FLAVIO CONDE

REDES DE MICROFONES EM TEMPO REAL: UMA IMPLEMENTAÇÃO COM HARDWARE DE BAIXO CUSTO E SOFTWARE DE CÓDIGO ABERTO

FLAVIO CONDE

REDES DE MICROFONES EM TEMPO REAL: UMA IMPLEMENTAÇÃO COM HARDWARE DE BAIXO CUSTO E SOFTWARE DE CÓDIGO ABERTO

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia

Área de Concentração: Engenharia Elétrica

Orientador: Prof. Dr. Phillip M. S. Burt

São Paulo

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.
São Paulo, 24 de março de 2010.
Assinatura do autor
Assinatura do orientador

FICHA CATALOGRÁFICA

Conde, Flavio

Redes de microfones em tempo real: uma implementação com hardware de baixo custo e software de código aberto / F. Conde. --- ed.rev. -- São Paulo, 2010.

123 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1. Processamento digital de sinais 2. Microfones 3. Software livre 4. Linux I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II. t.

AGRADECIMENTOS

Ao Professor e Orientador Phillip M. S. Burt pelas valorosas idéias e paciência no desenvolvimento deste trabalho.

Aos meus pais Oswaldo e Leni pela dedicação e incentivo.

À minha esposa e filho pelo carinho e compreensão.

A todos que contribuíram de forma direta ou indireta para a execução deste trabalho.

RESUMO

Este trabalho apresenta a implementação prática de uma rede de microfones para ser

utilizada em tempo real. A solução proposta envolve o uso de hardware de baixo

custo e software de código aberto. Em termos de hardware, a rede de microfones

utilizou dispositivos de áudio USB conectados diretamente a um computador pessoal

(PC). Em termos de software, foram utilizados a biblioteca de código aberto Advanced

Linux Sound Architecture (ALSA) e o sistema operacional Linux. Algumas

implementações foram realizadas na biblioteca ALSA para que fosse possível a

utilização da rede de microfones dentro do Linux. Os algoritmos implementados na

biblioteca ALSA foram o Delay and Sum, Generalized Sidelobe Canceller (GSC) e o

Post-Filtering.

Os aspectos teóricos dos principais algoritmos empregados nas redes de microfones

foram abordados de forma extensa. Os resultados teóricos e práticos desta

implementação são apresentados no final deste trabalho. Todo o desenvolvimento de

software foi publicado na Internet para que sirva de base para futuros trabalhos.

Palavras-chave: Processamento digital de sinais, microfones, software livre e Linux.

ABSTRACT

This work presents the practical implementation of a microphone array to be used in

real time. The proposed solution involves the use of low-cost hardware and open-

source software. In terms of hardware, the microphone array used USB audio devices

connected directly to a personal computer (PC). In terms of software, it was used the

open-source library Advanced Linux Sound Architecture (ALSA) and Linux operating

system. Some implementations were carried out in ALSA library to make it possible to

use the microphone array within Linux. The algorithms implemented in ALSA library

were the Delay and Sum, Generalized Sidelobe Canceller (GSC) and Post-Filtering.

The theory of the main algorithms used in microphone array were discussed

extensively. The results for the theoretical and practical implementation are presented

at the end of this work. All software development was published on the Internet to

serve as a basis for future works.

Keywords: Digital signal processing, microphones, free software and Linux.

LISTA DE ILUSTRAÇÕES

Figura 1.1 Resposta de um Beamforming típico	14
Figura 2.1 Sinal recebido por um sensor linear	20
Figura 2.2 Sensor linear contínuo de comprimento L	22
Figura 2.3 Função de resposta do padrão de diretividade de um sensor linear uniforme de	
comprimento L	23
Figura 2.4 Gráfico polar do padrão de diretividade	25
Figura 2.5 Rede de sensores discretos.	27
Figura 2.6 Padrão de diretividade variando-se o número de sensores $(f=1kHz, L=0.5m)$	28
Figura 2.7 Padrão de diretividade variando-se o comprimento da rede $(f=1kHz, N=5)$	28
Figura 2.8 Padrão de diretividade com $300Hz < f < 4000Hz$ ($N=5$, $d=0,1m$)	29
Figura 2.9 Chegada de ondas emitidas de longa distância (campo distante)	31
Figura 2.10 Chegada de ondas emitidas de curta distância (campo próximo)	33
Figura 2.11 Padrão de diretividade para fonte $near$ -field (campo próximo) $(r=1 m)$ e far -field	d
(campo distante) ($f=1 \text{ kHz}, N=9, d=0,1m$)	34
Figura 2.12 Função de coerência em ambiente de ruído difuso	36
Figura 3.1 Beamforming Delay and Sum.	38
Figura 3.2 Fontes de som e ruído visualizados com rede de microfones	40
Figura 3.3 Fontes de som e ruído com a rede de microfones	44
Figura 3.4 Delay and sum com sub-rede de 9 microfones e 3 faixas de frequência	47
Figura 3.5 Esquema do superdirective beamformer	48
Figura 3.6 Estrutura do Generalized Sidelobe Canceller (GSC): exemplo com 4 microfones	57
Figura 3.7 Diagrama de blocos do GSC LAF-LAF	61
Figura 3.8 Diagrama de blocos do GSC IIR: exemplo com 4 microfones	62
Figura 3.9 Esquema da estrutura de pós-filtragem	64
Figura 4.1 Dispositivo de áudio USB	74
Figura 4.2 Rede de microfones com 4 canais ligados a um <i>laptop</i> – vista de perto	75
Figura 4.3 Rede de microfones com 4 canais ligados a um <i>laptop</i> – vista lateral	75
Figura 4.4 Dispositivos de áudio mal conectados às portas USB	79
Figura 4.5 Dispositivos de áudio conectados perfeitamente às portas USB	79
Figura 5.1 Disposição das fontes sonoras utilizadas nos experimentos	81

Figura 5.2 Resultados do <i>Post-Filter</i> (SNR=3 dB, 8 microfones) - (a) Sinal desejado, (b) 1°
microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do Post-Filter (d)
Comparação entre as DEP's85
Figura 5.3 Resultados do <i>Post-Filter</i> (SNR=10 dB, 8 microfones) - (a) Sinal desejado, (b) 1°
microfone com sinal desejado + ruído (SNR=10 dB), (c) Saída do Post-Filter (d)
Comparação entre as DEP's86
Figura 5.4 Resultados do <i>Post-Filter</i> (SNR=3 dB, 4 microfones) - (a) Sinal desejado, (b) 1°
microfone com sinal desejado + ruído (SNR=3 dB, (c) Saída do Post-Filter (d)
Comparação entre as DEP's87
Figura 5.5 Resposta do <i>Post-Filter</i> em função do ângulo de incidência do sinal
Figura 5.6 Resultados do GSC LAF-LAF (SNR=3 dB, 8 microfones) - (a) Sinal desejado, (b) 19
microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do GSC (d) Comparação entre as DEP's
Figura 5.7 Resultados do GSC LAF-LAF (SNR=3 dB, 4 microfones) - (a) Sinal desejado, (b) 19
microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do GSC (d) Comparação entre as DEP's
Figura 5.8 Resposta do GSC LAF-LAF em função do ângulo de incidência do sinal
Figura 5.9 Resultados do GSC IIR (SNR=3 dB, 8 microfones) - (a) Sinal desejado, (b) 1°
microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do GSC IIR (d) Comparação
entre as DEP's94
Figura 5.10 Resultados do GSC IIR (SNR=3 dB, 4 microfones) - (a) Sinal desejado, (b) 1°
microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do GSC IIR (d) Comparação
entre as DEP's
Figura 5.11 Resposta do GSC IIR em função do ângulo de incidência do sinal96
Figura 5.12 Disposição das fontes sonoras utilizadas nos experimentos com o arranjo proposto
Figura 5.13 Resultados do <i>Post-Filter</i> (4 microfones) - (a) Sinal desejado, (b) 1º microfone com
sinal desejado + ruído, (c) Saída do <i>Post-Filter</i> (d) Comparação entre as DEP's100
Figura 5.14 Resposta do <i>Post-Filter</i> em função do ângulo de incidência do sinal101
Figura 5.15 Resultados do GSC LAF-LAF (4 microfones) - (a) Sinal desejado, (b) 1º microfone
com sinal desejado + ruído, (c) Saída do GSC (d) Comparação entre as DEP's103
Figura 5.16 Resposta do GSC LAF-LAF em função do ângulo de incidência do sinal104

Figura 5.17 Resultados do <i>Delay and Sum</i> (4 microfones) - (a) Sinal desejado, (b) 1° microfone
com sinal desejado + ruído, (c) Saída do Delay and Sum (d) Comparação entre as DEP's
106
Figura 5.18 Resposta do <i>Delay and Sum</i> em função do ângulo de incidência do sinal107

LISTA DE TABELAS

Tabela 4.1 Chamada dos algoritmos de rede de microfones no código-fonte	.72
Tabela 4.2: Lista dos dispositivos de áudio conectados no PC	.76
Tabela 4.3 Arquivo .asoundre para agrupamento dos dispositivos de áudio	.77

LISTA DE ABREVIATURAS E SIGLAS

ALSA Advanced Linux Sound Architecture

CLMS Constrained Least Mean Square

DEP Densidade Espectral de Potência

FFT Fast Fourier Transform

GJBF Griffths-Jim Beamformer

GSC Generalized Sidelobe Canceller

IFFT Inverse Fast Fourier Transform

IIR Infinite Impulse Response

LAF Leaky Adaptive Filters

LCMV Linear Constrained Minimum Variance

LMS Least Mean Square

LSI Large-Scale Integration

MVDR Minimum Variance Distortionless Response

NLMS Normalised Least Mean Square

PC Personal Computer

RLS Recursive Least Square

SNR Signal-to-Noise Ratio

USB Universal Serial Bus

SUMÁRIO

1	INTRODUÇÃO	12
1.1	MELHORIA DA QUALIDADE DOS SINAIS DE VOZ (SPEECH ENHANCEMEN	/T)
	12	
1.2	PRINCIPAIS ALGORITMOS UTILIZADOS NAS REDES DE MICROFONES	12
1.3	PROPOSTA DE TRABALHO	15
1.4	ORGANIZAÇÃO DO TEXTO	15
2	FUNDAMENTOS TEÓRICOS APLICADOS ÀS REDES DE MICROFONES.	17
2.1	PROPAGAÇÃO DE ONDAS	17
2.2	SENSOR CONTÍNUO	19
2.2.1	Função Matemática do Sensor	19
2.2.2	Padrão de Diretividade (Directivity Pattern)	19
2.2.3	Sensor Linear	21
2.3	REDE DE SENSORES DISCRETOS	25
2.3.1	Aliasing Espacial	29
2.4	FONTES DE SINAIS PRÓXIMAS ÀS REDES DE SENSORES (CAMPO PRÓXIMA	MO)
	30	
2.5	RUÍDOS	34
2.5.1	Ruídos Coerentes	35
2.5.2	Ruídos Incoerentes	35
2.5.3	Ruídos Difusos	35
3	PRINCIPAIS ALGORITMOS	37
3.1	DELAY AND SUM	37
3.1.1	Introdução	37
3.1.2	Tratamento Matemático	38
3.1.3	Rede de Sensores Aninhados (Nested Array)	45
3.2	SUPERDIRECTIVE BEAMFORMING	47
3.2.1	Introdução	47
3.2.2	Avaliação dos Beamformers	48
3.2.3	Ganho do Beamformer	49
3.2.4	Otimização do Ganho	51
3.2.5	Otimização do Ganho com Restrição	52

3.2.6	Ganho do Beamformer com Ruído Branco	52
3.2.7	Otimização do Ganho com Múltiplas Restrições Lineares	54
3.2.8	Considerações para o Campo Próximo	54
3.3	GENERALIZED SIDELOBE CANCELLER (GSC)	56
3.3.1	Introdução	56
3.3.2	Princípios do GSC	57
3.3.3	GSC Robusto	59
3.3.4	GSC com Filtros Adaptativos IIR	61
3.4	PÓS-FILTRAGEM (POST-FILTERING)	63
3.4.1	Introdução	63
3.4.2	Princípios da Pós-Filtragem	64
3.4.3	Pós-Filtro de Zelinsk	66
3.4.4	Pós-Filtro para Ruído Difuso	67
4	IMPLEMENTAÇÃO DA REDE DE MICROFONES	70
4.1	TRABALHANDO COM A BIBLIOTECA ALSA	70
4.2	HARDWARE UTILIZADO	73
4.3	LIMITAÇÕES DO SISTEMA PROPOSTO	78
5	EXPERIMENTOS E RESULTADOS	81
5.1	EXPERIMENTOS REALIZADOS NO MATLAB	81
5.1.1	Critérios de Avaliação	82
5.1.2	Análise dos Resultados do Post-Filtering	83
5.1.3	Análise dos Resultados do GSC LAF-LAF	88
5.1.4	Análise dos Resultados do GSC IIR	92
5.2	EXPERIMENTOS REALIZADOS NO SISTEMA PROPOSTO	96
5.2.1	Critérios de Avaliação	98
5.2.2	Análise dos Resultados do Post-Filtering	98
5.2.3	Análise dos Resultados do GSC LAF-LAF	101
5.2.4	Análise dos Resultados do Delay and Sum	104
6	CONCLUSÃO	108
REFE	ERÊNCIAS BIBLIOGRÁFICAS	110
ANEX	XO A – LISTAGEM DOS PROGRAMAS EM MATLAB	116
A.1	POST-FILTER	116
A.2	GSC LAF-LAF	118

A.3	GSC IIR	12	2(J
Λ .J	OSC IIIX	1 4	-'	ı

1 INTRODUÇÃO

1.1 MELHORIA DA QUALIDADE DOS SINAIS DE VOZ (SPEECH ENHANCEMENT)

A melhoria da qualidade dos sinais de voz, no ambiente das telecomunicações, vem sendo amplamente estudada nas últimas décadas [1] - [5]. Os sinais amostrados sofrem enormes degradações na presença de ruídos e reverberações. A filtragem destes sinais através de filtros de freqüência nem sempre é bem sucedida, uma vez que os sinais indesejados podem ocupar a mesma faixa do espectro de freqüência dos sinais de interesse. Neste caso, a rede de microfones pode ser utilizada com grande vantagem, pois ela se baseia nos métodos de filtragem espacial, que podem distinguir sinais oriundos de diferentes localidades físicas.

Em um ambiente típico de uma rede de microfones, o sinal desejado origina-se na voz de alguém, e é corrompido por sinais de interferência como vozes de outras pessoas, ruídos internos e externos ao local e reverberações do ambiente. A filtragem espacial pode ser útil neste caso, desde que as fontes das perturbações estejam em locais diferentes da origem do sinal desejado. Explorando a dimensão espacial do problema, a rede de microfones tenta melhorar a qualidade do sinal sem requerer que o locutor esteja próximo aos microfones.

1.2 PRINCIPAIS ALGORITMOS UTILIZADOS NAS REDES DE MICROFONES

Um dos mais simples e robustos métodos de filtragem espacial é o *Beamforming* [6], que utiliza a técnica de extrair o sinal desejado, contaminado por interferências, baseado na diretividade. Esta extração é executada processando-se os sinais obtidos nas redes de sensores. Estas redes podem ser de microfones, antenas, radares ou sonares localizados em diferentes posições no espaço.

O princípio de *Beamforming* é bem conhecido desde o início dos anos 60 [7] [8], sendo sua utilização bem difundida nos seguintes campos:

- Acústica: redes de microfones para a melhoria da qualidade dos sinais de voz [18]-[19], viva-voz [9], localização do locutor [13]-[17], reconhecimento de voz [11]-[12] e separação cega [20];
- Geologia: análise da crosta terrestre para prospecção de petróleo;
- Radar: controle de tráfego aéreo;
- Sonar: localização e classificação de fontes;
- Telecomunicações: transmissão e recepção direcional;
- Imagens médicas: ultra-sonografia e tomografia;
- Exploração astrofísica: imagens de alta resolução do universo;
- Biomedicina: monitoração de batimentos cardíacos de fetos, aparelhos auditivos [10] [26].

Devido à sua elevada complexidade computacional para as décadas de 60 e 70, o desenvolvimento de soluções de baixo custo não era viável. Desta maneira, as pesquisas do *Beamforming* foram direcionadas para a área de geologia com o uso de redes de sonares para a prospecção de petróleo, pois esta parecia ser a única área capaz de patrocinar este tipo de pesquisa. Com o advento da tecnologia LSI e o conseqüente aumento da capacidade computacional a baixos custos, iniciou-se um grande número de pesquisas para aplicações de baixo e médio custo desta técnica na área da acústica.

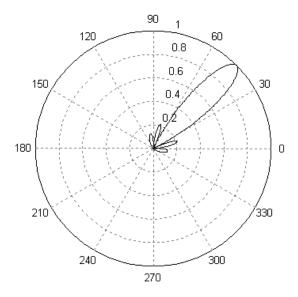


Figura 1.1 Resposta de um Beamforming típico

O termo *Beamforming* deriva do fato que os filtros espaciais eram projetados para que as plotagens polares de suas respostas tivessem o formato de *pencil beams*, conforme ilustra a Figura 1.1.

As técnicas de *Beamforming*, de maneira geral, podem ser classificadas como sendo:

- independentes dos dados: também chamados de Beamforming fixos, pois seus parâmetros não se modificam durante sua operação;
- dependentes dos dados: também chamados de Beamforming adaptativos, pois seus parâmetros se adaptam de acordo com os dados recebidos.

Dentro desta classificação, existe uma série de algoritmos empregados especificamente nas redes de microfones. Dentre eles, os principais a serem destacados são:

- Delay-and-Sum Beamforming: o mais simples de todos os algoritmos, discutido em detalhes na seção 3.1,
- Filter-and-Sum Beamforming,
- Superdirective Beamforming, detalhado na seção 3.2,
- Generalised Silelobe Canceller (GSC), detalhado na seção 3.3 e
- Pós Filtragem (Post-Filtering), detalhado na seção 3.4.

1.3 PROPOSTA DE TRABALHO

Grandes esforços vêm sendo empregados no aperfeiçoamento dos algoritmos das redes de microfones. No entanto, estas soluções ficam um pouco distantes da maioria do público porque necessitam de *hardware* específico e, na maioria das vezes, de valor elevado.

Este trabalho apresenta a implementação prática de uma rede de microfones para ser utilizada em tempo real [15]. A solução proposta envolve o uso de *hardware* de baixo custo e *software* de código aberto. O *hardware* utilizado consiste de dispositivos de áudio USB conectados diretamente a um computador pessoal (PC). Em termos de *software*, este trabalho propõe algumas implementações na biblioteca de código aberto *Advanced Linux Sound Architecture* (ALSA), que é a principal biblioteca de áudio do sistema operacional Linux. As implementações propostas na biblioteca ALSA foram realizadas no seu código-fonte, escrito em linguagem C, para possibilitar o uso dos principais algoritmos empregados nas redes de microfones, como o *Delay and Sum*, *Generalized Sidelobe Canceller* (GSC) e o *Post-Filtering*. As implementações realizadas na biblioteca ALSA foram documentadas e publicadas na Internet para que sirvam de base para futuros trabalhos.

Para a avaliação do sistema proposto, foram realizados ensaios em condições reais de uso da rede de microfones. Também foram feitas simulações numéricas no Matlab a título de comparação com os resultados práticos.

1.4 ORGANIZAÇÃO DO TEXTO

No capítulo 2, são apresentados os fundamentos teóricos das redes de microfones. O capítulo 3 descreve detalhadamente as características e o equacionamento dos principais algoritmos aplicados nas redes de microfones, quais sejam: *Delay-and-Sum, Superdirective Beamforming, Generalized Sidelobe Canceller* (GSC) e Pós-Filtragem (*Post-Filtering*). O capítulo 4 descreve a implementação prática em *hardware* e *software* da rede de microfones proposta, mostrando também as

limitações de sua implementação. O capítulo 5 descreve as análises teóricas e práticas da aplicação dos algoritmos propostos. Por fim, o capítulo 6 mostra as conclusões do método proposto e faz sugestões para a continuidade deste trabalho.

2 FUNDAMENTOS TEÓRICOS APLICADOS ÀS REDES DE MICROFONES

Uma rede de microfones é um caso particular de uma rede de sensores. No entanto, algumas características impedem que as redes de microfones utilizem os mesmos algoritmos desenvolvidos para as redes de sensores. As especificidades a serem destacadas são:

- O sinal de voz é considerado de banda larga (6 ou 7 oitavas), não estacionário, com elevadas taxas de variação de nível, além de apresentar períodos de silêncio;
- O espectro do sinal desejado pode ocupar a mesma faixa do espectro do sinal de interferência;
- A distância da fonte de sinal até os sensores em geral é pequena, fazendo com que não seja possível fazer as considerações de ondas planas.

No contexto deste trabalho, serão consideradas apenas as ondas acústicas geradas pela voz que serão recebidas por uma rede de microfones.

2.1 PROPAGAÇÃO DE ONDAS

Ondas sonoras se propagam através de fluidos como ondas longitudinais. No fluido, as moléculas se movem para frente e para trás na direção da propagação, produzindo regiões de compressão e expansão. Utilizando a equação de movimento de Newton para considerar um volume infinitesimal de fluido, pode-se desenvolver uma equação para a propagação de ondas. A equação geral para ondas acústicas é bastante complexa, pois leva em consideração as propriedades do fluido. Entretanto, se for considerado um fluido ideal com viscosidade zero, a equação da onda pode ser escrita como [21]

$$\nabla^2 p(t,\mathbf{r}) - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} p(t,\mathbf{r}) = 0, \qquad (2.1)$$

onde $p(t,\mathbf{r})$ é uma função que representa a pressão do ar no ponto \mathbf{r} do espaço e no instante t, sendo

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$
, x,y,z coordenadas cartesianas (2.2)

e ∇^2 o operador laplaciano. A velocidade de propagação c depende da pressão e da densidade do fluido, e é de aproximadamente *340 m/s* no ar.

A solução para a equação diferencial pode ser obtida utilizando-se o método da separação de variáveis. A solução para ondas planas monocromáticas é dada por [21]

$$p(t,\mathbf{r}) = Ae^{j(\omega t - k\mathbf{r})}, \qquad (2.3)$$

onde A é a amplitude da onda, $\omega=2\pi f$ é a freqüência em radianos por segundo e k é o vetor da onda, que indica a direção de sua propagação e é dado por:

$$k = \frac{2\pi}{\lambda} [sen\theta \cos \phi \quad sen\theta sen\phi \quad \cos \theta] , \qquad (2.4)$$

onde λ representa o comprimento de onda e está relacionado com c por $\lambda = c/f$. Alternativamente, a solução para ondas esféricas pode ser dada por

$$p(t,r) = -\frac{A}{4\pi r} e^{j(\omega t - kr)}, \qquad (2.5)$$

onde $r = |\mathbf{r}|$ é a distância radial da fonte e k é um escalar dado por $2\pi/\lambda$. A solução para ondas esféricas mostra que a atenuação da amplitude do sinal é proporcional à distância da fonte. Esta dependência da distância tem importantes implicações nos algoritmos das redes de microfones, particularmente para os casos onde as fontes de sinal estão situadas próximas aos microfones. Esta situação é chamada de *near-field* ou de campo próximo e será discutida em seções posteriores. Na natureza, as ondas sonoras são tipicamente esféricas. Porém, para distâncias suficientemente grandes de sua origem, uma onda pode ser considerada plana. Esta aproximação é bastante útil e é freqüentemente utilizada para simplificar a análise matemática do problema.

2.2 SENSOR CONTÍNUO

2.2.1 Função Matemática do Sensor

Considerando-se teoricamente um sensor contínuo de volume V onde o sinal $x(t,\mathbf{r})$ é recebido no instante t na posição espacial \mathbf{r} ,.com resposta impulsiva $a(t,\mathbf{r})$ para os sinais recebidos na posição \mathbf{r} , a resposta à saída do sensor será a convolução:

$$X_{R}(t,r) = \int x(\tau,r)a(t-\tau,r)d\tau$$
 (2.6)

ou, aplicando a transformada de Fourier

$$X_{R}(f,r) = X(f,r)A(f,r). \tag{2.7}$$

O termo $A(f,\mathbf{r})$ é conhecido como função de abertura ou função de sensibilidade, que define a resposta como uma função da posição espacial ao longo do sensor.

No contexto das redes de microfones, o termo abertura é utilizado para descrever a região do espaço capaz de receber a energia das ondas acústicas através de microfones e convertê-las em sinais elétricos.

2.2.2 Padrão de Diretividade (*Directivity Pattern*)

A resposta do sensor contínuo é notoriamente direcional devido à variação da quantidade de sinal e da fase com que atinge o sensor em função do seu ângulo de incidência.

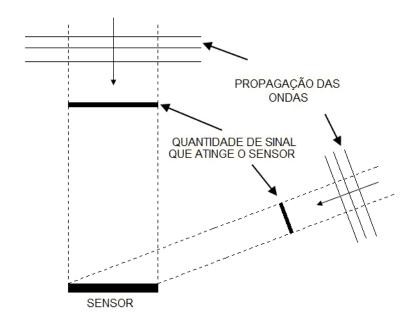


Figura 2.1 Sinal recebido por um sensor linear

Este princípio é ilustrado na Figura 2.1, onde ondas planas são recebidas por um sensor linear. A resposta do sensor como sendo uma função da freqüência e da direção do sinal que o atinge é conhecida como padrão de diretividade do sensor. Utilizando-se a equação da propagação de ondas, pode-se mostrar que o padrão de diretividade de um sensor, tendo A_R como função da abertura, é dado por [21]:

$$D_R(f,\alpha) = \Im_r \{ A_R(f,r) \}$$
 (2.8)

$$D_R(f,\alpha) = \int A_R(f,\mathbf{r})e^{j2\pi\alpha x} d\mathbf{r}$$
 (2.9)

onde:

 $\mathfrak{I}_{r}\{\cdot\}$: denota a transformada de Fourier tridimensional,

 $\mathbf{r} = [x_a \ y_a \ z_a]^T$: coordenadas espaciais no sensor,

 $\alpha = \frac{1}{\lambda}[sen\theta\cos\phi \ sen\thetasen\phi \ \cos\theta]$: direção do vetor da onda em coordenadas esféricas, conforme mostra a Figura 2.2. A dependência da freqüência está implícita, pois $\lambda = c/f$, onde λ é o comprimento de onda e c é a velocidade do som.

2.2.3 Sensor Linear

Para que sejam avaliadas algumas características importantes do padrão de diretividade, deve-se simplificar a equação anterior considerando o sensor como sendo linear de comprimento L posicionado sobre o eixo x, conforme a Figura 2.2 . Neste caso, o vetor \mathbf{r} pode ser reduzido para $\mathbf{r} = [x_a \ 0 \ 0]^T$ e a nova função do padrão de diretividade será

$$D_{R}(f,\alpha_{x}) = \int_{-L/2}^{L/2} A_{R}(f,x_{a}) e^{j2\pi\alpha_{x}x_{a}} dx_{a}$$
 (2.10)

$$\alpha_{x} = \frac{\sin\theta\cos\phi}{\lambda}.$$
 (2.11)

Se esta equação for escrita em função dos ângulos:

$$D_R(f,\theta,\phi) = \int_{-L/2}^{L/2} A_R(f,x_a) e^{j\frac{2\pi}{\lambda} \operatorname{sen}\theta \cos\phi x_a} dx_a. \qquad (2.12)$$

É importante lembrar que as expressões anteriores são válidas para o caso em que a origem do sinal está relativamente distante do sensor, e de forma que seja possível admitir que as ondas sejam planas. Esta situação é chamada de *far-field* ou campo distante, e é obtida quando a distância da origem do sinal em relação ao sensor satisfizer a seguinte relação dada por [22]:

$$\left|\mathbf{r}\right| > \frac{2\mathsf{L}^2}{\lambda} \,. \tag{2.13}$$

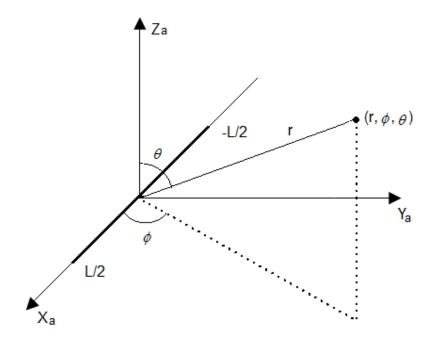


Figura 2.2 Sensor linear contínuo de comprimento L

Considerando-se o sensor linear com resposta uniforme e independente da freqüência, sua função de resposta pode ser escrita como

$$A_{R}(x_{a}) = \text{rect}(x_{a}/L), \qquad (2.14)$$

onde

$$rect(x/L) = \begin{cases} 1 \text{ para } |x| \le L/2 \\ 0 \text{ para } |x| > L/2 \end{cases}$$
 (2.15)

A função do padrão de diretividade resultante será:

$$D_{R}(f,\alpha_{x}) = \Im\{\operatorname{rect}(x_{a}/L)\}\tag{2.16}$$

$$= L \operatorname{sinc}(\alpha_{x}L), \qquad (2.17)$$

onde

$$\operatorname{sinc}(x) = \frac{\operatorname{sen}(x)}{x} \,. \tag{2.18}$$

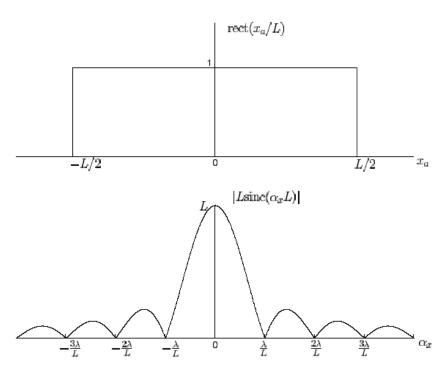


Figura 2.3 Função de resposta do padrão de diretividade de um sensor linear uniforme de comprimento L

A Figura 2.3 mostra que os zeros da função do padrão de diretividade estão posicionados em $\alpha_x = m\lambda/L$, onde m é um inteiro. A área da função do padrão de diretividade compreendida entre $-\lambda/L$ e λ/L é conhecida como lóbulo principal, e sua extensão é denominada largura do feixe. Nota-se que a largura do feixe de um sensor linear é dada por $2\lambda/L$, ou 2c/fL. Esta relação revela um problema na utilização deste tipo de comprimento fixo. À medida que a freqüência aumenta, a largura do feixe diminui. Com o objetivo de facilitar as comparações da resposta do padrão de diretividade, freqüentemente tomamos o seu valor na forma normalizada. A faixa de valores possíveis da função **sinc** varia entre -1 e 1, portanto o valor máximo do padrão de diretividade será $D_{\rm max} = L$; assim, teremos

$$D_N(f,\alpha_x) = \frac{D_R(f,\alpha_x)}{D_{\text{max}}} = \text{sinc}(\alpha_x L)$$
 (2.19)

Ou, em termos dos ângulos ϕ e θ

$$D_N(f,\theta,\phi) = \operatorname{sinc}\left(\frac{L}{\lambda}\operatorname{sen}\theta\cos\phi\right).$$
 (2.20)

O gráfico polar é uma ferramenta bastante utilizada para análise do padrão de diretividade. Para melhor visualização, a análise do comportamento da diretividade será feita apenas sobre o ângulo ϕ (fixando θ em 90°), de forma que

$$D_N\left(f, \frac{\pi}{2}, \phi\right) = \operatorname{sinc}\left(\frac{L}{\lambda}\cos\phi\right).$$
 (2.21)

Os gráficos polares são exibidos na Figura 2.4 para diferentes valores de L/λ , mostrando que a largura do feixe depende desta razão conforme discutido anteriormente. Analisando as respostas destes gráficos com auxílio da equação (2.21), verificamos que os maiores ganhos estão sempre em $\phi = \pi/2$ (90°) e $\phi = 3\pi/2$ (270°), o que corresponde à fonte de sinal situada perpendicularmente ao sensor. Esta posição da fonte é denominada *broadside*. Por outro lado, os menores ganhos são obtidos quando $\phi = 0$ (0°) ou $\phi = \pi$ (180°), o que corresponde à fonte de sinal situada na mesma linha do sensor. Esta posição da fonte é denominada *endfire*.

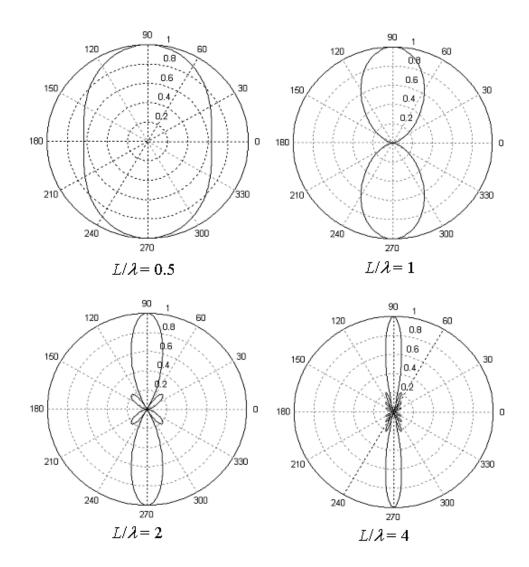


Figura 2.4 Gráfico polar do padrão de diretividade

2.3 REDE DE SENSORES DISCRETOS

Uma rede de sensores discretos pode ser considerada uma versão amostrada de um sensor contínuo. Cada sensor pode ser considerado como um pequeno sensor contínuo e assim a resposta deste modelo será a superposição das respostas de cada um dos sensores.

Considerando-se a rede de sensores mostrada na Figura 2.5, onde cada elemento possui a resposta em freqüência $r_n(f,x)$ e utilizando-se o princípio da superposição, a resposta da rede de sensores pode ser escrita como

$$A(f, X_a) = \sum_{n = -\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) r_n(f, X_a - X_n), \qquad (2.22)$$

onde $w_n(f)$ é o vetor complexo de pesos para o sensor n e x_n e x_a são suas posições no eixo x. Se substituirmos esta função discreta de abertura em (2.10), teremos o padrão de diretividade, definido para o campo distante, dado por

$$D(f,\alpha_{x}) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_{n}(f) R_{n}(f,\alpha_{x}) e^{j2\pi\alpha_{x}x_{n}}, \qquad (2.23)$$

onde $R_n(f,\alpha_x)$ é o padrão de diretividade do sensor n.

No caso de todos os sensores possuírem a mesma resposta em freqüência, isto é, $R_n(f,\alpha_x) = R(f,\alpha_x) \ \forall n$, a função de abertura pode ser simplificada para

$$A(f, x_a) = \sum_{n = -\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) \delta(x_a - x_n), \qquad (2.24)$$

e a nova função do padrão de diretividade será

$$D(f,\alpha_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j2\pi\alpha_x x_n} . {(2.25)}$$

Para o caso onde os sensores contíguos estiverem igualmente espaçados de uma distância *d* entre sensores, o padrão de diretividade será

$$D(f,\alpha_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j2\pi\alpha_x nd} . {(2.26)}$$

Considerando-se o padrão de diretividade apenas na horizontal, teremos

$$D(f,\phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j\frac{2\pi}{\lambda} nd\cos\phi}$$
 (2.27)

tornando explícita a dependência da fregüência

$$D(f,\phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j\frac{2\pi f}{c} nd\cos\phi}$$
 (2.28)

A equação (2.28) é válida para uma rede de sensores lineares, idênticos e pontuais equi-espaçados. Fica claro nesta equação que o padrão de diretividade depende:

- do número N de sensores da rede,
- do espaçamento d entre sensores e
- da freqüência f do sinal recebido.

Para análise da interdependência destas variáveis, a seguir serão plotadas as respostas do padrão de diretividade para três diferentes situações:

- variando-se o número de elementos N com L=Nd e f fixos;
- variando-se o comprimento da rede L=Nd, com N e f fixos;
- e variando-se a freqüência f, com N e L fixos.

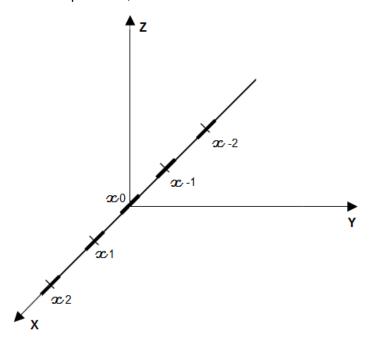


Figura 2.5 Rede de sensores discretos

No gráfico da Figura 2.6, pode-se verificar que à medida que se aumenta o número de sensores, ocorre uma diminuição não linear no nível dos lóbulos laterais, tendendo a

um ponto de saturação a partir do qual o aumento do número de sensores não produz redução no nível dos lóbulos laterais.

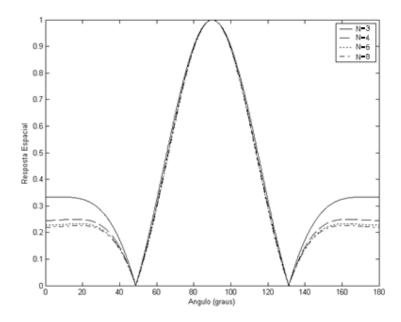


Figura 2.6 Padrão de diretividade variando-se o número de sensores (*f*=1*kHz*, *L*=0,5*m*)

Na segunda situação, o gráfico da Figura 2.7 mostra que a largura do feixe diminui à medida que se aumenta o comprimento da rede (aumentando o espaçamento entre sensores).

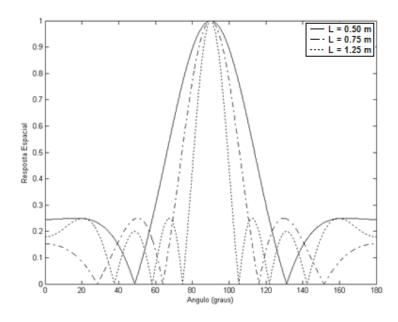


Figura 2.7 Padrão de diretividade variando-se o comprimento da rede (*f*=1*kHz*, *N*=5)

Na terceira situação, o gráfico da Figura 2.8 mostra que a largura do feixe diminui à medida que se aumenta a freqüência.

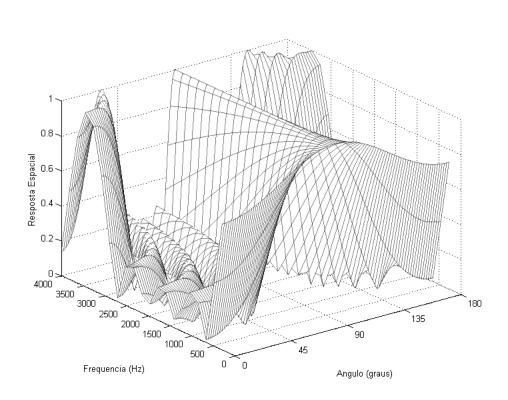


Figura 2.8 Padrão de diretividade com 300Hz < f < 4000Hz (N=5, d=0, 1m)

2.3.1 Aliasing Espacial

Em amostragens temporais, a freqüência de Nyquist é conhecida como sendo a freqüência mínima de amostragem para se evitar o efeito de *aliasing*, que impossibilita que se reconstrua o sinal original a partir do sinal amostrado devido à perda de informação [53]. Em essência, uma rede de sensores implementa uma amostragem espacial do sinal, e existe uma condição análoga para evitar a perda de informação do sinal original. Na amostragem temporal, o teorema de Nyquist diz que a freqüência de amostragem f_s com período T_s deve ser:

$$f_s = \frac{1}{T_s} > 2f_{\text{max}} \tag{2.29}$$

onde f_{max} é a maior frequência presente no sinal amostrado. Similarmente, para a amostragem espacial, existe a necessidade de que

$$f_{x_s} = \frac{1}{d} > 2f_{x_{\text{max}}} \,, \tag{2.30}$$

onde f_{x_s} é a freqüência de amostragem espacial em amostragens por metro, $f_{x_{\max}}$ é a maior freqüência espacial no espectro angular do sinal e d é o espaçamento entre sensores. A freqüência espacial ao longo do eixo x é dada por

$$f_{x} = \frac{\sin\theta\cos\phi}{\lambda}.$$
 (2.31)

O máximo valor desta razão ocorrerá quando o numerador for máximo e o denominador for mínimo, que produz a relação

$$f_{\chi_{\text{max}}} = \frac{1}{\lambda_{\text{min}}} \tag{2.32}$$

e, conseqüentemente, a necessidade que

$$d < \frac{\lambda_{\min}}{2} = \frac{c}{2f_{\max}} , \qquad (2.33)$$

onde λ_{\min} é o menor comprimento de onda do sinal de interesse. Em outras palavras, o espaçamento entre sensores deve ser menor que a metade do menor comprimento de onda do sinal amostrado para evitar o *aliasing* espacial no padrão de diretividade de uma rede de sensores. A equação (2.33) é conhecida como o *teorema da amostragem espacial*.

2.4 FONTES DE SINAIS PRÓXIMAS ÀS REDES DE SENSORES (CAMPO PRÓXIMO)

Nas análises anteriores, foram considerados apenas os casos em que a origem do sinal está a uma distância relativamente grande dos sensores (campo distante), fazendo com que as ondas possam ser consideras planas. Conforme definido em [22], para que esta condição seja verdadeira, deve-se ter a fonte do sinal distante de

$$\left|r\right| > \frac{2L^2}{\lambda} \,. \tag{2.34}$$

Em caso contrário, a curvatura das ondas passa a ter efeito importante, não podendo mais ser ignorada. Para facilitar a análise, será considerado um padrão de diretividade horizontal em que ondas planas atingem uma rede de sensores como mostrado no esquema da Figura 2.9. Neste esquema, nota-se que a distância percorrida por uma onda entre dois sensores adjacentes será

$$d' = d\cos\phi. \tag{2.35}$$

De maneira geral, a distância percorrida entre todos os sensores será

$$d' = nd\cos\phi. \tag{2.36}$$

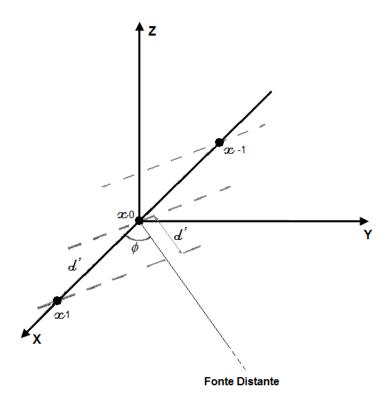


Figura 2.9 Chegada de ondas emitidas de longa distância (campo distante)

Já no esquema da Figura 2.10, ondas esféricas atingem a rede de sensores. Por este esquema, nota-se que a distância percorrida entre dois sensores adjacentes será

$$d' = d_1(r,\phi) - d_0(r,\phi). \tag{2.37}$$

De maneira geral,

$$d' = d_n(r, \phi) - d_0(r, \phi), \tag{2.38}$$

onde $d_n(r,\phi)$ é a distância da fonte até o n-ésimo sensor. Considerando apenas o plano horizontal e utilizando-se a lei dos cossenos numa rede em que os sensores estão equi-espaçados, pode ser mostrado que esta distância é dada por

$$d_n(r,\phi) = \sqrt{r_0^2 - 2r_0 nd\cos\phi_0 + (nd)^2},$$
 (2.39)

onde r_0 e ϕ_0 são considerados em relação ao sensor base.

Se observarmos a equação (2.27), do padrão de diretividade, que considera a caso da fonte a uma longa distância (campo distante), veremos que a exponencial possui o termo $nd\cos\phi$, que pode ser considerado apenas para o caso de ondas planas. Substituindo este termo pela expressão apropriada para o caso da fonte próxima à rede de sensores (campo próximo), teremos

$$D'(f,\phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j\frac{2\pi}{\lambda}(d_n(r,\phi) - d_0(r,\phi))}.$$
 (2.40)

Deve-se lembrar ainda que a amplitude de uma onda acústica diminui proporcionalmente com a distância percorrida. Para o caso de campo distante, a diferença relativa das amplitudes da onda que atinge cada sensor é desprezível; no entanto, estas diferenças de amplitude podem ser significativas para o caso da fonte próxima aos sensores. Incorporando na expressão do padrão de diretividade a dependência normalizada da amplitude, teremos

$$D_{nf}(f,\phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{d_0(r,\phi)}{d_n(r,\phi)} w_n(f) e^{j\frac{2\pi}{\lambda}(d_n(r,\phi)-d_0(r,\phi))}.$$
 (2.41)

Se a rede de sensores estiver sendo projetada para operar com fontes de sinal próximas aos sensores, deverá ser utilizada uma compensação nos pesos $w_n(t)$ de forma a se considerar o padrão de diretividade como sendo o mesmo do caso da fonte distante. Este procedimento é chamado de compensação para campo próximo. Os valores dos novos pesos serão dados por

$$W'_{n}(f) = \frac{d_{n}(r,\phi)}{d_{n}(r,\phi)}W_{n}(f)e^{j\frac{2\pi}{\lambda}(d_{n}(r,\phi)-d_{n}(r,\phi)+nd\cos\phi)}.$$
 (2.42)

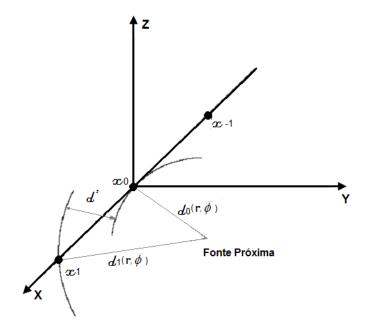


Figura 2.10 Chegada de ondas emitidas de curta distância (campo próximo)

O gráfico da Figura 2.11 ilustra a interferência da proximidade da fonte de sinais da rede de sensores, plotando o padrão de diretividade para ambos os casos: campo próximo e campo distante. Pode-se verificar que no caso do campo próximo, há uma sensível degradação na seletividade espacial, uma vez que a razão entre os máximos do lóbulo principal e dos lóbulos laterais do padrão de diretividade é bastante reduzida.

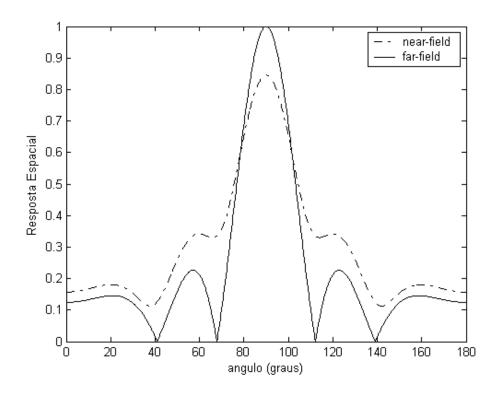


Figura 2.11 Padrão de diretividade para fonte *near-field* (campo próximo)(*r=1 m*) e *far-field* (campo distante) (*f=1 kHz, N=9, d=0,1m*)

2.5 RUÍDOS

Existem três categorias principais de ruídos a serem consideradas na análise das redes de microfones. Estas categorias são caracterizadas de acordo com o grau de correlação do sinal de ruído nas diferentes posições do espaço. Uma medida comum para avaliar essa correlação é a função espacial de coerência, que é definida em [45] como

$$\Gamma_{ij}(f) = \frac{\phi_{ij}(f)}{\sqrt{\phi_{ii}(f)\phi_{ij}(f)}},$$
(2.43)

onde ϕ_{ij} é a densidade espectral cruzada ente os sinais i e j. Os valores da função de coerência variam de forma que $0 \le \left|\Gamma_{ij}(f)\right| \le 1$.

2.5.1 Ruídos Coerentes

Em um ambiente com ruído coerente, os sinais de ruído captados em diferentes microfones de uma rede são altamente correlacionados, e portanto $\left| \varGamma_{ij}(f) \right| \approx 1$. Na prática, entre outras situações, ruídos coerentes podem ocorrer em ambientes a céu aberto onde não existam muitos obstáculos à propagação do som (desde que as condições de vento sejam favoráveis).

2.5.2 Ruídos Incoerentes

Em ambientes em que ocorrem ruídos incoerentes, os sinais de ruído captados em dois locais distintos são altamente não correlacionados, isto é $\left|\Gamma_{ij}(f)\right|\approx 0$, para $i\neq j$. Na prática este tipo de ruído é raramente encontrado. Ruídos incoerentes são também chamados de ruídos espacialmente brancos.

2.5.3 Ruídos Difusos

Em um ambiente de ruído difuso, a energia do ruído se propaga de forma igual em todas as direções, simultaneamente. Assim os sinais de ruído recebidos pelos microfones são pouco correlacionados e possuem aproximadamente a mesma energia. Na prática, muitos ambientes podem ser caracterizados pelo ruído difuso, como o interior de automóveis e salas de escritórios. A função de coerência do ruído difuso entre dois pontos quaisquer pode ser modelada como uma função da distância entre os dois pontos de medição [45]

$$\Gamma_{ij}(f) = \operatorname{sinc}\left(\frac{2\pi f d_{ij}}{c}\right),\tag{2.44}$$

onde d_{ij} é a distância entre os sensores i e j.

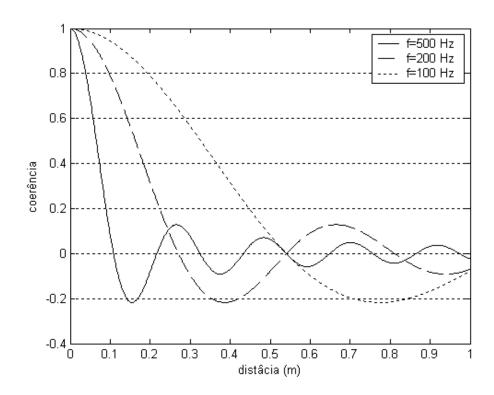


Figura 2.12 Função de coerência em ambiente de ruído difuso

Pode-se verificar que a coerência se aproxima de seu valor máximo (unitário) quanto menor for o espaçamento entre microfones. Outra característica importante ocorre em função da freqüência do sinal do ruído. Para freqüências menores, a coerência tende a ser maior se considerarmos o mesmo espaçamento entre microfones.

3 PRINCIPAIS ALGORITMOS

3.1 DELAY AND SUM

3.1.1 Introdução

A técnica mais simples de *beamforming* é o *Delay-and-Sum*. Seu funcionamento baseia-se no fato de que sinais situados em diferentes posições do espaço atingem os microfones da rede com diferentes conjuntos de atrasos. Considerando-se o espaçamento entre sensores fixo, cada conjunto de atrasos dependerá apenas do ângulo de incidência do sinal na rede. Conhecendo-se o ângulo do sinal desejado, pode-se aplicar a correção do atraso no sinal de cada microfone, fazendo com que se tenha o alinhamento no tempo do sinal desejado e simultaneamente o desalinhamento dos outros sinais. A soma das contribuições de todos os microfones com seus devidos atrasos incluídos fará com que todos os sinais desalinhados no tempo sejam atenuados, e desta forma, o sinal desejado se sobressaia.

O desempenho deste tipo de *beamforming* está rigidamente ligado ao número de microfones que compõem a rede, e à forma que estes microfones estão espaçados.

A grande desvantagem do *beamforming Delay-and-Sum* é o grande número de microfones necessários para se obter uma boa razão sinal-ruído. Outra desvantagem está no grande tamanho da rede, pois para que o sistema seja sensível às baixas freqüências, é necessário que o espaçamento entre sensores seja grande, o que exige que a rede tenha alguns metros de comprimento.

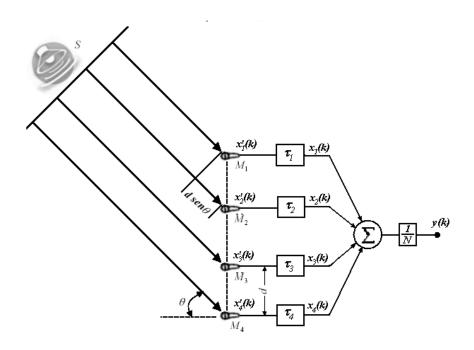


Figura 3.1 Beamforming Delay and Sum

3.1.2 Tratamento Matemático

A seguir será mostrado o tratamento matemático desenvolvido em [23] para uma rede de microfones na presença de uma fonte de som e apenas uma fonte de ruído.

Considerando a fonte de sinal s localizada nas coordenadas espaciais (x_s, y_s, z_s) , a fonte de ruído v localizada em (x_v, y_v, z_v) , N microfones $M_1, M_2, ..., M_N$ nas respectivas coordenadas (x_{M1}, y_{M1}, z_{M1}) , (x_{M2}, y_{M2}, z_{M2}) ,..., (x_{MN}, y_{MN}, z_{MN}) , como mostrado na Figura 3.2. O som emitido por s é definido como s(t), O som $M_j(t)$ recebido pelo microfone j que foi emitido pela fonte s, pode ser escrito como

$$M'_{j}(t) = K'_{j} s(t - \tau'_{j}),$$
 (3.1)

onde τ_{j} é o tempo de propagação do som entre a fonte de sinal e o microfone M_{j} , e pode ser escrito como

$$\tau_{j}' = \frac{\sqrt{(X_{Mj} - X_{s})^{2} + (Y_{Mj} - Y_{s})^{2} + (Z_{Mj} - Z_{s})^{2}}}{c},$$
(3.2)

sendo c a velocidade do som.

Da mesma forma, o sinal que é emitido pela fonte de ruído é definido como v(t). O som $M_j^r(t)$ recebido pelo microfone j emitido pela fonte de ruído v pode ser escrito como

$$M_{i}^{"}(t) = K_{i}^{"} v(t - \tau_{i}^{"}),$$
 (3.3)

onde $au_j^{"}$ é o tempo de propagação do som entre a fonte de ruído e o microfone $extit{M}_{j,}$ e pode ser escrito como

$$\tau_{j}^{"} = \frac{\sqrt{(X_{Mj} - X_{v})^{2} + (Y_{Mj} - Y_{v})^{2} + (Z_{Mj} - Z_{v})^{2}}}{C},$$
(3.4)

sendo c a velocidade do som.

Nas equações acima, $K_j^{'}$ e $K_j^{''}$ são constantes que representam as atenuações do sinal s(t) e do ruído v(t), respectivamente. Pode-se considerar que

$$K_j \alpha \frac{1}{\tau_j}$$
 (3.5)

е

$$\mathcal{K}_{j}^{"} \alpha \frac{1}{\tau_{j}^{"}}.$$
 (3.6)

Isto significa que a pressão do som no microfone é inversamente proporcional à distância da fonte.

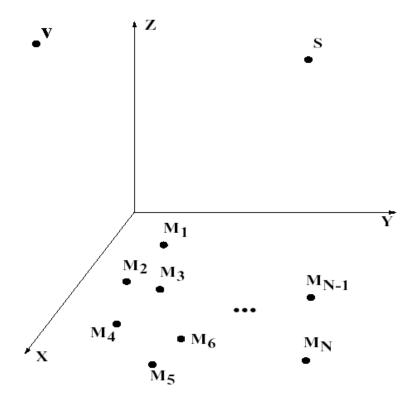


Figura 3.2 Fontes de som e ruído visualizados com rede de microfones

A rede de microfones proporcionará o máximo ganho no sinal proveniente da fonte s se configurarmos a linha de atraso d_i para cada microfone M_i de forma que

$$d_j + \tau_j' = C \tag{3.7}$$

para todo j, onde C é uma constante tal que

$$C \ge \max_{j \in 1..N} [\tau_j^{'}]. \tag{3.8}$$

Há um valor mínimo de C necessário para preservar a causalidade. É desejável que C seja o menor valor possível para evitar o excesso de atraso entre a fonte de sinal s(t) e o sinal recuperado pela rede de microfones. Este sinal recuperado será o somatório de todos os sinais dos microfones $M_j(t)$ com seus devidos atrasos:

$$M_R(t) = \sum_{j=1}^{N} M_j(t)$$
, (3.9)

onde

$$M_i(t) = M_i(t - d_i) + M_i(t - d_i).$$
 (3.10)

Expandindo a equação (3.9):

$$M_{R}(t) = \sum_{j=1}^{N} \left[K_{j}' s(t - \tau_{j}' - d_{j}) + K_{j}'' v(t - \tau_{j}'' - d_{j}) \right]$$
(3.11)

$$=\sum_{j=1}^{N} \left[\underbrace{K_{j}'s(t-C)}_{\sin al} + \underbrace{K_{j}''v(t-\tau_{j}''-d_{j})}_{\text{ruido}} \right]. \tag{3.12}$$

A razão sinal/ruído (SNR) na saída da rede de microfones é dada por

$$SNR = 10 \log \left[\frac{E\left\{ \left[\sum_{j=1}^{N} K_{j}^{"} s(t-C) \right]^{2} \right\}}{E\left\{ \left[\sum_{j=1}^{N} K_{j}^{"} v(t-\tau_{j}^{"}-d_{j}) \right]^{2} \right\}} \right].$$
(3.13)

Analisando o comportamento desta expressão ao incrementarmos o número *N* de microfones, podem-se fazer algumas considerações para sua simplificação.

Inicialmente deve-se assumir que a condição do campo distante seja válida:

$$\max_{j \in 1...N, k \in 1...N} |\tau_{j} - \tau_{k}| << \min_{j \in 1...N} \tau_{j}^{'},$$
(3.14)

ou seja, que a distância entre qualquer par de microfones é relativamente pequena se compararmos com a distância entre a fonte de som e o microfone mais próximo. A mesma consideração deve ser feita com relação à fonte de ruído:

$$\max_{j \in 1...N} \left| \tau_{j}^{"} - \tau_{k}^{"} \right| << \min_{j \in 1...N} \tau_{j}^{"}. \tag{3.15}$$

As considerações das equações (3.14) e (3.15) nos permitem fazer as seguintes substituições:

$$\tau_i \approx \overline{\tau}$$
 (3.16)

$$\tau_{i}^{"} \approx \overline{\tau_{i}}^{"}.$$
(3.17)

Substituindo as equações (3.16) e (3.17) em (3.5) e (3.6),

$$K_{i} = \overline{K}$$
 (3.18)

$$K_{j}^{"} = \overline{K^{"}}. \tag{3.19}$$

Reescrevendo a equação (3.13),

$$SNR = 10 \log \left[\frac{\overline{K}^{2} E\left\{ \left[\sum_{j=1}^{N} s(t-C)^{2} \right\} \right\}}{\overline{K}^{2} E\left\{ \left[\sum_{j=1}^{N} v(t-\tau_{j}^{2}-d_{j})^{2} \right\} \right]}$$
(3.20)

$$=10\log\left[\frac{\overline{K'}^{2}}{\overline{K''}^{2}}\right]+10\log\left[\frac{N^{2}E[s(t-C)^{2}]}{\sum_{j=1}^{N}\sum_{j=1}^{N}E[v(t-\tau_{i}''-d_{j})v(t-\tau_{j}''-d_{j})]}\right].$$
(3.21)

Assumindo também que o sinal e o ruído são estacionários no sentido amplo, com funções de autocorrelação $R_s(\tau)$ e $R_v(\tau)$, respectivamente, pode-se reduzir a equação (3.21) para

$$SNR = 10 \log \left[\frac{\overline{K}^2}{\overline{K}^2} \right] + 10 \log \left[\frac{N^2 R_s(0)}{\sum_{l=1}^{N} \sum_{j=1}^{N} R_v(\tau_l^{"} + d_l - \tau_j^{"} - d_j)} \right].$$
(3.22)

Avaliando-se o comportamento da equação (3.22) com a variação de seus parâmetros, observa-se apenas o segundo termo da equação é afetado com a variação do número de microfones *N*. O comportamento deste termo também depende das características espectrais do ruído. Para análise desta dependência, podem-se considerar dois casos extremos:

1. O ruído v(t) é quase um ruído branco e, portanto sua função de autocorrelação é impulsiva. Isto implica que

$$\left| R_{\nu}(\tau) \right| \approx 0 \quad \text{para} \quad \left| \tau \right| > \min_{\substack{j=1...N,\\j=1...N,\\i\neq j}} \left| \tau_{i}^{"} + d_{i} - \tau_{j}^{"} - d_{j} \right|. \tag{3.23}$$

O denominador do segundo termo da equação (3.22) pode ser reduzido em $\sum_{i=1}^{N} R_{\nu}(0)$, e a equação completa pode ser reescrita como

$$SNR = 10 \log \left[\frac{\overline{K'}^{2}}{\overline{K''}^{2}} \right] + 10 \log \left[\frac{N^{2}R_{s}(0)}{NR_{v}(0)} \right]$$
(3.24)

$$= 10 \log \left[\frac{\overline{K'}^{2}}{\overline{K''}^{2}} \right] + 10 \log \left[\frac{N R_{s}(0)}{R_{v}(0)} \right].$$
 (3.25)

O resultado da equação (3.25) indica que se dobrarmos o número de microfones, teremos um ganho de 3dB na razão sinal/ruído.

2. O espectro do ruído se concentra nas freqüências mais baixas, de modo que não haja variação perceptível na função de autocorrelação durante a propagação do ruído sobre a rede. Neste caso tem-se que

$$R_{\nu}(\tau) \approx R_{\nu}(0) \text{ para } \left|\tau\right| < \max_{\substack{j=1...N,\\j=1...N}} \left|\tau_{i}^{"} + d_{i} - \tau_{j}^{"} - d_{j}\right|.$$
 (3.26)

O denominador do segundo termo da equação (3.22) pode ser reduzido a $\sum_{i=1}^{N} \sum_{i=1}^{N} R_{\nu}(0)$, e a equação completa pode ser escrita como

$$SNR = 10 \log \left[\frac{\overline{K}^2}{\overline{K}^2} \right] + 10 \log \left[\frac{N^2 R_s(0)}{N^2 R_v(0)} \right]$$
 (3.27)

$$=10\log\left[\frac{\overline{K'}^{2}}{\overline{K''}^{2}}\right]+10\log\left[\frac{R_{s}(0)}{R_{v}(0)}\right]. \tag{3.28}$$

O resultado da equação (3.28) indica que a razão sinal/ruído é completamente independente do número de microfones, portanto o aumento do número de microfones não implicará nenhuma melhora de desempenho.

Os resultados acima indicam que a rede de microfones pode oferecer um ganho efetivo de desempenho sobre um único microfone quando as condições da equação (3.23) forem satisfeitas. Porém sua utilização não terá muita utilidade se as condições se aproximarem das estabelecidas na equação (3.26). Estas condições dependem principalmente:

- 1. da posição relativa entre a fonte do sinal desejado e a fonte do ruído;
- 2. do espaçamento entre os microfones;
- 3. da função de autocorrelação da fonte de ruído v(t).

A análise deve ser concentrada na condição 2, que trata do espaçamento entre os microfones, já que as demais condições não dependem da vontade do projetista da rede de microfones.

Ao variar o espaçamento entre os microfones, o objetivo é minimizar o duplo somatório do denominador do segundo termo da equação (3.22). Inicialmente podese examinar a expressão $\tau_i^{"} + d_i - \tau_j^{"} - d_j$ para o par de microfones M_j e M_l e para a fonte de sinal s e de ruído v dispostos conforme mostrado na Figura 3.3. Os microfones M_j e M_l são separados pela distancia D.

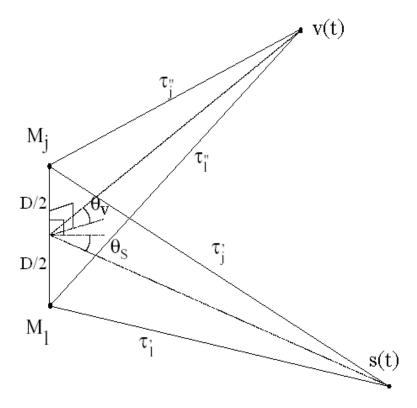


Figura 3.3 Fontes de som e ruído com a rede de microfones

Se a distância do par de microfones até a fonte de som s e a distância do par de microfones até a fonte de ruído v forem bem maiores que a distância D entre os microfones, então podem-se usar as seguintes aproximações:

$$\tau_i - \tau_j \approx \frac{D}{c} \operatorname{sen}(\theta_s)$$
(3.29)

$$\tau_{i}^{"} - \tau_{j}^{"} \approx \frac{D}{c} sen(\theta_{v}).$$
(3.30)

A linha dos atrasos da rede faz $d_j = C - \tau_j$ e $d_l = C - \tau_l$, então se pode produzir a seguinte igualdade:

$$\left|\tau_{i}^{"} + d_{i} - \tau_{j}^{"} - d_{j}\right| = \left|\tau_{i}^{"} - \tau_{j}^{"} + \tau_{j}^{'} - \tau_{i}^{'}\right| \tag{3.31}$$

$$= \frac{D}{c} |sen(\theta_s) - sen(\theta_v)|. \tag{3.32}$$

Se aumentarmos a distância D entre os microfones, estaremos diminuindo a autocorrelação do sinal da fonte de ruído e, portanto melhorando a razão sinal/ruído segundo a equação (3.22). A diferença entre os ângulos θ_s (formado pelo par de microfones e a fonte de som) e θ_v (formado pelo par de microfones e a fonte de ruído) também afeta a magnitude da expressão analisada. Se θ_s e θ_v forem iguais, a equação (3.32) se torna zero. Esta conclusão já era esperara, uma vez que a rede de microfones não é capaz de atribuir ganhos diferentes a dois sinais que venham da mesma direção.

3.1.3 Rede de Sensores Aninhados (Nested Array)

A equação do padrão de diretividade de uma rede de sensores uniformemente espaçados foi definida na seção 2.3 e é mostrada abaixo.

$$D(f,\phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j\frac{2\pi f}{c} nd\cos\phi}.$$
 (3.33)

Nesta equação, pode-se verificar que a resposta do beamformer depende da freqüência do sinal de interesse, do espaçamento e do número de sensores. Desta forma, pode-se facilmente concluir que a resposta do beamformer será constante apenas para sinais de banda estreita. Não é este o caso dos sinais de voz, o que significa que uma rede de espaçamento uniforme entre sensores não é a melhor alternativa quando se deseja uma resposta constante do beamformer em todo o espectro de freqüência.

Uma solução para este problema é fixar o comprimento da rede em termos do número de comprimentos de onda. Considere a rede com N elementos localizados em p_n , n = 1, 2, ..., N, para produzir o padrão de diretividade $b(\theta)$ para a freqüência f_1 . Então, para a freqüência f_2 , o mesmo padrão de diretividade $b(\theta)$ será produzido se os elementos estiverem localizados em $p_n(f_1/f_2)$, n = 1, 2, ..., N. Em outras palavras, o

comprimento da rede deve variar diretamente com a freqüência para se obter o mesmo padrão de diretividade. Na prática esta solução não é utilizada, pois seria necessário o uso de sensores contínuos. No entanto, é possível se fazer uma aproximação utilizando a rede de sensores discretos.

Um método simples para abranger sinais de banda larga é implementar uma série de sub-redes, onde o espaçamento é contínuo dentre de uma mesma sub-rede, porém variável entre as sub-redes. Estas sub-redes são projetadas para produzir uma determinada resposta dentro de uma faixa de freqüência. Devido às características do padrão de diretividade discutidas na seção 2.2.3, à medida que a freqüência aumenta, o tamanho da rede deve diminuir para se manter a mesma resposta. Além disso, para que se mantenham as características do lóbulo principal do padrão de diretividade, deve-se manter fixo o número de sensores. As sub-redes são geralmente implementadas aninhadas, de forma que um mesmo sensor pode ser utilizado em mais de uma sub-rede. Cada sub-rede fica restrita a uma diferente faixa de freqüência que é separada através de filtros passa-faixa. O resultado final é obtido somando-se as saídas das sub-redes. Um exemplo desta estrutura de sub-redes aninhadas é mostrado na Figura 3.4, que mostra a aplicação das sub-redes em um *beamformer Delay and Sum* contendo 9 microfones e 3 faixas de freqüência. O sinal de saída deste *beamformer* será a somatória de todas as sub-redes.

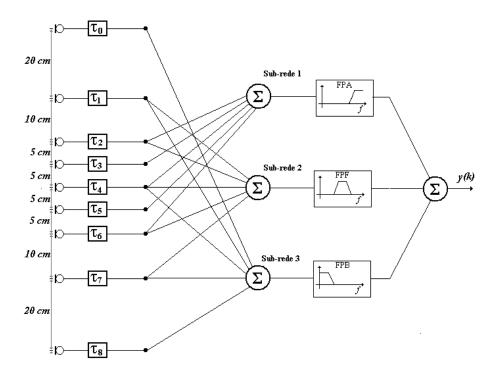


Figura 3.4 Delay and sum com sub-rede de 9 microfones e 3 faixas de freqüência

3.2 SUPERDIRECTIVE BEAMFORMING

3.2.1 Introdução

O termo *super* em *superdirective* indica que este tipo de *beamformer*, quando comparado com o tradicional *delay and sum*, é capaz de produzir uma maior diretividade, provando que a simples soma dos sinais não é a melhor escolha quando se deseja obter a diretividade ótima. Em parte da literatura, o *superdirective* é também conhecido como super-ganho. O termo diretividade denota a habilidade do *beamformer* de eliminar o ruído vindo de todas as direções sem afetar o sinal desejado vindo de uma direção principal.

Em meados da década de 50, Gilbert e Morgan [24] e Uzsoky e Solimár [25] utilizaram pela primeira vez o conceito do *superdirective beamforming* associado a um arranjo de transmissores com o objetivo de maximizar o seu ganho geométrico. O

ganho geométrico de um arranjo de transmissores é matematicamente equivalente ao ganho de um arranjo de receptores quando submetido a um ruído difuso. As soluções apresentadas naquela época eram apenas de interesse acadêmico, já que problemas associados aos microfones, como ruído próprio e variações de ganho e de fase impediam sua utilização prática. Somente na década de 90, o superdirective beamformer foi utilizado em aplicações com microfones. Pesquisas na área de aparelhos auditivos mostraram sua vantagem sobre os beamformers adaptativos [26]. Recentes estudos sobre o superdirective beamformer incluem as considerações do campo próximo e também a possibilidade de adaptar restrições com relação à situação real.

3.2.2 Avaliação dos Beamformers

Para o esquema mostrado na Figura 3.5, considera-se que uma amostra do sinal de entrada x(k) para cada sensor n consiste de uma versão atenuada e atrasada do sinal desejado $a_n s(k - \tau_n)$ e de uma componente de ruído $v_n(k)$.

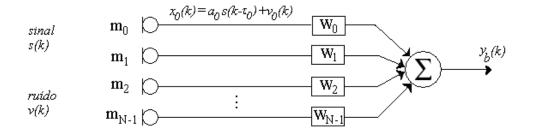


Figura 3.5 Esquema do superdirective beamformer

$$\begin{pmatrix} x_{0}(k) \\ x_{1}(k) \\ \vdots \\ x_{N-1}(k) \end{pmatrix} = \begin{pmatrix} a_{0}s(k-\tau_{0}) \\ a_{1}s(k-\tau_{1}) \\ \vdots \\ a_{N-1}s(k-\tau_{N-1}) \end{pmatrix} + \begin{pmatrix} v_{0}(k) \\ v_{1}(k) \\ \vdots \\ v_{N-1}(k) \end{pmatrix}$$
(3.34)

$$\mathbf{x}(k) = \mathbf{as}(k - \mathbf{T}) + \mathbf{v}(k). \tag{3.35}$$

Como todas as medidas relevantes dependem da freqüência, as análises seguintes serão passadas para o domínio da freqüência. A transformada de Fourier leva a

$$\mathbf{X}(e^{j\Omega}) = S(e^{j\Omega})\mathbf{d} + \mathbf{V}(e^{j\Omega}), \tag{3.36}$$

onde **d** é a representação dos atrasos e atenuações no domínio da freqüência e depende da geometria da rede e da direção da fonte do sinal desejado.

$$\mathbf{d}^{T} = [\mathbf{a}_{0} \mathbf{e}^{-j\Omega \tau_{0}}, \mathbf{a}_{1} \mathbf{e}^{-j\Omega \tau_{1}}, \dots, \mathbf{a}_{N-1} \mathbf{e}^{-j\Omega \tau_{N-1}}]. \tag{3.37}$$

Finalmente, o sinal de saída será

$$Y_{b}(e^{j\Omega}) = \sum_{n=0}^{N-1} W_{n}^{*}(e^{j\Omega}) X_{n}(e^{j\Omega}) = \mathbf{W}^{H} \mathbf{X} , \qquad (3.38)$$

onde $W_n(e^{j\Omega})$ denota o coeficiente do *beamformer* no domínio da freqüência no sensor n para a freqüência Ω e o operador H denota o conjugado transposto (operador Hermitiano). A transformada inversa de Fourier de $Y_b(e^{j\Omega})$ resulta na saída $y_b(k)$ de tempo discreto.

3.2.3 Ganho do Beamformer

O ganho do *beamformer* é a medida que mostra a melhora da razão sinal-ruído SNR (*signal to noise ratio*) entre um único sensor e a saída da rede:

$$G(e^{j\Omega}) = \frac{SNR_{rede}(e^{j\Omega})}{SNR_{sensor}(e^{j\Omega})}.$$
(3.39)

Considerando-se sinais estacionários, a SNR de um sensor é o quociente da densidade espectral de potência (DEP) do sinal desejado Φ_{ss} pela média das DEP do ruído $\Phi_{v_av_a}^{-1}$.

A SNR da rede pode ser obtida a partir da DEP do sinal de saída:

$$\Phi_{\gamma_b \gamma_b} = \mathbf{W}^{\mathsf{H}} \Phi_{\mathbf{X} \mathbf{X}} \mathbf{W} \,, \tag{3.40}$$

onde

 $^{^{1}}$ A partir deste ponto, a dependência do termo $e^{i\Omega}$ será omitida em benefício da legibilidade.

$$\mathbf{\Phi}_{\mathbf{XX}} = \begin{pmatrix} \Phi_{X_{0}X_{0}} & \Phi_{X_{0}X_{1}} & \cdots & \Phi_{X_{0}X_{N-1}} \\ \Phi_{X_{1}X_{0}} & \Phi_{X_{1}X_{1}} & \cdots & \Phi_{X_{1}X_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{X_{N-1}X_{0}} & \Phi_{X_{N-1}X_{1}} & \cdots & \Phi_{X_{N-1}X_{N-1}} \end{pmatrix}$$
(3.41)

é a matriz da densidade espectral de potência do sinal de entrada. Quando apenas o sinal desejado estiver presente, a saída será:

$$\Phi_{Y_b Y_b}\Big|_{\sin al} = \Phi_{ss} \Big| \mathbf{W}^{\mathsf{H}} \mathbf{d} \Big|^2; \tag{3.42}$$

para o caso de se ter apenas o ruído,

$$\Phi_{Y_b Y_b} \Big|_{uido} = \Phi_{V_a V_a} \mathbf{W}^{\mathsf{H}} \mathbf{\Phi}_{\mathbf{v} \mathbf{v}} \mathbf{W} , \qquad (3.43)$$

onde $\Phi_{\mathbf{w}}$ é a matriz normalizada da densidade espectral cruzada do ruído. O fator de normalização $\Phi_{\mathbf{v}_a\mathbf{v}_a}$ é escolhido de forma a forçar que que o traço da matriz $\Phi_{\mathbf{w}}$ seja igual a N.

A *SNR*_{rede} será obtida através da razão entre a equação (3.42) e a equação (3.43). Substituindo este valor em (3.39), o ganho da rede será:

$$G = \frac{\left|\mathbf{W}^{H}\mathbf{d}\right|^{2}}{\mathbf{W}^{H}\mathbf{\Phi}_{\text{tot}}\mathbf{W}}.$$
(3.44)

Assumindo um campo de ruído homogêneo, a equação (3.44) pode ser escrita em função da matriz de coerência, e pode ser definida como:

$$\Gamma_{\mathbf{vv}} = \begin{pmatrix}
1 & \Gamma_{V_0 V_1} & \cdots & \Gamma_{V_0 V_{N-1}} \\
\Gamma_{V_1 V_0} & 1 & \cdots & \Gamma_{V_1 V_{N-1}} \\
\vdots & \vdots & \ddots & \vdots \\
\Gamma_{V_{N-1} V_0} & \Gamma_{V_{N-1} V_0} & \cdots & 1
\end{pmatrix},$$
(3.45)

onde

$$\Gamma_{V_n V_m}(e^{j\Omega}) = \frac{\Phi_{V_n V_m}(e^{j\Omega})}{\sqrt{\Phi_{V_n V_n}(e^{j\Omega})\Phi_{V_m V_m}(e^{j\Omega})}}$$
(3.46)

representa a função de coerência.

Desta forma, o ganho poderá ser escrito como

$$G = \frac{\left|\mathbf{W}^{H}\mathbf{d}\right|^{2}}{\mathbf{W}^{H}\mathbf{\Gamma}_{VV}\mathbf{W}}.$$
(3.47)

Esta representação do ganho permite uma fácil avaliação dos *beamformers* para diferentes tipos de ruídos, uma vez que estes podem ser representados por suas funções de coerência.

3.2.4 Otimização do Ganho

O *beamformer* que maximiza o ganho da rede é considerado ótimo para uma variedade de aplicações de detecção e estimação de sinal. O problema da otimização pode ser representado matematicamente da seguinte forma:

$$\max_{\mathbf{w}} \frac{\left|\mathbf{W}^{H}\mathbf{d}\right|^{2}}{\mathbf{W}^{H}\mathbf{\Phi}_{\mathbf{W}}\mathbf{W}}.$$
 (3.48)

Utilizando o multiplicador de Lagrange, Cox [27] apresenta a seguinte solução para este problema:

$$\mathbf{W} = \mathbf{\alpha} \, \mathbf{\Phi}_{\mathbf{V}\mathbf{V}}^{-1} \, \mathbf{d} \,, \tag{3.49}$$

onde α é uma constante complexa arbitrária. Escolhendo-se α de forma a produzir ganho unitário e fase zero tal que $\mathbf{W}^H\mathbf{d}=\mathbf{1}$, tem-se

$$\mathbf{W} = \frac{\mathbf{\Phi}_{VV}^{-1} \mathbf{d}}{\mathbf{d}^{H} \mathbf{\Phi}_{VV}^{-1} \mathbf{d}}.$$
 (3.50)

A solução de Cox para o problema da equação (3.48) produz o *beamformer* de resposta sem distorção com mínima variância (*Minimum Variance Distortionless Response – MVDR*).

Substituindo a equação (3.50) em (3.44), o ganho da rede será

$$G = \mathbf{d}^H \; \mathbf{\Phi}_{\mathsf{VV}}^{-1} \; \mathbf{d} \; . \tag{3.51}$$

3.2.5 Otimização do Ganho com Restrição

A dificuldade prática para a otimização do ganho mostrado na seção anterior está no fato de não se ter como medir diretamente o ruído e sua matriz de densidade espectral cruzada Φ_{vv} . Uma importante sugestão foi apresentada por Levin [28], que propôs a minimização da potência de saída do *beamformer* $\Phi_{Y_bY_b}$, sujeita à restrição de resposta sem distorção e com ganho unitário para a direção desejada:

$$\min_{\mathbf{W}} \mathbf{W}^H \mathbf{\Phi}_{\mathbf{X}\mathbf{X}} \mathbf{W} \quad \text{sujeito a} \quad \mathbf{W}^H \mathbf{d} = \mathbf{1}. \tag{3.52}$$

Este problema é matematicamente equivalente ao apresentado pela equação (3.48), assim, sua solução, definida em [27], será

$$\mathbf{W} = \frac{\mathbf{\Phi}_{\mathbf{X}\mathbf{X}}^{-1} \mathbf{d}}{\mathbf{d}^{H} \mathbf{\Phi}_{\mathbf{Y}\mathbf{Y}}^{-1} \mathbf{d}}.$$
 (3.53)

Embora mais simples, esta solução para o problema causa a supressão de parte do sinal desejado na saída do *beamformer*, devido a erros de avaliação das características espaciais do sinal.

3.2.6 Ganho do Beamformer com Ruído Branco

Com o objetivo de resolver o problema da amplificação indesejada do próprio ruído dos microfones, Gilbert e Morgan [24], propuseram maximizar o ganho do beamformer sujeito à uma limitação no ganho com ruído branco.

$$\max_{\mathbf{W}} G \text{ sujeito a } G_{\mathbf{W}} = \delta^2 \le M, \tag{3.54}$$

onde G_w é o ganho com ruído branco, δ^2 deve ser escolhido menor ou igual a um valor máximo permitido para M.

De maneira equivalente, pode-se minimizar $((1/G) + \alpha(1/G_w))$ sendo α o multiplicador de Lagrange. Desta forma, tem-se que

$$\min_{\mathbf{W}} \frac{\mathbf{W}^{H} \mathbf{\Phi}_{\mathbf{w}} \mathbf{W}}{\left| \mathbf{W}^{H} \mathbf{d} \right|^{2}} + \frac{\alpha \mathbf{W}^{H} \mathbf{W}}{\left| \mathbf{W}^{H} \mathbf{d} \right|^{2}} = \min_{\mathbf{W}} \frac{\mathbf{W}^{H} (\mathbf{\Phi}_{\mathbf{w}} + \alpha \mathbf{I}) \mathbf{W}}{\left| \mathbf{W}^{H} \mathbf{d} \right|^{2}}.$$
 (3.55)

Analogamente à solução apresentada em (3.50) e também respeitando o ganho unitário e a fase zero, a solução será

$$\mathbf{W} = \frac{(\mathbf{\Phi}_{vv} + \alpha \mathbf{I})^{-1} \mathbf{d}}{\mathbf{d}^{H} (\mathbf{\Phi}_{vv} + \alpha \mathbf{I})^{-1} \mathbf{d}},$$
(3.56)

onde α é ajustado para satisfazer a restrição do ganho do *beamformer* com ruído branco. Verifica-se que a adição de α à diagonal de Φ_{vv} , produz a adição de α aos autovalores sem modificar os autovetores.

O problema da maximização do ganho do *beamformer* sujeito a restrições com o ruído branco envolve três formas quadráticas:

$$\left|\mathbf{W}^{H}\mathbf{d}\right|^{2}$$
, $\mathbf{W}^{H}\mathbf{\Phi}_{VV}\mathbf{W} \in \mathbf{W}^{H}\mathbf{W}$, (3.57)

que estão presentes nos cálculos do ganho do *beamformer*, na potência de saída e no ganho com ruído branco. Existem algumas outras propostas equivalentes para o problema da otimização que utilizam estas formas quadráticas:

 Problema 1: Maximizar o ganho do beamformer restringindo o ganho com ruído branco e a resposta do sinal [29]:

$$\max_{\mathbf{w}} \frac{\left| \mathbf{W}^{H} \mathbf{d} \right|^{2}}{\mathbf{W}^{H} \mathbf{\Phi}_{\mathbf{w}} \mathbf{W}} \text{ sujeito a } \frac{\left| \mathbf{W}^{H} \mathbf{d} \right|}{\mathbf{W}^{H} \mathbf{W}} = \delta^{2}, \mathbf{W}^{H} \mathbf{d} = \mathbf{1}.$$
 (3.58)

 Problema 2: Maximizar o ganho do beamformer restringindo a norma de W e a resposta do sinal:

$$\max_{\mathbf{w}} \frac{\left| \mathbf{W}^{H} \mathbf{d} \right|^{2}}{\mathbf{W}^{H} \mathbf{\Phi}_{\mathbf{w}} \mathbf{W}} \text{ sujeito a } \mathbf{W}^{H} \mathbf{W} = \delta^{-2}, \mathbf{W}^{H} \mathbf{d} = \mathbf{1}.$$
 (3.59)

 Problema 3: Minimizar a potência de saída restringindo o ganho com ruído branco e a resposta do sinal [30]:

$$\min_{\mathbf{w}} \mathbf{W}^{H} \mathbf{\Phi}_{\mathbf{x}\mathbf{x}} \mathbf{W} \text{ sujeito a } \frac{\left| \mathbf{W}^{H} \mathbf{d} \right|}{\mathbf{W}^{H} \mathbf{W}} = \delta^{2}, \mathbf{W}^{H} \mathbf{d} = 1.$$
 (3.60)

 Problema 4: Minimizar a potência de saída restringindo a norma de W e a resposta do sinal [31]:

$$\min_{\mathbf{w}} \mathbf{W}^{H} \mathbf{\Phi}_{\mathbf{x}\mathbf{x}} \mathbf{W} \text{ sujeito a } \mathbf{W}^{H} \mathbf{W} = \delta^{-2}, \mathbf{W}^{H} \mathbf{d} = \mathbf{1}.$$
 (3.61)

3.2.7 Otimização do Ganho com Múltiplas Restrições Lineares

A aplicação de múltiplas restrições lineares é uma generalização do caso com uma única restrição. Este método foi introduzido em 1971 por Booker e Ong [32] e utilizado por Frost em 1972 [34] em seu conhecido algoritmo recursivo adaptativo para cálculo do *beamformer*.

Este problema é formulado fazendo-se a minimização da potência de saída sujeita às múltiplas restrições lineares:

$$\min_{\mathbf{w}} \mathbf{W}^{H} \mathbf{\Phi}_{\mathbf{x}\mathbf{x}} \mathbf{W} \text{ sujeito a } \mathbf{C}^{H} \mathbf{W} = \mathbf{g}.$$
 (3.62)

A matriz \mathbf{C}^H $K \times M$ (K<M) tem linhas linearmente independentes. Uma das linhas de \mathbf{C}^H inclui a resposta de ganho unitário e de fase zero $\mathbf{W}^H\mathbf{d} = \mathbf{1}$. Esta importante restrição permite sua combinação para produzir respostas sem distorção de amplitude e fase nas freqüências desejadas. Outras restrições podem modificar o formato do lóbulo principal ou reduzir o ganho em determinadas direções.

O problema pode ser resolvido utilizando-se o multiplicador de Lagrange para computar as restrições à função objetivo. Sua solução será:

$$\mathbf{W} = \mathbf{\Phi}_{\mathbf{v}\mathbf{v}}^{-1} \mathbf{C} [\mathbf{C}^H \mathbf{\Phi}_{\mathbf{v}\mathbf{v}}^{-1} \mathbf{C}]^{-1} \mathbf{g} . \tag{3.63}$$

3.2.8 Considerações para o Campo Próximo

Conforme discutido na seção 2.4, sempre que tivermos a fonte de sinal próxima à rede de microfones, a curvatura das ondas passa a ter efeito importante, não podendo mais ser ignorada, sob a pena da perda de diretividade do *beamformer*.

Täger [33] propôs um método chamado *near-field superdirective*, e como o próprio nome diz, este método é uma adaptação do tradicional *superdirective*, descrito nas seções anteriores, para o caso do campo próximo. Este método consiste em modificar o vetor **d** substituindo seus valores por outros formulados para o caso do campo próximo.

Na seção 2.4, foi visto que em uma rede linear a distância da fonte de sinal até um sensor arbitrário é dada por

$$d_n(r,\phi) = \sqrt{r_0^2 - 2r_0 nd\cos\phi_0 + (nd)^2} , \qquad (3.64)$$

onde r_0 e ϕ_0 representam a distancia e o ângulo da fonte em relação ao microfone de referência.

A diferença de tempo na propagação da onda entre o microfone n e o microfone de referência (n=0) pode ser definida como

$$\tau_n = \frac{d_n(r,\phi) - d_0(r,\phi)}{c},$$
(3.65)

e o fator de atenuação de amplitude entre o microfone n e o microfone de referência como

$$\alpha_n = \frac{d_0(r,\phi)}{d_n(r,\phi)}.$$
 (3.66)

Então, o vetor **d** modificado para o campo próximo pode ser expresso como:

$$d_{nf}(\Omega) = [\alpha_0 e^{-j\Omega\tau_0}, \alpha_1 e^{-j\Omega\tau_1}, ..., \alpha_{N-1} e^{-j\Omega\tau_{N-1}}]. \tag{3.67}$$

O near-field superdirective apresenta melhores resultados que o superdirective tradicional, principalmente para as baixas freqüências. Isto ocorre porque ele leva em consideração as diferenças de amplitude e de fase. Enquanto as diferenças de fase podem ser consideradas desprezíveis para as baixas freqüências, as diferenças de amplitude são bastante significativas, particularmente quando a fonte de sinal estiver próxima ao end-fire.

3.3 GENERALIZED SIDELOBE CANCELLER (GSC)

3.3.1 Introdução

Dentre os várias estruturas de beamformers adaptativos, a mais utilizada é a Generalized Sidelobe Canceller (GSC), também conhecida como Griffths-Jim Beamformer (GJBF) [37]. O GSC pertence a uma classe de beamformers chamada Linear Constrained Minimum Variance (LCMV). As condições restritivas impostas por esta classe de algoritmos são empregadas com o objetivo de garantir que a resposta ao sinal desejado possua ganho constante e fase linear. Este conceito foi apresentado por Frost em 1972 [34] e posteriormente desenvolvido por Applebaum e Chapman em 1976 [35] e Griffiths e Jim em 1982 [37]. De maneira geral, o LCMV tenta minimizar o erro médio quadrático da saída do sistema utilizando geralmente algoritmos como o Least Mean Square (LMS). O mérito do algoritmo de Frost é tratar o processo de estimação dos filtros utilizando o LMS com restrições (CLMS), o que garante que a solução minimize o erro médio quadrático e ao mesmo tempo garanta uma função de transferência pré-determinada para o sinal desejado. O GSC é uma estrutura de beamformer que pode ser utilizada para implementar um grande número de algoritmos LCMV, entre eles, o algoritmo de Frost [34]. Outras técnicas combinadas ao GSC podem ser úteis na solução do problema de localização da fonte sonora, conforme demonstrado em [35].

3.3.2 Princípios do GSC

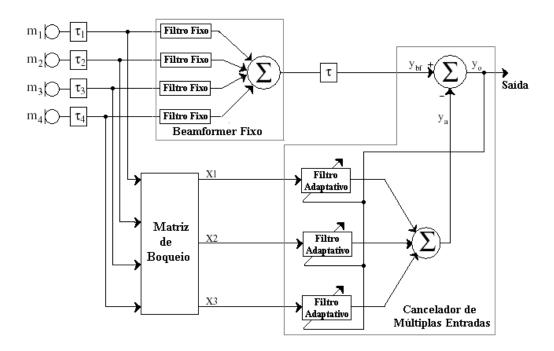


Figura 3.6 Estrutura do Generalized Sidelobe Canceller (GSC): exemplo com 4 microfones

A Figura 3.6 mostra a estrutura de um *beamformer* GSC, que é composto de um *beamformer* fixo, um cancelador de múltiplas entradas e uma matriz de bloqueio. O *beamformer* fixo é utilizado para permitir a passagem do sinal que vem da direção desejada e atenuar todos os demais sinais. De maneira oposta, a matriz de bloqueio deve anular o sinal que venha da direção desejada e permitir a passagem dos demais sinais. A estrutura mais simples da matriz de bloqueio é o *delay and subtract*, que efetua a diferença entre sinais de microfones adjacentes. No caso da figura com 4 microfones, a matriz de bloqueio é apresentada como sendo

$$B = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}. \tag{3.68}$$

O cancelador de múltiplas entradas pode ser composto por diferentes tipos de filtros adaptativos que recebem os sinais da matriz de bloqueio. Desta forma, os filtros adaptativos geram réplicas de componentes correlacionadas com as interferências.

Estas réplicas são subtraídas, formando o sinal $y_a(k)$. O sinal de saída $y_o(k)$ é gerado fazendo-se

$$y_o(k) = y_{bf}(k - \tau) - y_a(k)$$
. (3.69)

O atraso τ no sinal y_{bf} é inserido para que sejam compensados os atrasos nos processamentos da matriz de bloqueio e do cancelador de múltiplas entradas.

Para uma análise mais aprofundada do cancelador de múltiplas entradas, deve-se considerar a utilização do algoritmo NLMS nos seus filtros adaptativos, sendo os vetores $\mathbf{a}_1(k)$, $\mathbf{a}_2(k)$ e $\mathbf{a}_3(k)$ os coeficientes dos filtros adaptativos. Estes vetores de coeficientes podem ser combinados para criar o vetor global

$$\mathbf{a}(k) = \begin{bmatrix} \mathbf{a}_1(k) \\ \mathbf{a}_2(k) \\ \mathbf{a}_3(k) \end{bmatrix}. \tag{3.70}$$

Os vetores $\mathbf{x}_1(k)$, $\mathbf{x}_2(k)$ e $\mathbf{x}_3(k)$ contêm as entradas para os seus respectivos filtros adaptativos. Cada $\mathbf{x}_1(k)$ é apresentado como

$$\mathbf{x}_{i}(k) = \begin{bmatrix} x_{i}(k-p) \\ x_{i}(k-p+1) \\ \vdots \\ x_{i}(k) \end{bmatrix}, \tag{3.71}$$

onde p é a ordem do filtro adaptativo. De forma geral, o vetor de entrada pode ser apresentado como:

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \\ \mathbf{x}_3(k) \end{bmatrix}. \tag{3.72}$$

Após estas definições, uma iteração do filtro NLMS pode ser escrita como

$$\mathbf{a}(k+1) = \mathbf{a}(k) + \frac{\mu}{\|\mathbf{x}(k)\|^2} \mathbf{x}(k) y_o(k), \qquad (3.73)$$

onde μ é o passo de atualização. A saída do sistema será

$$y_o(k) = y_{bf}(k) - y_a(k),$$
 (3.74)

onde

$$y_a(k) = \mathbf{a}^H(k)\mathbf{x}(k). \tag{3.75}$$

Outro algoritmo que se pode usar nos filtros adaptativos é o RLS. Embora sua complexidade computacional seja bem maior que a do NLMS, o RLS pode apresentar uma taxa de convergência muito mais alta.

Os três filtros não podem ser adaptados separadamente, o que impede sua implementação em paralelo. O vetor de coeficientes $\mathbf{a}(k)$ e o vetor de entradas $\mathbf{x}(k)$ são os mesmos definidos anteriormente. Adicionalmente, será necessário considerar a matriz \mathbf{P} , inversa da matriz de correlações dos sinais de entrada. \mathbf{P} é uma matriz quadrada com (N-1)(p+1) linhas e colunas, onde N é o número de microfones da rede e p é a ordem do filtro adaptativo.

O algoritmo de adaptação do filtro é mostrado a seguir:

$$\mathbf{G}(k+1) = \frac{\mathbf{P}(k)\mathbf{x}(k+1)}{\beta + \mathbf{x}^{H}(k+1)\mathbf{P}(k)\mathbf{x}(k+1)},$$
(3.76)

onde β é uma constante que representa a memória do sistema. O vetor G(k) é também conhecido como vetor de ganho.

$$\mathbf{P}(k+1) = \frac{\mathbf{P}(k) - \mathbf{G}(k+1)\mathbf{x}^{H}(k+1)\mathbf{P}(k)}{\beta}$$
(3.77)

e finalmente, o vetor de coeficientes a(k) é calculado como

$$\mathbf{a}(k+1) = \mathbf{a}(k) + \mathbf{G}(k)y_{o}(k)$$
. (3.78)

3.3.3 GSC Robusto

O GSC é uma estrutura flexível devido à divisão do *beamformer* em uma parte fixa e outra adaptativa, o que possibilita sua implementação de muitas maneiras diferentes. Na prática, o GSC tal como apresentado anteriormente, é uma estrutura bastante sensível a erros no alinhamento do sinal desejado. Quando este problema ocorre, a matriz de bloqueio permite a passagem do sinal desejado, causando o fenômeno chamado de vazamento de sinal. Como resultado, o cancelador de múltiplas entradas provocará o cancelamento do sinal desejado na saída. Uma grande variedade de

técnicas foi proposta para reduzir o cancelamento do sinal desejado, principalmente nos campos de antenas e radares. Os *beamformers* que aplicam estas técnicas são conhecidos como *beamformers* robustos. As abordagens típicas para o problema estão em agir de forma direta na matriz de bloqueio eliminando o vazamento do sinal desejado; ou de forma indireta, restringindo o crescimento dos coeficientes do cancelador, assim, mesmo que haja um vazamento na matriz de bloqueio, o cancelador tenta minimizar os seus efeitos.

Em [38] é apresentado o GSC que utiliza na matriz de bloqueio e no cancelador de múltiplas entradas uma estrutura de filtros adaptativos conhecida como *Leaky Adaptive Filters* (LAF), por este motivo, a estrutura completa recebe o nome de GSC LAF-LAF. A Figura 3.7 mostra seu diagrama de blocos. Esta estrutura, com filtros adaptativos na matriz de bloqueio, é capaz de perseguir (em pequenas variações de direção) a fonte do sinal de interesse sem que haja seu cancelamento.

As saídas $z_n(k)(n=0,1,...,N-1)$ da matriz de bloqueio podem ser escritas como

$$Z_n(k) = X_n(k) - \mathbf{h}_n^T(k)\mathbf{b}(k), \tag{3.79}$$

$$\mathbf{h}_{n}(k) = [h_{n,0}(k), h_{n,1}(k), \dots, h_{n,M_{1}-1}(k)]^{T},$$
(3.80)

$$\mathbf{b}(k) = [b(k), b(k-1), \dots, b(k-M_1+1)]^T, \tag{3.81}$$

onde $x_n(k)$ é o sinal do n-ésimo microfone, $\mathbf{h}_n(k)$ é o vetor dos coeficientes do n-ésimo LAF, $\mathbf{b}(k)$ é o vetor com os últimos valores de b(k) que é a saída do *beamformer* fixo. Cada LAF possui M_1 coeficientes. O algoritmo de adaptação é o NLMS, e é descrito como

$$\mathbf{h}_{n}(k+1) = \mathbf{h}_{n}(k) - \delta \cdot \mathbf{h}_{n}(k) + \alpha \frac{Z_{n}(k)}{\mathbf{b}(k)^{T} \mathbf{b}(k)} \mathbf{b}(k), \qquad (3.82)$$

onde α é o passo para a adaptação do algoritmo e δ , $0 \le \delta \le 1$, é a constante de vazamento. O cancelador múltiplo de entradas também utiliza os LAFs para evitar o cancelamento indevido de parcelas remanescentes do sinal desejado.

Os vetores de coeficientes $\mathbf{w}_n(k)$ do cancelador de múltiplas entradas possuem M_2 coeficientes e suas adaptações são similares às descritas pela equação (3.82), onde

 $\mathbf{h}_n(k)$, $\mathbf{b}(k)$, $z_n(k)$, α e δ são substituídos por \mathbf{w}_n , \mathbf{z}_n , y(k), γ e β , respectivamente. As constantes γ e β podem ser valores diferentes de α e δ .

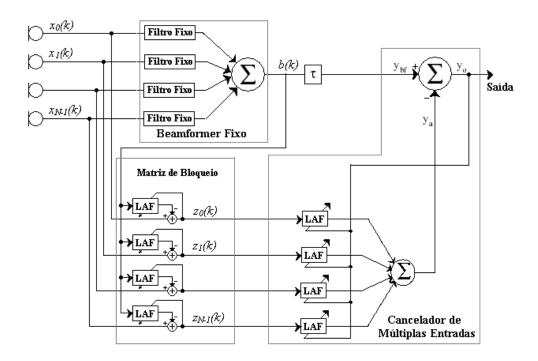


Figura 3.7 Diagrama de blocos do GSC LAF-LAF

3.3.4 GSC com Filtros Adaptativos IIR

Em um ambiente com reverberação moderada, os filtros adaptativos das estruturas dos GSCs podem necessitar de até 1000 coeficientes para a completa eliminação desta interferência. Este grande número de coeficientes leva a uma grande complexidade computacional e conseqüentemente a uma convergência demorada. A utilização de filtros adaptativos IIR em substituição aos filtros FIR podem produzir estruturas com menor número de coeficientes e conseqüentemente, menor complexidade computacional.

A estrutura proposta para este trabalho utiliza o algoritmo de Gauss-Newton para os filtros adaptativos IIR [39][40]. Estes filtros são utilizados no cancelador de múltiplas entradas conforme mostra o diagrama de blocos da Figura 3.8.

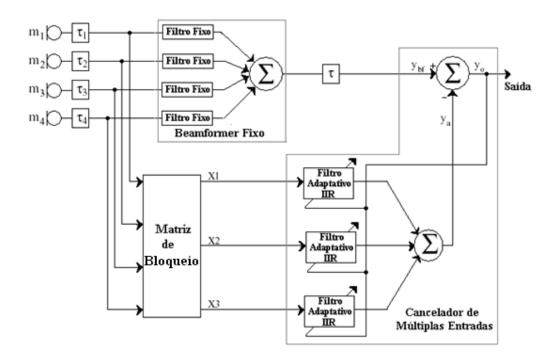


Figura 3.8 Diagrama de blocos do GSC IIR: exemplo com 4 microfones

Para cada filtro adaptativo IIR faremos as seguintes considerações:

$$y(k) = \sum_{m=0}^{N_b-1} b_m(k) x(k-m) + \sum_{m=1}^{N_a-1} a_m(k) y(k-m), \qquad (3.83)$$

onde x(k) é o sinal de entrada do filtro e é obtido na saída da matriz de bloqueio, N_b e N_a são as ordens do numerador e do denominador do filtro, respectivamente.

Seja

$$\phi(k) = [y(k-1)...y(k-N_a+1) \ x(k-1)...x(k-N_b+1)]^T \ (3.84)$$

o vetor de informação e

$$\theta(k) = [a_1(k) \dots a_{N_0-1}(k) \quad b_0(k) \dots b_{N_0-1}(k)]^T$$
(3.85)

o vetor dos coeficientes do filtro.

O algoritmo de adaptação será definido pelas seguintes equações:

$$P(k+1) = \frac{1}{\beta}P(k) - \frac{\alpha}{\beta} \left(\frac{P(k)\psi(k)\psi^{H}(k)P(k)}{\beta + \alpha\psi^{H}(k)P(k)\psi(k)} \right), \tag{3.86}$$

onde α e β definem o passo de adaptação do filtro,

$$\psi(k) = [y^{f}(k-1)...y^{f}(k-N_a+1) \ x^{f}(k)...x^{f}(k-N_b+1)]^{T}, \tag{3.87}$$

onde

$$x^{f}(k) = x(k) + \sum_{m=1}^{N_a - 1} a_m(k) \cdot x^{f}(k - m) \qquad , \tag{3.88}$$

$$y^{f}(k-1) = y(k-1) + \sum_{m=1}^{N_a-1} a_m y^{f}(k-m-1),$$
 (3.89)

$$y(k) = \theta^{\mathsf{T}}(k)\phi(k) \qquad e \tag{3.90}$$

$$\theta(k+1) = \theta(k) + \alpha P(k+1)\psi(k)e(k), \qquad (3.91)$$

onde e(k) é o sinal de erro e deve ser retirado na saída do GSC IIR pelo sinal $y_o(k)$.

A implementação deste algoritmo em Matlab está descrita no Apêndice A e os resultados obtidos com ensaios estão detalhados na seção 5.1.4.

3.4 PÓS-FILTRAGEM (POST-FILTERING)

3.4.1 Introdução

No contexto das redes de microfones, o termo pós-filtragem é utilizado para nomear o processamento do sinal de saída de uma rede através de um filtro de supressão de ruídos. O desempenho das estruturas como o *Delay-and-Sum* e o *Filter-and-Sum* dependem, dentre outros fatores, do número de microfones que compõem a rede, conforme demonstra a equação (3.25). Devido a demanda atual por redes de pequenas dimensões e de baixo custo, apenas as redes de poucos microfones garantirão aplicações de grande escala. Para melhorar o desempenho das redes de microfones que utilizam o *Delay-and-Sum* e o *Filter-and-Sum* sem aumentar o número de microfones, a técnica da pós-filtragem pode ser utilizada com sucesso.

A função de transferência deste novo filtro de freqüências é estimada a partir das definições do filtro ótimo de Wiener, e portanto dependem da matriz de correlação do

ruído de entrada e do vetor de correlação cruzada entre o ruído de entrada e o sinal desejado [41][42][43].

Para o bom funcionamento deste algoritmo, algumas considerações são necessárias, como: conhecimento prévio da função de correlação espacial do ruído, também devese considerar que estas perturbações não sejam correlacionadas com o sinal desejado.

O primeiro trabalho a abordar a utilização do filtro de Wiener em uma rede de microfones foi publicado por Zelinsk em 1988 [41].

3.4.2 Princípios da Pós-Filtragem

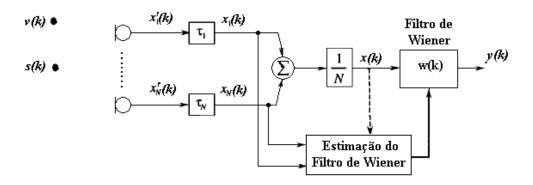


Figura 3.9 Esquema da estrutura de pós-filtragem

Analisando o esquema da rede de microfones mostrado na Figura 3.9, tem-se que o sinal x(k) é obtido na saída de um *Delay and Sum* e em seguida filtrado por $\mathbf{w}(k)$. O sinal x(k) pode ser escrito como

$$x(k) = s(k) + v(k), \tag{3.92}$$

onde s(k) é o sinal desejado e v(k) é o resíduo do ruído. Desta forma, pode-se escrever o sinal de saída y(k) como

$$y(k) = \mathbf{w}^{H}(k)\mathbf{x}(k) = \mathbf{w}^{H}(k)[\mathbf{s}(k) + \mathbf{v}(k)]. \tag{3.93}$$

O sinal de erro da saída em relação ao sinal desejado é dado por

$$e(k) = s(k) - \mathbf{w}^{H}(k)\mathbf{x}(k). \tag{3.94}$$

Para minimizar e(k), deve-se primeiro calcular o seu erro médio quadrático – MSE:

$$E[e^{2}(k)] = [\{s(k) - \mathbf{w}^{H}(k)\mathbf{x}(k)\}\{s^{*}(k) - \mathbf{x}^{H}(k)\mathbf{w}(k)\}].$$
(3.95)

Desenvolvendo a equação acima

$$\phi_{ee}(k) = \phi_{ss}(k) - w^{H}(k)\phi_{xs}(k) - \phi_{xs}^{H}(k)w(k) + w^{H}(k)\Phi_{xx}(k)w(k), \qquad (3.96)$$

onde

$$\phi_{ee}(k) = E[e(k)e(k)^{*}] \tag{3.97}$$

representa a potência do sinal complexo de erro e(k),

$$\phi_{ss}(k) = E[s(k)s(k)^*] \tag{3.98}$$

representa a potência do sinal complexo desejado s(k),

$$\phi_{xs}(k) = E[x(k)s^{*}(k)]$$
(3.99)

representa o vetor de correlação cruzada dos sinais x(k) e s(k) e

$$\mathbf{\Phi}_{xx}(k) = E[x(k)x^{H}(k)] \tag{3.100}$$

representa a matriz de autocorrelação do sinal x(k).

A potência ϕ_{ee} definida pela equação (3.96) é uma função quadrática com relação a \mathbf{w} , e possui um valor de mínimo. O valor de \mathbf{w} que minimiza o erro quadrático é obtido igualando-se a zero o gradiente de ϕ_{ee} com relação a \mathbf{w} [44]:

$$\nabla_{\mathbf{w}}(\phi_{ee}) = 2\frac{\partial \phi_{ee}}{\partial \mathbf{w}} = -2\phi_{xs} + 2\mathbf{\Phi}_{xx}\mathbf{w} = 0.$$
 (3.101)

O resultado da expressão representa a forma geral da equação de Wiener-Hopf

$$\mathbf{\Phi}_{xx}\mathbf{W}_{opt} = \mathbf{\phi}_{xs}, \qquad (3.102)$$

onde Φ_{xx} é a matriz de correlação do sinal de entrada e ϕ_{xs} é a correlação cruzada entre o sinal de entrada x(k) e o sinal desejado s(k). Considerando Φ_{xx} não singular, pode-se escrever (3.102) como:

$$\mathbf{W}_{opt} = \mathbf{\Phi}_{xx}^{-1} \mathbf{\phi}_{xs}. \tag{3.103}$$

Substituindo (3.92) em (3.100):

$$\mathbf{\Phi}_{xx}(k) = E[\{s(k) + v(k)\}\{s(k) + v(k)\}^{H}]$$
(3.104)

e substituindo (3.92) em (3.99)

$$\phi_{vs}(k) = E[\{s(k) + v(k)\}s^{*}(k)]. \tag{3.105}$$

Considerando-se que o ruído v(k) e o sinal desejado s(k) são não correlacionados, pode-se reescrever as equações (3.104) e (3.105) como:

$$\mathbf{\Phi}_{xx}(\mathbf{k}) = \mathbf{\Phi}_{ss}(\mathbf{k}) + \mathbf{\Phi}_{vv}(\mathbf{k}) \tag{3.106}$$

$$\phi_{vs}(\mathbf{k}) = \phi_{ss}(\mathbf{k}) \tag{3.107}$$

respectivamente. Desta forma, a equação (3.103), pode ser reescrita como:

$$\mathbf{W}_{\text{opt}}(\mathbf{k}) = \frac{\phi_{\text{ss}}(\mathbf{k})}{\Phi_{\text{vx}}(\mathbf{k})}$$
(3.108)

3.4.3 Pós-Filtro de Zelinsk

Para uma rede de microfones como mostra a Figura 3.9, Zelinsk [41] propôs um algoritmo para o cálculo do filtro de Wiener baseado nas funções de autocorrelação e de correlação cruzada dos sinais dos microfones. O termo Φ_{xx} da equação (3.108) pode ser obtido imediatamente através dos sinais de entrada $x_i(k)$, onde i representa o índice do microfone considerado. No entanto, $\phi_{ss}(k)$ deve ser estimado indiretamente. Assumindo que os sinais de ruído dos microfones $x_1(k)$ até $x_N(k)$ são mutuamente não correlacionados e independentes de s(k), tem-se que

$$E[x_i(k)x_i(k+l)] = E[\{s(k) + v_i(k)\}\{s(k+l) + v_i(k+l)\}] = \phi_{ss}(l) \text{ para } i \neq j.$$
 (3.109)

Assim pode-se dizer que:

$$\phi_{ss}(k) = \phi_{x_i x_j}(k) \text{ para } i \neq j.$$
 (3.110)

A estimativa do filtro de Wiener pode ser mais robusta se for feita uma média das densidades espectrais de potência de todas as combinações possíveis dos sensores, conforme mostrado a seguir.

$$\hat{w}_z = \frac{\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Re{\{\hat{\phi}_{x_i x_j}\}}}{\frac{1}{N} \sum_{i=1}^{N} \hat{\phi}_{x_i x_i}}.$$
(3.111)

O operador real $\Re\{\cdot\}$ é utilizado no numerador porque o termo estimado ϕ_{ss} é necessariamente um valor real.

3.4.4 Pós-Filtro para Ruído Difuso

Na formulação do algoritmo do pós-filtro de Zelinsk, o ruído foi considerado perfeitamente incoerente, o que raramente ocorre na prática, em se tratando de locais fechados. A correlação do ruído entre canais pode ser significativa particularmente nas baixas freqüências ou quando o espaçamento entre os sensores for pequeno, o que freqüentemente acontece nas aplicações de melhoria dos sinais de voz.

Para minimizar este problema, vários trabalhos [47] [48] [50] propuseram a consideração da função de coerência do ruído. O ruído difuso é uma boa aproximação empregada em um grande número de aplicações de melhoria dos sinais de voz para ambientes fechados como salas, auditórios e carros. Uma medida comum para caracterizar o ruído ambiente é a função de coerência. A função de coerência de um sinal em dois pontos diferentes *i* e *j* é definida pela equação (2.43).

Está demonstrado em [46], que a função complexa de coerência de um ambiente com ruído difuso é dada pela equação (2.44).

O desempenho da rede de microfones que se utiliza da pós-filtragem será melhor quanto mais próximo da realidade for o modelo do ruído ambiente utilizado.

Considerando-se que os sinais estejam alinhados no tempo em todos os sensores, e que o sinal desejado e o ruído sejam não correlacionados, pode-se escrever que:

$$\phi_{x_i x_i} = \phi_{ss} + \phi_{v_i v_i} \tag{3.112}$$

$$\phi_{x_i x_j} = \phi_{ss} + \phi_{v_i v_j} \tag{3.113}$$

$$\phi_{x_i x_i} = \phi_{ss} + \phi_{v_i v_i} \tag{3.114}$$

$$\Gamma_{\nu_{i}\nu_{j}} = \frac{\phi_{\nu_{i}\nu_{j}}}{\sqrt{\phi_{\nu_{i}\nu_{i}}\phi_{\nu_{j}\nu_{j}}}} \,. \tag{3.115}$$

Considerando que $\phi_{x_i x_j} = \phi_{x_j x_j}$, se houver um modelo para a função de coerência Γ_{vv} , então as três equações acima formam um sistema com três variáveis desconhecidas: ϕ_{ss} , $\phi_{v_i v_i}$ e $\phi_{v_i v_j}$. Aplicando-se as técnicas padrões de álgebra, este sistema pode ser resolvido para se encontrar qualquer uma das variáveis. No entanto, para a função de transferência do pós-filtro mostrado na equação (3.108), apenas interessa estimar a variável ϕ_{ss} . Resolvendo-se o sistema para ϕ_{ss} [47], tem-se equação do segundo grau com solução:

$$\phi_{ss} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{3.116}$$

onde:

$$a = \Gamma_{\nu_i \nu_i}^2 - 1 \tag{3.117}$$

$$b = 2\phi_{x_i x_i} - \Gamma_{v_i v_i}^2 (\phi_{x_i x_i} + \phi_{x_i x_i})$$
(3.118)

$$c = \Gamma_{v_i v_i}^2 \phi_{x_i x_i} \phi_{x_i x_i} - \phi_{x_i x_i}^2. \tag{3.119}$$

A ambigüidade das duas soluções é resolvida verificando-se que uma das soluções tornará a função de coerência negativa e, portanto poderá ser descartada.

Com o objetivo de simplificar a resolução do sistema de equações anterior, pode-se fazer a consideração que o espectro de potência é o mesmo em todos os sensores [48]

$$\phi_{v_i v_i} = \phi_{vv} \quad \forall i . \tag{3.120}$$

Desta forma, a equação (3.118) pode ser escrita como:

$$\boldsymbol{\Gamma}_{v_i v_j} = \frac{\phi_{v_i v_j}}{\phi_{vv}}, \tag{3.121}$$

e as equações (3.112), (3.113) e (3.114) podem ser simplificadas para:

$$\phi_{x_i x_i} = \phi_{ss} + \phi_{vv} \,, \tag{3.122}$$

$$\phi_{\mathbf{x}_i \mathbf{x}_i} = \phi_{\mathbf{s}\mathbf{s}} + \phi_{\mathbf{v}\mathbf{v}} \ \mathbf{e} \tag{3.123}$$

$$\phi_{x_i x_j} = \phi_{ss} + \Gamma_{\nu_i \nu_j} \phi_{\nu \nu} . \tag{3.124}$$

Desta forma, a densidade espectral de potência do sinal desejado pode ser estimada como:

$$\hat{\phi}_{ss}^{ij} = \frac{\Re{\{\hat{\phi}_{x_i x_j}\}} - \frac{1}{2} \Re{\{\hat{\Gamma}_{v_i v_j}\}} (\hat{\phi}_{x_i x_i} + \hat{\phi}_{x_j x_j})}{(1 - \Re{\{\hat{\Gamma}_{v_i v_j}\}})}.$$
(3.125)

Da mesma forma que na técnica apresentada por Zelinsk, a robustez do estimador é melhorada tirando-se a média das soluções obtidas combinando-se todos os sensores, o que resulta:

$$\hat{\mathbf{w}}_{p} = \frac{\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Re{\{\hat{\phi}_{ss}^{ij}\}}}{\frac{1}{N} \sum_{i=1}^{N} \Re{\{\hat{\phi}_{x_{i}x_{i}}\}}}.$$
(3.126)

Os resultados da simulação numérica realizada no Matlab com a técnica da pósfiltragem estão descritos na seção 5.1.2.

4 IMPLEMENTAÇÃO DA REDE DE MICROFONES

Neste trabalho, a implementação da rede de microfones possui duas etapas bem distintas, que envolvem a adaptação da biblioteca de código aberto *Advanced Linux Sound Architecture* (ALSA) [49], utilizada como principal biblioteca de áudio no sistema operacional Linux, e a escolha de um *hardware* de baixo custo e eficiente.

4.1 TRABALHANDO COM A BIBLIOTECA ALSA

A biblioteca ALSA possui algumas características muito importantes e que justificam sua utilização neste trabalho. As principais características que se destacam são:

- ser desenvolvida com política de código aberto;
- estar disponível para todas as distribuições do sistema operacional Linux;
- possuir suporte para a grande maioria das interfaces de áudio, desde as mais simples até as interfaces profissionais com múltiplos canais e
- ser desenvolvida em linguagem C.

O ponto negativo da biblioteca é a documentação, que abrange apenas de forma superficial as características das suas funções.

O *software* empregado na rede de microfones foi desenvolvido tomando-se como base os programas *aplay* e *arecord*, pertencentes à biblioteca ALSA.

Os programas *aplay* e *arecord* são programas de linha de comando, cujas principais funções são reproduzir e gravar arquivos de áudio. Estes programas utilizam grande parte dos recursos disponíveis da biblioteca e possuem uma grande quantidade de parâmetros de configuração. Os principais parâmetros são os que permitem configurar:

- o dispositivo ou conjunto de dispositivos de áudio para entrada e saída;
- o número de canais a serem gravados ou reproduzidos;
- a taxa de amostragem do sinal;
- o formato do arquivo (wav, voc, raw ou au);

71

o número de bits e tipo de dado de cada amostra: 8, 16, 24, 32 ou 64 bits, com

dado inteiro ou de ponto flutuante; e

o período de gravação.

Os programas aplay e arecord também podem ter como parâmetro de linha de

comando o nome de um arquivo de áudio para leitura ou gravação. No caso do

programa arecord, se o nome do arquivo for omitido, os dados que seriam gravados

no arquivo serão direcionados para o standard out (stdout) do sistema operacional.

No caso do aplay, se o nome do arquivo for omitido, os dados que seriam lidos do

arquivo serão procurados no standard in (stdin) do sistema operacional. Estas

características destes dois programas, quando combinadas,

implementação de um sistema de entrada e saída de áudio em tempo real, pois tudo

que é lido pelo *arecord* é reproduzido pelo *aplay*, quase que instantaneamente.

Exemplo: arecord <parâmetros> | aplay <parâmetros>

No sistema operacional Linux, o operador "|" (lê-se pipe) direciona a saída de um

programa para a entrada do outro através do uso dos buffers stdout e stdin. No caso

do exemplo anterior, a saída do arecord está direcionada para a entrada do aplay.

Por default, o programa arecord coloca em sua saída o mesmo número de canais de

sua entrada. Para o esquema de uma rede de microfones, esta característica precisou

ser modificada, pois neste caso existem múltiplos canais de entrada e apenas um

canal de saída. Para a solução deste problema, foi criado o parâmetro de linha de

comando --mic-array=<filtro>, onde <filtro> pode ser um dos filtros espaciais

implementados: gsc, post-filter ou delay-and-sum (com esta grafia).

Assim, após as implementações realizadas, para utilizar a rede de microfones é

necessário utilizar o seguinte comando no prompt do Linux:

arecord --mic-array=<filtro> <parâmetros> | aplay <parâmetros>

A codificação em linguagem C dos algoritmos GSC, *Post-Filtering* e *Delay and Sum* não produziu nenhum efeito colateral no funcionamento ou desempenho dos programas *aplay* e *arecord*, e da mesma forma nos demais programas da biblioteca ALSA.

Um dos objetivos deste trabalho é fornecer a base necessária para a continuidade do desenvolvimento dos algoritmos aplicados nas redes de microfones no ambiente Linux de código aberto. Seguindo esta diretriz, as implementações realizadas no código-fonte foram extensamente documentadas. Procurou-se também simplificar a parte do programa que interage com os algoritmos das redes de microfones. A tabela a seguir mostra a parte do código-fonte que ilustra esta idéia.

Tabela 4.1 Chamada dos algoritmos de rede de microfones no código-fonte

```
if (mic_array) { //mic-array:
    if (sf == SF_POST_FILTER) {
        Post_Filter(audiobuf, audiobuf_ma, f);
    }
    if (sf == SF_GSC) {
        GSC(audiobuf, audiobuf_ma, f);
    }
    if (sf == SF_DELAY_AND_SUM) {
        delay_and_sum(audiobuf, audiobuf_ma, f);
    }
    if (write(fd, audiobuf_ma, c / rhwparams.channels) !=
        c / rhwparams.channels) { //mic-array:
        perror(name);
        exit(EXIT_FAILURE);
    }
}
```

A Tabela 4.1 mostra a chamada dos algoritmos de rede de microfones no códigofonte. Deve-se observar que, para todos os algoritmos, os parâmetros passados por suas respectivas funções foram *audiobuf*, *audiobuf ma* e *f*, onde:

 audiobuf é o ponteiro do buffer com os múltiplos canais de entrada. Os dados dos canais de entrada estão intercalados, formando a seguinte seqüência:
 c0[0] c1[0] c2[0] ... cN[0] c0[1] c1[1] c2[1] ... cN[1] ...

- audiobuf_ma é o ponteiro do buffer do canal de saída que terá o dado já tratado pelo filtro espacial.
- f é o parâmetro que indica o número de frames amostrados a serem tratados pelo filtro espacial.

A alocação de memória dos buffers *audiobuf* e *audiobuf_ma* é realizada previamente. Os demais parâmetros, como freqüência de amostragem, número de canais e período total de amostragem, são passados por variáveis globais que já estavam disponibilizadas no programa original.

Embora tenham aplicações distintas, os programas *aplay* e *arecord* possuem o mesmo código-fonte, sendo que as funcionalidades de cada um são selecionadas apenas pelo nome do programa. As modificações realizadas no código-fonte destes programas apenas acrescentaram novas funcionalidades, sem alterar qualquer característica previamente existente.

O código-fonte dos programas com a implementação da rede de microfones está disponível em http://www. fcth.br/mic-array

4.2 HARDWARE UTILIZADO

O hardware utilizado nas redes de microfones pode ser considerado um dos grandes limitadores na popularização desta tecnologia. Normalmente, para as aplicações em tempo real, faz-se necessária a utilização de hardware específico e geralmente de custo elevado. Para o desenvolvimento deste trabalho, foi utilizado o critério de baixocusto para a escolha do hardware. A rede de microfones foi implementada em um computador pessoal (PC) com sistema operacional Linux.

A escolha do dispositivo de áudio foi mais difícil, pois para uma rede de microfones é necessária a utilização de placas com múltiplos canais de entrada, o que geralmente leva a componentes de alto custo. A solução encontrada para este problema foi a escolha de dispositivos de áudio USB como o mostrado na Figura 4.1. Este é um dispositivo compatível com USB 2.0 *full-speed* com velocidade de transferência de dados de até 12 Mbps, que não necessita de alimentação externa e não requer *driver* de instalação. Possui um plug de entrada mono e um plug de saída estéreo. O custo

unitário deste dispositivo de áudio é de aproximadamente R\$ 25,00. Como este elemento possui apenas uma entrada, são necessários tantos dispositivos de áudio quantos forem o número de canais da rede de microfones.



Figura 4.1 Dispositivo de áudio USB

Portanto, para cada canal da rede de microfones é necessária uma porta USB do microcomputador. Para redes com muitos canais, um hub USB pode ser utilizado.

A rede de microfones montada para este trabalho é mostrada na Figura 4.2 e Figura 4.3. Disponde de 4 canais (4 microfones), cada um ligado a um dispositivo USB de áudio. Os dispositivos de áudio utilizaram um cabo extensor USB para serem conectados ao laptop. Não foi necessária a utilização de hub USB, porque o *laptop* utilizado possuía 4 as portas. Para este estudo, os microfones não foram fixados a fim de facilitar as diferentes configurações de espaçamento entre eles.



Figura 4.2 Rede de microfones com 4 canais ligados a um laptop – vista de perto

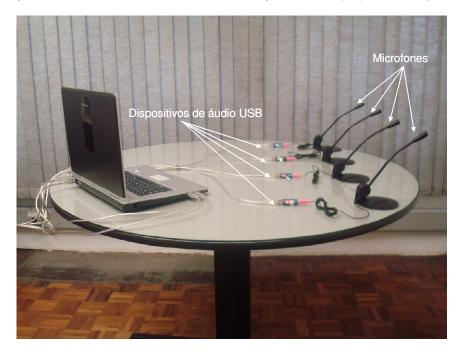


Figura 4.3 Rede de microfones com 4 canais ligados a um laptop – vista lateral

Através do parâmetro de linha de comando -l (list) dos programas *aplay* ou *arecord* pode-se listar os dispositivos de áudio conectados ao computador. No caso da rede mostrada na Figura 4.2 e Figura 4.3, o comando exibe o dispositivo de áudio interno do laptop, mais os 4 outros dispositivos USB externos, conforme mostra a tabela abaixo:

Tabela 4.2: Lista dos dispositivos de áudio conectados no PC

Card 0: Intel [HDA Intel]...

Card 1: [USB AUDIO]...

Card 2: [USB AUDIO]...

Card 3: [USB AUDIO]...

Card 4: [USB AUDIO]...

A Tabela 4.2 mostra todos os dispositivos de áudio conectados no PC. Embora exista número suficiente de dispositivos de áudio para se montar uma rede com 4 canais, todos os dispositivos listados funcionam de maneira independente e não sincronizada. Para aplicações de rede de microfones, o sincronismo dos canais é fundamental. A cada ciclo de amostragem que o programa *arecord* fizer, deverão ser lidos todos os dispositivos de entrada simultaneamente. Para solucionar este problema, o ALSA permite criar um dispositivo virtual formado pelo agrupamento dos dispositivos individuais, fazendo com que seu comportamento seja igual ao de um único dispositivo com todos os recursos agrupados. Para isto é necessário criar ou editar, caso já exista, um arquivo de configuração do ALSA chamado *.asoundrc* localizado em /etc/ ou /home/<user> no Linux. Se o arquivo for criado no /etc/, as funcionalidades serão estendidas a todos os usuários do computador; caso seja criado no /home/<user>, serão válidas apenas para o <user>.

As configurações colocadas no arquivo *.asoundrc* podem acrescentar um grande número de funcionalidades aos dispositivos de áudio. Neste caso, deseja-se agrupar todos os dispositivos externos em um único dispositivo. A Tabela 4.3 mostra esta configuração:

Tabela 4.3 Arquivo .asoundrc para agrupamento dos dispositivos de áudio

```
pcm.array4 {
  type multi;
  slaves.a.pcm "hw:1,0";
  slaves.a.channels 1;
  slaves.b.pcm "hw:2,0";
  slaves.b.channels 1;
  slaves.c.pcm "hw:3,0";
  slaves.c.channels 1;
  slaves.d.pcm "hw:4,0";
  slaves.d.channels 1;
  bindings.0.slave a;
  bindings.0.channel 0;
  bindings.1.slave b;
  bindings.1.channel 0;
  bindings.2.slave c;
  bindings.2.channel 0;
  bindings.3.slave d;
  bindings.3.channel 0;
}
ctl.array4 {
  type hw;
  card 0;
}
```

No caso da Tabela 4.3, foi criado um dispositivo virtual chamado **array4** com as funcionalidades dos dispositivos Card 1, representado por "hw:1,0" até Card 4 representado por "hw:4,0" e que se comportará como um dispositivo único de 4 canais de entrada.

Para a realização destes arranjos, outros dispositivos de áudio, internos ou externos, também podem ser utilizados, devendo ser verificadas as limitações da freqüência de amostragem, que podem interferir no sincronismo do agrupamento. Para o agrupamento mostrado anteriormente, foram testados com sucesso freqüências de amostragem de até 44100 Hz.

Foi dada especial atenção para a escolha dos microfones. Normalmente, em uma rede de microfones, o locutor está distante da rede, e isto requer que os microfones sejam de grande sensibilidade. Microfones de baixo custo tendem a não possuir esta característica. Este fator pode limitar a distância máxima do locutor. No âmbito deste trabalho foram testadas distâncias de até 2,5m entre o locutor e a rede de microfones.

4.3 LIMITAÇÕES DO SISTEMA PROPOSTO

As limitações encontradas no sistema proposto para a implementação de uma rede de microfones de baixo custo podem ser classificados em limitações de *software* e *hardware*.

A maior limitação com o *software* foi a documentação da biblioteca ALSA, que se apresenta bastante superficial, tornando difícil o entendimento da configuração dos dispositivos de áudio. De acordo com o grupo de desenvolvimento desta biblioteca, este é um problema reconhecido, e que deverá ter uma solução no médio prazo.

Com o *hardware*, a maior limitação foi verificada no nível de ruído acrescentado pelo dispositivo de áudio USB ao sinal amostrado. Este ruído limita a distância máxima que o locutor pode ter em relação à rede de microfones, pois à medida que ele se afasta a intensidade do sinal captado pelos microfones diminui. Como a intensidade do ruído aditivo é constante, a SNR diminui também.

Outro problema encontrado foi com o posicionamento dos dispositivos de áudio nas portas USB do PC. As dimensões do dispositivo de áudio não permitem que fiquem lado a lado, conforme mostrado na Figura 4.4. A solução encontrada para este problema foi a retirada dos invólucros dos dispositivos de áudio, conforme mostrado na Figura 4.5.



Figura 4.4 Dispositivos de áudio mal conectados às portas USB

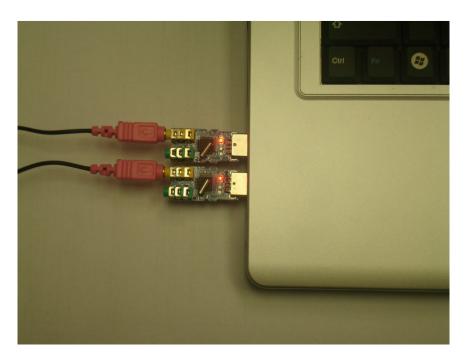


Figura 4.5 Dispositivos de áudio conectados perfeitamente às portas USB

Outra forma de se resolver este problema é a utilização de cabos extensores USB, porém, em alguns casos, esta medida pode levar a um aumento no nível de ruído do sinal de entrada do dispositivo de áudio.

Outro problema encontrado diz respeito à variação no sequenciamento dos dispositivos de áudio conectados ao PC. Como mostrado na Tabela 4.3, o arquivo .asoundrc, que configura o dispositivo virtual de múltiplos canais, necessita da posição exata dos dispositivos para atribuir cada canal a seu respectivo dispositivo de áudio, conforme mostrado na Tabela 4.2. Esta variação no seqüenciamento, sem padrão identificado, ocorre a cada reinício do sistema operacional, fazendo com que a ordem dos canais seja alterada. A solução adotada para este problema foi a de somente conectar os dispositivos de áudio após o início do sistema operacional, e na mesma seqüência dos números dos canais. Uma alternativa seria utilizar os dados fornecidos pelos comandos aplay ou arecord com a opção -l para listar os dispositivos e suas características como nome do fabricante, modelo ou versão. Com estes dados seria possível editar o arquivo .asoundrc e fixar suas posições, porém, no caso de aplicações como a deste trabalho, é fundamental que os dispositivos sejam iguais (de mesmos fabricantes, modelos e versões) para evitar diferenças significativas entre canais durante a amostragem dos sinais. Este é um problema recorrente entre os usuários da biblioteca ALSA, cuja solução definitiva ainda está sendo estudada.

5 EXPERIMENTOS E RESULTADOS

Para a avaliação do sistema proposto, foram realizados ensaios em condições reais de uso da rede de microfones em tempo real, testando os algoritmos *Post Filtering* e *GSC*. Com o objetivo de facilitar esta avaliação, e fornecer parâmetros de comparação, primeiro foram realizadas simulações no Matlab destes algoritmos. A seção 5.1 mostra os testes realizados no Matlab e a seção 5.2 mostra os testes realizados com o sistema proposto.

5.1 EXPERIMENTOS REALIZADOS NO MATLAB

Para a realização destes experimentos foram utilizadas uma rede com 4 e uma rede com 8 microfones de eletreto dispostos em linha, sempre com espaçamento uniforme de 16 cm. Os sinais de áudio foram amostrados por uma placa A/D da Advantec e software LabView da National Instruments. Para todos os ensaios a freqüência de amostragem foi de 16 kHz.

A sala onde foram realizados os experimentos possui as seguintes dimensões: 4,3m de largura, 5,7m de comprimento e 2,8m de altura, e não possui nenhuma proteção contra reverberação. A disposição das fontes sonoras é mostrada na figura a seguir.

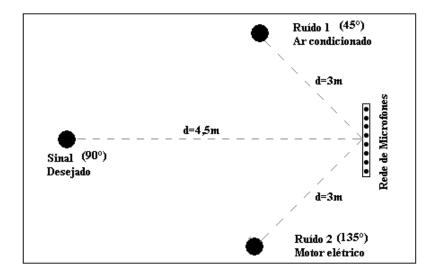


Figura 5.1 Disposição das fontes sonoras utilizadas nos experimentos

Os algoritmos foram escritos em Matlab versão 6 e rodados em um PC com processador Intel Pentium 4 com *clock* de 2,8 GHz.

Foram analisados os algoritmos *Post-Filtering* e dois algoritmos para o GSC, o LAF-LAF e o IIR. A fim de facilitar a implementação dentro da biblioteca ALSA, o desenvolvimento prévio destes algoritmos em Matlab buscou manter um compromisso entre a qualidade dos resultados e sua complexidade computacional, pois deveriam funcionar em tempo real.

5.1.1 Critérios de Avaliação

As avaliações foram realizadas através de critérios objetivos, e são descritas a seguir:

- Medição da atenuação do sinal de saída da rede com o sinal de entrada composto por um sinal limpo (sinal de interesse) e dois sinais ruidosos, conforme mostrado na Figura 5.1 . Para estas medições foram usadas duas razões sinal-ruído, SNRin=3dB e SNRin=10dB.
- Comparações entre as amostras no domínio do tempo do sinal de interesse e da saída da rede de microfones. Para estas medições foram usadas duas razões de sinal-ruído, SNRin=3dB e SNRin=10dB.
- Comparações entre as densidades espectrais de potência do sinal desejado, do sinal desejado com ruído e do sinal de saída da rede de microfones. Neste caso também foram testadas as situações com SNRin=3dB e SNRin=10dB.
- Todas as situações acima foram testadas em redes com 8 e com 4 microfones.
- Por último, foi realizado o teste de diretividade da rede através do desalinhamento dos sinais de entrada dos microfones. Este desalinhamento foi produzido aplicando-se atrasos proporcionais à distância entre a fonte de sinal e cada um dos microfones.

5.1.2 Análise dos Resultados do Post-Filtering

Para o *Post-Filtering* foram testados dois algoritmos, um baseado no *Post-Filtering* de Zelinsk descrito na seção 3.4.3 e outro baseado no *Post-Filtering* para ruído difuso descrito na seção 3.4.4. Para todos os testes realizados, os resultados destes dois algoritmos são muito parecidos, porém, por ter uma complexidade computacional menor, o *Post-Filtering* de Zelinsk roda em média cerca de 20% mais rápido.

Os testes realizados utilizaram quadros de 512 amostras. Para evitar o efeito indesejado de pequenos estalos no sinal de saída que ocorre devido a variações bruscas de energia de um quadro para outro, foi utilizado o recurso da sobreposição de quadro. Neste caso, foi adotada a sobreposição de 50%.

Para a condição com 8 microfones, o algoritmo do *Post-Filtering* de Zelinsk leva em média 1,6 s para cada segundo de sinal processado. Deve-se considerar que o código escrito em Matlab não foi otimizado em favor da velocidade de processamento.

Analisando as informações da Figura 5.2, que mostra os resultados do algoritmo *Post-Filter* com uma rede com 8 microfones e uma SNR de 3 dB, sendo o sinal e o ruído dispostos de acordo com a Figura 5.1, verifica-se que a filtragem espacial eliminou boa parte do ruído. Esta afirmação fica evidente quando se compara a saída da rede com o sinal de entrada em um dos microfones. Também pode-se observar que o espectro de potência do sinal de saída do *Post-Filter* se aproxima bastante do espectro do sinal original.

Na Figura 5.3, são mostrados os resultados da mesma análise anterior, porém para uma SNR de 10 dB. Neste caso, os resultados foram um pouco melhores que o anterior. Pode-se observar que a DEP do sinal obtido na saída do *Post-Filter* se assemelha bastante à DEP do sinal original. De forma subjetiva, verifica-se que a qualidade do áudio tem um enorme ganho ao se aplicar este tipo de filtro.

Na Figura 5.4 são mostrados os resultados da simulação realizada com o *Post-Filter* com 4 microfones e SNR de 3 dB. Os resultados obtidos mostram que o filtro consegue eliminar boa parte do ruído.

Quando se comparam os resultados do *Post-Filter* com 4 microfones, mostrados na Figura 5.4, e com 8 microfones, mostrados na Figura 5.2, nota-se que a diferença é

pequena a favor da situação com 8 microfones. Subjetivamente, também pode-se notar que o sinal da situação com 8 microfones possui menos ruído.

Post-Filter, 8 microfones e SNR=3dB

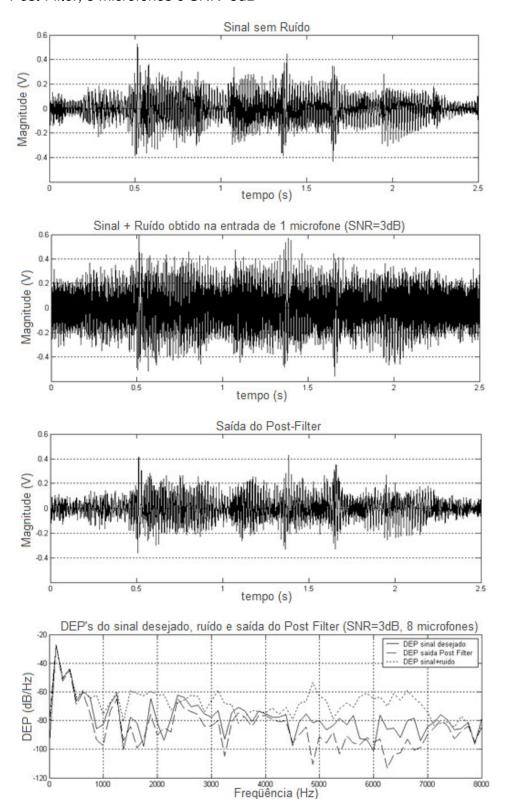


Figura 5.2 Resultados do *Post-Filter* (SNR=3 dB, 8 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do *Post-Filter* (d) Comparação entre as DEP's

Post-Filter, 8 microfones e SNR=10dB

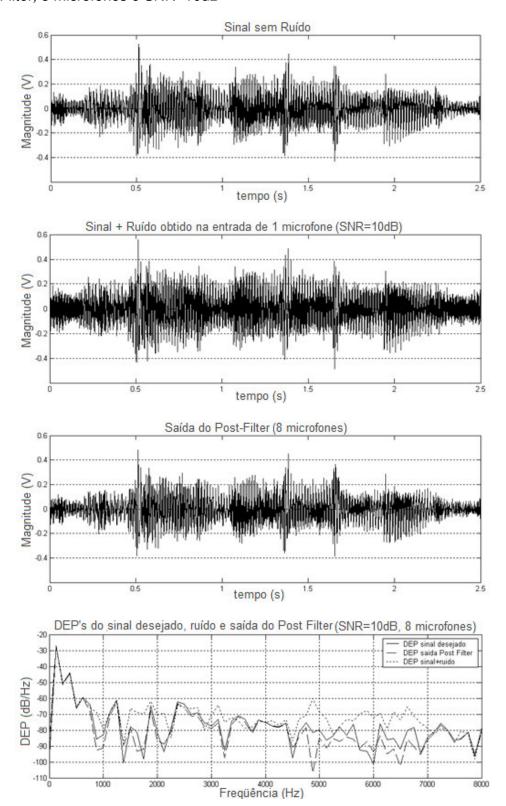


Figura 5.3 Resultados do *Post-Filter* (SNR=10 dB, 8 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído (SNR=10 dB), (c) Saída do *Post-Filter* (d) Comparação entre as DEP's

Post-Filter, 4 Microfones e SNR=3dB

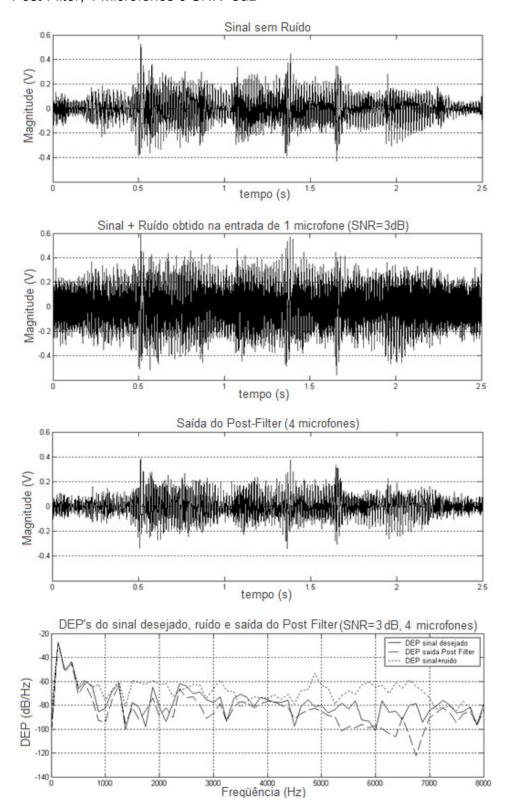


Figura 5.4 Resultados do *Post-Filter* (SNR=3 dB, 4 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído (SNR=3 dB, (c) Saída do *Post-Filter* (d) Comparação entre as DEP's

A seguir, é mostrado o resultado do ensaio de diretividade do *Post-Filter* com 8 microfones.

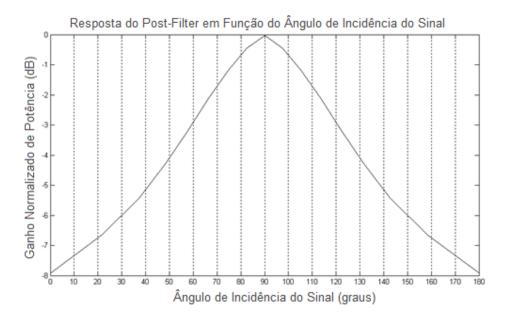


Figura 5.5 Resposta do Post-Filter em função do ângulo de incidência do sinal

A figura anterior mostra o comportamento da diretividade do *Post-Filter* no arranjo com 8 microfones. Os resultados obtidos mostram uma boa seletividade espacial dos sinais que chegam à rede, porém, é desejável que exista uma pequena faixa de 10 a 15º em torno do ângulo de 90º em que a atenuação seja pequena. Isto permite que pequenos deslocamentos do locutor não produzam grandes atenuações.

5.1.3 Análise dos Resultados do GSC LAF-LAF

Para o GSC LAF-LAF foi testado o algoritmo descrito na seção 3.3.3.

Os testes realizados utilizaram filtros com 32 coeficientes nos LAFs da matriz de bloqueio e 32 coeficientes nos LAFs do cancelador de múltiplas entradas. O tempo de processamento de cada segundo de sinal real foi de cerca de 5,1 s. Deve-se considerar que o código escrito em Matlab não foi otimizado em favor da velocidade de processamento.

A Figura 5.6 mostra os resultados da filtragem espacial utilizando-se o algoritmo GSC LAF-LAF com 8 microfones e SNR de 3 dB. O sinal e o ruído estão dispostos de acordo com a Figura 5.1. Da mesma forma que os resultados apresentados pelo *Post-Filter*, o GSC também produziu bons resultados, pois conseguiu eliminar boa pare do ruído captado pelos microfones individualmente. A comparação entre as DEP também mostra que o espectro do sinal obtido na saída do GSC se aproxima bastante do espectro do sinal original.

Na Figura 5.7 é mostrada a mesma análise do GSC LAF-LAF, porém com 4 microfones. Neste caso, houve uma pequena atenuação do sinal obtido na saída do GSC, mas que não compromete o resultado final. Avaliando-se o algoritmo em função do número de microfones, os resultados mostram que o GSC foi bem mais sensível à redução no número de microfones que o *Post-Filter*. Em análise subjetiva também se tem a mesma avaliação.

GCS LAF-LAF, 8 microfones e SNR=3dB

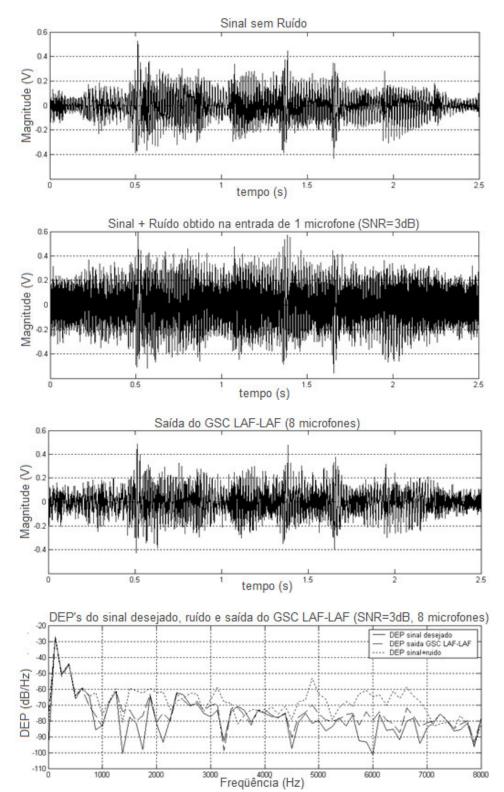


Figura 5.6 Resultados do GSC LAF-LAF (SNR=3 dB, 8 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do GSC (d) Comparação entre as DEP's

GSC LAF-LAF, 4 microfones e SNR=3dB

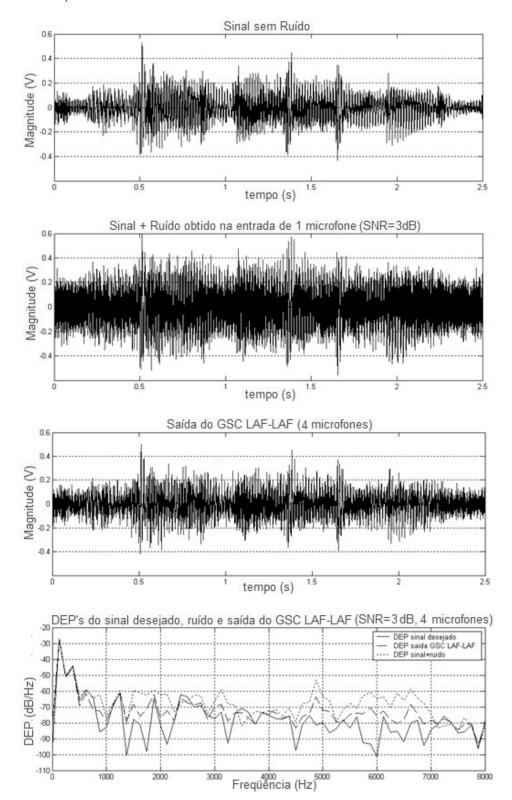


Figura 5.7 Resultados do GSC LAF-LAF (SNR=3 dB, 4 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do GSC (d) Comparação entre as DEP's

A seguir, é mostrado o resultado do ensaio de diretividade do GSC LAF-LAF com 8 microfones.

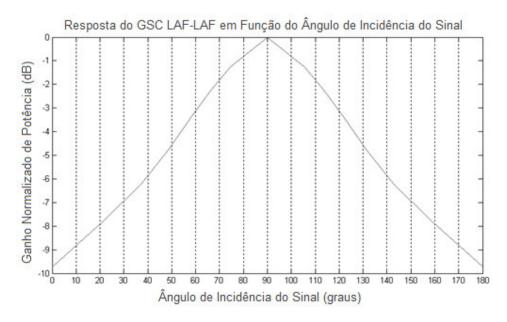


Figura 5.8 Resposta do GSC LAF-LAF em função do ângulo de incidência do sinal

Os resultados da diretividade, apresentados na Figura 5.8, mostram que o comportamento do GSC LAT-LAF é muito parecido com o do *Post-Filter*.

5.1.4 Análise dos Resultados do GSC IIR

Para o GSC IIR foi testado o algoritmo descrito na seção 3.3.4.

Os testes realizados utilizaram filtros com 8 coeficientes no numerador e 8 coeficientes no denominador. O tempo de processamento de cada segundo de sinal real levou cerca de 36 s. Deve-se considerar que o código escrito em Matlab não foi otimizado em favor da velocidade de processamento.

A Figura 5.9 mostra os resultados da filtragem espacial utilizando-se o algoritmo GSC IIR com 8 microfones e SNR de 3 dB. O sinal e o ruído estão dispostos de acordo com a Figura 5.1. Da mesma forma que os algoritmos anteriores, o GSC IIR também produziu bons resultados, pois conseguiu eliminar boa parte do ruído captado pelos microfones individualmente. A comparação entre as DEP's também mostra que o

espectro do sinal obtido na saída do GSC IIR se aproxima bastante do espectro do sinal original.

Na Figura 5.10 é mostrada a mesma análise do GSC IIR, porém com 4 microfones. Nesta situação, pode-se observar que o sinal obtido na saída do filtro é um pouco mais contaminado pelo ruído que na situação com 8 microfones. Em análise subjetiva também se tem a mesma avaliação.

GSC IIR, 8 microfones e SNR=3 dB

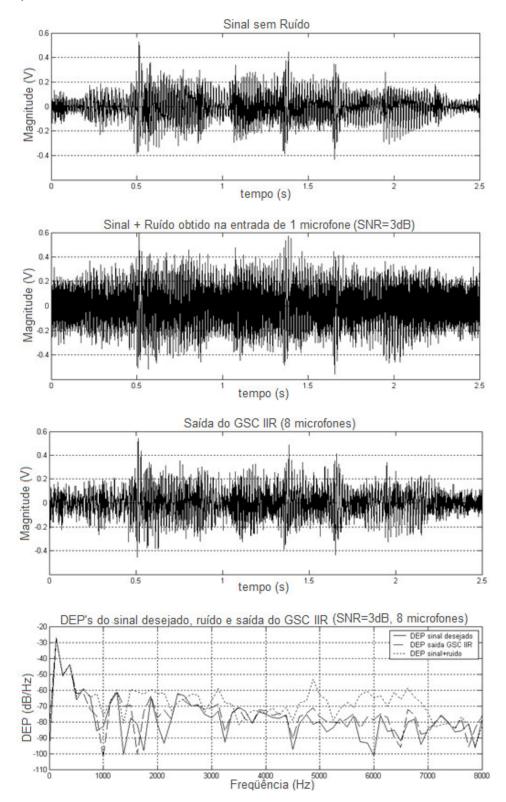


Figura 5.9 Resultados do GSC IIR (SNR=3 dB, 8 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do GSC IIR (d) Comparação entre as DEP's

GSC IIR, 4 microfones e SNR=3 dB

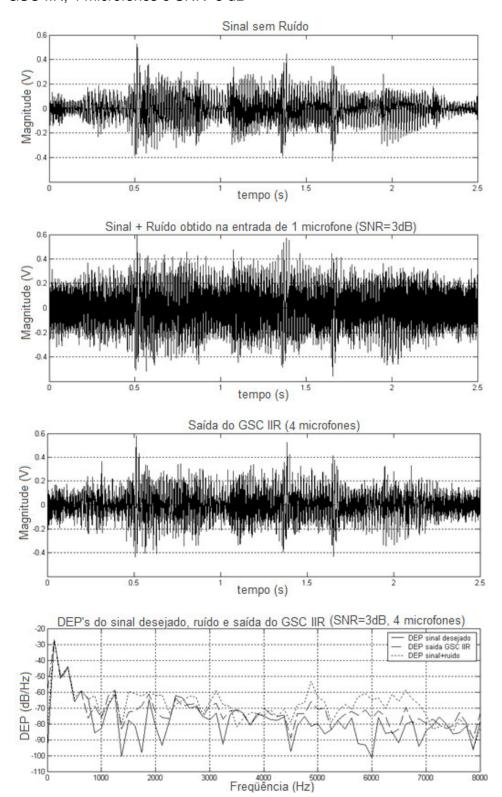


Figura 5.10 Resultados do GSC IIR (SNR=3 dB, 4 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído (SNR=3 dB), (c) Saída do GSC IIR (d) Comparação entre as DEP's

A seguir, é mostrado o resultado do ensaio de diretividade do GSC IIR com 8 microfones.

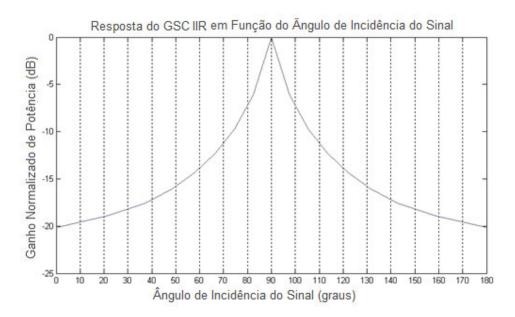


Figura 5.11 Resposta do GSC IIR em função do ângulo de incidência do sinal

Os resultados da diretividade apresentados na Figura 5.11 mostram que o GSC IIR produz maiores atenuações nos sinais que não se originam perpendicularmente a rede (*look direction*). Esta característica é muito útil para os casos em que se deseja uma alta seletividade espacial do sinal.

5.2 EXPERIMENTOS REALIZADOS NO SISTEMA PROPOSTO

Os resultados mostrados a seguir não têm o objetivo de mostrar a eficiência dos filtros espaciais, e sim mostrar que o arranjo de baixo custo proposto pode ser uma boa alternativa para a aplicação de redes de microfones funcionando em tempo real.

Os testes foram realizados em uma sala de escritório com a ocorrência de ruídos de ar-condicionado e do ambiente externo. Foi utilizado um arranjo como o mostrado na Figura 4.2, com 4 microfones com espaçamento uniforme de 16 cm. A freqüência de amostragem utilizada foi de 16 kHz. A fonte do sinal desejado foi colocada a 2,5 m do arranjo em um ângulo de 90° (*look direction*).

O comando utilizado para a coleta de dados com a rede de microfones proposta foi:

arecord --mic-array=<filtro> -D plug:array4 -c 4 -r 22000 -t wav -f S16_LE -d 10 <nome do arquivo>

onde:

- --mic-array=<filtro> ativa o modo de rede de microfones e seleciona o algoritmo do filtro a ser utilizado,
- -D plug:array4: informa que o dispositivo de gravação é o array4 que é um dispositivo virtual composto por 4 dispositivos USB,
- -c 4: informa o numero de canais a serem gravados: 4,
- -r 22000: informa a freqüência de amostragem da gravação: 22 kHz,
- -t wav: informa o formato do arquivo de áudio: wav,
- -f S16_LE: informa o tipo de dado de cada amostra: *short integer* de 16 bits (no formato *little endian*),
- -d 10: informa o tempo de gravação, neste caso 10s,
- <nome do arquivo>: informa o nome do arquivo wav a ser gravado com o resultado da filtragem espacial realizada pelo algoritmo selecionado.

A disposição das fontes sonoras é mostrada na figura a seguir.

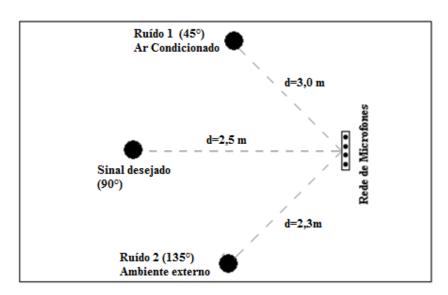


Figura 5.12 Disposição das fontes sonoras utilizadas nos experimentos com o arranjo proposto

Os algoritmos foram escritos em Linguagem C e rodados em um PC com processador Intel Pentium 4 com *clock* de 2,8 GHz.

Foram analisados os algoritmos que obtiveram melhor desempenho no Matlab, quais sejam: *Post-Filtering*, GSC LAF-LAF e o *Delay-and-Sum*.

Apenas para avaliar os limites do arranjo, foi testada uma rede com 16 dispositivos de áudio ligados através de 4 *hubs* USB com 4 portas cada. O sistema rodou normalmente sem apresentar interrupções de amostragens.

Alguns cuidados prévios foram tomados para a realização destes experimentos. Com relação ao hardware, procurou-se obter componentes como adaptadores e microfones de um mesmo fabricante e também de um mesmo lote. Esta medida foi necessária, pois, no caso dos microfones foram constatadas diferenças em sua sensibilidade, e no caso dos adaptadores de áudio, pôde-se verificar a diferenças no nível de ruído introduzido.

5.2.1 Critérios de Avaliação

As avaliações foram realizadas através de critérios objetivos, e são descritas a seguir:

- Comparações entre amostras no domínio do tempo do sinal de interesse e da saída da rede de microfones.
- Comparações entre as densidades espectrais de potência do sinal desejado, do sinal desejado com ruído e do sinal de saída da rede de microfones.
- Por último foi realizado o teste de diretividade da rede através da variação do ângulo de incidência do sinal desejado. Foram realizadas variações de 15 em 15° percorrendo de 0 a 90°. Os gráficos dos testes realizados mostram que o comportamento da diretividade da rede é simétrico ao seu eixo de 90°, ou seja, ela possui a mesma resposta com ângulos de incidência de 60° e de 120°.

5.2.2 Análise dos Resultados do Post-Filtering

Para o *Post-Filtering*, foi testado o algoritmo baseado no *Post-Filtering* de Zelinsk descrito na seção 3.4.3. Os testes realizados utilizaram quadros de 512 amostras sem

sobreposição. Para efeito de comparação, alguns testes também foram realizados com sobreposição de quadros na ordem de 50%. Como não houve diferença perceptível no resultado, optou-se por prosseguir os testes sem a utilização da sobreposição.

Analisando as informações da Figura 5.13, que mostra os resultados do algoritmo *Post-Filter* com uma rede com 4 microfones, verifica-se que a filtragem espacial produz uma sensível melhora na razão sinal-ruído. Esta afirmação fica evidente quando se compara a saída da rede com o sinal de entrada em um dos microfones. A densidade espectral de potência, ilustrada na Figura 5.13 (d), mostra que, para as médias e altas freqüências, o espectro da saída acompanha o espectro do sinal desejado, porém em algumas freqüências podem-se observar distorções. A análise subjetiva também mostra que o sinal obtido na saída do filtro possui sensível redução no ruído comparado ao sinal obtido no 1º microfone.

Post-Filter, 4 microfones:

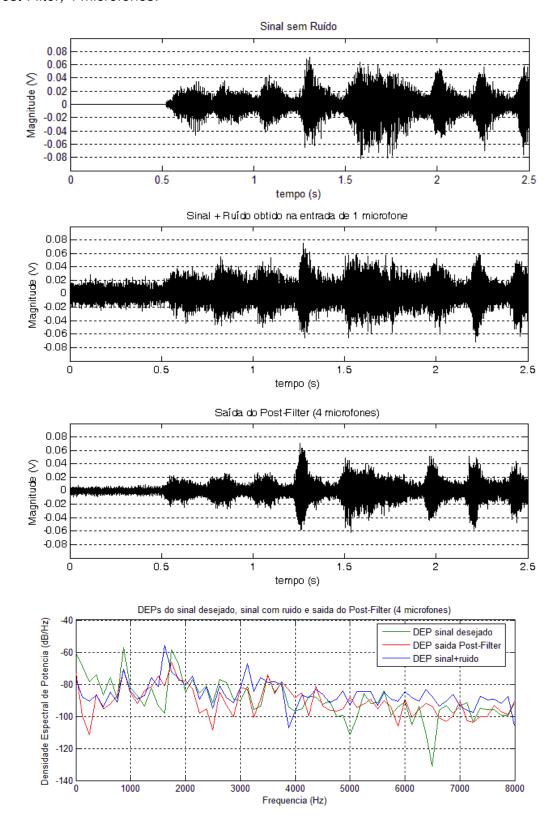


Figura 5.13 Resultados do *Post-Filter* (4 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído, (c) Saída do *Post-Filter* (d) Comparação entre as DEP's

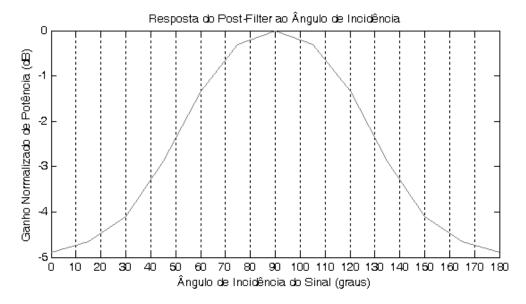


Figura 5.14 Resposta do Post-Filter em função do ângulo de incidência do sinal

A figura anterior mostra o comportamento da diretividade do *Post-Filter* no arranjo proposto. Comparando-se este resultado com o obtido pelas análises numéricas realizadas no Matlab (Figura 5.5), verifica-se que a diretividade do arranjo proposto apresentou bons resultados, embora com atenuações inferiores em todas as direções. Esta diferença é explicada pela diferença no número de microfones. O arranjo proposto foi ensaiado com 4 microfones, ao passo que o arranjo ensaiado no Matlab possuía 8.

Alguns trabalhos [47][48] mostram que melhores resultados podem ser obtidos, no entanto é necessário possuir um bom modelo para o ruído ambiente.

5.2.3 Análise dos Resultados do GSC LAF-LAF

O algoritmo GSC LAF-LAF testado está descrito na seção 3.3.3.

Os testes realizados utilizaram filtros com 32 coeficientes nos LAFs (*Leaky Adaptive Filters*) da matriz de bloqueio e 32 coeficientes nos LAFs do cancelador de múltiplas entradas.

A Figura 5.15 mostra os resultados da filtragem espacial no arranjo proposto utilizando-se o algoritmo GSC LAF-LAF com 4 microfones. O sinal e o ruído estão dispostos de acordo com a Figura 5.12.

Analisando os resultados, verifica-se que o filtro proposto elimina boa parte do ruído. A análise subjetiva também mostra que o áudio na saída do filtro possui bem menos ruído que o áudio do 1º microfone. Da mesma forma que no *Post-Filter*, a densidade espectral de potência, ilustrada na Figura 5.15 (d), mostra que, nas médias e altas freqüências, o espectro da saída acompanha o espectro do sinal desejado, porém em algumas freqüências podem-se observar distorções.

GSC LAF-LAF, 4 microfones:

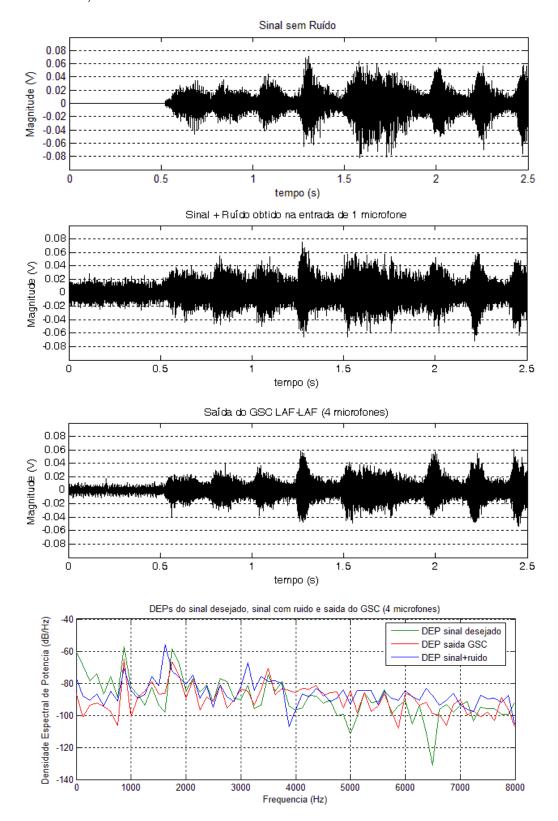


Figura 5.15 Resultados do GSC LAF-LAF (4 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído, (c) Saída do GSC (d) Comparação entre as DEP's

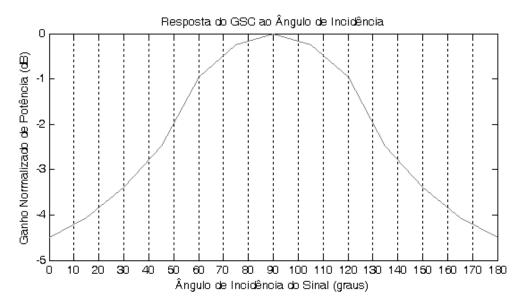


Figura 5.16 Resposta do GSC LAF-LAF em função do ângulo de incidência do sinal

A figura anterior mostra o comportamento da diretividade do GSC LAF-LAF no arranjo proposto. Comparando-se este resultado com o obtido pelas análises numéricas realizadas no Matlab (Figura 5.8), verifica-se que a diretividade do arranjo proposto apresentou atenuações inferiores em todas as direções. Esta diferença é explicada pelo número de microfones utilizados nos dois casos. O arranjo proposto foi ensaiado com 4 microfones ao passo que o arranjo ensaiado no Matlab possuía 8.

Em [38] são apresentadas algumas alternativas para melhorar o desempenho deste algoritmo. As alternativas apresentadas sugerem o controle do crescimento dos coeficientes dos filtros adaptativos.

5.2.4 Análise dos Resultados do Delay and Sum

O algoritmo *Delay and Sum* testado está descrito na seção 3.1.2.

A Figura 5.17 mostra os resultados da filtragem espacial no arranjo proposto utilizando-se o algoritmo *Delay and Sum* com 4 microfones. O sinal e o ruído estão dispostos de acordo com a Figura 5.12.

Analisando os resultados, verifica-se que o filtro proposto elimina boa parte do ruído, porém com desempenho pior que os algoritmos *Post-Filter* e GSC LAF-LAF. A análise

subjetiva mostra que o áudio na saída do filtro possui bem menos ruído que o áudio do 1º microfone. Da mesma forma que os algoritmos analisados anteriormente, a densidade espectral de potência, ilustrada na Figura 5.17 (d), mostra que para as médias e altas freqüências, o espectro da saída acompanha o espectro do sinal desejado, porém em algumas freqüências podem-se observar distorções.

Delay-and-Sum, 4 microfones:

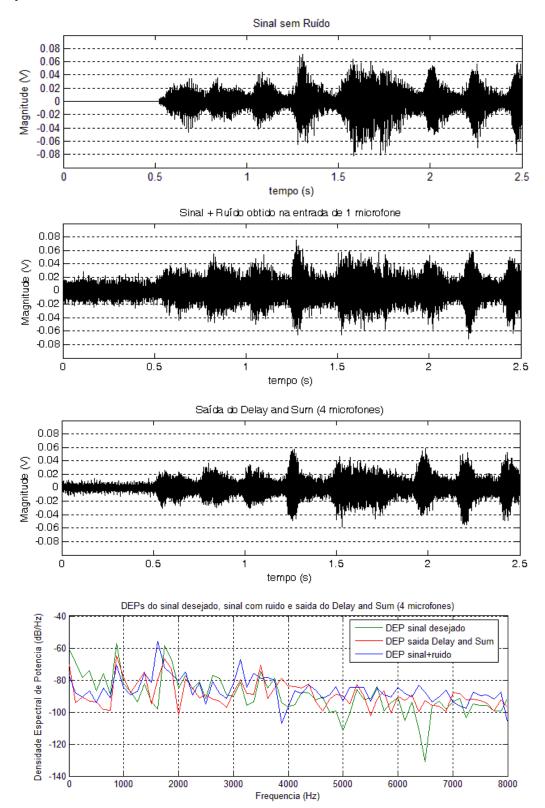


Figura 5.17 Resultados do *Delay and Sum* (4 microfones) - (a) Sinal desejado, (b) 1º microfone com sinal desejado + ruído, (c) Saída do *Delay and Sum* (d) Comparação entre as DEP's

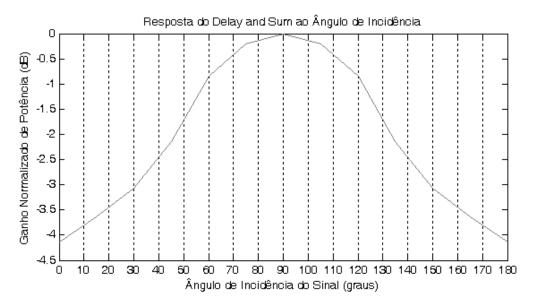


Figura 5.18 Resposta do Delay and Sum em função do ângulo de incidência do sinal

A figura anterior mostra o comportamento da diretividade do *Delay-and-Sum* no arranjo proposto. Dentre os algoritmos avaliados, este foi o que apresentou a menor atenuação dos sinais adjacentes ao sinal de interesse. Este resultado já era esperado, pois os demais algoritmos avaliados utilizam o *Delay-and-Sum* como ponto de partida para a filtragem espacial e em seguida aplicam técnicas para melhorar seu desempenho.

6 CONCLUSÃO

O objetivo do trabalho aqui apresentado foi o de pesquisar e tornar viável uma alternativa de baixo custo e fácil acesso para a implementação de uma rede de microfones que funcione em tempo real. Para isso, foram realizados estudos teóricos dos principais algoritmos utilizados nestas redes. Com o objetivo de facilitar a implementação prática destes algoritmos, vários testes foram realizados no Matlab. Estes testes foram importantes para definir quais algoritmos teriam complexidades computacionais compatíveis com aplicações de tempo real. Nestes testes foi descartado o algoritmo GSC com filtros adaptativos com IIR, pois para cada segundo de sinal foram necessários 36 segundos de processamento.

Após a definição dos algoritmos, o passo seguinte foi o de escolher uma base de hardware e software para a implementação da rede de microfones de baixo custo e fácil acesso. Com este objetivo optou-se por realizar a implementação em um computador pessoal (PC) com dispositivos de áudio USB. Os recursos de software escolhidos foram o sistema operacional Linux e a biblioteca de áudio com código aberto ALSA. Embora a documentação da biblioteca ALSA ainda esteja em um estágio inicial, a integração dos algoritmos estudados no Matlab ocorreu de maneira perfeita.

Durante o desenvolvimento deste trabalho, procurou-se documentar os principais passos com o objetivo de fornecer base para futuras pesquisas no campo das redes de microfones. Para isto, todo o código desenvolvido para o ALSA foi publicado na Internet na página http://www.fcth.br/mic-array e o código em Matlab está descrito em item anexo.

Os testes realizados com o arranjo proposto mostraram que o conjunto de soluções apresentada representa uma alternativa viável para a implementação de uma rede de baixo custo, e que possui grande potencial para evolução, uma vez que a qualidade dos dispositivos de áudio de baixo custo tende a aumentar.

As dificuldades encontradas neste desenvolvimento, descritas na seção 4.3, são atualmente as principais barreiras para um melhor desempenho da rede proposta. No entanto, todas estas dificuldades estão sendo analisadas e podem ter soluções no

curto e médio prazo. Para ilustrar esta afirmação, pode ser citado o exemplo dos dispositivos de áudio. Durante o desenvolvimento deste trabalho, os dispositivos disponíveis no mercado não possuíam boa razão sinal-ruído, o que prejudicava o desempenho dos algoritmos das redes de microfones. Nos últimos dias, foi lançado um dispositivo de áudio USB de excelente qualidade da marca Creative. Uma rede montada com este novo dispositivo, certamente produzirá um desempenho melhor para o arranjo proposto. O importante a ser considerado, é que a utilização de qualquer um destes novos dispositivos não implicará em qualquer necessidade de modificação do software apresentado, pois todos estes equipamentos respeitam as especificações da biblioteca ALSA.

Soluções como esta, permitem afirmar que o arranjo proposto neste trabalho, poderá ter, no médio prazo, sua aplicação em larga escala.

A continuidade deste trabalho pode se dar no desenvolvimento de um *driver* para o Linux partindo-se do software desenvolvido. Esta solução tornará esta aplicação muito mais versátil, pois poderá ser utilizada em *softwares* de comunicação como Skype, e outros aplicativos de voz sobre IP.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Perlmutter, Y.; Braids, L.; Frazier, R.; Oppenheim, A. "Evaluation of a speech enhancement system"; *IEEE International Conference on Acoustics, Speech, and Signal Processing 1977, (ICASSP '77).* Volume: 2, p. 212 215, maio 1977. Hartford, Connecticut, USA.
- [2] McAulay, R.; Malpass, M.; "A real-time noise suppression filter for speech enhancement and robust channel vocoding"; *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '80*, Volume: 5, p. 699 702, abril 1980. Denver, Colorado, USA.
- [3] Ramalho, M. A.; Mammone, R. J.; "New speech enhancement techniques using the pitch mode modulation model Circuits and Systems"; *Proceedings of the 36th Midwest Symposium on Circuits and Systems;* Volume 2 p. 1531 1534, 16-18 agosto 1993. Detroit, USA.
- [4] Lee, Te-Won; Yao, K.; "Speech enhancement by perceptual filter with sequential noise parameter estimation"; *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).*, Volume: 1, p. I 693-6 17-21, maio 2004. Montreal, Quebec, Canada.
- [5] Chen, Jingdong; Benesty, Jacob; Huang, Yiteng (Arden); "A minimum distortion noise reduction algorithm with multiple microphones"; *IEEE Transactions on Audio, Speech, and Language Processing*, Volume 16, n. 3, março 2008. Las Vegas, Nevada, USA.
- [6] Van Veen, B.D; Buckley, K.M.; "Beamforming: A versatil appoach to spacial filtering"; *IEEE ASSP 1998 Mag.*, Volume 5, n. 2, p. 4-24, abril 1998.
- [7] Bryn,F.; "Optimum signal processing of three-dimensional arrays operating on Gaussian signals and noise *The Journal of the Acoustical Society of America*; Volume 34, n. 3, p. 289-297, março 1962.

- [8] Middleton, D.; Groginski,H. I.; "Detection of random acoustic signals by receivers with distributed elements. Optimum receiver structures for normal signal and noise fields"; *The Journal of the Acoustical Society of America*; Volume 38, p. 727-737, novembro 1965.
- [9] Oh, S.; Viswanathan, V.; Papamichalis, P.; "Hands-free voice communication in an automobile with a microphone array"; *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1992. ICASSP-92.*; Volume: 1, p. 281 – 284, 23-26 Março 1992. San Francisco, California, USA.
- [10] Wang, A.; Yao, K.; Hudson, R. E.; Korompis, D.; Lorenzellii, F.; Soli, S.; Gao, S.; "Microphone array for hearing aid and speech enhancement applications", Proceedings of International Conference on Application Specific Systems, Architectures and Processors, 1996. ASSAP 96.; p. 231 – 239, 19-21 agosto 1996. Chicago, Ilinois, USA.
- [11] Seltzer, M. L.; Stern, R. M.; "Subband parameter optimization of microphone arrays for speech recognition in reverberant environments";. *Proceedings. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003.* (*ICASSP '03*). Volume: 1, p I-408 I-411, 6-10 abril 2003. Hong Kong.
- [12] Nguyen, D. Cuong; Shen, Guanghu; Jung, Ho-Youl; Chung, Hyun-Yeol; "Performance improvement of speech recognition system using microphone array"; *IEEE 2008*.
- [13] Dvorkind, T.; Gannot, S.; "Speaker localization in a reverberant environment" *The* 22nd Convention of Electrical and Electronics Engineers, p. 7 9, 1 dezembro 2002. Israel.
- [14] Hu, Jwu-Sheng; Cheng, Chieh-Cheng; Liu, Wei-Han; "Robust speaker's location detection in a vehicle environment using GMM models"; *IEEE Transactions on Systems, Man, and Cybernenetics—PART B: CYBERNETICS*; Volume 36, N. 2, abril 2006.
- [15] Conde, Flavio; Burt, P. M. S.; "Real time microphone arrays: a low-cost implementation with open-source code"; *IASTED 2010, Proceedings of*

- International Conference on Pattern Recognition and Applications fevereiro 2010. Innsbruck, Áustria.
- [16] Smaragdis, Paris; Boufounos, Petros; "Position and trajectory learning for microphone arrays"; IEEE Transaction on Audio, Speech, and Language Processing, Volume 15, n. 1, janeiro 2007.
- [17] Zhang, Cha; Florêncio, Dinei; Ba, Demba E.; Zhang, Zhengyou; "Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings"; *IEEE Transactions on Multimidia*, Volume 10, n. 3 abril 2008.
- [18] Saruwatari, H; Kajita, S.; Takeda K.; Itakura, F.; "Speech enhancement using nonlinear array based on complementary beamforming"; *IEICE Trans Fundamentals*; Volume E82-A, n. 8, agosto 1999.
- [19] Souden, Mehrez; Benesty, Jacob; Affes, Sofi`ene; "Microphone arrays for noise reduction with lowsignal distortion in room acoustics"; *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2008. ICASSP 2008*; p. 77 – 80, março-abril 2008. Las Vegas, Nevada, USA.
- [20] Douglas, S. C.; "Blind Separation of Acoustic Signals"; *Microphone Arrays*, M. Brandstein and D.Ward (Eds.) New York: Springer, 2001, cap. 16, p. 355 380.
- [21] Ziomek, L. J.; "Fundamentals of Acoustic Field Theory and Space-Time Signal Processing;" *CRC Press*; 1995.
- [22] B.D. Steinberg; "Principles of Aperture and Array System Design", (2nd Ed.), John Wiley and Sons, p. 12, New York 1976.
- [23] Rabinkin, Daniel V.; "Digital Hardware and Control for a Beam-Forming Microphone Array"; Thesis for the degree of Master of Science The State University of New Jersey; janeiro 1994.
- [24] Gilbert, E. N.; Morgan, S. P.; "Optimum design of directive antenna arrays subject to random variables;" *Bell System Technical Journal*; volume 34, p. 637-663, maio 1955.

- [25] Uzsoky, M; Solymrir, L; "Theory of superdirective linear arrays," *Acta Physics Academy.of Sciences. Hungary*, volume 6, n. 2, p. 185-205, 1956.
- [26] Kates, J. M.; Weiss, M. R,; "A comparison of hearing-aid array-processing techniques;" *The Journal of the Acoustical Society of America*; volume 99, p. 3138-3148; maio 1996.
- [27] Cox, H.; Zeskind, R. M.; Owen, M. M.; "Robust adaptive beamforming;" *IEEE Transactions on Acoustics, Speech, and Signal Processing;* volume 35, p. 1365-1375, outubro 1987.
- [28] Levin, M. J.; "Maximum-likelihood array processing;" *Seismic Discrimination Semi-Annu. Tech. Summary Rep.*, M.I.T. Lincoln Lab., Lexington, MA, Tech. Rep. DDC 455743, dezembro 1964.
- [29] Cox, H.; "Sensitivity considerations in adaptive beamforming;" Signal Processing (Proc. NATO Advanced Study Inst. Signal Processing with Particular Reference to Underwater Acoust., Loughborough, U.K., Aug. 1972.) J. W. R. Griffiths, P. L. Stocklin, and C. Van Schooneveld, Eds. New York and London: Academic, 1973.
- [30] Kooij, T.; "Adaptive array processors for sensitivity constrained optimization," *Ph.D. dissertation, Catholic Univ. Amer., Washington, DC;* junho 1977.
- [31] Hudson, J. E.; "Adaptive Array Principles;" *New York: Peter Peregrinus for IEE*; 1981.
- [32] Booker, A.; Ong, C. Y.; "Multiple constraint adaptive filtering;" *Geophysics;* volume 36, p. 498-509, junho 1971.
- [33] Tager, W; "Near field superdirectivity (NFSD);" Proceedings. IEEE International Conference on Acoustics, Speech, and Signal Processing 1998, (ICASSP '98); p. 2045-2048, 1998. Seattle, Washington, USA.
- [34] Frost, O. L.; "An algorithm for linearly constrained adaptive array processing"; *Proceedings of the IEEE*; volume 60, p. 926-935, agosto 1972.

- [35] Myllyla, V.; Hamalainen, M.; "Adaptive beamforming methods for dynamically steered array systems"; *IEEE International Conference on Acoustics, Speech, and Signal Processing 2008.(ICASSP 2008).* p.305-308. Las Vegas, Nevada, USA.
- [36] Applebaum, S. P.; Chapman, D.J.; "Adaptive Arrays with Main Beam Constraints;" *IEEE Transactions on Antennas and Propagation*; volume AP-24, p. 650-662, setembro 1976.
- [37] Griffths, L.; Jim, C. W.; "An alternative approach to linearly constrained adaptive beamforming;" *IEEE Transactions on Antennas and Propagation*; volume AP-30, p. 27-34, janeiro 1982.
- [38] Hoshuyama, O; Sugiyama, A.; "Robust Adaptive Beamformers"; *Microphone Arrays*, M. Brandstein and D.Ward (Eds.) New York: Springer, 2001, cap. 16, p. 87 109.
- [39] Shynk, J J; "A complex adaptive algorithm for iir filtering", *IEEE Transactions on Acoustics, Speech, and Signal Processing;* vol. ASSP-34, n. 5, outubro 1986.
- [40] Diniz, P.S.R.; "Adaptive IIR Filters" em *Adaptive Filtering Algorithms and Practical Implementation;* Kluwer Academic Publishers, 1999, cap. 9, p. 377-428.
- [41] Zelinski, R; "A microphone array with adaptive post- filtering for noise reduction in reverberant rooms"; *IEEE International Conference on Acoustics, Speech, and Signal Processing 1988, (ICASSP '88);* p. 2578-2581, abril 1988. New York, New York, USA.
- [42] Simmer, K.U; Wasiljeff, A.; "Adaptive microphone arrays for noise suppression in the frequency domain"; *Proc. 2nd COST-229 Workshop on Adaptive Algorithms in Communications;* p. 185-194, Bordeaux, France, 30 setembro 2 outubro 1992.
- [43] Marro, C.; Mahieux, Y.; Simmer, K.U.; "Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering"; *IEEE Transactions on Speech and Audio Processing*, Volume: 6, Issue: 3, p. 240-259, maio 1998.

- [44] Diniz, P.S.R.; "Wiener Filter" em *Adaptive Filtering Algorithms and Practical Implementation;* Kluwer Academic Publishers, 1999, cap. 2, p. 38-42.
- [45] Elko, G. W.; "Spatial coherence functions for differential microphones in isotropic noise fields"; *Microphone Arrays*, M. Brandstein and D.Ward (Eds.) New York: Springer, 2001, cap. 4, p. 61–85.
- [46] Cook, R. K.; Waterhouse, R. V.; Berendt, R. D.; Edelman, S.; Thompson Jr, M. C.; "Measurement of correlation coefficients in reverberant sound fields," *The Journal of the Acoustical Society of America*, Volume. 27, p. 1072–1077, 1955.
- [47] McCowan, Iain Bourlard, H "Microphone array post-filter for diffuse noise field"; *IEEE International Conference on Acoustics, Speech, and Signal Processing 2002.(ICASSP '02)*, vol. 1, 2002, p. 905–908. Orlando, Florida, USA.
- [48] McCowan, I. A.; Bourlard, H.; "Microphone array post-filter based on noise field coherence"; *IEEE Transactions on Speech and Audio Processing;* Volume 11, n. 6, novembro 2003.
- [49] ALSA: Advanced Linux Sound Architecture http://www.alsa-project.org
- [50] Meyer, J.; Siemer, K. U.; "Multi-channel speech enhancement in car environment using Wiener filtering and spectral subtraction"; *IEEE International Conference on Acoustics, Speech, and Signal Processing 1997. (ICASSP'97).* Volume 2, p.1167,. Munich, Bavaria, Germany
- [51] McCowan, I. A.; "Robust Speech Recognition Using Microphone Arrays"; PhD Thesis, Queensland University of Technology, Australia, 2001
- [52] Ward, D. B.; Kennedy, R. A.; Williamson, R. C.: "Constant directivity beamforming" *Microphone Arrays*, M. Brandstein and D.Ward (Eds.) New York: Springer, 2001, cap. 1, p. 3–17.
- [53] Oppenhein, A. V; Schafer, R. W.; "Sampling of continuous time signal" *Discrete-time signal processing*; 2ª edição, Prentice Hall, p. 140-239

ANEXO A – LISTAGEM DOS PROGRAMAS EM MATLAB

A.1 POST-FILTER

```
% Autor: Flavio Conde
% Criado em jun/2005 e modificado em jun/2009
% Programa desenvolvido para aplicar o algoritmo do Post-Filter a uma rede de
% microfones
% Os sinais de entrada foram gravados em ambiente real de uso
fs = 16000; % freq. amostragem
ts = 5; % tempo de amostragem em seg.
canal = 5; % canal do microfone de 1 a 8
N=8; %num. de microfones
np=512;
snr=3;
alfa=0.99;
load 'E:\fc\Pos\Dissertacao\Ensaio\sinal' Br; %carrega o sinal desejado
load 'E:\fc\Pos\Dissertacao\Ensaio\ruidos' Rc Rc2; %carrega os sinais de ruido
B=zeros(ts*fs,N);
if snr==3
  for i=1:N
     B(1:ts*fs,i)=Br(1:ts*fs,1)+3.56*Rc(1:ts*fs,i)+3.56*Rc2(1:ts*fs,i); % sinal + ruido
com SNR = 3dB
  end;
end;
if snr==10
  for i=1:N
     B(1:ts*fs,i)=Br(1:ts*fs,1)+1.57*Rc(1:ts*fs,i)+1.57*Rc2(1:ts*fs,i); % sinal + ruido
com SNR = 10dB
  end:
```

```
end;
nf = ceil(ts*fs/np);
R=zeros(N,N); %dimensiona a matriz das correlações
v=zeros(N,nf * np);
Swf=zeros(nf*np,1);
Sw=zeros(nf*np,1);
for j=1:nf-1
  for i=1:N
     ini = (j - 1) * np + 1;
    fim = j * np;
     v(i, ini:fim) = fft(B(ini:fim,i));
  end;
end;
for k=1:(nf-1)*np
  Rnum=0;
  for i=1:N-1
     for j=i:N
       R(i,j)=alfa*R(i,j)+(1-alfa)*real(v(i,k).*conj(v(j,k)));
       Rnum = Rnum + R(i,j);
     end;
  end;
  Rnum=2/(N^*(N-1))*Rnum;
  Rden=0;
  vm = 0;
  for i=1:N
     Rden=Rden+R(i,i);
     vm = vm + v(i,k);
  end;
  Rden=Rden/N;
  vm = vm/N;
  Wn=Rnum/Rden; %calculo o filtro de Wiener
```

```
Swf(k) = vm * Wn;
end;

%Calcula a saída do Post-Filter
for j=1:nf-1
    ini=(j-1)*np+1;
    fim=ini+np-1;
    Sw(ini:fim) = real(ifft(Swf(ini:fim)));
end;

wavplay(Sw,fs);
```

A.2 GSC LAF-LAF

```
% Autor: Flavio Conde
% Criado em jun/2005 e modificado em jun/2009
% Programa desenvolvido para aplicar o algoritmo do GSC LAF-LAF a uma rede de
% microfones
% Os sinais de entrada foram gravados em ambiente real de uso

fs=16000; % freq. amostragem
ts=5; % tempo de amostragem em seg.
N=4; %num. de microfones

%Parametros do filtro NLMS do Blocking Matrix (BM)
M1=32; %num. de coef. do filtro NLMS
alfa=0.1; %passo de adaptação do algoritimo
delta=0.8; % leakage constant

%Parametros do filtro NLMS do Multiple input canceler (MC)
M2=32; %num. de coef. do filtro NLMS
```

```
beta=0.1; %passo de adaptação do algoritimo
gama=0.8; % leakage constant
nz=0.0001;
L1=10;
L2=5;
snr=3;
Rc=zeros(N,ts*fs);
load 'E:\fc\Pos\Dissertacao\Ensaio\sinal' Br; %carrega o sinal desejado
load 'e:\fc\Pos\Dissertacao\Ensaio\ruidos' Rc Rc2; %carrega os sinais de ruido
z=zeros(fs*ts,N);
Z=zeros(M2,N);
Yo=zeros(fs*ts,1);
y=zeros(N-1,1);
B=zeros(ts*fs,N); %sinal de entrada
if snr==3
  for i=1:N
     B(1:ts*fs,i)=Br(1:ts*fs,1)+3.56*Rc(1:ts*fs,i)+3.56*Rc2(1:ts*fs,i); % sinal + ruido
com SNR = 3dB
  end;
end;
if snr==10
  for i=1:N
     B(1:ts*fs,i)=Br(1:ts*fs,1)+1.57*Rc(1:ts*fs,i)+1.57*Rc2(1:ts*fs,i); % sinal + ruido
com SNR = 10dB
  end;
end;
Ybf=mean(B')'; %sinal de saida do fixed beamform (FBF)
h=zeros(M1,N); % coeficientes do filtro NLMS do Blocking Matrix (BM)
```

```
w=zeros(M2,N); % coeficientes do filtro NLMS do Multiple input canceler (MC)
tic;
for n=M1:ts*fs
  for i=1:N
     b=Ybf(n:-1:n-M1+1);
     z(n,i)=B(n-L2,i)-h(:,i)'*b;
     h(:,i)=h(:,i)-delta*h(:,i)+alfa*(z(n,i)/(b'*b))*b;
  end;
  for i=1:N
     Z(:,i)=z(n:-1:n-M2+1,i)';
     y(i)=w(:,i)'*Z(:,i);
  end;
  Yo(n)=Ybf(n-L1)-mean(y);
  for i=1:N
     w(:,i)=w(:,i)-gama*w(:,i)+beta*(Yo(n)/(nz+Z(:,i))*Z(:,i))
  end;
end;
wavplay(Yo,fs);
```

A.3 GSC IIR

```
% Autor: Flavio Conde
% Criado em jun/2005 e modificado em jun/2009
% Programa desenvolvido para aplicar o algoritmo do GSC IIR a uma rede de
% microfones
% Os sinais de entrada foram gravados em ambiente real de uso

fs = 16000; % freq. amostragem
ts = 5; % tempo de amostragem em seg.
N = 8; %num. de microfones
ini=1;
fim=ts*fs;
```

```
L=0.999; %lambida
alf=1-L; %alfa
Mp=4; %num. de coefic. do numerador b
Np=4; %num. de coefic. do denominador a
a=zeros(Np-1,N-1);
b=zeros(Mp,N-1);
P=zeros(Mp+Np-1,Mp+Np-1,N-1);
for i=1:N-1
  P(:,:,i)=eye(Mp+Np-1).*0.1;
end;
NI=ts*fs;
teta=zeros(Mp+Np-1,N-1);
fi=zeros(Mp+Np-1,N-1);
psi=zeros(Mp+Np-1,N-1);
y=zeros(NI,N-1);
dV=zeros(NI,N-1);
Vn=zeros(N-1,1);
V=zeros(N-1,1);
x=zeros(NI,N-1);
yf=zeros(NI,N-1);
xf=zeros(NI,N-1);
ym=zeros(NI,1);
Yo=zeros(NI,1); %sinal de saida e de erro
snr=3;
Rc=zeros(N,ts*fs);
load 'E:\fc\Pos\Dissertacao\Ensaio\sinal' Br; %carrega o sinal desejado
load 'e:\fc\Pos\Dissertacao\Ensaio\ruidos' Rc Rc2; %carrega os sinais de ruido
if snr==3
  for i=1:N
```

```
B(1:ts*fs,i)=Br(1:ts*fs,1)+3.56*Rc(1:ts*fs,i)+3.56*Rc2(1:ts*fs,i); % sinal + ruido
com SNR = 3dB
  end:
end;
if snr==10
  for i=1:N
     B(1:ts*fs,i)=Br(1:ts*fs,1)+1.57*Rc(1:ts*fs,i)+1.57*Rc2(1:ts*fs,i); % sinal + ruido
com SNR = 10dB
  end:
end;
for n=ni:NI-2500
  y_sum=0;
  for i=1:N-1
     %Matriz de Bloqueio
     x(n,i)=B(n,i)-B(n,i+1);
     fi(:,i)=([y(n-1:-1:n-Np+1,i)' x(n:-1:n-Mp+1,i)'])';
     Vn(i)=teta(:,i)'*teta(:,i);
     dV(n,i)=Vn(i)-V(i);
     V(i)=Vn(i);
     y(n,i)=teta(:,i)'*fi(:,i);
     y_sum=y_sum+y(n,i);
  end;
  if max(dV(n,:))>0.0002
     teta=zeros(Mp+Np-1,N-1);
  end:
  L=1-alf;
  ym(n)=y_sum/(N-1);
  Yo(n)=s(n)-ym(n);
  for i=1:N-1
     a(:,i)=teta(1:Np-1,i);
     b(:,i)=teta(Np:Np+Mp-1,i);
     xf(n,i)=x(n,i)+sum(a(1:Np-1,i).*xf(n-1:-1:n-Np+1,i));
     yf(n-1,i)=y(n-1,i)+sum(a(1:Np-1,i).*yf(n-2:-1:n-Np,i));
     psi(:,i)=[yf(n-1:-1:n-Np+1,i)' xf(n:-1:n-Mp+1,i)']';
```

```
P(:,:,i)=P(:,:,i)/L-alf/L*((P(:,:,i)*psi(:,i)*Psi(:,i))*P(:,:,i))/(L+alf*psi(:,i)*P(:,:,i)*psi(:,i)));

teta(:,i)=teta(:,i)+alf*P(:,:,i)*psi(:,i)*Yo(n);

end;

if mod(n,2000)==0

n
dV(n,:)
teta
end;
end;
wavplay(Yo,fs);
```