GIULIO GUIYTI ROSSIGNOLO SUZUMURA

REMOÇÃO DE RUÍDO EM VOZ COM ARRAY DE MICROFONES E MODIFICAÇÃO ESPECTRAL

Relatório final de Trabalho de Graduação de Engenharia de Informação da Universidade Federal do ABC como parte dos requisitos para aprovação do curso.

Professor Dr. Irineu Antunes Jr.

Santo André - SP

2014

AGRADECIMENTOS

Aos melhores pais, Yoshikazu Suzumura Filho e Inêz Rossignolo Suzumura

À minha irmã Giorgia Yoshiko Rossignolo Suzumura

Ao grande amigo Tales Gouveia Fernandes e em especial à Gabriella Machado Pereira

Ao Professor Doutor Ricardo Suyama pelo auxílio e colaboração.

De modo especial, agradeço ao Professor Doutor Irineu Antunes Junior, pela excelente orientação e revisão minuciosa do texto, além da paciência demonstrada durante toda a realização desse trabalho.

RESUMO

O projeto aqui proposto considera a redução de ruído nas captações de sinais acústicos, onde um ruído concorrente com um sinal principal se soma a este e resulta em um sinal audível, porém ruidoso. O sinal principal foi considerado como sendo de voz, já o ruído foi representado por um ruído aditivo, gaussiano e branco. Para efetuar a redução do ruído, foi criado um método baseado num arranjo linear e uniforme de microfones seguido da subtração espectral da magnitude do espectro do ruído. Nas simulações, a voz humana incide a um ângulo conhecido em relação aos microfones e o ruído é suposto uniforme em todo o ambiente. A partir de simulações computacionais, foi possível obter uma boa estimativa do áudio principal. Foram realizadas comparações subjetivas (inspeção visual de espectrogramas e avaliação do áudio por ouvintes) e objetivas (relação sinal-ruído e PESQ, *Perceptual Evaluation of Speech Quality*) para três métodos diferentes de redução de ruído, a saber: apenas arranjo de microfones, apenas subtração espectral e o cascateamento destes o qual demonstrou ter maior redução de ruído do que os métodos individuais.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

AWGN	*Additive, White and Gaussian Noise*
dB	*deciBel*
STFT	*Discrete Time Fourier Transform*
FFT	*Fast Fourier Transform*
FIR	*Finite Impulse Response*
IIR	*Infinite Impulse Response*
MOS	*Mean Opinion Score*
PESQ	*Perceptual Evaluation of Speech Quality*
SegSNR	*Segmented Signal-to-noise Ratio*
SNR	*Signal-to-noise Ratio*
VAD	*Voice Activity Detector*

LISTA DE SÍMBOLOS

n	índice de tempo discreto
fs	frequência de amostragem do sistema
$s(n)$	sinal de voz limpo
$r(n)$	ruído
$x_i'(n)$	sinais atrasados da fonte limpa
$x_i(n)$	soma do ruído com sinais atrasados
θ	ângulo de emissão do sinal limpo
T_i	atraso entre os sinais dos microfones (amostras)
φ	suposto ângulo do sinal limpo
D	distância genérica entre microfones
N	quantidade genérica de microfones do *Array*
Δ	distância entre ondas recebidas
Δt	tempo do atraso entre microfones (enviado)
τ_i	quantidade de amostras defasadas entre microfones

$z(n)$	sinal de saída do *Delay-Sum Beamforming*
$y(n)$	sinal $z(n)$ tratado pela Subtração Espectral
$m(n)$	sinal $x_1(n)$ tratado pela Subtração espectral
w_i	pelo atribuído a cada microfone i
d	distância entre cada microfone i
τ_i	tempo do atraso entre microfones (recebido)
c	velocidade de propagação da onda sonora
K	constante genérica composta por d, theta, fs e c
S	transformada do sinal s(n)
R	transformada do sinal r(n)
Z	transformada do sinal z(n)
Y	transformada do sinal y(n)
$SNRm$	relação sinal-ruído do m-ésimo bloco

1. INTRODUÇÃO

Vivemos em um ambiente, onde, na maioria das vezes, o ruído é inevitável e ubíquo, de maneira que sinais de voz são geralmente imersos em ruído e raramente podem ser adquiridos e processados de forma pura.

O ruído pode afetar profundamente a comunicação humano-humano, modificando as características do sinal de fala e assim degradando a qualidade e a inteligibilidade da fala e, consequentemente, afetando a percepção do ouvinte. Dependendo do tipo de ruído a comunicação humano-máquina e máquina-máquina podem ser afetadas prejudicando o processamento de instruções transmitidas e recebidas. A fim de tornar possível a comunicação natural e confortável na presença de ruído, é desejável o desenvolvimento de técnicas de processamento de sinais para limpar o sinal recebido antes de ser armazenado ou transmitido [1].

De um modo geral, o ruído é um termo usado para representar qualquer sinal indesejado que interfere com a medição, processamento e comunicação do sinal de informação desejada. Esta definição no sentido amplo, no entanto, é demasiadamente abrangente por mascarar muitos aspectos técnicos importantes do problema real. Para permitir uma melhor modelagem e remoção dos efeitos do ruído, é vantajoso quebrar a definição geral de ruído nas seguintes subcategorias [1]:

- "Ruído Aditivo", originado de várias fontes de som ambiente,
- "Sinais Interferentes", entre falantes concorrentes,
- "Reverberação", causada pela propagação reflexões diversas (multipath) e
- "Eco", resultado do acoplamento entre alto-falantes e microfones.

O combate a esses quatro problemas levou à evolução de diversas técnicas de processamento de sinais, inclusive acústicos, que se encaixam nas quatro categorias sendo estas, redução de ruído (aperfeiçoamento de fala), separação de fontes, dereverberação de fala e supressão ou cancelamento de eco.

A pesquisa pioneira sobre a redução de ruído de um canal foi iniciada na década de 1960 com duas patentes de Schroeder [2][3]. Ele propôs uma implementação analógica de subtração espectral de magnitude. Este autor, no entanto, não recebeu muita atenção do público, provavelmente porque nunca foi publicado em revistas ou conferências [4].

As técnicas de processamentos de sinais sofreram consideradas evoluções ao decorrer de sua história e, após severas transformações tecnológicas somadas ao envolvimento computacional, são realizadas atualmente como processamento digital de sinais. O processamento digital de sinais tornou-se um tema importante em pesquisas contemporâneas, em virtude da ampla gama de aplicações na engenharia. Novas técnicas de separação de sinais podem ser aplicadas nos casos em que as técnicas mais antigas não podem ser usadas, possibilitando resoluções de problemas em áreas diversas, principalmente na ciência e engenharia.

Cerca de 15 anos depois das patentes de Schroeder, Boll, em seu *paper* informativo [5], reinventou o método de subtração espectral, além de passar a trabalhar no domínio digital. Quase ao mesmo tempo, Lim e Oppenheim, em um trabalho marcante [6], formularam de forma sistemática o problema de redução de ruído, estudaram e compararam os diferentes algoritmos

conhecidos na época e demonstraram que a redução de ruído não é apenas útil para melhorar a qualidade de discursos corrompidos por ruído, mas também útil para aumentar a qualidade de sistemas baseados em codificação preditiva linear (LPC, *Linear Predictive Coding*).

1.1. PROBLEMA DA PESQUISA

É comum ouvir em gravações de áudio, assim como no áudio embutido em vídeo, sons interferentes provenientes de fontes indesejáveis.

Figura 1-1 – Bluetooth Headset

Neste trabalho, é investigado o problema chamado de separação de sinais de áudio, nome que se dá pelo fato de que é realmente difícil separar as fontes sonoras (tanto para o cérebro humano, quanto para computadores) em locais com um ruído uniforme ao fundo, aqui considerado AWGN isotrópico. Foi definido o caso de apenas um falante em um ambiente uniformemente ruidoso.

Atualmente, alguns aparelhos celulares, além de muitos equipamentos *handsfree,* como o da figura 1-1, são capazes de eliminar ruídos a um nível aceitável por meio de um *hardware* que algumas vezes são dotados de dois ou três microfones. Alguns carros com sistema de áudio *bluetooth,* com função de telefone, são contemplados com um sistema parecido para reduzirem ruídos externos [14].

1.2. OBJETIVO

O projeto proposto tem como objetivo modelar e simular três formas de redução de ruído: 1) *Array* de microfones; 2) Subtração espectral de magnitude; 3) Método combinado usando os dois métodos anteriores em cascata. Além disto, é feita a comparação de desempenho das três formas propostas, buscando verificar que o método combinado proporciona maior redução de ruído.

2. FUNDAMENTAÇÃO TEÓRICA

Filtros, são comumente definidos como sendo sistemas projetados para extrair informações, sobre uma quantidade de interesse, a partir de um sinal ruidoso [7, p.1]. No trabalho realizado foram utilizados dois tipos de filtros, um espacial - técnica de processamento de sinais usada em *arrays* para transmissão ou recepção de sinais direcionais - e um espectral - filtro que reduz ruído de fundo com características de banda larga. Ambos podem ser implementados digitalmente, tornando possível a realização das simulações deste trabalho, o qual considerou o processamento "off-line" de sinais digitais previamente gravados.

De maneira geral, um sistema de comunicação é composto basicamente por um transmissor, um canal de transmissão e um receptor, como na figura 2-1.

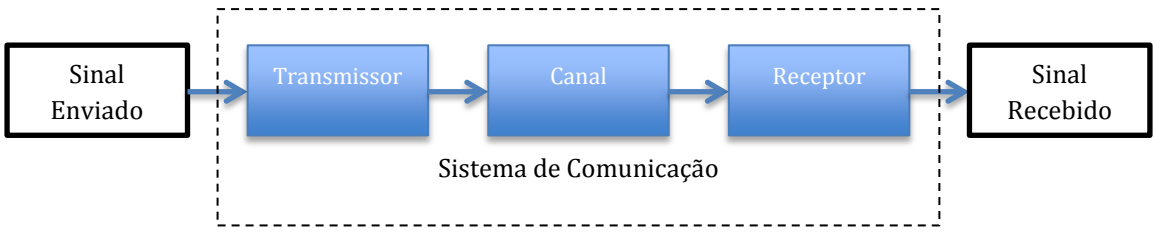

Figura 2-1 – Composição básica de um sistema de comunicação

O transmissor é o responsável por converter o sinal em formas de onda que possam ser transmitidas pelo canal de transmissão utilizado. O receptor por sua vez recebe o sinal transmitido e entrega, em sua saída, o sinal reconstituído.

Normalmente, o processo de filtragem fica localizado no receptor e sua eficiência é responsável pela inteligibilidade da comunicação. No caso aqui estudado, é possível expandir o conceito de receptor, pois o receptor final de um sistema acústico será o ser humano que intrinsecamente possui a capacidade de "filtragem e estimação" do sinal audível e pode ter esta capacidade explorada pelos métodos de melhoria de comunicação empregados.

A existência de ruído é inevitável em muitas aplicações de processamento de voz em ambientes reais. Num sistema de comunicação, um sinal acústico desejado, como a voz de uma pessoa captada por um microfone, é corrompido por ruído indesejado no ambiente, resultando em um sinal distorcido. Este sinal corrompido necessita ser processado por um filtro que buscará suprimir o ruído, deixando o sinal desejado relativamente inalterado. Este é o conceito básico de cancelamento de ruído que rege este trabalho.

O estudo das técnicas de cancelamento de ruído vem sendo feito desde 1960 e, desde então, vários métodos foram propostos e investigados [7 ,p.25]. Estes métodos podem ser classificados em três categorias básicas: Técnicas de *Beamforming* que exploram o uso de múltiplos sensores, Cancelamento Adaptativo de Ruído que utiliza um sensor primário que capta o sinal ruidoso e outro sensor para captar o sinal de referência de ruído e Modificação Espectral Adaptativa que se aplica a um único sensor. Serão detalhadas duas destas técnicas as quais foram utilizadas neste trabalho.

A técnica de *Beamforming*, que em uma tradução literal seria "Formação de feixe", é conhecida também por *Array* . Esta técnica utiliza um arranjo de sensores, onde para o caso acústico se dá por um arranjo de microfones, que tem a função de explicitar o sinal desejado e suprimir o ruído. Este conjunto de microfones é espacialmente posicionado no ambiente ruidoso onde se quer captar o sinal desejado. A posição de cada microfone e a geometria são conhecidos e referenciados a um ponto do ambiente. Estes sensores captam os sinais que chegam até eles e, através de pesos diferenciados para cada sensor, é construído um feixe direcional de captação de sinal. Desta forma, o sinal que estiver se propagando na direção de interesse será reforçado e os de outras direções, suprimidos. Informações sobre as técnicas de *Array* podem ser encontradas em [7,p.18] e [8].

A Modificação Espectral consiste em se trabalhar com o espectro de frequência do áudio ruidoso. De posse do sinal ruidoso, esta técnica tenta restaurar a magnitude espectral do sinal desejado, subtraindo do sinal ruidoso uma estimativa da magnitude espectral do ruído. Ao decorrer do estudo bibliográfico, foram encontradas diversas variantes da técnica de modificação espectral, porém todas referenciam a técnica de Boll [5] e, por este motivo, foi utilizada esta técnica. Informações sobre a filtragem por subtração espectral aqui realizada podem ser encontradas em [5], [7] e [11].

3. METODOLOGIA

A metodologia deste projeto foi baseada na modelagem e simulação de um problema. Em um sistema de gravação sem processamento a qualidade do áudio gravado não é necessariamente a melhor, já com o processamento digital de sinais é possível alterar a entrada do sistema de forma a melhorar a qualidade sonora na saída. Com a ajuda de algoritmos computacionais já conhecidos foi possível simular procedimentos para se obter esta melhora ~~e~~ lançando mão da união de procedimentos conhecidos.

No modelo de reprodução de áudio sem processamento, figura 3-1, o bloco "Entrada" fornece um único áudio ruidoso, o qual é repassado para o bloco "Saída". Neste caso é como se o áudio captado fosse reproduzido sem alterações.

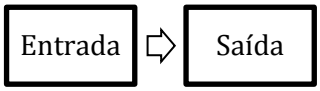

Figura 3-1 – Modelo sem processamento

$$y(t) = x(t), \tag{1}$$

onde $x(t)$ é o sinal de entrada, suposto ruidoso, e $y(t)$ o sinal de saída que se obteria na ausência de processamento.

Já no modelo com a presença de processamento, o sinal de entrada é trabalhado de forma a ser possível obter uma saída com maior qualidade em relação ao modelo sem processamento.

Figura 3-2 – Modelo com processamento

$$y(t) = h(x(t)), \tag{2}$$

em que h(.) é o "Processo", x(t) é a entrada ruidosa e y(t) a saída do sistema com menor presença de ruído.

No modelo com processamento, da figura 3-2, o bloco "Entrada" fornece sinais ruidosos para o bloco "Processo", que digitalmente processa os dados e fornece para o bloco "Saída" um áudio com inteligibilidade superior ao da entrada.

O bloco "Entrada" é descrito no capítulo 3.1 e se destina a criação de um ambiente onde a voz trafega em um ângulo definido até o *array* de microfones.

O processamento do áudio foi baseado em arranjo de microfones e subtração espectral. O primeiro método é largamente utilizado para redução de ruído em celulares modernos, quando há dois ou mais microfones. Já o segundo, é empregado para reduzir ainda mais o ruído de fundo, suposto AWGN. Todos os procedimentos do bloco "Processo" são descritos no capítulo 3.2.

O bloco "Saída" apresenta o sinal estimado da entrada, ou seja, o resultado final, descrito no capítulo 3.3.

3.1. ENTRADA

O bloco "Entrada" da figura 3-2 é mais bem representado pela figura 3-3. A entrada de áudio no sistema foi totalmente simulada, não sendo utilizados microfones reais para captação do sinal de voz. Utilizando-se do próprio software MATLAB® foi carregado um arquivo de voz $s(n)$, criado um ruído $r(n)$ e então foi simulada a recepção de cada microfone do sistema, obtendo-se os sinais $x_1(n)$, $x_2(n)$, ..., $x_N(n)$.

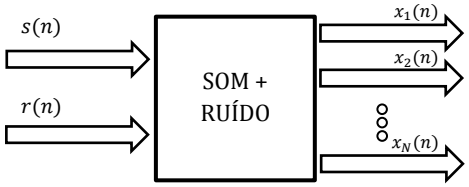

Figura 3-3 – Entrada simulada: s(n) sinal limpo, r(n) ruído, x_N(n) sinais de saída de cada microfone.

O sinal $s(n)$, foi retirado do banco de dados de voz TIMIT CORPUS SAMPLE, disponibilizado em [15]. Escolheu-se a frase em inglês "*The Thinker is a famous sculpture*" falada por uma voz masculina gravado em formato do tipo WAVform (WAV), por não ser comprimido e preservar a qualidade máxima, tendo sido amostrado a 16kHz. Já o sinal de ruído $r(n)$ foi gerado usando a função "*randn()*" do MATLAB®, tendo média nula e variância constante, simulando um AWGN isotrópico. Além disso, foram criados sinais ruidosos com uma relação sinal-ruído de 4.5dB.

A partir do sinal limpo $s(n)$, do ruído $r(n)$, da quantidade de microfones N, da distância d entre estes e do ângulo θ de incidência de $s(n)$, foi criado o script "ENTRADA.m" - ANEXO 1. Este script cria os áudios $x_i(n)$ defasados que serão recebidos em cada um dos microfones, desta forma, neste trabalho, não consideramos gravações de um ambiente real. O resultado da execução deste script é uma matriz com a quantidade de microfones referenciados nas linhas, e as amostras de cada microfone apresentadas nas colunas.

Com a função "CRIA_MICS.m" – ANEXO 1A – que é utilizada pelo script "ENTRADA.m", foi possível criar os áudios $x_i(n)$ de cada microfone com o ruído embutido.

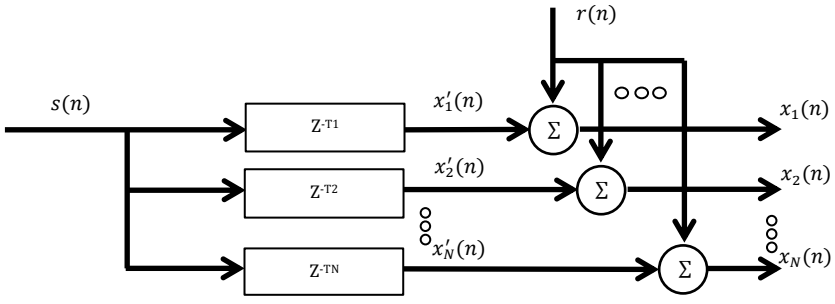

Figura 3-4 – Simulação de atrasos de cada microfone i: x'$_i$(n) sinais atrasados de T_i amostras.

Conforme o diagrama da figura 3-4, é possível observar que o mesmo sinal $s(n)$ é atrasado de T_i amostras para cada microfone, conforme (3) e (4).

$$x_i'(n) = s(n - T_i) \tag{3}$$

$$x_i(n) = x_i'(n) + r(n)$$

$$x_i(n) = s(n - T_i) + r(n)\,, \tag{4}$$

onde $x_i(n)$ é o sinal ruidoso no microfone i, $r(n)$ é o ruído e $s(n - T_i)$ é o sinal atrasado de T_i amostras.

Com o estudo de trigonometria é possível calcular o atraso do sinal de entrada em todos os microfones do *array* como apresentado na figura 3-5.

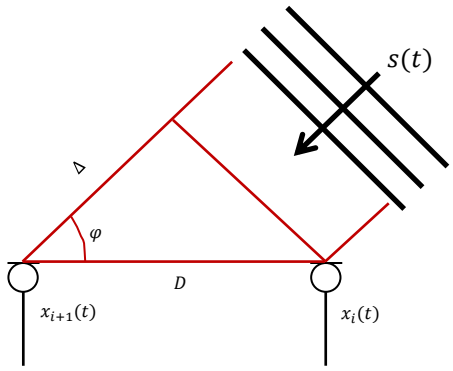

Figura 3-5 –Atraso de onda plana em relação a um ângulo φ, sendo D a distância entre microfones e s(t) a fonte de interesse.

Um ponto muito importante a ser considerado é a modelagem de onda plana atribuída ao sinal sonoro. Para tal, deve ser considerado que a distância da fonte sonora ao conjunto de microfones é bem maior do que a distância d entre os microfones, em outras palavras está sendo considerado *far-field* (campo distante). [8]

Para se encontrar a distância Δ entre o sinal recebido por um microfone x_i e outro x_{i+1}, pode-se empregar a expressão

$$\Delta = D.cos\varphi, \tag{5}$$

onde φ é o ângulo de incidência da onda e D a distância entre os microfones.

Para se determinar o atraso no tempo, é necessário se utilizar a velocidade de propagação da onda sonora, que no ar é de aproximadamente 340m/s, logo

$$\Delta t = \frac{D.cos\varphi}{340}, \tag{6}$$

onde φ é o ângulo de incidência da onda, D é a distância entre microfones e Δt é o tempo de diferença entre x_i e x_{i+1}.

Na maioria das situações práticas, este atraso se dá em frações de segundos e, como trabalhamos com sinais digitais, devemos fazer a conversão entre tempo e amostras. Esta conversão é obtida em razão da taxa de amostragem, desta forma é possível afirmar que o atraso fracionário do sinal de tempo discreto é de

$$\Delta n = \Delta t\,.fs\,, \tag{7}$$

onde Δt é o tempo desta defasagem para os sinais de tempo contínuo e fs é a frequência da amostragem.

A aplicação de um atraso fracionário aos sinais de tempo discreto foi realizada utilizando uma função pronta do Matlab chamada "*delayseq()*", que realiza o atraso fracionário com base na FFT, modificação linear de fase e IFFT de um bloco de amostras.

Antes de se determinar a quantidade N de microfones e a distância D entre estes, convém estudar o método "*Delay-sum Beamforming*" do bloco "Processo", conforme será visto na seção seguinte.

3.2. PROCESSO

O bloco "Processo" apresentado na figura 3-6 tem como referencia a figura 3-2 e foi criado dentro do script chamado "PROCESSO.m" – ANEXO 2. Este é subdividido em dois blocos nomeados como DS_BEAM (*Delay-Sum Beamforming*) - ANEXO 2A - e SUBSPEC (Subtração Espectral) – ANEXO 2B.

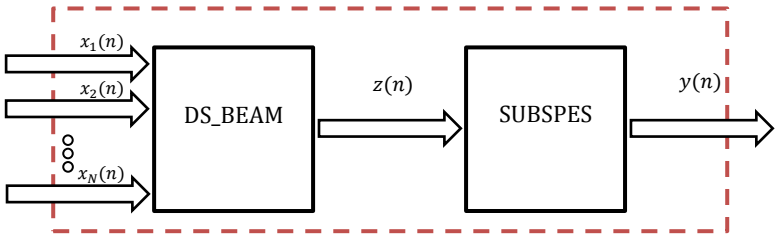

Figura 3-6 – Bloco Processo.

O bloco DS-BEAM em uma tradução literal seria "Formação de feixe por atraso e soma". Este método é conhecido na literatura como um modelo clássico de técnica de *beamforming*, onde os sinais de cada entrada $x_n(t)$ sofre um atraso correspondente, sendo, em seguida, calculado o sinal médio, $z(n)$.

Como o feixe tem a característica de reforçar o sinal proveniente de determinada direção, os sinais provenientes de outras direções acabam sendo atenuados. Desta forma, por exemplo, em um ambiente com ruído que possa ser aproximado por um AWGN isotrópico, geralmente, é possível realizar o direcionamento do feixe auditivo do *array* a fim de reforçar ou identificar a voz de determinada pessoa. Vale comentar que este método também pode ser usado caso existam diversas pessoas falando espalhadas pelo ambiente.

O bloco SUBSPEC implementado realiza a média espectral proposta por Boll [5]. As características do ruído são estimadas a partir do instante inicial de silêncio, ou seja, assume-se que o instante inicial é composto apenas por ruído e este é usado para modelar o ruído de todo o restante do sinal, supondo-se que o ruído seja estacionário.

Após esta análise geral do macro bloco "Processo", da figura 3-6, serão explicados a seguir os micro blocos DS_BEAM e SUBSPEC.

3.2.1. *DELAY-SUM BEAMFORMING*

É possível verificar no modelo da figura 3-7 um conjunto de microfones uniformemente espaçados, posicionados em uma mesma linha imaginária e a propagação de uma onda sonora, suposta plana, chegando a uma direção específica. Deve ser deixado claro que não é esta a única forma de posicionamento geométrico dos microfones que, por exemplo, podem ser dispostos em

forma circular, como em sonares de submarinos. Além disto, nesta figura, tomou-se a liberdade de modelar o problema já considerando sinais de tempo discreto.

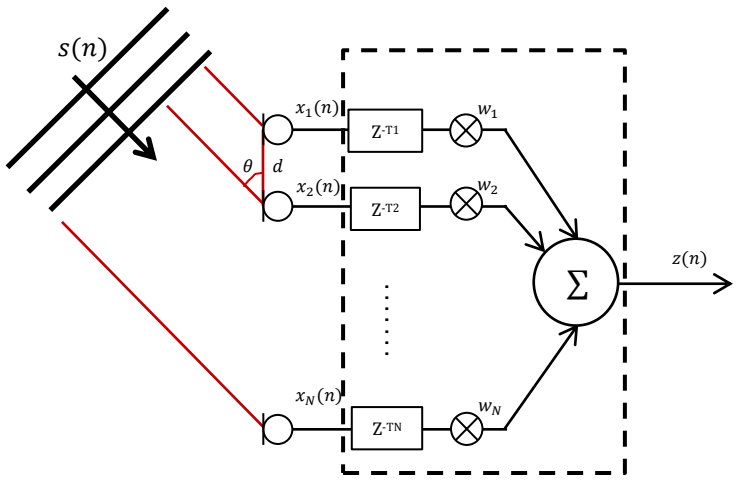

Figura 3-7 – *Delay-Sum Beamforming:* T_i amostras atrasadas de cada sensor i, w_i peso de cada sensor i, z(n) sinal de saída.

O sistema desta figura possui N microfones, onde $x_i(n)$ é o sinal, já amostrado, recebido pelo i-ésimo sensor (que vai de 1 até N), T_i é o atraso relativo de amostras de cada sensor, w_i é o peso atribuído a cada sensor, θ é o ângulo de incidência da frente de onda na linha onde estão localizados os microfones, d é a distância entre cada microfone e $z(n)$ é a saída deste sistema.

Usualmente, na técnica *delay-sum*, os pesos para cada microfone são iguais e, para se obter um ganho unitário na saída, o peso é o inverso da quantidade de microfones N:

$$w_i = \frac{1}{N}. \tag{8}$$

Toda a teoria aplicada no método *Delay-Sum Beamformer* é a mesma aplicada no bloco "Entrada". Pela figura 3-7 é possível perceber novamente este problema de geometria e então determinar o atraso τ_i (segundos) de cada microfone adjacente com as variáveis apresentadas e a velocidade de propagação c da onda [8], dada por

$$\tau_i = (i-1)\frac{d.cos\theta}{c}, \tag{9}$$

onde θ é o ângulo de incidência da onda *array*, d é a distância entre cada microfones, c é a velocidade de propagação e i o índice do microfone.

Este cálculo permite obter os atrasos dos sinais em tempo contínuo, isto é, antes da amostragem: o sinal do primeiro sensor é atrasado de τ_1 segundos, o sinal do segundo sensor é atrasado de τ_2 segundos e assim por diante. Com os valores τ_i é possível adiantar cada onda e somá-las para se obter a saída do *array*. Este tempo de atraso pode ter frações de segundos que quando convertido para amostradas podem resultar em frações de amostras. É possível calcular este atraso em amostras a partir de

$$T_i = \tau_i.fs. \tag{10}$$

onde τ_i é o tempo de atraso do i-ésimo microfone em relação ao primeiro e fs é a frequência de amostragem.

Se as distâncias d entre os microfones, o ângulo de incidência θ e a frequência de amostragem fs forem constantes durante todo o processo, é possível escrever

$$K = \frac{d.cos\theta}{c}.fs\,, \tag{11}$$

e então substituindo (10) em (11) é possível perceber que

$$T_i = (i-1).K, \tag{12}$$

para i=1, 2, ..., N, provando que cada microfone i é atrasado de T_i amostras em relação ao primeiro.

Os microfones captam o sinal proveniente da fonte principal além de ruídos provenientes de todas as direções, deste modo podemos modelar os sinais captados pelos sensores da seguinte forma:

$$\mathbf{x}(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_N(n) \end{bmatrix} = \begin{bmatrix} s(n-T_1) \\ s(n-T_2) \\ \vdots \\ s(n-T_N) \end{bmatrix} + \begin{bmatrix} r_1(n) \\ r_2(n) \\ \vdots \\ r_N(n) \end{bmatrix}, \tag{13}$$

para um bloco de amostras, com n=0,1,...,M.

Aplicando a Transformada de Fourier de tempo discreto (DTFT) aos sinais de (13), obtém-se

$$\begin{bmatrix} X_1(e^{j\omega}) \\ X_2(e^{j\omega}) \\ \vdots \\ X_N(e^{j\omega}) \end{bmatrix} = \begin{bmatrix} S(e^{j\omega})e^{-j\omega T_1} \\ S(e^{j\omega})e^{-j\omega T_2} \\ \vdots \\ S(e^{j\omega})e^{-j\omega T_N} \end{bmatrix} + \begin{bmatrix} R_1(e^{j\omega}) \\ R_2(e^{j\omega}) \\ \vdots \\ R_N(e^{j\omega}) \end{bmatrix}, \tag{14}$$

que pode ser escrito como

$$X_i(e^{j\omega}) = S(e^{j\omega})e^{-j\omega T_i} + R_i(e^{j\omega})\,, \tag{15}$$

onde i = 1,2,...,N e T_i é o atraso em amostras de cada microfone *i*.

Aplicando os atrasos T_i e os pesos w_i, a expressão geral do *beamformer* de atrasos e somas fica com a forma: [8]

$$Z(e^{j\omega}) = \sum_{i=1}^{N} w_i\, X_i(e^{j\omega})e^{j\omega T_i}. \tag{16}$$

Substituindo (15) em (16), pode-se escrever

$$Z(e^{j\omega}) = S(e^{j\omega}) + \frac{1}{N}\sum_{i=1}^{N} R_i(e^{j\omega})e^{j\omega T_i}. \tag{17}$$

Nesta expressão, é possível perceber que o sinal $s(n)$ se mantém inalterado na saída do *beamforming*, enquanto o ruído $r(n)$ é reduzido.

Após toda esta análise, finalmente, é possível finalizar a etapa de "Entrada", na qual não havia sido definida a quantidade de microfones e a distância de separação entre eles.

Neste trabalho, foi considerado um *array* de $N = 5$ microfones, e distância de $d = 5$ centímetros entre cada um deles e incidência de ângulo θ (figura 3-7). O projeto foi simplificado pelo fato de ser conhecido o valor do ângulo de incidência da fonte principal em relação ao *array* de microfones, logo $\theta = \varphi$.

Para um θ genérico de 45° e substituindo os valores de c e d em (9), obtemos a expressão para o atraso de cada microfone:

$$\tau_i = (i-1).\frac{0{,}05\ .\ cos45}{340}$$

$$= (i-1).10^{-4}\ . \qquad (18)$$

É possível verificar que nunca haverá atraso no primeiro microfone, pois $i = 1$ em (18) resulta $\tau_1 = 0$. Isto é facilmente entendido pelo fato de ser este o sinal de referência.

Após esta análise sobre o método *Delay-Sum Beamforming* e as características utilizadas no bloco "Entrada" do sistema, é apresentado a seguir o método de filtragem por Subtração Espectral.

3.2.2. SUBTRAÇÃO ESPECTRAL

A técnica de subtração espectral, também conhecida como supressão espectral, consiste em se descobrir uma estimativa do ruído no domínio da frequência e com isto modificar o espectro de frequência do áudio de entrada pela subtração desta estimativa do espectro do ruído. Na saída, normalmente, é obtido um sinal modificado e com menor presença de ruído em relação à entrada.

Se o ruído tiver uma distribuição espectral estreita, a filtragem digital pode ser facilmente aplicada para suprimir esta componente de ruído. Contudo, se a componente de ruído tem uma banda espectral larga, uma simples filtragem de supressão de banda pode não ser apropriada, devendo-se nestes casos empregar métodos como a subtração espectral.

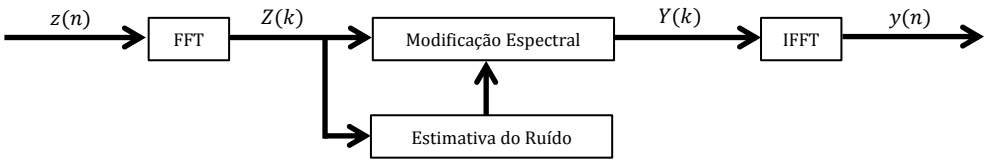

Figura 3-8 – Subtração espectral, sendo z(n) um bloco do sinal de entrada, Z(k) a transformada rápida de Fourier de z(n), Y(k) o espectro do sinal reconstruído e y(n) a transformada Inversa.

Na figura 3-8 é aplicada uma transformada ao sinal de entrada z(n) a fim de se trabalhar no domínio da frequência. Após esta transformação é realizada uma estimativa do ruído e esta estimativa é subtraída do sinal de entrada. Após uma transformação para o domínio temporal o resultado é um sinal com menos ruído, ou seja, o método de subtração espectral é uma abordagem simples e eficaz para suprimir ruído de fundo estacionário e de banda larga. É importante ressaltar que o sinal de voz não pode ser considerado um processo estacionário,

porém para intervalos de ordem de 30ms de duração os sinais podem ser caracterizados como tal [13].

Este método é baseado na hipótese de que o espectro do sinal $z(n)$ pode ser expresso como a soma do espectro de voz $s(n)$ com o espectro do ruído $r(n)$, sendo o processamento é feito no domínio da frequência, assim,

$$Z(k) = S(k) + R(k), \tag{19}$$

em que $S(k)$ é a FFT (*Fast Fourier Transform*) do de um bloco do sinal desejado, s(n), $R(k)$ é a transformada um bloco do ruído, $r(n)$, e $Z(k)$ é a transformada de um bloco do sinal corrompido, z(n). Tomando-se o quadrado na equação (19) é possível chegar em

$$|Z(k)|^2 = |S(k)|^2 + |R(k)|^2 + 2|S(k)||R(k)|cos\theta_k, \tag{20}$$

onde θ_k representa a diferença de fase entre as componentes espectrais do sinal de voz e do ruído.

Se o sinal de voz e o ruído são processos randômicos, estacionários e não correlacionados, em média, a expressão (20) pode ser aproximada por:

$$E\{|Z(k)|^2\} = E\{|S(k)|^2\} + E\{|R(k)|^2\}. \tag{21}$$

A subtração espectral é aplicada somente para o espectro de potência do sinal, preservando-se a fase do sinal ruidoso. A técnica da subtração espectral faz uso da propriedade de que o aparelho auditivo humano é pouco crítico a variações pequenas de fase. Desta forma, é possível combinar a magnitude do espectro instantâneo com a fase do sinal ruidoso.

Assim, estimando-se a potência do ruído $E\{|R(k)|^2\}$ da equação (21), obtém-se:

$$E\{|S(k)|\} = \sqrt{E\{|Z(k)|^2\} - E\{|R(k)|^2\}}. \tag{22}$$

Esta é a forma básica do método de subtração espectral. Um algoritmo geral pode ser desenvolvido no domínio da magnitude espectral [11]:

$$\left|\hat{S}(k)\right|^b = |Z(k)|^b - \eta\left|\hat{R}(k)\right|^b, \tag{23}$$

onde $\hat{S}(k)$ é a estimativa do espectro $S(k)$, $\hat{R}(k)$ é uma estimativa do espectro do ruído, b é um expoente inteiro, $b = 2$ representa a estimativa da magnitude quadrada; η é um parâmetro para controle da quantidade de ruído a ser retirada na subtração espectral, $\eta = 1$ se traduz como retirada máxima de ruído. Logo, a estimativa do espectro da voz será construída por

$$\hat{S}(k) = \left[|Z(k)|^b - \eta\left|\hat{R}(k)\right|^b\right]^{\frac{1}{b}} e^{j\varphi}. \tag{24}$$

Esta expressão é conhecida como a subtração espectral paramétrica [11], onde para $b = 2$ e $\eta = 1$, atribui-se o nome de IPS (*Instantaneous Spectral Subtraction*) e, para $b = 1$ e $\eta = 1$, o nome de MS (*Magnitude Spectral Subtraction*).

As técnicas de remoção de ruído que realizam modificação espectral podem introduzir uma distorção muitas vezes descrita como um "ruído musical". Normalmente, estas distorções aparecem quando há muita presença de ruído, situação esta que se espera evitar pelo uso do bloco *Delay-Sum Beamforming* antes de se fazer a Subtração Espectral.

Após apresentados os micro blocos *Delay-Sum Beamforming* e Subtração Espectral a próxima a ser descrita é etapa de "Saída".

3.3. SAÍDA

O bloco "Saída" da figura 3-2 apresenta o resultado audível dos procedimentos aplicados neste trabalho e é apresentado no script "SAIDA.m" - ANEXO 3.

Como foram simulados os blocos DS_BEAM e SUBSPEC separadamente também foi possível realizar comparações entre a filtragem pelo primeiro e pelo segundo método individualmente.

Para realizar a análise subjetiva baseada na audição foi reproduzido inicialmente o áudio da fonte $s(t)$ seguido da reprodução do áudio ruidoso sem processamento $x_1(t)$. Em seguida, foram reproduzidos os áudios filtrados:

$m(t)$ – processado apenas pela subtração espectral -,

z(t) - processado apenas pelo array de microfones - e finalmente o

$y(t)$ - cascateamento dos dois métodos.

4. RESULTADOS E DISCUSSÕES

Serão realizadas três comparações entre os áudios: visual, audível e numérica. As duas primeiras foram consideradas como subjetivas, por não terem resultados numéricos, onde serão apresentados gráficos e resultados da avaliação humana ao reproduzir os áudios resultantes. Para isto, foram criados os scripts "RESULTADO_subjetivo" – ANEXO 4 – e "RESULTADO_objetivo" – ANEXO 5. Lembrando que as simulações foram realizadas no ambiente MATLAB®. Todo o processamento realizado pode ser representado resumidamente pela figura 4-1.

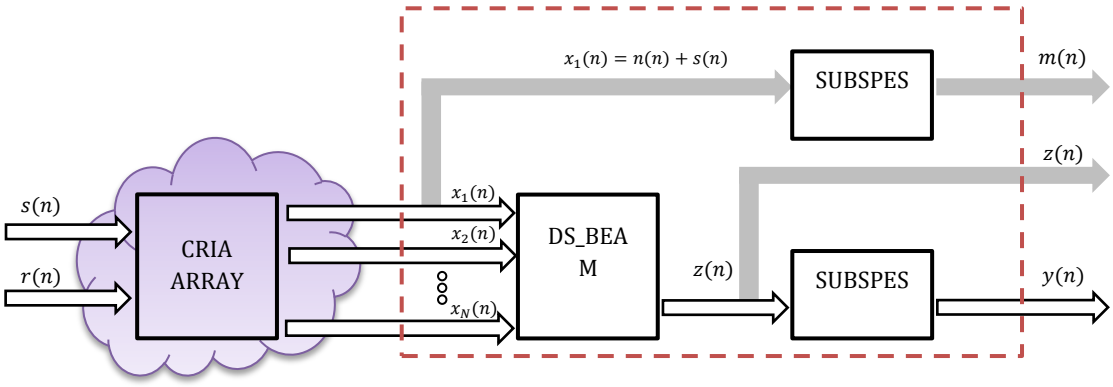

Figura 4-1 – Processos realizados

A nuvem na figura 4-1 representa a "entrada simulada", ou seja, a partir de um sinal de uma fonte sem ruído $s(t)$ e um ruído $r(t)$, foram criados cinco sinais defasados entre si. O primeiro sinal $x_1(t)$ foi tratado como sinal principal recebido, ou seja, este sinal seria o sinal recebido se não houvesse processamento pelo *array*. Foram então analisadas três diferentes tipos de filtragem em relação a este sinal $x_1(t)$:

- Modificação Espectral
- Arranjo de microfones
- Cascateamento dos métodos anteriores

A filtragem do sinal recebido $x_1(t)$ e processado apenas pela modificação espectral resulta em $m(t)$, os sinais $x_n(t)$ recebidos pelo *array* e trabalhados por *beamforming* resultam em $z(t)$, já os dois tipos de filtragem em sequência produzem $y(t)$.

4.1. AVALIAÇÃO SUBJETIVA

Para ser possível realizar a análise subjetiva visual foram plotados gráficos das formas de onda no tempo. Na figura 4-2(A) é apresentada uma comparação entre o sinal recebido pelo microfone principal $x_1(t)$ (preto) e o sinal da fonte $s(t)$ (vermelho). Uma observação a ser feita é que o ruído adicionado resultou em um aumento de amplitude, o que era de se esperar. Na figura 4-2(B) é apresentado o mesmo sinal do microfone $x_1(t)$, porém comparado ao sinal $m(t)$ processado apenas pelo bloco SUBSPEC (azul). Na figura 4-2(C), fornecemos a comparação entre $x_1(t)$ (preto) e o sinal $z(t)$ processado pelo ARRAY (verde). A figura 4-2(D) relaciona $x_1(t)$ (preto) com a saída $y(t)$ (ciano).

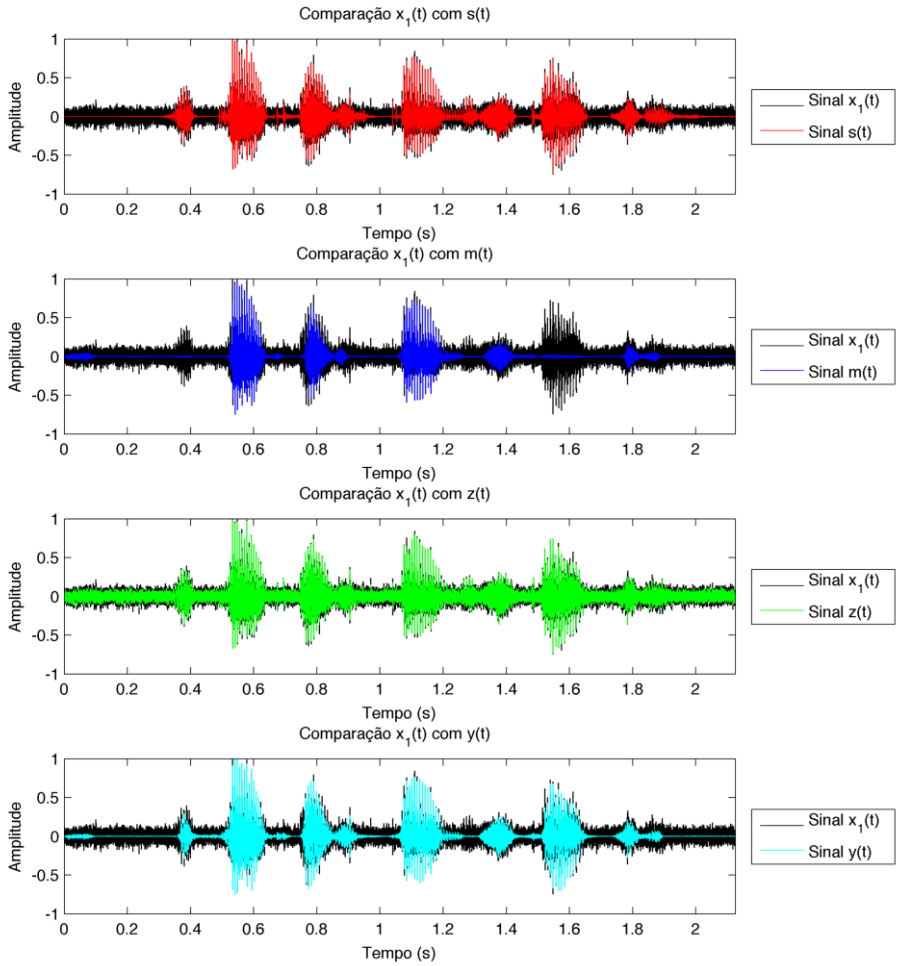

Figura 4-2 – Comparação dos sinais – (A) recebido x fonte, (B) recebido x processado SUBSPEC, (C) recebido x processado ARRAY, (D) recebido x processado completo

Destas comparações, pode-se observar que $m(t)$ (azul) parece não representar com fidelidade algumas poucas porções do sinal fonte, mas, apesar disto, parece reduzir drasticamente o ruído. Também pode-se observar que $z(t)$ (verde) ainda tem grande parcela de ruído enquanto que $y(t)$ (ciano) é o que mais se assemelha do sinal original. Não é possível determinar objetivamente a real qualidade de cada etapa do processamento apenas com a observação da amplitude em função do tempo, pois a única informação que se pode retirar destes gráficos é que os três métodos reduzem o ruído, mas podem estar afetando o sinal de voz original.

Para ser possível realizar uma melhor análise visual, foi realizada a análise espectral do sinal, utilizou-se então o espectrograma.

O espectrograma tem três dimensões geométricas, onde o eixo horizontal representa o tempo, o eixo vertical representa a frequência e a terceira dimensão é indicada por cores dentro dos eixos. A intensidade das cores representam a amplitude (em dB) para uma frequência apresentada no eixo vertical e um momento no eixo horizontal. Desta forma o espectrograma disponibiliza uma representação da potência do sinal no tempo e na frequência.

Uma escala de cores acompanha os espectrogramas. Quanto mais "quente" a cor, maior é a amplitude da frequência, consequentemente, as cores "frias" representam amplitudes

pequenas. Para ser possível realizar comparações entre os espectrogramas, foi adotada uma escala única. O espectrograma do sinal original $s(t)$ é apresentado na figura 4-3.

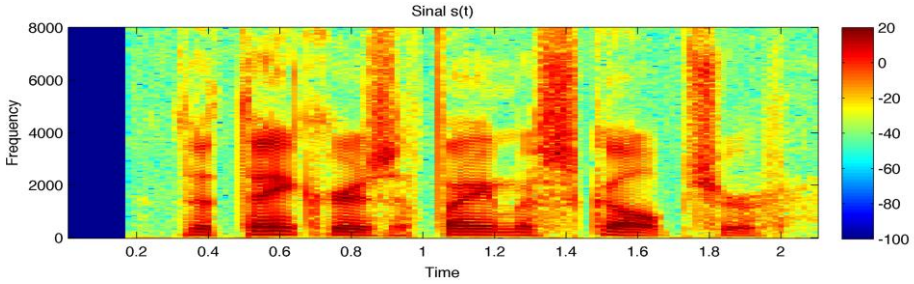

Figura 4-3 – Espectrograma do sinal da fonte

As mesmas comparações dos gráficos apresentados na figura 4-2 podem ser realizadas entre a figura 4-3 e 4-4.

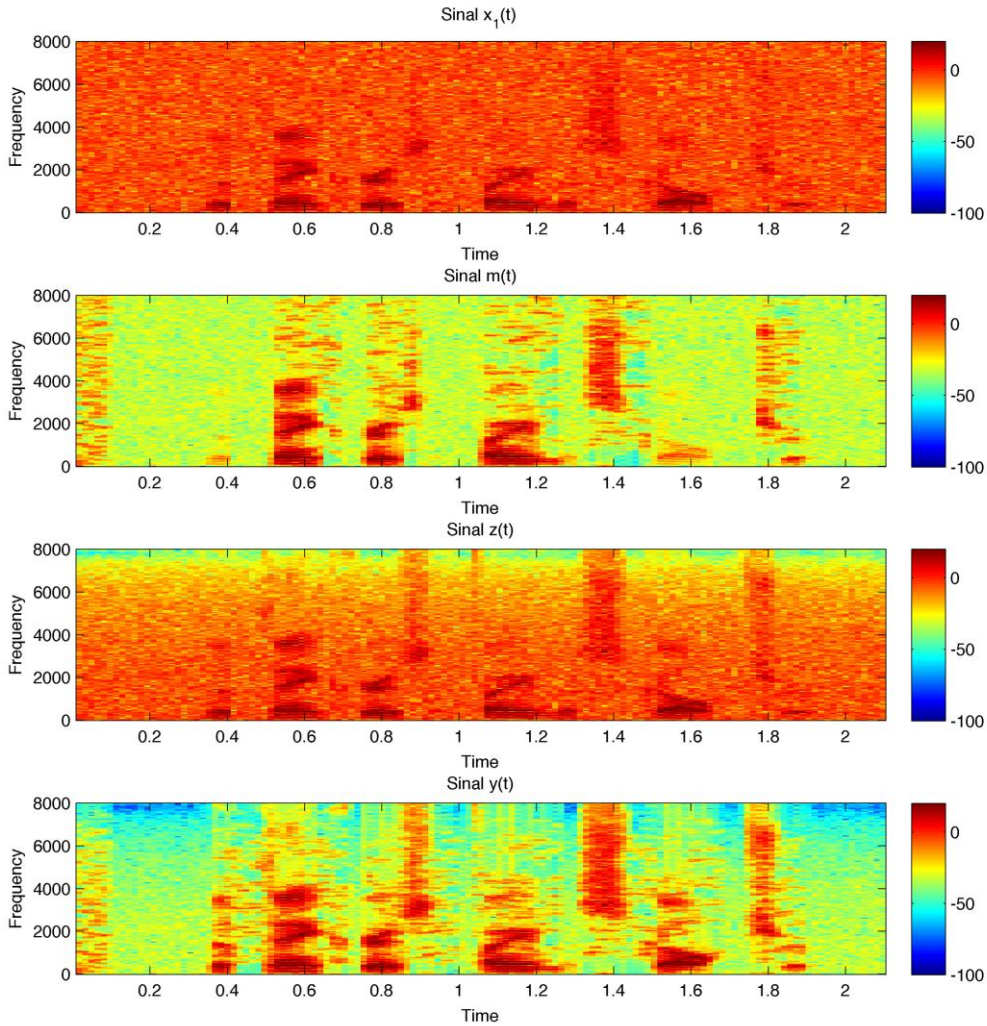

Figura 4-4 – Espectrograma do sinal: (A) recebido pelo microfone referência, (B) obtido após processamento SUBSPEC, (C) obtido após processamento ARRAY, (D) obtido após união dos métodos

Na figura 4-3, existe um alto contraste entre as cores quentes e as cores frias, o que simboliza distribuição da energia da voz em frequências definidas.

A figura 4-4(A) apresenta o sinal $x_1(t)$ que chega no microfone principal. É possível afirmar que o ruído corrompe o áudio em todos os microfones, pois a diferença entre estes é apenas um atraso no tempo. O ruído atinge todas as frequências resultando em um áudio pouco inteligível.

A figura 4-4(B) apresenta a saída após a filtragem de subtração espectral $m(t)$. É possível verificar que esta filtragem realmente reduz o ruído pelo fato deste espectrograma se assemelhar ao sinal $s(t)$ - figura 4-3.

A figura 4-4(C) apresenta a saída $z(t)$, sinal processado pelo arranjo de microfones. A comparação entre este espectrograma e o da figura 4-4(A) mostra que a técnica de "*delay-sum beamforming*" consegue reduzir uma parcela do ruído, ainda que em quantidade inferior em relação a $m(t)$ - figura 4-4(B).

A figura 4-4(D) apresenta o espectrograma do cascateamento do *delay-sum beamforming* com a subtração espectral. É possível perceber que, apesar de pouca diferença entre este e o da figura 4-4(B), ele ainda apresenta o espectrograma mais parecido com o da figura 4-3 e, consequentemente, deve proporcionar melhor resultado.

A análise subjetiva audível foi feita por duas pessoas que compararam todos os sinais descritos com o sinal original da fonte $s(t)$. O script "SAÍDA" – ANEXO 3 – foi construído para reproduzir todos os sinais áudios.

Para ser avaliado a qualidade do áudio foi implementado um sistema de escala, criada com uma faixa de valores entre 0 e 5, sendo 0 um áudio não inteligível e 5 um áudio de máxima qualidade.

	Pessoa 1	Pessoa 2
$x_1(t)$	0	1
$m(t)$	2	3
$z(t)$	1	2
$y(t)$	3	4

Tabela 1 – Análise subjetiva (audível)

As duas análises indicam que o cascateamento dos processos é a melhor forma de se obter redução de ruído em relação aos processos separados.

O próximo passo para se confirmar que a união dos processos é o melhor método para se reduzir ruído, é a avaliação objetiva.

4.2. AVALIAÇÃO OBJETIVA

Foram realizadas medidas quantitativas para determinar se cada método reduziu de forma satisfatória o ruído ambiente.

A primeira análise foi baseada na relação sinal-ruído estimada da seguinte forma

$$SNR = 10\log_{10}\left(\frac{\sum_{n=0}^{L-1}|s(n)|^2}{\sum_{n=0}^{L-1}|s(n)-y(n)|^2}\right). \quad (25)$$

em que L é o comprimento total dos sinais, $s(n)$ é o sinal de referência e $y(n)$ o sinal recuperado.

Por ser uma medida global do nível de ruído, a SNR não reflete bem a qualidade do sinal de voz obtido na saída. A quantidade é mais bem representada pelo valor médio da razão sinal-ruído entre blocos do sinal completo.

$$SegSNR = \frac{1}{B}\sum_{m=0}^{B-1} SNR_m. \quad (26)$$

na qual B é o número de blocos considerados, para blocos de 34 amostras de comprimento.

A medida SegSNR é conhecida como razão sinal-ruído segmentada, onde SNR_m é a razão sinal-ruído no m-ésimo bloco. Caso, para algum bloco, ocorra logaritmo de zero, tal problema é contornado limitando-se os valores de SNR_m a uma faixa dinâmica de 40 a -40 dB.

Além destas medidas objetivas, foi utilizada uma terceira chamada PESC (*Perceptual Evaluation of Speech Quality*), uma medida que avalia a qualidade perceptível de voz, padronizada pela ITU-T P.862. Esta medida é aplicada industrialmente, sendo usada por fabricantes de telefones, fornecedores de equipamentos de rede e operadores de telecomunicações.

Foram realizadas 10 simulações para o mesmo sinal de voz mas com diferentes realizações de ruído, sendo calculada a média de cada uma das medidas. Os resultados são apresentados na tabela 2.

	SNR (dB)	SegSNR (dB)	PESQ
$x_1(t)$	4.5	-12.17	1.86
$m(t)$	7.37	-9.26	1.89
$m(t)$ Boll [16]	3.24	-3.65	1.45
$z(t)$	7.98	-8.91	1.95
$y(t)$	11.19	-5.31	2.01

Tabela 2 – Análise objetiva

Na tabela 2, todos os sinais de saída, $m(t)$, $z(t)$ e $y(t)$ podem ser comparados com o sinal $x_1(t)$, o sinal gravado sem processamento.

Para descobrir se os valores apresentados pelo parâmetro PESQ são satisfatórios, foi criada a tabela 3 onde são comparados o sinal limpo com ele mesmo e o sinal limpo com o ruído puro. Nesta tabela, pode-se ver que a medida PESQ tem valor máximo de 4.5 e que o sinal ruidoso tem PESQ de 0.53, esperando-se os métodos de redução de ruído proporcionem valores intermediários a esses.

	SNR (dB)	SegSNR (dB)	PESQ
$s(t)$	Inf	40	4.5
$r(t)$	-1.31	-5.07	0.53

Tabela 3 – Análise objetiva para se obter parâmetros

Para todas as medidas de qualidade adotadas, quanto maior o valor nas tabelas 2 e 3, maior é a qualidade do áudio em relação ao sinal limpo $s(t)$ de entrada.

Com respeito à relação sinal-ruído segmentada, SegSNR, é possível afirmar que o sinal proveniente pelo método cascateado tem melhor resultado, igualmente apresentado pelo parâmetro PESQ, cujo o resultado foi de 2.01.

Os resultados das tabelas 1, 2 e 3 foram fornecidos como saída do script "RESULTADOS_objetivos.m" - ANEXO 5 – utilizando um tamanho de bloco equivalente a 30 ms, ou seja, 480 amostras , e após esta análise foram feitas outras simulações com sinais ruidosos na entrada do sistema conforme apresentamos a seguir.

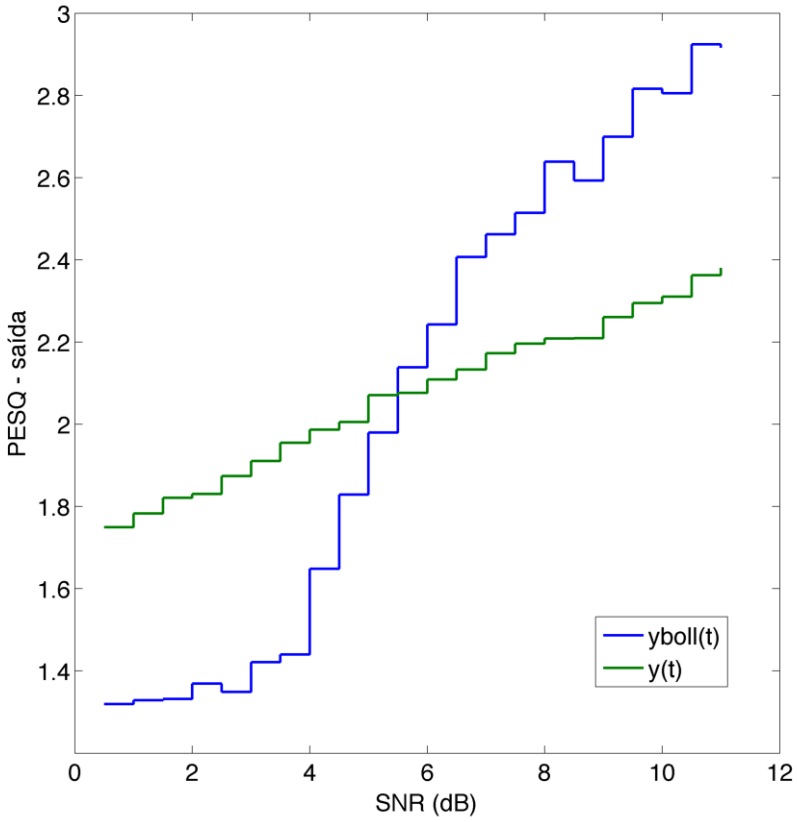

Figura 4-5 – Relação entre a saída utilizando-se o algoritmo criado SUBSPEC - y(t) e Boll [16] - yboll(t)

O resultado apresentado na figura 4-5 mostra que enquanto a subtração espectral de Boll [16] (azul) só apresenta bons resultados para SNR de entrada relativamente elevadas, o algoritmo de subtração espectral aqui implementado (verde) (ANEXO – 2B), mantém um bom desempenho, mesmo para sinais mais degradados (SNR abaixo de 5.5 dB), sendo, portanto, mais indicado para uma possível implementação prática.

5. CONCLUSÕES

A partir de um estudo bibliográfico foi possível modelar e simular os métodos de redução de ruído com *array* de microfones e com subtração espectral de magnitude. Com os resultados apresentados na tabela 1, a *performance* da medida PESC da tabela 2 e os espectrogramas apresentados na figura 4-4, é possível afirmar que o estudo subjetivo e objetivo apresentaram resultados coerentes.

Foi obtida melhor qualidade no sinal de saída quando utilizado a técnica combinada em comparação com os métodos separados. Isto se deu pelo fato de que o arranjo de microfones reduzir o ruído no sinal antes deste passar pela subtração espectral, proporcionando um sinal menos ruidoso na entrada deste segundo método.

Como visto na figura 4-5 o método de subtração espectral de Boll [16] revela que seus parâmetros são ajustados para sinais de entrada com elevada SNR global. Já o algoritmo criado (SUBSPEC) tem desempenho menos variável com a relação sinal-ruído da entrada, desta forma após o *array* de microfones, este último proporciona melhores resultados, principalmente para SNR de entrada abaixo de 5.5 dB.

Como sugestão para continuação deste trabalho, Um próximo passo seria utilizar a subtração espectral em cada canal do *Array*, para então ser realizado o *Delay-Sum Beamforming*.

6. CONSIDERAÇÕES FINAIS

O trabalho de graduação foi dividido em três grandes períodos, descritos como "Trabalho de Graduação I", "Trabalho de Graduação II" e "Trabalho de Graduação III". O primeiro período, comprimiu os meses de novembro de 2013 à janeiro de 2014, o segundo período os meses de março à maio de 2014, o terceiro e último de junho e agosto de 2014.

Como requisito para conclusão de cada período foi necessário realizar a entrega de relatórios genéricos, porem ampliados e melhorados a cada período. Durante os períodos foram realizadas diversas modificações a fim de beneficiar a qualidade científica do projeto final.

No "Trabalho de Graduação I", todo o projeto foi idealizado com o objetivo de reduzir ruído a partir de dois canais de entrada (dois microfones), sendo um canal com a qualidade de áudio mais claro do que ruidoso e outro canal mais ruidoso. Além da captação dos sinais, seria realizada a supressão espectral para se reduzir ainda mais o ruído, o que não ocorreu.

	1º período				2º período			3º período		
Fases da Pesquisa	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago
Tema e formulação do problema										
Levantamento Bibliográfico										
Elaboração do Projeto										
Testes com Tipos de Soluções										
Escolha da Solução Ideal										
Implementação										
Análise de Dados										
Redação Trabalho Final										

Tabela 4 – Cronograma inicial – apresentado em Trabalho de Graduação I

Foram feitas diversas modificações no projeto, o cronograma foi alterado tanto em etapas, quanto em períodos de cada uma. O cronograma da tabela 4 referencia o cronograma apresentado no “Trabalho de Graduação I”, ou seja, o inicialmente proposto, já a tabela 5 foi proposta no segundo período e mantida no terceiro.

No segundo período, "Trabalho de Graduação II", foi mantido o objetivo de se reduzir ruído com microfones e supressão espectral, porem a quantidade de microfones não havia sido estabelecida, e sim por um valor maior ou igual a dois. No segundo período foi projetado como seria o relatório final e também foram criados alguns dos scripts, como o cálculo de delay dos microfones e a subtração espectral.

	1º período				2º período			3º período		
Fases da Pesquisa	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago
Tema e formulação do problema										
Levantamento Bibliográfico										
Modelagem e Simulação										
Análise de dados										
Comparação de resultados										
Experimento real										
Redação Trabalho Final										

Tabela 5 – Cronograma cumprido – apresentado em Trabalho de Graduação II e III

No terceiro (último) período foram finalizados todos os scripts bem como o relatório final do trabalho de graduação.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] - BENESTY J., SONDHI M. M., HUANG Y., ***"*Fundamentals of noise reduction*"***, Springer Handbook of Speech Processing, EiCs, Springer-Verlag, Chapter 43, pp. 843-872, 2007.

[2] - SCHROEDER M. R.: U.S. Patent No. 3180936, filed Dec. 1, 1960, issued Apr. 27, 1965

[3] - SCHROEDER M. R.: U.S. Patent No. 3403224, filed May 28, 1965, issued Sept. 24, 1968

[4] - SCHROEDER M. R., ROSSING T. D., DUNN F., HARTMANN W.M., D.M. CAMPBELL, N.H. FLETCHER , ***"*Acoustic Signal Processing*"***, Springer Handbook of Acoustics, Rossing, Chapter 14, pp. 503-530, 2007.

[5] - BOLL S.F.: "***Suppression of acoustic noise in speech using spectral subtraction****"*, IEEE Trans. Acoust. Speech Signal Process. ASSP-27, 113–120 (1979)

[6] - LIM J.S., OPPENHEIM A.V.: ***"*Enhancement and bandwidth compression of noisy speech*"***, Proc. IEEE67, 1586–1604 (1979)

[7] - HAYKIN, S., ***"Adaptive Filter Theory"***, 4ª ed., Prentice-Hall, Englewood Cliffs, NJ, 2001b.

[8] - MCCOWAN I.A.. ***"*Robust Speech Recognition using Microphone Arrays*"*** PhD Thesis, Queensland Uni- versity of Technology, Australia, 2001.

[9] - WIENER N., ***"Extrapolation, Interpolation and Smoothing of Stationary Time Series, With Engineering Applications"***, Wiley, New York, NJ, 1949.

[10] - LEVITT, H., ***"Noise reduction in hearing aids: An overview"***, Journal of Rehabilitation ResearchandDevelopment, vol. 38, no. 1, Jan. 2001.

[11] - VASEGHI S. V., ***"Advanced Digital Signal Processing and Noise Reduction, Second Edition"*** , John Wiley & Sons Ltd, Chapter 11, pp. 333-354.

[12] - NURUZZAMAN, M., ***"Digital Audio Fundamentals in MATLAB"***, King Fahd University of Petroleum and Minerals, EED, 2010.

[13] - ANTUNES JÚNIOR, I., ***"Redução de ruído em sinais de voz usando curvas especializadas de modificação dos coeficientes da transformada em co-seno."*** Tese de Doutorado - 2006., Escola Politécnica, Universidade de São Paulo, São Paulo. Recuperado em 2014-08-26, de http://www.teses.usp.br/teses/disponiveis/3/3142/tde-05092006-103643/

[14] – Audi Bang-Olufsen - http://www.bang-olufsen.com/~/media/Files/Press%20releases/2014-03%20Bang%20%20Olufsen%20Audi%20TT%20Geneva%20PR%20UK_final.docx – acessado em 20/08/2014

[15] - 6.863J/9.611J Natural Language Processing: Fall 2012 - disponível em http://web.mit.edu/6.863/share/data/corpora/timit/dr4-maeb0/ (acessado em 23/08/2014)

[16] - ZAVAREHEI E. Boll Spectral Subtraction Algotrithm - disponível em http://www.mathworks.com/matlabcentral/fileexchange/7675-boll-spectral-subtraction (acessado em 10/08/2014)

ANEXOS - ALGORITMOS

ANEXO 1 – ENTRADA.m

```
%% Áudio principal
[s_t,fs] = audioread('sx90.wav');
%The thinker is a famous sculpture
s_t = s_t / max(abs(s_t));
s_t=[zeros(2870,1) ; s_t]; %IS da subtração espectral
%
%% Ruído genérico gaussiano
SNR=4.5;
sigma=std(s_t,1)*10^(-SNR/20);
r_t=sigma*randn(1,length(s_t))';
%
%% Criação dos áudios de cada microfone
%
n = 5;      % Quantidade de microfones do array
d = 0.05;    % Distâncida entre os microfones (m);
theta = 80;       % Ângulo de incidencia da fonte no array
%
[ x_t ] = CRIA_MICS( s_t, r_t, fs, n, d, theta);
%
```

ANEXO 1A – CRIA_MICS.m

```
function [ audios ] = CRIA_MICS( fonte, ruido, fs, quantidade_mic,
distancia_mic, angulo);

%% Cálculo de delay entre microfones
vel_som = 340;           % Velocidade de propagação do som: 340 m/s
delay_tempo = ( distancia_mic *  cos(angulo*(pi/180)) ) / vel_som;

%% Sons captados por cada microfone:
mic(:,1) = fonte;
x_linha(:,1) = mic(:,1) + ruido;

for i = 2 : quantidade_mic,
    mic(:,i) = delayseq(fonte,(i-1)*delay_tempo,fs);
    x_linha(:,i) = mic(:,i) + ruido(1:end);

end

audios = x_linha;

end
```

ANEXO 2 – PROCESSO.m

```
%
%% Variáveis do sistema
n = 5;      % Quantidade de microfones do array
theta = 80; % Ângulo de incidencia no microfone principal
d = 0.05;   % Distâncida entre os microfones (m);
%
%% Microfone de referência
x1_t = x_t(:,1);
%
%% Função para reduzir ruído apelas pela Subtração Espectral
[ mboll_t ] = SSBoll79( m_t, fs, 0.25);
[ m_t ] = SUBSPEC( x1_t, fs, 480);
[ m_t ] = [ m_t ; zeros(length(x1_t) - length(m_t),1)];
%
%% Função para reduzir ruído no áudio com base no Array de microfones
[ z_t ] = DS_BEAM( x_t, fs, n, d, theta);
%
%% Função para reduzir ruído no áudio com base no cascateamento
[ y_t ] = SUBSPEC( z_t, fs, 480);
[ y_t ] = [ y_t ; zeros(length(x1_t) - length(y_t),1)];
%
```

ANEXO 2A – DS_BEAM.m

```
function [ limpo ] = DS_BEAM( entrada, fs, quantidade_mic,
distancia_mic, angulo)

%% Cálculo de delay entre microfones
vel_som = 340;          % Velocidade de propagação do som: 340 m/s
const = cos(angulo*(pi/180));
delay_tempo = distancia_mic *  const / vel_som;

%% Corrigindo o delay dos microfones
reproc(:,1) = entrada(:,1);

for i = 2 : quantidade_mic,
    reproc(:,i) = delayseq(entrada(:,i),-(i-1)*delay_tempo,fs);
end
limpo = sum(reproc')'/quantidade_mic;

end
```

ANEXO 2B – SUBSPEC.m

```
function [ limpo ] = SUBSPEC( sujo, fs, bloco)

%% Constantes da Supressão Espectral
alfa = exp(-1/(4*fs/(bloco/2)));
beta = exp(-1/(1*fs/(bloco/2)));
b = 2;
n = 1;

%% Cálculo da quantidade de blocos existentes
num_blocos = floor(length(sujo)/bloco);

%% Chute inicial ruído
var=1e-3;
sd=sqrt(var);
R_b = (sd*sqrt(bloco))^b*(ones(1,bloco))';

%% Interações para Subtração Espectral
limpo = [];
janela = hann(bloco);

aux1 = (zeros(1,bloco))';

for i = 1:(num_blocos*2-1)

    modulo_Z_b = (abs(fft( sujo((i-1)*bloco/2 + 1:((i+1)*bloco/2))
.*sqrt(janela)))).^b;
    fase_Z_b = (phase(fft(sujo((i-1)*bloco/2 +
1:((i+1)*bloco/2)).*sqrt(janela))));

    %Comparações
    for k=1:bloco
        if modulo_Z_b(k) >= R_b(k)
            R_b(k) = R_b(k)*alfa + modulo_Z_b(k)*(1 - alfa);
        else
            R_b(k) = R_b(k)*beta + modulo_Z_b(k)*(1 - beta);
        end;
    end;

    S_b = (modulo_Z_b - n.*R_b);

    %Overlap and Add
    aux2 = aux1;
    aux1 = real(ifft((S_b.^(1/b)).*exp(j*fase_Z_b)));
    aux3 = aux1(1:bloco/2).*sqrt(janela(1:bloco/2))...
        + aux2(bloco/2+1:bloco).*sqrt(janela(bloco/2+1:bloco));

    limpo = [limpo; aux3];

end;

end
```

ANEXO 3 – SAIDA.m

```
%% Áudios de saída
%
tempo = ceil(length(s_t)/fs);
%
a1 = audioplayer(x_t(:,1), fs);% Áudio gravado por apenas um
microfone
play(a1); pause(tempo);
%
a2 = audioplayer(m_t, fs);% Áudio filtrado apenas supressão espectral
play(a2); pause(tempo);
%
a3 = audioplayer(z_t, fs);% Áudio filtrado apenas com array
play(a3); pause(tempo);
%
a4 = audioplayer(y_t, fs);% Áudio filtrado com array e supressão
espectral
play(a4); pause(tempo);
%
```

ANEXO 4 – RESULTADOS_subjetivos.m

```
clear all
close all
clc

ENTRADA; PROCESSO;

title_01 = 'Sinal s(t)'; title_02 = 'Sinal x_1(t)';
title_03 = 'Sinal m(t)'; title_04 = 'Sinal z(t)';
title_05 = 'Sinal y(t)';
%
%% Sinais de Áudio
t = linspace(0, length(x_t)/fs, length(x_t));
%
% Comparação Ruidoso x Fonte e Ruidoso x Array ----------------

figure('name','Comparação de Sinais','Color','white',...
        'PaperUnits', 'centimeters','PaperPosition',[10 1 24 24],...
        'Units','centimeters','Position',[10 1 24 24]);
for i =1:4
subplot(4,1,i);
plot(t,x1_t,'k'); hold on;
xlabel('Tempo (s)'); ylabel('Amplitude');
axis([0 length(x_t)/fs -1 1]);
end

subplot(4,1,1);
plot(t,s_t,'r');
title('Comparação x_1(t) com s(t)');
legend(title_02, title_01,'Location','EastOutside');

subplot(4,1,2);
plot(t,m_t,'b');
title('Comparação x_1(t) com m(t)');
legend(title_02, title_03,'Location','EastOutside');
```

```
subplot(4,1,3);
plot(t,z_t,'g');
title('Comparação x_1(t) com z(t)');
legend(title_02, title_04,'Location','EastOutside');

subplot(4,1,4);
plot(t,y_t,'c');
title('Comparação x_1(t) com y(t)');
legend(title_02, title_05,'Location','EastOutside');

print('figura_01', '-dpng', '-r300');
% ----------------------------------------------------------///
%
%% Espectrogramas
%Comparação com Filtrado ----------------------------------------
figure('name','Espectrograma','Color','white',...
        'PaperUnits', 'centimeters','Paperposition',[10 1 24 7],...
        'Units','centimeters','Position',[10 1 24 8]);

specgram(s_t,512,fs);
colorbar; caxis([-100 20]);
title(title_01);

print('figura_02', '-dpng', '-r300');
% ----------------------------------------------------------///
%
%Comparação Fonte x Ruidoso x Array -------------------------
figure('name','Espectrogramas','Color','white',...
        'PaperUnits', 'centimeters','PaperPosition',[10 1 24 24],...
        'Units','centimeters','Position',[10 1 24 24]);

subplot(4,1,1);
specgram(x1_t,512,fs);
colorbar; caxis([-100 20]);
title(title_02);

subplot(4,1,2);
specgram(m_t,512,fs);
colorbar; caxis([-100 20]);
title(title_03);

subplot(4,1,3);
specgram(z_t,512,fs);
colorbar; caxis([-100 20]);
title(title_04);

subplot(4,1,4);
specgram(y_t,512,fs);
colorbar; caxis([-100 20]);
title(title_05);


print('figura_03', '-dpng', '-r300');
% ----------------------------------------------------------///
%
%
```

ANEXO 5 – RESULTADOS_objetivos.m

```
clear all
close all
clc

for i = 1:10;

    ENTRADA
    PROCESSO


    % Valores para comparaões
    SNR(i,1)=SNRdB(s_t,s_t);
    SNR(i,2)=SNRdB(s_t,r_t);
    L=34;
    SegSNR(i,1)=SegSNRdB_fast(s_t,s_t,L);
    SegSNR(i,2)=SegSNRdB_fast(s_t,r_t,L);
    PESQ(i,1)=pesq(s_t,s_t,fs);
    PESQ(i,2)=pesq(s_t,r_t,fs);


    SNR(i,3)=SNRdB(s_t,x1_t);
    SNR(i,4)=SNRdB(s_t,m_t);
    SNR(i,5)=SNRdB(s_t,mboll_t);
    SNR(i,6)=SNRdB(s_t,z_t);
    SNR(i,7)=SNRdB(s_t,y_t);
    SNR(i,8)=SNRdB(s_t,yboll_t);

    SegSNR(i,3)=SegSNRdB_fast(s_t,x1_t,L);
    SegSNR(i,4)=SegSNRdB_fast(s_t,m_t,L);
    SegSNR(i,5)=SegSNRdB_fast(s_t,mboll_t,L);
    SegSNR(i,6)=SegSNRdB_fast(s_t,z_t,L);
    SegSNR(i,7)=SegSNRdB_fast(s_t,y_t,L);
    SegSNR(i,8)=SegSNRdB_fast(s_t,yboll_t,L);

    PESQ(i,3)=pesq(s_t,x1_t,fs);
    PESQ(i,4)=pesq(s_t,m_t,fs);
    PESQ(i,5)=pesq(s_t,mboll_t,fs);
    PESQ(i,6)=pesq(s_t,z_t,fs);
    PESQ(i,7)=pesq(s_t,y_t,fs);
    PESQ(i,8)=pesq(s_t,yboll_t,fs);


end

clc
disp(        '---- Sinal ------ SNR (dB) ---- SegSNR(dB) ----- PESQ -
---');
disp(sprintf('     s(t)           %.2f            %.2f           %.2f
',...
    mean(SNR(:,1)),mean(SegSNR(:,1)),mean(PESQ(:,1))))
disp(sprintf('     r(t)          %.2f         %.2f          %.2f
',...
    mean(SNR(:,2)),mean(SegSNR(:,2)),mean(PESQ(:,2))))
disp(        '-----------------------------------------------------
---');
```

```
disp(sprintf('     x_1(t)         %.2f           %.2f           %.2f
',...
    mean(SNR(:,3)),mean(SegSNR(:,3)),mean(PESQ(:,3))))
disp(sprintf('     m(t)           %.2f           %.2f           %.2f
',...
    mean(SNR(:,4)),mean(SegSNR(:,4)),mean(PESQ(:,4))))
disp(sprintf('     mboll(t)       %.2f           %.2f           %.2f
',...
    mean(SNR(:,5)),mean(SegSNR(:,5)),mean(PESQ(:,5))))
disp(sprintf('     z(t)           %.2f           %.2f           %.2f
',...
    mean(SNR(:,6)),mean(SegSNR(:,6)),mean(PESQ(:,6))))
disp(sprintf('     y(t)           %.2f           %.2f           %.2f
',...
    mean(SNR(:,7)),mean(SegSNR(:,7)),mean(PESQ(:,7))))
disp(sprintf('     yboll(t)       %.2f           %.2f           %.2f
',...
    mean(SNR(:,8)),mean(SegSNR(:,8)),mean(PESQ(:,8))))
```