



DOUGLAS HENRIQUE SIQUEIRA ABREU

**REDUÇÃO DE INTERFERÊNCIAS DESTRUTIVAS EM
ARRANJOS DE ALTO-FALANTES UTILIZANDO
PROCESSAMENTO DIGITAL DE SINAIS**

LAVRAS – MG

2017

DOUGLAS HENRIQUE SIQUEIRA ABREU

**REDUÇÃO DE INTERFERÊNCIAS DESTRUTIVAS EM ARRANJOS DE
ALTO-FALANTES UTILIZANDO PROCESSAMENTO DIGITAL DE SINAIS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

Prof. DSc. Wilian Soares Lacerda

Orientador

LAVRAS – MG

2017

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Abreu, Douglas Henrique Siqueira.

Redução de interferências destrutivas em arranjos de alto-falantes utilizando processamento digital de sinais / Douglas Henrique Siqueira Abreu. – Lavras : UFLA, 2017.

114 p. : il.

Dissertação (mestrado acadêmico) –Universidade Federal de Lavras, 2017.

Orientador: Prof. DSc. Wilian Soares Lacerda.

Bibliografia.

1. Processamento digital de sinais de áudio. 2. DSP. 3. Controle *comb filter* em arranjos de alto falantes. I. Lacerda, Wilian Soares. . II. Título


DOUGLAS HENRIQUE SIQUEIRA ABREU

**REDUÇÃO DE INTERFERÊNCIAS DESTRUTIVAS EM ARRANJOS DE
ALTO-FALANTES UTILIZANDO PROCESSAMENTO DIGITAL DE SINAIS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

APROVADA em 02 de Outubro de 2017.

Prof. DSc. Demóstenes Zegarra Rodriguez UFLA
Prof. DSc. Alexander Mattioli Pasqual UFMG


Prof. DSc. Wilian Soares Lacerda
Orientador

**LAVRAS – MG
2017**

Aos meu pais, irmãs e amigos.

AGRADECIMENTOS

Ao meu Orientador Wilian Soares Lacerda sou grato pela orientação e pela oportunidade de trabalhar com algo de meu interesse.

À Empresa Expert Electronics que prestou auxílio fundamental para o desenvolvimento deste trabalho.

Ao Centro Universitário de Lavras pela disponibilização do espaço para realização dos testes.

À minha família, pelo o apoio durante esta jornada.

À minha namorada Isabella pelo apoio nas horas mais difíceis durante a finalização da dissertação.

Ao amigo Renan Lopes, pela ajuda em escolher o tema durante uma conversa que tivemos em Paulínia.

Ao Colega Roberto Braga, pelo o incentivo a realizar a inscrição no processo seletivo PPGESISA.

Aos Membros do colegiado do PPGESISA, pela confiança, e convite para ser representante dos discentes do programa.

À CAPES, pelo apoio financeiro.

Ao amigo Serni Sarraf e seu pai Samin Sarraf, pela recepção em sua residência possibilitando as reuniões na empresa Expert Electronics.

A todos que estiveram comigo durante a jornada do curso, apoiando em todas as horas.

RESUMO

O uso de dispositivos para processamento digital de áudio, em tempo real, está se tornando cada vez mais comum. Desde os primeiros estudos, na área de áudio e acústica, é mostrada a possibilidade de controlar a frente de onda sonora reproduzida por diversas fontes. Com o avanço da tecnologia, a difusão dos processadores digitais de sinais denominados DSP, e a diminuição do custo desta tecnologia, foi proposto um dispositivo capaz de reduzir interferências destrutivas denominadas *comb filter* em um arranjo de alto-falantes. O dispositivo é de fácil utilização e configurável através de sua própria interface. Foi desenvolvida uma função de aproximação capaz de calcular parâmetros a serem inseridos em um DSP com a finalidade de alterar as características do áudio original, como fase e tempo, realizando o controle de interferências destrutivas em um ponto determinado no espaço. A função foi programada, e através do uso de plataformas específicas para programação de microcontroladores e DSPs, foi implementada em um protótipo capaz de realizar tal controle. Foram realizadas medições acústicas do sistema de som contendo o dispositivo desenvolvido, as medições são em um ambiente controlado acusticamente, com a finalidade de provar a eficiência do dispositivo. As medições abrangem dados como resposta em frequência, resposta ao impulso e distorção harmônica total (THD) do sistema de teste montado. Os resultados obtidos apresentam uma resposta em frequência mais homogênea no ponto de análise, controlando um problema comum na composição de ondas sonoras radiadas por duas ou mais fontes sonoras, o *comb filter*. Foi possível mostrar uma melhora da THD do sistema com o dispositivo de controle. Conclui-se que o dispositivo proposto é capaz de reduzir interferências destrutivas em um arranjo de alto falantes, melhorando a qualidade do áudio no ponto do escolhido no espaço, proporcionando uma nova experiência a usuários do dispositivo.

Palavras-chave: DSP. Direcionamento de áudio. *Comb filter*. Processamento digital de áudio. Varredura senoidal. Sistemas de som. Áudio Espacial

ABSTRACT

The use of devices for digital audio processing in real time is becoming more and more common. From the earliest studies in the area of audio and acoustics it is shown the possibility of controlling the wavefront reproduced by several sources. With the advancement of technology, the diffusion of the digital signal processors called DSP, and the reduction of the cost of this technology, a device capable of reducing destructive interferences denominated comb filter in an arrangement of loudspeakers has been proposed. The device is user-friendly and configurable through its own interface. An approximation function was developed to calculate parameters to be inserted in a DSP in order to alter the characteristics of the original audio, such as phase and time, controlling the destructive interferences at a given point in the space. The function was programmed, and through the use of specific platforms for programming microcontrollers and DSPs, was implemented in a prototype capable of performing such control. Acoustic measurements were made of the sound system containing the developed device, the measurements are in an acoustically controlled environment, in order to prove the efficiency of the device. The measurements cover data such as frequency response, impulse response and total harmonic distortion (THD) of the mounted test system. The results obtained present a more homogeneous frequency response at the point of analysis, controlling a common problem in the composition of sound waves radiated by two or more sound sources, comb filter. It was possible to show an improvement of the THD of the system with the control device. It is concluded that the proposed device is able to reduce destructive interference in a loudspeaker array, improving the quality of the audio at the chosen point in the space, providing a new experience for the users of the device.

Keywords: DSP. Audio Directing. comb filter. Digital audio processing. Sine Sweep. Sound systems. Spatial audio

LISTA DE FIGURAS

Figura 1.1 – Sistema de atraso para inclinar a frente de onda de um sistema de fontes sonoras em linha	14
Figura 2.1 – Características da onda e sua forma de interagir com o meio	17
Figura 2.2 – Relações de fase entre as formas de onda com a mesma amplitude e frequência	18
Figura 2.3 – Interferência construtiva de ondas	19
Figura 2.4 – Interferência destrutiva de ondas	19
Figura 2.5 – Sobreposição de ondas gerando <i>comb filter</i>	20
Figura 2.6 – Exemplo de <i>comb filter</i> com atraso de 1 ms	21
Figura 2.7 – Uso de geometria na determinação do campo distante em pistão circular plano	22
Figura 2.8 – As características de desempenho de vários microfones	23
Figura 2.9 – Padrão de captação omnidirecional	24
Figura 2.10 – Padrão de altas frequências em microfone omnidirecional	24
Figura 2.11 – Sistema de som duas vias	25
Figura 2.12 – Função de diretividade de um pistão	26
Figura 2.13 – A transformada de Hilbert ideal: a função de resposta ao impulso (a), o módulo(b), e a fase (c) da função de transferência da transformada de Hilbert	29
Figura 2.14 – Menu para seleção de blocos do SigmaStudio	31
Figura 2.15 – Exemplo de programa em SigmaStudio baseado em blocos	32
Figura 2.16 – Espectro de Fourier do sinal de teste (acima) e do filtro inverso (abaixo) . .	36
Figura 2.17 – Comparação entre as medidas MLS e varredura senoidal	37
Figura 2.18 – Filosofia básica utilizada na PESQ	38
Figura 3.1 – Diagrama de blocos Processador digital de Sinais de Áudio	41
Figura 3.2 – Foto da placada do protótipo	41
Figura 3.3 – Foto da placada do LCD e <i>push botton</i> do protótipo	42
Figura 3.4 – Foto externa do processador	42
Figura 3.5 – Representação do triângulo formado pelas fontes sonoras e o ponto de direcionamento	44
Figura 3.6 – Blocos da programação do protótipo no Sigma Studio	46
Figura 3.7 – Sub-bloco da programação da função de fase do protótipo no Sigma Studio	47
Figura 3.8 – Código em C da a função de rotação linear de fase	48
Figura 3.9 – Parâmetros de direcionamento	49

Figura 3.10 – Ajuste de volume da saída	49
Figura 3.11 – Roteamento de canais	50
Figura 3.12 – Projeto da caixa acústica do sistema de som 2 vias	50
Figura 3.13 – Curvas de impedância e fase dos tweeters	52
Figura 3.14 – Curvas de impedância e fase dos sistemas de caixas acústicas 2 vias	53
Figura 3.15 – Curvas de resposta em frequência a 0° e 45° do sistema de caixa acústica 2 vias	53
Figura 3.16 – Curva de resposta ECM8000	54
Figura 3.17 – Gráfico polar ECM8000	54
Figura 3.18 – Medições de canal simples e duplo	56
Figura 3.19 – Captura da tela principal do programa REW	58
Figura 3.20 – Coeficientes de absorção sonora Sonex	59
Figura 3.21 – Cenário de teste do sistema de medição acústica	60
Figura 3.22 – Cenário de teste com duas fontes sonoras	60
Figura 3.23 – Cenário de teste com três fontes sonoras	61
Figura 4.1 – Foto do cenário de teste	62
Figura 4.2 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 1 metro e 60% de porcentagem de direcionamento	63
Figura 4.3 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 1 metro e 80% de porcentagem de direcionamento	63
Figura 4.4 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 1 metro e 100% de porcentagem de direcionamento	64
Figura 4.5 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 2 metros e 80% de porcentagem de direcionamento	65
Figura 4.6 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 2 metros e 100% de porcentagem de direcionamento	65

Figura 4.7 – Gráfico da resposta em frequência da medição com três fontes, distância de 1,5 metros entre elas, distância de direcionamento de 2 metros e 13% de porcentagem de direcionamento	66
Figura 4.8 – Gráfico da resposta em frequência da medição com três fontes, distância de 1,5 metros entre elas, distância de direcionamento de 2 metros e 27% de porcentagem de direcionamento	66
Figura 4.9 – Gráfico da resposta em frequência da medição com três fontes, distância de 1,5 metros entre elas, distância de direcionamento de 2 metros e 40% de porcentagem de direcionamento	66
Figura 4.10 – Gráfico da resposta em frequência da medição com três fontes, distância de 1,5 metros entre elas, distância de direcionamento de 2 metros e 80% de porcentagem de direcionamento	67
Figura 4.11 – Distorção Harmônica Total com 3 fontes e 13% de direcionamento	67
Figura 4.12 – Distorção Harmônica Total com 3 fontes e 27% de direcionamento	68
Figura 4.13 – Distorção Harmônica Total com 3 fontes e 80% de direcionamento	68
Figura 4.14 – Resposta ao impulso do sistema com duas fontes sonoras	69
Figura 4.15 – Resposta ao impulso do sistema com três fontes sonoras	70
Figura 1 – Resposta em frequência do ambiente de teste.	76
Figura 2 – Cenários de teste.	78
Figura 3 – Resposta do sistema no cenário 1.	79
Figura 4 – Resposta do sistema no cenário 2.	79
Figura 5 – Resposta do sistema no cenário 3.	79
Figura 6 – Comparação da resposta do sistema nos cenários 1, 2 e 3.	80
Figura 7 – Resposta do sistema no cenário 4.	80
Figura 8 – Resposta do sistema no cenário 5.	80
Figura 9 – Resposta do sistema no cenário 6.	80
Figura 10 – Comparação da resposta do sistema nos cenários 4, 5 e 6.	81

LISTA DE TABELAS

Tabela 3.1 – Parâmetros dos transdutor de 4,5"	51
Tabela 1 – Resultados utilizando a recomendação ITU-T P.862	79

SUMÁRIO

1	Introdução	12
1.1	Objetivos	12
1.1.1	Objetivos Específicos	13
1.2	Justificativa	13
1.3	Organização do texto	14
2	REFERENCIAL TEÓRICO	15
2.1	Fundamentos de Áudio e Acústica	15
2.1.1	Decibel	15
2.1.2	Frequência e Comprimento de Onda	16
2.1.3	Sobreposição de ondas	18
2.1.4	<i>Comb Filter</i> (filtro pente)	19
2.2	Transdutores	21
2.2.1	Pistão Circular Plano	21
2.2.2	Microfones	22
2.2.2.1	Microfones Omnidirecional	23
2.2.3	O Alto-falante	24
2.2.3.1	Tipos de Sistema de Alto-falantes	25
2.2.3.2	Diretividade do Pistão e do Alto-falante	25
2.3	Processamento Digital de Sinais (DSP)	27
2.3.1	Amostragem	28
2.3.2	Transformada de Hilbert	28
2.3.3	Processadores digital de sinais	29
2.3.4	SigmaStudio: uma plataforma de programação gráfica	30
2.4	Métodos de medição acústica	31
2.4.1	Sequência de Máximo Tamanho	32
2.4.2	Varredura senoidal	34
2.5	Estimador intrusivo de qualidade de voz	37
3	MATERIAIS E MÉTODOS	40
3.1	Descrição do Hardware	40
3.2	Desenvolvimento do modelo para inserir atraso e variação de fase	42
3.3	Desenvolvimento do Software	45

3.4	Interface do Protótipo	48
3.5	Caixas de som 2 vias	50
3.6	Microfone Behringer ECM8000	52
3.7	Softwares de medições acústicas	55
3.7.1	Smaart: Software de análise de sinais de som	55
3.7.2	REW (<i>Room Eq Wizard</i>)	56
3.8	Cenário de Teste	58
3.9	Cenário de Medição	59
4	Resultados	62
4.1	Resultados da medição com duas fontes sonoras (esquerda e direita)	63
4.2	Resultados da medição com três fontes sonoras (esquerda, centro e direita)	64
4.3	Análise da resposta ao impulso do sistema	69
5	CONCLUSÃO	71
5.1	Dificuldades encontradas	72
5.2	Proposta de continuidade	72
	REFERÊNCIAS	73
	APENDICE A – Outros resultados obtidos	76
	APENDICE B – principal (<i>main</i>) do protótipo do processador	82
	APENDICE C – Código do DSP do protótipo do processador	98
	ANEXO A – Esquemático da placa do processador digital de sinais de áudio	105

1 INTRODUÇÃO

O fenômeno sonoro nada mais é que a perturbação de um meio formando uma onda, esse meio pode ser, por exemplo o ar, objetivo de estudo deste trabalho. Existem várias maneiras de estudar cientificamente este fenômeno. A acústica física estuda a parte material do fenômeno sonoro, enquanto a psicoacústica trata da percepção do fenômeno pelos sentidos. O estudo do processamento tanto analógico quanto digital do som, envolve também diversas áreas como matemática, física, eletrônica, computação, dentre outras.

Algumas necessidades influenciam na necessidade de controlar o som em pontos específicos do espaço em campo aberto e fechado. Uma delas é a diminuição da perturbação de sossego em locais externos (cidades, residências, etc.) não participantes a eventos como por exemplo a realização de shows, que utilizam sistema de som de alta intensidade sonora. Outro campo para o controle de som é a busca por uma melhor qualidade de reprodução sonora em um determinado ponto, abrangendo uma resposta em frequência mais homogênea, uma taxa de distorção harmônica total mais baixa. Existem também competições com a finalidade de atingir em um ponto específico um maior nível de pressão sonora (SPL – *Sound Pressure Level*), neste tipo de competição cada competidor com seu sistema de som busca atingir um maior SPL no ponto de captação de um microfone.

No encontro de duas ondas sonoras de mesma frequência, ocorre a soma ou cancelamento total ou parcial. Esse fenômeno ocorre e alterna entre a soma e o cancelamento de acordo com a fase da cada uma no momento de encontro. É possível controlar a fase de uma reprodução sonora, utilizando processamento digital de sinais. Esse modo de controle é chamado de controle ativo, pois processa o áudio antes do sinal ser enviado para o amplificador do sistema de som.

É possível, através do processamento digital de sinais, atrasar o tempo de reprodução de uma determinada fonte sonora. Esse tipo de processamento faz com que o mesmo som emitido por duas ou mais fontes sonoras distintas, e em distâncias diferentes ao ouvinte, chegue no mesmo instante de tempo em um ponto específico no espaço.

1.1 Objetivos

O presente estudo tem por objetivo estudar, investigar e testar métodos de soma, cancelamentos de ondas sonoras e controle de sinais de áudio. Após a avaliação dos métodos, desenvolver um protótipo capaz de executar o controle de fase do sistema de som, através do

ajuste de tempo do sinal de áudio enviado a cada uma das fontes e também do ajuste igual de fase em todas as frequências. Controle é realizado em tempo real utilizando um processador digital de sinais.

1.1.1 Objetivos Específicos

- Encontrar o estado da arte em métodos de soma e cancelamento de som.
- Compreender o funcionamento básico dos transdutores.
- Entender métodos e softwares de análises de dados e representação de resultados, utilizando resposta em frequência e níveis de pressão sonora.
- Compreender o funcionamento de equipamentos de processamento de sinais digitais de áudio presente no mercado, e propor um dispositivo que auxilie no desenvolvimento do projeto.
- Desenvolver um método de controle para redução de influências destrutivas em um arranjo de alto-falantes, tendo variáveis como distâncias e porcentagem, alterando o tempo e a fase do sinal.
- Avaliar o impacto na qualidade do sinal transmitido.
- Testar o método proposto e medir sua eficiência.

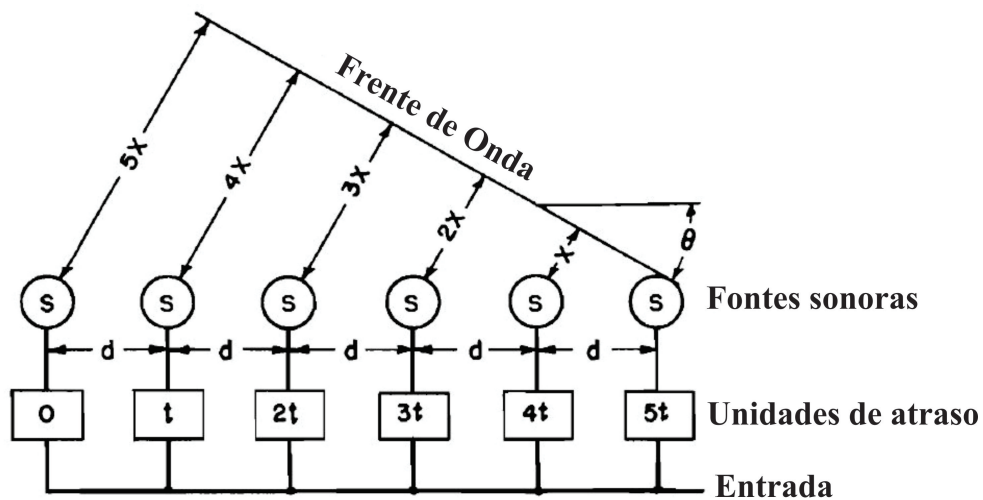
1.2 Justificativa

A área voltada a processamento digital de sinais de áudio e acústica permite diversas pesquisas de controle. O mercado não apresenta um produto baseado em manipulação ativa do sinal específico para controlar de influências destrutivas em arranjos de alto falantes. Este pressuposto motiva a criação de um protótipo voltado a essa finalidade, que seja capaz de ser utilizado por um usuário leigo, e que através da inserção de algumas variáveis como distâncias e porcentagem de direcionamento, execute a redução das influências destrutivas. O conceito de *Line Array* visa eliminar essas influências através do acoplamento de transdutores.

Para o desenvolvimento do protótipo, é necessária a criação de um modelo matemático, encontrar o estado da arte em sobreposição de ondas sonoras e como controla-las por processamento digital em tempo real. Com a modificação da fase de ondas complexas reproduzidas por

um sistemas de som, é possível modificar os pontos de soma e cancelamento de ondas sonoras em um meio como o ar, viabilizando o objetivo deste trabalho. Segundo Olson (1957), a direção e a forma da frente de onda produzidas por uma série de fontes sonoras podem ser alteradas pela introdução de um padrão de atraso na excitação das fontes, como motrado na Figura 1.1. É possível criar qualquer forma de frente de onda através da introdução do deslocamento de fase apropriado nas fontes sonoras (OLSON, 1957).

Figura 1.1 – Sistema de atraso para inclinar a frente de onda de um sistema de fontes sonoras em linha



Fonte: Adaptado de Olson (1957)

1.3 Organização do texto

Este capítulo apresenta a introdução do trabalho, demonstrando uma visão geral do tema, a justificativa do trabalho e os objetivos a serem alcançados. O Capítulo 2 apresenta o Referencial Teórico, desenvolvido para uma melhor compreensão do trabalho, sendo dividido em: Fundamentos de Áudio e Acústica, Transdutores, Processamento Digital de Sinais, Métodos de medições acústica, Softwares de medição acústica e Estimador intrusivo da qualidade de voz. O Capítulo 3 apresenta os Materiais e os Métodos utilizados. O Capítulo 4 apresenta os Resultados obtidos. O Capítulo 5 apresenta a Conclusão deste trabalho. O Apêndice A apresenta um Resultado obtido durante a realização do trabalho. O Apêndice B apresenta o Código principal (*main*) do protótipo do processador. O Apêndice C apresenta o Código do DSP do protótipo do processador. Por fim, o Anexo A apresenta o esquemático da placa do processador.

2 REFERENCIAL TEÓRICO

O referencial teórico apresenta uma base de informações para o entendimento do trabalho. Foi dividido em cinco principais assuntos: Fundamentos de Áudio e Acústica, Transdutores, Processamento Digital de Sinais, Métodos de Medição Acústica e Estimador intrusivo de qualidade de voz.

2.1 Fundamentos de Áudio e Acústica

O início deste trabalho é dedicado a estabelecer uma linha base de informação para o entendimento do áudio e seus princípios. Inúmeras ferramentas estão disponíveis para quem trabalha com áudio, as mais importantes são ferramentas matemáticas. A sua aplicação é independente do tipo de sistema ou sua utilização.

2.1.1 Decibel

O decibel é uma unidade logarítmica considerado muito útil por profissionais do áudio. Ele permite uma relação das alterações do nível ouvido com parâmetros do sistema como potência, tensão ou a distância. Em geral, o decibel é uma forma de expressar o quanto um som é relevante para a percepção humana (BROWN, 2013).

A maioria das pessoas tende a imaginar variáveis físicas em termos lineares, como por exemplo, o dobro de areia que produz o dobro de concreto. Essa verdade linear não é verdade para o sentido humano da audição. Mudanças perceptíveis na intensidade e frequência de som são baseadas na variação perceptual em relação a alguma condição inicial, isso significa que o áudio é relacionado a pessoa em forma de proporção. Testes subjetivos mostraram que a potência aplicada em um alto-falante deve ser aumentada cerca de 26% para produzir uma mudança audível. Portanto, em uma escala de 1.26:1 produz a mudança mínima audível, independentemente da quantidade de energia inicial. Uma potência inicial de 1 W, quando é aumentada para 1,26 W, é percebida uma mudança audível e respectivamente uma potência inicial de 100 W, é necessário ser aumentada para 126 W para ser percebida (BROWN, 2013).

Para manter uma proporcionalidade, o áudio utiliza logaritmo na base 10. Para simplificar, quando um parâmetro como por exemplo, potência, tem o seu nível alterado levando em consideração log de 10, o resultante dessa mudança de nível entre duas potências é expressa em

Bels. A conversão para decibéis é o dimensionamento da quantidade Bel por um fator de 10 (BROWN, 2013).

Por ser uma proporção entre duas quantidades, o decibel é sem dimensão, e por conseguinte, não é tecnicamente uma unidade no sentido clássico. Os valores em decibéis proporcionam uma gama mais fácil de se utilizar, o decibel é 1/10 bel, ou seja, um decibel (dB) é 10 vezes o logaritmo de base 10 da razão de duas quantidades (EVEREST; POHLMANN, 2009).

Assim, a razão de intensidades torna-se em decibéis conforme a Equação 2.1.

$$IL = 10 \log_{10} \frac{I}{I_{ref}} \quad (2.1)$$

O valor é chamado de nível de intensidade sonora (IL em decibéis) e difere da intensidade I que é definida fisicamente como potência sonora recebida por unidade de área de uma superfície (W/m^2), e respectivamente I_{ref} definida como mínima intensidade física, o menor valor da intensidade sonora ainda audível ($10^{-12}W/m^2$). O uso de decibéis é uma conveniência e o seu valor é o que mais se aproxima da forma de um ser humano ouvir (BROWN, 2013).

2.1.2 Frequência e Comprimento de Onda

Uma onda é produzida quando um meio é perturbado, esse meio pode ser ar, água, aço, terra, dentre outros. Essa perturbação produz uma oscilação na condição inicial do ambiente, a oscilação se irradia para fora a partir da fonte de perturbação. Se um segundo é utilizado como intervalo de tempo de referência, o número de oscilações acima e abaixo da condição inicial do ambiente por segundo é a frequência do evento, e é expressa em ciclos por segundos, ou Hertz (BROWN, 2013).

Como visto, na seção 2.1.1, o ouvido dos seres humanos é sensível a alterações proporcionais à potência, tensão e distância. Isto também é verdade para frequência. Se começar com a menor frequência audível 20Hz e aumentá-la a uma proporção de 2:1, o resultado é de 40 Hz, chamado intervalo de uma oitava. Se o mesmo for feito para 40 Hz o resultado será 80 Hz e assim por diante. Cada oitava maior terá duas vezes a largura de faixa da que está abaixo dela (BROWN, 2013). Isso faz com que a escala logarítmica seja apropriada para a visualização de frequências.

Se é conhecida a frequência f de uma vibração, o período de tempo T para que um ciclo da vibração se complete pode ser obtido pelo simples relacionamento da Equação 2.2.

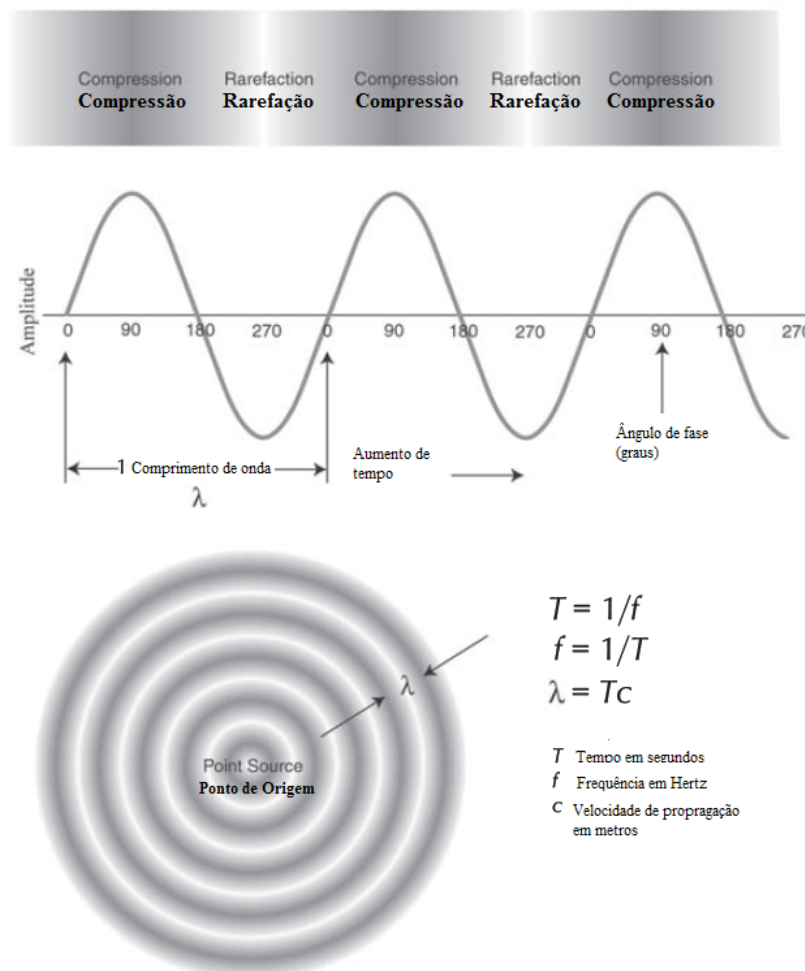
$$T = \frac{1}{f} [\text{segundos}] \quad (2.2)$$

Uma vez que a maioria das ondas se propagam e o seu período é conhecido, o seu tamanho físico pode ser calculado com a Equação 2.3 se a velocidade de propagação c é conhecida. Esse tamanho físico é conhecido como comprimento de onda, a Figura 2.1 demonstra as Equações 2.3 e 2.4.

$$\lambda = T \times c \quad (2.3)$$

$$\lambda = \frac{c}{f} \quad (2.4)$$

Figura 2.1 – Características da onda e sua forma de interagir com o meio

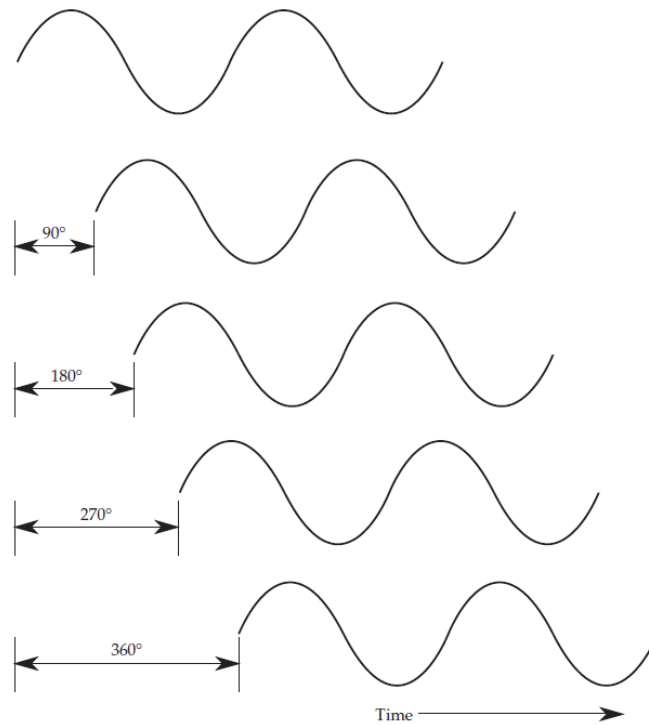


Fonte: Adaptado de Brown (2013)

2.1.3 Sobreposição de ondas

A sobreposição de ondas também é chamada de ondas complexas. As ondas complexas são formadas por duas ou mais ondas senoidais e cossenoidais periódicas e de frequência única. A amplitude de uma onda pode ser representada em função do tempo ou em uma função de rotação de fase, um ciclo completo de uma onda é de 360° (BROWN, 2013).

Figura 2.2 – Relações de fase entre as formas de onda com a mesma amplitude e frequência

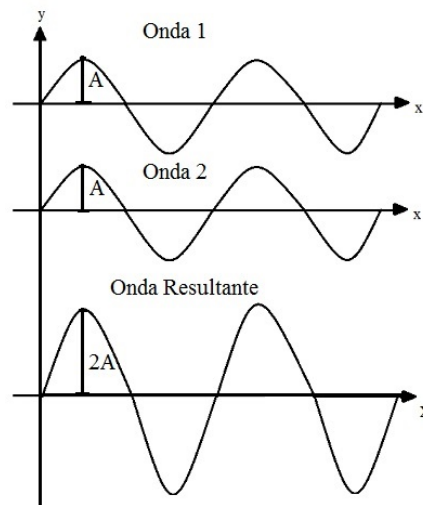


Fonte: Everest e Pohlmann (2009)

Quando várias ondas passam por um ponto de observação, a resposta da soma dessas ondas é uma onda composta, que é uma combinação de duas ou mais ondas individuais. A amplitude da soma de ondas individuais é determinada pela fase de cada uma delas, considerando duas ondas e um ponto de observação. Esse ponto de observação pode ser um microfone, e é importante destacar as duas extremidades da soma, quando essas duas ondas no ponto possuem mesma fase, o resultado é uma onda com duas vezes a amplitude das ondas individuais, ou seja, +6 dB. A outra extremidade da soma é quando as duas ondas possuem fase inversas, ou seja, uma diferença de 180° entre elas, o resultado é o cancelamento total das ondas. Um número infinito de condições podem ocorrer entre esses dois extremos (BROWN, 2013).

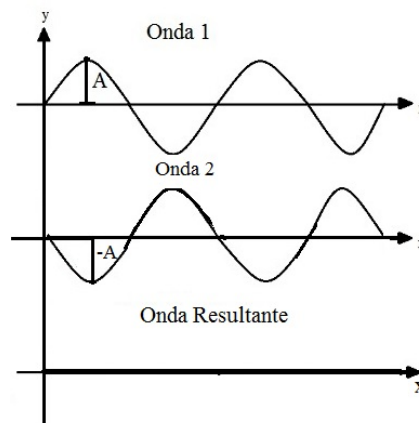
É possível observar na Figura 2.3 a situação onde duas ondas com fases iguais ocasionam em uma soma de amplitude. Quando duas ondas têm fases opostas resultando em cancelamento pode ser mostrado como na Figura 2.4 (TEIXEIRA, 2017).

Figura 2.3 – Interferência construtiva de ondas



Fonte: Teixeira (2017)

Figura 2.4 – Interferência destrutiva de ondas



Fonte: Teixeira (2017)

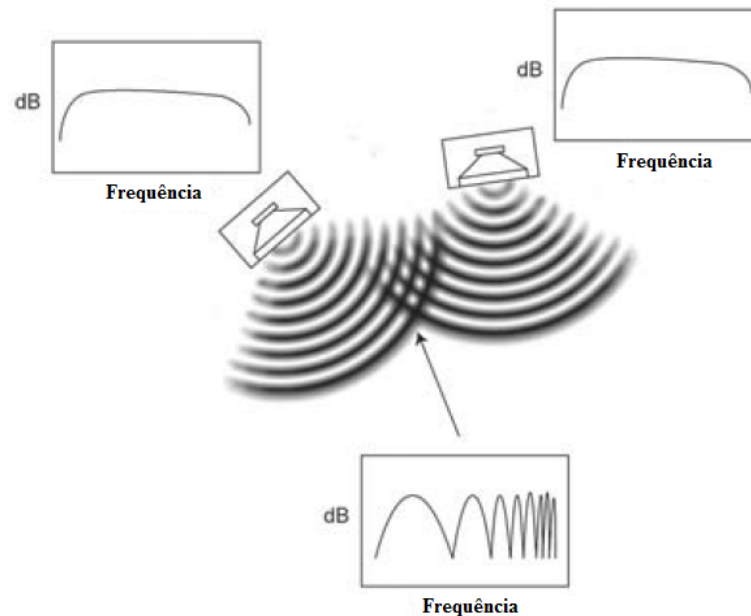
2.1.4 *Comb Filter* (filtro pente)

Em sistemas de sonorização, as interações de fase são tipicamente mais problemáticas em ondas acústicas do que em ondas eletromagnéticas. Somatórios e cancelamentos ocasionado pela variação de fase ocasionam uma mudança facilmente perceptível com a alteração drástica de nível audível (TOOLE, 2008).

Ondas sonoras radiadas por diversos alto-falantes interagem uma sobrepondo a outra. Dependendo da fase em cada frequência, a resposta em frequência da sobreposição sofre várias modificações de amplitude em todo o espectro sobreposto, gerando uma resposta diferente do padrão radiado. Essa mudança de padrão é devido aos alto-falantes cobrirem uma década ou mais do espectro sonoro, o que não resulta em um cancelamento completo da pressão sonora, e

sim um cancelamento em algumas frequências e um somatório coerente em outras. O resultado disso é uma tonalidade de coloração e uma mudança de imagem em relação ao som reproduzido e o som captado pelo ouvinte (BROWN, 2013).

Figura 2.5 – Sobreposição de ondas gerando *comb filter*



Fonte: Adaptado Brown (2013)

O efeito *comb filter* pode ocorrer tanto de ondas refletidas quanto em soma de duas fontes sonoras que reproduzem uma mesma frequência. Essa segunda opção é mais agravante, pois a intensidade sonora é maior (TOOLE, 2008).

Os cancelamentos ocorrem quando dois sons estão em oposição de fase, isto é, um comprimento de onda com período da metade do tempo do atraso. Se é conhecido o atraso entre duas fontes sonora, é possível calcular pela Equação 2.5 a primeira frequência de influência destrutiva, será aquela em que o período é duas vezes o atraso (TOOLE, 2008):

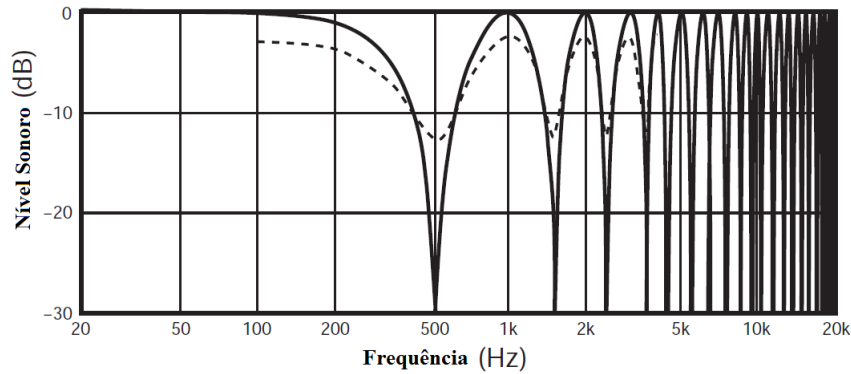
$$f = \frac{1}{T} = \frac{1}{2 \times \text{atraso}[\text{segundos}]} \quad (2.5)$$

Os cancelamentos de ordem superior ocorrem em frequências com número ímpar de meio comprimento de onda no intervalo de atraso, demonstrado na Equação 2.6 (TOOLE, 2008).

$$f_N = \frac{N}{2 \times \text{atraso}[\text{segundos}]} = \frac{1, 3, 5, 7, \text{etc.}}{2 \times \text{atraso}[\text{segundos}]} \quad (2.6)$$

A Figura 2.6 mostra a resposta em função de um *comb filter* que ocorre quando se tem 1ms (0,001s) de atraso. O primeiro cancelamento ocorre em $1 \div 2 \times 0,001 = 500\text{Hz}$. O segundo cancelamento ocorre em $3 \div 0,002 = 1500\text{ Hz}$, e assim por diante.

Figura 2.6 – Exemplo de *comb filter* com atraso de 1 ms



Fonte: Adaptado de Toole (2008)

Para encontrar os picos do sistema, onde as ondas sonoras somam, o ponto chave é que o atraso é sempre um múltiplo inteiro de comprimentos de onda, representado pela Equação 2.7 (TOOLE, 2008).

$$f_N = \frac{N}{\text{atraso}[\text{segundos}]} \quad (2.7)$$

2.2 Transdutores

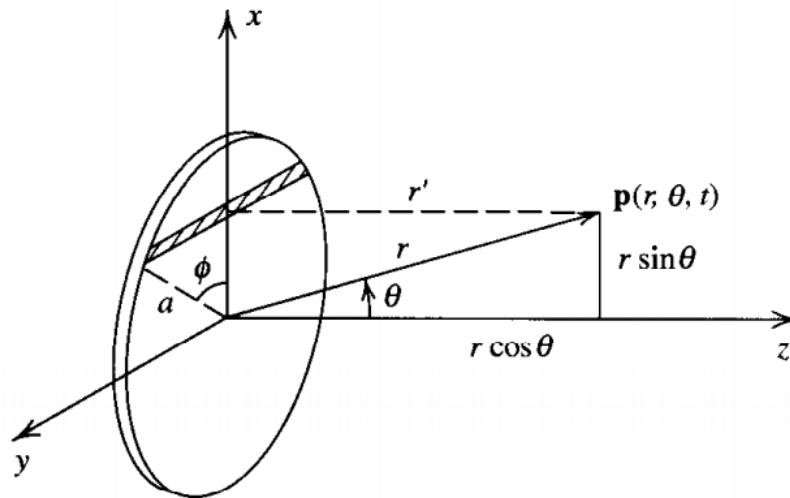
Para a compreensão dos meios de reprodução e captação sonora, torna-se importante revisar os transdutores em geral, tanto alto-falantes quando microfones.

2.2.1 Pistão Circular Plano

O pistão circular plano é visto como base de muitos transdutores como alto-falantes e microfones utilizados na prática. O conceito da reciprocidade acústica permite utilizar o modelo do pistão circular plano para atuadores tanto como radiador quanto receptor de ondas sonoras (KINSLER et al., 1999).

O pistão é uma estrutura idealizada, a obtenção de um modelo matemático é possível em situações de campo distante (*farfield*) e também em campo próximo (*nearfield*) (KINSLER et al., 1999), o modelo torna-se válido, pois no desenvolvimento deste trabalho, as situações dos testes realizados foram em uma distância entre receptor e radiador muito maior que o próprio raio do radiador (*farfield*).

Figura 2.7 – Uso de geometria na determinação do campo distante em pistão circular plano



Fonte: Kinsler, et al. (2000)

2.2.2 Microfones






Microfone pode ser definido como um equipamento eletroacústico que converte energia acústica em energia elétrica (BALLOU; CIAUDELLI; SCHMITT, 2008). Entretanto, a necessidade de se obter uma resposta mais eficiente para cada aplicação leva ao desenvolvimento de diversos tipos de microfones.

Os microfones podem ser classificados por padrões de captação, ou discriminados pela captação das diversas direções das fontes sonoras (BALLOU; CIAUDELLI; SCHMITT, 2008). Essas classificações são:

- Omnidirecional: capta o som igualmente em todas as direções.
- Bidirecional: capta o som igualmente em direções opostas (180°) e tende a zero em um ângulo de 90° a partir das duas direções preferenciais de captação.
- Unidirecional: capta o som em apenas uma direção.

Microfones são construídos com padrões simples ou de múltiplas captações, e são nomeados pelo padrão de captação. A Figura 2.8 mostra os padrões de captação e as características de vários tipos de microfones.

Figura 2.8 – As características de desempenho de vários microfones

Microfone	Omnidirecional	Bidirecional	Direcional	Supercardióide	Hípercardióide
Características de Resposta Rirrecional					
Voltagem de Saída	$E = E_0$	$E = E_0 \cos \theta$	$E = \frac{E_0}{2}(1 + \cos \theta)$	$E = \frac{E_0}{2}[(\sqrt{3}-1) + (3\sqrt{3})\cos \theta]$	$E = \frac{E_0}{4}(1 + 3 \cos \theta)$
Eficiência energética aleatória	100	33	33	27	25
Resposta da frente	1	1	∞	3.8	2
Resposta aleatória da frente	0.5	0.5	0.67	0.93	0.87
Resposta aleatória da traseira	1	1	7	14	7
Distância Equivalente	1	1.7	1.7	1.9	2
Ângulo de captação a 3 dB de atenuação	—	90°	130°	116°	100°
Ângulo de captação a 3 dB de atenuação	—	120°	180°	156°	140°

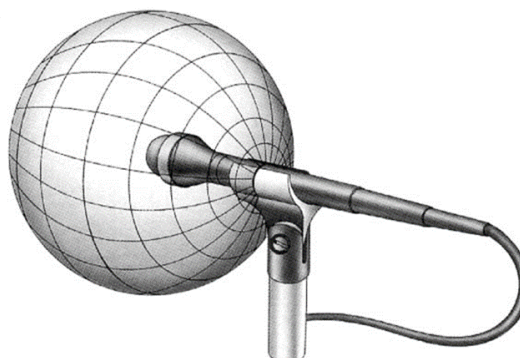
Fonte: Adaptado de Ballou, Ciaudelli e Schmitt (2013)

2.2.2.1 Microfones Omnidirecional

Omnidirecional ou esférica é a resposta polar da pressão sonora captada neste tipo de microfone. A Figura 2.9 mostra o comportamento da captação de energia acústica onde as ondas atingem o diafragma.

Os microfones omnidirecionais se tornam cada vez mais direcionais, à medida em que o comprimento da onda tende a ficar igual ou menor o diâmetro do microfone. A característica que faz com que a onda ultrapasse o objeto e atinja o diafragma, é chamada de difração, acontece

Figura 2.9 – Padrão de captação omnidirecional

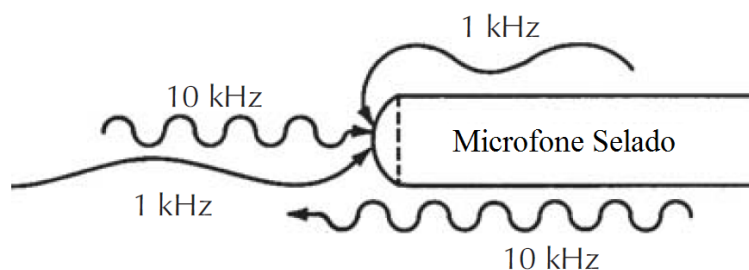


Fonte: Ballou, Ciaudelli e Schmitt (2013)

quando o comprimento de onda é maior que o objeto fazendo com que a onda contorne este objeto. À medida que o comprimento de onda se aproxima do tamanho deste objeto, ela não pode se dobrar o bastante para passar por ele e atingir o diafragma (BALLOU; CIAUDELLI; SCHMITT, 2008).

A resposta em função da direcionalidade começa a divergir quando o tamanho do diafragma é um décimo do comprimento de onda, a Equação 2.8 e a Figura 2.10 mostram isso.

Figura 2.10 – Padrão de altas frequências em microfone omnidirecional



Fonte: Adaptado de Shure (2014)

$$D = \frac{\lambda}{10} \quad (2.8)$$

2.2.3 O Alto-falante

Um sistema de reprodução de ondas sonoras consiste em um ou mais dispositivos, estes chamados de alto-falantes. Um alto-falante é chamado de transdutor eletroacústico, ou seja, um dispositivo que converte energia elétrica em energia acústica (MITCHELL, 2008).

Os alto-falantes podem ser desenvolvidos para diversas finalidades, como reproduzir graves, médios, agudos, a maioria baseiam-se em um conceito: um diafragma em movimento,

devido a um campo eletromagnético modulado em amplitude, atuando em conjunto com um ímã permanente (DICKASON, 2006).

O uso do alto-falante engloba uma enorme variedade de aplicações, porém essas aplicações têm como base uma combinação de quatro propósitos principais, são eles: comunicação, sonorização, produção de som e reprodução de som. Embora existam requisitos comuns para todos esses usos, cada um impõe sua própria demanda sobre o alto-falante (MITCHELL, 2008).

2.2.3.1 Tipos de Sistema de Alto-falantes

A forma mais simples de sistema de alto-falante utiliza um único transdutor fullrange para reproduzir todas as frequências, porém são sistemas de baixo custo e limitados em largura de banda, geralmente utilizados para reproduzir voz (MITCHELL, 2008). Em alguns casos, como nos aparelhos portáteis de música também utiliza-se um único transdutor.

Para sistemas residenciais de música, comumente é utilizada a configuração de duas vias. Esse sistema de duas vias geralmente é composto por um alto-falante de 6 ou 8 polegadas e um tweeter de domo. O sistema de duas vias é mostrado na Figura 2.11.

Figura 2.11 – Sistema de som duas vias



Fonte: Mitchell (2008)

2.2.3.2 Diretividade do Pistão e do Alto-falante

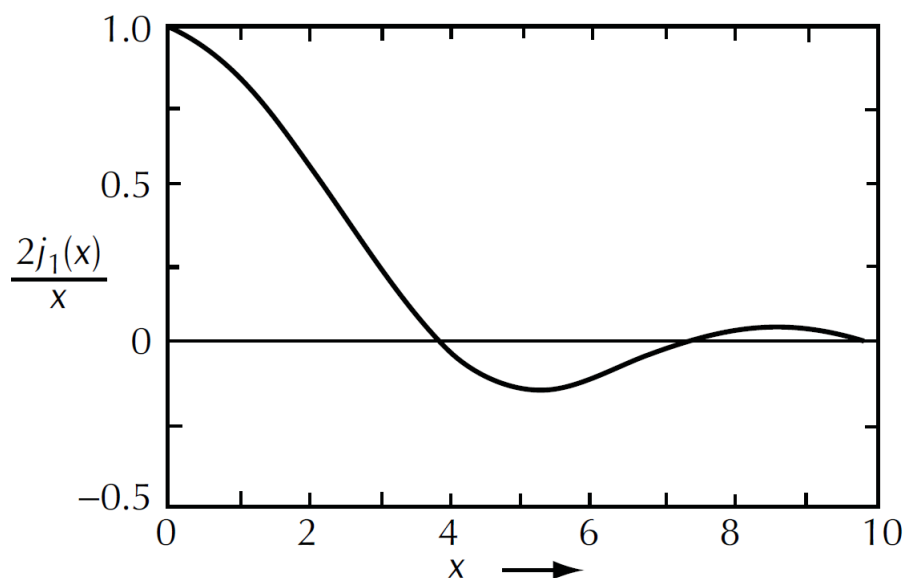
Segundo (DICKASON, 2006), os alto-falantes começam a ficar mais direcionais à medida que a frequência aumenta. Em altas frequências, a tendência é formar um feixe estreito. As frequências onde o comprimento de onda é grande em comparação à circunferência do cone (cerca do triplo do diâmetro), a radiação é aproximadamente uniforme. Quando a frequên-

cia aumenta ao ponto onde o comprimento de onda é igual ou menor que a circunferência do alto-falante, o lóbulo de radiação torna-se cada vez mais estreito.

Devido ser inevitável a natureza da diretividade, ela deve ser considerada. Uma técnica para calcular a função de diretividade do pistão, é considerar o pistão como sendo um composto de elementos diferenciais infinitesimais, cada um dos quais contribui para a radiação observada em um ponto no espaço. Essas contribuições individuais são combinados através da integração para produzir um valor para cada ponto específico no espaço.

Para chegar a uma expressão simples para a diretividade do pistão, uma hipótese deve ser cumprida: a distância desde o pistão até o ponto de observação é muito maior que o raio do pistão. O primeiro termo da relação é a dependência da pressão, velocidade, amplitude pelo tamanho do pistão e a distância a partir da origem. O segundo termo chamada de função de diretividade do pistão, é derivada a partir de uma função de Bessel, $J_1(x)$. O valor dessa função é representado graficamente na Figura 2.12. O termo ka pode ser entendido como cociente entre o perímetro do pistão e o comprimento de onda do sinal aplicado. Nota-se que, até $ka = 3,83$, o valor da função de diretividade do pistão é positiva. O padrão de radiação do pistão vai ter somente um lóbulo sob estas condições. Se $ka = 3,83$, o padrão terá um valor nulo a 90° fora do eixo. Para valores mais elevados de ka , o campo nulo ocorre em ângulos sucessivamente menores. Além disso, os lóbulos secundários aparecem de fora do lóbulo principal, porém em menor magnitude do que o primário. Esses lóbulos irão alternar no sinal, o primeiro conjunto será negativo e o segundo positivo, assim sucessivamente (MITCHELL, 2008).

Figura 2.12 – Função de diretividade de um pistão



Fonte: Mitchell (2008)

A análise realizada do modelo utilizando pistão se difere um pouco do que ocorre na realidade, a resposta de um alto-falante não cumpre vários pressupostos básicos do modelo do pistão. A primeira diferença é que o alto-falante não tem o cone ou diafragma perfeitamente rígido. A segunda diferença entre o pistão teórico e o alto-falante real, é que o pistão é plano, e os diafragmas do alto-falantes são raramente planos. Na maioria das vezes estão na forma de um cone côncavo, mas também é utilizado cones convexos (MITCHELL, 2008).

No caso do transdutor de cone, o diafragma é excitado em seu centro. A excitação viaja para fora da bobina como uma perturbação acústica no material do cone. A velocidade da propagação dessa perturbação é sempre finita. Nas frequências mais baixas, este efeito é insignificante, mas as frequências mais altas, nem em todas elas o diafragma irá vibrar em fase. Em muitos casos, a membrana é escolhida de modo a minimizar o efeito da propagação da onda de velocidade finita no material do diafragma, controlando o eixo de resposta do dispositivo (MITCHELL, 2008).

De um modo geral, a diretividade de um cone ou domo dos transdutores reais é qualitativamente semelhante ao de um pistão rígido e plano.

2.3 Processamento Digital de Sinais (DSP)

O campo de processamento digital de sinais tem crescido nos últimos cinquenta anos, através de suas origens como um conjunto de técnicas para simular o comportamento de sistemas analógicos em computadores digitais. É uma das ferramentas mais estudadas e universalmente utilizada da tecnologia moderna. O uso de algoritmos de DSP tornou-se a regra e não a exceção, com aplicações em muitas áreas, tais como: música, comunicações, radar, sonar, processamento de imagem, robótica, sismologia, meteorologia, física aplicada, etc. (RICHARDSON, 2008).

Em várias áreas existem diferentes perspectivas e diferentes níveis de abstração para explorar o DSP. O estudo do processamento digital de sinais são divididos principalmente em três: teórico, físico e perspectivas de software embarcado (RICHARDSON, 2008).

A perspectiva teórica consiste na construção de fundamentos da teoria de DSP. Esses fundamentos incluem teoria de sistemas lineares, teoria de números complexos e matemática aplicada (RICHARDSON, 2008).

A perspectiva física concentra em dispositivos que são usados para implementar sistemas DSP. Esses dispositivos incluem processadores digitais de sinais programáveis para exe-

cutar operações matemáticas em alta velocidade. Também apresenta detalhes para realizar de forma precisa as conversões: sinal analógico para digital e sinal digital para analógico (RICHARDSON, 2008).

A perspectiva de software embarcado está preocupada com o software real que faz com que os processadores digitais de sinais executem as tarefas desejadas. O software é chamado de embarcado por ser executado internamente em um dispositivo DSP, somente acessível ao usuário através de uma interface e eficazmente escondido ou incorporado no produto escondendo os detalhes de implementação do usuário (RICHARDSON, 2008).

2.3.1 Amostragem

Segundo (LATHI, 2007), o teorema de amostragem é a ponte entre os mundos de tempo contínuo e tempo discreto. Para que um sinal de tempo contínuo possa ser processado por um dispositivo de tempo discreto, é necessário que seja feito um processo de amostragem. A taxa com que o sinal é amostrado é de extrema importância para que posteriormente este sinal original possa ser reconstruído com menos erros.

Um sinal real cujo o espectro é limitado em uma frequência máxima B Hz, pode ser reconstruído com perdas aceitáveis se suas amostras forem tomadas a uma taxa de $f_s > 2B$ amostras por segundo, ou seja, a menor frequência aceitável de amostragem é $f = 2B$ Hz (LATHI, 2007).

Segundo (RICHARDSON, 2008), em aplicações de áudio é comumente utilizado uma frequência de amostragem de 48 kHz, o que produz um período de amostragem de $1/48000 = 20,83\mu s$. Usando uma taxa de amostragem de 48 kHz é possível adicionar um atraso ao sinal de um número inteiro multiplicado por $20,83\mu s$.

2.3.2 Transformada de Hilbert

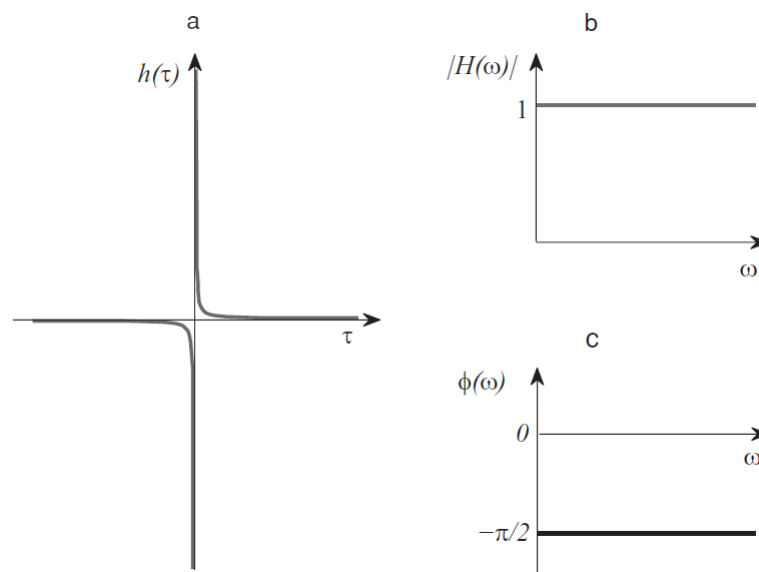
A transformada de Fourier permite o estudo de filtros capazes de separar sinais, baseados em suas frequências, porém, existem ocasiões onde a separação de sinais baseados em fase é mais apropriada, é neste ponto que a transformada de Hilbert se torna mais apropriada.

Em contraste com outras transformadas integrais, como Fourier e Laplace, a transformada de Hilbert (HT) não é uma transformada entre domínios. Em vez disso, atribui uma parte imaginária complementar a uma parte real, ou vice versa, deslocando cada componente do si-

nal em um quarto de um período, ou seja, 90 graus. Assim, A HT fornece um método para determinar a amplitude instantânea e a frequência instantânea de um sinal (FELDMAN, 2011).

Fisicamente, a transformada de Hilbert é equivalente a um tipo especial de filtro linear, onde todas as amplitudes e componentes espectrais permanecem inalteradas, mas suas fases são deslocadas por $\frac{\pi}{2}$. A função de resposta ao impulso da HT é mostrada na indicação *a* da Figura 2.13, e o módulo e as características de fase da função de transferência da HT são mostrados nas indicações *b* e *c* da Figura 2.13 (FELDMAN, 2011).

Figura 2.13 – A transformada de Hilbert ideal: a função de resposta ao impulso (a), o módulo(b), e a fase (c) da função de transferência da transformada de Hilbert



Fonte: Feldman (2011)

E termos de operações, a HT de uma constante é zero. A HT dupla, ou seja, HT de HT produz a função original com o sinal oposto, portanto, realiza uma mudança no sinal por $-\pi$ (180 graus). A HT utilizado quatro vezes na mesma função, retorna a função original. A HT da derivada de uma função é equivalente a derivada da HT da função, ou seja, a HT de uma função seno é uma função cosseno, o HT de uma função cosseno é uma função negativa de seno, mas para alguma forma de onda diferente, pode ter uma forma mais complicada (FELDMAN, 2011).

2.3.3 Processadores digital de sinais

Processadores de sinais digitais programáveis são microprocessadores com características particulares adequados para a realização de operações aritméticas, tais como multiplicação e adição de forma muito eficiente. Tradicionalmente, essas características tem melhorado o

desempenho do processador à custa de dificuldades de programabilidade. Um programa de aplicação DSP processa, tipicamente, um fluxo de entrada de dados para produzir alguns dados de saída. O processamento de dados é realizado pelo programa de aplicação, que inclui geralmente um ou mais algoritmos programados no DSP (RICHARDSON, 2008).

O Programa de aplicação DSP consiste em adquirir os dados de fluxo de entrada, usando algoritmos para processar dados e, em seguida, gera a saída dos dados processados para o fluxo de dados de saída.

Um dos fatores mais importantes desses processadores que utilizam entrada e saída é a idéia de tempo real. O tempo de produção de amostras e consumação delas consecutivamente conversões analógico/digital, e digital/analógico devem ser constantes a uma taxa fixa. Outro ponto importante em um sistema DSP de tempo real é o atraso imposto de entrada para a saída, principalmente deve ser considerado em aplicações de comunicação bidirecional (RICHARDSON, 2008).

2.3.4 SigmaStudio: uma plataforma de programação gráfica

Segundo (CHAVEZ; HUIN, 2006), ambientes de programação gráfica tem sido muito utilizado pelas indústrias nos últimos anos.

Uma plataforma de destaque é a SigmaStudio¹, que é um ambiente de desenvolvimento originalmente concebido para a família SigmaDSP de processadores de sinais específicos para áudio. Foi projetado para simplificar a programação e desenvolvimentos de aplicações de áudio para o mercado. A ideia principal foi facilitar a migração dos engenheiros de áudio analógico, para a utilização de metodologias DSP (CHAVEZ; GERMAN, 2011).

O SigmaStudio é uma ferramenta de programação gráfica desenvolvida pela Analog Devices². O software inclui uma vasta biblioteca de algoritmos para executar o processamento de áudio, tais como: filtragem, a mixagem e processamento dinâmico, bem como as funções básicas e controle de baixo nível DSP (POWERSOFT, 2011).

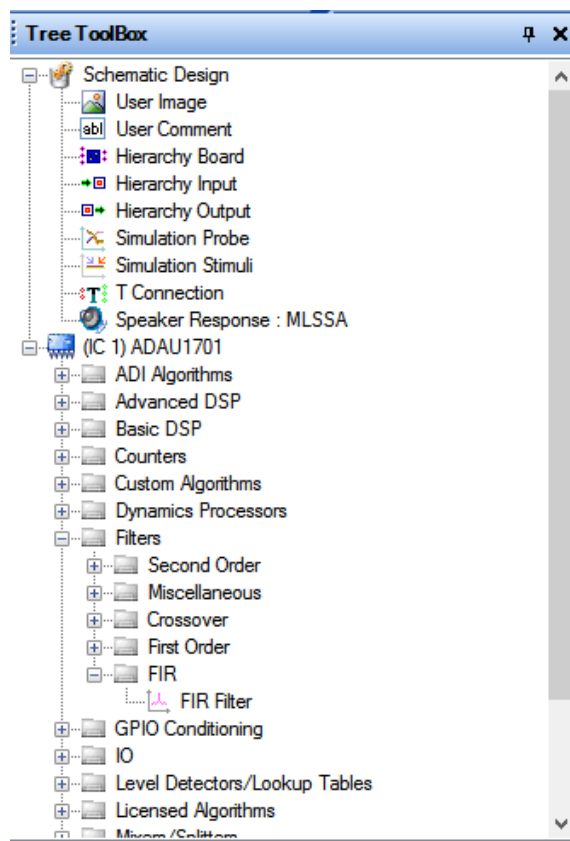
Todos esses algoritmos estão disponíveis como blocos de processamento que podem ser ligados entre si como em um esquema. Isso significa não é necessário utilizar um caminho de processamento de sinal pré-definido, pode ser definido cada um e cada detalhe desse caminho,

¹ <http://www.analog.com/en/design-center/processors-and-dsp/evaluation-and-development-software/ss_sigst_02.html#dsp-overview>

² <<http://www.analog.com/en/index.html>>

como por exemplo a quantidade de filtros (POWERSOFT, 2011). A Figura 2.14 mostra uma árvore de seleção desses blocos.

Figura 2.14 – Menu para seleção de blocos do SigmaStudio



Fonte: do Autor

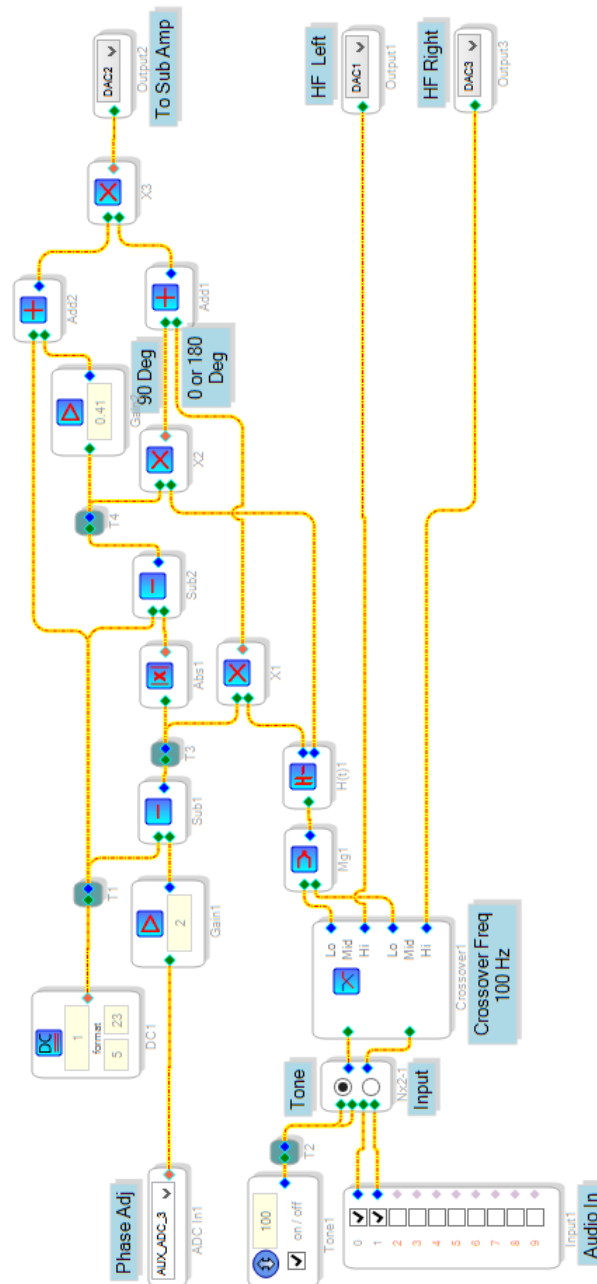
A plataforma permite que engenheiros, sem nenhuma experiência de escrever códigos, implementem projetos em DSP, e é suficiente para satisfazer a demanda de programadores mais experientes da área (POWERSOFT, 2011). A Figura 2.15 mostra um exemplo de programação em blocos.

A resposta a um impulso de um ambiente acústico ou transdutor é uma das caracterizações mais importantes no áudio. O atual estudo da arte é representado principalmente por dois tipos de medições citadas na Seção 2.4: Sequência de Máximo Tamanho (MLS) e Varredura senoidal de frequências.

2.4 Métodos de medição acústica

Dois dos principais métodos de medição acústica são as técnicas MLS (Sequência de Máximo Tamanho) e Varredura Senoidal. Os dois métodos utilizam o princípio de função de transferência, onde é comparado o sinal de entrada do sistema com o sinal de saída.

Figura 2.15 – Exemplo de programa em SigmaStudio baseado em blocos



Fonte: do Autor

2.4.1 Sequência de Máximo Tamanho

Inicialmente, é adotado que o sistema possui uma única entrada e uma única saída, denominado caixa preta, que um sinal de entrada $x(t)$ é introduzido, posteriormente obtendo o sinal de saída $y(t)$. Esses pressupostos são comuns para sistemas lineares e invariantes no tempo, dentro do sistema algum ruído pode ser gerado e inserido na parte determinística do sinal de saída. Normalmente, esse ruído é assumido como ruído branco gaussiano. Na prática o

signal de saída pode ser escrito como a soma do ruído gerado e a função determinística do sinal de entrada conforme a Equação 2.9 (FARINA, 2000).

$$y(t) = n(t) + F[x(t)] \quad (2.9)$$

Supondo que o sistema é linear e invariante no tempo, a função F assume a forma da convolução entre o sinal de entrada e a resposta ao impulso do sistema $h(t)$ conforme a Equação 2.10.

$$y(t) = n(t) + x(t) \otimes h(t) \quad (2.10)$$

Um sistema acústico não pode ser considerado linear e invariante no tempo, mas, muitas vezes, as não linearidades de um sistema estão no seu início e são substancialmente sem memória. Após essa distorção inicial, o sinal passa a ser linear subsequente. Esse caso é caracterizado por efeitos temporais evidentes, ou seja, memórias. Esse cenário é típico, pode ser descrito como um espaço reverberante excitado por um alto falante (FARINA, 2000). Em termos gerais, é possível afirmar que a distorção ocorre no transdutor eletromecânico, e à medida que o som é irradiado no ar, passa por um processo de propagação linear subsequente, incluindo múltiplas reflexões, ecos e reverberações.

A técnica de MLS baseia-se na excitação do espaço acústico por um sinal periódico pseudoaleatório, com propriedades estocásticas próximas as de um ruído branco puro (STAN; EMBRECHTS; ARCHAMBEAU, 2002).

Segundo (OPPENHEIM; SCHAFER, 2009), com a técnica MLS, a resposta ao impulso é obtida através da correlação cruzada circular entre a saída medida e a entrada determinada. Por causa do uso de operações circulares para a desconvolução da resposta ao impulso, a técnica de MLS entrega a resposta de impulso periódica $h(n)$, que está relacionada à resposta de impulso linear mostrada na Equação 2.11.

$$h'[n] = \sum_{l=-\infty}^{+\infty} h[n + lL] \quad (2.11)$$

A Equação 2.11 reflete um problema bem conhecido da técnica MLS, o *aliasing*. É um erro bastante conhecido se o comprimento de L por um período for menor que o comprimento da resposta ao impulso a ser medido. Portanto, a ordem da sequência MLS deve ser grande o suficiente para evitar este erro. Em termos de período e frequência de amostragem, o sistema

de medição permite sequências MLS até a ordem 19, o que corresponde a um período de 12 segundos e uma frequência de amostragem de 44,10 kHz (OPPENHEIM; SCHAFER, 2009).

A técnica MLS possui uma propriedade capaz de aleatorizar o espectro de fase de qualquer componente do sinal de saída que não esteja correlacionada com a sequência MLS de entrada. Essa propriedade faz com que cada sequência seja caracterizada por um espectro de fase que é fortemente errático, e com uma densidade de probabilidade uniforme no intervalo $(-pi, +pi)$ (RIFE, 1992).

Toda técnica possui suas vantagens e desvantagens, o principal problema do método MLS vem do aparecimento de artefatos de distorção, conhecidos como picos de distorção (VANDERKOOY, 1994). Segundo (STAN; EMBRECHTS; ARCHAMBEAU, 2002), esses artefatos são distribuídos de modo mais ou menos uniforme ao longo da desconvolução da resposta ao impulso. A origem dos picos de distorção reside nas não-linearidades inerentes do sistema de medição e especialmente no alto falante. Esses artefatos de distorção introduzem um barulho característico quando a resposta ao impulso medida é convoluída com algum sinal anecóico para realizar o processo de auralização.

Algumas técnicas podem ser usadas para atenuar esses picos de distorção como o uso de métodos de medição dedicados, como por exemplo, a técnica de sequência repetida inversa. É possível aumentar a amplitude do sinal de excitação, o que causa na prática um aumento das distorções em níveis elevados, porém diminui a relação de sinal/ruído. Quando se toma cuidado com a otimização dos parâmetros, é possível notar que os picos de distorção são significativamente reduzidos, mas não completamente removidos (STAN; EMBRECHTS; ARCHAMBEAU, 2002).

2.4.2 Varredura senoidal

A técnica MLS assume que o sistema é linear e invariante no tempo, e se caso apareça artefatos de distorção na resposta, quer dizer que essa condição não foi cumprida.

A técnica de varredura senoidal criada por Farina supera tais limitações (FARINA, 2000). Esse método possui amplo uso hoje em dia, muitas vezes é utilizada para medir resposta ao impulso com alta qualidade, essas medições podem ser usadas, por exemplo, para criar e aplicar filtros numéricos com espaços e reverberações realistas para a produção e gravação de música (FARINA, 2007). Essa técnica também é utilizada para caracterizar a resposta ao impulso de um sistema de alto-falantes e caixas acústicas.

Essa técnica utiliza uma varredura de frequências com crescimento exponencial, possibilitando a desconvolução simultânea da resposta linear ao impulso do sistema, e das distorções harmônicas do sistema (STAN; EMBRECHTS; ARCHAMBEAU, 2002). As distorções harmônicas aparecem antes da resposta linear ao impulso. Portanto, a resposta linear ao impulso medida é isenta de não-linearidades, e, ao mesmo tempo, a medição da distorção harmônica em várias ordens pode ser realizada.

Segundo (FARINA, 2007), utilizando como base uma varredura exponencial com desconvolução aperiódica, a medição fornece uma boa resposta a alguns dos problemas citados, tais como: a rejeição ao ruído é melhor do que um sinal MLS de mesmo comprimento, os efeitos não lineares são perfeitamente separados da resposta linear e o uso de uma varredura única e longa (sem média síncrona) evita qualquer problema no caso de o sistema tenha alguma variação de tempo.

A definição matemática do sinal de varredura é representado pela Equação 2.12, onde uma varredura começa na frequência angular ω_1 , e termina na frequência angular ω_2 , levando T segundos (FARINA, 2007).

$$x(t) = \text{sen}\left[\frac{\omega_1 \times T}{\ln\left(\frac{\omega_2}{\omega_1}\right)} \times \left(e^{\frac{t}{T}\left(\frac{\omega_2}{\omega_1}\right)} - 1\right)\right] \quad (2.12)$$

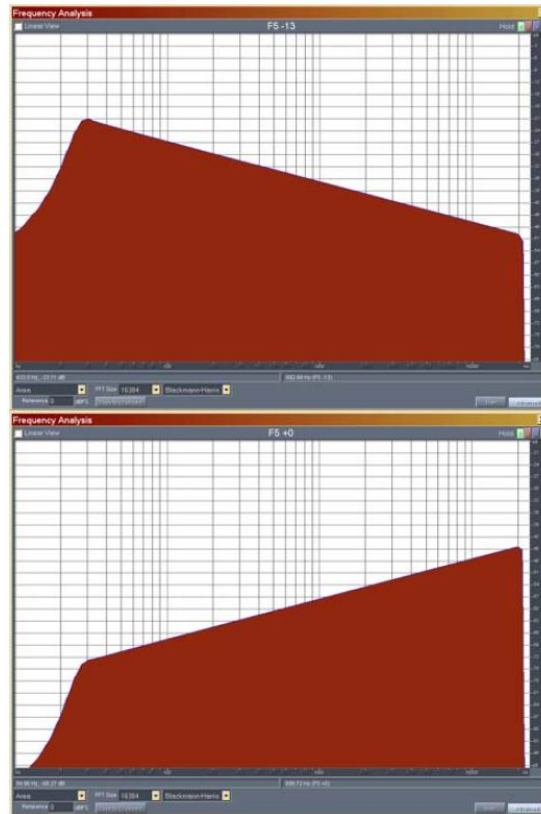
É possível separar a resposta da sala através de um método de análise do sinal gravado pelo microfone. Quando o sinal senoidal que tem amplitude constante é seguido por alguns instantes de silêncio, o microfone passa a gravar os efeitos de reflexões da sala (FARINA, 2007).

Com o sinal de saída gravado, é realizado um processo para extrair a resposta linear ao impulso. O que é feito é uma convolução do sinal de saída com um filtro inverso ao impulso utilizado. Dois cuidados devem ser tomados, o primeiro é implementar a convolução de forma aperiódica, evitando com que produtos de distorção contaminem a resposta linear. O segundo é realizar um espelho de reversão do impulso utilizado, para criar o filtro reverso (FARINA, 2007). Uma técnica de convolução rápida foi desenvolvida por Angelo Farina, que permite a computação da convolução proposta em um tempo significativamente mais curto que a duração do sinal (TORGER; FARINA, 2001).

O sinal de teste não possui um espectro plano, que é a característica de um ruído branco. O sinal de teste, analisado pelo espectro de Fourier, possui uma queda de $-3dB$ por cada oitava.

Isto se dá devido ao fato de que a varredura é realizada lentamente em baixas frequências e muito mais rápida em altas frequências. O filtro inverso deve compensar isso por meio de uma aplicação ao filtro inverso de uma modulação de amplitude, fazendo com que a amplitude do sinal cresça $+3dB$ por oitava (FARINA, 2007). A Figura 2.16 mostra o impulso enviado e o filtro inverso.

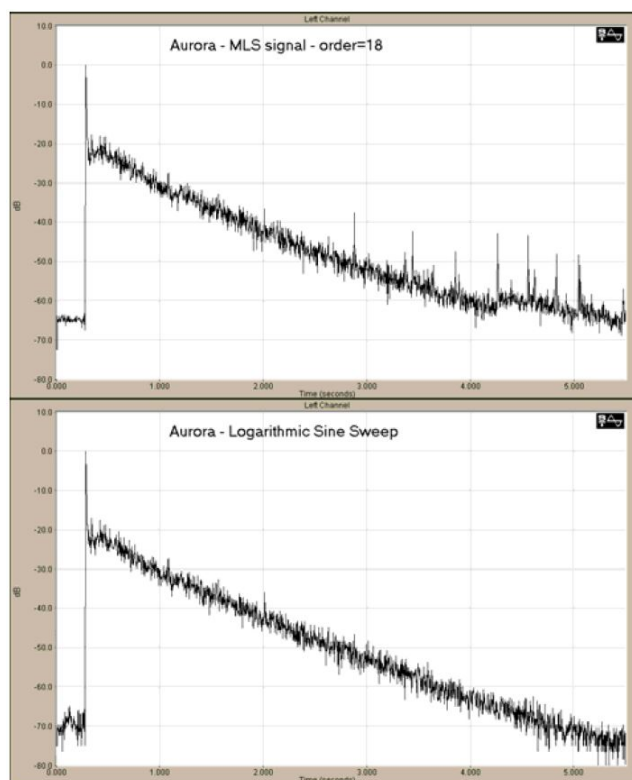
Figura 2.16 – Espectro de Fourier do sinal de teste (acima) e do filtro inverso (abaixo)



Fonte: Farina (2007)

Para mostrar a vantagem da técnica de varredura senoidal em relação a MLS, um cenário de teste com as mesmas condições foi analisado, utilizando as duas abordagens, presente na Figura 2.17. É fácil ver como o método de varredura exponencial produz uma melhor relação sinal/ruído, e também o desaparecimento de picos de distorção na parte tardia da medição MLS. Segundo (FARINA, 2007), esses picos são causados pela limitação da taxa de velocidade de amplificadores de potência e alto falante empregados para medição, que produz distorções harmônicas severas.

Figura 2.17 – Comparação entre as medidas MLS e varredura senoidal



Fonte: Farina (2007)

2.5 Estimador intrusivo de qualidade de voz

O método intrusivo utiliza uma amostra da fala como referência e compara com uma amostra no ponto de destino, o resultado é uma pontuação da avaliação de qualidade. O setor de Telecomunicação da União Internacional de Telecomunicações (ITU-T) propõe recomendações e métodos para a avaliação objetiva e subjetiva da qualidade (RODRÍGUEZ, 2009).

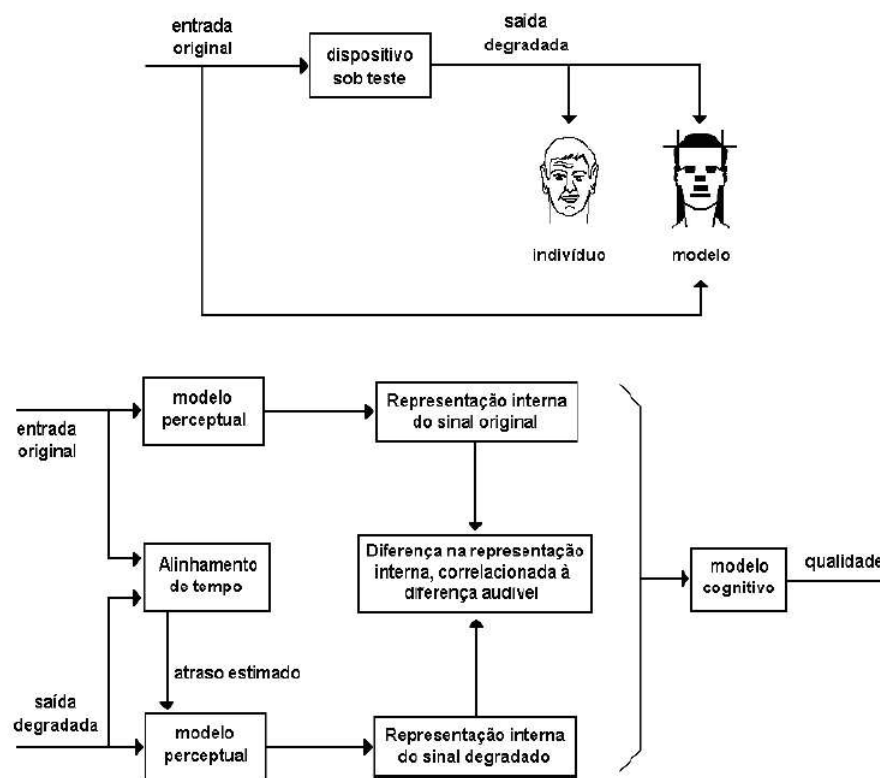
A recomendação ITU-T P.862 (ITU-T, 2001) foi preparada pelo grupo de estudo 12 da ITU-T (2001-2004) e aprovada sob o regime da resolução WTSA 1 em 23 de fevereiro de 2001. É um método objetivo e conhecido como Avaliação perceptual da qualidade de voz (PESQ - *Perceptual Evaluation of Speech Quality*).

O método PESQ considera diversos parâmetros como filtragem e retardo variável, distorções devidas a erros de canal e codificadores de baixa qualidade. Ele mede os efeitos de ruído e a distorção de voz unidirecional sobre a qualidade de voz. Os efeitos de perda de audibilidade, retardo, eco e outros fatores de degradação relacionados a uma comunicação bidirecional não são consideradas neste método (RODRÍGUEZ, 2009).

A PESQ utiliza um algoritmo para comparação de um sinal original com a saída do mesmo sinal, passando por um sistema de comunicação. O resultado é uma predição da percepção da qualidade de voz avaliada por um indivíduo em teste subjetivo de audição, chamado de MOS (*Mean Opinion Score*) que é um método subjetivo de teste de qualidade (ITU-T, 1996). O algoritmo realiza um tratamento nos sinais de origem e de saída para uma correção de tempo entre eles, com a finalidade de realizar a comparação.

A ideia do processo é analisar o sinal como se fosse um sistema auditivo humano, levando em consideração frequência perceptual e a intensidade. Para isso, é realizado alinhamento temporal, alinhamento do nível de intensidade, mapeamento de frequência em função do tempo, transformação da escala de frequências e compressão da escala de intensidade.

Figura 2.18 – Filosofia básica utilizada na PESQ



Fonte: Rodríguez (2009)

Os votos subjetivos são influenciados por muitos fatores, tais como: preferências individuais dos sujeitos e o contexto do experimento. Assim, um processo de regressão é necessário antes que seja feita uma comparação direta, a regressão deve ser monótona, ou seja, deve preservar a informação. O processo é normalmente usado para mapear a pontuação objetiva PESQ sobre a avaliação subjetiva. Uma boa medida da qualidade objetiva deve ter uma alta correlação

com muitas experiências subjetivas diferentes se esta regressão é realizada separadamente para cada um.

A proximidade do ajuste entre a PESQ e as pontuações subjetivas pode ser medida através do cálculo do coeficiente de correlação. Normalmente, isso é feito sob a condição de pontuação média, após mapear o objetivo das pontuações subjetivas. O coeficiente de correlação é calculado com a fórmula de Pearson, equação 2.13.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \quad (2.13)$$

Na formula x_i , é condição MOS para a condição i , \bar{x} é a média ao longo das condições dos valores MOS, $x_i \cdot y_i$ é o mapeamento da condição média da pontuação PESQ para a condição i , e \bar{y} é a média da condição MOS para previsão de y_i .

Segundo (ABREU; RODRÍGUEZ; LACERDA, 2016), qualidade de voz é degradada quando se altera linearmente a fase da reprodução sonora. Existe uma correlação entre a fase da reprodução e o índice de qualidade de voz. Quanto mais próximo de 180 graus, menor o índice de qualidade utilizando a recomendação ITU-T P.862

3 MATERIAIS E MÉTODOS

Este capítulo tem como objetivo apresentar os materiais e métodos utilizados e detalhar o desenvolvimento do protótipo do processador digital de sinais de áudio em tempo real.

3.1 Descrição do Hardware

O protótipo foi desenvolvido em parceria com a empresa Expert Electronics¹. É utilizado como base um produto de linha comercializado pela fábrica, este produto realiza operações com o áudio diferentes das de interesse deste trabalho. Para o desenvolvimento do protótipo, o produto de linha da fábrica foi reprogramado com o auxílio do engenheiro responsável para realizar operações de atraso de sinal por amostragem, e variação linear de fase, possibilitando o ajuste selecionável de um em um grau entre 0 a 180 graus. Isso torna possível realizar o controle de interferências destrutivas.

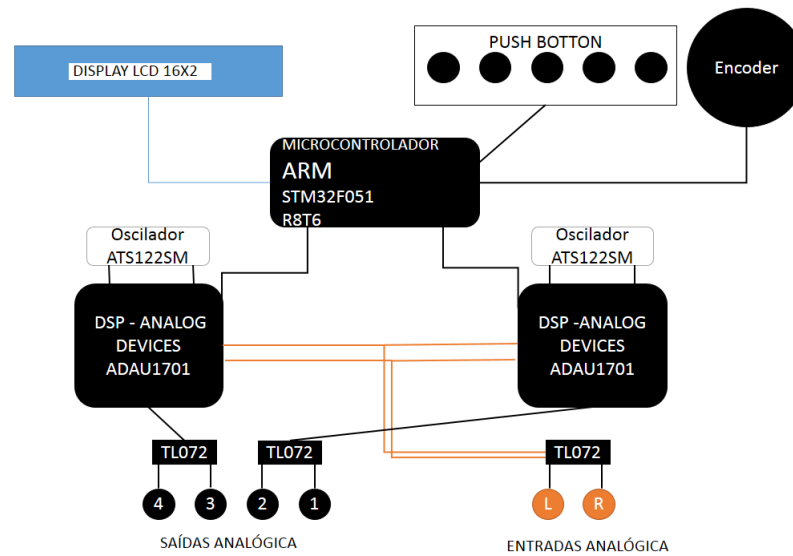
O dispositivo proposto é um processador digital de sinais capaz de receber um sinal analógico de som. O sinal é convertido de analógico para digital com 24 bits de resolução e uma taxa de amostragem de 48 kHz. Após a conversão, o processador realiza processamento do sinal digital, e fornece 4 saídas analógicas independentes. O Anexo A apresenta o esquemático completo do dispositivo.

Os principais componentes utilizados são: um microcontrolador, dois DSP (processador digital de sinais), conectores RCA para entrada e saída de sinais analógicos, um encoder, 4 push button e um display LCD 16x2 para comunicação com o usuário. A Figura 3.1 mostra o diagrama de blocos do dispositivo.

O microcontrolador escolhido é do fabricante STMicroelectronics, modelo STM32F105R8. Possui um processador ARM Cortex de 32 bits RISC, com velocidade máxima de 72 MHz, 256 kbytes de memória flash, 64 kbytes memória SRAM (STMicroelectronics, 2008). O microcontrolador tem a função de receber os dados inseridos pelo usuário por meio das entradas como encoder e *push buttons*. Transmite os dados ao usuário através do display LCD. Envia os dados inseridos pelo usuário para 2 DSPs utilizando comunicação SPI, onde é feito o processamento dos sinais.

¹ <<http://www.expertelectronics.com.br/>>

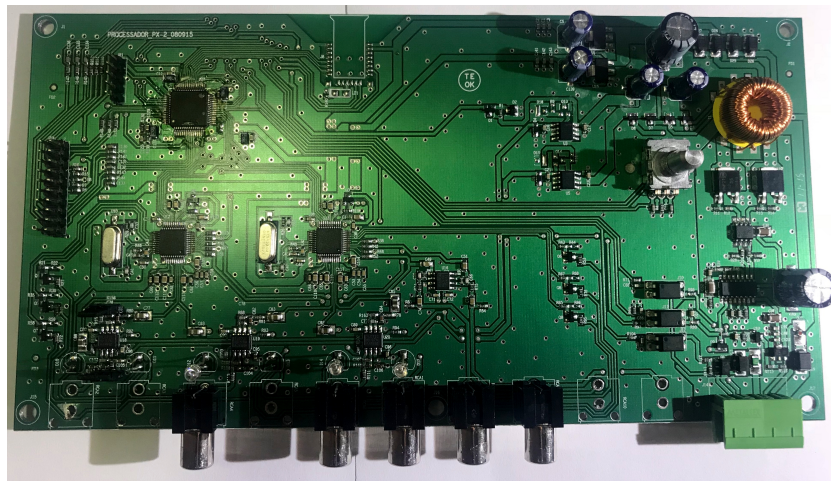
Figura 3.1 – Diagrama de blocos Processador digital de Sinais de Áudio



Fonte: do Autor

A Figura 3.2 apresenta a foto do circuito elétrico do protótipo. A placa com o cristal LCD é mostrado na Figura 3.3. e, por fim, a foto externa do protótipo é apresentado na Figura 3.4.

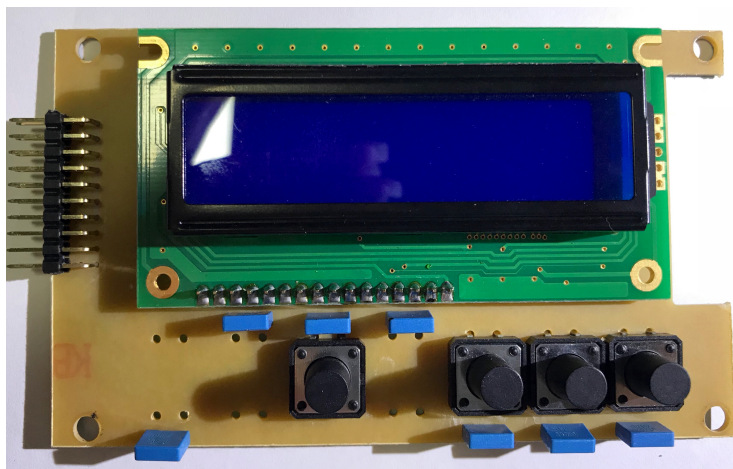
Figura 3.2 – Foto da placada do protótipo



Fonte: do Autor

Para desenvolver o protótipo, para que o mesmo possa de ser utilizado por qualquer pessoa, foi necessário desenvolver um modelo baseado em aproximações que utilize variáveis simples e calcule os valores de atraso por amostras e a alteração de fase.

Figura 3.3 – Foto da placada do LCD e *push botton* do protótipo



Fonte: do Autor

Figura 3.4 – Foto externa do processador



Fonte: do Autor

3.2 Desenvolvimento do modelo para inserir atraso e variação de fase

Desenvolvido a partir de aproximações, o modelo de controle dos sinais de áudio utiliza uma combinação de atraso por amostragem, com rotação linear de fase igual para todas as frequências. Através de funções trigonométricas, aproxima-se a variação de fase e quantidade de amostras atrasadas. São necessários que alguns parâmetros sejam inseridos pelo usuário, e através deles são realizados os cálculos no microcontrolador e enviado dados para o DSP realizar o processamento necessário para direcionar o áudio em um ponto, controlando interferências destrutivas. Os parâmetros são:

- Temperatura ambiente: utilizada para calcular a velocidade do som no ar.

- Distância entre as fontes: corresponde à distância entre os centros acústicos das fontes sonoras.
- Distância de direcionamento: corresponde à distância em linha, perpendicular à reta de distância entre as fontes.
- Porcentagem de direcionamento: variando de 0% a 100%, onde 0% é no centro entre as fontes e 100% é no eixo de uma das fontes, ou seja, se a distância entre as fontes é de 1 metro, 0% representa o centro, 40%, como exemplo, seria a 0,20 metros do centro para o lado de direcionamento.
- Escolha de Lado: o usuário deve escolher o lado desejado de direcionamento.

A inserção da temperatura é necessária para o cálculo da velocidade do som no ar em m/s , utilizado na Equação 3.1.

$$v = \sqrt{\frac{T(^{\circ}C) + 273,15}{273,15}} \times 331,15 \quad (3.1)$$

Após a inserção de todas as informações, é calculada a diferença entre os lados (dl), ou seja, diferença de distância paralelo a reta de distância entre as fontes, em que o som vai ser direcionado, calculado pela Equação 3.2. A Figura 3.5 ajuda a entender os cálculos.

$$dl = 2 \times \left(\frac{p}{200}\right) \times d1 \quad (3.2)$$

Onde $d1$ representa a distância entre os centros acústicos das fontes e p representa a porcentagem de direcionamento.

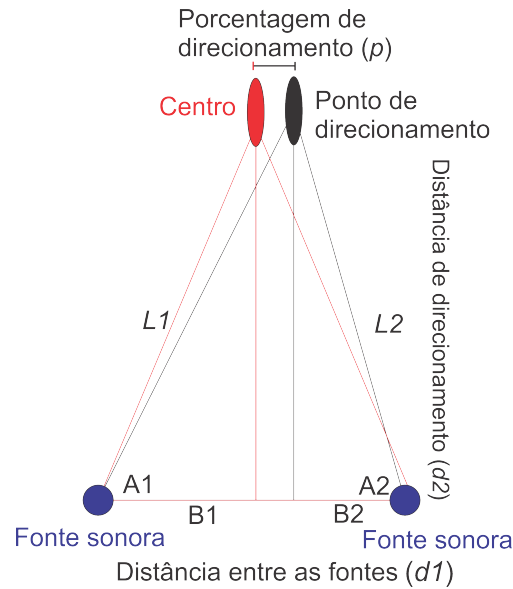
O próximo passo é calcular as duas bases de um triângulo formado pelo ponto de direcionamento e as duas fontes. As Equações 3.3 e 3.4 são utilizadas para calcular as duas bases.

$$B1 = \frac{d1}{2} + \frac{dl}{2} \quad (3.3)$$

$$B2 = \frac{d1}{2} - \frac{dl}{2} \quad (3.4)$$

Com as bases calculadas, é possível calcular as distâncias $L1$ e $L2$, que é a distância entre o ponto de direcionamento, as fontes sonoras. Para este cálculo é necessário $d2$, que é a

Figura 3.5 – Representação do triângulo formado pelas fontes sonoras e o ponto de direcionamento



distância de direcionamento. Os cálculos estão representados pelas Equações 3.5 e 3.6 utilizando uma adaptação do teorema de Pitágoras.

$$L_1 = \sqrt{d_1^2 - (2 \times d_1 \times B_2) + B_2^2 + d_2^2} \quad (3.5)$$

$$L_2 = \sqrt{d_1^2 - (2 \times d_1 \times B_1) + B_1^2 + d_2^2} \quad (3.6)$$

Conhecidos os valores de L_1 e L_2 , é possível realizar uma aproximação da quantidade de amostras a serem atrasadas, representado por at . É aplicado o atraso na fonte mais próxima do ponto de direcionamento. Para calcular o atraso, é utilizado a diferença entre L_1 e L_2 , a velocidade do som no ar e o período de amostragem do processador. Cálculo mostrado na Equação 3.7.

$$at = \frac{(L_1 - L_2)}{v \times 0,00002083} \quad (3.7)$$

Para que seja possível calcular o valor da alteração linear de fase a ser aplicada, é realizado uma aproximação da diferença angular (an) entre as duas fontes e o ponto de direcionamento. A primeira análise realizada é a razão entre a distância das fontes e a distância de direcionamento. Essa razão r deve estar em maior que 0 e menor ou igual a 1. É dividida em quatro partes as possibilidades de razão:

- Razão igual a 1: é utilizada a Equação 3.8.

- Razão maior que 0,75: é utilizada a Equação 3.9.
- Razão igual a 0,75: é utilizada a Equação 3.10.
- Razão menor que 0,75: é utilizada a Equação 3.11.

$$an = r \times 0,45 \times p \quad (3.8)$$

$$an = r \times 0,45 \times (1,111^{(1-(1-r))}) \times p \quad (3.9)$$

$$an = r \times 0,45 \times 1,111 \times p \quad (3.10)$$

$$an = r \times 0,45 \times (1,111^{(1+(1-r))}) \times p \quad (3.11)$$

Através dos resultados encontrados em an e at , é possível aplicar o controle do sinal de áudio, inserindo os valores no protótipo.

Os cálculos, citados nesta seção, também são utilizados na presença de uma terceira fonte (centro). Com três fontes, os cálculos são feitos considerando o triângulo externo formado pelas fontes da direita e da esquerda, e posteriormente é realizado um novo cálculo considerando o triângulo formado pela fonte do centro e a fonte do lado desejado de controle. Os dados são inseridos no DSP.

3.3 Desenvolvimento do Software

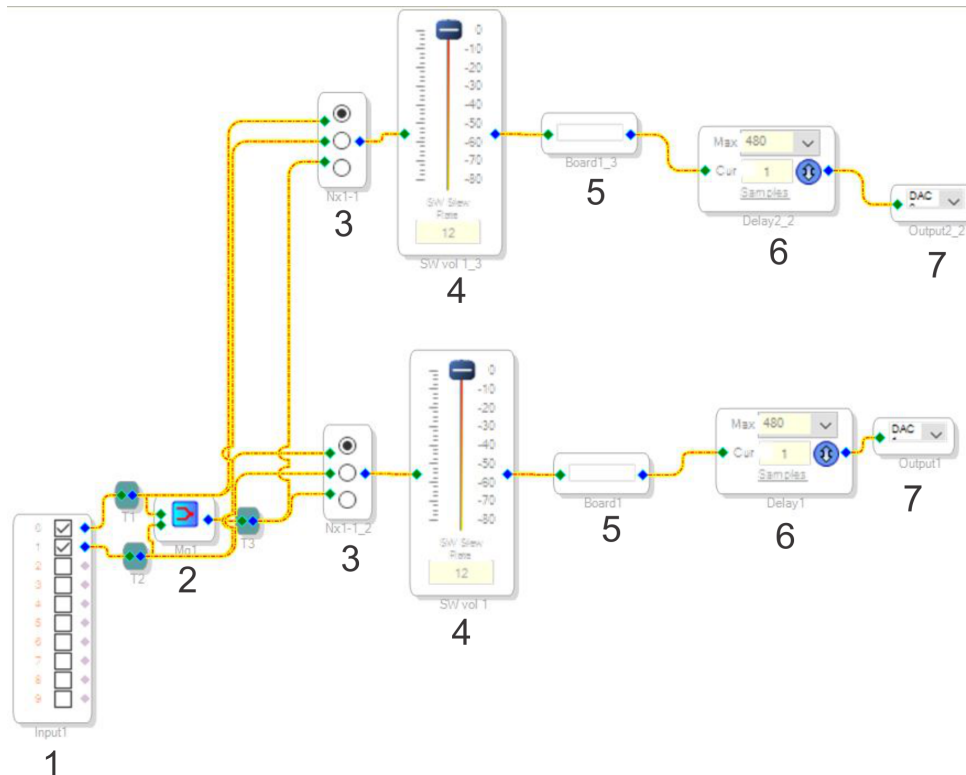
O protótipo foi desenvolvido a partir da reprogramação do produto da Expert Electronics, utilizado a mesma placa de componentes eletrônicos. A partir da nova programação, ficaram disponíveis 3 canais de saída de áudio, são eles, direita, centro e esquerda. Além de realizar o controle de interferências destrutivas, foram implementadas mais duas funções. São elas: roteamento de canais e ajuste de volume independente para cada saída.

O ajuste de canais pode ser utilizado para escolher a origem do áudio que irá para cada saída. O protótipo possui duas entradas de áudio, L e R, e no estágio de cada um dos canais de saída, direita, centro e esquerda, é possível escolher a origem do sinal. O sinal pode ser proveniente de L, R ou L+R, onde L+R é a mixagem das entradas.

O ajuste de volume permite ao usuário controlar o nível de cada uma das saídas, com um ajuste de volume independente variando de -36db a +6db.

O DSP é programado via a interface do Sigma Studio. Seu programa é formado por uma sequência de funções representada por blocos. A Figura 3.6 mostra os 7 blocos responsáveis pelo processamento do áudio.

Figura 3.6 – Blocos da programação do protótipo no Sigma Studio



A sequência de programação da Figura 3.6 é implementada em cada um dos dois DSP presentes no protótipo, onde um dos DSP é responsável pelas saídas centro e direita, e o outro DSP responsável pela esquerda.

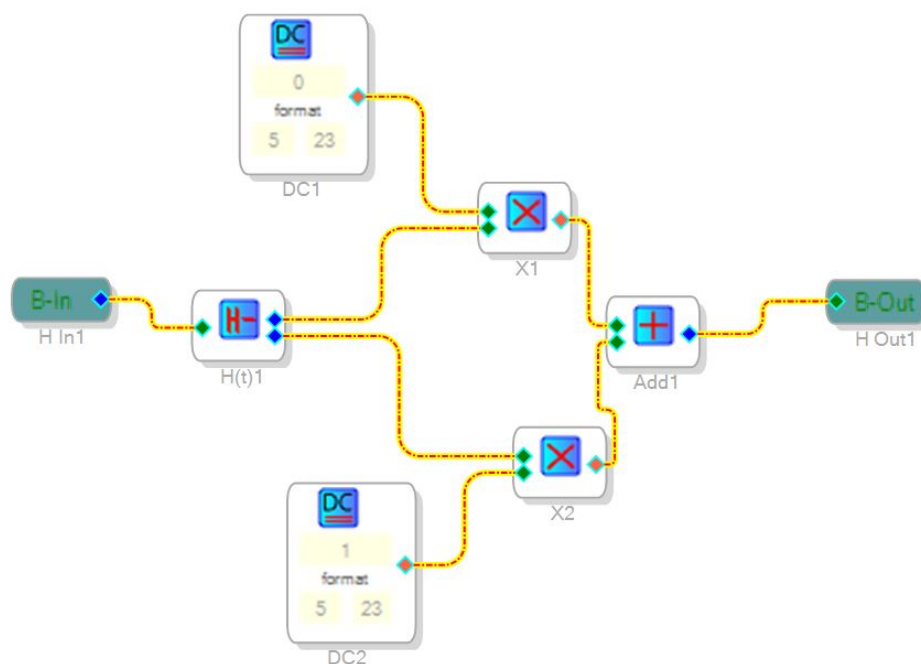
Cada um dos blocos enumerados é responsável por uma função de processamento do sinal de áudio. O sinal é processado por sequência da seguinte forma:

1. Entrada do sinal de áudio, são dois canais de entrada;
2. Responsável pela mixagem do sinal de entrada;
3. Opção de roteamento de sinal de entrada, este bloco permite que o usuário escolha qual a origem do sinal de entrada a ser utilizado na saída;
4. Este bloco é o responsável pela função de ajuste de volume de cada saída;

5. É um sub bloco responsável pelo ajuste da rotação linear de fase, a Figura 3.7 mostra este sub bloco em detalhes;
6. Bloco responsável pelo ajuste de atraso de sinal por amostragem
7. Saída do sinal digital de áudio, é a etapa de conversão do sinal digital em analógico.

A Figura 3.7 é o sub bloco de ajuste de rotação linear de fase. Sua ideia principal é a utilização da transformada de Hilbert. O sinal passa pela transformada e é defasado em todas as frequências em 90 graus, com isso é possível efetuar a comparação dos sinais e realizar a defasagem do sinal de grau em grau em todas as frequências.

Figura 3.7 – Sub-bloco da programação da função de fase do protótipo no Sigma Studio



Com o projeto de processamento do DSP pronto, foi gerado um código fonte na linguagem C que será utilizado para a programação através da plataforma Atollic True Studio² do protótipo. O protótipo utiliza o Microcontrolador para efetuar as operações de receber os dados do usuário, efetuar os cálculos da função de direcionamento de áudio, e enviar as operações aos DSPs. Um exemplo é o código da função de rotação linear de fase presente na Figura 3.8, utilizado na programação do microcontrolador. Os códigos de programação utilizados no segundo protótipo estão disponíveis no Apêndice B.

² <<https://atollic.com/truestudio/>>

Figura 3.8 – Código em C da a função de rotação linear de fase

```

float FASE_RAD=0.0;
void FASE_CANAL(void) {
    if(ESCOLHA_CANAL==0) { Sub_Address [0]=FASE_CH1; }
    if(ESCOLHA_CANAL==1) { Sub_Address [0]=FASE_CH2; }
    if(ESCOLHA_CANAL==2) { Sub_Address [0]=FASE_CH3; }

    if(ESCOLHA_CANAL==3) { Sub_Address [0]=FASE_CH1; }
    if(ESCOLHA_CANAL==4) { Sub_Address [0]=FASE_CH2; }
        FASE_RAD=cos(((PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ])*0.01745329));
        To523(FASE_RAD, &coebyte5 [0]);
        TXByteCtr=4; MSData=&coebyte5 [0]; SPI_TX ();1
        FASE_RAD=sin(((PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ])*0.01745329));
        To523(FASE_RAD, &coebyte5 [0]);
        Sub_Address [0]=Sub_Address [0]+1;
        TXByteCtr=4; MSData=&coebyte5 [0]; SPI_TX ();
}

```

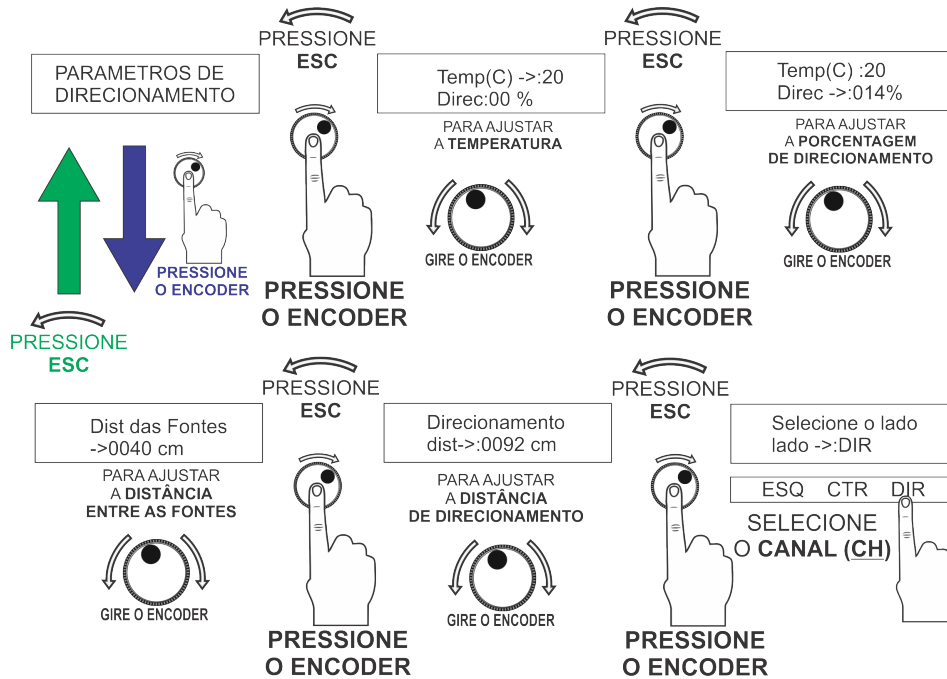
Fonte: do Autor

O ponto principal protótipo é a facilidade de uso da interface de configuração do usuário. Essa facilidade torna o uso do protótipo acessível a qualquer pessoa.

3.4 Interface do Protótipo

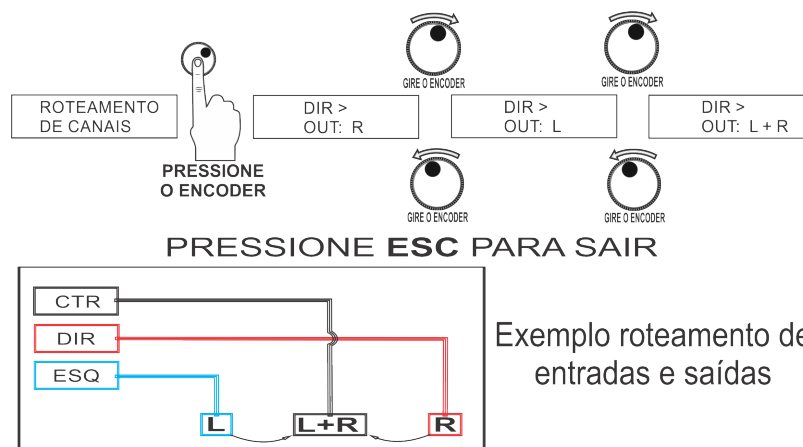
O protótipo tem como principal função realizar o direcionamento de ondas sonoras. A Figura 3.9 apresenta o modo como a interface entre o dispositivo e o usuário é configurada. Para selecionar os valores dos campos a serem configurados, é necessário girar o encoder sentido horário para aumentar o valor e sentido anti-horário para diminuir. O primeiro parâmetro a ser configurado é a temperatura ambiente, que ao ser ajustada deve-se pressionar o encoder para que passe para a seleção do próximo parâmetro. O segundo parâmetro a ser ajustado é a porcentagem de direcionamento; o terceiro é a distância entre as fontes; o quarto é a distância de direcionamento, e por último, é o lado de direcionamento. A qualquer momento, pode ser pressionado a tecla ESC que tem a função de retornar a seleção anterior.

Figura 3.9 – Parâmetros de direcionamento



Além da função principal, outras funções são passíveis de ajustes pelo usuário. O ajuste do volume de saída mostrado na Figura 3.10, que é realizado pela rotação do encoder, e após chegar ao valor desejado, o mesmo é pressionado para confirmar o ajuste.

Figura 3.10 – Ajuste de volume da saída

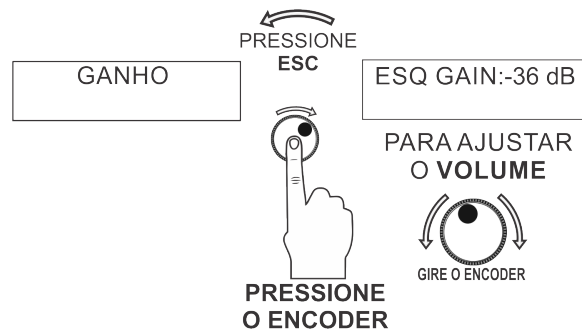


Exemplo roteamento de entradas e saídas

Outra função disponível é o ajuste de roteamento de canais explicado na Seção 3.3, e, assim, como a função de ajuste de ganho, é utilizada a rotação do encoder para seleção da origem do sinal e é necessário pressina-lo para confirmar a seleção. A função de ajuste de Roteamento de canais é mostrado pela Figura 3.11.

Para compor o cenário de teste, foi necessária a fabricação de caixas sonoras. A próxima seção descreve este processo.

Figura 3.11 – Roteamento de canais

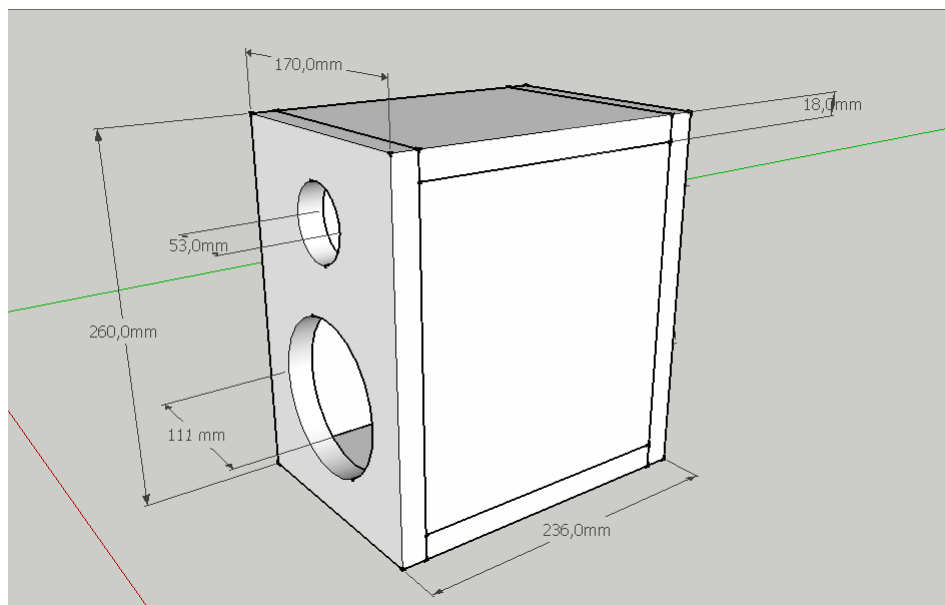


3.5 Caixas de som 2 vias

O sistema de som duas vias também pode ser chamado de caixa de som, é responsável em transformar energia elétrica em energia acústica. É composto por um alto falante de 4,5 polegadas e um tweeter (reprodutor de altas frequências) de 1/2 polegadas submetido a um filtro passa altas com ponto de crossover em 3,5 kHz. Para o desenvolvimento deste trabalho, foram utilizadas três caixas acústicas, fabricadas para a realização dos testes de direcionamento de áudio deste trabalho.

A caixa acústica foi produzida utilizando madeira do tipo MDF, seguindo as medidas presentes na Figura 3.12. Tais medidas foram obtidas através da simulação da caixa utilizando o software WinISD³.

Figura 3.12 – Projeto da caixa acústica do sistema de som 2 vias



³ <<http://www.linearteam.org/>>

Os três transdutores de 4,5" são iguais, porém, é comum no processo de fabricação alguma diferença entre as peças. A fim de tentar manter a maior igualdade possível entre as três caixas fabricadas, foi analisado a curva de impedância de cada um dos três transdutores utilizados. Com a análise da curva de impedância é possível extrair os parâmetros denominados de cada um deles, presente na Tabela 3.1.

Para realizar a medição da curva de impedância e parâmetros dos transdutores e das caixas, foi utilizado o kit de ferramentas de teste de áudio DATS V2⁴, que é um dispositivo versátil para medições e análises. Os testes foram realizados após 6 horas amaciando o alto falante com ruído rosa a 7,5 W. Para extração dos parâmetros foi utilizado um sinal de varredura logarítmica senoidal.

Tabela 3.1 – Parâmetros dos transdutor de 4,5"

Transdutor	1	2	3
Ressonância em ar livre	80,75 Hz	78,731 Hz	80,75 Hz
Q total (Qts)	0,58553	0,5708	0,5854
Q elétrico (Qes)	0,66966	0,65076	0,66718
Q mecânico(Qms)	4,6604	4,6458	4,7755
Volume Equivalente (Vas)	5,2084 litros	5,3226 litros	5,44 litros
Complância (Cms)	0,601 mm/N	0,614 mm/N	0,628 mm/N
Resistência DC (Re)	5,3545 Ohms	5,4455 Ohms	5,4547 Ohms
Impedância Máxima (Zmax)	40,618 Ohms	44,321 Ohms	44,498 Ohms
Impedância Mínima (Zmin)	5,3545 Ohms	5,4455 Ohms	5,4547 Ohms
Área do Pistão (Sd)	0,007854 m ²	0,007854 m ²	0,007854 m ²
Sensibilidade 1W/1m (dB SPL)	88,016	87,905	88,221
Sensibilidade 2,83 Vrms (dB SPL)	89,76	89,576	89,884
Eficiência (n0)	0,39%	0,38%	0,41%
Indutância da bobina 1 kHz (Le)	0,91641 mH	0,93676 mH	0,91376 mH
Indutância da bobina 10 kHz (Le)	0,46451 mH	0,47572 mH	0,46726 mH
Produto BL (BL)	5,12 N/Amp	5,2473 N/Amp	5,0659 N/Amp
Massa móvel efetiva (Mms)	6,4618 gramas	6,6516 gramas	6,1867 gramas

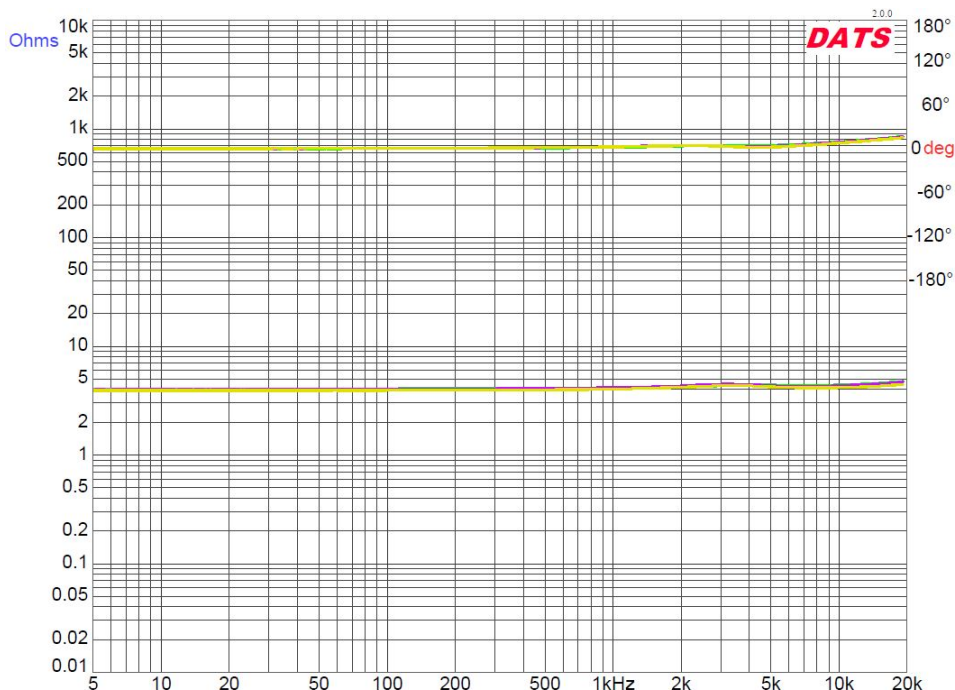
Com a análise dos parâmetros, é possível observar a proximidade de semelhança entre os três transdutores. Essa semelhança é fundamental para que os testes sejam realizados utilizando fontes sonoras com respostas sonoras mais próximas possíveis.

Os tweeters também possuem características próximas, como mostra a Figura 3.13, onde está disponível a curva de impedância e curva de fase dos três utilizados. Os três possuem ressonâncias em 4297,20 Hz, 3174,10 Hz e 3449,30 Hz, e Q total em 0,210, 0,150, 0,163 respectivamente, tal diferença na frequência de ressonância é devido ao fato de ambos serem

⁴ <<http://www.daytonaudio.com/index.php/dats-dayton-audio-test-system.html>>

bem controlados e não apresentarem um pico de impedância característicos para a determinação da mesma.

Figura 3.13 – Curvas de impedância e fase dos tweeters



Com a caixa montada, foi analisada a curva de impedância de cada uma delas e também a resposta de frequência medida a 1 metro com 0° (no eixo) e a um metro com 45° . A Figura 3.14 apresenta a curvas de impedância de cada uma das caixas acústicas, através delas é possível a similaridade entre as três.

As caixas possuem frequência de ressonância do sistema caixa fechada em 98,25 Hz, 96,90 Hz e 97,57 Hz, e os fatores de qualidade total em caixa fechada em 0,729, 0,712 e 0,714, respectivamente.

A resposta em frequência das caixas foi medida no mesmo ambiente utilizado para a realização dos testes e é apresentada pela Figura 3.15, onde é possível observar o fator de diretividade mostrado na Seção 2.2.3.2, onde a medida que a frequência aumenta, existe uma queda de SPL para 45 graus comparativamente a 0 graus.

3.6 Microfone Behringer ECM8000

O ECM8000 é altamente utilizado para análises e medições acústicas. A grande utilização é devida a característica de possuir uma resposta consideravelmente plana e com alta resolução, além de seu baixo custo, gerando um baixo custo benefício. O microfone utiliza uma

Figura 3.14 – Curvas de impedância e fase dos sistemas de caixas acústicas 2 vias

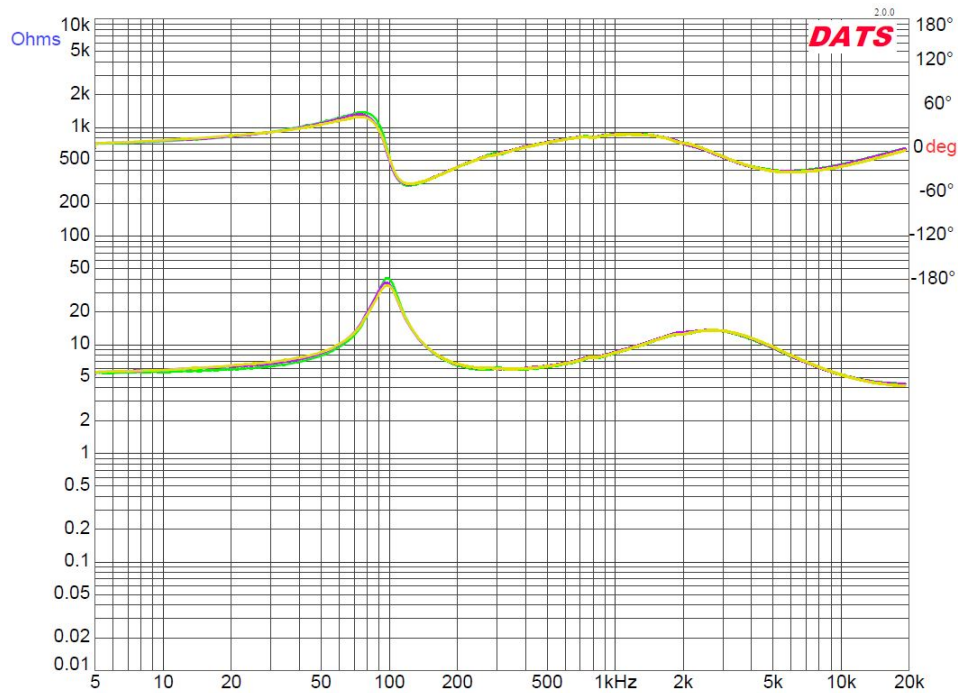
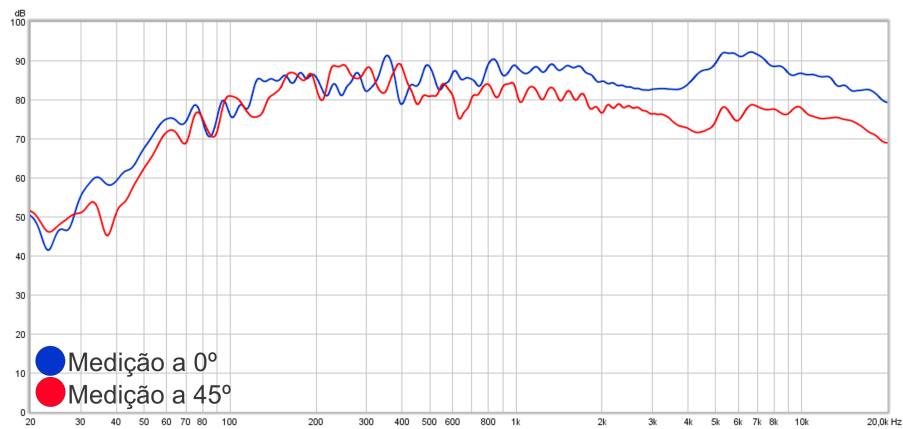


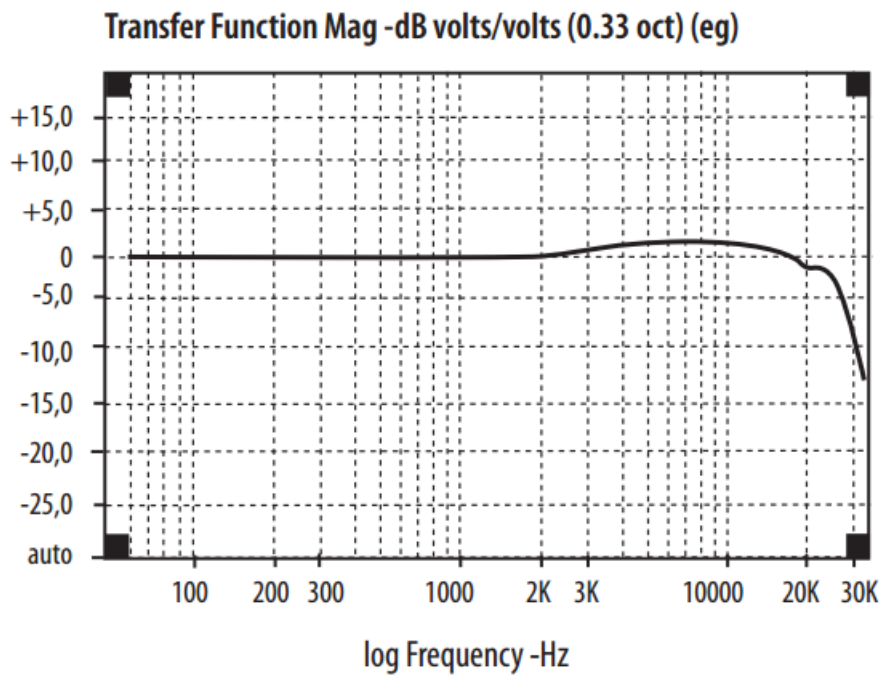
Figura 3.15 – Curvas de resposta em frequência a 0° e 45° do sistema de caixa acústica 2 vias



alimentação fantasma (*Phanton Power*) que pode variar de 15 a 48 Volts DC. Sua classificação se enquadra em um microfone condensador omnidirecional (BEHRINGER, 2008).

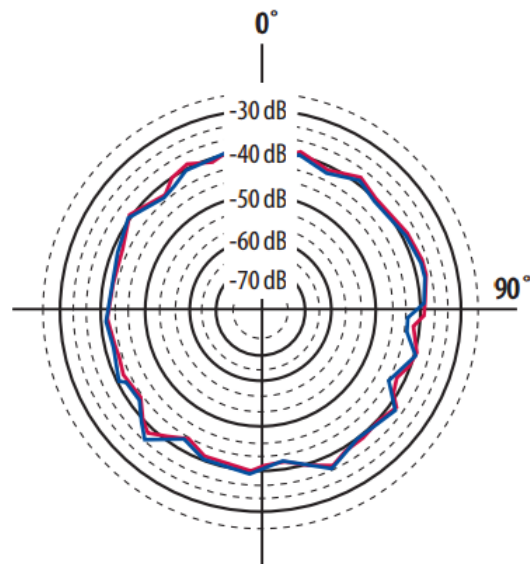
Devido às suas especificações (BEHRINGER, 2000), e sua curva de resposta, O microfone ECM8000 foi escolhido para a coleta dos dados gerados durante os testes. A sua curva de resposta é mostrada pela Figura 3.16.

Figura 3.16 – Curva de resposta ECM8000



Fonte: Behringer (2000)

Figura 3.17 – Gráfico polar ECM8000



Fonte: Behringer (2000)

Os dados foram coletados utilizando a técnica de varedura senoidal. O que possibilitou obter a resposta em frequência, resposta ao impulso e a distorção harmônica total do sistema, com a utilização do software REW.

3.7 Softwares de medições acústicas

Esta seção apresenta dois softwares de medição acústica, o primeiro deles é uma plataforma paga, que realiza medições em tempo real utilizando ruído rosa. O segundo é um software completo e gratuito, de simples utilização, e preciso nas medições.

3.7.1 Smaart: Software de análise de sinais de som

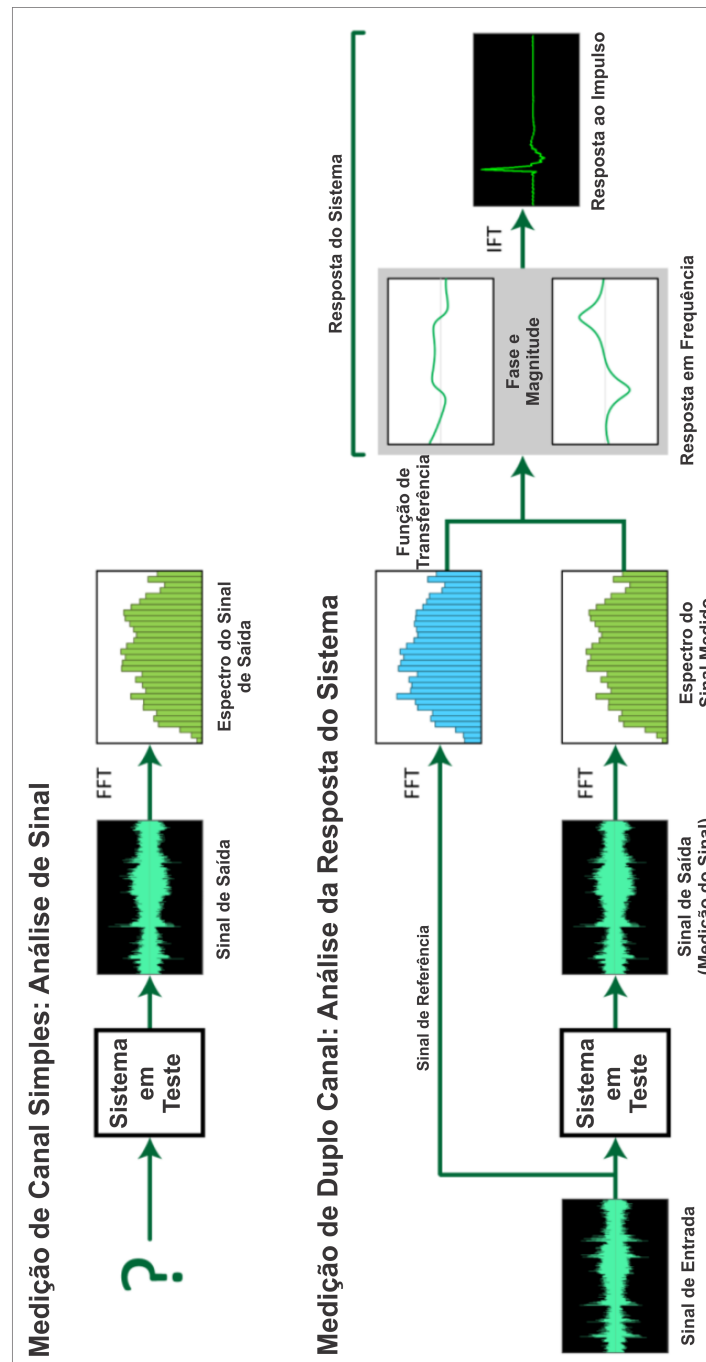
O Smaart⁵ é uma plataforma de software de análise acústica de duplo canal. O seu funcionamento é baseado na transformada rápida de Fourier (FFT – *Fast Fourier Transform*) que permite aos usuários medir e analisar as frequências contidas nos sinais de áudio, estudar a resposta em tempo e frequência de sistemas eletroacústicos, e realizar uma análise básica de salas acústicas. O software fornece análise do espectro em tempo real de sinais de áudio, análise da função de transferência da resposta de um sistema de som por meio de duplo canal, e a capacidade de análise e medição acústica da resposta a um impulso (ACOUSTICS, 2015).

Em tempo real, o Smaart fornece dois tipos básicos de medições: canal simples (análise de sinal) e canal duplo (análise de resposta). Medições de espectro de canal simples são medições de análise de sinal capaz de mostrar o teor das frequências e a amplitude de um sinal. Para esse tipo de análise, estão disponíveis espectrograma e analisador de espectro em tempo real (RTA – *Real Time spectrum Analyzer*) que são baseados na análise FFT de canal único (ACOUSTICS, 2015).

Medições de duplo canal comparam os dois sinais para encontrar as semelhanças e diferenças entre eles. A função de transferência e a medição de resposta ao impulso no Smaart são medições de duplo canal que comparam a saída de um dispositivo ou sistema com o sinal produzido na entrada do mesmo. Com isso, é realizada uma medição da resposta do sistema a um dado estímulo, e como ambos os sinais são conhecidos, o espectro do sinal de entrada torna-se quase irrelevante. Esse tipo de análise é capaz de medir com precisão a relação de tempo entre os dois sinais, o que permite analisar as relações de fase e encontrar tempos de atraso (ACOUSTICS, 2015).

⁵ <http://www.rationalacoustics.com/smaart/about-smaart/>

Figura 3.18 – Medições de canal simples e duplo



Fonte: adaptado de Acoustics (2015)

3.7.2 REW (*Room Eq Wizard*)

O REW⁶ é um programa gratuito de medição de alto falantes e sala acústica. Inclui ferramentas para a geração de sinais de teste de áudio; medição de SPL e impedância; medição de resposta de frequência e resposta ao impulso; medição de distorções; geração de gráficos de fase, atraso de grupo, decaimento espectral, cascatas, espectrogramas, curvas de tempo de

⁶ <https://www.roomeqwizard.com/>

energia; gera gráficos de análise em tempo real (RTA); calcula tempo de reverberação; calcula parâmetros *Thiele & Small*; determina frequência e o tempo de decaimento de ressonâncias modais (MULCAHY, 2017). É um programa bastante completo que funciona através da plataforma JAVA.

Segundo (PATE, 2014), o REW pode ser utilizado para realizar a caracterização da resposta em frequência de um sistema de alto falantes, essa caracterização requer minimizar não linearidades do processo. Através do uso do programa e de métodos que minimizam o erro de medição, é possível capturar uma representação precisa do desempenho do sistema.

O REW utiliza um sinal senoidal que é utilizado para realizar uma varredura logarítmica (MULCAHY, 2017). Torna-se muito mais rápido que medições manuais, mais precisas e menos propensas a sofrer cortes de ressonâncias, menos propícias a não linearidades do sistema que métodos MLS (MÜLLER; MASSARANI, 2001).

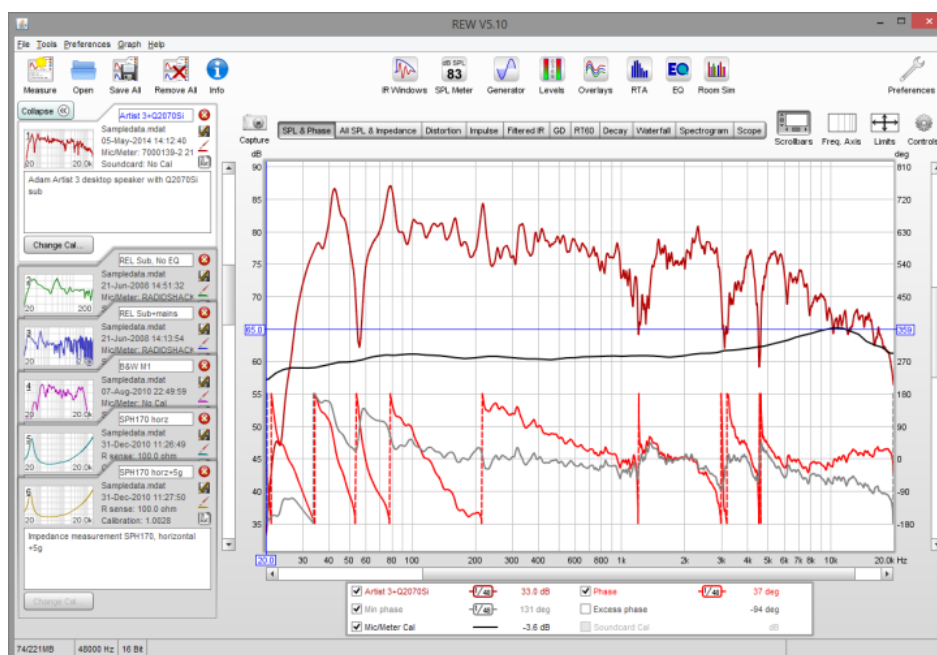
O programa trabalha do seguinte modo: com o uso de uma fonte sonora e um microfone de captação, um sinal de varredura logarítmica é enviado para a fonte. Esse sinal é um tom que começa em uma baixa frequência e esta frequência aumenta de forma constante. O que faz a varredura ser logarítmica é a taxa na qual a frequência muda, levando um tempo fixo para dobrar, ou seja, o tempo gasto para varrer de 20Hz a 40Hz é o mesmo gasto para varrer de 40Hz a 80Hz. O microfone capta o som que é irradiado diretamente da fonte, e também o som proveniente de reflexões das superfícies da sala (MULCAHY, 2017).

É aplicada a transformada rápida de Fourier (FFT) no sinal enviado e no sinal captado, a fim de calcular as frequências individuais (fase e amplitude) que compuseram a varredura. Ao realizar a função de transferência, ou seja, comparar as amplitudes e fases dos sinais, é possível saber como cada frequência foi afetada pelo sistema de medição. Com a função de transferência realizada, é utilizada uma FFT inversa para obter a resposta ao impulso do sistema (MULCAHY, 2017).

Com as entradas e saídas de áudio escolhidas, placa de som calibrada, níveis verificados, leitura de SPL calibrada, o REW está pronto para realizar as medições. A Figura 3.19 mostra a tela principal de medição do programa.

O tipo de pesquisa realizado neste trabalho enquadra-se em uma pesquisa experimental e de campo, onde foram realizados testes em campo e coletados os dados para análise, buscando métodos para dissolução do problema. É importante relevar que a ideia principal deste trabalho é aproximar com a realidade, porém a influência de variáveis externas podem negatar os

Figura 3.19 – Captura da tela principal do programa REW



resultados obtidos nos testes. Isso faz com que o cenário de teste utilizado seja um ambiente controlado, sem influências externas e com material absorvente, capaz de diminuir as reflexões internas.

3.8 Cenário de Teste

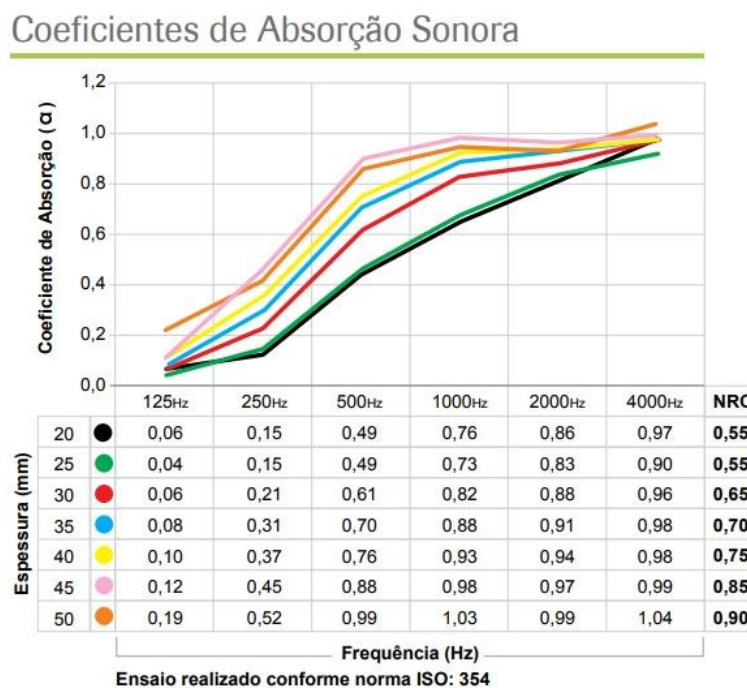
Os testes foram realizados com a intenção de mostrar os benefícios da utilização do protótipo responsável pelo direcionamento de som. Para que não haja interferências externas no processo de medição, foi escolhida uma sala acústicamente tratada construída utilizando o método box in box. Esse método é obtido com a sala acústica construída dentro de uma outra sala, mantendo um espaçamento entre as paredes internas e externas.

A sala acústica utilizada foi cedida pelo Centro Universitário de Lavras⁷ (UNILAVRAS), possui suas dimensões em $7,40 \times 5,20 \times 3,00$ m, sendo respectivamente comprimento, largura e altura. Durante a realização dos testes foi aferido a temperatura ambiente, que se manteve constante em 20 graus Celsius.

A sala é revestida internamente utilizando material absorvente sonoro com 50 mm de espessura, suas propriedades de absorção são mostradas na Figura 3.20 (SONEX, 2017). Esse material atenua as reverberações, melhorando a qualidade e a validação dos testes.

⁷ <<http://unilavras.edu.br/>>

Figura 3.20 – Coeficientes de absorção sonora Sonex



Fonte: Sonex (2017)

Para realizar a medição acústica do sistema, foi montado um cenário de teste dentro das especificações do software utilizado. A descrição deste cenário contribui para a compreensão dos testes.

3.9 Cenário de Medição

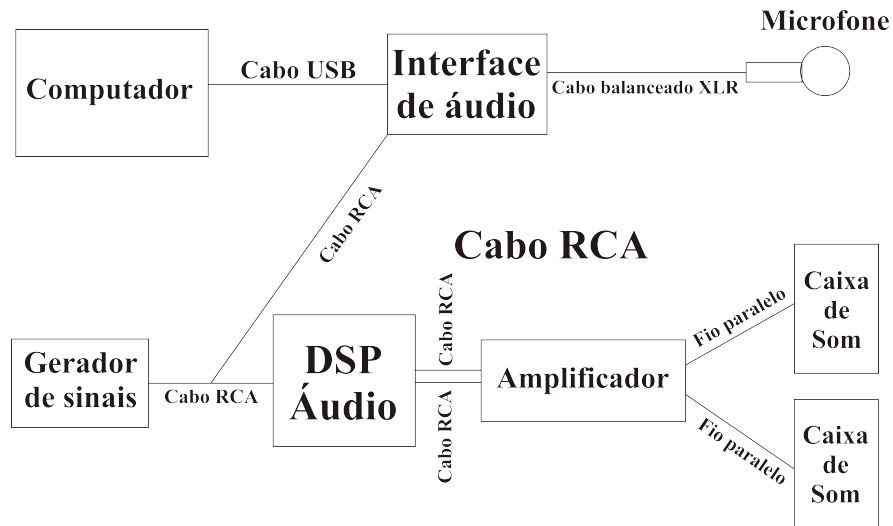
Um sistema de medição foi montado para caracterizar a resposta em frequência do sistema a ser testado. O sistema de medição é composto de um software capaz de utilizar o método de varredura senoidal, com uma variação exponencial de 20 Hz a 20 kHz e duração de 10,9 segundos.

O sinal é enviado a uma interface de áudio que possui duas saídas e duas entradas. Uma cópia do sinal de saída é enviado a uma das entradas da interface, pois assim é possível comparar o sinal enviado com o sinal medido. A outra cópia é enviada ao processador (protótipo) que realiza as operações necessárias ao sinal de áudio, posteriormente o sinal de áudio é enviado ao amplificador.

O amplificador, por sua vez, tem apenas a função de amplificar o sinal de áudio e enviar às fontes sonoras. Após a emissão do som pelas fontes sonoras, este sinal é captado pelo microfone omnidirecional balanceado que está ligado na outra entrada da interface. O sinal é

processado pelo computador e os resultados são exibidos pelo software. A Figura 3.21 exemplifica este cenário.

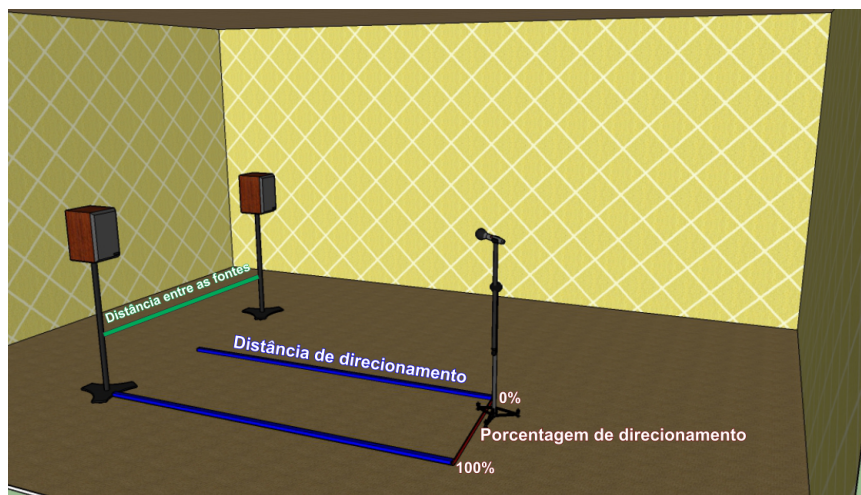
Figura 3.21 – Cenário de teste do sistema de medição acústica



Fonte: do Autor

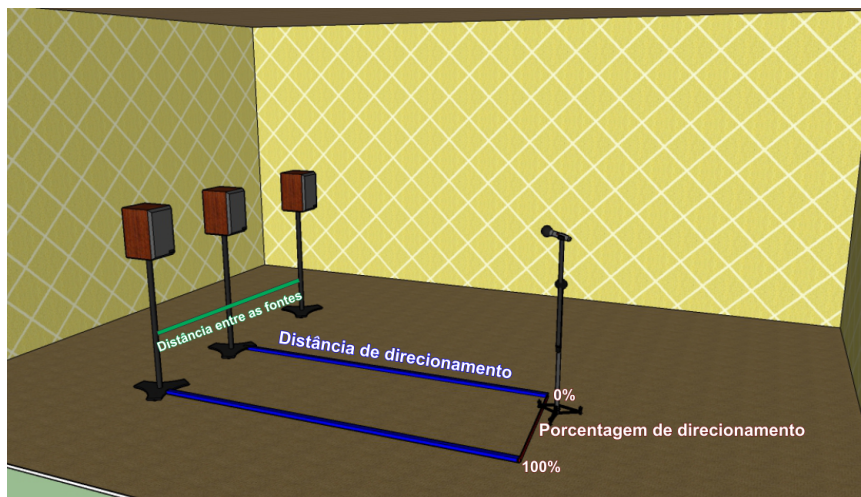
Para obter os resultados dos testes, foram realizados testes utilizando duas fontes sonoras (direita e esquerda) mostrado na Figura 3.22, e três fontes sonoras (direita, centro e esquerda) mostrado na Figura 3.23. Durante a realização dos testes, foi alterada a distância entre as fontes, a distância de direcionamento e a porcentagem de direcionamento.

Figura 3.22 – Cenário de teste com duas fontes sonoras



Fonte: do Autor

Figura 3.23 – Cenário de teste com três fontes sonoras



Fonte: do Autor

4 RESULTADOS

Para analisar a resposta acústica do sistema de teste, foi utilizado um sinal de entrada senoidal, cuja a frequência é variada dentro de uma faixa preestabelecida. Foram realizados um total de 139 testes com 17 variações, onde foi possível comprovar a eficácia do protótipo. Para cada um dos testes foram feitas 10 medições com objetivo de minimizar interferências externas, totalizando 1390 medições. A partir dessas 10 medições foi extraído uma média entre elas. A Figura 4.1 mostra o cenário real de teste.

Figura 4.1 – Foto do cenário de teste



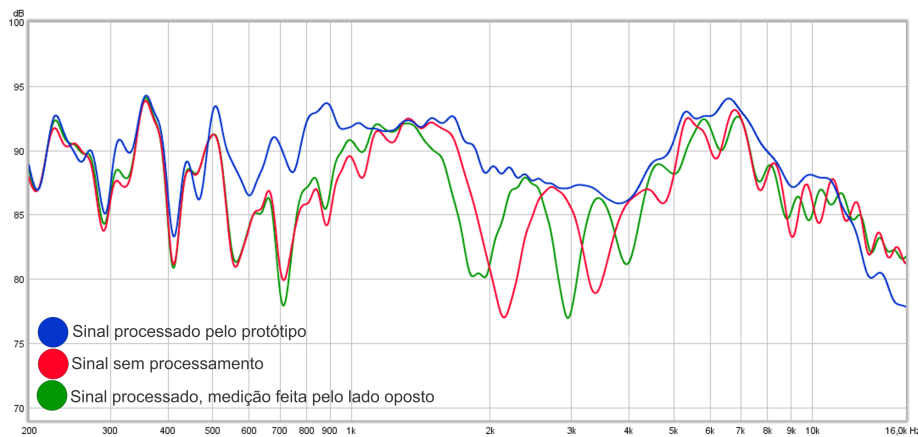
Fonte: do Autor

A avaliação dos resultados foi feita através da análise da resposta em frequência pela pressão sonora, obtido através da transformada discreta de Fourier da resposta ao impulso. Foi realizado uma comparação das curvas de resposta em frequência das medições do sistema com e sem a função de direcionamento.

4.1 Resultados da medição com duas fontes sonoras (esquerda e direita)

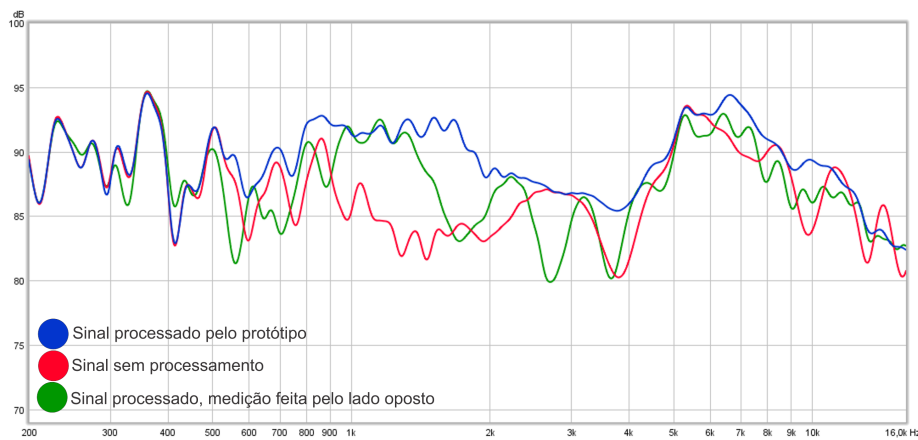
O primeiro cenário analisado é composto por duas fontes sonoras com os centros acústicos posicionados a um metro de distância um do outro. O uso de duas fontes representa lado esquerdo e lado direito. O microfone foi posicionado há um metro de distância perpendicular à linha imaginária que liga os centros acústicos das fontes. A configuração do protótipo foi feita com 20° C, distância entre as fontes 1 metro, distância de direcionamento 1 metro e porcentagem de direcionamento de 60% mostrado na Figura 4.2, 80% na Figura 4.3 e 100% na Figura 4.4.

Figura 4.2 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 1 metro e 60% de porcentagem de direcionamento



Fonte: do Autor

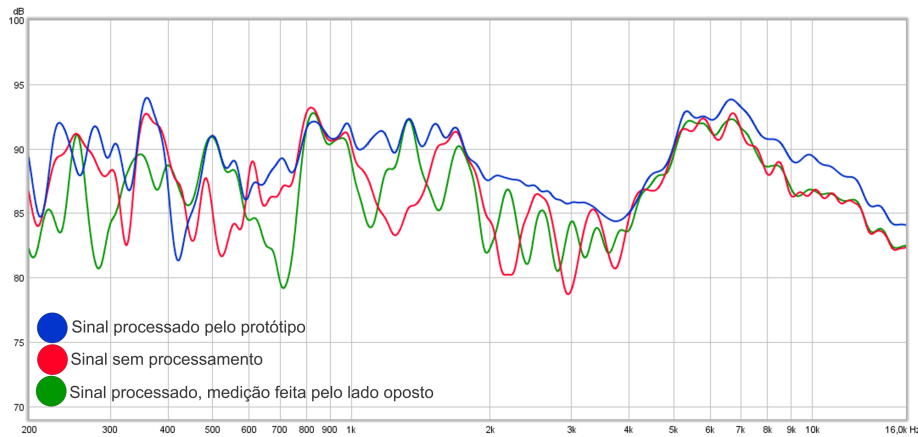
Figura 4.3 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 1 metro e 80% de porcentagem de direcionamento



Fonte: do Autor

É possível observar na Figura 4.2 que a medição do sinal processado, representado pela curva azul, possui uma pressão sonora maior ou igual ao sinal não processado. É considerável

Figura 4.4 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 1 metro e 100% de porcentagem de direcionamento



Fonte: do Autor

a medição acima de 350 Hz, pois abaixo disso, pela distância entre as fontes, o sistema está acoplado devido à distância entre as fontes.

Além de uma pressão sonora mais elevada, é possível visualizar claramente que a curva da resposta é mais homogênea. O ponto mais importante a se destacar é a correção do problema denominado *comb filter* descrito na Seção 2.1.4. Esse problema é claramente mostrado pelos picos e vales periódicos presentes nas curvas vermelha e verde. Quando esses picos e vales estão presentes em uma reprodução musical, causa uma degradação sonora. O mesmo ocorre nas Figuras 4.3 e 4.4, onde a medição do sinal processado apresenta as mesmas características da Figura 4.2.

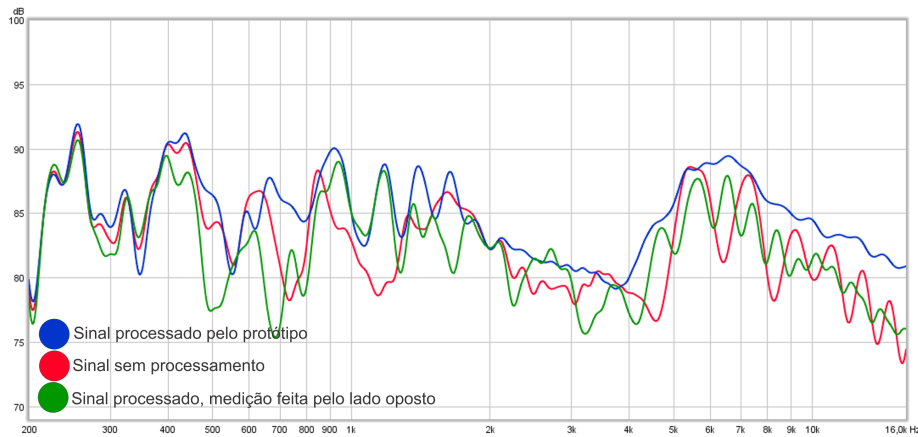
A demonstração de testes em cenários com outras dimensões é importante para validar os resultados. As Figuras 4.5 e 4.6 mostram as medições com a seguinte configuração: distância entre as fontes de 1 metro, distância de direcionamento de 2 metros, porcentagem de direcionamento de 80% e 100%.

Conforme apresentado nas figuras referentes à medição com duas fontes a 1 metro e distância de direcionamento de 1 metro, a resposta do sistema apresentada nas Figuras 4.5 e 4.6 manteve as características de homogeneidade, controle de *comb filters* e uma maior pressão sonora.

4.2 Resultados da medição com três fontes sonoras (esquerda, centro e direita)

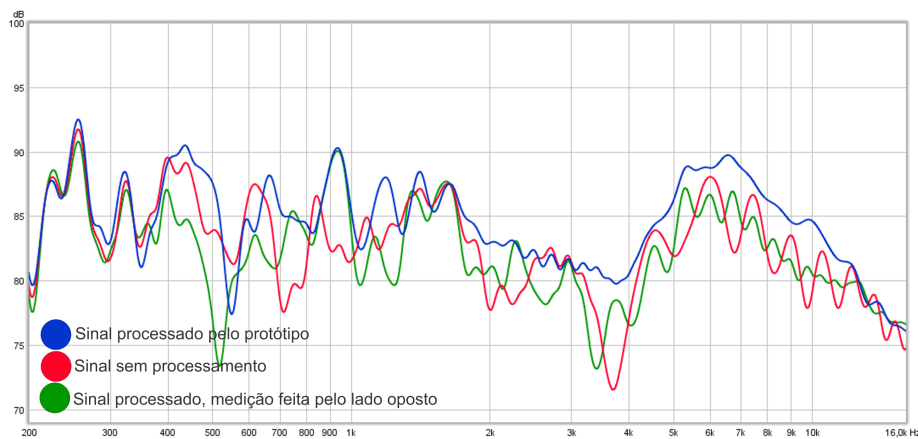
A adição de um canal central nas medições é importante para validar o uso do protótipo de direcionamento de áudio. A distância do canal central para as fontes da esquerda e da direita

Figura 4.5 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 2 metros e 80% de porcentagem de direcionamento



Fonte: do Autor

Figura 4.6 – Gráfico da resposta em frequência da medição com duas fontes, distância de 1 metro entre elas, distância de direcionamento de 2 metros e 100% de porcentagem de direcionamento



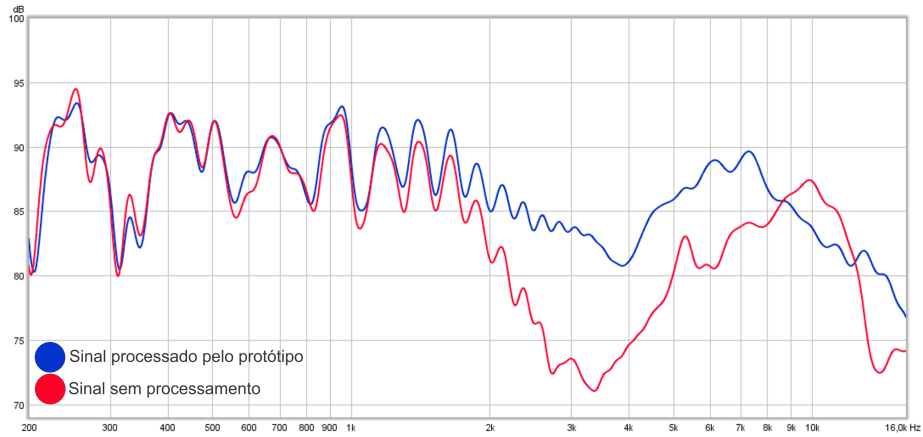
Fonte: do Autor

deve ser a mesma. O cenário com três fontes avaliado nos resultados deste trabalho é composto pela distância entre as fontes externas de 1,50 metros, distância de direcionamento de 2 metros, e porcentagens de direcionamento de 13%, 27%, 40% e 80%. Esses cenários são mostrados nas Figuras 4.7, 4.8, 4.9, e 4.10 respectivamente.

Em todos os gráficos do cenário de três fontes sonoras, é possível visualizar as propriedades mostradas nos testes com 2 fontes. Nesse local é realizado um controle de *comb filters*, é obtida uma curva resposta mais homogênea e maior pressão sonora. A Figura 4.9 do cenário referente a 40% de direcionamento deixa bem clara a maior linearidade na curva de resposta e também o controle de *comb filter*.

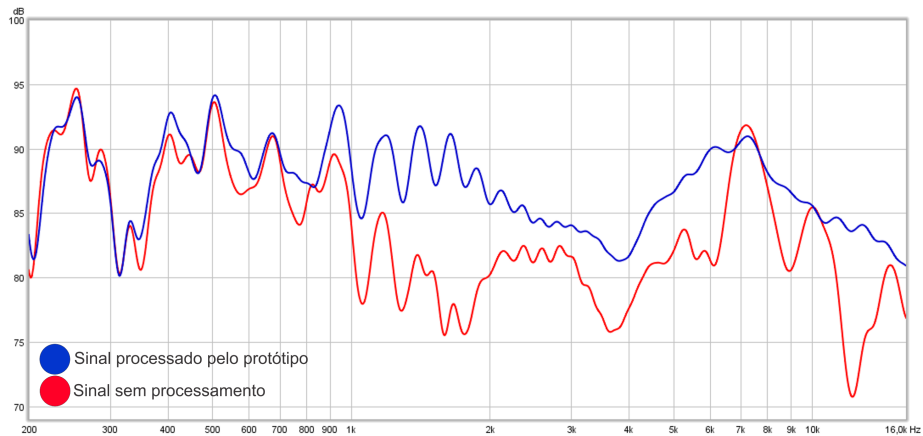
Através da medição por varredura senoidal, é possível extrair distorções harmônicas, dentre elas a distorção harmônica total denominada THD. Para o cenário de teste com três fon-

Figura 4.7 – Gráfico da resposta em frequência da medição com três fontes, distância de 1,5 metros entre elas, distância de direcionamento de 2 metros e 13% de porcentagem de direcionamento



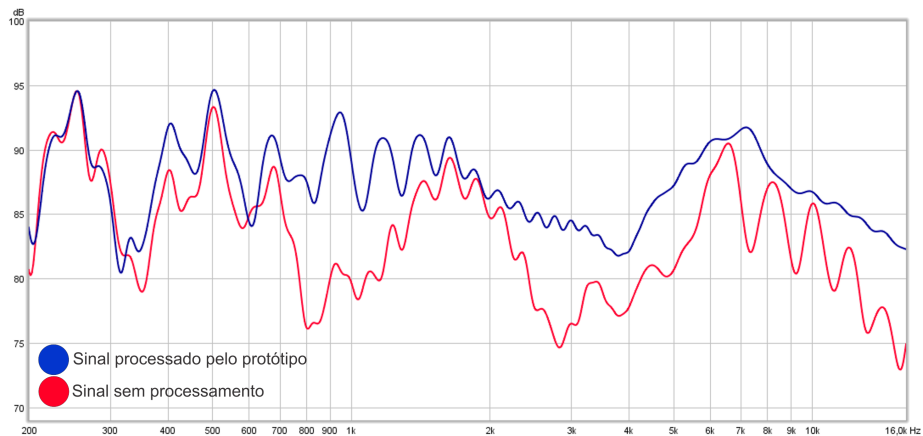
Fonte: do Autor

Figura 4.8 – Gráfico da resposta em frequência da medição com três fontes, distância de 1,5 metros entre elas, distância de direcionamento de 2 metros e 27% de porcentagem de direcionamento



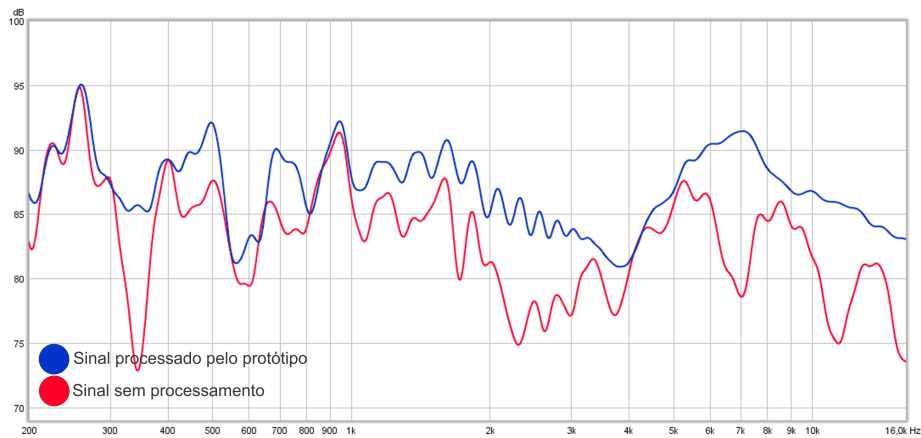
Fonte: do Autor

Figura 4.9 – Gráfico da resposta em frequência da medição com três fontes, distância de 1,5 metros entre elas, distância de direcionamento de 2 metros e 40% de porcentagem de direcionamento



Fonte: do Autor

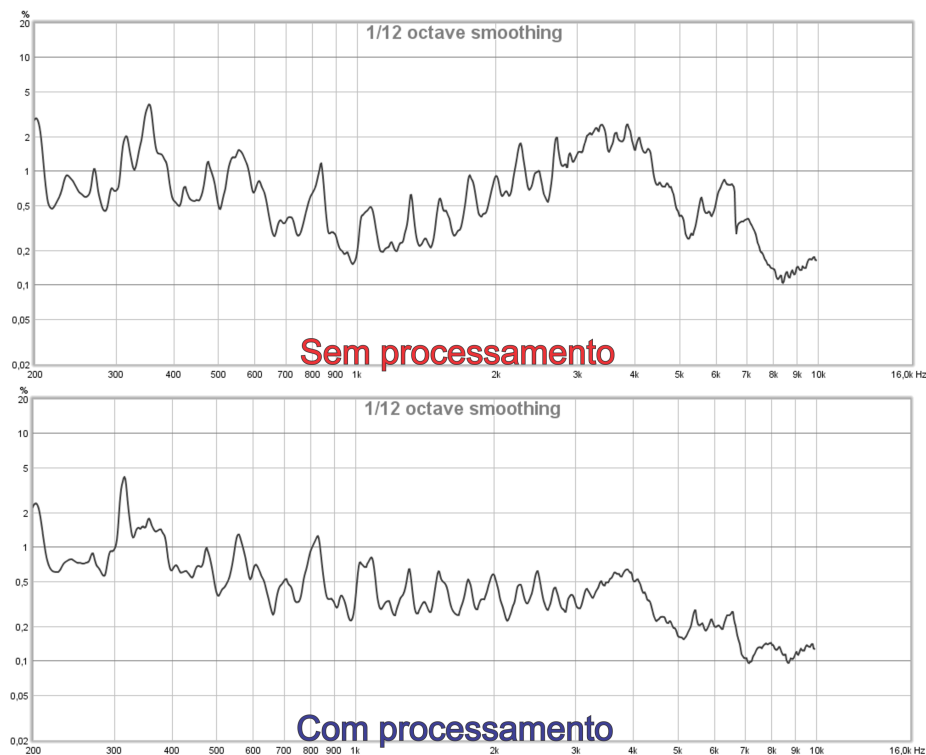
Figura 4.10 – Gráfico da resposta em frequência da medição com três fontes, distância de 1,5 metros entre elas, distância de direcionamento de 2 metros e 80% de porcentagem de direcionamento



Fonte: do Autor

tes sonoras destacado nesta seção, pode-se comparar a THD, quando se aplica o processamento no sinal, direcionando o áudio, e quando não se aplica. Essa comparação é mostrada nas Figuras 4.11, 4.12 e 4.13.

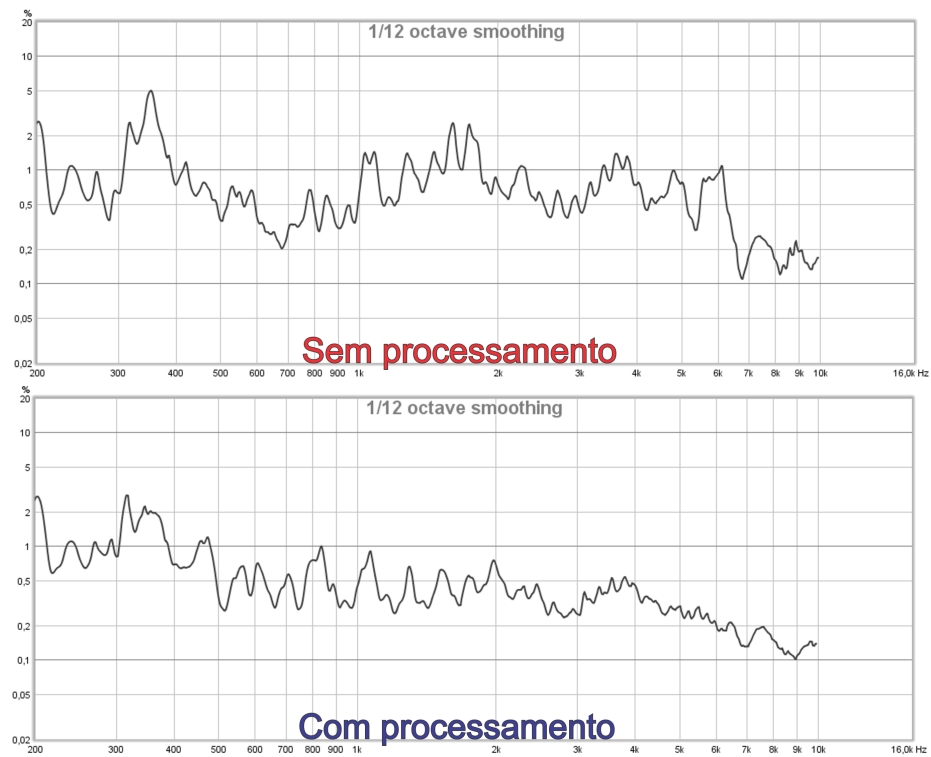
Figura 4.11 – Distorção Harmônica Total com 3 fontes e 13% de direcionamento



Fonte: do Autor

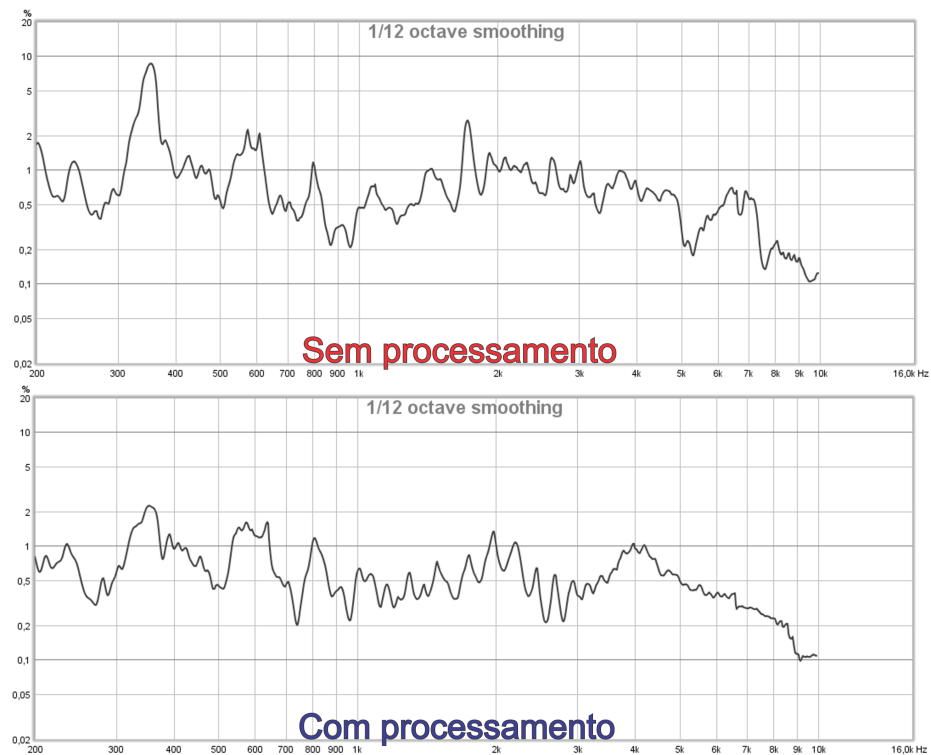
Analisando os resultados, é possível afirmar que a THD medida no ponto de direcionamento é menor do que a medida quando não se usa processamento. As Figuras 4.11 e 4.12

Figura 4.12 – Distorção Harmônica Total com 3 fontes e 27% de direcionamento



Fonte: do Autor

Figura 4.13 – Distorção Harmônica Total com 3 fontes e 80% de direcionamento



Fonte: do Autor

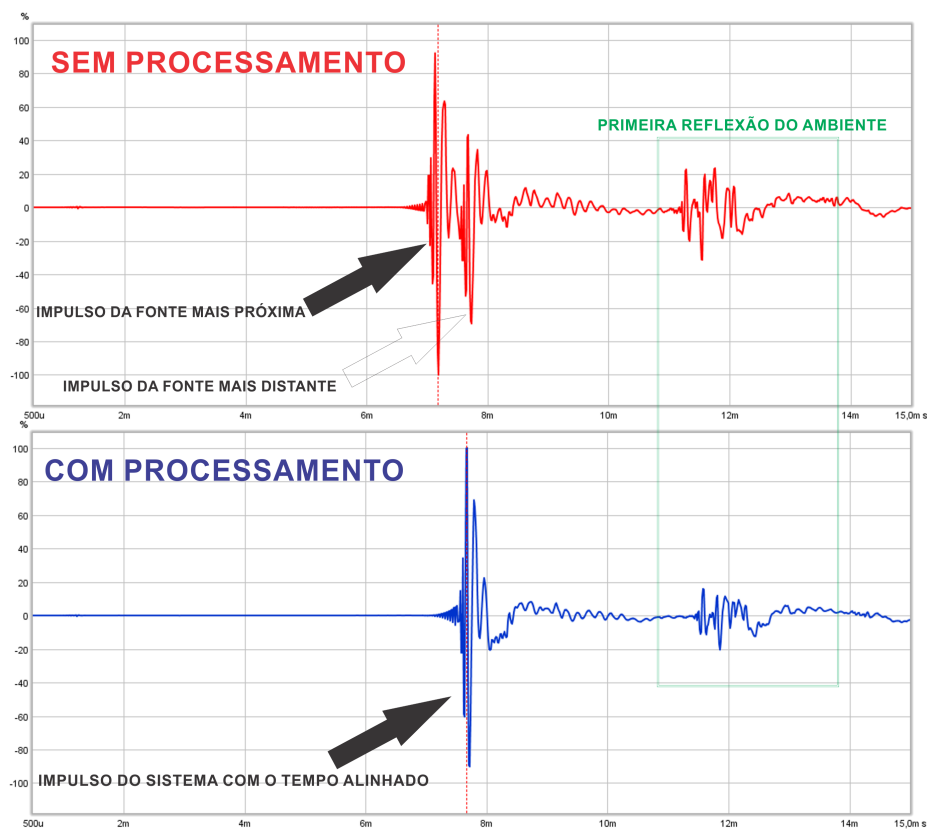
mostram uma THD mais linear, em relação a variação de frequências, na medição com sinal processado pelo protótipo.

4.3 Análise da resposta ao impulso do sistema

Interpretar a resposta ao impulso é uma parte importante da análise acústica. A resposta ao impulso é a responsável pela geração da resposta em frequência, pois nela está contida todas as ondas senoidais utilizadas na varredura do sistema de testes. É possível analisar o impulso no eixo do tempo, e verificar a reação do sistema sem o uso do protótipo e com o uso do protótipo.

A Figura 4.14 apresenta a comparação entre a resposta ao impulso do sistema utilizando o protótipo para processar o sinal, e a resposta sem utilizar o processamento. A medição foi extraída do sistema configurado com duas fontes sonoras posicionadas a 1 metro de distância e distância de direcionamento de 2 metros, microfone posicionado a 80%. É possível notar claramente a função do protótipo em alinhar os tempos das fontes sonoras.

Figura 4.14 – Resposta ao impulso do sistema com duas fontes sonoras

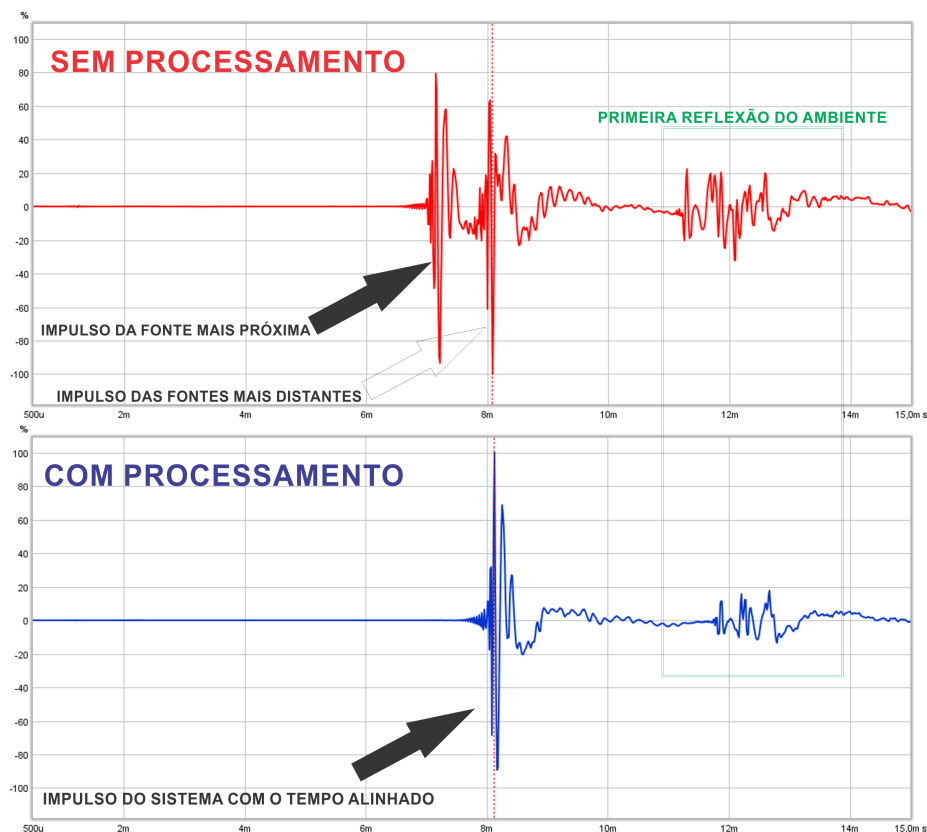


Fonte: do Autor

A Figura 4.15 apresenta a resposta ao impulso do sistema com três fontes sonoras externas posicionadas a 1,5 metros de distância, distância de direcionamento de 2 metros e porcen-

tagem de direcionamento de 80%. Ao comparar as respostas ao impulso com e sem processamento é possível afirmar que o conjunto formado pela função de direcionamento desenvolvida e o DSP capaz de atrasar amostras, realiza o direcionamento do som. É possível notar, claramente, o alinhamento temporal do sinal com processamento, e se for comparado com a Figura 4.14, as respostas ao impulso sem processamento, é perceptível a presença de mais de uma fonte sonora onde está indicado "impulso das fontes mais distantes".

Figura 4.15 – Resposta ao impulso do sistema com três fontes sonoras



Fonte: do Autor

Conforme citado na Seção 2.4.2, é possível observar nas Figuras 4.14 e 4.15, a captação de reflexões pelo microfone. Os efeitos das reflexões da sala acústica é captado posteriormente a alguns instantes de silêncio que ocorre após a captação da resposta ao impulso do sistema de som.

5 CONCLUSÃO

Para o desenvolvimento deste trabalho, foi necessário compreender os fundamentos do áudio e da acústica, o que possibilitou planejar um modo de controlar as interações entre ondas complexas. Esse controle é realizado pelo ajuste de fase das fontes sonoras, fazendo com que o som reproduzido por duas ou três fontes cheguem a um determinado ponto com a mesma fase ocasionando a soma entre elas.

O desenvolvimento do protótipo foi possível com o auxílio da empresa Expert Electronics, que disponibilizou os componentes eletrônicos e também auxiliou na programação. Com uso das ferramentas disponíveis, foi possível desenvolver o protótipo objetivo deste trabalho.

Para confirmar a eficiência do método de controle de interferências destrutivas, foi necessário a realização de testes. Esses testes envolvem a avaliação da resposta em frequência e resposta ao impulso do sistema de som composto por duas e três fontes sonoras, medição de distorção harmônica total do sistema com e sem processamento do áudio. As ferramentas e métodos de medição acústica foram imprescindíveis para a análise dos resultados, possibilitando validar, e comprovar o direcionamento do som.

Para complementar o trabalho e aumentar a coerência dos resultados, as caixas acústicas utilizadas foram projetadas de modo a possuírem a resposta ao um sinal mais próximas possíveis. Com a medição da curva de impedância tanto dos transdutores utilizados, quanto das caixas fechadas, é provada a semelhança entre as três fontes sonoras utilizadas nos testes deste trabalho.

Os resultados mostram favoráveis ao protótipo proposto. Foi possível obter uma resposta mais homogênea no ponto de direcionamento, amenizando a presença de *comb filters* e também uma taxa de distorção harmônica total menor e mais homogênea nas medições com a utilização do método de direcionamento proposto. Ao analisar a resposta ao impulso dos testes, é comprovada a eficácia do protótipo em realizar o alinhamento de tempo entre as caixas no ponto desejado.

É possível a utilização do protótipo por qualquer pessoa leiga no assunto, fazendo com que o ouvinte tenha uma experiência de ouvir um som mais homogêneo, sem a coloração provocada pelo fenômeno *comb filter*, e com uma distorção harmônica mais baixa.

5.1 Dificuldades encontradas

O desenvolvimento deste trabalho desenvolveu-se de forma favorável, porém algumas dificuldades foram encontradas durante o desenvolvimento. Uma delas é que os principais materiais próximos ao objetivo deste trabalho estão vinculados a produtos sob patentes.

A distância física em relação à empresa Expert Eletronics fez com que as visitas à empresa fossem somente com urgência, porém é destacável a dedicação e a prontidão para sanar possíveis dúvidas referentes tanto à programação quanto ao circuito elétrico e também à ajuda para a finalização do projeto.

A principal dificuldade encontrada foi localizar um ambiente acústico ideal disponível para a realização das medições, este ambiente seria uma câmara anecóica. O ambiente encontrado para utilização foi o mais próximo do ideal possível, e foi escolhido devido aos recursos disponíveis. Porém, a disponibilização de datas e horários mais longos para a utilização da sala, impossibilitou uma maior variação dos testes.

5.2 Proposta de continuidade

Como continuidade do produto, é possível o desenvolvimento de uma função que compense a diferença de absorção sonora pelo ar, provocada pela diferença de distância entre as fontes. Também uma função que ajuste os níveis sonoros observando o fator de diretividade do pistão, baseado no ângulo formado entre o ponto de direcionamento e a fonte sonora. A continuidade proposta é um equalizador dinâmico que compense as propriedades citadas, aperfeiçoando o método proposto por este trabalho.

Com a possibilidade de desenvolvimento de um equalizador dinâmico, pode ser realizada uma adaptação do protótipo para que seja utilizado em veículos e se transforme em um produto de mercado passível de comercialização.

REFERÊNCIAS

- ABREU, D. H. S.; RODRÍGUEZ, D. Z.; LACERDA, W. S. Impact of the linear phase variations on the voice quality index. **The 20th IEEE International Symposium on Consumer Electronics**, São Paulo, Brasil, Set 2016.
- ACOUSTICS, L. R. **Rational Acoustics Smart v7 User Guide**. Putnam, EUA, 2015. Disponível em: <<http://www.rationalacoustics.com/download/Smartv7UserGuide.pdf>>.
- BALLOU, G.; CIAUDELLI, J.; SCHMITT, V. Microphones. **Handbook for Sound Engineers**, Taylor & Francis, p. 101, 2008.
- BEHRINGER, M. G. Measurement microphone ecm8000: Technical specifications. **Acessado em: 26 Jun. 2016**, 2000. Disponível em: <<http://www.coutant.org/behringer/ecm8000c.pdf>>.
- BEHRINGER, M. G. Ecm8000: Ultra-linear measurement condenser microphone. **Acessado em: 15 Mar. 2016**, 2008. Disponível em: <<http://www.music-group.com/Categories/Behringer/Microphones/Condenser-Microphones/ECM8000/p/P0118>>.
- BROWN, P. Fundamentals of audio and acoustics. **Handbook for Sound Engineers**, Taylor & Francis, p. 22, 2013.
- CHAVEZ, M. A.; GERMAN, R. Automatic eq, crossover and alignment of speaker systems. In: **ALMA Winter Symposium**. [s.n.], 2011. Disponível em: <http://www.almainternational.org/yahoo_site_admin/assets/docs/chavez-automatic_eq_crossover_and_alignment_of_speaker_systems.106170930.pdf>.
- CHAVEZ, M. A.; HUIN, C. Sigmastudio. a user friendly, intuitive and expandable, graphical development environment for audio/dsp applications. In: **Audio Engineering Society Convention 120**. [s.n.], 2006. Disponível em: <<http://www.aes.org/e-lib/browse.cfm?elib=13518>>.
- DICKASON, V. Alto-falantes e caixas acústicas. **Tradução de Homero Sette Silva. 6ª ed. ampliada, Rio de Janeiro: H. Sheldon**, 2006.
- EVEREST, F. A.; POHLMANN, K. C. **The master handbook of acoustics**. [S.l.]: McGraw-Hill New York, 2009. v. 5.
- FARINA, A. Simultaneous measurement of impulse response and distortion with a swept-sine technique. In: AUDIO ENGINEERING SOCIETY. **Audio Engineering Society Convention 108**. Paris, França, 2000.
- FARINA, A. Advancements in impulse response measurements by sine sweeps. In: AUDIO ENGINEERING SOCIETY. **Audio Engineering Society Convention 122**. Vienna, Austria, 2007.
- FELDMAN, M. **Hilbert transform applications in mechanical vibration**. [S.l.]: John Wiley & Sons, 2011.
- ITU-T, R. P. 800. **Methods for subjective determination of transmission quality**, p. 800–899, 1996.
- ITU-T, R. P. 862. **Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs**, v. 23, 2001.

- KINSLER, L. E. et al. Fundamentals of acoustics. **Fundamentals of Acoustics, 4th Edition**, pp. 560. ISBN 0-471-84789-5. Wiley-VCH, December 1999., p. 560, 1999.
- LATHI, B. P. **Sinais e Sistemas Lineares Segunda Edição** .(GG PARMA, Trad.) Porto Alegre, Rio Grande do Sul. [S.l.]: Brasil: Bookman, 2007.
- LAZZARINI, V. E. Elementos de acústica. **apostila do Departamento de Artes da UEL, Londrina, Brasil**, 1998.
- MITCHELL, J. loudspeakers. **Handbook for Sound Engineers**, Taylor & Francis, p. 44, 2008.
- MULCAHY, J. Room eq wizard. **Acessado em: 26 Jul. 2017**, 2017. Disponível em: <<https://www.roomeqwizard.com/index.html>>.
- MÜLLER, S.; MASSARANI, P. Transfer-function measurement with sweeps. **Journal of the Audio Engineering Society**, Audio Engineering Society, Duque de Caxias, Brasil, v. 49, n. 6, p. 443–471, 2001.
- OLSON, H. F. **Acoustical engineering**. Nova York, EUA: van Nostrand, 1957.
- OPPENHEIM, A. V.; SCHAFER, R. W. **Discrete-Time Signal Processing**. 3rd. ed. Upper Saddle River, EUA: Prentice Hall Press, 2009. ISBN 0131988425, 9780131988422.
- PATE, T. Audio characterization primer. **Application Report**, Texas Instrument, Dallas, EUA, Ago 2014.
- POWERSOFT. **Powersoft SigmaStudio user guide v1.0**. Scandicci, Italy, 2011. Disponível em: <<http://www.powersoft-audio.com/it/downloads/documentazione/user-guides/software-1/419-sigmastudio-user-guide/file>>.
- RICHARDSON, D. C. Dsp technology. **Handbook for Sound Engineers**, Taylor & Francis, p. 17, 2008.
- RIFE, D. D. Modulation transfer function measurement with maximum-length sequences. **Journal of the Audio Engineering Society**, Audio Engineering Society, Sterling, EUA, v. 40, n. 10, p. 779–790, 1992.
- RODRÍGUEZ, D. Z. **Algoritmo para determinação da taxa de transmissão em uma rede IP**. Tese (Doutorado) — Universidade de São Paulo, São Paulo, Brasil, 2009.
- SONEX, O. **Sonex illtec Acústica e Design**. São Paulo, Brasil, 2017. Disponível em: <<http://www.owa.com.br/produtos/placas-acusticas-sonex-illtec/catalogo.pdf>>.
- STAN, G.-B.; EMBRECHTS, J.-J.; ARCHAMBEAU, D. Comparison of different impulse response measurement techniques. **Journal of the Audio Engineering Society**, Audio Engineering Society, Liège, Belgica, v. 50, n. 4, p. 249–262, 2002.
- STMicroelectronics. Datasheet stm32f051x4, stm32f051x6, stm32f051x8. **Acessado em: 22 fev. 2016**, 2008. Disponível em: <<http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00220364.pdf>>.
- TEIXEIRA, M. M. Interferência de ondas. **Acessado em: 15 setembro. 2017**, 2017. Disponível em: <<http://mundoeducacao.bol.uol.com.br/fisica/interferencia-ondas.htm>>.

TOOLE, F. E. **Sound reproduction: Loudspeakers and rooms**. [S.l.]: Taylor & Francis, 2008.

TORGER, A.; FARINA, A. Real-time partitioned convolution for ambiophonics surround sound. In: IEEE. **Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the**. Nova York, EUA, 2001. p. 195–198.

VANDERKOOY, J. Aspects of mls measuring systems. **Journal of the Audio Engineering Society**, Audio Engineering Society, Waterloo, Canada, v. 42, n. 4, p. 219–231, 1994.

APÊNDICE A – Outros resultados obtidos

O apêndice apresenta alguns resultados preliminares obtidos durante o desenvolvimento do projeto.

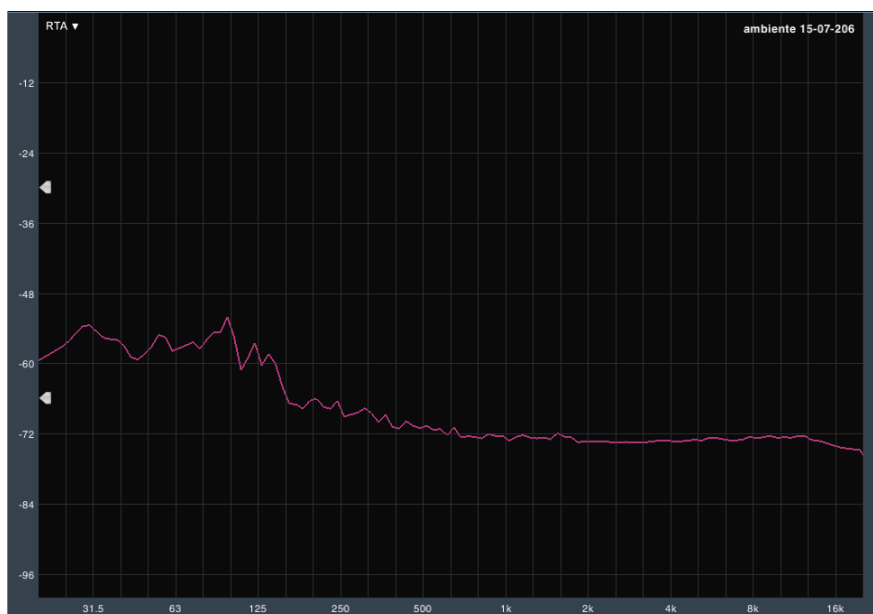
TESTE DE ESTIMATIVA DA QUALIDADE DE VOZ EM SISTEMAS COM ALINHAMENTO DE TEMPO

Foram realizados alguns testes com a finalidade de avaliar a qualidade de sinal de voz utilizando a recomendação ITU-T P.862 e a resposta em frequência obtida pelo software Smaart de um sistema de som duas vias com duas fontes sonoras.

Especificação do cenário de teste

O cenário de teste foi composto por uma sala comum sem tratamento acústico com pouca influência de ruído externo. A resposta em frequência do ruído no ambiente é apresentada na figura 1.

Figura 1 – Resposta em frequência do ambiente de teste.



O ambiente de teste foi dividido em 6 cenários para avaliação de resultados experimentais. Os cenários 1, 2 e 3 são comparados entre si, assim como os cenários 4, 5, e 6. Utilizando arquivos de áudio fornecidos pela P.862 (ITU-T, 2001) foram gravados uma voz masculina e

uma voz feminina para cada um dos cenários. Também foi capturado a resposta em frequência do sistema através do modo de duplo canal do software Smaart.

A temperatura no ambiente de teste foi aferida em 23 graus Celsius, ou 296 graus Kelvin. Segundo (LAZZARINI, 1998) a velocidade de propagação do som no ambiente de teste é de aproximadamente $345,8 \text{ ms}^{-1}$, obtida pela equação 1 onde T é a temperatura em graus Kelvin.

$$v = 20.1\sqrt{T}ms^{-1} \quad (1)$$

Os cenários de testes são apresentados pela figura 2. É importante destacar que nos cenários 3 e 6 foram inseridos atrasos de tempo na fonte mais próxima do microfone pelo processador digital de sinais. Os tempos de chegada do som de cada fonte sonora até o microfone foi calculado pelo software Smaart, e a diferença de tempo foi calculada. A diferença de tempo calculada foi inserida como um atraso na fonte mais próxima proporcionando que o som de ambas as fontes chegue ao microfone no mesmo tempo.

A inserção de atrasos é possível através da opção de atraso por amostras realizado pelo processador. Operando com uma taxa de amostragem de 48kHz é possível adicionar um atraso de um número inteiro multiplicado por $20,83\mu\text{s}$ ao sinal. No caso dos cenários 3 e 6 respectivamente foram adicionados atrasos de 58 e 94 amostras.

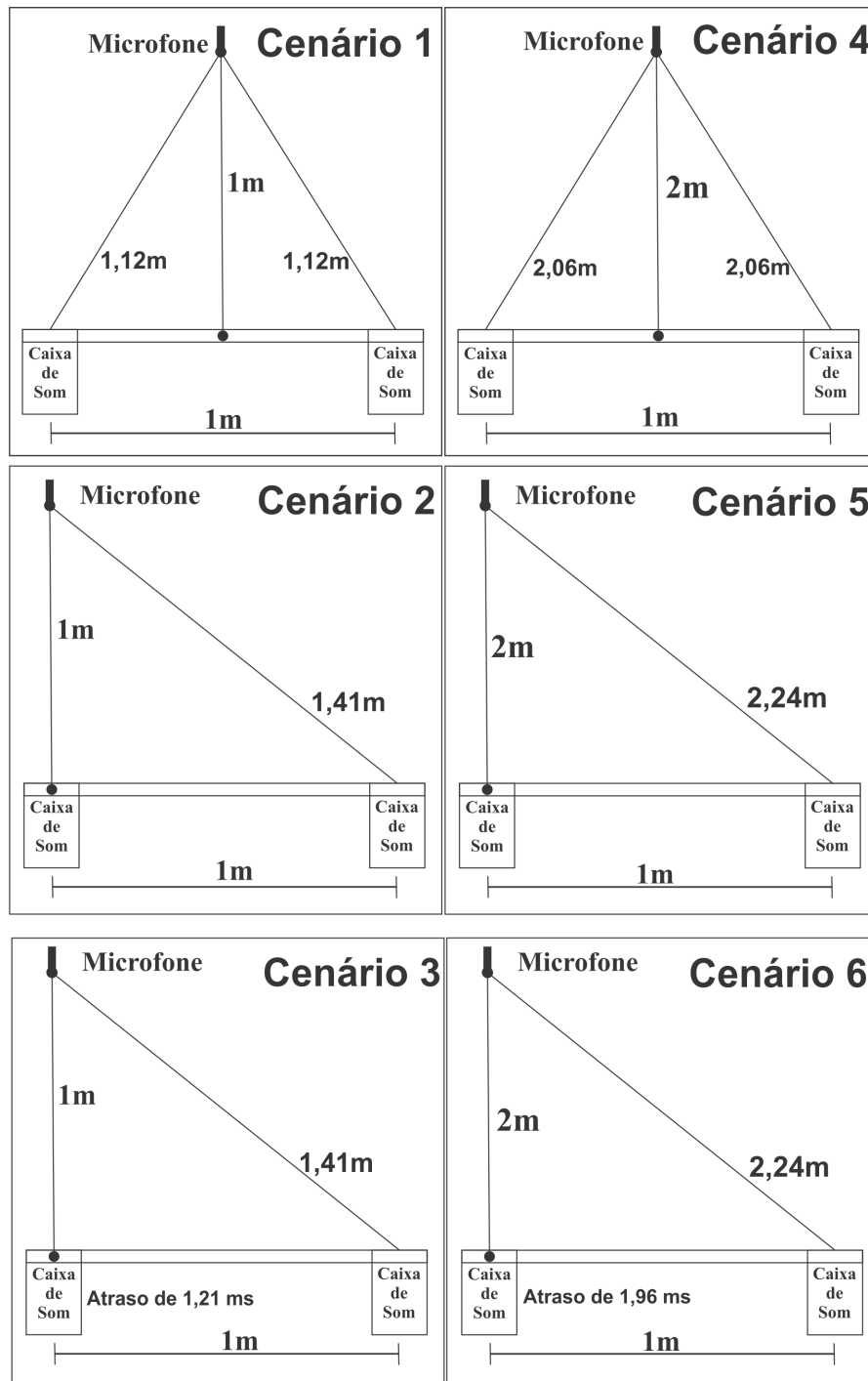
Resultados dos testes

Os resultados obtidos nos testes experimentais mostram que é possível obter uma melhor qualidade na reprodução de voz quando se realiza um alinhamento de tempo entre as fontes sonoras. A tabela 1 apresenta os resultados pelo método P.862 realizado comparando o sinal original enviado às fontes sonoras com a gravação em cada um dos cenários. É possível observar que os cenários 3 e 6 obtiveram uma maior nota se comparados com os demais (3 comparado com 1 e 2, 6 comparado com 4 e 5).

É observado também na figura 10 que a curva de resposta do sistema no cenário 6, representado pela linha verde, é mais plana e com uma amplitude maior. São sinais de uma maior cobertura das frequências (linha plana) e um nível maior de intensidade sonora (maior amplitude).

Os resultados dos testes experimentais é um avanço na pesquisa. Concluindo que através do alinhamento temporal entre as fontes é possível obter uma resposta que atenda uma gama

Figura 2 – Cenários de teste.



maior de frequências, uma maior amplitude no sinal, e um nível mais elevado de qualidade de voz. Considerando uma reprodução com duas fontes sonoras.

Tabela 1 – Resultados utilizando a recomendação ITU-T P.862

Cenários	Áudio 1	Áudio 2
Cenário 1	1,52	1,46
Cenário 2	1,58	1,46
Cenário 3	1,65	1,47
Cenário 4	1,73	1,72
Cenário 5	1,72	1,75
Cenário 6	1,76	1,78

Figura 3 – Resposta do sistema no cenário 1.

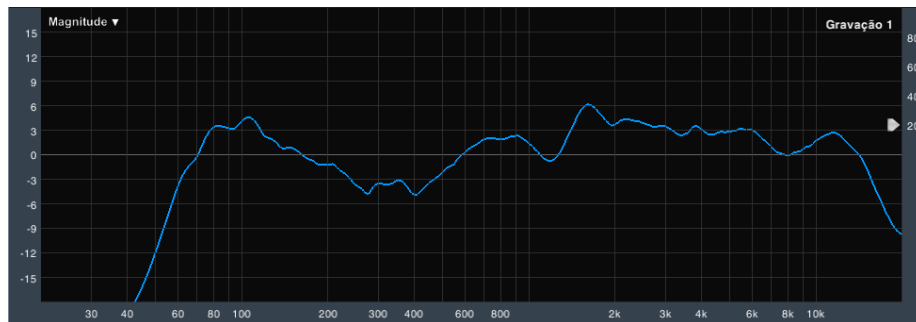


Figura 4 – Resposta do sistema no cenário 2.

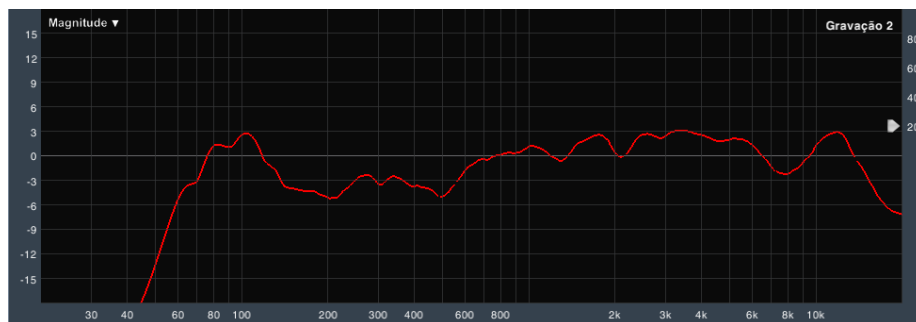


Figura 5 – Resposta do sistema no cenário 3.

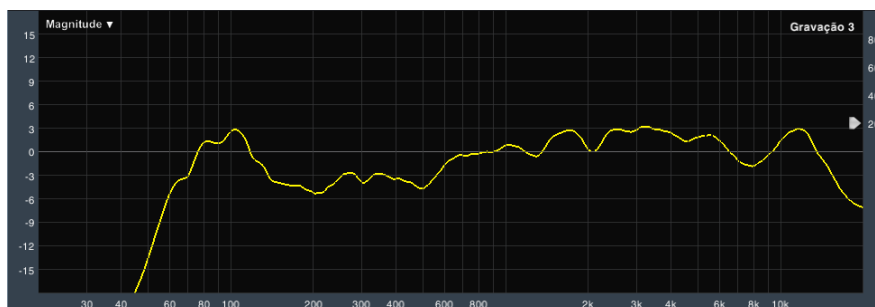


Figura 6 – Comparação da resposta do sistema nos cenários 1, 2 e 3.

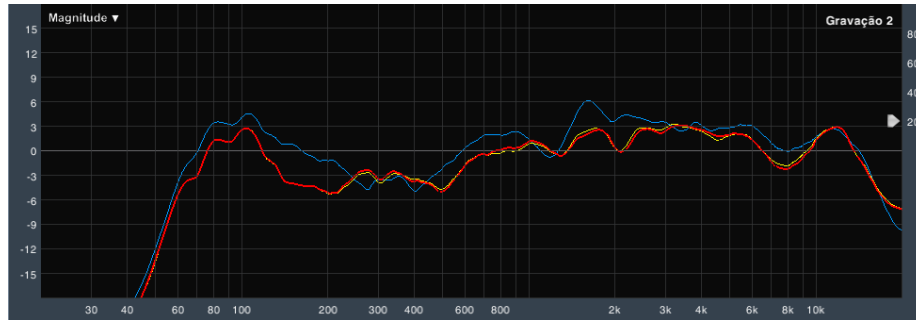


Figura 7 – Resposta do sistema no cenário 4.

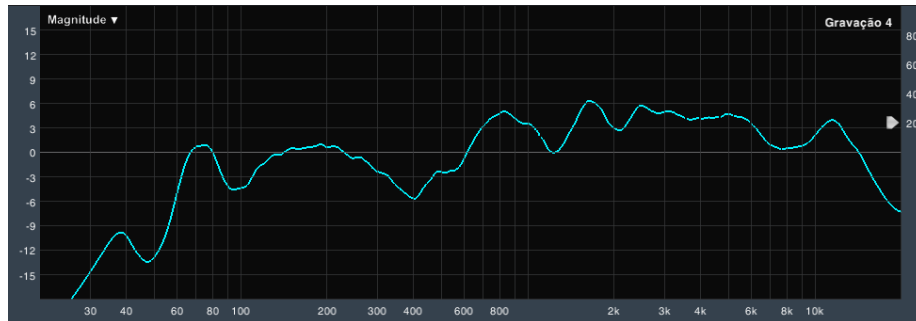


Figura 8 – Resposta do sistema no cenário 5.

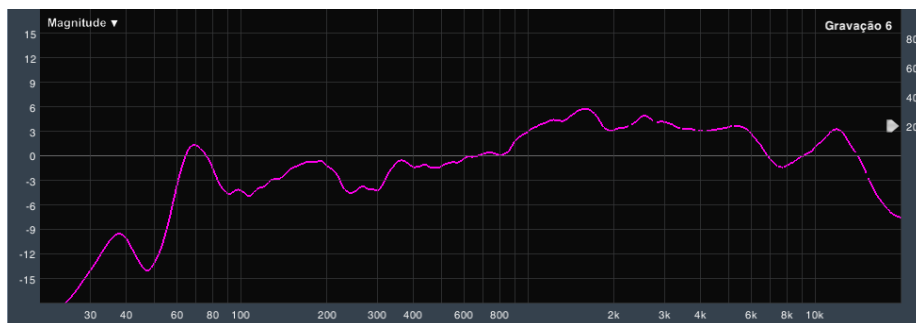


Figura 9 – Resposta do sistema no cenário 6.

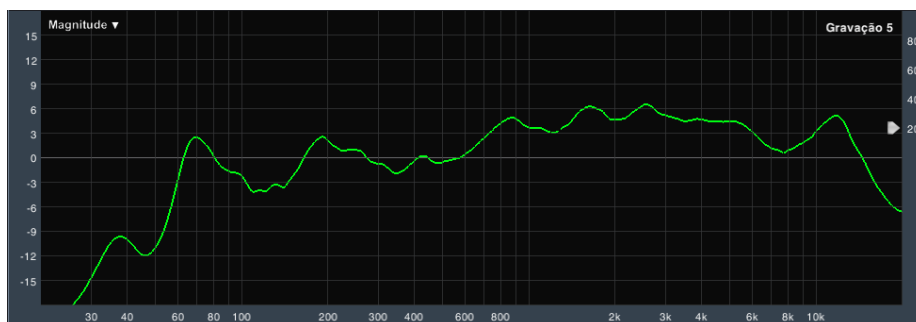
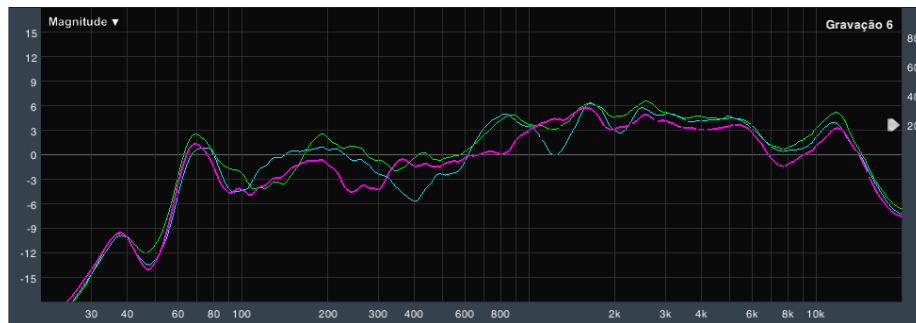


Figura 10 – Comparação da resposta do sistema nos cenários 4, 5 e 6.



APÊNDICE B – principal (*main*) do protótipo do processador

```
/**
*****
* File Name      : main.c
* Description    : Main program body
*****
*
* COPYRIGHT(c) 2017 STMicroelectronics
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
* 3. Neither the name of STMicroelectronics nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*****
*/
/* Includes -----*/
#include "main.h"
#include "stm32f0xx_hal.h"
#include "def_m0.h"
#include "lcd16.h"
#include "FUN_LCD.h"
#include "PORTAS.h"
#include "FUNC_DSP.h"
#include "FLASH_WR.h"
#include "math.h"
/* Private variables -----*/
SPI_HandleTypeDef hspi1;

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);
void Error_Handler(void);
/* USER CODE BEGIN PFP */
/* Private function prototypes -----*/
uint8_t PINK_VAR=0, PINK_GAIN=0, MUTE_PASS=0, POP_AV=0, rec_CH=0;
volatile uint8_t ROTA=1, ROTB=1;
//uint8_t buffer_UART[64], Rx_Data[2];
uint8_t AOK_1=0, AOK_2=0, AOK_3=0;
uint16_t TX_BYTES=0, TXByteCtr=0, DOISBYTES=0;
uint8_t ESCOLHA_CANAL=0, ESCOLHA_PARAM=0;
uint8_t GOUT_PARAM=1;
uint8_t GAIN_MASTER_PARAM=24;
uint8_t INVERT_MASTER_PARAM=0;

uint16_t DELAY_MASTER_PARAM=0;

uint16_t FREQ_CROSS_MASTER=0;

uint8_t ATEN_MASTER=0;

uint8_t UART_BIT=0, SEND_CMD=0;
```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
uint16_t SETA_CROSS=0,FREQIN_MASTER=0,GEQIN_MASTER=0;
volatile float Q_EQIN_MASTER=0.0, QUALY_MASTER=0.0;
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/
uint8_t EQIN_SETA=0;

uint16_t RMSTC_MASTER=0,TRESH_MASTER=0,DECAY_MASTER=0,SETA_LIM=0;

uint8_t G_PARAMETRIC_MASTER_PARAM=12;

uint8_t QUALITY_MASTER_PARAM=20;

uint16_t Sub_Address[2]={0,0};
uint16_t SAFE_BYTE[1]={0};
uint8_t Address_dsp[4]={0,0,0,0};

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

uint16_t VALZ=0,ADDRCtr=0;
uint8_t liga=0,FLASH_GRAV=0;
uint8_t CONT_EQ=0, INICIALIZACAO=0;
uint16_t
FREQ_PADRAO[28]={31,40,50,63,80,100,125,160,200,250,315,400,500,630,800,1000,1250,1600,2000,25
00,3150,4000,5000,6300,8000,10000,12500,16000};

uint8_t GAIN_EQ[28][PRST];

uint8_t ATEN_PARAM_HP[6][PRST];
uint8_t ATEN_PARAM_LP[6][PRST];
uint16_t FREQ_CROSS_HP[6][PRST];
uint16_t FREQ_CROSS_LP[6][PRST];

uint16_t FREQ_EQ_CH[PRST][18];
uint8_t GAIN_EQ_CH[PRST][18];
uint8_t QUAL_EQ_CH[PRST][18];

uint8_t MUTE_PARAM[6][PRST];
uint8_t GAIN_PARAM[6][PRST];
uint8_t OUT_PARAM[6][PRST];
uint8_t PHASE_PARAM[6][PRST];
uint16_t DELAY_PARAM[6][PRST];

uint16_t RMSTC_PARAM[6][PRST];
uint16_t DECAY_PARAM[6][PRST];
uint16_t TRESH_PARAM[6][PRST];

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//CROSS
uint8_t BYTE_PARAM=0;
uint16_t FREQ_HP=0,FREQ_LP=0,ADR_HI=0,ADR_LOW=0;

//CROSS
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//EQ
uint8_t VAL_EQ=0, F_CH=0;

volatile float QUAL_MASTER=0.0;
//EQ
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//SENHA
uint8_t SENHA_GRAVADA=0, SENHA_UM=0, SENHA_DOIS=0, SENHA_TRES=0, SENHA_QUATRO=0,INLR_GAIN=0;
uint8_t CONT_FUNC=8;
uint16_t RST_INI=0;

uint16_t distfont=0;

```

```
uint16_t distreta=0;
uint8_t temperatura=20, PORCENTAGEM_direc=00;

////////////////////////////////////

void LOAD_PROGRAM(void);
void AUX(void);
void ATUAL_PRESET(void);
void CANAIS(void);
void DELAY_TODOS(void);
void GANHO_TODOS(void);
void FASE_TODOS(void);
void ROTA_BOT(void);
void ROTA_ROTB(void);
void MULT_CHAN(void);
void RECARREGAR_SOFTWARE(void);
void CARREGAR_PARAM(void);
void DIRECIONAMENTO_porcentagem(void);
void DIRECIONAMENTO_temp(void);
void DISTANCIAS_direcionamento(void);
void DISTANCIAS_fonte(void);

void RESET_FUNC(void) {
    RST_INI=0;
    ADDRctr=0;DOISBYTES=0;

    LOAD_PROGRAM();

    //MEMORIA_PADRAO();

    //CARREGAR_PARAM();

    LOAD_PROGRAM_DEPOIS(); //*****arrumar

    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_13,GPIO_PIN_SET); //REM OUT

    PINK_VAR=0;
    PINK_GAIN=0;

    ROTA=1;
    ROTB=1;
    INICIALIZACAO=1;
    AUX();
}

int main(void)
{
    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* Configure the system clock */
    SystemClock_Config();

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_SPI1_Init();

    PORT_CONFIG();
    _delay_cycles(70000);
    RESET_ALL();

    INICIA_LCD();

    SPI_ATIVAR();
}
```

```

clr_lcm();

ADDRCtr=0;DOISBYTES=0;

LOAD_PROGRAM();

//MEMORIA_PADRAO();

//CARREGAR_PARAM();

LOAD_PROGRAM_DEPOIS();//*****arrumar

HAL_GPIO_WritePin(GPIOB,GPIO_PIN_13,GPIO_PIN_SET);//REM OUT

//*****daqui-----
// uint16_t n=0;
ROTA=1;
ROTB=1;
CONT_FUNC=0;
INICIALIZACAO=1;
RST_INI=0;
AUX();
}
void AUX(void){
uint16_t n=0;

EXP_E:RST_INI=0;
clr_lcm();
putstr("    EXPERT");
lcm_line2();
putstr("    ELECTRONICS    ");

PONTO_PRINCP:

while((ROTB==1) && (ROTA==1) && BOT && (CHAN6) && (CHAN5) && (CHAN4) && (CHAN3) && (CHAN2) && (CHAN1) && (ESC) && (PIN_REM));

//if((FLASH_GRAV==1) && (PIN_REM)){ROTB=1;ROTA=1;goto PONTO_PRINCP;}
if(ROTB==0){ROTB=1;CONT_FUNC++;if(CONT_FUNC>4){CONT_FUNC=0;}ROTA_BOT();}
if(ROTA==0){ROTA=1;if(CONT_FUNC==0){CONT_FUNC=4;}else{CONT_FUNC--;}ROTA_BOT();}
if(!BOT){while(!BOT){}ROTA_BOT();}

if(!(ESC)){while(!(ESC)){RST_INI++;__delay_cycles(400);if(RST_INI>55500){clr_lcm();__delay_cycles(28000);putstr("RESET INICIADO");__delay_cycles(700000);liga=0xFF;FLASH_WRITE_INI();RST_INI=0;__delay_cycles(70000);RESET_FUNC();}RST_INI=0;goto EXP_E;}//ESC

if(!(PIN_REM)){if(FLASH_GRAV==1){FLASH_GRAV=0;FLASH_WRITE();__delay_cycles(14000);}HAL_GPIO_WritePin(GPIOB,GPIO_PIN_13,GPIO_PIN_RESET);clr_lcm();putstr("    REM OFF");}
for(n=0;n<700;n++){__delay_cycles(77000);}
HAL_GPIO_WritePin(GPIOB,GPIO_PIN_15,GPIO_PIN_RESET);__delay_cycles(970000);main();}

goto PONTO_PRINCP;
}

void ROTA_BOT(void){

PONTO_ROTA:

if(BOT){

```

```

if (CONT_FUNC==0) {clr_lcm();putstr("   ROTEAMENTO");lcm_line2();putstr("   DE CANAIS");}
if (CONT_FUNC==1) {clr_lcm();putstr("PARAMETROS DE"); lcm_line2();putstr("DIRECIONAMENTO");}
if (CONT_FUNC==2) {clr_lcm();putstr("   GANHO");}
if (CONT_FUNC==3) {clr_lcm();putstr("   FASE");}
if (CONT_FUNC==4) {clr_lcm();putstr("   ATRASO");}

}

while ((ROTB==1) && (ROTA==1) && BOT && (CHAN6) && (CHAN5) && (CHAN4) && (CHAN3) && (CHAN2) && (CHAN1) && (ESC) && (PIN_REM));

if (ROTB==0) {ROTB=1;CONT_FUNC++;if (CONT_FUNC>4) {CONT_FUNC=0;}}
if (ROTA==0) {ROTA=1;if (CONT_FUNC==0) {CONT_FUNC=4;}else {CONT_FUNC--;}}

__delay_cycles(7000);
if (!BOT) {
if (CONT_FUNC==0) {while (!BOT) {} CANAIS ();}
if (CONT_FUNC==1) {while (!BOT) {} DIRECIONAMENTO_temp ();}
if (CONT_FUNC==2) {while (!BOT) {} GANHO_TODOS ();}
if (CONT_FUNC==3) {while (!BOT) {} FASE_TODOS ();}
if (CONT_FUNC==4) {while (!BOT) {} DELAY_TODOS ();}

}

if (!(!(CHAN6) || !(CHAN5) || !(CHAN4) || !(CHAN3) || !(CHAN2) || !(CHAN1))) {

MULT_CHAN ();

if (CONT_FUNC==0) {while (!BOT) {} CANAIS ();}
if (CONT_FUNC==1) {while (!BOT) {} DIRECIONAMENTO_temp ();}
if (CONT_FUNC==2) {while (!BOT) {} GANHO_TODOS ();}
if (CONT_FUNC==3) {while (!BOT) {} FASE_TODOS ();}
if (CONT_FUNC==4) {while (!BOT) {} DELAY_TODOS ();}

}

if (!(ESC)) {__delay_cycles(500000);FLASH_WRITE();AUX();} //ESC
if (!(PIN_REM)) {__delay_cycles(500000);FLASH_WRITE();AUX();}

goto PONTO_ROTA;

}

void MULT_CHAN(void) {
if (!(CHAN6)) {ESCOLHA_CANAL=5;while (!(CHAN6));}
if (!(CHAN5)) {ESCOLHA_CANAL=4;while (!(CHAN5));}
if (!(CHAN4)) {ESCOLHA_CANAL=3;while (!(CHAN4));}
if (!(CHAN3)) {ESCOLHA_CANAL=2;while (!(CHAN3));}
if (!(CHAN2)) {ESCOLHA_CANAL=1;while (!(CHAN2));}
if (!(CHAN1)) {ESCOLHA_CANAL=0;while (!(CHAN1));}
}
//////////-----SENHA
SIMPLES-----////////////////////////////////////
///
////////////////////////////////////-----
float  veloc_som=0.0,razao=0.0,dif_lados=0.0,PORCENTAGEM_direc_Dois=0.0,razao_centro=0.0;
float  ladoA=0.0,ladoB=0.0,ladoC=0.0,ladoD=0.0,lado1=0.0,lado2=0.0,lado3=0.0,lado4=0.0;
float  distfontmetros=0.0,distretametros=0.0,distfontmetroscentro=0.0,dif_lados_centro=0.0;
float  angulo_Um=0.0,angulo_Dois=0.0,amostras_atrasadas=0.0,amostras_atrasadas_centro=0.0;
void set_func_direcionamento(void) {
uint8_t var_aux=0;
ini_func_dir:
voltacanal:
clr_lcm();
putstr("Selecione o lado:");
lcm_line2();
putstr("lado ->:");
MOSTRA_CANAL();
}

```

```

while ((ROTB==1) && (ROTA==1) && BOT && (CHAN6) && (CHAN5) && (CHAN4) && (CHAN3) && (CHAN2) && (CHAN1) && (ESC)
&& (PIN_REM));
if (!BOT) {while (!BOT); var_aux=1;}
if (!ESC) {while (!ESC); __delay_cycles(900000); AUX();}

if ((!(CHAN6)) || (!(CHAN5)) || (!(CHAN4)) || (!(CHAN3)) || (!(CHAN2)) || (!(CHAN1))) {MULT_CHAN(); goto
o_voltacanal;}
if (ESCOLHA_CANAL==0 || ESCOLHA_CANAL==3) {

distfontmetros=distfont/100.0;
distretametros=distreta/100.0;
veloc_som=sqrt((temperatura+273.15)/(273.15))*331.45;
razao=(distfontmetros/distretametros);
dif_lados=2*(PORCENTAGEM_direc/200.0)*distfontmetros;

ladoA=(distfontmetros/2.0)+(dif_lados/2.0);
ladoB=(distfontmetros/2.0)-(dif_lados/2.0);

lado1=sqrt(pow(distfontmetros,2)-(2*distfontmetros*ladoB)+(pow(ladoB,2)+pow(distretametros
,2)));
lado2=sqrt(pow(distfontmetros,2)-(2*distfontmetros*ladoA)+(pow(ladoA,2)+pow(distretametros
,2)));
amostras_atrasadas=((lado1-lado2)/(veloc_som*0.00002083));

if (razao==1) {angulo_Um=(razao*0.45*PORCENTAGEM_direc);}
if (razao>0.75) {angulo_Um=(razao*0.45*(pow(1.111,(1-(1-razao))))*PORCENTAGEM_direc);}
if (razao<0.75) {angulo_Um=(razao*0.45*(pow(1.111,(1+(1-razao))))*PORCENTAGEM_direc);}
if (razao==0.75) {angulo_Um=(razao*0.45*1.111*PORCENTAGEM_direc);}

PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]=angulo_Um;
DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ]=amostras_atrasadas;
DELAY_MASTER_PARAM=DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ];
MUDANCA_DLY_MEDIDA();
FASE_CANAL();

if (ESCOLHA_CANAL==0) {
PHASE_PARAM[3][CONT_EQ]=0;
DELAY_PARAM[3][CONT_EQ]=0;
DELAY_MASTER_PARAM=DELAY_PARAM[3][CONT_EQ];
MUDANCA_DLY_MEDIDA();
FASE_CANAL();
}

if (ESCOLHA_CANAL==3) {
PHASE_PARAM[0][CONT_EQ]=0;
DELAY_PARAM[0][CONT_EQ]=0;
DELAY_MASTER_PARAM=DELAY_PARAM[0][CONT_EQ];
MUDANCA_DLY_MEDIDA();
FASE_CANAL();
}

}

ESCOLHA_CANAL=1;

PORCENTAGEM_direc_Dois=(PORCENTAGEM_direc-50)*2;
distfontmetroscentro=distfont/200.0;
distretametros=distreta/100.0;
veloc_som=sqrt((temperatura+273.15)/(273.15))*331.45;
razao_centro=(distfontmetroscentro/distretametros);
dif_lados_centro=2*(PORCENTAGEM_direc_Dois/200.0)*distfontmetroscentro;

ladoA=(distfontmetros/2.0)+(dif_lados/2.0);
ladoB=(distfontmetros/2.0)-(dif_lados/2.0);

ladoC=(distfontmetroscentro/2.0)+(dif_lados_centro/2.0);
ladoD=(distfontmetroscentro/2.0)-(dif_lados_centro/2.0);

```

```

lado1=sqrt(pow(distfontmetros,2)-(2*distfontmetros*ladoB)+(pow(ladoB,2)+pow(distretametros,2)));
lado2=sqrt(pow(distfontmetros,2)-(2*distfontmetros*ladoA)+(pow(ladoA,2)+pow(distretametros,2)));

lado3=sqrt(pow(distfontmetroscentro,2)-(2*distfontmetroscentro*ladoD)+(pow(ladoD,2)+pow(distretametros,2)));
lado4=sqrt(pow(distfontmetroscentro,2)-(2*distfontmetroscentro*ladoC)+(pow(ladoC,2)+pow(distretametros,2)));

amostras_atrasadas_centro=((lado3-lado4)/(veloc_som*0.00002083))*(-1.0)+amostras_atrasadas;

if(razao_centro==1){angulo_Dois=(razao_centro*0.45*PORCENTAGEM_direc_Dois);}
if(razao_centro>0.75){angulo_Dois=(razao_centro*0.45*(pow(1.111,(1-(1-razao_centro))))*PORCENTAGEM_direc_Dois);}
if(razao_centro<0.75){angulo_Dois=(razao_centro*0.45*(pow(1.111,(1+(1-razao_centro))))*PORCENTAGEM_direc_Dois);}
if(razao_centro==0.75){angulo_Dois=(razao_centro*0.45*1.111*PORCENTAGEM_direc_Dois);}

angulo_Dois=angulo_Um-angulo_Dois;

PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]=angulo_Dois;
DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ]=amostras_atrasadas_centro;
DELAY_MASTER_PARAM=DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ];
MUDANCA_DLY_MEDIDA();
FASE_CANAL();
}

ESCOLHA_CANAL=0;
if(var_aux==1){AUX();}

}

void DISTANCIAS_direcionamento(void){
    if(distreta<distfont){distreta=distfont;}
    clr_lcm();
    putstr("Direcionamento");
    dirc_ret:
    lcm_line2();
    putstr("dist->");
    VALZ=distreta;
    PART_DELAY_B();
    putstr(" cm");
    while((ROTB==1)&&(ROTA==1)&&BOT&&ESC&&(PIN_REM));

    if(ROTB==0){ROTB=1;if(distreta<9998){distreta++;} goto dirc_ret;}
    if(ROTA==0){ROTA=1;if(distreta>distfont){distreta--;} goto dirc_ret;}

    if(!ESC){while(!BOT);__delay_cycles(2000000);DISTANCIAS_fonte();}
    if(!BOT){while(!BOT);__delay_cycles(2000000);set_func_direcionamento();}
}

```

```

void DISTANCIAS_fonte(void) {

    clr_lcm();
    putstr("Dist das Fontes");
    dirc_font:
    lcm_line2();
    putstr("->:");

    VALZ=distfont;
    PART_DELAY_B();
    putstr(" cm");
    while((ROTB==1) && (ROTA==1) && BOT && ESC && (PIN_REM));

    if(ROTB==0) {ROTB=1; if(distfont<9998) {distfont++;} goto dirc_font;}
    if(ROTA==0) {ROTA=1; if(distfont>0) {distfont--;} goto dirc_font;}

    //distreta=distfont;
    if(!ESC) {while(!BOT); __delay_cycles(2000000); DIRECIONAMENTO_porcentagem();}
    if(!BOT) {while(!BOT); __delay_cycles(2000000); DISTANCIAS_direcionamento();}
}

void DIRECIONAMENTO_temp(void) {

    dirc_temp:
    clr_lcm();
    putstr("Temp(C) ->:");
    VALZ=temperatura;
    PART_GAIN();
    lcm_line2();
    putstr("Direc:");
    VALZ=PORCENTAGEM_direc;
    PART_GAIN();
    putstr(" %");

    while((ROTB==1) && (ROTA==1) && BOT && ESC && (PIN_REM));

    if(ROTB==0) {ROTB=1; if(temperatura<50) {temperatura++;} goto dirc_temp;}
    if(ROTA==0) {ROTA=1; if(temperatura>0) {temperatura--;} goto dirc_temp;}

    if(!ESC) {while(!ESC); __delay_cycles(2000000); AUX();}
    if(!BOT) {while(!BOT); __delay_cycles(2000000); DIRECIONAMENTO_porcentagem();}
}

void DIRECIONAMENTO_porcentagem(void) {

    clr_lcm();
    putstr("Temp(C) :");
    VALZ=temperatura;
    PART_GAIN();
    dirc_porc:
    lcm_line2();
    putstr("Direc ->:");
    VALZ=PORCENTAGEM_direc;
    PART_DELAY();
    putstr(" %");
    while((ROTB==1) && (ROTA==1) && BOT && ESC && (PIN_REM));

    if(ROTB==0) {ROTB=1; if(PORCENTAGEM_direc<100) {PORCENTAGEM_direc++;} goto dirc_porc;}
    if(ROTA==0) {ROTA=1; if(PORCENTAGEM_direc>0) {PORCENTAGEM_direc--;} goto dirc_porc;}

    if(!ESC) {while(!BOT); __delay_cycles(2000000); DIRECIONAMENTO_temp();}
    if(!BOT) {while(!BOT); __delay_cycles(2000000); DISTANCIAS_fonte();}
}

```

```

}

//////////////////////////////////-----ESCOLHA DE
CANAIS-----//////////////////////////////////
//
//////////////////////////////////-----
-----

void CANAIS(void){
    uint8_t RAPID_ROT=0;
    PONTO_CH:
    clr_lcm();
    if(ESCOLHA_CANAL==0){putstr("    CH1>");}
    if(ESCOLHA_CANAL==1){putstr("    CH2>");}
    if(ESCOLHA_CANAL==2){putstr("    CH3>");}
    if(ESCOLHA_CANAL==3){putstr("    CH4>");}
    if(ESCOLHA_CANAL==4){putstr("    CH5>");}
    if(ESCOLHA_CANAL==5){putstr("    CH6>");}
    lcm_line2();
    if(OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]==0){putstr("    OUT: R");if(RAPID_ROT==1){RAPID_ROT=0;
        OUT_AUDIO();}}
    if(OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]==1){putstr("    OUT: L");if(RAPID_ROT==1){RAPID_ROT=0;
        OUT_AUDIO();}}
    if(OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]==2){putstr("    OUT: L +
R");if(RAPID_ROT==1){RAPID_ROT=0;
        OUT_AUDIO();}}

    while((ROTB==1)&&(ROTA==1)&&(CHAN6)&&(CHAN5)&&(CHAN4)&&(CHAN3)&&(CHAN2)&&(CHAN1)&&(ESC)&&(PIN_REM));

    if(ROTA==0){ROTA=1;if(OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]>0){OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]
--;}/*OFF_CH();*/RAPID_ROT=1;}

    if(ROTB==0){ROTB=1;if(OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]<2){OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]
++;}/*OFF_CH();*/RAPID_ROT=1;}

    if(!(CHAN6)||!(CHAN5)||!(CHAN4)||!(CHAN3)||!(CHAN2)||!(CHAN1)){MULT_CHAN();}
    if(!(ESC)){_delay_cycles(50000);FLASH_WRITE();ROTA_BOT();}
    if(!(PIN_REM)){_delay_cycles(50000);FLASH_WRITE();AUX();}

    goto PONTO_CH;
}

//////////////////////////////////-----GANHO FASE MUTE-----//////////////////////////////////
//////////////////////////////////
//////////////////////////////////-----

void FASE_TODOS(void){

    clr_lcm();

    PONTO_FASE_1:lcm_line1();
    MOSTRA_CANAL();
    putstr(" PHASE: ");
    VALZ = PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ];
    PART_DELAY();
    //putstr("");
    //if(PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]==0){FASE_CANAL();putstr(" PHASE: 0    ");}

```

```

//if (PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]==1) {INV_CANAL();putstr(" PHASE: 180");}

while ((ROTB==1) && (ROTA==1) && BOT && (CHAN6) && (CHAN5) && (CHAN4) && (CHAN3) && (CHAN2) && (CHAN1) && (
ESC) && (PIN_REM));

if (ROTB==0) {ROTB=1; if (PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]<255) {PHASE_PARAM[ESCOLHA_CANAL]
[CONT_EQ]++;FASE_CANAL();}}

if (ROTA==0) {ROTA=1; if (PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]>0) {PHASE_PARAM[ESCOLHA_CANAL][
CONT_EQ]--;FASE_CANAL();}}

if ((!(CHAN6)) || (!(CHAN5)) || (!(CHAN4)) || (!(CHAN3)) || (!(CHAN2)) || (!(CHAN1))) {MULT_CHAN();}
if (!(ESC)) {__delay_cycles(5000);FLASH_WRITE();ROTA_BOT();}
if (!(PIN_REM)) {__delay_cycles(5000);FLASH_WRITE();AUX();}

goto PONTO_FASE_1;
}

void GANHO_TODOS(void) {
__delay_cycles(9000);
clr_lcm();
PONTO_GANHO_1:lcm_line1();
MOSTRA_CANAL();
VALZ = GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ];
if (VALZ<36) {VALZ = 36 - GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ];putstr(" GAIN:-");}
else {VALZ = GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ] - 36; putstr(" GAIN: ");}
PART_GAIN();
putstr(" dB");

while ((ROTB==1) && (ROTA==1) && BOT && (CHAN6) && (CHAN5) && (CHAN4) && (CHAN3) && (CHAN2) && (CHAN1) && (
ESC) && (PIN_REM));

if (ROTB==0) {ROTB=1; if (GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ]<42) {GAIN_PARAM[ESCOLHA_CANAL][CO
NT_EQ]++;VALZ = 42 - GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ];GANHO_CANAL();}}

if (ROTA==0) {ROTA=1; if (GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ]>0) {GAIN_PARAM[ESCOLHA_CANAL][CO
NT_EQ]--;VALZ = 42 - GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ];GANHO_CANAL();}}

if ((!(CHAN6)) || (!(CHAN5)) || (!(CHAN4)) || (!(CHAN3)) || (!(CHAN2)) || (!(CHAN1))) {MULT_CHAN();}

if (!(ESC)) {__delay_cycles(500000);FLASH_WRITE();ROTA_BOT();}
if (!(PIN_REM)) {__delay_cycles(500000);FLASH_WRITE();AUX();}

goto PONTO_GANHO_1;
}

//////////-----LIMITER_RMSTC_DECAY_TRESH-----//////////
//////////-----LIMITER_RMSTC_DECAY_TRESH-----//////////
//////////-----LIMITER_RMSTC_DECAY_TRESH-----//////////
-----

void DELAY_TODOS(void) {
uint16_t i=65100;
//PONTO_DELAY_1:
clr_lcm();
PONTO_DELAY_1:
lcm_line1();
MOSTRA_CANAL();

```

```

   _putstr(" DLY:");
    VALZ=DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ];
    PART_DELAY();
   _putstr("smp");

    while((ROTB==1) && (ROTA==1) && BOT && (CHAN6) && (CHAN5) && (CHAN4) && (CHAN3) && (CHAN2) && (CHAN1) && (
ESC) && (PIN_REM)) {i++; if(i>65000) {i=65100;}}
if(i==55000) {DELAY_MASTER_PARAM=DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ];MUDANCA_DLY_MEDIDA()
;}}

if(ROTB==0) {ROTB=1; if(DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ]<399) {DELAY_PARAM[ESCOLHA_CANAL
][CONT_EQ]++;}
i=0;
}

if(ROTA==0) {ROTA=1; if(DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ]>0) {DELAY_PARAM[ESCOLHA_CANAL][
CONT_EQ]--;}
i=0;
}

if(!(CHAN6) || !(CHAN5) || !(CHAN4) || !(CHAN3) || !(CHAN2) || !(CHAN1)) {MULT_CHAN();}

if(!(ESC)) {__delay_cycles(500000);FLASH_WRITE();ROTA_BOT();}
if(!(PIN_REM)) {__delay_cycles(500000);FLASH_WRITE();AUX();}

goto PONTO_DELAY_1;
}

void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct;
    RCC_ClkInitTypeDef RCC_ClkInitStruct;
    RCC_PeriphCLKInitTypeDef PeriphClkInit;

    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSISState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = 16;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL6;
    RCC_OscInitStruct.PLL.PREDIV = RCC_PREDIV_DIV1;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
        |RCC_CLOCKTYPE_PCLK1;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
    {
        Error_Handler();
    }

    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART1;
    PeriphClkInit.Usart1ClockSelection = RCC_USART1CLKSOURCE_PCLK1;
    if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
    {
        Error_Handler();
    }

    HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

```

```
HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

/* SysTick_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

/* SPI1 init function */
void MX_SPI1_Init(void)
{
    hspi1.Instance = SPI1;
    hspi1.Init.Mode = SPI_MODE_MASTER;
    hspi1.Init.Direction = SPI_DIRECTION_2LINES;
    hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
    hspi1.Init.CLKPolarity = SPI_POLARITY_HIGH;
    hspi1.Init.CLKPhase = SPI_PHASE_2EDGE;
    hspi1.Init.NSS = SPI_NSS_SOFT;
    hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_8;
    hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
    hspi1.Init.TIMode = SPI_TIMODE_DISABLED;
    hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLED;
    hspi1.Init.CRCPolynomial = 10;
    hspi1.Init.CRCLength = SPI_CRC_LENGTH_DATASIZE;
    hspi1.Init.NSSPMode = SPI_NSS_PULSE_DISABLED;
    HAL_SPI_Init(&hspi1);
}

/** Configure pins as
    * Analog
    * Input
    * Output
    * EVENT_OUT
    * EXTI
*/
void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct;

    /* GPIO Ports Clock Enable */
    __GPIOC_CLK_ENABLE();
    __GPIOA_CLK_ENABLE();
    __GPIOB_CLK_ENABLE();

    /*Configure GPIO pins : PC0 PC1 PC2 PC3 PC6 PC7 PC8 PC10 */
    GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_6|GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_10;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_PULLUP;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    /*Configure GPIO pins : PC9*/
    GPIO_InitStruct.Pin = GPIO_PIN_9;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    /*Configure GPIO pins : PC4 PC5 */
    GPIO_InitStruct.Pin = GPIO_PIN_4|GPIO_PIN_5;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    /*Configure GPIO pins : PA11 PA12*/
    GPIO_InitStruct.Pin = GPIO_PIN_11|GPIO_PIN_12;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
}
```

```

/*Configure GPIO pins : PA0 PA1 PA8 */
GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_8;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_PULLUP;
GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pins : PB0 PB1 PB2 PB3 PB4 PB5 PB6 */
GPIO_InitStruct.Pin =
GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pins : PB13 PB15 */
GPIO_InitStruct.Pin = GPIO_PIN_13|GPIO_PIN_15;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_PULLUP;
GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pin : PB14 PB7 PB8 PB9*/
GPIO_InitStruct.Pin = GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_14;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pins : PC11 PC12 */
GPIO_InitStruct.Pin = GPIO_PIN_11|GPIO_PIN_12;
GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
GPIO_InitStruct.Pull = GPIO_PULLUP;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

/* EXTI interrupt init*/
HAL_NVIC_SetPriority(EXTI4_15_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(EXTI4_15_IRQn);
}

/* USER CODE BEGIN 4 */
/* USER CODE END 4 */

#ifdef USE_FULL_ASSERT

/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t* file, uint32_t line)
{
/* USER CODE BEGIN 6 */
/* User can add his own implementation to report the file name and line number,
ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
/* USER CODE END 6 */
}

#endif

/**
 * @brief This function is executed in case of error occurrence.
 * @param None
 * @retval None
 */
void Error_Handler(void)

```

```
{
/* USER CODE BEGIN Error_Handler */
/* User can add his own implementation to report the HAL error return state */

    RESET_FUNC();

/* USER CODE END Error_Handler */
}
/**
 * @}
 */

/**
 * @}
 */

/***** (C) COPYRIGHT STMicroelectronics *****/

void CARREGAR_PARAM(void) {

uint8_t i=0;

    ESCOLHA_CANAL=0;OUT_AUDIO();
    ESCOLHA_CANAL=1;OUT_AUDIO();

    ESCOLHA_CANAL=3;OUT_AUDIO();
    ESCOLHA_CANAL=4;OUT_AUDIO();

    ESCOLHA_CANAL=0;DELAY_MASTER_PARAM=DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ];MUDANCA_DLY_MEDIDA(
);
    ESCOLHA_CANAL=1;DELAY_MASTER_PARAM=DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ];MUDANCA_DLY_MEDIDA(
);

    ESCOLHA_CANAL=3;DELAY_MASTER_PARAM=DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ];MUDANCA_DLY_MEDIDA(
);
    ESCOLHA_CANAL=4;DELAY_MASTER_PARAM=DELAY_PARAM[ESCOLHA_CANAL][CONT_EQ];MUDANCA_DLY_MEDIDA(
);

    ESCOLHA_CANAL=0;if(PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]==0){FASE_CANAL();}
    __delay_cycles(20000);

    ESCOLHA_CANAL=1;if(PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]==0){FASE_CANAL();}
    __delay_cycles(20000);

    ESCOLHA_CANAL=3;if(PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]==0){FASE_CANAL();}
    __delay_cycles(20000);

    ESCOLHA_CANAL=4;if(PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ]==0){FASE_CANAL();}
    __delay_cycles(20000);

    ESCOLHA_CANAL=0;VALZ = 42 - GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ];GANHO_CANAL();
    ESCOLHA_CANAL=1;VALZ = 42 - GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ];GANHO_CANAL();

    ESCOLHA_CANAL=3;VALZ = 42 - GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ];GANHO_CANAL();
    ESCOLHA_CANAL=4;VALZ = 42 - GAIN_PARAM[ESCOLHA_CANAL][CONT_EQ];GANHO_CANAL();
```

```
EQIN_SETA=0;
ESCOLHA_CANAL=0;

}

void RECARREGAR_SOFTWARE(void) {
    rec_CH=ESCOLHA_CANAL;
    RESET_ALL();
    INICIALIZACAO=0;
    SPI_ATIVAR();
    ADDRctr=0;DOISBYTES=0;
    LOAD_PROGRAM();
    LOAD_PROGRAM_DEPOIS();//*****arrumar
    INICIALIZACAO=1;
    ROTA=1;ROTB=1;
    ESCOLHA_CANAL=rec_CH;
}
```

APÊNDICE C – Código do DSP do protótipo do processador

```
#include "FUNC_DSP.h"
#include <math.h>
#include <float.h>
#include "SPI.h"
#include "def_m0.h"
#include "stm32f0xx_hal.h"

#include "degree_IC_1.h"
#include "degree_IC_1_REG.h"

float Pi=3.14159265358979323846264338;
#define pid180 (Pi*1.0/180)
#define Fs 48000
float coeffi[5] = {0.0,0.0,0.0,0.0,0.0};
byte coebyte5[20] = {0x00};
byte param_hex[4]={0};
//.....
.....

extern uint8_t INICIALIZACAO, CONT_EQ;
extern uint8_t BYTE_PARAM;
extern uint16_t FREQ_HP,FREQ_LP,ADR_HI,ADR_LOW;
extern uint8_t Address_dsp[4];

uint8_t ADR_CROSS[2];

uint8_t *MSData;
extern uint16_t Sub_Address[2];
extern uint16_t SAFE_BYTE[1];

extern uint8_t MUTE_PARAM[6][PRST];
extern uint8_t GAIN_PARAM[6][PRST];
extern uint8_t OUT_PARAM[6][PRST];
extern uint8_t PHASE_PARAM[6][PRST];
//.....
.....

extern uint16_t TX_BYTES, ADDRctr, TXByteCtr,DOISBYTES;

extern uint8_t ESCOLHA_CANAL, ESCOLHA_PARAM;
extern uint8_t GOUT_PARAM;
extern uint8_t GAIN_MASTER_PARAM;
extern uint8_t INVERT_MASTER_PARAM;

extern uint16_t DELAY_MASTER_PARAM;

extern uint16_t FREQ_CROSS_MASTER;

extern uint8_t ATEN_MASTER;

////////////////////////////////////

extern float Q_EQIN_MASTER, QUALY_MASTER;
////////////////////////////////////
extern uint16_t FREQIN_MASTER,GEQIN_MASTER;
////////////////////////////////////
/
extern uint8_t EQIN_SETA;

extern uint16_t RMSTC_MASTER,TRESH_MASTER,DECAY_MASTER,SETA_LIM;

extern uint8_t QUALITY_MASTER_PARAM;
//.....
```

```

.....
uint16_t Fx1=0,Fx2=0,Fx3=0,Fx4=0,Fx5=0,NUM_VAL=0;

uint8_t m1=0,m2=0,m3=0,m4=0,m5=0,m6=0;

extern uint16_t VALZ;

extern float QUAL_MASTER;

extern uint8_t F_CH;
//*****OPTIMIZED_FILTER*****
//*****OUT_PINKY*****

void OUT_AUDIO(void) {
    uint16_t VALOUT=0;

    if(ESCOLHA_CANAL==0){VALOUT=CHOUT1; }
    if(ESCOLHA_CANAL==1){VALOUT=CHOUT2; }
    if(ESCOLHA_CANAL==2){VALOUT=CHOUT3; }
    if(ESCOLHA_CANAL==3){VALOUT=CHOUT1; }
    if(ESCOLHA_CANAL==4){VALOUT=CHOUT2; }
    if(ESCOLHA_CANAL==5){VALOUT=CHOUT3; }

    Sub_Address[0]=VALOUT;
    TXByteCtr=4; MSData=(unsigned char *)Param_Data+4; SPI_TX();
    Sub_Address[0]=VALOUT+1;
    TXByteCtr=4; MSData=(unsigned char *)Param_Data+4; SPI_TX();
    Sub_Address[0]=VALOUT+2;
    TXByteCtr=4; MSData=(unsigned char *)Param_Data+4; SPI_TX();

    if(OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]==1){Sub_Address[0]=VALOUT;TXByteCtr=4;
    MSData=(unsigned char *)SPDIF_TAB; SPI_TX();}
    if(OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]==0){Sub_Address[0]=VALOUT+1;TXByteCtr=4;
    MSData=(unsigned char *)SPDIF_TAB; SPI_TX();}
    if(OUT_PARAM[ESCOLHA_CANAL][CONT_EQ]==2){Sub_Address[0]=VALOUT+2;TXByteCtr=4;
    MSData=(unsigned char *)SPDIF_TAB; SPI_TX();}

}

float FASE_RAD=0.0;
void FASE_CANAL(void) {
    if(ESCOLHA_CANAL==0){Sub_Address[0]=FASE_CH1; }
    if(ESCOLHA_CANAL==1){Sub_Address[0]=FASE_CH2; }
    //if(ESCOLHA_CANAL==2){Sub_Address[0]=FASE_CH3; }

    if(ESCOLHA_CANAL==3){Sub_Address[0]=FASE_CH1; }
    if(ESCOLHA_CANAL==4){Sub_Address[0]=FASE_CH2; }
    //if(ESCOLHA_CANAL==5){Sub_Address[0]=FASE_CH3; }

    FASE_RAD=cos(((PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ])*0.01745329));
    To523(FASE_RAD, &coebyte5[0]);
    TXByteCtr=4; MSData=&coebyte5[0]; SPI_TX();//1
    FASE_RAD=sin(((PHASE_PARAM[ESCOLHA_CANAL][CONT_EQ])*0.01745329));
    To523(FASE_RAD, &coebyte5[0]);
    Sub_Address[0]=Sub_Address[0]+1;
    TXByteCtr=4; MSData=&coebyte5[0]; SPI_TX();//1

}
//.....
//.....GANHO.....
void GANHO_CANAL(void) {

```

```

if(ESCOLHA_CANAL==0){MSData=(unsigned char *) GANHO_CH1; }
if(ESCOLHA_CANAL==1){MSData=(unsigned char *) GANHO_CH2; }
//if(ESCOLHA_CANAL==2){MSData=(unsigned char *) GANHO_CH3; }
if(ESCOLHA_CANAL==3){MSData=(unsigned char *) GANHO_CH1; }
if(ESCOLHA_CANAL==4){MSData=(unsigned char *) GANHO_CH2; }
// if(ESCOLHA_CANAL==5){MSData=(unsigned char *) GANHO_CH3; }

// if(INICIALIZACAO==1){
Sub_Address[0]=0x0815;
TXByteCtr=2;
SPI_TX();

Sub_Address[0]=0x0810;
TXByteCtr=5;
if(MUTE_PARAM[ESCOLHA_CANAL][CONT_EQ]==1){VALZ=54;}
MSData=(unsigned char *)VOL_Data+5*VALZ;
SPI_TX();

__delay_cycles(200);

if(ESCOLHA_CANAL==0){MSData=(unsigned char *) TAU_GANHO_CH1; }
if(ESCOLHA_CANAL==1){MSData=(unsigned char *) TAU_GANHO_CH2; }
if(ESCOLHA_CANAL==2){MSData=(unsigned char *) TAU_GANHO_CH3; }
if(ESCOLHA_CANAL==3){MSData=(unsigned char *) TAU_GANHO_CH1; }
if(ESCOLHA_CANAL==4){MSData=(unsigned char *) TAU_GANHO_CH2; }
if(ESCOLHA_CANAL==5){MSData=(unsigned char *) TAU_GANHO_CH3; }

Sub_Address[0]=0x0816;
TXByteCtr=2;
SPI_TX();

Sub_Address[0]=0x0811;
TXByteCtr=5;
MSData=(unsigned char *)TAU_TAB;
SPI_TX();

Sub_Address[0]=0x081C;
TXByteCtr=2;
if(INICIALIZACAO==1){MSData=(unsigned char *)SAFELOAD_TAB;}
if(INICIALIZACAO==0){MSData=(unsigned char *)SAFELOAD_TAB+2;}
SPI_TX();

__delay_cycles(700);
}

//.....

//.....

//.....

void MUDANCA_DLY_MEDIDA(void){
uint16_t CT=0;

NUM_VAL=DELAY_MASTER_PARAM+1;
FUNCAO_PARTICIONAR();

CT=Fx3*100+Fx4*10+Fx5;
Address_dsp[2]=0;
Address_dsp[3]=Fx3*100+Fx4*10+Fx5;
if(CT>255){Address_dsp[2]=0x01;Address_dsp[3]=(Fx3*100+Fx4*10+Fx5)-256;}
Address_dsp[0]=0;Address_dsp[1]=0;

```

```

if(ESCOLHA_CANAL==0){Sub_Address[0] = DELAY_1;}
if(ESCOLHA_CANAL==1){Sub_Address[0] = DELAY_2;}
if(ESCOLHA_CANAL==2){Sub_Address[0] = DELAY_3;}
if(ESCOLHA_CANAL==3){Sub_Address[0] = DELAY_1;}
if(ESCOLHA_CANAL==4){Sub_Address[0] = DELAY_2;}
if(ESCOLHA_CANAL==5){Sub_Address[0] = DELAY_3;}

MSData=&Address_dsp[0];
TXByteCtr = 4;
SPI_TX();
__delay_cycles(700);
}

//.....

////////////////////////////////////
void To523(float param_dec, byte* param_hex)
{
    long param223;
    long param227;
    //multiply decimal number by 2^23
    param223 = param_dec * (1L << 23);
    //convert to positive binary
    param227 = param223 + (1L << 27);
    param_hex[3]=(byte)param227; //get byte 3 (LSBs) of parameter value
    param_hex[2]=(byte)(param227>>8); //get byte 2 of parameter value
    param_hex[1]=(byte)(param227>>16); //get byte 1 of parameter value
    //get byte 0 (MSBs) of parameter value
    param_hex[0]=(byte)(param227>>24);
    //invert sign bit to get correct sign
    param_hex[0] = param_hex[0] ^ 0x08;
    if(param_hex[0] & 0x08) { param_hex[0] |= 0xf0;}
}
//.....
//-----//
//.....
//.....

void LOAD_PROGRAM(void){
// uint8_t n=0;
ESCOLHA_CANAL=0;
//SIGMA_WRITE_REGISTER_BLOCK( DEVICE_ADDR_IC_1, REG_COREREGISTER_IC_1_ADDR,
REG_COREREGISTER_IC_1_BYTE, R0_COREREGISTER_IC_1_Default );
Sub_Address[0] = REG_COREREGISTER_IC_1_ADDR;

MSData = (uint8_t *)R0_COREREGISTER_IC_1_Default;
TXByteCtr = REG_COREREGISTER_IC_1_BYTE;
SPI_TX();

//SIGMA_WRITE_REGISTER_BLOCK( DEVICE_ADDR_IC_1, PROGRAM_ADDR_IC_1, PROGRAM_SIZE_IC_1,
Program_Data_IC_1 );
Sub_Address[0] = PROGRAM_ADDR_IC_1;

MSData = (uint8_t *)Program_Data_IC_1;
TXByteCtr = PROGRAM_SIZE_IC_1;
SPI_TX();
//SIGMA_WRITE_REGISTER_BLOCK( DEVICE_ADDR_IC_1, PARAM_ADDR_IC_1, PARAM_SIZE_IC_1,
Param_Data_IC_1 );
Sub_Address[0] = PARAM_ADDR_IC_1;
}

```

```

MSData = (uint8_t *)Param_Data_IC_1;
TXByteCtr = PARAM_SIZE_IC_1;
SPI_TX();

ESCOLHA_CANAL=5;
//SIGMA_WRITE_REGISTER_BLOCK( DEVICE_ADDR_IC_1, REG_COREREGISTER_IC_1_ADDR,
REG_COREREGISTER_IC_1_BYTE, R0_COREREGISTER_IC_1_Default );
Sub_Address[0] = REG_COREREGISTER_IC_1_ADDR;

MSData = (uint8_t *)R0_COREREGISTER_IC_1_Default;
TXByteCtr = REG_COREREGISTER_IC_1_BYTE;
SPI_TX();

//SIGMA_WRITE_REGISTER_BLOCK( DEVICE_ADDR_IC_1, PROGRAM_ADDR_IC_1, PROGRAM_SIZE_IC_1,
Program_Data_IC_1 );
Sub_Address[0] = PROGRAM_ADDR_IC_1;

MSData = (uint8_t *)Program_Data_IC_1;
TXByteCtr = PROGRAM_SIZE_IC_1;
SPI_TX();
//SIGMA_WRITE_REGISTER_BLOCK( DEVICE_ADDR_IC_1, PARAM_ADDR_IC_1, PARAM_SIZE_IC_1,
Param_Data_IC_1 );
Sub_Address[0] = PARAM_ADDR_IC_1;

MSData = (uint8_t *)Param_Data_IC_1;
TXByteCtr = PARAM_SIZE_IC_1;
SPI_TX();

ESCOLHA_CANAL=0;
}

void LOAD_PROGRAM_DEPOIS(void){
ESCOLHA_CANAL=0;
//SIGMA_WRITE_REGISTER_BLOCK( DEVICE_ADDR_IC_1, REG_COREREGISTER_IC_1_ADDR ,
R3_HWCONFIGURATION_IC_1_SIZE, R3_HWCONFIGURATION_IC_1_Default );
Sub_Address[0] = REG_COREREGISTER_IC_1_ADDR;

MSData = (uint8_t *)R3_HWCONFIGURATION_IC_1_Default;
TXByteCtr = R3_HWCONFIGURATION_IC_1_SIZE;
SPI_TX();

Sub_Address[0] = REG_COREREGISTER_IC_1_ADDR;

MSData = (unsigned char *)R4_COREREGISTER_IC_1_Default;
TXByteCtr = REG_COREREGISTER_IC_1_BYTE;
SPI_TX();

ESCOLHA_CANAL=5;
//SIGMA_WRITE_REGISTER_BLOCK( DEVICE_ADDR_IC_1, REG_COREREGISTER_IC_1_ADDR ,
R3_HWCONFIGURATION_IC_1_SIZE, R3_HWCONFIGURATION_IC_1_Default );
Sub_Address[0] = REG_COREREGISTER_IC_1_ADDR;

MSData = (uint8_t *)R3_HWCONFIGURATION_IC_1_Default;
TXByteCtr = R3_HWCONFIGURATION_IC_1_SIZE;
SPI_TX();

Sub_Address[0] = REG_COREREGISTER_IC_1_ADDR;

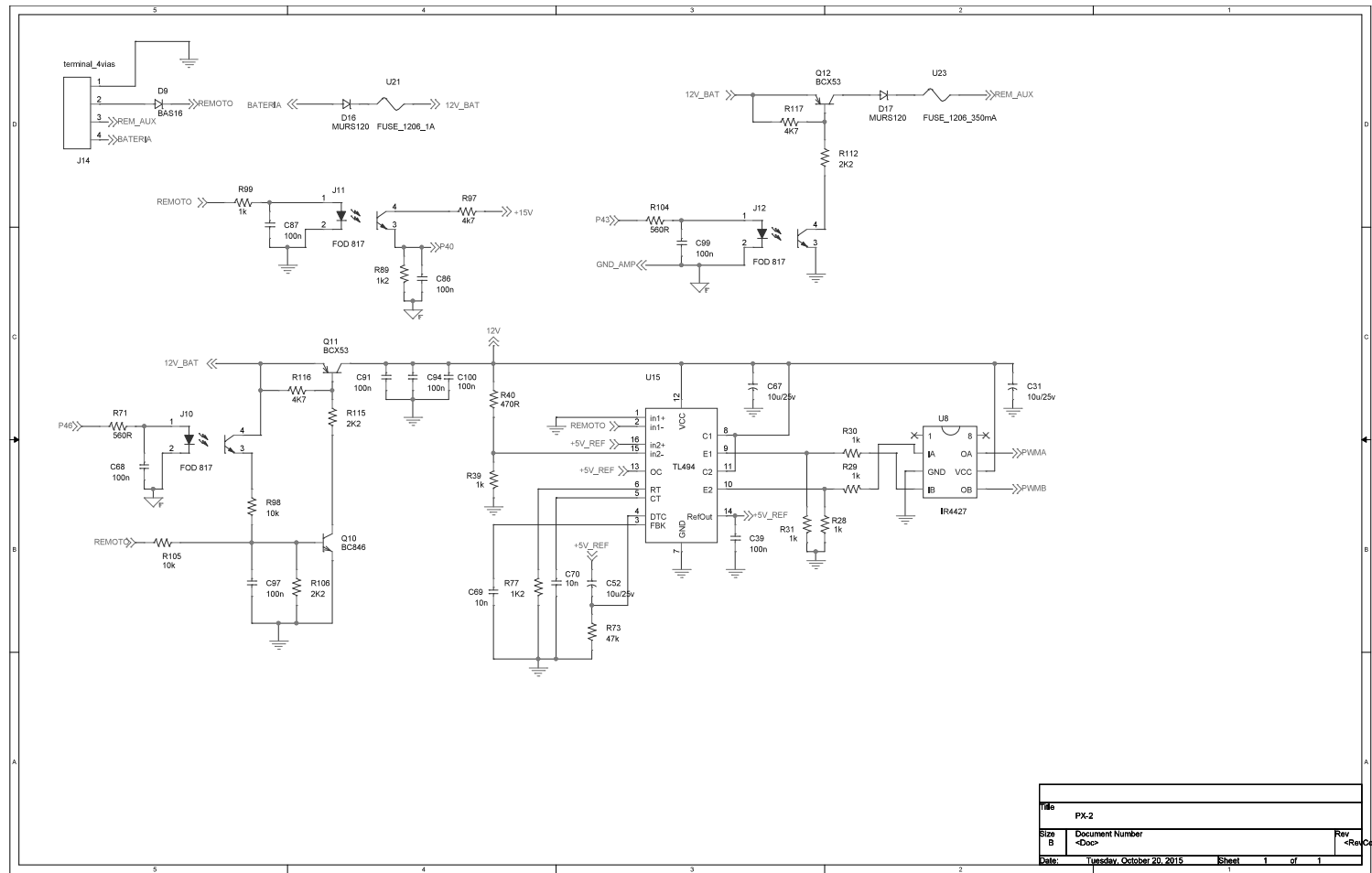
MSData = (unsigned char *)R4_COREREGISTER_IC_1_Default;
TXByteCtr = REG_COREREGISTER_IC_1_BYTE;
SPI_TX();
ESCOLHA_CANAL=0;
}

void FUNCAO_PARTICIONAR(void){
Fx1=0;Fx2=0;Fx3=0;Fx4=0;Fx5=0;
Fx1 =(int) (NUM_VAL/10000);

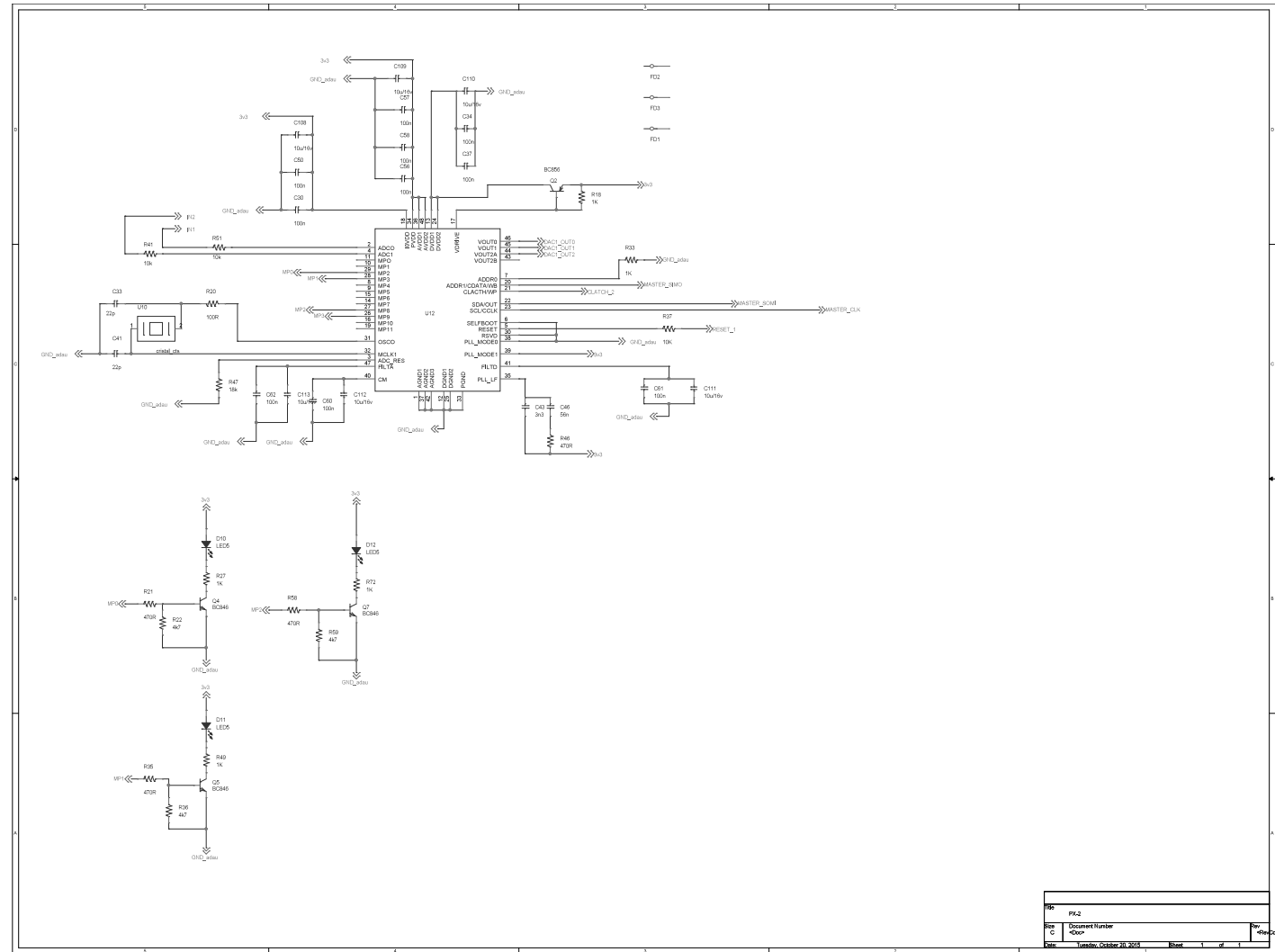
```

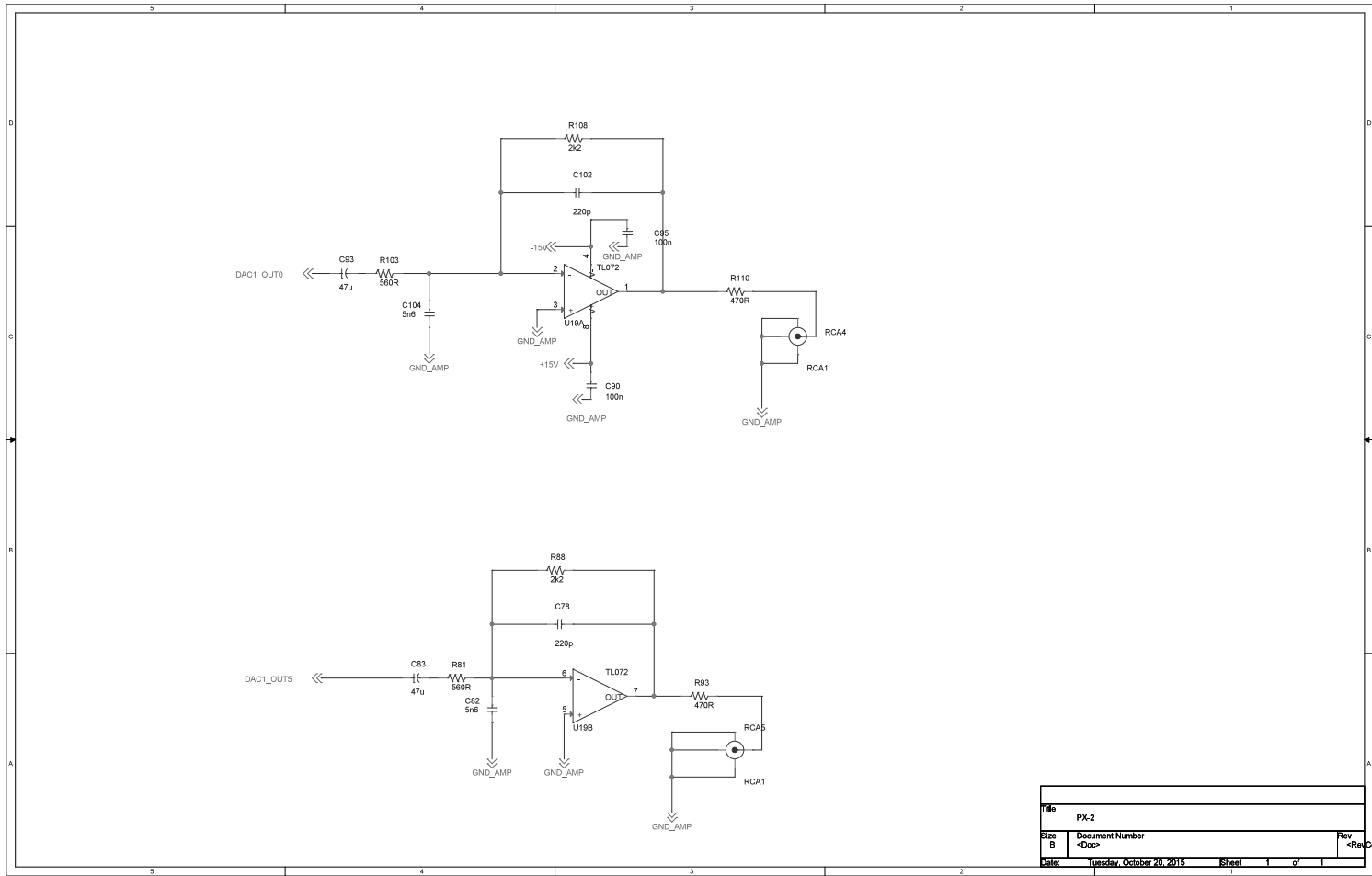
```
Fx2 =(int) ((NUM_VAL-Fx1*10000)/1000);  
Fx3 =(int) ((NUM_VAL-(Fx1*10000+Fx2*1000))/100);  
Fx4 =(int) ((NUM_VAL-(Fx1*10000+Fx2*1000+Fx3*100))/10);  
Fx5 =(int) (NUM_VAL-(Fx1*10000+Fx2*1000+Fx3*100+Fx4*10));  
//FREQ_HP= FQ5+FQ4*10+FQ3*100+FQ2*1000+FQ1*10000;  
}
```

ANEXO A – Esquemático da placa do processador digital de sinais de áudio

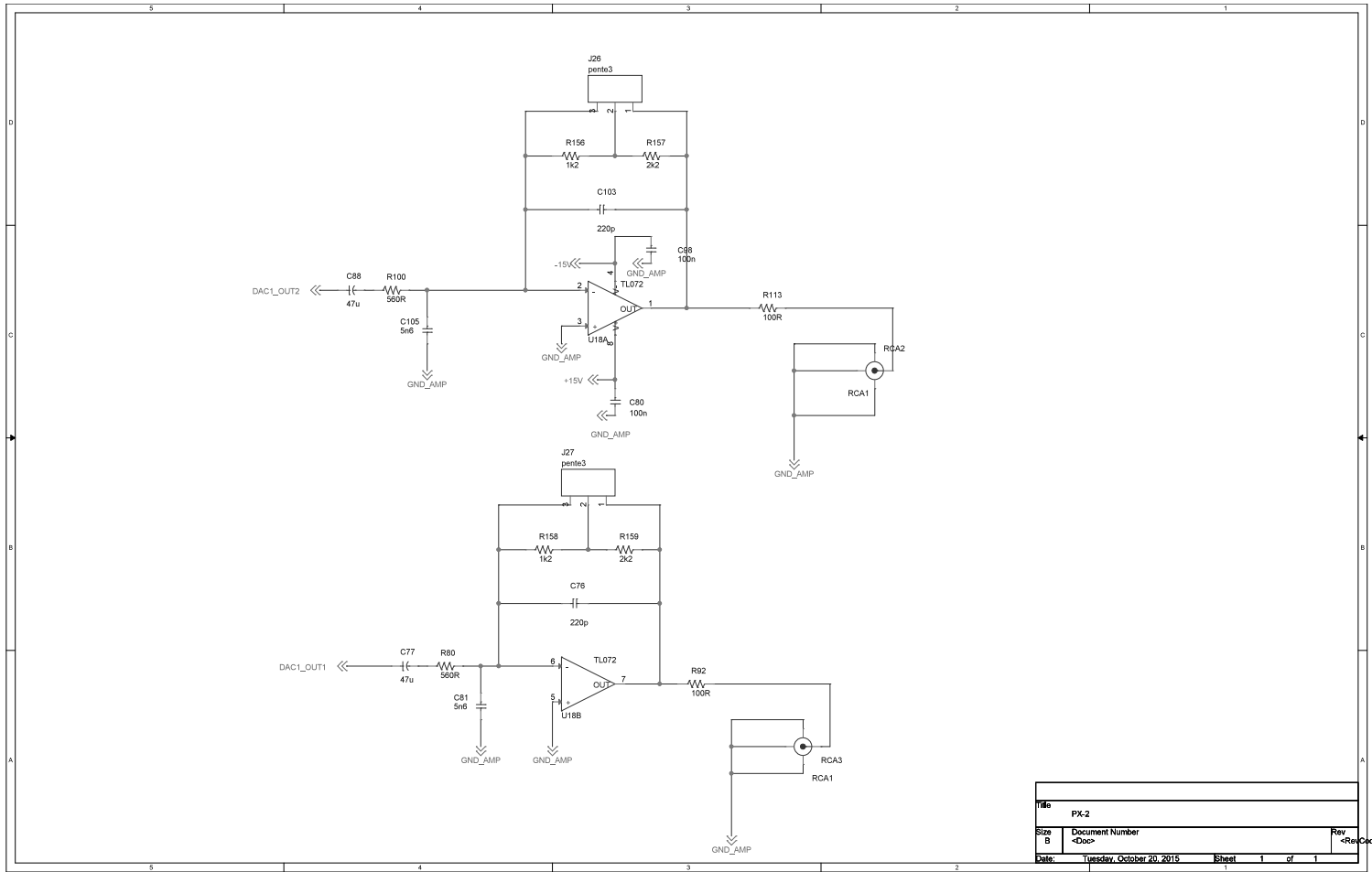


Title			PX-2
Size	Document Number	Rev	
B	<Doc>	<Rev>	
Date:	Tuesday, October 20, 2015	Sheet	1 of 1

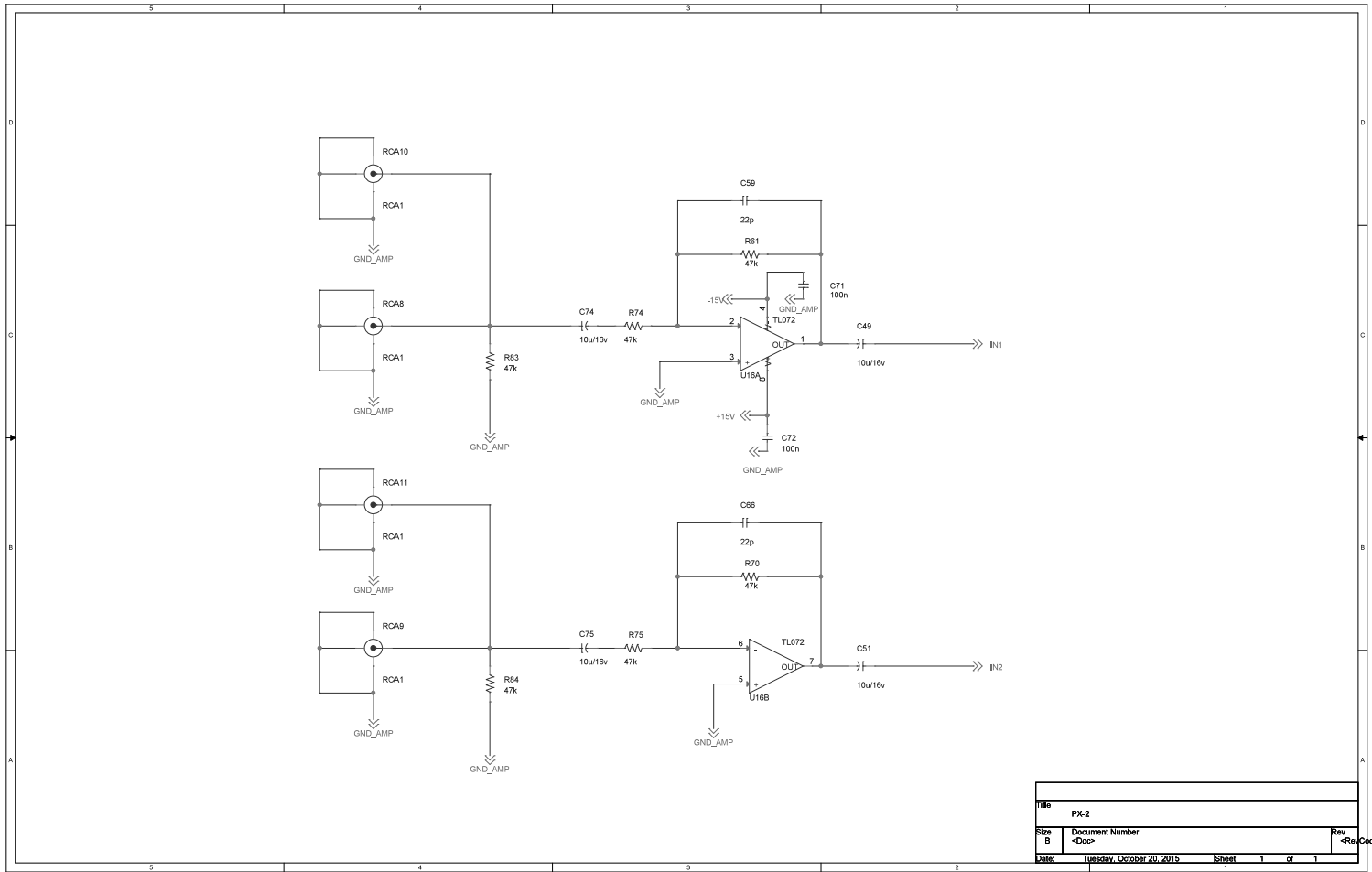




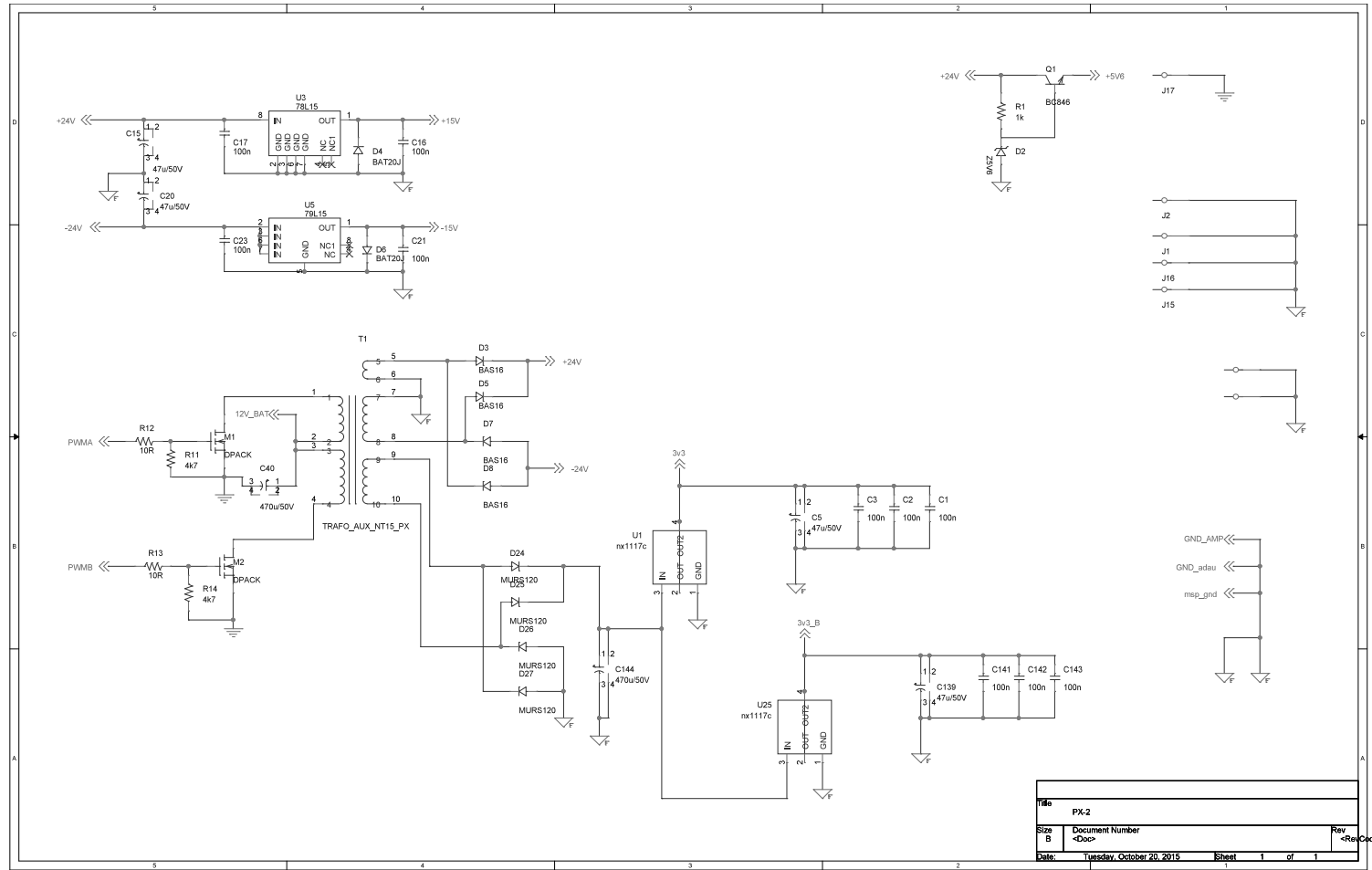
Title		PX-2	
Size	Document Number	Rev	<Rev>
B	<Doc>		
Date:	Tuesday, October 20, 2015	Sheet	1 of 1



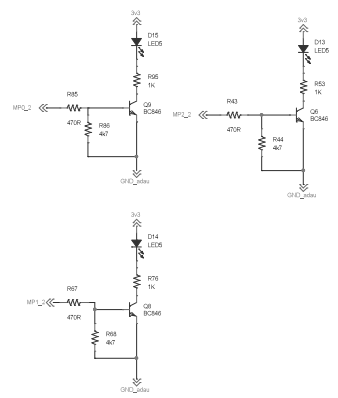
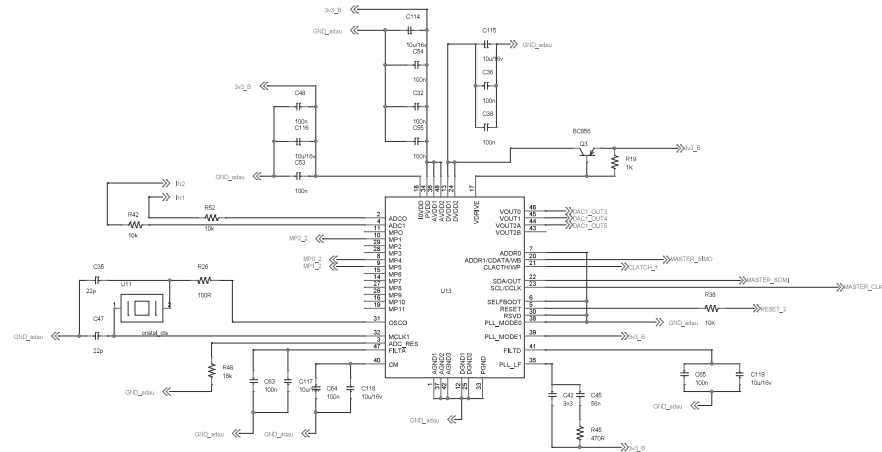
Title	PX-2	
Size	Document Number	Rev
B	<Doc>	<Rev>
Date:	Tuesday, October 20, 2015	Sheet 1 of 1



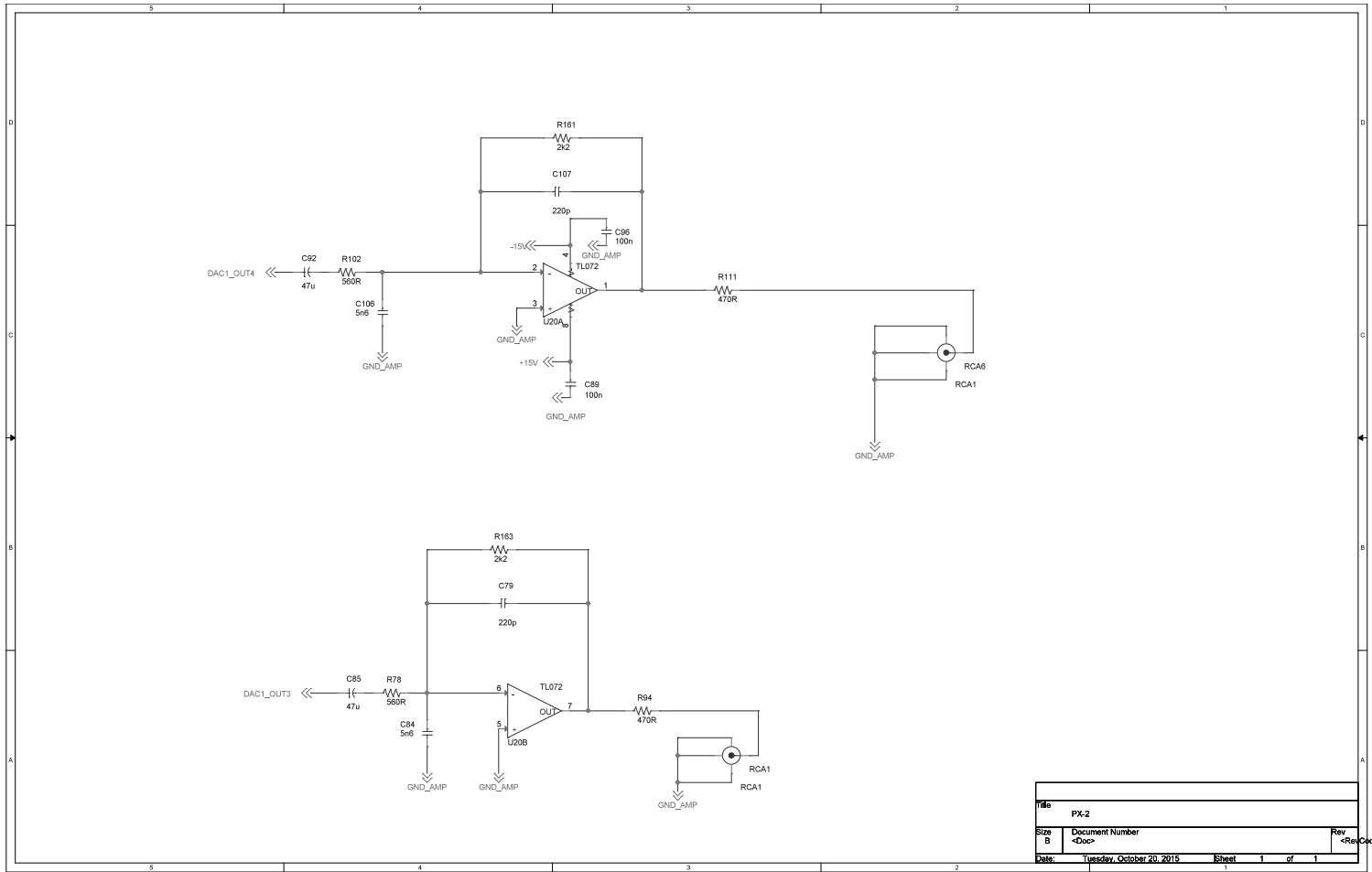
Title		PX-2	
Size	Document Number	Rev	
B	<Doc>	<Rev>	
Date:	Tuesday, October 20, 2015	Sheet	1 of 1



Title			PX-2
Size	Document Number	Rev	
B	<Doc>	<Rev>	
Date:	Tuesday, October 20, 2015	Sheet	1 of 1



Rev	PL2
Docu	Docu#111111
Date	Tuesday, October 21, 2015
Sheet	1 of 1



Title	PX-2		Rev	<Rev>
Size	Document Number			
B	<Doc>			
Date:	Tuesday, October 20, 2015	Sheet	1	of 1