

Universidade de Brasília – UnB
Campus Gama – FGA
Engenharia Eletrônica

**DESENVOLVIMENTO E AVALIAÇÃO DE UM SISTEMA PARA
MODELAGEM E EMULAÇÃO DE AMPLIFICADORES DE ÁUDIO**

GUSTAVO HENRIQUE RODRIGUES VIANA DE OLIVEIRA

Orientador: PROF. DR. CRISTIANO J. MIOSSO, PHD



UNB – UNIVERSIDADE DE BRASÍLIA

FGA – FACULDADE GAMA

ENGENHARIA ELETRÔNICA

**DESENVOLVIMENTO E AVALIAÇÃO DE UM SISTEMA PARA
MODELAGEM E EMULAÇÃO DE AMPLIFICADORES DE ÁUDIO**

GUSTAVO HENRIQUE RODRIGUES VIANA DE OLIVEIRA

ORIENTADOR: PROF. DR. CRISTIANO J. MIOSSO, PHD

**TRABALHO DE CONCLUSÃO DE CURSO
ENGENHARIA ELETRÔNICA**

BRASÍLIA/DF, SETEMBRO DE 2024

UNB – UNIVERSIDADE DE BRASÍLIA
FGA – FACULDADE GAMA
ENGENHARIA ELETRÔNICA

**DESENVOLVIMENTO E AVALIAÇÃO DE UM SISTEMA PARA
MODELAGEM E EMULAÇÃO DE AMPLIFICADORES DE ÁUDIO**

GUSTAVO HENRIQUE RODRIGUES VIANA DE OLIVEIRA

**TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À FACULDADE UNB GAMA DA
UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE BACHAREL EM ENGENHARIA ELETRÔNICA**

APROVADA POR:

Prof. Dr. Cristiano J. Miosso, PhD

(Orientador)

Prof. Dr. Diogo Caetano Garcia, PhD

Prof. Dr. Henrique Gomes de Moura, PhD

FICHA CATALOGRÁFICA

OLIVEIRA, GUSTAVO

Desenvolvimento de Algoritmos para Modelagem e Emulação de Amplificadores de Áudio, [Distrito Federal], 2024.

58p., 210 × 297 mm (FGA/UnB Gama, Bacharelado em Engenharia Eletrônica, 2022).

Trabalho de Conclusão de Curso, Faculdade UnB Gama, Engenharia Eletrônica

1. 2.

3. 4.

I. FGA UnB/UnB. II. Título (série)

REFERÊNCIA

OLIVEIRA, GUSTAVO (2024). Desenvolvimento e Avaliação de um Sistema para Modelagem e Emulação de Amplificadores de Áudio. Trabalho de Conclusão de Curso, Engenharia Eletrônica, Faculdade UnB Gama, Universidade de Brasília, Brasília, DF, 58p.

CESSÃO DE DIREITOS

AUTOR: Gustavo Henrique Rodrigues Viana de Oliveira

TÍTULO: Desenvolvimento e Avaliação de um Sistema para Modelagem e Emulação de Amplificadores de Áudio

GRAU: Bacharel em Engenharia Eletrônica

ANO: 2024

É concedida à Universidade de Brasília permissão para reproduzir cópias desta monografia de conclusão de curso e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta monografia pode ser reproduzida sem a autorização por escrito do autor.

RESUMO

Amplificadores de áudio são dispositivos eletrônicos utilizados em diversos setores. No cenário da música, eles compõem a cadeia do sinal de áudio produzido por uma guitarra elétrica. Com o avanço da tecnologia e a popularidade dos instrumentos musicais digitais, foram surgindo novas soluções para substituir os amplificadores analógicos em aplicações como apresentações ao vivo e até gravações em estúdio, devido à facilidade e versatilidade dos equipamentos. Por se tratar de um mercado altamente competitivo, a maioria dos equipamentos possuem soluções fechadas, sem revelar como é feito o processamento de sinais internamente. Dessa forma, esse trabalho apresenta uma metodologia para realizar a modelagem de amplificadores de áudio utilizando a teoria de processamento de sinais e modelagem de sistemas lineares e invariantes no tempo. São mostrados os métodos utilizados para obtenção dos modelos de resposta ao impulso finita (FIR) e resposta ao impulso infinita (IIR), a partir da definição de um dispositivo capaz de adquirir as amostras aplicadas na entrada e geradas na saída do sistema de forma simultânea. Os modelos são obtidos a partir da análise da amplitude e fase entre os sinais adquiridos calculados por um algoritmo matemático. Logo após, são apresentados resultados dos modelos obtidos para 12 sistemas distintos, utilizando dois equipamentos diferentes. Em geral, a relação sinal-ruído para os modelos FIR foi acima de 24.8 dB, e o valor do erro médio quadrático na ordem de 10^{-7} , já para os modelos IIR, houveram resultados de 6.5 dB de SNR e outros valores que mostram uma alta potência do erro em relação à potência do sinal. O MSE para o caso IIR foi na ordem de 10^{-4} . Para emulação, é feita uma análise e comparação entre os tempos de execução da função responsável por realizar a emulação dos modelos de resposta ao impulso finita em dois ambientes computacionais diferentes. Os valores de latência obtidos, em média, foram próximos da 5 milissegundos, para o *hardware* com menor poder de processamento. Por fim, é feita uma conclusão dos resultados obtidos com objetivo de propor melhorias para trabalhos futuros, em que é efetizado uma reavaliação da obtenção dos modelos IIR, por meio da obtenção de mais medidas. Também são apresentadas explicações detalhadas sobre as principais funções desenvolvidas para os algoritmos de modelagem e emulação, destacando as funções para estimação da relação entre as amplitudes e fases dos sinais de entrada e saída do sistema, e o procedimento utilizado para convolução durante a emulação.

Palavras-chave: modelagem de sistemas de áudio; modelos acústicos de amplificadores de áudio; resposta em frequência de amplificadores de áudio.

ABSTRACT

Audio amplifiers are electronic devices used in various sectors. In the music industry, they form part of the signal chain produced by an electric guitar. With advances in technology and the rise of digital musical instruments, new solutions have emerged to replace analog amplifiers in applications such as live performances and even studio recordings, due to the ease and versatility of digital equipment. Since this is a highly competitive market, most of the equipment features closed solutions, without revealing how audio processing and effects are performed. This work presents a methodology for modeling audio amplifiers using signal processing theory and linear time-invariant system modeling. The methods used to obtain finite impulse response (FIR) and infinite impulse response (IIR) models are shown, starting from the definition of a device capable of acquiring samples applied at the input and generated at the output of the system simultaneously. The models are obtained through the analysis of amplitude and phase between the acquired signals, calculated by a mathematical algorithm. Subsequently, results of the models obtained for 12 distinct systems using two different devices are presented. In general, the signal-to-noise ratio (SNR) for FIR models was above 24.8 dB, and the mean squared error (MSE) was on the order of 10^{-7} . For IIR models, results showed an SNR of 6.5 dB, with other values indicating a high error power relative to the signal power. The MSE for the IIR case was on the order of 10^{-4} . For emulation, an analysis and comparison of execution times of the function responsible for emulating the finite impulse response models are performed in two different computational environments. The average latency values obtained were close to 5 milliseconds for the hardware with lower processing power. Finally, a conclusion is made about the results obtained, with the aim of proposing improvements for future work, such as a reevaluation of the IIR model acquisition by obtaining more measurements. Detailed explanations of the main functions developed for modeling and emulation algorithms are also provided, highlighting the functions for estimating the relationship between the amplitudes and phases of the system's input and output signals, and the procedure used for convolution during emulation.

Keywords: audio system modeling; acoustic models of audio amplifiers; frequency response of audio amplifiers.

SUMÁRIO

1	Introdução	1
1.1	Equipamentos de amplificação e modelagem de áudio	1
1.2	Definição do Problema Científico e Proposta de Pesquisa	4
1.3	Objetivos	5
1.3.1	Objetivos Específicos	5
1.4	Estrutura da monografia	6
2	Fundamentação teórica e Estado da Arte	7
2.1	Amplificadores de áudio	7
2.2	Modelagem de sistemas lineares e invariantes no tempo (LIT)	9
2.2.1	Sinais, sistemas e autofunções de sistemas LIT	9
2.2.2	Função de Transferência e Resposta ao Impulso	11
2.2.3	Amostragem e Análise em Tempo Discreto	13
2.2.4	Análise em Domínio Transformado e Relações entre a Transformada de Fourier e a Transformada Z	13
3	Materiais e Métodos	17
3.1	Projeto de modelagem	17
3.1.1	Introdução de sinais senoidais a entrada do amplificador de áudio	19
3.1.2	Aquisição das amostras dos sinais de áudio gerados pelo gerador de senoides antes e depois de passarem pelo amplificador de áudio . .	20

3.1.3	Obtenção do modelo de resposta ao impulso finita (FIR)	21
3.1.4	Obtenção do modelo de resposta ao impulso infinita (IIR)	23
3.1.5	Avaliação dos modelos obtidos	26
3.2	Projeto de emulação	27
3.2.1	Emulação a partir da convolução do sinal de entrada e a resposta ao impulso do modelo	27
3.2.2	Emulação a partir da equação das diferenças	28
3.2.3	Avaliação da Latência do Processo de Emulação	29
4	Resultados e Discussões	31
4.1	Resultados da modelagem	31
4.1.1	Resultados da modelagem dos arquivos presentes no pedal Sonic IR	32
4.1.2	Resultados da Modelagem de Amplificadores de Áudio	39
4.2	Resultados da Emulação	45
5	Conclusão	49
6	Apêndices	54
6.1	Algoritmos de modelagem	54
6.2	Algoritmos de emulação	56

LISTA DE TABELAS

4.1	Valores do MSE e SNR para os arquivos de áudio provenientes do pedal Sonic IR e os modelos FIR obtidos. Os arquivos listados na Tabela são comparados entre os sinais originais do pedal e os sinais modelados pelos arquivos FIR correspondentes.	36
4.2	Valores do MSE e SNR para a saída obtida pelo modelo IIR de 4 ^a ordem em comparação com os arquivos do pedal Sonic IR.	37

LISTA DE FIGURAS

1.1	Espectro da resposta ao impulso presente no arquivo “01 Champion 1x8.wav” retirado do pedal Sonic IR da Sonicake [1] que emula as características sonoras de um gabinete Fender Champion 1x8.	2
1.2	Espectro da resposta ao impulso presente no arquivo “04 UK 2550 2x12.wav” retirado do pedal Sonic IR da Sonicake [1] que emula as características sonoras de um gabinete Marshall 2550 2x12.	3
1.3	Pedal Tonex One da IK Multimedia [2] que modela e emula amplificadores de guitarra, situado à frente de diversos equipamentos que podem ser modelados e substituídos pelos modelos presentes no pedal.	4
2.1	Curva da resposta em frequência comumente encontrada em amplificadores de áudio. Imagem retirada de [3].	8
2.2	Magnitude da resposta em frequência de dois amplificadores com tecnologias de construção diferentes.	8
2.3	Região de convergência da Transformada Z	15
3.1	Diagrama de blocos das conexões utilizadas para modelagem de um amplificador de áudio, em que o mesmo dispositivo é utilizado para gerar os sinais de entrada e capturar os sinais de saída do amplificador.	18
3.2	Interface de áudio Presonus Studio 24c [4] utilizada para aquisição e reprodução dos sinais de áudio.	19
3.3	Amplificador Blackstar ID Core 20 v2 [5] e microfone dinâmico Santo Angelo SAS 57 [6] utilizados como sistema a ser modelado a partir da introdução de senoides ao amplificador e captura do sinal de saída pelo microfone.	21

3.4	Fluxograma do algoritmo de modelagem de amplificadores de áudio, que permite o usuário definir os parâmetros dos sinais, selecionar os dispositivos de gravação e reprodução, e por fim gera e salva os modelos FIR e IIR.	25
3.5	Diagrama do processo de emulação a partir da convolução do sinal de entrada e da resposta ao impulso finita obtida pela modelagem.	28
3.6	Diagrama do processo de emulação a partir da equação das diferenças, utiliza para os modelos IIR.	29
4.1	Conexões utilizadas para processo de modelagem do pedal Sonic IR. A saída da placa de áudio é conectada ao pedal ABY Box que divide a entrada em duas saídas: um ligada ao canal 1 da placa de áudio e, outra ao pedal Sonic IR. A saída do pedal é ligada no canal 2 da placa de áudio.	31
4.2	Arquivo “01 Champion 1x8.wav” extraído do pedal Sonic IR, que emula as características sonoras de um gabinete Fender Champion 1x8.	32
4.3	Modelo FIR obtido do sistema formado apenas pelo pedal Sonic IR e com o arquivo “01 Champion 1x8.wav” selecionado para emulação.	33
4.4	Modelo FIR obtido do sistema formado apenas pelo pedal Sonic IR com o arquivo “01 Champion 1x8.wav” selecionado, com atraso e inversão de fase compensados. O sinal do modelo foi sobreposto ao sinal do próprio sistema, o arquivo “01 Champion 1x8.wav”.	34
4.5	Magnitude da resposta em frequência do modelo FIR obtido do sistema formado apenas pelo pedal Sonic IR com o arquivo “01 Champion 1x8.wav” selecionado, com atraso e inversão de fase compensados. A resposta em frequência do modelo foi sobreposta à própria resposta do sistema, o arquivo “01 Champion 1x8.wav”.	35
4.6	Comparação entre a magnitude da resposta em frequência do modelo IIR de quarta ordem, em azul, e o arquivo “01 Champion 1x8.wav” do pedal, em vermelho.	38
4.7	Comparação entre a magnitude da resposta em frequência do modelo IIR de quarta ordem, em azul, e o arquivo “11 A Bass 8x10.wav” do pedal, em vermelho.	39
4.8	Configuração do amplificador Blackstar ID Core 20 v2 para modelagem do canal “Clean Warm”.	40

4.9	Modelo FIR obtido para o sistema formado pelo amplificador Blackstar ID Core 20 v2 e o microfone Santo Angelo SAS57.	40
4.10	Magnitude da resposta em frequência do modelo FIR obtido para o sistema formado pelo amplificador Blackstar ID Core 20 v2 e o microfone Santo Angelo SAS57.	41
4.11	Sinal de áudio gravado por uma guitarra elétrica utilizado como entradas para os modelos FIR e IIR.	42
4.12	Sinal de saída obtido a pelo amplificador Blackstar ID Core 20 v2.	43
4.13	Sinal de saída obtido a pela convolução do sinal gravado pela guitarra elétrica e o modelo FIR obtido para o amplificador Blackstar ID Core 20 v2.	44
4.14	Sinal de saída obtido a pela aplicação da equação das diferenças ao sinal gravado pela guitarra elétrica com os pesos do modelo IIR obtido para o amplificador Blackstar ID Core 20 v2	44
4.15	Comparação no domínio da frequência entre o sistema real e os modelos FIR e IIR obtidos do amplificador Blackstar ID Core 20 v2.	45
4.16	Gráfico de dispersão mostrando o tempo de execução da função de convolução para as 100 primeiras chamadas, executada em um computador. A linha tracejada vermelha indica a média, enquanto os pontos amarelo e verde representam os valores máximo e mínimo, respectivamente.	46
4.17	Gráfico de dispersão mostrando o tempo de execução da função de convolução para as 100 primeiras chamadas, executada em um Orange Pi 3B [7]. A linha tracejada vermelha indica a média, enquanto os pontos amarelo e verde representam os valores máximo e mínimo, respectivamente.	47

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

- DSP* – *Digital Signal Processor*
LIT – *Linear e invariante no tempo*
dB – *Decibel*
LIT – *Linear e invariante no tempo*
SNR – *Signal noise rate*
MSE – *Mean square error*
DTFT – *Discrete-Time Fourier Transform*
DFT – *Discrete Fourier Transform*
FIR – *Finite impulse response*
IIR – *Infinite impulse response*

1 INTRODUÇÃO

1.1 Equipamentos de amplificação e modelagem de áudio

O avanço tecnológico do século 20 causou um grande impacto no cenário musical. Novos equipamentos e instrumentos surgiram juntamente com novas possibilidades, resultantes de uma acessibilidade maior a essas novas tecnologias. A guitarra elétrica surgiu por volta de 1930 na Califórnia [8]. A tecnologia das válvulas, que são utilizadas até os dias de hoje, foi estimulada pela Segunda Guerra Mundial, e posteriormente, a introdução dos transistores nos equipamentos musicais, juntamente com o acesso aos computadores, possibilitou escalá-los e torná-los acessíveis, popularizando o interesse das pessoas em não apenas aprender a tocar um instrumento, mas também na criação de músicas [9].

Dessa forma, o surgimento de vários artistas e bandas no século 20 ainda influencia as produções musicais nos dias de hoje. Jimi Hendrix foi um dos responsáveis por introduzir diversos efeitos no contexto da guitarra elétrica [10], os quais são utilizados em vários estilos da guitarra. Muitos músicos têm como referência artistas que modificaram, de alguma forma, a história da música e possuem um timbre único. Dessa forma, aliado ao avanço tecnológico, surgiram os equipamentos de modelagem e emulação, que têm o papel de aproximar o som de saída que passou pelo modelo ao som de saída do sistema real, que pode ser de amplificadores, pedais e alto-falantes situados em contextos específicos, como um grande estúdio de gravação ou uma apresentação ao vivo. Para entender como o som é efetivamente gerado e reproduzido nesses contextos, é essencial conhecer o processo envolvido na produção do sinal de áudio.

Um sinal de áudio gerado por uma guitarra elétrica passa por várias etapas antes de ser reproduzido por um alto-falante. O primeiro passo ocorre no captador, cuja função é converter o movimento das cordas da guitarra elétrica em um sinal de tensão. De forma resumida, o funcionamento do captador segue a Lei de Faraday, que estabelece que um campo magnético variável ao longo do tempo pode gerar uma força eletromotriz, ou seja, uma tensão elétrica [11]. Desse modo, o movimento das cordas, geralmente feitas de aço ou níquel, induz o captador (um enrolamento de bobinas) a gerar o sinal de tensão

correspondente [12].

Em seguida, esse sinal atravessa diferentes estágios no amplificador de guitarra até alcançar o alto-falante. Esses estágios compreendem: pré-amplificação, equalização e amplificação de potência. A pré-amplificação tem como função elevar o nível de tensão do sinal captado, sem modificar significativamente suas características. A equalização, por sua vez, realiza uma filtragem para compensar a ressonância do captador e proporcionar ao usuário um controle sobre o ajuste de ganho para determinadas frequências. Finalmente, a amplificação de potência intensifica o sinal de tensão, tornando-o apto para acionar o alto-falante [13].

Além dos componentes eletrônicos, o ambiente onde o alto-falante está localizado também afeta o resultado sonoro. A reverberação altera a estrutura do som, modificando aspectos como a duração e o tom [14]. Esses efeitos de reverberação podem mudar a forma como o som do alto-falante é ouvido. Portanto, a acústica do ambiente é essencial para a percepção do som produzido pelo alto-falante.

Existem diversos equipamentos especializados em emular não apenas o processo completo, mas também etapas específicas que o sinal de áudio percorre. As Figuras 1.1 e 1.2 mostram a representação no domínio da frequência de dois sinais, que são respostas ao impulso de dois sistemas diferentes: um gabinete com um alto-falante de oito polegadas e outro com dois alto-falantes de doze polegadas, respectivamente.

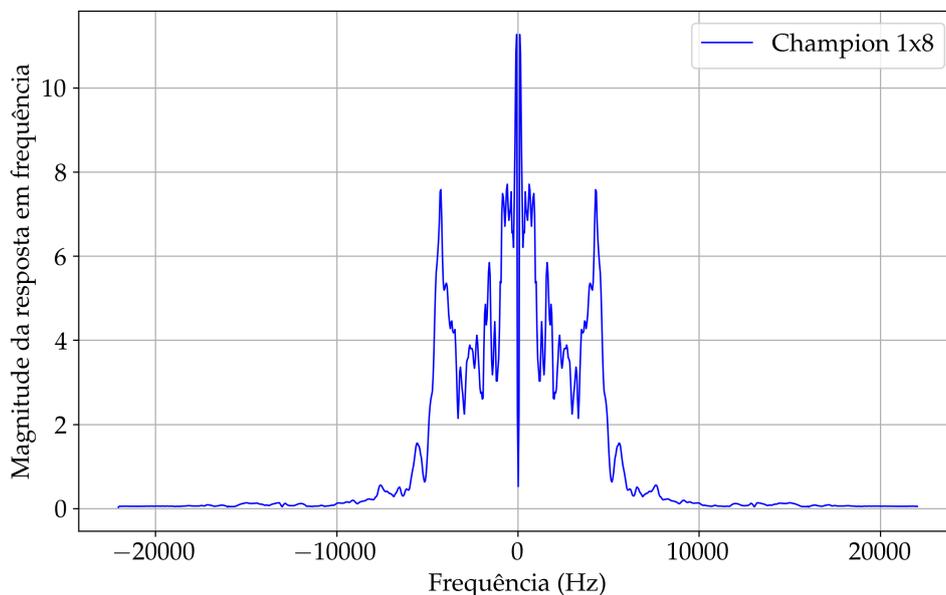


Figura 1.1. Espectro da resposta ao impulso presente no arquivo “01 Champion 1x8.wav” retirado do pedal Sonic IR da Sonicake [1] que emula as características sonoras de um gabinete Fender Champion 1x8.

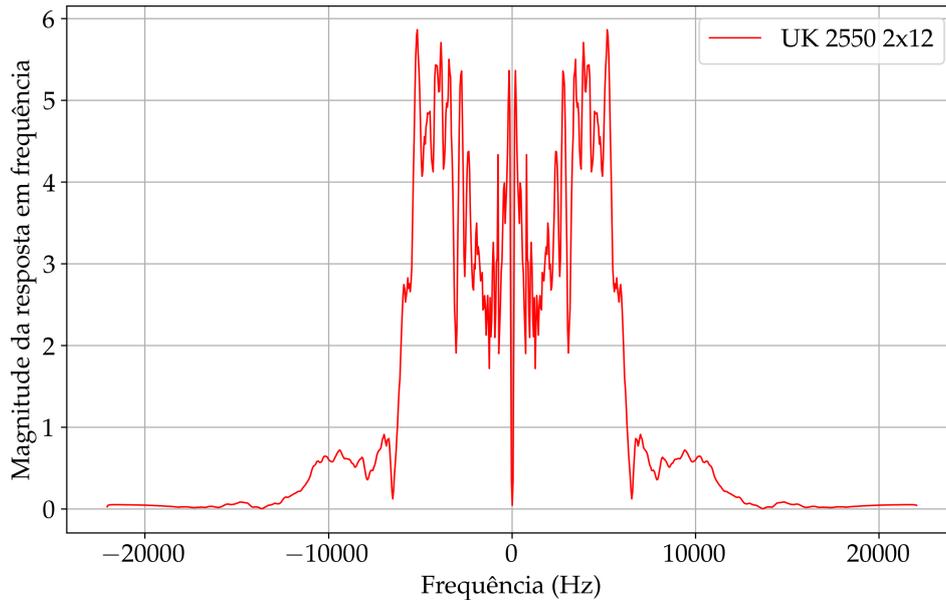


Figura 1.2. Espectro da resposta ao impulso presente no arquivo “04 UK 2550 2x12.wav” retirado do pedal Sonic IR da Sonicake [1] que emula as características sonoras de um gabinete Marshall 2550 2x12.

Os espectros evidenciam as particularidades de cada sistema. Se o mesmo sinal de áudio passasse por cada um desses sistemas, os resultados seriam diferentes, pois a magnitude do espectro mostra o ganho ou atenuação do sinal para cada frequência. Dessa forma, a sonoridade resultante que cada sistema proporciona é única, e ao adicionar outros equipamentos na cadeia do sinal, as possibilidades aumentam.

Para que um músico tenha acesso a essas diferentes possibilidades, é necessário possuir todos os equipamentos, o que exige um grande investimento, nesse caso, os equipamentos de emulação surgiram para facilitar o acesso a timbres variados sem a necessidade de adquirir os sistemas físicos, mas sim seus modelos digitais. A utilização de emuladores traz dois benefícios diretos: a facilidade e praticidade de poder acessar diversos timbres em um só equipamento, e o acesso a timbres de artistas e estúdios renomados, visto que esses modelos podem ser comercializados.

Alguns equipamentos de emulação conhecidos são os chamados *Impulse Response Loaders*, como o pedal Sonic IR da Sonicake [1], que emula modelos de gabinetes com um ou mais alto-falantes. Existem também equipamentos de captura e modelagem, como o Kemper Profiler [15], a pedaleira Quad-Cortex da Neural DSP [16] e o pedal Tonex One da IK Multimedia [2], mostrado na Figura 1.3.



Figura 1.3. Pedal Tonex One da IK Multimedia [2] que modela e emula amplificadores de guitarra, situado à frente de diversos equipamentos que podem ser modelados e substituídos pelos modelos presentes no pedal.

A Figura 1.3 apresenta o pedal Tonex One à frente de outros amplificadores de guitarra que podem ser modelados e substituídos pelo próprio pedal.

Embora o objetivo seja reduzir os custos, e os equipamentos de emulação permitam o acesso a timbres variados em um único dispositivo, esses equipamentos de modelagem ainda possuem um custo elevado e têm propriedade intelectual fechada, ou seja, o processo pelo qual a modelagem é executada não é divulgado, dificultando o avanço e melhorias por parte de comunidade.

1.2 Definição do Problema Científico e Proposta de Pesquisa

Diante essas limitações, esse trabalho apresenta um possível processo para modelagem digital da resposta em frequência de amplificadores de áudio para guitarras elétricas. O processo é validado a partir da análise dos resultados obtidos para a modelagem de um sistema composto por um amplificador, o qual já possui pré-amplificação e um amplificação de potência, alto-falante e microfone. O modelo do sistema é criado utilizando equipamentos de gravação e algoritmos matemáticos. Os resultados são analisados a partir de comparações entre o sistema original e a emulação do modelo digital obtido do mesmo. O objetivo é identificar as influências sonoras que o equipamento exerce sobre o sinal captado, definindo métricas objetivas para qualificar o resultado do modelo em comparação com o sistema original.

Os algoritmos matemáticos foram desenvolvidos para realizar não apenas a modelagem da resposta em frequência de um amplificador de áudio, mas também a emulação do modelo criado. Essa emulação, ao substituir o amplificador na cadeia do sinal, busca obter uma saída comparável àquela do sistema original, quando ambos são excitados pela mesma entrada.

Para garantir uma avaliação sistemática e objetiva, a pesquisa propõe a utilização de métricas específicas que permitem a comparação entre a saída obtida pela modelagem e o sistema real, ambos estimulados pela mesma entrada. Dentre essas métricas, são analisadas a relação sinal-ruído e o cálculo do erro médio quadrático entre os modelos e o sistema de referência. Além disso, foi necessário desenvolver algoritmos com baixa latência para a emulação, não ultrapassando 20 milissegundos entre o momento em que o sinal é adquirido e processado, e depois liberado na saída [17], garantindo que não afete a tocabilidade do guitarrista durante o processo de gravação, por exemplo. A latência obtida tem duas origens fundamentais: o próprio processo de modelagem, que requer o armazenamento de amostras passadas do sinal de áudio, e o processamento computacional realizado a partir dos cálculos e do processo de amostragem.

Em suma, a proposta abrange não apenas a modelagem da resposta em frequência, mas também a implementação prática dessa modelagem, assegurando que a emulação resultante seja não apenas acurada, mas também utilizável em contextos musicais reais.

1.3 Objetivos

O objetivo desse trabalho é desenvolver e avaliar o desempenho de algoritmos e do *hardware*, responsável por adquirir e reproduzir sinais de áudio para efetuar a modelagem e emulação de amplificadores de áudio. Inicialmente, esses algoritmos são implementados em um ambiente computacional, com a perspectiva de adaptá-los posteriormente para sistemas embarcados, ampliando assim a aplicabilidade do equipamento resultante. Quanto ao *hardware*, a intenção é utilizar equipamentos de gravação comerciais para facilitar o acesso do usuário e as conexões relacionadas ao projeto.

1.3.1 Objetivos Específicos

Para atingir o objetivo geral de desenvolver algoritmos para modelagem e emulação de amplificadores de áudio, foram delineados os seguintes objetivos específicos:

- Definir um *hardware* para gerar, introduzir e capturar sinais de áudio de amplificadores;
- Desenvolver um algoritmo que obtém o modelo FIR e IIR do amplificador a partir do processamento das amostras dos sinais;

- Realizar uma avaliação a partir do cálculo do erro médio quadrático entre os modelos e o sistema de referência;
- Avaliar a relação sinal-ruído para os modelos em comparação ao sistema de referência;
- Implementar um algoritmo que utiliza o modelo para realizar a emulação do amplificador em tempo real;
- Avaliar a latência do processo de emulação.

É importante enfatizar que o foco são as comparações entre os modelos e o sistema real. Uma comparação entre os modelos IIR e FIR pode ser abordada em trabalhos futuros.

1.4 Estrutura da monografia

A estrutura da monografia é dividida em capítulos que buscam apresentar os conhecimentos necessários, o desenvolvimento da solução, os resultados obtidos e as conclusões para o problema proposto. Inicialmente, é apresentada uma fundamentação teórica abordando a teoria existente e o estado da arte sobre análise da resposta em frequência amplificadores de áudio, sistemas lineares e invariantes no tempo (LIT), modelagem de sistemas LIT, autofunções de sistemas LIT, análise em domínio transformado e a relação entre a Transformada de Fourier e a Transformada Z . Dessa forma, o Capítulo 2 apresenta uma base para o desenvolvimento da solução, o qual é apresentado no Capítulo 3 a partir dos materiais necessários e os métodos utilizados para o desenvolvimento da solução, mostrando o processo de desenvolvimento do projeto de modelagem e emulação de amplificadores de áudio. Nesse mesmo capítulo é apresentado o desenvolvimento da arquitetura de *hardware* utilizada, o processo de aquisição e processamento de sinais, a lógica para o desenvolvimento dos algoritmos de modelagem e emulação, bem como a metodologia adotada para analisar e comparar os resultados entre os modelos obtidos e os sistemas modelados.

O Capítulo 4 apresenta os resultados para os modelos obtidos, apresentando os sinais das respostas ao impulso obtidas e a resposta em frequência dos modelos. Além disso, é mostrado uma comparação entre os modelos e os sistemas modelados em termos da relação sinal-ruído e erro quadrático médio, e uma avaliação da latência obtida pelo processo de emulação em comparação com uma solução comercial existente.

2 FUNDAMENTAÇÃO TEÓRICA E ESTADO DA ARTE

2.1 Amplificadores de áudio

Os amplificadores de áudio são dispositivos eletrônicos utilizados para aumentar o ganho de sinais de áudio, permitindo que esses sinais sejam processados e reproduzidos em volumes audíveis por meio de alto-falantes. No contexto de uma guitarra elétrica, a amplificação do sinal envolve dois estágios principais: a pré-amplificação e a amplificação de potência. O pré-amplificador tem a função de aumentar a amplitude do sinal gerado pela guitarra, elevando o nível de tensão do sinal. Já o amplificador de potência, além de aumentar ainda mais a tensão, também amplifica a corrente, fornecendo a potência necessária para acionar os alto-falantes [3].

A linearidade de um amplificador é uma característica crucial na maioria das aplicações, pois um sinal de áudio transporta informações as quais podem ser distorcidas por não linearidades no dispositivo. No entanto, em algumas aplicações para amplificadores de áudio, uma certa distorção não é apenas tolerada, mas muitas vezes desejada, pois pode conferir ao som características únicas que são apreciadas por músicos e entusiastas da música. A linearidade, no contexto de amplificadores de áudio, refere-se a um ganho constante ou proporcional ao longo de uma ampla faixa de frequências, limitada pelas propriedades do próprio amplificador. O conceito de linearidade nos sistemas, em um contexto mais amplo, é discutido na subseção 2.2.1.

A Figura 2.1 ilustra uma típica curva de resposta em frequência encontrada em amplificadores de áudio. Nessa Figura, o eixo vertical representa o ganho em decibéis, enquanto o eixo horizontal representa a frequência ω . Essa curva é essencial para entender como o amplificador responde a diferentes frequências, o que pode impactar significativamente a qualidade do som reproduzido.

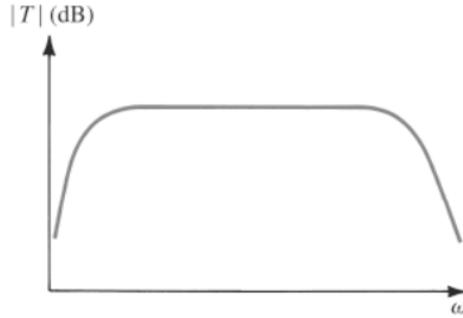


Figura 2.1. Curva da resposta em frequência comumente encontrada em amplificadores de áudio. Imagem retirada de [3].

A faixa de operação considerada para um amplificador de áudio geralmente é de 20 Hz a 20 kHz, que é a faixa audível ao ouvido humano [18]. Essa faixa de frequência garante que todas as nuances do som sejam capturadas e amplificadas de maneira que sejam perceptíveis ao ouvinte. A Figura 2.2 apresenta um comparativo da resposta em frequência entre dois amplificadores de diferentes tecnologias [19]. Esse comparativo ajuda a destacar as diferenças no comportamento das diferentes abordagens tecnológicas.

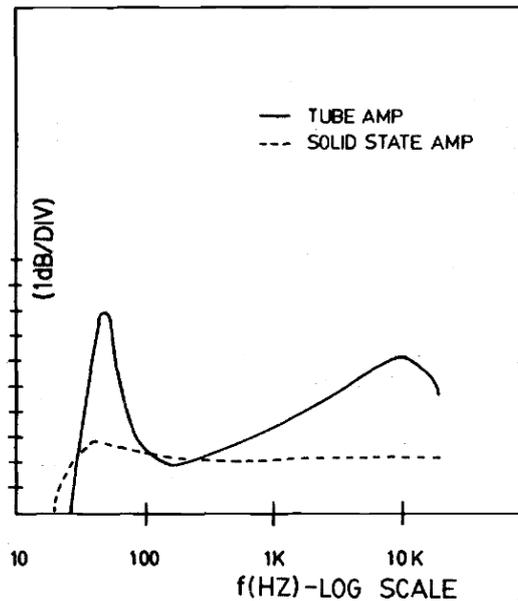


Figura 2.2. Magnitude da resposta em frequência de um amplificador *solid state* (estado sólido), construído por transistores, e de um amplificador *tube* (valvulado), construído por válvulas. Imagem retirada de [19], com tradução.

A Figura 2.2 destaca a diferença significativa entre os dois tipos de amplificadores, mostrando como o amplificador valvulado apresenta uma resposta em frequência menos plana, com variações notáveis na magnitude ao longo de diferentes frequências. Essa característica é justamente o que atrai muitos músicos. Surgidos no início do século 20 [3],

os amplificadores valvulados são apreciados por seu som único, e, apesar do avanço de novas tecnologias, continuam a ser preferidos por muitos entusiastas devido ao seu caráter sonoro distinto.

2.2 Modelagem de sistemas lineares e invariantes no tempo (LIT)

Com o objetivo de modelar a resposta em frequência de amplificadores, é preciso compreender a matemática por trás dos sinais de áudio, como eles são modificados pelo amplificador e por outros dispositivos da cadeia do sinal, e como obter um modelo que sintetiza todas as características do sinal de saída.

2.2.1 Sinais, sistemas e autofunções de sistemas LIT

Um sinal pode ser entendido como uma coleção de dados que são processados por um sistema, resultando em um sinal de saída modificado [20]. Por exemplo, o som produzido por uma guitarra elétrica é um sinal, e o amplificador de áudio é um sistema que processa esse sinal. Como discutido anteriormente, o amplificador de áudio pode ser classificado como linear ou não linear, e o conceito de linearidade é descrito pela propriedade matemática da superposição.

A superposição é uma característica dos sistemas lineares e pode ser compreendida por meio de duas propriedades principais: aditividade e homogeneidade. A aditividade implica que a resposta de um sistema a uma soma de sinais de entrada é igual à soma das respostas individuais a cada sinal de entrada. Em outras palavras, se um sistema recebe dois sinais $x_1(t)$ e $x_2(t)$, a resposta combinada ao sinal $x_1(t) + x_2(t)$ será a soma das respostas individuais a $x_1(t)$ e $x_2(t)$ separadamente. A homogeneidade, por outro lado, indica que se um sinal de entrada for multiplicado por uma constante escalar, a resposta do sistema será multiplicada pela mesma constante. Isso significa que, se o sinal $x_1(t)$ for escalado por um fator a , a saída correspondente será escalada pelo mesmo fator.

Para que um sistema seja classificado como linear, ele deve satisfazer as condições de aditividade e homogeneidade simultaneamente [21]. Formalmente, um sistema é considerado linear se, para duas entradas $x_1(t)$ e $x_2(t)$ e coeficientes escalares a e b , é válida a relação

$$H[a \cdot x_1(t) + b \cdot x_2(t)] = a \cdot H[x_1(t)] + b \cdot H[x_2(t)] = a \cdot y_1(t) + b \cdot y_2(t), \quad (2.1)$$

onde H é o operador que descreve a transformação do sistema, $x_1(t)$ e $x_2(t)$ são os sinais de entrada, $y_1(t)$ e $y_2(t)$ são os sinais de saída, e a e b são constantes.

Além da linearidade, os sistemas podem ser classificados de acordo com a sua variância no tempo [20]. Um sistema é dito invariante no tempo se suas propriedades não se alteram ao longo do tempo, ou seja, se a resposta do sistema a uma determinada entrada não depende do instante em que essa entrada é aplicada. Isso significa que se um sinal de entrada for deslocado no tempo por uma quantidade t_0 , a saída correspondente será deslocada pela mesma quantidade t_0 , sem qualquer alteração na forma do sinal. Formalmente, para um sistema invariante no tempo, deve ser satisfeita a relação

$$H[x_1(t - t_0)] = y_1(t - t_0), \quad (2.2)$$

onde H é o operador que descreve a transformação do sistema, $x_1(t)$ é o sinal de entrada, $y_1(t)$ é o sinal de saída, e t_0 é o deslocamento no tempo.

A classificação de um sistema como linear e invariante no tempo (LIT) simplifica o processo de modelagem, visto que as propriedades descritas podem ser usadas para determinar características do sistema. Dessa forma, são utilizados sinais específicos, que ao serem aplicados a sistemas LIT, facilitam a análise dessas características. Esses sinais específicos são as autofunções, definidas como funções que, ao passarem pelo sistema, possuem sua essência inalterada [20]. Por exemplo, as exponenciais complexas de duração infinita, como as senoides, ao serem injetadas em um sistema LIT, preservam sua frequência na saída, sofrendo apenas alterações em amplitude e/ou fase. Isso pode ser mostrado por

$$H[e^{j\omega t}] = A(\omega) \cdot e^{j(\omega t + \phi(\omega))}, \quad (2.3)$$

onde H representa o operador do sistema, $e^{j\omega t}$ é a autofunção aplicada ao sistema, $A(\omega)$ é o ganho de amplitude, e $\phi(\omega)$ é a mudança de fase. A equação demonstra que, em um sistema LIT, a forma da exponencial complexa é preservada, enquanto a amplitude e a fase podem ser modificadas.

Além das classificações de linearidade e invariância no tempo, dois outros aspectos cruciais a serem considerados em um sistema são a estabilidade e a causalidade. Um sistema é considerado estável se, para qualquer entrada $x(t)$ limitada, a saída $y(t)$ também

for limitada [20]. Matematicamente, isso pode ser por

$$\text{Se } \|x(t)\| < \infty, \text{ então } \|y(t)\| < \infty, \quad (2.4)$$

onde $\|x(t)\|$ e $\|y(t)\|$ representam as magnitudes dos sinais de entrada e saída, respectivamente. Essa propriedade é essencial para garantir que o sistema não gere respostas que cresçam indefinidamente para entradas limitadas, o que seria impraticável em aplicações reais.

Por outro lado, a causalidade refere-se à propriedade de que a saída de um sistema em um dado instante de tempo t depende apenas dos valores de entrada em instantes presentes ou passados, e não de valores futuros. Formalmente, um sistema é causal se, para duas entradas $x_1(t)$ e $x_2(t)$, onde $x_1(t) = x_2(t)$ para $t < t_0$, então $y_1(t) = y_2(t)$ para $t < t_0$ [20], expressado por

$$y(t) = H[x(t)] \text{ para } t \geq t_0, \quad (2.5)$$

indicando que a resposta $y(t)$ em qualquer instante t depende apenas dos valores de entrada $x(t)$ até aquele instante. A causalidade é uma característica fundamental em sistemas físicos, pois implica que o sistema não pode “ver” o futuro, ou seja, a resposta em um dado momento só pode ser influenciada por entradas ocorridas até aquele momento.

2.2.2 Função de Transferência e Resposta ao Impulso

Para entender melhor a utilização das autofunções no processo de modelagem, é preciso apresentar a importância da função de transferência e da resposta ao impulso. Quando conhecida a resposta ao impulso de um sistema, é possível determinar a resposta do sistema a uma entrada $x(t)$ ao somar as respostas da decomposição de $x(t)$ em componentes impulso. O impulso unitário, denotado como $\delta(t)$, é uma função matemática que possui uma duração que tende a zero, mas com uma área total igual a um [20]. A função impulso unitário é definida por

$$\delta(t) = \begin{cases} 0, & \text{se } t \neq 0, \\ 1, & \text{se } t = 0. \end{cases} \quad (2.6)$$

Considerando $h(t)$ como a resposta ao impulso unitário, a resposta $y(t)$ do sistema a uma entrada qualquer $x(t)$ pode ser obtida pela convolução de $h(t)$ com $x(t)$, conforme

$$y(t) = (h * x)(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau) d\tau. \quad (2.7)$$

Esta relação é especialmente relevante quando a entrada do sistema é uma exponencial complexa de duração infinita, como $x(t) = e^{st}$, onde s é uma variável complexa. Nesse caso, a saída do sistema pode ser expressa por

$$y(t) = h(t) * e^{st} = \int_{-\infty}^{\infty} h(\tau)e^{s(t-\tau)} d\tau = e^{st} \int_{-\infty}^{\infty} h(\tau)e^{-s\tau} d\tau. \quad (2.8)$$

A expressão acima utiliza a propriedade da convolução e define a integral da convolução como uma função da variável complexa s . Esta função é comumente reescrita como $H(s)$, que representa a função de transferência do sistema, dada por

$$H(s) = \int_{-\infty}^{\infty} h(\tau)e^{-s\tau} d\tau. \quad (2.9)$$

Dentro da região de convergência da integral, $H(s)$ assume um valor constante para um dado s , indicando que a entrada e^{st} e a resposta do sistema possuem a mesma forma, mas escaladas por $H(s)$. A função de transferência $H(s)$ estabelece a relação entre o sinal de entrada e o de saída do sistema [20], sendo definida por

$$H(s) = \frac{Y(s)}{X(s)}, \quad (2.10)$$

onde $Y(s)$ é a transformada de Laplace da saída e $X(s)$ é a transformada de Laplace da entrada. A função de transferência é um conceito aplicável exclusivamente a sistemas Lineares e Invariantes no Tempo (LIT) [20]. Para um sistema LIT, a transformada de Laplace é uma ferramenta poderosa que descreve a relação entre a excitação e a resposta do sistema, especialmente para entradas que são exponenciais complexas. A transformada de Laplace de um sinal $x(t)$ é definida por

$$L\{x(t)\} = X(s) = \int_{-\infty}^{\infty} x(t)e^{-st} dt, \quad (2.11)$$

onde $L\{\cdot\}$ denota o operador da transformada de Laplace e e^{-st} é a função base da transformada.

2.2.3 Amostragem e Análise em Tempo Discreto

A transformada de Laplace é definida para o domínio contínuo e fornece uma ferramenta essencial para a análise de sistemas LIT. No entanto, no contexto de sistemas digitais, a análise é realizada no domínio do tempo discreto. Para isso, o processo de amostragem dos sinais de áudio é fundamental.

O Teorema da Amostragem afirma que um sinal em tempo contínuo $x_c(t)$, que é limitado em banda, pode ser completamente reconstruído a partir de suas amostras discretas $x[n]$, desde que a frequência de amostragem seja pelo menos o dobro da máxima frequência presente no espectro do sinal contínuo. Esse critério para o valor da frequência de amostragem é um critério suficiente, mas não obrigatório e é conhecido como Critério de Nyquist [22]. A relação entre o sinal contínuo e o sinal discreto é expressa por

$$x[n] = x_c(n \cdot T_s), \quad (2.12)$$

onde T_s representa o período de amostragem, que é o inverso da taxa, ou frequência de amostragem

$$T_s = \frac{1}{f_s}. \quad (2.13)$$

Conforme discutido na seção 2.1, a maior frequência audível para os sensores humanos é cerca de 20 kHz. Portanto, para capturar adequadamente o sinal de áudio, é possível utilizar uma taxa de amostragem de pelo menos 40 kHz, que é o dobro da maior frequência audível. Na prática, as taxas de amostragem mais comuns para sinais de áudio são 44,1 kHz, 48 kHz, 96 kHz e 192 kHz. Esses valores garantem que o sinal de áudio seja amostrado de maneira adequada e que a reconstrução do sinal discreto seja fiel ao sinal contínuo original.

2.2.4 Análise em Domínio Transformado e Relações entre a Transformada de Fourier e a Transformada Z

A análise em domínio transformado é essencial para entender o comportamento de sistemas em diferentes frequências. A Transformada de Fourier é uma ferramenta fundamental nessa análise, permitindo a conversão de um sinal entre os domínios do tempo e da frequência [22]. A seguir, são apresentadas as definições da Transformada de Fourier em tempo contínuo e discreto, bem como a Transformada Z e suas relações com a

Transformada de Fourier.

2.2.4.1 Transformada de Fourier

Para sinais em tempo contínuo, a Transformada de Fourier (FT) é definida por

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt, \quad (2.14)$$

onde $x(t)$ é o sinal no domínio do tempo, $X(f)$ é o sinal transformado no domínio da frequência, e f é a frequência em hertz. A Transformada de Fourier converte um sinal contínuo no tempo em uma representação contínua na frequência [22].

Para sinais em tempo discreto, é definida a Transformada de Fourier em Tempo Discreto (DTFT) por

$$X(f) = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-j2\pi fn}, \quad (2.15)$$

onde $x[n]$ é o sinal discreto no domínio do tempo, $X(f)$ é o sinal transformado no domínio da frequência discreta, e f é a frequência em hertz. A DTFT é a extensão periódica da FT, ou seja é uma composição de réplicas da transformada contínua, em que a réplica central corresponde as mesmas frequências da transformada contínua divididas pela frequência de amostragem [23, text].

Já para sinais de duração finita, é definida a Transformada Discreta de Fourier (DFT), que corresponde a amostras da DTFT entre 0 e 1, de acordo com a quantidade de amostras no tempo [23]. A DFT para um sinal pode ser obtida por

$$X(f) = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi \frac{f}{N}n}, \quad (2.16)$$

em que N corresponde a quantidade de amostras.

2.2.4.2 Transformada Z

A Transformada Z é uma generalização da Transformada de Fourier para sinais discretos [22] e é definida por

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] \cdot z^{-n}, \quad (2.17)$$

onde z é uma variável complexa. A Transformada Z é particularmente útil na análise e projeto de sistemas LIT em tempo discreto, pois permite a modelagem do sistema em termos de exponenciais complexas.

2.2.4.3 Relações entre a Transformada Z e a Transformada de Fourier

Em sistemas estáveis e causais, existe uma conexão direta entre a Transformada Z e a DTFT. A Transformada Z é definida no domínio complexo e pode ser avaliada ao longo de uma circunferência específica no plano z . Quando essa região inclui a circunferência unitária, caracterizada pela condição $z = e^{j2\pi f}$, onde f é a frequência normalizada entre valores de -0.5 e 0.5 , a Transformada de Fourier pode ser obtida como um caso especial da Transformada Z [22]. Em termos matemáticos, a substituição de z por $e^{j2\pi f}$ na Transformada Z fornece a resposta em frequência $H(f)$ do sistema, que descreve como o sistema responde a diferentes frequências. A Região de Convergência (ROC) da transformada Z é ilustrada na Figura 2.3, que mostra a área do plano complexo onde a Transformada Z converge.

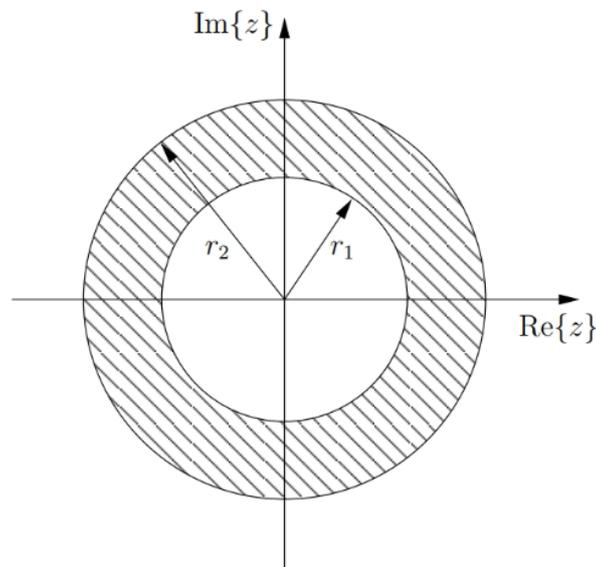


Figura 2.3. Região de Convergência da Transformada Z representada no plano complexo, com $r_1 = \lim_{n \rightarrow \infty} |x[n]|^{1/n}$ e $r_2 = \lim_{n \rightarrow -\infty} |x[n]|^{1/n}$. Imagem adaptada de [22].

A Transformada de Fourier de um sinal discreto é, portanto, obtida substituindo z

por $e^{j2\pi f}$ na expressão da Transformada Z do sinal. Assim, a expressão para a resposta em frequência $H(f)$ de um sistema pode ser escrita como

$$H(f) = X(z) \Big|_{z=e^{j2\pi f}} = \sum_{n=-\infty}^{\infty} x[n]e^{-j2\pi fn}, \quad (2.18)$$

onde $X(z)$ é a Transformada Z do sinal discreto $x[n]$, e $H(f)$ é a resposta em frequência do sistema.

Essa relação entre as duas transformadas, além das propriedades de sistemas LIT apresentadas, são essenciais para o processo de modelagem, visto que permitem uma análise completa tanto no domínio do tempo quanto no domínio da frequência. A Transformada Z possibilita a implementação de modelos no tempo discreto através da equação das diferenças, mostrada em detalhes no capítulo 3, enquanto a Transformada de Fourier facilita a caracterização do comportamento do sistema, oferecendo uma visão de como o sistema reage a diferentes frequências de entrada em termos de magnitude e fase.

3 MATERIAIS E MÉTODOS

Utilizando os conceitos de análise de sistemas lineares e invariantes no tempo, este trabalho propõe o desenvolvimento de algoritmos para obtenção de modelos digitais a partir da análise da resposta em frequência. É feita uma divisão em dois algoritmos distintos: um para a modelagem e outro para a emulação, ambos utilizando o mesmo *hardware*. Com isso, o dispositivo responsável pela modelagem também poderá emular os modelos, sendo necessário apenas ajustar a conexão dos equipamentos ao dispositivo.

3.1 Projeto de modelagem

A modelagem da resposta em frequência de um amplificador de áudio pode ser realizada a partir de sua função de transferência. Para isso, foi necessário desenvolver um dispositivo capaz de gerar sinais na entrada do amplificador e capturar os sinais de saída, visando obter a função de transferência. O dispositivo deve possuir um canal de saída compatível com a entrada do amplificador, para introduzir os sinais, e um canal de entrada conectado ao equipamento que grava a saída do amplificador. No entanto, para calcular a função de transferência com precisão, é necessário também capturar o sinal aplicado na entrada do amplificador por um segundo canal de gravação no dispositivo. Dessa forma, a função de transferência é determinada a partir de dois sinais gravados: um representando a saída do amplificador e outro a entrada. A Figura 3.1 apresenta o diagrama de blocos mostrando as conexões utilizadas para realizar o processo de modelagem.

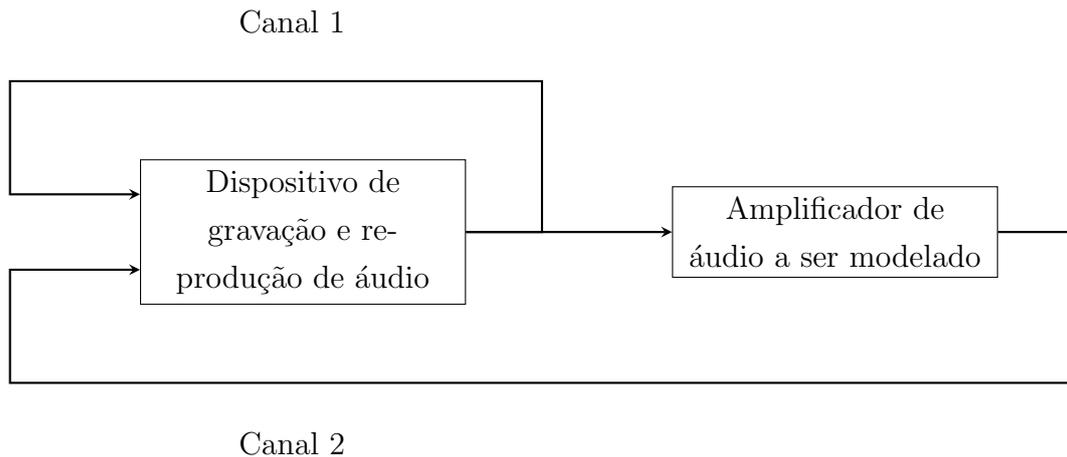


Figura 3.1. Diagrama de blocos das conexões utilizadas para modelagem de um amplificador de áudio, em que o mesmo dispositivo é utilizado para gerar os sinais de entrada e capturar os sinais de saída do amplificador.

Embora o sinal de entrada especificado seja ideal, o sinal gerado está sujeito as limitações do dispositivo. Por isso, utilizar o sinal gravado na entrada garante que o processo de modelagem considere exatamente o sinal aplicado ao amplificador, evitando influências do sistema de geração e assegurando a precisão na obtenção da função de transferência.

Além disso, para garantir que o resultado da modelagem seja aplicável em contextos musicais, os sinais aplicados devem ser senoidais, com frequências definidas dentro da faixa utilizada por equipamentos de áudio. Para atender a essas exigências, uma interface de áudio foi utilizada como gerador de sinais e dispositivo de gravação, a qual é apresentada na Figura 3.2.



Figura 3.2. Interface de áudio Presonus Studio 24c [4] utilizada para aquisição e reprodução dos sinais de áudio.

A interface de áudio Presonus Studio 24c [4] permite taxas de amostragem de até 192 kHz e resolução de até 24 bits. Ela possui dois canais de entrada com controle de ganho independente e um canal de saída estéreo, o que atende as especificações para aquisição e reprodução de sinais durante a modelagem.

3.1.1 Introdução de sinais senoidais a entrada do amplificador de áudio

Para iniciar o projeto de modelagem, é necessário utilizar o gerador de sinais para introduzir senoides ao amplificador, as quais possuem frequência e amplitude definidas. A amplitude das senoides permanece constante, mas o processo de introdução ocorre para diversas frequências contidas no espectro a ser modelado. Os sinais podem ser introduzidos de duas formas: um único sinal composto por uma soma de senoides de várias frequências ou vários sinais, cada um sendo uma senoide de frequência diferente, introduzidos um após o outro. O usuário pode selecionar entre essas duas abordagens e

definir a frequência mais grave, a mais aguda, a duração de cada sinal, e a quantidade de frequências a serem utilizadas.

De forma a garantir que as senoides sejam igualmente espaçadas em frequência, o algoritmo calcula a diferença entre a frequência mais alta e a mais baixa selecionadas pelo usuário e divide esse intervalo pelo número de senoides desejado. Assim, a diferença de frequência entre duas senoides consecutivas será constante, gerando um conjunto de frequências uniformemente distribuídas ao longo do espectro definido. Esse conjunto de frequências é então utilizado como entrada para a função geradora de senoides.

3.1.2 Aquisição das amostras dos sinais de áudio gerados pelo gerador de senoides antes e depois de passarem pelo amplificador de áudio

A gravação dos sinais ocorre de forma simultânea à geração das senoides pelo gerador de sinais, que é a própria saída da interface de áudio. As senoides geradas são injetadas nos dois canais da interface. O primeiro canal recebe diretamente o sinal gerado pela interface, enquanto o segundo canal está conectado a um microfone que capta o sinal após ele passar pelo amplificador e ser reproduzido por um alto-falante. Dessa forma, os dois sinais são armazenados simultaneamente, e as amostras podem ser salvas em um arquivo de áudio.

A configuração da taxa de amostragem e da resolução das amostras é baseada nas especificações do equipamento de emulação Sonic IR [1], que é um dos equipamentos utilizados para a avaliação do modelo. O Sonic IR suporta arquivos de resposta ao impulso amostrados a 44,1 kHz e com uma resolução de 24 bits, com 1024 pontos. O amplificador e o microfone utilizados são apresentados na Figura 3.3.



Figura 3.3. Amplificador Blackstar ID Core 20 v2 [5] e microfone dinâmico Santo Angelo SAS 57 [6] utilizados como sistema a ser modelado a partir da introdução de senoides ao amplificador e captura do sinal de saída pelo microfone.

O amplificador Blackstar ID Core 20 v2 é um combo de pré-amplificador, amplificador de potência e dois alto-falantes que entrega uma potência de 20 W [5]. Já o microfone Santo Angelo SAS 57 é um microfone dinâmico com resposta em frequência de 60 Hz até 14 kHz e impedância de 500 Ω [6].

3.1.3 Obtenção do modelo de resposta ao impulso finita (FIR)

Após a aquisição das amostras dos sinais de entrada e saída do sistema, a magnitude e a fase da resposta em frequência do sistema podem ser obtidas utilizando a Transformada de Fourier, conforme a equação:

$$H(f) = \frac{|H_s(f)|}{|H_e(f)|} \cdot e^{j(\theta_s - \theta_e)} = |H(f)| \cdot e^{j\theta}, \quad (3.1)$$

onde $|H(f)|$ é o módulo, obtido pela razão entre as amplitudes dos sinais de entrada e saída, e θ é a fase entre esses sinais.

Com essa informação, é possível determinar a resposta do sistema para cada frequência, o que permite a formulação de um sistema de equações no qual, para cada frequência f , a função de transferência $H(f)$ é dada por

$$H(f) = \sum_{n=0}^{N-1} h[n] \cdot e^{-j2\pi fn}. \quad (3.2)$$

Escrevendo essas equações em termos de multiplicação matricial, temos

$$H = M \cdot h, \quad (3.3)$$

onde M é a matriz de coeficientes que multiplicam o vetor h . Cada linha da matriz M representa os coeficientes que multiplicam $h[n]$ para calcular $H(f)$ em uma frequência específica. No caso de um modelo de resposta ao impulso finita (FIR), esses coeficientes são exponenciais complexas, e a matriz M pode ser expressa como:

$$M = \begin{bmatrix} 1 & e^{j2\pi f_1} & e^{j2\pi f_1 \cdot 2} & \dots & e^{j2\pi f_1 \cdot N-1} \\ 1 & e^{j2\pi f_2} & e^{j2\pi f_2 \cdot 2} & \dots & e^{j2\pi f_2 \cdot N-1} \\ 1 & e^{j2\pi f_3} & e^{j2\pi f_3 \cdot 2} & \dots & e^{j2\pi f_3 \cdot N-1} \\ 1 & e^{j2\pi f_4} & e^{j2\pi f_4 \cdot 2} & \dots & e^{j2\pi f_4 \cdot N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{j2\pi f_m} & e^{j2\pi f_m \cdot 2} & \dots & e^{j2\pi f_m \cdot N-1} \\ 1 & e^{-j2\pi f_1} & e^{-j2\pi f_1 \cdot 2} & \dots & e^{-j2\pi f_1 \cdot N-1} \\ 1 & e^{-j2\pi f_2} & e^{-j2\pi f_2 \cdot 2} & \dots & e^{-j2\pi f_2 \cdot N-1} \\ 1 & e^{-j2\pi f_3} & e^{-j2\pi f_3 \cdot 2} & \dots & e^{-j2\pi f_3 \cdot N-1} \\ 1 & e^{-j2\pi f_4} & e^{-j2\pi f_4 \cdot 2} & \dots & e^{-j2\pi f_4 \cdot N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi f_m} & e^{-j2\pi f_m \cdot 2} & \dots & e^{-j2\pi f_m \cdot N-1} \end{bmatrix} \quad (3.4)$$

Utilizando um modelo de mesma ordem dos utilizados pelo pedal Sonic IR, é necessário possuir medidas de amplitude e fase para 1024 frequências diferentes. A resposta ao impulso finita $h[n]$ pode ser obtida a partir da multiplicação da inversa da matriz M pela matriz

$$H = \begin{bmatrix} H^*(f_1) \\ H^*(f_2) \\ H^*(f_3) \\ \vdots \\ H^*(f_m) \\ H(f_1) \\ H(f_2) \\ H(f_3) \\ \vdots \\ H(f_m) \end{bmatrix} \quad (3.5)$$

Dessa forma, a matriz com os coeficientes de $h[n]$ pode ser obtida por

$$h = M^{-1}H, \quad (3.6)$$

para o caso em que a matriz M é quadrada. Caso o sistema de equações seja sobre-determinado, ou seja, possua mais equações do que incógnitas, a matriz M pode ser multiplicada pela matriz M^h , em que

$$h = (M^h M)^{-1}(M^h H). \quad (3.7)$$

A matriz M^h é a matriz hermitiana, ou seja, a transposta conjugada da matriz M . Ela é utilizada para obter a solução ótima que minimiza o erro em termo do erro quadrático médio [24], dessa forma é recomendado que sempre seja utilizado um sistema sobredeterminado, resolvido por essa abordagem, para que o erro causado pelos ruídos das medições seja minimizado. A matriz h representa o modelo FIR, o qual pode ser utilizado para emular o sistema modelado.

3.1.4 Obtenção do modelo de resposta ao impulso infinita (IIR)

Já para o caso de resposta ao impulso infinita (IIR), é preciso saber a relação entre a Transformada Z e a Transformada de Fourier, apresentada no capítulo 2, e obter a função de transferência do sistema no domínio Z , visto que

$$X[j\omega] = X[z], \quad (3.8)$$

onde $z = e^{j\omega}$. Essa substituição é possível visto que ao obter as medidas da resposta em frequência do sistema a partir da Transformada de Fourier, a região de convergência inclui a circunferência de raio unitário. Portanto, utilizam-se os valores da Transformada de Fourier para montar a função de transferência expressa como

$$H(f_m) = \frac{N_0 + N_1 e^{-j2\pi f_m} + N_2 e^{-j2\pi f_m \cdot 2} + \dots + N_o e^{-j2\pi f_m \cdot o}}{D_0 + D_1 e^{-j2\pi f_m} + D_2 e^{-j2\pi f_m \cdot 2} + \dots + D_o e^{-j2\pi f_m \cdot o}}, \quad (3.9)$$

para uma dada frequência f_m e para uma dada ordem o .

Após multiplicar os termos do denominador em ambos os lados da equação, temos que

$$\begin{aligned}
N_0 + N_1 e^{-j2\pi f_m} + N_2 e^{-j2\pi f_m \cdot 2} + \dots + N_o e^{-j2\pi f_m \cdot o} &= D_0 H(f_m) \\
+ D_1 e^{-j2\pi f_m} H(f_m) + D_2 e^{-j2\pi f_m \cdot 2} H(f_m) + \dots + D_o e^{-j2\pi f_m \cdot o} H(f_m). &
\end{aligned} \tag{3.10}$$

Fixando $D_0 = 1$ para não obtermos um sistema homogêneo, temos que

$$\begin{aligned}
N_0 + N_1 e^{-j2\pi f_m} + N_2 e^{-j2\pi f_m \cdot 2} + \dots + N_o e^{-j2\pi f_m \cdot o} - D_1 e^{-j2\pi f_m} H(f_m) \\
+ D_2 e^{-j2\pi f_m \cdot 2} H(f_m) + \dots + D_o e^{-j2\pi f_m \cdot o} H(f_m) &= H(f_m).
\end{aligned} \tag{3.11}$$

Caso não fosse fixado $D_0 = 1$ não seria possível isolar $H(f_m)$ e todas as equações estariam igualadas a 0, o que torna o sistema homogêneo. Por fim, a matriz M resultante é dada por

$$M = \begin{bmatrix}
1 & e^{j2\pi f_1} & \dots & e^{j2\pi f_1 \cdot o} & e^{j2\pi f_1} H^*(f_1) & \dots & e^{j2\pi f_1 \cdot o} H^*(f_1) \\
1 & e^{j2\pi f_2} & \dots & e^{j2\pi f_2 \cdot o} & e^{j2\pi f_2} H^*(f_2) & \dots & e^{j2\pi f_2 \cdot o} H^*(f_2) \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
1 & e^{j2\pi f_m} & \dots & e^{j2\pi f_m \cdot o} & e^{j2\pi f_m} H^*(f_m) & \dots & e^{j2\pi f_m \cdot o} H^*(f_m) \\
1 & e^{-j2\pi f_1} & \dots & e^{-j2\pi f_1 \cdot o} & e^{-j2\pi f_1} H(f_1) & \dots & e^{-j2\pi f_1 \cdot o} H(f_1) \\
1 & e^{-j2\pi f_2} & \dots & e^{-j2\pi f_2 \cdot o} & e^{-j2\pi f_2} H(f_2) & \dots & e^{-j2\pi f_2 \cdot o} H(f_2) \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
1 & e^{-j2\pi f_m} & \dots & e^{-j2\pi f_m \cdot o} & e^{-j2\pi f_m} H(f_m) & \dots & e^{-j2\pi f_m \cdot o} H(f_m)
\end{bmatrix} \tag{3.12}$$

É possível observar que a matriz M é ligeiramente diferente para o IIR em relação ao FIR, visto que existem termos multiplicados pelo valor de $H(f)$ para cada frequência. Para obter os coeficientes do modelo, é realizado o mesmo procedimento do FIR, resolver o sistema linear a partir da multiplicação da inversa da matriz M com a matriz H , que é a mesma do caso FIR.

A grande diferença entre os dois modelos está no fato de que o modelo IIR utiliza coeficientes que multiplicam amostras de saídas passadas para obter uma nova saída, gerando uma realimentação. Esse aspecto permite que modelos IIR possuem ordens menores em comparação aos modelos FIR, podendo utilizar menos de 1024 termos como estabelecido para o FIR.

A Figura 3.4 sintetiza o algoritmo utilizado para o projeto de modelagem.

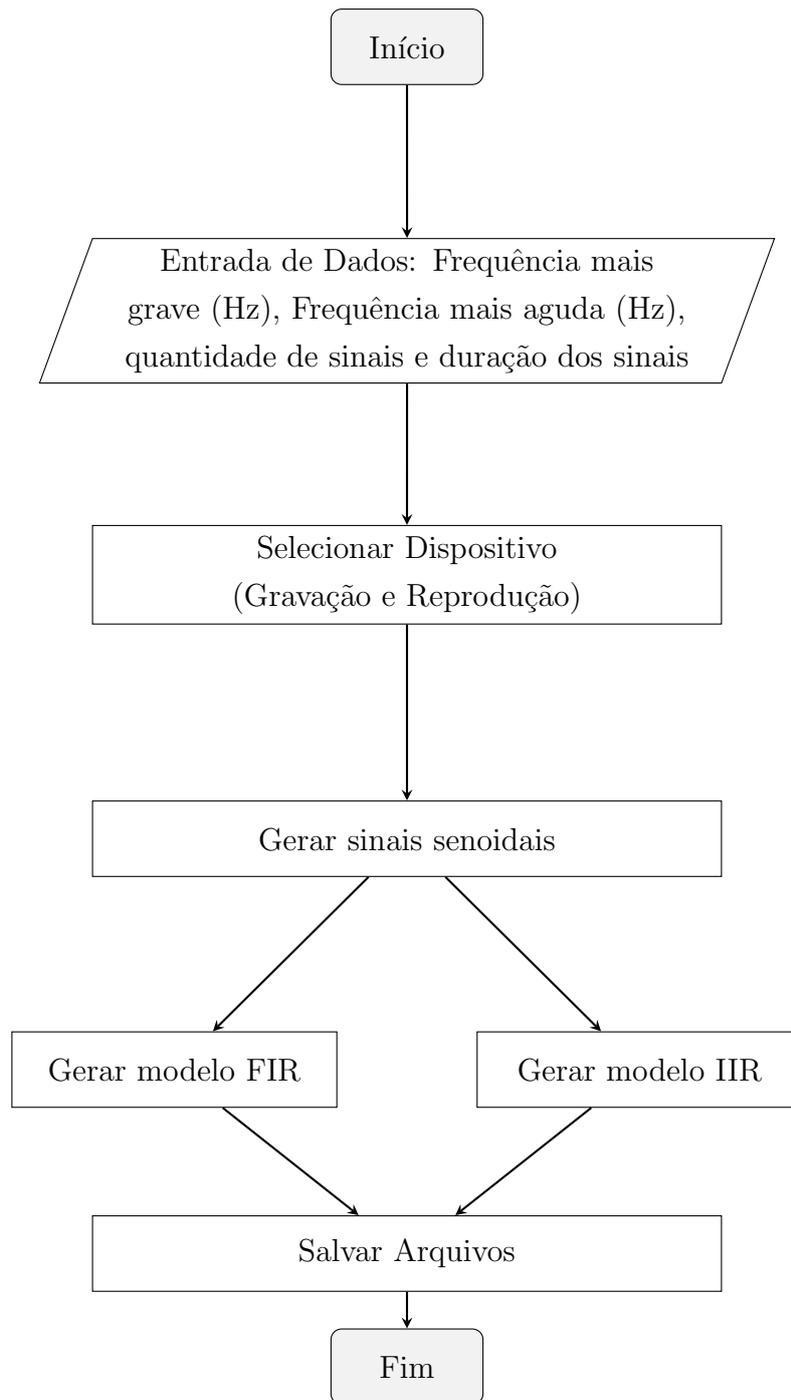


Figura 3.4. Fluxograma do algoritmo de modelagem de amplificadores de áudio, que permite o usuário definir os parâmetros dos sinais, selecionar os dispositivos de gravação e reprodução, e por fim gera e salva os modelos FIR e IIR.

A Figura 3.4 evidencia a flexibilidade dos parâmetros do sinal introduzido no sistema a ser modelado escolhidos pelo usuário que executa o algoritmo.

3.1.5 Avaliação dos modelos obtidos

A avaliação dos modelos é realizada aplicando um sinal de áudio gravado tanto aos modelos quanto ao sistema modelado. A implementação dos modelos é feita de maneiras diferentes para cada tipo. Para o modelo FIR, a saída é obtida através da convolução entre o sinal gravado e a resposta ao impulso. A operação de convolução é expressa pela fórmula:

$$y[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n - k], \quad (3.13)$$

onde $x[n]$ é o sinal de entrada, $h[n]$ é a resposta ao impulso e $y[n]$ é a saída do modelo após a convolução.

Além da convolução, o modelo IIR pode ser implementado a partir da equação de diferenças, que requer o armazenamento das amostras de saídas anteriores devido à sua estrutura recursiva. A equação geral para o modelo IIR é dada por

$$y[n] = \sum_{k=0}^M N_k x[n - k] - \sum_{j=1}^N D_j y[n - j], \quad (3.14)$$

em que N_k e D_j são os coeficientes do modelo, $x[n]$ é o sinal de entrada e $y[n]$ é a saída do modelo. Esta fórmula reflete como as saídas passadas são utilizadas para calcular a saída atual.

Para quantificar a diferença entre as saídas do modelo e do sistema modelado, calculamos o Erro Quadrático Médio (MSE). O MSE é obtido por

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (3.15)$$

onde y_i é o valor real da saída, \hat{y}_i é o valor estimado pelo modelo e N é o número total de amostras.

Adicionalmente, a qualidade da saída do modelo é avaliada pela Relação Sinal-Ruído (SNR), que quantifica a qualidade da saída em comparação com o ruído presente. A SNR é dada por

$$\text{SNR} = 10 \log_{10} \left(\frac{P_{\text{sinal}}}{P_{\text{ruído}}} \right), \quad (3.16)$$

onde P_{sinal} é a potência do sinal obtido pelo sistema real e $P_{\text{ruído}}$ é a potência do ruído, obtida pela diferença de potência entre o sinal gerado pelo sistema e o sinal gerado pelo modelo. As potências do sinal e do ruído são calculadas como a média dos quadrados das amostras correspondentes.

Essas métricas proporcionam uma avaliação da performance dos modelos em comparação com o sistema modelado. O MSE quantifica a discrepância entre as saídas esperadas e as obtidas, já a SNR mede a qualidade da saída em relação ao ruído. Elas não apenas validam a precisão dos resultados, mas também asseguram que o modelo seja útil em aplicações práticas, refletindo com precisão o comportamento do sistema real.

3.2 Projeto de emulação

A emulação do sistema ocorre ao aplicar o modelo obtido em um contexto prático, substituindo o sistema original. Utiliza-se o mesmo *hardware* do projeto de modelagem, mas sem a necessidade de amplificador e microfone. Apenas a interface de áudio é necessária, que agora é responsável por capturar o sinal que deveria ser introduzido no sistema, como o sinal de uma guitarra elétrica. O sinal resultante é então disponível nas saídas da interface de áudio e pode ser encaminhado para um alto-falante ou fone de ouvido para reprodução.

3.2.1 Emulação a partir da convolução do sinal de entrada e a resposta ao impulso do modelo

O algoritmo de emulação utiliza a convolução entre dois sinais: o sinal do instrumento e a resposta ao impulso finita obtida pela modelagem. Para a realização dessa convolução, o sinal de resposta ao impulso deve estar armazenado na memória do dispositivo responsável pela emulação, permitindo que a operação seja realizada de maneira adequada e eficiente. O sinal de entrada é adquirido à medida que é gerado, e a convolução é aplicada conforme o sinal é processado.

O sinal de saída gerado deve corresponder exatamente ao tamanho do sinal de entrada, isso significa que apenas o resultado parcial da convolução, que se ajusta ao tamanho do sinal de entrada, será reproduzido na saída. O tamanho total do sinal resultante da convolução é calculado como a soma do tamanho do sinal de entrada e da resposta ao impulso, menos um. As amostras excedentes da convolução são armazenadas para o processamento dos dados do próximo sinal de entrada.

A Figura 3.5 apresenta um fluxograma do algoritmo de emulação que implementa a convolução.

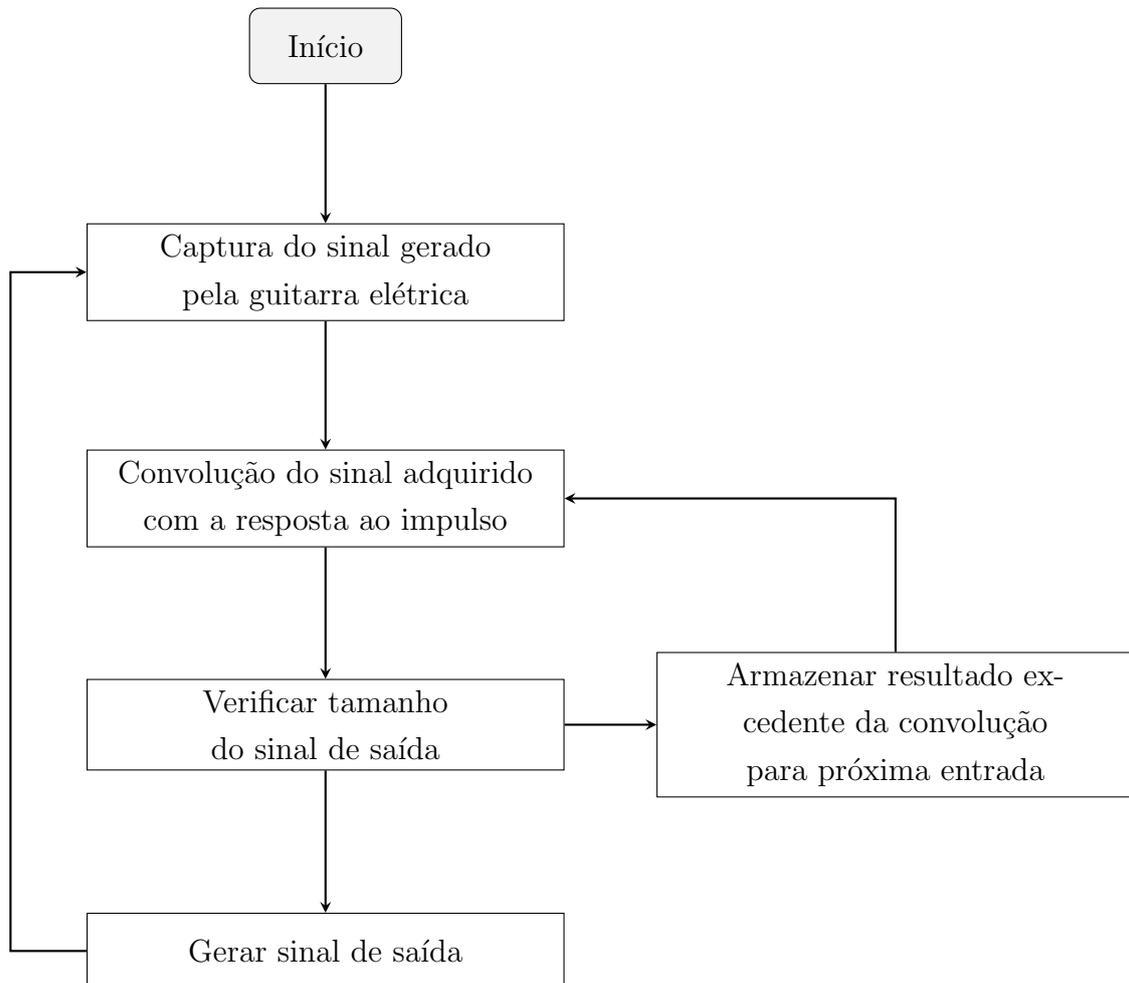


Figura 3.5. Diagrama do processo de emulação a partir da convolução do sinal de entrada e da resposta ao impulso finita obtida pela modelagem.

É possível observar que o processo de emulação apresentado na Figura 3.5 se repete à medida que novas amostras do sinal de entrada são capturadas.

3.2.2 Emulação a partir da equação das diferenças

A equação das diferenças apresenta os pesos, em termos das constantes a serem multiplicadas por cada amostra tanto para o sinal de entrada, quanto para o sinal de saída. Dessa forma, é necessário que o processo de aquisição armazene as entradas e saídas passadas, visto que as mesmas serão utilizadas para as saídas futuras. Vale ressaltar que no início do processo, o valor das entradas e saídas passadas serão adotadas como 0.

O algoritmo responsável pela emulação do sistema adquire as amostras do sinal de

áudio e realizar as somas e multiplicações da equação das diferenças, além de armazenar as entradas e saídas em *buffers* de memória. O armazenamento das amostras pode ser feito utilizando um vetor com quantidade de elementos definida pela ordem do modelo, em que há um deslocamento das amostras a medida uma nova amostras é adquirida, semelhante a um registrador.

A Figura 3.6 apresenta um fluxograma do algoritmo de emulação que implementa a equação das diferenças.

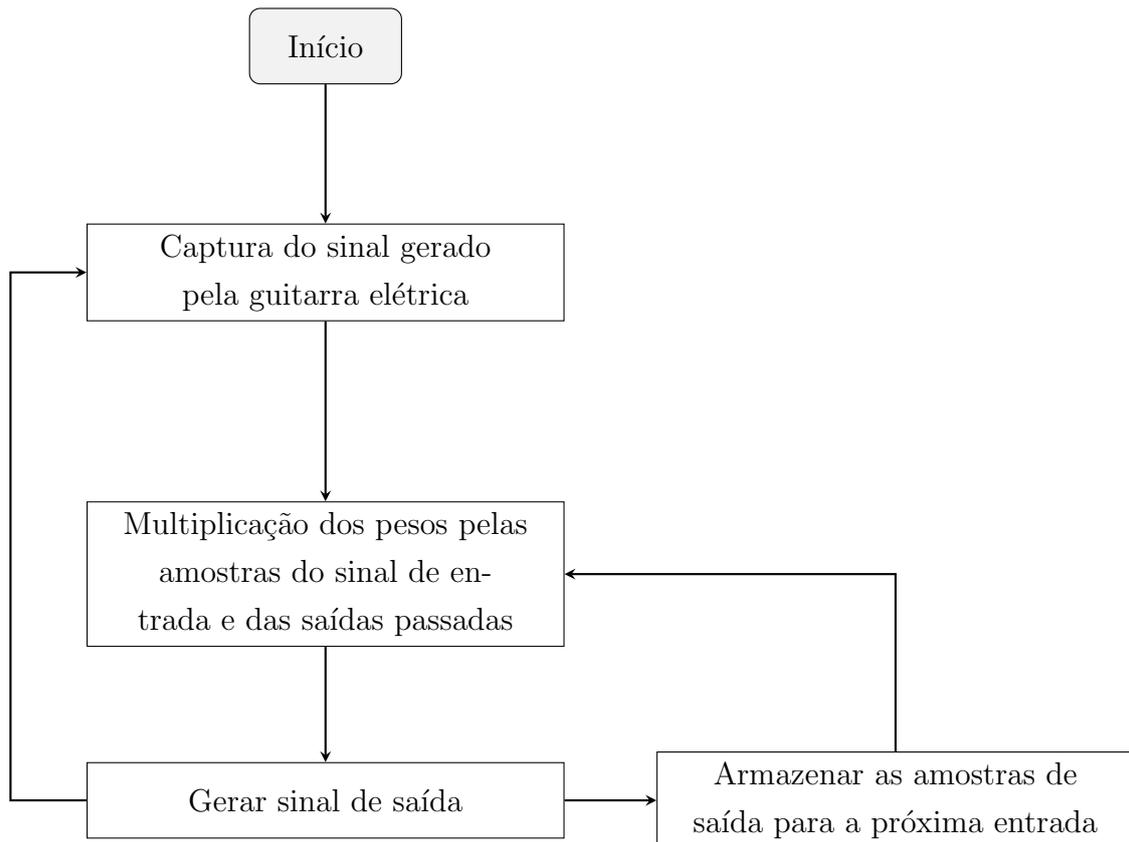


Figura 3.6. Diagrama do processo de emulação a partir da equação das diferenças, utiliza para os modelos IIR.

Novamente, o processo de emulação apresentado na Figura 3.6 se repete à medida que novas amostras do sinal de entrada são captadas.

3.2.3 Avaliação da Latência do Processo de Emulação

A avaliação da latência é fundamental para medir a eficiência dos métodos de emulação, definida pelo tempo entre a captura do sinal de entrada e a geração do sinal de saída. A latência pode variar significativamente entre os métodos de convolução e equação das diferenças, devido às diferenças na complexidade dos algoritmos e no processamento en-

volvido.

Para o método de convolução, a latência depende do tempo necessário para executar a operação de convolução entre o sinal de entrada e a resposta ao impulso. A convolução pode ser realizada no domínio do tempo ou então sua operação pode ser substituída por uma multiplicação no domínio da frequência usando a Transformada de Fourier.

No método baseado na equação das diferenças, a latência é determinada pelo tempo necessário para calcular as saídas com base nas entradas e saídas passadas, bem como pelo tempo para atualizar os buffers de memória. A equação das diferenças tende a ter uma latência menor, especialmente para modelos de ordem baixa, devido à menor quantidade de amostras a serem processadas comparado à convolução.

Essencialmente, a latência será medida em milissegundos e obtida a partir da diferença entre o sinal de saída e o de entrada. O objetivo é garantir que a latência não ultrapasse os 20 milissegundos, como mencionado anteriormente na seção 1.2.

Os algoritmos descritos nesse capítulo para modelagem e para emulação foram implementados nas linguagens Python e C++, respectivamente. A apresentação e descrição das principais funções implementadas podem ser vista no capítulo 6, e os códigos completos estão no repositório do projeto [25]. As principais bibliotecas utilizadas para implementação do algoritmo de modelagem foram a NumPy [26], SciPy [27], Soundfile [28] e Sounddevice [29], e para o algoritmo de emulação foram a PortAudio [30] e a Thread [31].

A escolha da linguagem Python para o algoritmo de modelagem foi baseada na disponibilidade de materiais e suporte às funções utilizadas em projetos de processamento de sinais, além da facilidade para implementação de funções causada pela simplicidade da sintaxe da linguagem. Já para a emulação, como há um limite estabelecido para o tempo de execução, optou-se utilizar a linguagem C++ por, ao contrário do Python, ser uma linguagem compilada e também ser amplamente utilizada para desenvolvimento de programas que lidam diretamente com recursos de *hardware* e do sistema operacional.

4 RESULTADOS E DISCUSSÕES

4.1 Resultados da modelagem

O processo de modelagem foi realizado para dois sistemas. Inicialmente, foram gerados modelos FIR para os arquivos do pedal Sonic IR e em seguida para o sistema apresentado na Figura 3.3. Essa abordagem permite validar e avaliar os resultados do algoritmo de modelagem, uma vez que os resultados esperados para o primeiro caso são os próprios modelos presentes no pedal. As conexões utilizadas para aquisição e reprodução de sinais ao pedal é mostrada na Figura 4.1.



Figura 4.1. Conexões utilizadas para processo de modelagem do pedal Sonic IR. A saída da placa de áudio é conectada ao pedal ABY Box que divide a entrada em duas saídas: um ligada ao canal 1 da placa de áudio e, outra ao pedal Sonic IR. A saída do pedal é ligada no canal 2 da placa de áudio.

A montagem apresentada segue a mesma conexão do diagrama de blocos da Figura

3.1, apenas alterando o sistema modelado.

4.1.1 Resultados da modelagem dos arquivos presentes no pedal Sonic IR

O pedal Sonic IR possui um total de 11 modelos FIR distintos, para os quais não há muitas informações disponíveis sobre os sistemas modelados. Não são fornecidos detalhes sobre o pré-amplificador ou o microfone utilizados durante a modelagem, apenas informações sobre o gabinete e a quantidade e tamanho dos alto-falantes.

O primeiro arquivo presente no Sonic IR é o “Champion 1x8”, como ilustrado na Figura 4.2.

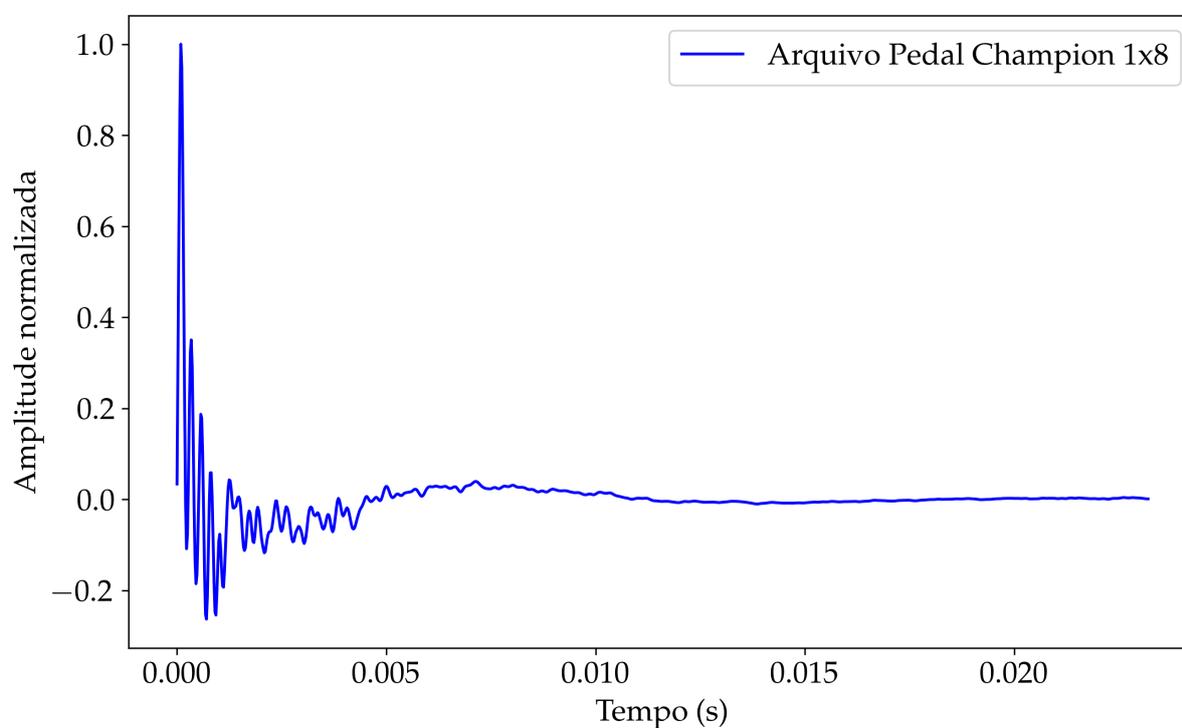


Figura 4.2. Arquivo “01 Champion 1x8.wav” extraído do pedal Sonic IR, que emula as características sonoras de um gabinete Fender Champion 1x8.

O resultado obtido para este modelo é apresentado na Figura 4.3.

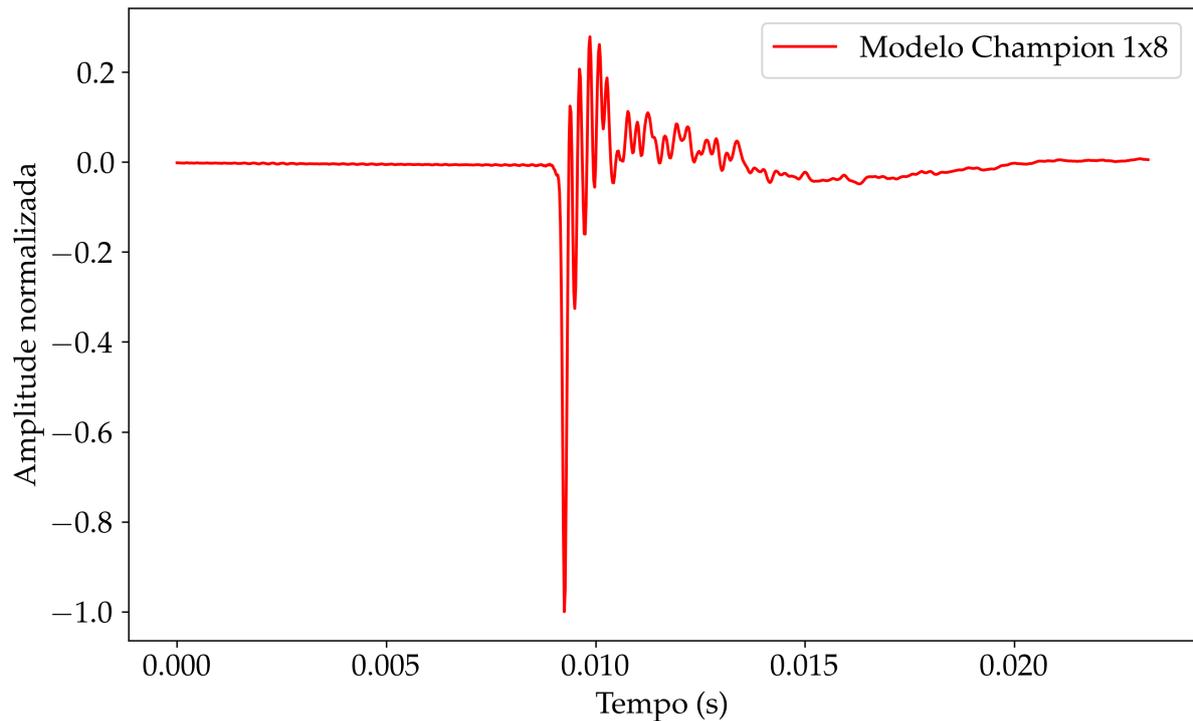


Figura 4.3. Modelo FIR obtido do sistema formado apenas pelo pedal Sonic IR e com o arquivo “01 Champion 1x8.wav” selecionado para emulação.

Observa-se um atraso no modelo obtido em relação ao arquivo presente no pedal, além de uma variação de fase entre os dois sinais. Essa diferença é atribuída ao fato de que o pedal Sonic IR realiza a emulação através de processamento digital, introduzindo uma latência no processo. Atrasos semelhantes e inversões de fase foram observados em todos os modelos adquiridos que utilizam o pedal Sonic IR. As Figuras 4.4 e 4.5 mostram os dois sinais e a magnitude da resposta em frequência sobrepostos, respectivamente. Em vermelho, está o modelo apresentado na Figura 4.3 com atraso e inversão de fase compensados, e em azul, o arquivo presente no pedal, mostrado na Figura 4.2.

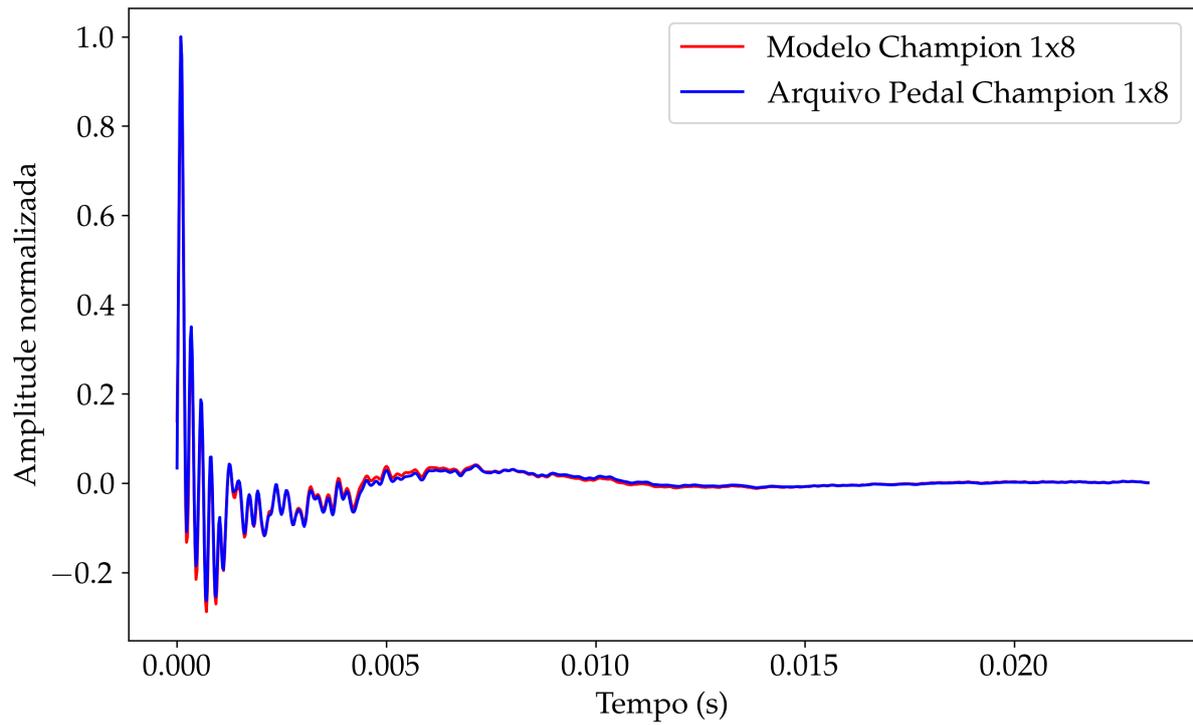


Figura 4.4. Modelo FIR obtido do sistema formado apenas pelo pedal Sonic IR com o arquivo “01 Champion 1x8.wav” selecionado, com atraso e inversão de fase compensados. O sinal do modelo foi sobreposto ao sinal do próprio sistema, o arquivo “01 Champion 1x8.wav”.

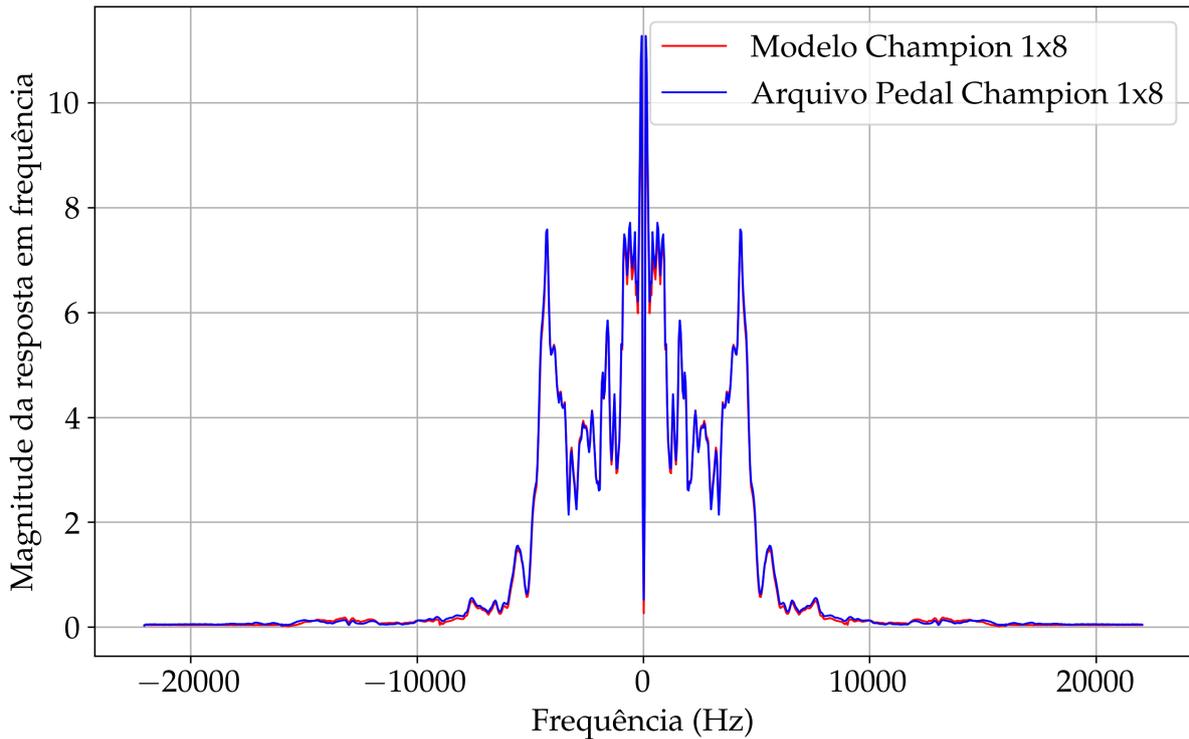


Figura 4.5. Magnitude da resposta em frequência do modelo FIR obtido do sistema formado apenas pelo pedal Sonic IR com o arquivo “01 Champion 1x8.wav” selecionado, com atraso e inversão de fase compensados. A resposta em frequência do modelo foi sobreposta à própria resposta do sistema, o arquivo “01 Champion 1x8.wav”.

Os resultados mostram grande semelhança visual entre o modelo FIR obtido e o arquivo presente no pedal Sonic IR, sendo possível observar que todo o espectro presente no arquivo do pedal foi modelado. O modelo foi obtido usando um sinal composto pela soma de várias senoides espaçadas em 10 Hz cada uma que variam de 40 Hz até 16 kHz, e a saída do pedal foi conectada diretamente a entrada da placa de aquisição.

Os mesmos procedimentos de modelagem e comparação foram realizados para todos os 11 arquivos disponíveis no pedal, e a Tabela 4.1 apresenta o erro quadrático médio e a relação sinal-ruído para cada um dos arquivos analisados.

Arquivo	MSE	SNR (dB)
01 Champion 1x8.wav	4.9468×10^{-8}	33.1
02 Voks 2x12.wav	1.4558×10^{-7}	28.4
03 TR65 2x12.wav	3.4873×10^{-8}	34.6
04 UK 2550 2x12.wav	8.2382×10^{-8}	30.8
05 BM59 4x10.wav	1.2457×10^{-7}	29.0
06 Tanger 4x12.wav	2.9339×10^{-7}	25.3
07 UK 1960 4x12.wav	7.9619×10^{-8}	31.0
08 6505 4x12.wav	2.6432×10^{-7}	25.8
09 Mebo 4x12.wav	5.7519×10^{-8}	32.4
10 M Bass 2x10.wav	6.8938×10^{-8}	31.6
11 A Bass 8x10.wav	3.3112×10^{-7}	24.8

Tabela 4.1. Valores do MSE e SNR para os arquivos de áudio provenientes do pedal Sonic IR e os modelos FIR obtidos. Os arquivos listados na Tabela são comparados entre os sinais originais do pedal e os sinais modelados pelos arquivos FIR correspondentes.

A Tabela 4.1 permite avaliar a qualidade dos modelos obtidos a partir dos sinais processados. Antes dos cálculos, os sinais foram normalizados, permitindo uma comparação entre eles por estarem na mesma ordem de grandeza. O Erro Quadrático Médio quantifica o desvio entre o sinal original e o sinal gerado pelo modelo, e quanto menor o valor de MSE, mais próximo o sinal gerado está do original. Na análise dos valores, o menor MSE foi observado no arquivo “3 TR65 2x12.wav”, indicando uma boa proximidade entre o sinal real e o modelo, enquanto o maior valor de MSE foi registrado para “11 A Bass 8x10.wav”, embora ainda dentro de uma margem aceitável de erro. Os erros quadráticos foram obtidos na ordem de 10^{-7} e 10^{-8} , refletindo em erros percentuais de 0.03% e 0.01%. Os erros estão em uma margem aceitável, que os valores das amostras variam entre -1 e 1 .

Já a SNR avalia a relação entre a potência do sinal e a do ruído. Valores mais altos de SNR indicam que o sinal do modelo se destaca significativamente em relação ao ruído. O maior valor de SNR foi encontrado no arquivo “03 TR65 2x12.wav”, sugerindo uma preservação da qualidade do sinal com relação ao ruído. O menor valor de SNR foi observado em “11 A Bass 8x10.wav”, mas mesmo esse menor valor indica que a potência do sinal é mais de 100 vezes maior que a do ruído, conforme a escala logarítmica utilizada.

Foi feita também a modelagem IIR para as mesmas medidas utilizadas para os modelos FIR, em que a ordem dos modelos foi definida de forma arbitrária. A Tabela 4.2 apresenta os resultados de MSE e SNR para os modelos IIR de ordem quatro, calculados

a partir da resposta em frequência dos modelos e do arquivo do pedal.

Arquivo	MSE	SNR (dB)
01 Champion 1x8.wav	5.9542×10^{-5}	2.3
02 Voks 2x12.wav	8.3799×10^{-5}	0.8
03 TR65 2x12.wav	8.6030×10^{-5}	0.7
04 UK 2550 2x12.wav	1.2176×10^{-4}	-0.9
05 BM59 4x10.wav	3.3983×10^{-5}	4.7
06 Tanger 4x12.wav	1.3012×10^{-4}	-1.1
07 UK 1960 4x12.wav	7.7268×10^{-5}	1.1
08 6505 4x12.wav	5.2529×10^{-5}	2.8
09 Mebo 4x12.wav	5.0562×10^{-5}	3.0
10 M Bass 2x10.wav	1.1369×10^{-4}	-0.6
11 A Bass 8x10.wav	2.2203×10^{-5}	6.5

Tabela 4.2. Valores do MSE e SNR para a saída obtida pelo modelo IIR de 4ª ordem em comparação com os arquivos do pedal Sonic IR.

É possível observar uma diferença entre os resultados obtidos para cada modelo IIR. Em geral, os modelos IIR apresentaram valores de MSE e SNR que deixam claros a presença de erros e a inadequação em representar o sistema, a partir da análise numérica feita. Os valores de SNR negativos evidenciam que para alguns modelos a potência do ruído é maior que a potência do sinal. A Figura 4.6 apresenta o espectro da magnitude obtida do modelo IIR de quarta ordem e do arquivo “01 Champion 1x8.wav”.

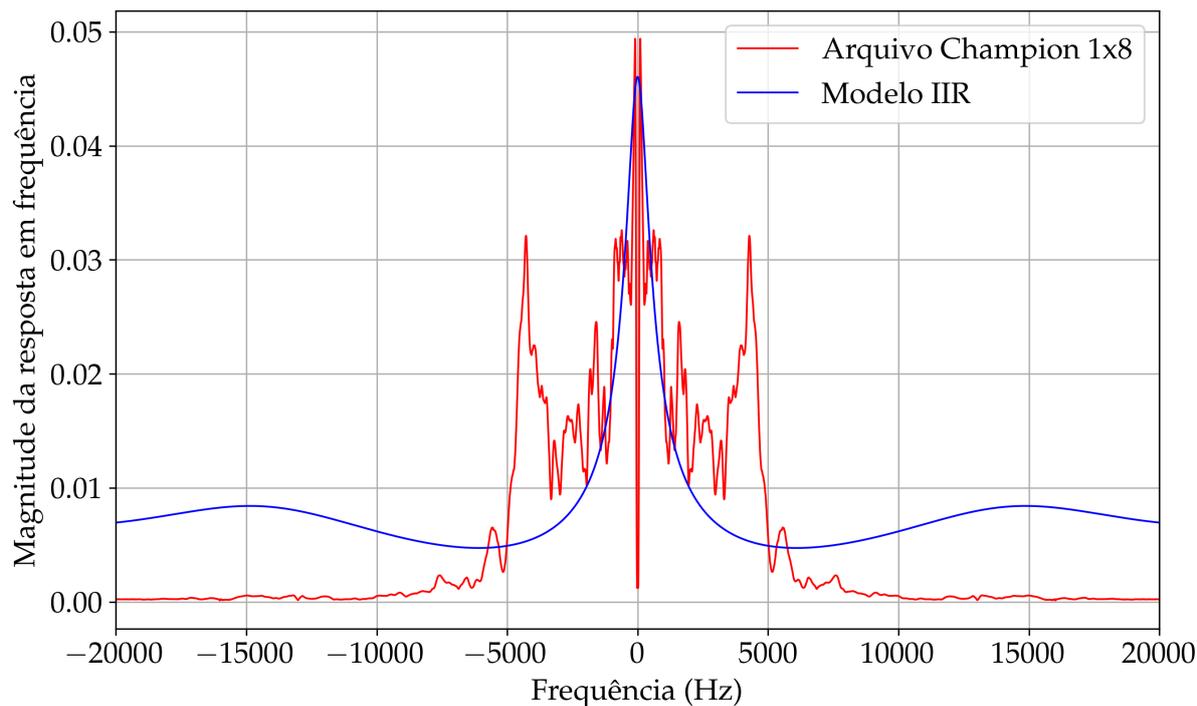


Figura 4.6. Comparação entre a magnitude da resposta em frequência do modelo IIR de quarta ordem, em azul, e o arquivo “01 Champion 1x8.wav” do pedal, em vermelho.

A Figura 4.6 mostra que a resposta em frequência obtida pelo modelo IIR está bem diferente em relação ao arquivo, o que confirma os valores obtidos pela tabela 4.2. Contudo, ao analisar a resposta em frequência do modelo IIR obtido do arquivo “11 A Bass 8x10.wav” presente na Figura 4.7, que obteve o maior SNR, é possível observar que, por mais que a resposta em frequência não possua uma variação semelhante em relação ao arquivo, há uma coincidência maior entre os picos na faixa de frequência observada.

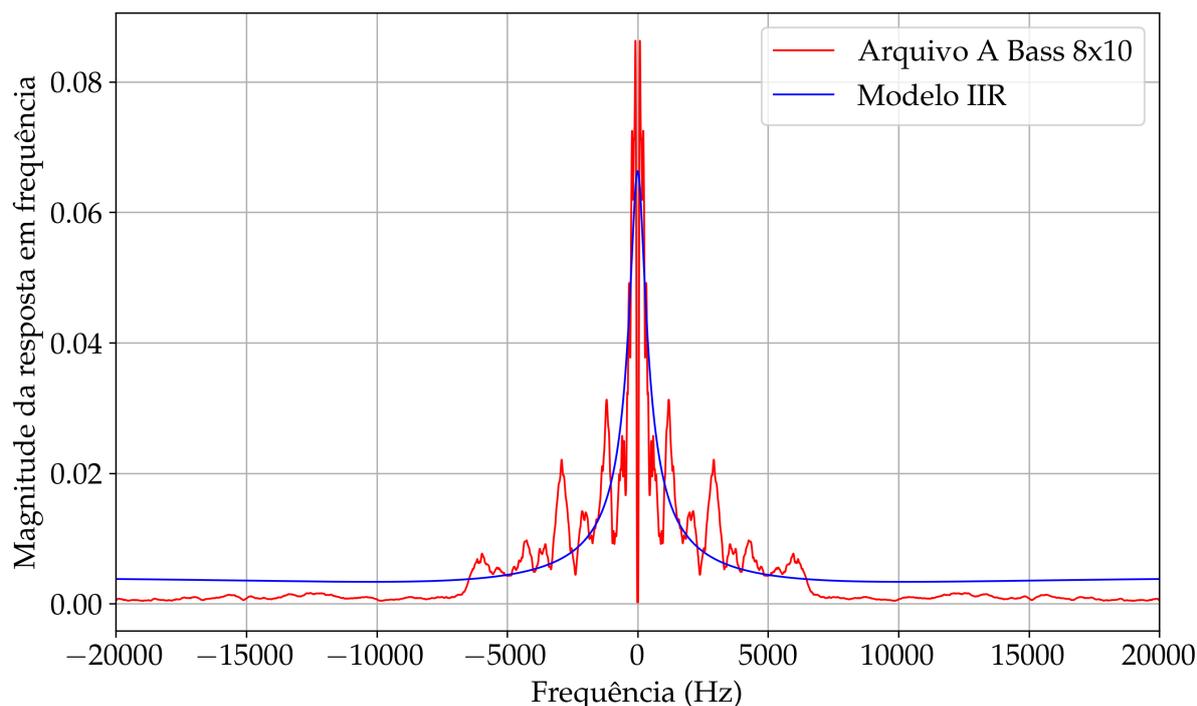


Figura 4.7. Comparação entre a magnitude da resposta em frequência do modelo IIR de quarta ordem, em azul, e o arquivo “11 A Bass 8x10.wav” do pedal, em vermelho.

Por conta dessa maior semelhança, não é possível afirmar com clareza que o modelo 4.7 não pode emular o sistema de forma fidedigna. Porém em geral, os resultados para os modelos IIR sugerem uma necessidade maior de medidas e uma reavaliação de valores de ordem utilizados para a modelagem. O foco principal do trabalho não é realizar uma comparação entre os modelos IIR e FIR, visto que para os modelos IIR não foram efetuadas comparações com modelos comerciais.

4.1.2 Resultados da Modelagem de Amplificadores de Áudio

Após a validação dos métodos com os arquivos do pedal Sonic IR, foi realizada a modelagem do amplificador Blackstar ID Core 20 v2. Este amplificador possui um estágio de pré-amplificação e amplificação de potência. A saída é distribuída entre dois alto-falantes, cujos sinais foram captados utilizando o microfone Santo Angelo SAS57, conforme mostrado na Figura 3.3. Foram obtidos modelos FIR e IIR para a configuração do amplificador com os potenciômetros de ganho, volume e equalização ajustados para a posição média, e o potenciômetro “Voice” configurado na posição “Clean Warm”, como ilustrado na Figura 4.8.



Figura 4.8. Configuração do amplificador Blackstar ID Core 20 v2 para modelagem do canal “Clean Warm”.

Os modelos obtidos seguem as mesmas especificações dos modelos gerados a partir dos arquivos do pedal Sonic IR. Um sinal composto por senoides de 40 Hz a 16 kHz, espaçadas por 10 Hz, foi introduzido. O modelo FIR obtido é apresentado na Figura 4.9, e a magnitude da resposta em frequência do modelo está na Figura 4.10.

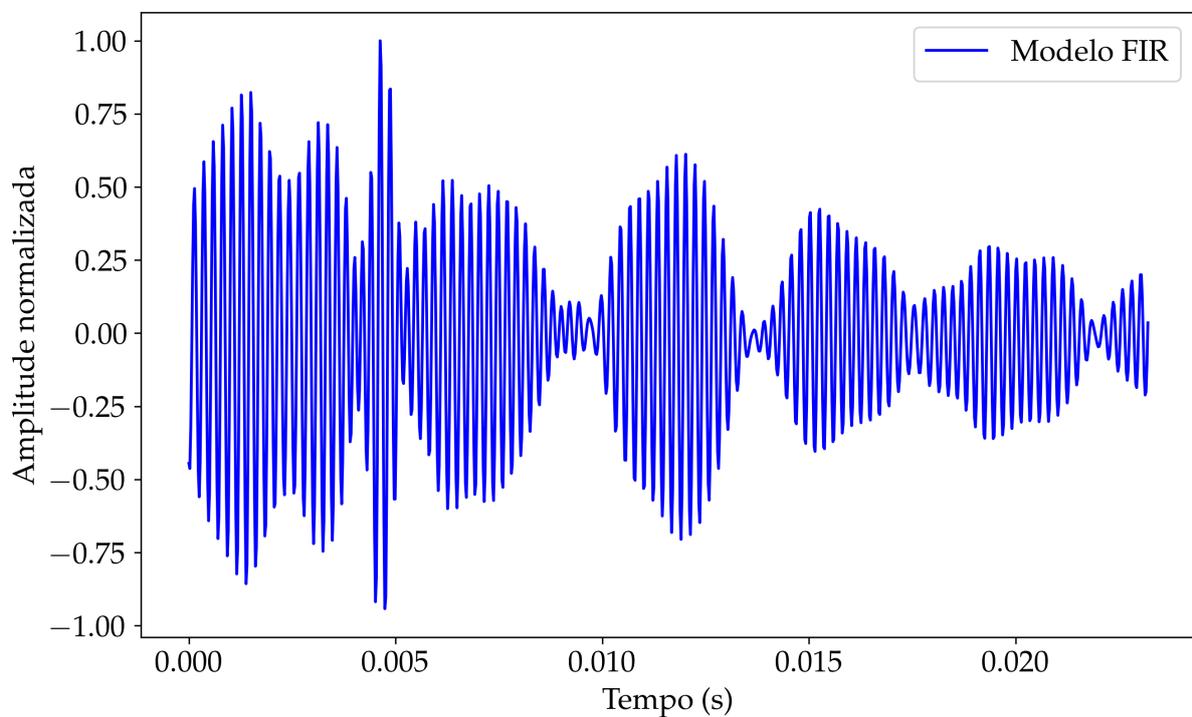


Figura 4.9. Modelo FIR obtido para o sistema formado pelo amplificador Blackstar ID Core 20 v2 e o microfone Santo Angelo SAS57.

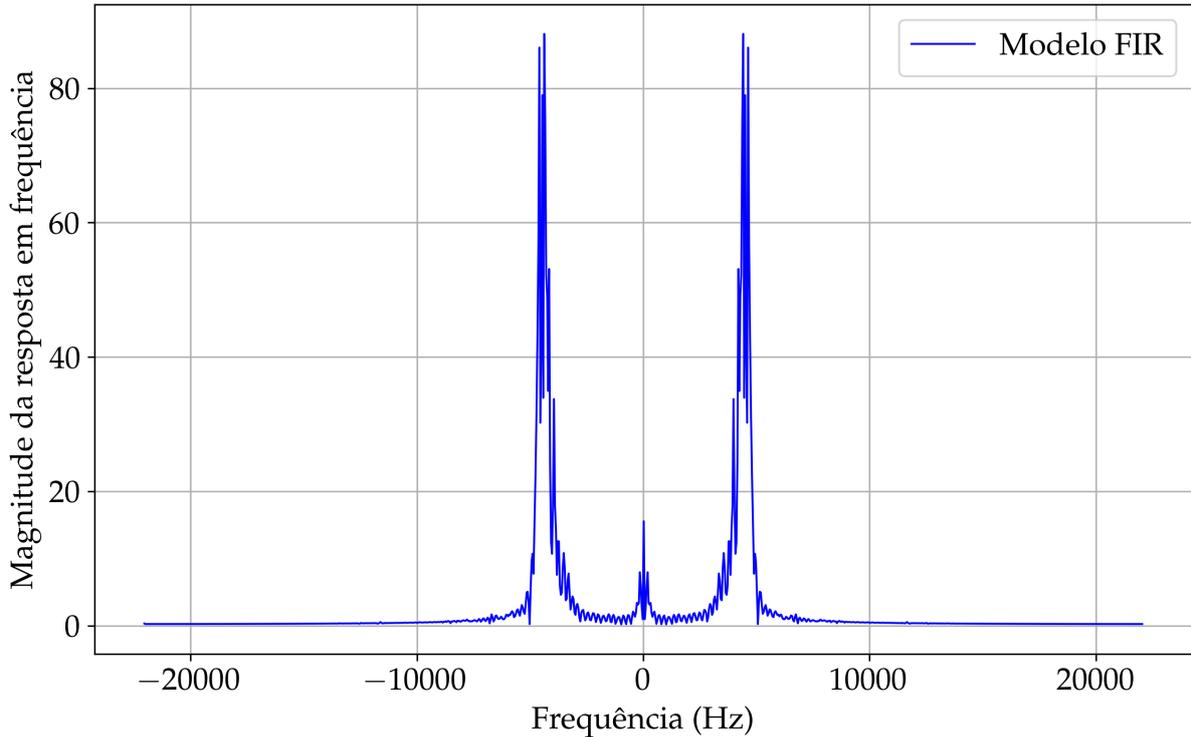


Figura 4.10. Magnitude da resposta em frequência do modelo FIR obtido para o sistema formado pelo amplificador Blackstar ID Core 20 v2 e o microfone Santo Angelo SAS57.

Observa-se que o modelo FIR para o pedal exibe um comportamento menos oscilante, enquanto o modelo do amplificador Blackstar possui uma faixa de frequência mais acentuada. Diferente do caso do pedal, não há modelos para realizar a comparação, e portanto não é possível afirmar que os mesmos parâmetros e a mesma ordem utilizada para criar o modelo FIR é adequada para modelar esse amplificador. Vale ressaltar que a posição do microfone influencia a captação do sinal, que é responsável por capturar o som emitido pelos alto-falantes e propagado pelo ar, o que torna o sistema mais suscetível a ruídos sonoros do ambiente.

Além do modelo FIR, foi obtido um modelo IIR de ordem 4 utilizando os mesmos sinais usados para o modelo FIR. A equação de diferenças do modelo IIR é dada por

$$\begin{aligned}
 y[n] = & -0.00229285x[n] - 0.00083447x[n-1] + 0.00062927x[n-2] + 0.00181103x[n-3] \\
 & - 2.46860021y[n-1] + 2.37715818y[n-2] - 0.85248921y[n-3],
 \end{aligned}
 \tag{4.1}$$

onde y representa o sinal de saída e x representa o sinal de entrada.

Após a obtenção dos modelos, foi realizada uma avaliação comparativa entre os mode-

los gerados e o amplificador real, utilizando um sinal de áudio gravado por uma guitarra elétrica, mostrado na Figura 4.11.

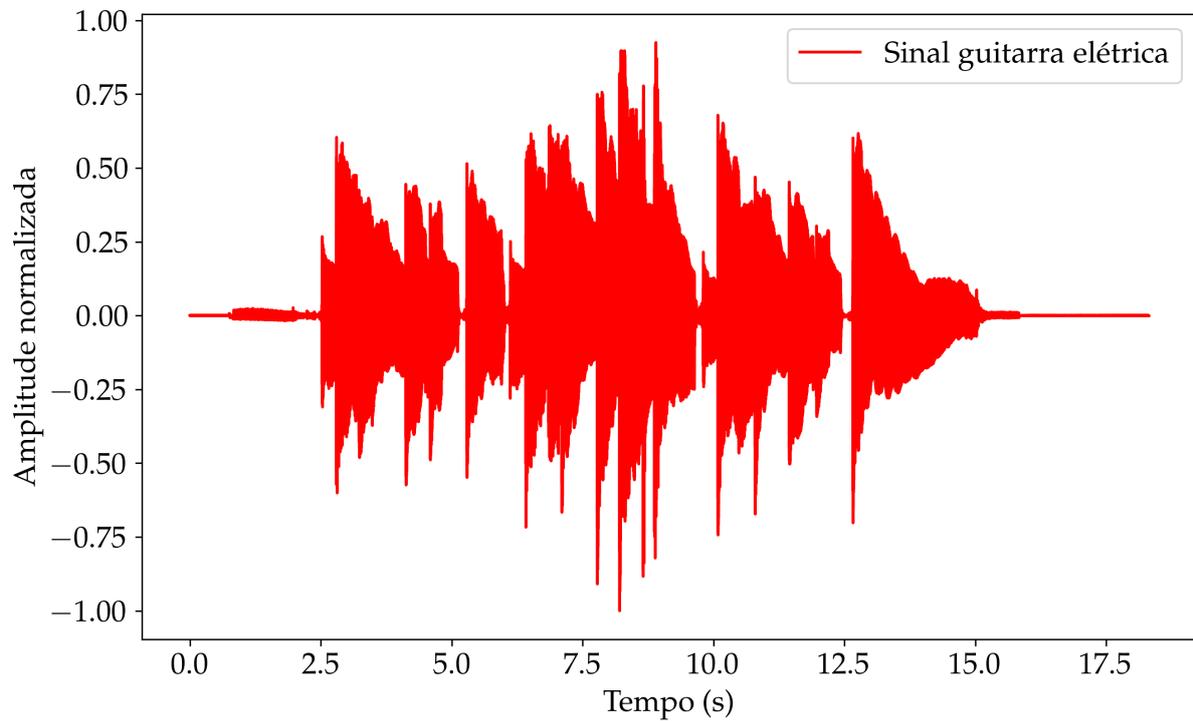


Figura 4.11. Sinal de áudio gravado por uma guitarra elétrica utilizado como entradas para os modelos FIR e IIR.

O sinal de áudio foi aplicado ao sistema e a saída do amplificador pode ser vista na Figura 4.12.

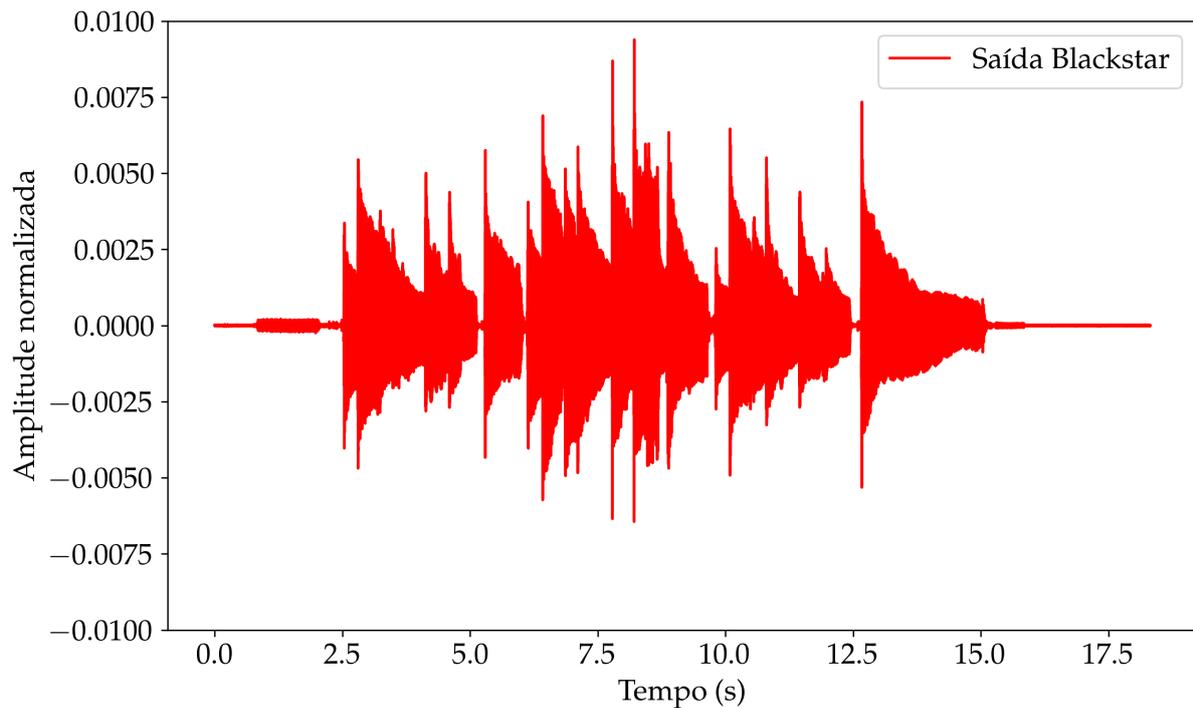


Figura 4.12. Sinal de saída obtido a pelo amplificador Blackstar ID Core 20 v2.

O sinal introduzido no amplificador foi aplicado aos modelos obtidos. Para o modelo FIR, foi feita a convolução entre a resposta ao impulso e o sinal gravado pela guitarra. Já para o IIR, foi feita a aplicação da equação das diferenças, multiplicando os pesos entre as amostras do sinal de entrada e as saídas passadas, e realizando a somatória dos valores. A saída obtida pela o modelo FIR pode ser vista em 4.13, e para o modelo IIR em 4.14.

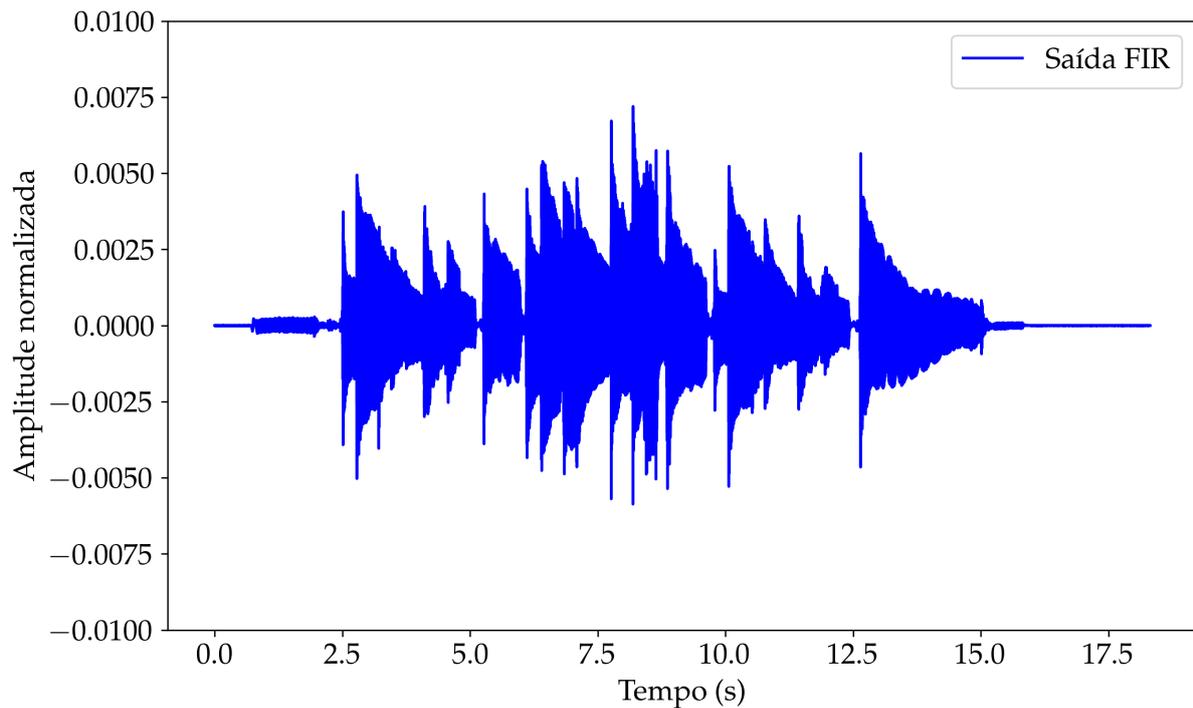


Figura 4.13. Sinal de saída obtido a pela convolução do sinal gravado pela guitarra elétrica e o modelo FIR obtido para o amplificador Blackstar ID Core 20 v2.

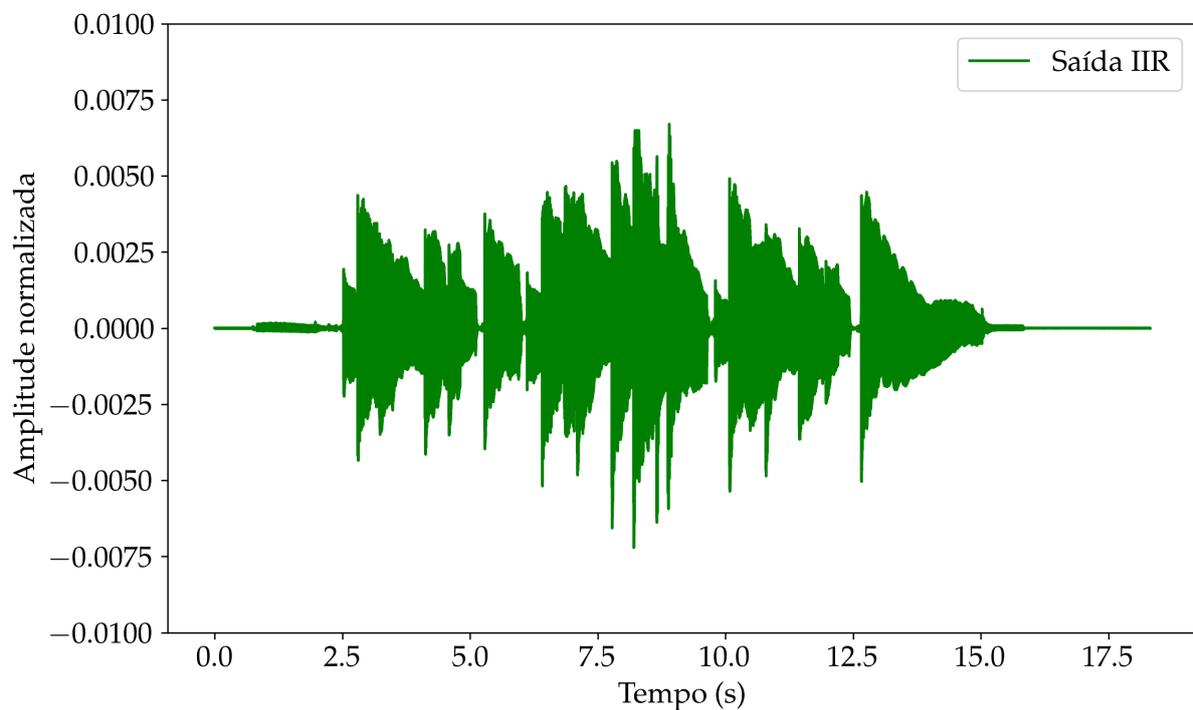


Figura 4.14. Sinal de saída obtido a pela aplicação da equação das diferenças ao sinal gravado pela guitarra elétrica com os pesos do modelo IIR obtido para o amplificador Blackstar ID Core 20 v2

As resposta em frequência para as saídas podem ser vistas sobrepostas na Figura 4.15,

com objetivo de verificar se há uma semelhança entre os resultados. Os valores de MSE e SNR não foram calculados para esse modelo, visto que não há um modelo comercial utilizado para esse amplificador, e com isso não é possível afirmar que os modelos obtidos são os quais melhor representam o sistema.

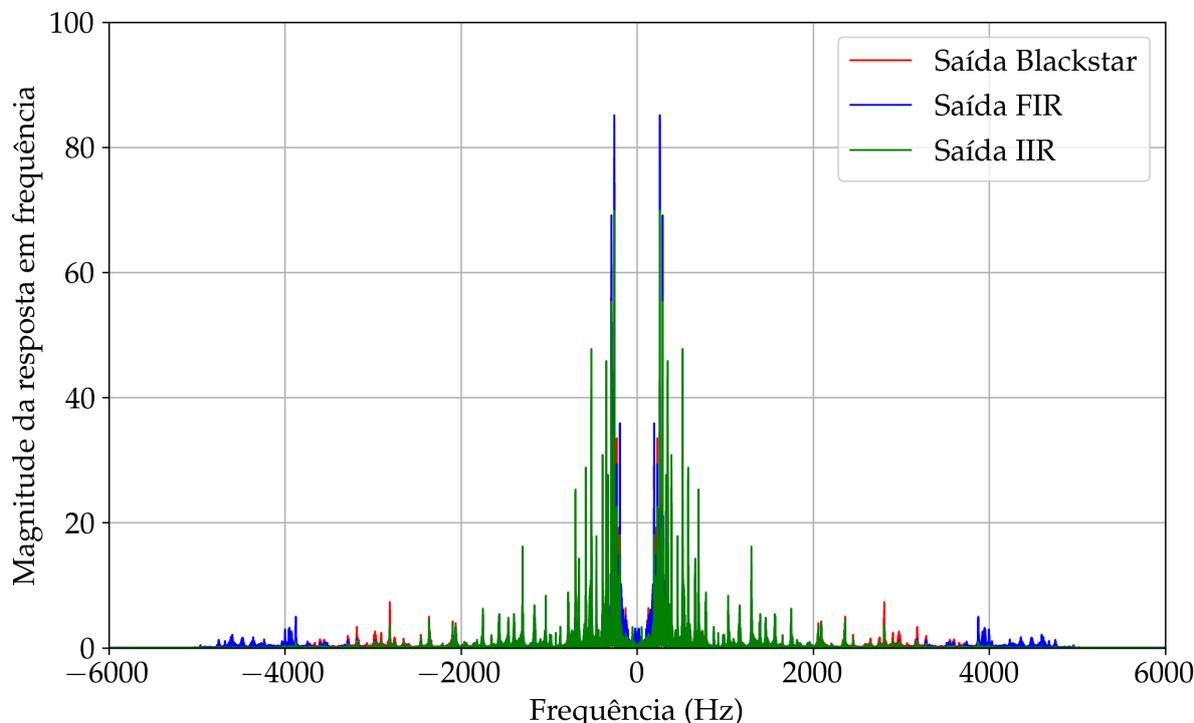


Figura 4.15. Comparação no domínio da frequência entre o sistema real e os modelos FIR e IIR obtidos do amplificador Blackstar ID Core 20 v2.

Os espectros mostram que há uma diferença entre os picos para os três sinais analisados, evidenciando que os resultados obtidos pelos modelos não coincidem com a saída obtida pelo amplificador. Dessa forma, pode ser reavaliado a ordem dos modelos e a quantidade de medidas realizadas para criação dos modelos.

4.2 Resultados da Emulação

Os resultados da emulação baseiam-se na implementação do algoritmo de convolução, que emula os modelos FIR. O algoritmo processa um *stream* de áudio, adquirindo amostras de entrada e gerando a saída correspondente. Esse *stream* foi configurado para operar com 256 amostras por *buffer*, no formato “paFloat32”, utilizado pela biblioteca PortAudio, e uma taxa de amostragem de 44.100 Hz, compatível com a utilizada na geração dos modelos. Apenas um canal de entrada e um de saída foram utilizados.

Para realizar a convolução e emular os modelos FIR, foram empregadas estruturas de repetição *for* e quatro *threads*, com o objetivo de otimizar o processamento. A Figura 4.16 apresenta um gráfico de dispersão, que ilustra o tempo de execução da função de convolução ao emular um modelo FIR com 1024 amostras.

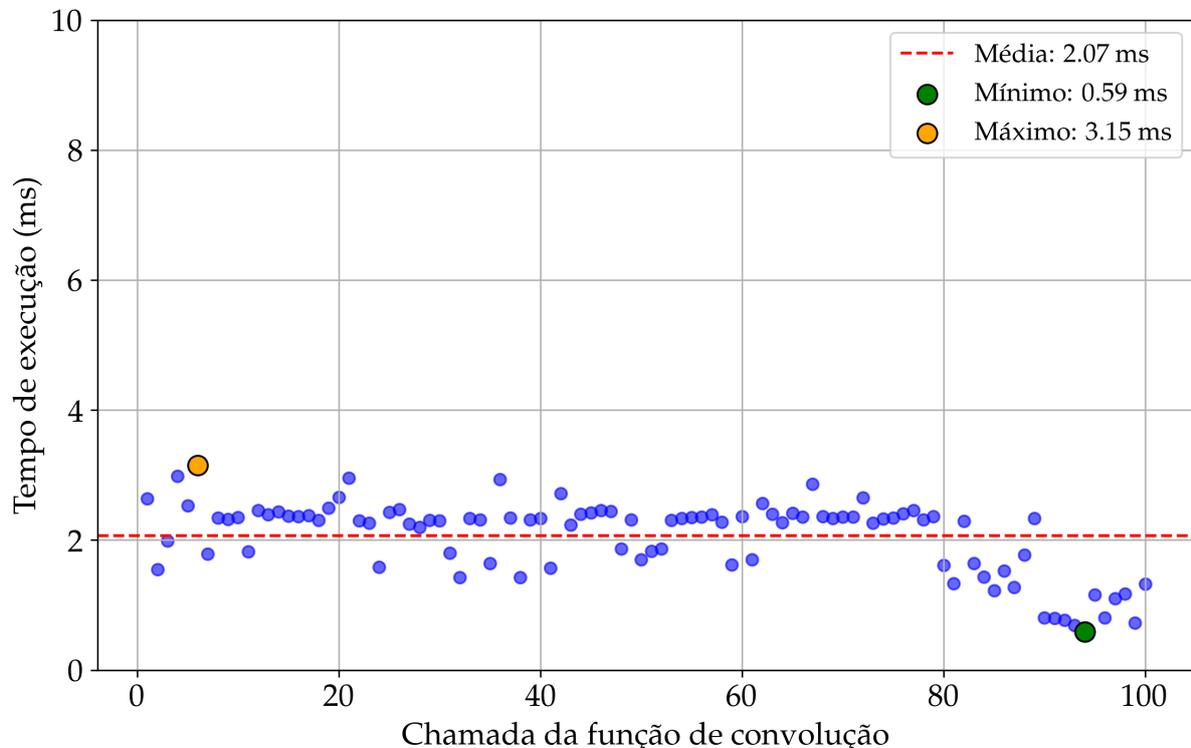


Figura 4.16. Gráfico de dispersão mostrando o tempo de execução da função de convolução para as 100 primeiras chamadas, executada em um computador. A linha tracejada vermelha indica a média, enquanto os pontos amarelo e verde representam os valores máximo e mínimo, respectivamente.

Os resultados indicam que, apesar de algumas variações, o tempo de execução da função de convolução permanece abaixo do limite de 20 ms estabelecido. A diferença entre o tempo máximo e mínimo não causa impactos perceptíveis para muitos cenários de uso do emulador. No entanto, para contextos mais exigentes, como gravações em estúdios, essas flutuações podem gerar atrasos indesejados. Os valores de latência da Figura 4.16 foram obtidos a partir da execução em um computador com sistema operacional de propósito geral, o que explica a inconsistência observada nos tempos de latência.

O algoritmo de emulação também foi testado em um Orange Pi 3B [7], um *Single Board Computer* (SBC) comumente utilizado em projetos de sistemas embarcados. Embora o sistema operacional usado no SBC também seja de propósito geral, ele é minimalista, dessa forma, é esperado tempos de execução mais consistentes, devido à menor quantidade de processos paralelos em execução. A Figura 4.17 apresenta o gráfico de dispersão

para o tempo de execução da função de convolução nesse dispositivo.

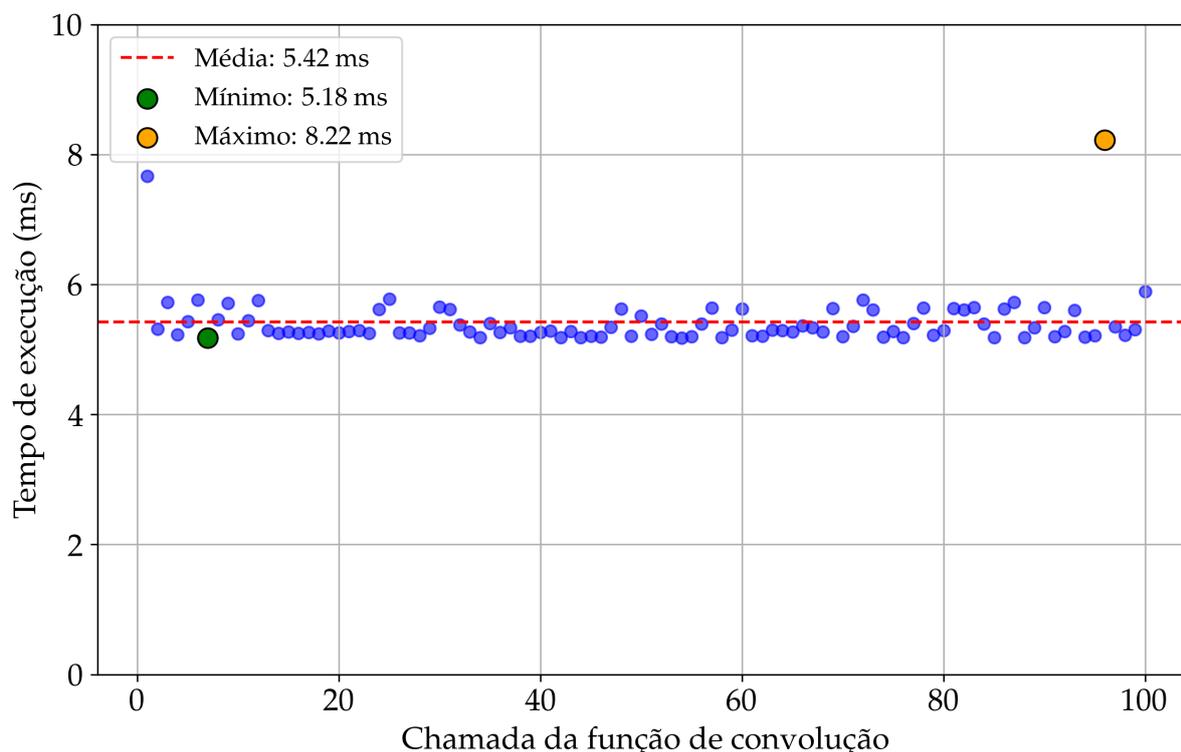


Figura 4.17. Gráfico de dispersão mostrando o tempo de execução da função de convolução para as 100 primeiras chamadas, executada em um Orange Pi 3B [7]. A linha tracejada vermelha indica a média, enquanto os pontos amarelo e verde representam os valores máximo e mínimo, respectivamente.

Ao analisar os resultados no Orange Pi 3B, observa-se uma menor dispersão nos tempos de execução em comparação com o computador, embora ainda haja valores discrepantes em casos isolados. Essas discrepâncias são atribuídas ao uso de um sistema operacional de propósito geral, que, mesmo sendo minimalista, ainda apresenta variações ocasionais devido à concorrência do programa de emulação com outros processos.

Vale destacar que, no início do processo de emulação, foi identificado um erro de *overflow* de entrada. Esse *overflow* ocorre quando um novo conjunto de dados da amostra de entrada está disponível e o *buffer* ainda não foi liberado para escrita, resultando em uma fila de entrada que excede o tamanho do *buffer*. Dessa forma, o sistema não conseguiu processar as amostras de entrada rapidamente o suficiente, fazendo com que o tempo de execução da função de convolução atingisse valores muito acima da média obtida nas quatro primeiras chamadas, chegando até 24.724 ms. Esse problema foi resolvido após essas chamadas iniciais, e o tempo de execução se estabilizou nas iterações subsequentes, eliminando o erro de *overflow*.

Outro problema que pode ser mencionado é que durante a saída gerada pelo algoritmo

do emulador, foi percebido a presença de um ruído possivelmente obtido por interferências no *hardware*, aproximações ou conversão de dados durante os cálculos feitos pelo algoritmo. Infelizmente não foi possível realizar a captura do sinal para comparar com o sinal obtido pela convolução em modo *offline*, visto que a placa de aquisição é utilizada para emulação, sendo inviável a sua utilização para captura da saída. Apesar do ruído está presente, ele não prejudica a identificação dos modelos através da percepção auditiva.

5 CONCLUSÃO

Este trabalho apresentou uma metodologia para a modelagem, emulação e avaliação de sistemas de áudio utilizando modelos FIR e IIR, com base na teoria de sistemas lineares e invariantes no tempo. O *hardware* desenvolvido e os algoritmos implementados foram empregados para validar a eficácia da metodologia proposta. A modelagem dos sinais gerados pelo pedal Sonic IR apresentou um bom alinhamento entre os modelos FIR e os sinais originais, resultando em baixos valores de erro quadrático médio e excelentes relações sinal-ruído. No entanto, os modelos IIR gerados para o mesmo sistema não conseguiram reproduzir adequadamente todo o espectro de frequência, especialmente nas mesmas condições avaliadas para os modelos FIR.

Ao verificar os modelos FIR e IIR obtidos para o amplificador Blackstar ID Core 20 v2 e o pedal Sonic IR, foi observado que a gravação utilizando o microfone é mais suscetível a ruídos do que a gravação em linha. Os modelos do amplificador Blackstar não puderam ser verificados, visto que não existem modelos comerciais para comparação. Dessa forma, é possível que os modelos obtidos não estejam com a ordem adequada e uma maior quantidade de medidas e testes devem ser feitos. Vale ressaltar que existe também uma influência do ruído do ambiente, visto que para o caso da captura dos sinais da saída do amplificador Blackstar foi utilizado um microfone.

Os resultados da emulação destacam a capacidade do algoritmo de convolução em processar *streams* de áudio e emular de forma eficaz os modelos FIR. Essa abordagem confirma a viabilidade da emulação digital de sistemas de áudio em ambientes computacionais, utilizando o *hardware* adequado.

Para trabalhos futuros, pode ser avaliado a implementação de técnicas para melhorar a captura de sinais com microfones, como o uso de filtros e a otimização do posicionamento dos microfones, além de avaliar ordens e outras metodologias para implementação de modelos IIR. Já para a emulação pode ser feita uma avaliação do algoritmo em ambientes controlados e projetados para processamento em tempo real, com objetivo de garantir que não hajam valores inconstantes de latência, e também verificar e resolver as possíveis causas para o ruído gerado na saída do emulador implementado. Além disso, sugere-

se implementar o algoritmo de emulação para modelos IIR e realizar uma comparação sistemática entre os modelos IIR e FIR.

Referências Bibliográficas

- [1] Sonicake. Pedal de guitarra sonic ir. Disponível em: <https://sonicake.com.br/products/pedal-de-guitarra-sonicake-sonic-ir>. Acesso em: 19 ago. 2024.
- [2] IK Multimedia. Tonex one. Disponível em: <https://www.ikmultimedia.com/products/tonexone/>. Acesso em: 23 set. 2024.
- [3] A. Sedra. *Microeletrônicos: Circuitos e Dispositivos*. Editora Técnica, São Paulo, 2002.
- [4] Presonus. Interface de áudio studio 24c. Disponível em: <https://www.presonus.com/en/interfaces/usb-audio-interfaces/studio-series/2777700403.html>. Acesso em: 19 ago. 2024.
- [5] Blackstar Amplifiers. Id:core v2 handbook. Disponível em: <https://blackstaramps.com/wp-content/uploads/2021/10/id-core-v2-handbook.pdf>. Acesso em: 19 ago. 2024.
- [6] Santo Angelo. Microfone sas57. Disponível em: <https://www.santoangelo.com.br/detalhes-do-produto/476/microfone-sas57>. Acesso em: 19 ago. 2024.
- [7] Orange Pi. Orange pi 3b - service and support. Disponível em: <http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-3B.html>. Acesso em: 9 set. 2024.
- [8] J. Smith. *Electric Guitar Techniques*. Guitar Publications, Los Angeles, 2005.
- [9] G. Weinreich. Musical acoustics and signal processing. *Journal of Acoustics*, 25:123–145, 1999.
- [10] T. Fricke. *Jimi Hendrix: A Musical Biography*. Rock Legends, New York, 1998.
- [11] J. R. Buck. *Electromagnetismo*. Editora Científica, São Paulo, 2013.
- [12] S. De. *A Guitarra: História e Desenvolvimento*. Editora Musical, Rio de Janeiro, 2007.

- [13] A. Smith. Techniques in modelling. *Journal of Modelling*, 15:45–67, 2010.
- [14] B. Shinn-Cunningham. How sound localization works. *Journal of the Acoustical Society of America*, 128:123–135, 2010.
- [15] Kemper. Kemper profiling amplifier. Disponível em: <https://www.kemper-amps.com/profiler/overview>. Acesso em: 19 ago. 2024.
- [16] Neural DSP. Quad cortex. Disponível em: <https://neuraldsp.com/quad-cortex>. Acesso em: 19 ago. 2024.
- [17] L. Cox. *The Modern Audio Systems*. Audio Publications, New York, 1984.
- [18] R. Magalhães. *Escutar Sons Graves*. Editora Sonora, Belo Horizonte, 2020.
- [19] T. Johnson. *Amplificadores de Tubo: Projeto e Construção*. Tecnologia Amplificada, São Paulo, 1999.
- [20] B. Lathi. *Sinais e Sistemas Lineares*. Editora Matemática, São Paulo, 2008.
- [21] M. Roberts. *Fundamentos de Sinais e Sistemas*. Editora Técnica, Rio de Janeiro, 2015.
- [22] P. Diniz. *Processamento de Sinais*. Editora Técnica, São Paulo, 2014.
- [23] A.V. Oppenheim e A.S. Willsky. *Sinais e Sistemas*. Pearson Universidades, 2010.
- [24] Stephen Boyd e Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [25] Gustavo Henrique Oliveira. Repositório git projeto modelagem de amplificador de áudio. Disponível em: <https://gitlab.com/GHRV0/modelagem-audio>. Acesso em: 9 set. 2024.
- [26] NumPy. Numpy: The fundamental package for scientific computing with python. Disponível em: <https://numpy.org>. Acesso em: 9 set. 2024.
- [27] SciPy. Scipy: Fundamental algorithms for scientific computing in python. Disponível em: <https://scipy.org>. Acesso em: 9 set. 2024.
- [28] SoundFile. Soundfile: An audio library based on libsndfile, cffi and numpy. Disponível em: <https://pypi.org/project/soundfile/>. Acesso em: 9 set. 2024.
- [29] SoundDevice. Sounddevice: Play and record sound with python. Disponível em: <https://pypi.org/project/sounddevice/>. Acesso em: 9 set. 2024.

- [30] PortAudio. Portaudio - portable cross-platform audio i/o. Disponível em: <https://www.portaudio.com>. Acesso em: 9 set. 2024.
- [31] cplusplus.com. std::thread reference in c++. Disponível em: <https://cplusplus.com/reference/thread/thread/>. Acesso em: 9 set. 2024.

6 APÊNDICES

6.1 Algoritmos de modelagem

O algoritmo de modelagem, mostrado em 6.1, apresenta o programa principal utilizado para o todo processo de criação dos modelos FIR e IIR. Inicialmente, é feita uma etapa de aquisição dos parâmetros a partir da entrada de valores pelo usuário para posteriormente efetuar cálculos para criação dos modelos. As saídas do programa são três arquivos, sendo eles: o arquivo de gravação dos sinais utilizados para modelagem em formato WAV, com o sinal aplicado ao sistema no canal 1 e o sinal de saída do sistema no canal 2, o arquivo WAV do modelo FIR e, o arquivo de texto contendo os coeficientes do modelo IIR.

```
# main.py

import numpy as np
import threading
import sounddevice as sd
import soundfile as sf
from matplotlib import pyplot as plt
from audio_utils import generate_sine_wave, record_audio,
    plot_signals, play_audio, calculate_amplitude_phase,
    calculate_amplitude_phase_fft, create_fir_model, create_iir_model

def main():

# ----- Entradas do Usuario ----- #
    print("Controle do Algoritmo de Sinais Sonoros")

    freq_min = float(input("Digite a frequencia mais grave (Hz): "))
    freq_max = float(input("Digite a frequencia mais aguda (Hz): "))
    quantity = int(input("Digite a quantidade de sinais: "))
    duration = float(input("Digite a duracao do sinal (em segundos):
    "))
    print("Dispositivos de Reproducao:")
    print(sd.query_devices())
    playback_device = int(input("Escolha o dispositivo de reproducao
    (ID): "))

    print("Dispositivos de Gravacao:")
    print(sd.query_devices())
    recording_device = int(input("Escolha o dispositivo de gravacao
    (ID): "))

    filename = input("Digite o nome do arquivo para salvar a gravacao
    e os modelos (ex: saida): ")
```

```

# ----- #

sample_rate = 44100
playback_signal = None
recorded_signal = None

freqs = []
amps = []
phases = []

t = np.linspace(0, duration, int(sample_rate * duration), False)
combined_signal = np.zeros_like(t)

for i in range(quantity):
    freq = np.linspace(freq_min, freq_max, quantity)[i]
    sine_wave, _ = generate_sine_wave(freq, duration, sample_rate)
    combined_signal += sine_wave

print(f"Preparando sinal combinado de {freq_min} Hz a {freq_max}
Hz por {duration} segundos")

audio_container = {}
play_thread = threading.Thread(target=play_audio,
args=(combined_signal, sample_rate, playback_device))
record_thread = threading.Thread(target=record_audio,
args=(duration, sample_rate, 2, recording_device, audio_container))

play_thread.start()
record_thread.start()

play_thread.join()
record_thread.join()

playback_signal = combined_signal
recorded_signal = audio_container["data"]

# Salvando gravacao
recorded_file = filename + ".wav"
sf.write(recorded_file, recorded_signal, sample_rate)

freqs, amps, phases = calculate_amplitude_phase_fft(recorded_file,
freq_min, freq_max, quantity)

# Criar modelo FIR
h_fir = create_fir_model(freqs, amps, phases)
fir_file = "fir_" + filename + ".wav"
sf.write(fir_file, h_fir, sample_rate)

# Criar modelo IIR
N, D = create_iir_model(freqs, amps, phases, 9)

iir_file = "iir_" + filename + ".txt"
with open(iir_file, 'w') as file:
    file.write("Coeficientes Numerador:\n")
    np.savetxt(file, [N.real], fmt='%0.8f', delimiter=',')

    file.write("\nCoeficientes Denominador:\n")
    np.savetxt(file, [D.real], fmt='%0.8f', delimiter=',')

print("Modelos FIR e IIR salvos.")

# ----- #
#

```

```
if __name__ == "__main__":
    main()
```

Algoritmo 6.1. Algoritmo principal utilizado para o projeto de modelagem

O processo de reprodução e gravação dos sinais de áudio é feito de forma paralela utilizando *threads*. As funções desenvolvidas são encontradas no programa “audio_utils.py”, e para visualizá-las com todos os detalhes, é recomendado a visita ao repositório do projeto [25]. É destacada a função *calculate_amplitude_phase_fft*, apresentada em 6.2, que utiliza a função *dtfft*, a qual implementa a Transformada de Fourier em Tempo Discreto, chamada para cálculo do valor da transformada para as frequências utilizadas no processo de aquisição dos dois sinais. Vale destacar que no início da função, é feito um ajuste nos sinais removendo 20% do início e do fim deles para eliminar possíveis transitórios.

```
def calculate_amplitude_phase_fft(filename, freq_min, freq_max,
    quantity):
    data, fs = sf.read(filename)

    entrada = data[:, 0]
    saida = data[:, 1]

    # Remover 20% do inicio e do fim dos sinais, para tirar possiveis
    transitorios
    samples = len(entrada)
    adjust = int(0.2 * samples)
    entrada = entrada[adjust:samples-adjust]
    saida = saida[adjust:samples-adjust]

    freqs = np.arange(freq_min, fs/2, 10) / fs # pegando as
    frequencias de todo o espectro

    fft_entrada = dtfft(entrada, freqs, freq_max/fs)
    fft_saida = dtfft(saida, freqs, freq_max/fs)

    # Calcular modulo e fase
    amps = np.zeros(shape=fft_entrada.shape)
    phases = np.angle(fft_saida) - np.angle(fft_entrada) #+ np.pi

    amps = np.where(np.abs(fft_saida) != 0, np.abs(fft_saida) /
    np.abs(fft_entrada), 0) # evitar divisao por 0

    freqs, amps, phases = add_negative_frequencies(freqs, amps, phases)
```

Algoritmo 6.2. Função *calculate_amplitude_phase_fft* utilizada para cálculo de amplitude e fase dos sinais gravados da entrada e saída do sistema modelado

6.2 Algoritmos de emulação

O algoritmo de emulação mostra o programa principal utilizado para emulação dos modelos FIR. Ainda podem ser feitas melhorias no programa, principalmente para o processo de seleção dos arquivos modelos utilizados, porém a função responsável pela emulação mostrada em 6.3 aborda a metodologia utilizada.

```

void emulation(float *buffer, audioBuffer ir)
{
    convResult convolve_buffer;
    float output[FRAMES_PER_BUFFER];
    memset(output, 0, sizeof(output));

    int total_samples = FRAMES_PER_BUFFER + ir.n_samples - 1; //
    tamanho da saída da convolucao

    float res[total_samples - FRAMES_PER_BUFFER];
    memset(res, 0, sizeof(res));

    while(1)
    {
        audio.recordSamples(buffer, FRAMES_PER_BUFFER);

        convolve_buffer = calc.convolution(buffer,
FRAMES_PER_BUFFER, ir.samples, ir.n_samples);

        for (int i = 0; i < (total_samples - FRAMES_PER_BUFFER);
i++)
        {
            if (i < FRAMES_PER_BUFFER)
                output[i] = (convolve_buffer.conv[i] + res[i]);

            // res[i] += convolve_buffer.conv[i +
FRAMES_PER_BUFFER];
        }

        audio.playbackSamples(output, FRAMES_PER_BUFFER);
    }
    return;
}

```

Algoritmo 6.3. Função de emulação que realiza a convolução do sinal adquirido e do modelo FIR selecionado pelo usuário

É destacado o método *convolution* da classe *Calculation*, chamada pelo objeto *calc*, em que é realizado a convolução dos dois sinais. O método é apresentado em 6.4 e utiliza quatro *threads* para otimizar o tempo de execução do processo de convolução.

```

convResult Calculation::convolution(float *input, int input_len, float
*impulse_response, int impulse_response_len) {
    int conv_len = input_len + impulse_response_len - 1;

    // Alocar buffer para o resultado da convolucao
    float *conv = new float[conv_len];
    float acc = 0;

    auto start = std::chrono::high_resolution_clock::now();

    // Dividir o trabalho entre threads
    int num_threads = 4;
    int segment_size = conv_len / num_threads;
    std::vector<std::thread> threads;

    for (int t = 0; t < num_threads; t++) {
        int start_idx = t * segment_size;
        int end_idx = (t == num_threads - 1) ? conv_len : start_idx +
segment_size;
    }
}

```

```

        threads.emplace_back(&Calculation::convolutionSegment, this,
input, input_len, impulse_response, impulse_response_len, conv,
start_idx, end_idx);
    }

    // Esperar todas as threads terminarem
    for (auto& th : threads) {
        th.join();
    }

    // Calcular a soma acumulada
    for (int i = 0; i < conv_len; i++) {
        acc += conv[i] * conv[i];
    }

    auto end = std::chrono::high_resolution_clock::now();
    std::chrono::duration<double, std::milli> duration = end - start;
    std::printf("Tempo de execucao da convolucao: %.3f ms\n",
duration.count());

    convResult result;
    result.conv = conv;
    result.acc = std::sqrt(acc);
    return result;
}

```

Algoritmo 6.4. Algoritmo principal utilizado para o projeto de emulação

O sinal de saída da convolução possui um tamanho maior do que a entrada gravada, o que é esperado de acordo com a própria definição do processo. Dessa forma, o sinal de saída é reproduzido a partir do resultado parcial da convolução, e as amostras excedentes são armazenadas no vetor *res*. Porém, não é utilizado o resultado excedente armazenado em *res* para a próxima saída, pois durante os testes, foi percebido que o áudio de saída ultrapassou os limites e ficou distorcido. Como mencionado no capítulo 4, há um ruído na saída perceptível mesmo sem a adição desse resultado excedente.