

Protótipo de Auto-tune e Exclutor de Nota

Ana Carolina Ribeiro Miranda, 2208407, André Luís de Oliveira, 2270170, Lucas Eduardo Pires Parra, 2208490,

Abstract—Este resumo descreve um projeto de desenvolvimento de uma aplicação gráfica em MATLAB, utilizando o App Designer, com base em conhecimentos adquiridos em uma disciplina de processamento digital de sinais, para criar um sistema similar ao Auto-Tune. O projeto envolve a implementação de um filtro que extrai o espectro de frequência de um áudio, realiza o deslocamento desse espectro para modificar a frequência com maior amplitude e reconstrói o sinal resultante, permitindo a alteração do tom do áudio.

A aplicação permite ao usuário importar um arquivo de áudio e exibe sua plotagem do espectro de frequência. Ao deslocar o espectro, é possível modificar o tom do áudio, proporcionando funcionalidades similares a um "protótipo de auto-tune". O processo de deslocamento do espectro é realizado através de cálculos e manipulação adequada das frequências com maior amplitude.

Além disso, o projeto tem como objetivo implementar a funcionalidade de exclusão de uma nota específica usando um filtro IIR. Essa adição permitirá remover seletivamente uma nota indesejada do áudio, oferecendo maior controle sobre o resultado final.

A abordagem utilizada neste projeto é baseada em técnicas de processamento digital de sinais, incluindo transformadas de Fourier e manipulação de espectro de frequência. O sistema desenvolvido demonstra eficácia na modificação do tom do áudio, proporcionando uma afinação aprimorada e abrindo possibilidades criativas na produção musical.

Em conclusão, este projeto apresenta o desenvolvimento de uma aplicação GUI MATLAB inspirada no Auto-Tune, que utiliza um filtro para extrair, deslocar, excluir e reconstruir o espectro de frequência de um áudio. Os resultados demonstram o potencial do sistema para aplicações em processamento de áudio.

Index Terms—Auto-tune, fourier, fft, filtros, IIR.

I. INTRODUÇÃO

O Processamento digital de sinais desempenha um papel crucial na manipulação e aprimoramento de áudio. Neste trabalho, propomos o desenvolvimento de uma interface gráfica baseada no

MATLAB, utilizando o App Designer, para criar um sistema semelhante ao Auto-Tune. O objetivo principal é permitir que os usuários modifiquem o tom de áudio de forma intuitiva e eficaz, proporcionando uma experiência aprimorada de produção musical.

A tecnologia do Auto-Tune tem sido amplamente utilizada na indústria da música para corrigir desafinações e criar efeitos vocais desejados, assim, por meio de uma interface gráfica amigável e utilizando as capacidades de processamento digital de sinais do MATLAB iremos reproduzir algo semelhante.

A interface desenvolvida permite ao usuário gravar ou importar um áudio e visualizar sua representação no domínio do tempo. Em seguida, é possível escolher uma nota musical e sua oitava, bem como definir se a nota possui sustenido ou não. Ao clicar em um botão "Gerar Gráfico", o espectro de frequência do áudio é extraído e deslocado para modificar a frequência com maior amplitude, alterando assim o tom do áudio. O espectro deslocado é então utilizado para reconstruir o sinal de áudio.

Para realizar esse processo, são aplicadas técnicas de processamento digital de sinais, incluindo a transformada de Fourier para análise espectral e manipulação das componentes de frequência. Além disso, o sistema é projetado para ser capaz de excluir uma nota específica do áudio utilizando um filtro IIR.

No decorrer deste artigo, apresentaremos em detalhes a implementação da interface gráfica, os métodos utilizados para extração e manipulação do espectro de frequência e os resultados obtidos com a aplicação em diferentes amostras de áudio.

II. DESENVOLVIMENTO

Utilizou-se para o desenvolvimento do projeto, visando uma melhor visualização, a interface gráfica em MATLAB, denominada App Designer, ao qual é dividida em duas janelas, sendo a primeira o "Design View" e o "Code View", onde respectivamente era atribuída a forma visual e na outra implementada o código.

A lógica e o comportamento do aplicativo no App Designer são implementados usando programação textual em MATLAB. Em vez de usar blocos visuais para definir a lógica, se escreve o código MATLAB diretamente nos callbacks dos componentes, como funções que são acionadas quando um botão é pressionado ou quando um valor é alterado.

Para o desenvolvimento foi-se criado um "Properties", uma seção da interface gráfica do usuário (GUI) onde você pode visualizar e modificar as propriedades de componentes ou objetos de uma aplicação, também criou-se um "Methods", uma seção na GUI onde você pode definir e gerenciar os métodos associados aos eventos dos componentes do seu aplicativo, tais como a de gerar as cores de forma aleatória nos gráficos, a função para se verificar a nota predominante da frequência e por fim a função responsável por modificar o espectro.

O código possui no total 9 "botões", sendo eles:

- Gravar: onde pega em tempo real um áudio de até 5 segundos gerados na hora;
- Importar: onde pega um arquivo de áudio;
- Nota: seleciona uma das notas (Dó, Ré, Mi, Fá, Sol, Lá, Si);
- Oitava: a oitava da nota (1, 2, 3, 4, 5);
- Sustenido: onde seleciona se a nota possui ou não sustenido, quando a nota em questão não possui a variação de sustenido, o botão passa a ser desativado, impossibilitando sua escolha;
- Mudar tom: responsável por fazer a alteração com base nas notas escolhidas, gerando o novo gráfico do espectro;
- Remover tom: responsável pela exclusão do tom selecionado
- Reproduzir: dois botões responsáveis por reproduzir o áudio, além de atualizar o gráfico para a forma de onda, ambos os botões só irão

aparecer, embaixo dos respectivos gráficos, quando tiver alguma informação presente.

Para a mudança de tom predominante do sinal, é utilizado um algoritmo que vai pegar a frequência predominante do espectro original. A partir da seleção de notas e oitavas, será escolhida uma frequência alvo para ser a nova maior frequência. Com isso, é calculada a diferença entre as duas frequências. Dependendo se esse resultado for positivo, ou não, o espectro será deslocado para a esquerda ou para a direita. Para o deslocamento do áudio, foi utilizada uma iteração em loop por meio de um 'for'. Para as frequências que ficariam vazias devido o deslocamento, foram dados o valor '0'. Por fim, com o novo espectro de frequência gerado, o áudio foi reconstruído por meio da função 'ifft'.

Já para a remoção de uma nota específica, a seleção de notas e oitavas é reaproveitada. Com isso, é selecionada uma faixa de frequência para o Ω_d e para o Ω_s de uma matriz. O R_p escolhido foi 1 e o R_s escolhido varia entre 20 e 13, dependendo da nota que está sendo filtrada. Com essas variáveis, foi obtido os coeficientes para o filtro de Butterworth com a função 'buttord', e, por fim, após a função 'butter' ter sido utilizada, o áudio é filtrado utilizando a função 'filtfilt', sendo gerado o novo espectro de frequência em seguida.

III. RESULTADOS E CONCLUSÃO

Como resultado do trabalho desenvolvido, foi obtido o aplicativo com interface gráfica em MATLAB representado na figura a seguir.

Fig. 1. Interface do programa com o áudio importado e com tom removido

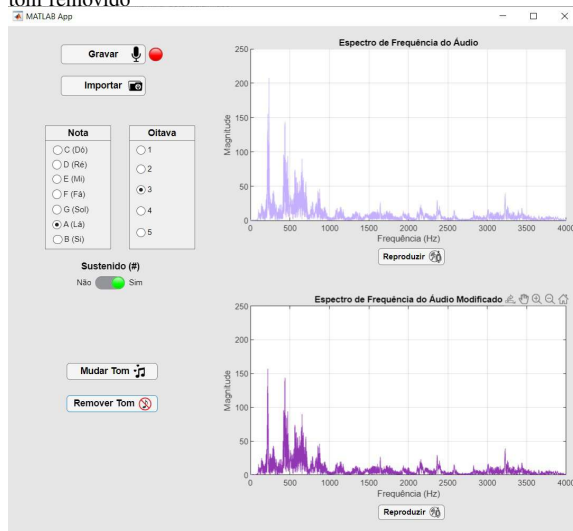


Fig. 2. *

Fonte: Autoria própria.

Utilizando o botão 'Gravar', é iniciada a gravação de um áudio para ser modificado no programa. Esse áudio tem a duração de 5 segundos e uma taxa de amostragem de 8 KHz. Alternativamente, é possível utilizar um áudio gravado anteriormente usando o botão 'Importar'. O áudio a ser importado precisa estar salvo em um arquivo do tipo ".mat".

Abaixo do botão importar, é possível selecionar uma nota e uma oitava. Também é possível determinar se será usado o subtom sustenido dessa nota ou não. A partir da nota escolhida, é possível realizar duas modificações com ela. Utilizando o botão 'Mudar Tom', é possível mudar o tom predominante do áudio. Já com o botão 'Remover Tom', é possível atenuar a nota escolhida. Como o faixa de frequência a ser filtrada é muito pequena, a atenuação não é tão grande quanto o desejado, mas ainda sim é possível notar diferenças no espectro de frequência.

Fig. 3. Interface do programa com o áudio importado e com o tom modificado

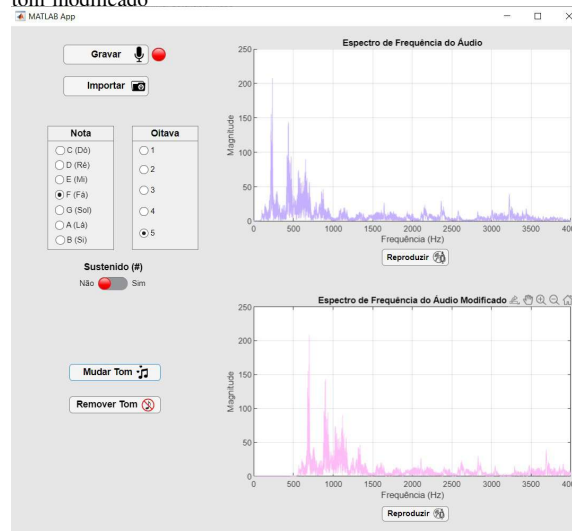


Fig. 4. *

Fonte: Autoria própria.

Por fim, há dois botões 'Reproduzir'. Com eles é possível reproduzir o áudio original e o áudio modificado, respectivamente.

Todos os diagramas produzidos aparecem dentro da própria interface gráfica.

A partir desses resultados obtidos, é possível concluir que a interface gráfica foi produzida de maneira satisfatória. Todos os botões funcionam corretamente e ela se comporta da exata maneira que era para funcionar. Já na parte de deslocamento e reconstrução do áudio, para mudar a nota predominante, é possível concluir que também foi obtido um bom resultado. É facilmente perceptível que o áudio fica ou mais agudo ou mais grave. Também é possível perceber uma pequena distorção. Por fim a remoção de tons não proveu um resultado tão satisfatório. Como as faixas de frequências a serem excluídas eram muito pequenas, a atenuação da banda de rejeição não pôde ser tão alta. Dessa forma, apesar da atenuação ser facilmente perceptível no diagrama de frequências, ela não é tão perceptível no áudio reconstruído.

Ao finalizar o projeto proposto, foi criado um aplicativo do trabalho com ajuda do "Share" uma

opção do app designer, onde adiciona como "Standalone Desktop App", que em termos de software, "standalone" refere-se a um programa ou aplicativo que é independente e autônomo, não requerendo a instalação de outros componentes ou software adicional para funcionar.

Um aplicativo "standalone" é capaz de executar suas funções sem depender de outros aplicativos ou bibliotecas externas. Isso significa que ele contém todas as dependências necessárias incorporadas em si mesmo, para que possa ser executado em um ambiente isolado, neste caso, ao se importar um áudio, se mantém constantemente a mesma gravação padrão salva no aplicativo.

AGRADECIMENTOS

Gostaríamos de expressar nosso sincero agradecimento ao professor Daniel Prado de Campos pela sua dedicação e orientação durante a disciplina de Processamentos Digitais de Sinais. Seu profundo conhecimento e entusiasmo pela área foram fundamentais para o desenvolvimento deste trabalho.

Também queremos destacar toda a forma como o professor incentivou a exploração criativa e aprofundamento nos conceitos de processamentos digitais de sinais. Sua abordagem dinâmica e didática enriqueceu nossa compreensão do tema, permitindo-nos aplicar os conhecimentos adquiridos de forma prática neste projeto. Expressamos nossa gratidão por sua disponibilidade e prontidão em fornecer feedback construtivo, esclarecer dúvidas e promover discussões enriquecedoras durante as aulas e orientações.

REFERENCES

- [1] A. V. Oppenheim and A. S. Willsky, *Sinais e Sistemas*, 2ª ed. Pearson Universidades, 2010, ISBN: 9780138147570.
- [2] A. V. Oppenheim and R. W. Schaffer, *Processamento em tempo discreto de sinais*, 3ª ed. Pearson Universidades, 2012, ISBN: 9788581431024.

- [3] NTi Audio AG, *Transformação Rápida de Fourier (FFT)*, <https://www.nti-audio.com/pt/suporte/saber-como/transformacao-rapida-de-fourier-fft>, Acessado em 20 de Junho, 2023, s.d.
- [4] *Butterworth filter design*, <https://www.mathworks.com/help/signal/ref/butter.html>, Acessado em 25 de Junho, 2023.
- [5] *Create a simple app or gui using app designer*, https://www.mathworks.com/help/matlab/creating_guis/create-a-simple-app-or-gui-using-app-designer.html, Acessado em: 25/06/2023.
- [6] *Página do prof. dullius*, http://www.ece.ufrgs.br/~cpereira/sistema_sinais/dullius.htm, Acessado em 25 de Junho, 2023.