**Análise da implemetação de filtros FIR e IIR**

**Kaline B. F Mesquita.**

**Sumário**

**1. Introdução…………………………………………………………………………………………..2**

**2. Fundamentação teórica…………………………………………………………………………..2**

Sinais……………………………………………………………………………………………….......2

Processamento de sinais……………………………………………………………………………..3

Propriedade do LTI………………………………………………………………………………….....3

2.1. Filtros FIR……………………………………………………………………………………....4

2.1.1.Projeto de filtros FIR………………………………………………………………………….5 Método das Janelas…………………………………………………………………………………...6

2.2. Filtros IIR…………………………………………………………………………………….....7

2.2.1 Projeto de filtros IIR…………………………………………………………………………..8

Transformação bilinear……………………………………………………………………..…..8

2.2.1. Protótipos analógicos………………………………………………………………………10

Filtro Butterworth……………………………………………………………………………....10

Filtro Chebychev………………………………………………………………………….…....11

Filtro Elíptico…………………………………………………………………………………....12

2.3. Tipos de filtros……………………………………………………………………………….....12

Filtros passa-baixa………………………………………………………………………..…..…..12

Filtros passa-alta…………………………………………………………………………….........12

Filtros passa-faixa………………………………………………………………………….……...12

Filtros Notch………………………………………………………………………………….….....13

**3. Metodologia……………………………………………………………………………………......14**

3.1. Filtragem com IIR…………………………………………………………………………...….14

3.2. Filtragem com FIR…………………………………………………………………………......14

**4. Resultados……………………………………………………………………………………..….14**

4.1. Resultados com FIR………………………………………………………………………......15

4.2. Resultados com IIR…………………………………………………………………………....18

4.3. Comparação: IIR vs. FIR……………………………………………………………………....21

**5. Conclusão………………………………………………………………………………………....22**

**Bibliografia………………………………………………………………………………………..22**

**Anexos………………………………………………………………………………………...…..23**

**Introdução**

Um filtro digital é um sistema que retira alguns componentes de frequência de um sinal, mas considerando um contexto mais amplo, qualquer sistema que modifica certas frequências em relação ao seu estado inicial é também chamado de filtro. Em sistemas de comunicação, particularmente em sinais de áudio é possível encontrar sons não originais juntos ao sinal de voz, e consequentemente interferindo na transmissão de dados, e umas das soluções que pode ser encontrada para eliminar essas interferências no áudio é com a aplicação de filtros.

O objetivo deste trabalho é apresentar a base teórica de um projeto para filtrar um áudio em arquivo WAV. No áudio há uma pessoa falando em um ambiente com muita interferência e tem-se como objetivo reduzir significamente essa interferência sobre o som da voz. Para a realização deste projeto optou-se pela ferramenta Matlab® que é programada na linguagem Matlab, devido a documentação extensa e de fácil acesso,possuindo a vantagem de possuir bibliotecas extensas e de fácil implementação.

Os requisitos para a filtragem do áudio, para fins comparativos, é que precisa haver duas formas de limpeza do sinal: a primeira implementando o IIR, a segunda o FIR. Desse modo, pode-se comparar as vantagens e desvantagem de cada um, tanto a nível de facilidade em programar, como no resultado obtido na filtragem.

O primeiro tópico deste trabalho aborda a fundamentação teórica sobre a área de processamento digital de sinais. A segunda parte aborda sobre a metodologia e processo para filtrar o áudio. Em seguida, é comentado os resultados e conclusões do resultado obtido na filtragem do áudio. Por fim, na seção de Anexos está à íntegra os códigos usados no desenvolvimento dos dois filtros.

**2. Fundamentação teórica**

Os filtros digitais são selecionados de componentes do sinal, isso significa que, analogamente a um utensílio de limpeza, o filtro limpa o sinal dos componentes interferentes na qualidade da informação.Em outras palavras, no processamento digital de sinais, a função dos filtros digitais consiste em retirar as partes indesejadas do sinal, tais como o ruído aleatório, ou para extrair partes úteis do sinal.

Os filtros digitais são classificados em dois grupos: FIR (inglês: finite impulse response) e IIR (inglês, infinite impulse response). A seguir será discutida a base do projeto de filtros digitais. [3]

## **Sinais**

Um sinal, de um modo em geral, é a descrição de um fenômeno ocorrido em um determinado tempo e espaço.[[1]](https://paperpile.com/c/cTtsKG/7XLV) Um simples exemplo é a variação da temperatura ambiente: podemos convencionar em nosso modelo que a variação da temperatura é um sinal e assim registrar a evolução da temperatura ao longo de um tempo.[[2]](https://paperpile.com/c/cTtsKG/bfj3)

## **Processamento digital de sinais**

A área de processamento digital de sinais caracteriza-se em capturar um sinal do mundo real transformar em digital, a fim de efetuar alguma computação.[[3], [4]](https://paperpile.com/c/cTtsKG/MbmQ+L88T) Diversos tipos de sinais podem ser capturados, como sinal de vídeo, temperatura, áudio, assim como diversas computações podem ser realizadas, como soma, subtração, média.[[3]](https://paperpile.com/c/cTtsKG/MbmQ)

Apesar dos sinais do mundo real poderem ser tratados e processados ainda no mundo real, assim sem precisar ter o trabalho de converter para o digital, os meios digitais provém maior velocidade e precisão, este fato é percebido pela otimização de processos diários criados pelo uso da computação.[[3]](https://paperpile.com/c/cTtsKG/MbmQ) Outra vantagem é a flexibilidade, da qual pode-se implementar diferentes operações simplesmente fazendo modificações no software.[[3]](https://paperpile.com/c/cTtsKG/CvZ0)

Uma operação comum a se realizar é filtrar o sinal. Essa operação caracteriza-se em reduzir ou melhorar determinada característica de um sinal ou, simplesmente determinar um intervalo do sinal desejado. Como estamos trabalhando em sinais digitais, essa nossa operação é realizada por um filtro digital, da qual trabalha no tempo discreto com uma amostragem de um determinado sinal.

**Propriedade dos sistemas LTI**

Causalidade

Um sistema é definido como causal caso a saída não depende de valores futuros da entrada. Na prática, um sistema causal é o único tipo de sistema em “tempo-real” que, conseguimos implementar atualmente, considerando que os dados referentes ao futuro não é normalmente uma opção na vida real.Ainda assim, os filtros não-causais, que dependem em parte do futuro, ainda mantém em algumas aplicações(do inglês, batch processing).[1]

Estabilidade

Um sistema LTI é estável para toda entrada limitada é produzida uma saída limitada. Considerando um sistema com a entrada x(t) e saída y(t), se:

No entanto, considerando a entrada e a saída do LTI relacionado a uma convolução, temos:

Dessa forma:

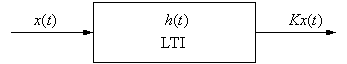
, Resumindo :

O sistema LTI é estável, de acordo com a integral, se a resposta ao impulso é absolutamente integrável, o que significa que o sistema pode ser resumido numa expressão matemática .[10]

Memória

Em sistemas sem memórias,a saída the output é uma função da entrada *x*(*t*) no instante de tempo t somente. Iss significa que não depende de outras entradas passadas e futuras. Portanto, um sistema LTI, sem memória, tem apenas esta forma:

Nesta forma, *K* é o ganho do sistema e deve ser uma constante, caso o contrário o sistema variaria com o tempo.[11]



Considerando *y*(*t*)= *Kx*(*t*),a resposta ao impulso h(t) deve ser da forma de uma unidade de impulse ponderado por uma constante k:

*Invertibilidade*

Um sistema é invertível, se é possível encontrar hI(t) que a entrada original possa ser reconstruída da saída.Para isso manter-se,o sistema precisa ser um a um.

**2.1 Filtros FIR**

Os filtros FIR são definidos pela sequência de impulsos finitos. Neste caso, o operador de convolução engloba apenas um número finito de somas e produtos; se h[n] = 0 para e é possível reescrever usando a propriedade comutativa:

Portanto, somas de convolução envolvendo um suporte finito de resposta ao impulso são sempre bem definidos. Este tipo de filtro apresenta a função de transferência discreta, como apresentado na Equação 1, ou pode ser reescrito como uma função polinomial com potências negativas de Z na Equação 2.

(1)

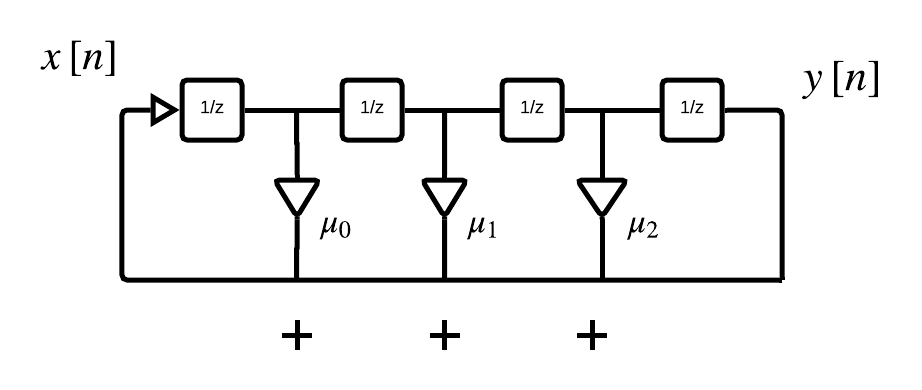
(2)

**Características:**

* Possuem resposta ao impulso finita.
* BIBO estável.
* Regidos por equações de diferença sem recursividade. Exemplo:
* Implementam resposta em fase linear.
* Descrito pela função de transferência:
* Descrito pela resposta em frequência:
* Descrito pela resposta ao impulso:

Um filtro FIR é normalmente implementado por meio de uma série de atraso,

multiplicadores e adicionadores para criar uma saída do filtro, como o da Figura 1.

**Figura 1.**Diagrama de blocos FIR. 

**2.1.1 Projeto de filtros FIR**

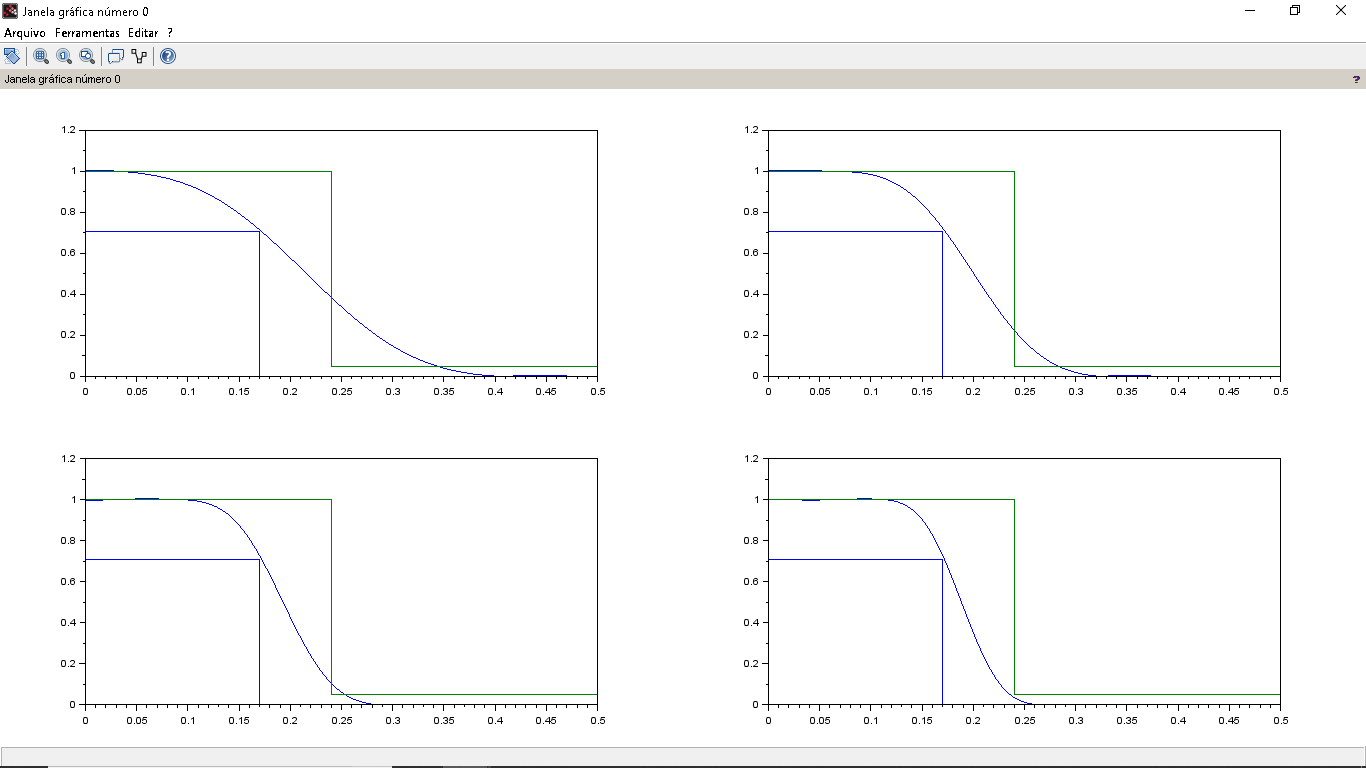
Os projetos de filtro FIR majoritariamente são baseados em aproximações de filtros ideais.O resultado se aproxima da característica ideal proporcionalmente ao aumento da ordem, no entanto, uma desvantagem ocorre pelo aumento da complexidade de implementação.

**Método das janelas**

O projeto de um filtro FIR significa determinar seus coeficientes, de maneira a minimizar o erro de aproximação. Considerando a função como a resposta ao impulso do filtro desejado e a resposta ao impulso do filtro FIR. O erro de aproximação é dado por:

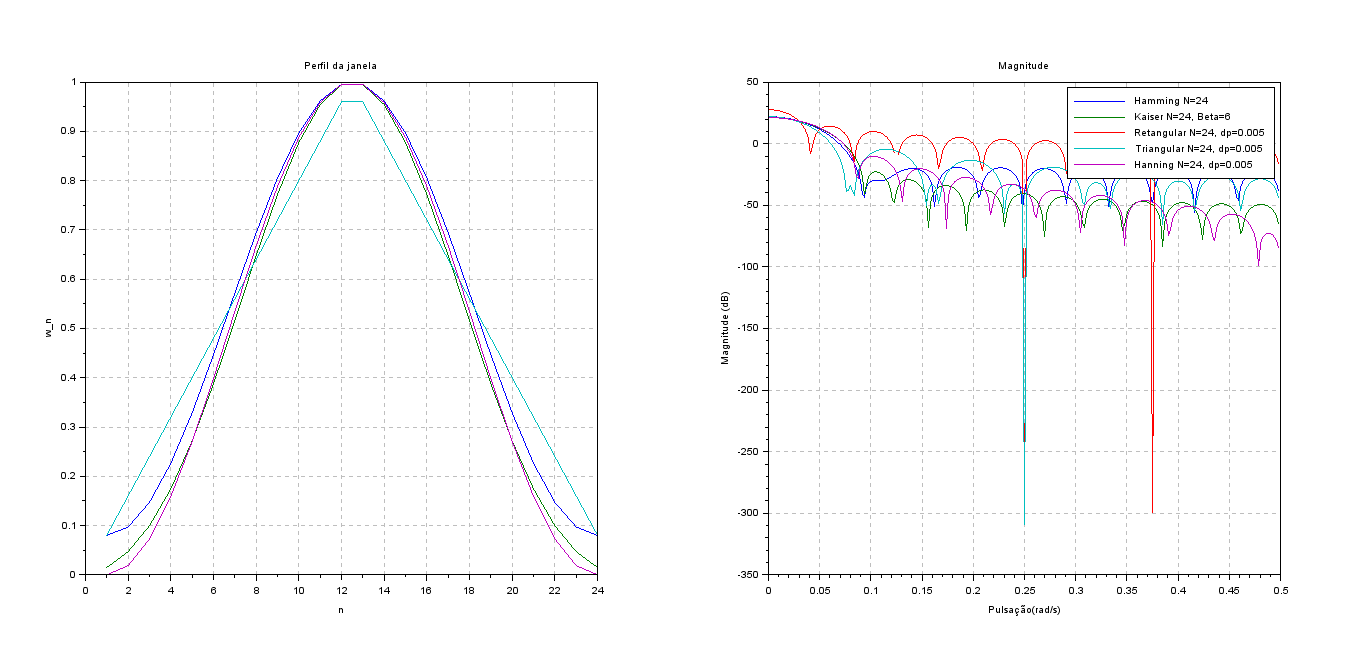
A determinação dos coeficientes é o principal objetivo no projeto de um filtro FIR de ordem N de forma que a função de erro acima seja atenuada. Para isso, é necessário a atribuição dos valores à função:

O método das janelas para o projeto de filtros digitais é rápido, conveniente e robusto, mas geralmente sub-ótimo. Facilmente entendido em termos de teorema da convolução *para* transformadas de Fourier. O método das janelas consiste simplesmente do ‘janelamento’ de uma resposta ao impulso de um filtro teoricamente ideal a partir da escolha de uma função janela adequada , produzindo:

**Figura 2.** Testes de gabarito do filtro FIR. ****

Na figura 4 são apresentadas algumas janelas, na quais foram utilizadas Hamming, Kaiser, Chebyshev, Retangular, Triangular e Hanning utilizando o filtro passa banda de ordem 24:

**Figura 3**. 6 Janelas- (1)Hamming;(2)Kaiser;(3)Retangular;(4)Triangular;(5)Hanning.

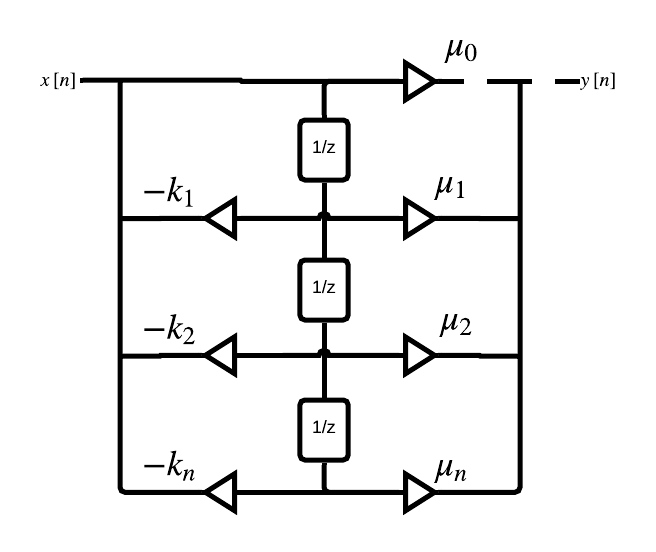


**2.3 Filtros IIR**

Os filtros IIR são definidos pela pela sequência de impulsos infinita. Algumas características básicas dos filtros IIIR são:

* Possuem resposta ao impulso infinita.
* A estabilidade depende da posição dos polos.
* São regidos por equações de diferença de natureza recursiva.
* Não é possível implementar resposta em fase linear.
* Considerando um mesmo grau de aproximação, é necessário uma ordem menor em comparação com um FIR nas mesmas condições.
* Descrito pela equação de diferença:
* Descrito pela função de transferência: .
* Descrito pela resposta em frequência: .

**Figura 4.** Diagrama IIR



**2.2.1 Projetos de filtros IIR**

Nesse projeto, o filtro digital com função de transferência 𝐻(𝑧) é criado a partir de uma mudança de variável, a função 𝐻(𝑠) em Laplace é convertida em 𝐻(𝑧), funções que geram respostas em frequência que atendam as especificações de filtro.[3]

As funções de aproximações mais comuns de um filtro IIR são:

* Butterworth;
* Chebyshev;
* Elíptica.

As especificações de um filtro consistem de:

* Banda passante;
* Banda de rejeição;
* Banda de transição;
* Ripple na banda passante;
* Ripple na banda de rejeição.

**Transformação bilinear**

No projeto de um filtro IIR implementando este método é primariamente determinado são os parâmetros do filtro a ser obtido, o que é chamado gabarito do filtro. Em sequência, um protótipo analógico com função de transferência é projetado. O equivalente discreto definido pela função é obtido usando uma mudança de variável conhecida como transformação bilinear.

A transformação bilinear é uma relação matemática que pode ser utilizado para

converter a função de transferência de um filtro específico no domínio de Laplace

complexo para o domínio Z, e vice-versa. Por este método, a função de transferência do

filtro digital IIR é obtido a partir da função de transferência de um filtro analógico

(protótipo). Abaixo é demonstrado algumas equivalências de matemáticas para resolver o problema de transformação de .[6]

Considerando , obtém-se a função de transferência em Z. Um número de aproximações são usados. A transformação bilinear é:

Portanto, a transformação bilinear de z é:

**Warping e Pré-warping**

É um procedimento realizado para compensar a distorção de frequência (warping), produzida pela transformação bilinear. Assim, as frequências a serem usadas no protótipo analógico devem ser obtidos.

A transformação bilinear é um método que transforma funções adequadas de filtros de para , como a seguir:

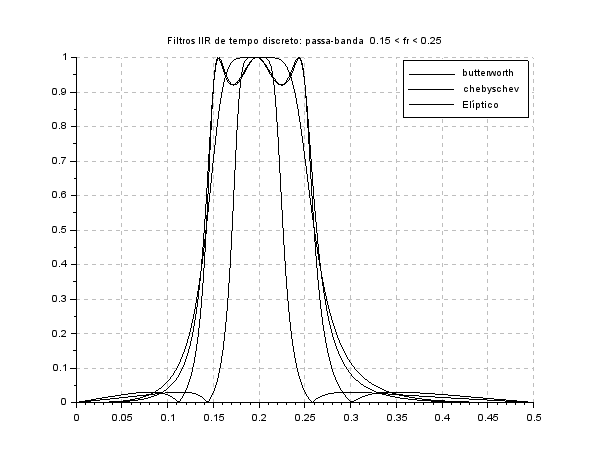


Ambos os filtros irão ter o mesmo destino, mas o resultado não é compatível nas mesmas frequência no domínio analógico e digital. A causa desta distorção ou *warping* é o fato que a aproximação é a mesma aproximação .[6]

Considerando como a frequência no domínio analógico e no domínio da frequência. O warping pode ser reduzido substituindo s com com a descrição da frequência , resumindo:

**Protótipos analógicos**

**Figura 5.** Tipos de filtro analógico IIR.

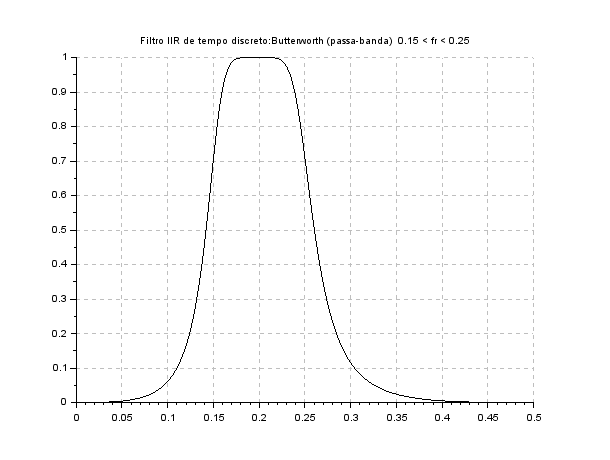


Há três tipos de protótipos analógicos usados em projetos de filtros IIR:

**Filtro Butterworth**

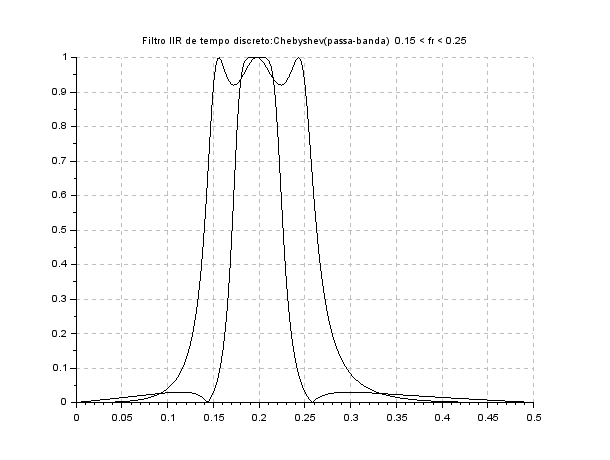
* Possui resposta maximamente plana.
* Não possui oscilações na banda passante e de rejeição.

**Figura 6.** Tipo de filtro Butterworth.



**Filtro Chebyshev**

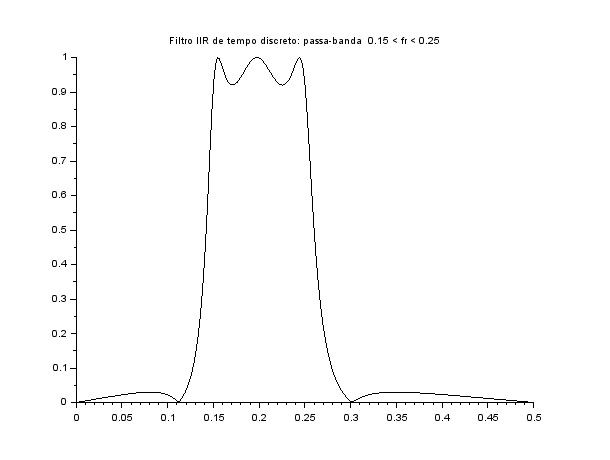
* Tipo I - Possui ondulações na banda passante.
* Tipo II - Possui ondulações na banda de rejeição. Na mesma ordem, possuem menor banda de transição em comparação com o filtro Butterworth.

**Figura 7.** Tipo de filtro Chebyshev

**Filtro Elíptico**

* Possui oscilações na banda de passagem e rejeição.
* Em comparação com os outros filtros analógicos, possui a menor banda de transição de todos.

**Figura 8.** Tipo de filtro Elíptico.

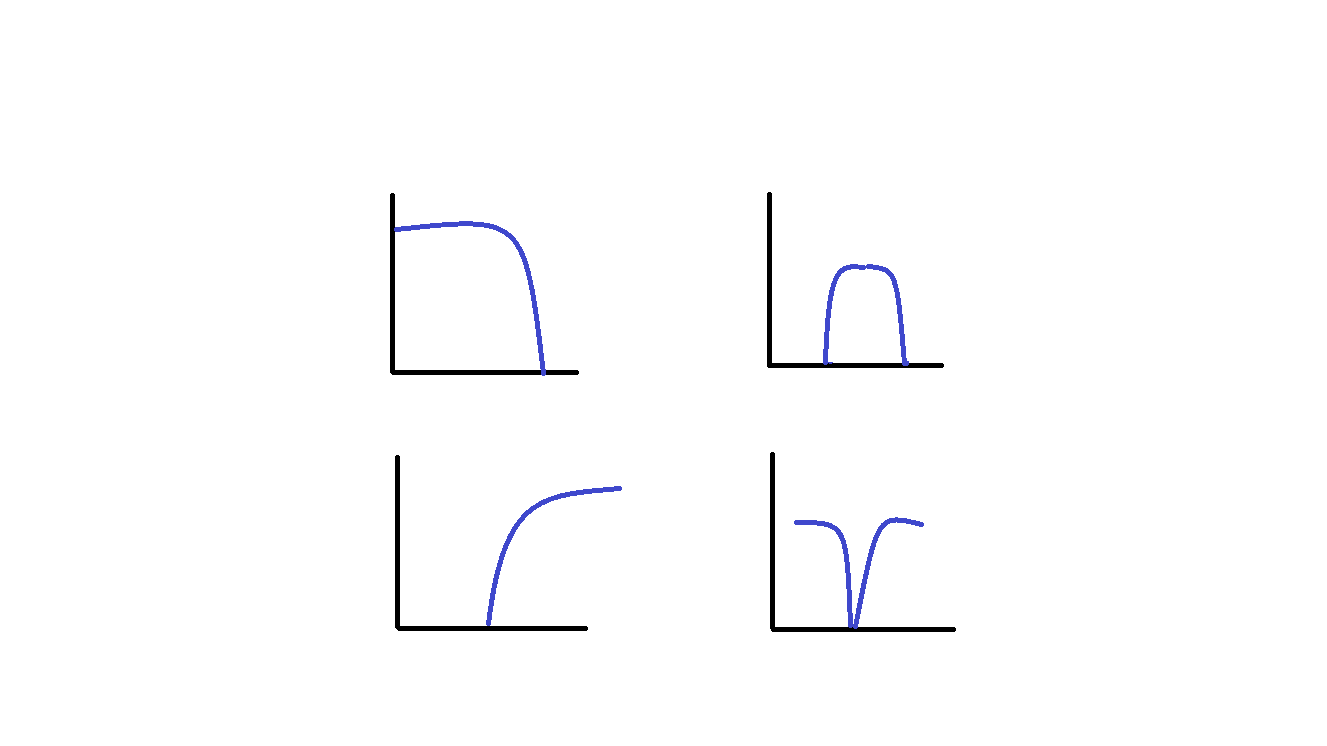


**2.3 Tipos de filtros**

A precisão da resposta em frequência do filtro depende do design do filtro.

Além da divisão dos filtros nos grupos FIR e IIR, existem a subdivisão destes filtros em: passa-baixa(primeiro), passa-alta(segundo inferior), passa-banda(segundo esquerda) e rejeita-banda (último), como está descrito na figura abaixo:

**Figura 9. Filtros passa-baixa, alta, banda e Notch.**



**Filtros passa-baixa**

Um filtro passa-baixa (do inglês, LPF)é um filtro que passa o sinal com uma frequência menor que a frequência de corte selecionada e atenua sinais com frequências maiores que a frequência de corte.

**Filtros pasa-alta**

O filtros passa alta é complementar ao passa alta. Neste caso, ele deixa passar sinais nas frequências maiores que a frequência de corte e atenuar sinais na frequência menor que a de corte.

**Filtros passa-banda**

Um filtro passa banda é caracterizado por deixar passar frequências numa determinada faixa, enquanto atenua as frequências externas a esta faixa.

**Filtros Notch**

O filtro Notch, também conhecido por rejeita-banda, este tipo de filtro passa todas as frequências acima e abaixo de um determinado intervalo definido pelos valores dos componentes. O filtro Notch pode ser obtido como resultado da junção de um filtro passa baixa e um filtro passa alta,assim como o projeto de passa banda, com exceção que são conectados as duas seções do filtro em paralelo, ao contrário de em série.

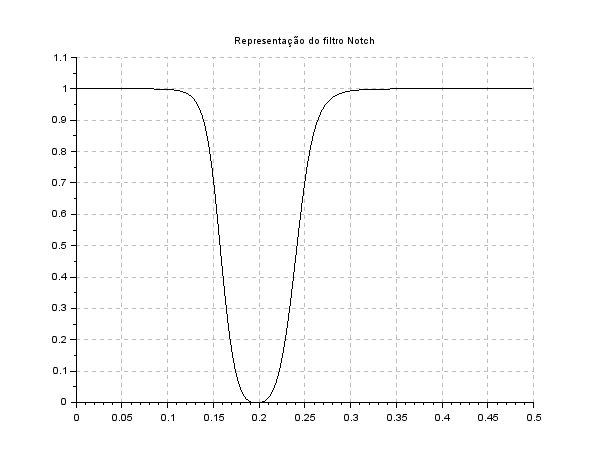
* Um filtro de para-banda funciona para extrair frequências do sinal que estão em um determinado intervalo, consequentemente proporcionando uma passagem flexível apenas para frequências externas a este intervalo.
* O filtro de para-banda podem ser feitos colocando um filtro passa-baixa em paralelo com um filtro passa-alta.Comumente,ambas as seções dos filtros passa-baixas e passa altas são de configuração “T”.
* A frequência de atenuação máxima é chamada de frequência de notch.

Por padrão, O filtros Notch é descrito matematicamente por uma função de transferência:



Na qual é a frequência central excluída e é a largura de banda desta frequência excluída.

**Figura 10**. Representação de um filtro rejeita-faixa.



**3. Metodologia**

Para a implementação deste projeto, foi utilizado um sinal de voz de áudio,

com ruídos e interferências, e o mesmo deveria passar por um tratamento de filtragem, a um patamar possível de verificação da informação que o sinal de voz transmite. Mais

precisamente foi utilizado um sinal áudio onde a mesma continha o ruído de uma sirene,

e vibrações sendo emitida ao mesmo tempo em que o tom da fala humana, deixando a

voz humana sem som audível, por causa das interferências.

Para a análise do sinal, é utilizado o software de computação numérica Matlab®,

que foi de grande importância no processamento do sinal utilizado. O mesmo captura

informações do arquivo de áudio, como por exemplos, números de amostras, frequência,

e número de bits. Com essas informações é possível plotar o espectro do sinal de áudio e

verificar o seu comportamento.

Analisando o espectro do sinal é possível verificar quais ruídos devem ser filtrados.

Isso porque a fala humana está no intervalo de 50Hz à 4000Hz, assim, frequências

maiores que as do intervalo podem ser descartadas do sinal que realmente deseja-se ter.

Para o procedimento de filtragem, foram utilizados os dois tipos de filtros, FIR e IIR.

**3.1.Filtragem com FIR**

.

Para a filtragem com o filtro FIR, foi utilizado o método das janelas, de onde foi dividido em dois estágios. No primeiro estágio foi desenvolvido uma equação que gera o rejeita-faixa. No segundo estágio foi utilizado uma função própria do Matlab© que gera o filtro passa-baixa.

**3.2.Filtragem com IIR**

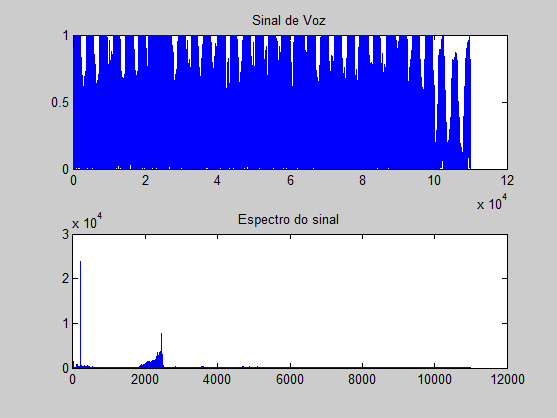
Para a filtragem do sinal com o filtro IIR, foi utilizado dois tipos de projetos de filtros, divididos em duas etapas, no primeiro estágio é aplicado duas vezes o filtro rejeita-faixa de notch para filtrar uma frequência indesejada, que se encontra no intervalo 11 de frequências da voz humana, e no segundo estágio é utilizado um filtro passa-baixa de butterworth.

O passa-baixa de butterworth foi escolhido por ser o mais adequado para retirar as frequências que estão acima da faixa de frequências da voz humana, além disso, esse filtro possui uma resposta mais próxima à ideal. O rejeita-faixa de notch foi escolhido por conseguir retirar uma pequena faixa de frequência que se encontrava dentro do intervalo de frequências da voz.

**4. Resultados**

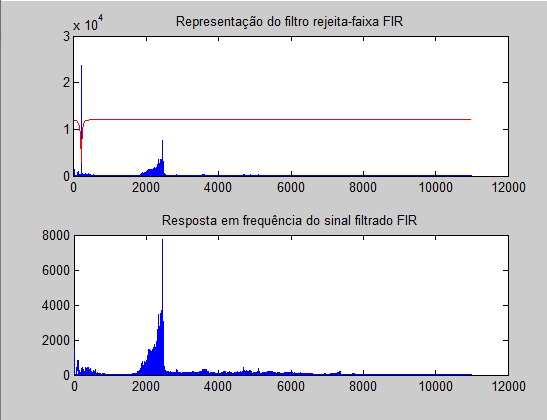
Os dois tipos de filtros filtram o mesmo arquivo de áudio disponibilizado, e os resultados se comportam de maneiras semelhantes. Na figura 10 é apresentado o sinal do áudio e o seu espectro em função da frequência, comum aos dois tipos de filtros.

**Figura 11.** PlotagemSinal de áudio e espectro na frequência.



**Resultados com FIR**

Na filtragem do filtro FIR, foi verificado que a frequência de 200Hz interferir no sinal de áudio, com isso, foi projetado um filtro rejeita-faixa com as seguintes características. fc=200 Hz, de frequência de corte; r=0.9878, coeficiente de regulação; wc= 0.0570, de frequência de corte discreta. Na Figura 11 é apresentado o filtro rejeita-faixa do tipo FIR, projetado com os valores acima. Percebe-se que o filtro eliminou somente a frequência de 200Hz do sinal, a Figura 12 apresenta a respectiva resposta ao impulso desse filtro.

**Figura 12.** Plotagemdo Notch (vermelho) e resposta em frequência. 

Na Tabela 1 são mostrados os respectivamente os coeficientes numerador e denominador do filtro rejeita-faixa FIR implementado neste projeto.

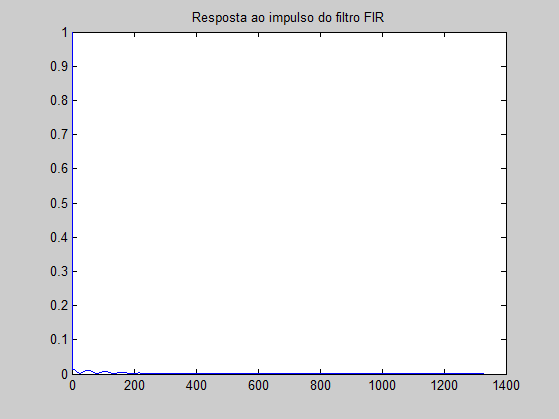
**Tabela 1**. Coeficientes dos filtros rejeita-faixa FIR.

|  |  |
| --- | --- |
| Numerador | 1.0000 -1.9968 1.0000 |
| Denominador | 1.0000 -1.9724 0.9757 |

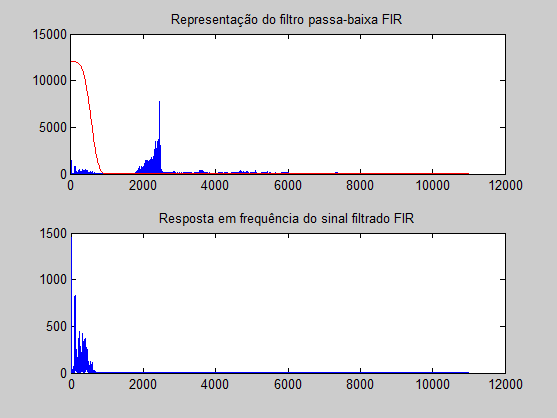
Foi verificado que frequências acima de 900Hz deveriam ser eliminadas do sinal para se alcançar resultados mais satisfatórios, com isso foi empregado um filtro passa-baixa FIR com as seguintes características:

* N=100, ordem do filtro;
* Wn=0.04, frequência de corte.

Na Figura 13 é apresentado o filtro passa-baixa do tipo FIR, projetado com os valores acima. Percebe-se que o filtro eliminou as frequências acima de 900 Hz do sinal, a Figura 14 apresenta a respectiva resposta ao impulso desse filtro.

**Figura 13.** Plotagemda respostaao impulso do rejeita-faixa FIR.

**Figura 14**. Representação do filtro passa-baixa FIR (vermelho) filtrando o sinal, e a forma do sinal depois de filtrado.

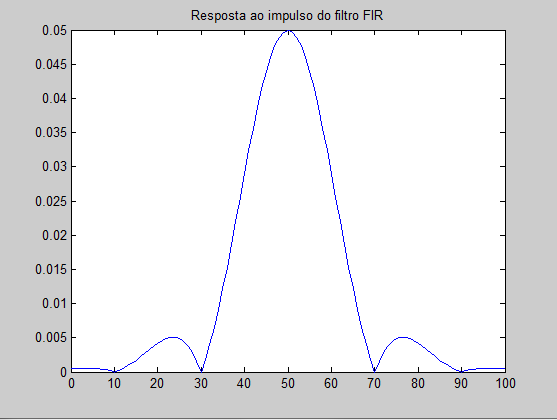


A ordem do filtro passa-baixa é N=100, ocasionando, o numerador B ter um tamanho de coeficientes equivalente a 100. Com isso, na Tabela 2 optou-se por apresentar somente alguns coeficientes, entretanto no console do Matlab©, são apresentado todos os coeficientes do numerador e denominador. Assim no numerador só serão apresentados os 3 primeiros e 3 últimos coeficientes do console.

**Tabela 2**. Coeficiente do filtro passa-faixa FIR.

|  |  |
| --- | --- |
| Numerador | 0.0005 0.0005 0.0005 ….. 0.0005 0.0005 0.0005 |
| Denominador | 1 |

**Figura 15.** Resposta ao impulso do filtro passa-baixa FIR.

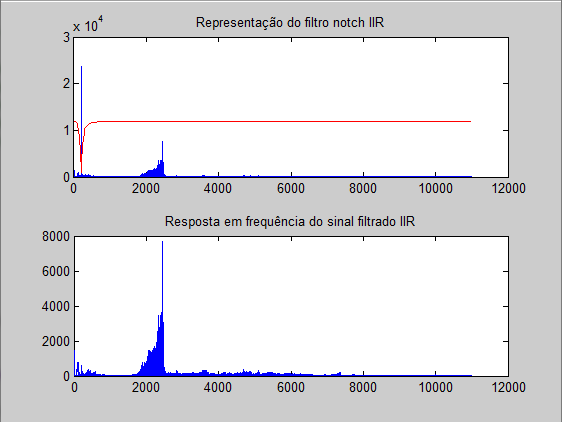


**Resultados com o IIR**

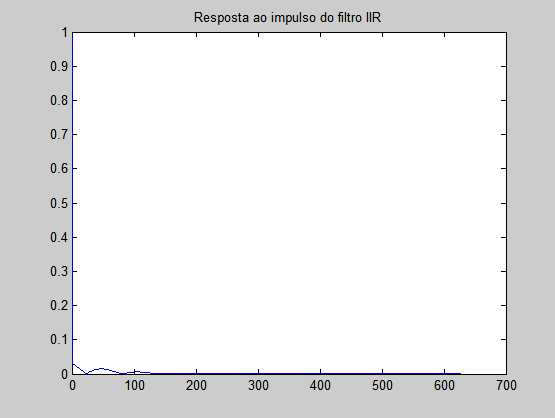
Semelhante como na filtragem do filtro FIR, foi verificado que a frequência de 300Hz interferia no sinal de áudio, com isso, também foram projetados dois filtros notch com as seguintes parâmetros Wo=0.0275 e BW=0.01 utilizando a função iirnotch. Foram projetados dois filtros notch para se alcançar mais consistência na filtragem. Na Figura 16 é apresentado o filtro notch do tipo IIR, projetado com os parâmetros mencionados acima. Percebe-se que o filtro eliminou somente a frequência de 200Hz do sinal, a Figura 17 apresenta a respectiva resposta ao impulso desse filtro.

**Figura 16.** Representação do filtro rejeita-faixa IIR (cor vermelho) filtrando o sinal,

e a forma do sinal depois de filtrado.



**Figura 17**. Resposta ao impulso do filtro rejeita faixa IIR.



Na Tabela 3 são apresentados os coeficientes do numerador e denominador do filtro rejeita-faixa IIR implementado neste projeto.

**Tabela 3.** Coeficientes dos filtros rejeita-faixa IIR.

|  |  |
| --- | --- |
| Numerador | 0.9845 -1.9659 0.9845 |
| Denominador | 1.0000 -1.9659 0.9691 |

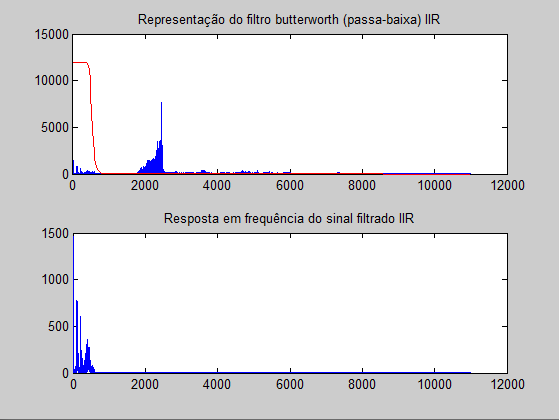
Para retirar as frequências acima de 900Hz também foi utilizado um filtro passa-baixa, mas de butterworth do tipo IIR. Este filtro foi empregado com as seguintes características:

N=10; ordem do filtro;

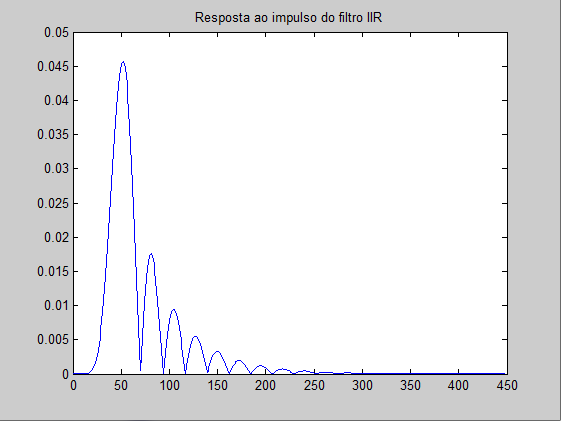
Wn=0.05, frequência de corte.

Na Figura 18 é apresentado o filtro passa-baixa do tipo FIR, projetado com os valores acima. Percebe-se que o filtro eliminou as frequências acima de 900Hz do sinal, a Figura 19 apresenta a respectiva resposta ao impulso desse filtro.

**Figura 18.** Representação do filtro passa-baixa IIR (vermelho) filtrando o sinal, e a forma do sinal depois de filtrado.



**Figura 19.** Resposta ao impulso do filtro passa-baixa IIR.



Na Tabela 4 são apresentados os coeficientes do numerador e denominador do filtro passa-baixa IIR de ordem N=10 implementado neste projeto.

**Tabela 4.** Coeficientes dos filtros passa-baixa IIR.

|  |  |
| --- | --- |
| Numerador | 0.0020 0.0201 0.0906 0.2417 0.4230 0.5076 0.4230 0.2417 0.0906 0.0201 0.0020 |
| Denominador | 1.0000 -9.0963 37.2722 -90.5916 144.6357 -158.4931 120.7194 -63.1060 21.6674 -4.4123 0.4047 |

**Comparação: FIR vs. IIR**

Os dois tipos filtros realização a filtragem do arquivo de áudio, de forma que o mesmo pudesse se tornar audível. Os filtros rejeita-faixa de FIR e IIR também funcionam de maneiras semelhantes na filtragem da frequência, assim como também, os filtros passa-baixa FIR e IIR, no final o resultado foi semelhantes nos dois tipos de filtros. O filtro IIR obteve um comportamento melhor considerando a ordem do filtro, 10 vezes melhor que o filtro FIR. Considerando a opinião do autor deste trabalho, o som depois de filtrado utilizando o filtro IIR se mostrou mais adequado audivelmente.

**5. Conclusão**

Conclui-se que implementação de filtros digitais para o tratamento de sinais de áudio é bastante útil, principalmente no sentido de ser observado na prática a filtragem de sinais. Os resultados se comportaram de maneira fiel a fundamentação teórica, e foi possível utilizar vários projetos de filtros na implementação deste trabalho. Também a escolha dos projetos de filtros foram as mais adequadas para a realização deste trabalho. Por fim, a utilização de software matemáticos foi de grande importância para calcular e simular os resultados obtidos.

**Bibliografia**

[1] M. V. Paolo Prandoni, Signal Processing For Communications.

[2] A. Device, “A Beginner’s Guide to Digital Signal Processing (DSP)”, Analog Device. [Online]. Disponível em:http://www.analog.com/en/design-center/landing-pages/001/beginners-guide-to-dsp.html. [Acessado: 15-jun-2019]

[3] HAYKIN, SIMON S.; VAN VEEN, Barry. Sinais e sistemas. Bookman, 2001.

[4 ] “Window” [Online]. Disponível em:https://www.recordingblogs.com/wiki/window. [Acessado: 15-jun-2019]

[5] “High pass filter - Wikimedia Commons”. [Online]. Disponível em: https://commons.wikimedia.org/wiki/File:High\_pass\_filter.svg. [Acessado: 15-jun-2019]

[6] “Design IIR filters”. [Online]. Disponível em: http://lucienne.landau.free.fr/Erasmus/userfiles/Cours/Signals/Cours/1190627256.doc. [Acessado: 15-jun-2019]

[7 ]“Bandpass Filter - Wikipedia”. [Online]. Disponível em: https://en.wikipedia.org/wiki/File:Bandpass\_Filter.svg. [Acessado: 15-jun-2019]

[8] “Band-Reject Filter - Wikimedia Commons”. [Online]. Disponível em: https://commons.wikimedia.org/wiki/File:Band-Reject\_Filter.svg. [Acessado: 15-jun-2019]

[9] “Digital Filter Design”. [Online]. Disponível em: https://www.mikroe.com/ebooks/digital-filter-design/finite-impulse-response-fir-filter-design-methods. [Acessado: 15-jun-2019].

[10] “Causality and Stability for LTI Systems”.[Online]. Disponível em:http://fourier.eng.hmc.edu/e101/lectures/LTI/node3.html.[Acessado:18-jun-2019]

[11] “Properties of continous time LTI systems”.[Online].Disponível em https://class.ece.uw.edu/235dl/EE235/Project/lesson9old/lesson9.html.[Acessado:18-jun-2019]

**Anexos**

**FIR**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Projeto filtros FIR

%Kaline B.F Mesquita

%Ferramentas:MÉTODO DAS JANELAS

%Descrição: Neste código são utilizado dois filtro FIR para a filtragem de

% uma som. No som há uma pessoa falando, uma sirene e um chiado.

% O código filtra o som, retirando o ruído da sirene e do chiado. Resultando no som da

% voz audível.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clc; clear all;

% Carregamento do sinal de áudio

[x,fs]=audioread('C:\Users\Alunos\Downloads\fala\_sirene\_tm0.wav');

nbits=16;

% Play na musica original

%wavplay(x,fs);

% Capturando informações

% Ts - Período de amostragem

% t -Intervalo de amostragem

% fa - fa percorre vetor de amostras dos sinal

% f - Vetor de frequências em Hz

% X - Espectro de frequência do sinal de áudio

Ts = 1/fs;

t = 0:Ts:1-Ts;

fa=[0:1:length(x)-1];

f=fa.\*fs/(length(x)-1);

X=abs(fft(x));

% Apresentação de plots

subplot(2,1,1);

plot(fa,abs(x));

title('Sinal de Voz');

subplot(2,1,2);

plot(f(1:round(length(x)/2)),X(1:round(length(x)/2)));

title('Espectro do sinal');

% ------------------------- 1° seção de filtragem ---------------

% Filtro rejeita-faixa.

% fc - Frequência a ser eliminada

% r - coeficiente variável do raio

% wc - Conversão da frequência de corte

% b - Coeficientes do numerador do filtro

% a - Coeficientes do denominador do filtro

% [H1,W1] - Resposta em Frequência do Filtro Notch

% e - Filtrando o sinal

% E - Resposta em frequência do sinal filtrado

% [I,T] - Resposta ao impulso do filtro FIR

fc = 200;

r = 0.9878;

wc = (2\*pi\*fc)/fs;

b = [1 -2\*cos(wc) 1];

a = [1 -2\*r\*cos(wc) r^2];

[H1,W1] = freqz(b, a, 512);

e = filter(b,a,x);

E = abs(fft(e));

[I,T] = impz(b\*10000, a\*10000);

% Coeficientes do segundo filtro notch

disp('Coeficientes do filtro rejeita-faixa FIR');

disp(b);

disp(a);

% Apresentando de plots

figure;

subplot(2,1,1);

plot(f(1:round(length(x)/2)),X(1:round(length(x)/2)));

hold on

plot(W1\*(fs/(2\*pi)), abs(H1\*12000), 'r');

title('Representação do filtro rejeita-faixa FIR');

subplot(2,1,2);

plot (f(1:round(length(e)/2)), E(1:round(length(e)/2)));

title('Resposta em frequência do sinal filtrado FIR');

figure;

plot(T,abs(I));

title('Resposta ao impulso do filtro FIR');

% ------------------------- 2° seção de filtragem ---------------

% Filtro passa-baixa.

% [H1,W1] - Resposta em Frequência

% y - Filtrando o sinal

% Y - Resposta em frequência do sinal filtrado

% [I1, T1] - Resposta ao impulso do filtro FIR

B=fir1(100,0.05, 'low');

A=1;

[H1,W1] = freqz(B, 1, 1024);

y = filter(B,A,e);

Y = abs(fft(y));

[I1,T1] = impz(B, A);

% Coeficientes do segundo filtro notch

disp('Coeficientes do filtro passa-baixa FIR');

disp(B);

disp(A);

% Apresentando de plots

figure;

subplot(2,1,1);

plot(f(1:round(length(e)/2)),E(1:round(length(e)/2)));

hold on

plot(W1\*(fs/(2\*pi)), abs(H1\*12000), 'r');

title('Representação do filtro passa-baixa FIR');

subplot(2,1,2);

plot (f(1:round(length(y)/2)), Y(1:round(length(y)/2)));

title('Resposta em frequência do sinal filtrado FIR');

figure;

plot(T1,abs(I1));

title('Resposta ao impulso do filtro FIR');

% Salvando sinal filtrado

audiowrite('C:\Users\Alunos\Desktop\fala\_sirene\_tmFIR.wav',y,fs);

% Play no voz filtrado

[x1,fs1]=audioread('C:\Users\Alunos\Desktop\projetopds\fala\_sirene\_tmFIR.wav');

am=(1)\*x1; % Amplificação do sinal.

audioplayer(am,fs1);

disp('100%!');

**IRR**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Projeto filtros IIR

%kaline B.F Mequita

%Ferramentas: FILTRO DE BUTTERWORTH E NOTCH.

%Descrição: Neste código são utilizado dois filtro IIR para a filtragem de

% uma som. No som há uma pessoa falando, uma sirene e um chiado.

% O código filtra o som, retirando o ruído da sirene e do chiado. Resultando no som da

% voz audível.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clc; clear all;

% Carregamento do sinal de audio

[x,fs]=audioread('C:\Users\Alunos\Downloads\fala\_sirene\_tm0.wav');

nbits=16;

% Play na musica original

%wavplay(x,fs);

%1° parte:

%Ts - período de amostragem

%t - intervalo de amostragem

%fa - percorre um vetor de amostras do sinal.

%f - vetor em frequência em Hz

%X - espectro de frequência do sinal de áudio

% Capturando informações

Ts = 1/fs;

t = 0:Ts:1-Ts;

fa=[0:1:length(x)-1];

f=fa.\*fs/(length(x)-1);

X=abs(fft(x));

% Apresentando de plots

subplot(2,1,1);

plot(fa,abs(x));

title('Sinal de Voz');

subplot(2,1,2);

plot(f(1:(round(length(x)/2))),X(1:(round(length(x)/2))));

title('Espectro do sinal');

% ------------------------ 1° seção de filtragem ---------------

% 1° Filtro notch

%Wo e BW - Parâmetros de entrada

%[b,a] - Filtro Notch

%[H1, W1] - resposta em frequência do filtro

%Y - Resposta em frequência do sinal filtrado

Wo = 0.0275; BW=0.01;

[b,a] = iirnotch(Wo,BW);

[H1, W1]= freqz(b, a, 512);

y = filter(b,a,x);

Y = abs(fft(y));

% Coeficientes do primeiro filtro notch

disp('Coeficientes do primeiro filtro notch');

disp(b);

disp(a);

% 2° Filtro notch

%Wo e BW - Parâmetros de entrada

%[b1,a1] - Filtro Notch

%[H1, W1] - Resposta em frequência do filtro

%e - filtrando o sinal

%E - resposta em frequência do sinal filtrado

%[G, U] - Resposta ao impulso do filtro IIR

Wo = 0.018; BW=0.01;

[b1,a1] = iirnotch(Wo,BW);

[H1, W1]= freqz(b1, a1, 512);

e = filter(b1,a1,y);

E = abs(fft(e));

[G,U] = impz(b1, a1);

% Coeficientes do segundo filtro notch

disp('Coeficientes do segundo filtro notch');

disp(b1);

disp(a1);

% Apresentação de plots

figure;

subplot(2,1,1);

plot(f(1:(round(length(x)/2))),X(1:(round(length(x)/2))));

hold on;

plot(W1\*(fs/(2\*pi)), abs(H1\*12000), 'r');

title('Representação do filtro notch IIR');

subplot(2,1,2);

plot (f(1:round(length(e)/2)), E(1:round(length(e)/2)));

title('Resposta em frequência do sinal filtrado IIR');

figure;

plot(U,abs(G));

title('Resposta ao impulso do filtro IIR');

% -------------------------- 2° seção de filtragem ---------------

% Filtro de passa-baixa de butterworth

%[b1, a2] - Parâmetros de entrada

%[H1 , W1] - Resposta em frequência do filtro

%s - Filtrando o sinal

%S- Filtrando o sinal

%[T, T] Resposta ao impulso do filtro IIR

[b2, a2] = butter(10, 0.045, 'low');

[H1, W1]= freqz(b2, a2, 512);

s = filter(b2,a2,e);

S = abs(fft(s));

[I,T] = impz(b2, a2);

% Coeficientes do filtro de butterworth

disp('Coeficientes do filtro de butterworth');

disp(b2);

disp(a2);

% apresentando de plots

figure;

subplot(2,1,1);

plot(f(1:(round(length(e)/2))),E(1:(round(length(e)/2))));

hold on;

plot(W1\*(fs/(2\*pi)), abs(H1\*12000), 'r');

title('Representação do filtro butterworth (passa-baixa) IIR');

subplot(2,1,2);

plot (f(1:round(length(s)/2)), S(1:round(length(s)/2)) );

title('Resposta em frequência do sinal filtrado IIR');

figure;

plot(T,abs(I));

title('Resposta ao impulso do filtro IIR');

% Salvando sinal filtrado

audiowrite('C:\Users\Alunos\Downloads\fala\_sirene\_tmIIR.wav',s,fs);

% Play no voz filtrado

[x1,fs1]=audioread('C:\Users\Alunos\Desktop\projetopds\fala\_sirene\_tmFIR.wav');

am=(1)\*x1; % Amplificação do sinal.

audioplayer(am,fs1);

disp('100%!');