

Relatório de Projeto 2019.1

ES265 - Processamento Digital de Sinais

Igor Dias da Silva - ids3@cin.ufpe.br

10 de Junho de 2019

I. SINAIS

A. Questão 1.1

Para o projeto do filtro foi escolhida a janela de Kaiser. Dadas as especificações na questão, M e β foram calculados. A questão foi implementada em python e a função `signal.firwin()` da biblioteca `scipy` foi usada para criar o filtro com a janela de Kaiser dados M e β . Os resultados são apresentados na figura 1.

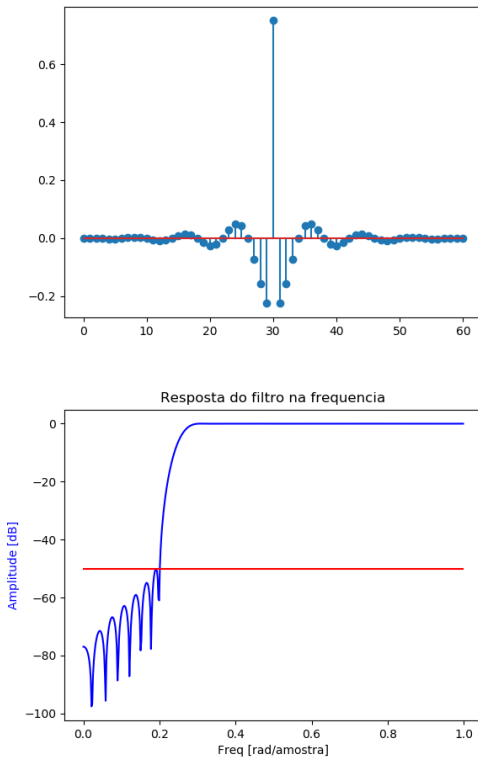


Fig. 1. O filtro projetado e sua resposta na frequência.

B. Questão 1.2

O áudio com eco foi gerado conforme especificado na questão. Contudo, a questão foi implementada em python e ao ler o arquivo, a taxa de amostragem recebida pela função é 22050. Consequentemente, o eco existe e é perceptível, mas não tão perceptível como no áudio de outros alunos já que a taxa de amostragem é maior e portanto o atraso de 500 amostras é menor.

Foram definidos o numerador e denominador da função de transferência, que são os parâmetros da função `lfilter()`, usada para filtrar o sinal com eco gerado. Dos resultados, o melhor foi com $a = +0,5$. O áudio com eco e o melhor resultado são mostrados na figura 2. Os outros áudios ficaram bem distorcidos, com uma voz metálica. Os arquivos de áudio e a visualização dos sinais gerados estão disponíveis na pasta da questão.

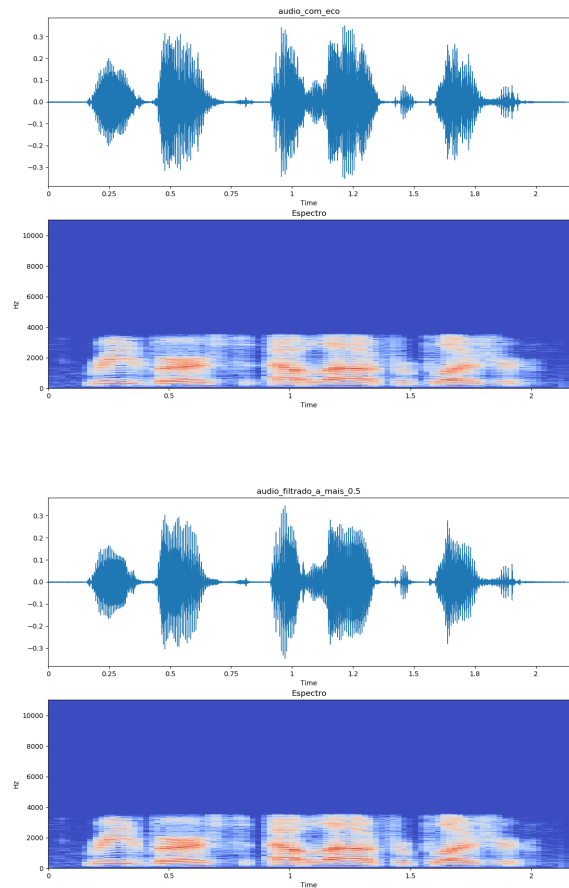


Fig. 2. A forma de onda e o espectro do sinal antes e depois de ser filtrado respectivamente.

II. IMAGEM

A. Questão 2.1

Essa questão foi implementada em python. A ideia para resolvê-la foi atenuar o efeito usando um blur. Assim, as variações bruscas seriam eliminadas. O que de fato ocorreu. Entretanto, usando um filtro gaussiano, não foi possível remover totalmente o efeito sem borrar tanto a imagem. O resultado pode ser visto na figura 3.



Fig. 3. A imagem sem o efeito de rings após o blur.

B. Questão 2.2

Essa questão foi implementada no MATLAB. A ideia geral foi separar bem as regiões e então contá-las. Para isso, o primeiro passo foi usar uma técnica de detecção de bordas. Canny e Sobel foram testados, e Sobel produziu menos ruídos. Após a detecção das bordas, foi utilizada a função `bwconncomp()` que recebe o resultado da detecção de bordas e conta o número de regiões. Com isso o algoritmo contou 45 regiões na imagem.

Para chegar ainda mais próximo do resultado desejado foi utilizado um filtro gaussiano no início do processo para borrar a imagem e reduzir a quantidade de bordas irrelevantes encontradas pelo Sobel. Para o Sobel com $\sigma = 1.3$, o algoritmo passou a contar 35 regiões.

C. Questão 2.3

Essa questão foi implementada no MATLAB. Aqui a imagem foi convertida para HSV, onde era mais fácil manipular as cores desejadas. A imagem foi percorrida pixel a pixel e os pixels com hue menor que 0.15, que estão próximos ao tom do vermelho, tiveram 0.6 adicionado ao seu hue, jogando o pixel para o tom azul.

Já os pixels com hue menor que 0.45 e maior que 0.15, próximos ao tom do verde, tiveram 0.3 retirado indo para o magenta. Para esses pixels, 0.3 foi retirado do value. Esses valores foram definidos através da experimentação para melhorar a visualização do resultado final, que pode ser visto na figura 4.

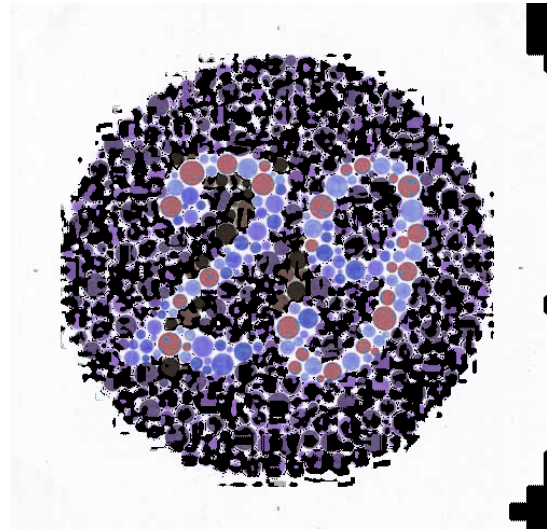


Fig. 4. Resultado após variação dos tons dos pixels.

III. VÍDEO

A. Questão 3.1

Essa questão foi implementada no MATLAB. A ideia foi simplesmente verificar se um pixel mudou de um frame para o outro. Em caso positivo, o pixel é replicado no vídeo de saída e em caso negativo ele recebe valor zero no vídeo de saída. Foi necessário utilizar um threshold nessa comparação para melhorar o resultado, visto que existem pequenas alterações nos valores dos pixels de um frame pro outro que não decorrem do movimento dos carros.

IV. VOZ E SOM

A. Questão 4.1

Essa questão foi implementada em python. A primeira ideia foi usar um filtro passa-faixa para pegar só a parte mais importante do sinal de voz assim como nos sistemas de telefonia antigamente, na esperança de que ao menos parte do ruído dos carros fosse eliminado. O que não ocorreu já que o ruído está muito espalhado na frequência. Foi utilizado um filtro Butterworth passa-faixa de 2 Khz a 4 Khz mas o sinal só perdeu qualidade e manteve o ruído. Os resultados podem ser vistos nas figura 5 e 6.

B. Questão 4.2

A segunda ideia foi elevar as amostras da onda ao quadrado na tentativa de distanciar os sinais mais fortes que em sua maioria são a voz do ruído que em boa parte do áudio é mais baixo, e então eliminar as amostras com valor abaixo de um limiar. Dessa forma, esperava-se que fosse possível eliminar o ruído quando não houvesse fala, e quando o interlocutor falasse, o sinal seria o mesmo do original. Entretanto, o resultado não correspondeu a expectativa. O resultado pode ser visto na figura 7.

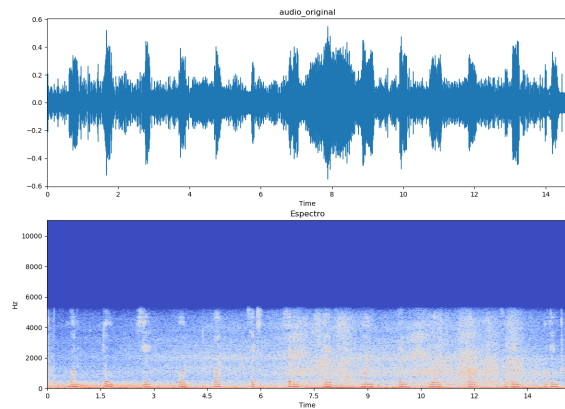


Fig. 5. O sinal e seu espectro antes de ser filtrado.

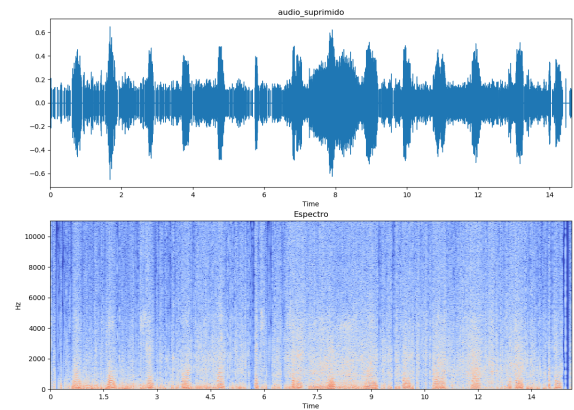


Fig. 7. O sinal com ruído senoidal e seu espectro após ser suprimido, respectivamente.

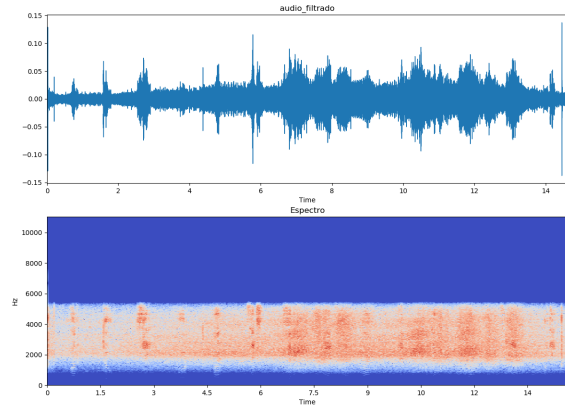


Fig. 6. O sinal e seu espectro depois de ser filtrado.

C. Questão 4.2

Mais uma vez a ideia foi replicar a ideia dos sistemas de telefonia antigos, o mesmo filtro butterworth foi utilizado de 2 KHz a 4 KHz. Assim, o áudio perdeu qualidade e ficou mais baixo. Porém o ruído senoidal foi eliminado. Os resultados podem ser vistos na figura 8.

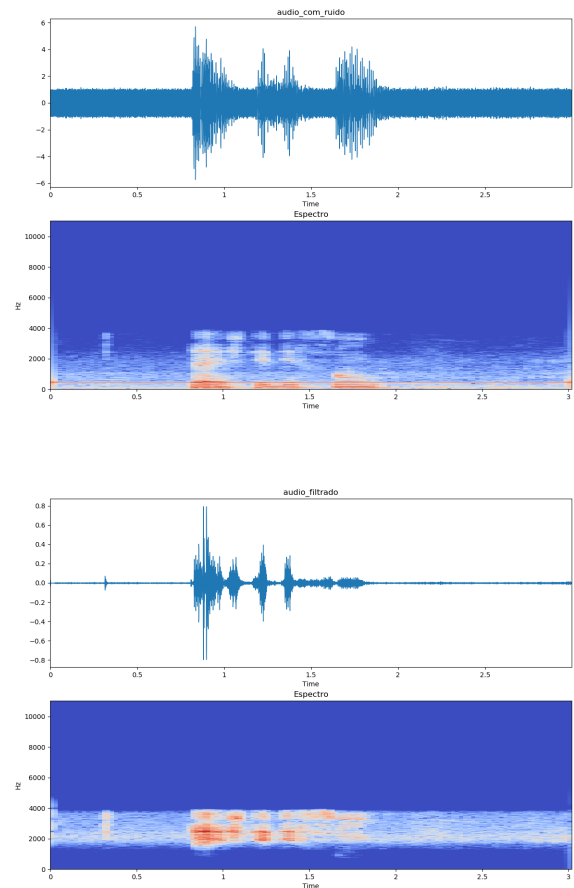


Fig. 8. O sinal com ruído senoidal e seu espectro antes e depois de ser filtrado.