

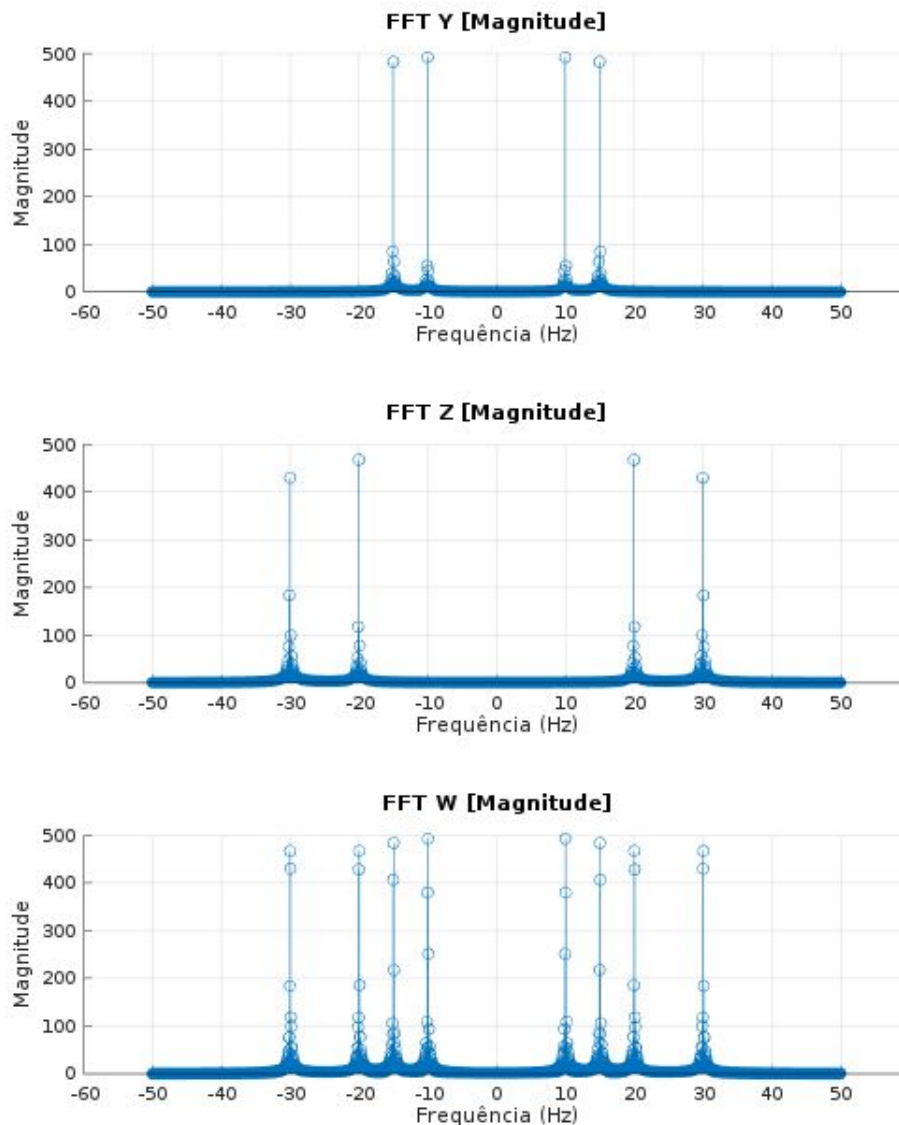
## Processamento Digital de Sinais

### - Projeto -

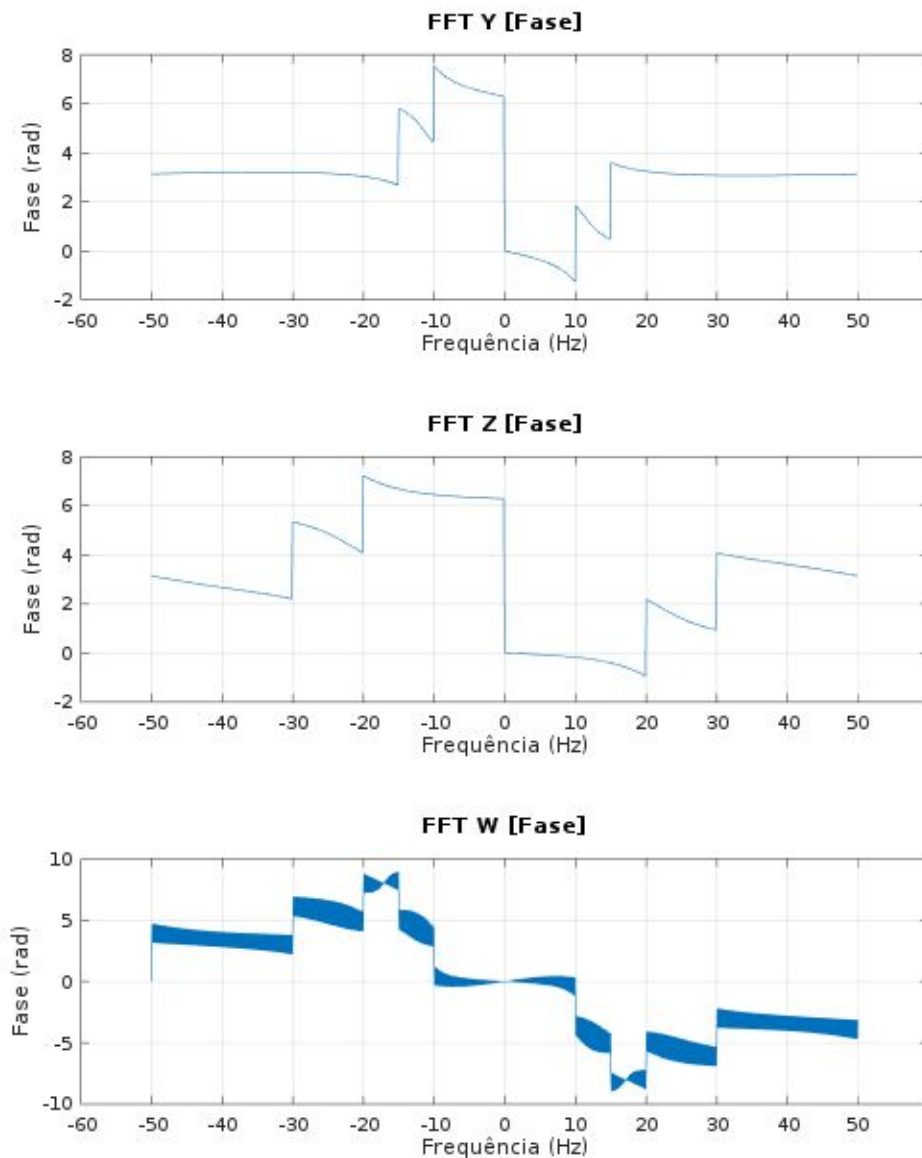
**OBS:** Todo projeto foi desenvolvido com o software **GNU Octave 4.2.2** numa máquina com Ubuntu 16.04. Algumas funções não têm a mesma sintaxe e modo de uso que o MATLAB, tais como: *spectrogram* (MATLAB) e *specgram* (Octave), então é provável que alguns códigos não funcionem em sua totalidade quando executados no MATLAB. Não tive como converter o código para fazer a compatibilidade a tempo, então peço que os scripts sejam executados no Octave durante a correção.

**1.1** Os plots pedidos na questão se encontram a seguir:

**Figura 1 - Magnitude das FFTs dos sinais**



**Figura 2 - Fases das FFTs dos sinais**



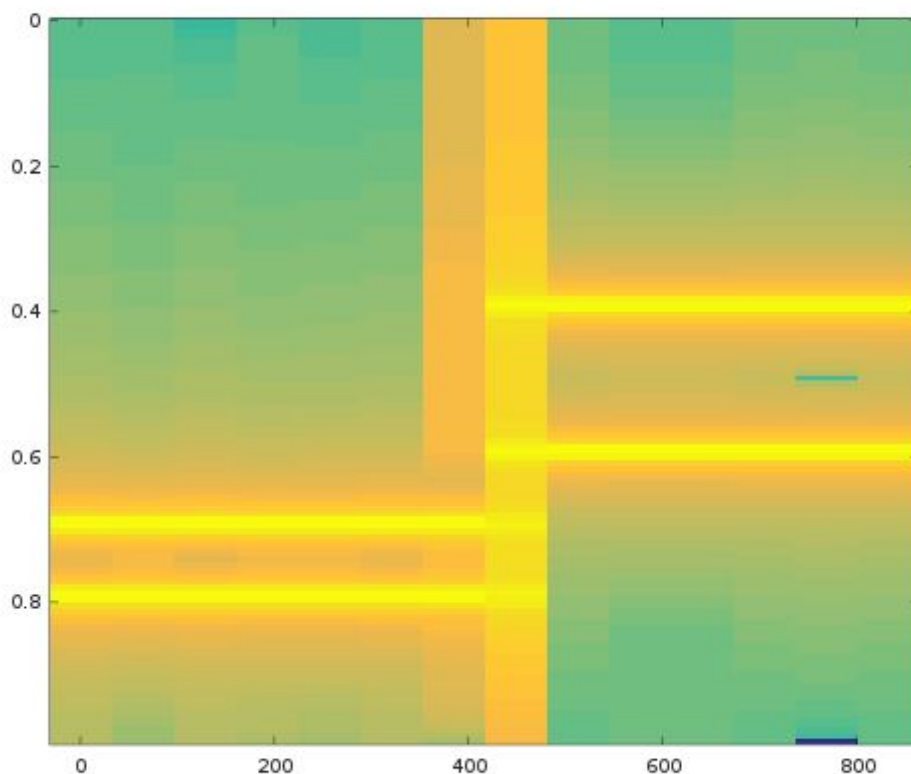
Como se percebe nos plots das magnitudes, as frequências dos sinais se mesclaram. Isso ocorre porque a magnitude das frequências é um sistema invariante no tempo. No entanto, a transformada de Fourier em si não é invariante no tempo, ou seja, deslocamentos no tempo ocasionam mudanças na fase da transformada. Isso justifica a imagem da fase do sinal **w** com “oscilações parasitas”. Observando-se a propriedade do deslocamento no tempo da Transformada de Fourier, percebe-se que aparece uma exponencial complexa ao resultado final, e esse termo exponencial causa essas oscilações maiores na fase do sinal concatenado.

**Figura 3 - Propriedade da Transformada de Fourier de um sinal deslocado no tempo**

$$\begin{aligned}\mathcal{F}\{g(t-a)\} &= \int_{-\infty}^{\infty} g(t-a)e^{-i2\pi ft} dt \\ &= \int_{-\infty}^{\infty} g(u)e^{-i2\pi f(u+a)} du \\ &= e^{-i2\pi fa} \int_{-\infty}^{\infty} g(u)e^{-i2\pi fu} du \\ &= e^{-i2\pi fa} G(f)\end{aligned}$$

A Figura 3 exibe espectrograma do sinal  $w$ . Nela é perceptível os pares de componentes das senoides dos sinais  $y$  e  $z$ . Esses dois pares de faixas horizontais amarelas informam que existem dois sinais distintos bem definidos dentro do sinal  $w$  e, pela mesma tonalidade desses dois pares de faixas horizontais, ambas componentes possuem a mesma amplitude. Além disso, as duas faixas verticais no meio do sinal  $w$  deixa bem claro que este é um sinal composto pela concatenação brusca de dois outros sinais, por isso essa “confusão” na componente das frequências.

**Figura 3 - Espectrograma do sinal  $w$**



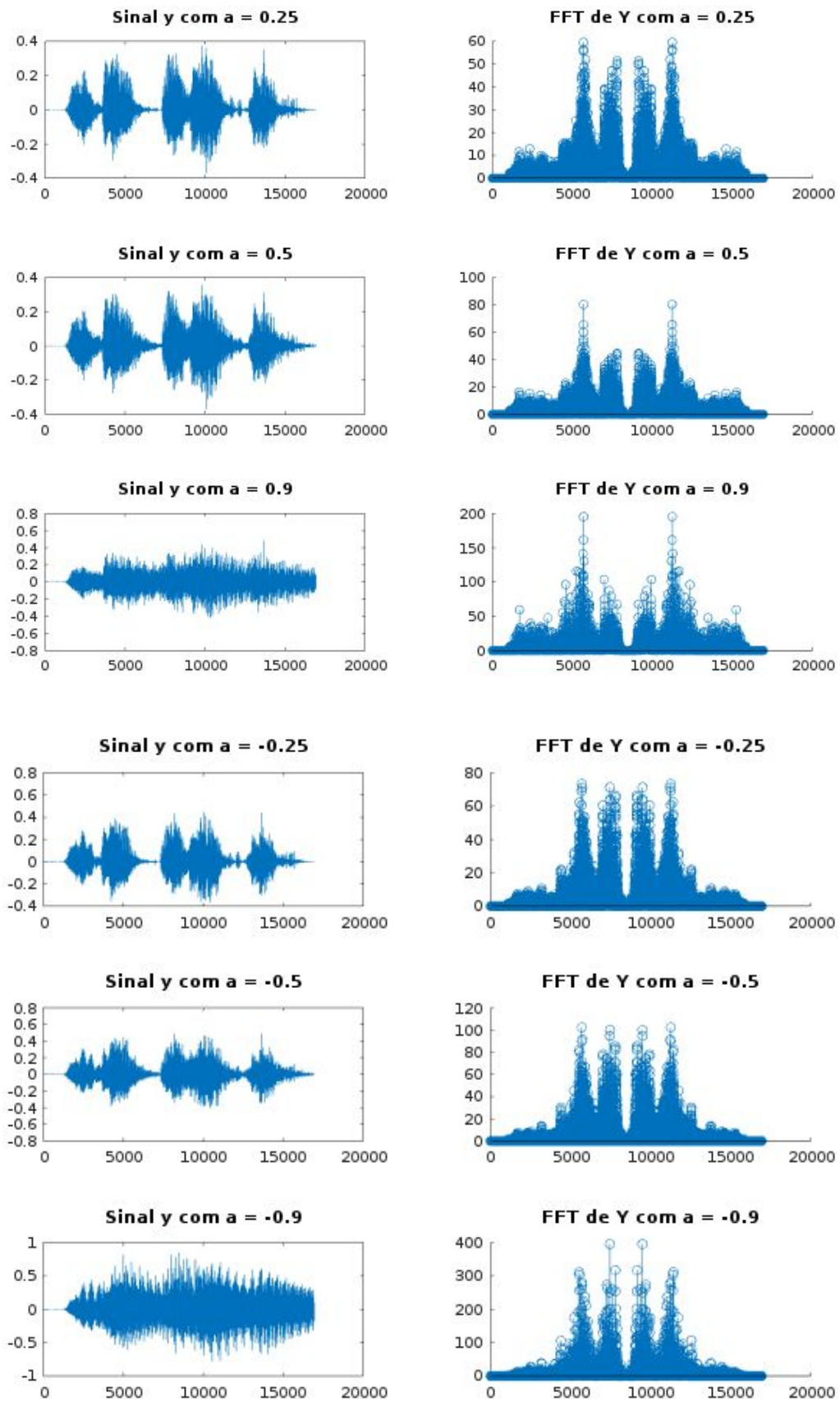
**1.2** Nesta análise, ao invés de considerar duas funções de filtro reverberador, irei considerar apenas a segunda função, com a soma dos termos do denominador, e será alterado o sinal do coeficiente **a** para obter a primeira função do enunciado.

$$H(z) = \frac{1}{1 + a \cdot z^{-D}}$$

Com os coeficientes negativos, ou seja, **a** variando entre -0,25, -0,5 e -0,9, nesta respectiva ordem, percebe-se que o sinal fica cada vez mais ecoado. Com os coeficientes positivos, **a** variando entre 0,25, 0,5 e 0,9, também nesta ordem, o sinal também fica mais ecoado. A diferença entre as duas é que os coeficientes negativos ecoam tons mais agudos e os coeficientes positivos ecoam tons mais graves, deixando o aspecto de um eco de um ambiente mais aberto e de um ambiente mais fechado, respectivamente. É interessante comentar também que quanto mais o sinal ecoa, mais o som fica com um aspecto robótico.

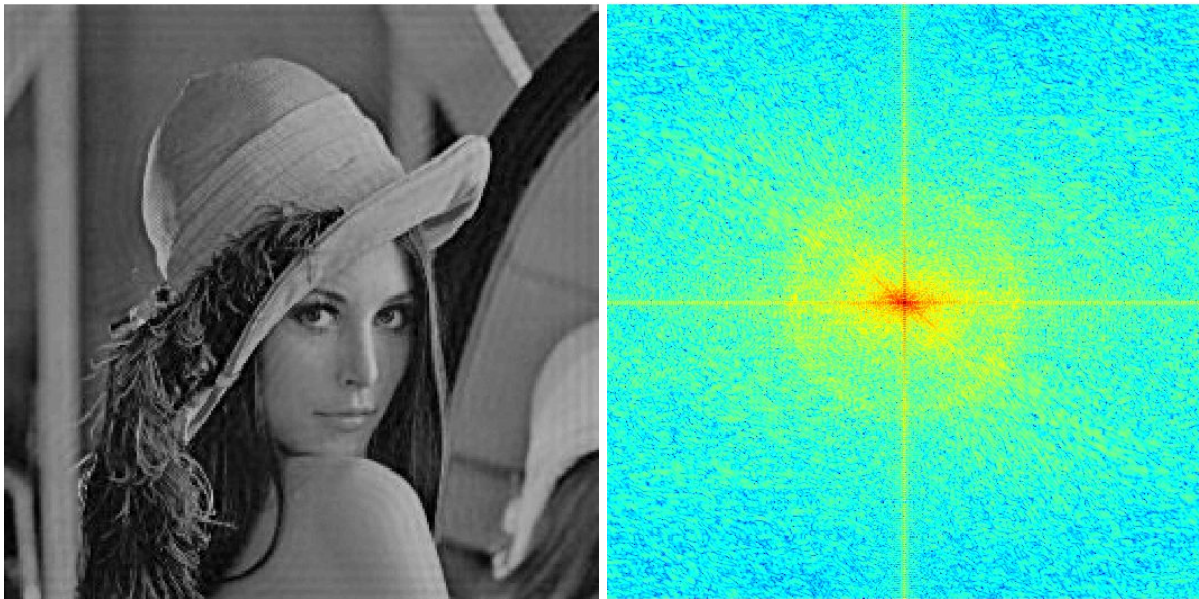
Abaixo estão os pares de sinal e sua transformada, a fim de ilustrar o que ocorre com o sinal quando se aplica o filtro reverberador.

**Figura 4 - Pares sinal-FFT com variações nos coeficientes**



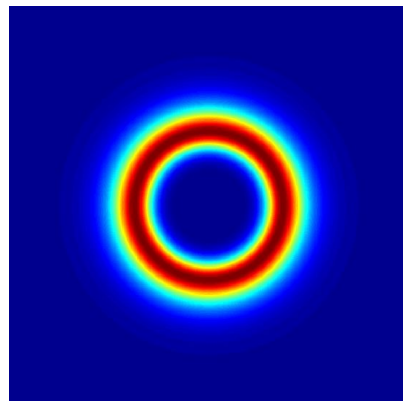
**2.1** A primeira operação que foi feita na imagem foi tirar sua FFT, de modo a identificar algum padrão que pudesse estar associado ao *ringings* (ver Figura 5). É possível perceber, pela sua transformada, que há um padrão circular no centro da imagem que é um forte indicativo da fonte do problema. Com ferramenta de zoom e utilizando a escala dos eixos da transformada de fourier, foi identificado que esse aro está entre as frequências de 90 e 100, radialmente.

**Figura 5 - À esquerda, imagem de lena com efeito *ringings*; à direita, transformada de Fourier da imagem**



Foi escolhido um filtro de Butterworth passa-faixa de 4ª ordem com as frequências de corte em 90 e 100 apenas para selecionar essa coroa circular.

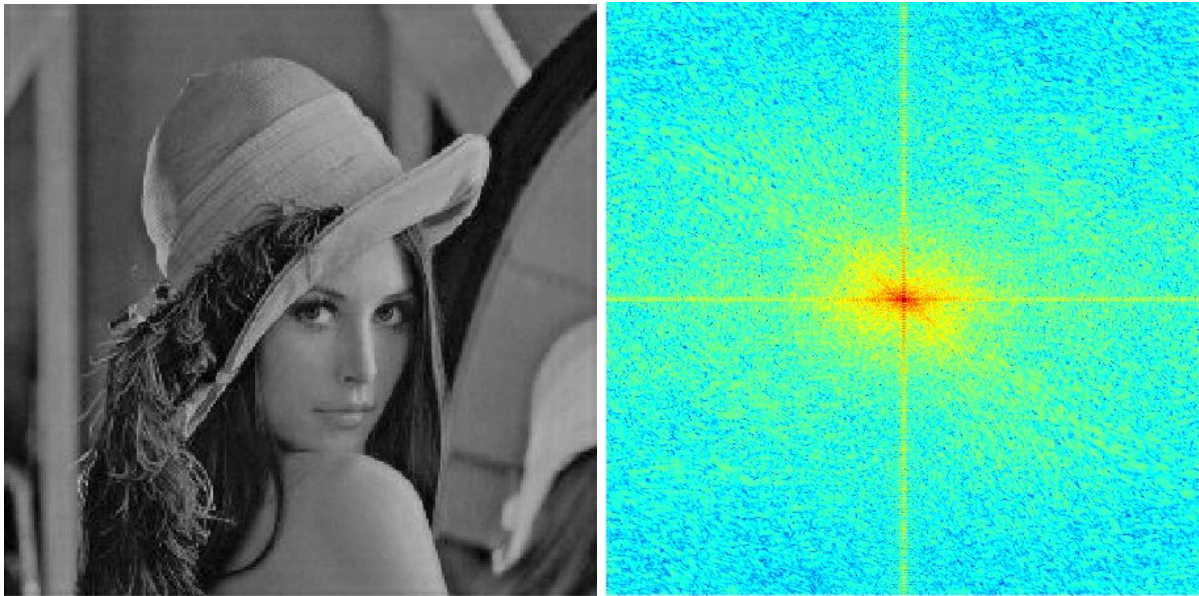
**Figura 6 - Filtro passa-faixa de Butterworth de 4ª ordem com frequências de corte em 90 e 100**





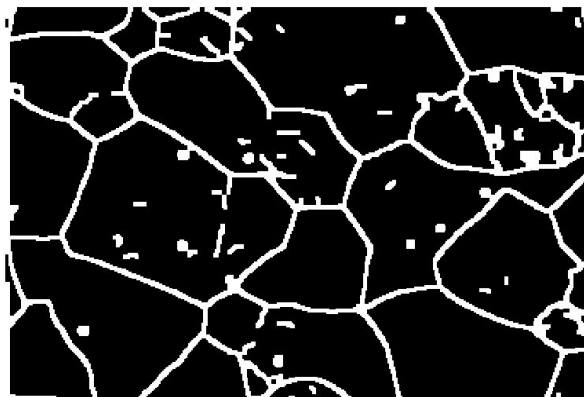
A partir da seleção dessas frequências radiais relativa aos *ringings*, foi possível subtrair da transformada de Fourier da imagem inicial, deixando apenas a imagem sem os *ringings*. Para se retirar totalmente os rings, foi necessário multiplicar por um fator de 3 para aumentar a intensidade do filtro, pois com um fator normal de 1 a filtragem era quase imperceptível e os *ringings* ainda predominavam.

**Figura 7 - Imagem filtrada e sua respectiva FFT com o anel de frequências subtraído**



**2.2** Primeiro, para se detectar as texturas, foi necessário encontrar as bordas da imagem. Dos filtros testados, o de Canny foi o que teve melhor resultado. A partir disso foi feita uma dilatação e erosão com a forma quadrada e o resultado está exibido na Figura 8.

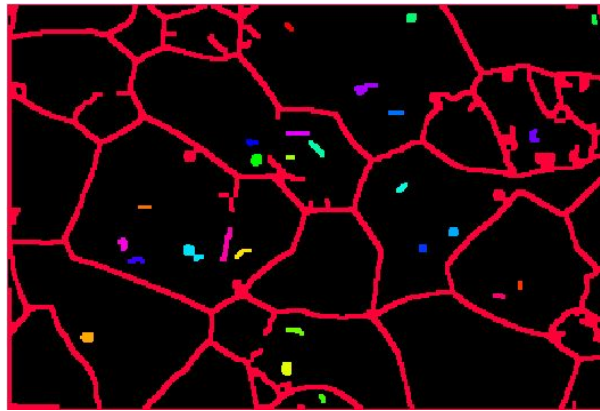
**Figura 8 - Resultado do Canny Edge Detection**



Com esse resultado, utilizou-se a função *bwlabel* do MATLAB para rotular todos os componentes conectados na cor branca. Com isso foi possível contar todos essas mini

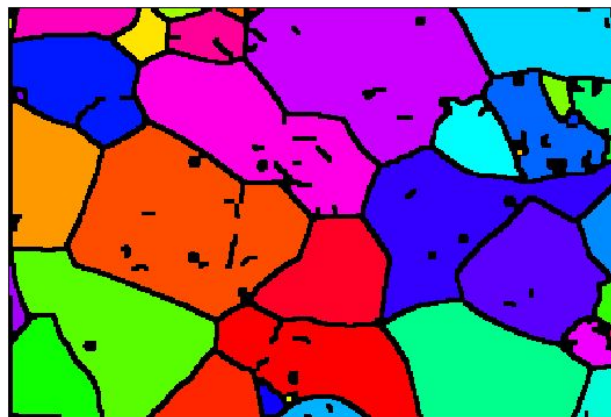
texturas espalhadas. Foi necessário subtrair 1 da quantidade total devido à fronteira principal (em vermelho na Figura 9) também entram nessa contagem data pelo *bwlabel*. Essa contagem deu **26 texturas**.

**Figura 9 - Todas regiões em branco rotuladas**



Invertendo a Figura 8, ou seja, fazendo o que é branco virar preto e o que é preto virar branco, pode-se executar novamente o *bwlabel* para se detectar as texturas maiores. O resultado dessa operação está na Figura 10.

**Figura 10 - Texturas maiores também rotuladas**

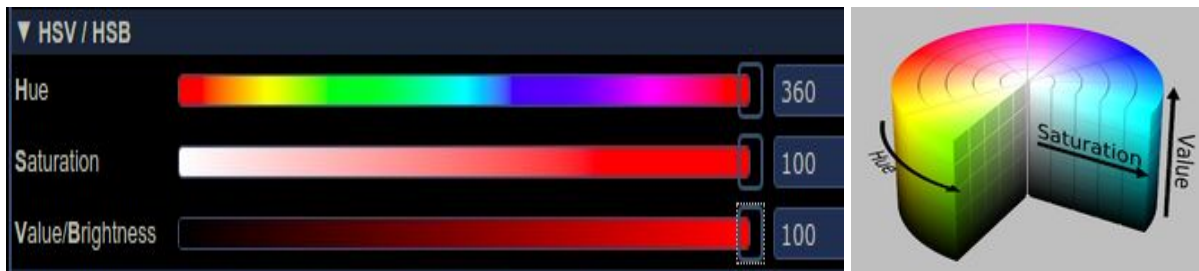


Essa contagem resultou em **40 texturas**. Então, o algoritmo desenvolvido, detectou que a imagem possui  $40 + 26 = \mathbf{66 \text{ texturas}}$ .

**2.3** Para se trabalhar nesse problema, foi necessário mudar o espaço de cores de RGB para HSV. Este sistema de cores possui 3 componentes: *Hue* (matiz), *Saturation* (saturação) e *Value* (valor, também encarado como brilho).



**Figura 11 - Espaço de cor HSV**



Para destacar melhor as cores da imagem do teste de daltonismo, pode-se mexer na saturação, deixando as cores mais vívidas:

$$newS = factor * originalS$$

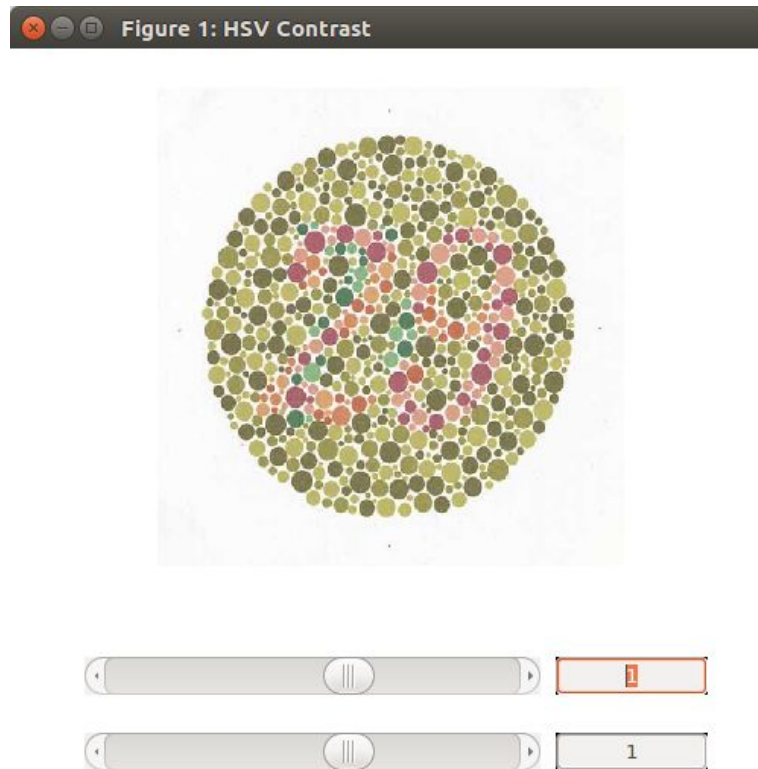
Pode-se também mexer no contraste entre as cores da imagem alterando-se os valores do brilho:

$$meanV = mean2(originalV)$$

$$newV = meanV + factor * (originalV - meanV)$$

Para deixar a atividade mais interativa e prática, foi desenvolvida uma pequena ferramenta com componentes UI de modo a encontrar os melhores fatores para ambas fórmulas (ver Figura 12). Com essa interação, foram encontrados os valores de 4 e -4 para os fatores de saturação e de valor/brilho. Observe a Figura 13 para ver o resultado dessas operações.

**Figura 12 - Ferramenta UI para encontrar fatores das fórmulas. A primeira barra deslizante refere-se ao fator de *Saturation*, e a segunda, ao de *Value***



**Figura 13 - À esquerda, o fator de saturação alterado para 4 já distingue bem as cores; à direita, o fator de Valor/Brilho também alterado para -4 faz com que as cores fiquem bem contrastadas, incluindo o fundo**



**3.1** A questão pede que os frames perdidos sejam recuperados. Então, uma ideia para resolver essa questão seria tirar uma média entre o frame anterior e o frame seguinte e aplicar/somar essa média no frame anterior, formando assim um frame intermediário.

O resultado individual ficou com um efeito de sombra/fantasma, como é possível ver na Figura 14. Foi feito um *Gaussian Blur* na imagem da média para tentar reduzir esse efeito fantasma, mas os resultados não foram tão bons e, como ponto negativo, esse *blur* acabava deteriorando todos os detalhes nas periferias da imagem.

Contudo, devido a esses frames serem parte de um vídeo, a percepção humana no meio da animação, não perceberá tal efeito, visto que é algo pontual na sequência de frames. Essa hipótese foi testada e aprovada compilando-se todos os frames em um arquivo de vídeo a 25 FPS. Ao visualizar o vídeo, realmente não se percebe os frames fantasmagóricos.

A fim de formalizar a comparação entre as duas imagens, calculou-se o erro quadrático médio (MSE) entre o frame recuperado e o frame original. Os resultados podem ser conferidos abaixo.

**Figura 14 - Efeito sombra/fantasma quando se tira a média entre os frames**



Abaixo está uma métrica do Erro Médio Quadrático para verificar o erro entre a imagem esperada e a imagem gerada pelo algoritmo desenvolvido.

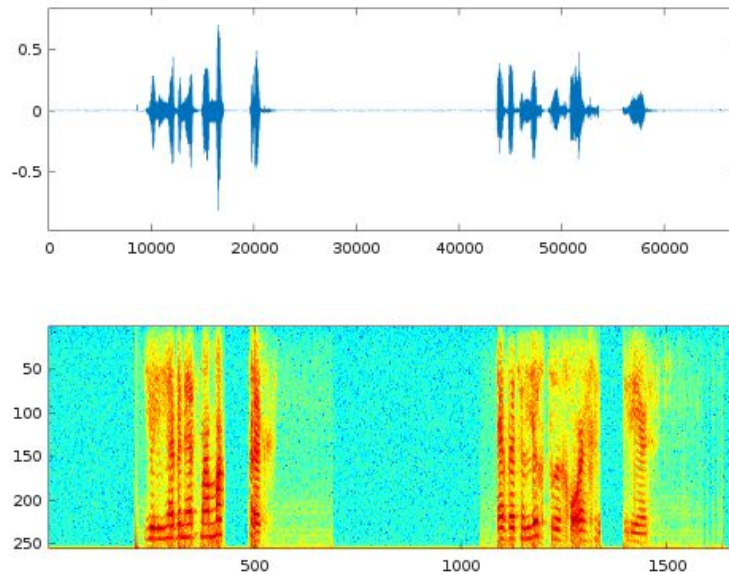
**Frame 102:** MSE Correct and Recovered = 0.1961

**Frame 110:** MSE Correct and Recovered = 41.8639

**Frame 118:** MSE Correct and Recovered = 26.2208

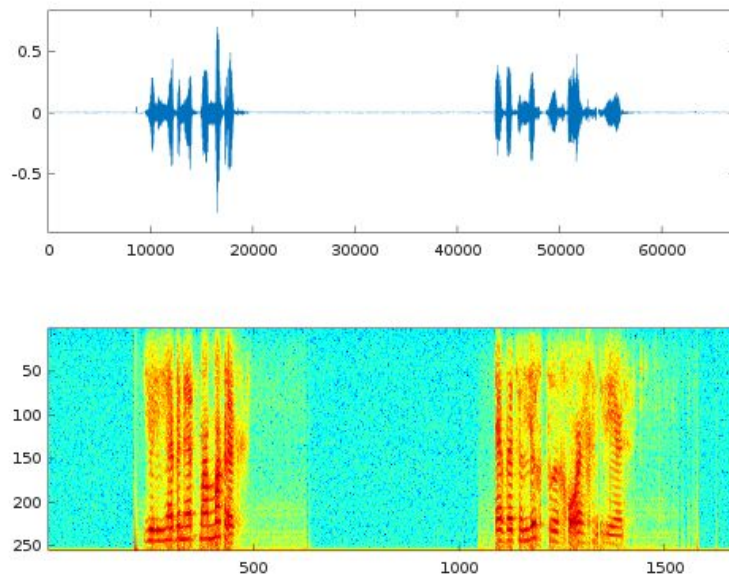
**4.1** O áudio modificado (*dg105.wav*) possui duas pausas notáveis no sinal, como pode-se notar na Figura 15:

**Figura 15 - Sinal de áudio dg105.wav no domínio do tempo e seu espectrograma**



Aproximando-se com o zoom, identifica-se que o primeiro espaço vazio fica entre a amostra 17100 a 19570, e o segundo espaço vai da amostra de número 53800 a 55930. Com essas demarcações, esses espaços vazios foram posicionados logo antes do próximo silêncio, deixando as informações de voz em seu lugar. Com essa operação, o áudio gerado se assemelha bastante do sinal de áudio original (ver Figura 16).

**Figura 16 - Sinal de áudio corrigido no domínio do tempo e seu espectrograma**



Ainda percebe-se uma compressão no domínio do tempo nas porções de áudio que ficavam antes dos *gaps*. Para consertar essa degradação, é necessário esticar essas porções para que o áudio fique mais parecido ainda com o original. Uma reamostragem pode resolver a questão do esticamento do sinal no domínio do tempo, porém, quando se trata de áudio, uma reamostragem comum ocasiona na mudança do tom de voz. A reamostragem foi feita, como mostra a Figura 17, mas o som fica distorcido. Para se esticar essa porção de áudio sem alterar o tom da voz, pode-se implementar um *phase vocoder*. Este *phase vocoder* não foi implementado na resolução dessa questão por agregar uma complexidade desnecessária ao problema visto que o resultado já está bastante aceitável apenas com o deslocamento dos *gaps*.

**Figura 17 - Reamostragem estica o sinal para deixar na proporção certa de tempo, no entanto, esse processo altera o tom da voz do locutor, desfigurando a fala.**



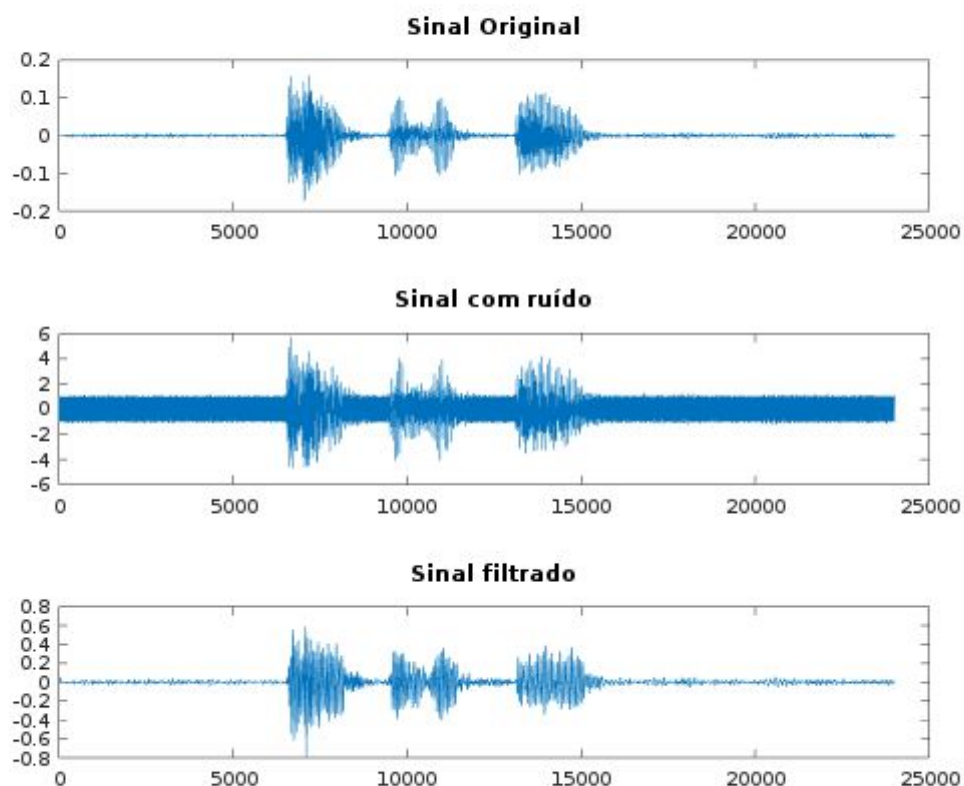
**4.2** A Figura 18 mostra o sinal original “teste\_de\_som.wav”, o sinal com o ruído solicitado e, por último, o sinal filtrado (essa figura foi gerada com um filtro gaussiano). A Figura 19 mostra a transformada de Fourier para o sinal com ruído. Os dois picos simétricos se referem ao ruído senoidal. Independente de se saber ou não da existência desse ruído senoidal, sabe-se que é um ruído pois se estende por todo sinal, como pode-se ver no



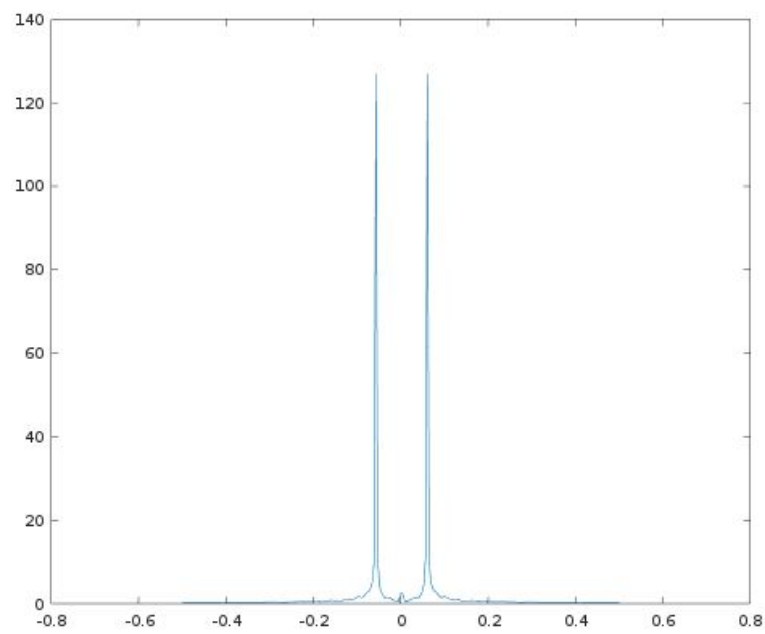
espectrograma (Figura 20). Então, para eliminar esse ruído evidente, implementou-se um filtro rejeita-faixa para anular uma faixa de frequência que o ruído estivesse contido. Analisando com a ferramenta de zoom, verificou-se que o ponto de início do pico é aproximadamente  $0,04\pi$  e o ponto de fim é um pouco indefinido. Alguns valores foram testados para esse ponto de fim e o melhor resultado filtrado se deu com  $0,35\pi$ .

Filtros FIR rejeita-faixa de ordem 60 foram, então, implementados utilizando as janelas de Hamming, Hanning, Gaussiana, Blackman e Kaiser. Todas as janelas tiveram resultados semelhantes, mas a que ficou com a frase mais nítida foi a implementação utilizando a janela de Hamming (ver Figura 21).

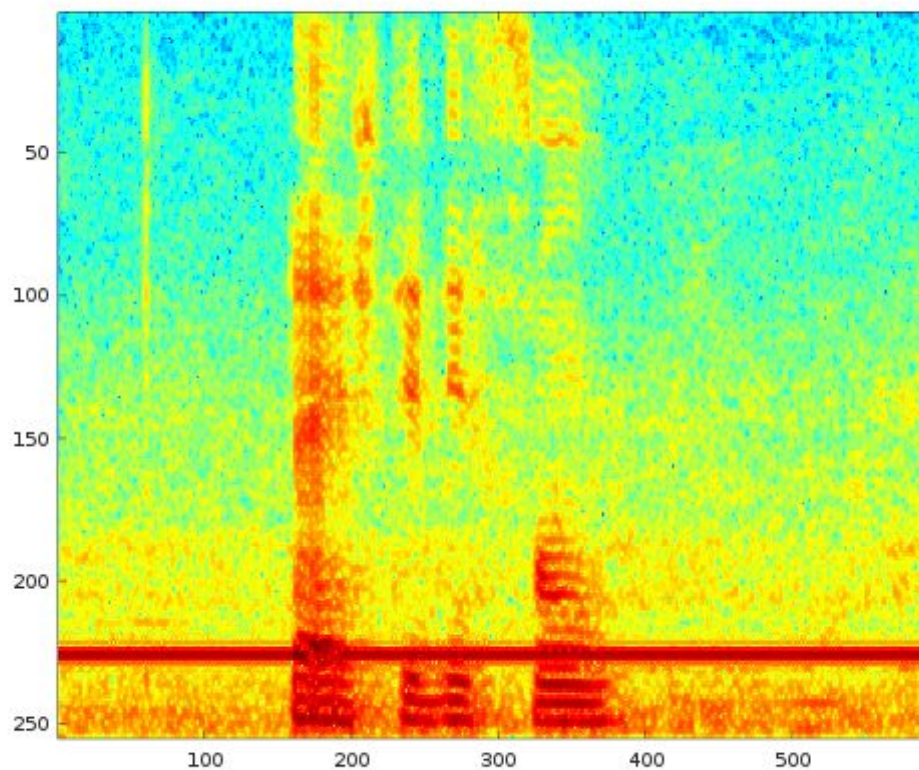
**Figura 18 - Sinal original, com ruído e filtrado, respectivamente**



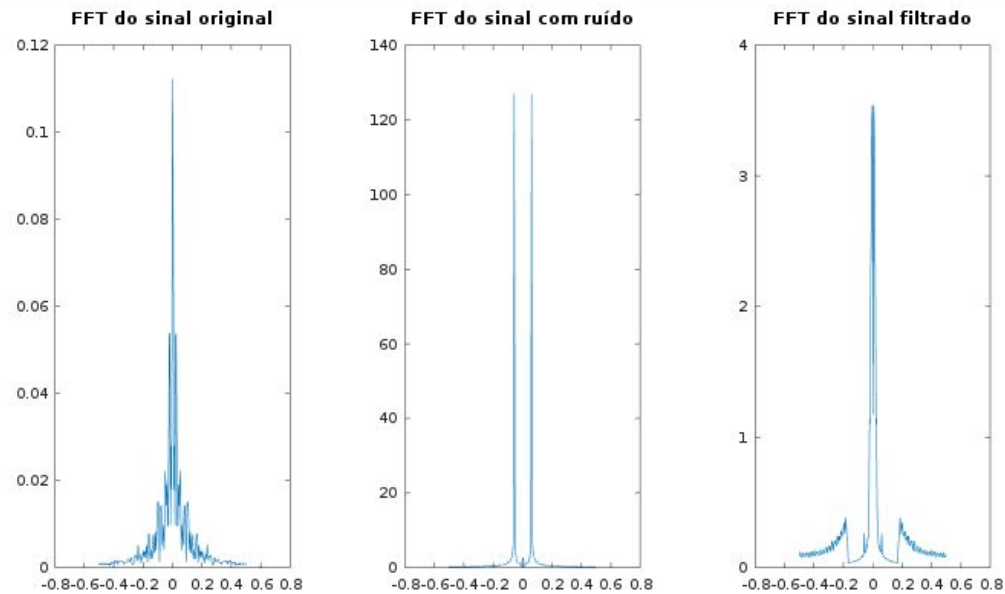
**Figura 19 - FFT do sinal com ruído**



**Figura 20 - Espectrograma do sinal com ruído mostrando o ruído persistindo ao longo do tempo entre as faixas de frequência 200 e 250**



**Figura 21 - FFT do sinal original, com ruído e filtrado por um filtro FIR rejeita-faixa, respectivamente**



Os filtros IIR de Butterworth e Chebyshev também foram testados e mostraram resultados satisfatórios. Ainda assim, o som produzido pelo filtro de Hamming ainda é mais próximo do original.

**Figura 22 - FFT do sinal original, com ruído e filtrado por um filtro IIR rejeita-faixa de Butterworth, respectivamente**

