

Processamento Digital de Sinais - Projeto

Victor Ximenes Carneiro de Oliveira

25 de setembro de 2023

1 Sinais

Com base nos sinais apresentados de YN, ZN e WN, assim como o seu intervalo:

$$yn = \sin(20 * \pi * n) + \cos(30 * \pi * n) \quad (1)$$

$$zn = \sin(40 * \pi * n) + \cos(60 * \pi * n) \quad (2)$$

$$wn = [yn \quad zn] \quad (3)$$

$$n = 0:0.01:10$$

a) Plotagem do YN

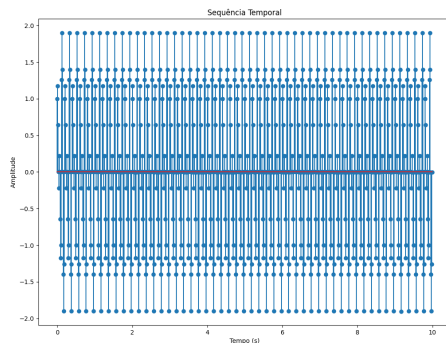


Figura 1: Sequência temporal de YN

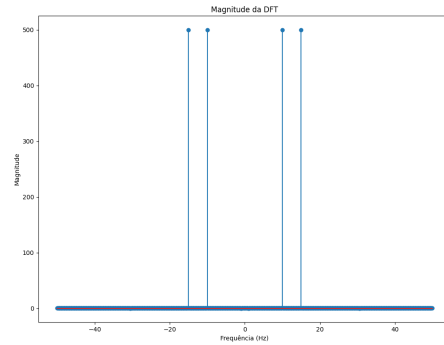


Figura 2: Magnitude de YN

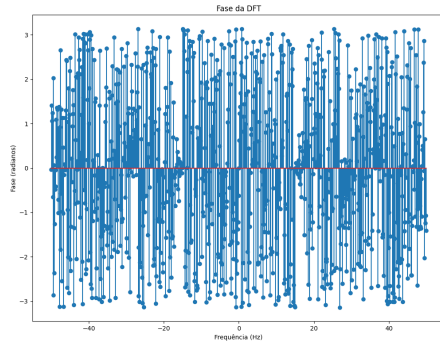


Figura 3: Fase de YN

b) Plotagem do ZN

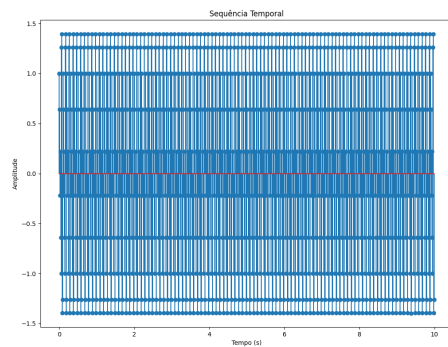


Figura 4: Sequencia temporal de ZN

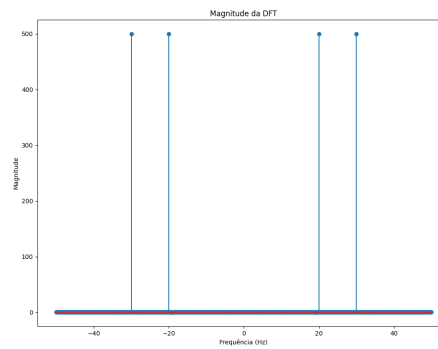


Figura 5: Magnitude de ZN

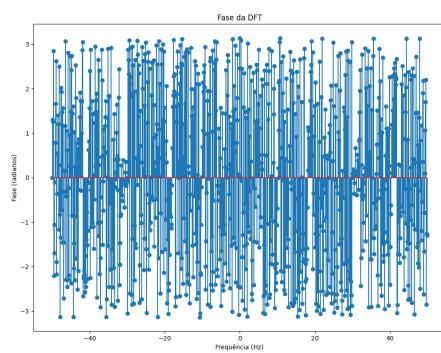


Figura 6: Fase de ZN

c) Plotagem do WN

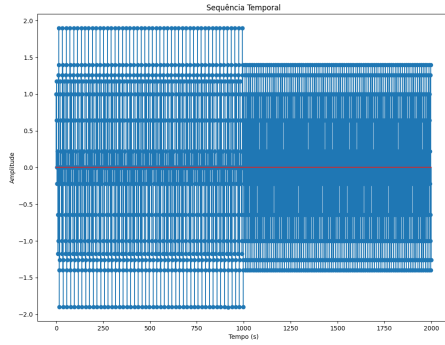


Figura 7: Sequencia temporal de WN

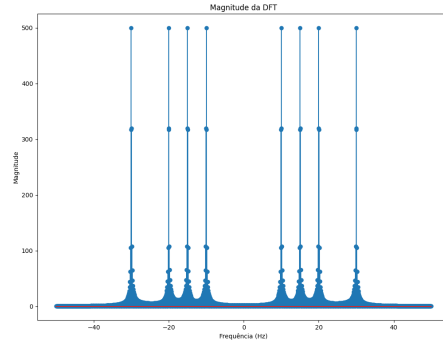


Figura 8: Magnitude de WN

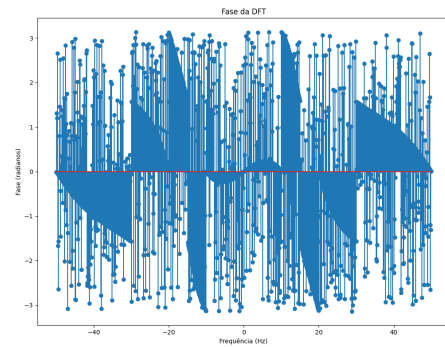


Figura 9: Fase de WN

d) A relação entre os sinais está em serem funções senoidais e cossenoidais com frequência angular diferentes, por isso atingem pico de magnitude em momentos diferentes. Entretanto para como WN é a concatenação de YN e ZN, isso interfere na DFT de WN pois como são sinais de frequências angulares diferentes, houve uma sobreposição na magnitude e na fase de WN, dessa forma WN tem os 8 picos (4 de YN e 4 de ZN) na magnitude, com isso a relação é de $wn = yn + zn$ mesmo sendo de concatenação, o que não afetou os valores maximos dos picos pois possuem fases diferentes.

e) Espectrograma de WN

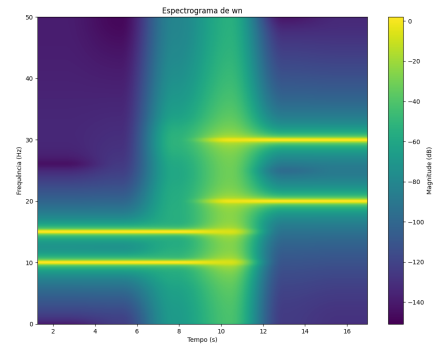


Figura 10: Espectrograma com janela maior

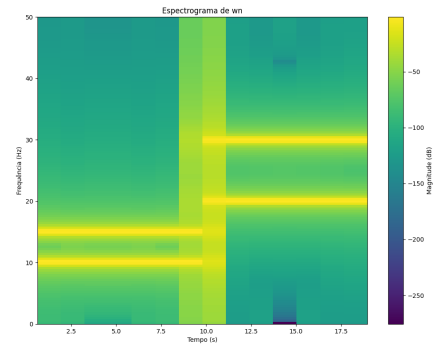


Figura 11: Espectrograma com janela menor

A conclusão obtida pelo espectrograma de WN é que existe uma frequência dominante nas duas metades do sinal, justamente as frequências dominantes de YN e ZN, em que para YN as frequências dominantes são 10Hz e 15Hz, e em ZN são 20Hz e 30Hz, pois em ambos os casos é devido à frequência angular dos senos e cossenos em YN e ZN. Por exemplo, temos: $y_n = \sin(20 * \pi * n) + \cos(30 * \pi * n)$, em que ao dividir a frequência angular por 2π , temos as frequências dominantes de 10Hz e 15Hz que se expressam no espectrograma, o mesmo se aplica para ZN, que demonstra frequências dominantes em 20Hz e 30Hz.

2 Imagens

2.1 Diferença de Gaussianas

- Visto que a mesma imagem é passada em dois filtros gaussianos com desvio padrão diferente, eu imagino que a subtração (diferença) disso vá gerar uma imagem destacando apenas as bordas, porque os dois filtros são passa-baixa, logo na diferença só sobrará os pixels de alto contraste
- Como esperado, após gerar as duas gaussianas com sigmas diferentes e subtraído, obteve-se uma imagem que destaca as bordas, inclusive os detalhes na face das moedas

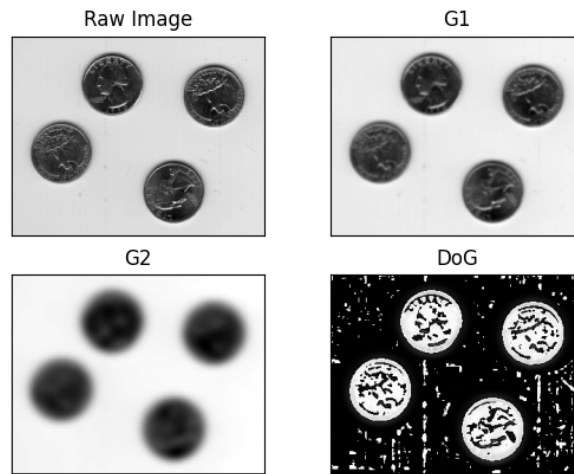


Figura 12: Diferença de gaussianas

2.2 Segmentação

A imagem original apresenta *low field of depth*, ou seja, existe um foco na imagem da frente (*foreground*) com relação a de trás (*background*), pois na de trás está com **Blur**, como demonstrado na figura.



Figura 13: Flor e Joaninha

A metodologia utilizada para segmentar a parte em destaque da imagem (a flor e a joaninha) consistiu em:

- Carregar a imagem e converter para *Grayscale*
- Aplicar filtro gaussiano para destacar as bordas da imagem, visto que o *background* já está borrado, com isso destaca a borda entre parte borrada e não borrada.
- Calcular o absoluto da diferença entre o filtro gaussiano e a imagem original para definir a mascara de contorno
- Desenhar o contorno em uma nova imagem de fundo preto para distinguir qual parte da imagem deve ser segmentada
- Segmentar a imagem original com a mascara gerada, obtendo o resultado abaixo



Figura 14: Imagem segmentada

3 Video

No video, 3 frames foram perdidos (102, 110 e 118) e para recuperar esses frames, foi utilizado um algoritmo de interpolação com os frames anteriores e posteriores aos faltantes. Para cada um

dos frames, eram carregados os frames antes e depois, e estes convertidos para nummppy array para que pudesse obter os valores numéricos com o objetivo de calcular a interpolação por meio da função demonstrada no código abaixo. Após esse cálculo, o array 2d era convertido e salvo em imagem.

Função de interpolação

```
def interpolation(prevFrame , nextFrame):
    return np.uint8(prevFrame*0.75 + nextFrame*0.25)
```



(a) Frame 102



(b) Frame 110



(c) Frame 118

Figura 15: Os 3 frames recuperados

Com o resultado das imagens recuperadas, foram utilizados as métricas *MSE*, *SSIM* e *PSNR* para comparar com os frames perdidos e avaliar a qualidade da recuperação.

Métricas			
Frame	MSE	SSIM	PSNR
102	0.175146	0.99927	55.6967
110	41.9766	0.97375	31.9007
118	26.2115	0.97874	33.9458

4 Voz e som

No audio analisado, existem dois momentos de fala e é possível observar tanto pela escuta quanto pelos graficos no tempo e na frequência que existe quebras abruptas no meio de ambos momentos de fala.

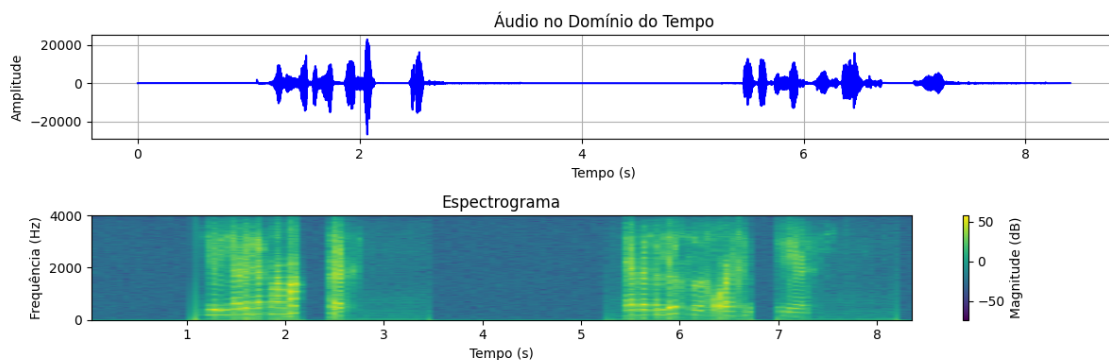


Figura 16: Audio degradado no tempo e na frequência

Com esses momentos de quebra no meio da fala, foi desenvolvido um código que remove esses curtos momentos de silencio entre falas, em que por meio de testes chegou-se aos melhores valores de tamanho minimo para silencio de quebra(intra-fala) e silencio de fala (inter-falas)

Código para identificação de quebras

```

for idx , amplitude in enumerate(audio_data):
    if np.abs(amplitude) <= limiar_quebra and not quebrado:
        idx_quebra = idx
        quebrado = True
    else:
        if np.abs(amplitude) > limiar_quebra and quebrado:
            duracao = idx - idx_quebra
            if duracao < duracao_min_silencio and duracao > duracao_min_quebra:
                :
            momentosQuebras.append((idx_quebra , idx))
        quebrado = False

```

Com a lista de tuplas que possuem os índices de início e fim da quebra, foi criado uma máscara para marcar como falso os trechos a serem removidos, para no final utilizar a máscara sobre o áudio degradado para remover as quebras.

Código para criação da máscara

```

mascara_audio = np.ones_like(audio_data, dtype=bool)
for inicio, fim in momentosQuebras:
    mascara_audio[inicio:fim] = False

```

Com isso, pode-se eliminar os trechos quebrados no meio da fala e tornar o áudio mais natural, o que se refletiu nos gráficos:

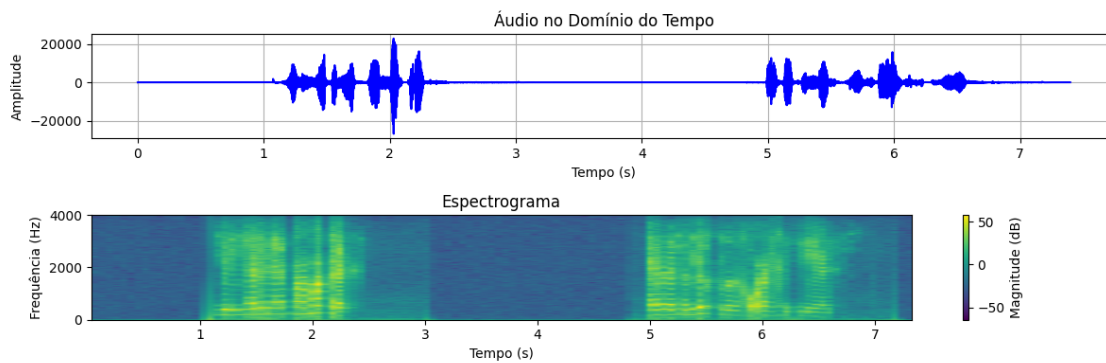


Figura 17: Áudio filtrado no tempo e na frequência

Ao comparar com o áudio original, sem degradação, observou-se que ele é menos acentuado nos valores máximos da amplitude, enquanto o degradado e filtrado passam do 20000 que ultrapassam o limite audível, então foi aplicado um filtro *Hanning* para redução suave dos ruídos, o que pode ser comparado nos gráficos abaixo entre o áudio filtrado com Hanning e o áudio original sem degradação.

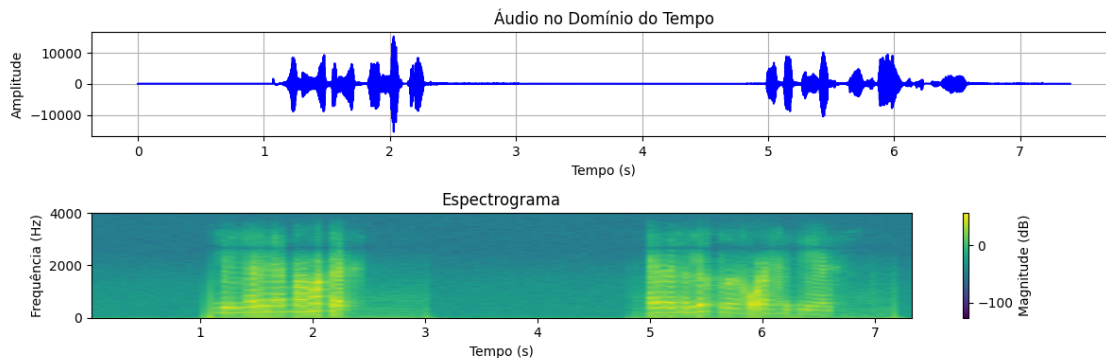


Figura 18: Áudio filtrado com Hanning no tempo e na frequência

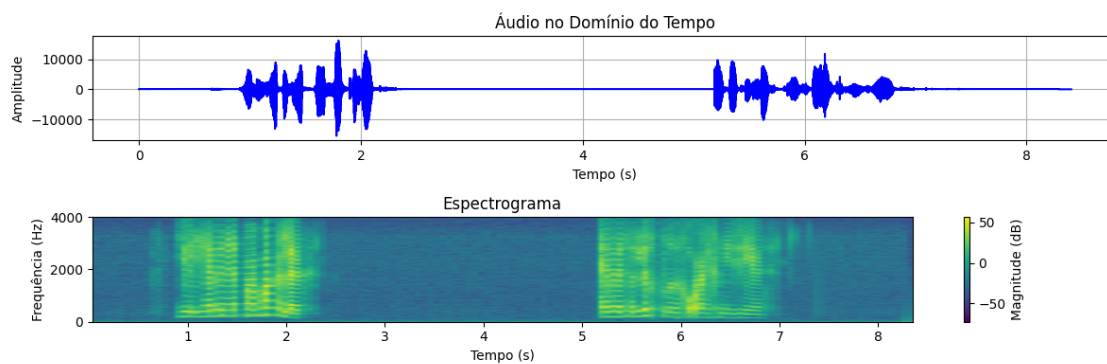


Figura 19: Audio original sem degradação no tempo e na frequência

5 Apêndice

Para melhor visualização, todos os códigos, graficos e imagens foram compartilhados junto com esse PDF em *folders* respectivos as suas questões.