Processamento Digital de Sinais - PDS

Python em PDS: Amostragem:

Para transportar um sinal analógico para dentro de dispositivos digitais, precisamos Amostrá-los, Quantizá-los e Codificá-los. O dispositivo que faz a amostragem, a quantização e a codificação de um sinal analógico (tempo contínuo) é chamado de conversor Analógico para Digital ou conversor A/D, enquanto o dispositivo que converte sinais Digitais em sinal Analógicos de tempo contínuo é chamado de conversor Digital para Analógico ou conversor D/A.

Destes três processos nós vamos estudar o primeiro a amostragem. Que é a captura de amostras do sinal contínuo em intervalos uniformemente espaçados. Transformando em um sinal de tempo discreto. E também inclui o retorno do sinal discreto no tempo para um sinal de tempo contínuo.

Teorema de Amostragem:

Representação de um sinal de tempo contínuo por suas amostras. Quando amostramos um sinal de tempo contínuo, temos que escolher o período que o sinal será amostrado.

Se escolhermos um período com um valor extremamente pequeno para o período de amostragem, obteremos um sinal discreto com uma diferença significativa pequena entre ele e o sinal contínuo, tanto no visual como no conteúdo de informação. Esse sinal discreto será formado por mais pontos e desta forma necessitara de mais recursos para ser tratado.

Se escolhermos um valor grande para o período de amostragem, obteremos uma compressão de dados, uma diminuição no número de amostras, mas corremos o risco de perder algumas informações importantes fornecidas pelo sinal contínuo.

Então a pergunta é:

Como fazer a amostragem sem perder as informações do sinal contínuo e não gerar muitas amostras? Veremos que a resposta é mais bem percebida no domínio da frequência do que no domínio do tempo.

Para fazer a conversão do sinal contínuo $x(t)$ em um sinal digital o primeiro ato é discretizá-lo na variável do tempo. Ou seja, amostrar $x(t)$ em períodos uniformes de $t = nT_s$ ou

$$x(nT_s) = x(t)|_{t=nT_s},\ n \text{ inteiro}$$

Onde T_s é o período de amostragem.

Uma forma bem conveniente de apresentar a amostragem e através da multiplicação do sinal contínuo por um trem de impulsos.

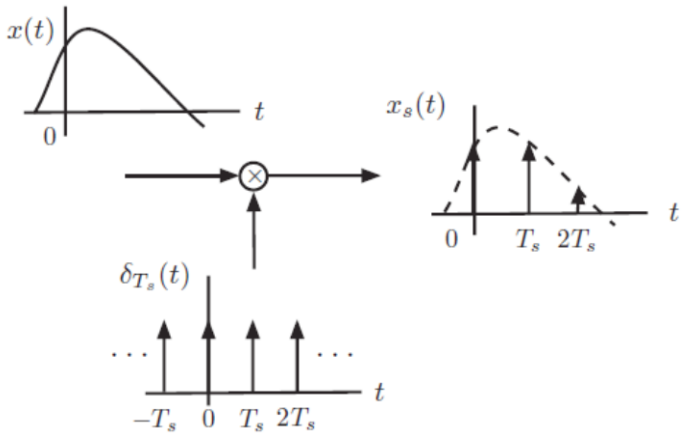

Figura 1 – Amostragem por Trem de impulsos.

Exemplo 1: Vamos pegar o sinal contínuo:

$$x(t)=100\cos(2\pi\cdot 100\,t)$$

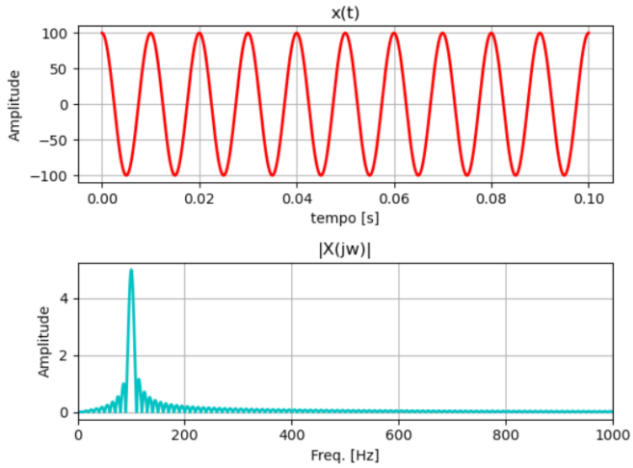

Figura 2 - Sinal x(t) no tempo e na frequência.

E temos o trem de impulsos $\delta_{T_s}(t) = \sum_n \delta(t - nT_s)$, com T_s = 0.001 s, apresentado na figura 3.

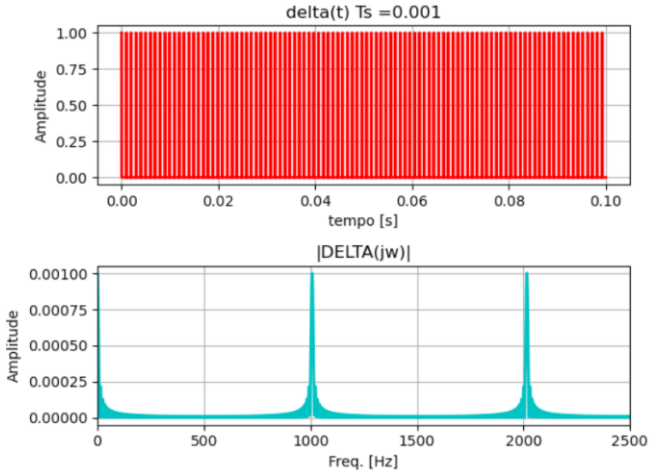

Figura 3 - Trem de impulsos, T_s = 0.001 s.

Vamos amostrar o sinal original $x(t)$ com a multiplicação do trem de impulsos $\delta_{Ts}(t)$. Assim obtemos: $x_s(t) = x(t)\delta_{T_s}(t)$. A figura 4 representa o sinal obtido no domínio do tempo e da frequência.

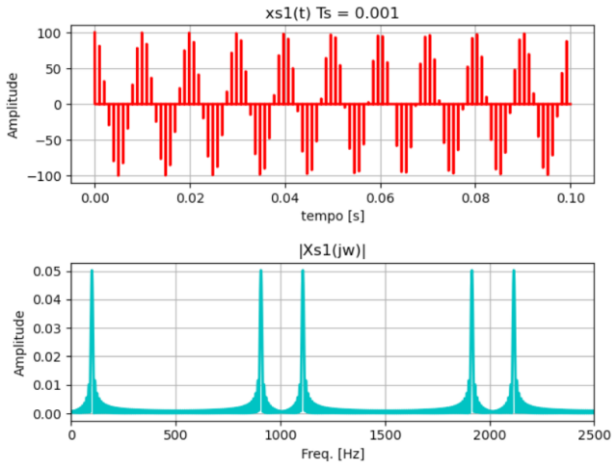

Figura 4 - $x_{s1}(t)$ - Sinal amostrado.

A figura 5 apresenta $x_{s1}[n]$ que representa $x(t)$ no tempo discreto.

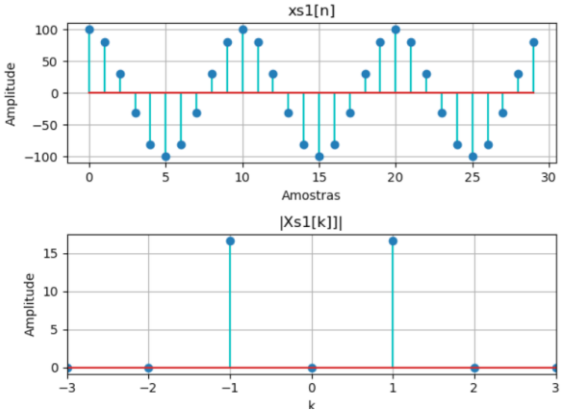

Figura 5 - $x_{s1}[n]$ – Versão discreta do sinal e sua representação serie de Fourier.

Vamos aumentar o período de amostragem mudando o período do trem de impulsos para T_s = 0.0025 s:

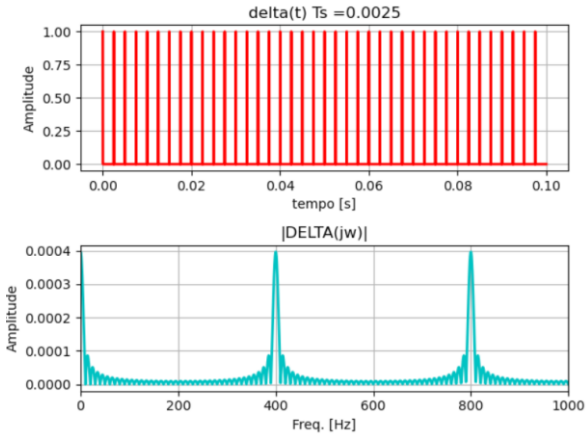

Figura 6 - Trem de impulsos, T_s = 0.0025 s.

Assim o sinal amostrado muda para o apresentado na figura 7.

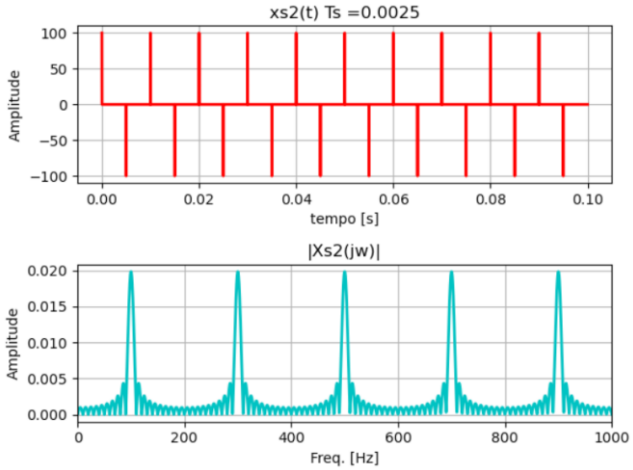

Figura 7 - $x_{s2}(t)$ - Sinal amostrado.

A figura 8 apresenta $x_{s2}[n]$ e sua Serie de Fourier.

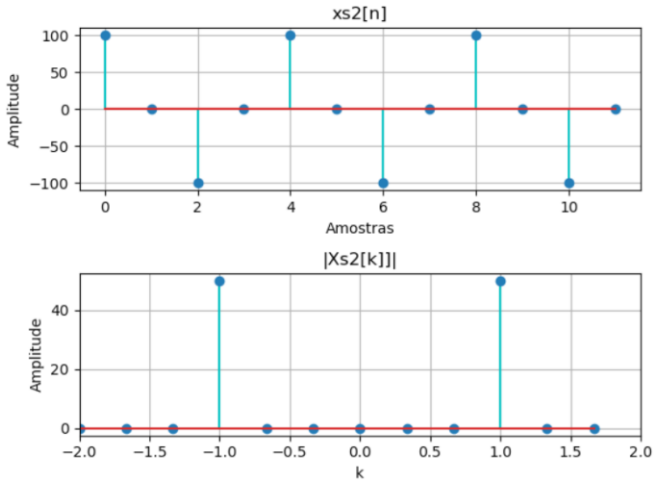

Figura 8 - $x_{s1}[n]$ – Versão discreta do sinal.

Vamos novamente aumentar o período de amostragem. Agora com T_s = 0.008 s.

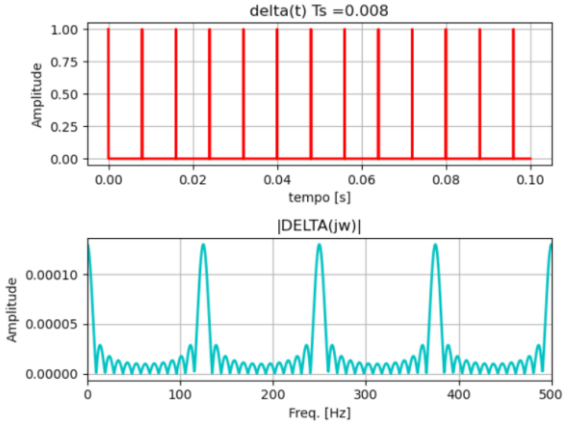

Figura 9 - Trem de impulsos, T_s = 0.008 s.

A figura 10 apresenta o sinal amostrado com T_s = 0.008 s.

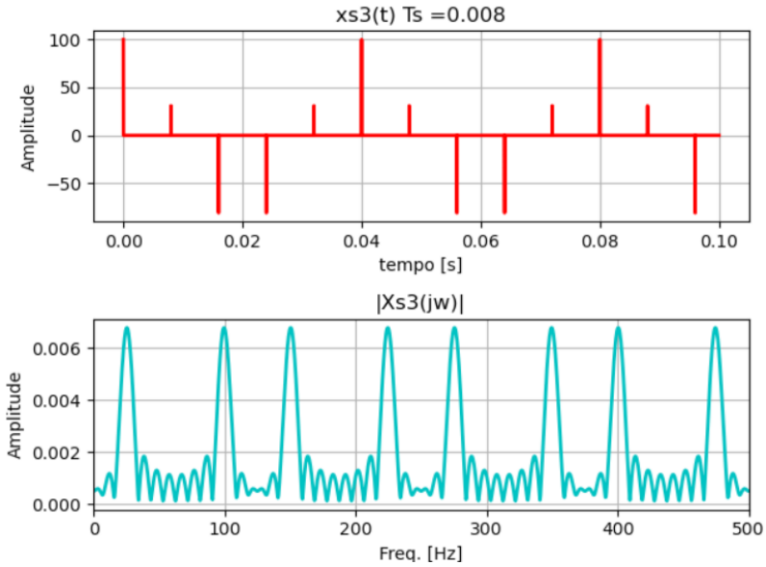

Figura 10 - $x_{s3}(t)$ - Sinal amostrado.

Na figura 11 vamos juntar os gráficos no domínio do tempo e da frequência dos três sinais amostrados comparando com o sinal original.

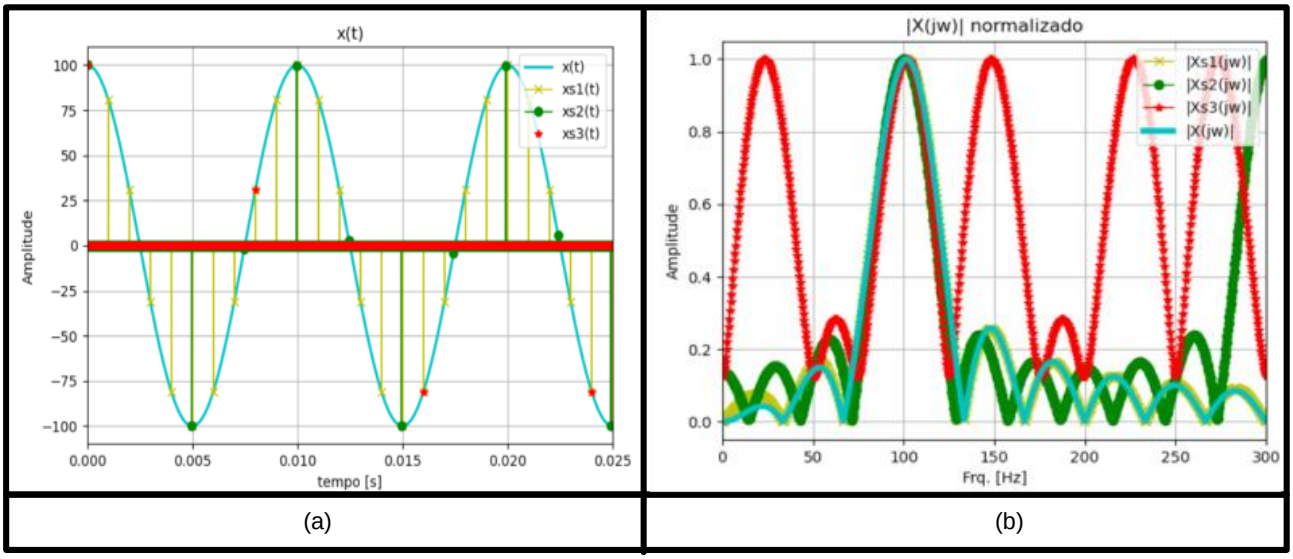

Figura 11 – *Comparando os sinais amostrados com o sinal original x(t) (a) domínio do tempo (b) no domínio da frequência.*

Note na figura 11.a que os sinais amostrados acompanham o sinal original. E difícil avaliar no domínio do tempo quais sinais amostrados podem ser recuperados. Porem no domínio da frequência, figura 11.b, é mais fácil perceber quais sinais amostrados são recuperáveis. Está claro que o sinal $x_{s3}(t)$ apresenta componentes de frequência diferente do sinal original. Isso é conhecido com *aliasing*. Isso ocorre porque o período de amostragem não foi pequeno o suficiente. Todos os outros sinais podem ser recuperados através de um filtro passa-baixas.

Exemplo 2: Na figura 12 temos o sinal contínuo

$$x(t)=u(t-0.005)-u(t-0.02)$$

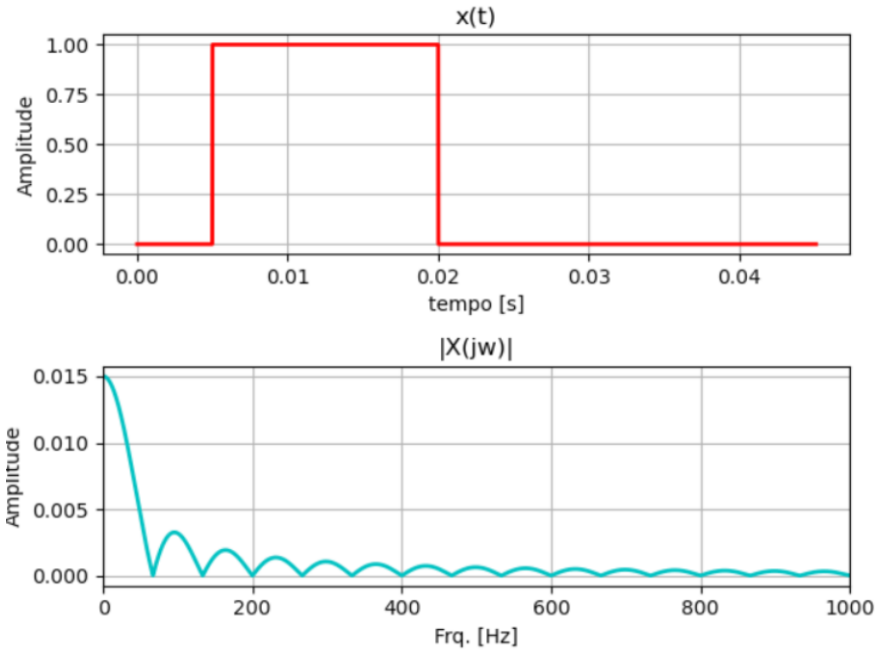

Figura 12 - Sinal $x(t) = u(t-0.005)-u(t-0.002)$

E vamos aplicar dois trens de impulsos $\delta_{T_s}(t) = \sum_n \delta(t - nT_s)$, o primeiro com um período de T_s = 0.0015 s e o segundo com T_s = 0.01 s. A figura 13 apresenta respectivamente os dois trens de impulsos, no domínio do tempo e da frequência.

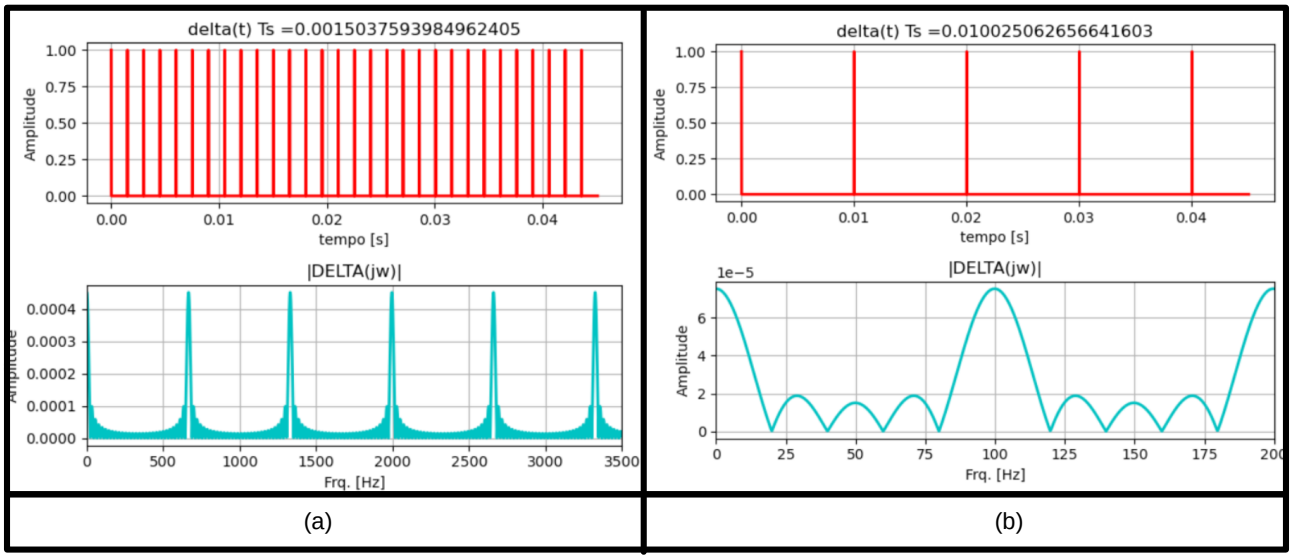

Figura 13 - Trens de impulsos (a) T_s = 0.0015 s e (b) T_s = 0.01 s.

A figura 14 apresenta a multiplicação do sinal contínuo com os dois trens de impulso: $x_s(t) = x(t)\delta_{T_s}(t)$:

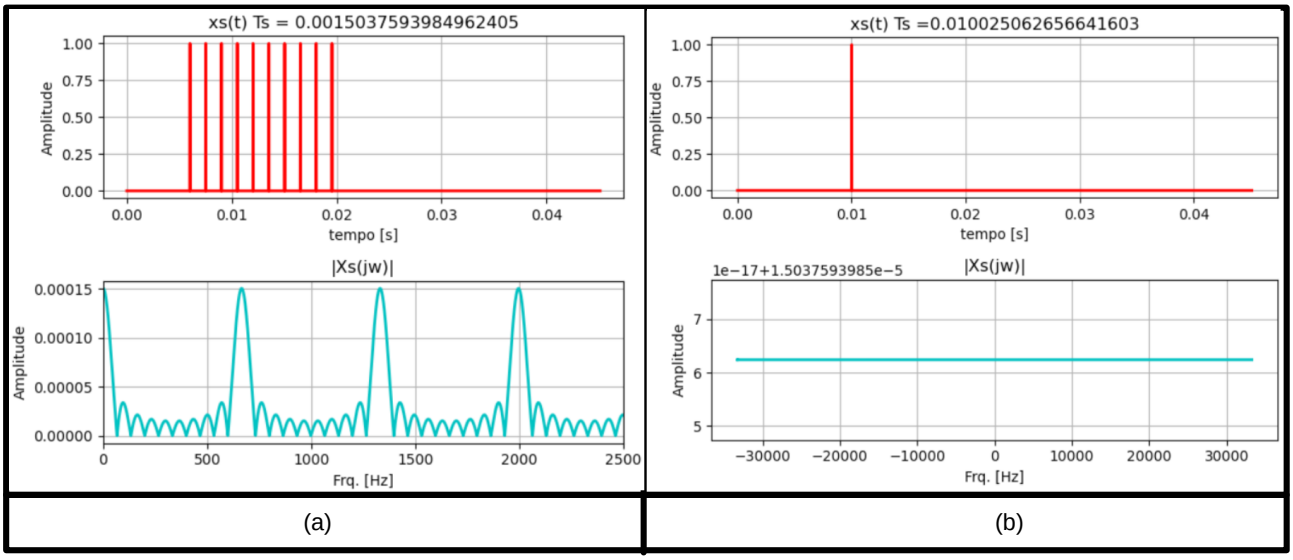

Figura 14 - $x_s(t)$ - Sinal amostrado com (a) T_s = 0.0015 s e (b) T_s = 0.01 s.

Note que na figura 15.a o sinal amostrado tem varias amostra dentro do pulso quadrado enquanto na figura 15.b o sinal amostrado só consegue pegar uma amostra do sinal dentro do pulso quadrado. Assim o sinal com T_s = 0.01 s parece que é um impulso e isso se reflete na sua representação no domínio da frequência.

A figura 15 reuniu a representação no domínio da frequência dos sinais amostrado. Também aparece um terceiro sinal $x_{s2}(t)$ que foi amostrado com um T_s = 0.003756 s.

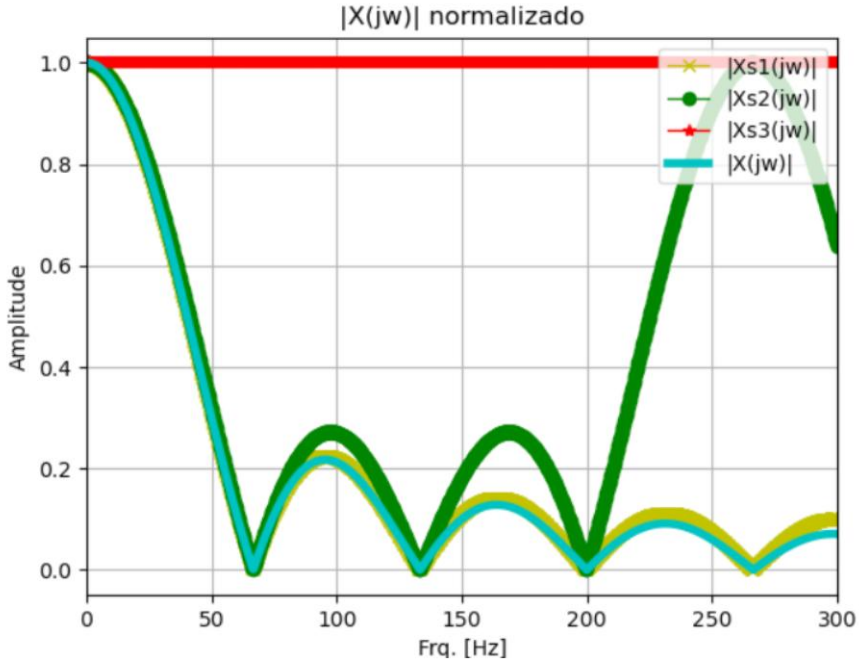

Figura 15 -

Os sinais $x_{s1}(t)$ e $x_{s2}(t)$ podem ser recuperados através de um filtro passa-baixas. Porem o sinal $x_{s3}(t)$ como apresenta *aliasing* não pode ser recuperado.

Reconstrução do Sinal:

A operação de reconstrução de um sinal amostrado consiste em obter, a partir de amostras de um sinal discreto, o sinal analógico correspondente.

Sendo que o período de amostragem $T_s = 1/f_s$, onde f_s é conhecida como frequência de amostragem. O teorema de *Nyquist* diz que se $f_s > 2\, f_{máx}$ ($f_{máx}$ é a freq. máxima do sinal de tempo continuo) o sinal pode retornar ao domínio de tempo contínuo. Como podemos ver dos exemplos anteriores o sinal pode retornar a sua forma original apenas passando por um filtro passa-baixas ideal.

Porem sabe que um filtro ideal não é causal e assim não realizável. Então a recuperação é realizada através de outros tipos de filtros (sistemas). Um bem conhecido por sua simplicidade é o retentor de ordem Zero. O qual retém o valor até a próxima amostra, apresentado na figura 16. E assim produz um sinal que é descontinuo.

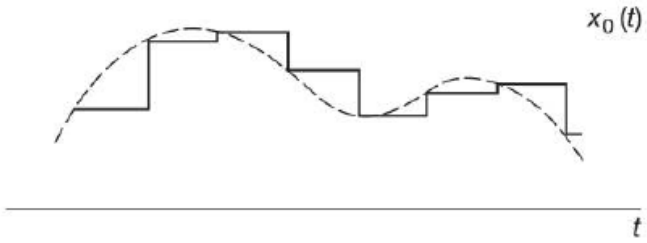

Figura 16 - Retentor de ordem zero

Exemplo 3: Vamos aplicar no exemplo 1 o Retentor de ordem zero.

A figura 17 apresenta o resultado da aplicação do Retentor de ordem zero no sinal amostrado com $f_s = 10f_{máx}$ ou $T_s = 0.001$ s.

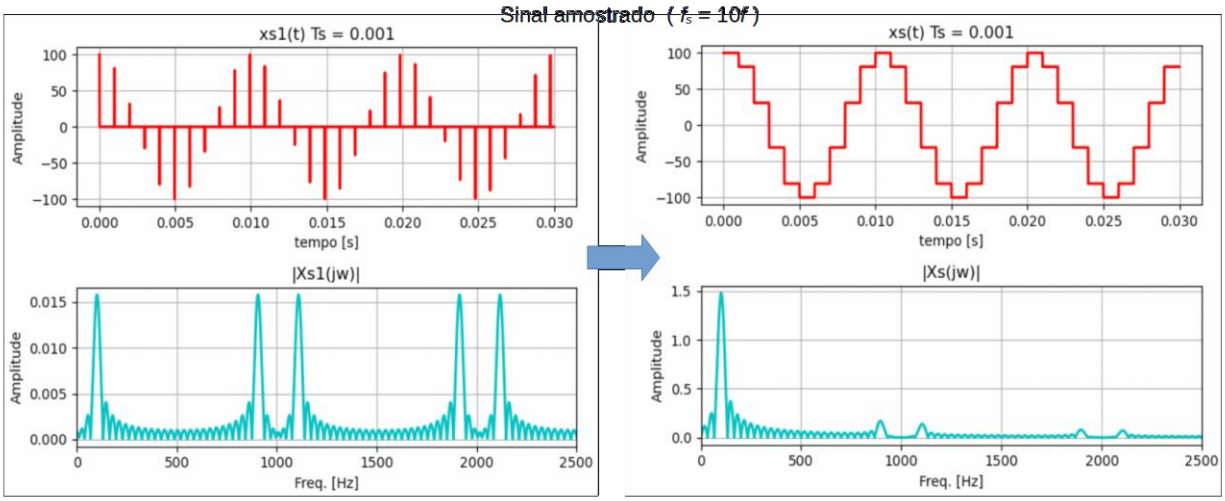

Figura 17 – Aplicação do Retentor de ordem zero no sinal amostrado $T_s = 0.001$ s.

Observando a representação no domínio da frequência note como o retentor de ordem zero atuou como um filtro passa-baixas. Note também a presença de componentes de frequência perto das harmônicas da frequência de amostragem.

Vamos também aplicar o Retentor de ordem zero no sinal amostrado com T_s = 0.0025 s, apresentado na figura 18. Observamos que o retentor de ordem zero continua funcionando como um filtro passa-baixas. Note a maior presença dos componentes de frequência perto das harmônicas da frequência de amostragem.

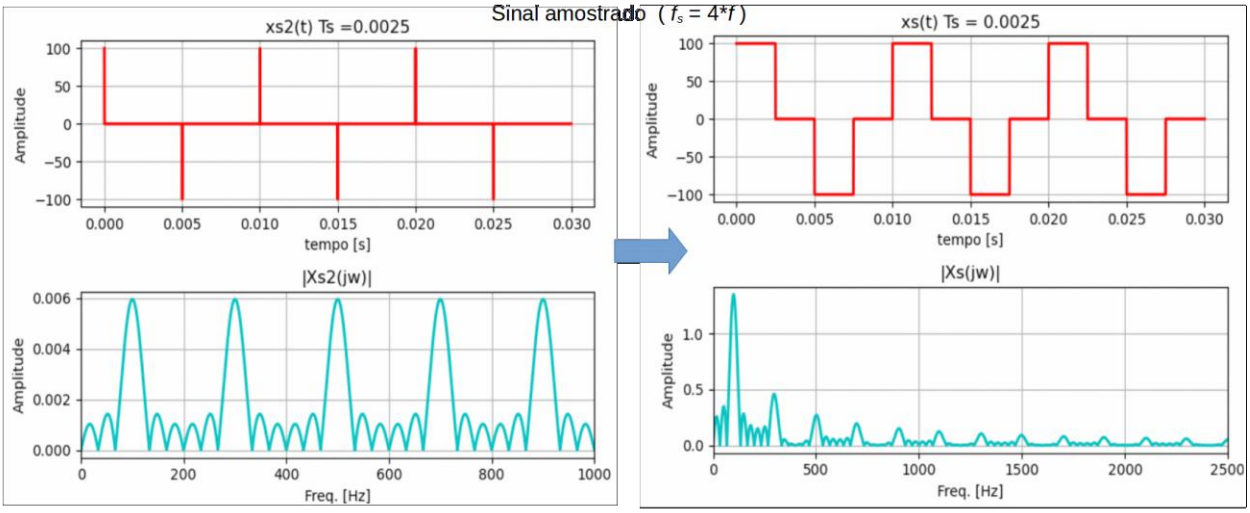

Figura 18 – Aplicação do Retentor de ordem zero no sinal amostrado T_s = 0.0025 s.

Dependendo da aplicação que o sinal reconstruído será utilizado, as formas apresentadas no exemplo anterior já são suficientemente razoáveis boas. Porem algumas aplicações necessitam de sinais reconstruído mais parecido com o sinal original. Isso implica em utilizar técnicas mais complexas para a reconstrução do sinal, com, por exemplo, o retentor de primeira ordem.

O sinal reconstruído também pode passar por um processo filtragem para eliminar ou mitigar a presença de elementos de frequência fora do sinal original. Só para demonstração vamos utilizar um filtro passa-baixas RC. O esquema do filtro RC e sua função de transferência podem ser visto na figura 19.

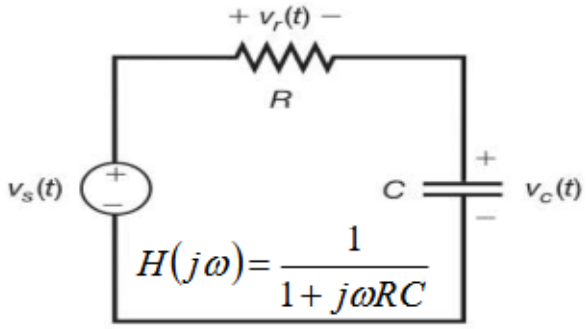

Figura 19 - Filtro passa-baixas RC em serie.

A figura 20 apresenta o gráfico com o sinal reconstruído T_s = 0.001 s e sem filtragem e com filtragem. Comparando os dois percebemos que o sinal filtrado é um pouco melhor. Pode-se conseguir um sinal mais parecido com o original com um filtro melhor.

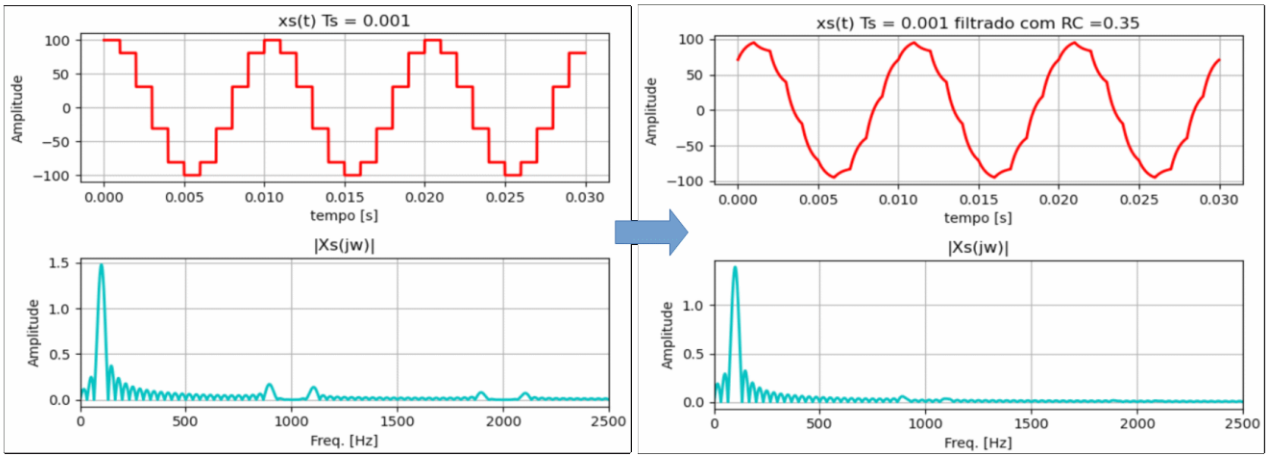

Figura 20 - Comparando o sinal reconstruído T_s = 0.001 s sem filtragem e com filtragem.

A figura 21 apresenta a mesma operação de filtragem para o sinal reconstruído T_s = 0.0025 s.

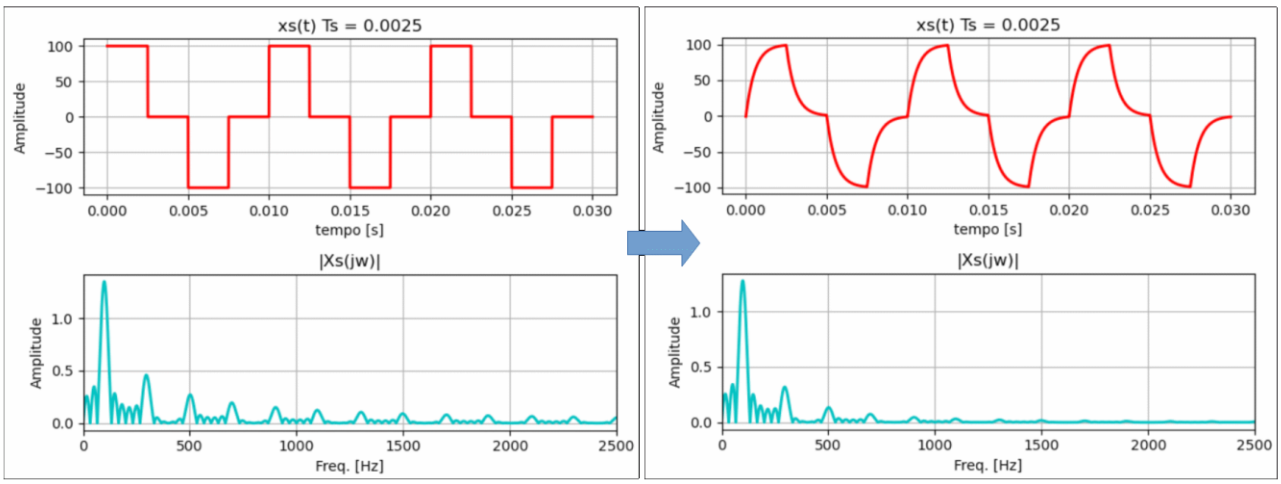

Figura 21 - Comparando o sinal reconstruído T_s = 0.0025 s sem filtragem e com filtragem.

Observando a figura 21 notamos a presença de mais componentes de frequência indesejadas. Isso vem da influencia da taxa de amostragem do sinal. E que sinais com períodos maiores de amostragem necessitam de filtros com melhores para atingir a mesma qualidade no sinal reconstruído.

Processamento Digital de Sinais - PDS

Projeto Amostragem:

O código a seguir permite gravar o microfone do PC.

```
import numpy as np
import matplotlib.pyplot as plt
import sounddevice as sd
from scipy.io.wavfile import write

# Frequência de amostragem
freqAmost = 44100
# Duração da gravação
duration = 5

recording = sd.rec(int(duration * freqAmost),
                   samplerate=freqAmost, channels=1)

print('Gravando...')

sd.wait()

print('Salvando gravação ......')

write("Gravacao_0.wav ", freqAmost, recording)

t = np.arange(0,len(recording)/freqAmost,(1/freqAmost))

fig, ax = plt.subplots()
ax.plot(t recording, 'r-', lw=2)
ax.set_ylabel("Amplitude")
ax.set_xlabel("tempo [s]")
ax.grid(True)
# ax.set_xlim(0,1 )
ax.set_title('Gravação')

plt.show()
```

Grave a sua voz em um arquivo, variando a frequência de amostragem em:

I. 1000 Hz;
II. 8000 Hz
III. 44100 Hz.

Depois plote o gráfico da transformada de Fourier dos arquivos. Analisando os três arquivos quanto ao seu tamanho, sua transformada de Fourier e sua capacidade de transmitir a informação (entender o que está sendo falado e identificar a pessoa que está falando) contida na mensagem de voz, responda:

a) Qual tem a melhor qualidade de áudio?
b) Qual arquivo se perde menos informação?
c) Qual arquivo consegue entregar a informação com menor custo (menor tamanho)?
d) Qual arquivo tem o melhor custo beneficio (qualidade e tamanho)?