



**INSTITUTO
FEDERAL**

Santa Catarina

Câmpus
São José

Modelo COM1

Sistemas de Comunicações I

Gabriel Luiz Espindola Pedro

18 de Março de 2024

Sumário

1. Introdução	3
2. Fundamentação teórica	3
3. Análise dos resultados	4
4. Conclusões	9

1. Introdução

O trabalho tem como objetivo a análise e manipulação de sinais no domínio do tempo e da frequência utilizando linguagens de programação e ferramentas para simulação das características e comportamento da interação de sinais em sistemas de comunicação.

Foi utilizada a linguagem de programação Python juntamente com Jupyter Notebook para a descrição e simulação dos sinais. Os pacotes utilizados em Python foram Numpy, para manipulação de vetores e materiais matemáticos, Matplotlib, para a geração de gráficos e SciPy para a análise de sinais.

2. Fundamentação teórica

A fundamentação teórica do trabalho é baseada em sinais e sistemas, onde sinais são funções de uma ou mais variáveis independentes, e sistemas são entidades que manipulam sinais. A análise de sinais e sistemas é uma área fundamental para a compreensão de sistemas de comunicação, pois permite a análise de sinais em diferentes domínios, como o domínio do tempo e da frequência.

Foi utilizado o conceito de transformada de Fourier para a análise de sinais no domínio da frequência. A transformada de Fourier é uma ferramenta matemática que permite a análise de sinais no domínio da frequência, ou seja, permite a decomposição de um sinal em suas componentes de frequência. Como algoritmo foi utilizada a transformada rápida de Fourier (FFT), que é uma implementação eficiente da transformada de Fourier discreta.

Para realizar uma medição da distribuição de energia de um sinal no domínio da frequência, utilizamos a densidade espectral de potência, pela função `scipy.signal.welch` que realiza o cálculo utilizando o método de Welch, gerando uma aproximação para o valor de densidade espectral.

3. Análise dos resultados

Foi iniciado o experimento criando três cossenos e gerando um sinal que é a composição da soma dos três cossenos. Na coluna da esquerda, temos a representação dos sinais no domínio do tempo, e na coluna da direita, temos a representação dos sinais no domínio da frequência. Sendo os sinais:

$$\begin{aligned}x_1(t) &= 6 \cos(2\pi 1000t) \\x_2(t) &= 2 \cos(2\pi 3000t) \\x_3(t) &= 4 \cos(2\pi 5000t)\end{aligned}\tag{1}$$

$$s(t) = x_1(t) + x_2(t) + x_3(t)\tag{2}$$

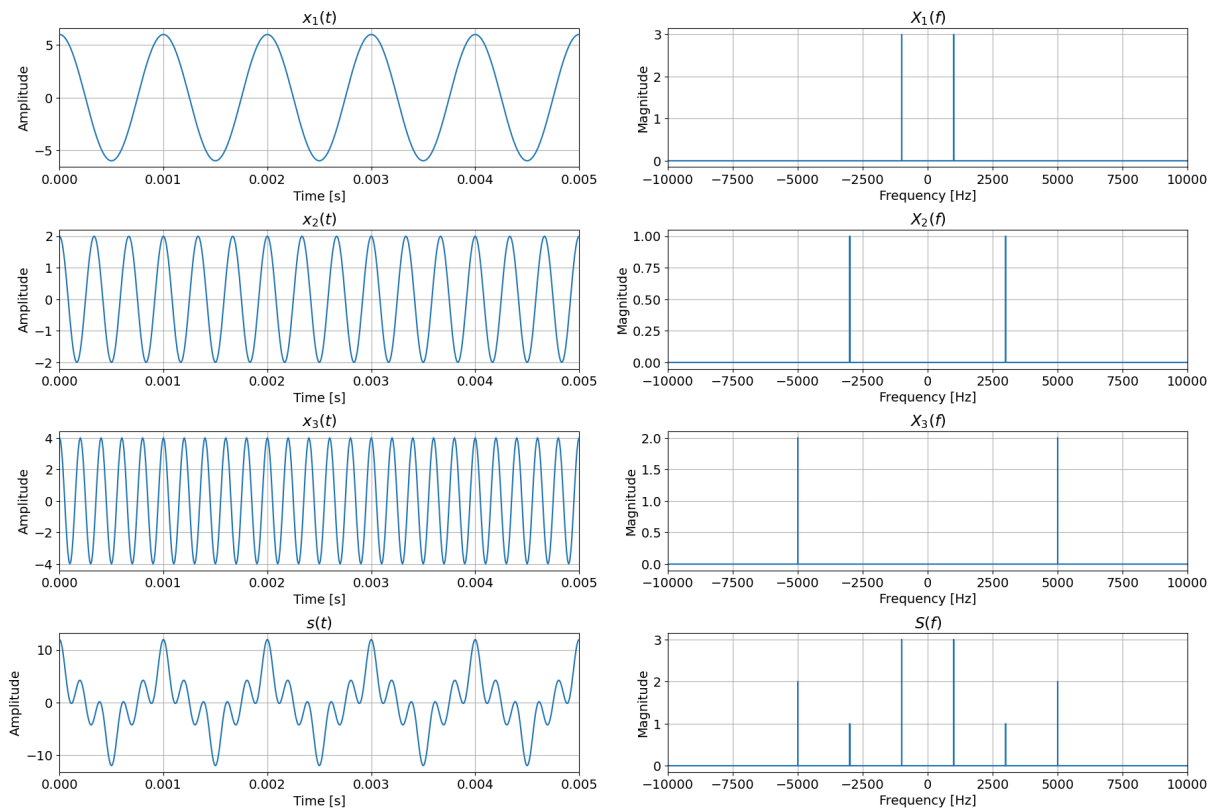


Figura 1: sinais no domínio do tempo e da frequência

Notamos que no domínio da frequência, o sinal é muito mais claro e fácil de ser analisado, pois conseguimos visualizar as componentes de frequência que compõem o sinal.

Calculamos a potência média do sinal utilizando a função `np.linalg.norm` e encontramos um valor de 28, que é a soma do quadrado das amplitudes dos impulsos do sinal analisado na figura anterior.

Utilizando a função `scipy.signal.welch` para calcular a densidade espectral de potência do sinal, encontramos o gráfico que descreve a distribuição de energia do sinal no domínio da frequência, se aproximando do valor esperado.

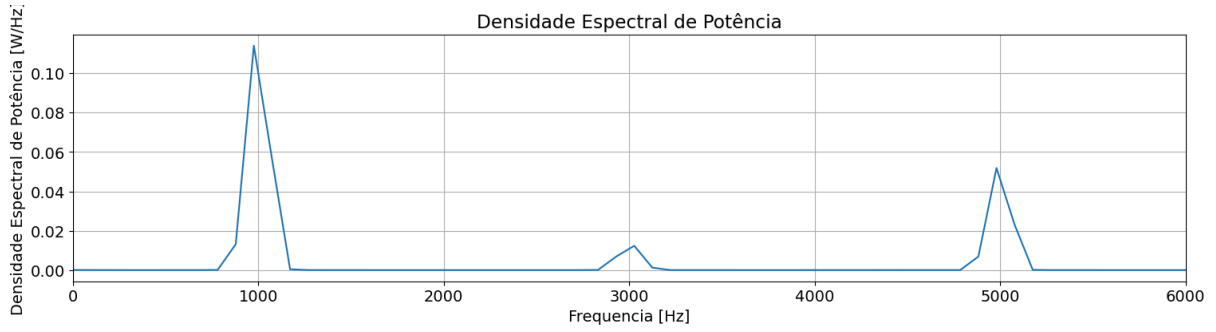


Figura 2: densidade espectral de potência utilizando o algoritmo de Welch

Trabalhando agora com senos foi possível gerar um sinal que se aproxima de uma onda quadrada, porém com poucas harmônicas. A figura a seguir mostra os sinais no domínio do tempo e da frequência:

$$\begin{aligned}
 x_1(t) &= 6 \sin(2\pi 1000t) \\
 x_2(t) &= 2 \sin(2\pi 3000t) \\
 x_3(t) &= 1 \sin(2\pi 5000t) \\
 s(t) &= x_1(t) + x_2(t) + x_3(t)
 \end{aligned} \tag{3}$$

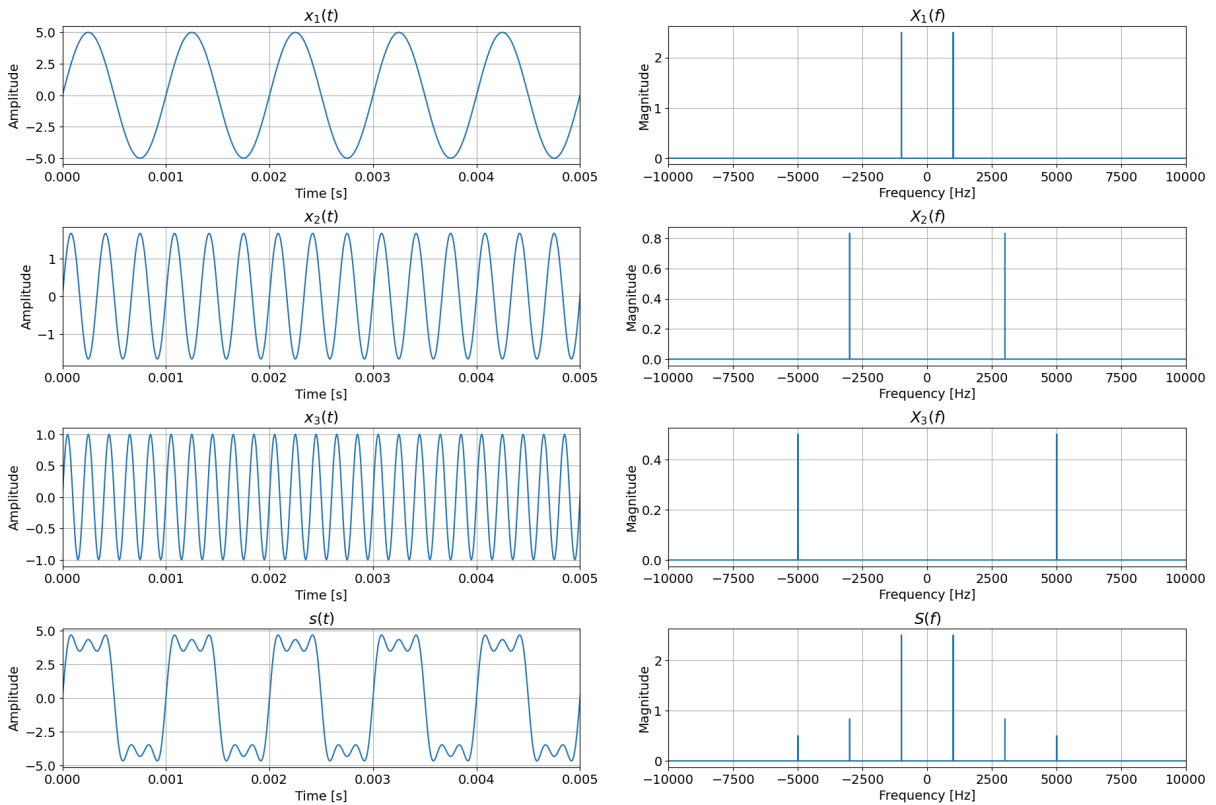


Figura 3: sinais no domínio do tempo e da frequência

Assim como no exemplo foram utilizados os cossenos, é muito mais simples analisar o sinal no domínio da frequência, pois conseguimos visualizar as componentes de frequência que compõem o sinal.

Notamos também que adicionando harmônicas ao sinal, o sinal cada vez mais se aproximaria de uma onda quadrada.

Para retornarmos ao sinal original, geramos por meio de vetores sinais que funcionam como filtros passa-baixa, passa-alta e passa-faixa. A figura a seguir mostra os sinais no domínio da frequência:

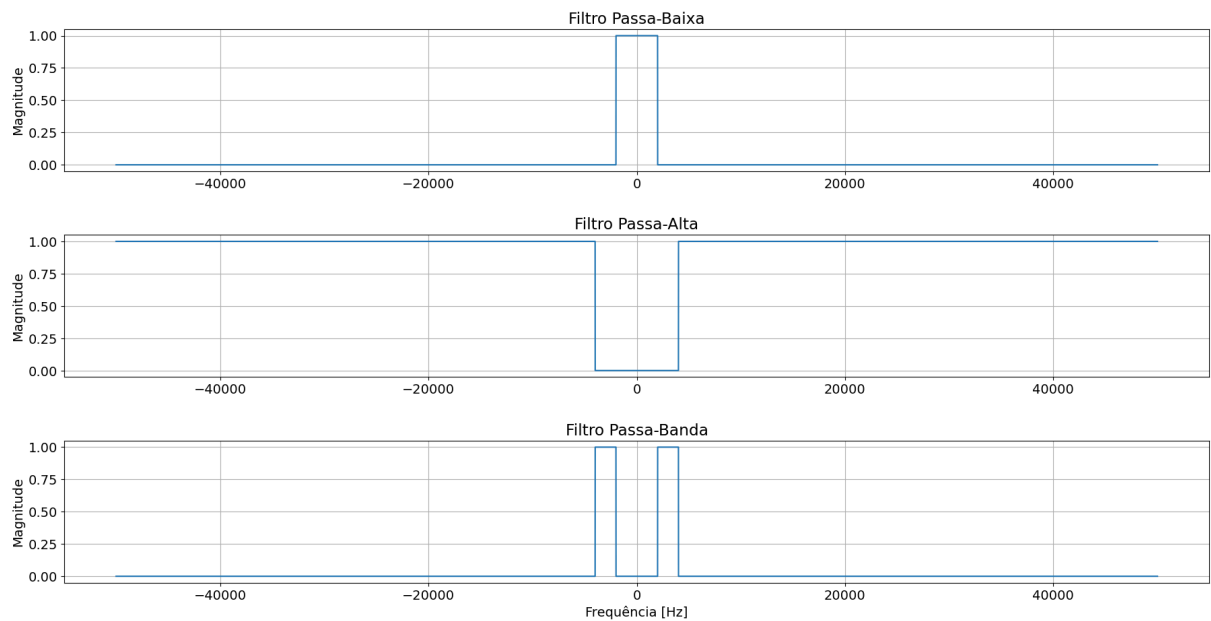


Figura 4: filtros ideais, passa-baixa, passa-alta e passa-faixa

Desta maneira quando multiplicarmos o sinal $s(t)$ composto pelos senso, podemos separar as componentes do sinal original, como mostra a figura a seguir:

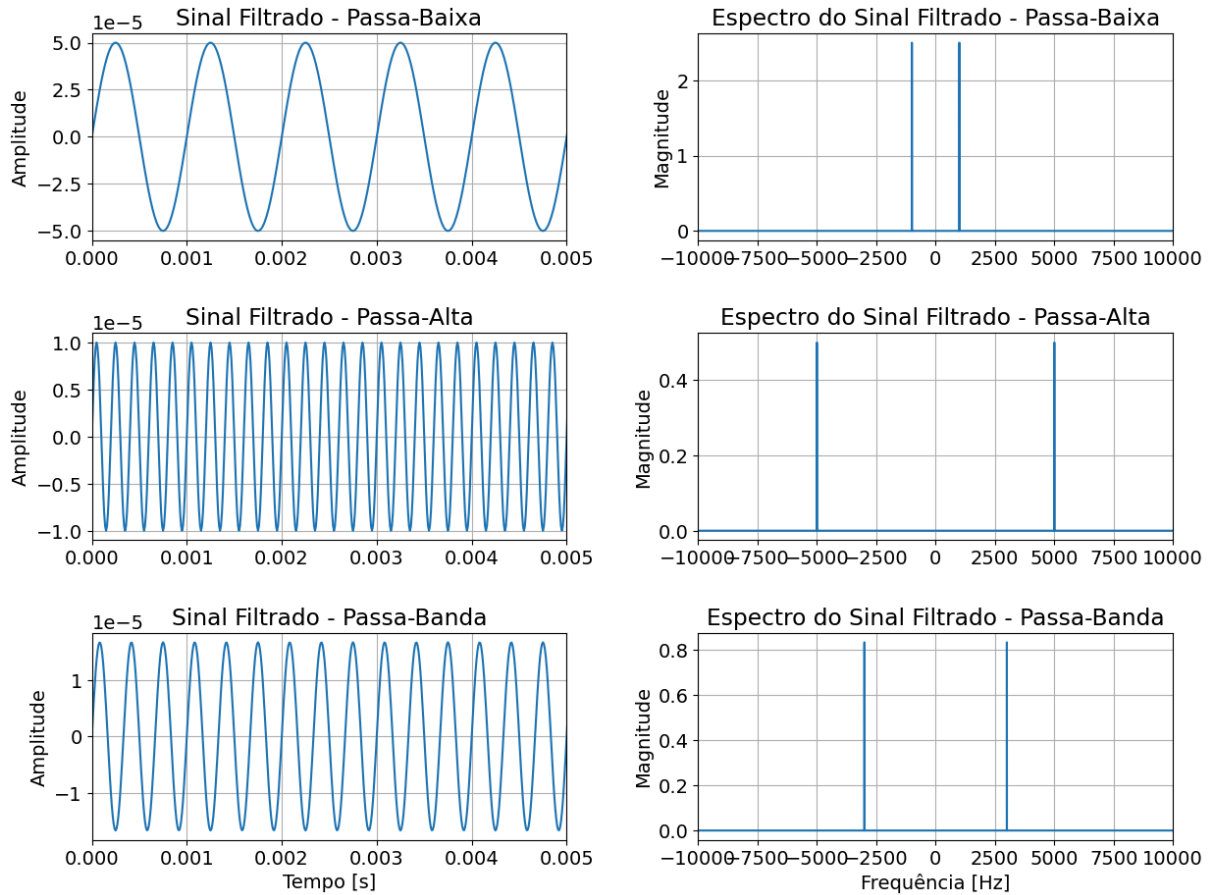


Figura 5: componentes recuperadas do sinal $s(t)$

Com o sinal original filtrado, podemos observar que as componentes do sinal original foram separadas, e que o sinal original foi recuperado.

Por fim foi gerado um ruído e analisado ele a parte. A figura a seguir mostra um ruído gaussiano gerado utilizando a função `np.random.normal`:

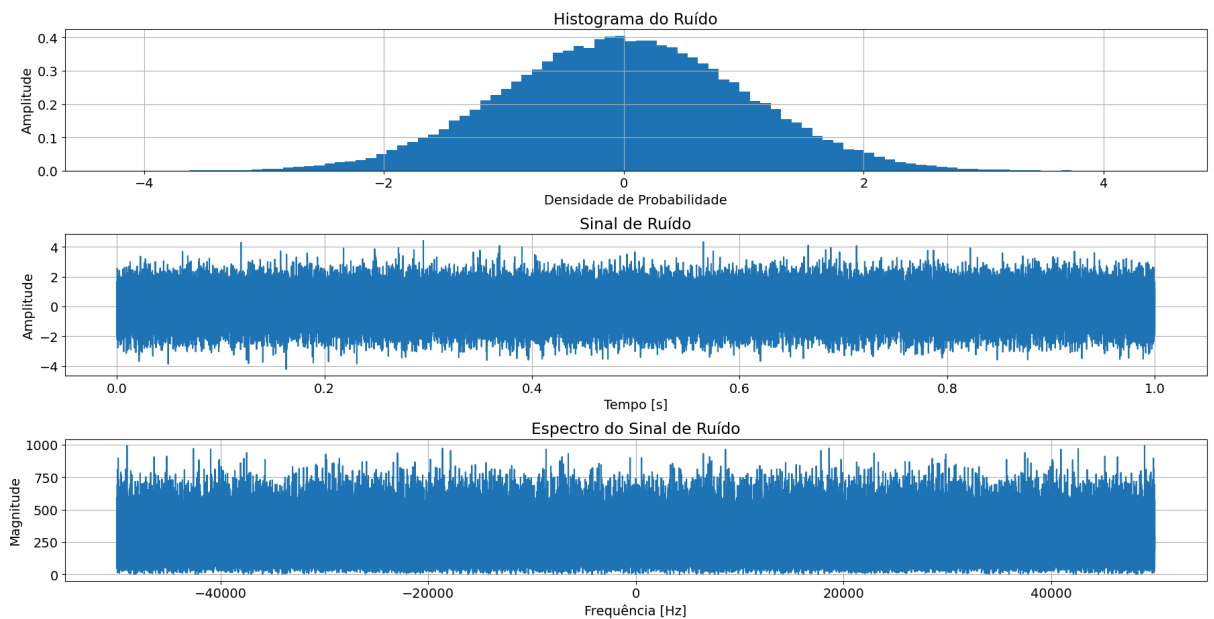


Figura 6: ruído gaussiano

É possível verificar que é um ruído gaussiano analisando o histograma gerado, verificamos que a distribuição do ruído é normal, com média 0 e desvio padrão 1. É notável também que ele possui componentes em todo o espectro de frequência.

Sua autocorrelação indica que o ruído é branco, ou seja, não possui correlação entre as amostras, apenas quando o ruído se sobrepõe completamente, ou seja em zero na autocorrelação, ele possui um valor que se distancia de zero. A figura a seguir mostra a autocorrelação do ruído:



Figura 7: Autocorrelação do ruído

Para filtrar o ruído, foi utilizado um filtro FIR, que é um filtro de resposta finita ao impulso. A figura a seguir mostra o filtro FIR de ordem 50 e frequência de corte de 1 kHz:

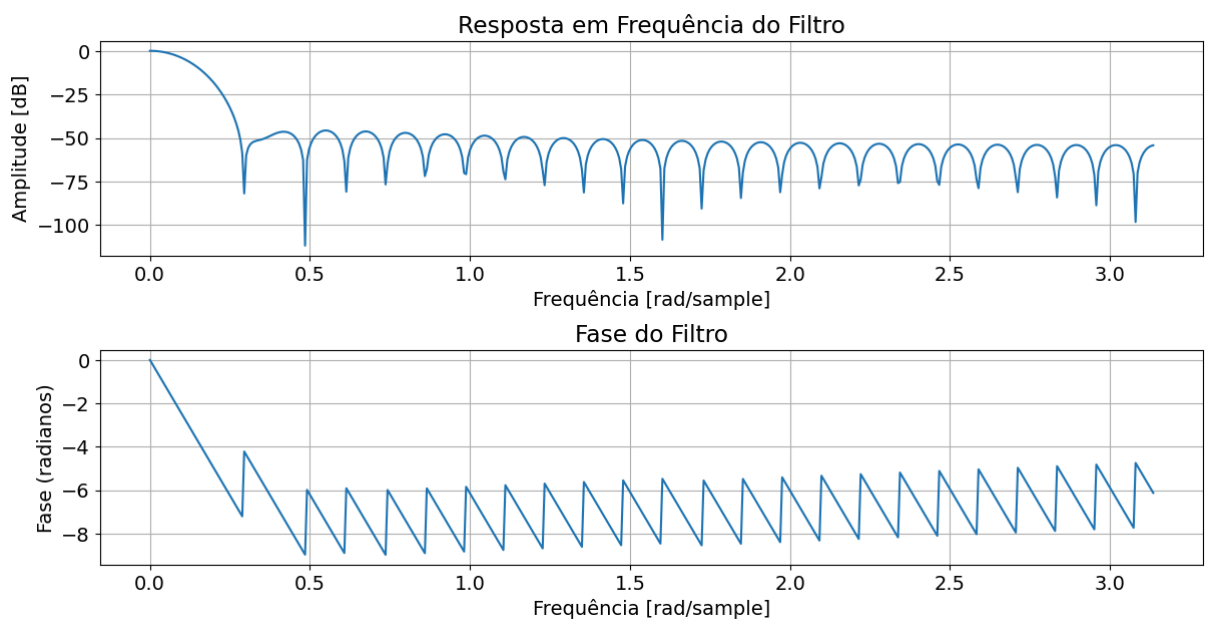


Figura 8: filtro FIR

Ao ser aplicado este filtro ao ruído, verificamos uma diminuição das componentes de alta frequência no espectro de frequência e um estreitamento da distribuição gaussiana.

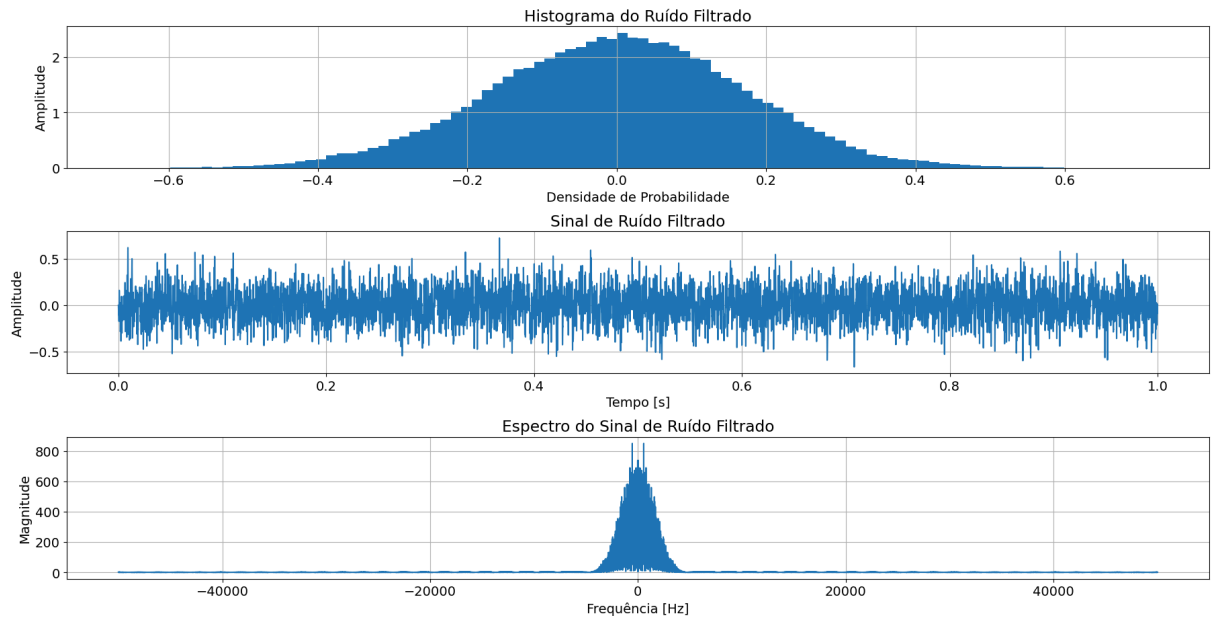


Figura 9: ruído filtrado

4. Conclusões

Realizada a análise é possível verificar que a recuperação de sinais no domínio da frequência é muito mais simples e eficiente do que no domínio do tempo, pois conseguimos visualizar as componentes de frequência que compõem o sinal.