



**INSTITUTO  
FEDERAL**

Santa Catarina

---

Câmpus  
São José

## **Projeto de Filtros IIR**

Processamento de Sinais Digitais

**Arthur Cadore Matuella Barcella**

12 de Agosto de 2024

Engenharia de Telecomunicações - IFSC-SJ

# Sumário

<b>1. Introdução .....</b>	<b>3</b>
<b>2. Questão 1 .....</b>	<b>3</b>
2.1. Projeto do Filtro IIR no Simulink: .....	3
2.1.1. Pré-Distorção do Sinal: .....	3
2.1.2. Normalização do Filtro: .....	3
2.1.3. Cálculo das atenuações: .....	4
2.1.4. Cálculo dos parâmetros do filtro: .....	4
<b>3. Questão 2 .....</b>	<b>5</b>
3.1. Montagem do Filtro IIR no Simulink: .....	5
3.1.1. Resposta em Frequência (Magnitude): .....	5
3.1.2. Resposta em Frequência (Fase): .....	5
3.1.3. Magnitude e Fase do Filtro: .....	6
3.1.4. Resposta ao impulso: .....	6
3.1.5. Polos e Zeros do Filtro: .....	7
3.1.6. Espalhamento de potência do filtro: .....	7
3.2. Comparativo entre outros Desings: .....	8
3.3. Aplicando o filtro no sinal de entrada: .....	9
3.3.1. Script Matlab: .....	9
3.3.2. Resultados Obtidos: .....	12
3.3.2.1. Sinal de Entrada: .....	12
3.3.2.2. Sinal de 770Hz Filtrado: .....	13
3.3.2.3. Sinal de 852Hz Filtrado: .....	14
3.3.2.4. Sinal de 941Hz Filtrado: .....	15
<b>4. Conclusão .....</b>	<b>16</b>
<b>5. Referências .....</b>	<b>16</b>

# 1. Introdução

## 2. Questão 1

Usando a transformação bilinear, projete um filtro passa-baixas Butterworth que atenda as seguintes especificações:

$$0.9 \leq |H(e^{j\omega})| \leq 1 \quad (1)$$

$$0 \leq \omega \leq 0.2\pi \quad (2)$$

$$|H(e^{j\omega})| \leq 0.2 \quad (3)$$

$$0.3\pi \leq \omega \leq \pi \quad (4)$$

Considere também que:

- $T_s$  (tempo de amostragem) = 2
- Faça o mesmo projeto usando o MATLAB ou simulink. Plote a resposta em frequência

### 2.1. Projeto do Filtro IIR no Simulink:

#### 2.1.1. Pré-Distorção do Sinal:

O primeiro passo no desenvolvimento do projeto foi a montagem do filtro IIR no Simulink. Utilizamos a ferramenta para alterar os parâmetros do filtro procurando pelos valores que atendessem as especificações de banda de passagem e de rejeição.

Seguindo a formula para transformação bilinear para transformar a frequência analógica em frequência digital:

$$\omega = \frac{2}{T_s} \tan\left(\frac{\omega}{2}\right) \quad (5)$$

Desta forma, temos que:

```
1 wp = 0.2*pi;  
2 wr = 0.3*pi;  
3 ts = 2;  
4  
5 omega_ap = (2/ts)*tan(wp/2)  
6 omega_ar = (2/ts)*tan(wr/2)
```

#### 2.1.2. Normalização do Filtro:

A normalização do filtro é realizada para que o filtro projetado atenda as especificações de banda de passagem e de rejeição.

Para o processo de normalização, seguimos a tabela apresentada em aula que contém a seguinte transformada:

$$\Omega'_p = \frac{1}{a} \quad (6)$$

$$\Omega'_r = \frac{1}{a} \left( \frac{\Omega_r}{\Omega_p} \right) \quad (7)$$

Desta forma, temos que:

```
1 a = 1;
2 omega_p_linha = 1/a
3 omega_r_linha = omega_p_linha * (omega_ar/omega_ap)
```

### 2.1.3. Calculo das atenuações:

Para o calculo das atenuações, utilizamos a formula apresentada em aula.

Para a banda de passagem:

$$G_p = 20 \log_{10}(1 - \sigma_p) \quad (8)$$

Para a banda de rejeição:

$$G_r = 20 \log_{10}(\sigma_r) \quad (9)$$

Desta forma:

```
1 sigma_p = 0.9;
2 sigma_r = 0.2;
3
4 atenuacao_p = -1 * (20*log10(sigma_p))
5 atenuacao_r = -1 * (20*log10(sigma_r))
```

### 2.1.4. Calculo dos parâmetros do filtro:

$$e = \sqrt{10^{0,1,A_p} - 1} \quad (10)$$

$$n \geq \frac{\log_{10}\left(\frac{10^{0,1,A_{r-1}}}{e^2}\right)}{2 \log_{10} \Omega'_r} \quad (11)$$

```
1 eps = sqrt((10^(0.1*atenuacao_p))-1)
2
3 numerador = log10((10^(0.1*atenuacao_r)-1) / eps^2);
4 denominador = 2*log10(omega_r_linha);
5
6 n = ceil(numerador/denominador)
```

## 3. Questão 2

Crie, usando MATLAB, um sinal de entrada composto de três componentes senoidais, nas frequências 770Hz, 852Hz e 941Hz, com  $\Omega_s = 8$  kHz. Projete, usando o simulink ou o MATLAB, um filtro IIR para isolar cada componente. Documente as especificações utilizadas. Faça comentários.

### 3.1. Montagem do Filtro IIR no Simulink:

O primeiro passo no desenvolvimento da questão foi a montagem do filtro IIR no Simulink seguindo as especificações solicitadas pela questão. Utilizamos a ferramenta para alterar o parâmetros do filtro procurando pelos valores que atendessem as especificações de banda de passagem e de rejeição.

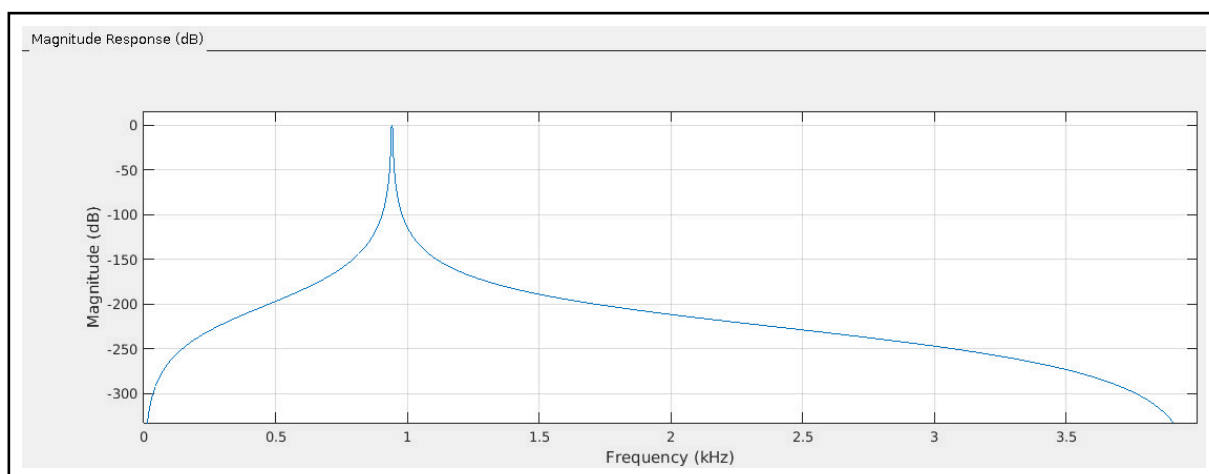
Os valores apresentados abaixo foram obtidos para o filtro de 941Hz, porem, foram realizados mais duas baterias de testes exatamente como as apresentadas abaixo, para as frequências de 770Hz e 852Hz.

#### 3.1.1. Resposta em Frequência (Magnitude):

Inicialmente utilizamos a ferramenta de plot da resposta em frequência do filtro para analisar a magnitude do filtro. A figura abaixo apresenta a resposta em frequência do filtro para a frequência de 941Hz.

O filtro que estamos procurando precisa possuir uma alta atenuação na banda de rejeição e uma queda abrupta na banda de transição para a banda de rejeição. O filtro abaixo apresenta uma atenuação de aproximadamente 80dB em toda a banda de rejeição.

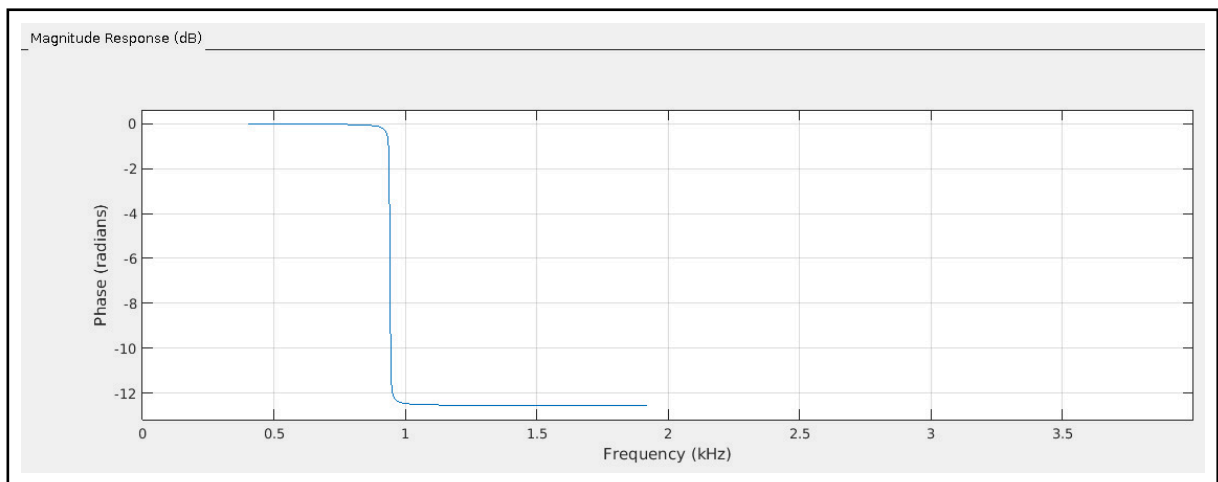
Figura 1: Elaborada pelo Autor



#### 3.1.2. Resposta em Frequência (Fase):

A fase do filtro é apresentada na figura abaixo. A fase do filtro é linear e apresenta apenas uma variação de  $11^\circ$  na banda de passagem.

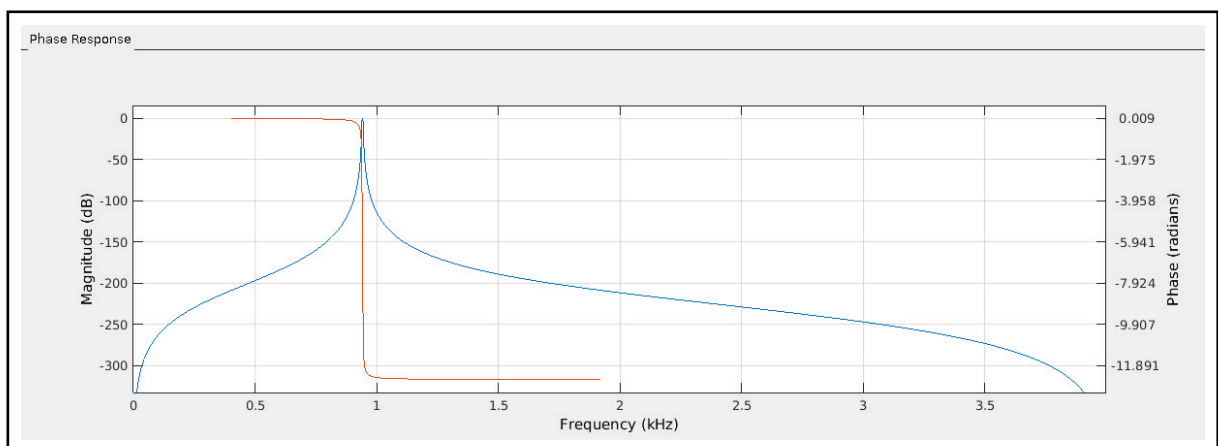
Figura 2: Elaborada pelo Autor



### 3.1.3. Magnitude e Fase do Filtro:

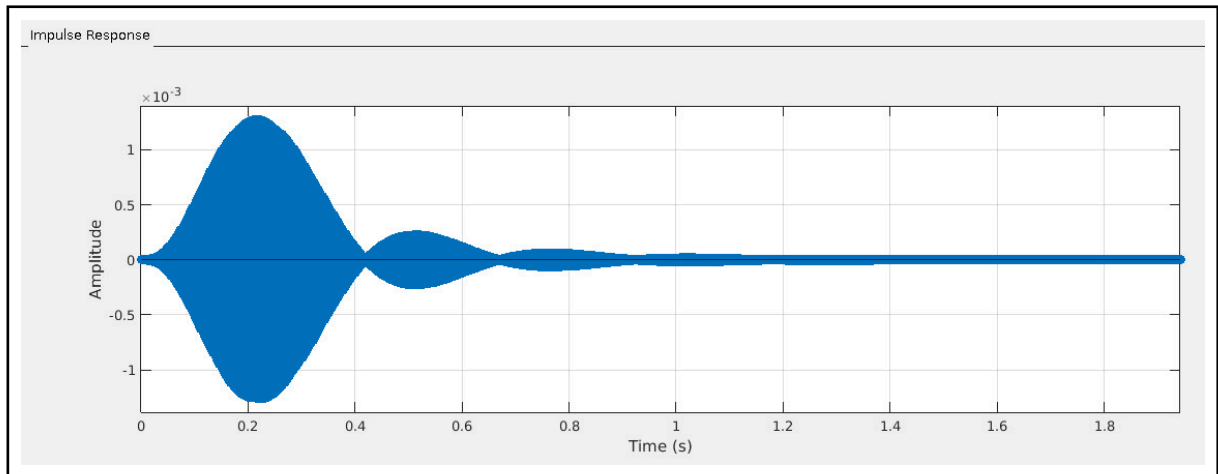
Abaixo podemos ver a magnitude e a fase do filtro sendo apresentadas juntas, note que a fase é linear dentro da banda de passagem, e a magnitude apresenta uma queda abrupta na banda de transição:

Figura 3: Elaborada pelo Autor



### 3.1.4. Resposta ao impulso:

Figura 4: Elaborada pelo Autor

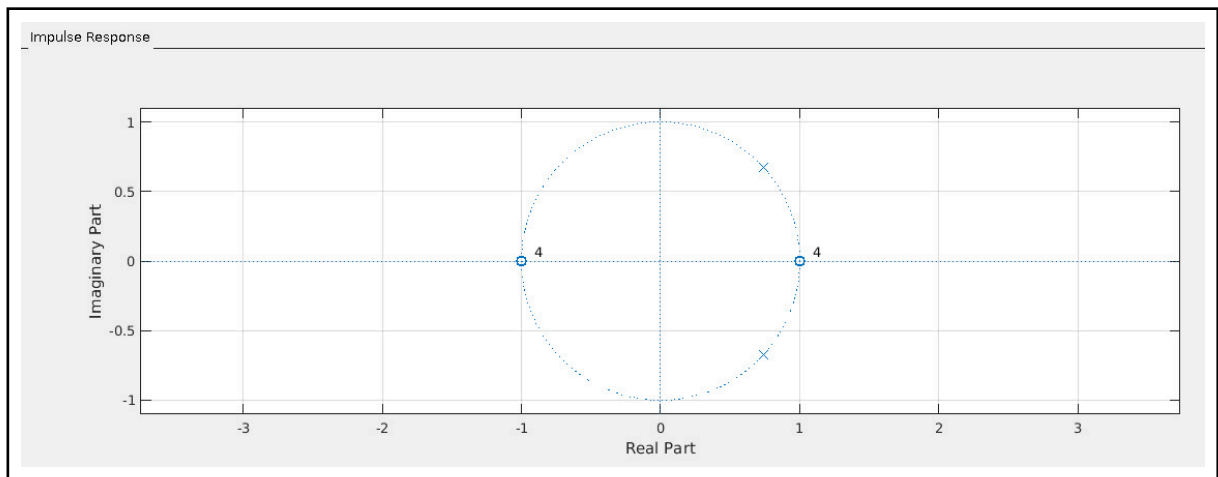


### 3.1.5. Polos e Zeros do Filtro:

Os polos e zeros do filtro são apresentados na figura abaixo. Note que o filtro possui 4 polos e 4 zeros, e que os polos estão localizados na região de interesse, ou seja, todos os polos estão dentro do círculo unitário.

Desta forma, o filtro é estável e possui uma resposta ao impulso finita oque é um requisito para o desenvolvimento deste projeto por se tratar de um filtro para sinais digitais.

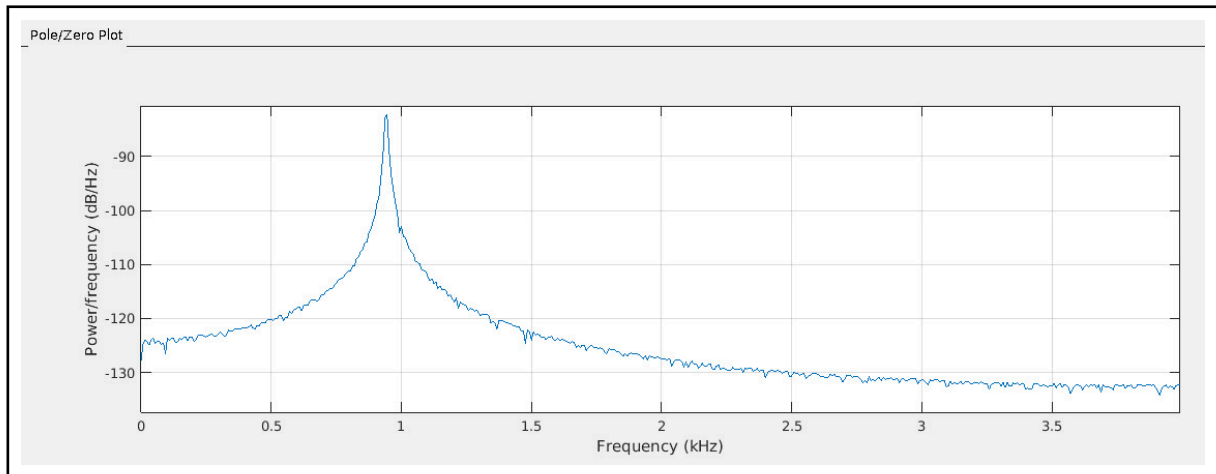
Figura 5: Elaborada pelo Autor



### 3.1.6. Espalhamento de potência do filtro:

Abaixo também está apresentado o espalhamento de potência do filtro. Note que o filtro apresenta uma atenuação de aproximadamente 80dB na banda de rejeição, também é possível verificar que grande parte da potência do sinal está concentrada na banda de passagem como esperado para as características do filtro.

Figura 6: Elaborada pelo Autor



Com base nestes valores, obtivemos os seguintes parâmetros para o filtro de 942Hz:

- Ordem do filtro: 8
- Banda de passagem: 941Hz - 943Hz
- Banda de transição Baixa: 930Hz - 941Hz
- Banda de transição Alta: 943Hz - 954Hz
- Frequência de amostragem: 8kHz
- Astop1 = 40dB
- Astop2 = 60dB
- Método de Desing: IIR-Butterworth

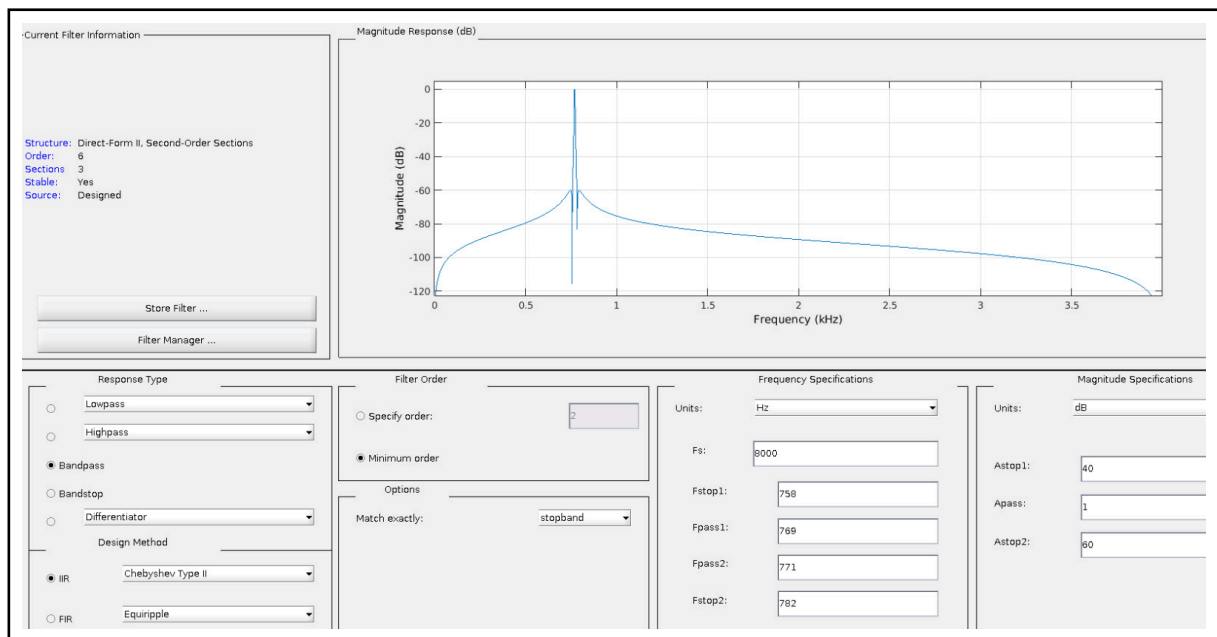
Os mesmos parâmetros foram utilizados para os filtros de 1kHz e 1.1kHz.

### 3.2. Comparativo entre outros Desings:

Neste projeto foi decidido utilizar o método de Butterworth para o desing do filtro, porem, também foram testados outros métodos, como Chebyshev:



Figura 7: Elaborada pelo Autor



Filtro de 941Hz Utilizando Chebyshev

Entretanto, durante os testes com os demais filtros, foi visto que a taxa de atenuação apresentada pelo filtro de Butterworth era superior a dos demais filtros, como é possível verificar acima para o filtro chebyshev em relação ao filtro Butterworth apresentado abaixo.

Note que na banda de passagem, os filtros se assemelham com atenuação próxima de zero e queda abrupta no início da banda de rejeição, porém, o filtro Butterworth apresenta uma atenuação de aproximadamente 80dB na banda de rejeição, enquanto o filtro Chebyshev apresenta uma atenuação de até 60dB e em seguida o filtro se mantém constante neste valor por uma faixa de frequência maior, o que atenua menos as componentes espectrais indesejadas.

Desta forma, o filtro que foi escolhido para o projeto e filtragem das componentes cossenoidais, foi o filtro Butterworth.

### 3.3. Aplicando o filtro no sinal de entrada:

Uma vez com o filtro projetado e os parâmetros definidos, é possível gerar um script matlab (octave) para gerar as três componentes de entrada, soma-las de maneira a criar um sinal único, análogo ao que seria transmitido pelo meio de transmissão, e aplicar o filtro projetado “no receptor” para cada uma das componentes, separando-as e verificando a resposta em frequência de cada sinal filtrado.

#### 3.3.1. Script Matlab:

```
1 clc; close all; clear all;
2
3 tmin = 0;
4 tmax = 2;
5 Omega_s = 8000;
```

```

6  Fs = Omega_s;
7  Ts = 1/Fs;
8  L = (tmax - tmin)/Ts;
9  t = 0:Ts:tmax-Ts;
10
11 f1 = 770;
12 f2 = 852;
13 f3 = 941;
14
15 s_t = sin(2*pi*f1*t) + sin(2*pi*f2*t) + sin(2*pi*f3*t);
16 S_f = fft(s_t);
17 S_f = abs(2*S_f/L);
18 S_f = fftshift(S_f);
19 freq = Fs*(-(L/2):(L/2)-1)/L;
20
21 figure;
22 subplot(2,1,1);
23 plot(t,s_t);
24 ylabel('Amplitude');
25 xlabel('Tempo (s)');
26 title('Sinal de entrada');
27
28 subplot(2,1,2);
29 plot(freq,S_f);
30 title('Espectro do sinal composto de três componentes senoidais');
31 xlabel('Frequência (Hz)');
32 ylabel('Amplitude');
33 xlim([-1000 1000]);
34 ylim([-0.1 1.1]);
35
36 % Filtro de 770Hz Butterworth
37 Num = Numerator = [
38     0.000000000007648974625344080593812088956; 0;
39     -0.000000000030595898501376322375248355826; 0;
40     0.000000000045893847752064483562872533739; 0;
41     -0.000000000030595898501376322375248355826; 0;
42     0.000000000007648974625344080593812088956
43 ];
44
45 Den = [1;
46     -6.573968478516325930627317575272172689438;
47     20.197703598146055981032986892387270927429;
48     -37.435767531546105146844638511538505554199;
49     45.612066895335409810741111868992447853088;
50     -37.354424964373507123127637896686792373657;
51     20.110025517173959030969854211434721946716;
52     -6.5312086462963758748401232878677546978;
53     0.991336860368820738109718604391673579812
54 ];
55
56 s_770hz = filter(Num, Den, s_t);
57 S_770hz = fft(s_770hz);
58 S_770hz = abs(2*S_770hz/L);
59 S_770hz = fftshift(S_770hz);
60 figure;
61 subplot(2,1,1);
62 plot(freq,S_f);
63 title('Espectro do sinal composto de três componentes senoidais');

```

```

64 xlabel('Frequência (Hz)');
65 ylabel('Amplitude');
66 xlim([-1000 1000]);
67 ylim([-0.1 1.1]);
68 subplot(2,1,2);
69 plot(freq, S_770hz);
70 title('Componente do sinal de 770 Hz filtrado com passa-faixa
Butterworth. ');
71 xlabel('Frequência (Hz)');
72 ylabel('Amplitude');
73 xlim([-1000 1000]);
74 ylim([-0.1 1.1]);
75
76 % Filtro de 852Hz Butterworth
77 Num = [
78     0.000000000007674623107715503777944271234; 0;
79     -0.000000000030698492430862015111777084937; 0;
80     0.000000000046047738646293025898839895191; 0;
81     -0.000000000030698492430862015111777084937; 0;
82     0.000000000007674623107715503777944271234
83 ];
84
85 Den = [1;
86     18.723854036105780807019982603378593921661;
87     -34.153268845102040529582154704257845878601;
88     41.404068891609767888439819216728210449219;
89     -34.07899654791336274683999363332986831665 ;
90     18.642505949254868369280302431434392929077;
91     -6.22711645042142070138879716978408396244;
92     0.991329630860964927663303569715935736895
93 ];
94
95
96 s_852hz = filter(Num, Den, s_t);
97 S_852hz = fft(s_852hz);
98 S_852hz = abs(2*S_852hz /L);
99 S_852hz = fftshift(S_852hz );
100 figure;
101 subplot(2,1,1);
102 plot(freq,S_f);
103 title('Espectro do sinal composto de três componentes senoidais');
104 xlabel('Frequência (Hz)');
105 ylabel('Amplitude');
106 xlim([-1000 1000]);
107 ylim([-0.1 1.1]);
108 subplot(2,1,2);
109 plot(freq, S_852hz );
110 title('Componente do sinal de 852 Hz filtrado com passa-faixa
Butterworth. ');
111 xlabel('Frequência (Hz)');
112 ylabel('Amplitude');
113 xlim([-1000 1000]);
114 ylim([-0.1 1.1]);
115
116 % Filtro de 940Hz Butterworth
117 Num = [0.000000000007698496068837793746781982269; 0;
118     -0.000000000030793984275351174987127929078; 0;

```

```

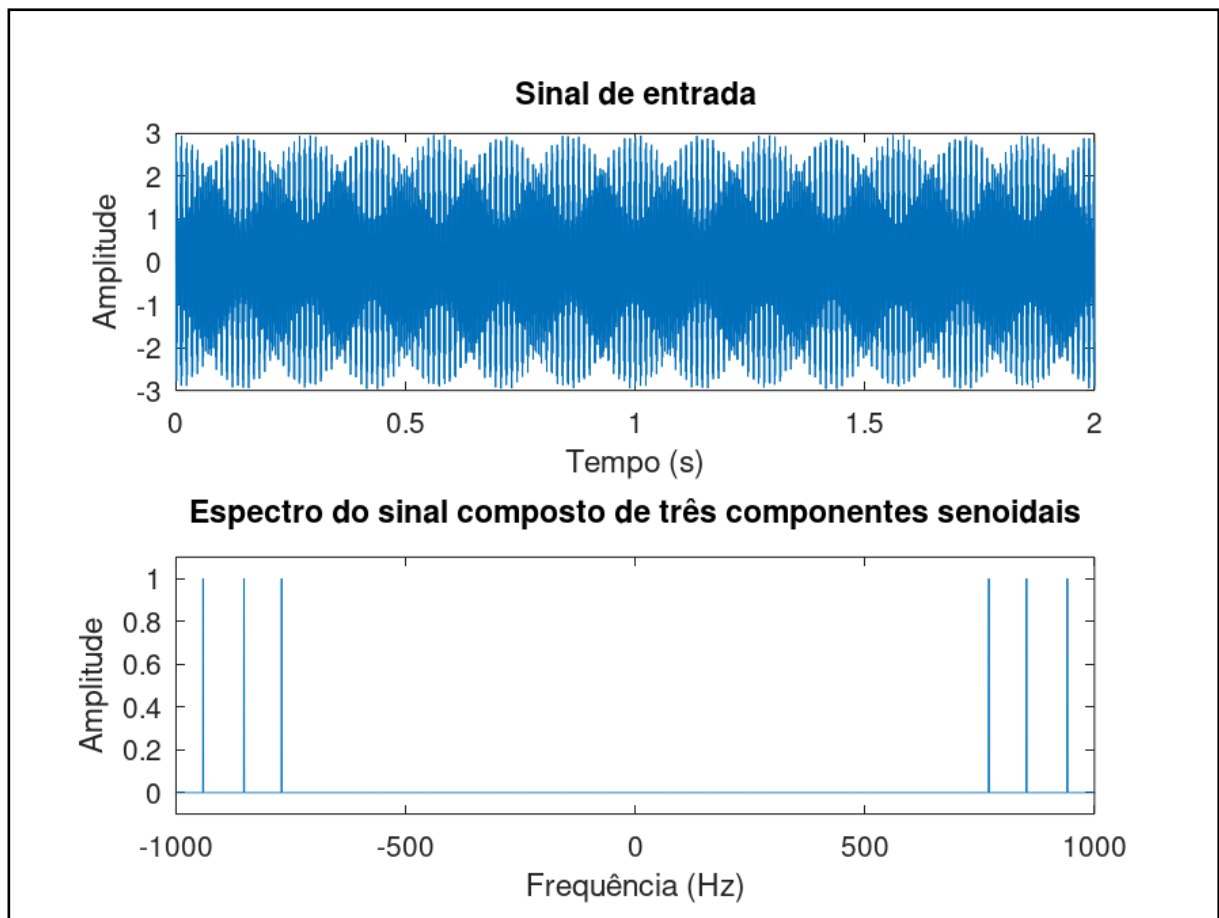
119         0.000000000046190976413026762480691893617; 0;
120     -0.000000000030793984275351174987127929078; 0;
121     0.000000000007698496068837793746781982269
122 ];
123 Den = [1;
124     -5.902149339413338857696089689852669835091;
125     17.054552786823688137474164250306785106659;
126     -30.518120678802585388211809913627803325653;
127     36.783803548311759357147820992395281791687;
128     -30.451702102949198547321429941803216934204;;
129     16.980399695100125256885803537443280220032;
130     -5.863697447546361019021787797100841999054;
131     0.991322918116968265778155000589322298765
132 ];
133
134 s_940hz = filter(Num, Den, s_t);
135 S_940hz = fft(s_940hz);
136 S_940hz = abs(2*S_940hz/L);
137 S_940hz = fftshift(S_940hz);
138 figure;
139 subplot(2,1,1);
140 plot(freq,S_f);
141 title('Espectro do sinal composto de três componentes senoidais');
142 xlabel('Frequência (Hz)');
143 ylabel('Amplitude');
144 xlim([-1000 1000]);
145 ylim([-0.1 1.1]);
146 subplot(2,1,2);
147 plot(freq, S_940hz);
148 title('Componente do sinal de 940 Hz filtrado com passa-faixa
    Butterworth. ');
149 xlabel('Frequência (Hz)');
150 ylabel('Amplitude');
151 xlim([-1000 1000]);
152 ylim([-0.1 1.1]);

```

### 3.3.2. Resultados Obtidos:

#### 3.3.2.1. Sinal de Entrada:

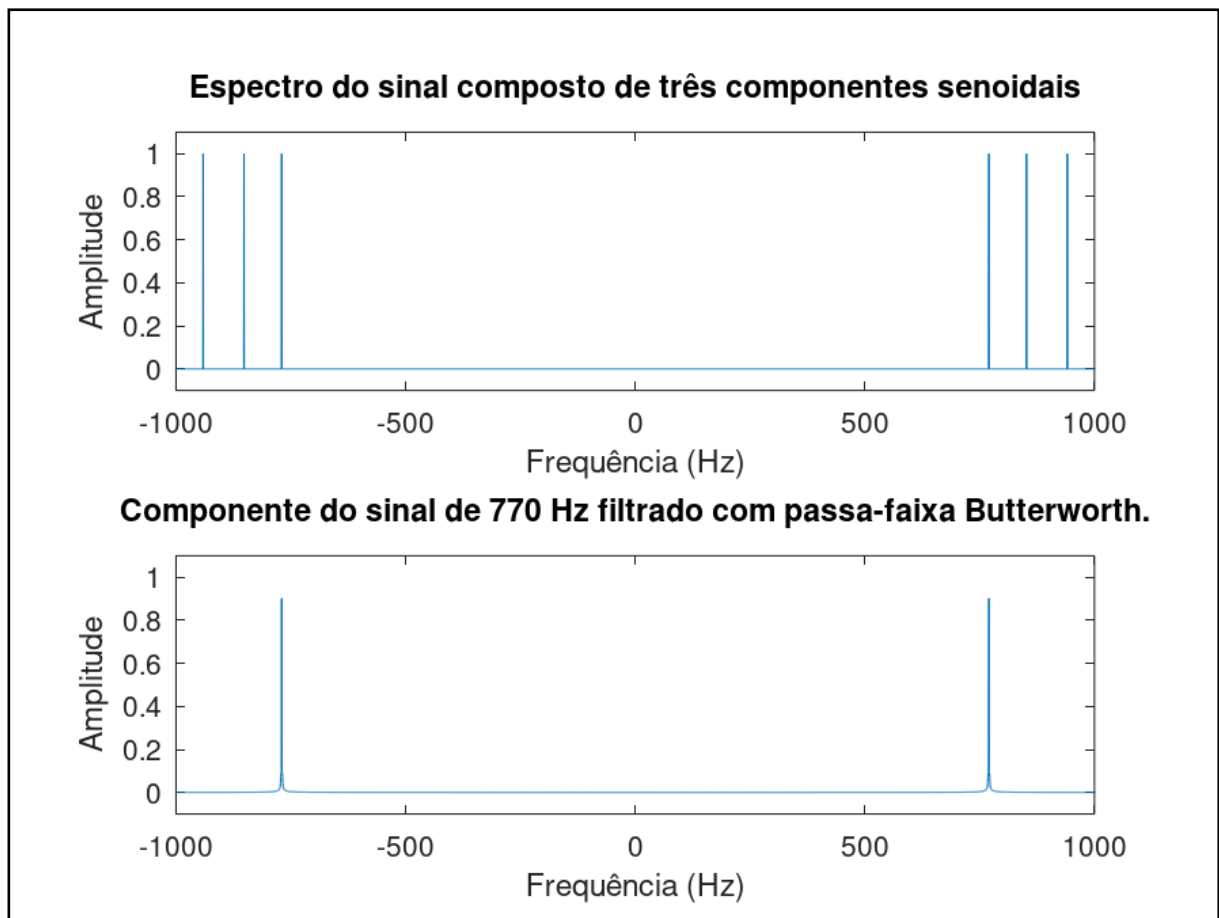
Figura 8: Elaborada pelo Autor



Filtro de 941Hz Utilizando Chebyshev

#### 3.3.2.2. Sinal de 770Hz Filtrado:

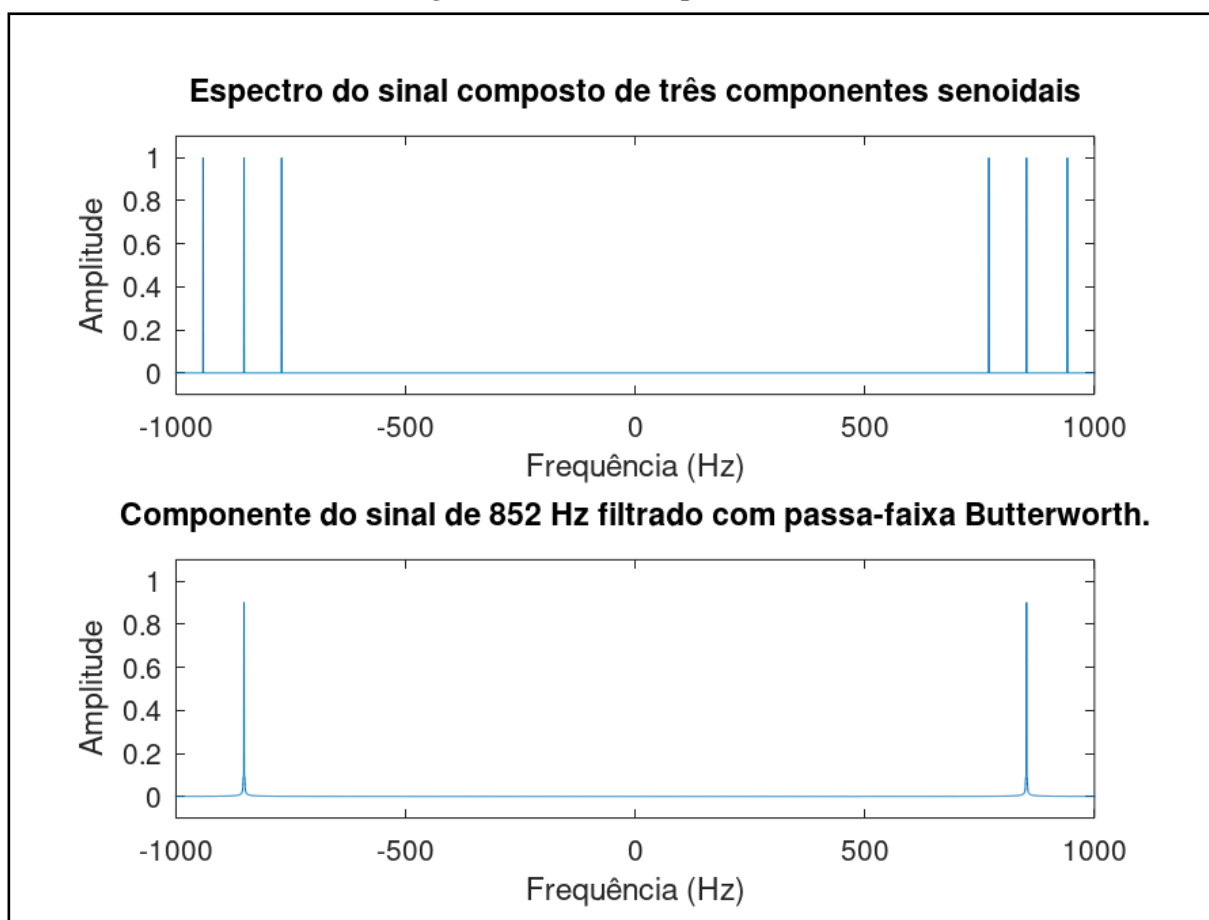
Figura 9: Elaborada pelo Autor



Filtro de 941Hz Utilizando Chebyshev

#### 3.3.2.3. Sinal de 852Hz Filtrado:

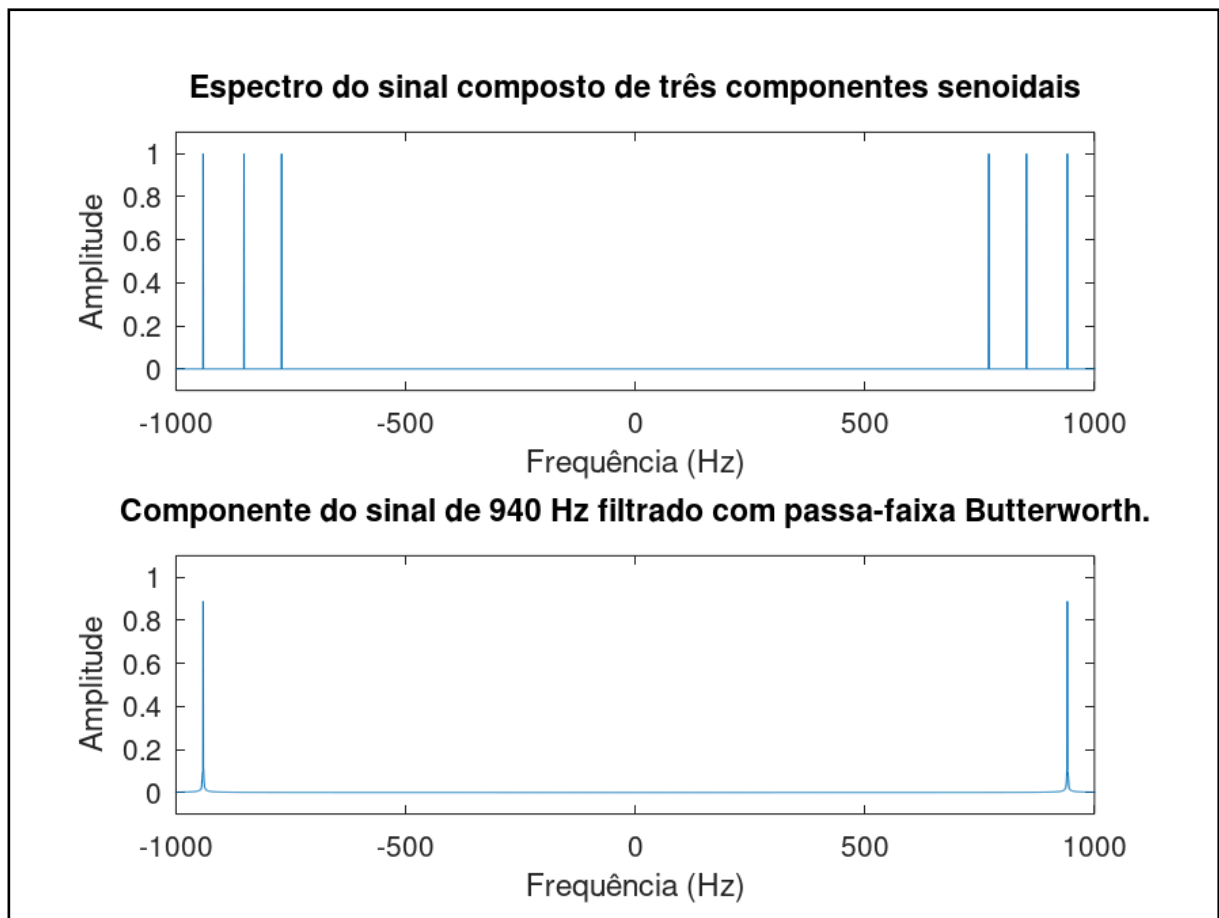
Figura 10: Elaborada pelo Autor



Filtro de 941Hz Utilizando Chebyshev

#### 3.3.2.4. Sinal de 941Hz Filtrado:

Figura 11: Elaborada pelo Autor



Filtro de 941Hz Utilizando Chebyshev

#### 4. Conclusão

#### 5. Referências