

## Simulation of DSP HW:

```
clear; clc; close all;
```

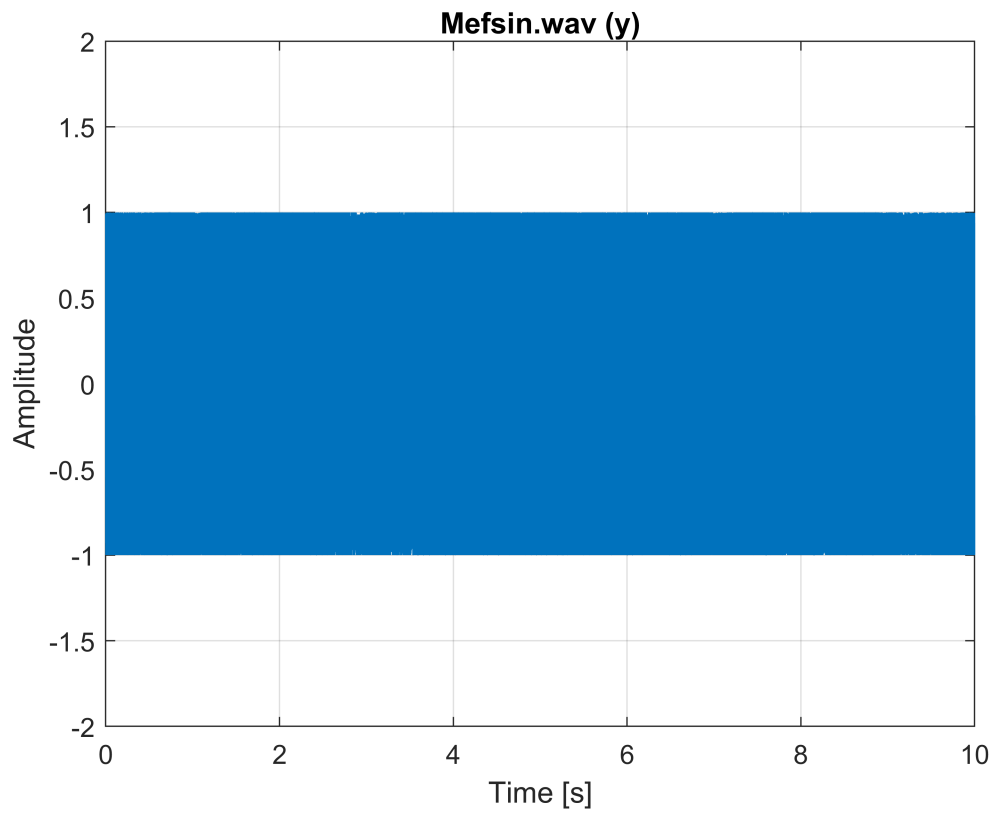
### Part-1:

الف) با استفاده از دستور **audioread** صوتی نویزدار **Mefsin.wav** را بخوانید. با استفاده از دستور **sound** فایل را پخش کنید. همانطور که مشاهده میکنید این فایل دارای نویز **Single Tone Noise** می باشد. با استفاده تبدیل فوریه اندازه طیف متفازن سیگنال را در بازه زیر رسم کنید.

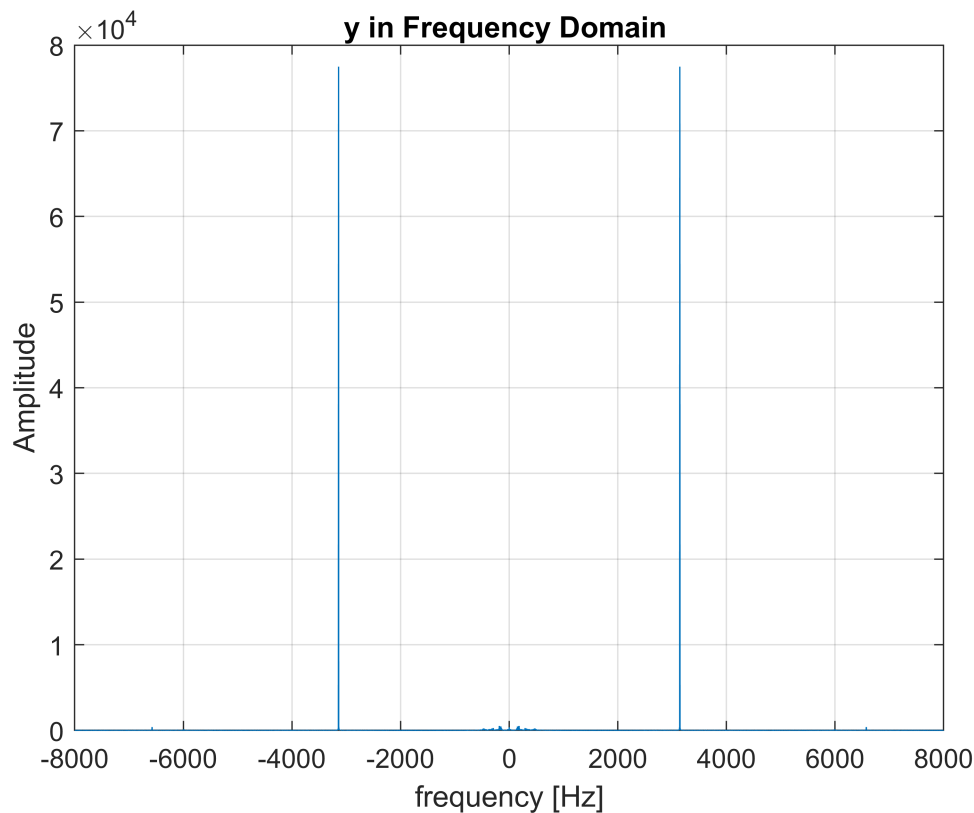
$$-\frac{F_s}{2} \leq \omega \leq \frac{F_s}{2}$$

```
[y , fs] = audioread("Mefsin.wav");
T_record = length(y)/fs;
ts = 1/fs;
t = 0:ts:T_record-ts;
delta_f = 1/T_record;
freq = -fs/2:delta_f:fs/2-delta_f ;

figure()
plot(t,y)
grid on
title("Mefsin.wav (y)" )
xlabel("Time [s]")
ylabel("Amplitude")
ylim([-2*max(abs(y)) , 2*max(abs(y)) ] )
```



```
Y_F = fftshift(fft(y));  
plot(freq,abs(Y_F))  
grid on  
title("y in Frequency Domain ")  
xlabel("frequency [Hz]")  
ylabel("Amplitude")
```



As it is vivid in the figure above, the frequency of single tone noise is:

$$\pm 3141 \text{ Hz}$$

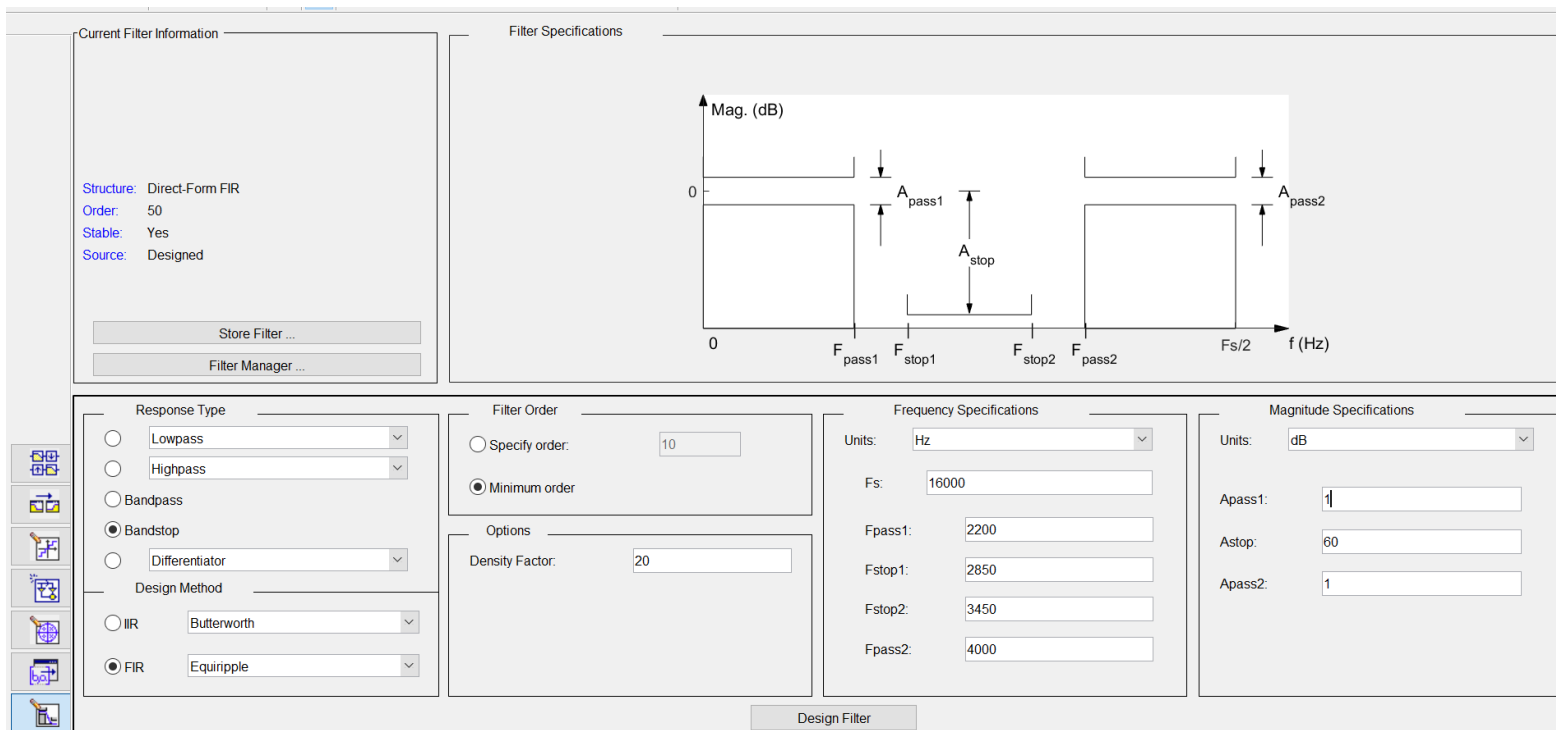
## Part-2:

ب) با مشاهده طیف فرکانس رسم شده در قسمت قبل، فرکانس نویز تک تون را گزارش کنید. با استفاده از ابزار **Filter Designer** فیلتری طراحی کنید تا این نویز را حذف کند. برای طراحی از فیلتر میان گذر به صورت **FIR** استفاده کنید و باید مرتبه آن حداقل باشد. مقادیر فرکانسهای قطع و همچنین تضعیف باندهای مختلف را میتوانید به دلخواه خود تنظیم کنید. به عنوان نمونه :

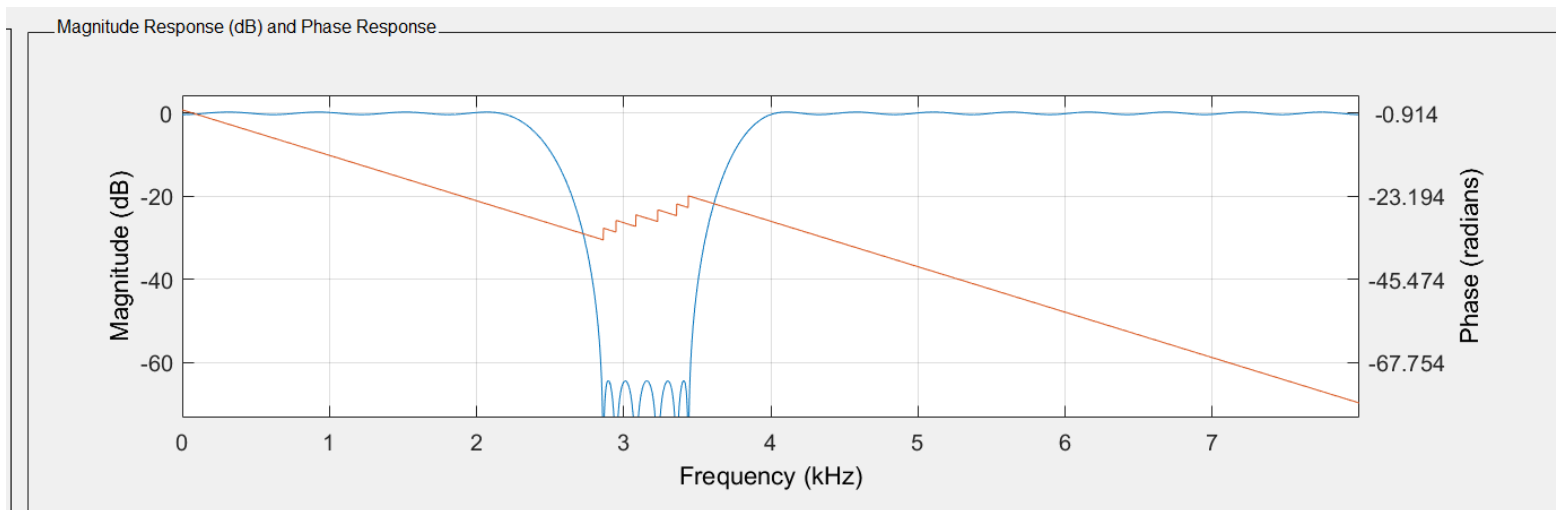
$$F_s = 16000, \quad F_{pass1} = 2200, \quad F_{stop1} = 2850, \quad F_{stop2} = 3450, \quad F_{pass2} = 4000$$

بعد از طراحی فیلتر آن را به درون **workspace** انتقال دهید (آن را **export** کنید) و با استفاده از دستور **freqz** پاسخ فرکانسی آن را رسم کنید. متغیری که درون **workspace** برای فیلتر شما آمده است را در نهایت آپلود نمایید.

After designing the filter we have:

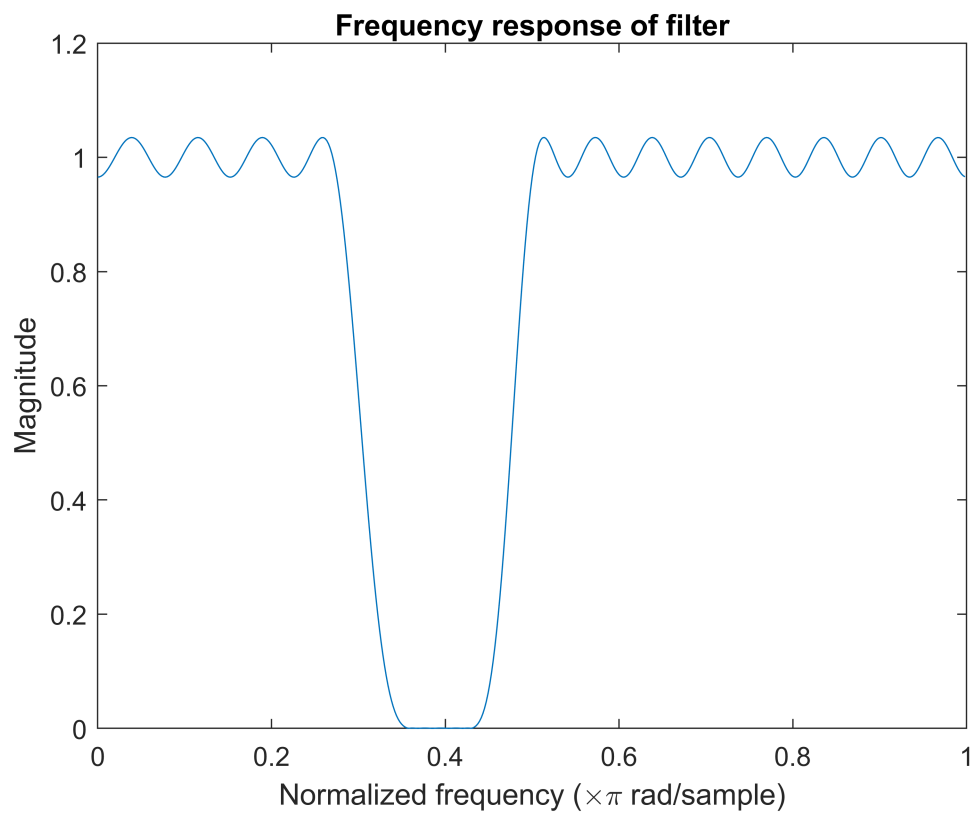


which gives the result as below:

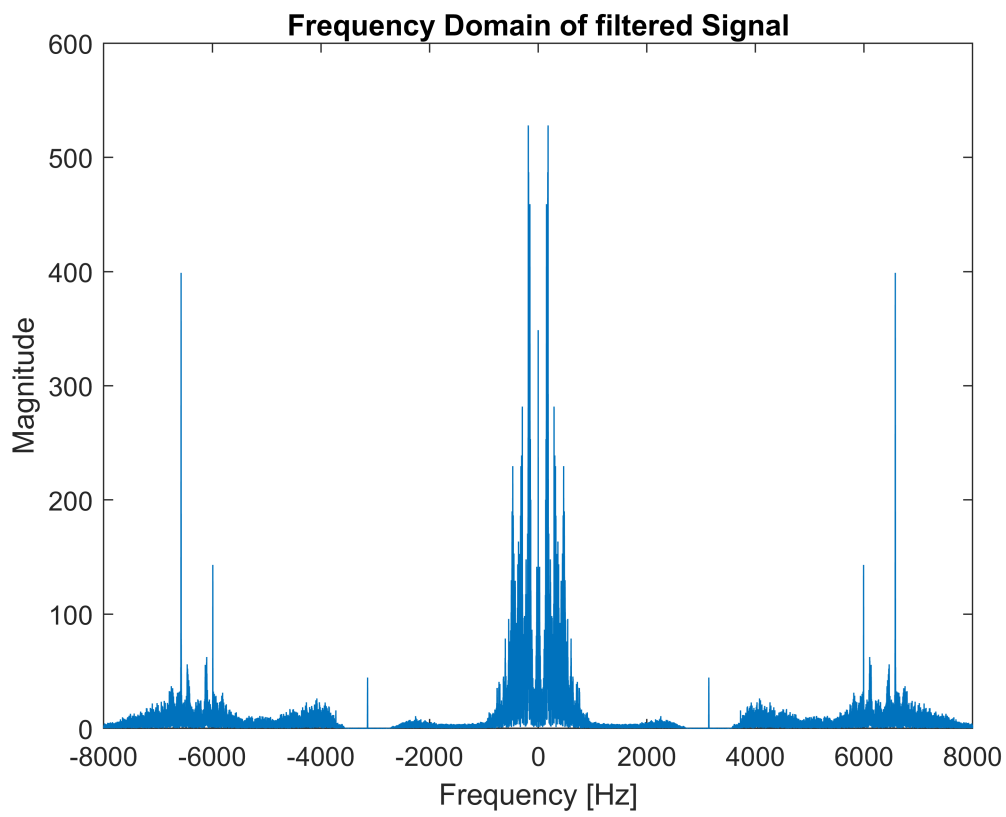


```
Band_Stop_Filter_Part2 = Part_2_Sim_DSP;

% Exporting to workspace gives us:  Num --> 1*63 &
% Hd_Exported_Filter_Designer.mat
b = Band_Stop_Filter_Part2.Numerator;
a = 1; % For FIR filters, the denominator is always 1
[h, w] = freqz(b, a);
plot(w/pi, abs(h));
xlabel('Normalized frequency (\times\pi rad/sample)');
ylabel('Magnitude');
title('Frequency response of filter');
```



```
% Filter previous Signal using this filter:  
y_filtered_Part2 = filter(b,a,y);  
  
figure()  
plot(freq , abs(fftshift(fft(y_filtered_Part2))));  
xlabel('Frequency [Hz]');  
ylabel('Magnitude');  
title('Frequency Domain of filtered Signal');
```



**Part-3:**

ج) با استفاده از دستور **sound** سیگنال بدون نویز را پخش کنید. در ادامه به کمک دستور **audiowrite** آن را به درون فایل با نام **Nonoise.wav** نوشته و ذخیره کنید.

د) فایل ساخته شده در قسمت قبل را بخوانید و آن را از فیلتر دیجیتال **FIR** با مشخصات زیر عبور دهید.

- فرکانس قطع باند عبور : **2000Hz**
- فرکانس قطع باند حذف : **2500Hz**
- مرتبه **40**
- **Equiripple**

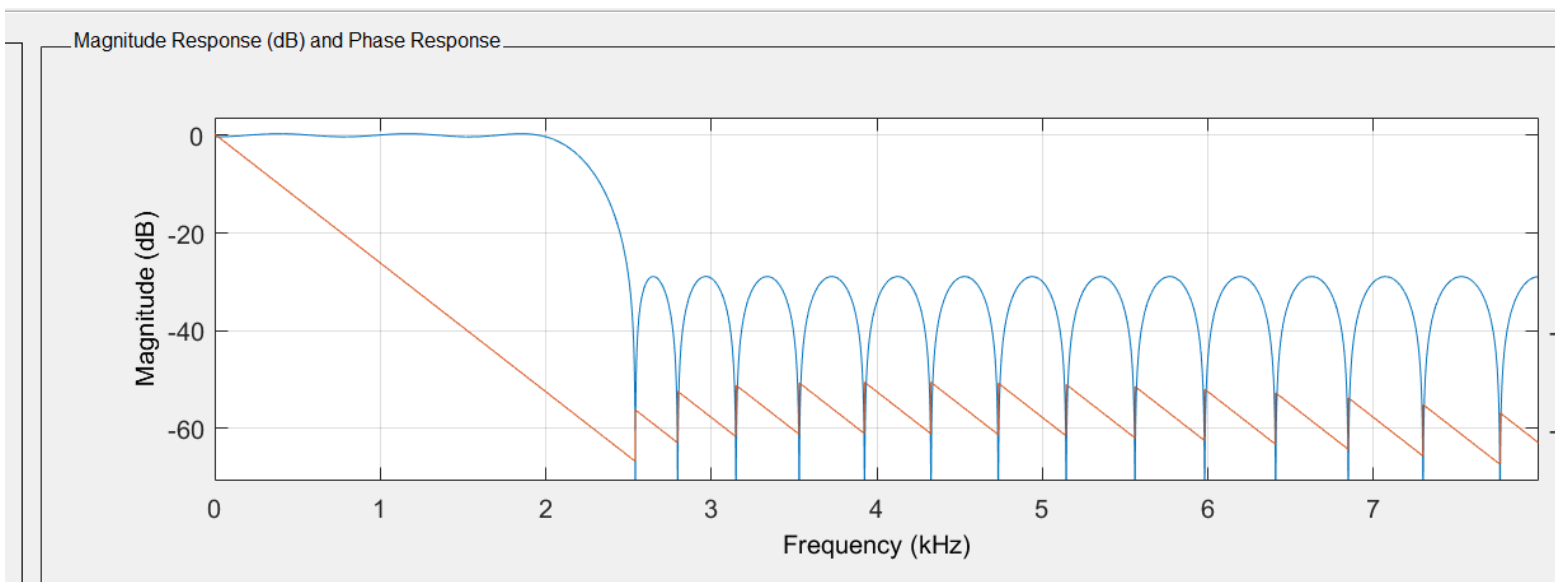
سپس به ترتیب مراحل زیر را انجام دهید.

1. صوت را قبل و بعد از فیلتر کردن پخش کنید. فیلتر را همانند بخش ب درون پوشه ذخیره کنید.
2. سیگنال ها را قبل و بعد از فیلتر در حوزه زمان رسم کنید.
3. اندازه طیف سیگنال ها را به صورت متقارن را قبل و بعد از فیلتر کردن رسم کنید.
4. مانند قسمت قبلی فیلتر را به **workspace** منتقل کرده و به کمک دستور **freqz** پاسخ فرکانسی آن را رسم کنید.

\*در هر جا که نیاز شد  $F_s = 16000$  در نظر گرفته شود.

```
sound(y_filtered_Part2,fs);  
  
audiowrite("Nonoise.wav",y_filtered_Part2,fs);  
  
y_part_3 = audioread("Nonoise.wav");
```

<p>Response Type</p> <p><input checked="" type="radio"/> Lowpass</p> <p><input type="radio"/> Highpass</p> <p><input type="radio"/> Bandpass</p> <p><input type="radio"/> Bandstop</p> <p><input type="radio"/> Differentiator</p> <p>Design Method</p> <p><input type="radio"/> IIR Buttenworth</p> <p><input checked="" type="radio"/> FIR Equiripple</p>	<p>Filter Order</p> <p><input checked="" type="radio"/> Specify order: 40</p> <p><input type="radio"/> Minimum order</p> <p>Options</p> <p>Density Factor: 20</p>	<p>Frequency Specifications</p> <p>Units: Hz</p> <p>Fs: 16000</p> <p>Fpass: 2000</p> <p>Fstop: 2500</p>	<p>Magnitude Specifications</p> <p>Enter a weight value</p> <p>Wpass: 1</p> <p>Wstop: 1</p>
---	---	---	---



```

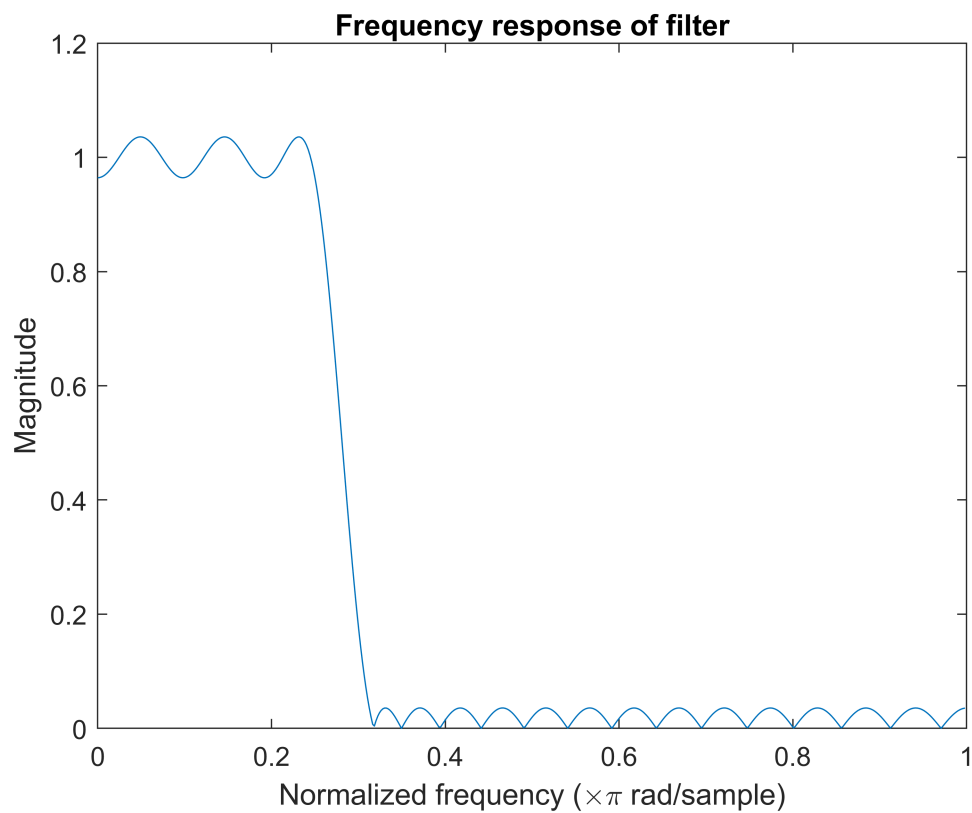
LowPass_Part4_Filter = Part_4_Sim_DSP;
sound(y_part_3,fs);

b = LowPass_Part4_Filter.Numerator;
a = 1; % For FIR filters, the denominator is always 1

[h, w] = freqz(b, a);
plot(w/pi, abs(h));
xlabel('Normalized frequency (\times\pi rad/sample)');
ylabel('Magnitude');
title('Frequency response of filter');

```

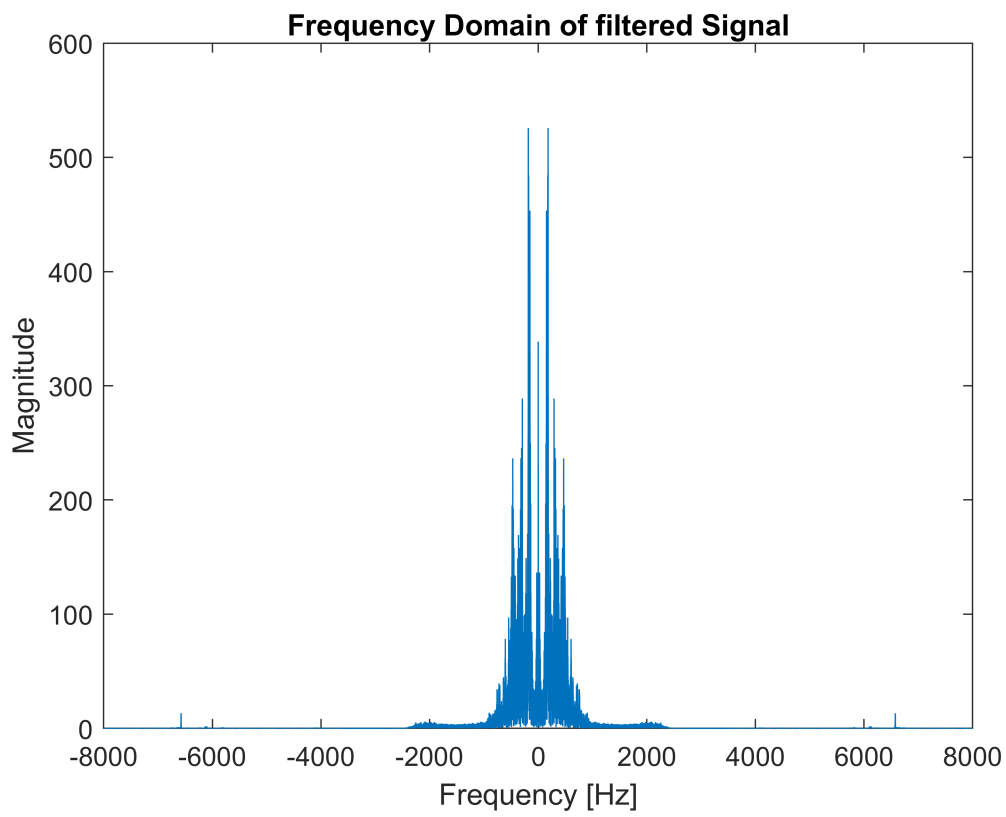




```
% Filter previous Signal using this filter:
```

```
y_filtered_Part4 = filter(b,a,y_part_3);
```

```
figure()  
plot(freq , abs(fftshift(fft(y_filtered_Part4))));  
xlabel('Frequency [Hz]');  
ylabel('Magnitude');  
title('Frequency Domain of filtered Signal');
```



```
sound(y_filtered_Part4,fs);
```