

%↵

```

----- ↵
% Name Lamin Jammeh
% Class: EE417 Summer 2024
% Lesson 10 HW Question 3
% Group: Ron Kalin/ Lamin Jammeh
% Project Description: The verilog coeff are used for creating the filter
%↵
----- ↵
-----

```

```

% Defining the parameters of the synthetic input signal:
% Creating a sine signal

```

```

fs = 2000;      % the sampling frequency = 2KHz
fmax = fs/2;    % Maximum fundamental frequency that can be sampled by fs

```

```

f1 = 50;        % frequency f1 is 50Hz
f2 = 800;       % frequency f2 is 800Hz
f3 = 900;       % frequency f3 is 900Hz

```

```

A1 = 10;        % Amplitude of the sine wave s1
A2 = 5;         % Amplitude of the sine wave s2
A3 = 8;         % Amplitude of the sine wave s3

```

```

L = 200;        % the number of samples

```

```

t = (0:L-1)/fs;      % Generating the time vector for L samples

```

```

s = A1*sin(2*pi*t*f1) + A2*sin(2*pi*t*f2) + A3*sin(2*pi*t*f3);

```

```

%----- ↵
% Defining the filter parameters:
% Use the filter coeff from verilog and create Sampling signal in matlab

```

```

fc = 400;        % the filter cutoff frequency is 400Hz
Wn = fc/fmax;    % Wn = fc/fmax
order = 4;       % the order of the filter

```

```

%coeff = fir1(order,Wn,'low');    % low pass filter
%coeff = fir1(order,Wn,'high');   % high pass filter
coeff = [3, 7, 20, 7, 3];         % same coefficient used in verilog code

```

```

%-----
%           Filtering the signal using 3 approaches
%-----
% 1. Filtering the signal in MATLAB using the 'filter' function:

```

```
output_signal = filter(coeff,1,s);
```

```
% Plotting the input signal & the filtered signals in the time domain
```

```
subplot(2,2,1)
plot(t,s)
title("Original Signal Created with matlab")
subplot(2,2,2)
plot(t,output_signal)
title("Filtered Signal")
xlabel("Time in s")
```

```
%-----
% Creating the fft for the signals:
```

```
f = fs/L*(0:(L/2));
```

```
Y = fft(s);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
```

```
Y_out = fft(output_signal);
P2_out = abs(Y_out/L);
P1_out = P2_out(1:L/2+1);
P1_out(2:end-1) = 2*P1_out(2:end-1);
```

```
subplot(2,2,3)
stem(f,P1,"LineWidth",2)
title("fft Spectrum in the Positive Frequencies")
xlabel("f (Hz)")
ylabel("|fft(input signal)|")
```

```
subplot(2,2,4)
stem(f,P1_out,"LineWidth",2)
title("fft Spectrum in the Positive Frequencies")
xlabel("f (Hz)")
ylabel("|fft(output filtered signal)|")
```

```
%-----
```