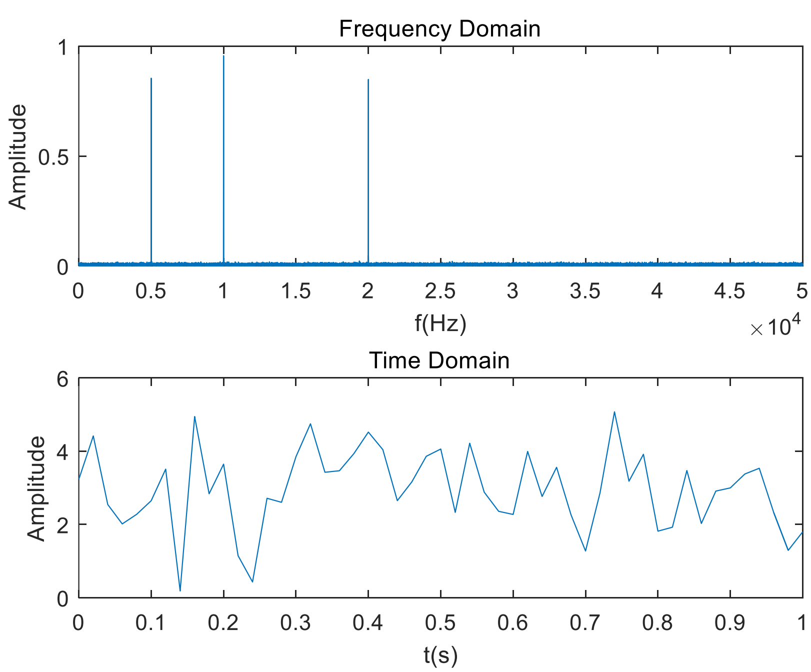
Q1 The spectrum of x(t) based on DFT analysis

MATLAB CODE

```
FS=100000;
T=1/FS; N=FS; n=0:N-1;
f1=5000; f2=10000; f3=20000;
xn=cos(2*pi*f1*n*T)+cos(2*pi*f2*n*T)+cos(2*pi*f3*n*T)+randn(size(n));
NFFT=2^nextpow2(N);
y=fft(xn,NFFT)/N;
f=FS/2*linspace(0,1,NFFT/2+1);
figure(1);
subplot(2,1,1);
plot(f,2*abs(y(1:NFFT/2+1)));
title('Frequency Domain');
xlabel('f(Hz)');
ylabel('Amplitude');

n1=0:(2000/N):1;
xn1=cos(2*pi*f1*n1)+cos(2*pi*f2*n1)+cos(2*pi*f3*n1)+randn(size(n1));
subplot(2,1,2);
plot(n1,xn1);
title('Time Domain');
xlabel('t(s)');
ylabel('Amplitude');
```



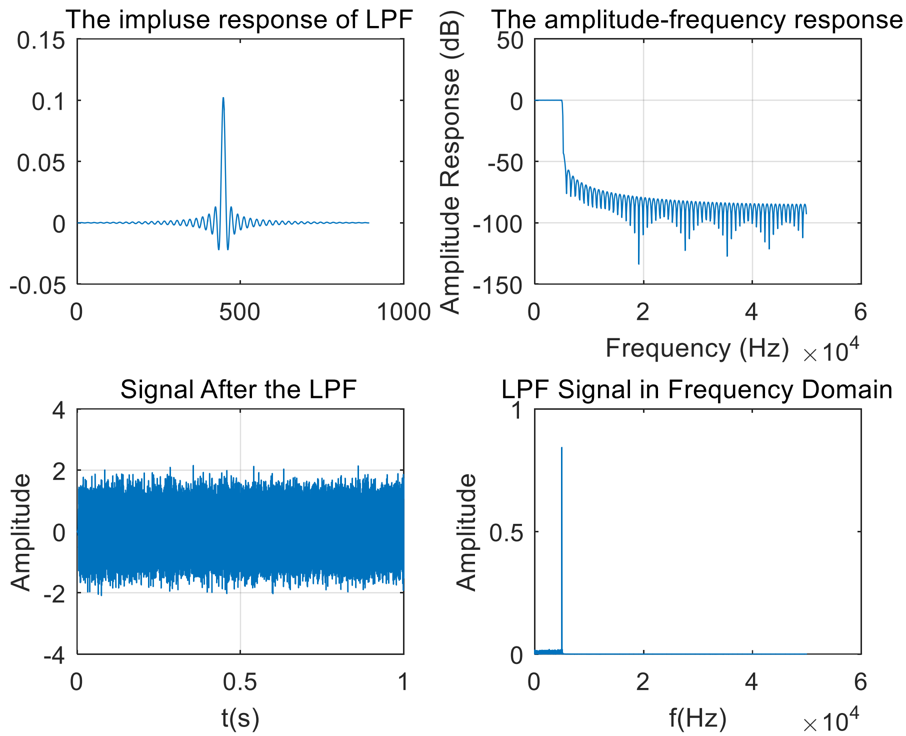
Q2 LP filter to extract the signal with frequency f1

MATLAB CODE

```
fsamp=FS;
fcuts1=[5000 5250];
mags1=[1 0];
devs1=[0.01 0.01];
[nlp1,Wn1,beta1,ftype1] = kaiserord(fcuts1,mags1,devs1,fsamp);
hn1 = fir1(nlp1,Wn1,ftype1,kaiser(nlp1+1,beta1),'noscale');
figure(2);
subplot(2,2,1);
plot(hn1);
title('The impluse response of LPF');
[H1,f1]=freqz(hn1,1,512,FS);
mag1=20*log10(abs(H1));
subplot(2,2,2);
plot(f1,mag1),grid on;
title('The amplitude-frequency response');
xlabel('Frequency (Hz)');
ylabel('Amplitude Response (dB)');

lp1=filter(hn1,1,xn);
subplot(2,2,3);
plot(n*T,lp1),grid on;
title('Signal After the LPF');
xlabel('t(s)')
ylabel('Amplitude');

ylp1=fft(lp1,NFFT)/N;
ff=FS/2*linspace(0,1,NFFT/2+1);
subplot(2,2,4);
plot(ff,2*abs(ylp1(1:NFFT/2+1)));
title('LPF Signal in Frequency Domain');
xlabel('f(Hz)');
ylabel('Amplitude');
```



Q3 BP filter to extract the signal with frequency f2

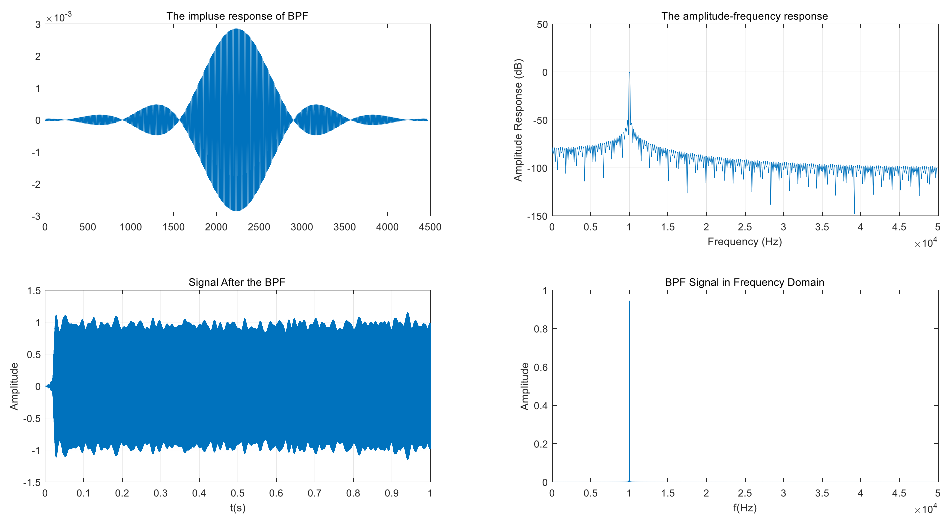
MATLAB CODE

```
fcuts3=[9900 9950 10050 10100];
mags3=[0 1 0];
devs3=[0.01 0.05 0.01];
[nlp3,Wn3,beta3,ftype3] = kaiserord(fcuts3,mags3,devs3,fsamp);
hn3 = fir1(nlp3,Wn3,ftype3,kaiser(nlp3+1,beta3),'noscale');
figure(3);
subplot(2,2,1);
plot(hn3);
title('The impluse response of BPF');
[H3,f3]=freqz(hn3,1,512,FS);
mag3=20*log10(abs(H3));
subplot(2,2,2);
plot(f3,mag3),grid on;
title('The amplitude-frequency response');
xlabel('Frequency (Hz)');
ylabel('Amplitude Response (dB)');

bp3=filter(hn3,1,xn);
subplot(2,2,3);
```

```
plot(n*T,bp3),grid on;
title('Signal After the BPF');
xlabel('t(s)')
ylabel('Amplitude');

ybp3=fft(bp3,NFFT)/N;
subplot(2,2,4);
plot(ff,2*abs(ybp3(1:NFFT/2+1)));
title('BPF Signal in Frequency Domain');
xlabel('f(Hz)');
ylabel('Amplitude');
```



Q4 HP filter to extract the signal with frequency f3

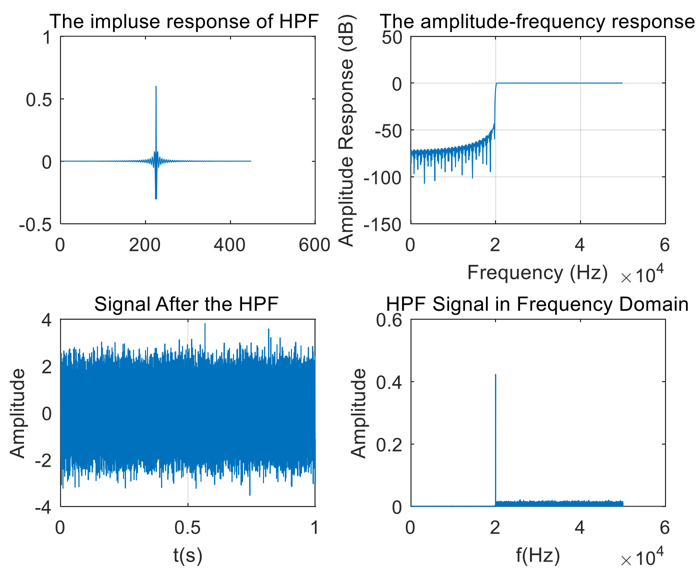
MATLAB CODE

```
fcuts5=[19750 20250];
mags5=[0 1];
devs5=[0.01 0.01];
[nlp5,Wn5,beta5,ftype5] = kaiserord(fcuts5,mags5,devs5,fsamp);
hn5 = fir1(nlp5,Wn5,ftype5,kaiser(nlp5+1,beta5),'noscale');
figure(4);
subplot(2,2,1);
plot(hn5);
title('The impluse response of HPF ');
[H5,f5]=freqz(hn5,1,512,FS);
mag5=20*log10(abs(H5));
subplot(2,2,2);
plot(f5,mag5),grid on;
```

```
title('The amplitude-frequency response');
xlabel('Frequency (Hz)');
ylabel('Amplitude Response (dB)');

hp5=filter(hn5,1,xn);
subplot(2,2,3);
plot(n*T,hp5),grid on;
title('Signal After the HPF');
xlabel('t(s)')
ylabel('Amplitude');

yhp5=fft(hp5,NFFT)/N;
subplot(2,2,4);
plot(ff,2*abs(yhp5(1:NFFT/2+1)));
title('HPF Signal in Frequency Domain');
xlabel('f(Hz)');
ylabel('Amplitude');
```



Annotation:

(1) The filter is designed using Kaiser Window. It can meet the requirements in the assumptions. The "kaiserord" function is used to estimate the important parameters of the Kaiser window filter. The "fir1" function is then used to implement the method of FIR filter design. By default, the filter is normalized so that the magnitude response of the filter at the center frequency of the passband is 0 dB. The "freqz" function is used to calculate the frequency response based on the current filter coefficients. The last step is to use the "filter" function to filter the input signal. (2) The Kaiser window filter is obtained by using the [n, Wn, beta, ftype] = kaiserord (f, a, dev) function to estimate the filter order. Here the function kaiserord (F, A, DEV) or kaiserord (F, A, deviation, FS): F is the corresponding frequency and FS is the sampling frequency. When F is expressed in digital

frequency, FS does not need to write. a = [1 0] is the amplitude vector of each frequency band specified by f, generally only 0 and 1; devs = specifies the maximum output error or deviation between the frequency response of the output filter of each band and its expected amplitude, equal to a. (3) In the designs of the band pass filter and the high pass filter, Kaiser window gives satisfying results.