DSP Lab Experiments Using MATLAB

1. BASIC_OPERATIONS_ON_AUDIO_SIGNALS
2. CONVOLUTION OF TWO DISCRETE SIGNALS
3. DISCRETE FOURIER TRANSFORM
4. POWER SPECTRAL DENSITY
5. ECG_SIGNAL_READ AND PLOT

1.BASIC_OPERATIONS_ON_AUDIO_SIGNALS

Aim:-To write a code for Basic operations of audio signals using MATLAB

Tool:-MATLAB&SIMULINK

Version:-R2018b

Procedure:-

1. Open MATLAB file
2. Give a correct MATLAB path
2. Write a code for basic_operations on audio signals in "Editor"
3. First write clear all and close all because sometimes before program values are stored ,those will be execute.
4. Download audio songs,those are in same MATLAB path only
5. Using the "audioread" function we have to read the all songs
6. In the same way read all signals (which you want to operate)
7. Important thing is that if you want to add or subtract the "matrix length" must be same
8. using "plot" function you can see the signal individually and if you want to see all signals at a time use "subplot" function
9. save the file only lowercase letters only and it doesn't take uppercase and space also.

Code:-

```
clear;
%%
[g,f3]=audioread('guitar.wav');
sound(g,f3);
%%clear sound
gg=size(g,1);
%figure(1);
%plot(g);
%%
[b,f1]=audioread('bass.wav');
%sound(b,f1);
%%clear sound
```

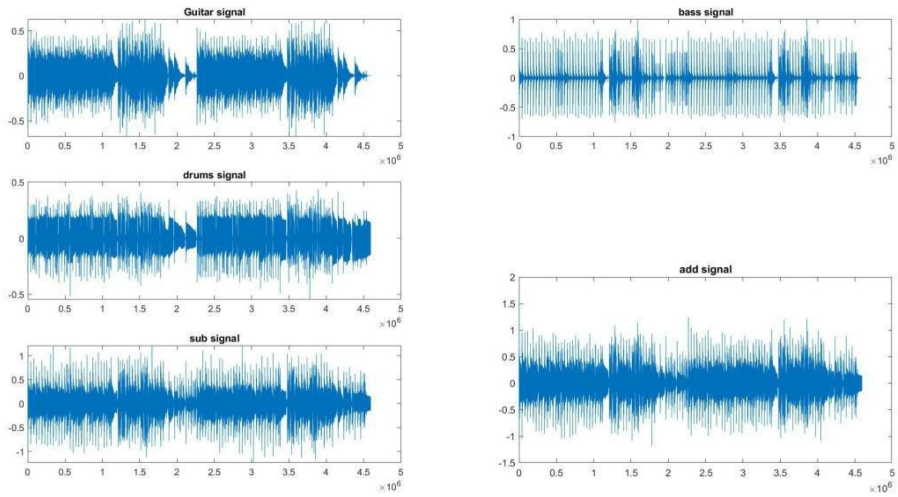
```
bb=size(b,1);
b=b(1:gg);
%figure(2);
%plot(b);
%%
[d,f2]=audioread('drums.wav');
%sound(d,f2);
%%clear sound
dd=size(d,1);
d=d(1:gg);
%figure(3);
%plot(d);
%%
compound=b+d+g;
compound2=g-b-d;
%sound(compound,44100);
%figure(4);
%plot(compound);
%%
subplot(3,2,1);plot(g);title('Guitar signal');
subplot(3,2,2);plot(d);title('bass signal');
subplot(3,2,3);plot(b);title('drums signal');
subplot(3,2,4);plot(compound);title('add signal');
subplot(3,2,5);plot(compound2);title('sub signal');
subplot(3,2,6);plot(d_ramp);title('Ramp sig signal');
%%
dnew=d(1:100000);
gnew=g(1:100000);
bnew=b(1:100000);
reapt=[dnew;gnew;bnew];
%sound(reapt,f1);
%%
dnew=d(1:100000);
gnew=g(1:100000);
reapt=[dnew,gnew];
%sound(reapt,f1);
%%
ramp=0:(1/size(d,1)):1;
ramp=ramp(1:size(d,1));
ramp=(ramp)';
d_ramp=d.*ramp;
%plot(d_ramp)
```

Simulated output:-

Name	values

b,	4593084x1 double
bb,	4646687
compound,	4593084x1 double
compound2,	4593084x1 double
d,	4593084x1 double
dd,	4725975
f1,	44100
f2,	44100
f3,	44100
g,	4593084x1 double
gg	4593084

Output figure:-



Conclusion:-

Using MATLAB the output of the Basic_operations_on_signals are executed.

2.CONVOLUTION OF FREQUENCY SIGNALS

Aim:-To write a code for convolution of two discrete signals using "Direct method" and direct "conv" function in MATLAB

Tool:-MATLAB&SIMULINK

Version:-R2018b

Procedure:-

1. Open MATLAB file
2. Give a correct MATLAB path
3. Write a code for convolution in "Editor Window"
4. first initialize the sample frequency(fs)and time(t)
5. Write an expression for 20hz sine wave sig and plot it.
6. In the same way write an expression for 1hz sine wave sig and plot it.
7. Now add the both signals(`sig_20hz+sig_1hz`)
8. Design a filter using filter design analysis(fda)tool with 1 to 6 bandwidth and order 39
9. Now using 'conv' function we can directly convolve the signals
10. And also find convolution using direct method also
11. If we want to see this plots in single figure use subplot function

Code:-

```
fs=100;
%% 20hz sine wave
f1=20;
ts=1/fs;
t=0:ts:3;
sig_20hz=sin(2*pi*f1*t);
% figure(1);
% plot(sig_20hz);
%% 1hz sinewave
f2=1;
sig_1hz=sin(2*pi*f2*t);
% figure(2);
```

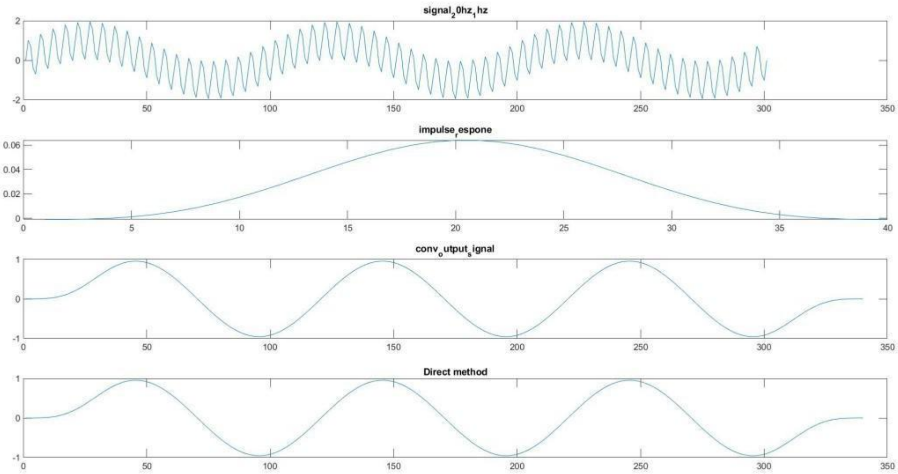
```
% plot(sig_1hz);
%% 20hz+1hz sine wave
 signal_20hz_1hz=sig_20hz+sig_1hz;
% figure(3);
% plot(sig_20hz+sig_1hz);
%% design  a filter using filter design analysis(FDA)tool
%fdatool
%pause
%%
 impulse_response=impulse;
%  plot(impulse_response);
%% convolution
output_sig=conv(signal_20hz_1hz,impulse_response);
%  plot(sig_20hz+sig_1hz);
%  figure(4);

subplot(4,1,1);plot(sig_20hz+sig_1hz);title('signal_2
0hz_1hz');

subplot(4,1,2);plot(impulse_response);title('impulse_
response');

subplot(4,1,3);plot(output_sig);title('conv_output_si
gnal');
%% direct method of conv
m=length(signal_20hz_1hz);
m=length(impluse_response);
l=m+n-1;
y=zeros(1,1);
for i=1:m;
    x1=signal_20hz_1hz(i)*impulse_response;
    x2=[zeros(1,i-1),x1,zeros(1,m-i)];
    y=y+x2;
end
% plot(y);
subplot(4,1,4);plot(y);title('Direct method');
```

Simulated output:-

NAME	value
f1	20
f2	1
Fs	100

Impulse	1*40 double
M	301
output_sig	1*340 double
sig_1hz	1*301 double
sig_20hz	1*301 double
signal_20hz_1hz	1*301 double
T	1*301 double
Ts	0.0100

Output figure:-



Conclusion:-

Using MATLAB the convolution of two frequency signals using “Direct method” and direct “conv” function are executed

3. DISCRETE FOURIER TRANSFORM

Aim:-To write a code for Discrete Fourier transform using direct method and FFT command and comparision of two methods performance

Tool:-MATLAB&SIMULINK

Version:-R2018b

Procedure:-

1. open MATLAB file
2. Give a correct MATLAB path
3. Write a code for ‘DTFT’ in “Editor window”
4. First consider the sample rate(srate)
5. Signal length(l) depends upon the srate
6. Signal length is equal to csw(complex sine wave)length
7. Then csw length is depends on FourTime
8. FourTime depends on points(srate)
9. Now dot product on csw and signal,then it develop a new signal
10. Then new signal length will be “summable”
11. Then create only one single signal
12. That is the Fourier Transform signal
13. Here we use “tic toc” to calculate the speed of the operation
14. Then completion of the code save and Run it.we can observe the single signal

Code:-

```
clc
clear
close all
%%  Creating signal
srate =1000;
time=0:1/srate:2;
pnts=length(time);
signal=2.5*sin(2*pi*4*time)+1.5*sin(2*pi*6.5*time);
%Fourier Transeform
```

```matlab
tic;
fourTime=(0:pnts-1)/pnts;
fCoefs=zeros(size(signal));
for fi=1:pnts
    %create complex sine wave
    csw=exp(-1i*2*pi*(fi-1)*fourTime);
    %compute dot product between sine wave and signal
    fCoefs(fi)=sum(signal.*csw)/pnts;
end
t(1)=toc
%amplitudes
ampls=2*abs(fCoefs);
%compute frequencie vector
hz=linspace(0,srate/2,floor(pnts/2)+1);

figure
subplot(311)
plot(time,signal,'k')
xlabel('Time(s)'),ylabel('Amplitude')
title('Time domain')

subplot(312)
stem(hz,ampls(1:length(hz)),'ks-')
set(gca,'xlim',[0  8],'ylim',[-.01 3])
xlabel('Frequency(Hz)'),ylabel('Amplitude(a.u.)')
title('Frequency domain')
%% Fourier Tranceform using fft command
tic;
fcoefsF=fft(signal)/pnts;
t(2)=toc;
amplsF=2*abs(fcoefsF);
subplot(313)
stem(hz,amplsF(1:length(hz)),'or')
set(gca,'xlim',[0 8],'ylim',[-.01 3])
xlabel('Frequency Transform using FFT')
title('Fourier Trance form using FFT')
%% the inverse fourier transform
% initialize time-domain reconstuction
reconSignal=zeros(size(signal));
tic;
for fi=1:pnts
    %create coefficient-modulated complex sine wave
    csw1=fCoefs(fi)*exp(1i*2*pi*(fi-1)*fourTime);
    %sum them together
    reconSignal=reconSignal+csw1;
end
%divide by N
```

```
reconSignal=reconSignal/pnts;
t(3)=toc;
%figure(2),clf
%subplot(2,1,1)
%plot(time,signal)
%legend({'original'})
%subplot(2,1,2)
%plot(time,real(reconSignal),'r-')
%legend
%hold on
%% Inverse legend({'Original';'Reconstructed'})fourier
tic;
reconTS=real(ifft(fCoefs))*pnts;
t(4)=toc;
subplot(3,1,1),hold on
plot(time(1:3:end),reconTS(1:3:end),'ro');
legend({'Original';'Reconstructed'})
zoom on
%% time computations plot
figure,bar(t);
title('Computing time values')
%plot(time,signal)
```

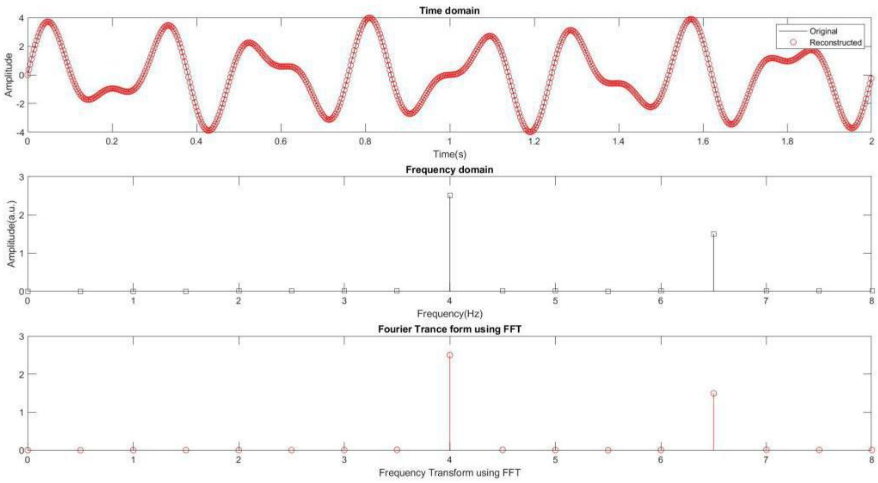
Simulated output:-

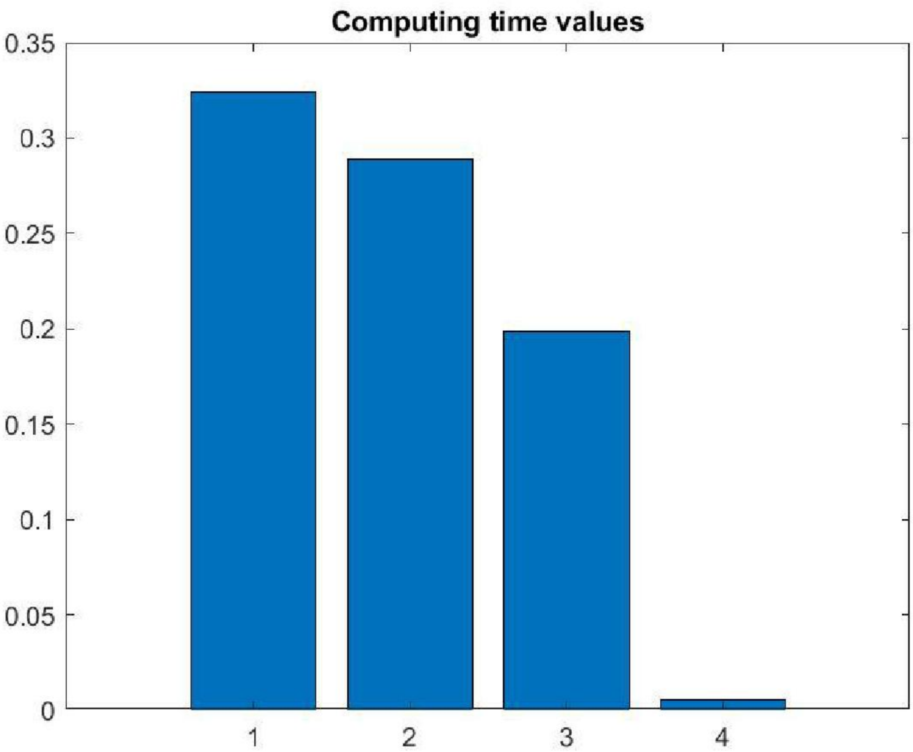
NAME	VALUES
Ampls	1x2001 double
amplsF	1x2001 double
Csw	1x2001 complex
csw1	1x2001 complex
fCoef	1x2001 complex
fcoefsF	1x2001 complex
Fi	2001
fourTime	1x2001 double
Hz	1x2001 double
Pnts	2001
reconSignal	1x2001 complex
reconTS	1x2001 double
Signal	1x2001 double
Srate	1000
T	

[0.323965151558465,0.288773243209917,0.198709844782867,0.005325228938764]

Time 1x2001 double

Output figure:-





Conclusion:-

Using MATLAB we have observe the output of the Discrete Fourier transform for given signal.

4.POWER SPECTRAL DENSITY

AIM:- To generate power spectral density by using pwelch method in MATLAB

Tool:- MATLAB&SIMULINK

Version:-R2018b

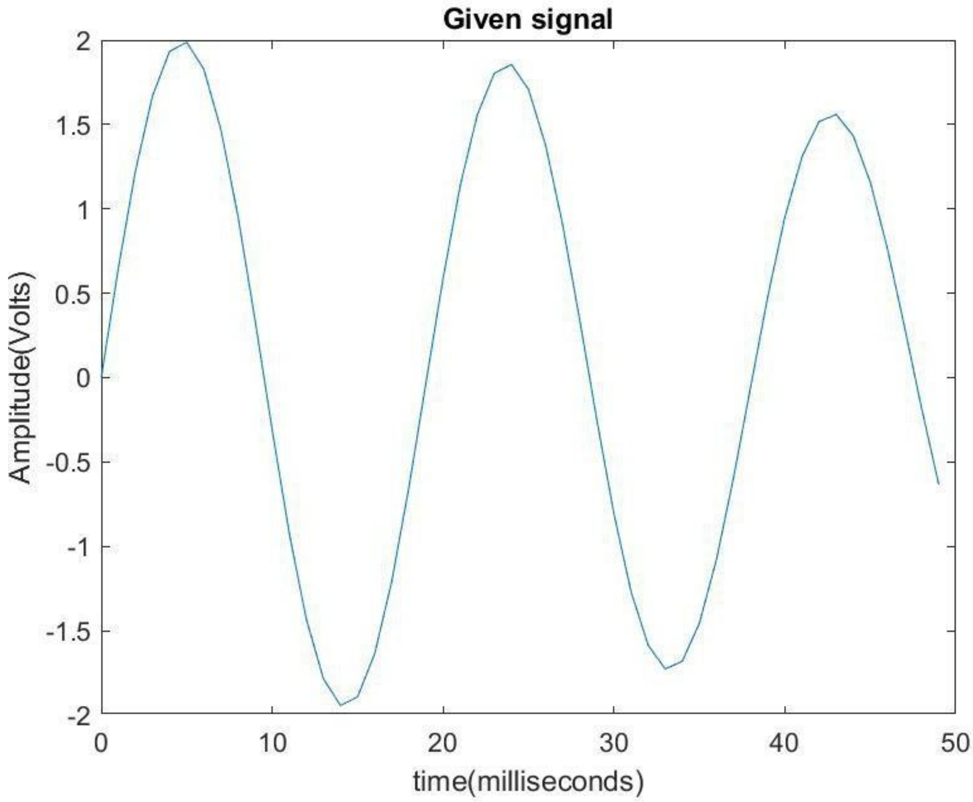
Procedure:-

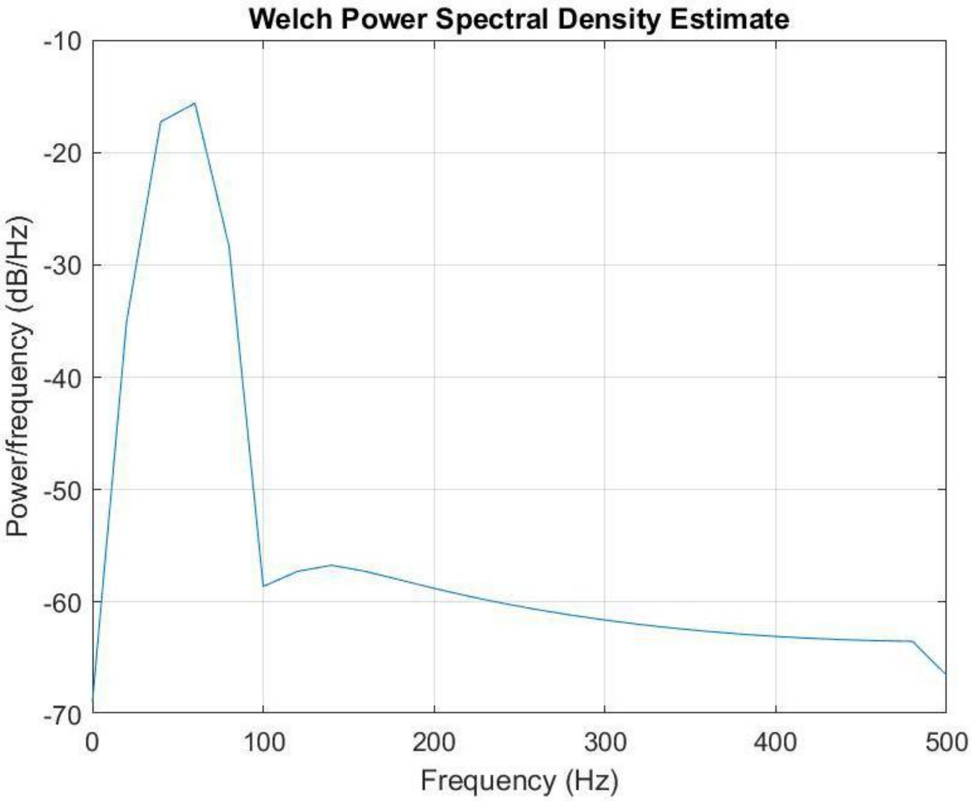
1. Open MATLAB file.
2. Give a correct path
3. Type “edit”in the MATLAB prompt ‘>>’that Appear in command window
4. Write the program in the “Editor window” and save it in m-file.
5. Run the program.
6. Enter the input in the command window
7. The result is displayed in the command window
8. The graphical output is displayed in the figure window

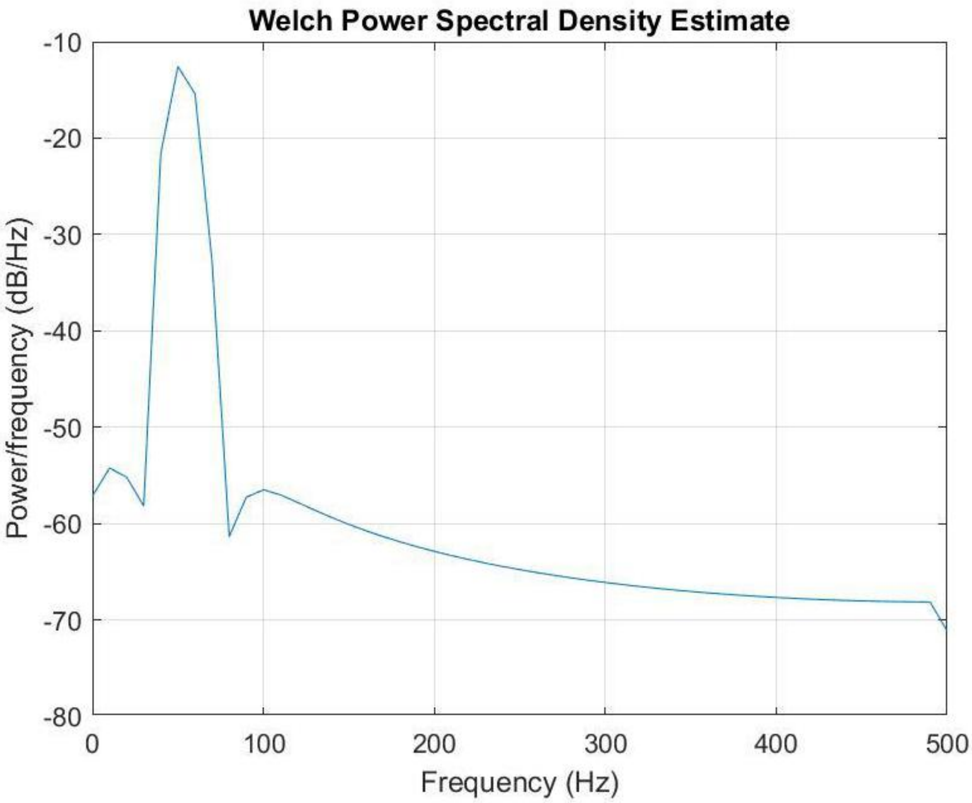
Code:-

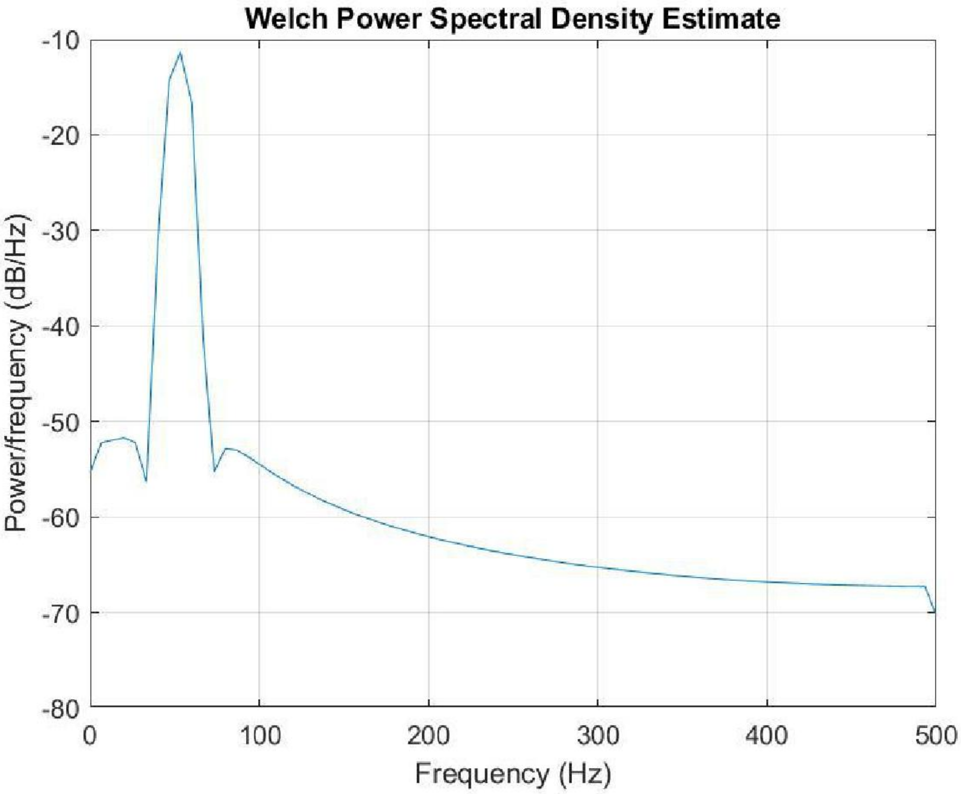
```
clc
clear
close all
% Generating a signal
fs=1000;
t=0:1/fs:0.6;
x=sin(2*pi*50*t)+sin(2*pi*55*t);
figure,
plot(fs*t(1:50),x(1:50))
title('Given signal')
xlabel('time(milliseconds)');
ylabel('Amplitude(Volts)');
%% Pwelch method(default window is Hamming window)
for len=50:50:600
    figure;pwelch(x,len,[],len,fs);
end
```

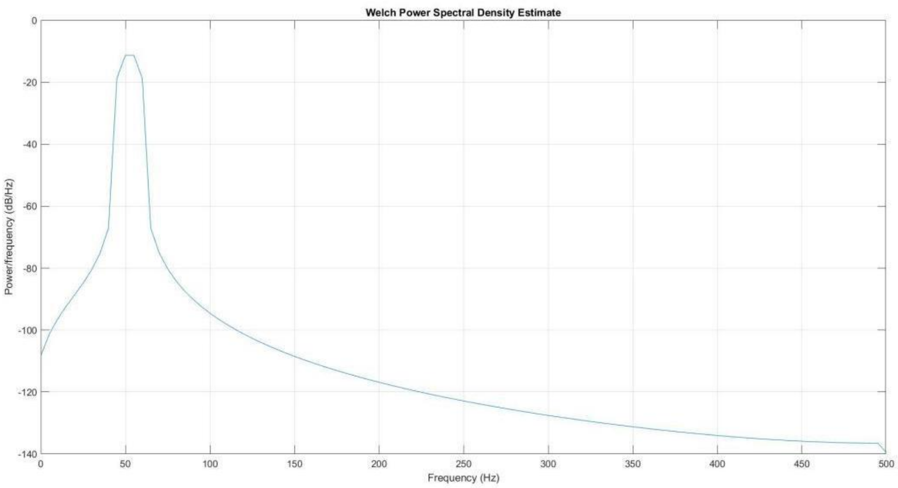
Simulated output:-

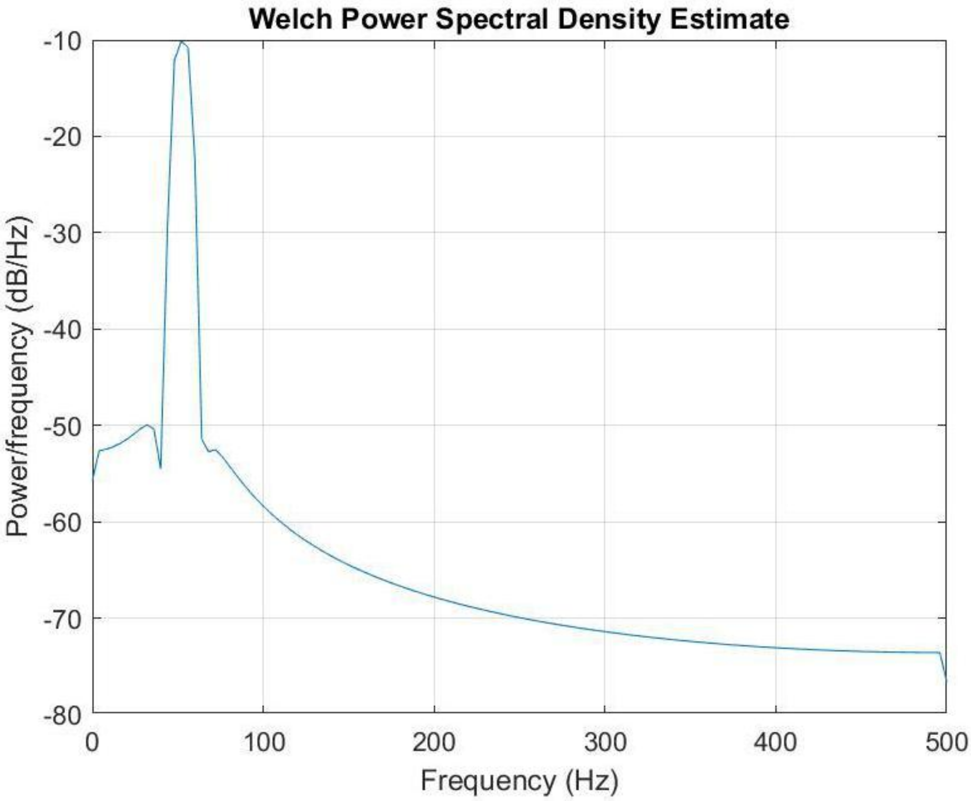


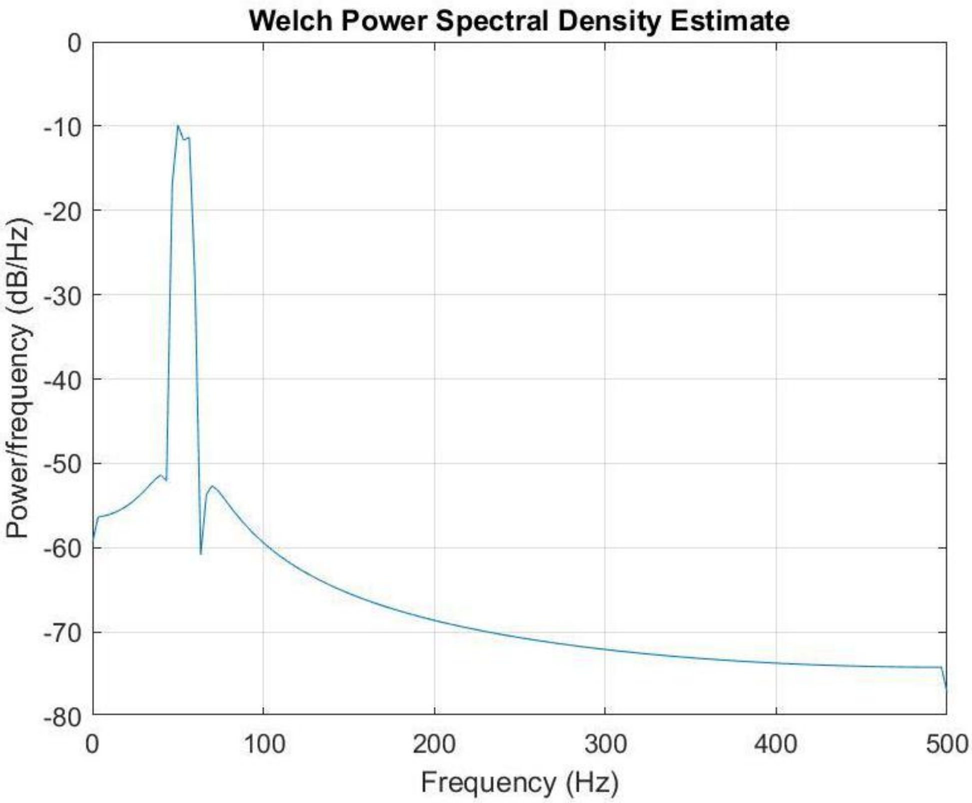

Welch Power Spectral Density Estimate
-10
-20
-30
-40
-50
-60
-70
Power/frequency (dB/Hz)
0
100
200
300
400
500
Frequency (Hz)

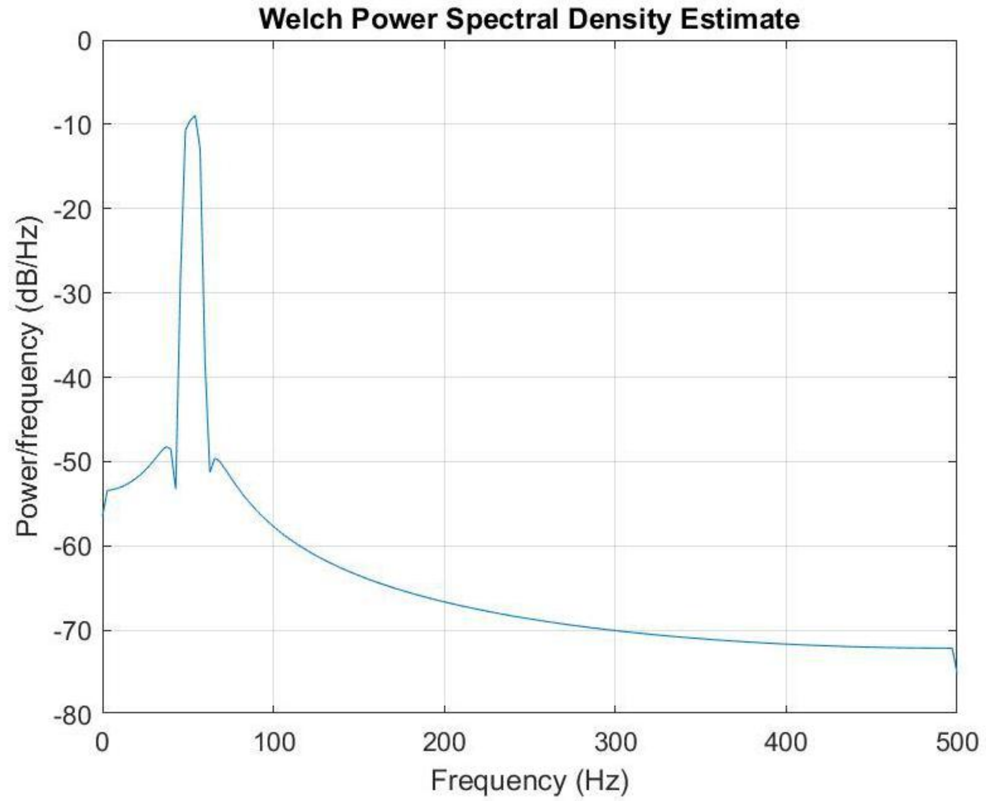

Welch Power Spectral Density Estimate
-10
-20
-30
-40
-50
-60
-70
-80
Power/frequency (dB/Hz)
0
100
200
300
400
500
Frequency (Hz)

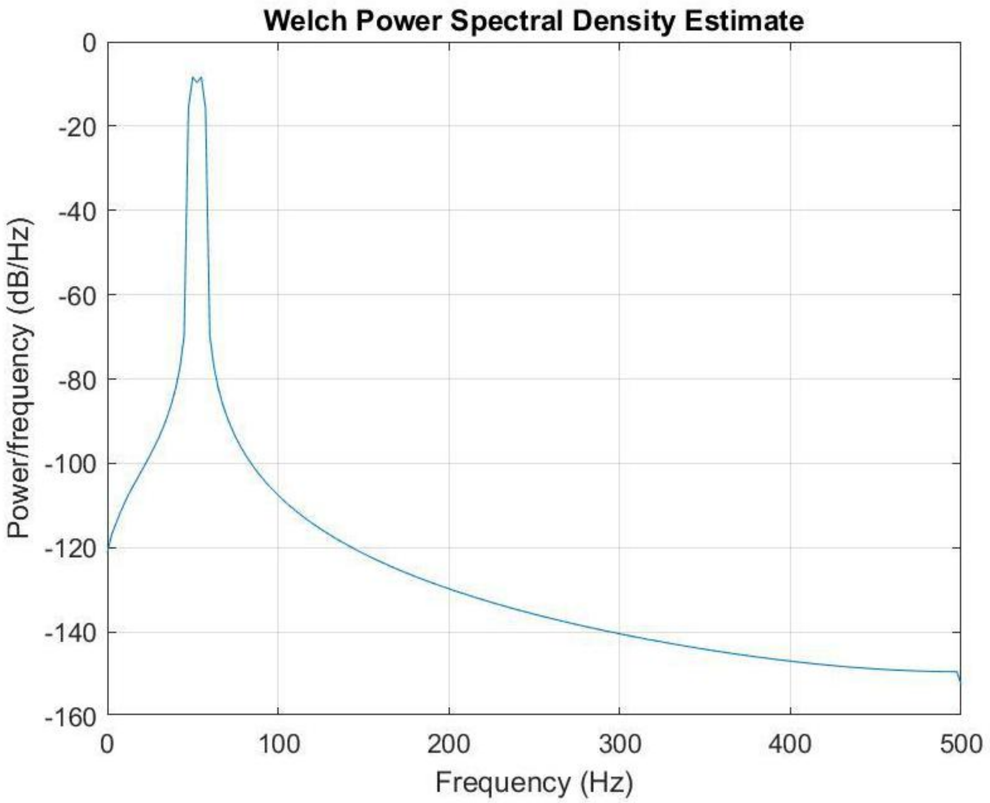

Welch Power Spectral Density Estimate
-10
-20
-30
-40
-50
-60
-70
-80
Power/frequency (dB/Hz)
0
100
200
300
400
500
Frequency (Hz)

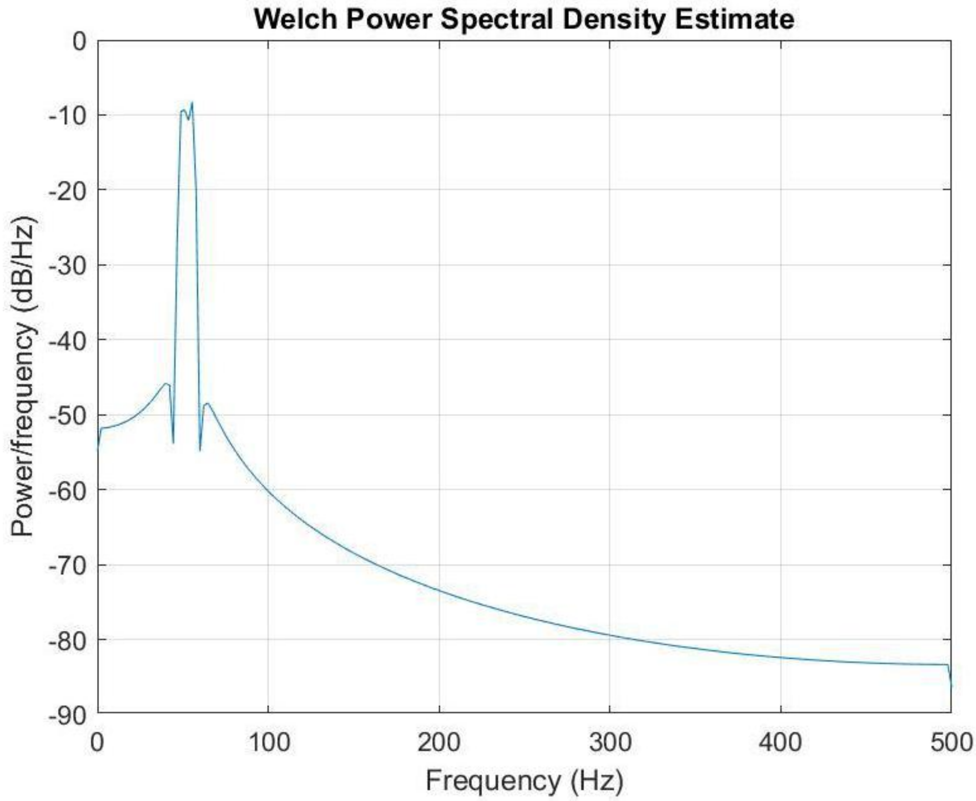

Welch Power Spectral Density Estimate
0
-20
-40
-60
-80
-100
-120
-140
Power/frequency (dB/Hz)
0
50
100
150
200
250
300
350
400
450
500
Frequency (Hz)

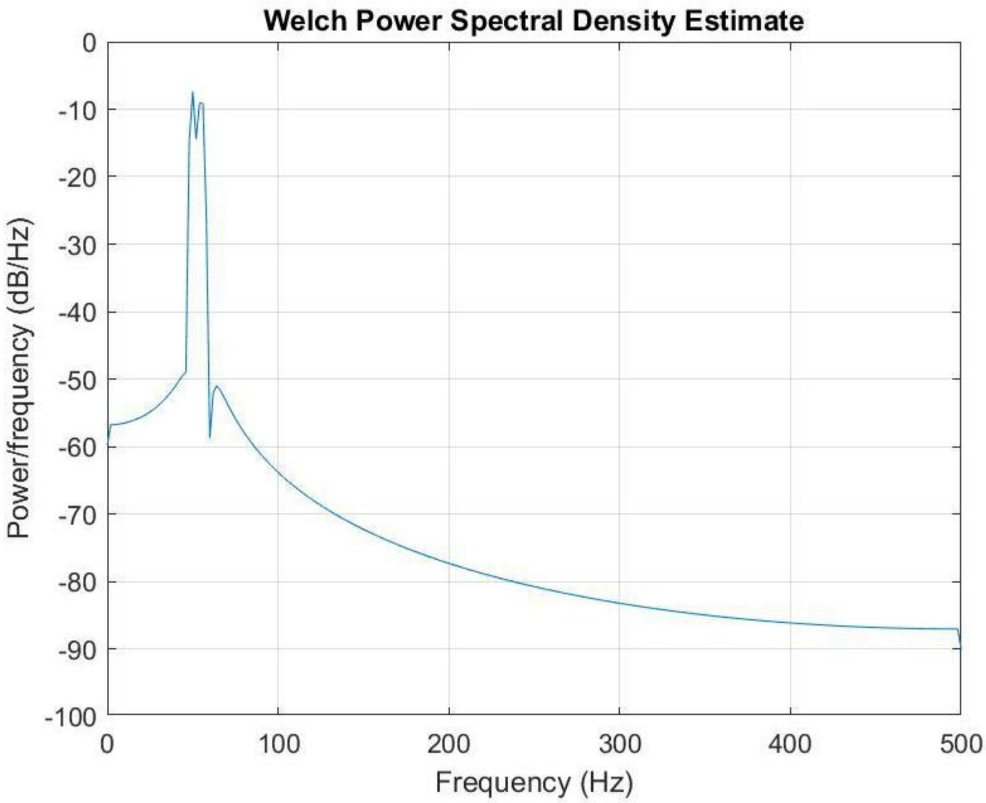

Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
-10
-20
-30
-40
-50
-60
-70
-80
0
100
200
300
400
500
Frequency (Hz)

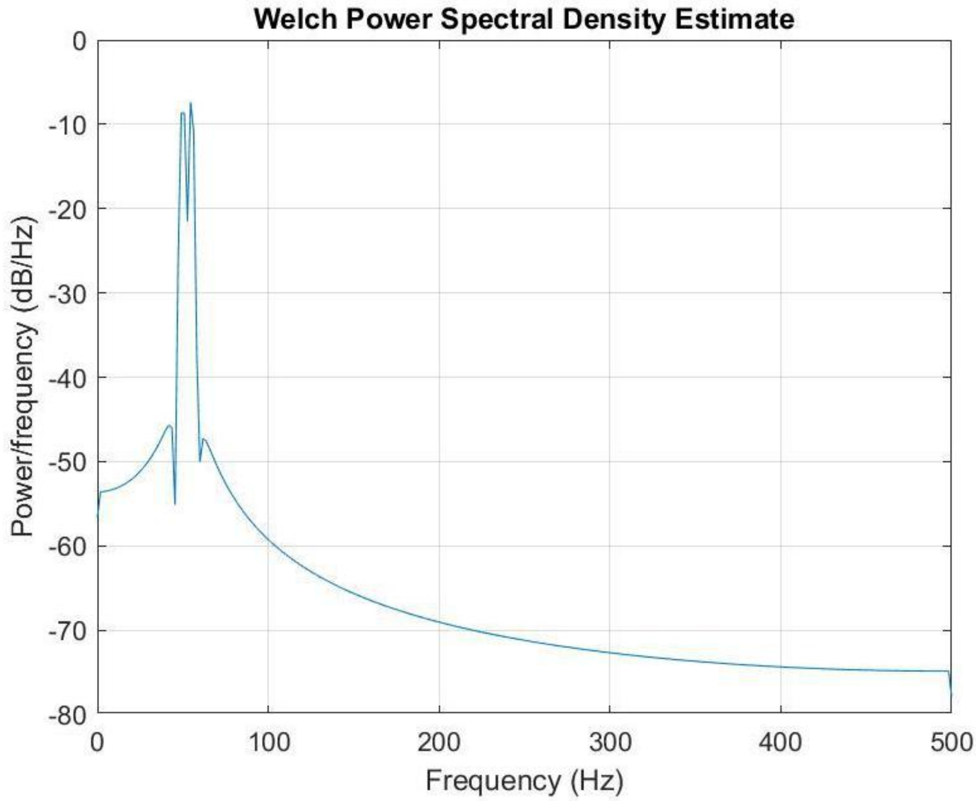

Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
0
-10
-20
-30
-40
-50
-60
-70
-80
0
100
200
300
400
500
Frequency (Hz)

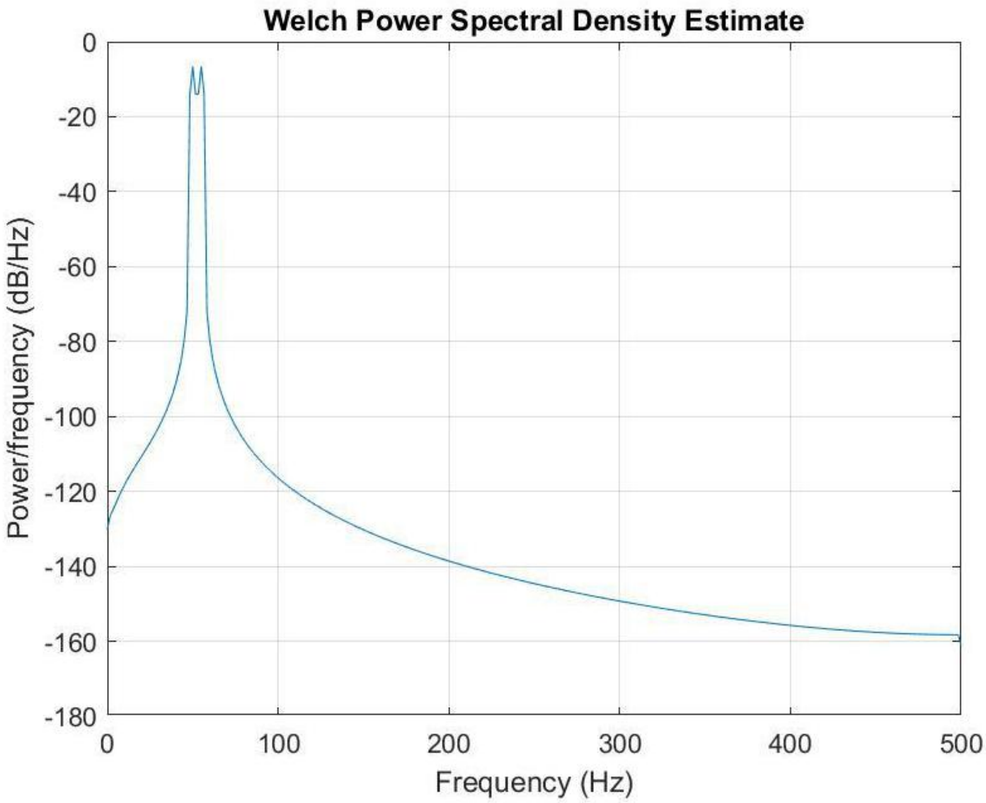

Welch Power Spectral Density Estimate
0
-10
-20
-30
-40
-50
-60
-70
-80
Power/frequency (dB/Hz)
0
100
200
300
400
500
Frequency (Hz)

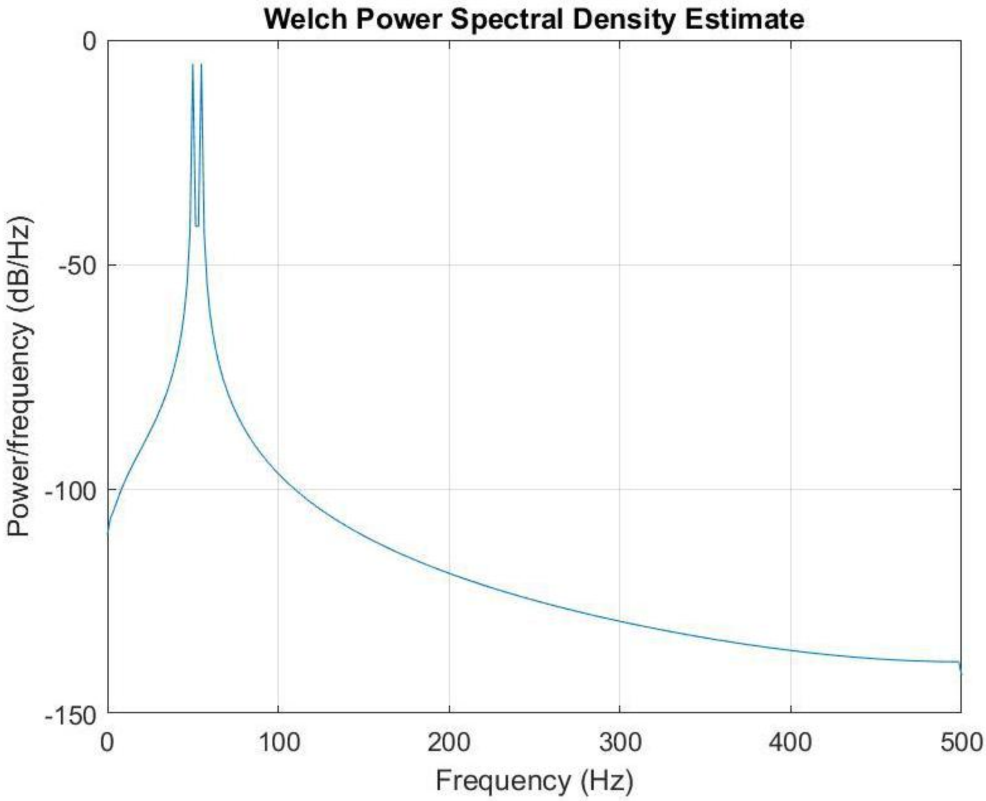

Welch Power Spectral Density Estimate
0
-20
-40
-60
-80
-100
-120
-140
-160
Power/frequency (dB/Hz)
0
100
200
300
400
500
Frequency (Hz)

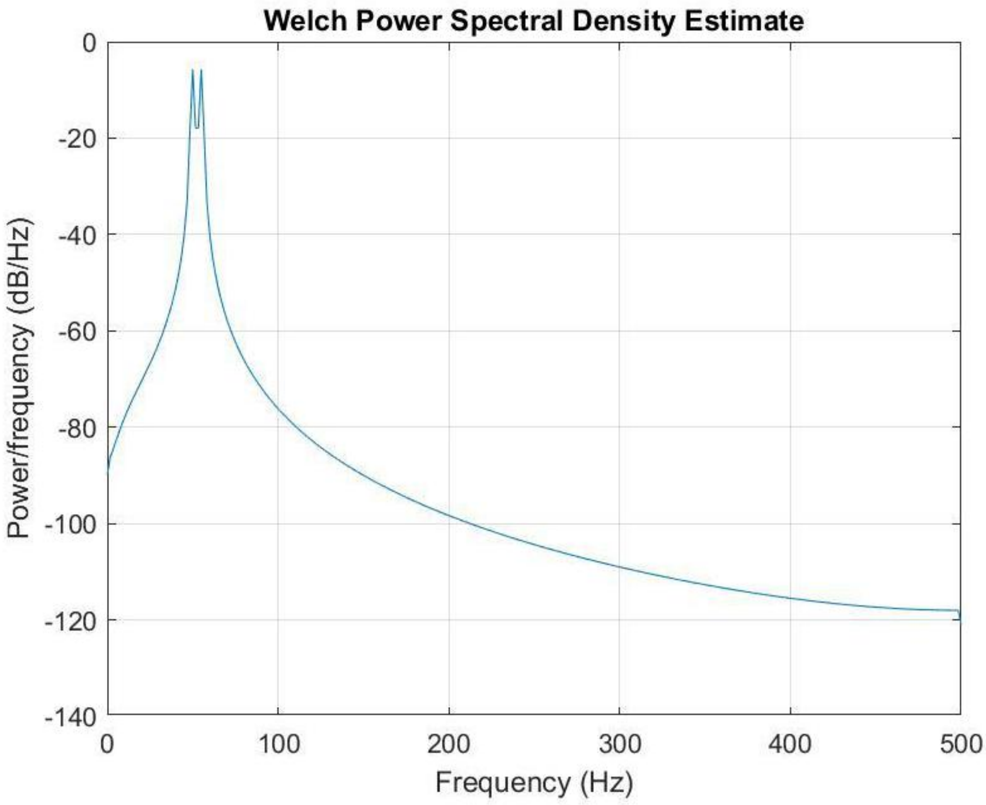

Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
0
100
200
300
400
500
Frequency (Hz)

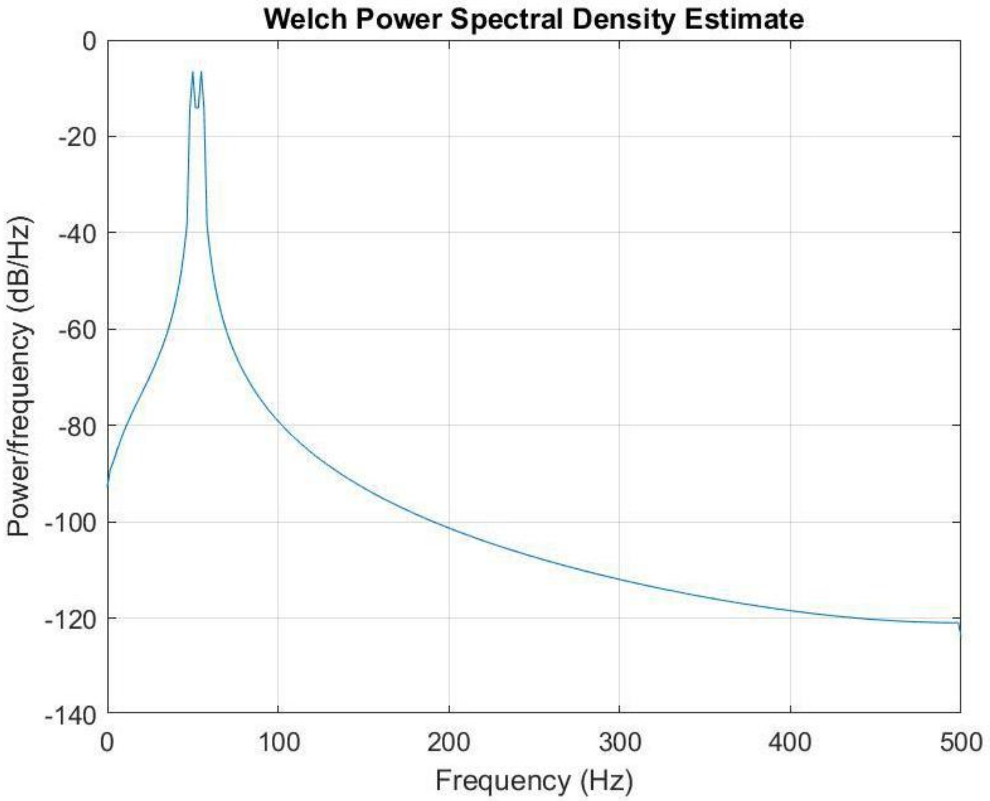

Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
0
100
200
300
400
500
Frequency (Hz)


Welch Power Spectral Density Estimate
0
-10
-20
-30
-40
-50
-60
-70
-80
Power/frequency (dB/Hz)
0
100
200
300
400
500
Frequency (Hz)

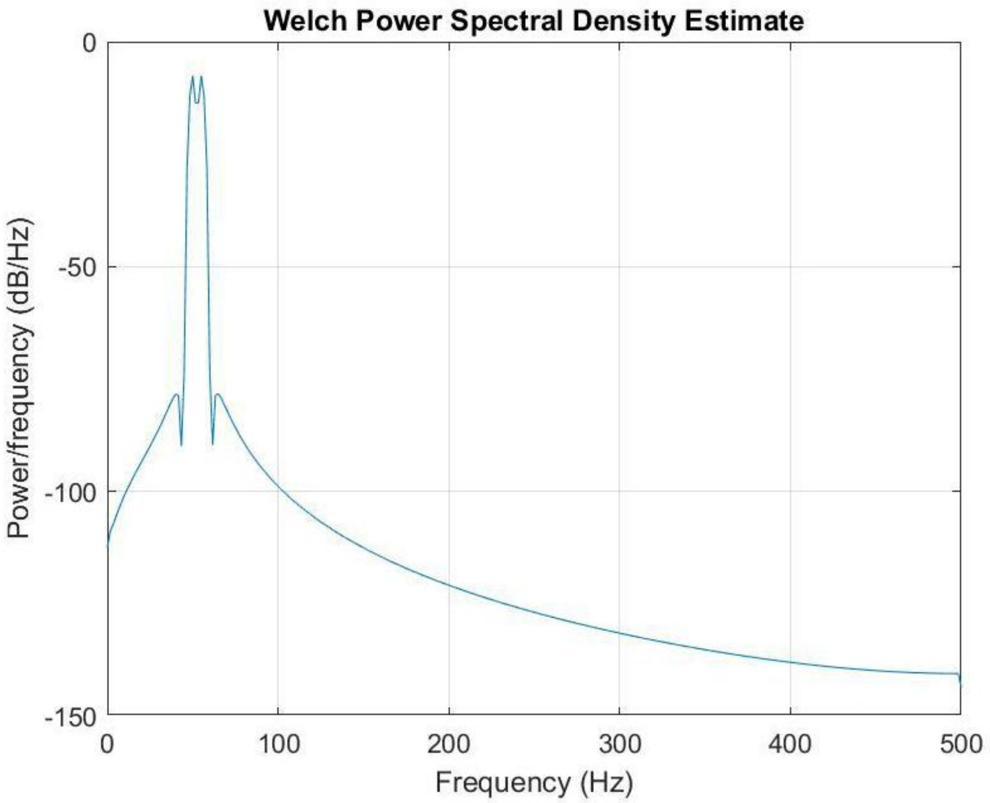

Welch Power Spectral Density Estimate
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
Power/frequency (dB/Hz)
0
100
200
300
400
500
Frequency (Hz)


Welch Power Spectral Density Estimate
0
-50
-100
-150
Power/frequency (dB/Hz)
0
100
200
300
400
500
Frequency (Hz)

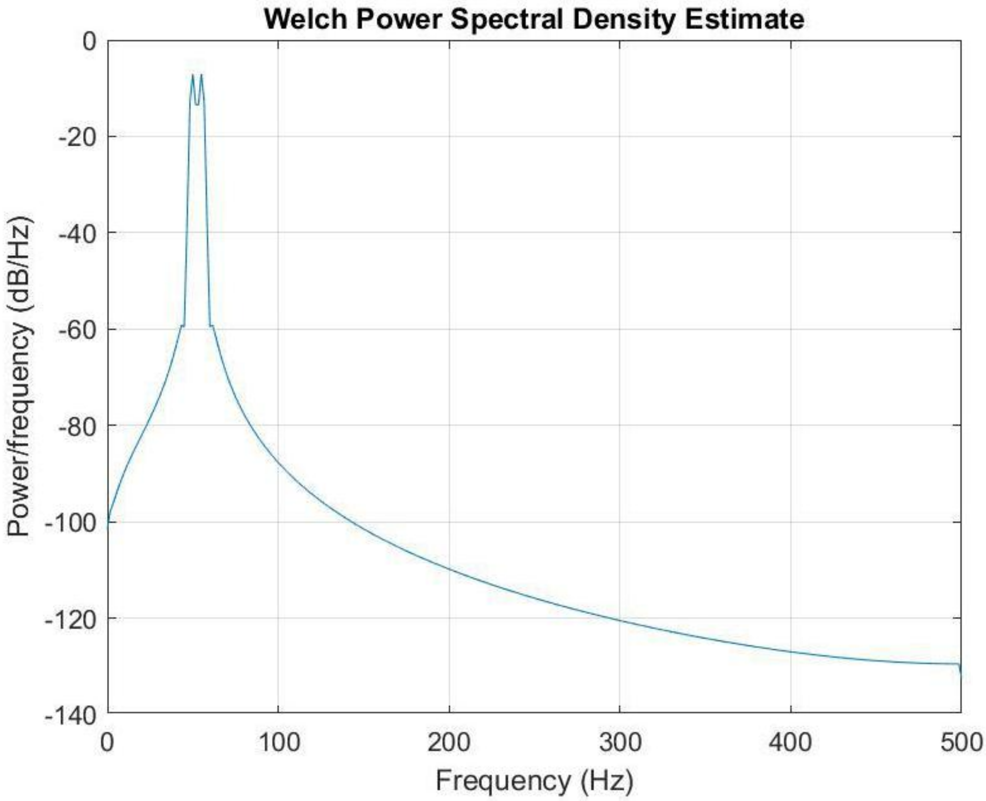

Welch Power Spectral Density Estimate
0
-20
-40
-60
-80
-100
-120
-140
Power/frequency (dB/Hz)
0
100
200
300
400
500
Frequency (Hz)

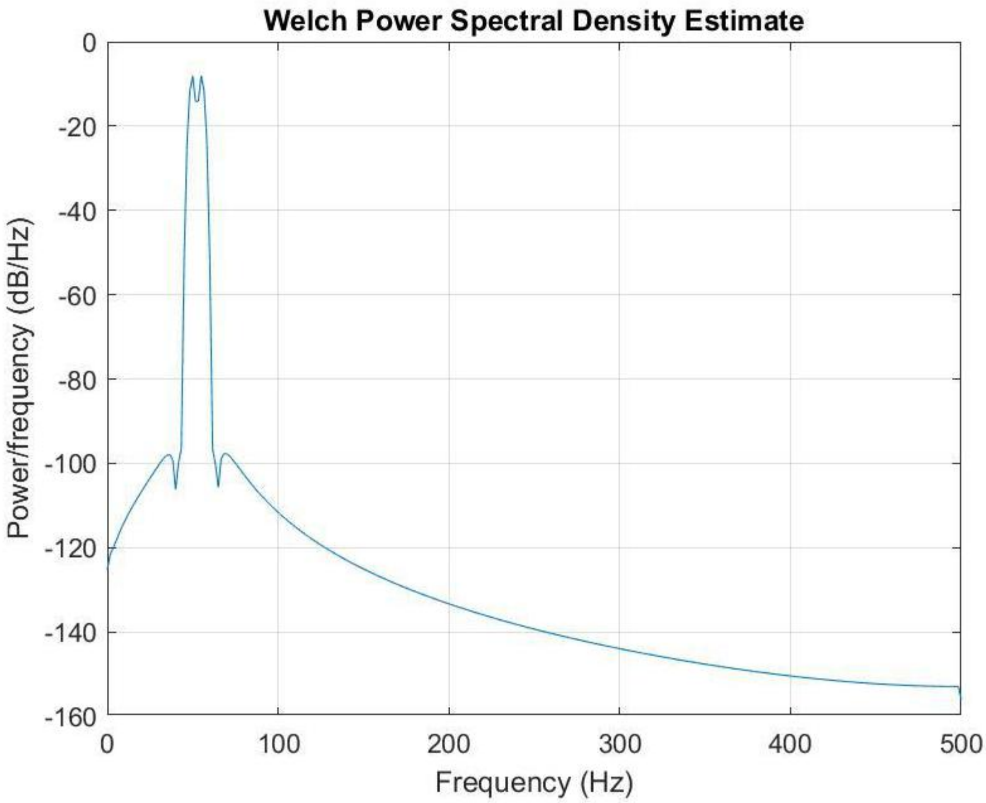


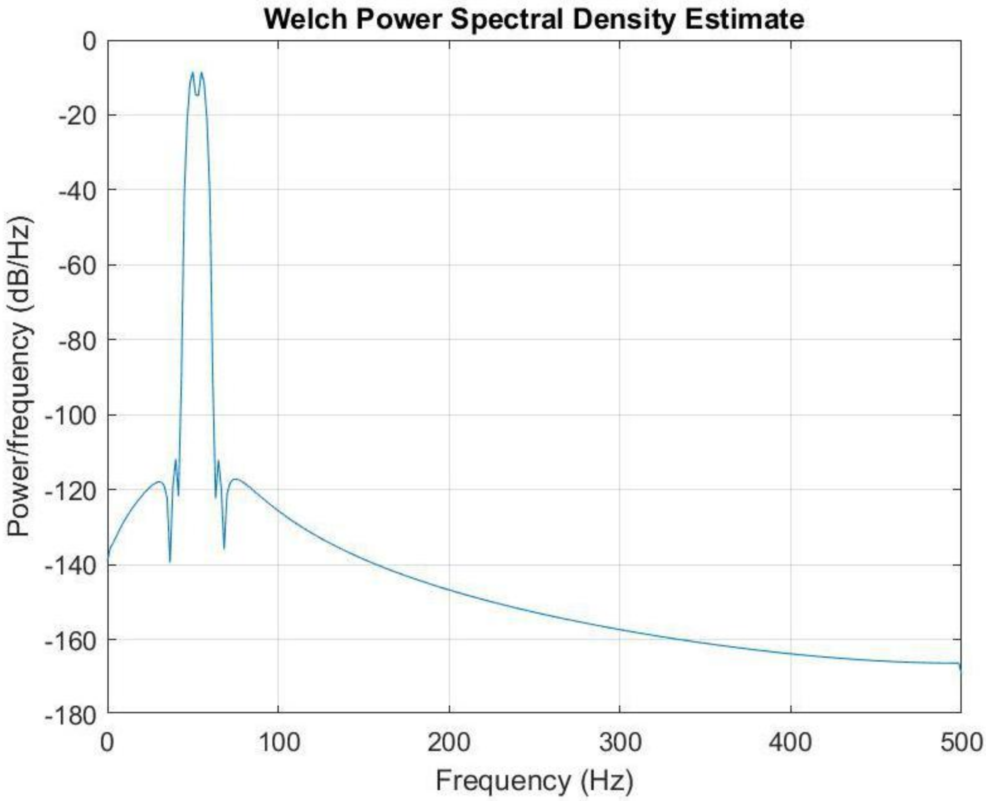
Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
0
-20
-40
-60
-80
-100
-120
-140
0
100
200
300
400
500

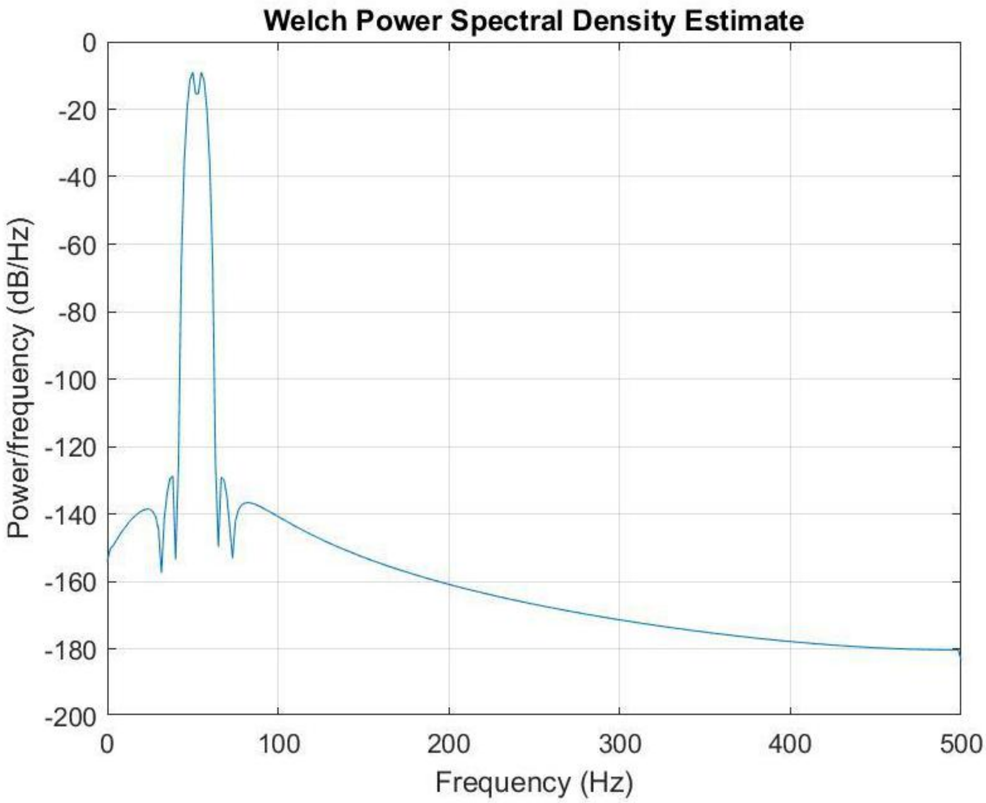


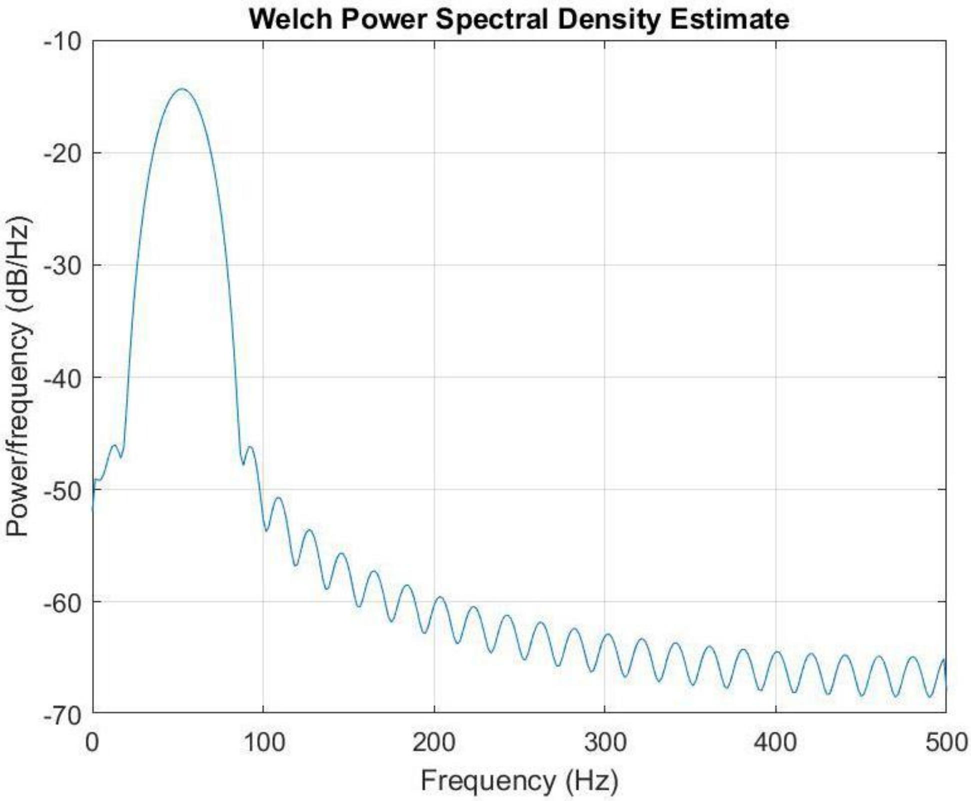
Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
0
-50
-100
-150
0
100
200
300
400
500

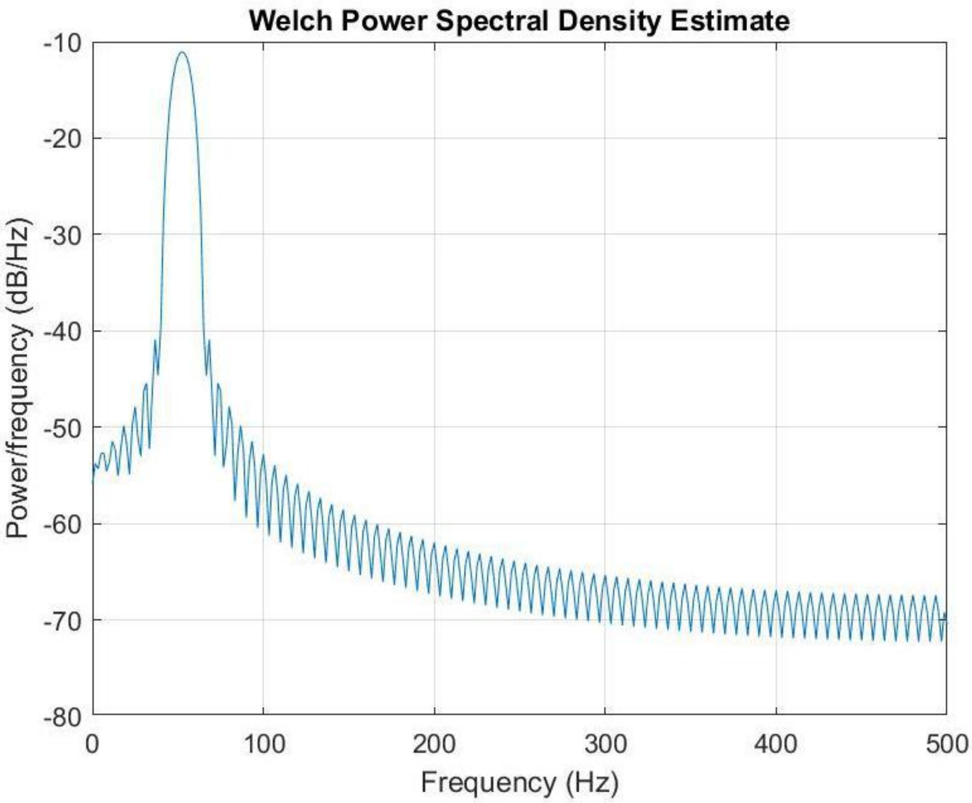

Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
0
-20
-40
-60
-80
-100
-120
-140
0
100
200
300
400
500

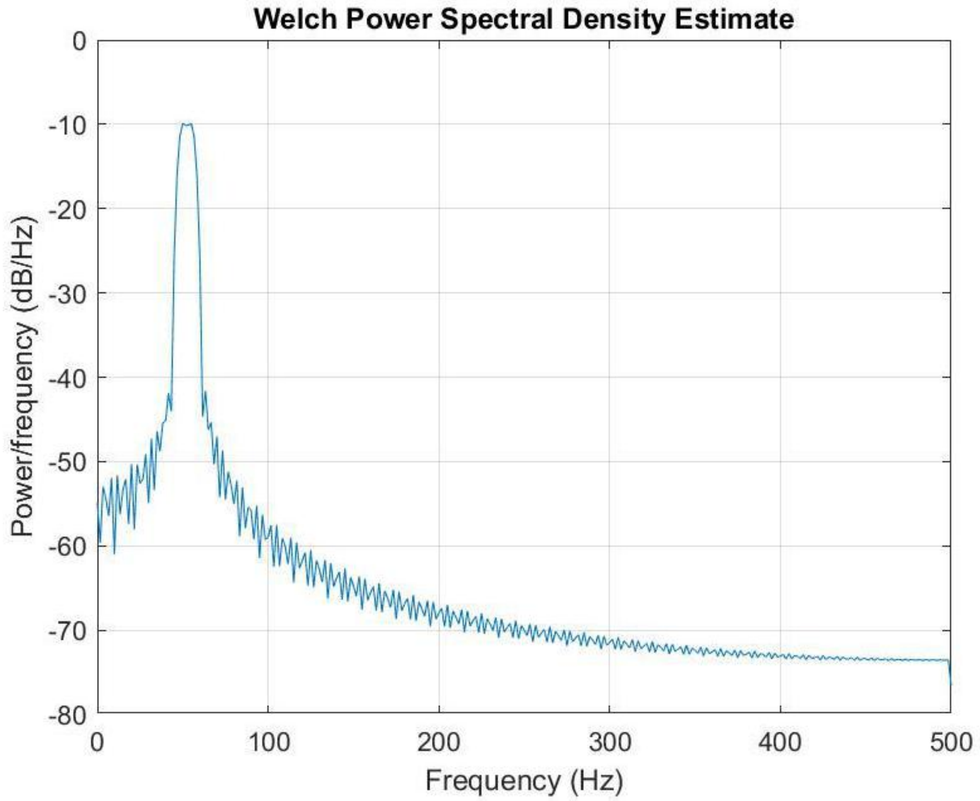

Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
0
-20
-40
-60
-80
-100
-120
-140
-160
0
100
200
300
400
500

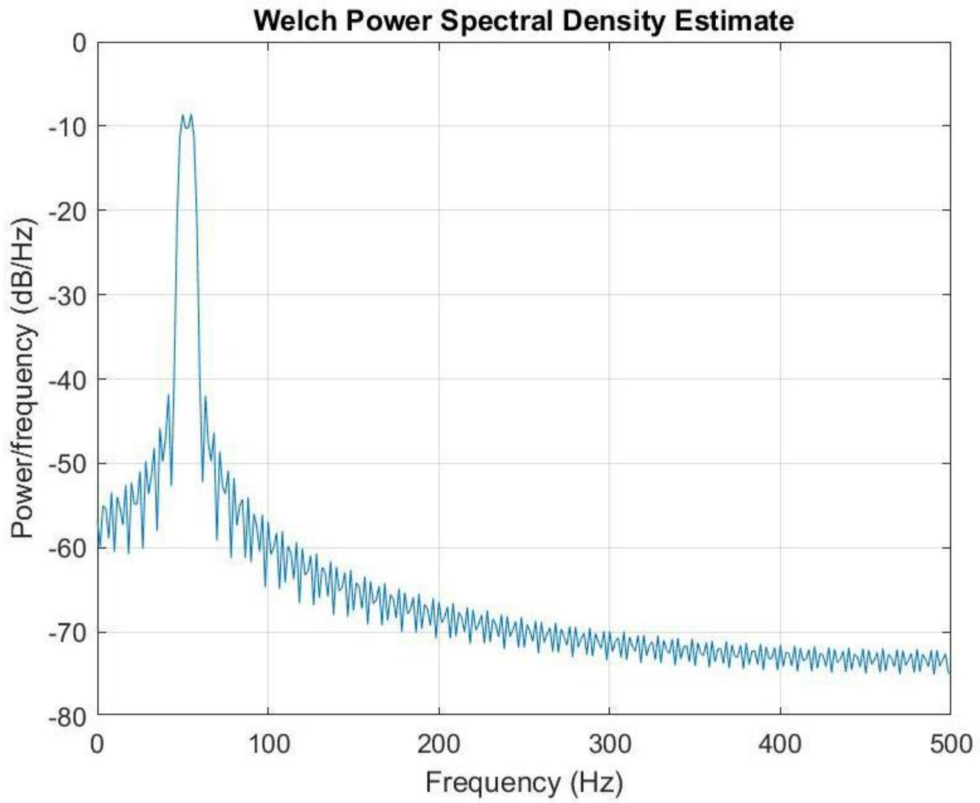

Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
0
100
200
300
400
500

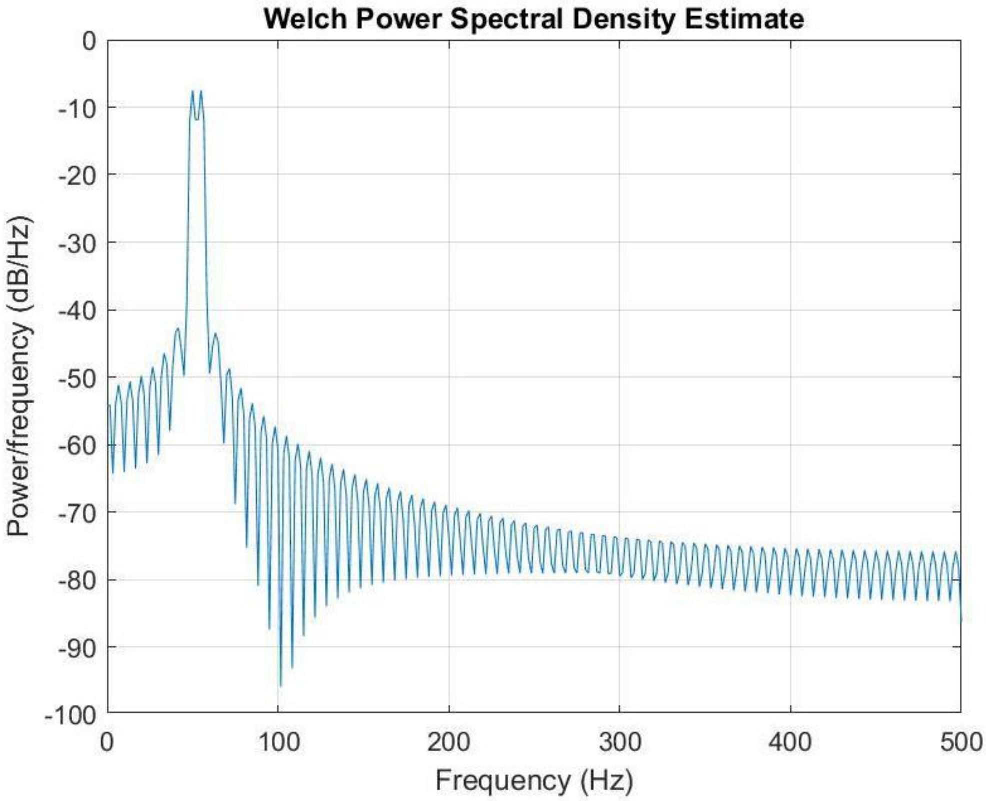

Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
-200
0
100
200
300
400
500

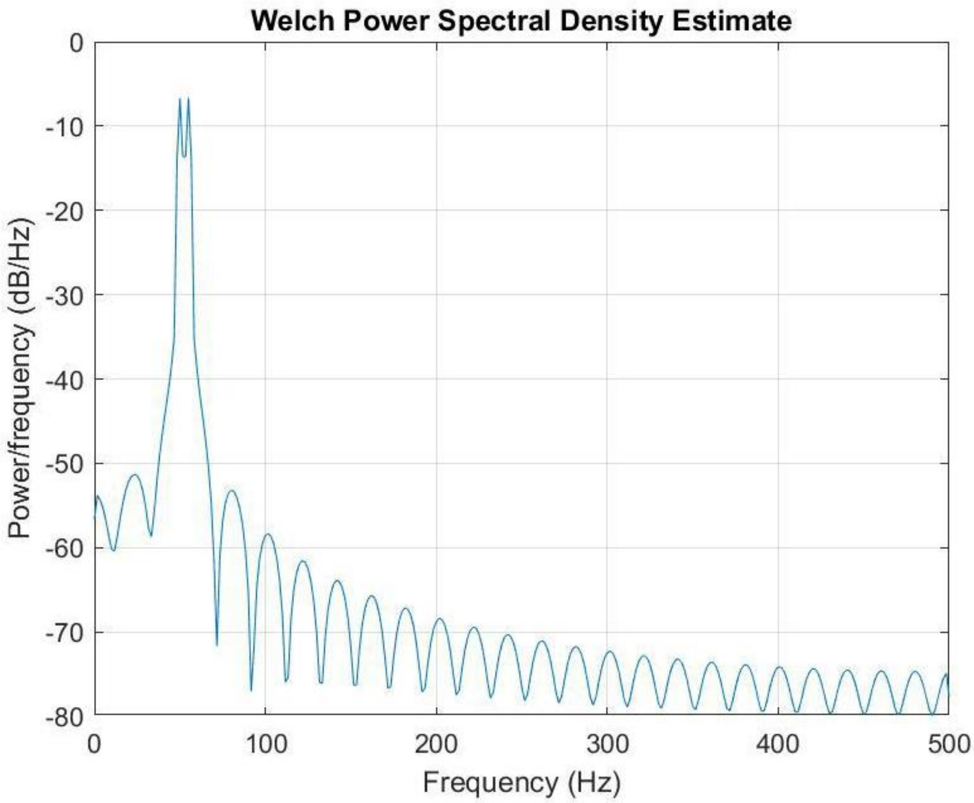

Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
-10
-20
-30
-40
-50
-60
-70
0
100
200
300
400
500


Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
-10
-20
-30
-40
-50
-60
-70
-80
0
100
200
300
400
500


Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
0
-10
-20
-30
-40
-50
-60
-70
-80
0
100
200
300
400
500


Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (Hz)
0
-10
-20
-30
-40
-50
-60
-70
-80
0
100
200
300
400
500





Conclusion:-

PSD of a signal has been performed in MATLAB

5.ECG_SIGNAL_READ_PLOT

Aim:-To write a code to read ECG signal using MATLAB

Tool:-MATLAB&SIMULINK

Version:-R2018b

Procedure:-

1.Open MATLAB file
2. Make sure your database path must be clear
3.Write a code for ECG_ Signal read and plot in "Editor"
3.First write clear all and close all because sometimes before program values are stored ,those will be execute.
4.After write the programming run the program
5.It will ask enter the path -you have to give correct path
6.output will be displayed.
7. ECG signals Database : From MIT-BIH - **physionet.org**

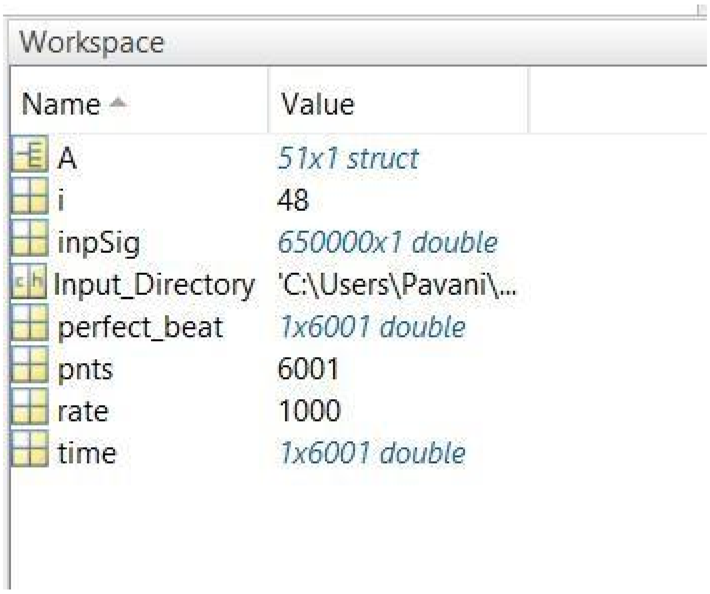
get the ECG signals database link here:
https://drive.google.com/drive/folders/1nj70UY20fkoj0v-N2TFCfMqIXGkdEKOB?usp=sharing

Code:-

```
clc; clear;
close all;
%% Overall samples in the data is 6,50,000. As it takes
much time to load.
 %So we take up to 6000 samples with sampling rate 1000.
rate=1000;
time=0:1/rate:6;
pnts=length(time);
Input_Directory = input('Enter the Input Path : '); %
Enter the path to get database..
A = dir(Input_Directory);  % Getting entire
directory...
%% Loading persons ECG data one by one %%%%%%%%%%
for i=1:size(A,1)-3         % Loop runs for  persons in
the database
   %% database consists 48 persons ECG signal data
excelsheets.each excel
   %named with "rec_1",it consists of 650000 sample
values of that ECG
```

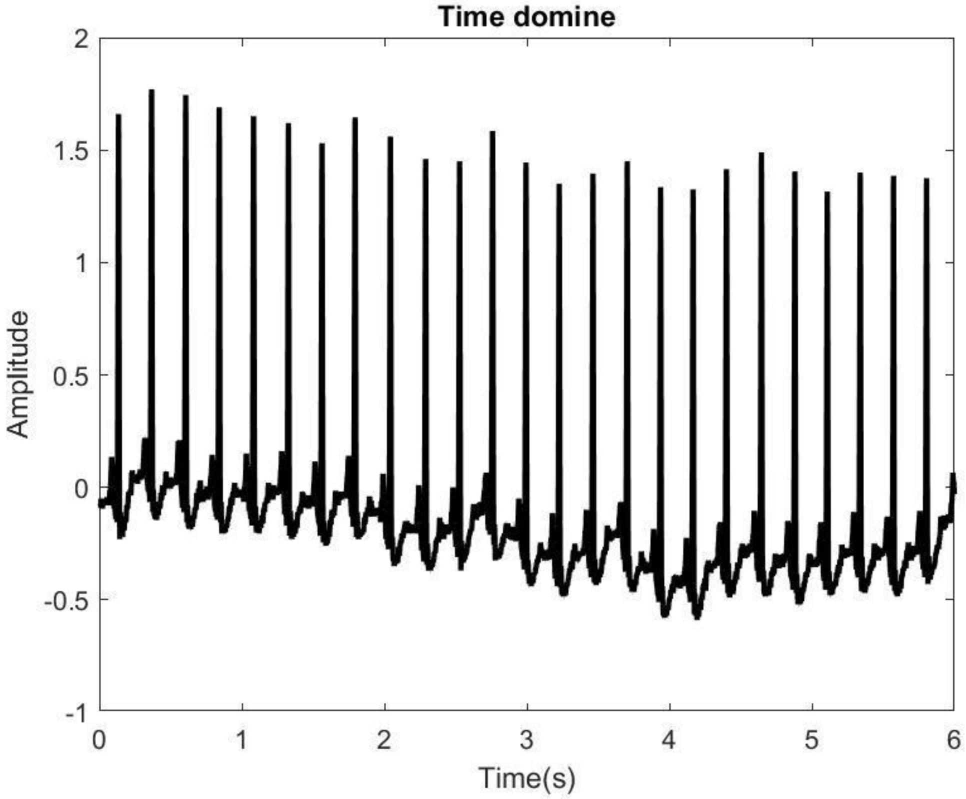
```
   %signal. Now we read xlsheet from given database. for reading the xlsheet we
   % use"xlsread" command.
   inpSig = xlsread([Input_Directory '\' A(i+3,1).name '\rec_1.xlsx']);% loading persons ecg samples record file
   inpSig=inpSig(:,1);  %taking only first ecg channel.
   disp(strcat(num2str(A(i+3,1).name)))%disply person ID
   perfect_beat=(inpSig(1:6001))';
   plot(time,perfect_beat,'k','linew',2) %ploting the ecg signal
   xlabel('Time(s)'),ylabel('Amplitude')
   title('Time domine')
   sound(perfect_beat);% we can hear the person heartbeat sound
 end
```

WORKSPACE:-


Workspace

Name	Value
A	51x1 struct
i	48
inpSig	650000x1 double
Input_Directory	'C:\Users\Pavani\...
perfect_beat	1x6001 double
pnts	6001
rate	1000
time	1x6001 double

Output plot:-



Conclusion:-

The ECG signal to be plotted using given database in MATLAB

MATLAB CODES , AUDIO FILES AND ECG RECORDS AVAILABLE IN BELOW LINK (please download these files)

(https://drive.google.com/drive/folders/1mNpJNQo2rtAkE1jDKlpSn4fIbUo2e0se?usp=sharing)

........THANK YOU........