

USING MATLAB TO ANALYZE SIGNALS IN TIME & FREQUENCY DOMAINS

ECE 321 – PRINCIPLES OF COMMUNICATION SYSTEMS

Tuesday, 02 December 2025

WHAT IS THE AUDIO FREQUENCY RANGE

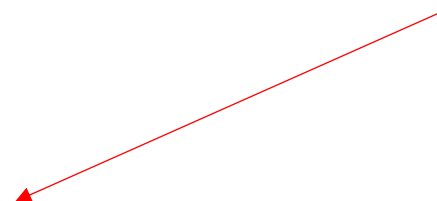
PROCEDURE

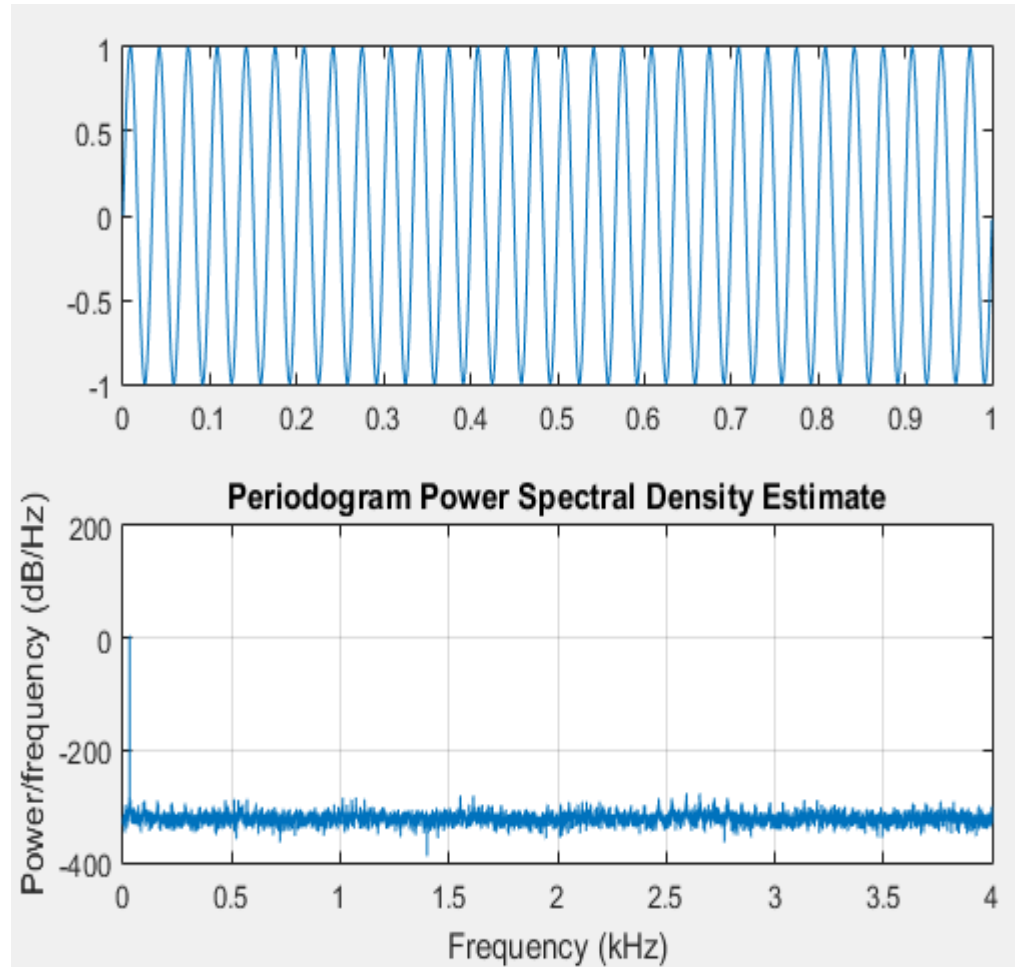
1. Create 40,000 samples of a sinewave given by
 $y(t) = \sin(2\pi f t)$ $f=10, 30, 100, 1000, 10,000, 20,000$
at a sampling rate of 8,000samples/sec.
2. Plot the signal
3. Send the signal to the computer speaker
4. Graph the Spectrum

MATLAB CODE

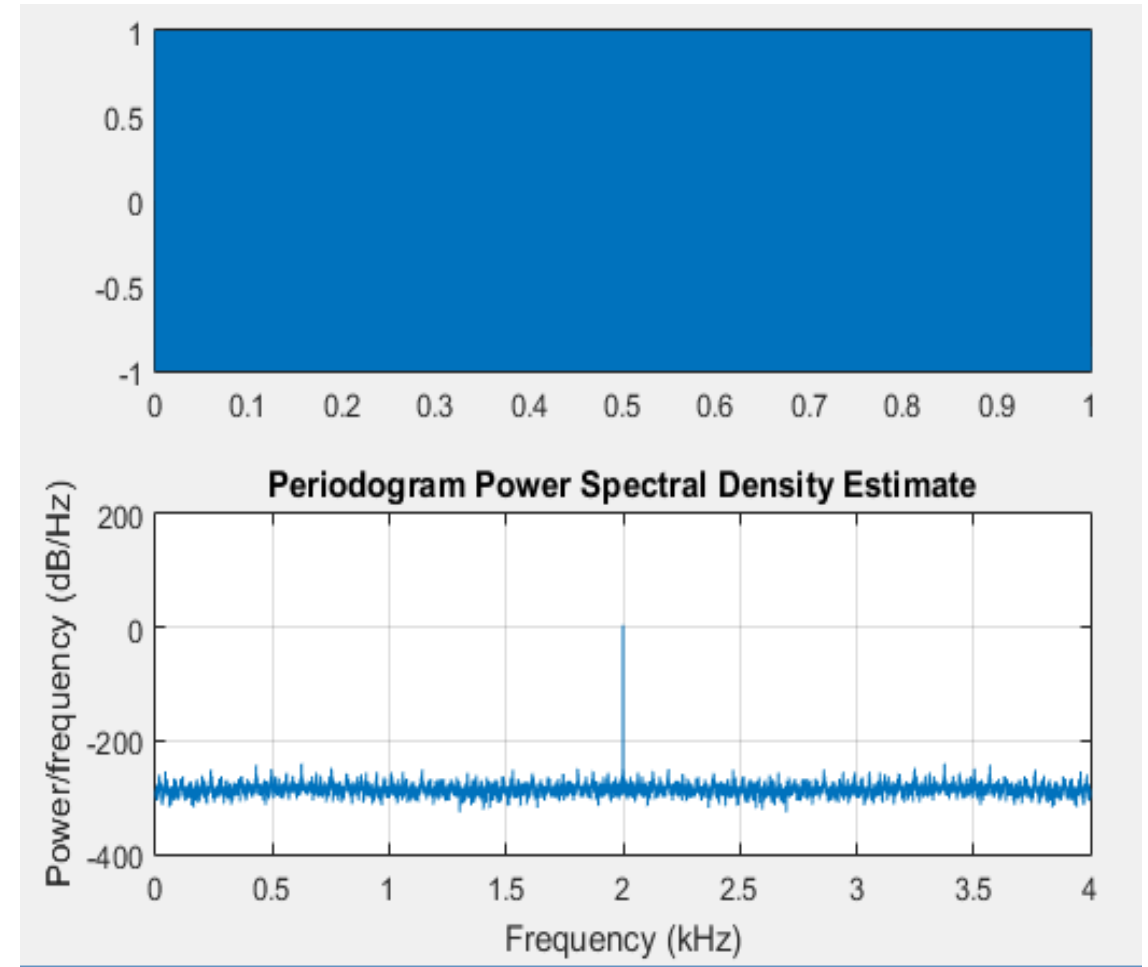
```
fs = 8000;  
N = 40000;  
t= 0:1/fs:(N-1)/fs  
y = sin(2*pi*1000*t);  
subplot(2,1,1);  
plot(t,y,'r');  
sound(y);  
subplot(2,1,2)  
Y = psd(spectrum.periodogram,y,'Fs',fs,'NFFT',length(y));  
plot(Y);
```

Change the
frequency to: 30,
200, 100, 2,000,
3,500 Hz





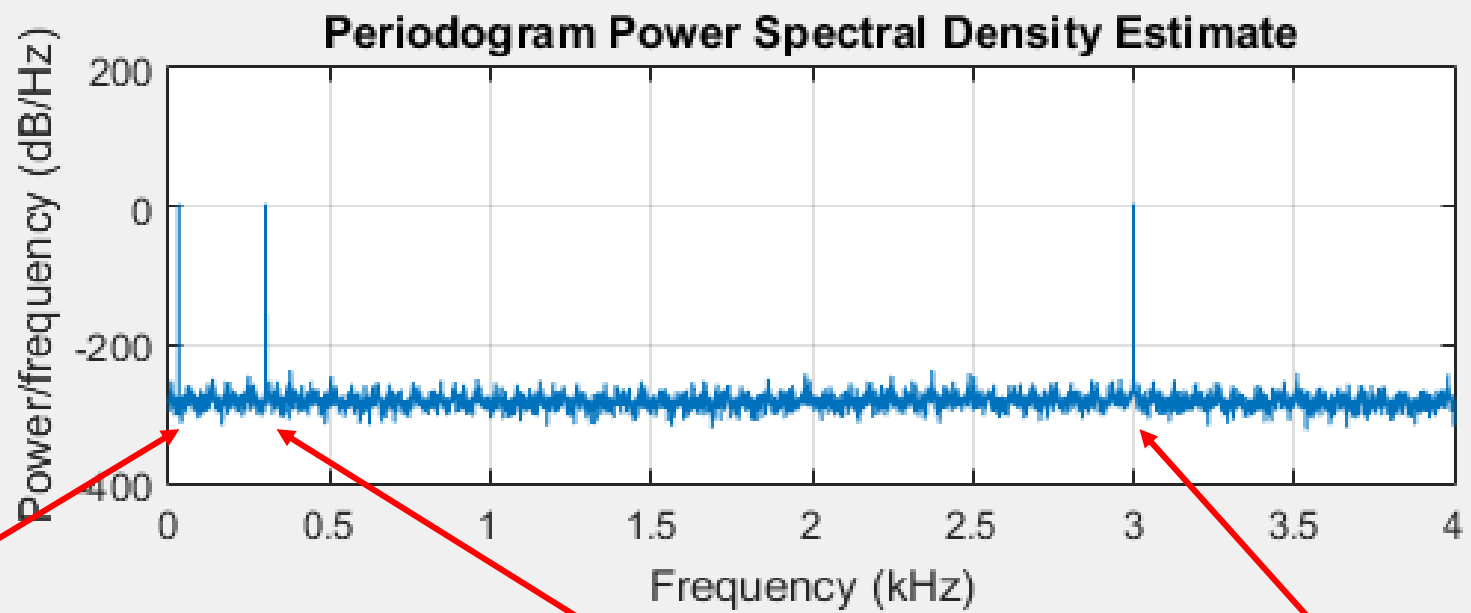
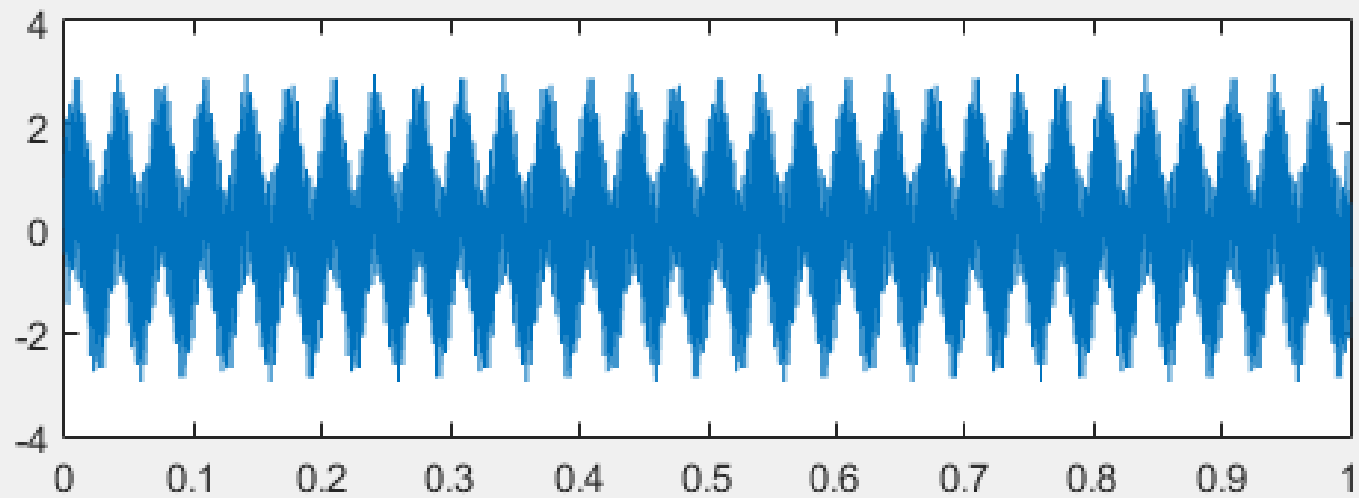
(a) Display for 30 Hz



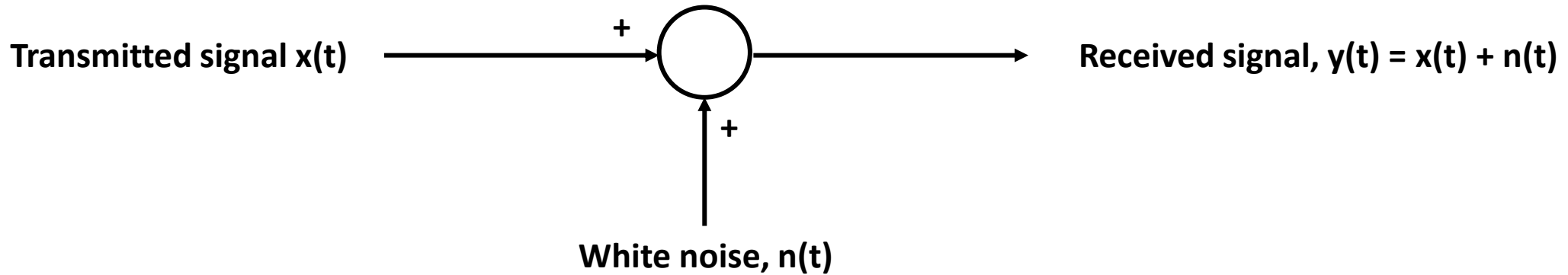
(b) Display for 2,000 Hz

SIGNAL WITH THREE SINUSOIDS

```
fs = 8000;  
N = 40000;  
t = 0:1/fs:(N-1)/fs  
y = sin(2*pi*30*t) + sin(2*pi*300*t) + sin(2*pi*3000*t);  
subplot(2,1,1);  
plot(t(1:8000), y(1:8000));  
sound(y);  
subplot(2,1,2)  
Y = psd(spectrum.periodogram, y, 'Fs', fs, 'NFFT', length(y));  
plot(Y);
```



SIMULATING CHANNEL INDUCED NOISE



MATLAB IMPLEMENTATION

- **$y = \text{awgn}(x, \text{snr})$** adds white Gaussian noise to the vector signal x .
- The scalar snr specifies the signal-to-noise ratio per sample, in dB.
- If x is complex, awgn adds complex noise.
- This syntax assumes that the power of x is 0 dBW.

SPECTRUM OF A SIGNAL WITH ADDITIVE WHITE GAUSSIAN NOISE

```
fs = 8000;  
N = 40000;  
t= 0:1/fs:(N-1)/fs  
y = sin(2*pi*30*t)+ sin(2*pi*300*t)+ sin(2*pi*3000*t)  
y1=awgn(y,20);    % Pass signal through a noisy channel  
subplot(2,1,1);  
plot(t(1:8000),y1(1:8000));  
sound(y);  
subplot(2,1,2)  
Y = psd(spectrum.periodogram,y1,'Fs',fs,'NFFT',length(y1));  
plot(Y);
```

ENERGY SPECTRAL DENSITY OF AUDIBLE SIGNAL /01

PROCEDURE

1. Record your voice for 4 seconds
2. Convert the recording into a double precision real value
3. Graph the signal
4. Play the recording on computer speakers
5. Compute the power spectral density
6. Display the power spectral density

ENERGY SPECTRAL DENSITY OF AUDIO SIGNAL /02

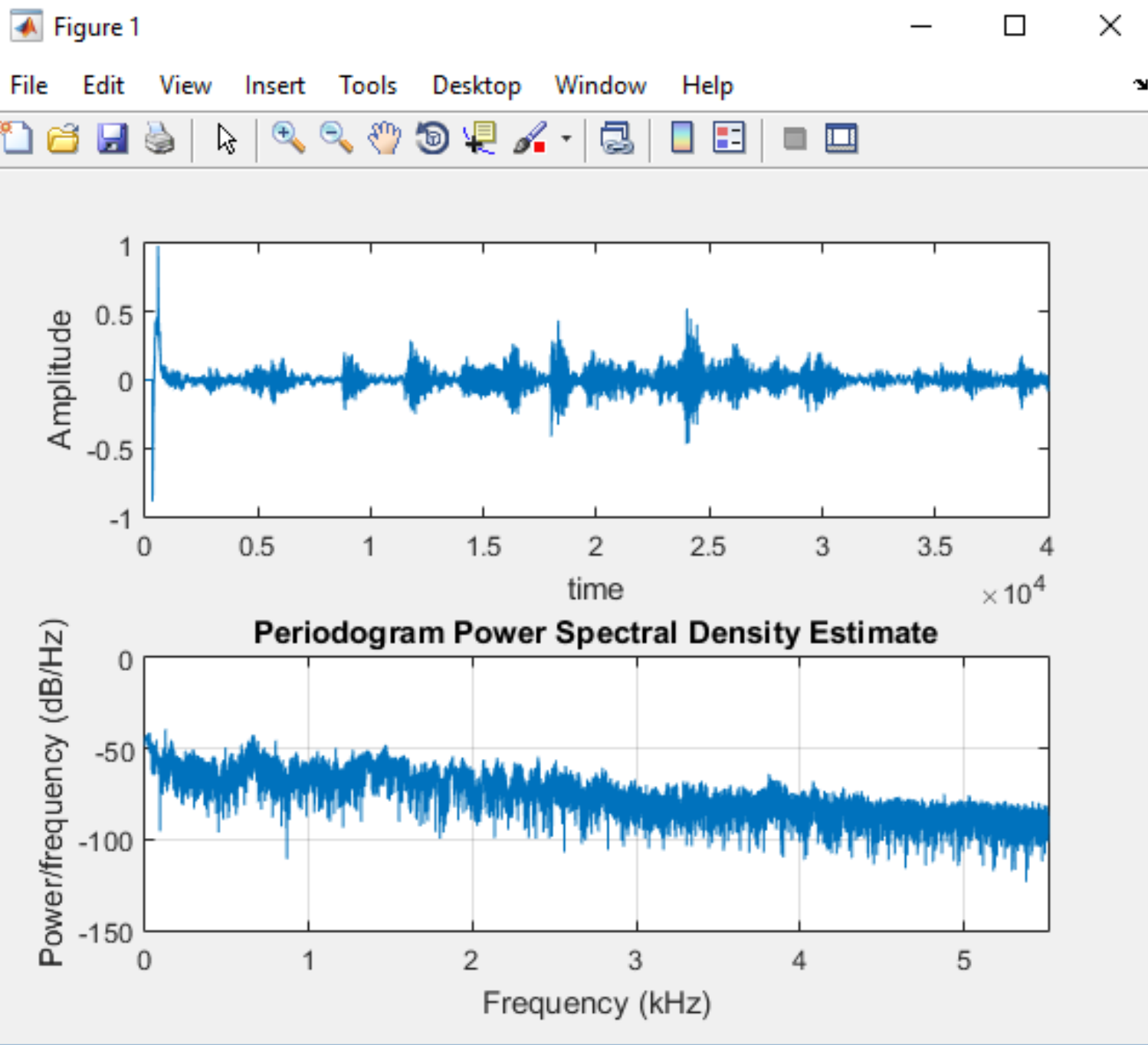
```
rec = audiorecorder
disp('Start speaking....')           % Prompt user to start speaking
recordblocking(rec, 4);              % Record your voice for 4 seconds
disp('End of Recording.');
```



```
play(rec);                          % Play sound on computer speaker
y = getaudiodata(rec);               % Store data in double-precision array, y.
subplot(2,1,1)
plot(y);
```



```
Y = psd(spectrum.periodogram,y,'Fs',fs,'NFFT',length(y));
subplot(2,1,2);
plot(Y);
```



POWER SPECTRAL DENSITY OF AUDIO SIGNAL /03

```
rec = audiorecorder
disp('Start speaking.')
recordblocking(rec, 5);
disp('End of Recording. ');
y = getaudiodata(rec);
subplot(2,1,1)
plot(y)
play(rec)
zz = size(y)
N=8000;
k=1:N/2;
F=fft(y);
magF=abs(F/N);
subplot(2,1,2)
plot(k(1:N/2),magF(1:N/2),'r');
title('Spectrum Using FFT')
```

