Signal Analysis Examples - MAT 367

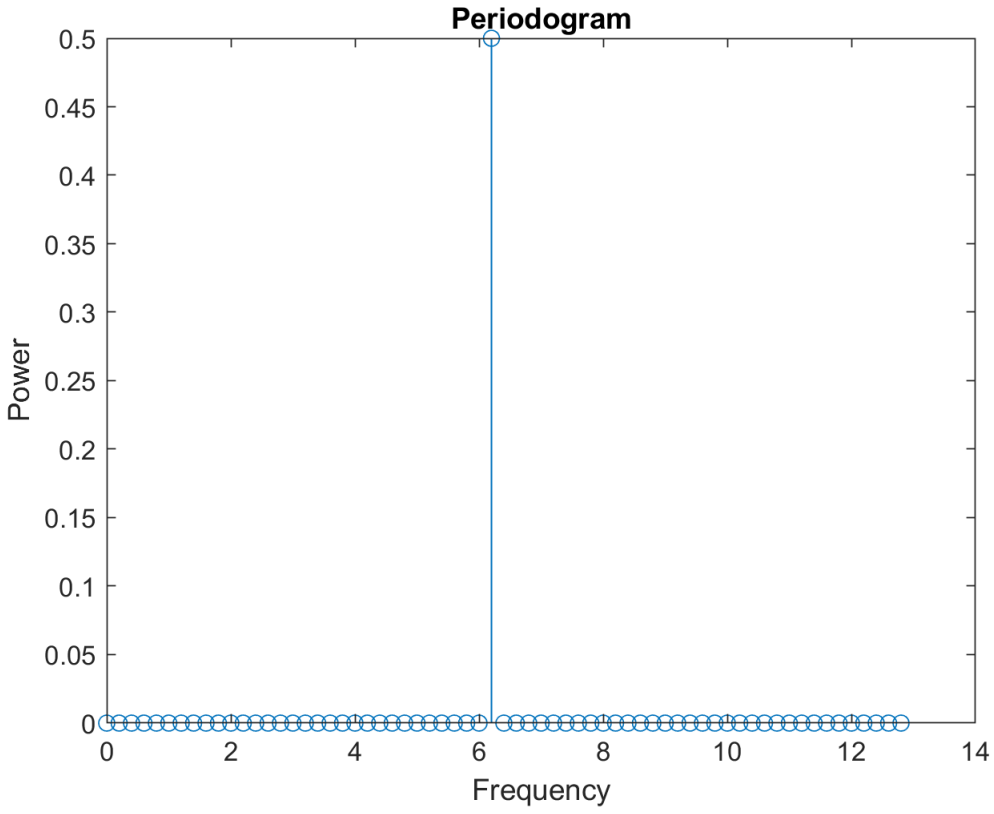
Here are some examples based on Chapter 7 of the textbook.

Using FFT

First, we demonstrate the use of the function (below) fanalfun (fanalf in project) in which the FFT function is used and a periodogram is drawn.

This example is a simple sine function with frequency f_0.

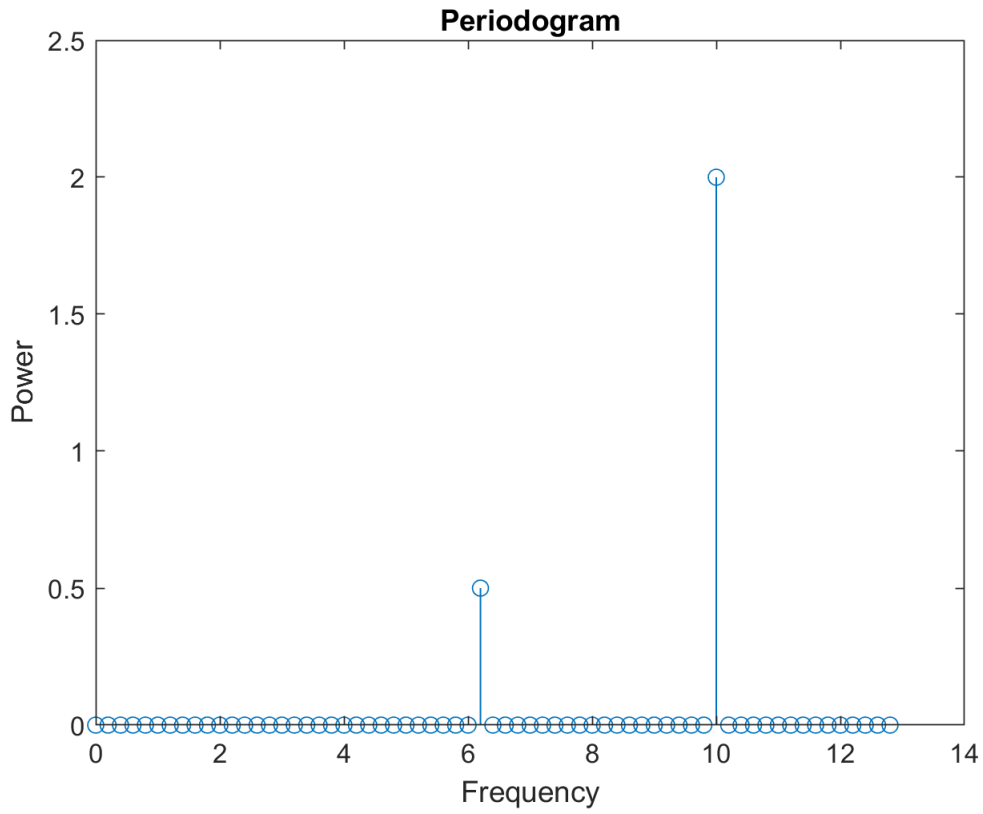
```
clear
n=128;
T=5;
dt=T/n;
f0=6.2;
t=(1:n)*dt;
y=sin(2*pi*f0*t);
fanalfun(y,T);
```



Now we add another sine function with frequency f_1 and note the DFT.

```
n=128;
T=5;
dt=T/n;
```

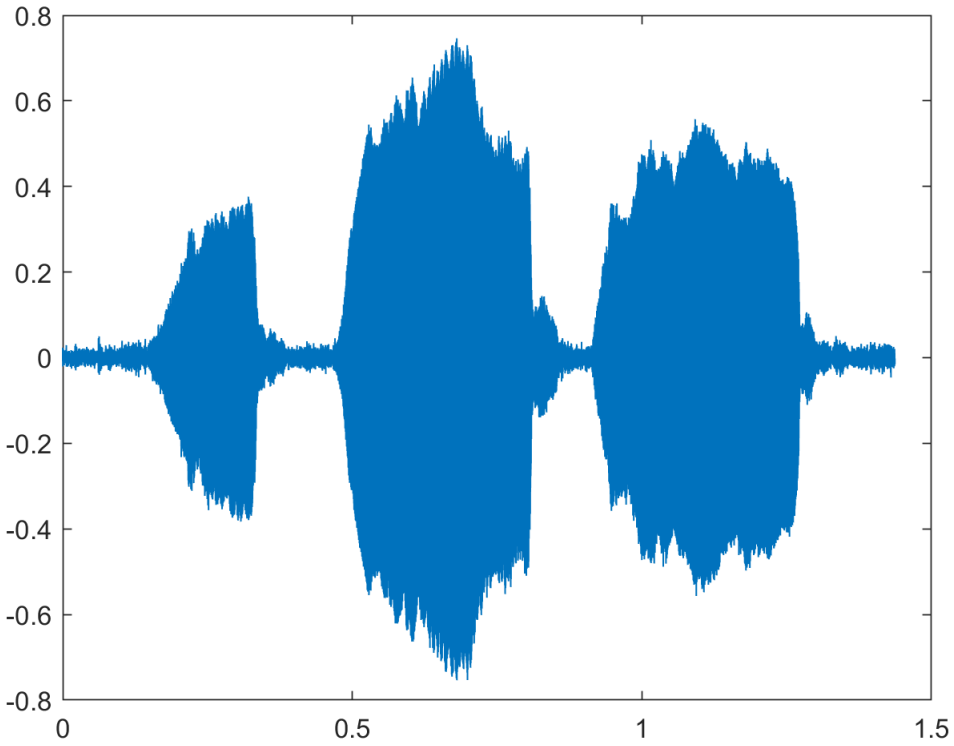
```
f0=6.2;
f1=10;
t=(1:n)*dt;
y=sin(2*pi*f0*t)+2*sin(2*pi*f1*t);
fanalfun(y,T);
```



Analyzing Audio Data

It is useful to apply this analysis to time series or sound files. Here we load a sound file using audioread and plot the signal.

```
[y,NS]=audioread('firstbird.wav');
n=length(y);
T=n/NS;
dt=T/n;
figure(1)
plot(dt*(1:n),y)
```

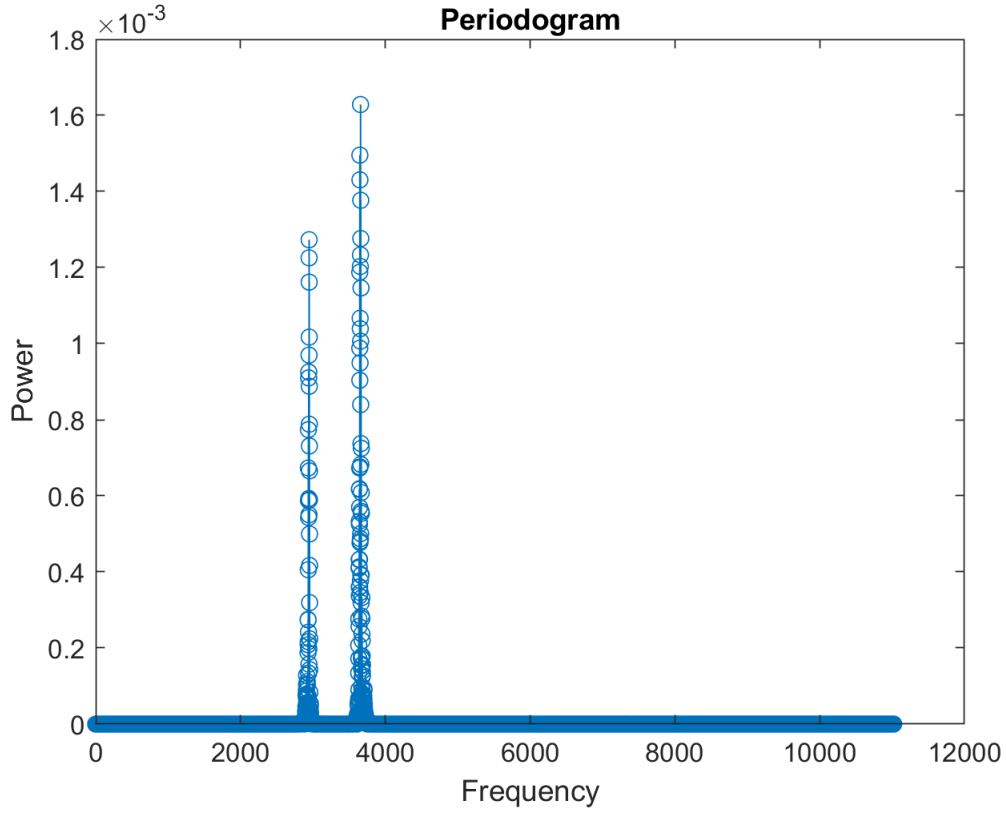


We can listen to the sound.

```
sound(y,NS);
```

We then look at the periodogram.

```
figure(2)
fanalf(y,T);
```



We can also create our own sound and listen to it. Then, we save it using audiowrite.

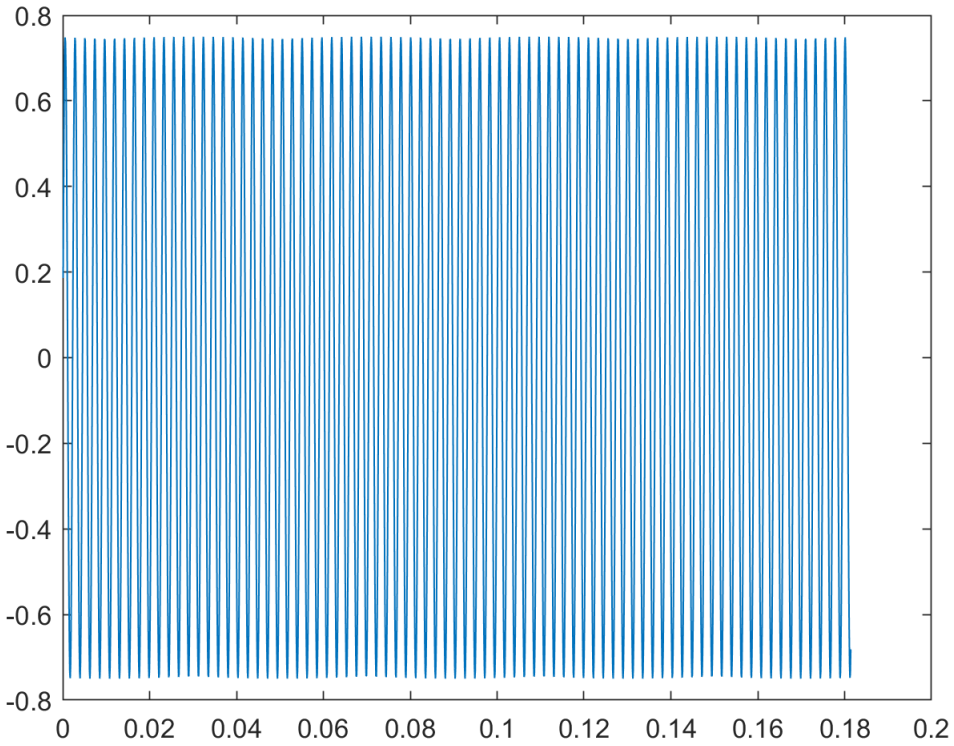
```
smp=11025;
t=(1:2000)/smp;
y=0.75*sin(2*pi*440*t);
sound(y,smp,8);
audiowrite('mysound.wav',y,smp);
```

Let's verify that the created file can be read back.

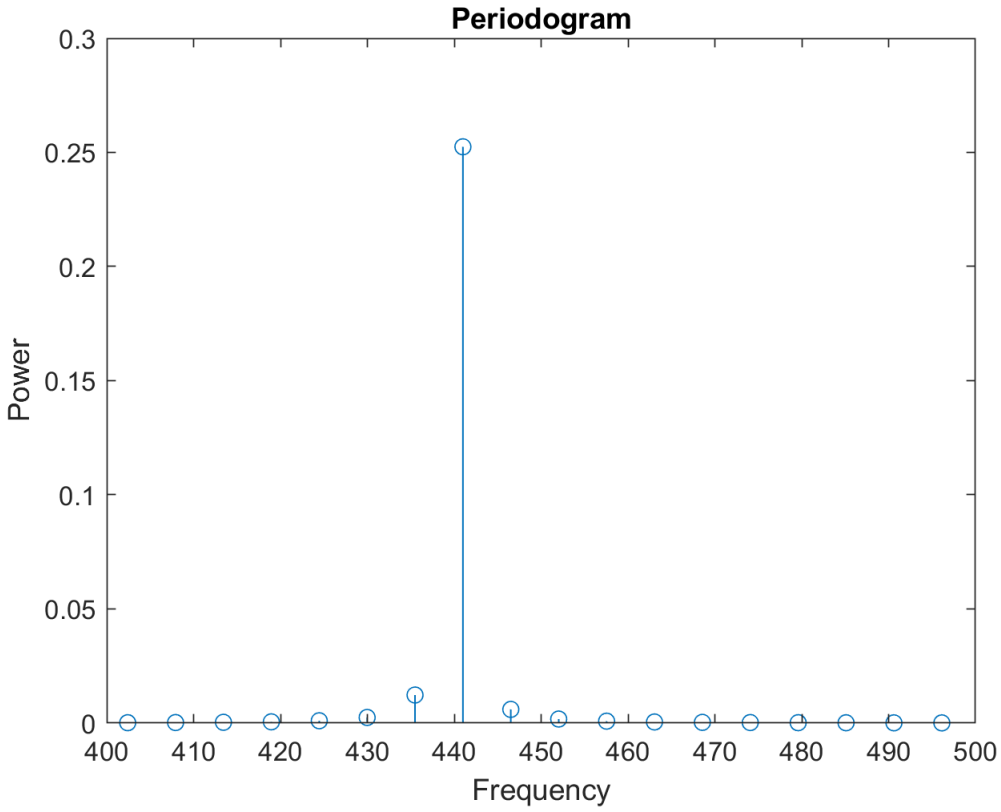
```
[y2,smp] = audioread('mysound.wav');
sound(y2,smp)
```

The periodogram is now drawn and we zoom in to the interesting part of the frequency domain.

```
n=length(y2);
T=n/smp;
dt=T/n;
figure(1)
plot(dt*(1:n),y2)
```



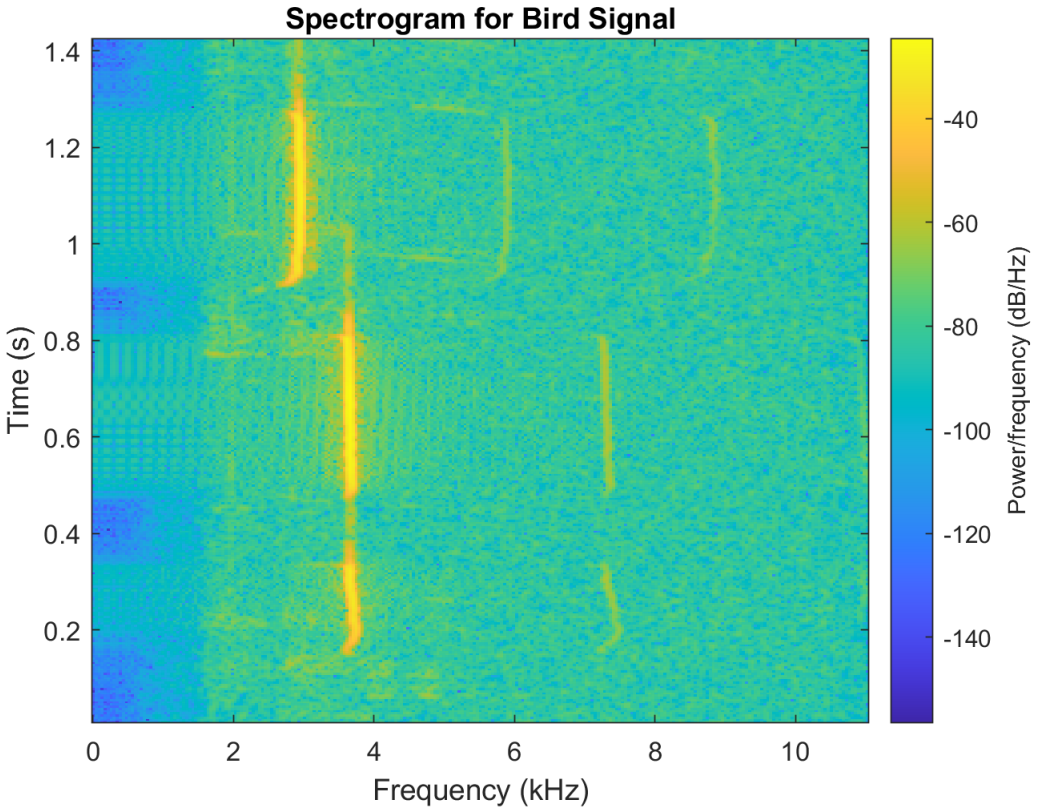

```
figure(2)
fanalf(y,T);
axis([400,500,0,0.3])
```



The Spectrogram

In the above analysis we determined the frequency content but do not know when particular notes were present. So, we use MATLAB's spectrogram function in the Signal Processing Toolbox.

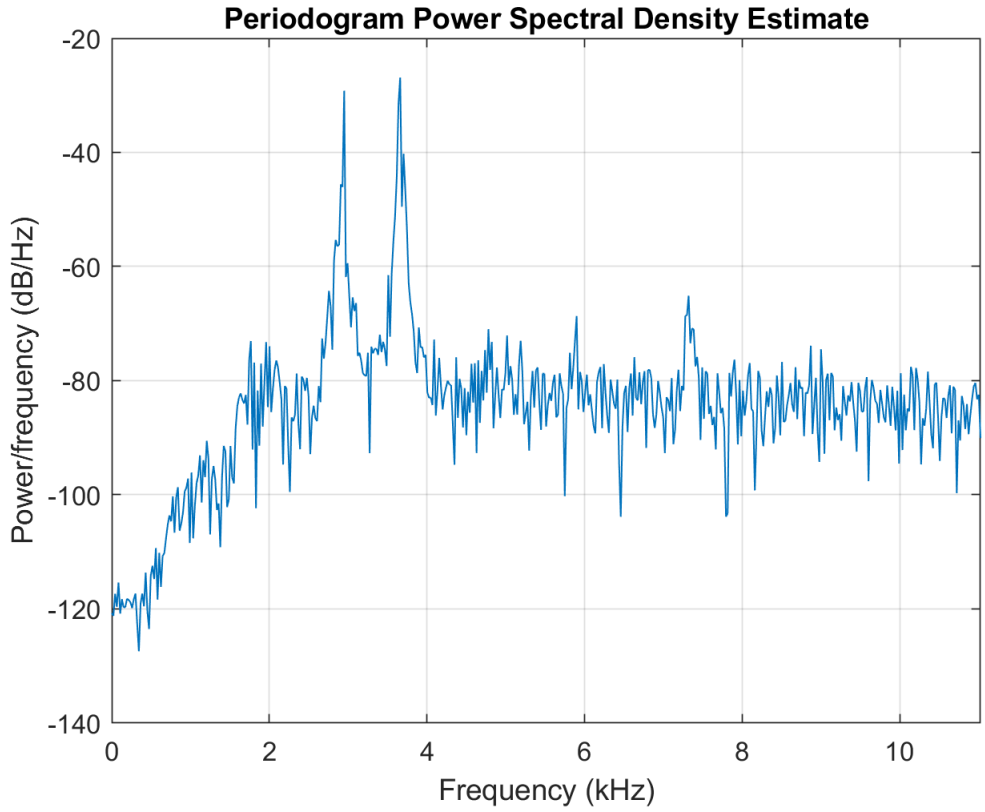
```
[y,NS]=audioread('firstbird.wav');
spectrogram(y,400,300,[],NS);
title('Spectrogram for Bird Signal')
ylabel('Time (s)')
```




```
%colormap bone
```

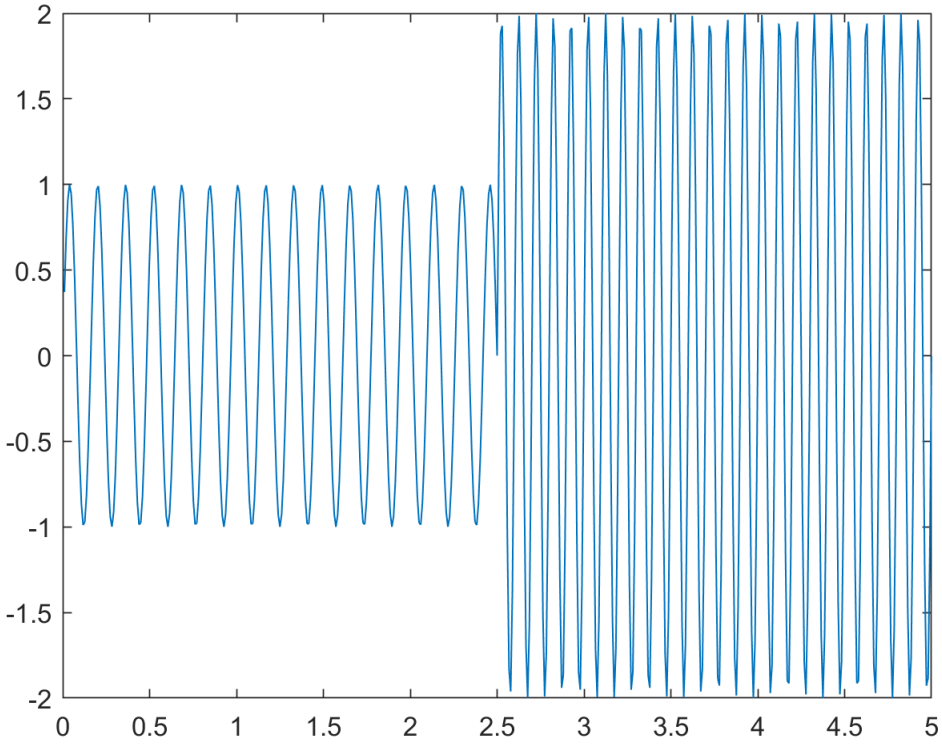
MATLAB also has a built-in periodogram function.

```
periodogram(y,[],'onesided',1024,NS)
```

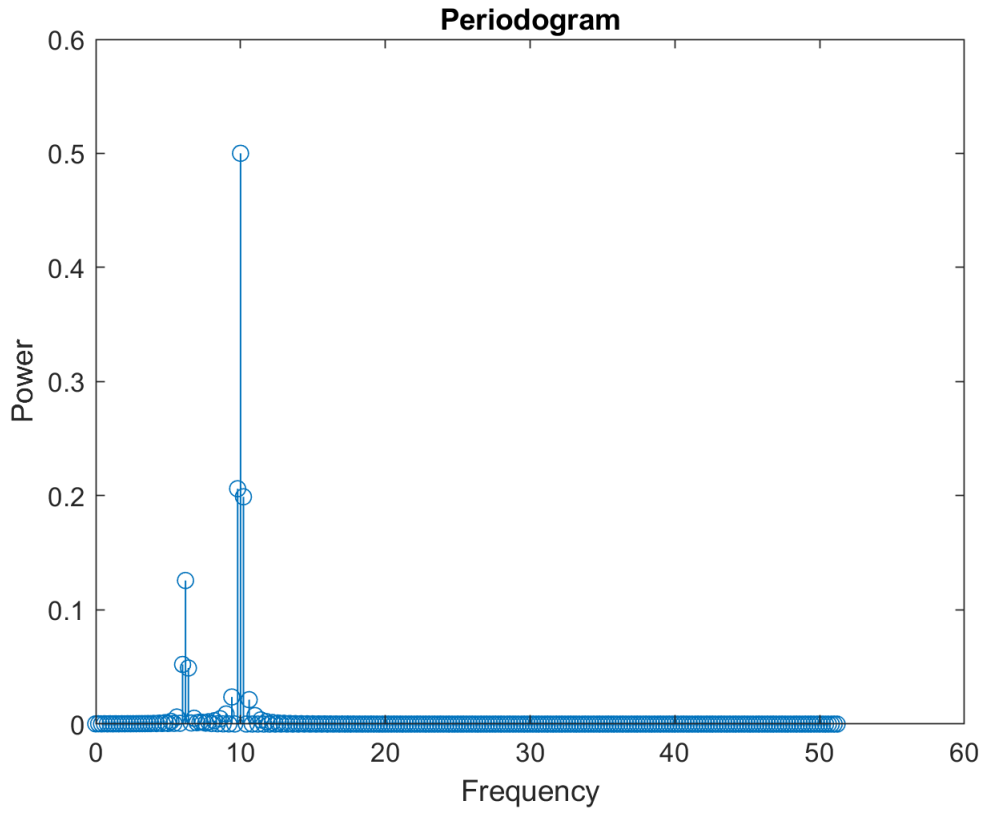


We can create a function in which the previous two frequencies occur one after the other and analyze the new signal.

```
clear
n=512;
T=5;
dt=T/n;
f0=6.2;
f1=10;
t=(1:n)*dt;
y(1:n/2)=sin(2*pi*f0*t(1:n/2));
y(n/2+1:n)=+2*sin(2*pi*f1*t(n/2+1:n));

figure(1)
plot(t,y)
```

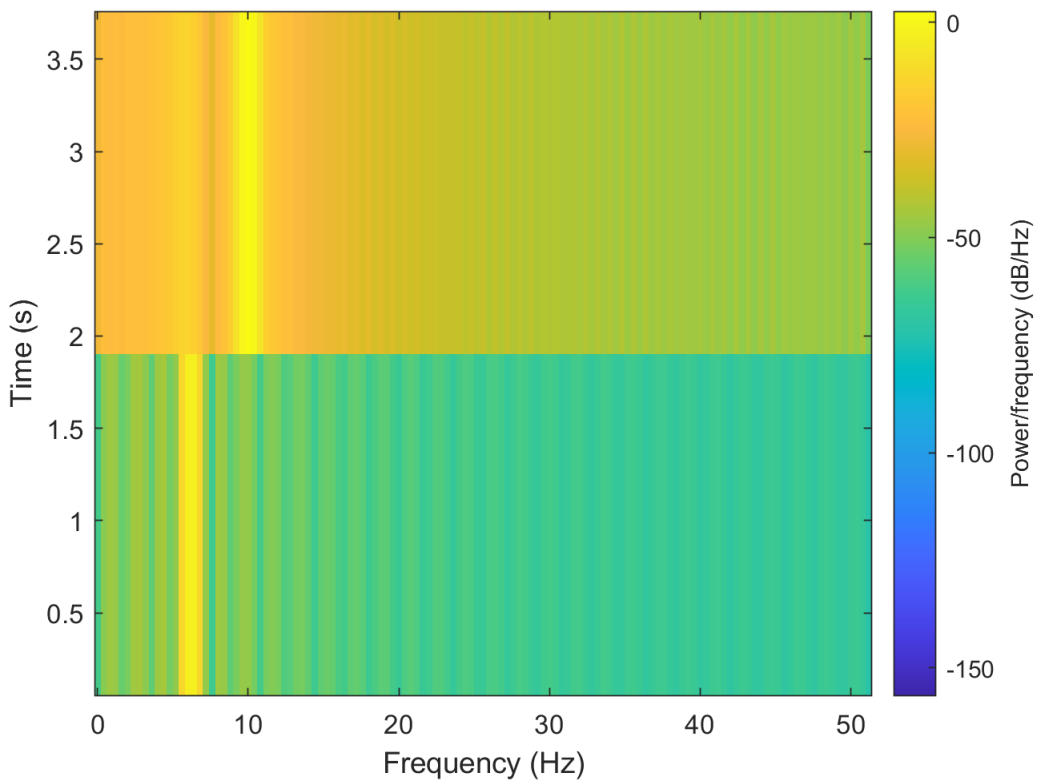



```
figure(2)
fanalf(y,T);
```



The spectrogram shows roughly when the frequencies occur.

```
spectrogram(y,200,10,[],n/T);
```

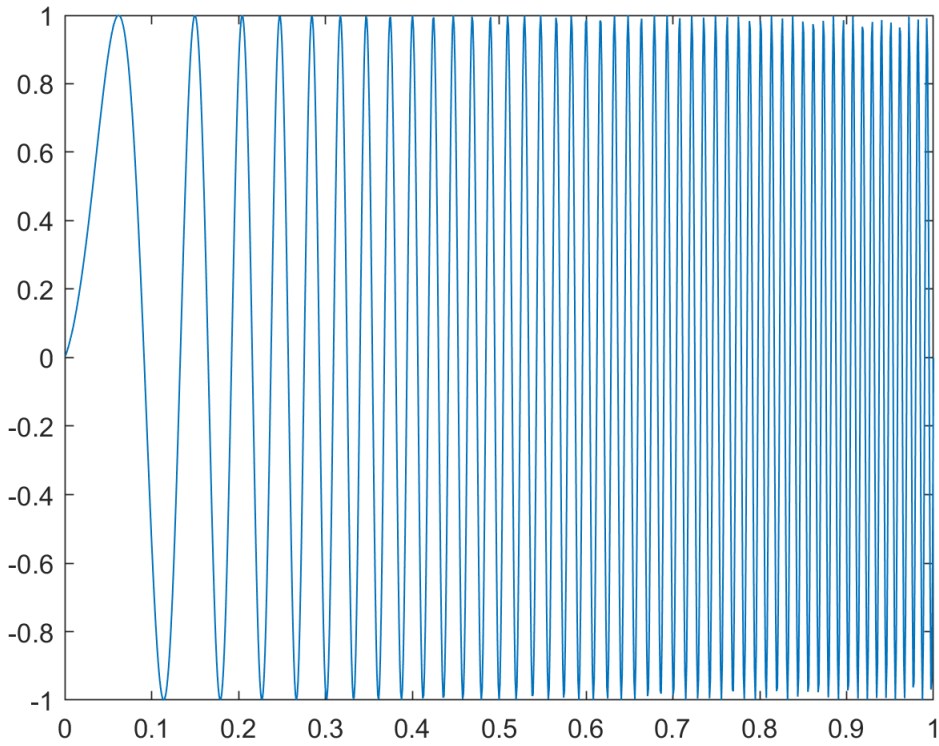



```
%axis([4,12,0,T])
```

The Chirp

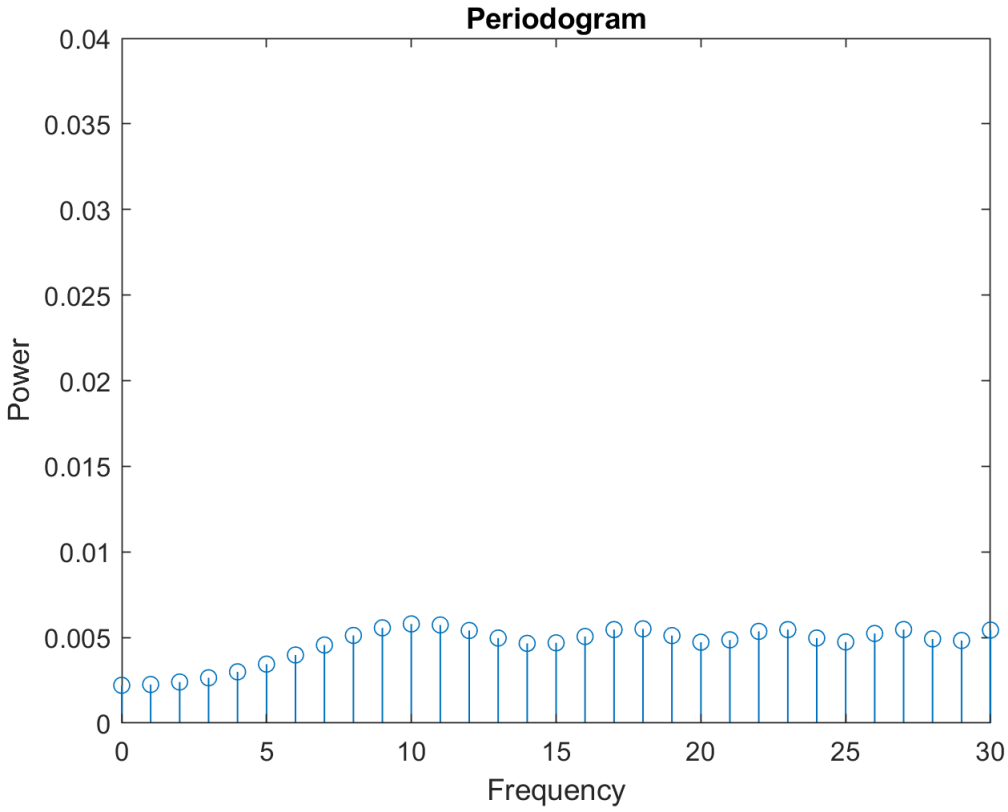
Another interesting signal is the chirp. We create our own signal which starts with frequency f_0 and linearly increases to frequency f_1. See more at MATLAB's Help Center: https://www.mathworks.com/help/signal/ref/spectrogram.html#bupbguu

```
n=1024;
T=1;
dt=T/n;
t=(1:n)*dt;
f0=1;
f1=50;
k=(f1-f0)/T;
x=sin(2*pi*(f0+k*t).*t);
plot(t,x)
```

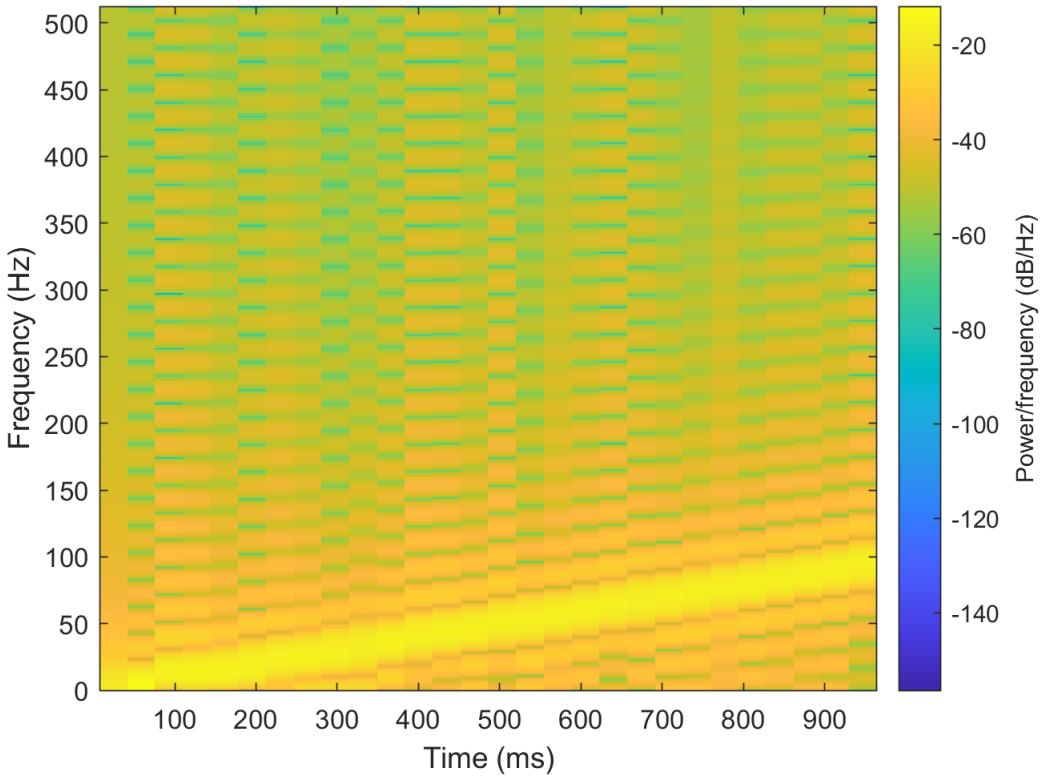


We now analyze the chirp with fanalfun and spectrogram.

```
fanalfun(x,T);
axis([0,30,0,.04])
```

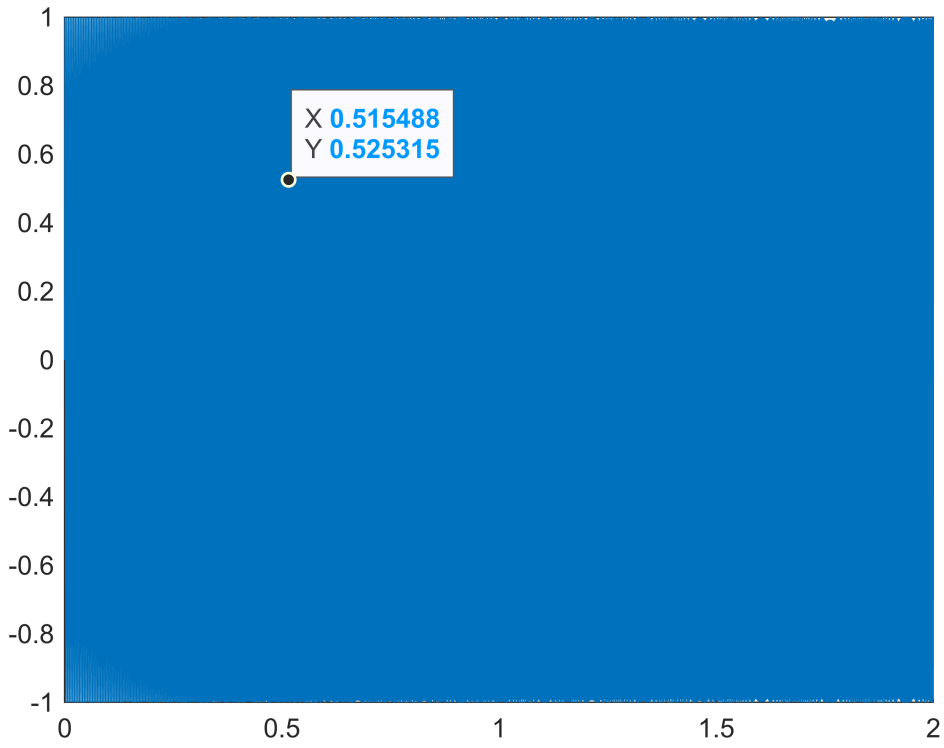



```
spectrogram(x,rectwin(50),15,n,1/dt,'yaxis')
```

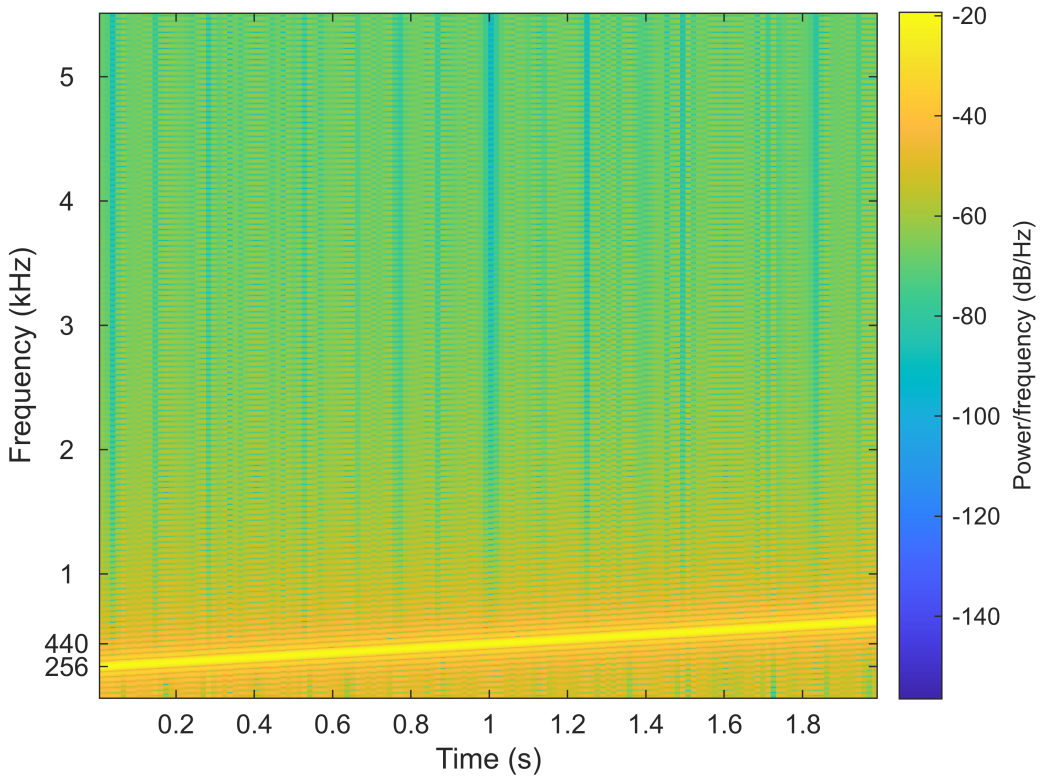


Hear a Chirp

```
smp=11025;
nsec=2;
t=linspace(0,nsec,smp*nsec);
f0=256;
f1=440;
k=(f1-f0)/nsec;
y=sin(2*pi*(f0+k*t).*t);
sound(y,smp,8);
plot(t,y)
```




```
n=length(t);
dt=1/smp;
spectrogram(y,rectwin(250),100,n,1/dt,'yaxis')
set(gca,'YTick',[.256 .44 1 2 3 4 5])
set(gca,'YTickLabel',[256 440 1 2 3 4 5])
```



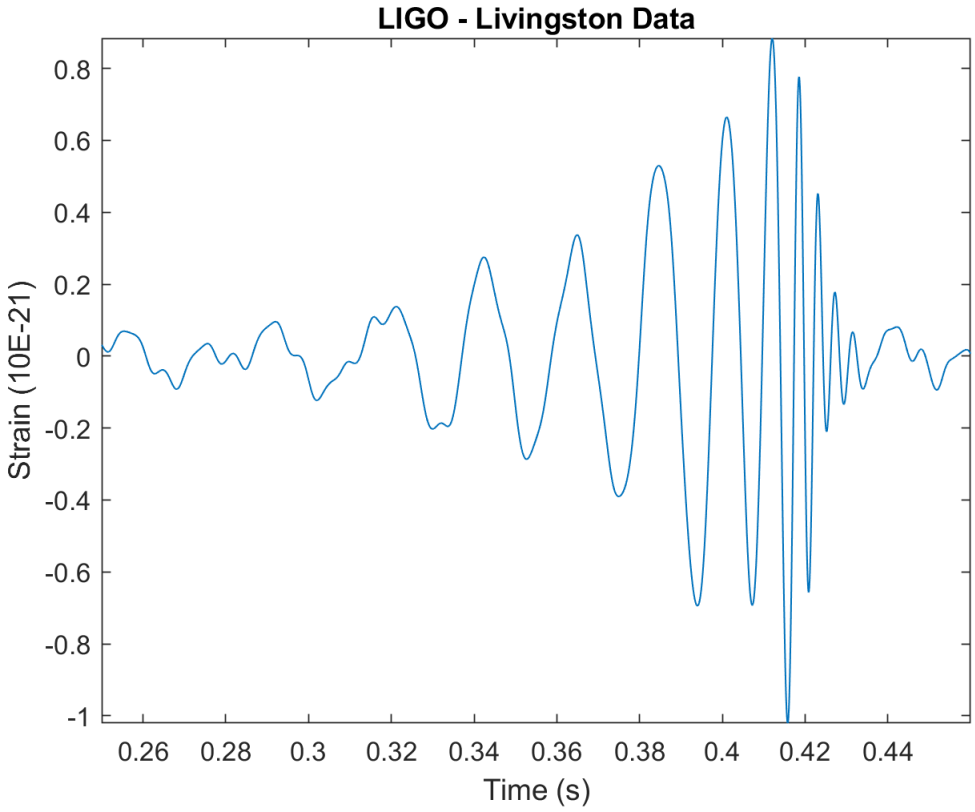
Inspiralling black-hole binaries

On September 14, 2015, LIGO observed gravitational waves from the merger of two black holes, each about 30 times the mass of our sun. It was reported February 11, 2016.

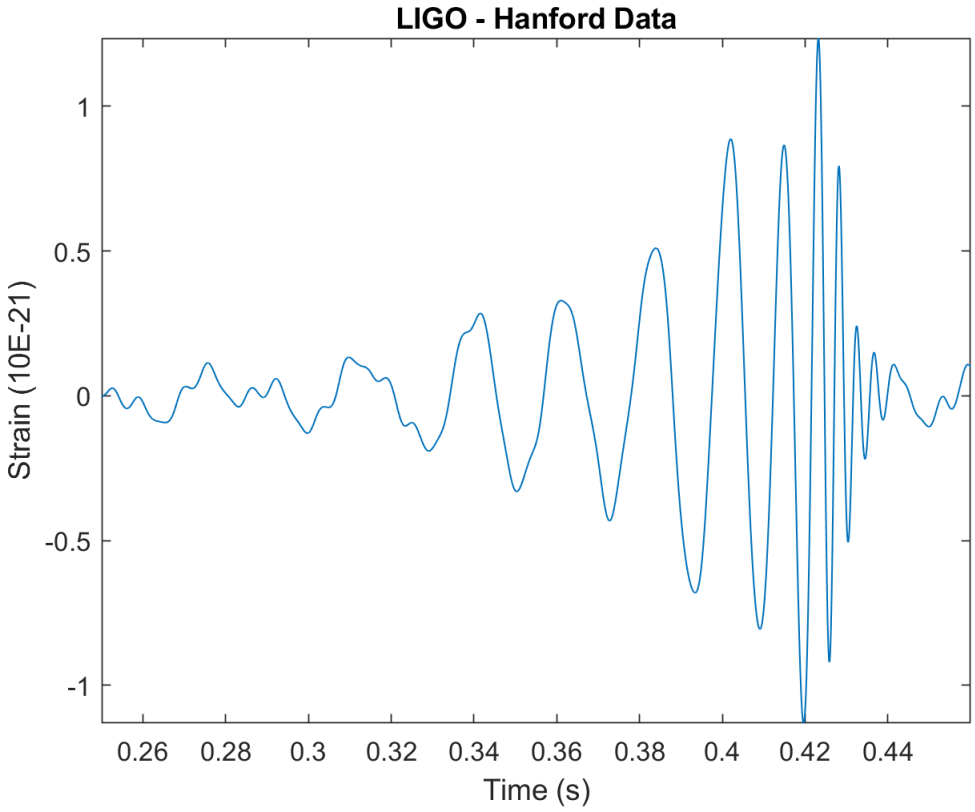
The signal seen was a chirp - https://www.youtube.com/watch?v=QyDcTbR-kEA&t=7s As the black holes spiraled towards each other, the frequency of the gravitational waves increased.

Hanover, WA data and Livingston, LA data.

```
clear

MH = dlmread('NumH.txt', ' ', 1, 0);
ML = dlmread('NumL.txt', ' ', 1, 0);

time=ML(:,1);
strain=ML(:,2);
n=length(strain);
t=time;
y=strain-mean(strain);
plot(t,strain)
xlabel('Time (s)')
ylabel('Strain (10E-21)')
axis([t(1), t(n), min(strain), max(strain)])
title('LIGO - Livingston Data')
```

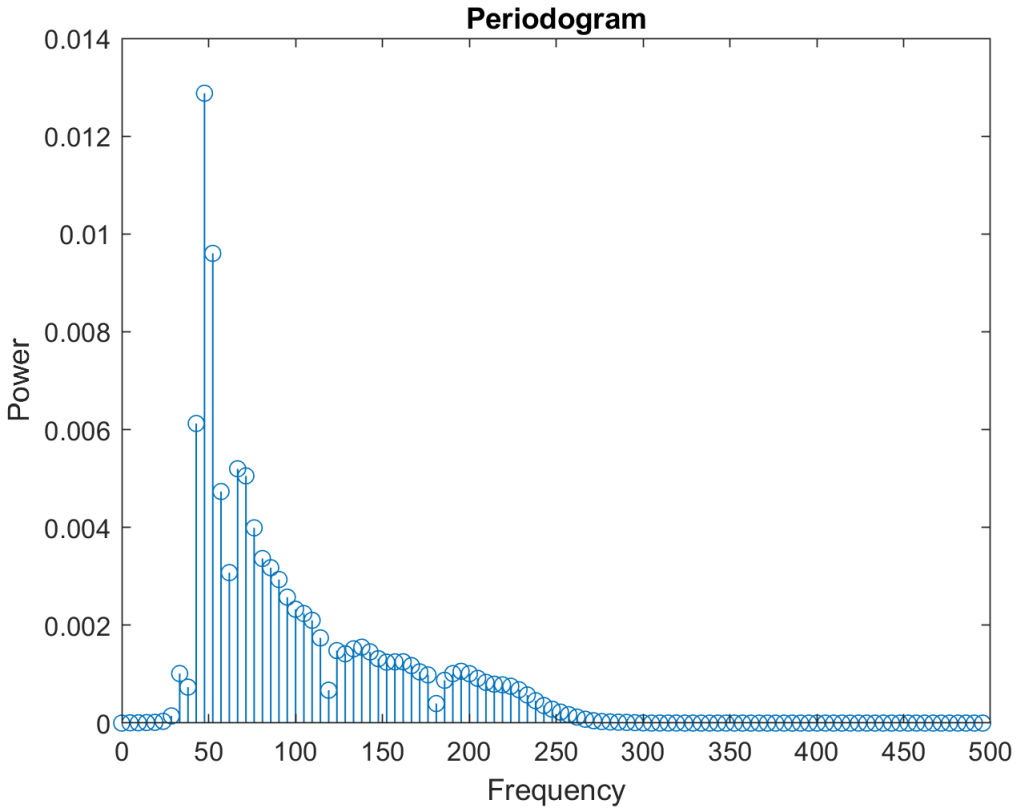



```
time=MH(:,1);
strain=MH(:,2);
n=length(strain);
t=time;
y=strain-mean(strain);

plot(t,strain)
xlabel('Time (s)')
ylabel('Strain (10E-21)')
axis([t(1), t(n), min(strain), max(strain)])
title('LIGO - Hanford Data')
```

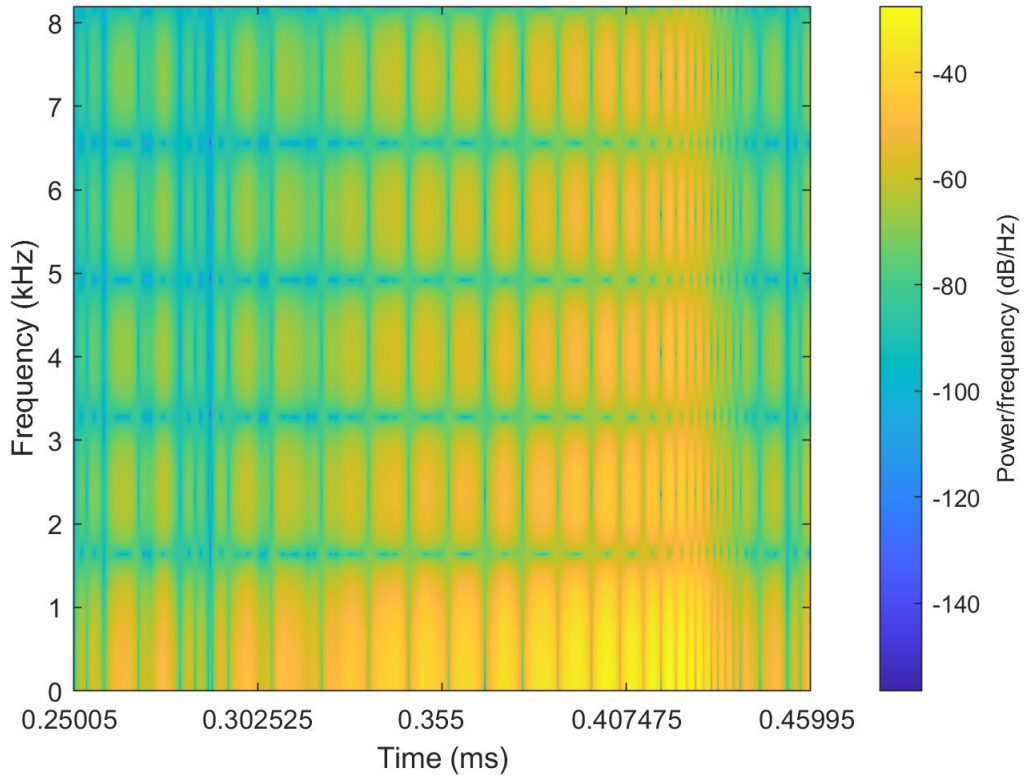


Fourier Analyze the Hanford data using fanalfun and spectrogram.

```
T=t(n)-t(1);
dt=T/n;
fanalfun(y',T);
axis([0,500,0,.014])
```

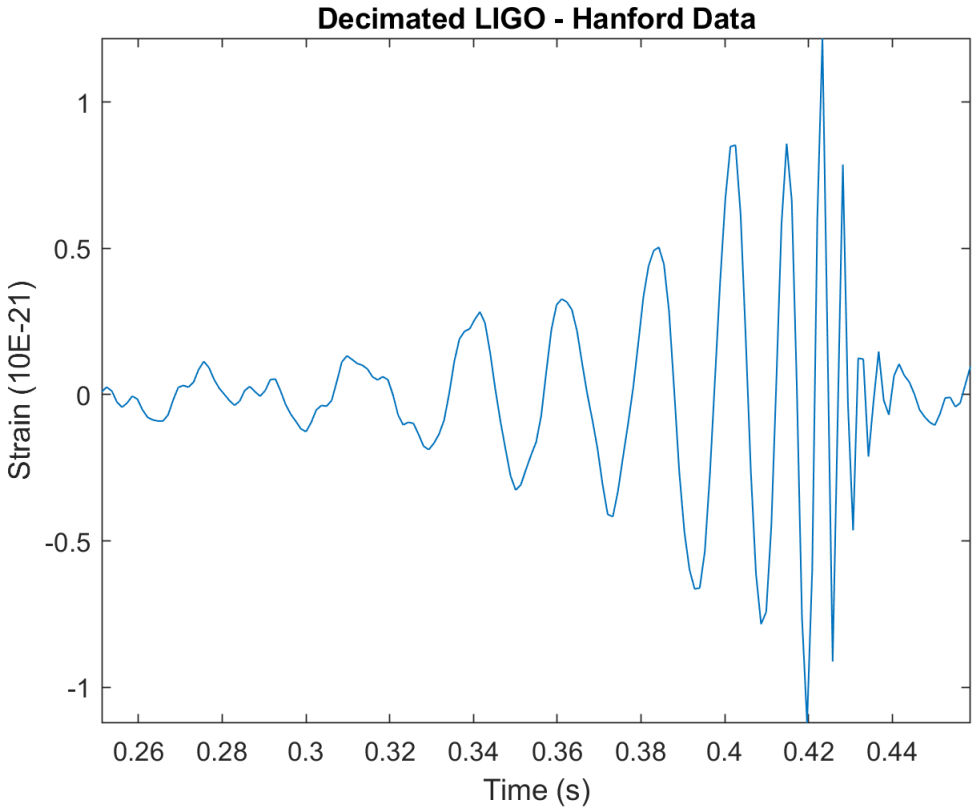



```
spectrogram(strain,rectwin(10),5,n,1/dt,'yaxis')
Xlim = get(gca, 'xlim');
set(gca, 'XTick', linspace(Xlim(1), Xlim(2), 5));
set(gca, 'XTicklabel', time(1):T/4:t(n));
```



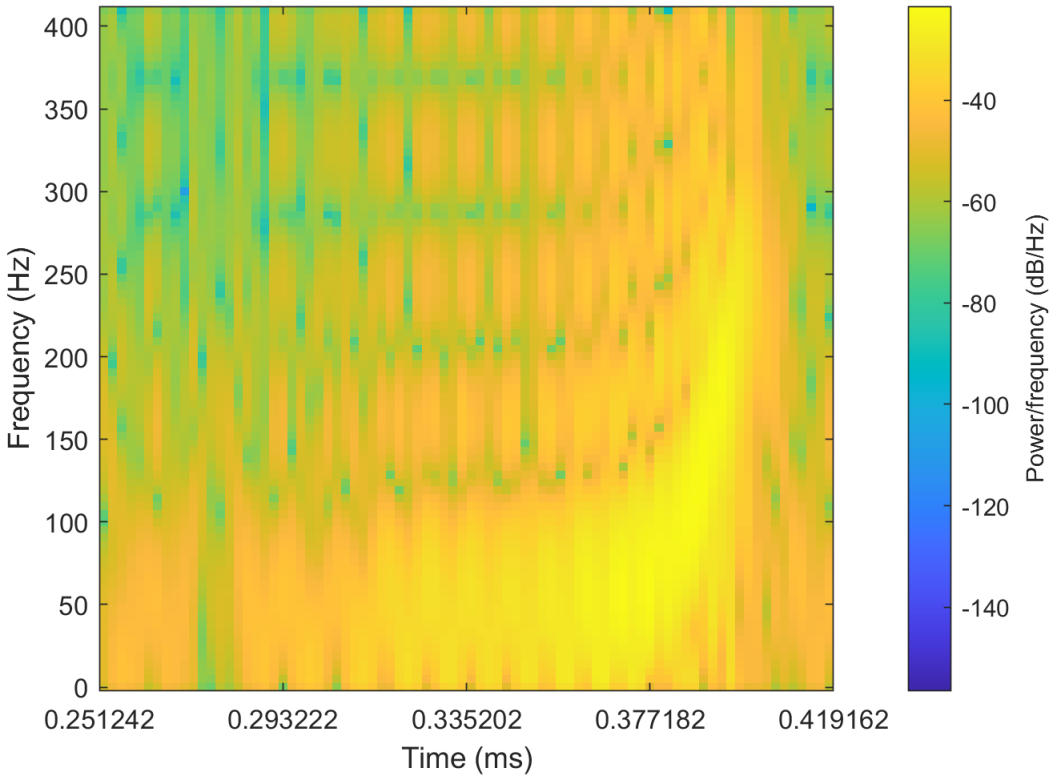
Decimation filters the data with a lowpass filter and resamples the signal at a lower rate.

```
decN=20;
y2 = decimate(strain,decN);
tt=decimate(time,decN);
N2=length(tt);
plot(tt,y2)
xlabel('Time (s)')
ylabel('Strain (10E-21)')
axis([tt(1), tt(N2), min(y2), max(y2)])
title('Decimated LIGO - Hanford Data')
```



Spectrogram of decimated data.

```
spectrogram(y2,rectwin(10),8,n/decN,1/dt/decN,'yaxis')
Xlim = get(gca, 'xlim');
set(gca, 'XTick', linspace(Xlim(1), Xlim(2), 5));

set(gca, 'XTicklabel', tt(1):T/5:tt(N2-1));
```



End of Examples

Here is fanalfun (fanalf in the project).

```
function z=fanalfun(y,T)
%
% FFT Analysis Function
%
% Enter Data in y and record length T
% Example:
%    n=128;
%    T=5;
%    dt=T/n;
%    f0=6.2;
%    f1=10;
%    fanal(sin(2*pi*f0*(1:n)*dt)+2*sin(2*pi*f1*(1:n)*dt));
% or
%    fanal(sin(2*pi*6.2*(1:128)/128*5),5);
n=length(y);
dt=T/n;
Y=fft(y,n);
n2=floor(n/2);
Power=Y.*conj(Y)/n^2;
f=(0:n2)/(n2*2*dt);
z=Power;
```

```
stem(f,2*Power(1:n2+1))
xlabel('Frequency')
ylabel('Power')
title('Periodogram')
end
```