

DIGITAL SIGNAL PROCESSING

% A signal can be pictured as the resulting wave form received by an electromagnetic, sonar or other detector. For example: A sinusoid pattern. More technically, given an analog (continuous) signal as a function of time, we say we have a digital signal by sampling the analog at a given interval of time say T. Thus we will have values of the signal for times 0,T,2T,3T..... These discrete values are the digital signal.

Once we have an equation for the signal we say they are deterministic. In most applications we study these types of signals. In nature, we also have to deal with Random signals, like noise on a communications line (static on the radio) and can model these with a "random" function. We use a random function to test DSP techniques. In MATLAB we use the function **rand**.

» help rand

RAND Random numbers and matrices.... RAND with no arguments is a scalar whose value changes each time it is referenced.

Ordinarily, random numbers are uniformly distributed in the interval (0.0,1.0). RAND('normal') switches to a normal distribution with mean 0.0 and variance 1.0. RAND('uniform') switches back to the uniform distribution.

RAND('seed') returns the current value of the seed for the generator. RAND('seed',n) sets the seed to n. RAND('seed',0) resets the seed to 0, its value when MATLAB is first entered.

%TYPICAL BEHAVIOR OF 'rand' FUNCTION

```
» rand
ans = 0.2190
» rand
ans = 0.0470
» rand
ans = 0.6789
```

% Since **rand** normally returns numbers between 0 and 1 we can stretch the range of random values by clever formulas. consider random values % between 0 and 5 obtained as in

```
» 5*rand
ans = 3.3965
» 5*rand
ans = 4.6735
» 5*rand
ans = 1.9175
```

% Random range -5 to +5

```
» 10*rand-5
ans = 0.1942
» 10*rand-5
ans = 3.3097
» 10*rand-5
```

```

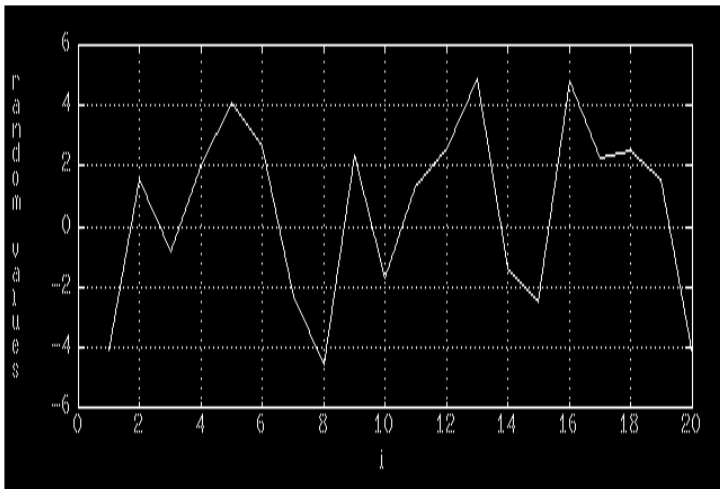
ans = -4.6543
» 10*rand-5
ans = -4.4654
% Building a vector of random numbers
» for i=1:20
y(i)=10*rand-5
end
y = Columns 1 through 7
-4.0804  1.5392  -0.8400  2.0119  4.1032  2.6220  -2.3755
   Columns 8 through 14
-4.5254  2.3608  -1.7177  1.3264  2.5641  4.9104  -1.3466
   Columns 15 through 20
-2.5296  4.8255  2.2266  2.5336  1.5152  -4.2731

```

```

% We can see this purely Random Signal by graphing
» plot(y)

```



% When working with random signals, we bring in statistically derived concepts to define various aspects for these signals.

For example **MEAN or average value...**

% MATLAB "mean" does the trick with

```
» AVERAGE = mean(y)
```

```
AVERAGE = 0.5425
```

% also Theory average:

```
» THEORY_AVERAGE=(min(y)+max(y))/2
```

```
THEORY_AVERAGE = 0.1925
```

% Variance and standard deviation are also important

```
» help std
```

STD Standard deviation. For vectors, STD(x) returns the standard deviation....

```
» help cov
```

COV Covariance matrix.

COV(X), if X is a vector, returns the variance....

```
» VARIANCE =cov(y)
```

```
VARIANCE = 9.0588
```

```
» STANDARDDEV=std(y)
STANDARDDEV = 3.0098
```

```
» Theoryvar=(max(y)-min(y)).^2 /12
Theoryvar = 7.4194
```

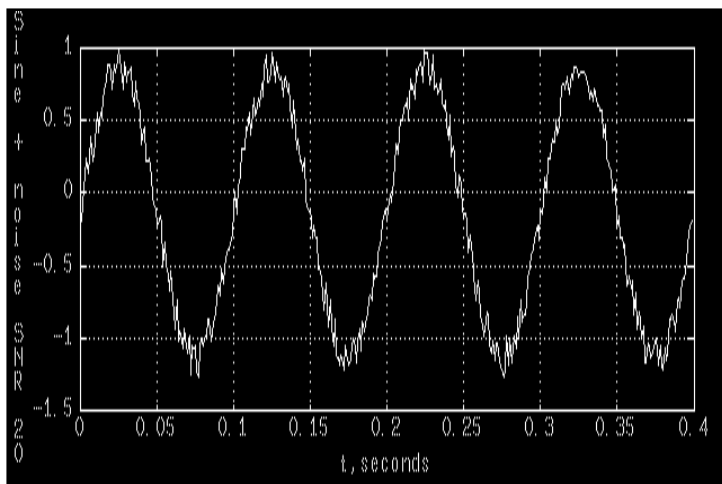
SIGNALS WITH NOISE

% We want to generate signals with noise to test many DSP applications. We usually specify the signal-to-noise ratio or SNR which is the ratio of signal power to noise power. The ratio is used to predict the amplitude of the noise so that we can get a signal +noise plot or formula. The ratio is used to decide if our signal is sufficient to get information to a reasonable degree of precision. % A typical M-file to generate such a combined signal is
» type spn

```
t=0;
i=1;
while t<=.4
    noise = 0.274.*rand-0.274;
    SPN(i)=sin(2.*pi.*10.*t)+noise;
    x(i)=t;
    t=t+.001;
    i=i+1;
end;

plot(x,SPN);
xlabel('t,seconds');
ylabel('Sine + noise SNR 20');
grid;
```

% This M-file contains the information that our original "no-noise" signal was a sinusoid at 10 Hz and amplitude of 1.0. The noise is a SNR of 20 which by theoretical considerations leads to a noise amplitude of approximately 0.274 ie. Noise varies from -0.274 to +0.274 amplitude. The resulting plot is as follows



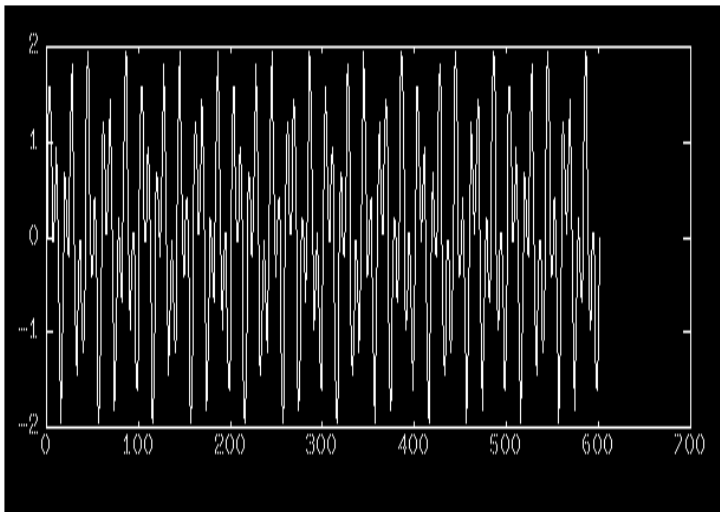
SIGNAL ANALYSIS

% Once we have a signal with noise we usually want to find the original signal without the noise. Finding the signal in the last case is relatively easy, but consider the following MATLAB code and resulting Generated graph of Signal plus Noise(using 'rand' below) NOTE: HOW we generate the vector (SIGNAL) y here as opposed to what we did in the m-file above.

```
» t=0:.001:.6;  
» x=sin(2*pi*50*t)+sin(2*pi*120*t);
```

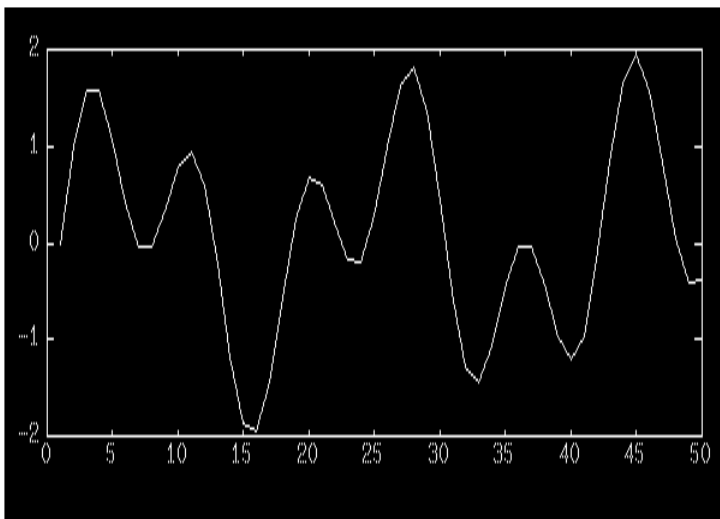
% Here x is a signal consisting of a 50 Hz and 120 Hz component which if they had no noise would look like over the full range

```
» plot(x)
```



% We can get a better idea of this combined frequency by the magnifying effect of plotting a smaller range of value as in

```
» plot(x(1:50))
```



% It can be seen that there is a pattern.
% but when we add noise with

```

» rand('normal');
» y = x + 2*rand(t);

```

```

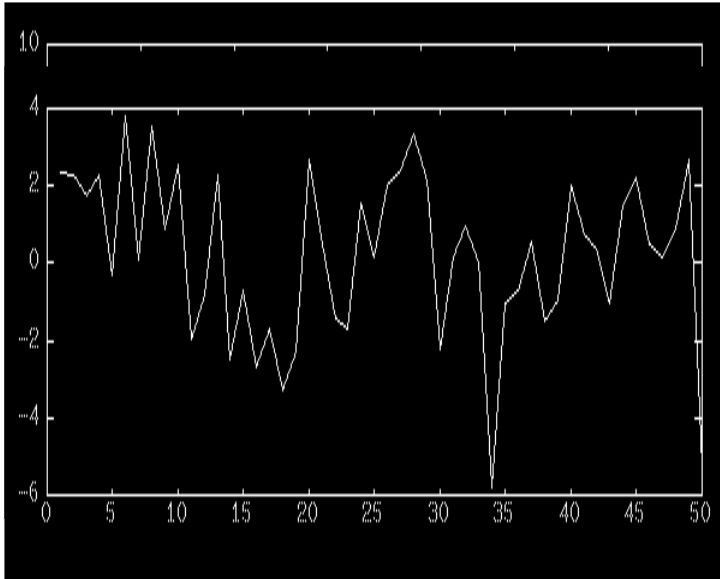
% NOTE: use of rand(t) see help rand to see that rand(t) returns a scalar
since t is a scalar
% The signal over the entire range now looks like

```

```

» plot(y)

```



```

% We can try to magnify with
plot(y(1:50))

```

```

% It can be seen by this last plot that the frequency components are
difficult to identify.

```

```

% We can extract the original frequencies by a technique called Fourier
Transforms.. MATLAB provides some functions to do this. Especially useful is the
fft() (fast Fourier transform) function.
(In fact there is an entire library of routines for DSP in general)
» help fft

```

```

FFT  FFT(X) is the discrete Fourier transform of vector X...
      FFT(X,N) is the N-point FFT....

```

```

% To find the original frequencies above we proceed as follows
First, we convert to the frequency domain, the discrete Fourier transform "Y" of
the noisy signal "y" above is found by taking the 512 point fast
% Fourier transform(FFT) with

```

```

» Y = fft(y,512);

```

```

% Y is a complex vector of 512 rows

```

```

» Y

```

```

Y = 1.0e+002 *

```

```

Columns 1 through 4

```

```

0.5305      -0.0016 + 0.1593i   0.1152 - 0.2822i  -0.0468 + 0.3554i

```

```

Columns 5 through 8

```

```

0.1512 - 0.4357i  0.1457 + 0.2767i  0.4708 + 0.1943i  -0.1518 + 0.1173i

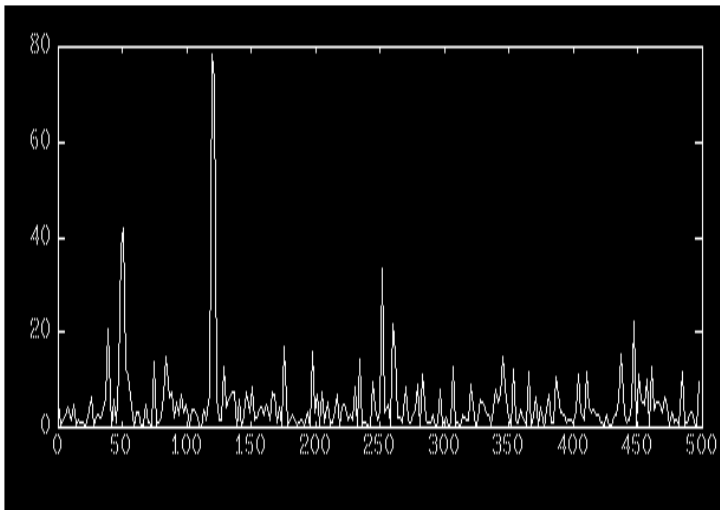
```

```

.....
% We determine next the Power Spectrum density which is a measurement of
energy at various frequencies and is defined as the real values
» Psd =Y.*conj(Y)/512;
% NOTE
» help conj
CONJ CONJ(X) is the complex conjugate of X.
Psd =
Columns 1 through 7
    5.4977    0.4955    1.8152    2.5096    4.1540    1.9099    5.0665
Columns 8 through 14
    0.7189    1.4510    1.1974    1.2711    0.3606    2.2991    6.2825
.....
% We plot only the first 256 points since the other 256 points are
symmetric about the middle.
% TRY PLOTTING "Psd"

% We choose a meaningful frequency "f" for the x axis
and then plot as follows
» f=1000*(0:255)/512;
» plot(f,Psd(1:256))

```



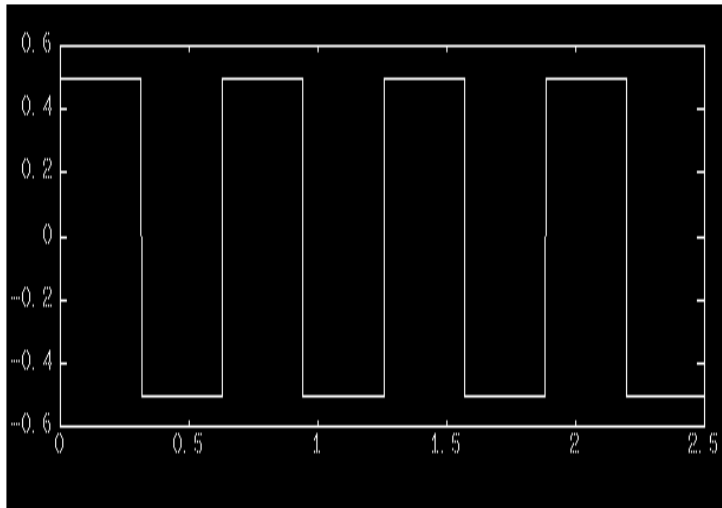
% One sees from the plot the original 50 Hz and 120 Hz signals stand out clearly which is what we set out to do in the first place!

SQUARE WAVE

% Another type of important signal is the square wave and MATLAB's **Signal Processing Toolbox** section has the 'square' function to generate it. One application is to Digital devices, which are usually driven over a clock cycle to be at High or Low Voltage states

```
% A signal that has amplitude 0.5 is illustrated as follows
» t=0:.001:2.5;
» y=0.5*square(10*t);
```

```
» plot(t,y)
```



5

% LABORATORY TASK 42 (20 pts)

STARTING WITH THE DIGITAL SIGNAL PROCESSING SECTION ABOVE
REPRODUCE ALL CURVES STUDYING THE TECHNIQUES CAREFULLY!

Given a complicated **noisy** radio signal from a distant galaxy whose values are contained in the function $R(t)$ and are in the frequency range of 0 to 500 megacycles. $R(t)$ can be represented algebraically by

$$R(t) = .3\sin(678t) + .4\sin(986t) + \text{noise}$$

$\text{noise} = 0.3\text{rand} - 0.3$

with t from 0 to 0.4

Write the MATLAB code to obtain the **Power Spectrum Density** in a similar manner to above in order to separate out any strong signals that may be present. What are they if any?