EE 484

Homework 2 Solutions

Note: For any Matlab exercises you should submit your Matlab code along with the rest of your solution to the given problem.

Note: Code is provided for some of the problems in the solutions below. It is example code only; there are many ways to solve each problem (and likely more efficient solutions than those presented below).

Problem 1. This is a Matlab exercise. Consider the carrier signal, denoted as $s(t) = A \cdot sin(2\pi f_c t)$ and a message waveform $m(t) = M \cdot \cos(2\pi f_m t + \phi)$. Let $x(t) = [1 + m(t)] \cdot s(t)$ represent an amplitude modulated (AM) signal.

a. Let $t = n\Delta t$, $f_c = 1$ Hz, $f_m = 0.1$ Hz, and $\varphi = \pi/2$, then generate a plot of $x(n) = x(n\Delta t)$ in Matlab for A = 1.0 and M = 0.5, M = 1.0, and M = 1.5. Plot x(n) for five cycles of the message waveform in each case.

   Solution: Example code is below. The code is shown for one value of the modulation index, M, only. Plots for each of the three values of M are also shown below.

```
% AM Modulation problem. HW4 Problem 1 for EE484.

M = 0.5;                                % Amplitude of modulating waveform
f_c = 1;                                % Carrier frequency
f_m = 0.10;                             % Message frequency
phi = pi/2;                             % Message phase

N = 1000;                               % Number of time samples
A = 1.0;                                % Carrier amplitude
num_cycles = 5;                         % Number of cycles of the low-rate
                                        % modulation tone to model
delta_t = 1.0/(N*f_m/num_cycles);       % Sample step size in time
max_f = (1/delta_t)/2.0;                % Largest frequency

t = [1:N] .* delta_t;                   % Create vector of time samples
unity = ones(1,N);                      % Create vector of 1's

s = A * sin(2.0 * pi * f_c * t);        % Create carrier signal vector
plot(s);

m = M * cos(2.0 * pi * f_m * t + phi); % Create message signal vector
plot(m);

x = [unity + m] .* s;                   % Create amplitude modulated signal vector

plot(x);
xlabel('Time(samples)');
ylabel('Amplitude');
figure;
```

```
X = (1.0/N) * fft(x');          % Take FFT of amplitude modulated signal

f_axis = [0:N-1] .* 2*max_f/N;          % Build frequency axis for plot,

plot(f_axis(1:100), abs(X(1:100)));
xlabel('Frequency');
ylabel('Magnitude');
```

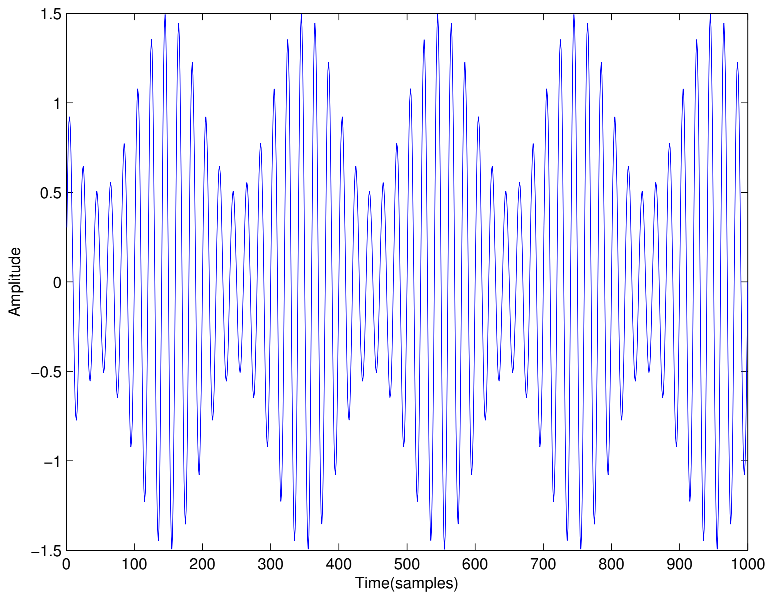

Figure 1: x(n) for M=0.5

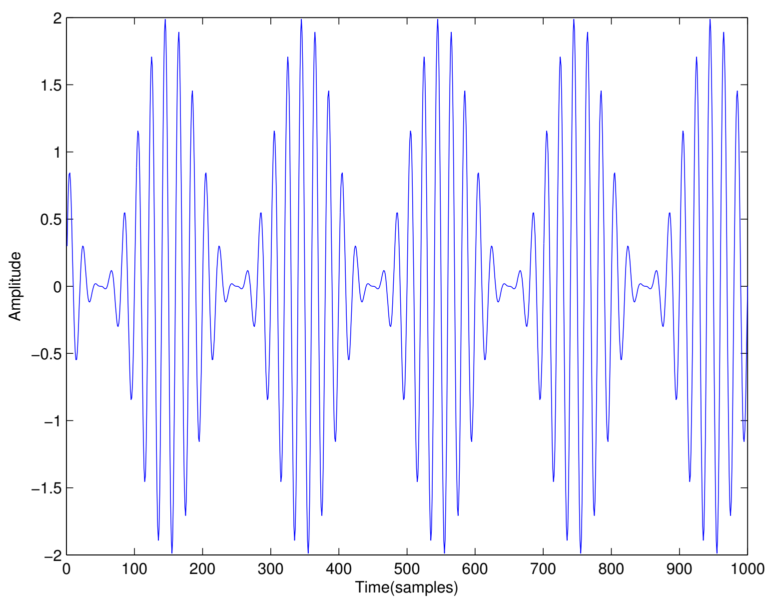

Figure 2: x(n) for M=1.0

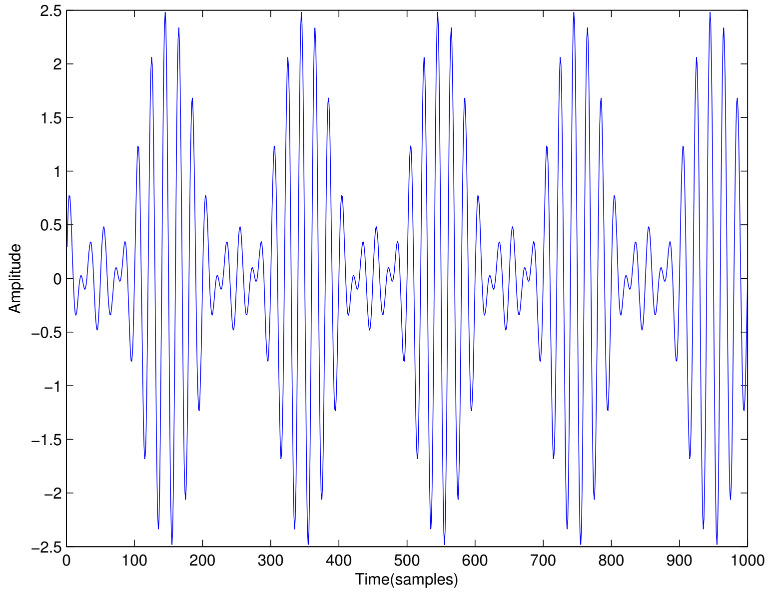

Figure 3: x(n) for M=1.5

b. Derive the Fourier Transform of x(t) analytically.

Solution: $X(f) = jA/2 \cdot (\delta(f + f_c) - \delta(f - f_c)) + jAM/4[e^{-j\phi}\delta(f + (f_c + f_m)) - e^{-j\phi}\delta(f - (f_c + f_m))] + jAM/4[e^{-j\phi}\delta(f + (f_c - f_m)) - e^{-j\phi}\delta(f - (f_c - f_m)]$

c. Generate and plot the magnitude of the Fourier Transform of x(n) in Matlab for the three cases of M identified in part a. Include the frequency axis in each plot.

Solution: See plots of $\mid X(k) \mid$.

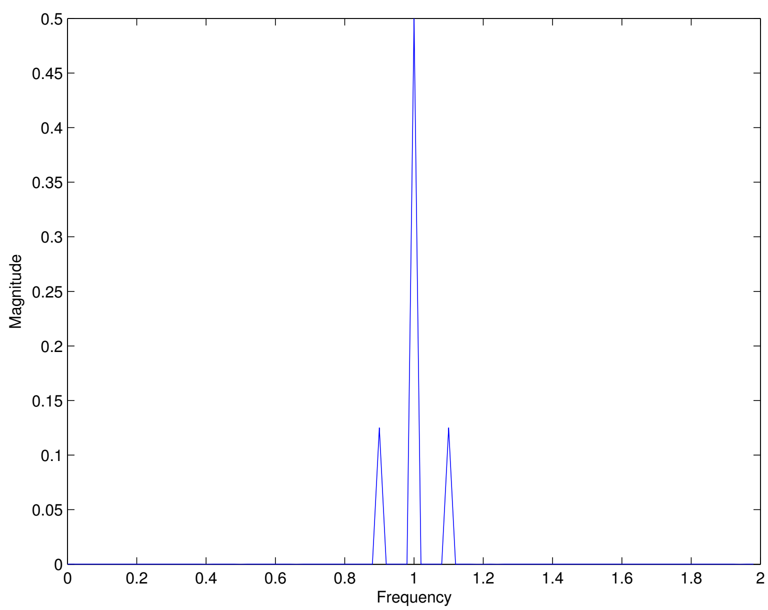

Figure 4: $\mid X(k) \mid$, the magnitude of the FFT of x(n) for M=0.5

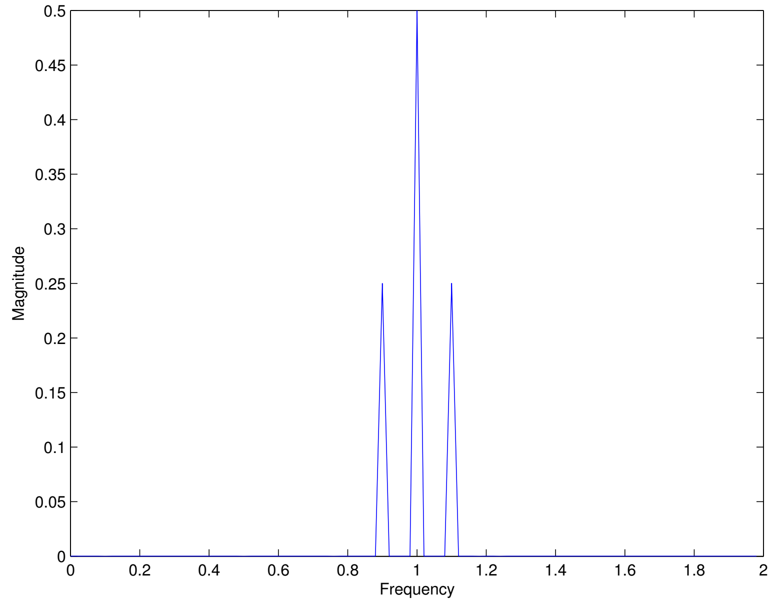

Figure 5: $\mid X(k) \mid$, the magnitude of the FFT of x(n) for M=1.0

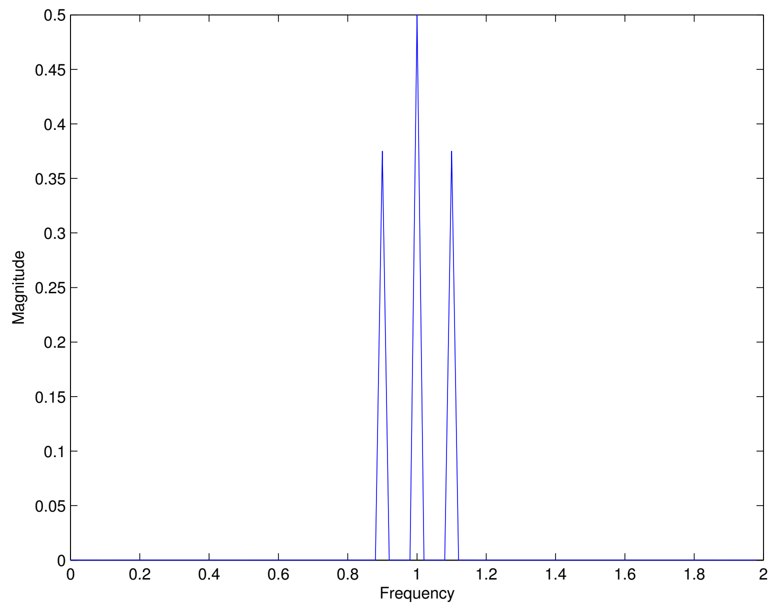

Figure 6: $\mid X(k) \mid$, the magnitude of the FFT of x(n) for M=1.5

d. Which of the three values of M is most suitable for transmission of information and why?

Solution: The modulation index for an AM waveform is defined as $\frac{M}{A}$. Amplitude modulated signals are typically demodulated via envelope detection. Consider the plots of $x(n)$ shown below, for $M = 0.5, 1.0, 1.5$, in which the envelope, corresponding to the message waveform, $m(t)$ is highlighted. For the case of $M = 0.5$, this envelope signal is readily recovered or demodulated. For the case of $M = 1.0$ the signal can also be recovered in a straightforward manner, though, from a practical perspective, it would be challenging to control a more random modulating waveform (such as voice or music, for example), so that the envelope never crossed zero. If the modulating signal, $m(t)$, causes the envelope of $s(t)$ to cross zero, then distortion of the received signal can occur. Though the waveform can be demodulated in a coherent manner, as mentioned above, envelope detection is often used. In other words, the red line in each of the three plots is detected. From this, it can be seen that for the case of M=1.5 (as well as any case in which the modulation index is greater than 1.0), the envelope is distorted due to this "overmodulation". Note the reversal of the carrier phase for this case when the envelope crosses zero, and the associated sharp transition in the envelop, which would introduce distortions, as discussed in class.

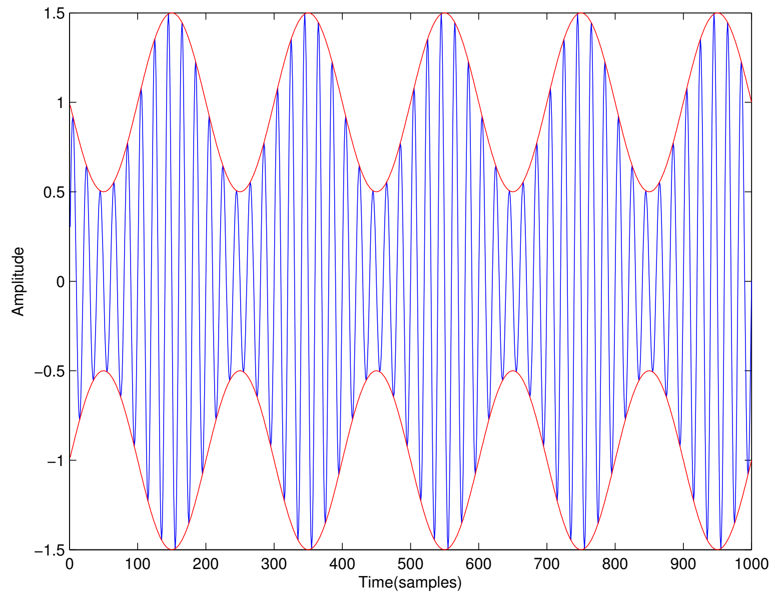

Figure 7: $x(n)$ for M=0.5

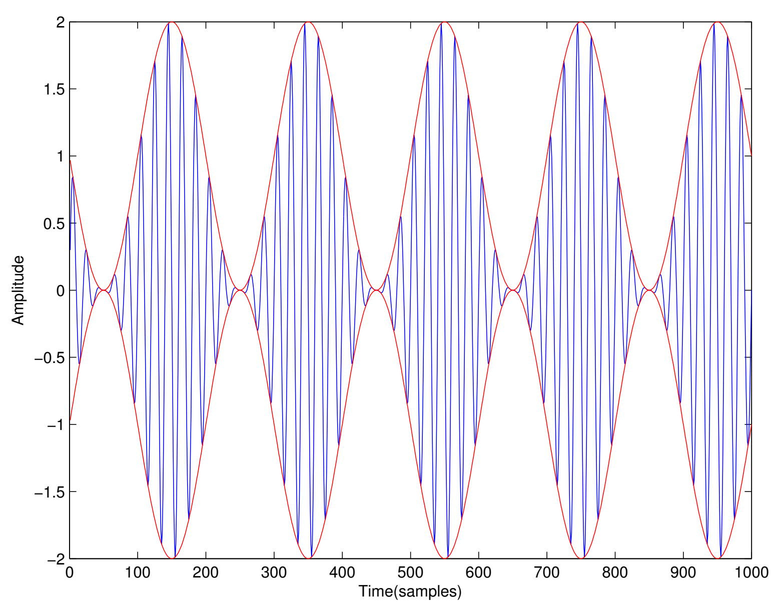

Figure 8: $x(n)$ for M=1.0

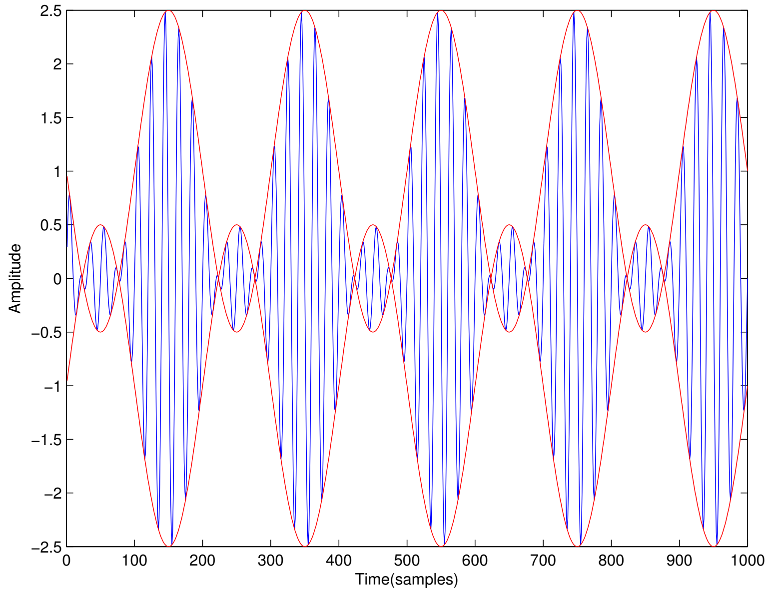

Figure 9: $x(n)$ for M=1.5

Problem 2. This is a Matlab exercise.

a. Generate the complex signal $s(n) = A \cdot cos(2\pi f_c \Delta t \cdot n) + jA \cdot sin(2\pi f_c \Delta t \cdot n)$, for $f_c = 1$, A=1.0, $\Delta t = 0.01$, and $1 \leq n \leq 10,000$. Plot this signal with the in-phase values on the x-axis and the quadrature components on the y-axis. What is the magnitude of this signal?
   Solution: See below for plot. Magnitude is 1, since this is just a complex exponential. Example code for the problem is also below.

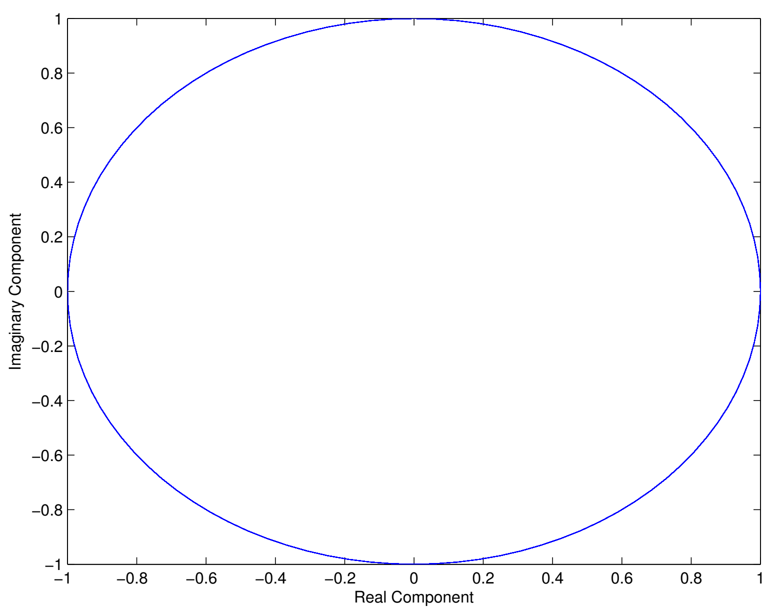

Figure 10: Plot for problem 2a

```
% Signal in noise visualization problem. HW4 problem 2 for EE484.
```

```
A = 1.0;                                          % Amplitude of modulating waveform (divide by A t
f_c = 1;                                          % Carrier frequency
delta_t = 0.01;                                   % Sample step size in time
phi = pi/2;                                       % Message phase

N = 10000;                                        % Number of time samples
delta_f = 1/delta_t;                              % Sample step size in frequency
t = [1:N] .* delta_t;                             % Create vector of time samples

s_Q = A * sin(2.0 * pi * f_c * t);                % Create quadrature carrier signal vector
s_I = A * cos(2.0 * pi * f_c * t);                % Create in-phase carrier signal vector
s = s_I + 1i*s_Q;                                 % Create complex signal

plot(s);
xlabel('Real Component');
ylabel('Imaginary Component'); figure;

sigma_1 = sqrt(0.1);                              % Set the standard deviation for random signal 1
sigma_2 = sqrt(0.5);                              % Set the standard deviation for random signal 2
sigma_3 = sqrt(1.0);                              % Set the standard deviation for random signal 3

k_1 = (randn(1,N) + 1i*randn(1,N)) * sigma_1/sqrt(2.0);     % Generate random sequence 1
k_2 = (randn(1,N) + 1i*randn(1,N)) * sigma_2/sqrt(2.0);     % Generate random sequence 2
k_3 = (randn(1,N) + 1i*randn(1,N)) * sigma_3/sqrt(2.0);     % Generate random sequence 3

plot(k_1);
xlabel('Real Component');
ylabel('Imaginary Component'); figure;

var_1 = var(k_1);                                 % Check the variances of the generated complex se
var_2 = var(k_2);
var_3 = var(k_3);

[dist_1, M_1] = hist(abs(k_1),100);
[dist_2, M_2] = hist(abs(k_2),100);
[dist_3, M_3] = hist(abs(k_3),100);

mean_1 = mean(k_1);
mean_2 = mean(k_2);
mean_3 = mean(k_3);

var_1 = var(k_1);
var_2 = var(k_2);
var_3 = var(k_3);

plot(M_1, dist_1); hold on;
plot(M_2, dist_2, '-g');
plot(M_3, dist_3, '-c');
xlabel('Magnitude');
ylabel('Occurences');
legend('\sigma = 0.1', '\sigma = 0.5', '\sigma = 1.0'); hold off; figure;

r_1 = s+k_1;
r_2 = s+k_2;
r_3 = s+k_3;
```

```
mag_r_1 = abs(r_1); % Generate magnitude vectors for signal plus noise
mag_r_2 = abs(r_2);
mag_r_3 = abs(r_3);

[dist_1, X_1] = hist(mag_r_1,100);  % Generate histograms
[dist_2, X_2] = hist(mag_r_2,100);
[dist_3, X_3] = hist(mag_r_3,100);

plot(X_1, dist_1); hold on;         % Generate distributions for signal plus noise
plot(X_2, dist_2, '-g');
plot(X_3, dist_3, '-c');
xlabel('Magnitude');
ylabel('Occurences');
legend('\sigma = 0.1', '\sigma = 0.5', '\sigma = 1.0'); hold off;
```

b. Let $r_1(n) = s(n) + k_1(n)$, $r_2(n) = s(n) + k_2(n)$, and $r_3(n) = s(n) + k_3(n)$ be the complex signal plus noise vectors for each of the three difference noise sequences calculated above. Calculate the magnitude of each of these signals, then generate the histograms as above. Describe the distribution that each of these signals follows. What distribution(s) does the phase of the signals follow?

Solution: See below for plots of the magnitude histograms for each case. These signals are drawn from a complex Gaussian distribution, with non-zero mean. It can be shown that this will follow the Rician distribution. For the combination of a complex Gaussian noise sequence and a sinusoid, the phase will not show a preference for any given value, therefore it will follow a uniform distribution, from $-\pi$ to π.

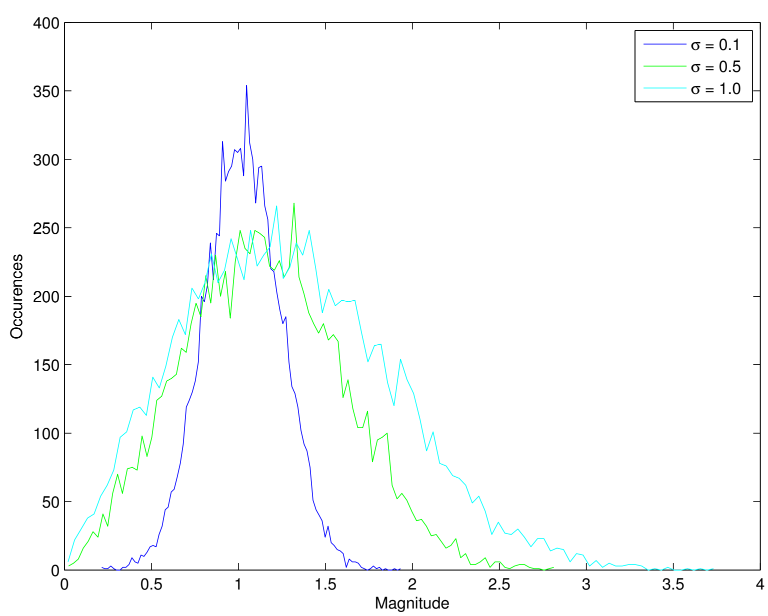

Figure 11: Plot for problem 2b