Cmpe362 First Homework Report

Berkay Kozan

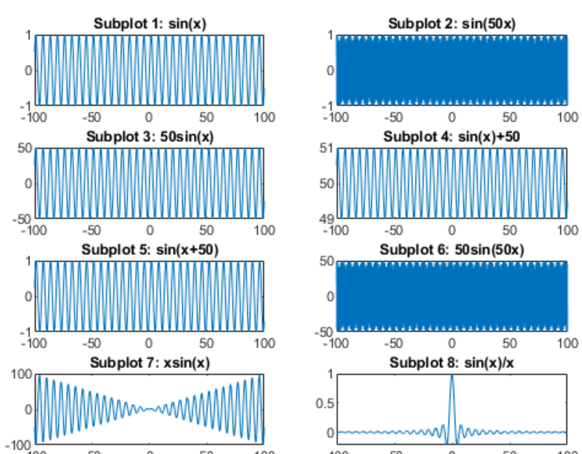
06 March 2019

1 Introduction

This is the first assignment I have in MATLAB. It was a nice assignment to learn MATLAB and see a new powerful tool. I saw that I don't need to write specifically functions and classes to work with code. You can just dive into code. And it is so straightforward, you can run the code and see the plots you created, and see if it is true or wrong quickly. As a disadvantage, I am a newbie to MATLAB, thus I had to search a lot from Google. As advantage, I felt MATLAB is a strong mathematical tool that can make intangible words to a tangible shape.

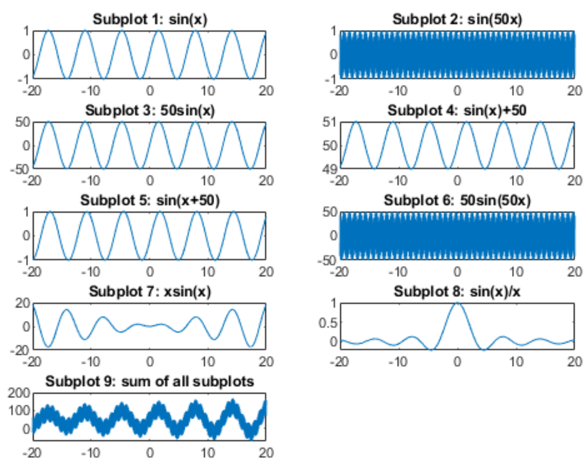
2 Question 1

As we can see from the image, when frequency increases, the wavelength is shorter, and graph gets more dense. When you multiply the image by a constant, the shape just gets longer, the frequency and wavelength stays constant. If we compare 4th subplot to 1st, adding 50 just changed y values, in fact they're increased by 50. When comparing 5th subplot to 1st, we see the change of phase. The shape is a shifted version of 1st by 50 degrees. In 6th subplot, we see a dense graph as 2nd. Multiplying 2nd with 50 made the shape 50 times longer by y axis. 7th shape is kind of special, no shape is like it. It goes like a damped sinusoid oscillation, when approached to 0, y value is decreased. 8th shape is also special, it again has oscillation but increasing this time. 0 is the peak point, although the 0 point is nothing, since we can't divide 0 by 0.



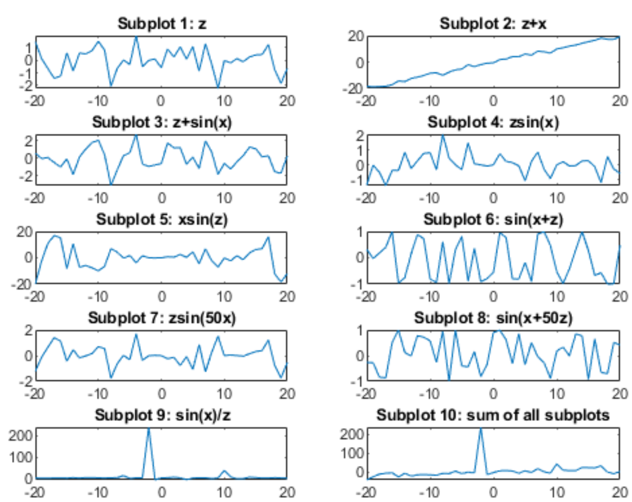
3 Question 2

It is very similar to first question, only the x axis is more narrow. Shapes are pretty similar, and sum of all plots looks like sinusoidal wave.



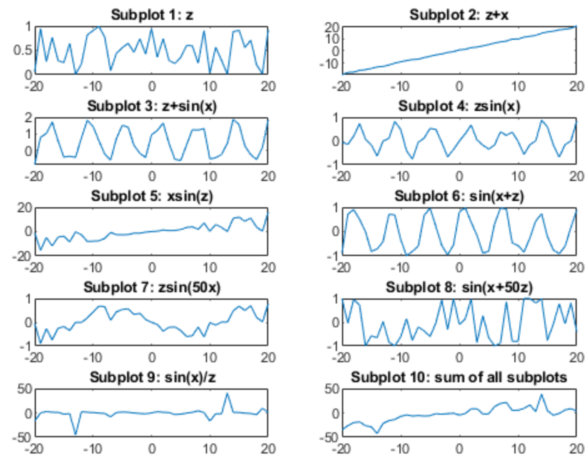
4 Question 3

Z is Gaussian random number, but when we add it with x, we see a pretty orderly increasing graph. Rest of the graphs looks a random mess. Sum is highly influenced by 9th subplot, since it has the biggest values.



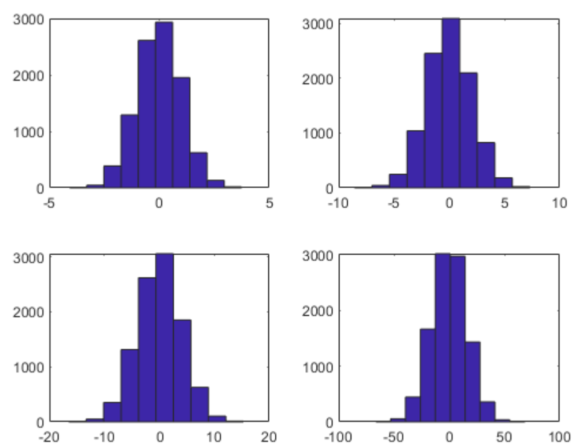
5 Question 4

Z is uniform random number, but when we add it with x, we see a pretty orderly increasing graph. Rest of the graphs looks a random mess. Sum is highly influenced by 9th subplot, since it has the biggest values. It is almost identical to Question 3.



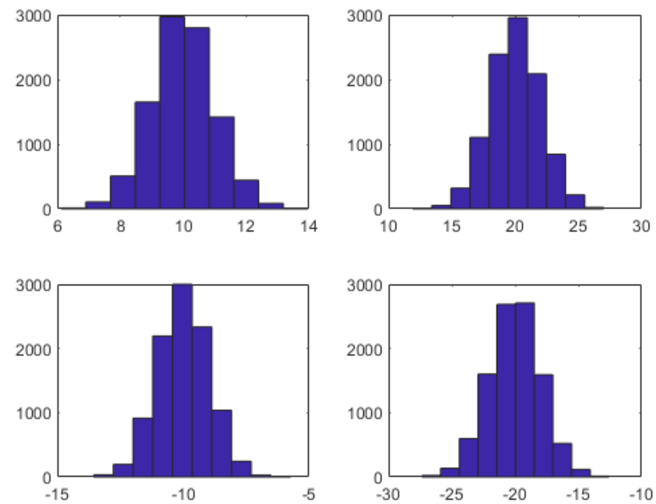
6 Question 5

I see when variance increases, the same values are achieved by a longer range of x-axis. They all have 0 mean.



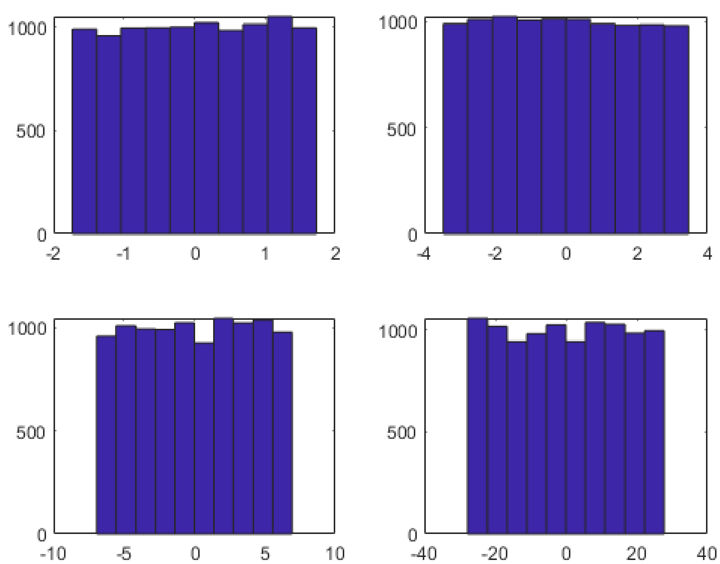
7 Question 6

I see when variance increases, the same values are achieved by a longer range of x-axis again. The mean is the middle number below the graph



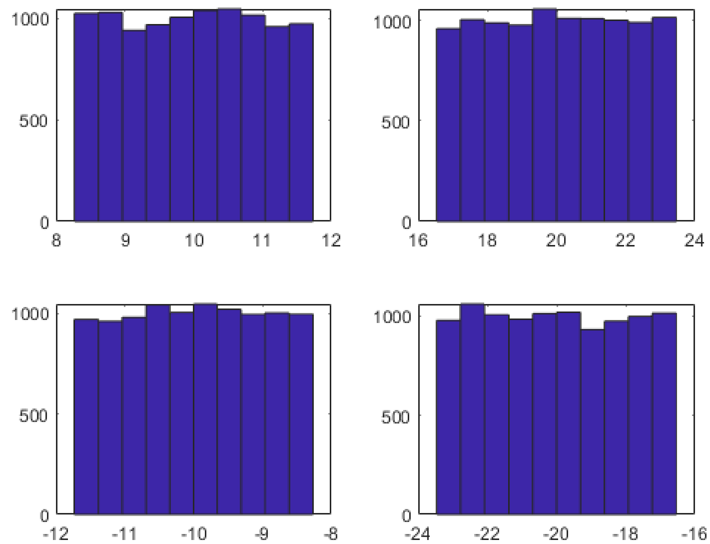
8 Question 7

Since they are all uniformly distributed, the length of all sticks are almost equal. Their mean is 0.



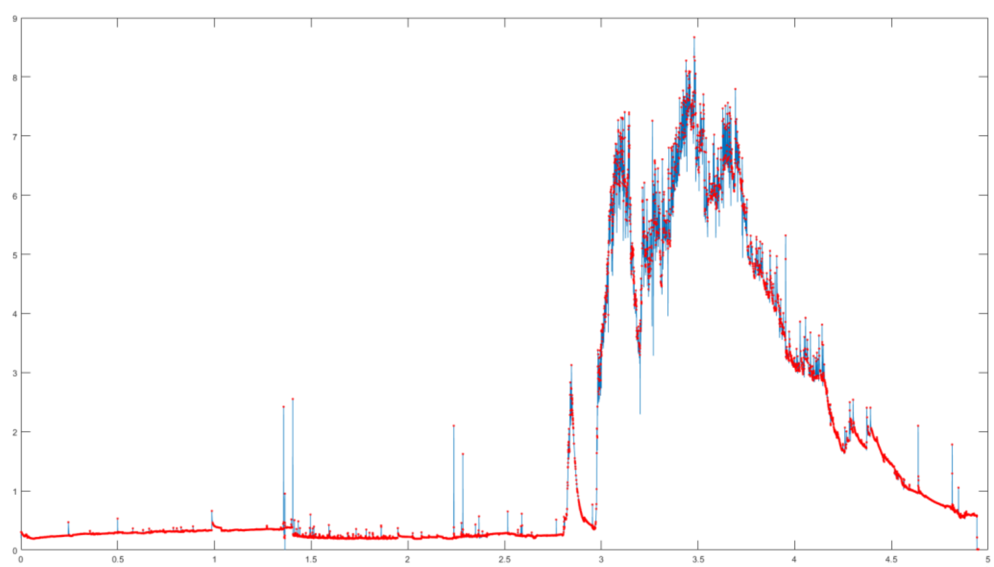
9 Question 8

Since they are all uniformly distributed, the length of all sticks are almost equal. Their mean is the middle number below the graph.



10 Question 9

There are red points on all local maxima's. However, the peaks are all maxima. One cannot see lower peaks. Also, when there is a straight line, there is only one red mark, we see only a maximum.



11 The code - signal and noise

```
%---------------------------PROBLEM 1---------------------------
figure(1);
subplot(4,2,1);
x = linspace(-100,100,1000); %1000 is the number of points in this plot. I selected a big number
y1 = sin(x);                 %so that the shape goes smoothly. The vector starts from -100, ends in 100.
plot(x,y1)
title('Subplot 1: sin(x)')

subplot(4,2,2);
y2 = sin(50*x);
plot(x,y2)
title('Subplot 2: sin(50x)')

subplot(4,2,3);
y3 = 50*sin(x);
plot(x,y3)
title('Subplot 3: 50sin(x)')

subplot(4,2,4);
y4 = sin(x)+50;
plot(x,y4)
title('Subplot 4: sin(x)+50')

subplot(4,2,5);
y5 = sin(x+50);
plot(x,y5)
title('Subplot 5: sin(x+50)')

subplot(4,2,6);
y6 = 50*sin(50*x);
plot(x,y6)
title('Subplot 6: 50sin(50x)')

subplot(4,2,7);
y7 = x.*sin(x);
plot(x,y7)
title('Subplot 7: xsin(x)')

subplot(4,2,8);
y8 = sin(x)./x;
plot(x,y8)
title('Subplot 8: sin(x)/x')

%---------------------------PROBLEM 2---------------------------
figure(2);
x = linspace(-20,20,1000); %1000 is the number of points in this plot. I selected a big number
subplot(5,2,1);            %so that the shape goes smoothly. The vector starts from -100, ends in 100.
```

```
y1 = sin(x);
plot(x,y1)
title('Subplot 1: sin(x)')

subplot(5,2,2);
y2 = sin(50*x);
plot(x,y2)
title('Subplot 2: sin(50x)')

subplot(5,2,3);
y3 = 50*sin(x);
plot(x,y3)
title('Subplot 3: 50sin(x)')

subplot(5,2,4);
y4 = sin(x)+50;
plot(x,y4)
title('Subplot 4: sin(x)+50')

subplot(5,2,5);
y5 = sin(x+50);
plot(x,y5)
title('Subplot 5: sin(x+50)')

subplot(5,2,6);
y6 = 50*sin(50*x);
plot(x,y6)
title('Subplot 6: 50sin(50x)')

subplot(5,2,7);
y7 = x.*sin(x);
plot(x,y7)
title('Subplot 7: xsin(x)')

subplot(5,2,8);
y8 = sin(x)./x;
plot(x,y8)
title('Subplot 8: sin(x)/x')

subplot(5,2,9);
y9 = y1+y2+y3+y4+y5+y6+y7+y8;
plot(x,y9)
title('Subplot 9: sum of all subplots')

%---------------------------PROBLEM 3---------------------------
figure(3);
z= randn(1,41); % 41 Gaussian distributed random numbers are created.
x = linspace(-20,20,41); %The vector starts from -20, ends in 20. There must
be
subplot(5,2,1);         %41 points, so that I can do operations on x and z.
y10 = z;
plot(x,y10)
title('Subplot 1: z')

subplot(5,2,2);
```

```
y11 = z+x;
plot(x,y11)
title('Subplot 2: z+x')

subplot(5,2,3);
y12 = z+sin(x);
plot(x,y12)
title('Subplot 3: z+sin(x)')

subplot(5,2,4);
y13 = z.*sin(x);
plot(x,y13)
title('Subplot 4: zsin(x)')

subplot(5,2,5);
y14 = x.*sin(z);
plot(x,y14)
title('Subplot 5: xsin(z)')

subplot(5,2,6);
y15 = sin(x+z);
plot(x,y15)
title('Subplot 6: sin(x+z)')

subplot(5,2,7);
y16 = z.*sin(50*x);
plot(x,y16)
title('Subplot 7: zsin(50x)')

subplot(5,2,8);
y17 = sin(x+(50*z));
plot(x,y17)
title('Subplot 8: sin(x+50z)')

subplot(5,2,9);
y18 = sin(x)./z;
plot(x,y18)
title('Subplot 9: sin(x)/z')

subplot(5,2,10);
y19 = y10+y11+y12+y13+y14+y15+y16+y17+y18;
plot(x,y19)
title('Subplot 10: sum of all subplots')

%----------------------------PROBLEM 4---------------------------
figure(4);
z= rand(1,41); % 41 uniformly distributed random numbers are created.
x = linspace(-20,20,41); %The vector starts from -20, ends in 20. There must
be
subplot(5,2,1);          %41 points, so that I can do operations on x and z.
y20 = z;
plot(x,y20)
title('Subplot 1: z')

subplot(5,2,2);
```

```
y21 = z+x;
plot(x,y21)
title('Subplot 2: z+x')

subplot(5,2,3);
y22 = z+sin(x);
plot(x,y22)
title('Subplot 3: z+sin(x)')

subplot(5,2,4);
y23 = z.*sin(x);
plot(x,y23)
title('Subplot 4: zsin(x)')

subplot(5,2,5);
y24 = x.*sin(z);
plot(x,y24)
title('Subplot 5: xsin(z)')

subplot(5,2,6);
y25 = sin(x+z);
plot(x,y25)
title('Subplot 6: sin(x+z)')

subplot(5,2,7);
y26 = z.*sin(50*x);
plot(x,y26)
title('Subplot 7: zsin(50x)')

subplot(5,2,8);
y27 = sin(x+(50*z));
plot(x,y27)
title('Subplot 8: sin(x+50z)')

subplot(5,2,9);
y28 = sin(x)./z;
plot(x,y28)
title('Subplot 9: sin(x)/z')

subplot(5,2,10);
y29 = y10+y11+y12+y13+y14+y15+y16+y17+y18;
plot(x,y29)
title('Subplot 10: sum of all subplots')

%----------------------------PROBLEM 5---------------------------
figure(5);
r1 = randn(1,10000); %Mean=0, variance=1
r2 = 2*randn(1,10000); %Mean=0, variance=4
r3 = 4*randn(1,10000); %Mean=0, variance=16
r4 = 16*randn(1,10000); %Mean=0, variance=256

% I multiplied randn with squareroot of variance, so that the variance of
% the sample increases variance times. randn function default creates a set
% with variance 1 and mean 0.
vectors = [r1;r2;r3;r4];
```

```
for i=1:4
    subplot(2,2,i);
    hist(vectors(i,:));
end
%---------------------------PROBLEM 6---------------------------
figure(6);
r6 = randn(1,10000)+10; %Mean=10, variance=1
r7 = 2*randn(1,10000)+20; %Mean=20, variance=4
r8 = randn(1,10000)-10; %Mean=-10, variance=1
r9 = 2*randn(1,10000)-20; %Mean=-20, variance=4

% I multiplied randn with squareroot of variance, so that the variance of
% the sample increases variance times. randn function default creates a set
% with variance 1 and mean 0. I manually added/substracted the amount of mean
we want
% to/from all random variables, so that the mean increases/decreases by that
amount.
vectors = [r6;r7;r8;r9];

for i=1:4
    subplot(2,2,i);
    hist(vectors(i,:));
end
%---------------------------PROBLEM 7---------------------------
figure(7);
r11 = sqrt(12)*(rand(1,10000)-0.5); %Mean=0, variance=1
r21 = sqrt(12)*2*(rand(1,10000)-0.5); %Mean=0, variance=4
r31 = sqrt(12)*4*(rand(1,10000)-0.5); %Mean=0, variance=16
r41 = sqrt(12)*16*(rand(1,10000)-0.5); %Mean=0, variance=256

vectors = [r11;r21;r31;r41];
   %We substract 0.5 since rand gives a number with mean 0.5. sqrt(12)
   %comes from the formula of variance.

for i=1:4
    subplot(2,2,i);
    hist(vectors(i,:));
end
%---------------------------PROBLEM 8---------------------------
figure(8);
r61 = sqrt(12)*(rand(1,10000)-0.5)+10; %Mean=10, variance=1
r71 = sqrt(12)*2*(rand(1,10000)-0.5)+20; %Mean=20, variance=4
r81 = sqrt(12)*(rand(1,10000)-0.5)-10; %Mean=-10, variance=1
r91 = sqrt(12)*2*(rand(1,10000)-0.5)-20; %Mean=-20, variance=4

vectors = [r61;r71;r81;r91];
   %We substract 0.5 since rand gives a number with mean 0.5. sqrt(12)
   %comes from the formula of variance.

for i=1:4
    subplot(2,2,i);
    hist(vectors(i,:));
    %stats1 = [mean(vectors(i,:)) std(vectors(i,:)) var(vectors(i,:))]
    %for seeing stats of each vector.
```

```
end
```

12 The code – Problem 9

```
%----------------------------PROBLEM 9----------------------------
figure(9);
M = csvread('exampleSignal.csv',3); %Read the file
plot(M) %Plot the signal itself.
hold on %Wait for the peaks to enter.
[pks,locs] = findpeaks(M); %pks are peaks, locations are locations of peaks.
plot(locs,pks,'.r') %Add the peaks from findpeaks method to the same figure
with color red.
```

13 The code – Problem 10

```
%----------------------------PROBLEM 10----------------------------
Image=imread('lena.png'); %Take the image as input.
Image=rgb2gray(Image); %Convert the image from rgb to gray picture.

%Mean of the Gray image. The variable mean is the mean of the image input.
mean=mean2(Image);

%Standart deviation of the Gray image. The variable std is the standart
deviation of the image input.
std=std2(Image);

%the maximum element of the image file
max1= max(Image); %max1= 1x512 array.
max2= transpose(max1); %Flip the matrix, so that we can use max method once
again.
max=max(max2); %The maximum element is called max.

%location of the maximum element.
[max_row, max_column] = find(Image == max); %max_row is the row of the
maximum element.
%max_column is the column of the maximum element.

%the minimum element of the image file
min1= min(Image); %min1= 1x512 array.
min2= transpose(min1); %Flip the matrix, so that we can use min method once
again.
min=min(min2); %The minimum element is called min.

%location of the minimum element.
[min_row, min_column] = find(Image == min); %min_row is the row of the
minimum element.
%min_column is the column of the minimum element.
```