ECE 438 Lab 1 Report

David Dang & Benedict Lee

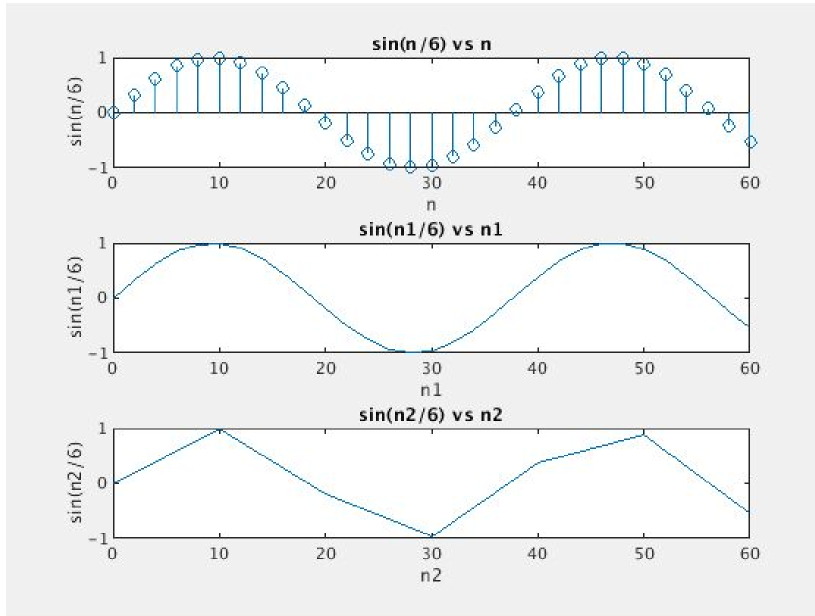
**2 Laboratory Ethics**

**INLAB REPORT:** Each student on the team must write by hand the following statement in the lab report, sign and date. “I have read and understood the Laboratory Ethics section (Section 2) of Laboratory 1. I pledge to behave ethically and with honesty in ECE438 this semester. The reports I will hand in will be the product of original work by myself and my teammate, and no one else. I will not look at other people’s laboratory. I will not use other people’s code. I will not make my labs available to other students beyond my teammates, even after the semester is over. In particular, I will not post my labs on the Internet or make my files available to other people. I will not be a cheater. ” Name, Signature, Date.

## 4.1 Displaying Continuous-Time and Discrete-Time Signals in Matlab

**INLAB REPORT:** Submit a hard copy of the plots of the discrete-time function and two continuous-time “looking” functions. Label them with the title command, and include your names. Comment on the accuracy of each of the continuous time plots.

- The plot of sin(n1/6) vs n1 is more accurate than the plot of sin(n2/6) vs n2 because we use a smaller step size in sin(n1/6) vs n1.

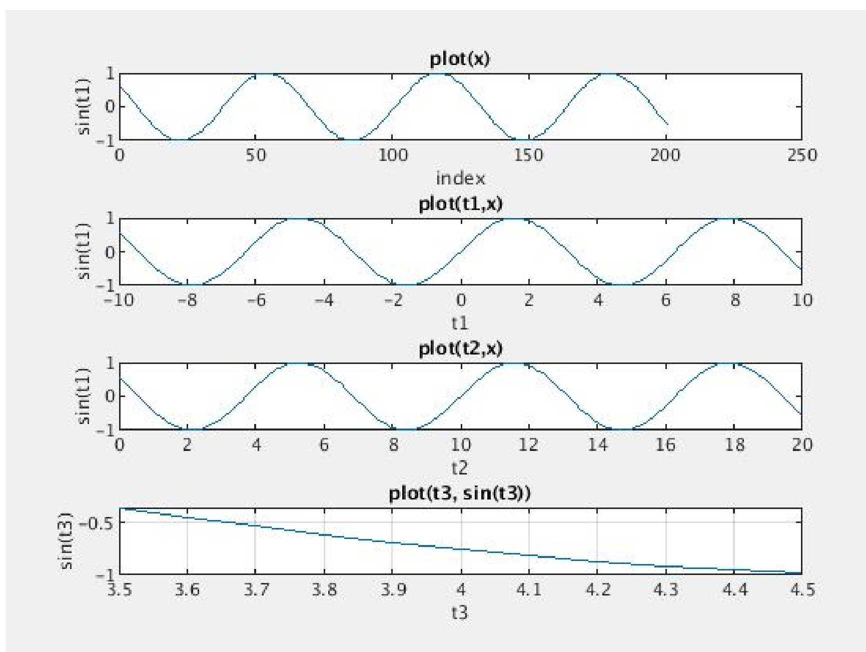



## 4.2 Vector Index versus Time

**INLAB REPORT:** Print the three subplots and explain the difference between the three signals represented. Write Matlab command(s) that would print the graph of sin(t) for the values of t on the interval [3.5,4.5]. (Pick a suitable increment for t.)

- The first signal titled plot(x) shows sin(x) plotted against its vector indices. The second signal titled plot(t1, x) shows sin(x) plotted from -10 to 10 with a stepsize of 0.1. The third signal titled plot(t2,x) shows sin(x) plotted from 0 to 20 with a stepsize of 0.1.

```
t3 = 3.5:0.1:4.5;
plot(t3, sin(t3));
title('plot(t3, sin(t3))')
xlabel('t3')
ylabel('sin(t3)')
grid on
```



## 4.3 Analytical Calculation

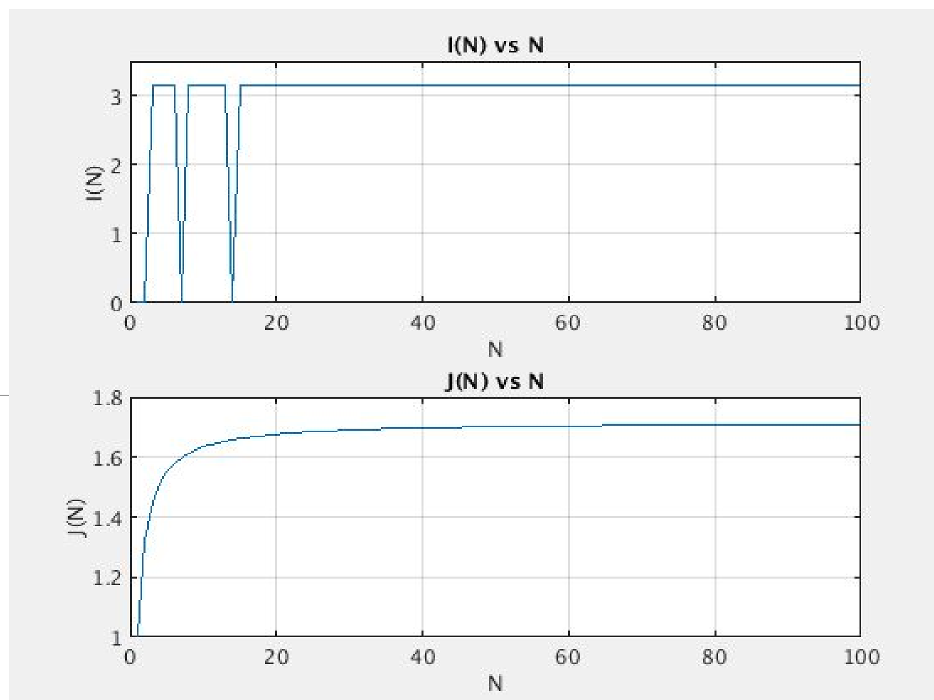
**INLAB REPORT:** Hand in your calculations of these two integrals. Show all work.

## 4.4 Numerical Computation of Continuous-Time Signals

**INLAB REPORT:** Submit plots of I(N) and J(N) versus N. Use the subplot command to put both plots on a single sheet of paper. Also submit your Matlab code for each function. Compare your results to the analytical solutions from Section 4.3. Explain why I(7) = I(14) = 0.

```
function [ I ] = integ1( N )
%UNTITLED5 Summary of this function goes here
%   Detailed explanation goes here

x = linspace(0, (2*pi), N+1);
y = ((sin(7 .* x).^2));
deltax = ((2*pi) - 0) / N;
I = deltax * sum(y);

end
```

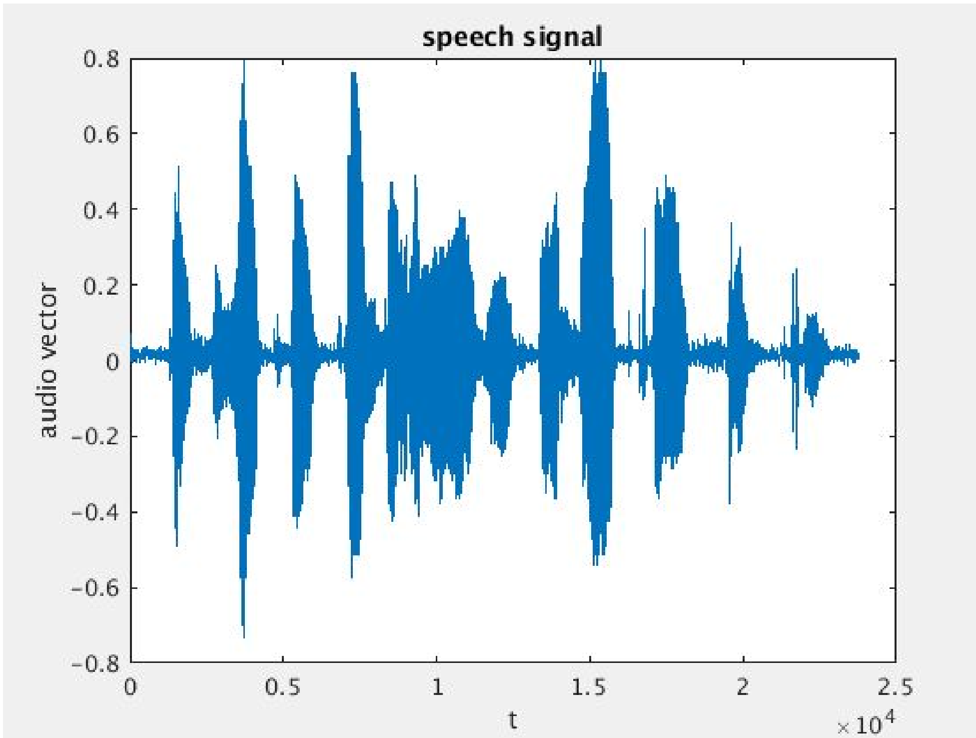
```
function [ J ] = integ2( N )
%UNTITLED9 Summary of this function goes here
%   Detailed explanation goes here

x = linspace(0, 1, N+1);
x=x(1:N);
y = exp(x);
deltax = ((1) - 0) / N;
J = deltax * sum(y);

end
```




- I(7) = I(14) = 0 because when you sum up the left corner value of ((sin7t)^2) when using 7 or 14 rectangles, some values cancel each other out and you end up with I(N) being 0.

## 5 Processing of Speech Signals

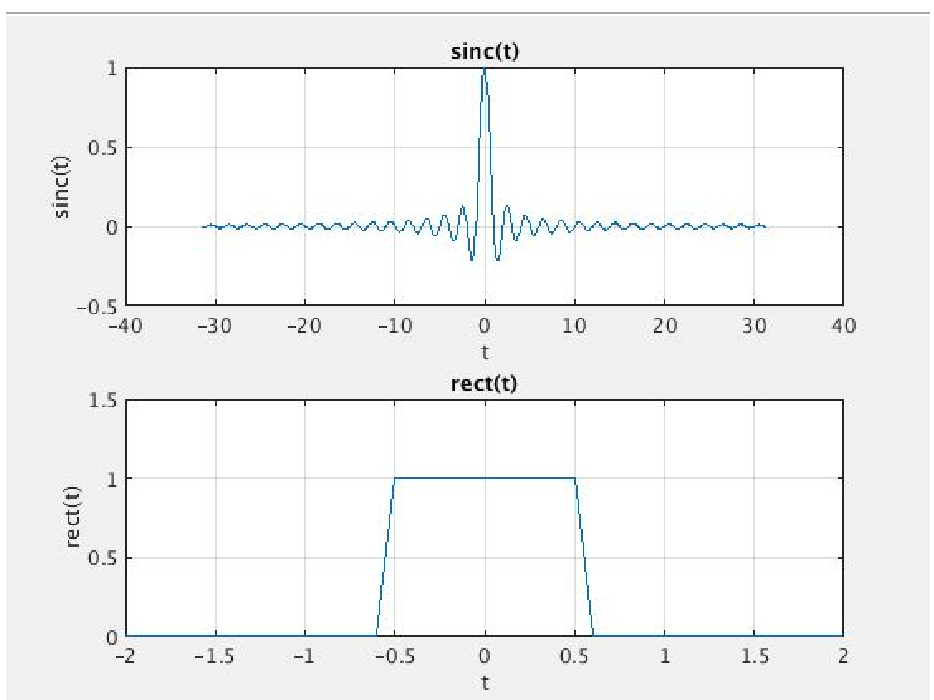
**INLAB REPORT:** Submit your plot of the speech signal.



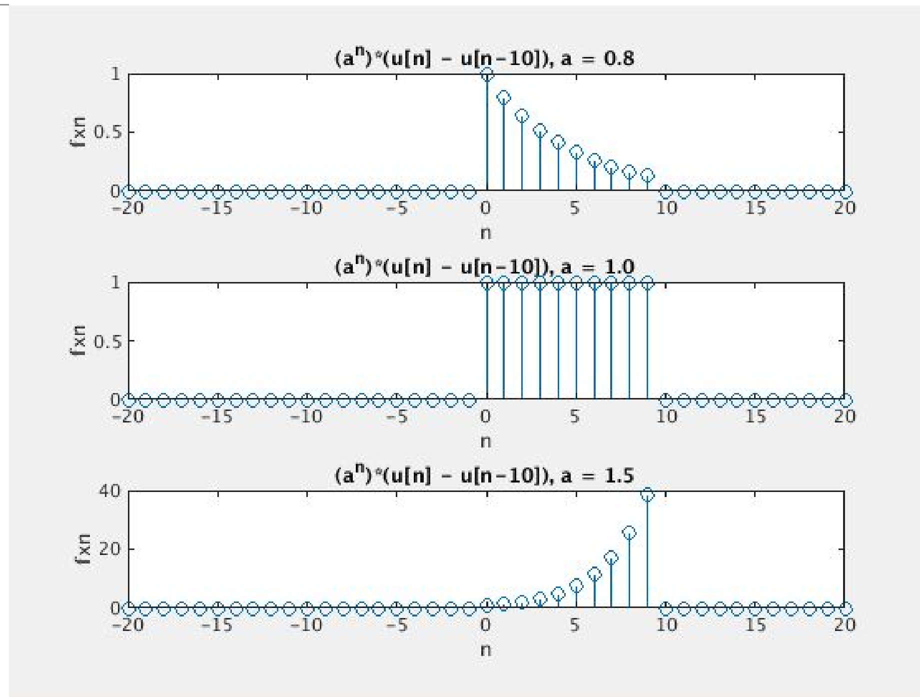

## 6 Special Functions

**INLAB REPORT:** Submit all three figures, for a total of 8 plots. Also submit the printouts of your Matlab .m-files.

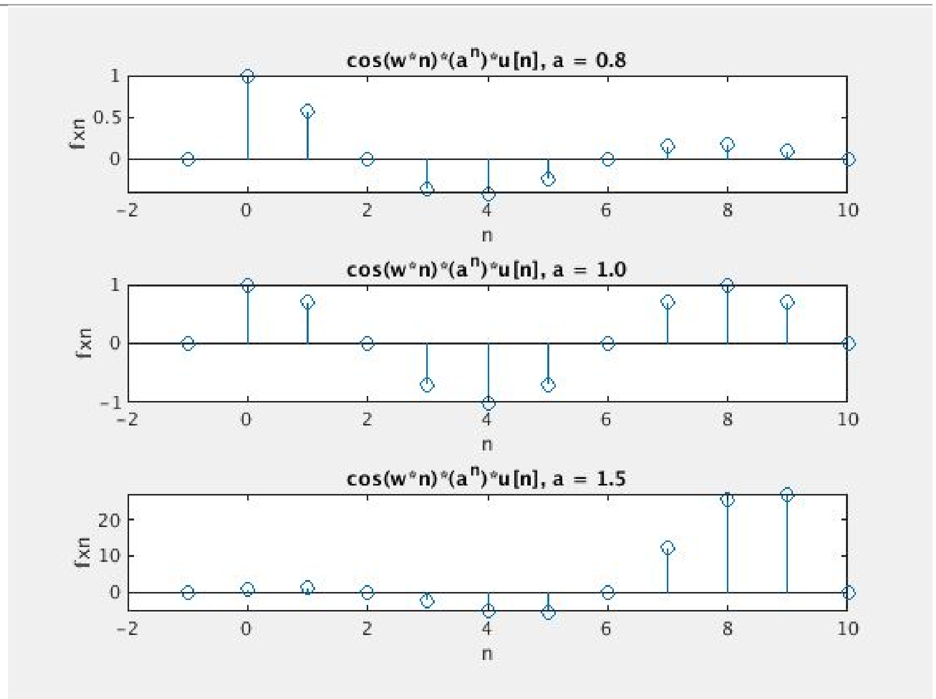
```
subplot(2,1,1)
x = -10*pi:0.1:10*pi;
y = sin(pi*x) ./ (pi*x);
y(x==0) = 1;
plot(x,y)
title('sinc(t)')
xlabel('t')
ylabel('sinc(t)')
grid on

subplot(2,1,2)
x = -2:0.1:2;
y = (abs(x)<=0.5);
plot(x,y)
ylim([0, 1.5])
title('rect(t)')
xlabel('t')
ylabel('rect(t)')
grid on
```



```
subplot(3,1,1)
n = -20:1:20;
u1 = (n>=0);
u2 = (n>=10);
a = 0.8;
fxn = (a.^n).*(u1 - u2);
stem(n, fxn)
title('(a^n)*(u[n] - u[n-10]), a = 0.8')
xlabel('n')
ylabel('fxn')

subplot(3,1,2)
n = -20:1:20;
u1 = (n>=0);
u2 = (n>=10);
a = 1.0;
fxn = (a.^n).*(u1 - u2);
stem(n, fxn)
title('(a^n)*(u[n] - u[n-10]), a = 1.0')
xlabel('n')
ylabel('fxn')

subplot(3,1,3)
n = -20:1:20;
u1 = (n>=0);
u2 = (n>=10);
a = 1.5;
fxn = (a.^n).*(u1 - u2);
stem(n, fxn)
title('(a^n)*(u[n] - u[n-10]), a = 1.5')
xlabel('n')
ylabel('fxn')

orient('tall')
```

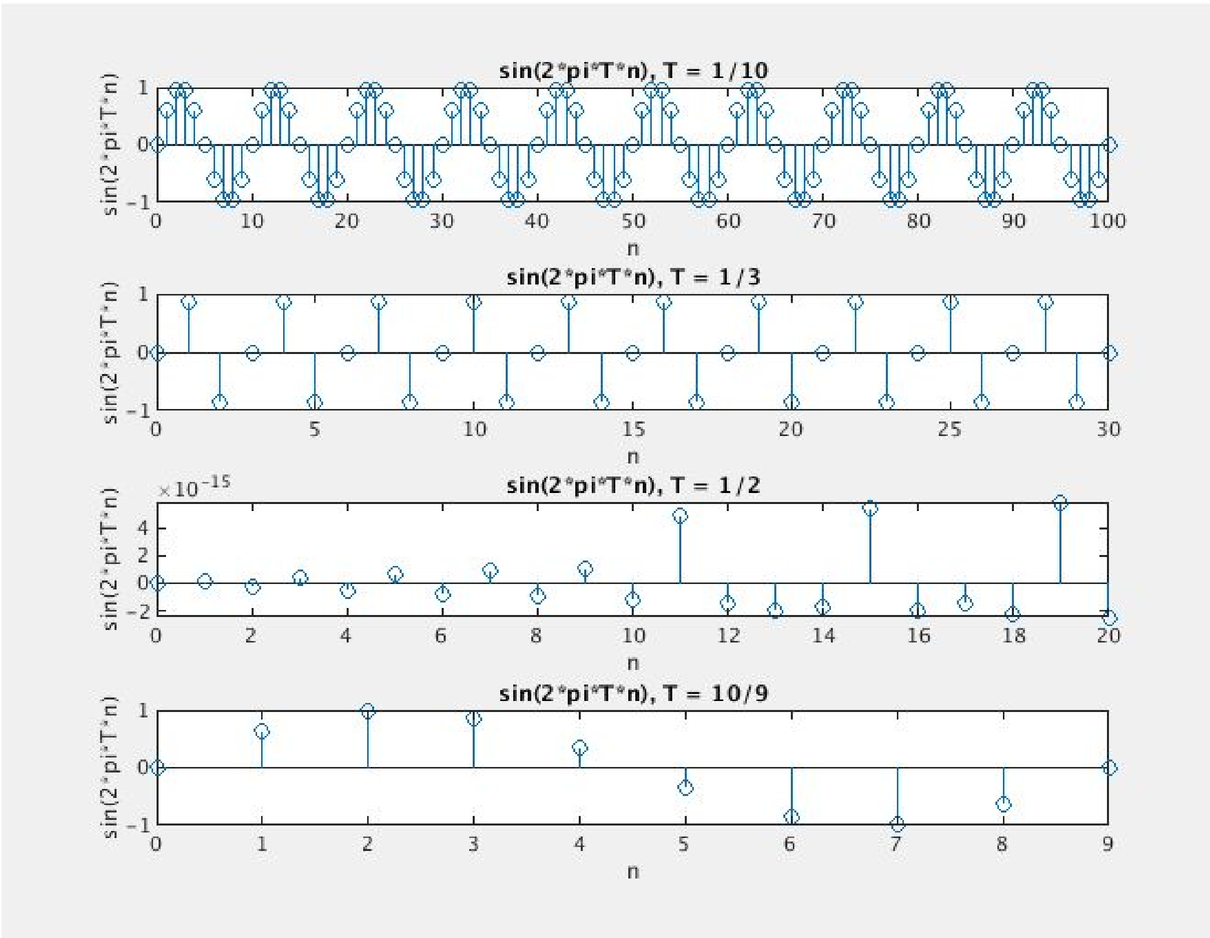


```
w = pi/4;

subplot(3,1,1)
n = -1:1:10;
u1 = (n>=0);
a = 0.8;
fxn = cos(w.*n) .* (a.^n).*u1;
stem(n, fxn)
title('cos(w*n)*(a^n)*u[n], a = 0.8')
xlabel('n')
ylabel('fxn')

subplot(3,1,2)
n = -1:1:10;
u1 = (n>=0);
a = 1.0;
fxn = cos(w.*n) .* (a.^n).*u1;
stem(n, fxn)
title('cos(w*n)*(a^n)*u[n], a = 1.0')
xlabel('n')
ylabel('fxn')

subplot(3,1,3)
n = -1:1:10;
u1 = (n>=0);
a = 1.5;
fxn = cos(w.*n) .* (a.^n).*u1;
stem(n, fxn)
title('cos(w*n)*(a^n)*u[n], a = 1.5')
xlabel('n')
ylabel('fxn')

orient('tall')
```



**7 Sampling**

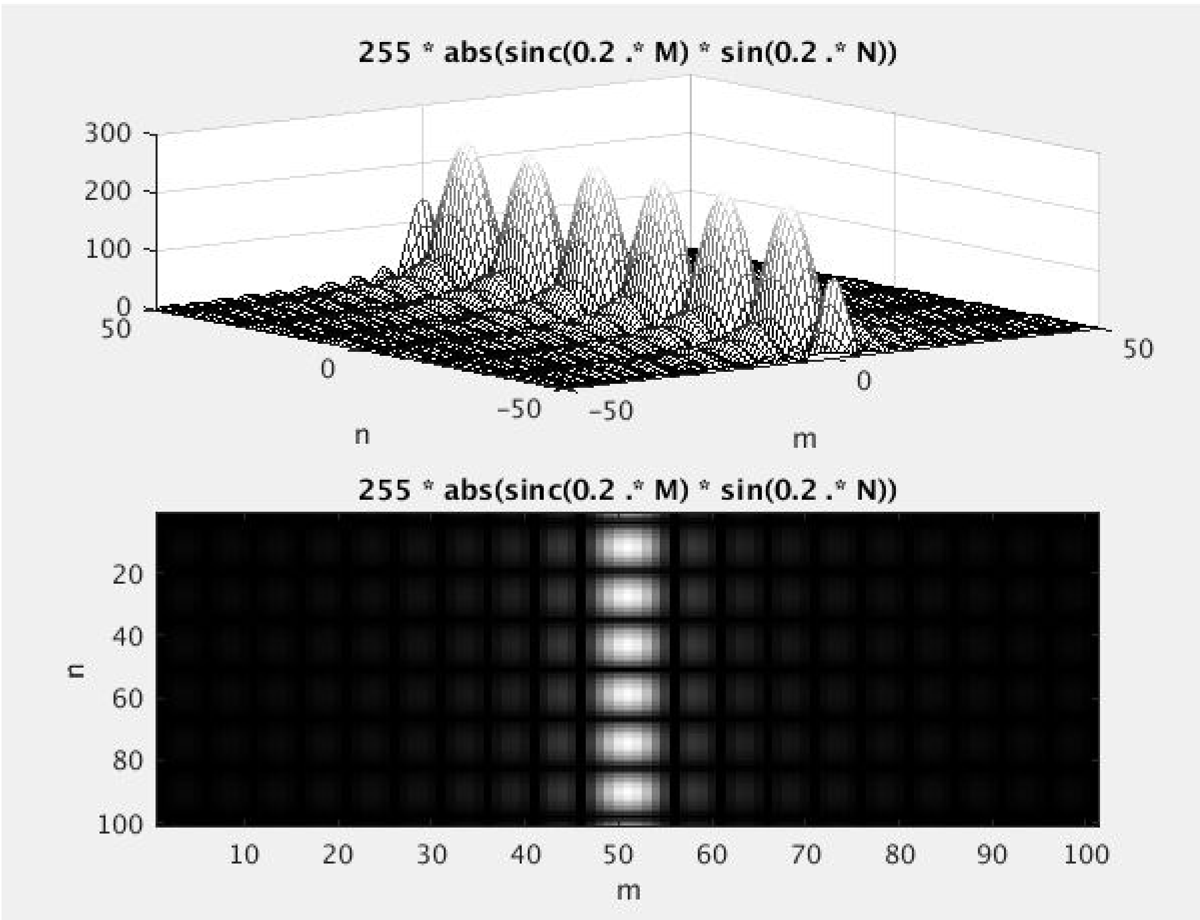
**INLAB REPORT:** Submit a hardcopy of the figure containing all four subplots. Discuss your results. How does the sampled version of the signal with Ts = 1/10 compare to those with Ts = 1/3, Ts = 1/2 and Ts = 10/9?




- When Ts = 1/10, the plot closely resembles a sine wave we sample the sine wave at a small interval. As Ts increases to ⅓, we sample the value of the sine wave at multiples of (2*pi) / 3. As Ts increases to ½, we sample the value of the sine wave at multiples of pi, resulting in the plotted values of 0. When Ts = 10/9, we make a whole trip around the unit circle plus a little bit more and then the value of the sine wave is sampled at that point.

**8 2-D Signals**

**INLAB REPORT:** Hand in hardcopies of your mesh plot and image. For which applications do you think the surface plot works better? When would you prefer the image?




- The surface plot is best suited for viewing the signal in three dimensions (amplitude, sin, sinc) while the image is suited for viewing the signal in two dimensions (sin & sinc).