Date Submitted: 2/25/2015

**EECS 360**

**Discrete Convolution in MATLAB**

**Lab Report #4**

Murtha, Daniel

KUID: 2213976

Objective:

The purpose of this lab was to practice using loops with matlab and then perform convolution on two discrete signals using loops.

Description:

Matlab provides the capability of utilizing for loops and while loops in it’s scripts. For loops are generally used when you know how many times you need to run a particular operation. While loops , however can be used in the case when you don’t actually know how many times you’ll need to perform an operation.

Below is a small example of a for loop in matlab that adds all the numbers from 1 to 100 and then displays the result.

>> j = 0;

>> for i = 1:100

j = i+j;

end

>> j

It could also be implemented using a while loop seen below

>> i = 0;

>> j = 0;

>> while j < 100

i = i+1;

j = j + i;

end

>> j

In the cases above, Matlab can utilize vector operations to perform calculations faster than conventional loops. For the sum of all numbers from 1 to 100,

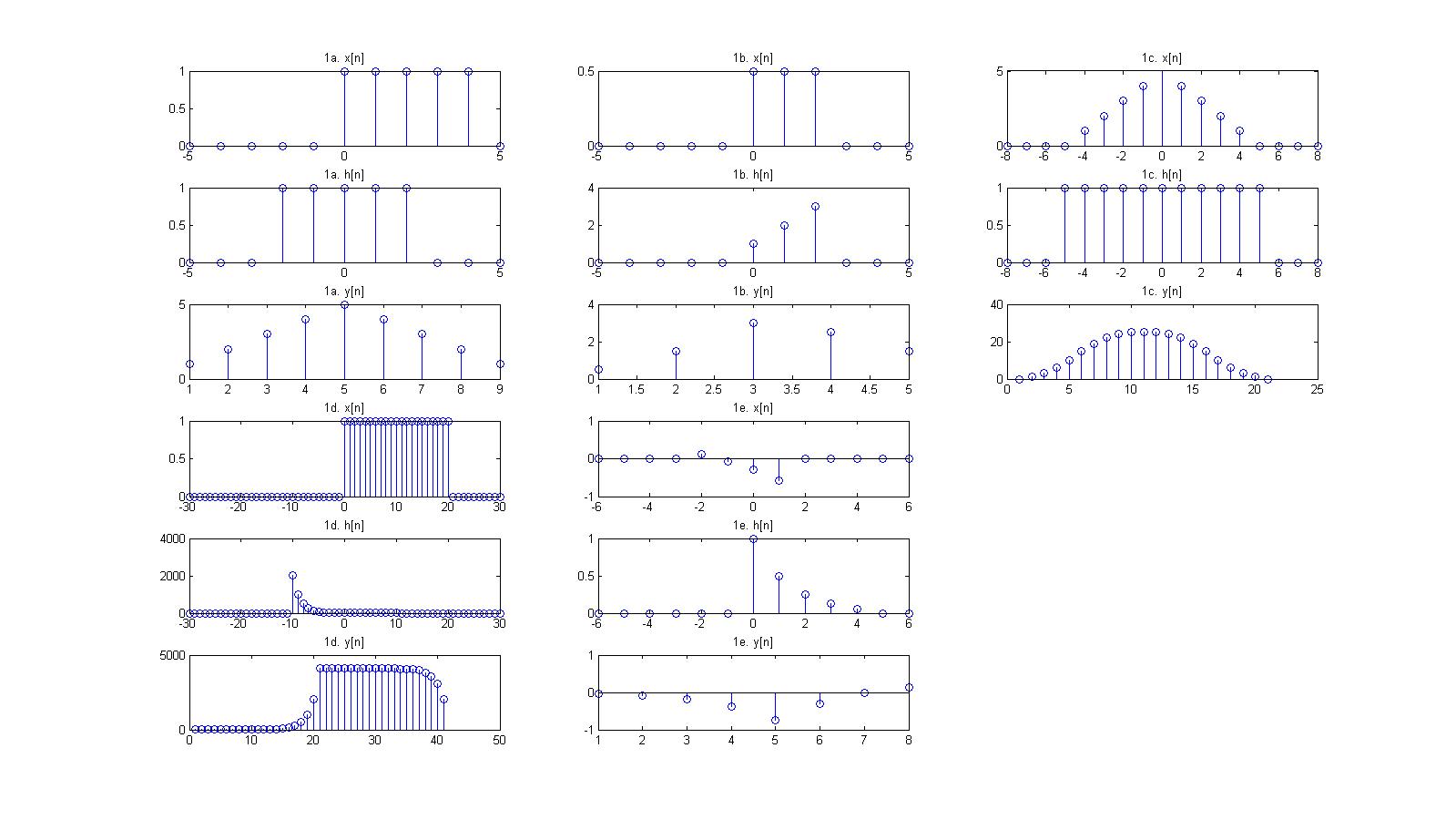
>> j = sum(1:100)

is enough to produce the same output as both of the loops above with only one line of code.

In the results section, the two signals x(t) = [] in blue and y(t) = [] in green are convoluted using loop operations in matlab and the resulting signal is displayed in red.

Results:

The results below are generated using the matlab script from the Appendix section of the report.

 Problem 1a-e

Problem 2

Problem 3

The work for convolving signals a-c by hand is in the attached .xlsx file. The graphs below were produced with the data from the calculations.

1.a

1.b

1.c

Conclusion:

In conclusion, matlab loops can be used to perform complex operations in relatively small amounts of code. Demonstrated in the code below, loops can be combined with multiple functions and vector operations expedite most tasks matlab is required to perform.

Appendix:

Lab\_4\_tests.m

clear all; close all; clc;

figure('Position',[100 100 1600 900],'Name','Lab 4 Problem 1')

%%%%%%%

subplot(6,3,1)

stem(-5:5,[0 0 0 0 0 1 1 1 1 1 0])

title('1a. x[n]')

subplot(6,3,4)

stem(-5:5,[0 0 0 1 1 1 1 1 0 0 0])

title('1a. h[n]')

subplot(6,3,7)

stem(convfunc(0,4,-2,2,ones(1,5),ones(1,5)))

title('1a. y[n]')

%%%%%%%

subplot(6,3,2)

stem(-5:5,[0 0 0 0 0 0.5 0.5 0.5 0 0 0])

title('1b. x[n]')

subplot(6,3,5)

stem(-5:5,[0 0 0 0 0 3 2 1 0 0 0])

title('1b. h[n]')

subplot(6,3,8)

stem(convfunc(0,2,0,2,[0.5 0.5 0.5],[3 2 1]))

title('1b. y[n]')

%%%%%%%

x = [];

for i = -5:5

x = [x (5-abs(i))];

end

subplot(6,3,3)

stem(-8:8,[0 0 0 x 0 0 0])

title('1c. x[n]')

subplot(6,3,6)

stem(-8:8,[0 0 0 ones(1,11) 0 0 0])

title('1c. h[n]')

subplot(6,3,9)

stem(convfunc(-5,5,-5,5,x,ones(1,11)))

title('1c. y[n]')

%%%%%%%

x = [];

for i = -10:10

x = [x ((1/2)^(i-1))];

end

subplot(6,3,10)

stem(-30:30,[zeros(1,30) ones(1,21) zeros(1,10)])

title('1d. x[n]')

subplot(6,3,13)

stem(-30:30,[zeros(1,20) x zeros(1,20)])

title('1d. h[n]')

subplot(6,3,16)

stem(convfunc(0,20,-10,10,ones(1,21),x))

title('1d. y[n]')

%%%%%%%

x = [];

for i = -2:1

x = [x (1-1.3\*exp(i/5))];

end

y = [];

for i = 0:4

y = [y (exp(-(0.7\*i)))];

end

subplot(6,3,11)

stem(-6:6,[0 0 0 0 x 0 0 0 0 0])

title('1e. x[n]')

subplot(6,3,14)

stem(-6:6,[zeros(1,6) y 0 0])

title('1e. h[n]')

subplot(6,3,17)

stem(convfunc(-2,1,0,4,x,y))

title('1e. y[n]')

convfunc.m

function [varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar,varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar] = convfunc(var, varvar, varvarvar, varvarvarvar, varvarvarvarvar, varvarvarvarvarvar)

varvarvarvarvarvarvar=var;

varvarvarvarvarvarvarvar=varvar;

varvarvarvarvarvarvarvarvar=varvarvar;

varvarvarvarvarvarvarvarvarvar=varvarvarvar;

varvarvarvarvarvarvarvarvarvarvar = varvarvarvarvarvarvar+varvarvarvarvarvarvarvarvar:varvarvarvarvarvarvarvar+varvarvarvarvarvarvarvarvarvar;

varvarvarvarvarvarvarvarvarvarvarvar=varvarvarvarvar;

varvarvarvarvarvarvarvarvarvarvarvarvar=varvarvarvarvarvar;

varvarvarvarvarvarvarvarvarvarvarvarvarvar=length(varvarvarvarvarvarvarvarvarvarvarvar)+length(varvarvarvarvarvarvarvarvarvarvarvarvar)-1;

varvarvarvarvarvarvarvarvarvarvarvarvarvarvar=[zeros(1,length(varvarvarvarvarvarvarvarvarvarvarvarvar)) varvarvarvarvarvarvarvarvarvarvarvar];

varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar=[fliplr(varvarvarvarvarvarvarvarvarvarvarvarvar) zeros(1,length(varvarvarvarvarvarvarvarvarvarvarvar))];

varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar=zeros(1,varvarvarvarvarvarvarvarvarvarvarvarvarvar);

for varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar=1:varvarvarvarvarvarvarvarvarvarvarvarvarvar

varvarvarvarvarvarvarvarvarvarvarvarvarvarvar=[varvarvarvarvarvarvarvarvarvarvarvarvarvarvar(end) varvarvarvarvarvarvarvarvarvarvarvarvarvarvar(1:end-1)];

varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar(varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar)=sum(varvarvarvarvarvarvarvarvarvarvarvarvarvarvar.\*varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar);

end

varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar = varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar;

varvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvarvar = varvarvarvarvarvarvarvarvarvarvar;

end