**Experiment-5**

**Name of the Experiment:** To study and implement the autocorrelation and cross-correlation functions on different

signals.

**Theory:** Correlation is a measure of similarity between two signals.

There are two types of correlation

1. Auto-correlation

2. Cross-correlation

Auto Correlation Function:

It is defined as correlation of a signal with itself. Auto correlation function is a measure of similarity between a signal & its time delayed version. It is represented with R(τ). Consider a signal x (t). The auto correlation function of x (t) with its time delayed version is given by: R(τ) = Integration(x(tau)x(t-tau))[from infinity to minus infinity]

Cross-correlation Function:

Cross correlation is the measure of similarity between two different signals. Consider two signals x1 (t) and x2(t). The cross correlation of these two signals R12(τ) is given by: R12(τ) = Integration(x(1tau)x2(t-tau))[from infinity to minus infinity]

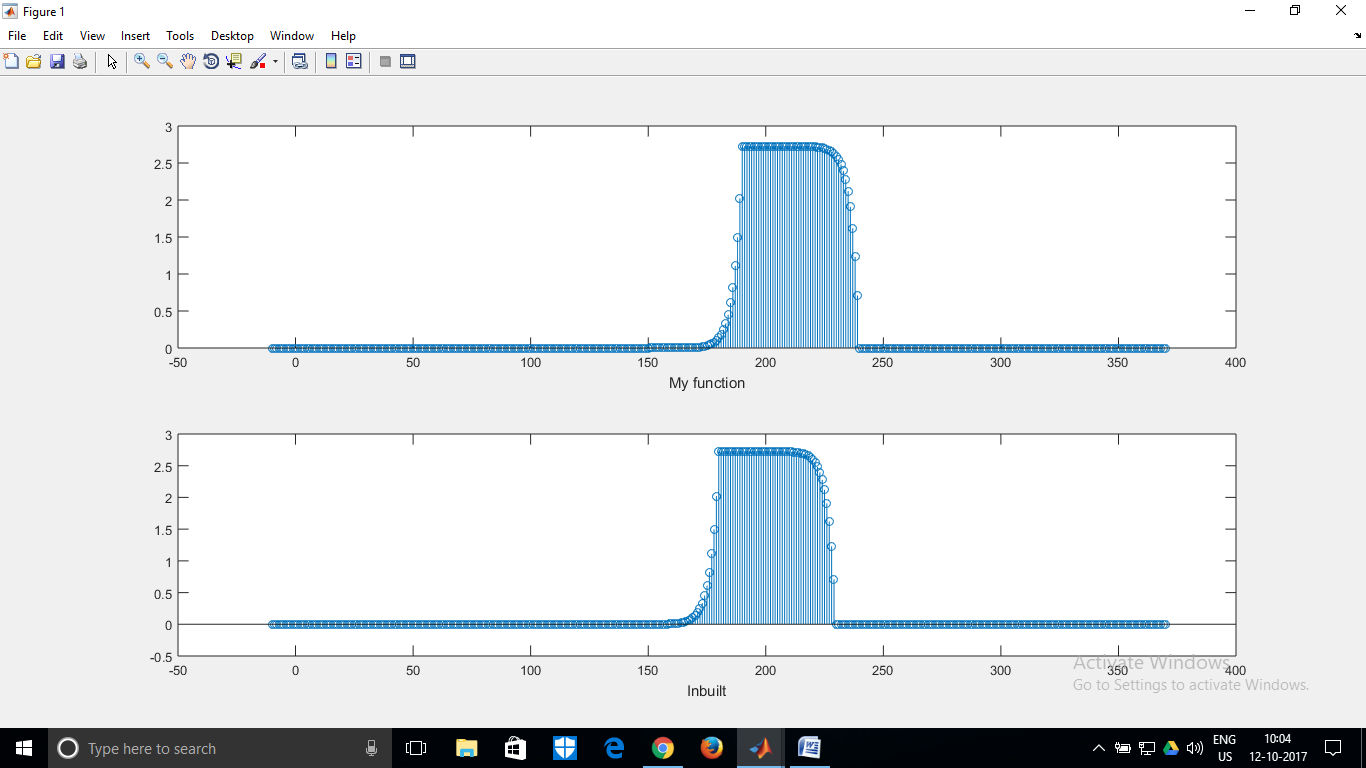
Correlation coefficient:

If f and g are two signals, then the correlation coefficient is given by

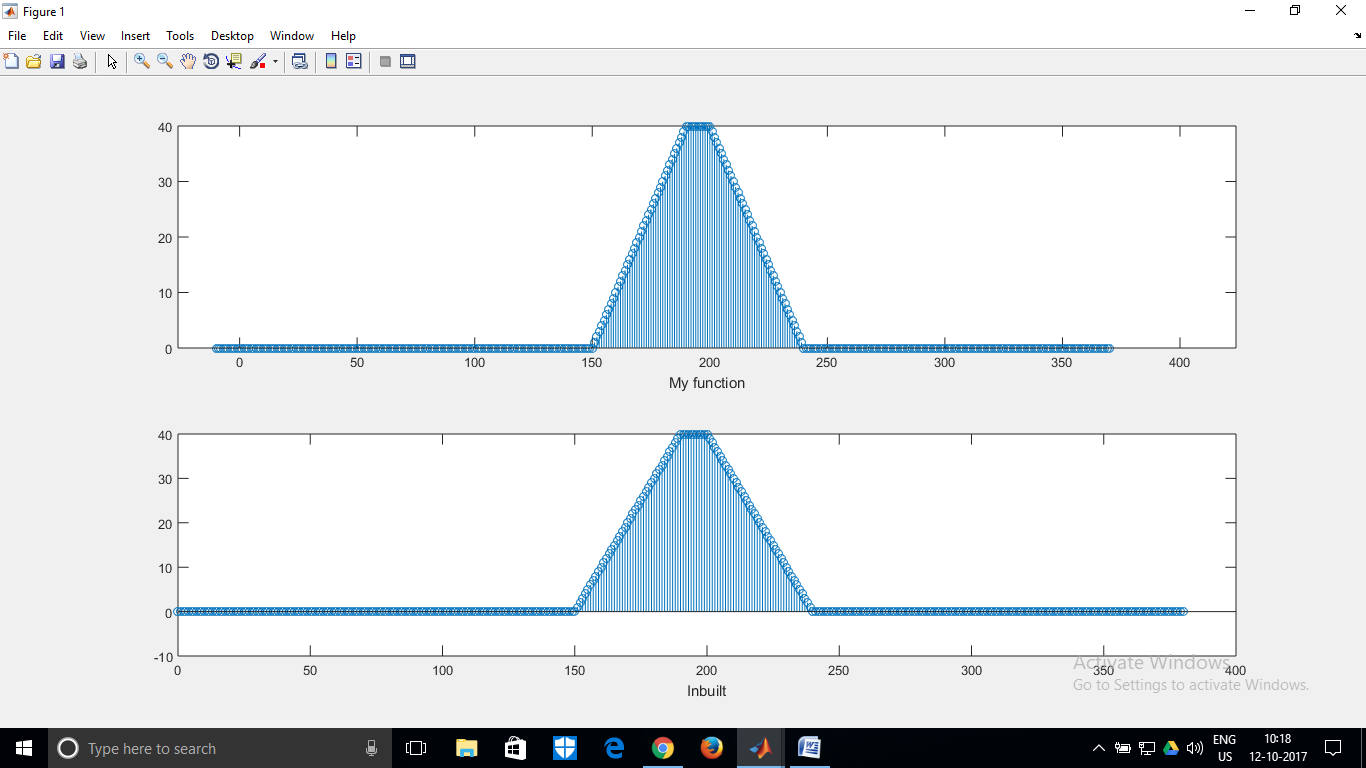
Cn =1/[√(E(f)\*E(g)]\*∫ f(t)g(t)dt {Integration of f(t)\*g(t) from minus infinity to infinity divided by square root of their respective energies}. It is a normalized method of comparing similarities between the signals.

**Results:**

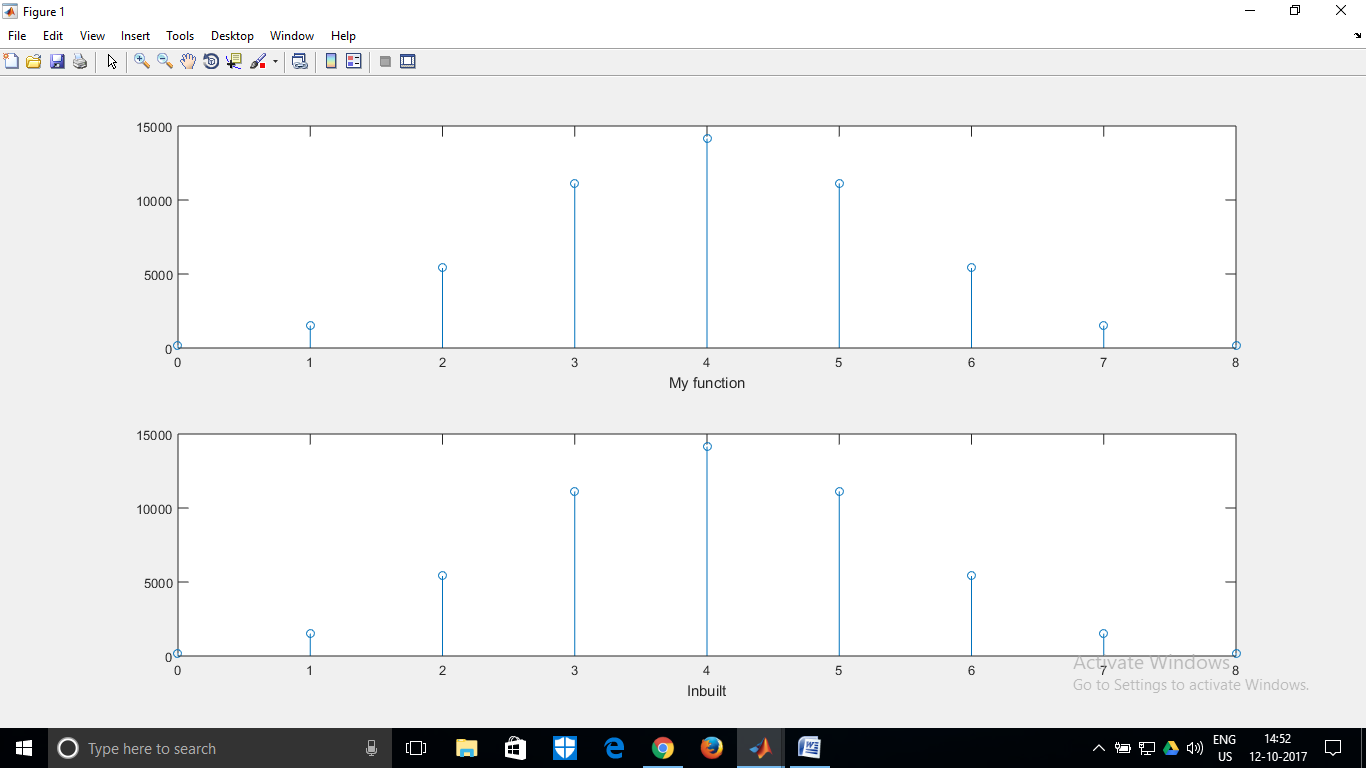
Signal 1: 1)x(t) = (sin (2πn/8)) \*(u(t) − u(t − 5)) and h(t) = e^−3t\*(u(t) − u(t − 4))



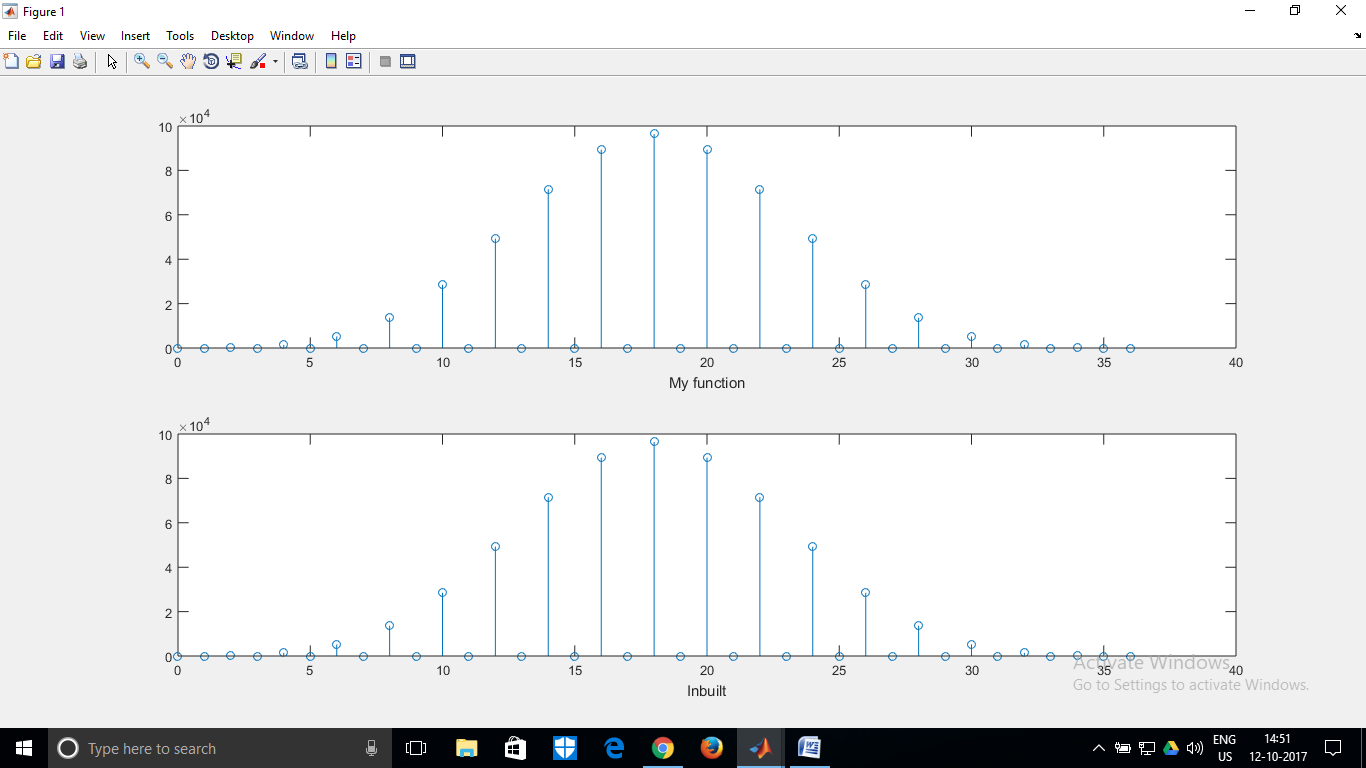
Signal 2: x(t)=u(t) − u(t − 5) and h(t) = u(t) − u(t − 4)



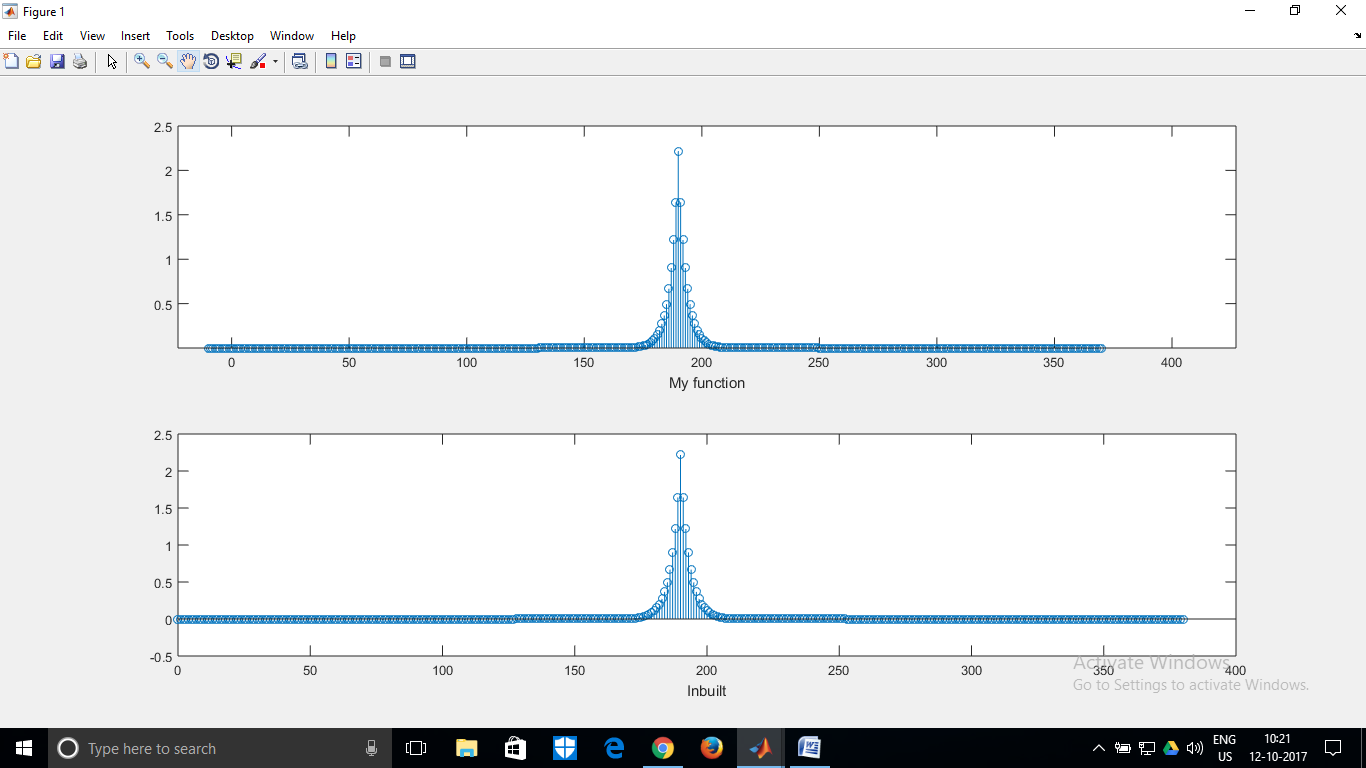
Signal 3(i):auto-correltion of (x[2n]\*x[2n])



Signal 3(ii): auto-correlation of x[n/2]\*x[n/2]



Signal 4: e^−3t\*u(t)



5. The correlation coefficients are: 1. 0.9614 for functions x(t) = u(t) − u(t − 5) and y(t) =( e^−t/5)\*(u(t) − u(t − 5))

2. -2.3095e-13 for functions x(t) = u(t) − u(t − 5) and y(t) = sin(2πt)\*(u(t) − u(t − 5))

Two functions for which correlation coefficient will be 1 and -1 are [2(u(t)-u(t-1)), 3(u(t)-u(t-1))] and [2(u(t)-u(t-1)), -3(u(t)-u(t-1))] respectively. The code for verifying them has been attached.

Conclusion: This experiment helped us understand the concept of correlation of a function. We also observed the close similarities between correlation and convolution and how they can be used interchangeably by clever manipulation of the signal’s time domain. We also verified that the auto-correlation of a function after self-convolution produces even signals as in question 3. However, the matrix shifting method which is accurate for convolution proved to be slightly unsuitable for correlation if the input time domains varied (say one signal was from -2 to 5, other 0 to 6). Hence, integration, as given by the mathematical definition seems to be the more accurate way to perform correlation.

Appendix: MATLAB Code:

clc;

clear all;

close all;

%Questions 1-4

choice=input('Enter choice ');

start=-10;

b0=-10;

endl=10;

end2=10;

scale=1;

switch (choice)

case 1

%Uncomment for dynamic inputs

% start=input('Signal x(t) starts at? ');

% b0=input('Signal h(t) starts at? ');

% endl=input('Signal x(t) ends at? ');

% end2=input('Signal h(t) ends at? ');

alpha=(start:0.1:endl-1);

beta=(b0:0.1:end2-1);

if(start>b0)

start=b0;

end;

x=(sin(2\*pi/8)).\*(unitstep(alpha)-unitstep(alpha-5));

h=exp(-3\*-beta).\*(unitstep(-beta)-unitstep(-beta-4));

case 2

%Uncomment for dynamic inputs

% start=input('Signal x(t) starts at? ');

% b0=input('Signal h(t) starts at? ');

% endl=input('Signal x(t) ends at? ');

% end2=input('Signal h(t) ends at? ');

alpha=(start:0.1:endl-1);

beta=(b0:0.1:end2-1);

if(start>b0)

start=b0;

end;

x=1.\*(unitstep(alpha)-unitstep(alpha-5));

h=1.\*(unitstep(-beta)-unitstep(-beta-4));

case 3

ch2=input('Enter secondary choice ');

x=[2 5 6 8 7];

h=x;

switch (ch2)

case 1

mult=2;

x\_temp=zeros(1,ceil(length(x)/mult));

x\_temp(1)=x(1);

for i=2:length(x\_temp)

if(mult\*i-(mult-1)<=length(x))

x\_temp(i)=x(mult\*i-(mult-1));

else

x\_temp(i)=0;

end;

end

disp(x\_temp);

x=x\_temp;

h=x\_temp;

case 2

mult=0.5;

x\_temp=zeros(1,ceil(length(x)/mult));

x\_temp(1)=x(1);

for i=1/mult+1:length(x\_temp)

t=mult\*i;

if(t==floor(t)&&t>1)

x\_temp(i-(ceil(1/mult)-1))=x(t);

else

x\_temp(i-(ceil(1/mult)-1))=0;

end;

end;

disp(x\_temp);

x=x\_temp;

h=x\_temp;

end

case 4

% Uncomment for dynamic inputs

% start=input('Signal x(t) starts at? ');

% b0=input('Signal h(t) starts at? ');

% endl=input('Signal x(t) ends at? ');

% end2=input('Signal h(t) ends at? ');

alpha=(start:0.1:endl-1);

beta=(b0:0.1:end2-1);

if(start>b0)

start=b0;

end;

x=exp(3\*-alpha).\*(unitstep(alpha));

h=exp(3\*alpha).\*unitstep(-alpha);

case 5

otherwise

disp('Oops,function does not exist');

return;

end

%x=Result(x,h); %Uncomment line for question 3

%res=Result(x,fliplr(x)); %Uncomment for question 3

res=Result(x,h);%Comment for all others other than 3

time=(start:1:length(res)+start-1);

figure;

disp('Length of convolution matrix is');

disp(length(res));

subplot(2,1,1);

stem(time,res);

xlabel('My function')

subplot(2,1,2);

time=time+10;

h=exp(-3\*beta).\*(unitstep(beta)-unitstep(beta-4));%To be uncommented for question 1

%h=1.\*(unitstep(beta)-unitstep(beta-4)); %To be uncommented for question 2

%h=exp(3\*-alpha).\*(unitstep(alpha)); %To be uncommented for question 4

%stem(time,xcorr(x,x)); %To be uncommented for question 3

stem(time,xcorr(x,h));

xlabel('Inbuilt');

disp(length(xcorr(x,x)));

%Correlation coefficients calculation, uncomment the below section

% y= @(t) (unitstep(t)-unitstep(t-5)).\* (exp(-t/5).\*unitstep(t)-unitstep(t-5));

% k=integral(y,-10,10);

% y=@(t) (unitstep(t)-unitstep(t-5)).\*(unitstep(t)-unitstep(t-5));

% k1=integral(y,-10,10);

% y=@(t) (exp(-t/5).\*(unitstep(t)-unitstep(t-5))).\*(exp(-t/5).\*(unitstep(t)-unitstep(t-5)));

% k2=integral(y,-10,10);

% disp(k/sqrt(k1\*k2));

% y= @(t) (unitstep(t)-unitstep(t-5)).\* ((sin(2\*pi\*t)).\*unitstep(t)-unitstep(t-5));

% k=integral(y,-10,10);

% y=@(t) (unitstep(t)-unitstep(t-5)).\*(unitstep(t)-unitstep(t-5));

% k1=integral(y,-10,10);

% y=@(t) ((sin(2\*pi\*t)).\*unitstep(t)-unitstep(t-5)).\*((sin(2\*pi\*t)).\*unitstep(t)-unitstep(t-5));

% k2=integral(y,-10,10);

% disp(k/sqrt(k1\*k2));

%Verification for my functions.

% y= @(t) (2\*(unitstep(t))-unitstep(t-1)).\*(3\*(unitstep(t)-unitstep(t-1)));

% k=integral(y,-10,10);

%

% y=@(t) (2\*(unitstep(t)-unitstep(t-1))).\*(2\*(unitstep(t)-unitstep(t-1)));

% k1=integral(y,-10,10);

% y=@(t) (3\*(unitstep(t)-unitstep(t-1))).\*(3\*(unitstep(t)-unitstep(t-1)));

% k2=integral(y,-10,10);

% disp(k/sqrt(k1\*k2));

% y= @(t) (2\*(unitstep(t))-unitstep(t-1)).\*(-3\*(unitstep(t)-unitstep(t-1)));

% k=integral(y,-10,10);

% y=@(t) (2\*(unitstep(t)-unitstep(t-1))).\*(2\*(unitstep(t)-unitstep(t-1)));

% k1=integral(y,-10,10);

% y=@(t) (-3\*(unitstep(t)-unitstep(t-1))).\*(-3\*(unitstep(t)-unitstep(t-1)));

% k2=integral(y,-10,10);

% disp(k/sqrt(k1\*k2));