EE387 – LAB 1

E/19/166

JAYATHUNGA W.W.K.

SEMESTER 06

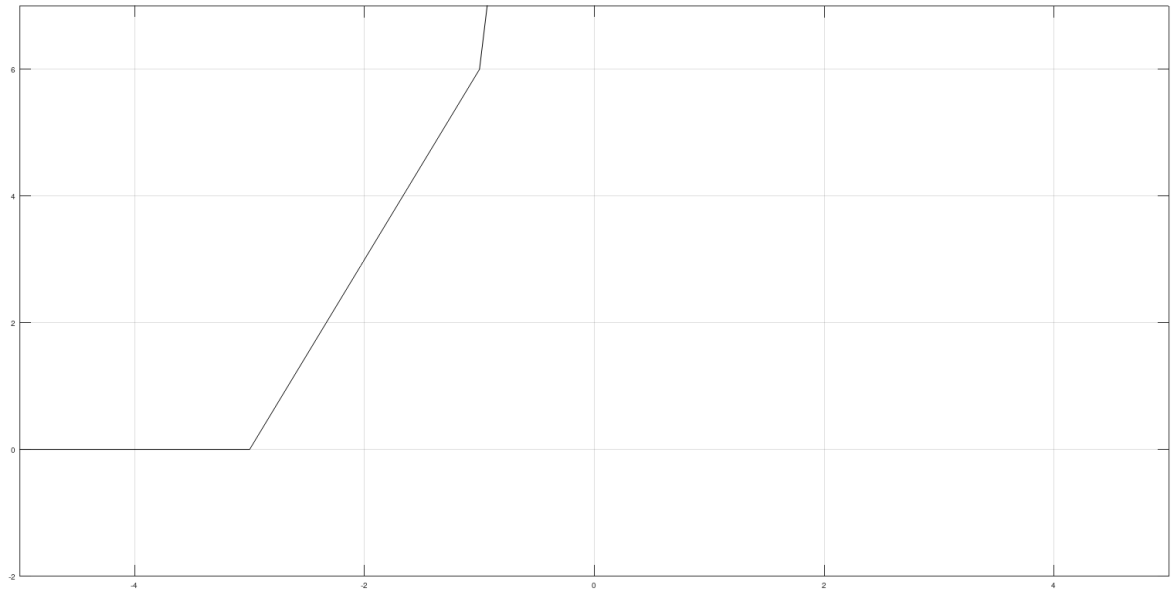
23/04/2024

PART 1: Basic Signal Representation in MATLAB

1. Part1_1.m

```
% Clear all variables
clear all;

% Define the sample period and time vectorize
Ts = 0.01;
t = -5:Ts:5;

% Define the ramp function
function y = ramp(t, m, ad)
  y = max(0, t + ad) * m;
endfunction

% Define the unit step function
function y = ustep(t, ad)
  y = (t+ad) >= 0;
endfunction

% Generate the signal
y1 = ramp(t, 3, 3);
y2 = ramp(t, -6, 1);
y3 = ramp(t, 3, 0);
y4 = ustep(t, -3);
y = y1 - 2 * y2 + 3 * y3 - y4;

% Plot the signal
plot(t, y, 'k');
axis([-5 5 -2 7]);
```

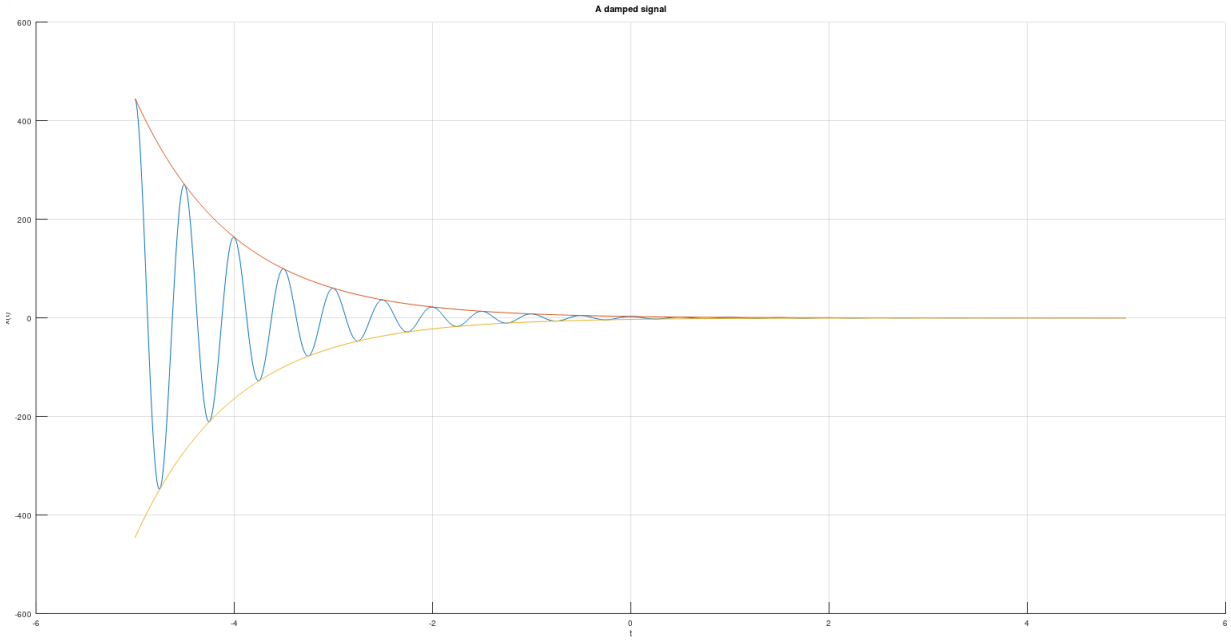
```
grid on;
```



2. Part1_2.m

```
clear all;
close all;
clc;

Ts=0.01; %Sampling time
t=-5:Ts:5; %Time vector

x= 3 * exp(-1*t) .* cos(4*pi*t);
envelope= 3 * exp(-1*t);

figure;
hold on;
plot(t,x);
plot(t, envelope);
plot(t, -1*envelope);
title('A damped signal');
```

```
xlabel('t');
ylabel('x(t)');
grid
```



PART2: Time-Domain Convolution and Elementary Signal Operations

Q: Are there any disadvantages if a high sampling frequency is used?

Yes.

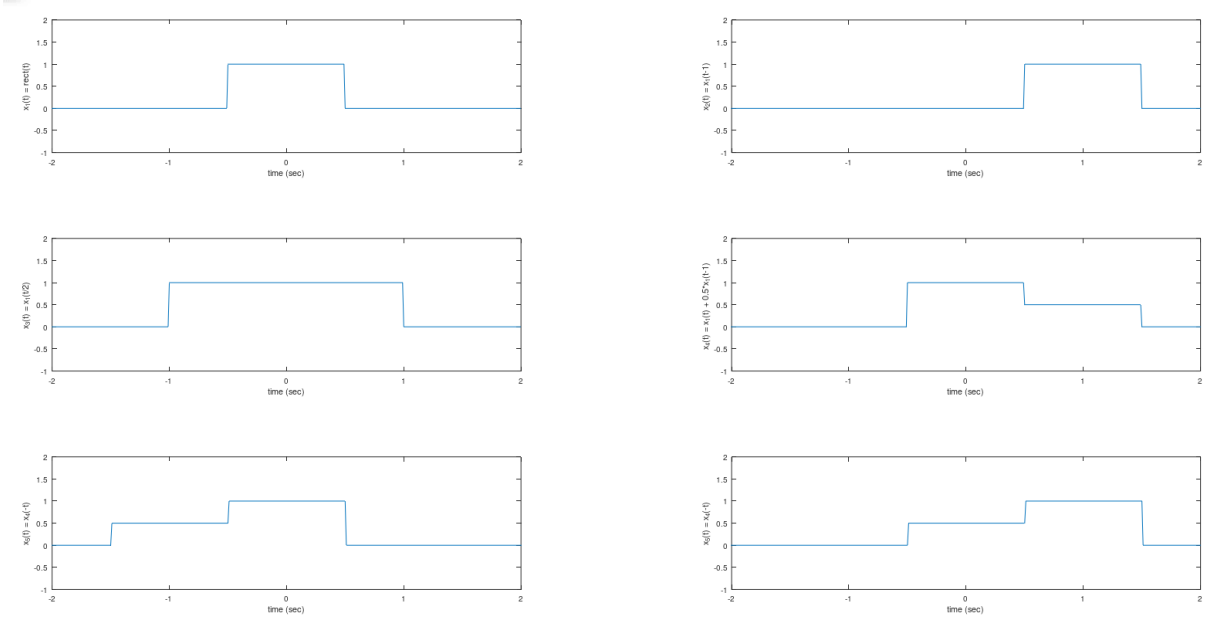
1. Increased Computational Demand: The process will require more computational power for effective processing.
2. High-Frequency Hardware Requirements: Utilizing higher sampling frequencies will necessitate hardware components, such as sensors and ADC (Analog-to-Digital Converter)

units, that can operate at higher clock speeds. These components are generally more expensive.

Part2.m

```
clear all;
close all;
clc;

Ts=0.01; %Sampling time
t=-5:Ts:5; %Time vector

% Define the rectangular pulse function
function x = rect(t)
  x = (t >= -0.5 & t < 0.5);
 endfunction


x1=rect(t);
x2 = rect(t-1);
x3 = rect(t/2);
x4 = rect(t) + (1/2)*rect(t-1);
x5 = rect(-t) + (1/2)*rect(-t-1);
x6 = rect(1-t)+ (1/2)*rect(-t);

subplot(3,2,1);
plot(t,x1) ;
axis( [-2 2 -1 2]);
xlabel( 'time (sec)' );
ylabel('x_1(t) = rect(t)');
```

```
subplot(3,2,2);
plot(t,x2);
axis( [-2 2 -1 2]);
xlabel( 'time (sec)' );
ylabel('x_2(t) = x_1(t-1)');


subplot(3,2,3);
plot(t,x3);
axis( [-2 2 -1 2]);
xlabel( 'time (sec)' );
ylabel('x_3(t) = x_1(t/2)');

subplot(3,2,4);
plot(t,x4);
axis( [-2 2 -1 2]);
xlabel( 'time (sec)' );
ylabel('x_4(t) = x_1(t) + 0.5*x_1(t-1) ');

subplot(3,2,5);
plot(t,x5);
axis( [-2 2 -1 2]);
xlabel( 'time (sec)' );
ylabel('x_5(t) = x_4(-t)');

subplot(3,2,6);
plot(t,x6);
axis( [-2 2 -1 2]);
xlabel( 'time (sec)' );
ylabel('x_5(t) = x_4(-t)');
```

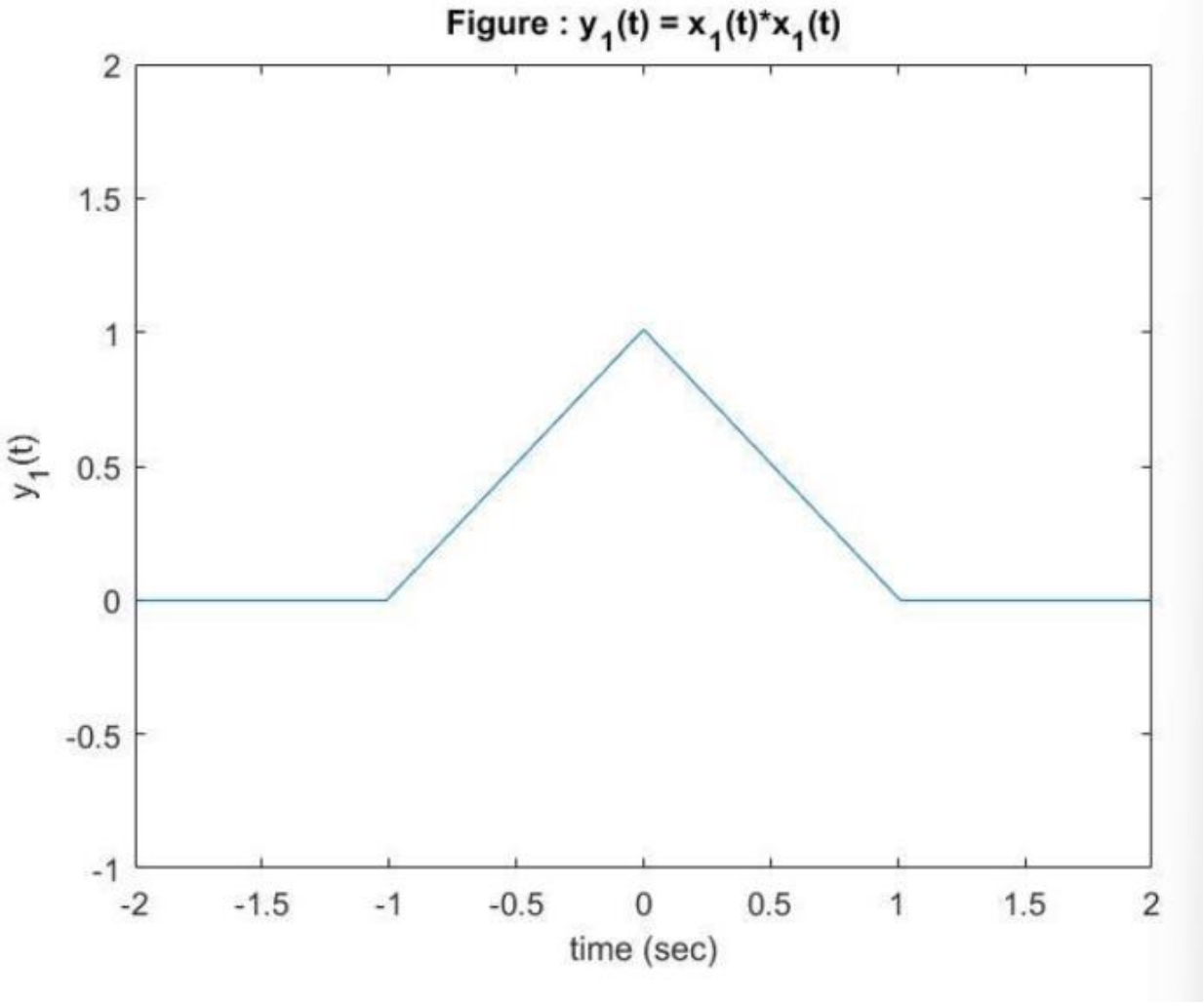


Convolution

Convolution.m

```
clear all;


Ts=0.01; %Sampling time
t=-5:Ts:5; %Time vector


% Define the rectangular pulse function
function x = rect(t)
  x = (t >= -0.5 & t < 0.5);
endfunction


x1=rect(t);
y=conv(x1,x1);


try
   plot(t,y)
```

```
catch
   disp('The dimensions do not match for t and y');
end

t_y=-10:Ts:10;

%plot(t_y,y);

y1 = Ts* conv(x1,x1);
plot(t_y, y1);
axis( [-2 2 -1 2] ) ;
xlabel( 'time (sec)');
ylabel('y_1(t)');
title('Figure : y_1(t) = x_1(t)*x_1(t)');
```



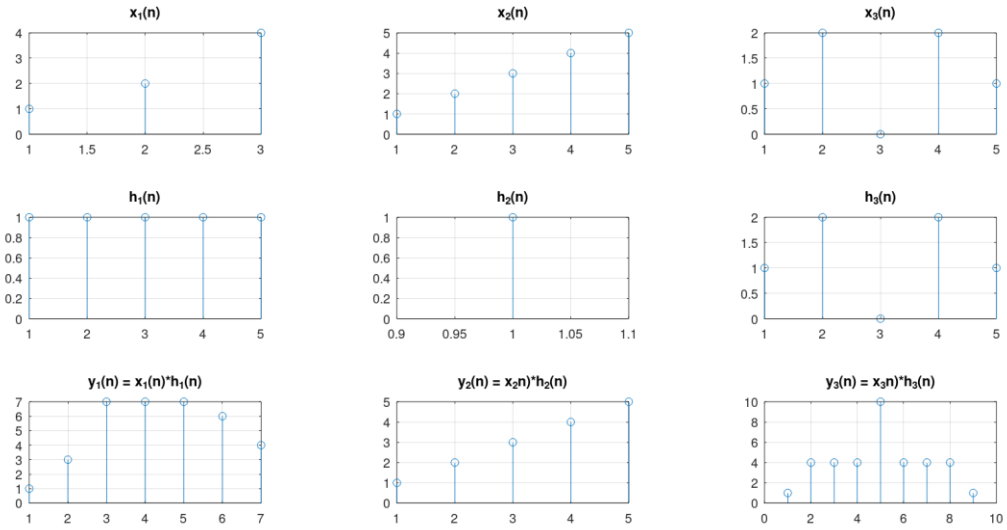
Exercise

Ex1.m

```
clear all;
close all;
clc;

x1 = [1, 2, 4];
h1 = [1, 1, 1, 1, 1];
```

```
y1 = conv(x1,h1);

x2 = 1:5;
h2 = 1;
y2 = conv(x2, h2);

x3 = [1, 2, 0, 2, 1];
h3 = x3;
y3 = conv(x3, h3);

% Plot the graphs for (1)
subplot(3, 3, 1)
stem(x1);
title('x_1(n)');
grid

subplot(3, 3, 4);
stem(h1);
title('h_1(n)');
grid

subplot(3, 3, 7);
stem(y1);
title('y_1(n) = x_1(n)*h_1(n)');
grid

% Plot the graphs for (2)
subplot(3, 3, 2)
stem(x2);
title('x_2(n)');
```

```
grid

subplot(3, 3, 5);
stem(h2);
title('h_2(n)');
grid

subplot(3, 3, 8);
stem(y2);
title('y_2(n) = x_2n)*h_2(n)');
grid

% Plot the graphs for (3)
subplot(3, 3, 3)
stem(x3);
title('x_3(n)');
grid

subplot(3, 3, 6);
stem(h3);
title('h_3(n)');
grid

subplot(3, 3, 9);
stem(y3);
title('y_3(n) = x_3n)*h_3(n)');
grid
```



Ex2.m

```
clear all;
close all;
clc;
%[Q, R] = deconv(B, A)
% B = conv(A, Q) + R.

n = 0:3;
h = 0.5.^n;

y = [1, 2, 2.5, 3, 3, 3, 2, 1,0, 0, 0, 0, 0, 0, 0, 0];

[x, R] = deconv(y, h);

figure

subplot(3, 1, 1);
stem(h);
```

```
title('Impulse Response');
xlabel('n');
ylabel('h(n)');

subplot(3, 1, 2);
stem(y);
title('Output');
xlabel('n');
ylabel('y(n)');

subplot(3, 1, 3);
stem(x);
title('Input');
xlabel('n');
ylabel('x(n)');
```