ELE632 Lab 2 Report

Table of Contents

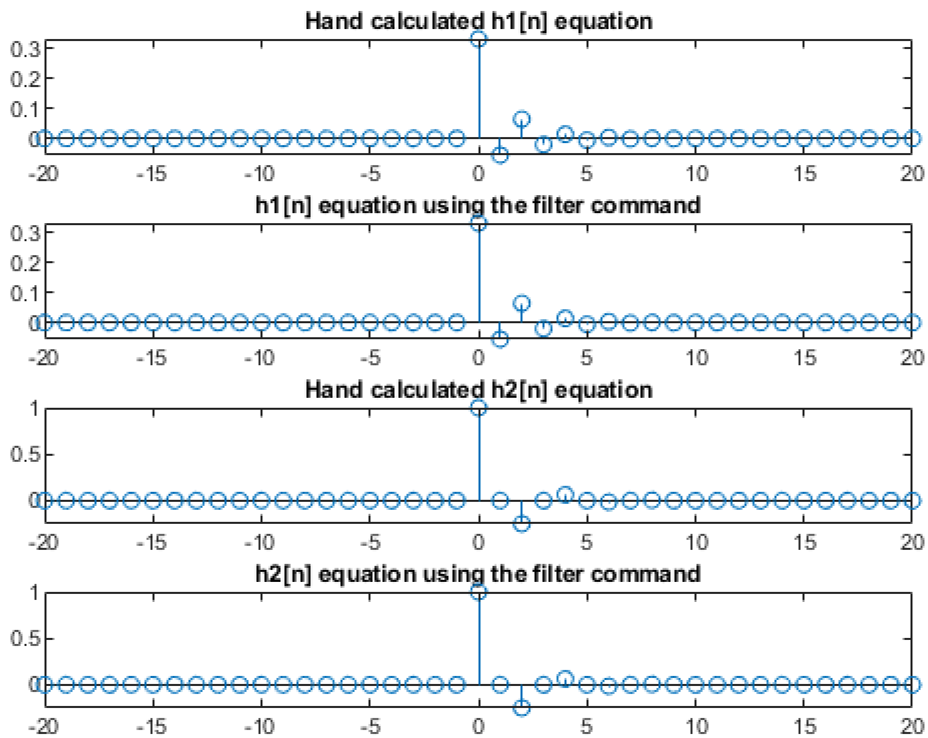
Authors: Mai Abdelhameed & Omar Ahmad

Part A: Unit Impulse Response

```
n = [-20:20];

u = @(n) (n >= 0) * 1.0 .* (mod(n,1)==0);
% Hand calculations using the z-transform
h1_calculated = @(n) ((2/15).*(1/3).^n + 0.2.*(-0.5).^n).*u(n);
h2_calculated = @(n) ((1/2).^n).*cos(n*pi/2).*u(n);

% Calculating it using the filter command
impulse = @(n) (n == 0) * 1.0 .* (mod(n, 1) == 0);
B = 1/3;
A = [1 1/6 -1/6];
h1_filter = filter(B,A,impulse(n));
B2 = [1];
A2 = [1 0 1/4];
h2_filter = filter(B2,A2,impulse(n));

figure;
subplot(4,1,1);
stem(n,h1_calculated(n));
title("Hand calculated h1[n] equation");
subplot(4,1,2);
stem(n,h1_filter);
title("h1[n] equation using the filter command");

subplot(4,1,3);
stem(n,h2_calculated(n));
title("Hand calculated h2[n] equation");
subplot(4,1,4);
stem(n,h2_filter);
title("h2[n] equation using the filter command");

%checking if h[3] is the same
threshold = 1e-10;
eq1 = all(h1_calculated(n)-h1_filter <= threshold)
eq2 = all(h2_calculated(n)-h2_filter <= threshold)
```

```
disp('The value @ n=3 is the same for both methods');
```

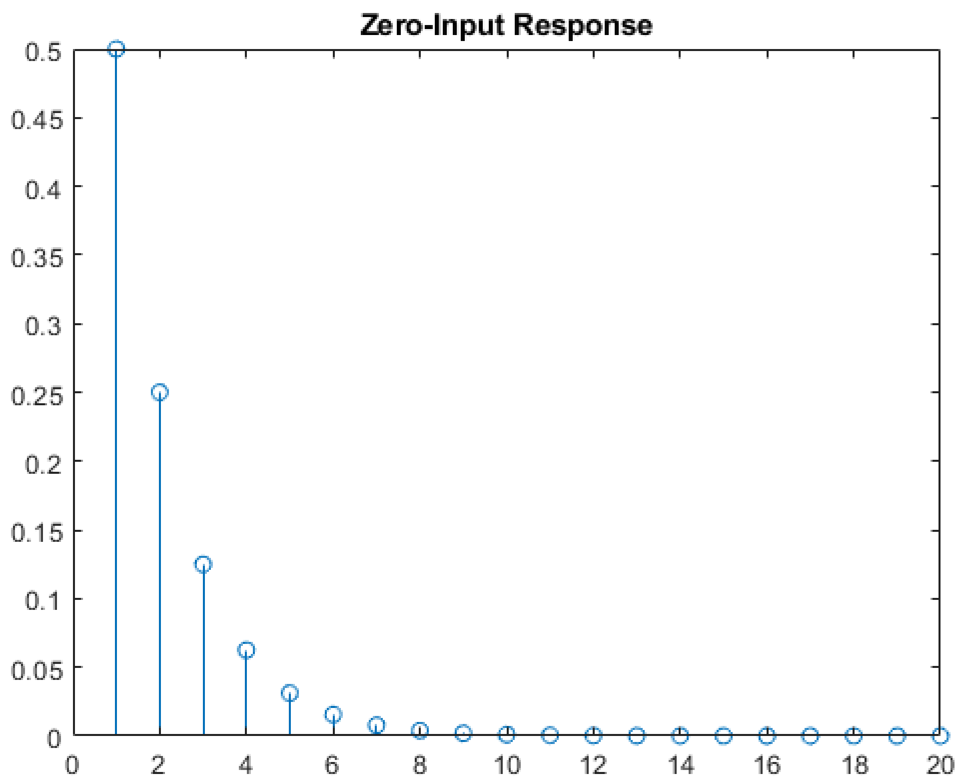
```
eq1 =

  logical

   1


eq2 =

  logical

   1

The value @ n=3 is the same for both methods
```



Part B: Zero-input response

```
%zero input response is the output of the system when input is set to
 zero
%and only initial conditions
clear;

b = [2];          % Numerator Coefficients
```

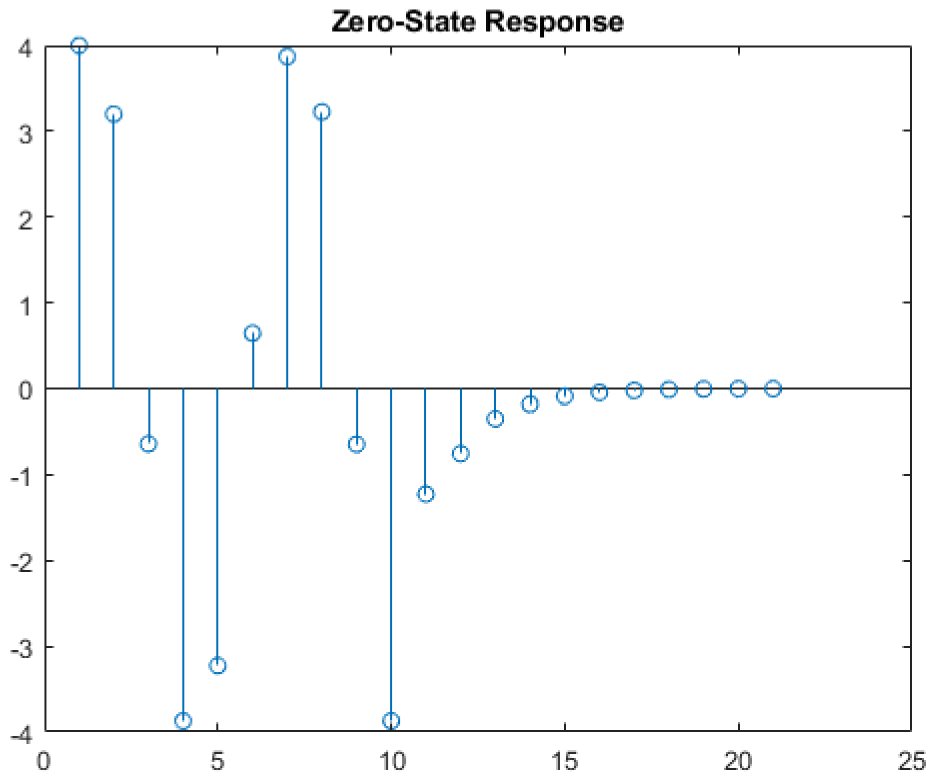
```
a = [1, -0.3, -0.1];          % Denominator Coefficients
Y = [1, 2];                % Initial conditions for output
xic = filtic(b,a,Y);   % Finding initial conditions for the system
yzi = filter(b,a,zeros(1,20),xic);   % Zero Input response

figure;
stem(yzi);
title('Zero-Input Response');
```



Part C: Zero-state response

```
%zero state response is the response of system to an input x[n] when
 all
%the initial conditions are zero
%we can find this response by using the impulse response of the system

clear;

n = [0:20];
u = @(n) (n >= 0) * 1.0 .* (mod(n,1)==0);
x = @(n) 2*cos((2/6)*pi*n).*(u(n) - u(n-10));

b = [2];          % Numerator Coefficients
a = [1, -0.3, -0.1];          % Denominator Coefficients

yzs = filter(b,a,x(n));   % Zero State response
```

```
figure;
stem(yzs);
title('Zero-State Response');
```

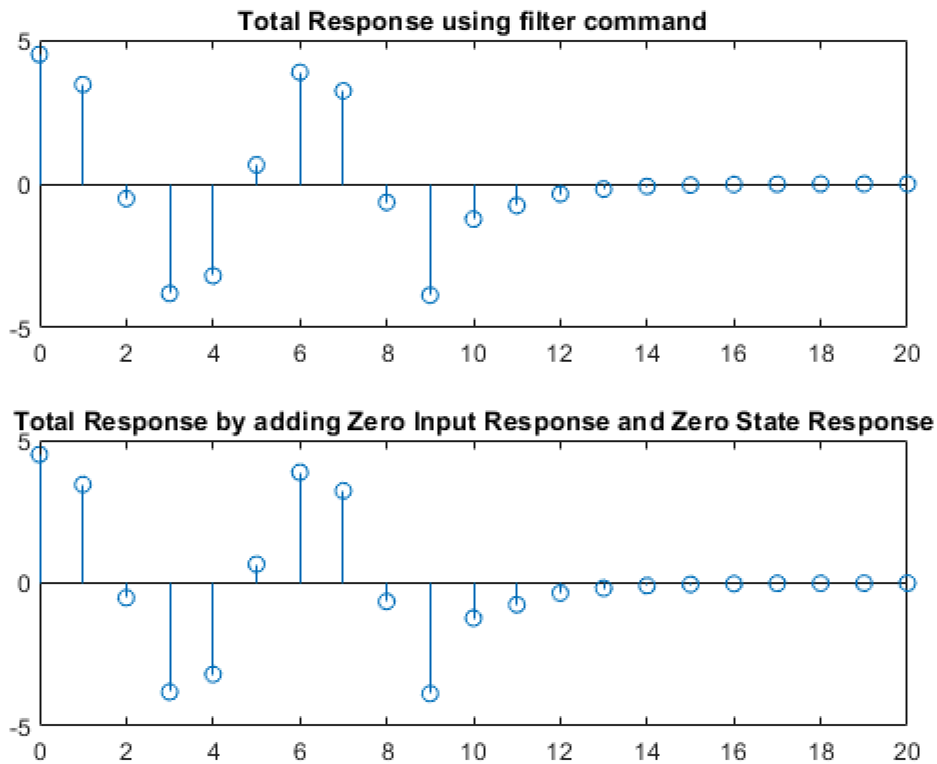


Part D: Total Response

```
%output = zero input response + zero state response
clear;
n = [0:20];
u = @(n) (n >= 0) * 1.0 .* (mod(n,1)==0);
x = @(n) 2*cos((2/6)*pi*n).*(u(n) - u(n-10));

b = [2];         % Numerator Coefficients
a = [1, -0.3, -0.1];          % Denominator Coefficients
Y = [1, 2];                % Initial conditions for output
xic = filtic(b,a,Y);   % Finding initial conditions for the system
yzi = filter(b,a,zeros(1,21),xic);   % Zero Input response
yzs = filter(b,a,x(n));   % Zero State response
yadded = yzs + yzi;
y = filter(b,a,x(n),xic);

figure;
subplot(2,1,1);
stem(n,y);
title("Total Response using filter command");
```

```
subplot(2,1,2);
stem(n,yadded);
title("Total Response by adding Zero Input Response and Zero State
 Response");

disp('It is clear to see that both graphs are equal.');
```

It is clear to see that both graphs are equal.

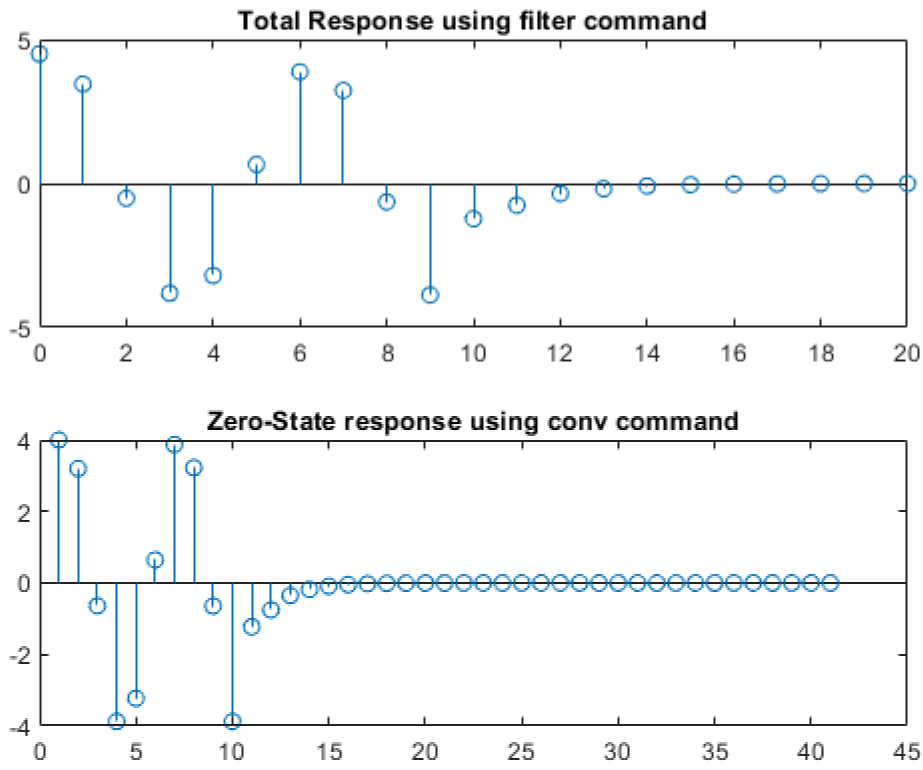


Part E: Convolution and System Stability

```
clear;
n = [0:20];
%system from B
impulse = @(n) (n == 0) * 1.0 .* (mod(n, 1) == 0);
b = [2];          % Numerator Coefficients
a = [1, -0.3, -0.1];          % Denominator Coefficients
h = filter(b,a,impulse(n));   % Zero Input response

%input from C
u = @(n) (n >= 0) * 1.0 .* (mod(n,1)==0);
x = @(n) 2*cos((2/6)*pi*n).*(u(n) - u(n-10));
%yzs = filter(b,a,x(n));   % Zero State response

S = conv(x(n), h);
stem(S);
```

```
title("Zero-State response using conv command");

disp('Matches part C');
disp('It asymptotically stable as the bended input converges to 0.');
```

Matches part C
It asymptotically stable as the bended input converges to 0.

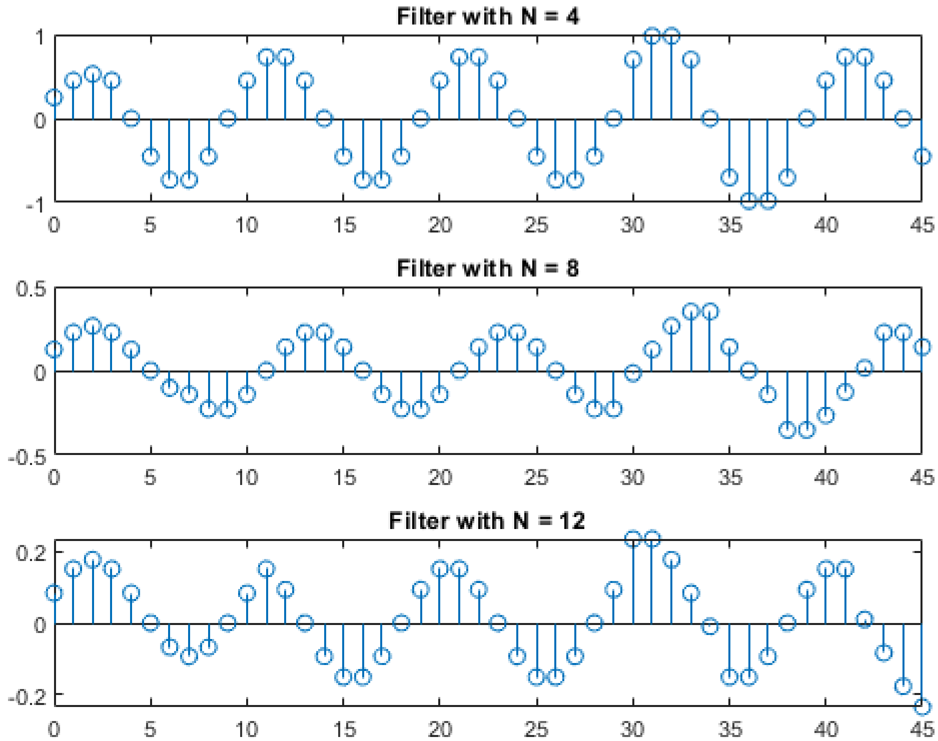


Part F: Moving average filter

```
clear;
% #1
disp('For Q.1:');
disp('A is equal to 1');
disp('B is equal to the N numbered sum of 1/N');

% #3
impulse = @(n) (n == 0) * 1.0 .* (mod(n, 1) == 0);
x = @(n) cos(pi*n/5)+impulse(n-30)-impulse(n-35);

n = [0:45];

[a,b] = params(4);
[a2,b2] = params(8);
[a3,b3] = params(12);
```

```
figure;
subplot(3,1,1);
stem(n,filter(b,a,x(n)));
title("Filter with N = 4");

subplot(3,1,2);
stem(n,filter(b2,a2,x(n)));
title("Filter with N = 8");

subplot(3,1,3);
stem(n,filter(b3,a3,x(n)));
title("Filter with N = 12");

disp('The average value for a cosine signal is 0, so as N increases,
 the');
disp('resulting filtered signal approaches 0, and the effect of the
 impulse');
disp('on the size of the output signal decreases.');
```

For Q.1:
A is equal to 1
B is equal to the N numbered sum of 1/N
The average value for a cosine signal is 0, so as N increases, the resulting filtered signal approaches 0, and the effect of the impulse on the size of the output signal decreases.



Published with MATLAB® R2018b