
Table of Contents

.....	1
Question 1: Convolution of Non-Causal Signals	1
Question 2: Smoothing Data	2
Question 3: Difference Equations	14

```
% Signals and Systems: Lab 2
% Name: Omar Rayyan
% Net-Id: olr7742

close all;
clear;

% Setting up required functions
u = @(n)double(n>=0);
del = @(n)double(n==0);
```

Question 1: Convolution of Non-Causal Signals

```
n = [-5:10];
f = 3*del(n+2)-del(n-1)+2*del(n-3);
g = u(n+4)-u(n-3);

% Plot f(n)
figure;
subplot(3,1,1);
stem(n,f);
title('f(n)');
grid on;
xlabel('n');
ylabel('f(n)');
axis([-4,10,-4,4]);

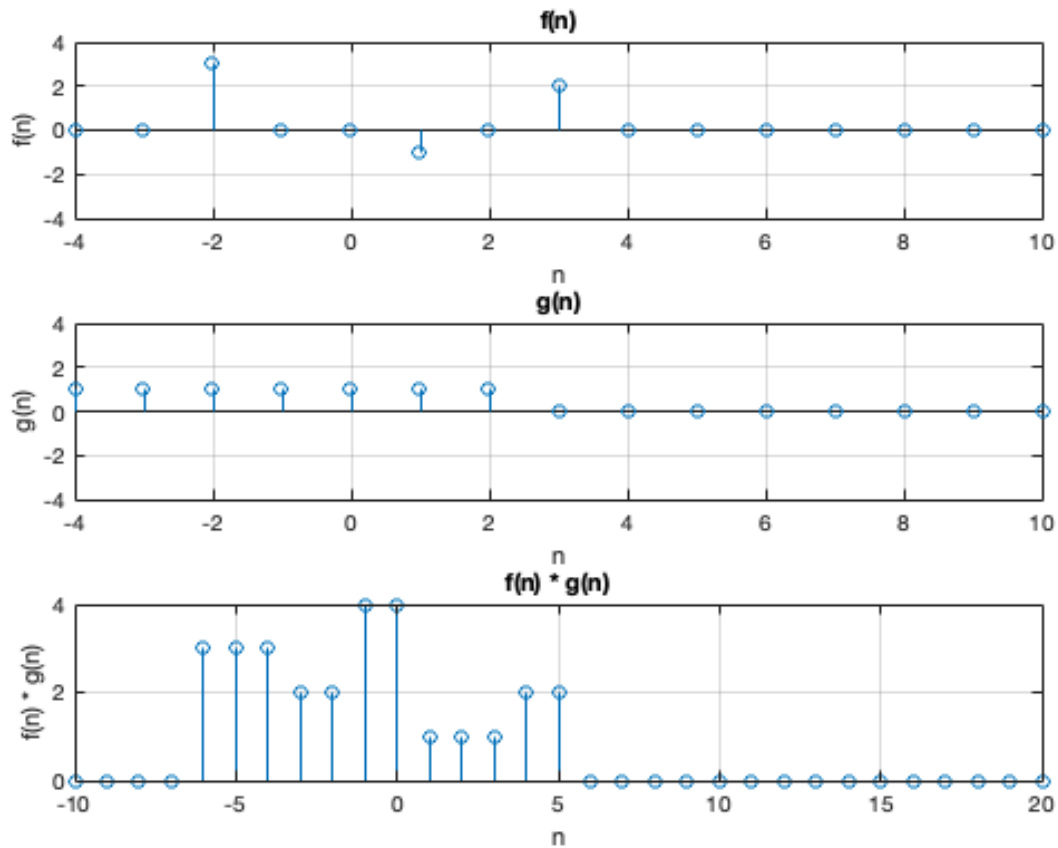
% Plot g(n)
subplot(3,1,2);
stem(n,g);
title('g(n)');
grid on;
xlabel('n');
ylabel('g(n)');
axis([-4,10,-4,4]);

% Finding f(n)*g(n)
x = conv(f,g);
% x(1:14) = [];
% x(end-13:end) = [];
length_x = -10: length(x)-11;
subplot(3,1,3);
```

```

stem(length_x,x);
title('f(n) * g(n)');
xlabel('n');
ylabel('f(n) * g(n)');
grid on;

```



Question 2: Smoothing Data

```

load DataEOG.txt;
x = DataEOG;

% Stem for DataEOG

figure; stem(x);
title('DataEOG(n)');
grid on;
xlabel('n');
ylabel('DataEOG(n)');

% Plot for DataEOG

figure; plot(x);
title('DataEOG(n)');

```

```

grid on;
xlabel('n');
ylabel('DataEOG(n)');

% Simple Impulse Response

h = ones(1,11)/11;
figure; stem(h);
title('h(n)');
grid on;
xlabel('n');
ylabel('h(n)');

% Convoluting h(n) with DataEOG

y = conv(x,h);
figure; plot(y);
title('x(n) * h(n)');
xlabel('n');
ylabel('x(n) * h(n)');
grid on;

% (a) What is the effect of the convolution in this example?
% Answer: It made the curve smoother. The effect of this convultion is
% equivalent to going to every point on the graph and replacing it with the
% average of that point and the 10 points to it's left. In a sense, it can
% be thought of as a noise remover

% (b) How is the length of y(n) related to the length of x(n) and h(n)?
% Answer: Size of y is equal to the summation of the sizes of x and h
% subtracted by 1. That is because every point is replaced with the avarage
% of it and the 10 points to it's left so there will be a point 10 steps
% after the last point in the graph.

% Plotting x(n) and y(n) on same graph
y = conv(x,h);
figure;
plot(y); hold on; plot(x);
title('x(n) * h(n)');
xlabel('n');
ylabel('x(n) * h(n)');
grid on;

% (c) What problem do you see? Can we line y and x?
% The convulated graph (y) is slightly shifted to the right
% of the original graph x(n). It can be translated to the left to make it
% line up with x(n). The reason behind this shifting is because every poitn
% is replaced with teh avergae of itself and the previous 10 points so in a
% sense the average delay the effect of a point to make the change happen
% gradually.

y2 = y;
y2(1:5) = [];
y2(end-4:end) = [];

```

```

% Plotting x(n) and y2(n) on same graph
figure;
plot(y2); hold on; plot(x);
title('y2 and x');
xlabel('n');
ylabel('y2 and x');
grid on;

% (d) What is the effect of these commands?
% They shift the graph to the left by 5. It basically remove the first 5 and
% last 5 elements from the vector to make them line
% up on the plot.
% What is the length of y2?
% 600x1, it is 10 rows less than y as 5 are removed from each side to shift
% it to the left
% What do you notice now?
% The graoh seem to match the original x more after beign shifterd

% (e) Repeat the problem, but use a different impulse response

h = ones(1,31)/31;

y = conv(x,h);
figure;
plot(y); hold on; plot(x);
title('x(n) * h(n)');
xlabel('n');
ylabel('x(n) * h(n)');
grid on;

y2 = y;
y2(1:15) = [];
y2(end-14:end) = [];

% Plotting x(n) and y2(n) on same graph
figure;
plot(y2); hold on; plot(x);
title('y2 and x');
xlabel('n');
ylabel('y2 and x');

grid on

% What should the parameters in part (d) be now? 15 and 14 as shown in the
% script above

% (f) Repeat the problem, but use a different impulse response

h = ones(1,67)/67;

y = conv(x,h);

```

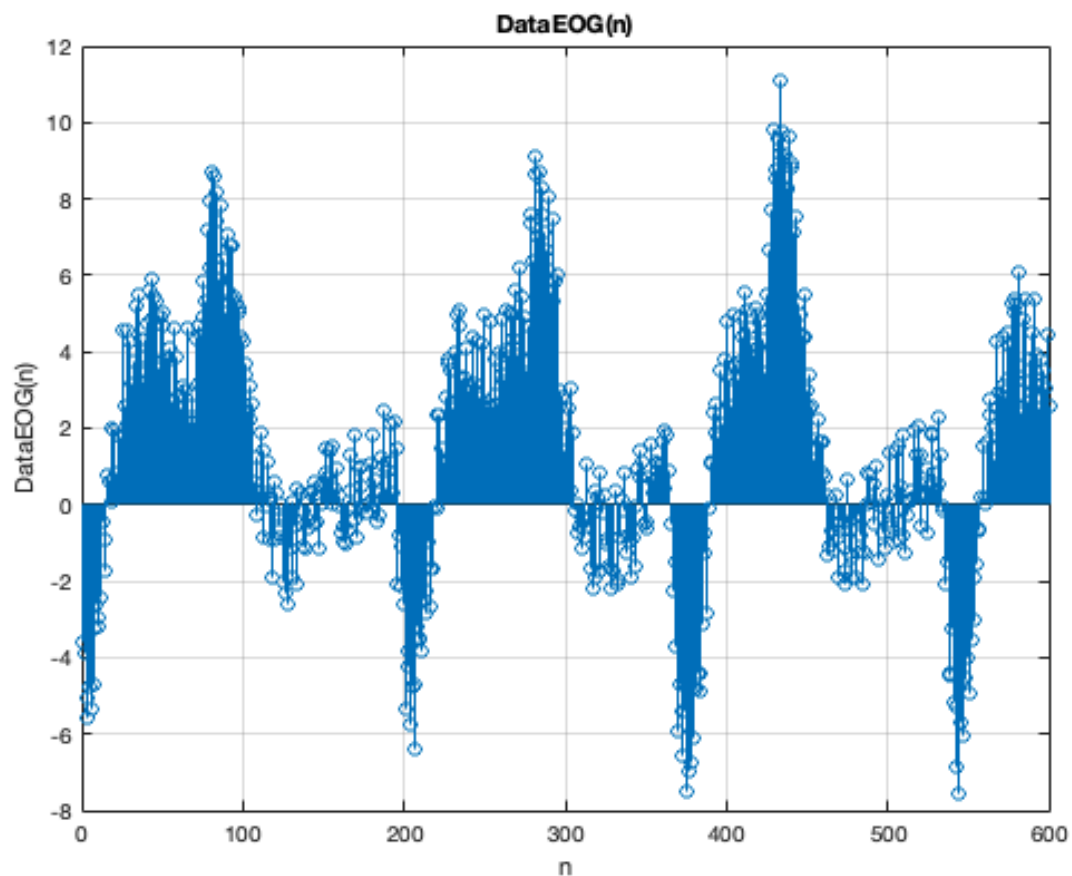
```
figure;  
plot(y); hold on; plot(x);  
title('x(n) * h(n)');  
xlabel('n');  
ylabel('x(n) * h(n)');  
grid on;
```

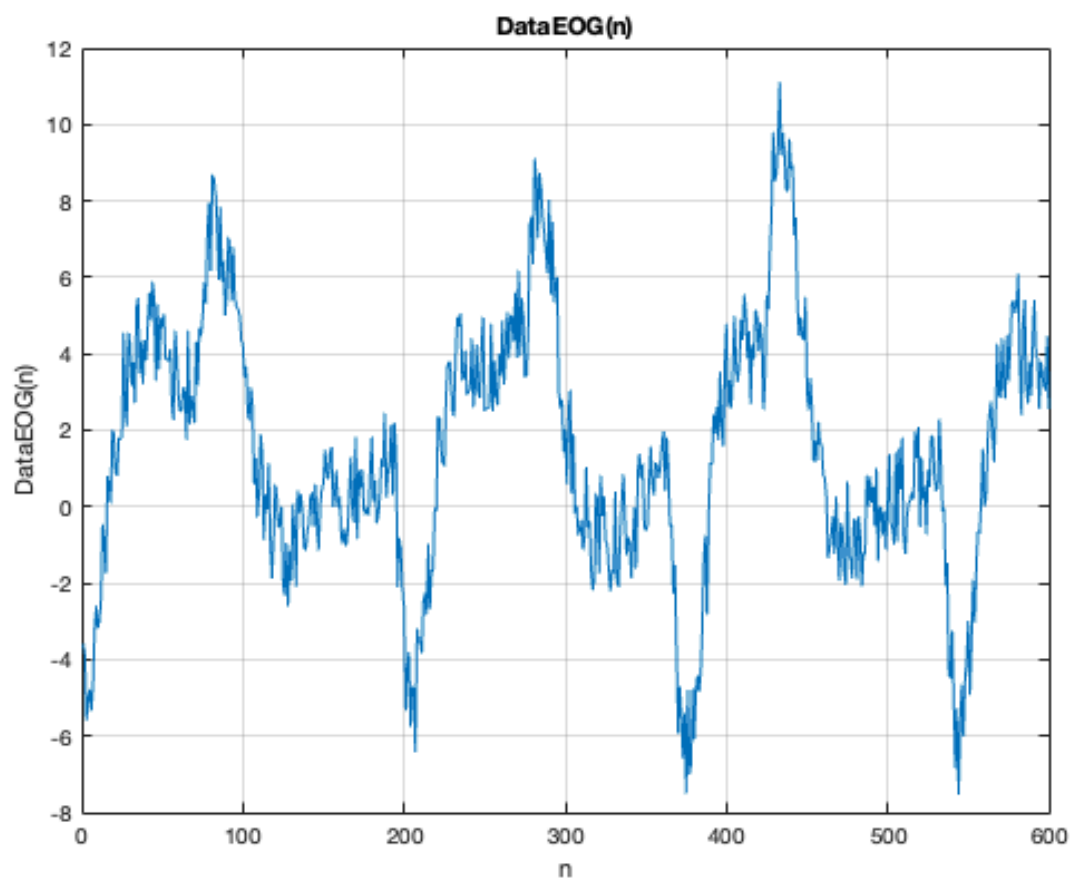
```
y2 = y;  
y2(1:33) = [];  
y2(end-32:end) = [];
```

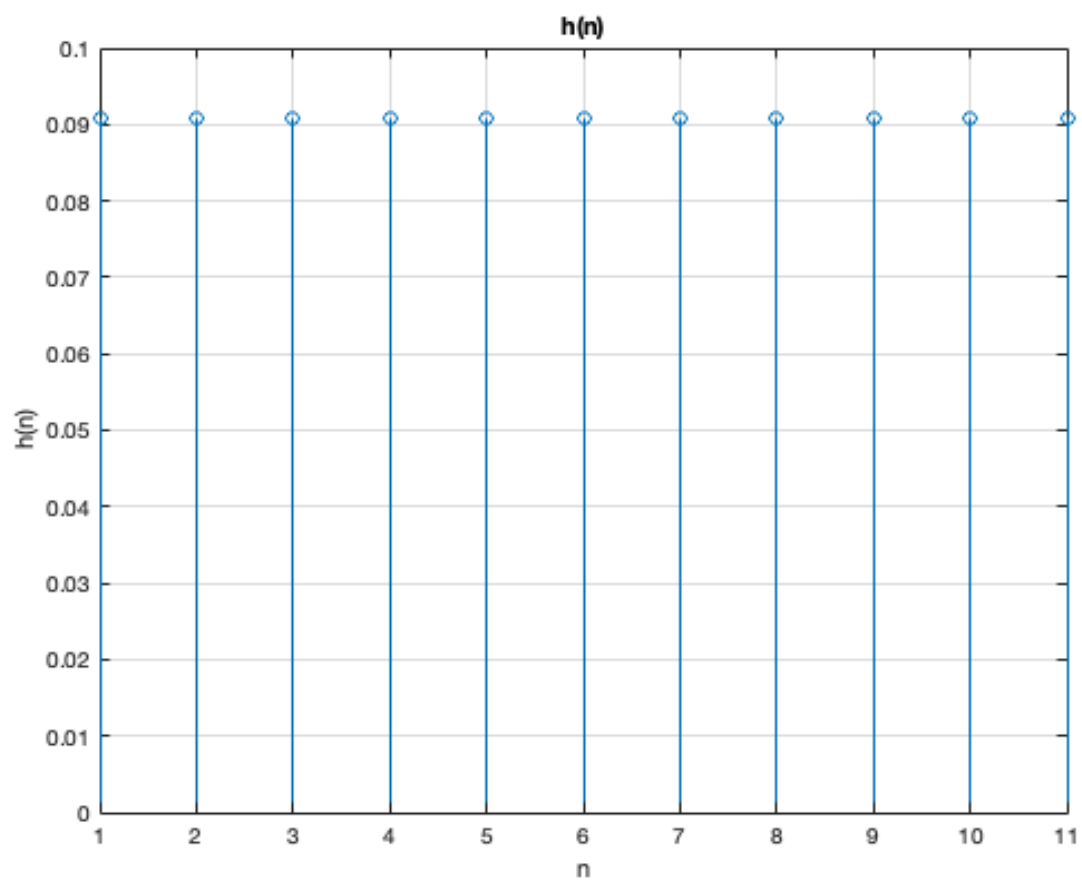
```
% Plotting x(n) and y2(n) on same graph  
figure;  
plot(y2); hold on; plot(x);  
title('y2 and x');  
xlabel('n');  
ylabel('y2 and x');
```

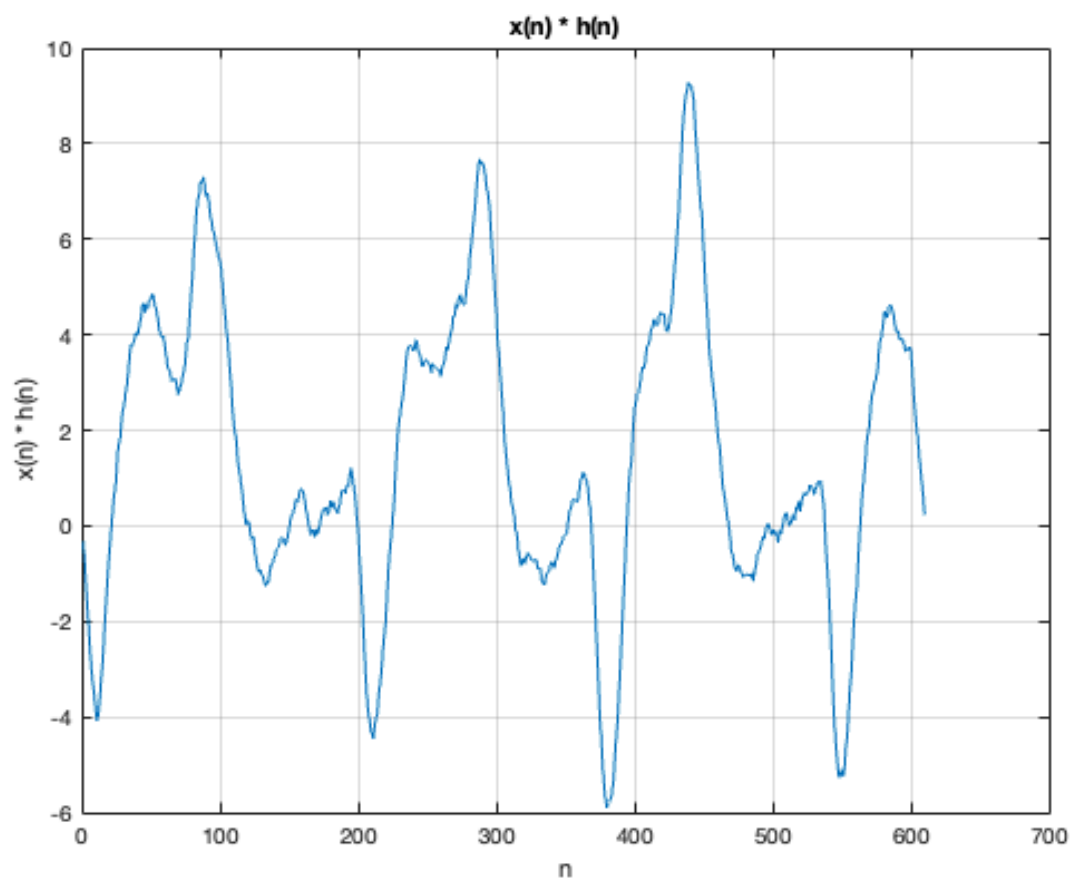
```
grid on;
```

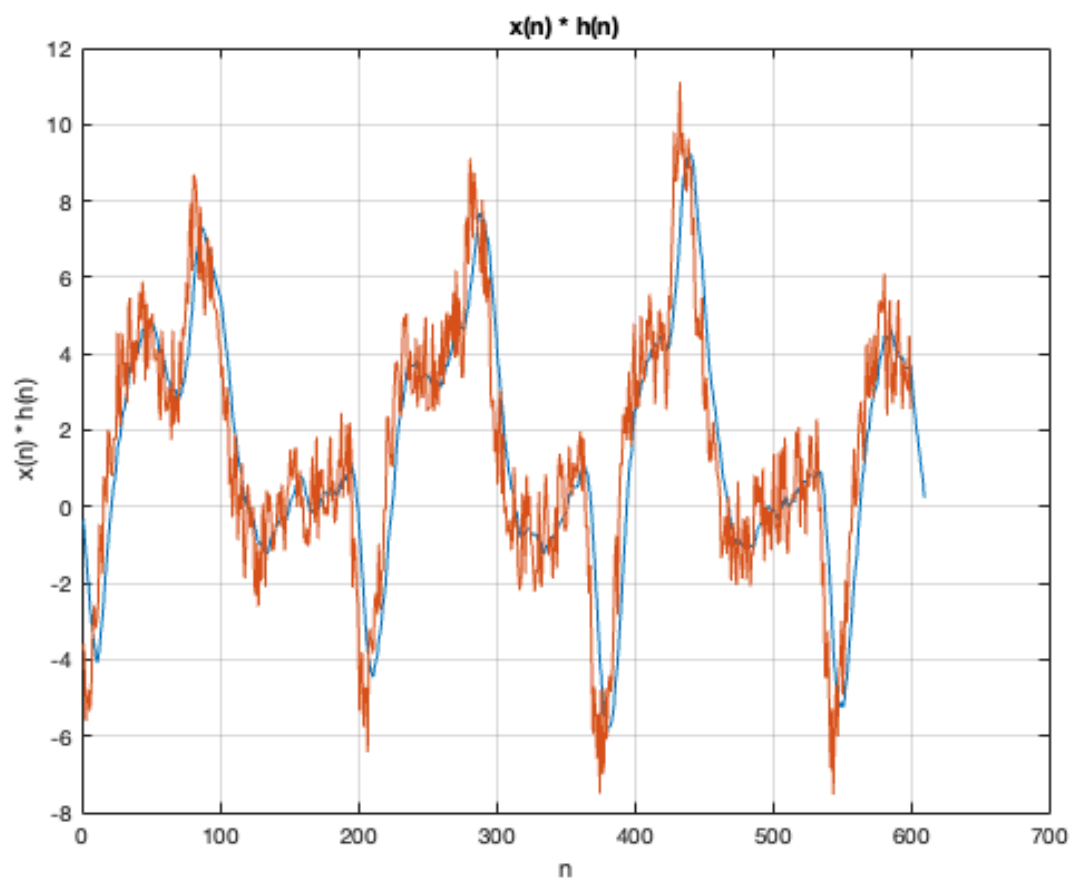
```
% What should the parameters in part (d) be now? 15 and 14 as shown in the  
% script above
```

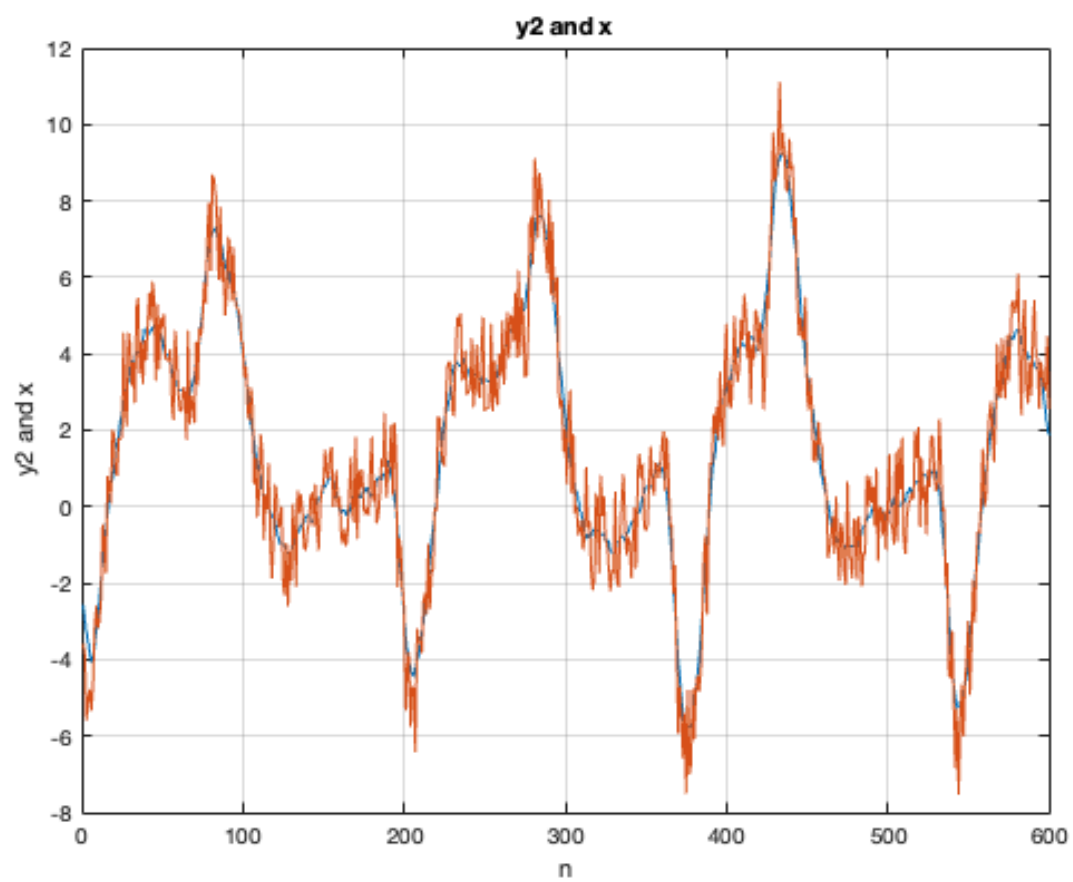


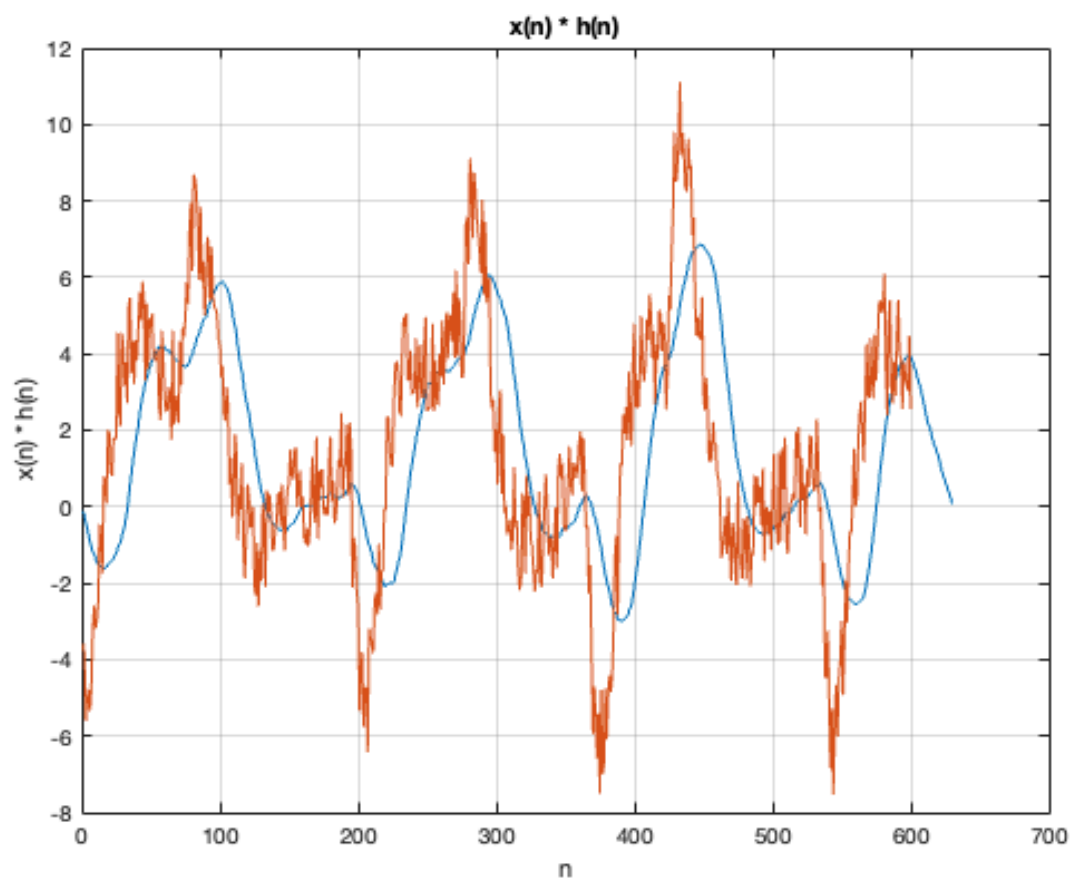


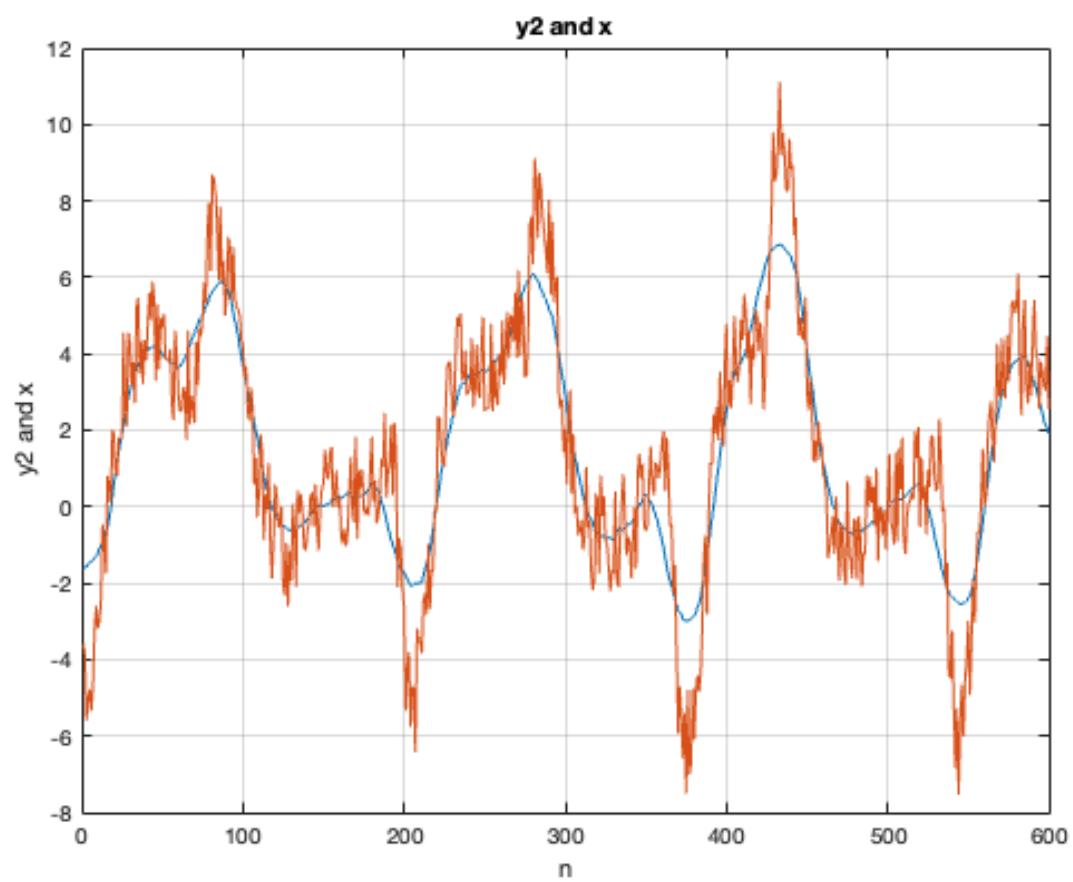


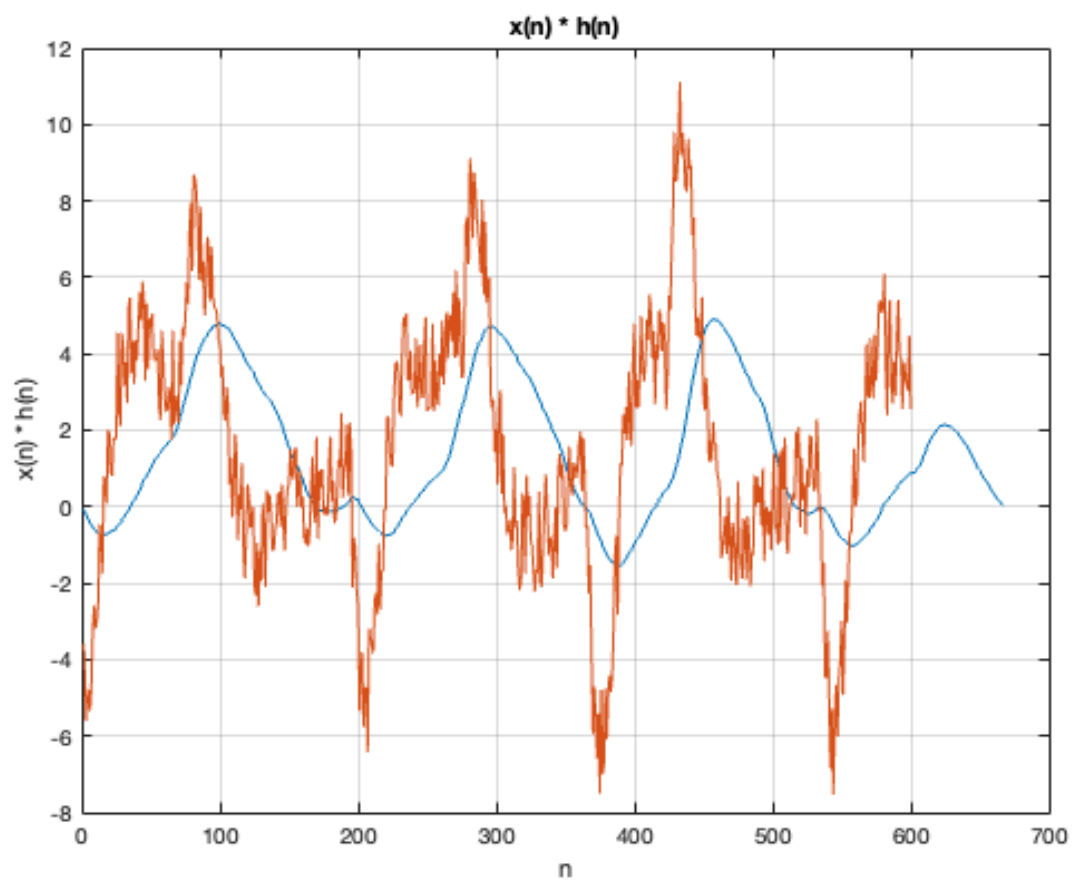


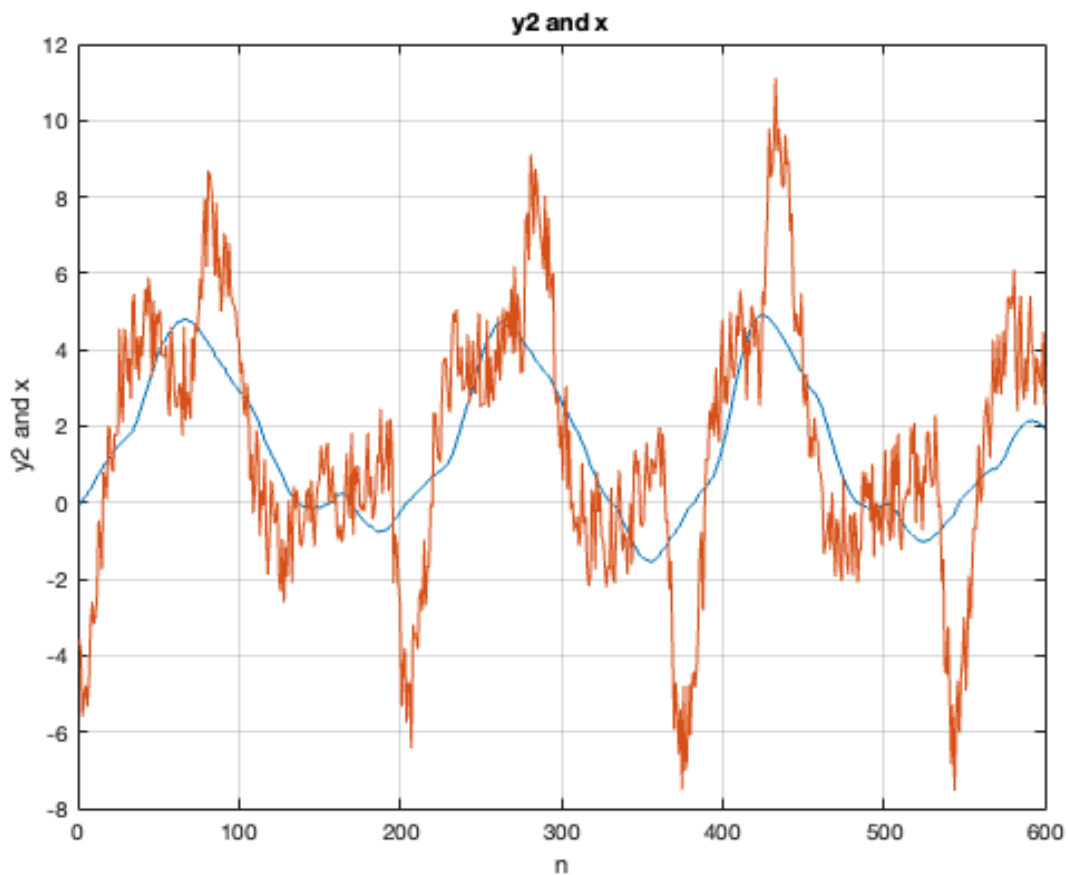












Question 3: Difference Equations

```
% (a) Is this system linear?
% Yes.
% Scalability Property: If we multiply all x and y by c, we can factor out
% the c
% Addition Property: If we add y1 attached to x1 and y2 attached to x2 we
% can factor the x1 and x2 together as well as y1 and y2
% This can be also shown using the superposition property faster
% This can be confirmed in matlab using the signals choosen.
% This is shown in the plotting below where  $y(x1+2*x2) = y(x1)+2*y(x2)$ 
```

```
N=11;
n=0:1:N-1;

x1 = n;
x2 = -n+4;

y1 = mydiffeq(x1,N);
y2 = mydiffeq(x2,N);
y3 = mydiffeq(x1+2*x2,N);
```

```

% Plotting y1, y2, y3 on the same graph
figure;
subplot(4,1,1);
stem(n,y1);
title('y(x1(t))');
xlabel('n');
ylabel('y(x1(t))');
grid on;

subplot(4,1,2);
stem(n,y2);
title('y(x2(t))');
xlabel('n');
ylabel('y(x2(t))');
grid on;

subplot(4,1,3);
stem(n,y3);
title('y(x1(t)+2x2(t))');
xlabel('n');
ylabel('y(x1(t)+2x2(t))');
grid on;

subplot(4,1,4);
stem(n,y1+2*y2);
title('y(x1(t))+2y(x2(t))');
xlabel('n');
ylabel('y(x1(t))+2y(x2(t))');
grid on;

% (b) Is this system time-invariant? Yes, we don't have any variable n
% outside x and y. We can confirm this by plotting y shifted to the right
% for a signal, and then shifting the signal to the right before feeding it
% to y and showing that they give the same output

N=200;
n = -N:1:N-1;
shift = 3;

x = n;
y = mydiffeq(x,N*2);
y(1:shift) = [];
y(end+1:end+3) = [0,0,0];

% Plotting y with normal x
x = n;
y = mydiffeq(x,N*2);
figure;
subplot(3,1,1);
stem(n,y);
axis([0 10 0 20]);
title('y');
xlabel('n');
ylabel('y');

```

```

grid on;

% Plotting y shifted to the left
x1 = n;
y1 = mydiffeq(x1,N*2);
y1(1:shift) = [];
y1(end+1) = 0;
y1(end+1) = 0;
y1(end+1) = 0;
subplot(3,1,2);
stem(n,y1);
axis([0 10 0 20]);
title('shifting y to the right by 3');
xlabel('n');
ylabel('y(t+3)');
grid on;

% Plotting y with shifted x by 3 to the left (x(x+3))
n_shifted = -N+shift:1:N-1+shift;
x1 = n_shifted;
y1 = mydiffeq(x1,N*2);
subplot(3,1,3);
stem(n,y1);
axis([0 10 0 20]);
title('shifting x of y to the right by 3');
xlabel('n');
ylabel('y(x(t+3))');
grid on;

% (c) Compute and plot the impulse response of this system. Use x = [1,
    zeros(1,100)];
% as input.

x = [1, zeros(1,100)];
impulse_response = mydiffeq(x,101);

figure;
stem(impulse_response);
title('h(n)');
xlabel('n');
ylabel('h(n)');
grid on;

% (d) Define x(n). Compute the output of the system in two ways

N=100;
n = 0:1:N;
x = cos((pi/8)*n).*(u(n)-u(n-50));

% (1) y(n) = h(n)*x(n) using the conv command.
y = conv(impulse_response,x);
figure;
subplot(2,1,1);

```

```

plot(y);
axis([0 100 -2 2]);
title('h(n)');
xlabel('n');
ylabel('y(n) using conv command');

% (2) Use your function to find the output for this input signal.
y = mydiffeq(x,N);
subplot(2,1,2);
plot(y);
axis([0 100 -2 2]);
xlabel('n');
ylabel('y(n) using function made');

% Are the two output signals you compute the same? Yes

% (e) Write a new Matlab function for the system with the different
% equation

x = [1, zeros(1,100)];
impulse_response = mydiffeq_2(x,101);
length_impulse_response = 0:length(impulse_response) - 1;

figure;
stem(length_impulse_response,impulse_response);
title('h(n)');
xlabel('n');
ylabel('h(n)');
grid on;
axis([0 20 0 3]);

% (e) What is the difference between the two impulse responses?
% This impulse response is only composed of two impulse response, at 0
% and 1 of height 1 and 2 respectively.

% (f)

x = [1, zeros(1,100)];
impulse_response = mydiffeq_3(x,101);
length_impulse_response = 0:length(impulse_response) - 1;

figure;
stem(length_impulse_response,impulse_response);
title('h(n)');
xlabel('n');
ylabel('h(n)');
grid on;
axis([0 20 -6 6]);

% (f) What do you find? Discuss your observations.
% amplitude keeps increasing since the coefficient of y(n-1) is more than 1
% unlike the first case. So the amplitude increases as every y is more than
% the one before it.

```

```
function y = mydiffeq(x,N)
    % Initialize the output vector y
    y = zeros(1, N);

    % Initialize the initial values
    x_minus_1 = 0;
    y_minus_1 = 0;

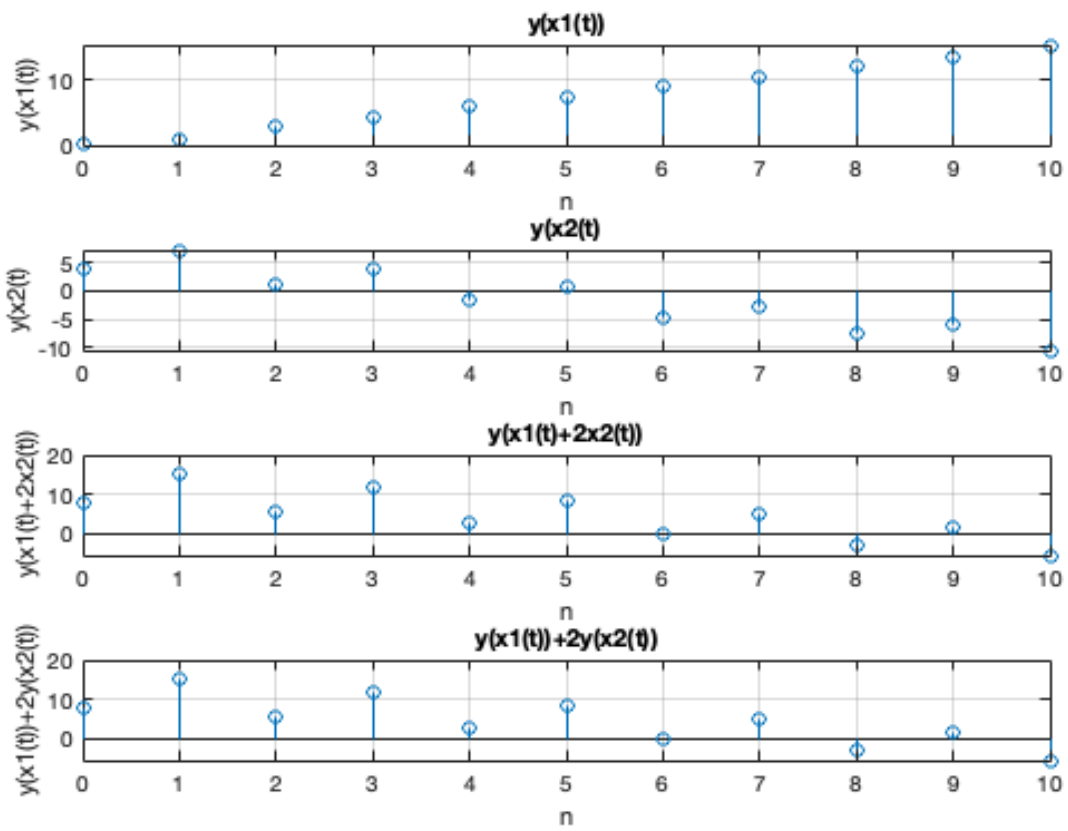
    % Implement the difference equation using a for loop
    for n = 1:N
        y(n) = x(n) + 2 * x_minus_1 - 0.95 * y_minus_1;
        % Update the past values for the next iteration
        x_minus_1 = x(n);
        y_minus_1 = y(n);
    end
end

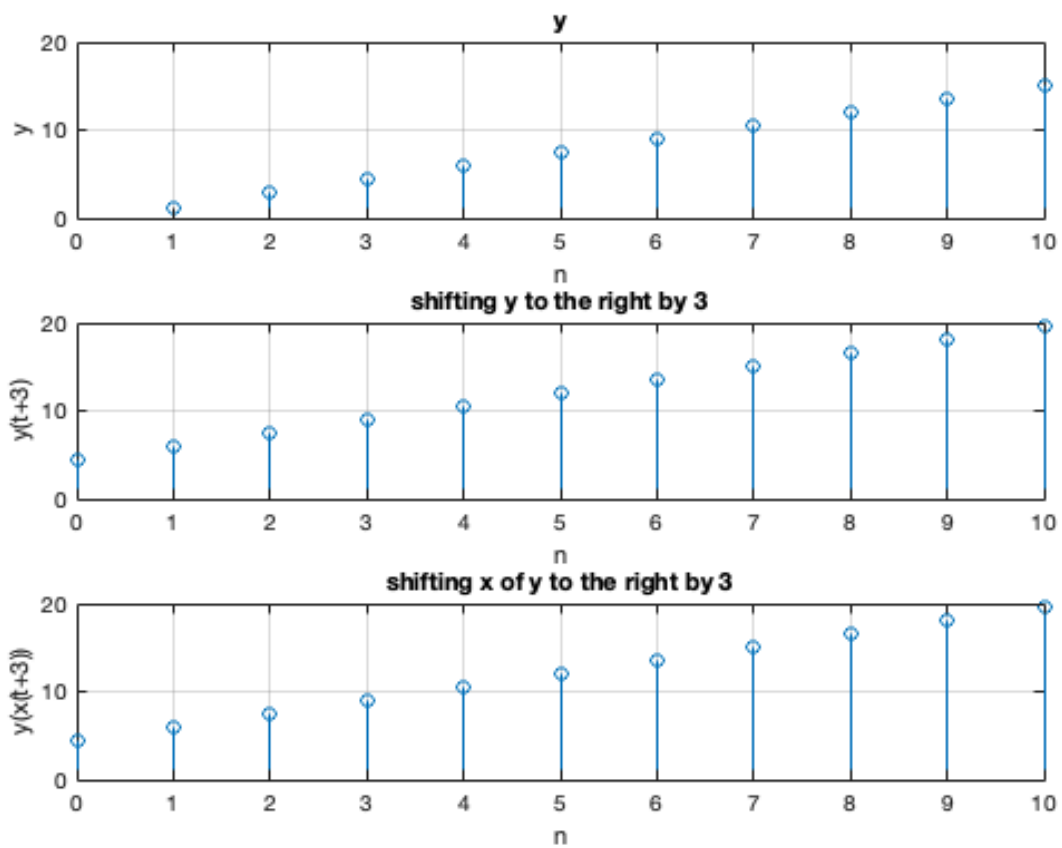
function y = mydiffeq_2(x,N)
    % Initialize the output vector y
    y = zeros(1, N);
    y(1) = x(1);
    % Implement the difference equation using a for loop
    for n = 2:N
        y(n) = x(n)+2*x(n-1);
    end
end

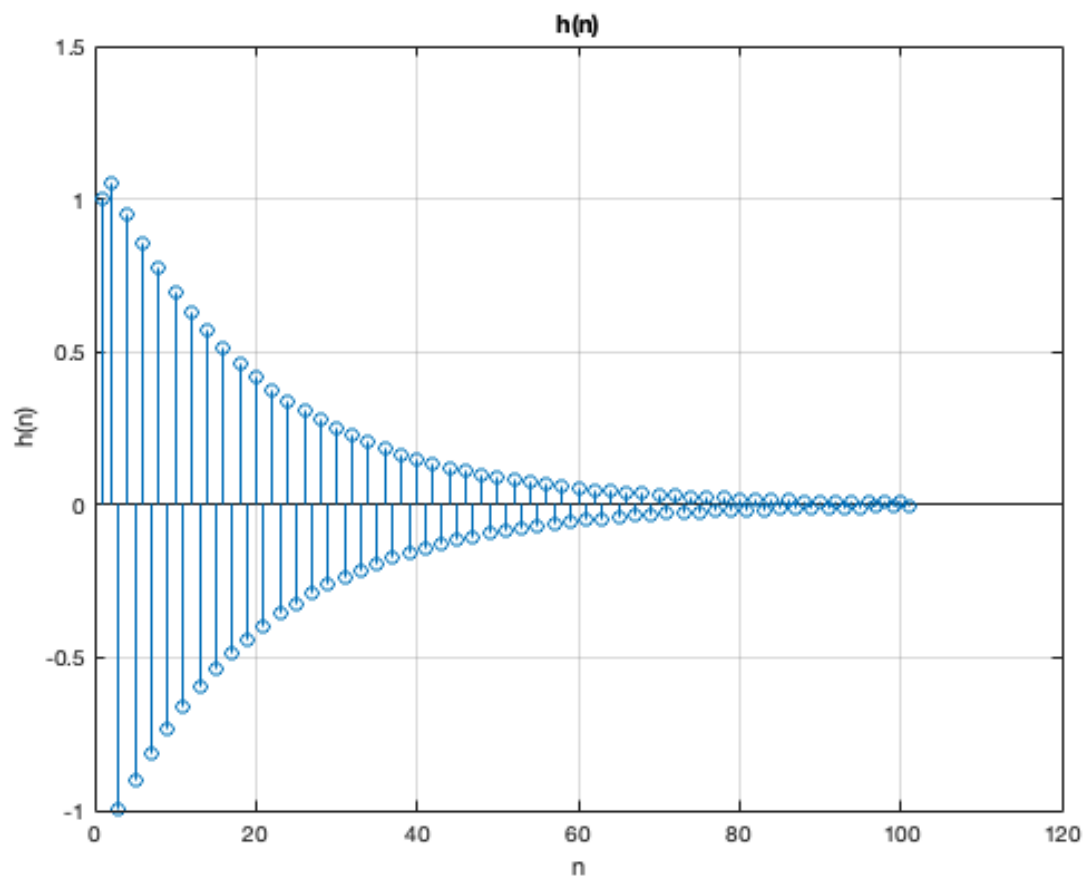
function y = mydiffeq_3(x,N)
    % Initialize the output vector y
    y = zeros(1, N);

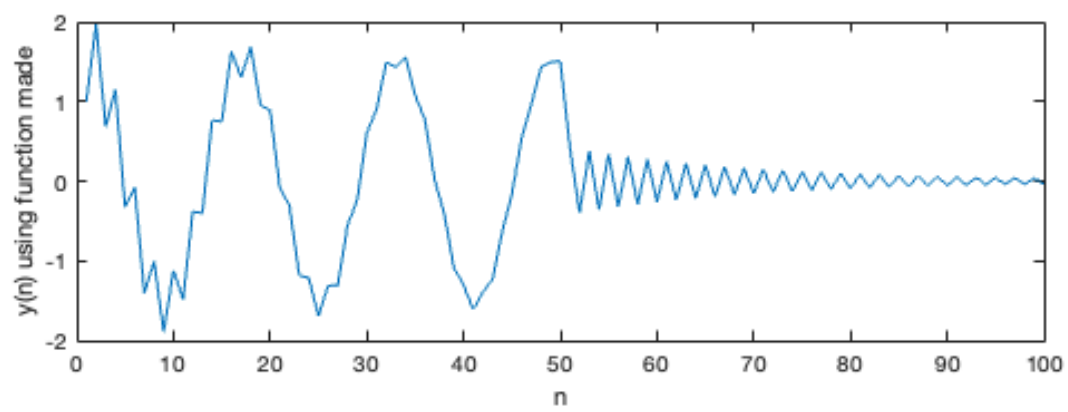
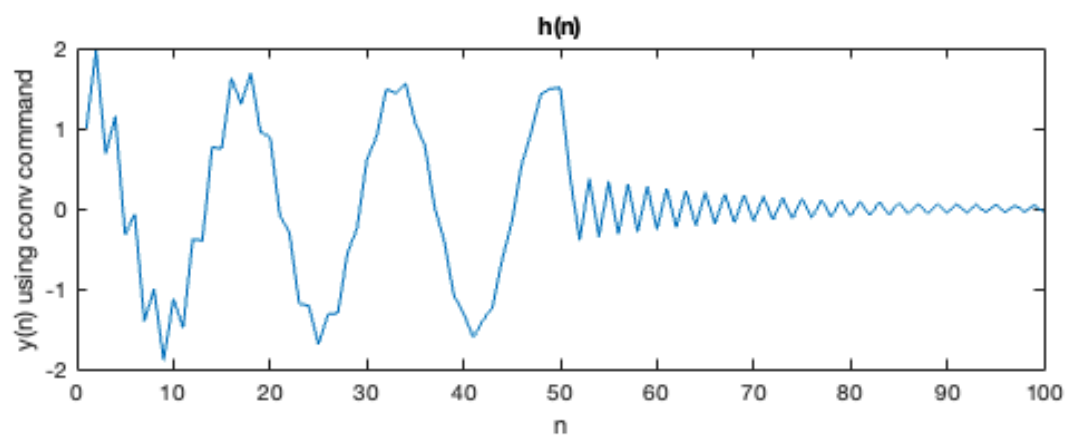
    % Initialize the initial values
    x_minus_1 = 0;
    y_minus_1 = 0;

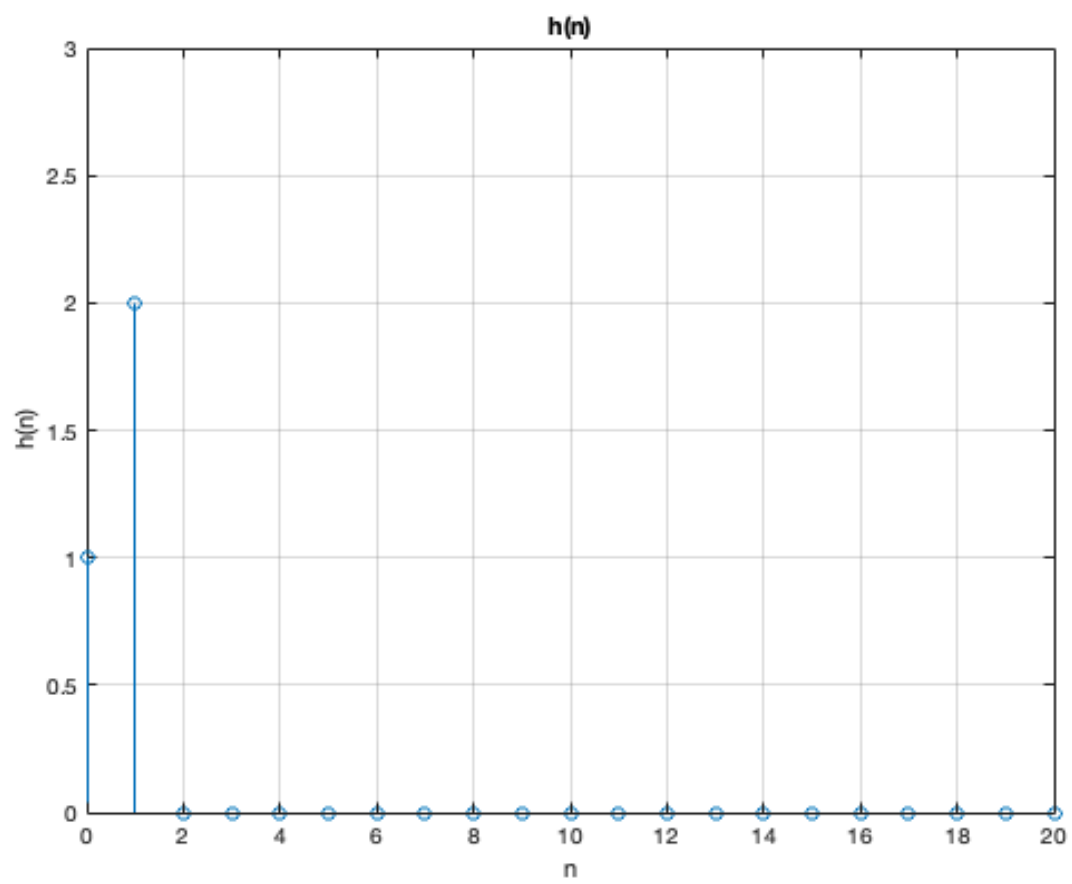
    % Implement the difference equation using a for loop
    for n = 1:N
        y(n) = x(n) + 2 * x_minus_1 - 1.1 * y_minus_1;
        % Update the past values for the next iteration
        x_minus_1 = x(n);
        y_minus_1 = y(n);
    end
end
```

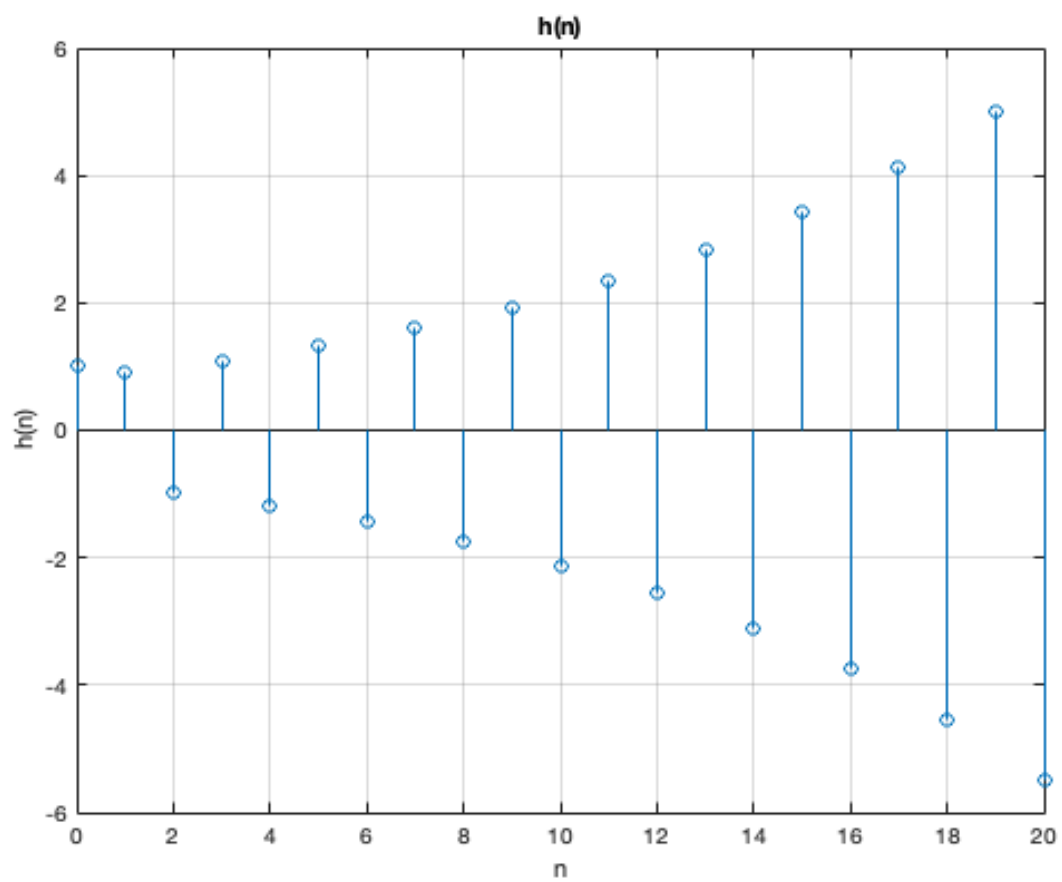












Published with MATLAB® R2023a