Table of Contents

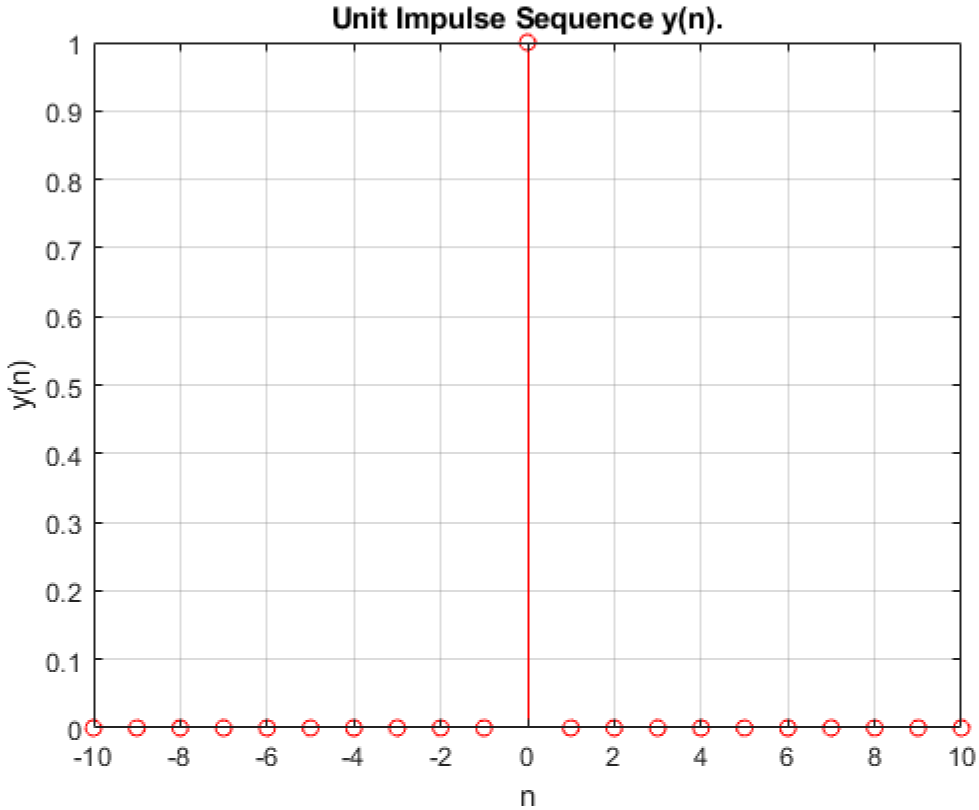
Dan Otieno. EE 384 -> Spring '24. Classwork 1. Due date: 01/23/24.

Problem 1.

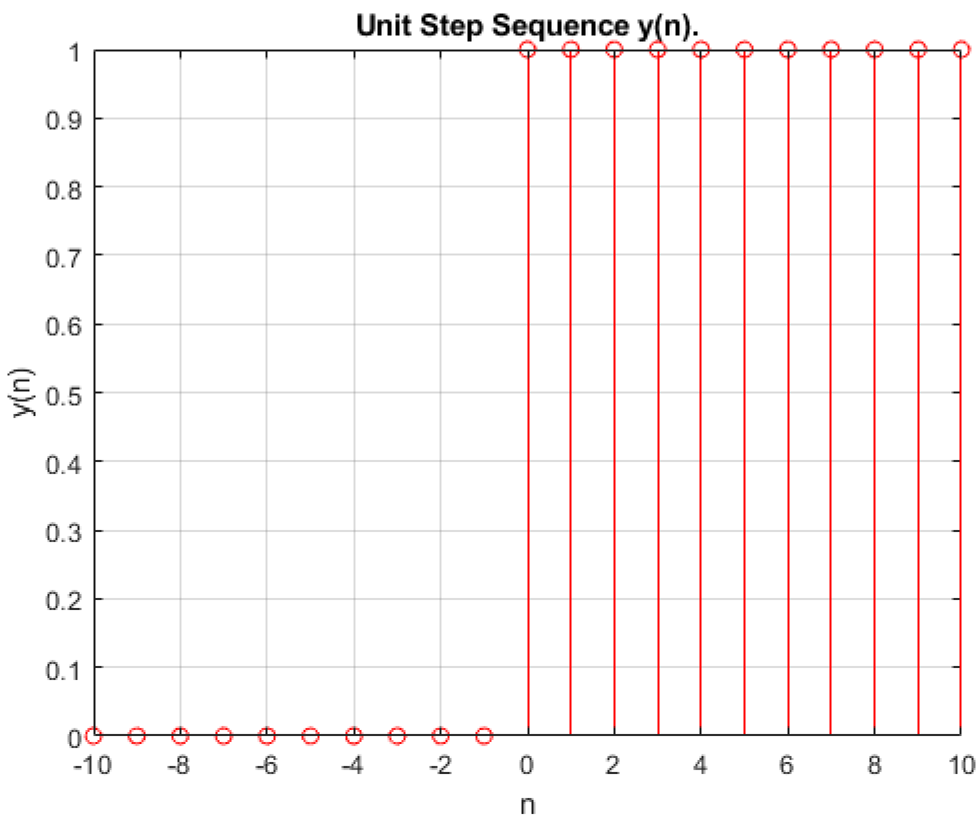
Q1.1: Unit Impulse. 'unit_impulse' function that creates a unit impulse sequence y(n). The function accepts input arguments of n_p, n1, and n2. n1 and n2 are the start and the end of the sequence. n_p is the position of the pulse.

```
[y, n] = unit_impulse(0, -10, 10);
figure(1);
% Plot/Display Results.
stem(n, y, 'r');
grid on;
xlabel('n');
ylabel('y(n)');
title('Unit Impulse Sequence y(n).');

% Q1.2: Unit Step.
% 'unit_step' function that creates a unit step sequence y(n).
% The function accepts input arguments of n_s, n1, and n2.
% n1 and n2 are the start and the end of the sequence.
% n_s is the position of the step.
[y, n] = unit_step(0, -10, 10);
figure(2);
% Plot/Display Results.
stem(n, y, 'r');
grid on;
xlabel('n');
ylabel('y(n)');
title('Unit Step Sequence y(n).');
```

==



Unit Impulse Sequence y(n).
y(n)
n



Unit Step Sequence y(n).
y(n)
n

==

Problem 2.

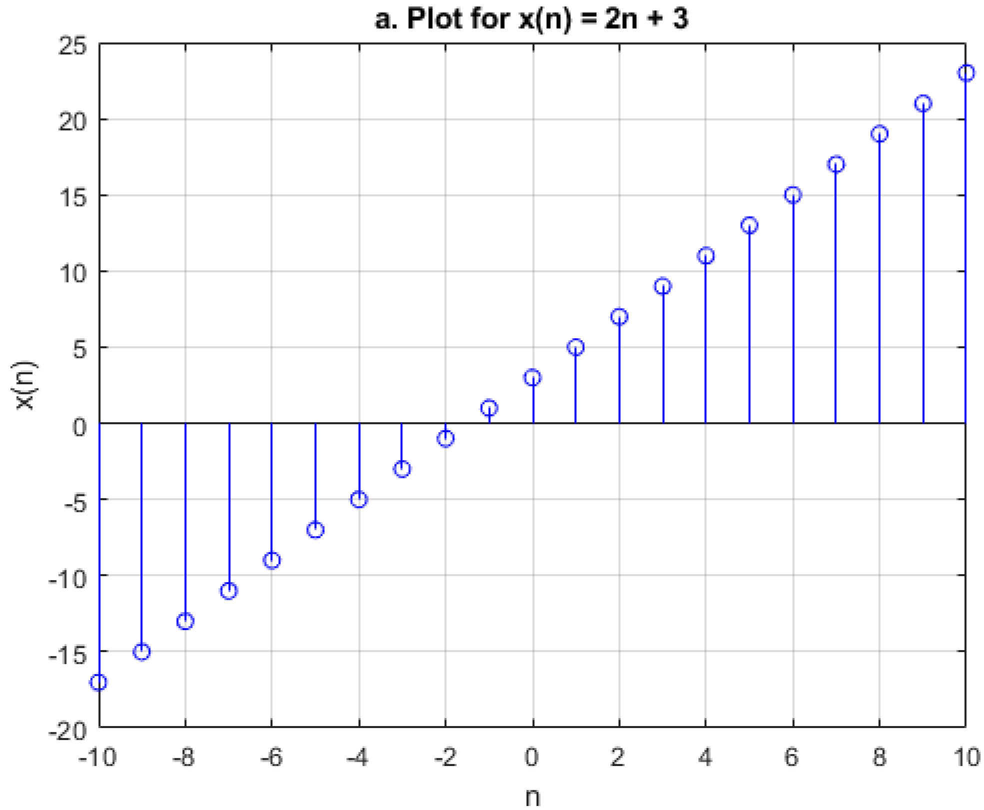
Q2.1: Time-Shift function. .time_shift. function that create an output sequence y(n)... ...as a delayed version of the input sequence x(n). Function should accept input arguments of x, n, and n_d. n_d is the number of samples delayed.

```
% Time indices.
n = 0:10;
% Input sequence.
x = [1, 2, 3, 4, 5, 4, 3, 2, 1, 0, -1];
% Number of samples delayed.
n_d = 3;
y = time_shift(x, n, n_d);

% Print outputs.
[y,z] = time_shift(x, n, n_d);
fprintf("Question 2.1:\n");
disp('Original sequence x(n):');
disp(x);
disp('Time indices for x(n):');
disp(n);
disp('Delayed sequence y(n):');
disp(y);
disp('Time indices for y(n):');
disp(z);

% Q2.2: Let x(n) = 2n + 3, n = -10:10.
n2 = -10:10;
x2 = 2*n2 + 3;

% a). Plot x(n).
figure(3);
stem(n2, x2, 'b');
title('a. Plot for x(n) = 2n + 3');
grid on;
xlabel('n');
ylabel('x(n)');

% b). Plot a time delayed version of x(n) delayed by 3 samples.
[y, z] = time_shift(x2, n2, 3);
figure(4);
stem(z, y, 'b');
title('b. Plot for x(n) delayed by 3 samples: x(n - 3)');
grid on;
xlabel('n');
ylabel('x(n - 3)');

% c). Plot the time-reversal of x(n).
[y,z] = time_reverse(x2, n2);
figure(5);
stem(z, y, 'b');
title('c. Plot for time-reversal of x(n): x(-n)');
grid on;
xlabel('n');
```

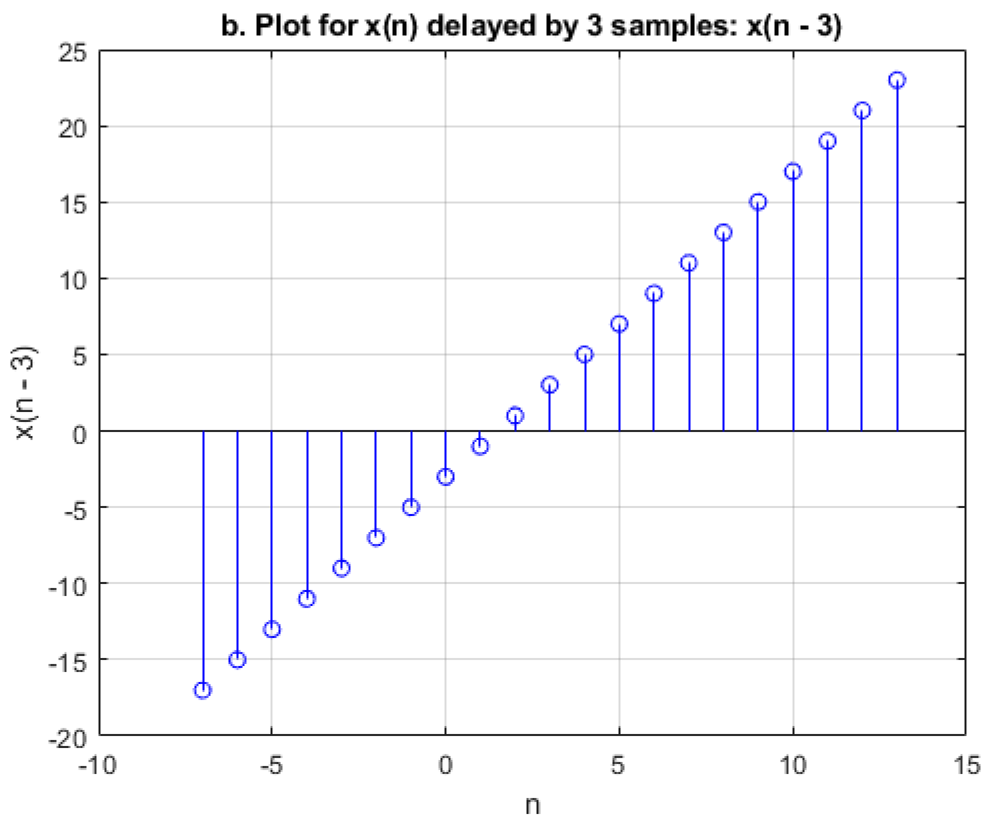
```
ylabel('x(-n)');

% Q2.3: Plot y(n) = 5*d(n + 4) – 2* d(n -2); n = -10:10.
% where d(n) is the unit impulse sequence.
[y1, ~] = unit_impulse(-4, -10, 10);
[y2, z] = unit_impulse(2, -10, 10);
y = 5 * y1 - 2 * y2;
figure(6);
stem(z, y, 'm');
title('Plot for Question 2.3');
grid on;
xlabel('n');
ylabel('y(n)');

% Q2.4: Plot z(n) = u(n) – u(n-4) ; n = -10:10.
% where u(n) is the unit step sequence.
z = unit_step(0, -10, 10) - unit_step(4, -10, 10);
figure(7);
stem(n2, z, 'm');
title('Plot for Question 2.4');
grid on;
xlabel('n');
ylabel('z(n)');
```

```
Question 2.1:
Original sequence x(n):
     1     2     3     4     5     4     3     2     1     0    -1

Time indices for x(n):
     0     1     2     3     4     5     6     7     8     9    10

Delayed sequence y(n):
     1     2     3     4     5     4     3     2     1     0    -1

Time indices for y(n):
     3     4     5     6     7     8     9    10    11    12    13
```

==

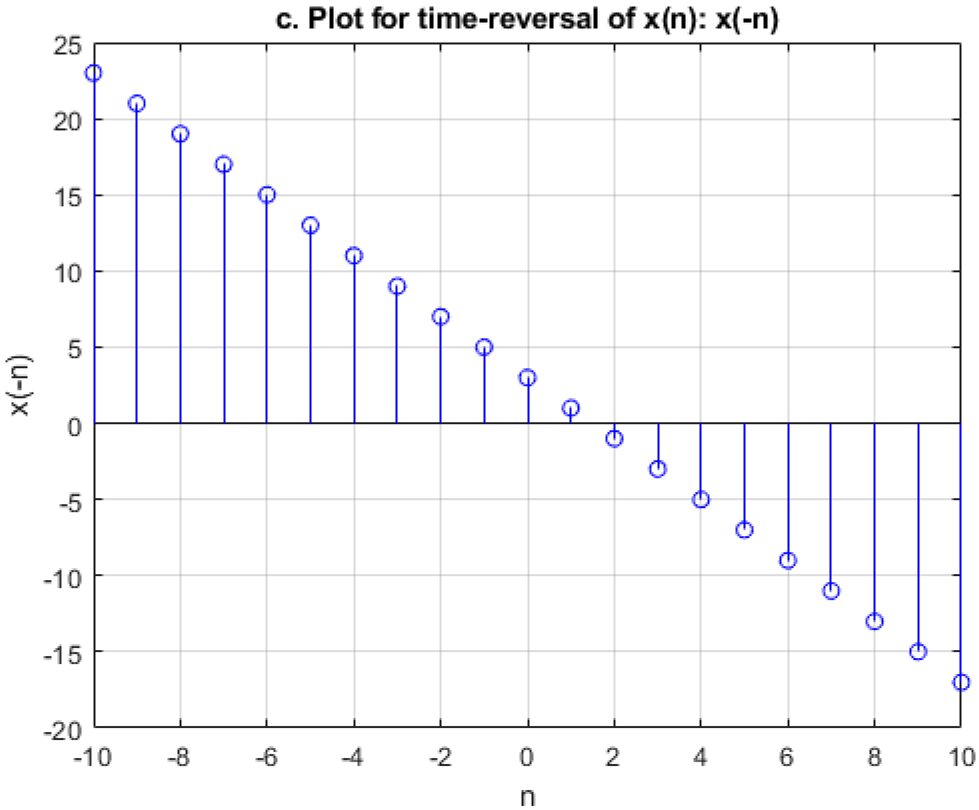


a. Plot for x(n) = 2n + 3
x(n)
n



b. Plot for x(n) delayed by 3 samples: x(n - 3)
x(n - 3)
n

==


c. Plot for time-reversal of x(n): x(-n)
x(-n)
n
25
20
15
10
5
0
-5
-10
-15
-20
-10
-8
-6
-4
-2
0
2
4
6
8
10


Plot for Question 2.3
y(n)
n
5
4
3
2
1
0
-1
-2
-10
-8
-6
-4
-2
0
2
4
6
8
10

==

Problem 3.

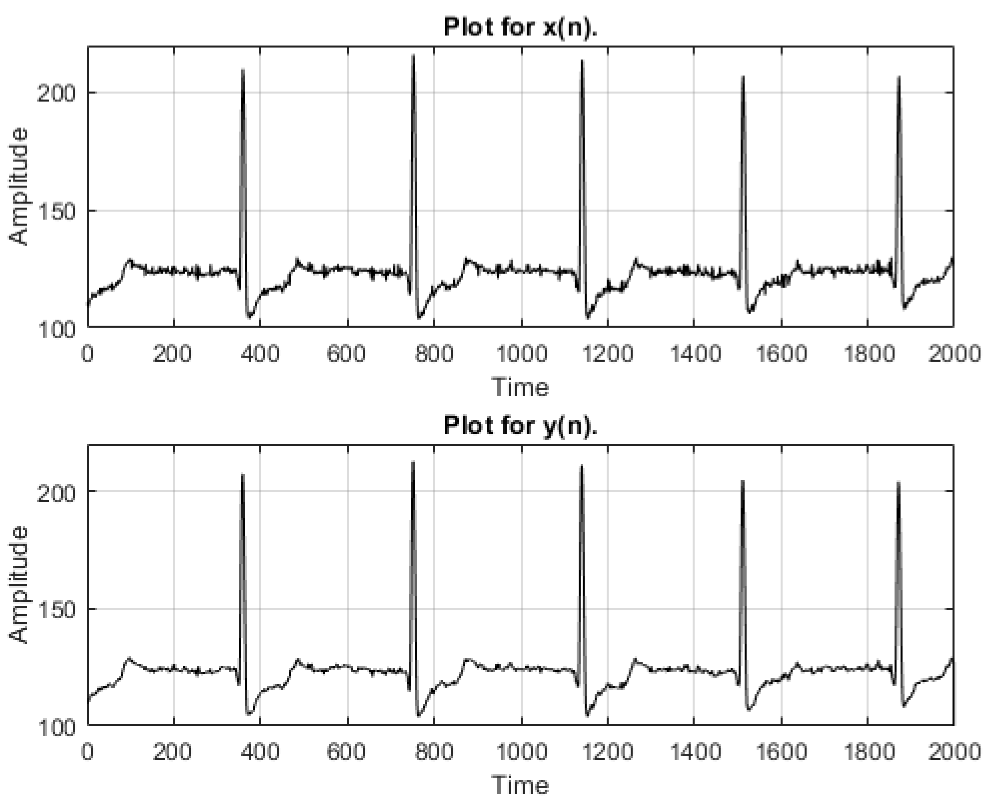
Using additional file "SAMPLE_ECG.mat" Q3.1: Load and plot signal, call it x(n).

```
load SAMPLE_ECG;
x = ECG_Data;

% Q3.2: Using for loop, write a program that creates an output y(n)...
% ...where y(n) = x(n) + x(n+1) + x(n+2) / 3
y = zeros(size(x));
% Loop.
for i = 1:(length(x)-2)
    y(i) = (x(i) + x(i+1) + x(i+2)) / 3;
end

% Q3.3: Plot both x(n) and y(n) in a same figure using the subplot command.
% The vertical axis varies from 100 to 200 for both plots.
% The horizontal axis varies from 0 to 2000 for both plots.
% Using the command axis([0 2000 100 220]).

figure(8);
subplot(2,1,1);
plot(x, 'k');
title('Plot for x(n).');
axis([0 2000 100 220]);
xlabel('Time');
```

```
==================================================================
```

```
ylabel('Amplitude');
grid on;

subplot(2,1,2);
plot(y, 'k');
title('Plot for y(n).');
axis([0 2000 100 220]);
xlabel('Time');
ylabel('Amplitude');
grid on;

% Q3.4: Does y(n) look smoother than x(n)?
fprintf("=================================================================
\n");
fprintf("Question 3.4:\n");
fprintf("Yes, the signal for y(n) looks smoother than the x(n) signal.")
fprintf("\n=================================================================
\n");
```

```
=================================================================
Question 3.4:
Yes, the signal for y(n) looks smoother than the x(n) signal.
=================================================================
```



===

Functions.

Unit Impulse.

```
function [y, n] = unit_impulse(n_p, n1, n2)
    n = n1:n2;
    y = (n == n_p);
end
% Unit Step.
function [y, n] = unit_step(n_s, n1, n2)
    n = n1:n2;
    y = (n >= n_s);
end
% Time shift.
function [y, z] = time_shift(x, n, n_d)
    y = x;
    z = n_d + n;
end
% Time Reverse.
function [y, z] = time_reverse(x, n)
    y = fliplr(x);
    z = n;
end
```

Published with MATLAB® R2023a