

Nolan Anderson

EE 384 Classwork 1 Due 29 August 2021

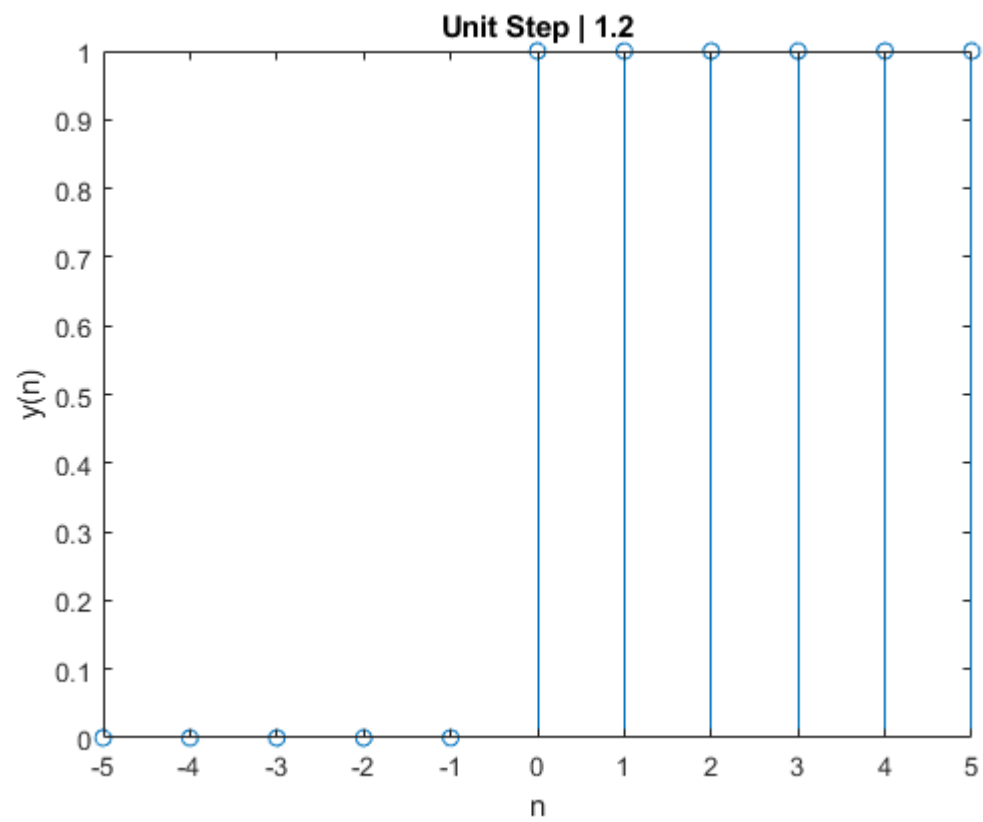
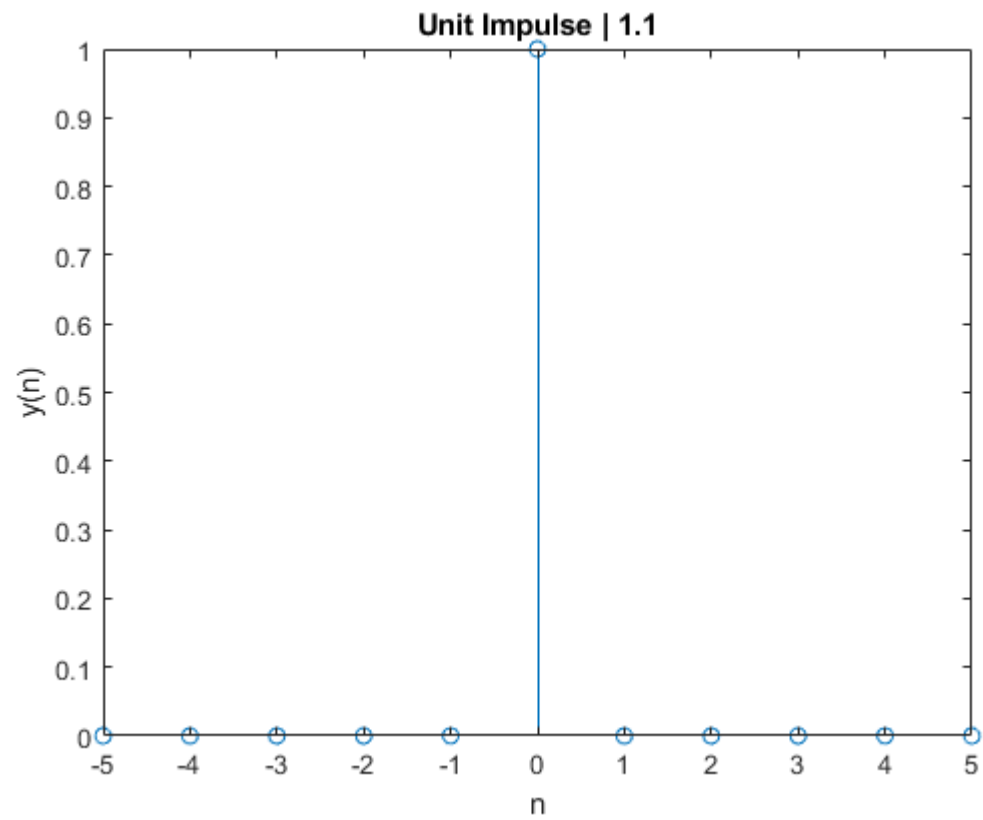
Functions are at the bottom of report

Notes

```
clc  
clear all  
close all
```

Problem 1

```
% 1.1 Unit Impulse  
n_p = 0; n1 = -5; n2 = 5;  
[y, n] = u_impulse(n_p, n1, n2);  
figure  
stem(n,y), xlabel('n'), ylabel('y(n)')  
title('Unit Impulse | 1.1')  
  
% 1.2 Unit Step  
n_s = 0; n1 = -5; n2 = 5;  
[y, n] = u_step(n_s, n1, n2);  
figure  
stem(n,y), xlabel('n'), ylabel('y(n)')  
title('Unit Step | 1.2')
```



Problem 2

```
% 2.1 see Functions section.
```

```
% 2.2
```

```
n = -10:10; x = 2 * n + 3; figure
stem(n, x), xlabel('n'), ylabel('x'), title('x(n) = 2n + 3 | 2.2a')
```

```
[x_2, n] = time_shift(x, n, 3); figure
stem(n, x_2), xlabel('n'), ylabel('x'), title('x(n) = 2(n+3) + 3 | 2.2b')
```

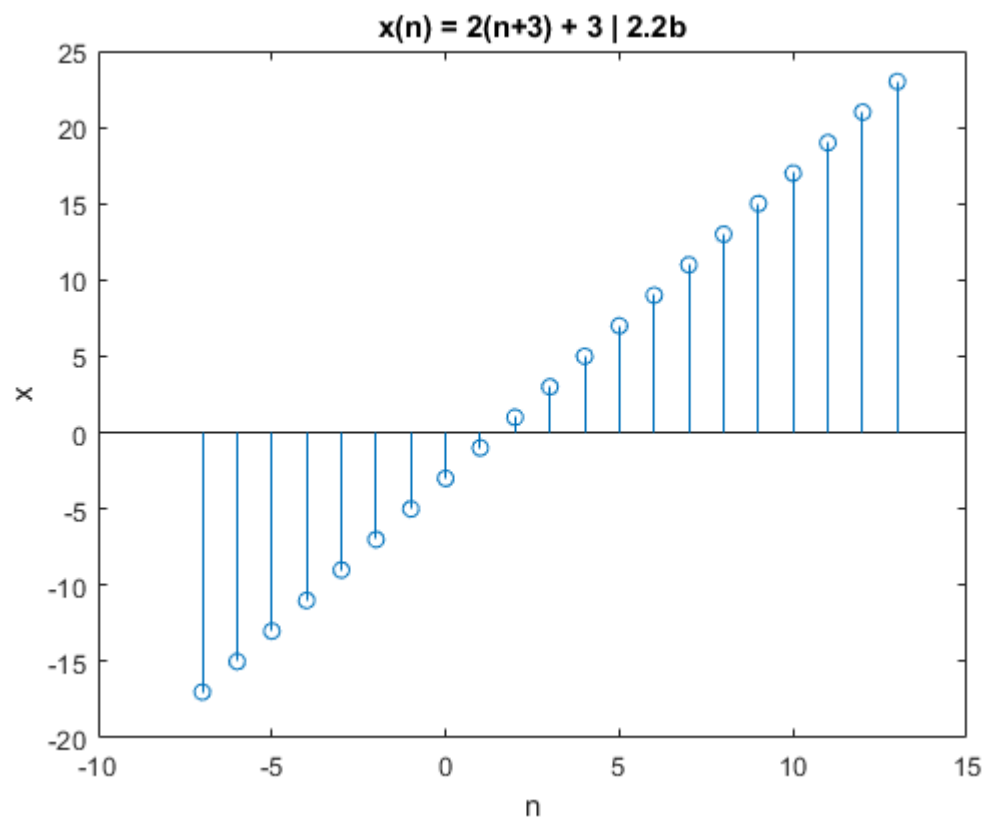
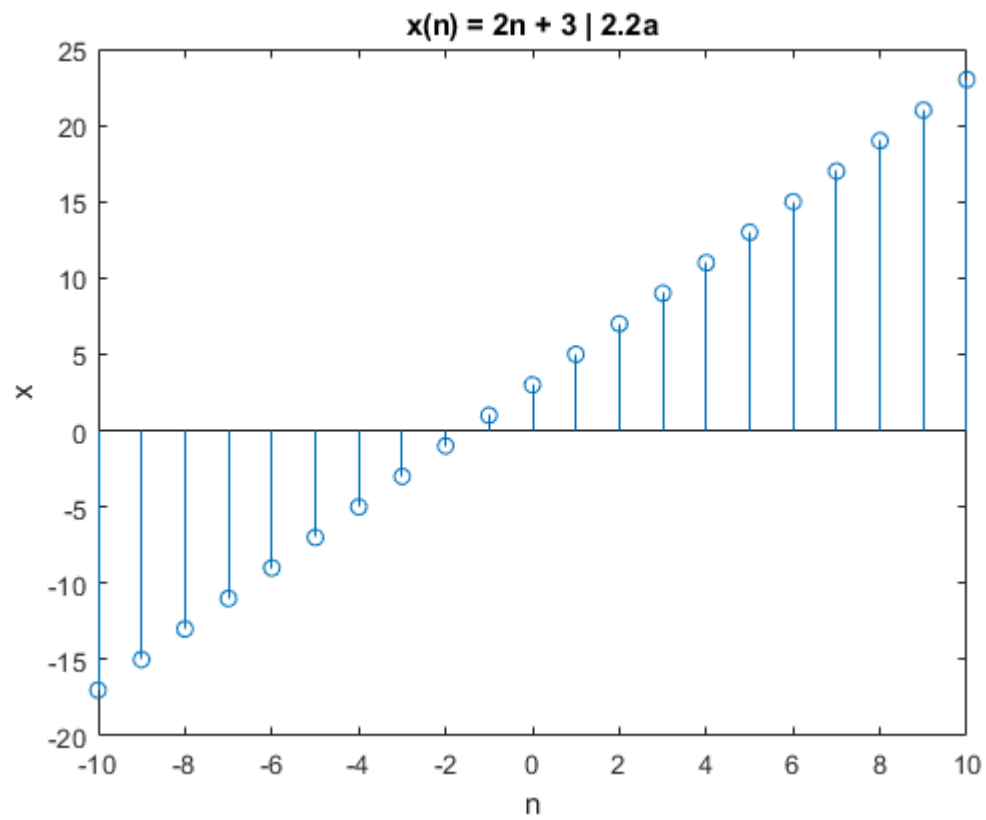
```
[x_3, n] = time_reverse(x, n); figure
stem(n, x_3), xlabel('n'), ylabel('x'), title('x(n) = -2n + 3 | 2.2c')
```

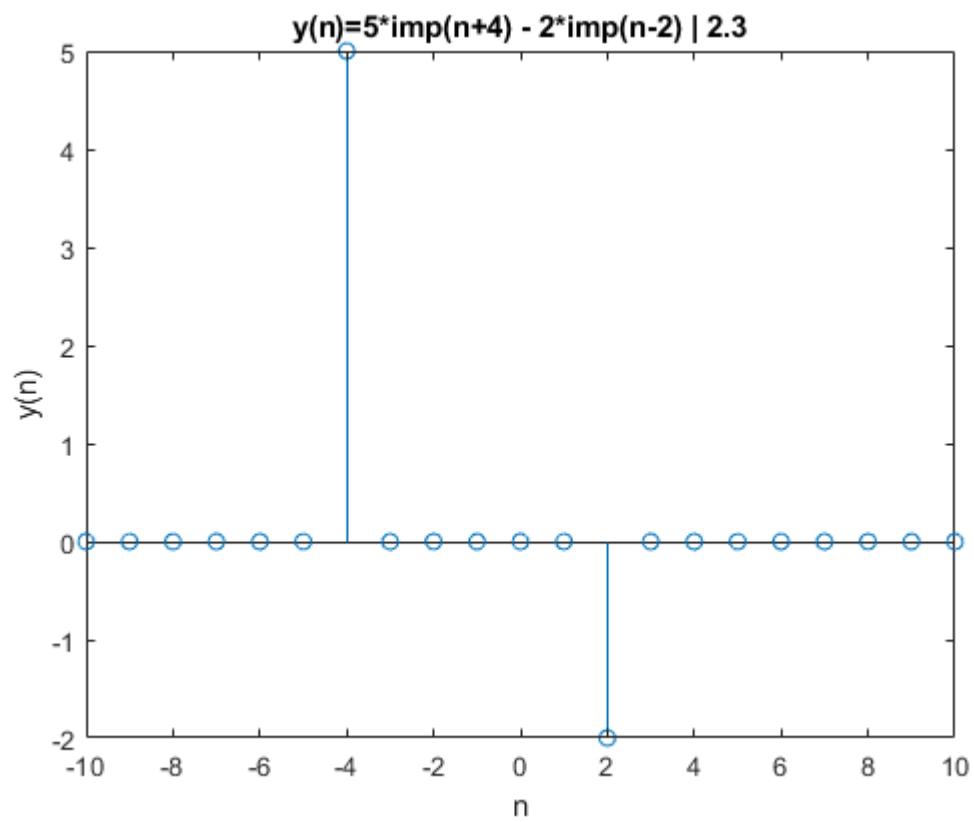
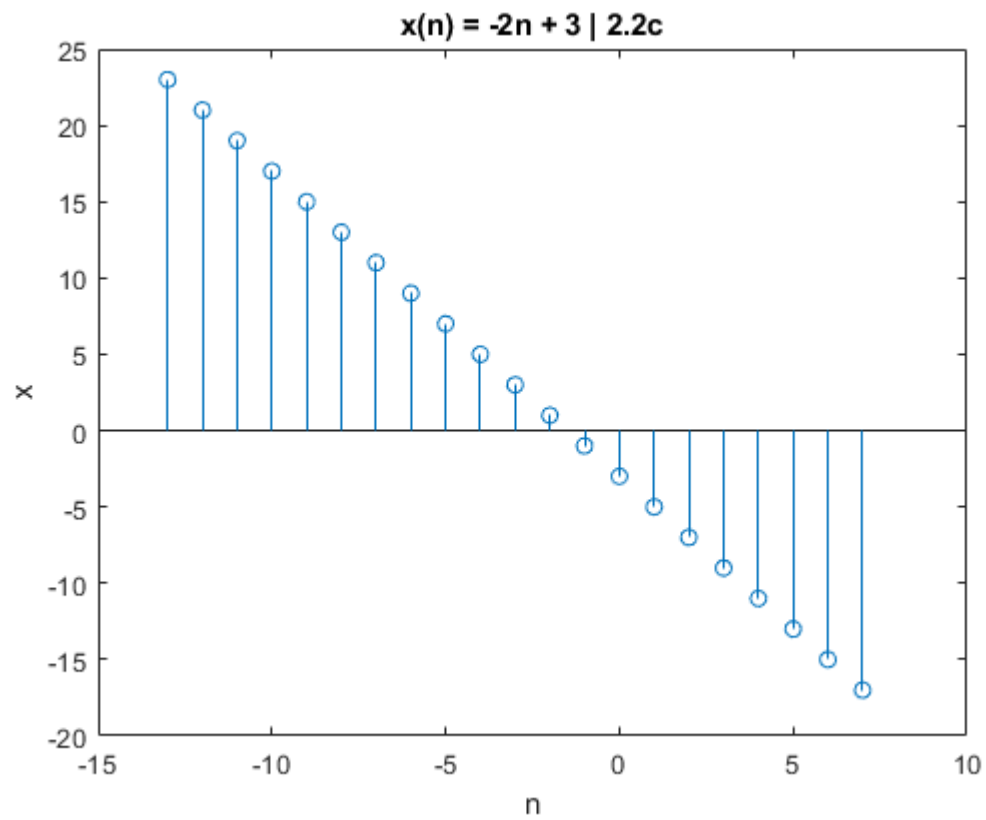
```
% 2.3
```

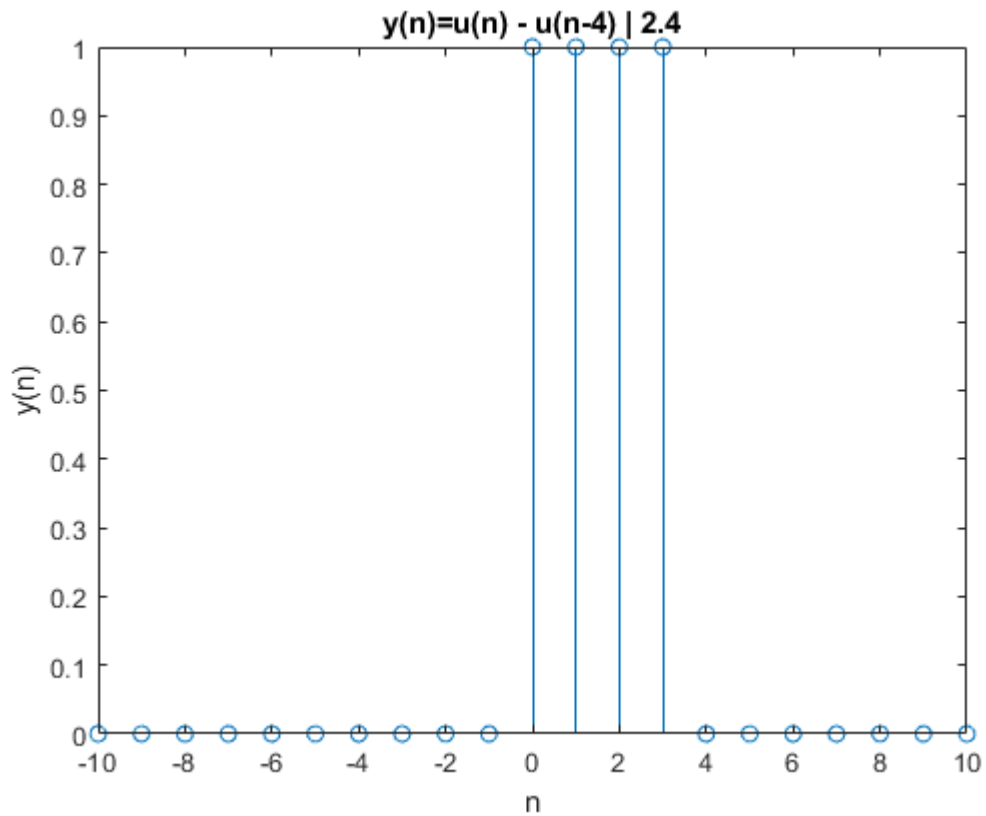
```
figure
n = -10:10;
[y1, m] = u_impulse(-4, -10, 10);
[y2, m] = u_impulse(2, -10, 10);
y = 5*y1 - 2*y2;
stem(n, y), title('y(n)=5*imp(n+4) - 2*imp(n-2) | 2.3')
ylabel('y(n)'), xlabel('n')
```

```
% 2.4
```

```
figure
[z1, m] = u_step(0, -10, 10);
[z2, m] = u_step(4, -10, 10);
z = z1 - z2;
stem(n, z), title('y(n)=u(n) - u(n-4) | 2.4'), xlabel('n'), ylabel('y(n)')
```





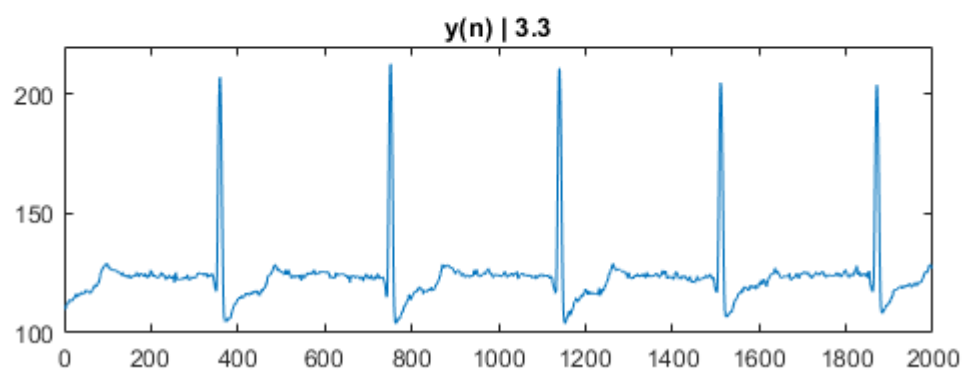
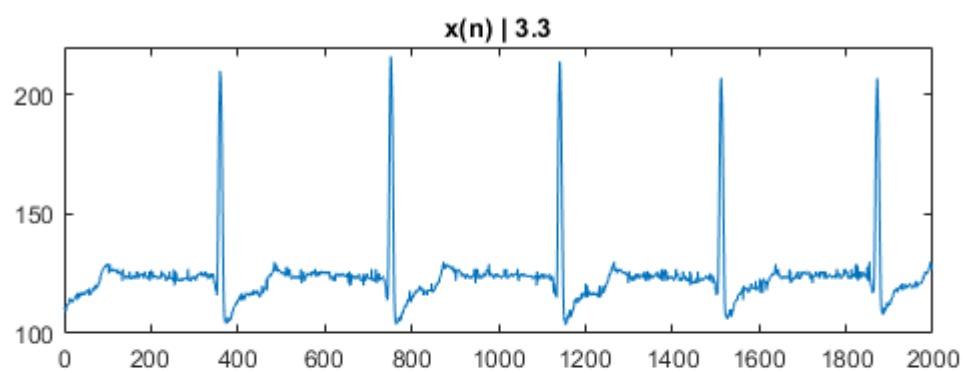
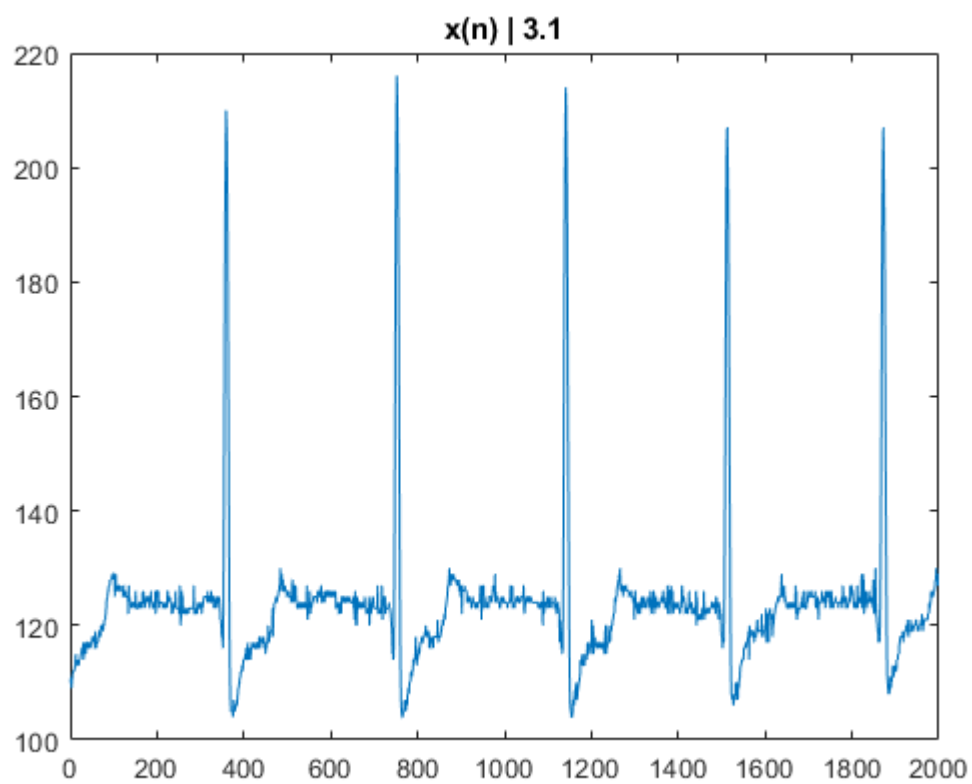


Problem 3

```
% 3.1
load('SAMPLE_ECG.mat');
x = ECG_Data;
figure
plot(x), title('x(n) | 3.1'), axis([0 2000 100 220])
% 3.2
for i = 1 : length(x) - 2
    y(i) = (x(i) + x(i + 1) + x(i + 2)) / 3;
end

% 3.3
figure
subplot(2, 1, 1), plot(x), title('x(n) | 3.3'), axis([0 2000 100 220])
subplot(2, 1, 2), plot(y), title('y(n) | 3.3'), axis([0 2000 100 220])

% 3.4
% Yes it is smoother. You can tell that there aren't as many spikes between
% the values > 200.
```



Functions

```
% Unit Impulse
function[y, n] = u_impulse(n_p, n1, n2)
    n = n1 : n2;
    y = (n - n_p) == 0;
end

% Unit Step
function[y, n] = u_step(n_s, n1, n2)
    n = n1 : n2;
    y = (n - n_s) >= 0;
end

% Time Shift
function[x, n] = time_shift(x, n, n_d)
    n = n + n_d;
end

% Time reversal.
function[x, n] = time_reverse(x, n)
    n = -n;
end
```

Published with MATLAB® R2020b