Name:	Muhammad Attiq
Registration Number:	FA23 – BCE – 060
Lab No:	3
Instructor:	Dr. Bilal Qasim
Class:	BCE – 4A

LAB 3

Study of Signal characteristics using MATLAB

Task 1

Create a function “impseq”, which performs following operations:

Function [x,n]=impseq(n0,n1,n2)

- Takes three parameters (n0, n1, n2) as input, where ‘n1’ and ‘n2’ are lower and upper limits of n axis, and ‘n0’ is the delay.
- Generates a unit-impulse sequence using above mentioned three parameters.
- There should be two output arguments [x, n] of function ‘impseq’, where ‘x’ is impulse sequence and ‘n’ is its corresponding n-axis.
- Finally, plot unit impulse ‘x’ against vector ‘n’.
- On the main window, type “[x,n]=impseq(0,-5,5)”

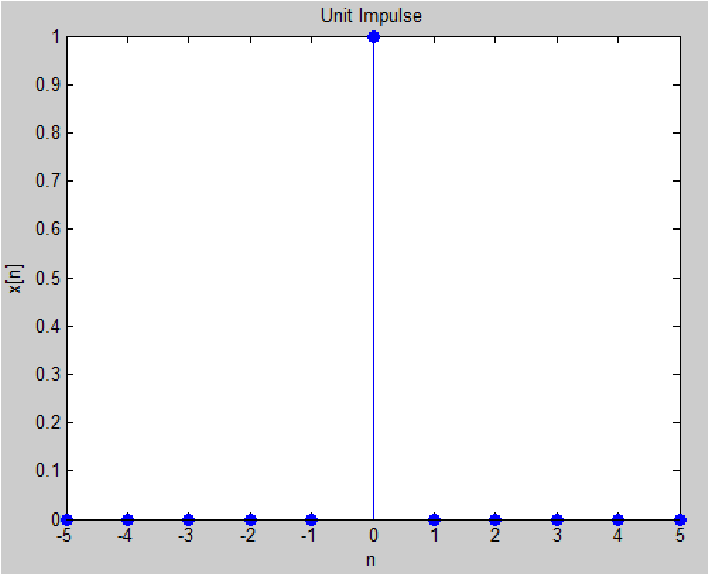
Script:

```
function [x, n] = impseq(n0, n1, n2)

    n = n1:n2;

    x = (n == n0);

    stem(n, x, 'filled');

    title('Unit Impulse');

    xlabel('n');

    ylabel('x[n]');

end
```

Command Window:

```
>> MATLAB_3(0, -5, 5)

ans =

     0     0     0     0     0     1     0     0     0     0     0
```

Figure:



Task 2

Make a function to form “stepseq” function which will output unit-step sequence. Function [x,n]=stepseq(n0,n1,n2)

- Unit Step Sequence
- We can have another elegant way to produce a step function
- Alternatively, we can use the “ones” function
- Type “stepseq[x,n]=(0,-5,5)” we get:

Script:

- Using Logic Comparison:

```
function [x,n]= stepseq(n0, n1, n2);

n = n1:n2;

x = (n >= n0);

stem(n, x, 'filled');

xlabel('n');

ylabel('x[n]');

grid on;
```

- Using Ones Function:

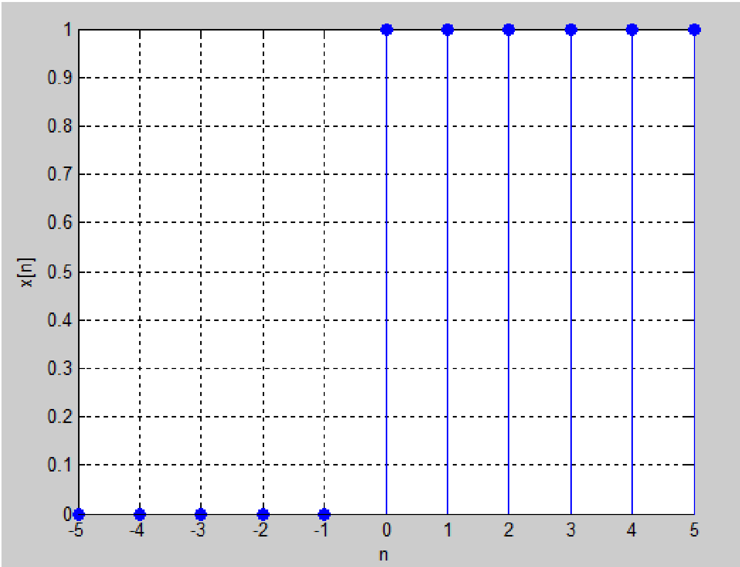
```
function [x,n]= stepseq(n0, n1, n2);

n = n1:n2;
```

```
x = zeros(1, length(n));

x(n >= n0) = ones(1, sum(n>=n0));

stem(n, x, 'filled');

xlabel('n');

ylabel('x[n]');

grid on;
```

Command Window:

```
>> stepseq(0, -5, 5)

ans =

     0     0     0     0     0     1     1     1     1     1     1
```

Figure:



Task 3

Create a function “rampseq”, which performs following operations: Function [x,n]=rampseq(n0,n1,n2)

- Takes three parameters (n0, n1, n2) as input, where ‘n1’ and ‘n2’ are lower and upper limits of n axis, and ‘n0’ is the delay.
- Generates a ramp sequence using above mentioned three parameters.
- There should be two output arguments [x, n] of function ‘rampseq’, where ‘x’ is impulse sequence and ‘n’ is its corresponding n-axis.
- Finally, plot ramp impulse ‘x’ against vector ‘n’.

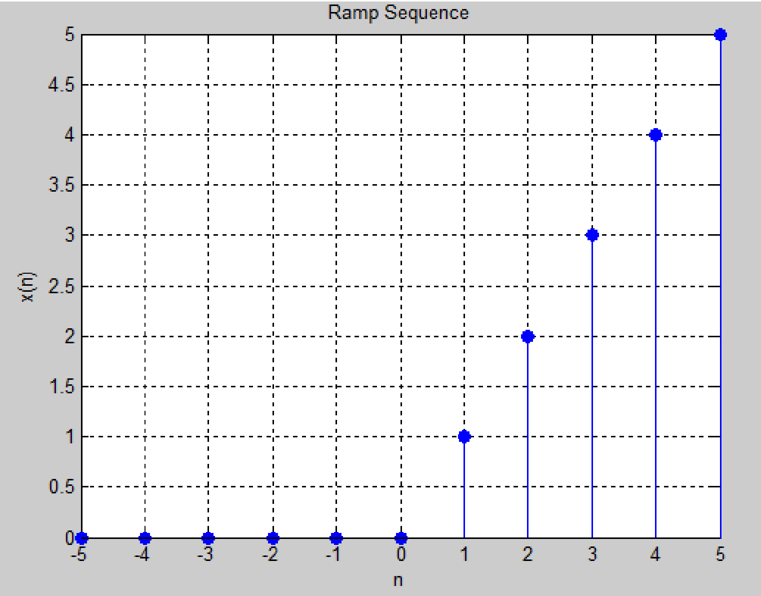
Script:

```
function [x, n] = rampseq(n0, n1, n2);

n = n1:n2;

x = (n - n0) .* (n >= n0);

stem(n, x, 'filled');

xlabel('n');

ylabel('x(n)');

title('Ramp Sequence');

grid on;
```

Command Window:

```
>> rampseq(0, -5, 5)

ans =

    0    0    0    0    0    0    1    2    3    4    5
```

Figure:



Task 4

Create a function "sigseq", which performs following operations:

Function [x,n]=sigpseq(n0,n1,n2)

- Takes three parameters (n0, n1, n2) as input, where 'n1' and 'n2' are lower and upper limits of n axis, and 'n0' is the delay.
- Generates a signum sequence using above mentioned three parameters.
- There should be two output arguments [x, n] of function 'sigseq', where 'x' is impulse sequence and 'n' is its corresponding n-axis.
- Finally, plot signum sequence 'x' against vector 'n'.

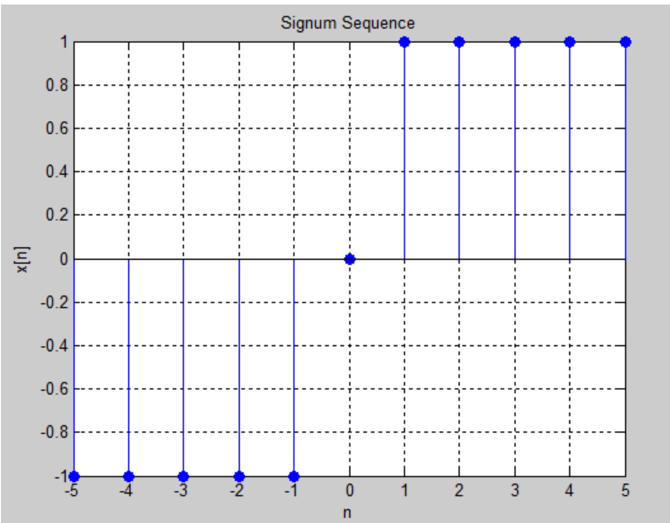
Script:

```
function [x, n] = sigseq(n0, n1, n2)

n = n1:n2;

x = -1*(n < n0) + 1*(n > n0);

stem(n, x, 'filled');

grid on;
```

Command Window:

```
>> sigseq(0, -5, 5)

ans =

  -1   -1   -1   -1   -1    0    1    1    1    1    1
```

Figure:



Task 5

Find $E\infty$ for the following signal

$$tri(t) = \begin{cases} 1 - |t| & |t| < 1 \\ 0 & |t| \geq 1 \end{cases}$$

Script:

```
t = -1:0.001:1;

tri_t = 1 - abs(t);

tri_squared = tri_t.^2;

E_infinity = trapz(t, tri_squared);

fprintf('The total energy E(infinty) is: %.4f\n', E_infinity);
```

Command Window:

```
>> task_5

The total energy E? is: 0.6667
```

Task 6

Find P∞ for the following signal

$$x[n] = \cos\left(\frac{\pi}{4} n\right)$$

Script:

```
n = 0:99999;

x = cos(pi/4 * n);

sum_squared = sum(abs(x).^2);

N = length(n);

P_infinity = sum_squared / N;

disp(['The average power P_infinity is: ', num2str(P_infinity)]);
```

Command Window:

>> task_6

The average power P_infinity is: 0.5

Task 7

Write a function which plot or stem a unit impulse and unit step signals. The function takes values for starting and ending value of independent variable, i.e. t and n, and a character for identification of discrete and continuous signal. Finally t plot or stem the function or signal. e.g; function f_name (arg1 (start) , arg2 (end) , arg3 (D/C)).

Script:

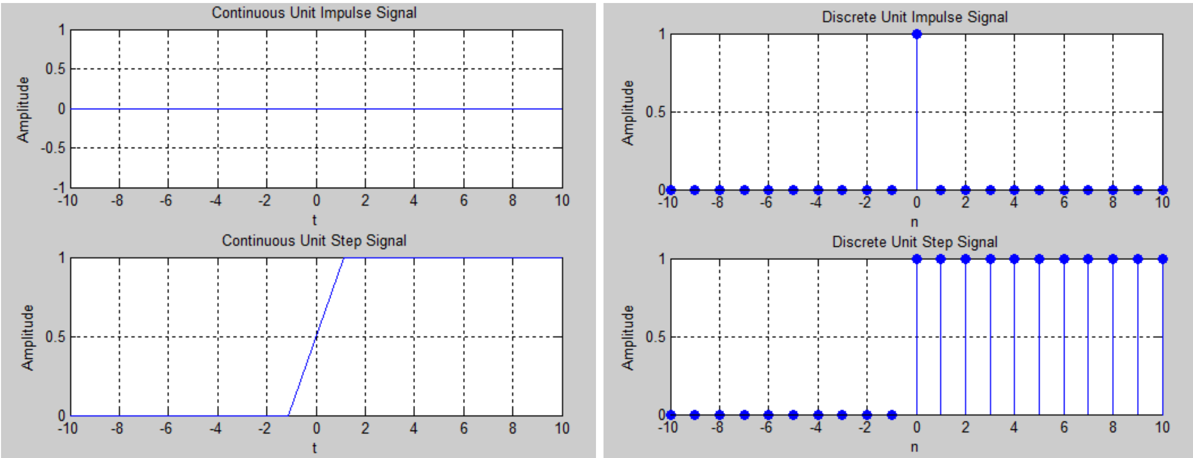
```
function plot_signals(start_val, end_val, signal_type)

  t = start_val:end_val;

  if signal_type == 'D'

    impulse = zeros(size(t));

    impulse(t == 0) = 1;

    step = double(t >= 0);

    subplot(2, 1, 1);

    stem(t, impulse, 'filled');

    title('Discrete Unit Impulse Signal');

    xlabel('n'); ylabel('Amplitude');

    grid on;

    subplot(2, 1, 2);

    stem(t, step, 'filled');

    title('Discrete Unit Step Signal');

    xlabel('n'); ylabel('Amplitude');

    grid on;

   elseif signal_type == 'C'

    t = linspace(start_val, end_val, 10);

    impulse = zeros(size(t));

    impulse(t == 0) = 1;
```

```
        step = double(t >= 0);

        subplot(2, 1, 1);

        plot(t, impulse);

        title('Continuous Unit Impulse Signal');

        xlabel('t');  ylabel('Amplitude');

        grid on;

        subplot(2, 1, 2);

        plot(t, step);

        title('Continuous Unit Step Signal');

        xlabel('t'); ylabel('Amplitude');

        grid on;

    else

        disp('Invalid signal type. Use "D" for discrete or "C" for continuous.');  end

end
```

Command Window:

```
>> task_7(-10, 10, 'C')

>> task_7(-10, 10, 'D')
```

Figure:



Post-lab Task

Critical Analysis / Conclusion

In this lab, I explored how to plot various sequences, including unit sample, step, ramp, and signum, using MATLAB. I also worked on calculating their energy and power. The process of plotting these graphs became clear once I input the proper sequences. By using the "stem" function, I was able to generate discrete-time graphs easily, which made plotting both discrete and continuous time sequences much simpler.

<table>
<tr><th colspan="3">Lab Assessment</th></tr>
<tr><td>Lab Task Evaluation</td><td>/6</td><td rowspan="2">/10</td></tr>
<tr><td>Lab Report</td><td>/4</td></tr>
<tr><td colspan="3">Instructor Signature and Comments</td></tr>
</table>