Shri Ramdeobaba College of Engineering and Management, Nagpur

Department of Electronics and Communication Engineering

Teacher's Assessment

Course: Signals and Systems **Course Code: ECT3004**

Teacher's Assessment Task: Students should **plot at least FIVE signals** using **MATLAB onramp** and submit a printed report in the format provided **on or before 21st October, 2024**. The detailed steps to be followed are mentioned below.

1. Go through the Tutorial Videos of "**MATLAB onramp**" for acquaintance with MATLAB.
2. Go to the webpage of MathWorks using the link https://in.mathworks.com/
3. Create MathWorks account using your email address.
4. Once the account is created, verify the account through the email received on your mail address
5. Login to https://in.mathworks.com/ using your login credentials
6. Go to "***MATLAB***" tab MATLAB
7. Go to "***Open MATLAB Online (basic)***" tab
8. Now you can write the program in MATLAB and Plot the signals.
9. Take the screenshot of Program and the Corresponding Signal Plotted.
10. Prepare a Report in the Format Provided
    i. 1st Page should be the Title Page
    ii. Next 2-3 pages about MATLAB onramp
    iii. Thereafter 1 signal on each page (Matlab Program + Signal Plot)

[illegible] [illegible]

Dr. P. R. Selokar **Dr. R. S. Ochawar**

Course Co-ordinators

Teachers Assessment

SIGNALS AND SYSTEMS (ECT 3004)

3rd Semester B. Tech. (ECE – B Section)
Session: 2024-25

<u>Report on Teacher's Assessment Activity</u>

SHRI RAMDEOBABA SARVAJANIK SAMITI
SHRI RAMDEOBABA COLLEGE OF ENGINEERING AND MANAGEMENT
NAGPUR
ESTD. - 1984
श्रेष्ठम् तंत्र शिक्षणम्

Name of the Student: Vanshika N. Ramtekkar
Roll No: 60 Section: B

Course Coordinator
Dr. P. R. Selokar [illegible]

Department of Electronics and Communication Engineering

Contents

1. Introduction:

MATLAB Onramp is a free, interactive online course offered by MathWorks, designed to introduce beginners to the fundamentals of MATLAB, a powerful programming environment widely used for numerical computing, data analysis, and visualization. The course provides a structured, self-paced learning experience, allowing users to explore core MATLAB concepts through hands-on exercises and real-time feedback. Participants start with an overview of the MATLAB interface and basic syntax, learning how to create and manipulate variables, arrays, and matrices, which are essential for mathematical computations. The curriculum includes data visualization techniques, enabling learners to generate plots and graphical representations of data, enhancing their ability to communicate findings effectively. Additionally, the course covers control structures, such as loops and conditional statements, which are vital for programming logic and flow. Users can practice coding directly in their web browser, making the experience accessible and convenient. By the end of the course, participants gain a solid foundation in MATLAB, preparing them for more advanced topics and applications in various fields, including engineering, finance, and scientific research. Completing MATLAB Onramp often results in a certificate, adding value to learners' professional development. Overall, MATLAB Onramp is an excellent resource for anyone looking to build their skills in MATLAB and leverage its capabilities for data analysis and modeling.

The MATLAB courses contains:

- Basic MATLAB syntax and commands
- Working with variables and array
- Data visualization techniques
- Control flow and programming constructs

Here are some key features and functionalities it offers MATLAB offers:

Interactive Learning: The course includes interactive tutorials and exercises, allowing users to practice coding directly within the browser.

Core Concepts: It covers fundamental MATLAB concepts, such as:

- Variables and data types
- Arrays and matrices
- Basic mathematical operations
- Functions and scripts

Data Visualization: Users learn how to create plots and visualizations, enabling them to present data effectively.

Control Structures: The course introduces programming constructs like loops and conditional statements, which are essential for writing efficient code.

Real-World Applications: Examples and exercises often relate to practical applications in engineering, science, and mathematics, helping learners see how MATLAB is used in various fields.

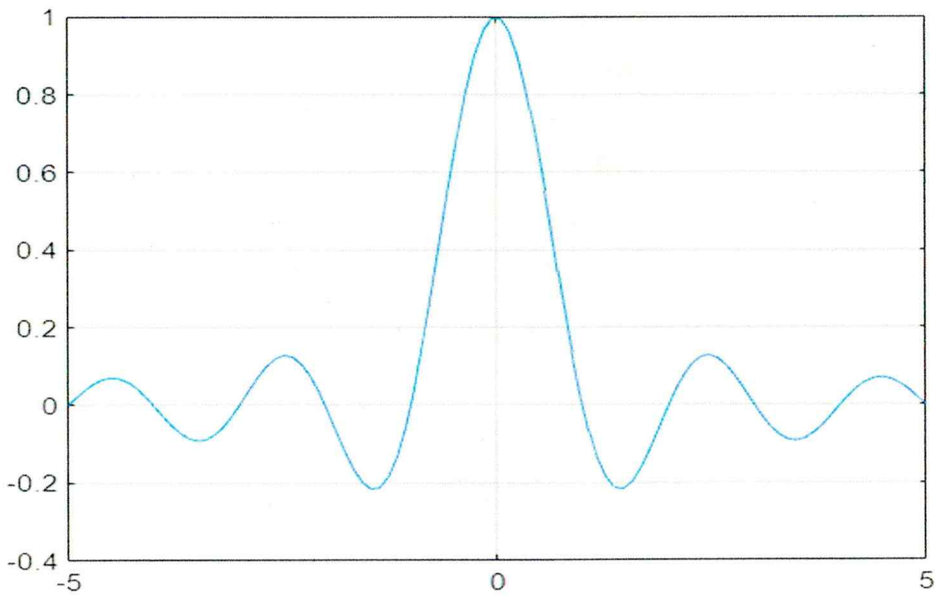
Self-Paced: Users can progress at their own speed, revisiting topics as needed.

Completion Certificate: Upon finishing the course, participants may receive a certificate of completion, which can be useful for professional development.

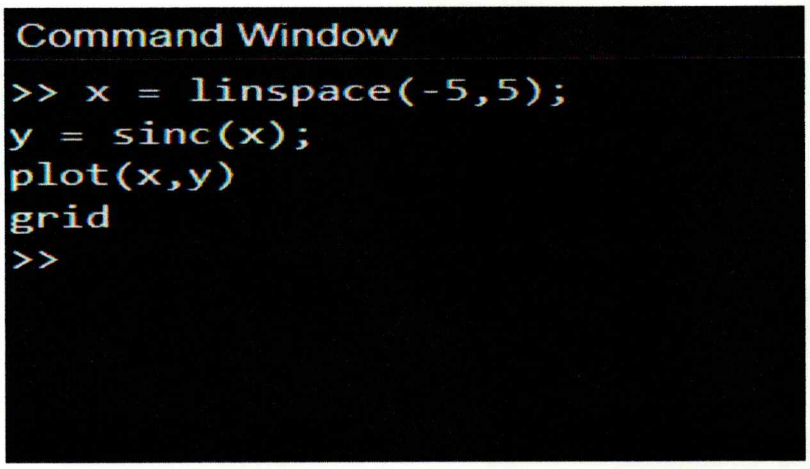
Overall, MATLAB Onramp serves as a comprehensive introduction to the software, equipping users with the skills to tackle more advanced topics and projects in MATLAB.

2. Signal-1 (Sinc Function)

Plotted Signal:



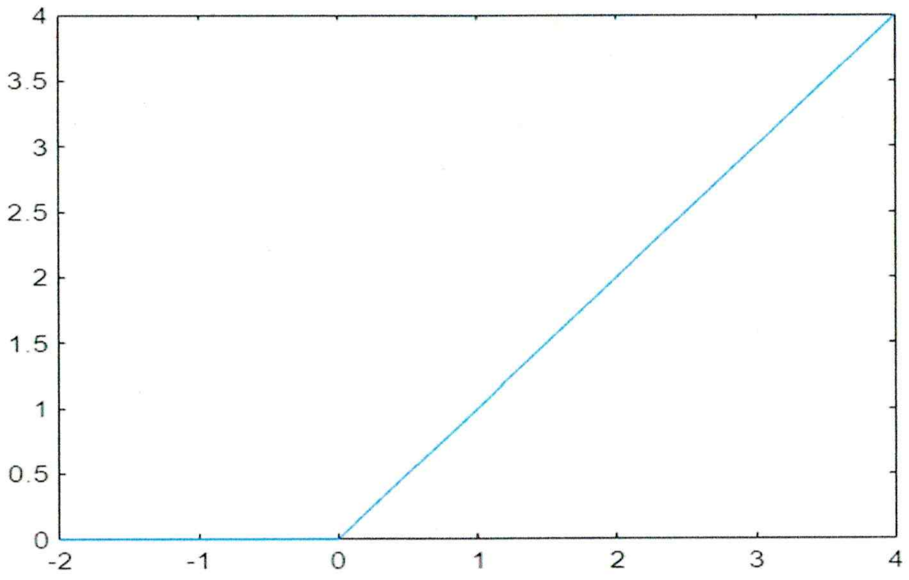
MATLAB Code:



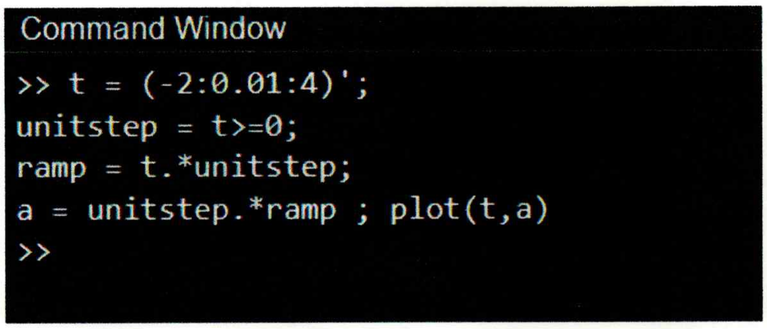
```
Command Window
>> x = linspace(-5,5);
y = sinc(x);
plot(x,y)
grid
>>
```

3. Signal-2 (Multiplication of unit step and ramp function)

Plotted Signal:



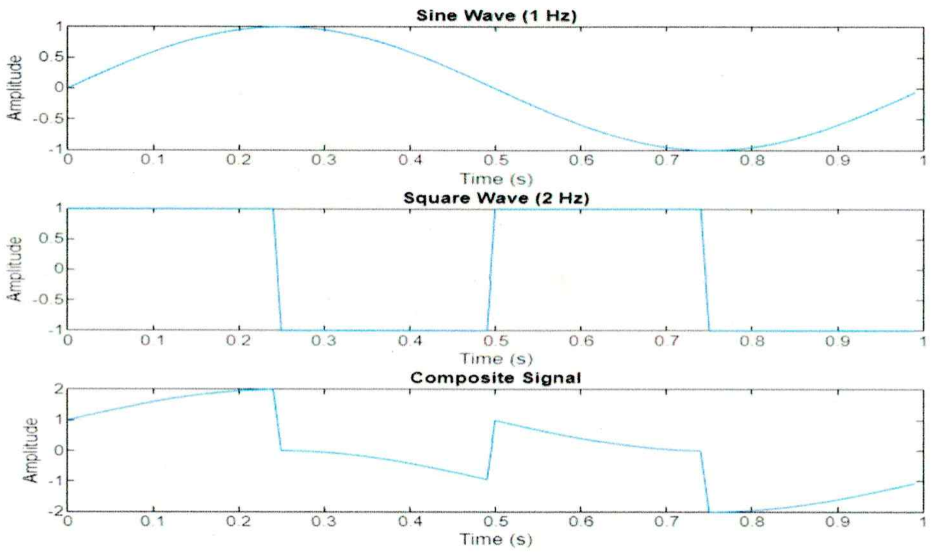
MATLAB Code:



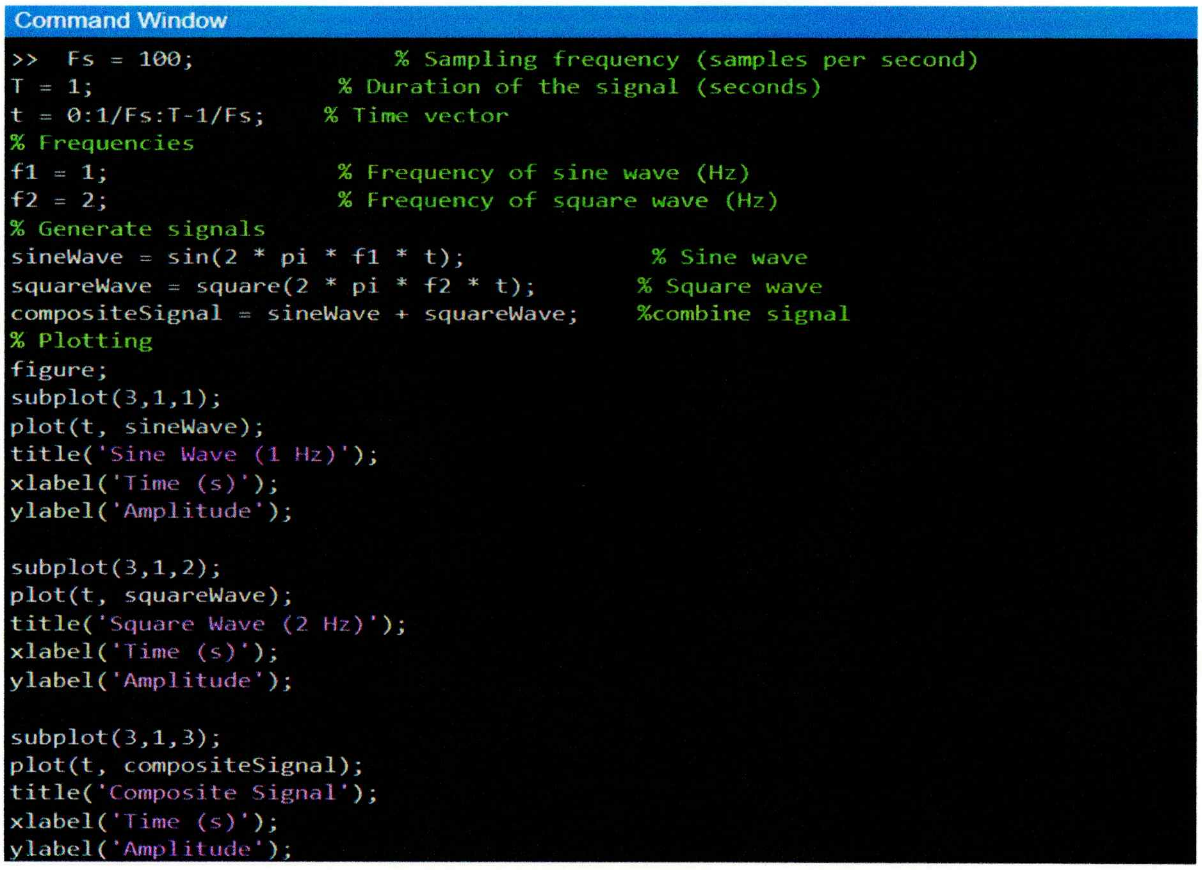
```
Command Window
>> t = (-2:0.01:4)';
unitstep = t>=0;
ramp = t.*unitstep;
a = unitstep.*ramp ; plot(t,a)
>>
```

4. Signal-3 (Addition of sine and square wave)

Plotted Signal:



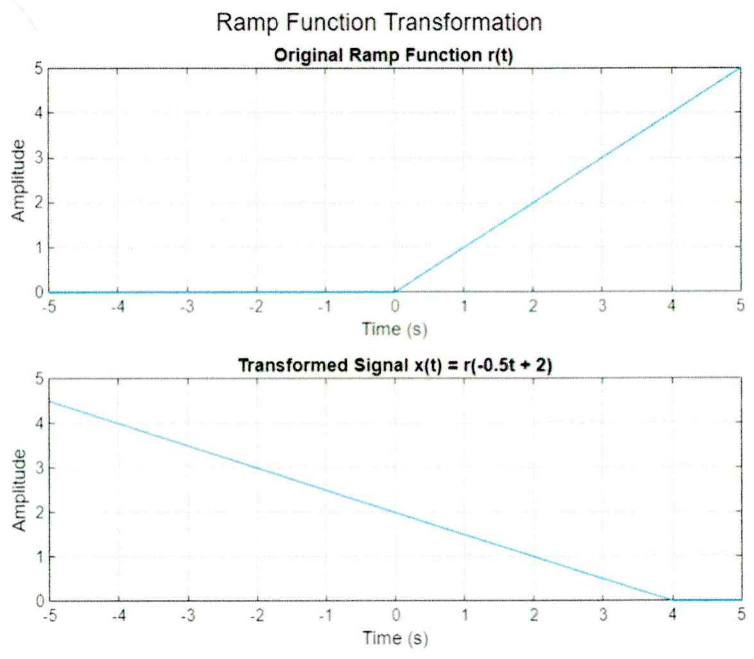
MATLABCode:



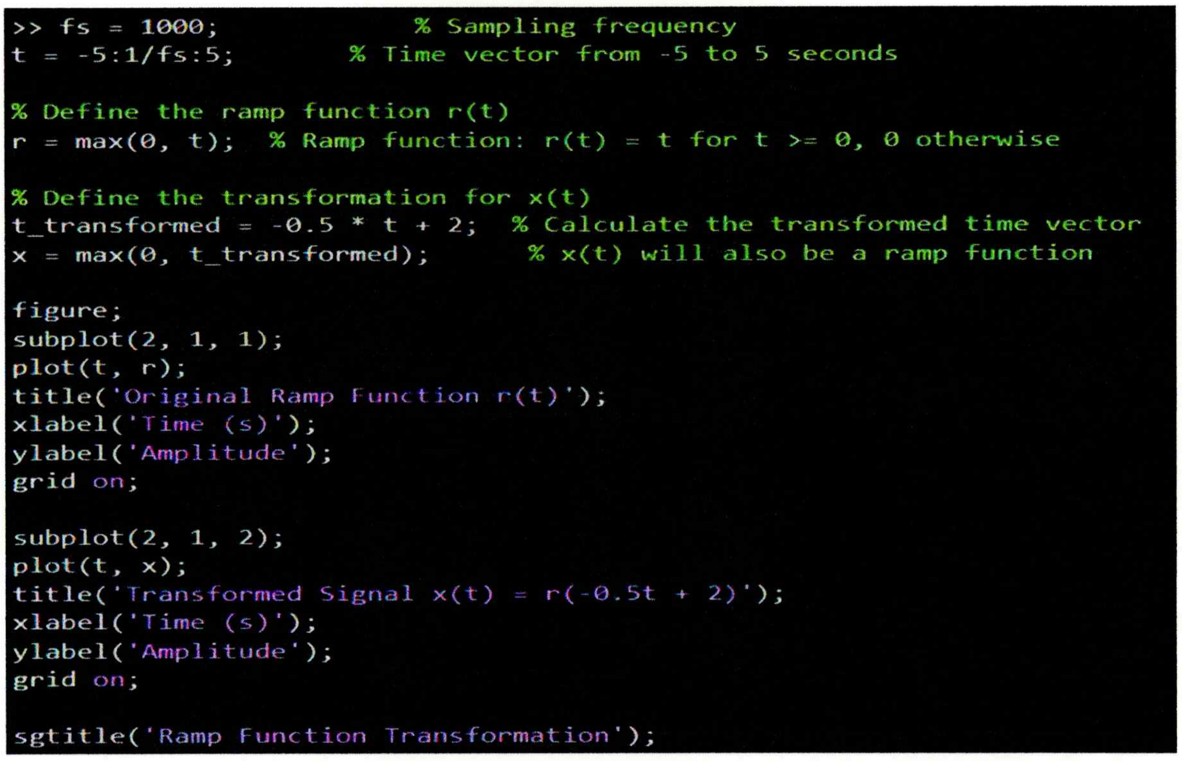
```
>>  Fs = 100;                    % Sampling frequency (samples per second)
T = 1;                       % Duration of the signal (seconds)
t = 0:1/Fs:T-1/Fs;          % Time vector
% Frequencies
f1 = 1;                      % Frequency of sine wave (Hz)
f2 = 2;                      % Frequency of square wave (Hz)
% Generate signals
sineWave = sin(2 * pi * f1 * t);            % Sine wave
squareWave = square(2 * pi * f2 * t);      % Square wave
compositeSignal = sineWave + squareWave;    %combine signal
% Plotting
figure;
subplot(3,1,1);
plot(t, sineWave);
title('Sine Wave (1 Hz)');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(3,1,2);
plot(t, squareWave);
title('Square Wave (2 Hz)');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(3,1,3);
plot(t, compositeSignal);
title('Composite Signal');
xlabel('Time (s)');
ylabel('Amplitude');
```

5. Signal-4 (Transformed Ramp Signal)

Plotted Signal:



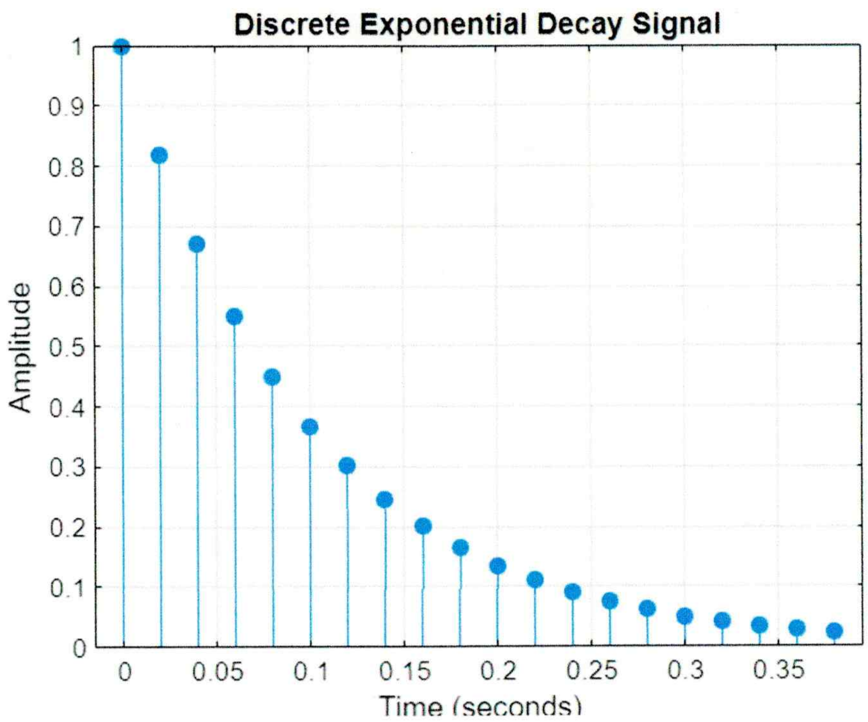
MATLAB Code:



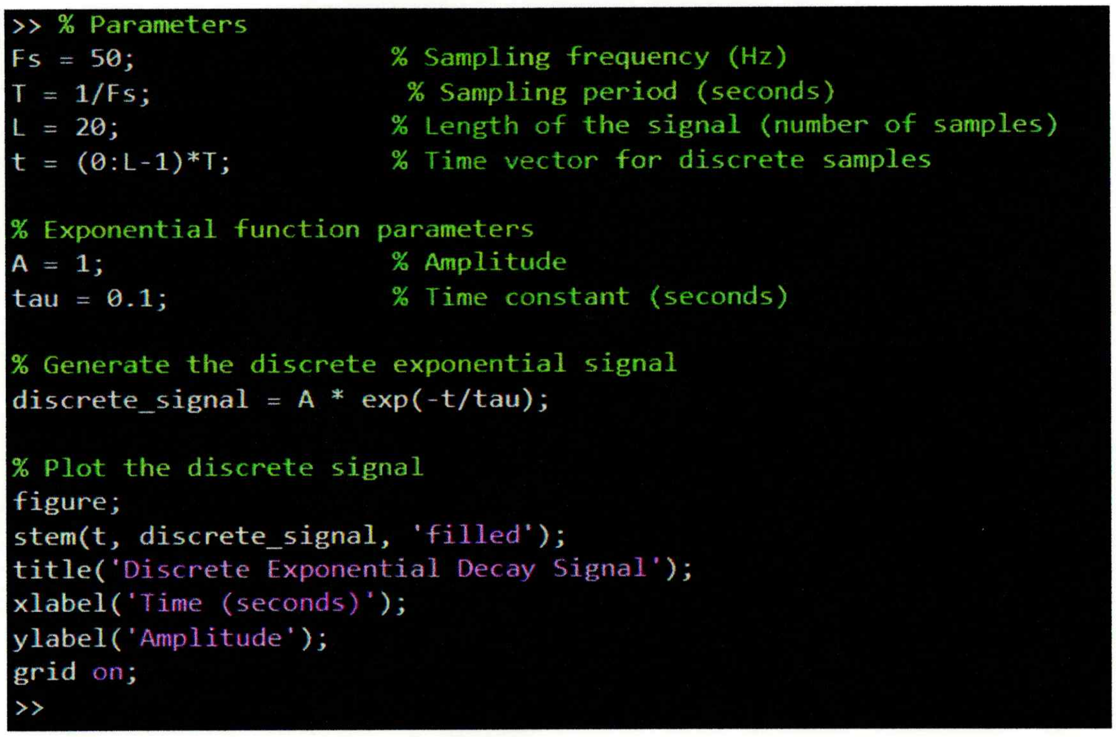
```
>> fs = 1000;                    % Sampling frequency
t = -5:1/fs:5;           % Time vector from -5 to 5 seconds

% Define the ramp function r(t)
r = max(0, t);   % Ramp function: r(t) = t for t >= 0, 0 otherwise

% Define the transformation for x(t)
t_transformed = -0.5 * t + 2;  % Calculate the transformed time vector
x = max(0, t_transformed);       % x(t) will also be a ramp function

figure;
subplot(2, 1, 1);
plot(t, r);
title('Original Ramp Function r(t)');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;

subplot(2, 1, 2);
plot(t, x);
title('Transformed Signal x(t) = r(-0.5t + 2)');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;

sgtitle('Ramp Function Transformation');
```

6. Signal-5 (Discrete exponential signal)

Plotted Signal:



MATLAB Code:



```
>> % Parameters
Fs = 50;                    % Sampling frequency (Hz)
T = 1/Fs;                    % Sampling period (seconds)
L = 20;                     % Length of the signal (number of samples)
t = (0:L-1)*T;              % Time vector for discrete samples

% Exponential function parameters
A = 1;                      % Amplitude
tau = 0.1;                  % Time constant (seconds)

% Generate the discrete exponential signal
discrete_signal = A * exp(-t/tau);

% Plot the discrete signal
figure;
stem(t, discrete_signal, 'filled');
title('Discrete Exponential Decay Signal');
xlabel('Time (seconds)');
ylabel('Amplitude');
grid on;
>>
```

[illegible]
Dr. P.R. Selokar