

Report on Matlab projects (SEC 101)

Priyanshu Gupta

Roll No.: 39

M.Sc 1st Year

20 November 2025

Contents

1	Project 1: Random Walk Simulator	2
2	Project 2: Guess the Number Game	3
3	Project 3: Voice Changer	4
4	Project 4: Two Images Merger	5
5	Conclusion	6

1 Project 1: Random Walk Simulator

Problem Statement

This project aims to simulate a 2D random walk. The program generates 200 random movements in the X and Y directions, calculates the cumulative path, and visualizes the complete motion. This helps illustrate the concept of randomness and basic plotting in MATLAB.

MATLAB Code

```
% RANDOM WALK SIMULATOR
steps = 200;
moves = randi([-1 1], steps, 2); % random steps
path = cumsum(moves); % cumulative motion

figure;
plot(path(:,1), path(:,2), 'LineWidth', 2);
title('2D Random Walk Simulator');
xlabel('X Position');
ylabel('Y Position');
axis equal;
```

Output

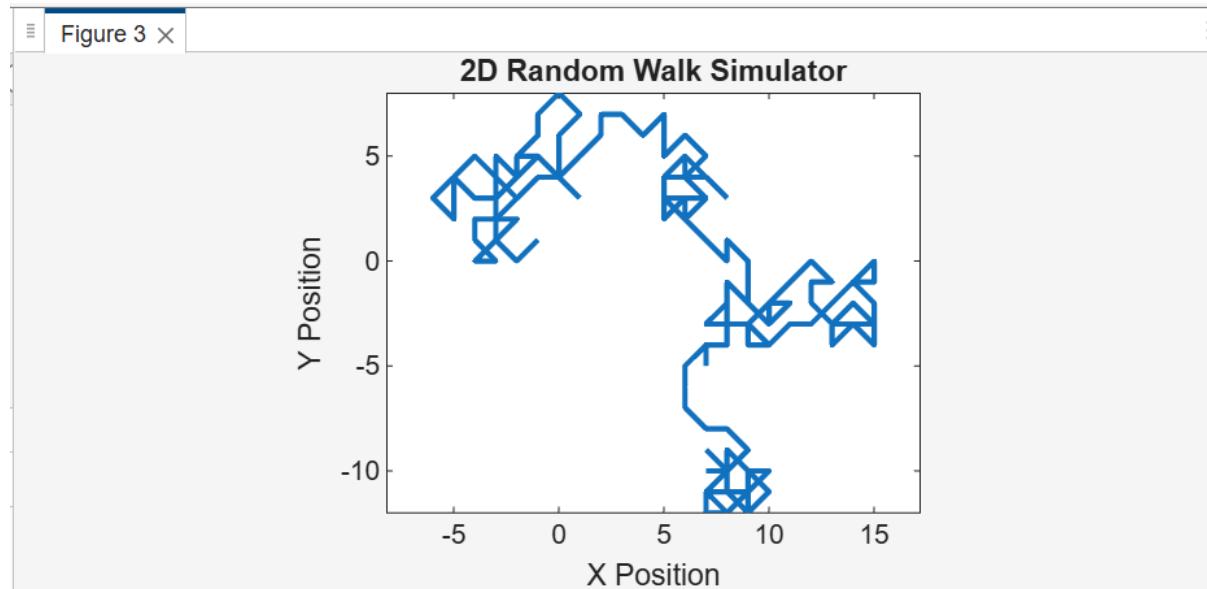


Figure 1: Random Walk Output

2 Project 2: Guess the Number Game

Problem Statement

The objective of this project is to create an interactive guessing game. MATLAB randomly selects a number between 1 and 50. The user repeatedly enters guesses and receives hints whether the guess is too high, too low, or correct. This project demonstrates loops, conditionals, user input, and interaction.

MATLAB Code

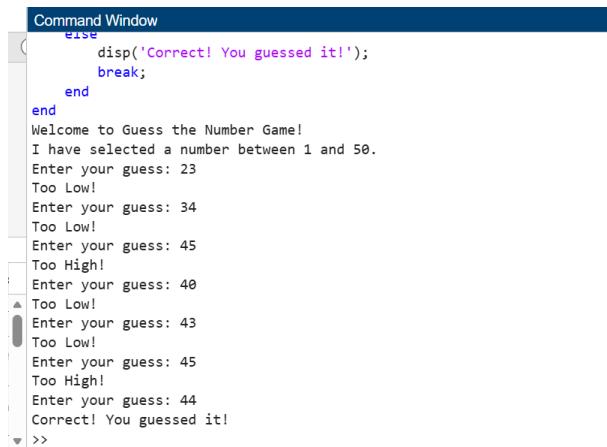
```
% GUESS THE NUMBER GAME
target = randi([1 50]);      % picks a number
guess = -1;

disp('Welcome to Guess the Number Game!');
disp('I have selected a number between 1 and 50.');

while guess ~= target
    guess = input('Enter your guess: ');

    if guess > target
        disp('Too High!');
    elseif guess < target
        disp('Too Low!');
    else
        disp('Correct! You guessed it!');
    end
end
```

Output



The screenshot shows the MATLAB Command Window with the title bar "Command Window". The window displays the execution of a script. The script starts with a series of comments and variable definitions. It then displays a welcome message and a hint about the selected number. A while loop begins, prompting the user for a guess. Inside the loop, an if-else block handles the logic for whether the guess is too high, too low, or correct. The user's input is shown as "Enter your guess: 23", followed by several iterations where the user guesses 34, 45, 40, 43, 45, 44, and finally 45, which results in the message "Correct! You guessed it!".

Figure 2: User guesses on console

3 Project 3: Voice Changer

Problem Statement

This project demonstrates basic signal processing concepts using MATLAB's built-in audio file `handel.mat`. The program plays the audio once, then allows the user to select from different playback modes including **fast**, **slow**, and **echo effects**. Based on user choice, the sampling rate is modified or an echo filter is applied. This helps understand signal processing.

MATLAB Code

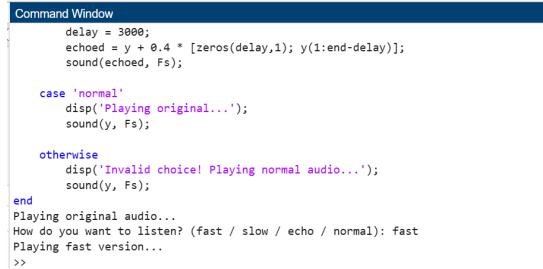
```
% INTERACTIVE VOICE CHANGER
load handel.mat    % loads 'y' and 'Fs'

disp('Playing original audio...'); 
sound(y, Fs);
pause(4);

choice = lower(input('How do you want to listen? (fast / slow / echo / normal): ', 's'));

switch choice
    case 'fast'
        sound(y, Fs * 1.5);
    case 'slow'
        sound(y, Fs * 0.7);
    case 'echo'
        delay = 3000;
        echoed = y + 0.4 * [zeros(delay,1); y(1:end-delay)];
        sound(echoed, Fs);
    otherwise
        sound(y, Fs);
end
```

Output



```
Command Window
delay = 3000;
echoed = y + 0.4 * [zeros(delay,1); y(1:end-delay)];
sound(echoed, Fs);

case 'normal'
    disp('Playing original...'); 
    sound(y, Fs);

otherwise
    disp('Invalid choice! Playing normal audio...'); 
    sound(y, Fs);
end
Playing original audio...
How do you want to listen? (fast / slow / echo / normal): fast
Playing fast version...
>>
```

Figure 3: Modified sound playing

4 Project 4: Two Images Merger

Problem Statement

This project focuses on image blending using MATLAB's image processing tools. Two built-in sample images (`peppers.png` and `saturn.png`) are loaded, resized to identical dimensions, converted to double precision, and blended using pixel-wise averaging. The final output combines features of both images, demonstrating how images are stored as matrices and manipulated numerically.

MATLAB Code

```
% TWO IMAGES MERGER

img1 = imread('peppers.png');           % Image 1
img2 = imread('saturn.png');            % Image 2

img2 = imresize(img2, [size(img1,1), size(img1,2)]);

A = im2double(img1);
B = im2double(img2);
merged = (A + B) / 2;

figure;

subplot(1,3,1); imshow(img1); title('Image 1');
subplot(1,3,2); imshow(img2); title('Image 2');
subplot(1,3,3); imshow(merged); title('Merged Output');
```

Output

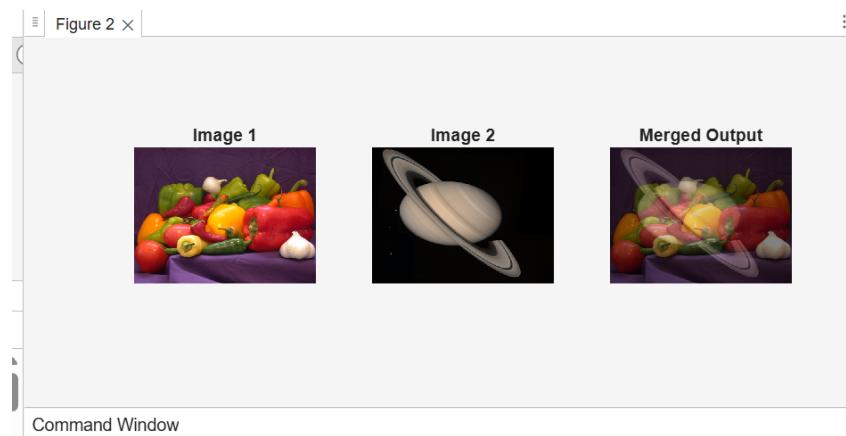


Figure 4: Merged Image Output

5 Conclusion

This report demonstrates four MATLAB applications covering plotting, logic building, digital audio manipulation, and image blending. Each project uses fundamental programming concepts and MATLAB topics specified in the course syllabus.