LINEAR SYSTEMS
SECOND YEAR
LABORATORY SESSION 1 - ACADEMIC YEAR 2023/2024

1. Introduction

1.1. Creating an environment

Create a new folder on the desktop called "P1 Linear Systems". Run Matlab. Place the *path* in a folder created on the desktop so that it is located as shown in the following image:

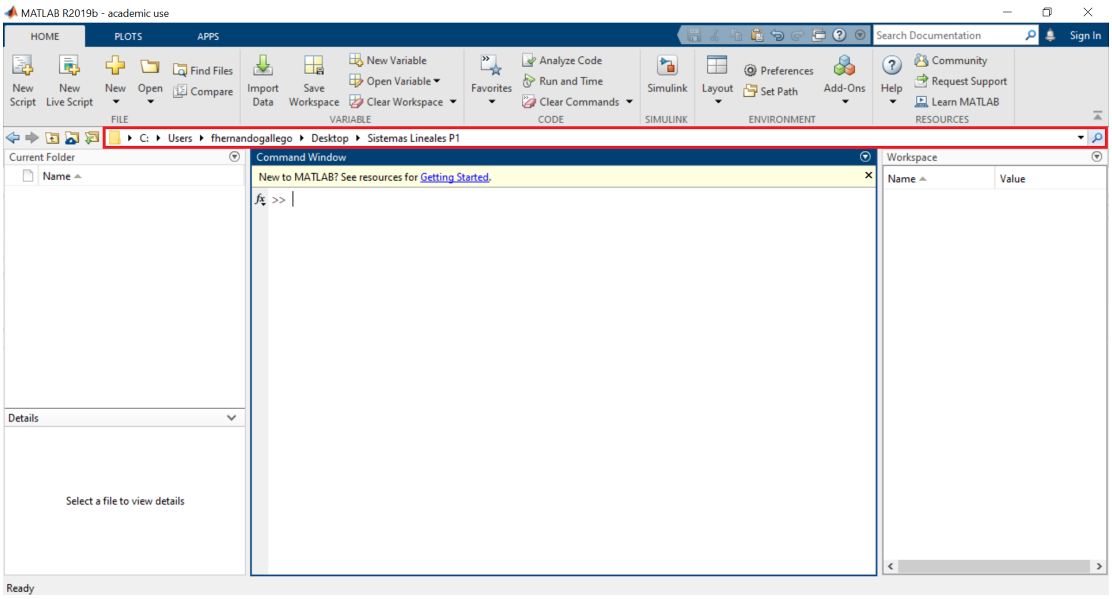

Figure 1: Path Placement

All the necessary files will be placed in the folder for the realization of the laboratory session.

1.2. Main file

Create a new file named "lab1.m", which will be saved in the *path* that was previously created. All the commands and functions attached to the practice will be executed from that file. It is recommended to include the following commands at the beginning of the file:

```
clc % Cleans the command window
clear all % Cleans the workspace
close all % Closes all figures
```

The main screen will look like this:

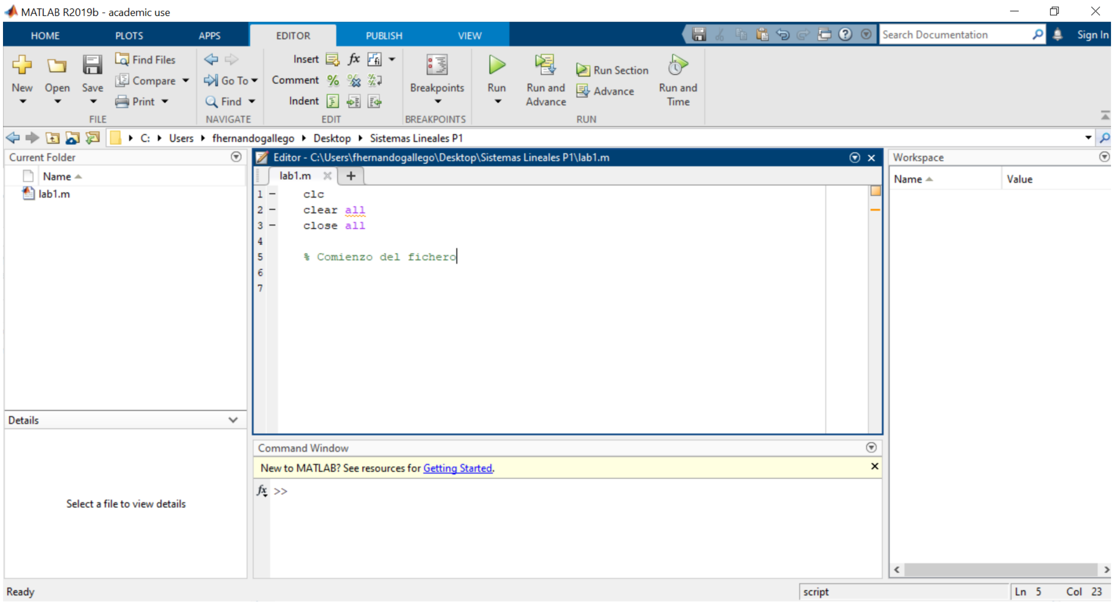

Figure 2: Main file

1.3. Functions

From the main file "lab1.m", a new function “square.m” will be called. To do this, create a new file with the following content:

```
function [tout,y] = square(tin,x)
   tout = tin; % The time axis does not vary
y = x.*x;   % Returns the square of the entered signal
```

In the function, *tin* and *tout* represent the input and output time vectors and *x* and *y* are the arrays of input and output values. The two files will be stored in the same directory, as follows:

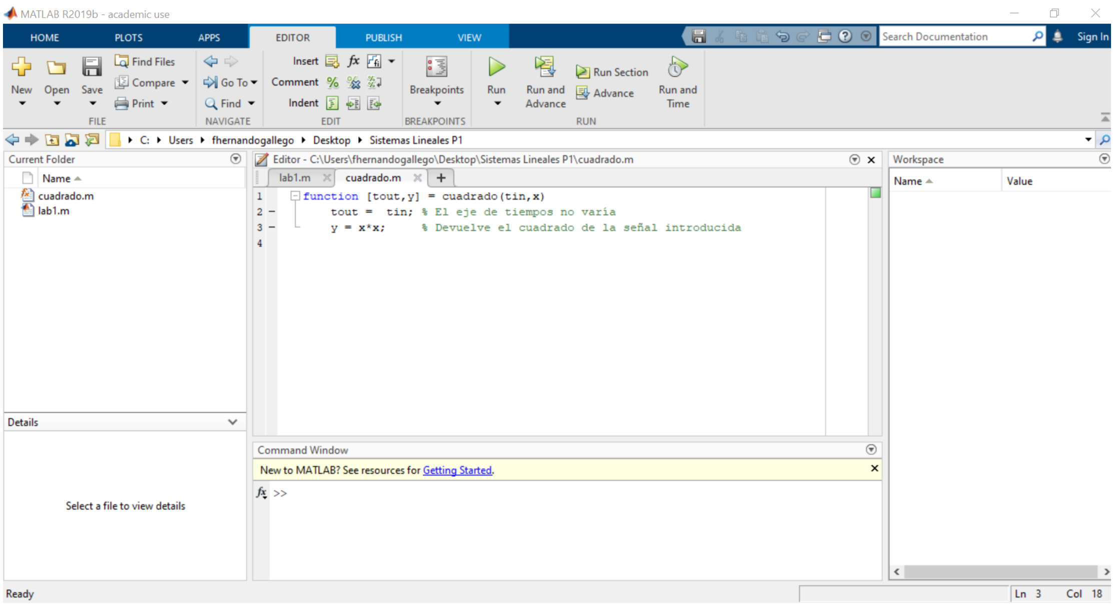

Figure 3: Additional function

1.4. Signal representation

Once the function is created, it can be invoked from the main file "lab1.m". Create a time vector *tin* and the vector *x* with the corresponding values. These vectors will be the inputs of the function ***square***. The output time vector and the corresponding vector with the output values will be denoted by *tout* and *y*, respectively:

```
clc
clear all
close all
%% Start of the file
tin = -5:1:5; % Time vector
x = -5:1:5; % Value vector
[tout,y] = square(tin,x); %  Function invocation
%% Creation of the figure
figure(1)     % Figure 1
subplot(2,1,1) % Figure 1.1 // 2 rows, 1 column, first figure
plot(tin,x)
title('Input'),xlabel('time'),ylabel('values'),ylim([min(x)  max(y)])
subplot(2,1,2) % Figure 1.2 // 2 rows, 1 column, second figure
plot(tout,y)
title('Output'),xlabel('time'),ylabel('values'),ylim([min(y)  max(y)])
```

We believe the above commands together with their comments are self-explanatory. In any case, if you have any questions, you can always use the Matlab command ***help*** or consult the online documentation: https://es.mathworks.com/help/

The composition of Matlab should look as follows:

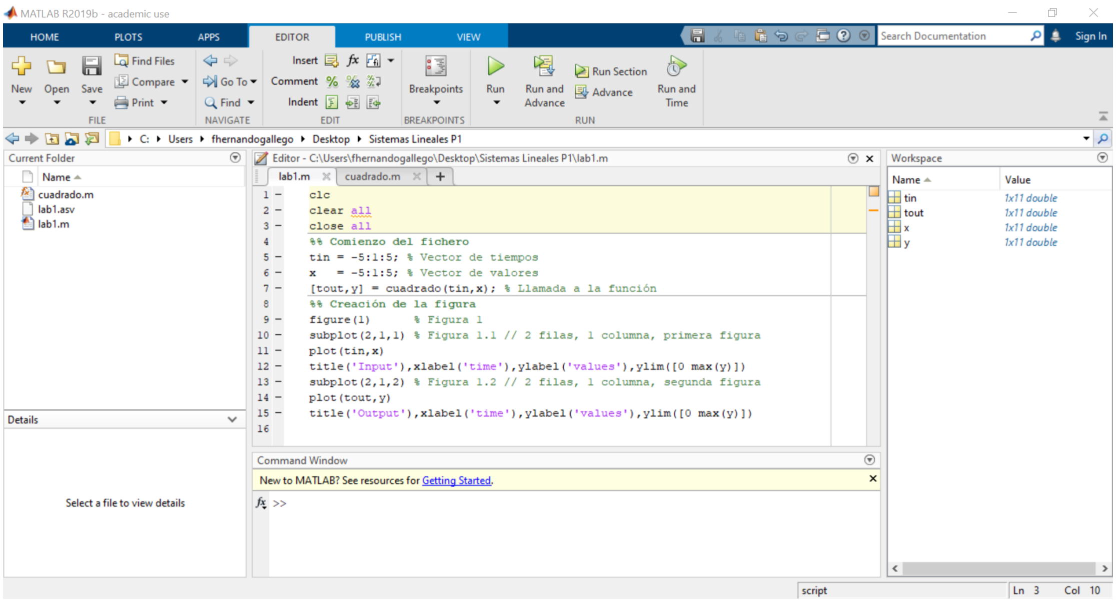

Figure 4: Signal representation file

The resulting figure will be as follows:

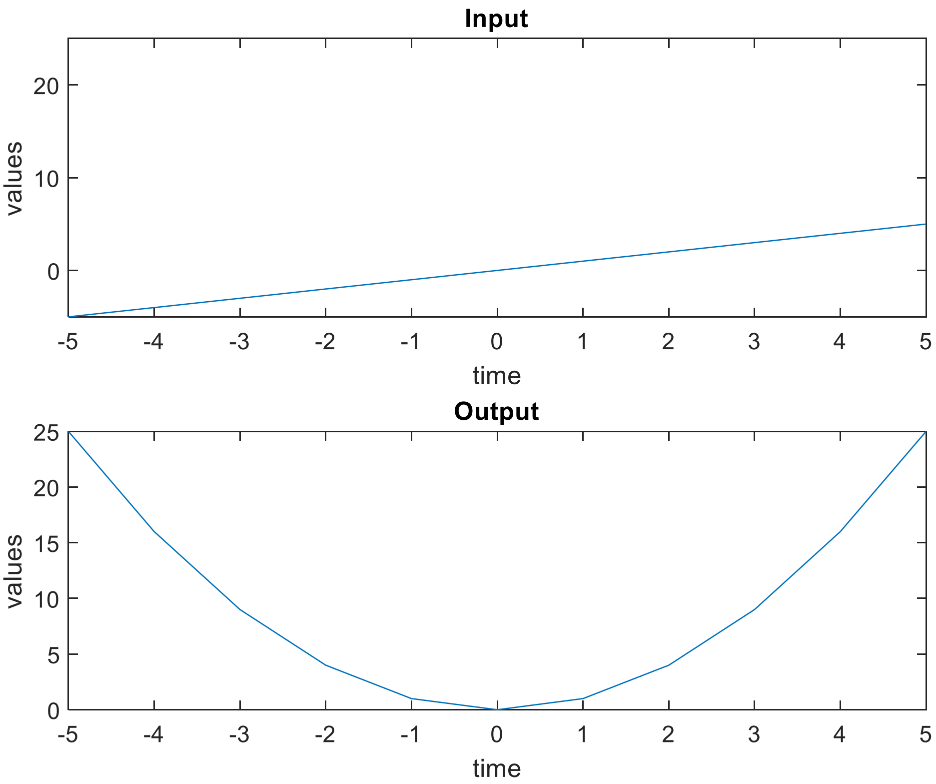

Figure 5: Signal representation

Exercise 1

Write a function called “genera_sinusoide.m”, which generates sinusoidal signals with the following input parameters: initial time instant (*n0*), final time instant (*n1*), sampling period (*step*), frequency (*f0*), and phase (*phi*). The function will return the generated sequence in a vector (*y*) and the corresponding time reference vector (*tout*).

Using this function and the ***subplot*** command, plot three sinusoids with a time range from 0 to 2 and frequencies π, 2π, and 4π. The time steps must be sufficiently small to make the sinusoidal shape recognizable. The three graphs will have to be seen in the same figure, separated into three rows.

Note: In MATLAB, π can be obtained by typing "pi" and the imaginary part of a signal is given by the letter "i".

Exercise 2

Write a function called “genera_exponencial.m”, which generates exponential signals with the following input parameters: initial time instant *(n0),* final time instant *(n1),* sampling period *(step),* and base (*b*). The function will return the generated sequence (*y*) and the corresponding time reference vector (*tout*).

Using this function and the ***subplot*** command, plot three exponentials with a time range from 0 to 4 and bases $^1/_2$, $^1/_4$, and $e^{j\pi}$ (for the last base, plot the real and imaginary part separately).

Exercise 3

In the file “lab1file.m:”

a) Define a discrete-time base vector *N* that ranges from -50 to +50.

b) Define a pulse signal that has unit height between -10 and +10 and that is zero elsewhere.

c) Define a delta signal δ[N-20] + δ[*N*+20] that is equal to one at *N*=20 and *N*=-20 and that is zero elsewhere.

d) Using the ***conv*** function in Matlab, compute the convolution of the above two signals and plot it in a figure.

Exercise 4

Consider the following sequences:

$$x_1[n] = 3\sin\left(\frac{\pi}{7}n\right) + j4\cos\left(\frac{\pi}{7}n\right), \quad n = 0, \ldots, 20$$
$$x_2[n] = (1.1)^n \cos\left(\frac{\pi}{11}n + \frac{\pi}{4}\right), \quad n = 0, \ldots, 50$$

Express $x_1[\cdot]$ and $x_2[\cdot]$ using complex exponentials with the help of Euler's formula and generate new vectors $x_{11}[\cdot]$ and $x_{22}[\cdot]$ using the ***exp*** function in Matlab. Next, using the ***subplot*** command, plot the functions $x_1[\cdot]$, $x_2[\cdot]$, $x_{11}[\cdot]$, and $x_{22}[\cdot]$ and verify that the new vectors match the representations obtained from the sinusoids. (For $x_1[\cdot]$ and $x_{11}[\cdot]$, plot the real and imaginary part separately.)