NUMBER WRITING ROBOTIC MANIPULATOR

A PROJECT REPORT

Submitted by

BL.EN.U4AIE19041	M. TANUJ
BL.EN.U4AIE19049	P. VENKATA AKHIL
BL.EN.U4AIE19068	V. AISHWARYA

for the course

19AIE201 – Introduction to Robotics

in partial fulfillment for the award of the degree

0f

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE ENGINEERING

AMRITA SCHOOL OF ENGINEERING, BANGALORE

BANGALORE 560 035

PROBLEM STATEMENT

Design and implement a planer robotic manipulator with suitable degree of freedom to write characters {1, 3, 4, 5, 9, 0}. Input for the system is character to be written.

PROBLEM STATEMENT EXPLANATION:

In this problem we can see the working of a number writing planar robotic manipulator in MATLAB with graphical interface. We can build the number writing manipulator using any degree of freedom. Inverse kinematics is used by the manipulator to draw the number. When the user runs the code, he will be shown the list of numbers the manipulator can draw. The list of numbers includes (1, 3, 4, 5, 9, 0). So, when the user chooses a particular number among the list, the robotic manipulator will draw that number in a graphical plot.

THEORY

1. We have implemented a 2 Linked 3R Planar Robotic Manipulator for writing numbers.

2. The degree of freedom used is 3 because of the three revolute joints, where one revolute joint is used for the end effector.

$$\text{DOF} = 3$$

3. The Inverse kinematics of our 2 Linked manipulator is shown as follows:

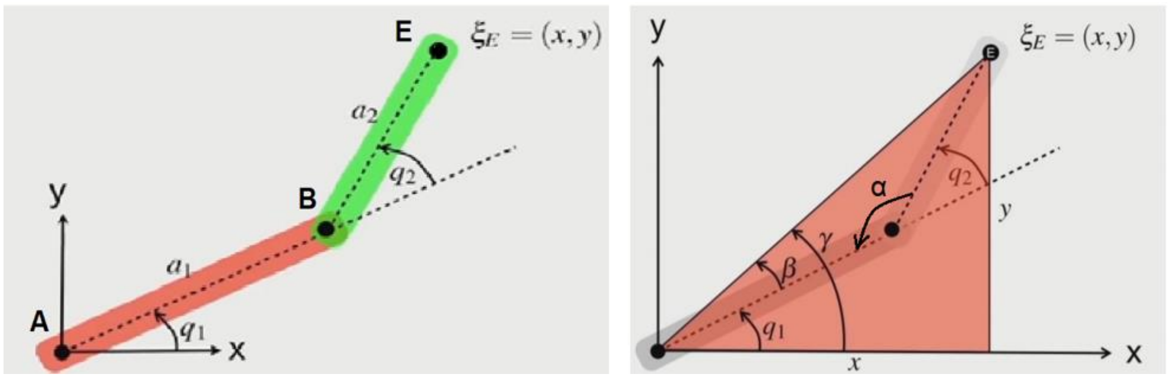


Fig 1.1: 2Linked Planar Robotic Manipulator

$$r^2 = x^2 + y^2$$

$$r^2 = a_1^2 + a_2^2 - 2a_1a_2\cos\alpha$$

$$\cos\alpha = \frac{a_1^2 + a_2^2 - r^2}{2a_1a_2}$$

$$= \frac{a_1^2 + a_2^2 - x^2 - y^2}{2a_1a_2}$$

$$q_2 = \pi - \alpha$$

$$\cos q_2 = -\cos\alpha$$

$$\cos q_2 = \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2}$$

$$\sin q_2 = \sqrt{1 - \cos q_2^2}$$

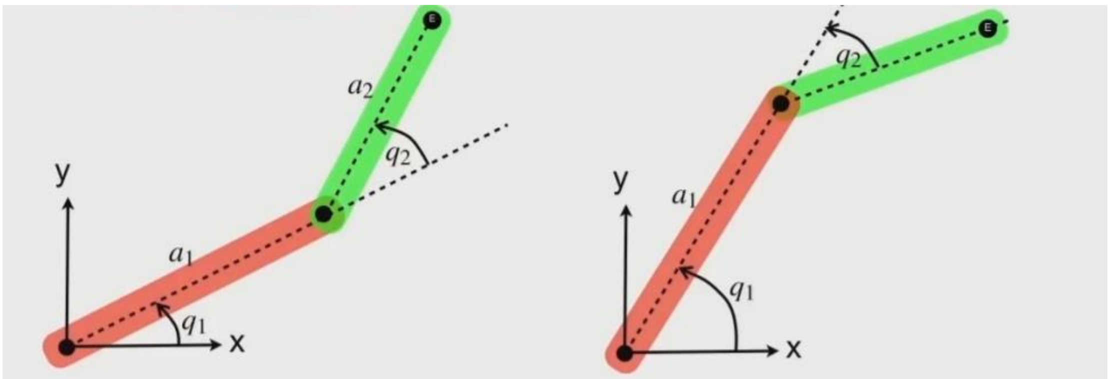
$$\beta = \tan^{-1}\frac{a_2\sin q_2}{a_1 + a_2\cos q_2}$$

$$\gamma = \tan^{-1}\frac{y}{x}$$

$$q_1 = \gamma - \beta$$

$$q_1 = \tan^{-1}\frac{y}{x} - \tan^{-1}\frac{a_2\sin q_2}{a_1 + a_2\cos q_2}$$

Fig 1.2: Inverse Kinematics of 2Linked Planar Robotic Manipulator



$$q_2 = \cos^{-1}\frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2}$$

$$q_1 = \tan^{-1}\frac{y}{x} - \tan^{-1}\frac{a_2\sin q_2}{a_1 + a_2\cos q_2}$$

$$q_2 = -\cos^{-1}\frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2}$$

$$q_1 = \tan^{-1}\frac{y}{x} + \tan^{-1}\frac{a_2\sin q_2}{a_1 + a_2\cos q_2}$$

Fig 1.3: Two possible representations of a planar 2Linked robotic manipulator

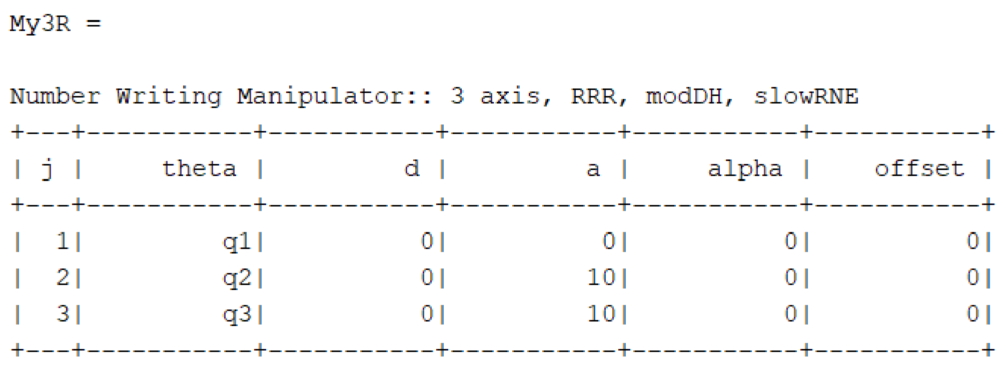
IMPLEMENTATION

We have implemented this using two matlab files:

Manipulator.m:

This code contains a function called 'Manipulator (x1, x2, y1, y2)' with 4 arguments where the arguments are the x and y coordinates of the starting point and the ending point of the desired line to be drawn. Thus, when this function is called it draws the line for the given coordinates.

- **DH parameters** of our manipulator are defined as:



```
My3R =

Number Writing Manipulator:: 3 axis, RRR, modDH, slowRNE
+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|          0|          0|          0|          0|
|  2|         q2|          0|         10|          0|          0|
|  3|         q3|          0|         10|          0|          0|
+---+-----------+-----------+-----------+-----------+-----------+
```

Fig 2.1: DH Parameter Representation of 2Link 3R planar manipulator

- '**SerialLink( )**' function is used to build the manipulator with the above defined DH parameters.

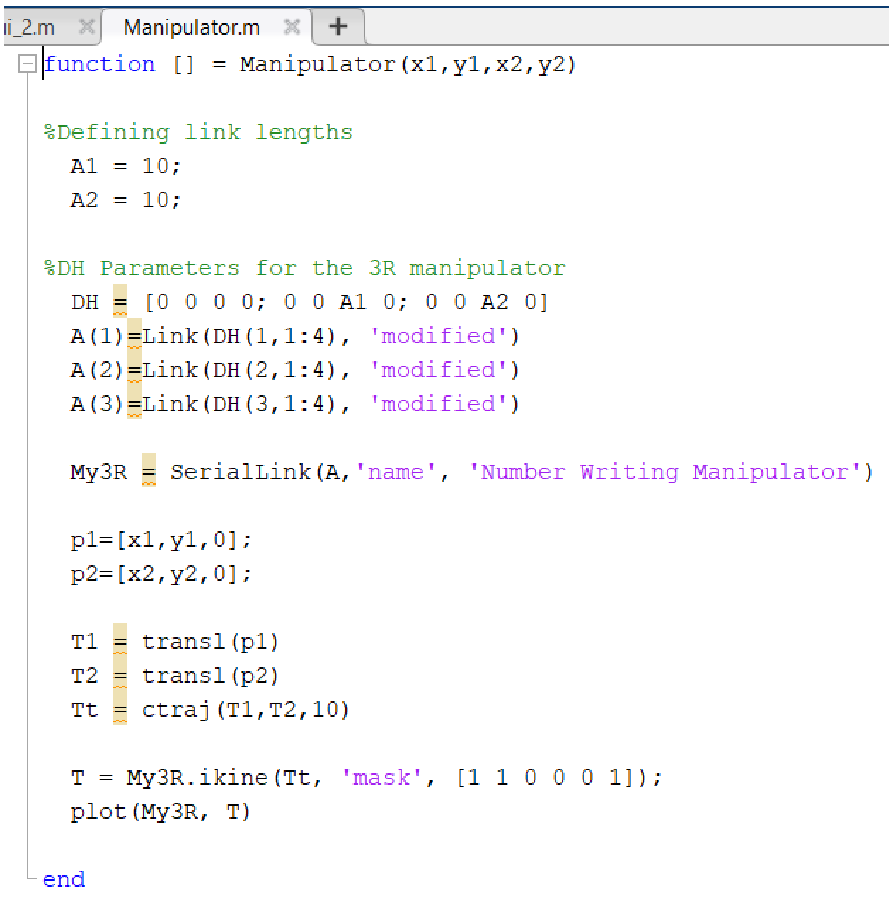
- Matlab command used for trajectory planning is '**ctraj'**. ctraj returns straight path in cartesian space. Command syntax requires beginning and end points in form of translational matrix.

$$Tt = ctraj(Ti, Td, n)$$

where, Ti = initial transformation matrix
Td = desired transformation matrix
n = number of steps

- ‘**ikine**’ function calculates the inverse kinematics of the manipulator.

  T = My3R.ikine(Tt, m, options) are the joint coordinates (1xN) corresponding to the robot end-effector pose Tt (4x4) which is a homogenenous transform where m is a mask vector(1x6) which specifies the Cartesian DOF. The mask vector has six elements that correspond to translation in X, Y and Z, and rotation about X, Y and Z respectively.



```matlab
function [] = Manipulator(x1,y1,x2,y2)

%Defining link lengths
  A1 = 10;
  A2 = 10;

%DH Parameters for the 3R manipulator
  DH = [0 0 0 0; 0 0 A1 0; 0 0 A2 0]
  A(1)=Link(DH(1,1:4), 'modified')
  A(2)=Link(DH(2,1:4), 'modified')
  A(3)=Link(DH(3,1:4), 'modified')

  My3R = SerialLink(A,'name', 'Number Writing Manipulator')

  p1=[x1,y1,0];
  p2=[x2,y2,0];

  T1 = transl(p1)
  T2 = transl(p2)
  Tt = ctraj(T1,T2,10)

  T = My3R.ikine(Tt, 'mask', [1 1 0 0 0 1]);
  plot(My3R, T)

end
```

Fig 2.2: Code in Manipulator.m file

gui_2.m:

- **GUI:**

In command window, type ‘guide’. We get a window for graphical interface as in [Fig 2.3]. There we get multiple options to create a graphical window. We’ve used buttons, text and graphical axes to represent the working of our manipulator in this GUI.

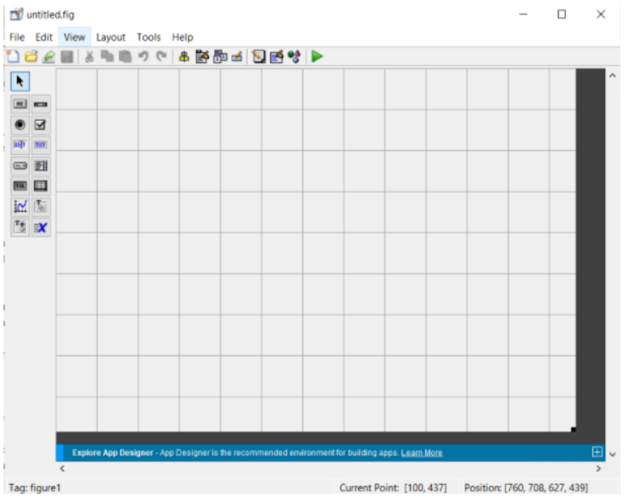
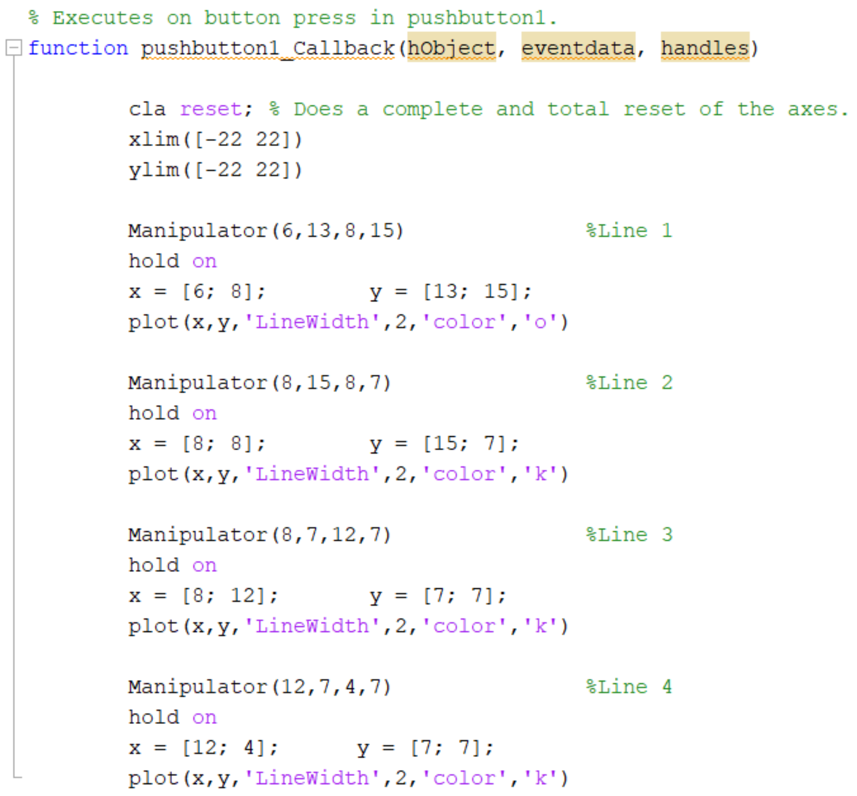

untitled.fig
File Edit View Layout Tools Help
Explore App Designer - App Designer is the recommended environment for building apps. Learn More
Tag: figure1
Current Point: [100, 437]
Position: [760, 708, 627, 439]

Fig 2.3: New GUI window in matlab

- ‘gui_2.m’ file contains the gui interface code, which is used to represent our final manipulator which draws a number based on user’s choice. The user will be shown 6 buttons on one side of the screen with numbers {1, 3, 4, 5, 9, 0} and a empty graph on the other side. If the user clicks ‘1’, the below code is executed:



```
% Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

        cla reset; % Does a complete and total reset of the axes.
        xlim([-22 22])
        ylim([-22 22])

        Manipulator(6,13,8,15)              %Line 1
        hold on
        x = [6; 8];         y = [13; 15];
        plot(x,y,'LineWidth',2,'color','o')

        Manipulator(8,15,8,7)               %Line 2
        hold on
        x = [8; 8];         y = [15; 7];
        plot(x,y,'LineWidth',2,'color','k')

        Manipulator(8,7,12,7)               %Line 3
        hold on
        x = [8; 12];        y = [7; 7];
        plot(x,y,'LineWidth',2,'color','k')

        Manipulator(12,7,4,7)               %Line 4
        hold on
        x = [12; 4];       y = [7; 7];
        plot(x,y,'LineWidth',2,'color','k')
```

Fig 2.4: Code to draw number '4' in gui_2.m

In the above code, Manipulator() function is called whenever a new line is to be drawn. Here, 4 lines are required to draw '1'.

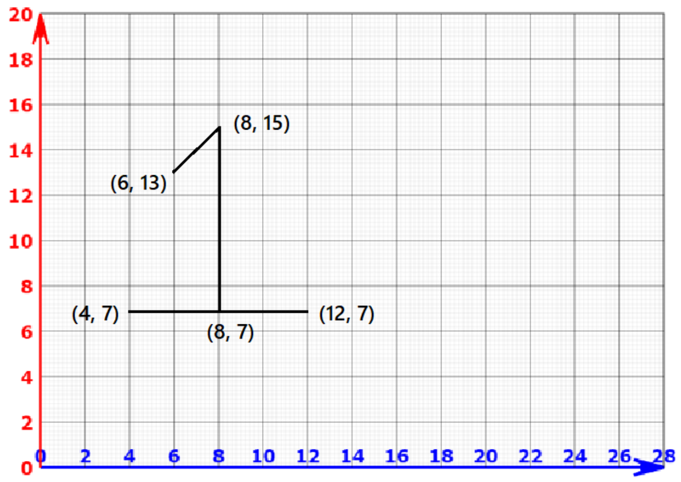


Fig 2.5: Coordinate representation of number '1' as drawn in gui_2.m

Similarly, it is coded in the same way for all the other cases with numbers 3, 4, 5, 9, 0 with the desired lines to draw that particular number.

RESULTS

Implementation of the number writing manipulator is obtained as desired.

After running the Matlab code, a new window pops up like in (fig 3.1)

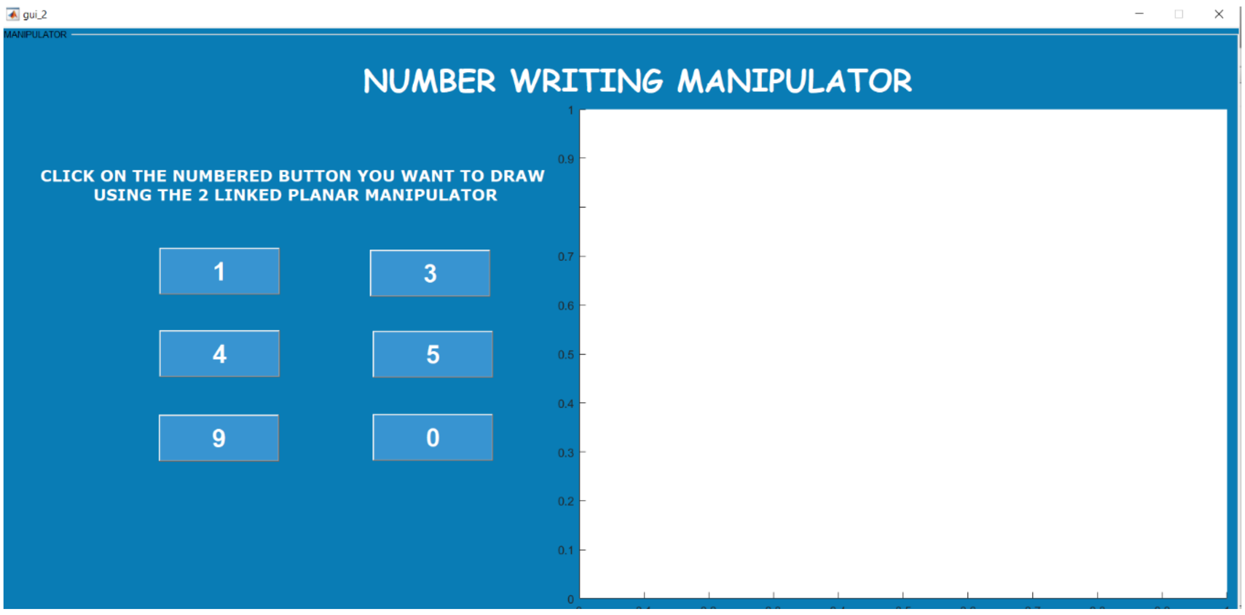

gui_2
MANIPULATOR
NUMBER WRITING MANIPULATOR
CLICK ON THE NUMBERED BUTTON YOU WANT TO DRAW USING THE 2 LINKED PLANAR MANIPULATOR
1
3
4
5
9
0

Figure 3.1

CASE 1:

Click on 1 and you can see the manipulator writing '1' in the graph (fig 3.2)

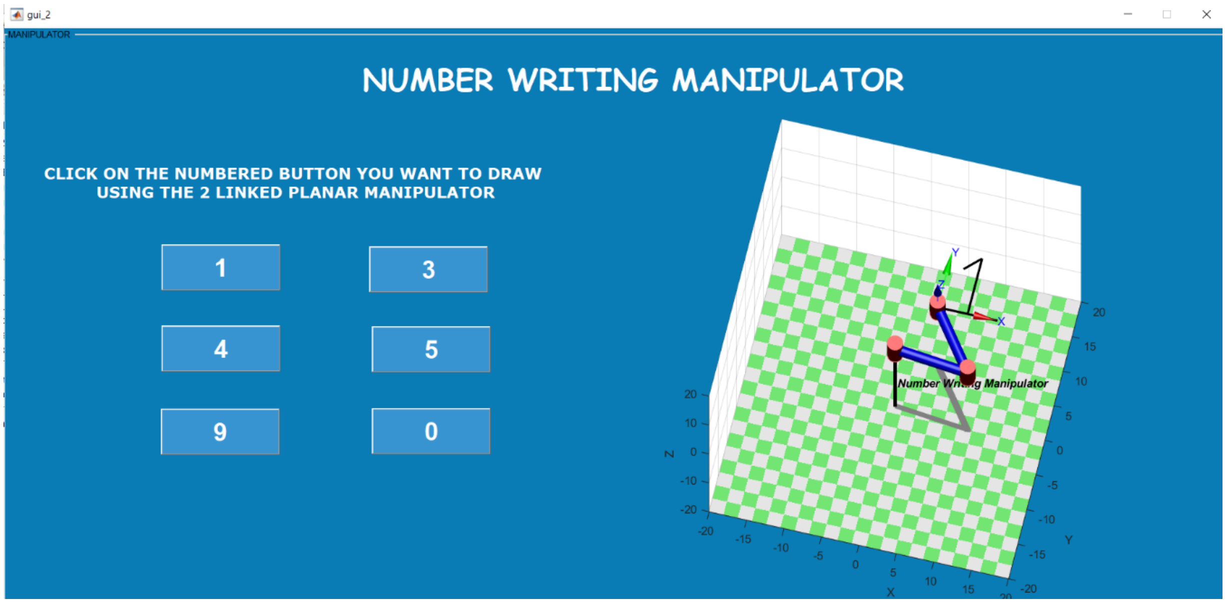

gui_2
MANIPULATOR
NUMBER WRITING MANIPULATOR
CLICK ON THE NUMBERED BUTTON YOU WANT TO DRAW USING THE 2 LINKED PLANAR MANIPULATOR
1
3
4
5
9
0
Number Writing Manipulator
X
Y
Z

Figure 3.2

CASE 2:

Click on 3 and you can see the manipulator writing ‘1’ in the graph (fig 3.3)

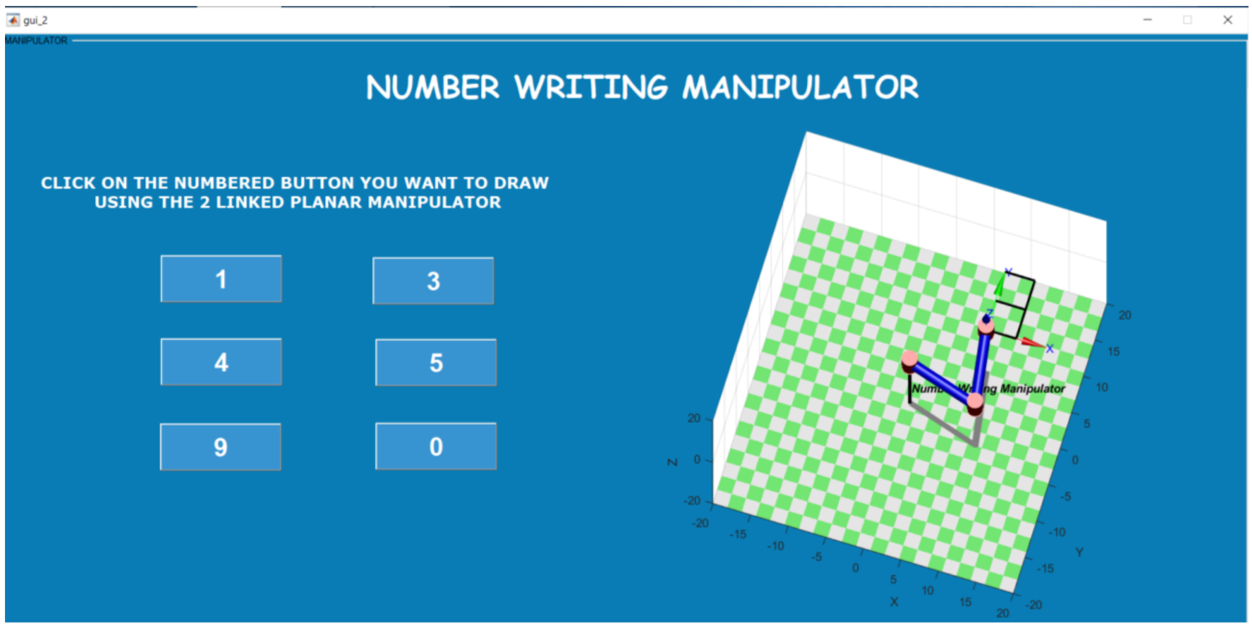

gui_2
MANIPULATOR
NUMBER WRITING MANIPULATOR
CLICK ON THE NUMBERED BUTTON YOU WANT TO DRAW
USING THE 2 LINKED PLANAR MANIPULATOR
1
3
4
5
9
0
Number Writing Manipulator
X
Y
Z

Figure 3.3

CASE 3:

Click on 4 and you can see the manipulator writing ‘4’ in the graph (fig 3.4)

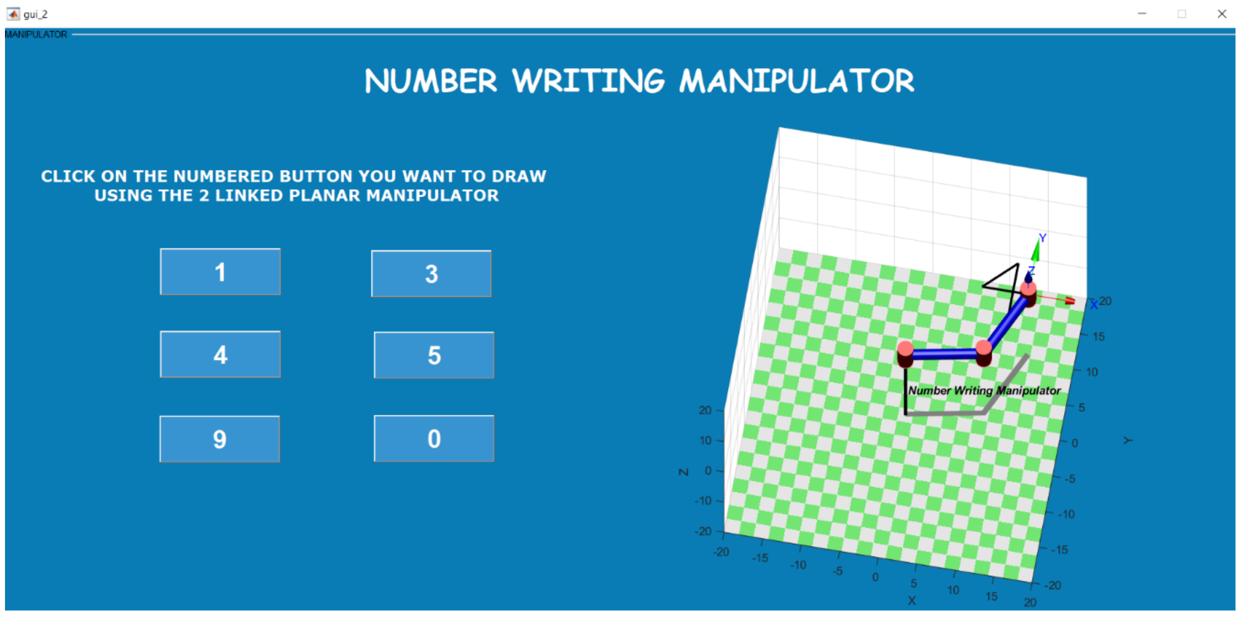

gui_2
MANIPULATOR
NUMBER WRITING MANIPULATOR
CLICK ON THE NUMBERED BUTTON YOU WANT TO DRAW
USING THE 2 LINKED PLANAR MANIPULATOR
1
3
4
5
9
0
Number Writing Manipulator
X
Y
Z

Figure 3.4

CASE 4:

Click on 5 and you can see the manipulator writing ‘5’ in the graph (fig 3.5)

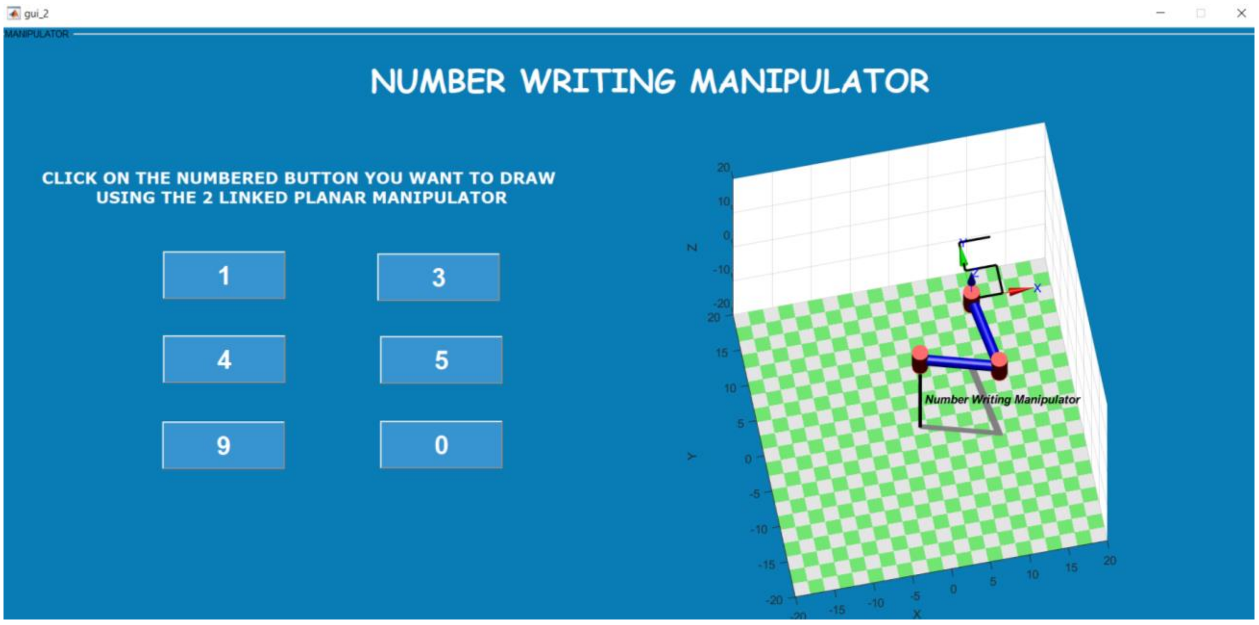

gui_2
MANIPULATOR
NUMBER WRITING MANIPULATOR
CLICK ON THE NUMBERED BUTTON YOU WANT TO DRAW
USING THE 2 LINKED PLANAR MANIPULATOR
1
3
4
5
9
0
Number Writing Manipulator
X
Y
Z

Figure 3.5

CASE 5:

Click on 9 and you can see the manipulator writing ‘9’ in the graph (fig 3.6)

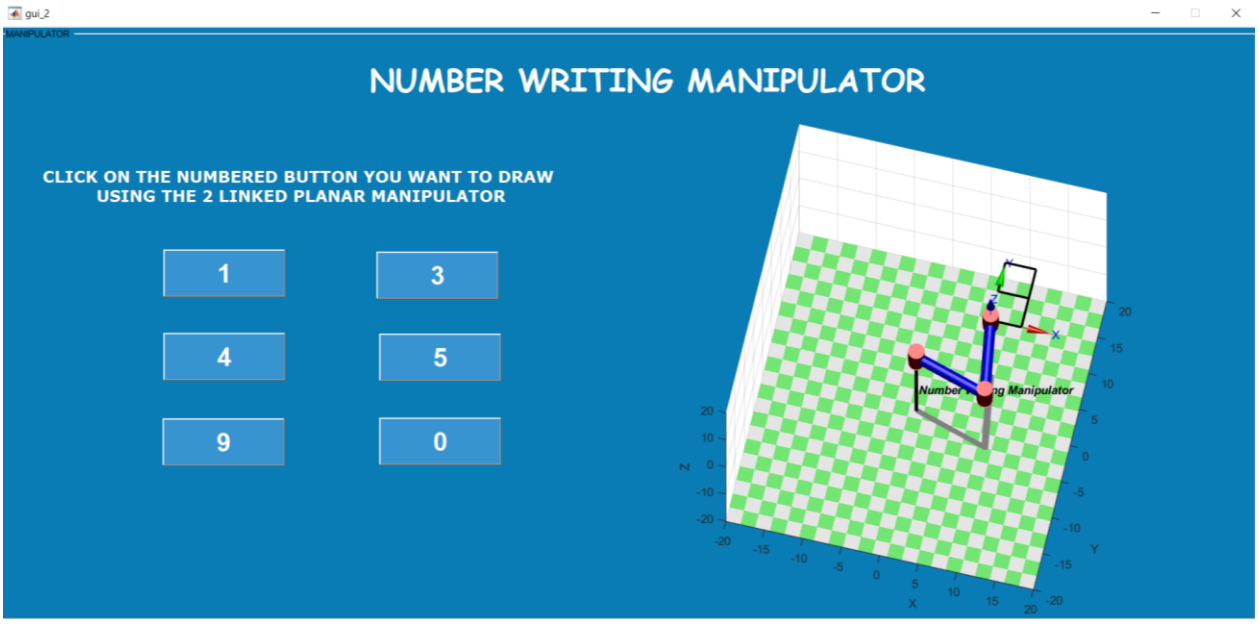

gui_2
MANIPULATOR
NUMBER WRITING MANIPULATOR
CLICK ON THE NUMBERED BUTTON YOU WANT TO DRAW
USING THE 2 LINKED PLANAR MANIPULATOR
1
3
4
5
9
0
X
Y
Z

Figure 3.6

CASE 6:

Click on 0 and you can see the manipulator writing ‘0’ in the graph (fig 3.7)

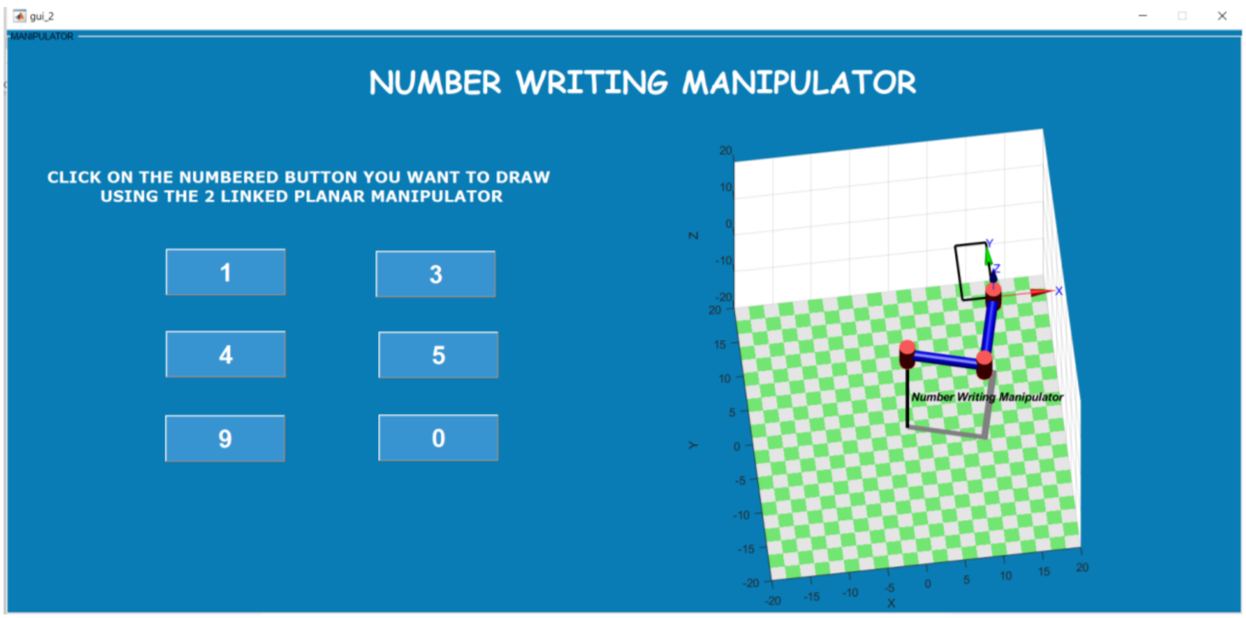

gui_2
MANIPULATOR
NUMBER WRITING MANIPULATOR
CLICK ON THE NUMBERED BUTTON YOU WANT TO DRAW
USING THE 2 LINKED PLANAR MANIPULATOR
1
3
4
5
9
0
Number Writing Manipulator
X
Y
Z

Figure 3.7

REFERENCES

- *http://www.sml.ee.upatras.gr/UploadedFiles/InverseKinematics.pdf*
- *https://www.researchgate.net/publication/341070239_Design_of_2-DOF_Robot_Manipulator*
- *https://www.youtube.com/watch?v=Oh68VtdjiLk*