**WALL – PAINTING MANUPULATOR**

A PROJECT REPORT

### **TEAM** – PREFECT DEALERS

BL.EN.U4AIE19028 K. VISHNU SAINADH

BL.EN.U4AIE19034 K. SATWIK

BL.EN.U4AIE19066 V.ASHRITH

***19AIE201 – Introduction to Robotics***

***in partial fulfilment for the award of the degree 0f***

## BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE ENGINEERING



AMRITA SCHOOL OF ENGINEERING, BANGALORE AMRITA VISHWA VIDYAPEETHAM

**BANGALORE 560 035**

**PROBLEM STATEMENT**

Design and implement a wall painting robot with a suitable degree of freedom. The inputs for the system should be the dimensions of the wall patch to be painted and number of coatings.

**IMPLEMENTATION**

We have implemented a 2P-planar manipulator. It has 2 DOF (degree of freedom).

Here we are having two matlab files for the implementation of our manipulator.

* Painting\_Robot\_A.m
* IR\_Project\_A.m

**Painting\_Robot\_A.m:**

This file is our function file, it consists of the position of our end effector and it has 3 arguments (Pi,Pd,Y\_plane) where,

Pi - it is the initial point of our end effector

Pd - it is the destination point of our end effector

Y\_plane - it is the plane on which we paint (wall)

DH – PARAMETERS: -

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| j | Link | theta | d | a | alpha | offset | Q-Limit |
| 1  2 | Prismatic  Prismatic | 90 90 | 0  0 | 0  0 | 90  90 | 10  10 | 0-100  0-100 |

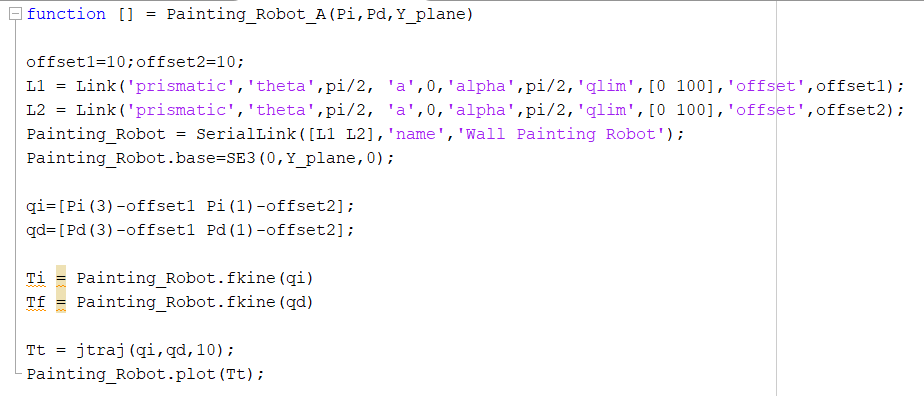


Fig (1)

**IR\_Project\_A.m:**

This is our main matlab file. Here our code starts with asking these inputs from the user:

1. Dimensions of the wall
2. Number of coatings
3. Width of the paint brush
4. The start coordinates of the wall patch

Then, from the given information we see whether we need to draw either horizontal or vertical strokes. If the length is greater than breadth then we draw horizontal strokes, if the breadth is greater than length then we draw vertical strokes. Now we calculate the number of strokes by dividing the width of the paint brush with either length or breadth depending on the dimensions of the wall. Now with the help of start coordinate we calculate all the other points of our patch.

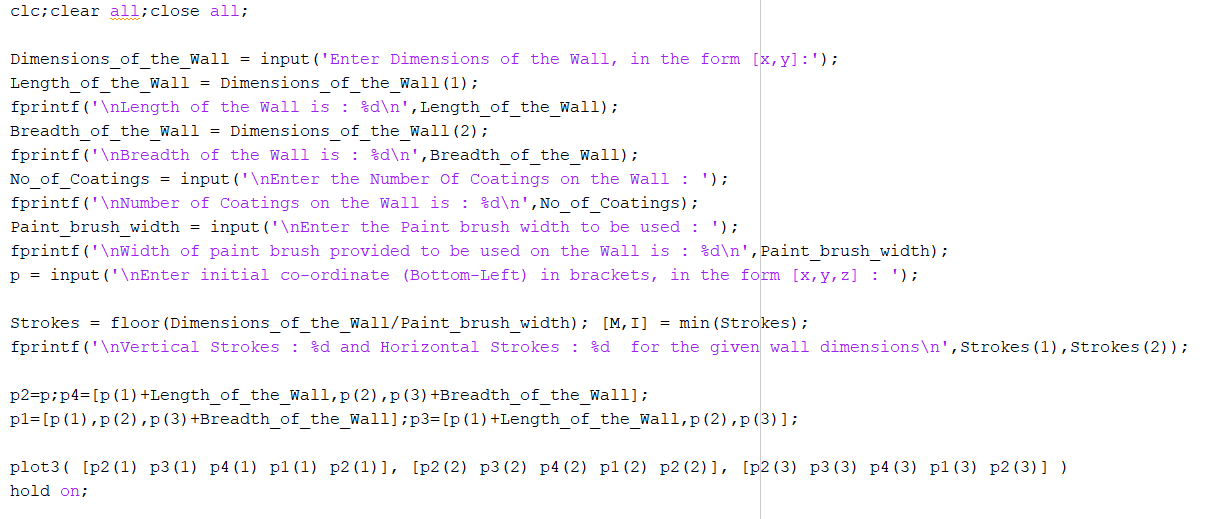


Fig (2)

Here we now have two cases:

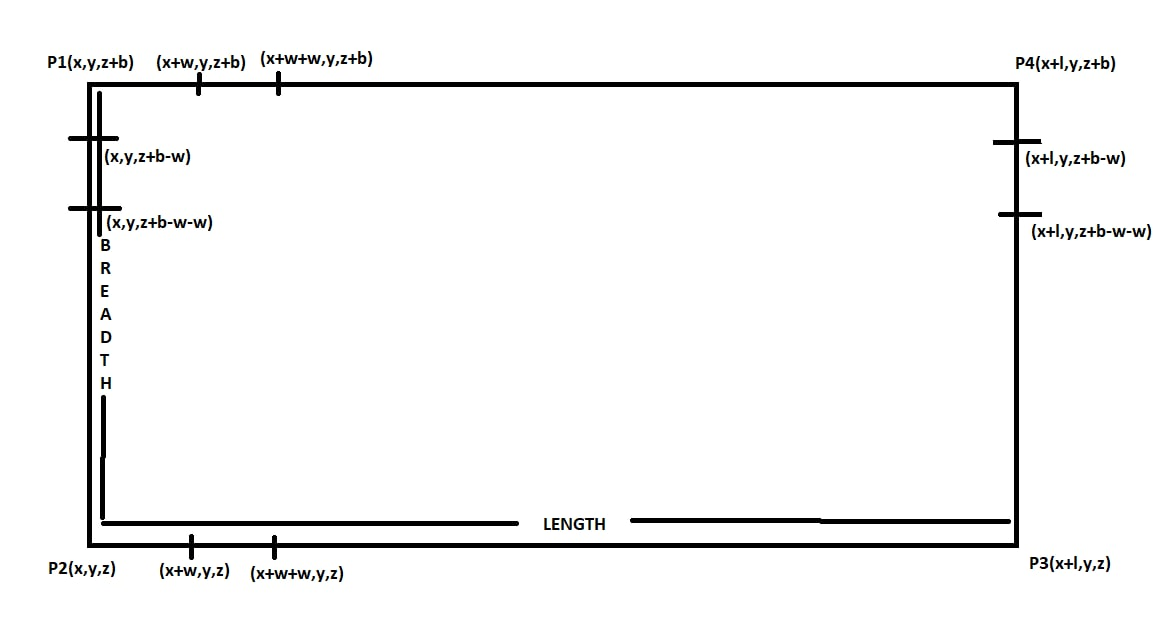


Fig (3)

Case 1: vertical strokes

This case is considered when the breadth of the wall patch is more than the length of the wall patch. If you see in Fig (3), P2 is the starting point and our end effector goes towards P1 increasing the z-coordinate by the width of the wall patch and then our end effector moves by the width of the paintbrush in x-direction and then it comes down to the base height. And then this process repeats until the wall is filled with paint.

Exception situation: when the width of the paint brush is not perfectly divisible with the length of the wall then the last stroke’s starting x-coordinate will be x-coordinate of point P3 - width of the brush.

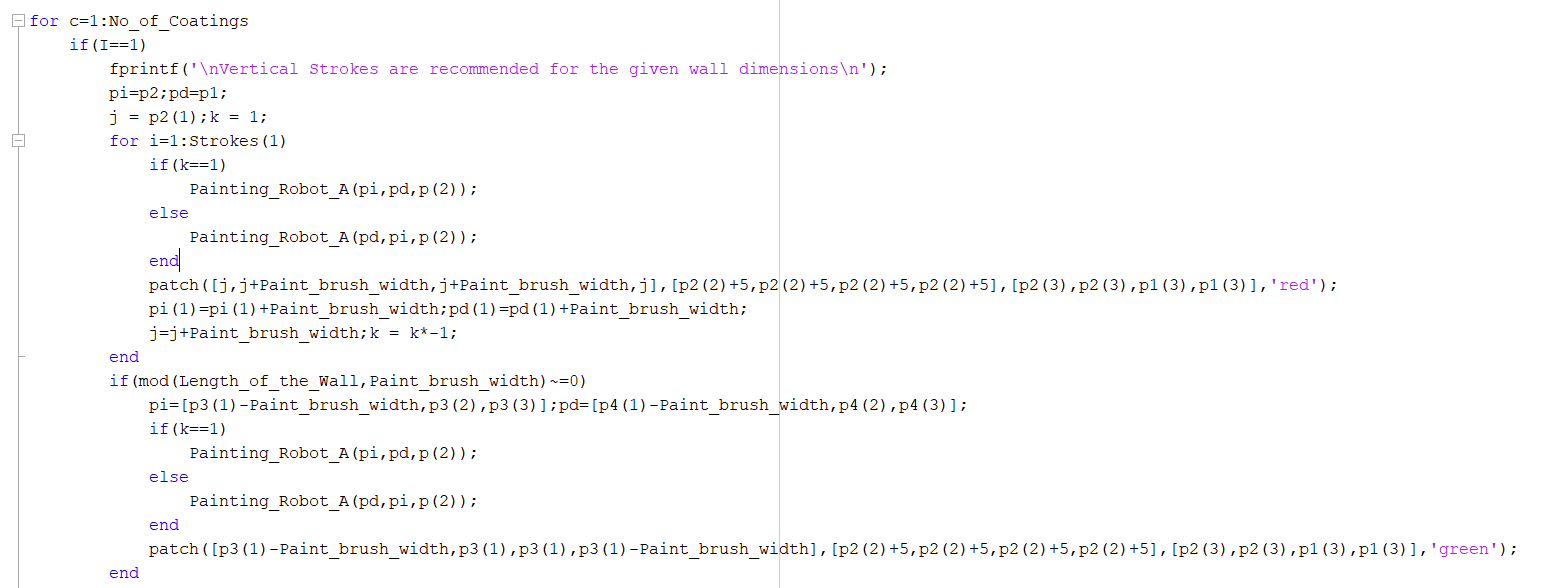


Fig (4)

Case 2: horizontal strokes

This case is considered when the length of the wall patch is more than the breadth of the wall patch. If you see in Fig (3), P2 is the starting point and in horizontal case the end effector starts painting from the top so, P1 becomes the starting point and our end effector goes towards P3 increasing the x-coordinate by the width of the wall patch and then our end effector moves by the width of the paintbrush in negative z-direction and then it goes in negative x- direction to the initial x-coordinate of point P1. And then this process repeats until the wall is filled with paint.

Exception situation: when the width of the paint brush is not perfectly divisible with the breadth of the wall then the last stroke’s starting z-coordinate will be z-coordinate of point P3 + width of the brush.

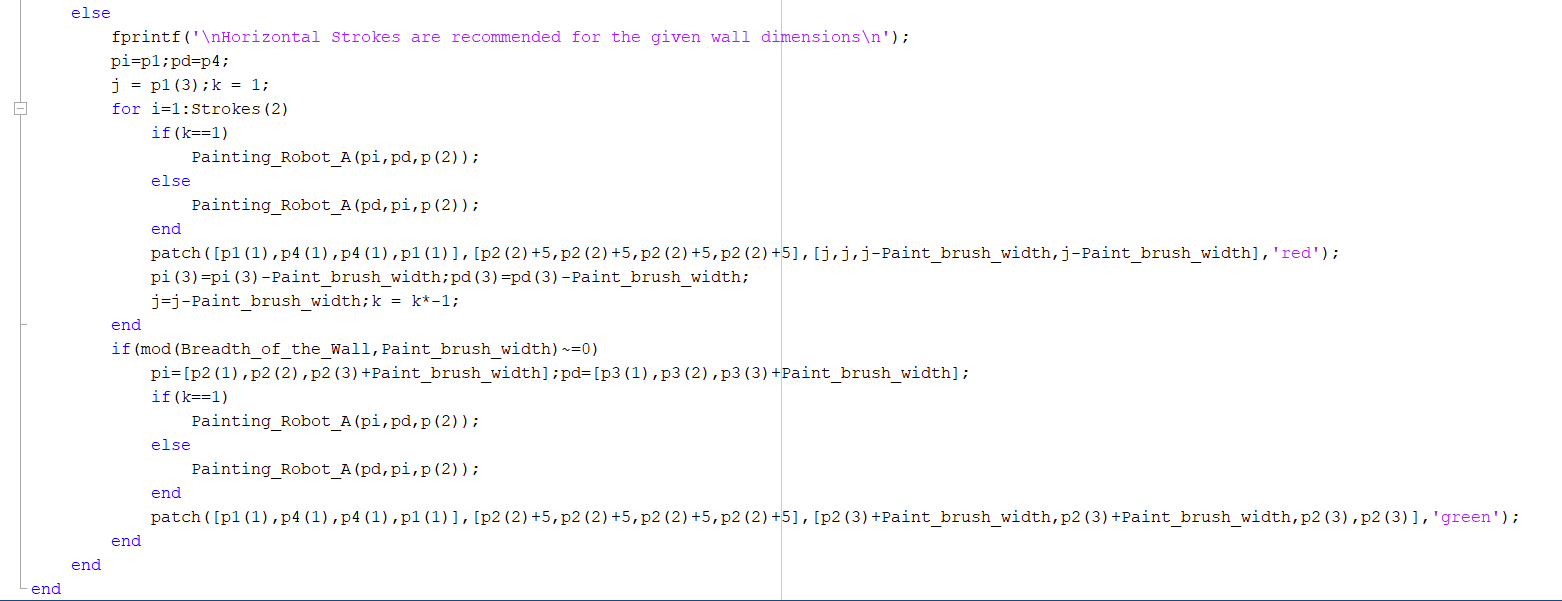


Fig (5)

**RESULTS**

Case 1: vertical strokes

After giving the necessary inputs

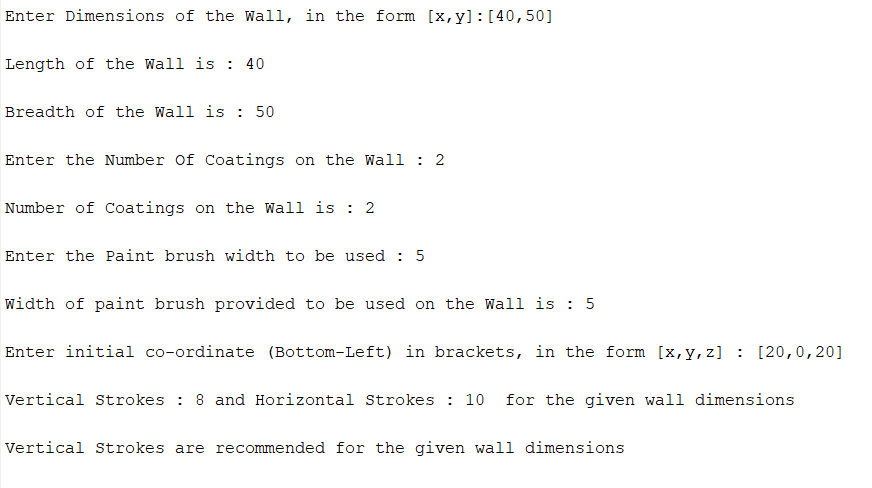


Fig (6)

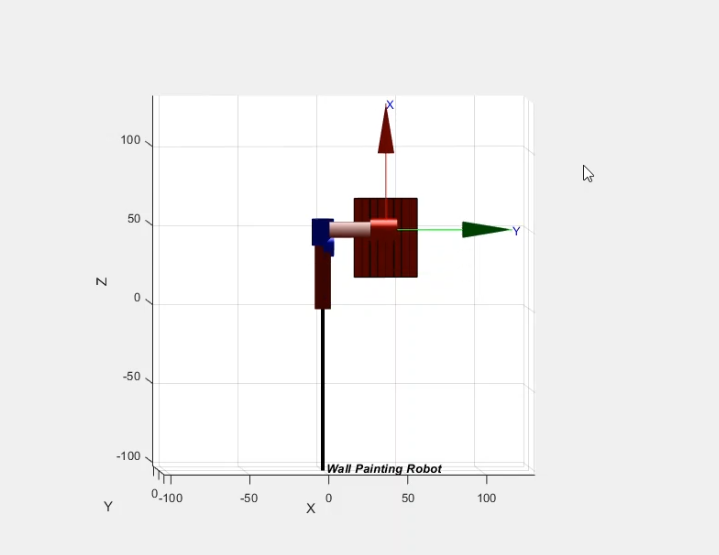


Fig (7)

Case 2: horizontal strokes

After giving the necessary inputs

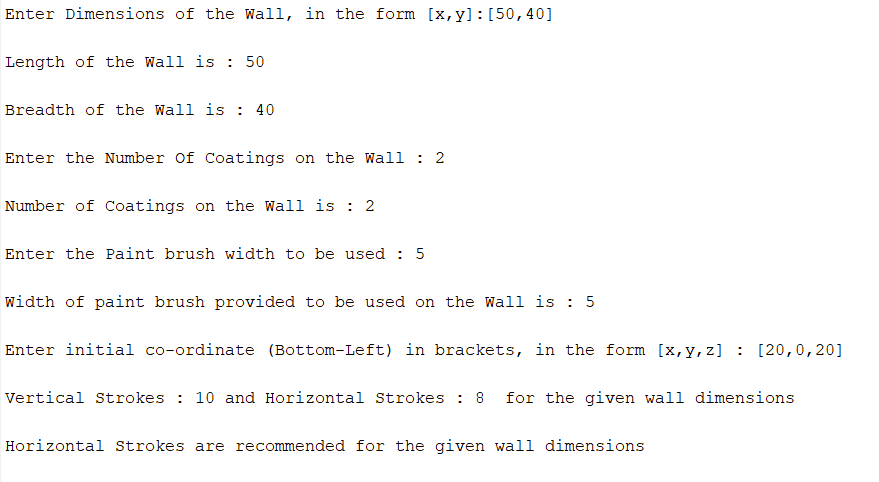


Fig (8)

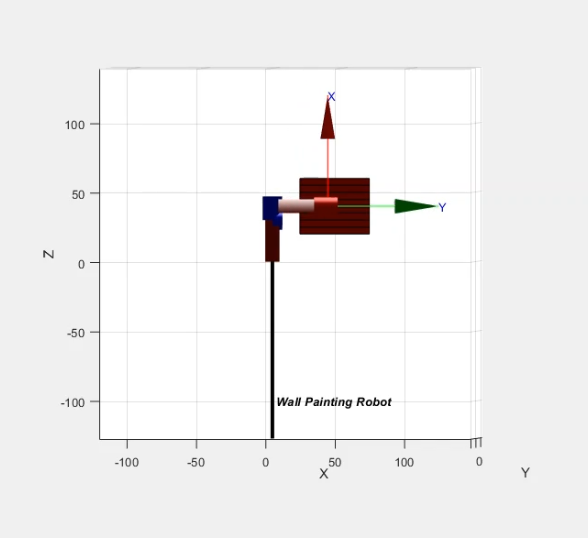


Fig (9)

**REFERENCES:**

[1]<https://www.researchgate.net/publication/236659040_Design_of_An_Autonomous_Wall_Painting_Robot>

[2]Robot Kinematics: Forward and Inverse Kinematics a book by Serdar Kucuk and Zafer Bingul