University of the West of England

Serial and Parallel Robot Kinematics

Yi Li and Shuhao Kang

January 14, 2021

UFMF4X-15-M Robotic Fundamentals Lecturer: Dr Aghil Jafari

Coursework Report - Group Cover Page

Coursework: Serial and Parallel Robot Kinematics

Student name and number	PART I.A	PART I.B	PART II.1	PART II.2	PART III
Shuhao Kang 20052575	√	√	√	√	√
Yi Li 20053963	√	√	√	√	√

Note: Students must tick the sections that they have been undertaking or helped with.

Abstract

The concept of kinematics has always been an important part of robot control. This report demonstrates our understanding of tandem and parallel robots. In the first part the DH representation of the Lynxmotion arm is emphasised, while the second part focuses on a planar parallel robot for shell surgery, with the main knowledge involved being inverse kinematics. In addition, in part 3 this paper investigates the theory, functions and implementation of the PRM algorithm using matlab. It also presents the significance of applying the path planning algorithm to a robot arm. The following sections provide a detailed discussion.

Keywords: MATLAB; Denavit-Hartenberg; Planar parallel robot; Path Planning; PRM;

Contents

PART 1

Kinematics of Lynxmotion Arm

1.1 Forward Kinematics

Forward kinematics refers to the use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters (Paul, 1981). This chapter demonstrates the relevant knowledge of DH representation of forward kinematics.

1.1.1 DH Representation

Assigning the Coordinate Frames

DH convention needs to assign a coordinate system on every link so as to describe the relative motion and the relation of attitudes between the adjacent links. Lynxmotion arm's coordinate systems are assigned following the regulation below:

1. Assigning base frame (Link 0):

   It is a revolute joint, so:

$$a_0 = 0$$
$$\alpha_0 = 0$$
$$d_1 = 0$$

2. Assigning intermedia links' coordinate frames (Link i):

   Axis Z_i of Coordinate frame {i} and the axis of join i are collinear, and the direction is determined by Right-handed rules. Axis X_i of Coordinate frame {i} coincides with the common perpendicular of joint i and joint i+1, and the direction is from i to i+1. Correspondingly, axis Y_i can be defined via Right-handed rules. The origin O_i is the point of intersection of X_i and Z_i. When Z_i and Z_i +1 intersect, the point of intersection is origin. When Z_i and Z_i +1 are parallel, the origin should make the d_i =0, which means that X_i and $X_{\mathrm{i}+1}$ are collinear.

3. Assigning of end-effector's frame (Link n):

   It is a revolute joint, and θ_n is assumed as 0, so X_n is parallel with axis $X_{\mathrm{n}-1}$. d_n is defined as 0.

Based on the regulation above, the model of Lynxmotion arm with coordinate systems is shown in Figure 1.1.

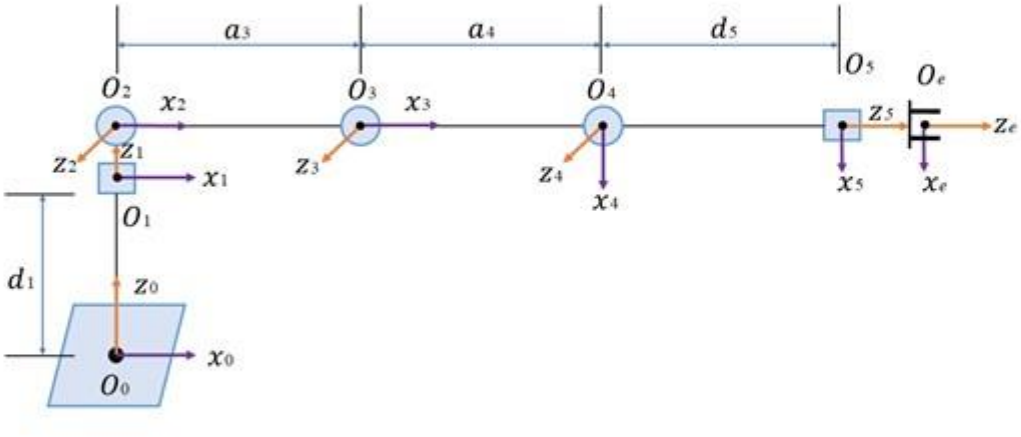


Figure 1.1: Modified D-H Convention Model

D-H Parameters

After the establishment of coordinate frames, DH parameters can be determined straightly due to the adjacent links. For the coordinate frames {i-1} and {i}, the way to determine DH parameters is shown as below:

a_{i-1}: The distance from Z_{i-1} to Z_i along X_{i-1}

α_{i-1}: The rotation angle from Z_{i-1} to Z_i around X_{i-1}

d_i: The distance from X_{i-1} to X_i along Z_i

θ_i: The rotation angle from X_{i-1} to X_i around Z_i

All link lengths are defined as 0.1m, and the DH parameters table is shown as Table 1.1.

Table 1.1. DH Parameters.

Link i	α_{i-1}	a_{i-1}	θ_i	d_i
1	0	0	θ_1	d_1
2	90°	0	θ_2	0
3	0	a_3	θ_3	0
4	0	a_4	θ_4	0
5	-90°	0	θ_5	d_5

Specially, because θ_5 does not influence the position of end-effector, it may be assumed as 0.

D-H Matrix

By means of the transformation of coordinate frames, every coordinate frame rotates or translates relating to its own axis, so the transformation matrix from link i-1 to link i is demonstrated as follow:

$$^{i-1}T_i = R(x_{i-1}, \alpha_{i-1})T(x_{i-1}, a_{i-1})R(z_i, \theta_i)T(z_i, d_i) \tag{1.1}$$

$$= \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & a_{i-1} \\ sin\theta_i cos\alpha_{i-1} & cos\theta_i cos\alpha_{i-1} & -sin\alpha_{i-1} & -sin\alpha_{i-1}d_i \\ sin\theta_i sin\alpha_{i-1} & cos\theta_i sin\alpha_{i-1} & cos\alpha_{i-1} & cos\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2}$$

According to the equation and parameters above, the homogeneous transformations between every two adjacent coordinate frames ($^0T_1, {}^1T_2, {}^2T_3, {}^3T_4, {}^4T_5$) can be calculated.

The position and orientation of end-effector can be figured:

$$^0T_5 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \tag{1.3}$$

The numbers in the top three rows and the fourth column are the coordinates of joints.

MATLAB Test

In this test, four sets of parameters about the joints' angles are given in Table 1.2 (All the parameters' unit is degree):

Table 1.2. Four Sets of Joint Parameters.

q_1	q_2	q_3	q_4
50	-25	45	-60
60	-30	50	-55
70	-35	55	-50
80	-40	60	-45

Using Equation1.2 and Equation1.3, the position of every joint will be figured, and the end-effector's positions and orientations are plotted as below:

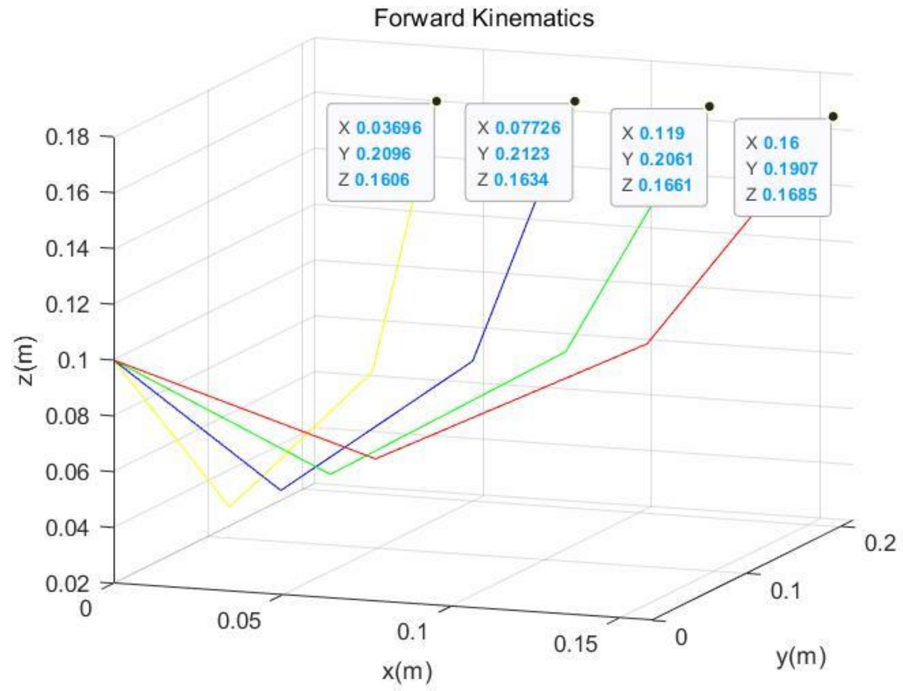
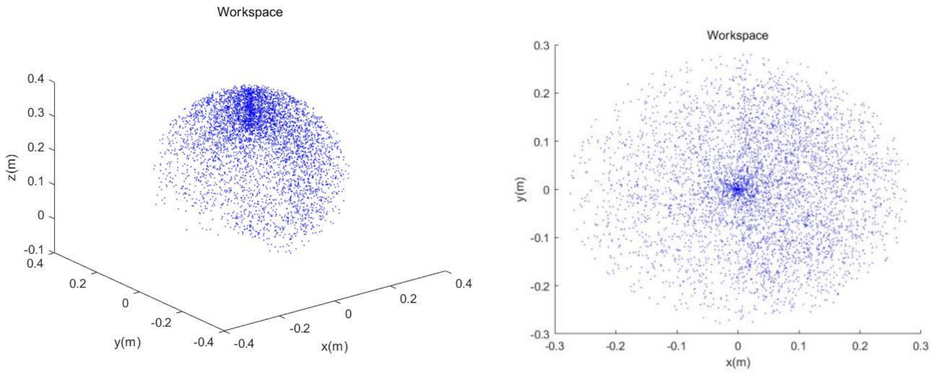


Figure 1.2: FK Test in MATLAB

The corresponding MATLAB file is FK_test.m.

1.1.2 Workspace

5000 random numbers of every link's rotational angle (θ) are set, and due to the forward kinematics, the positions of the end-effector are calculated. Because the samples are large enough, the sample points can basically cover the real workspace (Fig 1.3). Specially, all the random numbers are set under the limitation of every link's angle (Table 1.3). The corresponding MATLAB file is FK_workspace.m.



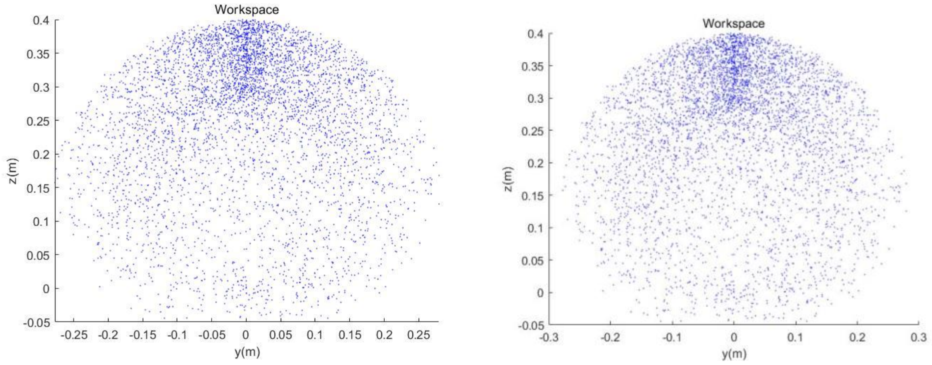


Figure 1.3: Workspace with 3D and 2D Views

Table 1.3. Limitation of Joint Angles.

Joint Angles	Range(°)
q_1	(-90,90)
q_2	(30,150)
q_3	(-135,45)
q_4	(-150,30)

1.2 Inverse Kinematics

There are two solutions of inverse kinematics: Numerical Solution and Analytical Solution, and Analytical Solution is introduced particularly.

1.2.1 Analytical Approach

The analytical Approach is also named Closed Form Solution, which can use the analytical expression to calculate joint parameters. In inverse kinematics, the known information including the end-effector's orientation and position, and it can be formed via a 4-by-4 homogeneous transformation matrix, seen in 1.12.

There are two approaches of Analytical Solution: Geometric Method and Algebraic Method, and Geometric Method is proposed as following.

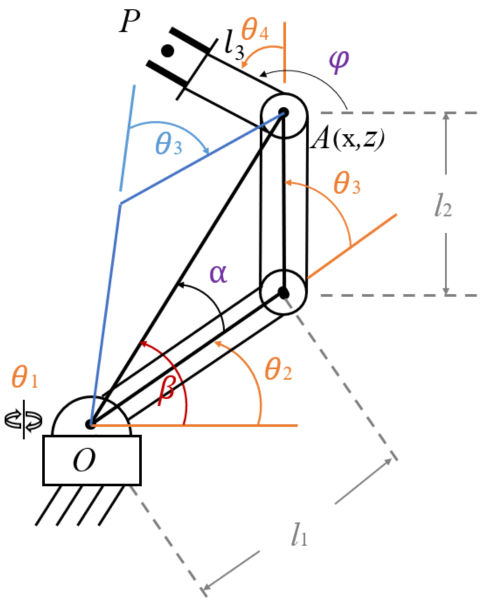


Figure 1.4: Ichnography of Lynxmotion Arm

At first, the joint's anticlockwise rotation is defined as positive value, and the clockwise rotation is defined as negative value. Assuming the coordinate of A as (x_1, y_1, z_1), θ_1 is a rotation of vertical orientation, and it can be given using Equation 1.4:

$$\theta_1 = arctan2(y_1\,, x_1) \tag{1.4}$$

As Fig 1.4, the end-effector's coordinate is given:

$$x\ = x_1 + l_3 cos\varphi \tag{1.5}$$

$$z = z_1 + l_3 sin\varphi \tag{1.6}$$

The length of OA can be represented:

$$|OA|^2 = {x_1}^2 + {z_1}^2 \tag{1.7}$$

Additionally, based on law of cosines,

$$|OA|^2 = {x_1}^2 + {z_1}^2 = l_1^2 + l_2^2 - 2l_1l_2\cos{(180 - |\theta_3|)} \tag{1.8}$$

Then,

$$\theta_3 = \arccos{(\frac{x_1^2+z_1^2-l_1^2-l_2^2}{2l_1l_2})} \tag{1.9}$$

x_1 and y_1 need to meet the condition:

$$\sqrt{x_1^2 + z_1^2} \leq l_1 + l_2 \tag{1.10}$$

θ_3 may have two solutions, and another is $\theta_3' = -\theta_3$, as the blue outline in Fig 1.4.

Moreover,

$$\beta = arctan2(\, z_1, x_1) \tag{1.11}$$

$$cos\alpha = \frac{x_1^2+z_1^2+l_1^2-l_2^2}{2l_1\sqrt{x_1^2+z_1^2}} \tag{1.12}$$

So,

$$\theta_2 = \begin{cases} \beta + \alpha & \theta_3 < 0 \\ \beta - \alpha & \theta_3 \geq 0 \end{cases} \tag{1.13}$$

$$\theta_4 = \varphi - \theta_1 - \theta_2 - \theta_3 \tag{1.14}$$

The corresponding MATLAB file is a function written by ourselves named IK.m.

1.3 Trajectory Planning

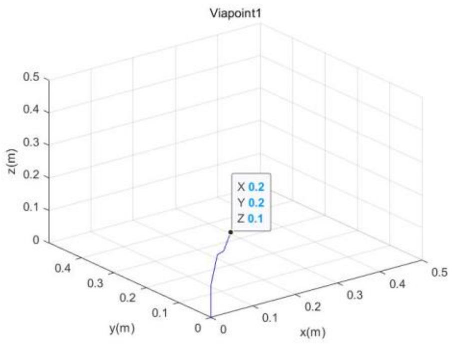
1.3.1 Task Planning

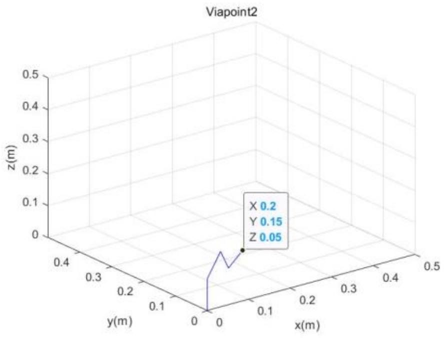
The orientation of the end-effector is determined to be horizontal, and all the via points' parameters are given in Table 1.4.

Table 1.4 Position of Via Points

Via Points	x	y	z
1	0.2	0.2	0.1
2	0.2	0.15	0.05
3	0.2	0.1	0.1
4	0.2	0.05	0.05
5	0.2	0	0.1

On the basis of inverse kinematics, the arm's attitude and position in every via point are plotted in MATLAB, and the corresponding file is FK_TaskPlanning.m. All the pictures are shown below:





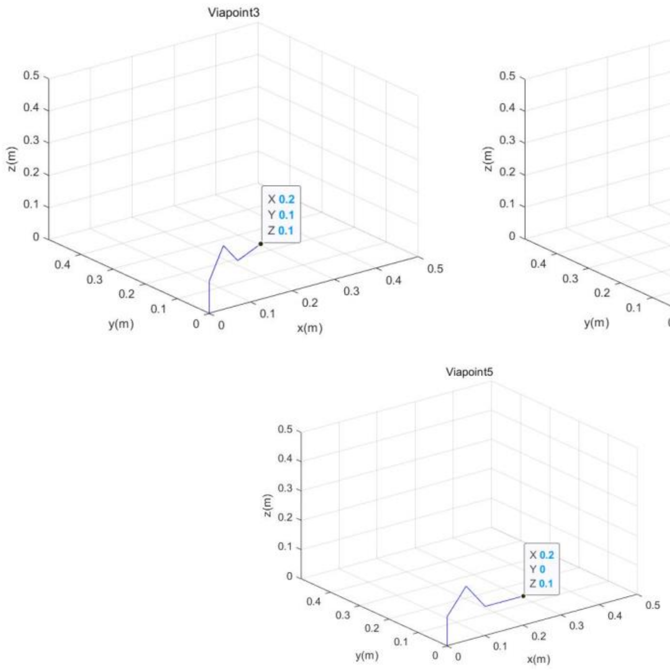


Figure 1.5: Positions and Attitudes of Via Points

1.3.2 Trajectories Implementation

Free Motion

For the free motion's trajectories, five via points are fitted in two conics, and the length of steps of each conic is 50. Additionally, time interval of every step is 0.01s. The corresponding MATLAB file is free_motion.m, and the screenshot of its animation is presented in Fig. 1.6.

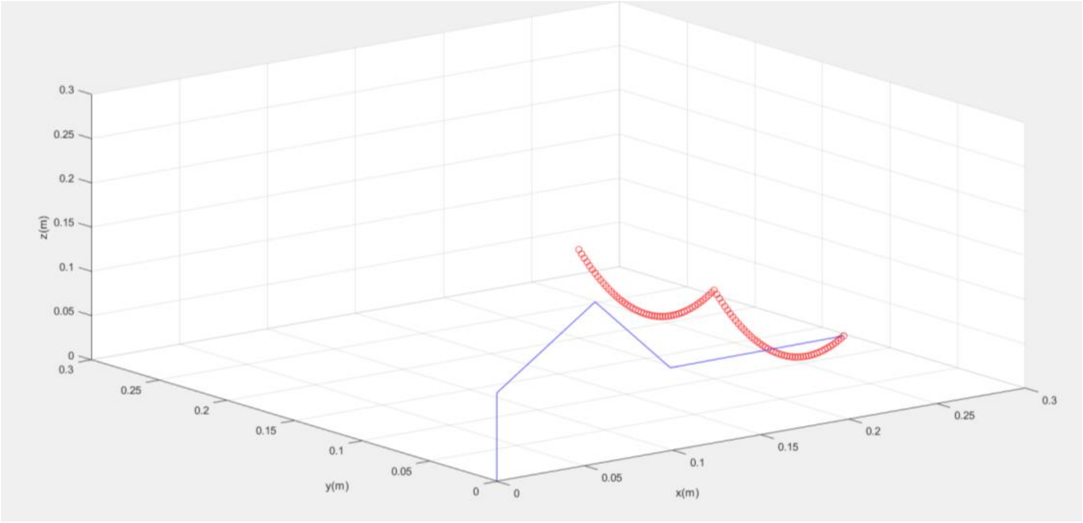


Figure 1.6: Free Motion

Trajectories with Straight Line

In this part, trajectories with linear segments are divided into three parts: uniformly accelerated motion, uniformly velocity motion and uniformly decelerated motion. The relation of velocity and time can be shown in Fig. 1.7.

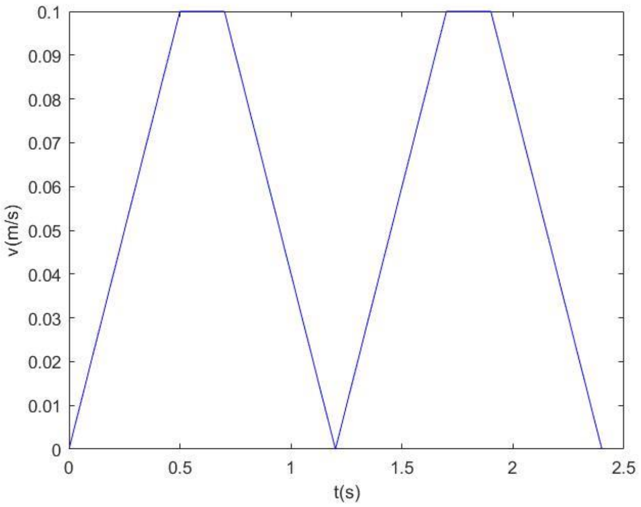


Figure 1.7: Straight Line Trajectories

The screenshot of its animation and the relation of joint values and time are shown in Fig. 1.8 and Fig. 1.9, and the corresponding MATLAB files are Trajectory_straight_line.m and Trajectory_straight_line_qt.m.

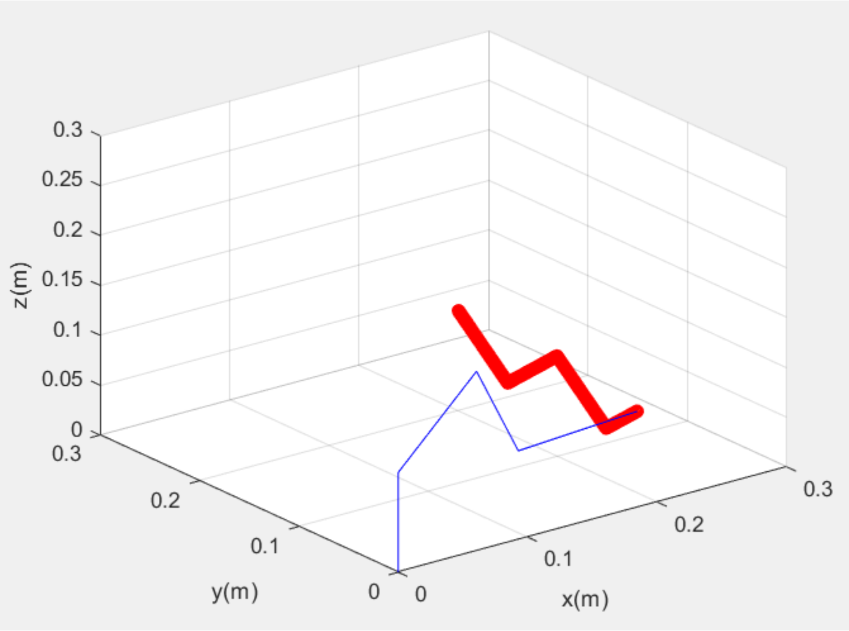


Figure 1.8: Relation of Velocity and Time

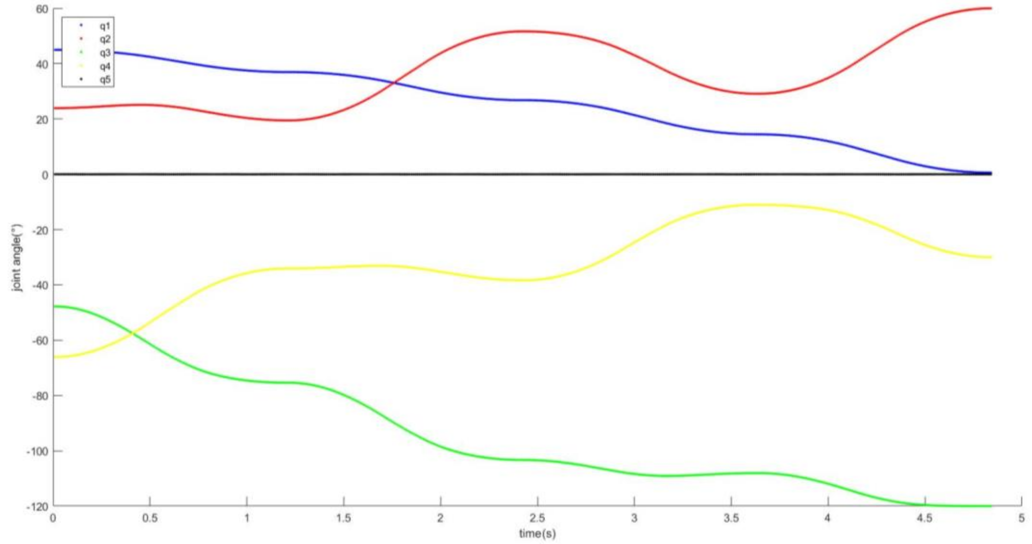


Figure 1.9: Relation of Joint Values and Time

Linear Trajectories with Obstacle Avoidance

In this part, a ball is inserted between the third point and the fourth point. To avoid the obstacle, a point above the tangent of the third and fourth points is selected to be the new via point. Then, new straight-line trajectories are created. The corresponding MATLAB file is obstacle_avoidance.m, and the screenshot of animation is shown below:

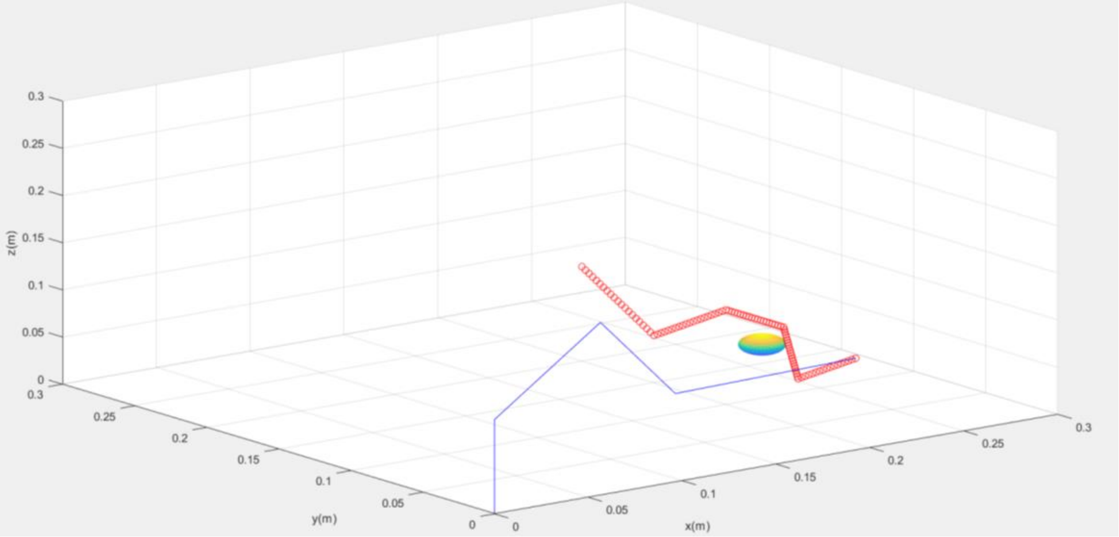


Figure 1.10: Obstacle Avoidance

Trajectories with Quintic Polynomial function of Joint Angle

Assuming every via point's velocity and acceleration are 0, and the relation of every joint's angle and time follows a quintic polynomial function. Additionally, time consuming between two via points is 1s, and trajectories can be figured. The MATLAB file is quitic_polynomial.m, and the animation is in Fig. 1.11.

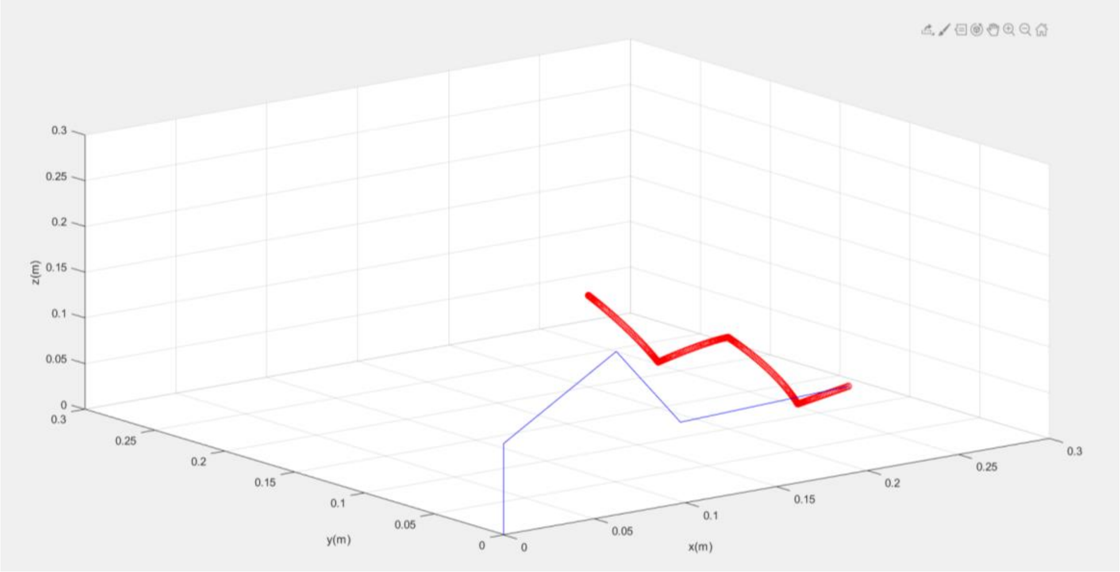


Figure1.11: Trajectories with Quintic Polynomial function of Joint Angle

PART 2

Kinematics simulation of the Parallel Robot in MATLAB

2.1 Inverse Kinematics of Parallel Robot

The figure 2.1 show the planar parallel robot kinematic model. The parallel robot is composed of a fixed base, a mobile platform, 6 passive rotating joints (PPi where i=1:3) and 3 active rotating joints (Pbi where i=1:3). Therefore, the i-th pillar of the robot is composed of three joints, respectively in the upper SA and the lower L. The corresponding matlab code can be found in the Part2a.m file. In this appendix, MATLAB is used to simulate the spatial position of the parallel robot for different coordinates and orientations. The origin of the coordinate system is the midpoint of the platform.

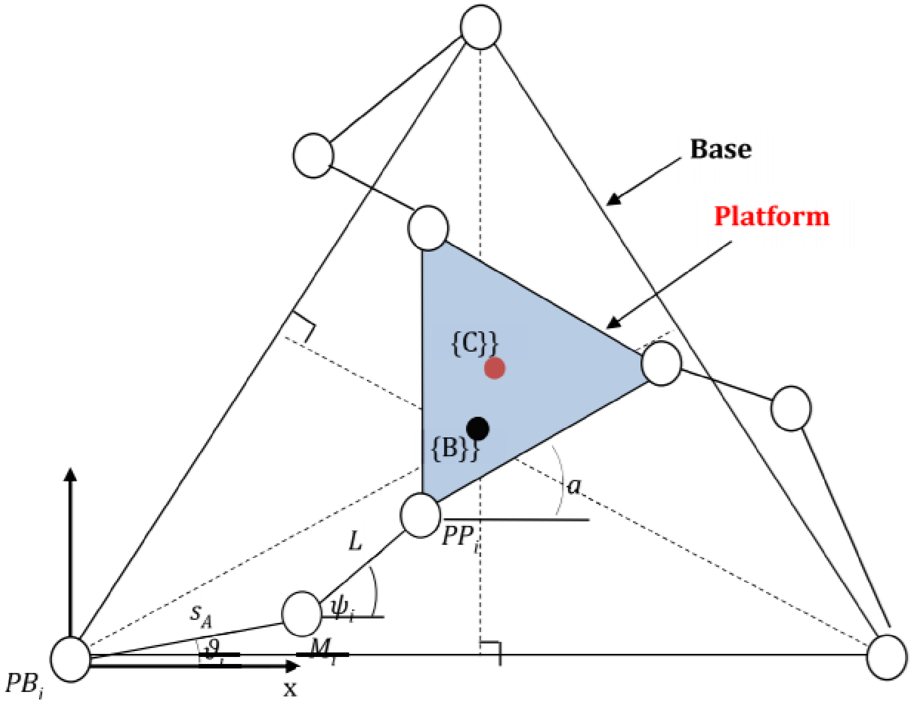


Figure 2.1: Planar Parallel Robot Kinematic Model

The inverse kinematic tasks in this section are solve, implement parallel robot inverse Kinematics and regardless of how the Cartesian input parameters are changed each leg will be adapted to the new Cartesian position.

To accomplish this part of the task, the flowchart shown in figure 2.2 gives a clear idea.

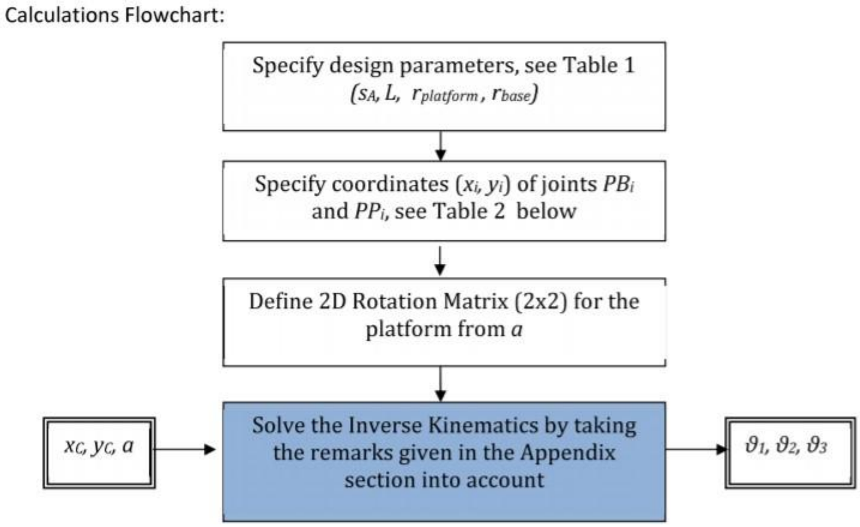

Calculations Flowchart:
Specify design parameters, see Table 1 (s_A, L, $r_{platform}$, r_{base})
Specify coordinates (x_i, y_i) of joints PB_i and PP_i, see Table 2 below
Define 2D Rotation Matrix (2x2) for the platform from a
x_C, y_C, a
Solve the Inverse Kinematics by taking the remarks given in the Appendix section into account
ϑ_1, ϑ_2, ϑ_3

Figure 2.2: The flow chart of the inverse kinematics

1.The first step is to determine the given parameters of the parallel robot (e.g., SA, L, etc.) which can be found in Table 2.1.

Table 2.1. Design parameters of planar parallel robot.

Parameter – Data Geometry	**Value**
S_A	170mm
L	130mm
$r_{platform}(joint\ circle\ radius)$	130mm
$r_{base}(joint\ circle\ radius)$	290mm

Table 2.2. Angular joint positions.

PB_1	$pi/2$
PB_2	$pi + pi/6$
PB_3	$2pi + pi/6$
PP_1	$pi/2$
PP_2	$pi + pi/6$
PP_3	$2pi + pi/6$

2.In the second step, the detailed coordinates of PPi and PBi (where i=1:3) are determined in the coordinate system using the centre point of the platform as the origin through the geometric relationship of the triangle. The angular joint positions can be seen in the table 2.2 above.

Assuming that the origin of the axes is the centre of the Base, the coordinates of PPi (where i=1:3) can be derived from the geometric relationship of the equilateral triangle. Suppose the position of the origin of the coordinates is (X_c, Y_c) and theorientation of the platform is a. The formula is shown below:

$$Xpp_i = -r_{platform} * cos((\frac{\pi}{6} + a + \frac{2\pi}{3} * (i-1)) + Xc; \text{(B.1.1)}$$

$$Ypp_i = -r_{platform} * sin((\frac{\pi}{6} + a + \frac{2\pi}{3} * (i-1)) + Yc; \text{(B.1.2)}$$

Where **i=1:3**,

$\boldsymbol{r_{platform}}$ = 130mm, it is the joint circle radius of the platform.

a is the orientation of the platfrom, here it is used as an input parameter.

$(\boldsymbol{Xc}, \boldsymbol{Yc})$ is the centre of the platform, it is also used here as an input parameter, which can be modified as required in the following.

Since the origin of the coordinate system is the centroid of Base, the coordinates of the three vertices of Base are fixed and PBi is calculated as shown below:

$$X_{PB_i} = -r_{base} * cos((\frac{7\pi}{6} + \frac{2\pi}{3} * (i-1)); \text{(B.1.3)}$$

$$Y_{PB_i} = -r_{base} * sin((\frac{7\pi}{6} + \frac{2\pi}{3} * (i-1)); \text{(B.1.4)}$$

Where i=1:3,

$\boldsymbol{r_{base}}$ = 290mm, it is the joint circle radius of the base.

Because the three vertices of base do not change depending on the input parameters, there are no deviations and no coordinates $(\boldsymbol{Xc}, \boldsymbol{Yc})$in the formula.

3. In the third step, Define the 2D rotation matrix (2x2) from a as a platform. Finally, the information given in the appendix is used to solve the inverse kinematics of the parallel robot.

In this step, the joint coordinates between PBi and PPi are determined. In this step, the distribution is calculated, firstly by calculating the angle θ between the link and the horizontal plane directly.

For leg i the coordinated for point PPi, w.r.t to frame (PB) are:

$$X_{PP2PB_i} = S_A \times \cos(\theta_i) + L \times \cos(\psi_i); \text{(B.1.5)}$$

$$Y_{PP2PB_i} = S_A \times sin(\theta_i) + L \times \sin(\psi_i);\ \text{(B.1.6)}$$

Where i=1:3,

$S_A = 170mm$ is the length between the joint M and PBi.

θ_i is the angle between the connecting rod M-PBi and the horizontal line and is the key value to be found here.

$L = 170mm$ is the length between PPi and the node M.

ψ_i is the angle between the connecting rod PPi-M and the horizontal line.

Combining the two formulas above (2.5)和(2.6), $(2.5)^2 + (2.6)^2 \xrightarrow{yields}$

$$X_{PP2PB_i}{}^2 + Y_{PP2PB_i}{}^2 - 2S_A\left(X_{PP2PB_i} + Y_{PP2PB_i}\right) + S_A{}^2 - L^2 = 0;\ \text{(B.1.7)}$$

Simplify the equation according to the hints in the RF taskbook by bringing in the algebra e_i (where i=1:3).

$$e_1 = -2Y_{PP2PB_i}S_A;\ \text{(B.1.8)}$$

$$e_2 = -2X_{PP2PB_i}S_A;\ \text{(B.1.9)}$$

$$e_3 = \left(X_{PP2PB_i} + Y_{PP2PB_i}\right) + S_A{}^2 - L^2;\ \text{(B.2.0)}$$

B.1.7 after bringing in e_i can be simplified to the following equation:

$$e_1 sin(\theta_i) + e_2 \cos(\theta_i) + e_3 = 0;\ \text{(B.2.1)}$$

According to the hint in the handbook of RF. The parameter t can be defined as $t = \tan(\frac{\theta_i}{2})$, then:

$$\sin(\theta_i) = \frac{2t}{1+t^2};\ \text{(B.2.2)}$$

$$\cos(\theta_i) = \frac{1-t^2}{1+t^2};\ \text{(B.2.3)}$$

Using (B.2.2) and (B.2.3) in equation (B.2.1), the following equation can be obtained:

$$(e_3 - e_2)t^2 + 2e_1 t + e_2 + e_3 = 0;\ \text{(B.2.4)}$$

The general formula for solving quadratic equations gives us the value of t, and furthermore the value of θ_i:

$$t_{1,2} = \frac{-e_1 \pm \sqrt{e_1{}^2 + e_2{}^2 - e_3{}^2}}{e_3 - e_2};\ \text{(B.2.5)}$$

$$\theta_i = 2\,\text{atan}\left(t_{1,2}\right);\ \text{(B.2.6)}$$

The value of θ_i calculated by the above formula can be used to calculate the coordinates of the joint M by the following formula:

$$X_{M_i} = -X_{PB_i} + S_A \times \cos(\theta_i)\,; \text{(B.2.7)}$$

$$Y_{M_i} = -X_{PB_i} + S_A \times \cos(\theta_i)\,; \text{(B.2.8)}$$

All the values for the MATLAB simulation have now been calculated and the MATLAB code is attached. The following article will show the position state of the parallel robot working plane for a given orientation a.

The changing of the Cartesian input parameters should cause each leg to adapt to the new Cartesian position. And the results can be seen in figures 2.3, 2.4 and 2.5.

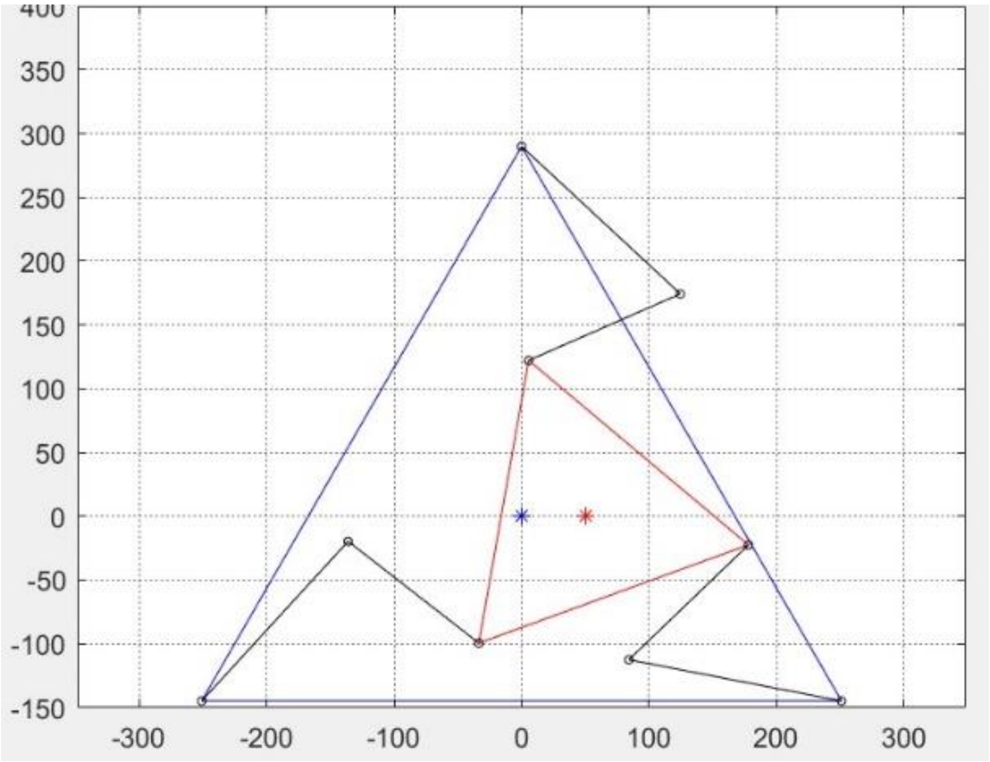


Figure 2.3: $(X_c, Y_c, \alpha) = (20, 50, 0)$

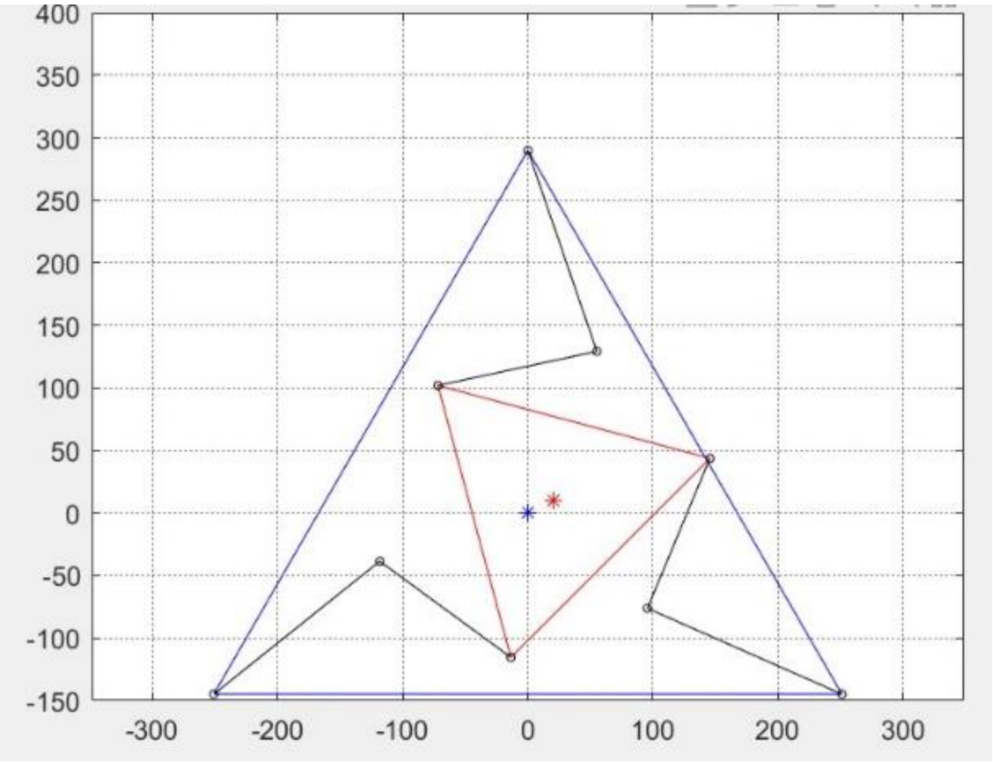


Figure 2.4: $(X_c, Y_c, \alpha) = (45, 20, 10)$

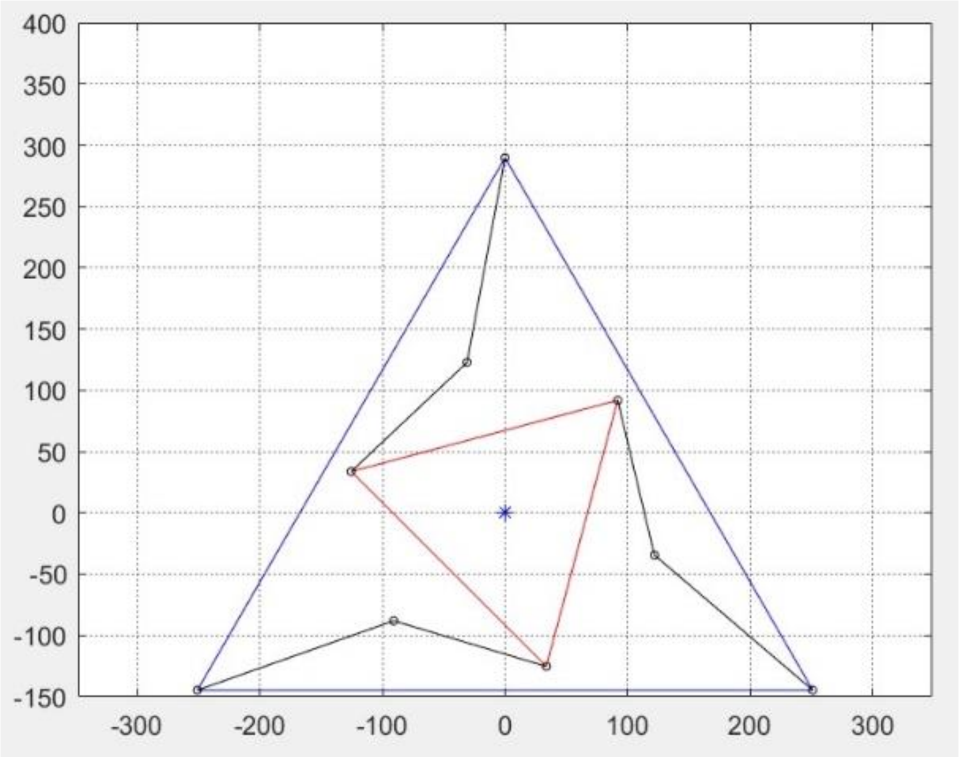


Figure 2.5: $(X_c, Y_c, \alpha) = (75, 0, 0)$

2.2 Parallel robots' workspace for a given orientation a.

The idea is to calculate all possible locations of the centre point of each possible parallel robot working plane, given the orientation a. The most important point is that the presence of θ_i is also guaranteed when estimating the location of each point. If a possible centre point is calculated that does not have a true θ_i, then this centre point needs to be discarded.

Using MATLAB to plot the parallel robots' workspace for a given orientation a. The result is as follow:

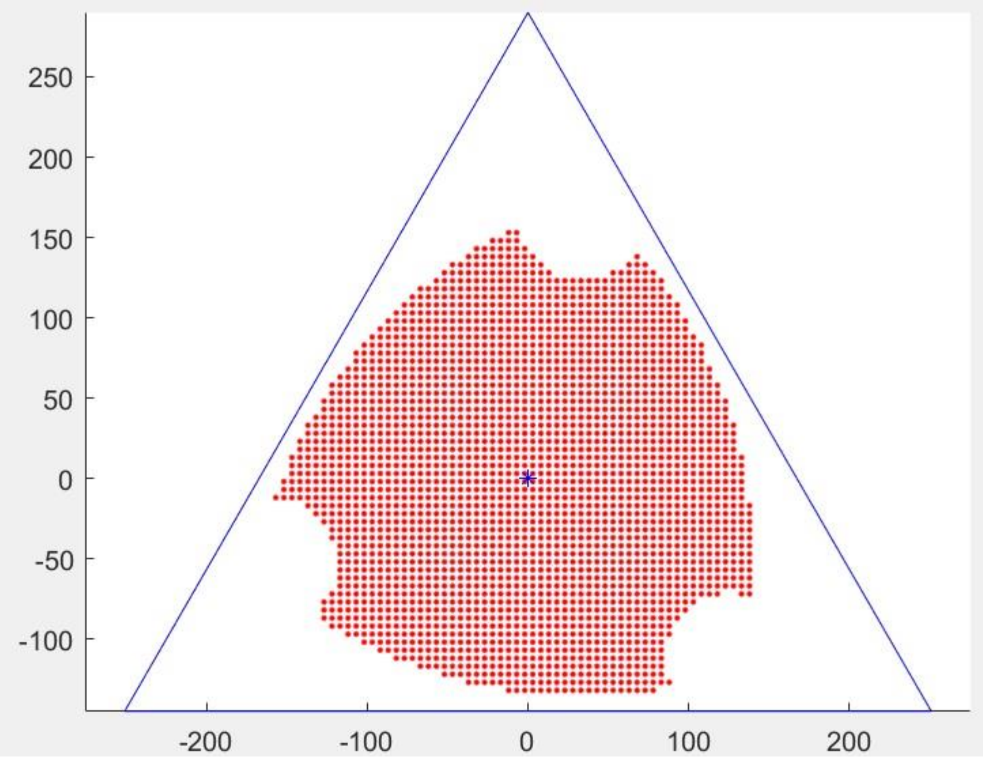


Figure 2.6: the given orientation is 15

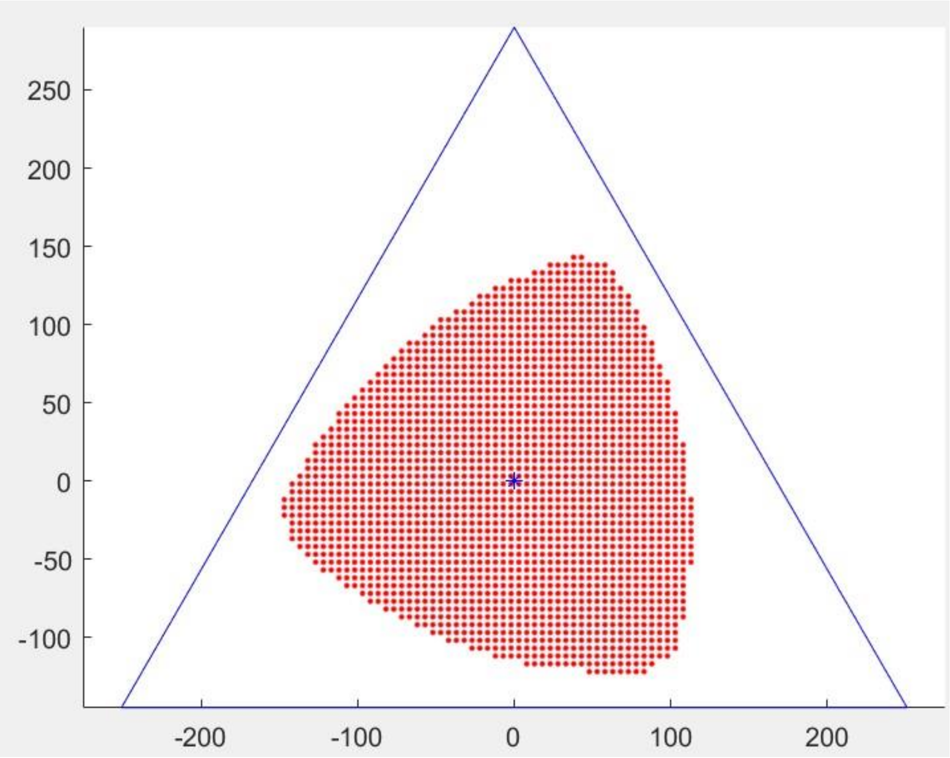


Figure 2.7: the given orientation is 30

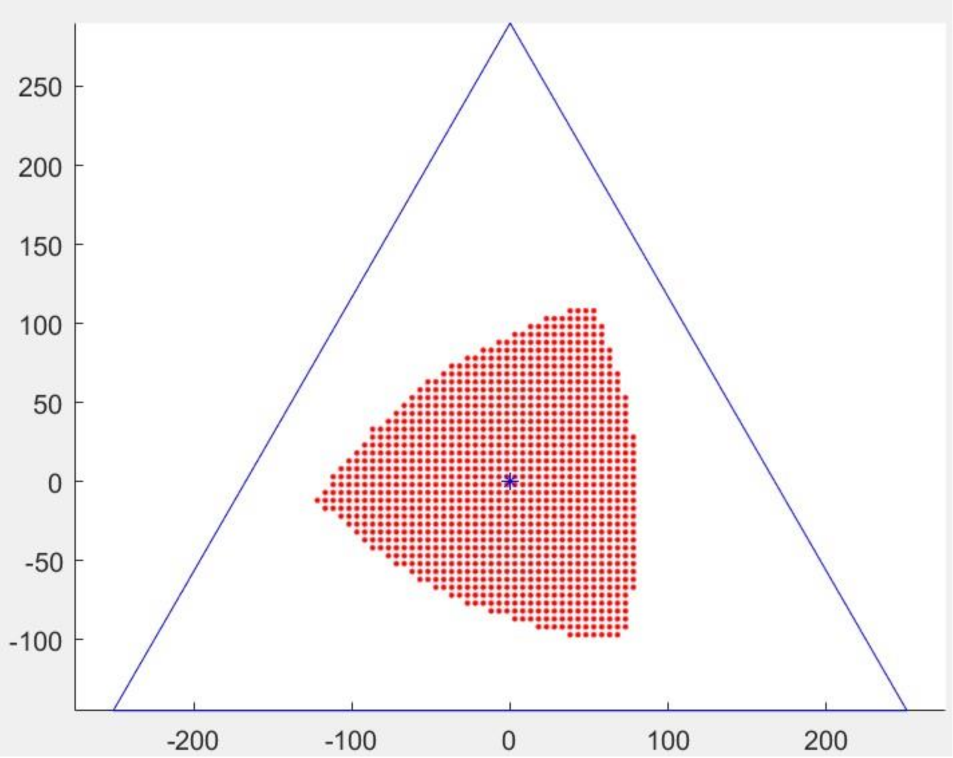


Figure 2.7: the given orientation is 45

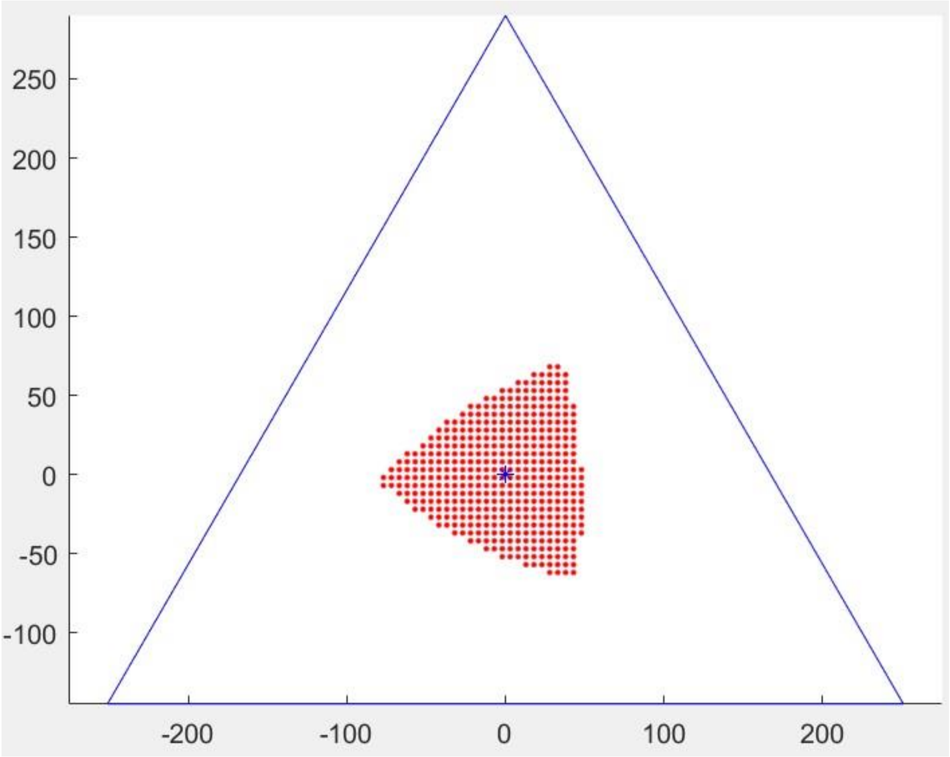


Figure 2.7: the given orientation is 60

PART 3

3.1 Introduction— (PRM)

3.1.1 The meaning of path planning

Motion planning involves trajectory planning and path planning. Path is a sequence of points or curves connecting a beginning position to an ending position. The path planning is the strategy for forming such paths. (LaValle, 2006) Additionally, depending on the environment in which path planning needs to be applied (different information features). The Part3 investigates the application of robotic arms in industrial environments, defined as path planning problems in the discrete domain.

3.1.2 The significance of path planning

In China, the ageing of society is inevitably happening. It is predictable that in the future the Chinese workforce will be decreased, so the increase in average labour productivity will need to be done by robots. This will be effective in preventing a slower GDP growth rate (Zhao, 2019). Traditional robots are mainly used in the car industry, however, because the car industry is characterised by the production of vehicles that can be used for many years and the high profits per vehicle. So, it seems that the car industry is now saturated with robots. Therefore, many engineers in industry are focusing their interest on the 3C industry.

3.1.3 Types of path planning

There are four main types of algorithms for path planning: traditional, graphical, intelligent bionic and others. (Zhang, et al., 2011)

a) Traditional algorithms

b) Graphical Alogrithms

c) Intelligent bionic algorithms

d) Other algorithms

3.2 The principle of the PRM algorithm

Probabilistic Road Maps (PRM), the method used in Part 3 of this report is a graph-based search method in which PRM converts a continuous space into a discrete space and then uses search algorithms such as A* to search for paths on the roadmap to optimise search efficiency. (Kavraki, et al., 1996). In this experiment, Dijkstra was used to search for paths on the roadmap. This method provides a solution with a relatively small number of randomly sampled points, which is sufficient to cover most of the available space for most problems, and the probability of finding a path is 1.

In Part 3, the PRM is designed to solve the problem by randomly sampling the non-colliding points, and then connecting the two points using a simple local planner such as the Walk to algorithm, with the start and end points connected to the roadmap using graph search.3.3 Evaluation of the PRM algorithm

The code for the Part 3 section can be viewed in the PRM.m file.

3.3 Experiment

Figure 3.1 shows the map used in this experiment. The green dots represent the starting position of the robot arm, the yellow dots represent the target position of the robot arm. The black areas represent obstacles that may be present in the real workshop.

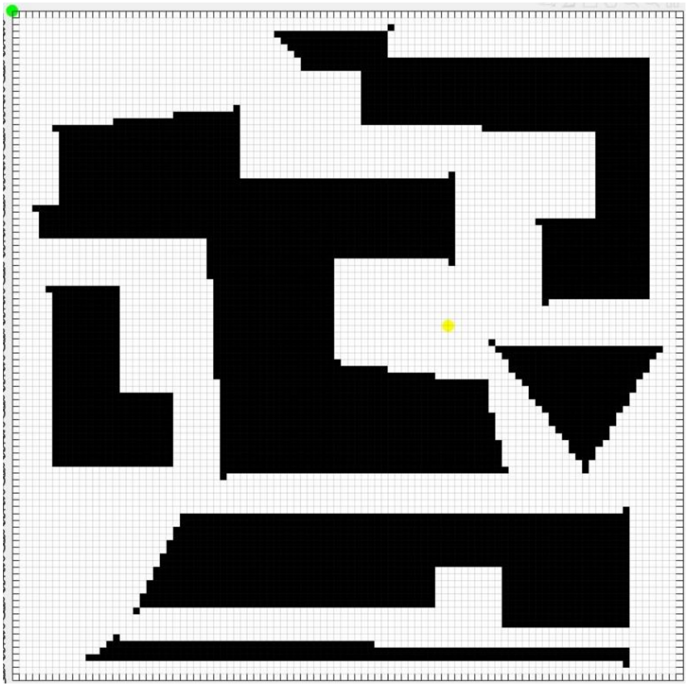


Figure 3.1: Map of PRM algorithm applications

There were two parameters that affect this experiment, namely the number of sampling points and the step size between sampling points. Figures 3.2, 3.3 and 3.4 represent the real change in the PRM algorithm at different sampling points and step sizes.

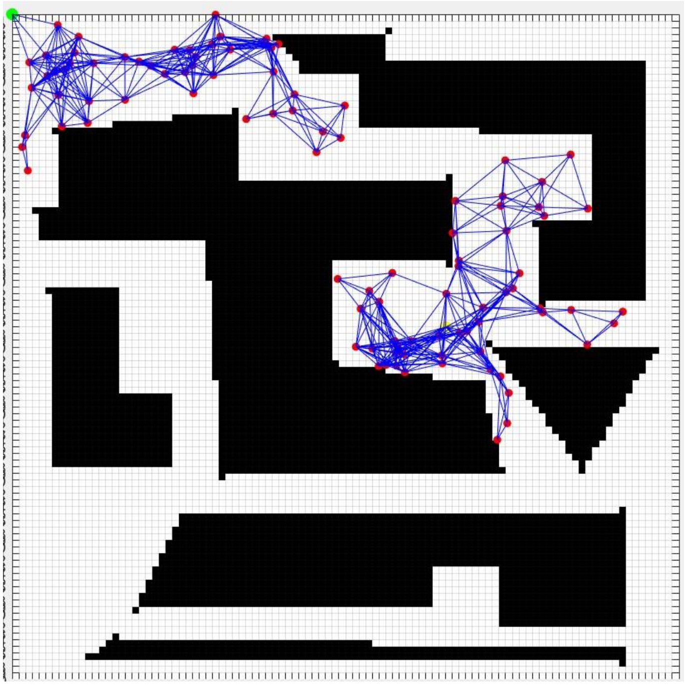


Figure 3.2: sampling_points = 100, step_size = 10

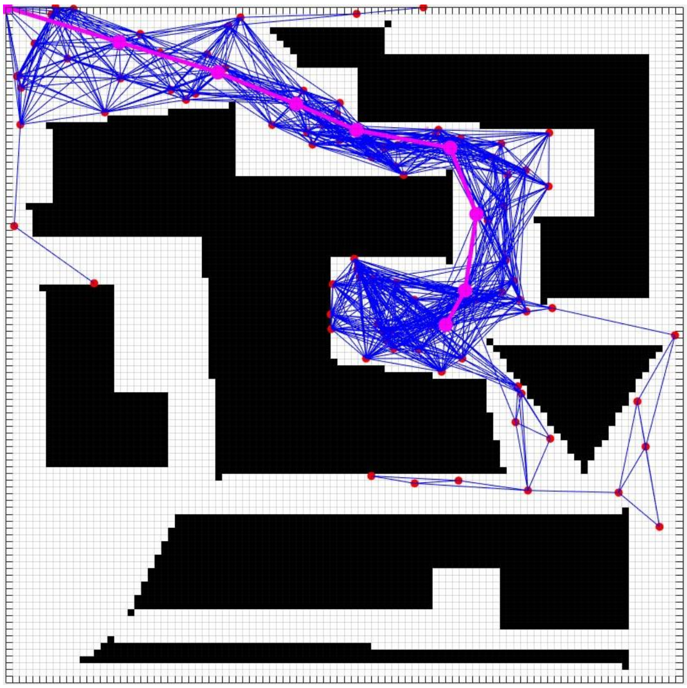


Figure 3.2: sampling_points = 100, step_size = 20

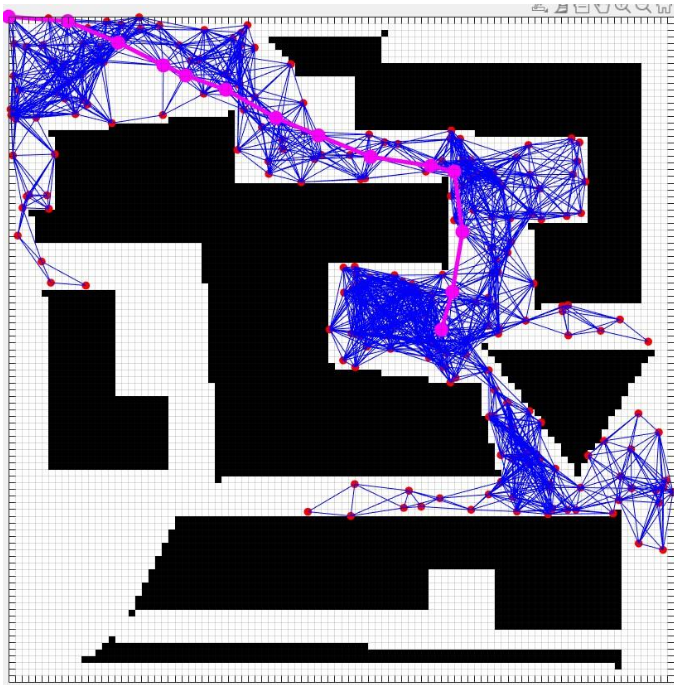


Figure 3.2: sampling_points = 200, step_size = 10

It is obvious from the three graphs above that when the step sizes are the same, if there are not enough sampling points, the path planning will be failed. When the sampling points are the same, the larger the step length, the more dispersed the layout of the sampling points becomes, so that only a relatively small number of sampling points are needed to complete the path planning. Moreover, the more points are sampled, the more time it takes for the computer to calculate the exact path.

Conlcusion

This report consists of three parts.

The first part solves the trajectory planning problem of the robot arm.

Secondly, a parallel robot is a combination of several tandem robots with the same end-effector. The inverse kinematics of a parallel robot can therefore be broken down into several simple inverse kinematics. When plotting the workspace of a parallel robot, there are also conditions that determine whether the point is in the workspace or not. For example, we can set whether the length of the second arm is 130. if so, then the point is in the workspace and conversely.

Thirdly, it is obvious from theory and experiment that PRM algorithm has the same drawbacks as other sampling planning algorithms. The PRM algorithm is incomplete when the number of sampling points is too small or badly distributed, although it is possible to achieve completeness as the number of points used increases. Thus, in the later stages of the study, completeness will be studied. The result of the learning will be the path solution between the starting point and the target point, which is expected to be obtained even when the number of points sampled is low.

References

Kavraki, L., Svestka, P., Latombe, J. and Overmars, M., (1996). "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". *IEEE Transactions on Robotics and Automation,* 12(4), pp. 556-580.

LaValle, S., (2006). *Planning Algorithms.* 1 ed. s.l.:Cambridge Univeristy Press.
Zhang, G. et al., 2011. Summary of Path Planning Algorithm and its Application. *Modern machinery,* Issue 05, pp. 85-90.

Paul, Richard (1981). *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. *MIT Press, Cambridge, Massachusetts. ISBN 978-0-262-16082-7.*

Zhao, D., 2019. Analysis of the development of China's ageing population. *Financial News,* 000(012), pp. 9-10.