ROBOTIC PATH PLANNING

Project report submitted in

Partial fulfilment of requirement for the award of the degree of

Bachelor of Technology

in

Artificial Intelligence

by

Om Barde

Monali Patle

Aditya Bhagwat

Khushal Shinde

Under the guidance of

Prof. Rahul Suryawanshi

Assistant Professor

AI Department

Department of Artificial Intelligence

G H Raisoni Institute of Engineering and Technology, Nagpur

(An Autonomous Institute Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur)

Accredited by NAAC with A+ Grade

2022-23

ROBOTIC PATH PLANNING

Project report submitted in

Partial fulfilment of requirement for the award of the degree of

Bachelor of Technology

in

Artificial Intelligence

by

Om Barde

Monali Patle

Aditya Bhagwat

Khushal Shinde

Under the guidance of

Prof. Rahul Suryawanshi

Assistant Professor

AI Department

Department of Artificial Intelligence

G H Raisoni Institute of Engineering and Technology, Nagpur

(An Autonomous Institute Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur)

Accredited by NAAC with A+ Grade

2022-23

G H Raisoni Institute of Engineering and Technology, Nagpur

(An Autonomous Institute)

Department of Artificial Intelligence

Certificate

The report of project titled **Robotic Path Planning** submitted by Aditya Bhagwat, Khushal Shinde, Monali Patle, Om Barde in the partial fulfilment of the degree of Bachelor of Technology in **Artificial Intelligence** during academic year 2022-23, has been carried out under our supervision at the Department of Artificial Intelligence of G H Raisoni Institute of Engineering and Technology, Nagpur. The work is comprehensive, complete and fit for evaluation.

Prof. Rahul Suryawanshi
Asst. Prof. AI Department
(Guide)

Prof. Shweta Bhelonde
Asst. Prof. AI Department
(Project - Coordinator)

Dr. Smita Nirkhi
HOD, AI Department

Dr. Vivek Kapur
Director, GHRIETN

G H Raisoni Institute of Engineering and Technology, Nagpur

(An Autonomous Institute)

Department of Artificial Intelligence

Declaration

We certify that

a. The work contained in this project has been done by us under the guidance of our supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. We have followed the guidelines provided by the Institute in preparing the project report.

d. We have confirmed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever We have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

(Om Barde) (Monali Patle)

(Aditya Bhagwat) *(Khushal Shinde)*

G H Raisoni Institute of Engineering and Technology, Nagpur

(An Autonomous Institute)

Department of Artificial Intelligence

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help from many individuals and organizations. We would like to extend my sincere thanks to all of them.

Our deepest thanks to **Prof. Rahul Suryawanshi** lecturer from Artificial Intelligence department In G. H. Raisoni Institute of Engineering and Technology, Nagpur for guiding, supervising, providing us with necessary information about the project and correcting various prototypes and documents of us with attention and care. He has gone with us through the project and without his help we were not able to complete our project.

We also would like to express our gratitude towards our college and all our professors in the college who have given us a great chance to undergo this project and also provide us with knowledge to complete it.

Our thanks and appreciations also go to our colleagues in developing this project and people who have willingly helped us out with their abilities.

Name of Projectees

1. Om Barde
2. Monali Patle
3. Aditya Bhagwat
4. Khushal Shende

TABLE OF CONTENT

ABSTRACT

Currently, the path planning problem is one of the most researched topics in autonomous robotics. That is why finding a safe path in a cluttered environment for a robot is an important requirement for the success of any such robot project. Here we developed an algorithm based on free segments and a turning point strategy for solving the problem of robot path planning in a static environment is presented. The aim of the turning point approach is to search a safe path for the robot, to make the robot moving from a starting position to a destination position without hitting obstacles.

CHAPTER 1: INTRODUCTION

1.1 Robot Locomotion

Robots is either mobile or stationary. Robots embrace rolling robots, creep robots, swimming robots and lots of a lot of. Stationary mechanisms embrace robot arm, robot face, industrial robots etc. though called stationary, these robots don't seem to be truly unmoving, however square measure confined to a tiny low boundary. every of those robot’s square measure designed to figure on totally different platforms and also the most typical one’s work either ashore, Air, Water, space etc. a number of the robots square measure designed to figure on quite one platform and may shift from land to water to air. supported the manner robots move, they will be more classified as "Holonomic" or "Non-Holonomic" drive Robots[3].

2D Maps are extensively employed in the trail designing field, as they will be simply enforced by the image files; many algorithms were developed and tested in several environments. Here, we have a tendency to describe the 3 reference maps accustomed check the planned path designing algorithms. during this explicit reference map[4], the obstacles are mounted in nature. it's a path or assortment of methods, generally from a supply purpose to a goal purpose that has walls because the obstacles. There may well be some dead-end; therefore, plenty of backtracking ability of the rule is needed to seek out a path from the supply to the goal node in path designing[7]. Then it's more taken as input to implement A*, PRM, RRT, and RRT sleek Algorithms. The sample image is conferred within the result section, wherever the planned rule is tested with four algorithms in association with snake algorithms. The second form of map chosen consists of random obstacles; four forms of obstacles (circle, square, triangle, and termini shapes) are chosen. this sort of map is beneficial in representing a posh town that consists of 3D buildings and obstacles which require to be scanned for sleek path traversing [9].

This setting could comprise either disjoint obstacles or overlapping obstacles from the on top of list of obstacles. Then the automaton needs to calculate the optimum path within the structured setting[2]. The developed setting consisting of the obstacle from the list is then taken as input for implementing A*, PRM, RRT, and RRT sleek Algorithms. Usually, this sort of image is beneficial for drones or 2/3D hemisphere obstacles that represent measuring instrument sites. This setting consists of pictures that are densely inhabited that contain colour pixels throughout the image. this sort of setting could embrace an image captured by a camera or by a device to find the optimum path from the supply to the destination node. Here, an image of the NIT Raipur field has been taken to account for experimentation[5]. once the implementation of a snake rule, the output offers a transparent specific plan of free house and also the space occupied by the obstacles.

Path Length is one of the most important parameters of path planning. It is the summation of variable lv for all the coordinate value pair (x, y) in the final optimal path obtained. The path length is calculated between the source and destination[10]. The algorithm traverses completely the free space based on which it derives the shortest path, which is considered for robots to move from source to destination. The distance of the path is calculated with the help of Euclidian Distance. The distance between two locations in one dimension is defined as the absolute value of the difference between their coordinates in the Euclidean Distance system. In mathematical notation, this is written as $|p1-q1|$ where p1, q1 are the first and second coordinates, respectively. This difference is expressed as an absolute value as the distance is typically thought of as having only positive values [8].

The number of Moves is another important parameter when a robot needs to perform in some uneven planes where the number of moves becomes important to count, it is calculated by the angle between the x-coordinate and the hypotenuse of the final optimal path. This is calculated pixel by pixel, and thus, if any change is detected between the current and previous slope, then the counter is incremented by one [10]. The function Math.arccosine() is used to calculate and count the number of moves taken by the algorithm, and the function Math.arccosine() returns the arc cosine (in radians) of a number. The Math.arccosine() method returns a numeric value between 0 and π radians for x between -1 and 1. Computation time is the third parameter taken to record the results of the proposed algorithm, and it is also one of the essential parameters to be observed where the task which is to be accomplished is time-bound, it is the total times consumed by the algorithms to produce output, and can be calculated as the difference between the start time and exit time[10].

Total Time Taken = Current Time – Start Time

1.2 Holonomic and Non-Holonomic Drive

1.2.1 Holonomic Drive

Holonomic refers to the link between governable and total degrees of freedom of a golem. If the governable degree of freedom is capable total degrees of freedom, then the golem is claimed to be Holonomic. A golem engineered on castor wheels or Omni-wheels may be an exemplar of Holonomic drive because it will freely move in any direction and therefore the governable degrees of freedom is capable total degrees of freedom. The image shows a castor wheel which might rotate in each coordinate axis and coordinate axis creating it move in each the directions[6].

1.2.2 Non-Holonomic Drive

If the governable degree of freedom is a smaller amount than the whole degrees of freedom, then it's called non-Holonomic drive. An automobile has 3 degrees of freedom; i.e., its position in 2 axes and its orientation. However, there are a unit solely 2 governable degrees of freedom that area unit acceleration (or braking) and turning angle of handwheel[6]. This makes it tough for the motive force to show the automobile in any direction (unless the automobile skids or slides).

1.2.3 Redundant Drive

What if the governable degrees of freedom square measure quite the whole degrees of freedom? Then the controls square measure thought of to be redundant. An automaton arm or perhaps an individual's arm has solely six degrees of freedom, however seven governable degrees of freedom[6]. (Try twisting and rotating your arm and determine what square measure the seven degrees of freedom, as well as shoulder, elbow and wrist).

1.3 Problem statement

Many algorithms associated with path designing were developed and tested in several environments. The live of Performances becomes sophisticated because of playacting in several pictures, so first, the matter is to possess the right illustration of the appliance situation and may follow some standards. every application has totally different operational environments with constraints on the platforms. it's needed to possess reference take a look at maps to exactly compare the performance with one another. Thesis deals with the second situation solely, and for that, it's taken 3 totally different reference maps for testing the trail designing algorithms. Once the quality maps square measure designated, subsequent downside is to urge associate degree optimized obstacle-free path. The performance of path designing algorithms is greatly suffering from the obstacle, thus, poignant the ends up in many ways. Most of path designing algorithmic rule suggests a special thanks to alter the obstacles whereas traversing for locating ways. They largely take issue in their traversing mechanism and take alternative ways to derive ways. It proposes the employment of the Snake algorithmic rule for detective work obstacles before with its complete periphery; it splits the space into 2 components, free house and sophisticated house, wherever free house is that the house that's to be traversed by the trail designing algorithms and sophisticated house is that the house to be voided by the algorithms. Snake algorithmic rule helps to cut back the traversing house for path designing algorithms. Thus, it helps to boost performance.

1.4 Objectives and scope of study

First of all, we understand cost as the metric that the robot accumulates by moving. The objective of the path planner is to minimize this accumulation by producing the optimal path. The cost in question can be uniform, in the sense that the regions that can be accessed by the robot always have the same value. This approach can be used for collision-avoidance path planning, in which metrics such as the path length in a 2D plane are minimized. Non-uniform cost maps can be used to assign different values of cost to different accessible areas. It can be useful to, for example, define the energetic performance of the robot at each location. Moreover, the cost can be also defined according to a direction vector. This means that the robot will experience different values of cost depending on its heading. In this case, the cost is categorized as anisotropic [31], whereas in the contrary case the cost is isotropic. Furthermore, the steering manoeuvre of the robot can also have different values of cost according to its locomotion. Finally, it is worth noting that the environment can be fully known, partially known or even fully unknown, requiring for the latter two a planning strategy that is capable of replanning when this knowledge is updated.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

With the event of science and industrial automation, golem technology has been greatly developed in recent decades, and step by step applied in military, aerospace, industry, medical, service, and alternative fields [1,2]. Single-arm industrial robots have achieved notable development and application in China, wide exchange manual casting, welding, palletizing, and alternative operations [3,4]. However, several advanced operational tasks need collaboration between the robotic arms. The dual-arm golem includes a larger operating area, stronger load capability, and obvious blessings in work and assembly eventualities [5,6]. However, in contrast to a straightforward combination of 2 single-arm robots, a dual-arm golem has some overlap in its space. the trail coming up with of 2 arms shouldn't solely think about static obstacles in area, however additionally think about the interference between the 2 arms. within the field of dual-arm AI, a way to understand obstacle dodging motion coming up with is often a stock [5,6]. Path coming up with is to maneuver swimmingly from one purpose to a different with none hitch and hindrance. it's one in every of the prime aspects for robots to search out the shortest path or otherwise best path. The best path in AI might be the trail that reduces the computation time, variety of turnings[6], and therefore the distance between the supply and destination. Assume a case wherever a golem is positioned in a very sq. space, the 2 endpoints of the diagonal of that sq. space square measure the supply and destination’s purpose for the golem[7]. it's currently needed to travel from supply purpose to destination, there square measure multiple things placed between these 2 points, and to achieve the destination the golem should avoid those objects placed in its path to avoid any clash[8].

The first step in doing path coming up with for robots needs Associate in Nursing surroundings set-up or a map of the environment. These surroundings are termed the space or configuration area for robots. when finalizing the surroundings, the golem is then positioned on the map and is assumed to bear in mind of its location on the map[8,9]. It, thus, localizes itself and is capable of avoiding the obstacle drawing near its method. choice of map illustration is additionally one in every of the essential tasks because it ought to match the acceptable application[9]. an extra factor that has to listen to is that the golem ought to be a point-sized golem, because it wouldn't be possible each time to check algorithms on physical robots. therefore, before truly absorbing the algorithms, these 2 things ought to be obvious, i.e., map illustration resembling the applying space and point-sized robots. In Map illustration, 1st of all, it's needed to delineate the surroundings within the laptop. This surroundings constitutes the configuration area for robots. There square measure 2 approaches to try and do that, distinct and continuous approximations. The map taken for testing is split into equal components or in several sizes (grids) within the distinct approximation technique [3].

Path Planning

After fixing the surroundings for the trail designing formula to check, successive step to pursue the target is to relishes the idea of path designing. Understanding the idea of path designing needs obtaining aware of some basic notions associated with path designing. the primary one is that the configuration space/workspace [4], which implies the platform or the environmental set-up wherever the experiments got to be performed, generally it's going to either be 2D Environments. To represent 2nd, we have a tendency to need 3 parameters, (x, y, θ), whereas, within the 3D surroundings, it needs six parameters, 3 for translation and 3 for outlining mathematician angles to explain the configuration area[4,6]. during this article, we've got taken 3 totally different situations supported a 2nd surroundings, specifically Maze, random obstacles, and dense case situations to perform our experiment. successive vital notion is that the sort of mechanism thought-about for work, and it's going to be a point/zero-sized mechanism, 2nd form mechanism, and in some cases, 3D also [6]. Here during this article, we've got taken point-sized robots for our experimentation. successive most vital notion is that the totally different components of the space. Here space suggests that the entire environmental set-up wherever the experiment is dole out. It includes free area, which implies the area wherever traversal is feasible and most popular. complicated area is that the space that has to be discarded and skipped as this is often the area occupied by the obstacle gift within the space. Moreover, within the last, the choice of 2 points/coordinates between the trail is to be found. they're known as the supply purpose and destination purpose [4].

RRT could be a quick search formula projected by LaValle [9]. However, it's a random search formula, and also the search path might not be within the direction of the target, so the convergence speed is comparatively slow. so as to enhance the target orientation of the RRT formula, Chris Urmson et al. [10] projected a P likelihood RRT formula supported goal bias strategy. Li et al. [8] projected a variable step size trunk quick exploration random tree (VT-RRT) formula. By reworking the search area of random nodes within the RRT formula and adaptively adjusting the step size in step with the target position, the search potency is effectively improved, and also the path designing time is reduced. Jiang Hong et al. [10] projected a replacement node enlargement methodology that's biased towards the target purpose. the tactic combines the target purpose gravity, obstacle repulsion, and random purpose gravity, associate degreed adds an adaptative operate associated with the obstacle distance. A pruning optimisation methodology is projected to optimize the trail length. Lei Shao et al. [3] used associate degree pismire colony formula to optimize RRT. Experiments show that the mixture of associate degree pismire colony formula and RRT formula effectively reduces the quantity of nodes and also the average computing time. Kun Wei dynasty [7] projected a dynamic path designing methodology for mechanism autonomous obstacle shunning supported associate degree improved RRT formula, specifically sleek RRT (S-RRT). This methodology takes the directional node because the goal, that greatly improves the sampling speed and potency of RRT.

In the field of path designing for dual-arm robots, Andreas [9] projected a dual-arm path designing methodology supported cyclic mechanics to fulfil the motion constraints of the manipulator. Kim [6,7] projected a dimension reduction RRT methodology, that reduced the dimension of high-dimensional path designing area in step with the task necessities of the dual-arm mechanism. to make sure that the RRT formula has higher potency, Li rule [3,5] projected a cooperative path designing methodology for a dual-arm mechanism supported gravity adaptative step length RRT. The simulation results show that the gravity adaptative step size RRT methodology will effectively constrain the step size within the space to make sure the effectiveness of the collision detection formula [10].

CHAPTER 3: METHODOLOGY

3.1 Research Methodology

This idea of research we got from the Robotic Path Decision where the user wants to predict the data using RRT.

3.1.1 Model Building

According to the DH modelling method, The joint diagram of the main manipulator is shown in Figure 1. Using the parameters of the robotic arm and DH method, we obtain the DH parameters in Table 1. The joint diagram of the slave manipulator, and the joint diagram of the main manipulator, are mirror-symmetric with the robot body. For example, the DH parameters α and θ of the master manipulator are opposite to those of the slave manipulator.

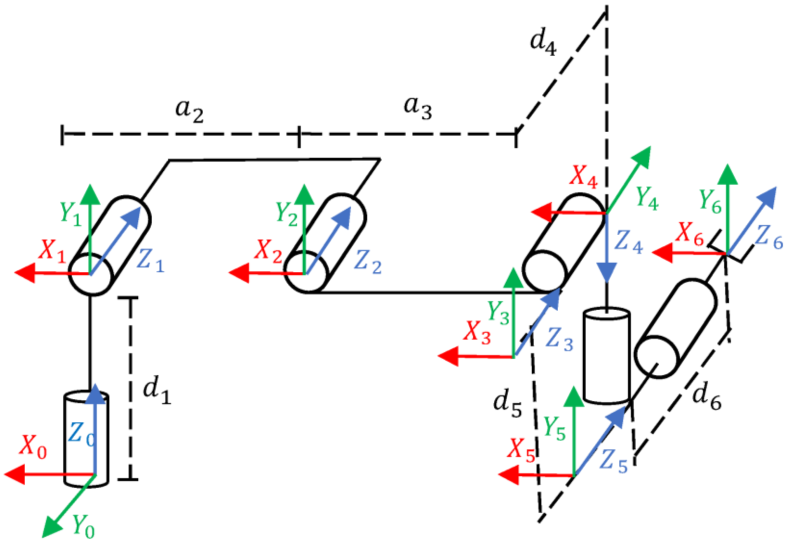


Figure 1: *Coordinate system of the main manipulator joint.*

Table 1. DH parameters of the main robot arm.

Joint Number	α	a	θ	d
1	90	0	θ_1	89.2
2	0	−425	θ_2	0
3	0	−392	θ_3	0
4	90	0	θ_4	109.3
5	−90	0	θ_5	94.75
6	0	0	θ_6	82.5

The homogeneous coordinate transformation matrix of a six-axis manipulator with this parameter is shown in Equation (1).

$$^{i-1}_{i}T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & \alpha_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) & -\cos(\theta_i)\sin(\alpha_i) & \alpha_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation matrix of all adjacent coordinate systems can be obtained by substituting DH parameters. The transformation matrix relative to the base coordinate system is shown in Equation (2).

$$T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T$$

Finally, the dual-arm manipulator needs to be unified into the base coordinate system using Equation (3).

$$S = S_0T$$

3.1.2. Path Planning Method

This proposes a hybrid way to optimize the problem of robot path planning. To do so here, we propose the concept of snake algorithm along with the traditional path planning algorithm. As discussed in the previous section, the working of the snake function helps in finding the contours present in the workspace which reduces the effort applied by the path planning algorithms to reach the robot from its source position to the destination. Here, we have considered three different path planning algorithms along with the snake algorithm to find the solution. We have considered three algorithms namely A*, PRM, RRT & RRT Smooth. The reason to pick this algorithm among all the available algorithms for path planning is that these three algorithms differ from each other entire in terms of their working mechanism. The working mechanism here refers to the working of algorithms while finding the best route for the robot, or it can also be defined as the traversing behaviour of these three algorithms that are entirely different from each other. The purpose of picking algorithms whose traversing behaviour should differ is that when it is combined with the snake algorithm, it will validate the concept in multiple aspects. The working of A* is based on the concept of selecting the next best node based on the heuristic function, whereas PRM is quite fast as compared to A*. The working of PRM mainly focuses on selecting the random nodes out of available ones in the workspace, whereas the working of RRT is based on creating a parent-child node tree and it keeps on creating the tree till it reaches the destination. We also have considered the advanced version of RRT. Finally, we have combined all these algorithms to form a hybrid combination with the snake algorithm to test the proposed hypothesis. Finally, the result obtained was very encouraging and is presented in the later section of the manuscript.

3.2. Path Planning Algorithms for Robotic Systems

Basic RRT

RRT constructs a random tree through random sampling. In the space with obstacles, the algorithm starts to explore from the initial node, X_{init}. This process is shown in Figure 2. Firstly, the random point, X_{rand}, is generated. In all the added path nodes of the tree $[X_{init}, X_1, X_2, X_3, X_{nearest}]$ the node $X_{nearest}$, which is nearest to X_{rand},is selected. Taking $X_{nearest}$as the root node, a fixed step R is added to the direction of $X_{rand}$to obtain X_{new} is added to the random tree, T. Alternatively, if there is a collision, X_{rand} will be regenerated. Repeat the whole process until the distance between the latest node and X_{goal} is less than the step value, and there are no obstacles between them; at this point it is considered that the algorithm has converged. When the planning algorithm converges, we start from X_{goal} and trace back along its parent node to find an effective path between the start point and the end point.

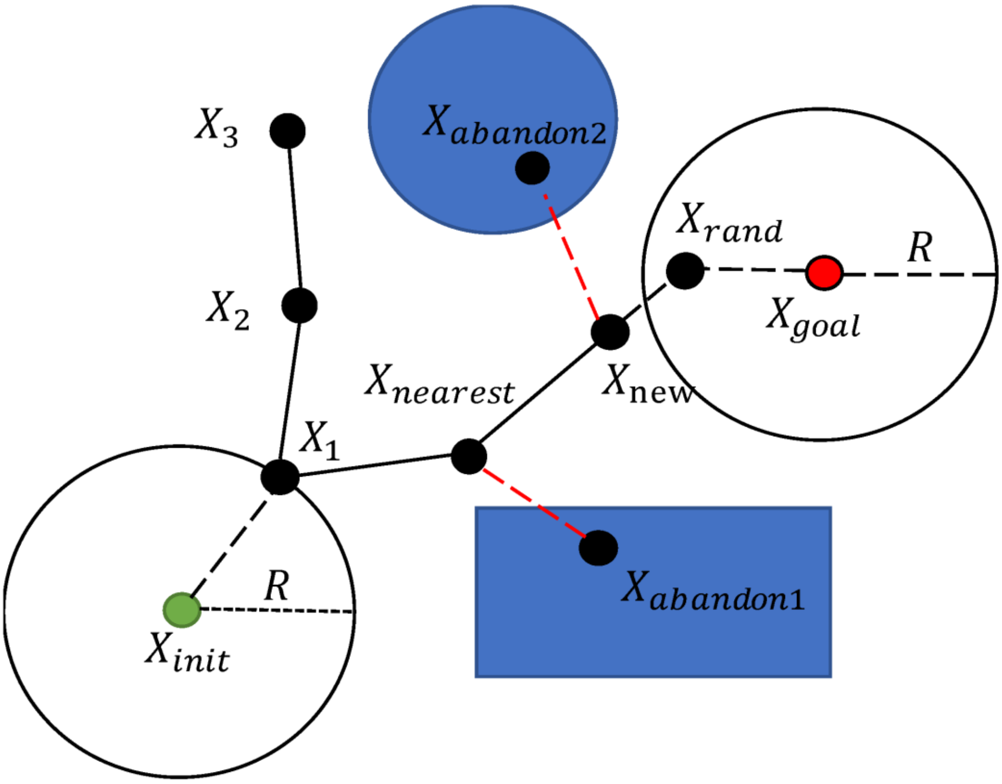


Figure 2: *RRT algorithm random tree expansion process.*

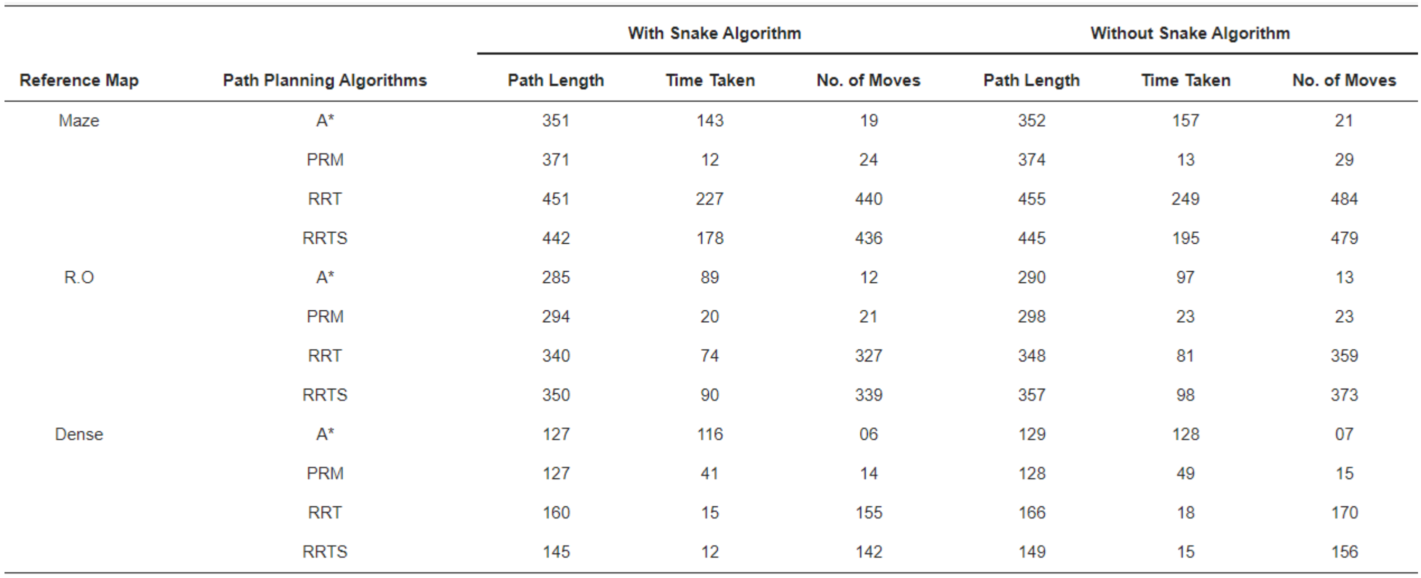
RRT algorithm

```
Step1:  T = init Tree (); // Initialize the random tree
Step 2:  R = init R (); // Initialize the step size
Step 3:  T [0] = Node ($X_{init}$);
Step 4:  for i = 1 to N:
Step 5:      $X_{rand}$= Random sampling (); // Random sampling
Step 6:      Xnearest= min the distance ($X_{rand}$)Nodetree); // Find the nearest node
Step 7:      Xnew = extend ($X_{nearest}$,$X_{rand}$, R); // Expand towards random points
Step 8:      if not obstacle($X_{nearest}$, $X_{new}$):
Step 9:         T[i] = add Node(Xnew); // Add a new node to the tree
Step 10:     end if
Step 11:  end for
Step 12:  return T
```

3.3. Work Flowgraph of Proposed Work

In the above section, the working principles of the Algorithm (A*, PRM, RRT, and RRT-S) are discussed in detail, and the working procedure is summarised in Table 2. The graph explains the basic steps of the algorithms, initially the presence of contour is detected by the snake algorithm (Here in this case contours are the obstacles present in the reference maps) and then the nodes are initialized, and the complete periphery of an obstacle is then traced out which is being provided to the path planning algorithms, and then points the unique procedures followed by each algorithm is carried out. In the end output of each procedure would be the parameter taken to test the results shown in Table 2.

Table 2. Comparative analysis of results obtained on the reference maps using path planning algorithms with snake algorithms and without snake algorithms.



		With Snake Algorithm			Without Snake Algorithm		
Reference Map	Path Planning Algorithms	Path Length	Time Taken	No. of Moves	Path Length	Time Taken	No. of Moves
Maze	A*	351	143	19	352	157	21
	PRM	371	12	24	374	13	29
	RRT	451	227	440	455	249	484
	RRTS	442	178	436	445	195	479
R.O	A*	285	89	12	290	97	13
	PRM	294	20	21	298	23	23
	RRT	340	74	327	348	81	359
	RRTS	350	90	339	357	98	373
Dense	A*	127	116	06	129	128	07
	PRM	127	41	14	128	49	15
	RRT	160	15	155	166	18	170
	RRTS	145	12	142	149	15	156

3.4. TOOLS

The tools/ software required to develop this website are:

- Frontend Programming Language: Python
- Styling and Front End: Python
- Backend: Spyder 5.0.3
- Database: Environment Obstacles

CHAPTER 4: RESULTS AND DISCUSSIONS

4.1. Results

- In order to investigate the effectiveness of the GA_RRT algorithmic rule for the trail coming up with drawback, Thesis compares with the standard RRT algorithmic rule and also the RRT algorithmic rule supported goal bias chance (G_RRT) in 2 totally different eventualities.
- The simulation chiefly adopts 2 common strategies to verify the validity of the trail coming up with algorithmic rule. situation A is shown in Figure half dozen, the initial purpose is one, one and also the target purpose is 750,750. As are often seen in Figure 6a, the RRT algorithmic rule explores the complete house, whereas the G_RRT algorithmic rule explores the target purpose with a chance of zero.
- This greatly reduces the number of nodes for special search. The GA_RRT algorithmic rule performs node pruning by hard the price perform whereas exploring the goal purpose. This additional effectively reduces the amount of tree node branches and quickens the search potency.
- The best way to do a qualitative check on holonomy vs. non-holonomy is if you look at a robot in its operating space, can you pick any two positions not blocked by an obstacle where the robot will need to reposition itself before directly moving to its goal position.

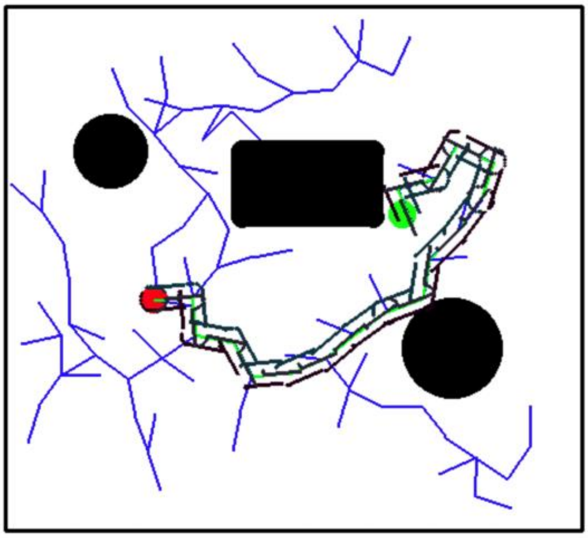


Figure 3: *Holonomic case*

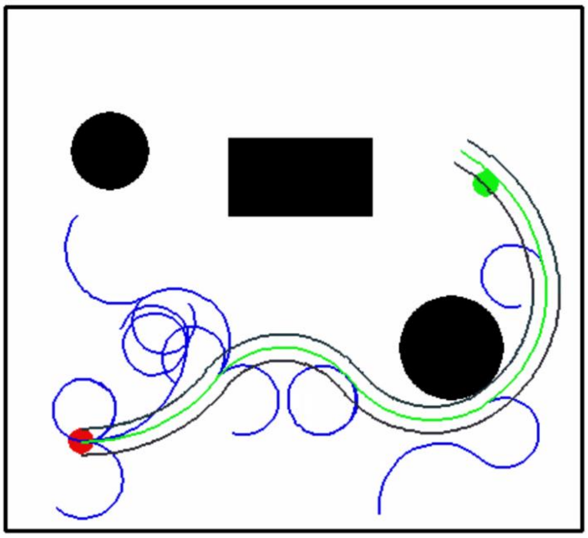


Figure 4: *Non-Holonomic case*

Figure. 3 refers holonomic, to a restriction (or not) among translational axes. If a robot is holonomic with respect to N dimensions, it's capable of moving in any direction in any of those N physical dimensions available to it. Figure. 4 refers non-holonomic robot are subject to nonintegrable equality nonholonomic constraints involving the velocity. the dimension of the admissible velocity space is smaller than the dimension of the configuration space.

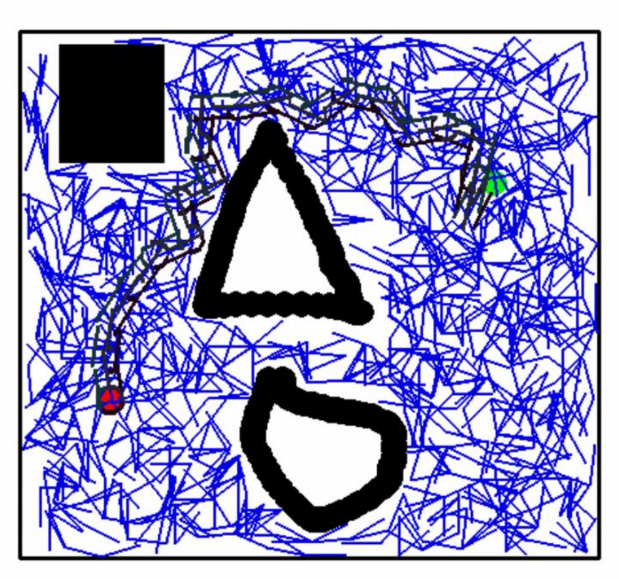
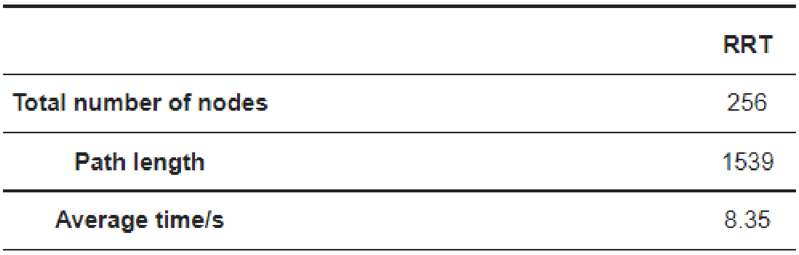


Figure 5: *Hand drawn obstacles*

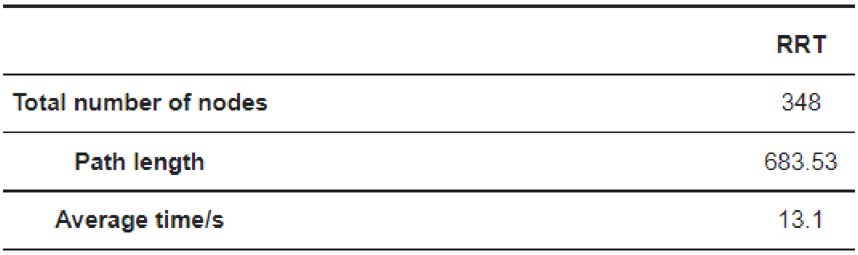
In the experiment, since the paths generated by the RRT algorithm have high randomness, the number of nodes and time of each path may be quite different. Therefore, it conducts ten experiments for each algorithm, and takes the average value as the final data. The path data for Scenario shown in Table 3.

Table 3. The average experimental data for Scenario A.



	RRT
Total number of nodes	256
Path length	1539
Average time/s	8.35

Table 4. The average experimental data for Scenario B.



	RRT
Total number of nodes	348
Path length	683.53
Average time/s	13.1

4.2. System Flow Chart

System flowcharts are a way of displaying how data flows in a system and how decisions are made to control events. System Architecture of Stock Market Prediction using LSTM.

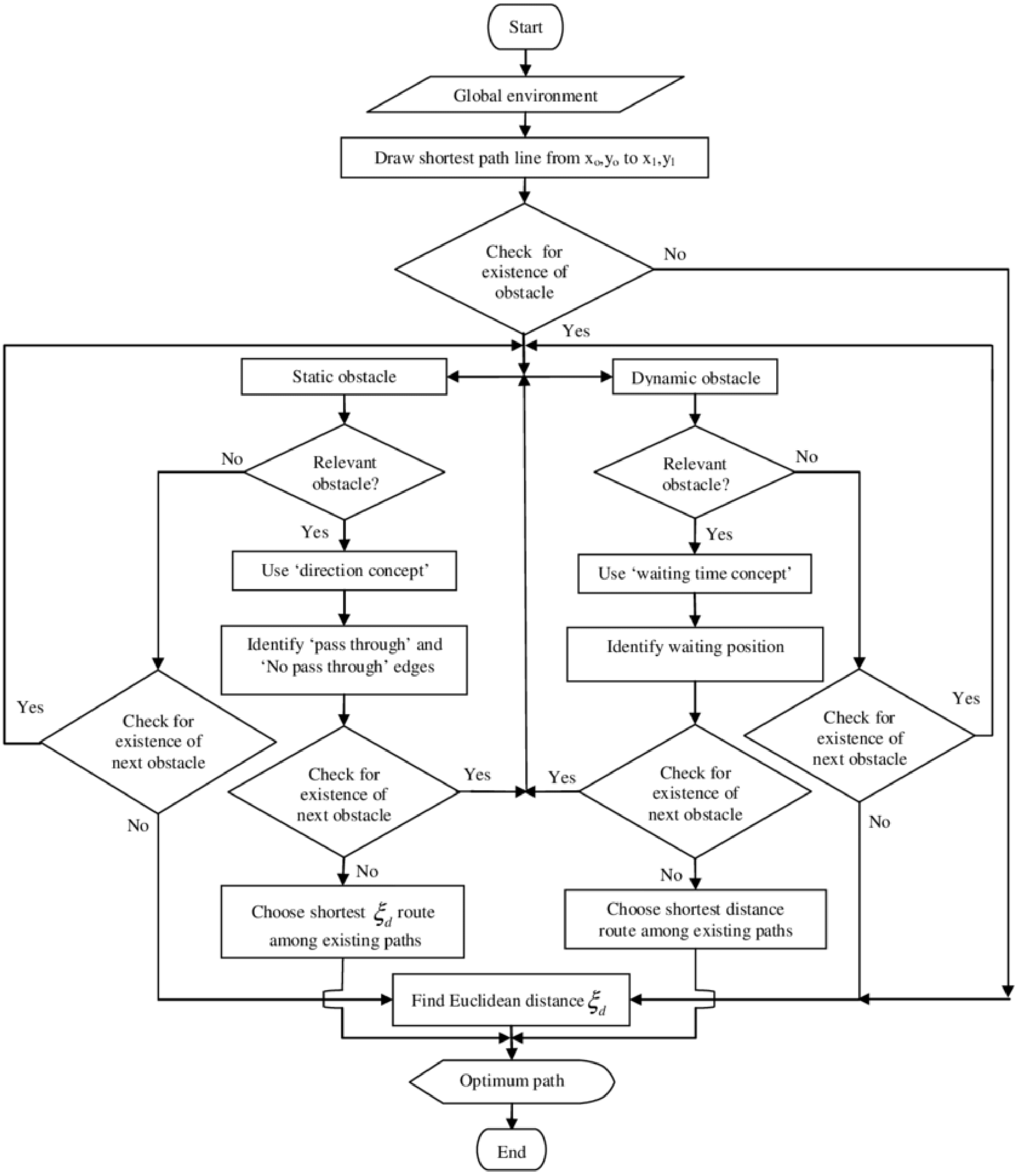

Start
Global environment
Draw shortest path line from x_o, y_o to x_1, y_1
Check for existence of obstacle
No
Yes
Static obstacle
Dynamic obstacle
Relevant obstacle?
No
Yes
Use 'direction concept'
Identify 'pass through' and 'No pass through' edges
Relevant obstacle?
No
Yes
Use 'waiting time concept'
Identify waiting position
Yes
Check for existence of next obstacle
No
Check for existence of next obstacle
Yes
No
Yes
Check for existence of next obstacle
No
Yes
Check for existence of next obstacle
No
Choose shortest ξ_d route among existing paths
Choose shortest distance route among existing paths
Find Euclidean distance ξ_d
Optimum path
End

Figure 6: *System Flow Chart for Robotic Path Planning*

4.3. Data Flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a system(usually an information system). The DFD also provides Flow Diagram: System Flow Chart information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.

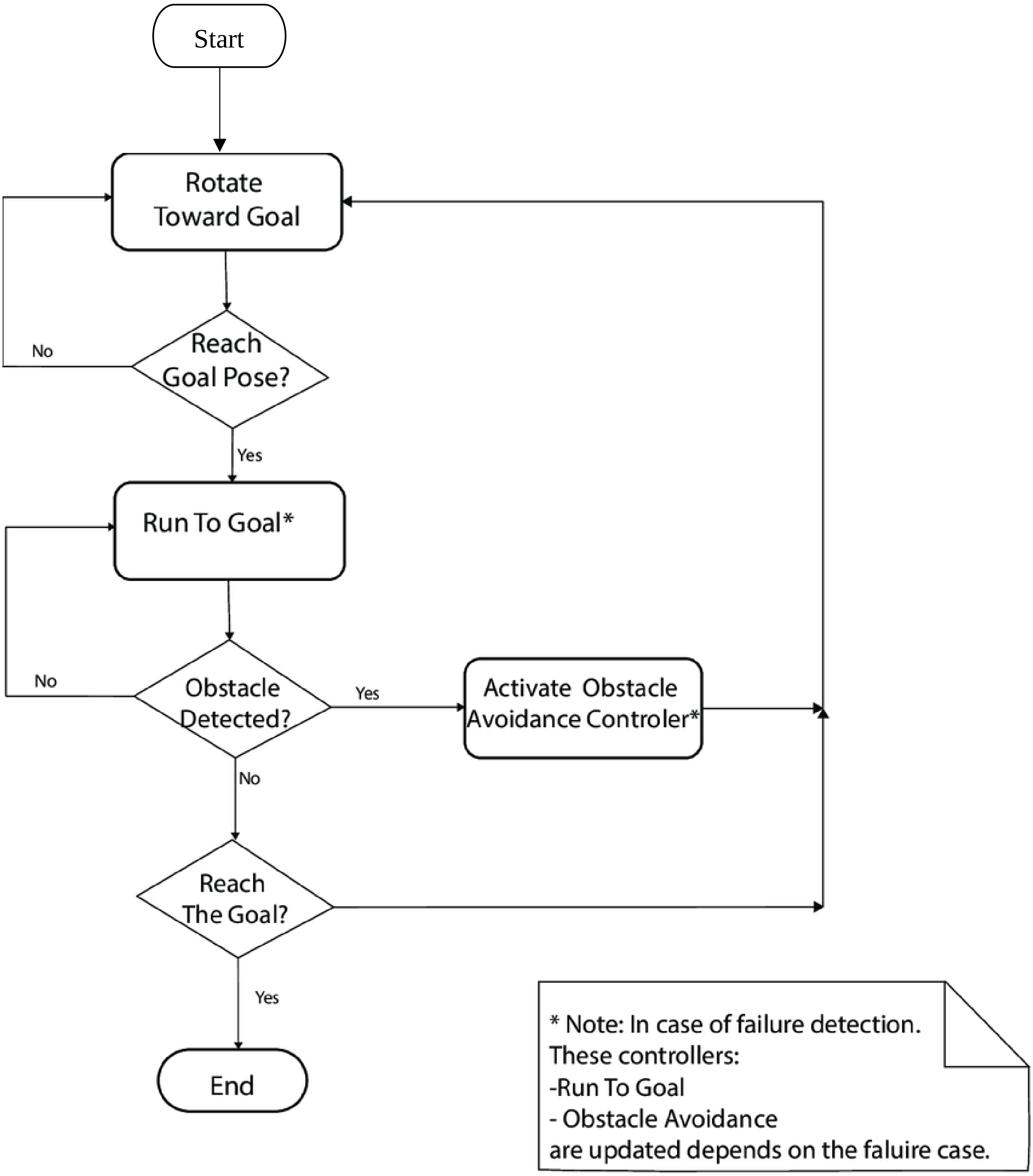


Figure 7: *Data Flow Diagram for Robotic Path Planning*

4.4. Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

The purpose of the class diagram can be summarized as −

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

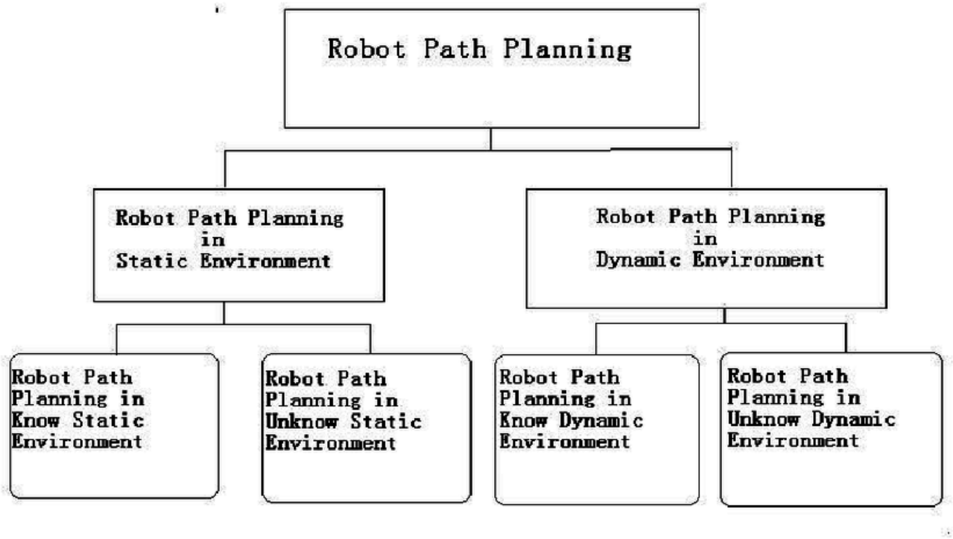

Robot Path Planning
Robot Path Planning in Static Environment
Robot Path Planning in Dynamic Environment
Robot Path Planning in Know Static Environment
Robot Path Planning in Unknow Static Environment
Robot Path Planning in Know Dynamic Environment
Robot Path Planning in Unknow Dynamic Environment

Figure 8: *Class Diagram for Robotic Path Planning*

CHAPTER 5: SUMMARY AND CONCLUSIONS

Based on the traditional RRT algorithm, we introduce the goal probability bias and A* cost function algorithm. The search time and efficiency of the RRT algorithm is greatly optimized.

The advantage of the developed algorithm is that the robot always can move from the initial position to the target position, not only safely, but also on the shortest path regardless the shape of the obstacles and the change of goal position in the known environment. In the other side, the proposed sliding mode control is an important method to deal with the system. Simulation results are performed on a platform Spyder to demonstrate that the proposed method is a good alternative to solve the path planning and trajectory tracking problems.

Having a smoother route with a lower risk of collision helps the robot to move more safely while working with humans. The results of this work indicate the potential applications of hybrid A-star methods in the indoor environment that require autonomous driving tasks. However, the proposed method needs to be verified in real environments, which we will investigate in our future work.

REFERENCES

[1] *Huang, Jiunn-Kai, Yingwen Tan, Dongmyeong Lee, Vishnu R. Desaraju, and Jessy W. Grizzle. "Informable Multi-Objective and Multi-Directional RRT* System for Robot Path Planning." arXiv preprint arXiv:2205.14853 (2022).*

[2] *Dam, Tuan, Georgia Chalvatzaki, Jan Peters, and Joni Pajarinen. "Monte-Carlo Robot Path Planning." IEEE Robotics and Automation Letters 7, no. 4 (2022): 11213-11220.*

[3] *Hu, Zijiang, Jian Qin, Zhongxin Wang, and Jian He. "Robot Path Planning Based on Multi-strategy Improved RRT* Algorithm." In 2022 6th International Conference on Automation, Control and Robots (ICACR), pp. 17-24. IEEE, 2022.*

[4] *Wu, Daohua, Lisheng Wei, Guanling Wang, Li Tian, and Guangzhen Dai. "APF-IRRT*: An Improved Informed Rapidly-Exploring Random Trees-Star Algorithm by Introducing Artificial Potential Field Method for Robot Path Planning." Applied Sciences 12, no. 21 (2022): 10905.*

[5] *Wang, Jiankun, Tingguang Li, Baopu Li, and Max Q-H. Meng. "GMR-RRT*: Sampling-based Path Planning Using Gaussian Mixture Regression." IEEE Transactions on Intelligent Vehicles (2022).*

[6] *Shi, Wubin, Ke Wang, Chong Zhao, and Mengqi Tian. "Obstacle Avoidance Path Planning for the Dual-Arm Robot Based on an Improved RRT Algorithm." Applied Sciences 12, no. 8 (2022): 4087.*

[7] *Wang, Xuewu, Jianbin Wei, Xin Zhou, Zelong Xia, and Xingsheng Gu. "AEB-RRT*: an adaptive extension bidirectional RRT* algorithm." Autonomous Robots 46, no. 6 (2022): 685-704.*

[8] *Sharma, Kaushlendra, Chetan Swarup, Saroj Kumar Pandey, Ankit Kumar, Rajesh Doriya, Kamred Udham Singh, and Teekam Singh. "Early Detection of Obstacle to Optimize the Robot Path Planning." Drones 6, no. 10 (2022): 265.*

[9] *Li, Jian, Rui Cui, Peng Su, Lifang Ma, and Hao Sun. "A Computer-assisted Preoperative Path Planning Method for the Parallel Orthopedic Robot." Machines 10, no. 6 (2022): 480.*

[10] *Zhang, Junning, Hao Wang, Yajing Guo, Fan Yang, and Qing Zhao. "Research on Dual-Arm Robot Assembly Path Planning Based on Improved RRT* Algorithm." In Chinese Intelligent Systems Conference, pp. 786-798. Springer, Singapore, 2022.*