Moving Domestic Robotics Control Method Based on Creating and Sharing Maps with Shortest Path Findings and Obstacle Avoidance

Karan Bhakuni
IIT Jodhpur
m22rm003@iitj.ac.in

Padhye Soham Satish
IIT Jodhpur
m22rm007@iitj.ac.in

April 28, 2023

Abstract

This paper deals with the methods for obstacle avoidance, and finding a shortest pass based on Dijkstra algorithm.Concept of Place Identifier is used by the Robot.Places are identified by using identifiers placed outside the rooms.Also control method for moving robotic in closed areas is proposed.Path is planned by creating and sharing maps using shortest path techniques and obstacle avoidance. The effect of map sharing among multiple robotics is confirmed. The proposed method is validated through a simulation study in MATLAB and Then the effectiveness of obstacle avoidance using cameras and ultrasonic sensors is also confirmed in the paper. The camera is continuously taking the images and comparing it with the reference image and finding the norm image.In summary paper discuss about the concept of creating and sharing the map,finding the shortest path using Dijkstra algorithm and obstacle avoidance using the concept of norm image.

1 Introduction

The integration of robotics in our daily lives has become increasingly prevalent in recent years. With the advancement of technology, the potential for domestic robotics has become limitless. One of the key challenges in this field is developing efficient control methods for robots that can navigate through complex environments while avoiding obstacles. In this report, we will explore a new approach to domestic robotics control, based on creating and sharing maps with shortest path findings and obstacle avoidance. This approach utilizes Dijkstra algorithm for path planning, which is a method used in computer science to calculate the shortest path between two points while considering obstacles in the environment. We will examine the benefits and limitations of this approach and evaluate its potential to improve the functionality and practicality of domestic robotics. We will also make the use of camera to detect the obstacles in the path of the robot by taking the norm image. This concept is explained in detail next in the paper. The new concept of map sharing is proposed in the paper.Map sharing is very helpful in case of multi robot system. As one robot encounters an obstacle it can update the map and share with others, such that other robots can plan the path efficiently.

2 Path Planning and Obstacle Avoidance

2.1 Fundamental Idea

Movement of mobile robot is based on Place Identifiers. If we take example of a hospital, so each room is equipped with some identifiers on it.The tag information includes various details such as the name of the hospital, floor, room, interior, and data acquisition time.It is possible to identify any specific item within the hospital's interior, room, or floor using PI.So with the help of the concept of Place Identifier we can create the nodes for Dijkstra algorithm. And by the help of Dijkstra algorithm we can find the shortest path.To control the robot we have used the proportional controller. By selecting the proper value of proportional gain Kp we can control the robot and we can successfully move the robot to the goal position.

2.2 Process Flow

1.First the robot will determine start and end of the path.
2.Then based on this the robot starts creating the map.
3.Then using Dijkstra algorithm shortest path is cal-

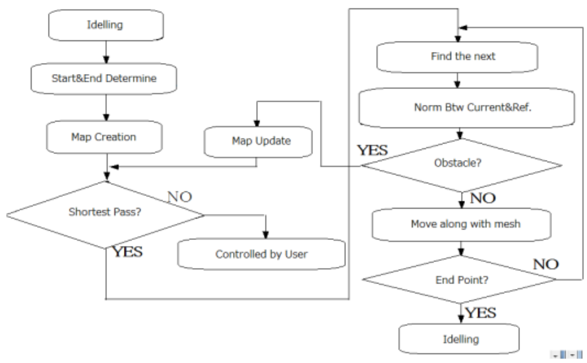

Figure 1: Process Flow Diagram

culated between start and end point.
4.It will start moving and follow the nodes created by Dijkstra.
5.Compare the reference and current image to check for the obstacle.
6.If no obstacle is there, move along the node.But if obstacle is there, then update the map.
7.Check if you reached to end or not?
8.If not start from point 4. 9. When the second robot starts moving from the start position it will have updated map. Map updation is done by first robot. Updated map contains information of transient obstacles.

2.3 Dijkstra Algorithm for finding shortest path

Dijkstra's algorithm is a graph traversal algorithm that is used to find the shortest path between two nodes in a graph.

The algorithm works by starting at the source node and iteratively visiting neighboring nodes, updating the distance to each node as it goes. At each iteration, it selects the node with the shortest distance from the source that has not been visited yet, and adds it to the set of visited nodes. The algorithm then updates the distances of all its neighboring nodes, taking into account the distance from the newly added node.

This process continues until the algorithm reaches the target node or has visited all nodes reachable from the source. Once the target node is reached, the algorithm backtracks along the shortest path to the source node, recording the nodes and edges along the way.

Dijkstra's algorithm is guaranteed to find the shortest path between the source and target nodes, as long as the graph is weighted, connected, and has no negative-weight edges. Negative-weight edges can cause the algorithm to fail, as it relies on the assumption that the shortest path to any node can only be improved upon by visiting nodes with shorter distances.

Overall, Dijkstra's algorithm is a powerful and efficient algorithm for finding shortest paths in graphs, and has many practical applications in areas such as network routing, transportation planning, and logistics.
The Simulation results in MATLAB are shown below:

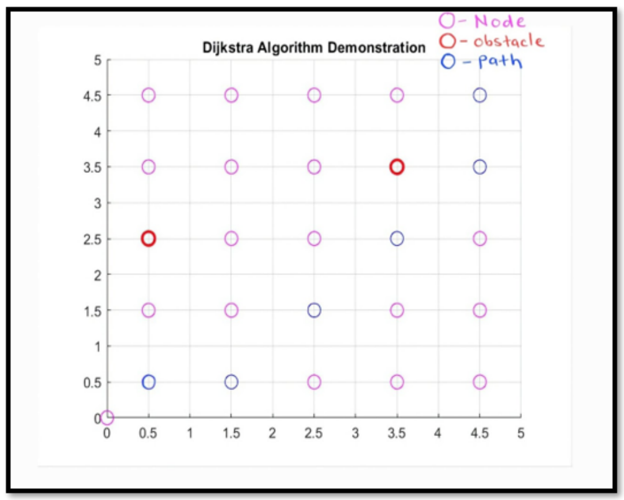

Figure 2: Shortest Path by Dijkstra Algorithm

2.4 Method for obstacle avoidance

There are three types of obstacles: static, dynamic, and transient. Static obstacles include walls, doors, and interiors, while dynamic obstacles include moving objects such as nurses, doctors, and patients. Transient obstacles suddenly appear and disappear. The robot must use the Dijkstra algorithm to avoid both static and transient obstacles that appear suddenly in front of it while moving. Suppose the first robot encounters another obstacle then it must change its route using the Dijkstra algorithm. Obstacle node is also removed from the graph so that robot will not visit it again. When second robot begins its journey towards the same destination it already have updated map so that it will reach the goal position in less time.

3 Map Sharing

In the paper, map sharing between the multiple robots is proposed. It has made the planning easy as when one robot encounter the obstacle it updates the map. Then next robot will get the updated map and if another obstacle appear it again updates the map for next robot. In this way shared map has made the task easier. The effect of the map sharing is saturated when number of moving obstacles crosses a certain threshold value.Sharing maps among multiple robots requires communication between the robots. As the number of obstacles increases, the amount of information that needs to be communicated also increases. The number of resources available to each robot in a multi-robot system, such as processing power and memory, is limited. As the number of obstacles increases, the amount of data that needs to be stored and processed also increases, which can put a strain on the available resources.

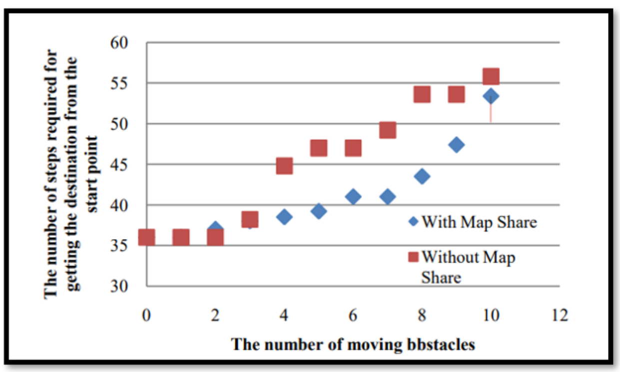

Figure 3: Map Sharing Saturation Graph

4 Control of Unicycle Robot

Algorithm for controlling the unicycle robot using proportional control- Initialize the robot's position and orientation. Set the desired position and orientation for the robot.In this case the goal position is not given directly . We have nodes given by the Dijkstra algorithm and we will set each node as the current goal for the robot. robot will trace all the nodes one by one.

While the robot is not at the desired position(current goal) and orientation, repeat the following steps:

a. Measure the robot's current position and orientation.

b. Calculate the error between the desired position and orientation and the current position and orientation.

c. Calculate the control input as a proportional function of the error. For example, the control input could be proportional to the difference between the desired and current orientation.

d. Apply the control input to the robot's motors or actuators to move it towards the desired position(node) and orientation.

e. Repeat this process until the robot is within an acceptable margin of error.

Results of Simulation in MATLAB is shown below:

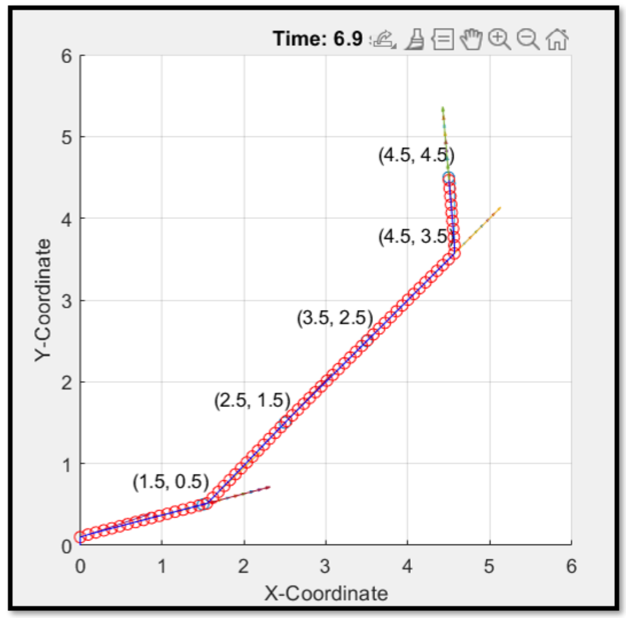

Figure 4: Control of Uncicyle Robot over Desired Points

4.1 Selection of Kp

Start with a small value of Kp and increase it gradually until we observe oscillations in the system's response. Once we have identified the oscillation threshold, reduce the Kp value to a point where the system response is stable and not oscillating. If the

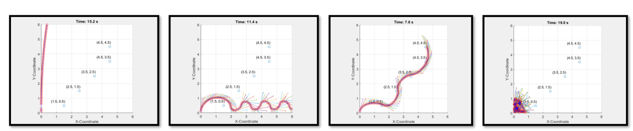

Figure 5: Control for different values of Kp

system's response is too slow, increase Kp gradually until the system responds faster but still stable. If the system's response is too fast and overshoots the desired setpoint, decrease Kp gradually until the system responds without overshooting.
In this way by we have found out the optimal value of Kp is 10. The system response for the different values of Kp is shown below:

5 Norm(Difference) Image

Unicycle robot is equipped with the Near Infrared camera. With the help of this camera it is continuously taking the images of the environment and finding the norm image it is finding the obstacles as shown in the figure below:

For finding the norm image calculate the absolute difference between the reference image and the current image.Then calculate the norm of the absolute difference image.If the norm of the image is greater than zero the there is a obstacle in the image.

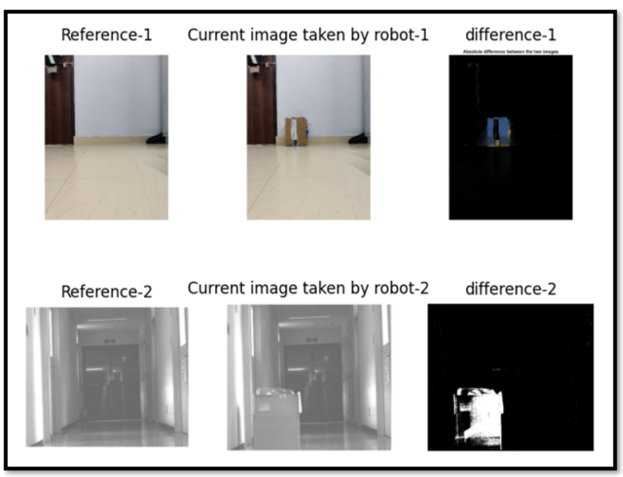

Figure 6: Norm Images for Obstacle Avoidance

6 Conclusion

With the help of Dijkstra algorithm, we found out the shortest path between Start and Goal position of the robot while avoiding two obstacles in between them. Obstacles are identified by the camera using norm images. The graph is made using MATLAB.Also with the help of shared map between the robots, planning will become an easier task as compared to when it is not shared.With the help of Map sharing we can reduce the time to reach the goal position. But if the number of obstacles are more then the map sharing result saturates and it wont show any improvement in total time to reach the end goal.
We have developed the code for finding the shortest path using Dijkstra algorithm in MATLAB. We also have demonstrated the simulation of control of unicycle robot on MATLAB that follows the shortest path and reach the destination in minimum time.

7 Future Work

1. We are unable to do the simulation of the Map Sharing concept proposed in the research paper.
2. Also we have not tested/implemented obstacle avoidance using ultrasonic sensor.
3. Value for proportional gain is chosen by trial and error method.
4. We have developed only Proportional controller, we can go for more advanced controller like MPC .

8 References

1.Moving Domestic Robotics Control Method Based on Creating and Sharing Maps with Shortest Path Findings and Obstacle Avoidance.By Kohei Arai
2.Understanding Dijkstra's Algorithm.By Muhammad Adeel Javaid