



A NOVEL APPROACH TO PATH PLANNING OF ROBOTS BY DETECTING DYNAMIC OBSTACLES

Syed Nawaz Pasha¹, D. Ramesh² & G Roopa³

Abstract-Artificial intelligence has gained importance in the area of robotics. The robot that thinks like humans and act according is a big area of research. The problem of finding optimal path from source to destination in robotic movement is important. The path planning in robots typically consists of two types. One is the static path in which the obstacles are static and the path is also fixed. The robot moves in that specified path in static path planning. In dynamic path planning the obstacles will be in motion and there is no fixed path. Hence the robot has to determine the collision free path identifying the moving obstacles and plan its path accordingly. Before identifying the optimal path first we need to identify optimal obstacles so that it reduces the search space. The obstacles are static if the distance between the robot and the obstacle is constant. If the obstacle is a dynamic one then the distance between the robot and the obstacle will be changing. The robot will be incorporated with sensors that will be able to sense the obstacles within the specified range. In this paper we propose a method how to detect moving obstacle and plan a path that is collision free. We assume that if the distance between the robot and obstacle is increasing then the obstacle is moving away from the robot and the path is free so that the robot can move forward. But if the movement of the obstacle is towards the robot then we calculate the distance take an alternative path. The Movement of the robot is based on the sensory information of the obstacles.

KeyWords: Artificial Intelligence, Path planning, Mobile Robot.

1. INTRODUCTION

One of the greatest challenges for a mobile robot is finding a path from one point to another point in any environment. So the path finding strategies are playing a major role in an autonomous system. Path finding strategies have the responsibility of finding a path from one coordinate to another coordinate in any environment by taking a starting point and a destination point. Then they find a series of points that together comprise a path to the destination. For this a set of pre-compiled instructions are employed into the system to guide the robot. And these instructions extract the information from the environment and returns the coordinates to which the robot have to move in an environment when we give start and goal positions. Algorithms to find a shortest path are important not only in robotics, but also in network routing, video games and gene sequencing. Path-planning requires a map of the environment and the robot to be aware of its location with respect to the map.

So before planning the path we need to find the number of obstacles in the sensing area of the robot and differentiate them as static or dynamic. If the obstacles are static we follow the offline path planning, but if the obstacles are dynamic we need to find the distance between the robot and the obstacle and plan the path accordingly based on the sensory information. In dynamic environment we only consider the obstacles that are moving towards the robot. Hence we optimize the path calculation of all the obstacles and calculate the path only for obstacles whose motion is towards the robot. This technique optimizes the search space.

2. LITERATURE SURVEY

A* algorithm [7] [6] is a space-search algorithm that can be used to find solutions too many problems. This Algorithm works on best-first Algorithm just it offers qualities to the hubs in an unexpected way. Every hub's worth is the total of the genuine expense to that hub from the begin and the heuristic assessment of the rest of the expense from the hub to the objective. Thusly it consolidates the following of past length from Dijkstra's Algorithm with the assessment of the rest of the way from the best first inquiry.

Genetic Algorithms are optimizing algorithm search in a space of potential solutions. The solution to the problem of planning by Genetic Algorithms is proposed for the first time for static environment only. The method involves choosing the initial population. Most of these methods use a set of paths that encoded in the chromosomes. To obtain the optimal path we perform several rounds.. Next we determine the fitness function which is determined by the linear combination of distance, the smoothing angle and the robot position from the obstacles.

¹ Assistant professor, Department of CSE, S R Engineering College Warangal 506371, Telangana.

² Assistant professor, Department of CSE, S R Engineering College Warangal 506371, Telangana.

³ Assistant professor, Department of CSE, S R Engineering College Warangal 506371, Telangana.

Most of the path planning algorithms searches for all the obstacles within sensory area and Considers all the available obstacles to find final path. But this approach is not an effective one because the search area is increased which leads to decrease in effectiveness.

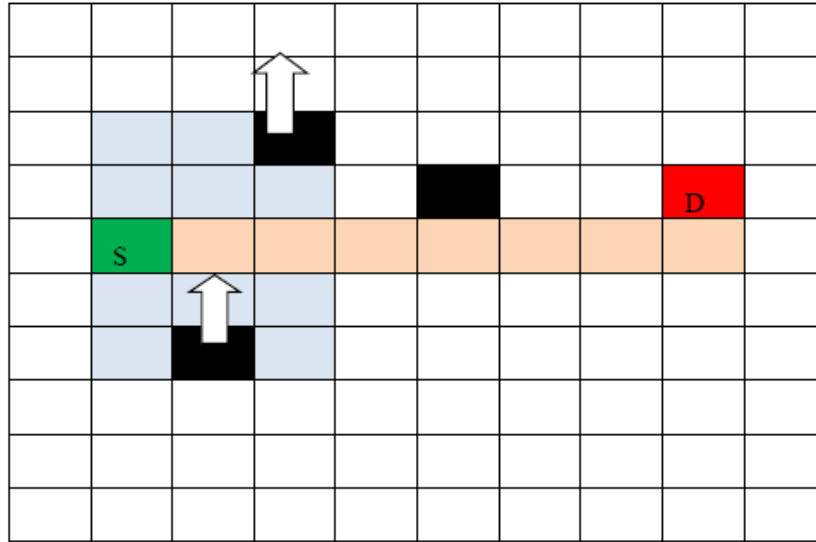


figure 1.1: Path Planning with static obstacles S-Source D-Destination

The problem with the above approach is that the robot encounters many difficulties in identifying the obstacles such as the identification of static obstacles, dynamic obstacles, the number of obstacles that are moving towards the robots, the number of obstacles that are moving away from the robot.

To overcome this approach of considering all obstacles in the sensor area we propose a new approach of detecting obstacles and differentiate those obstacles based on the sensory information.

3. PROPOSED WORK:

In this paper we propose a way of differentiating the obstacle which are to be considered and not to be considered and the robot will decide a path based on the sensory information of the obstacles.

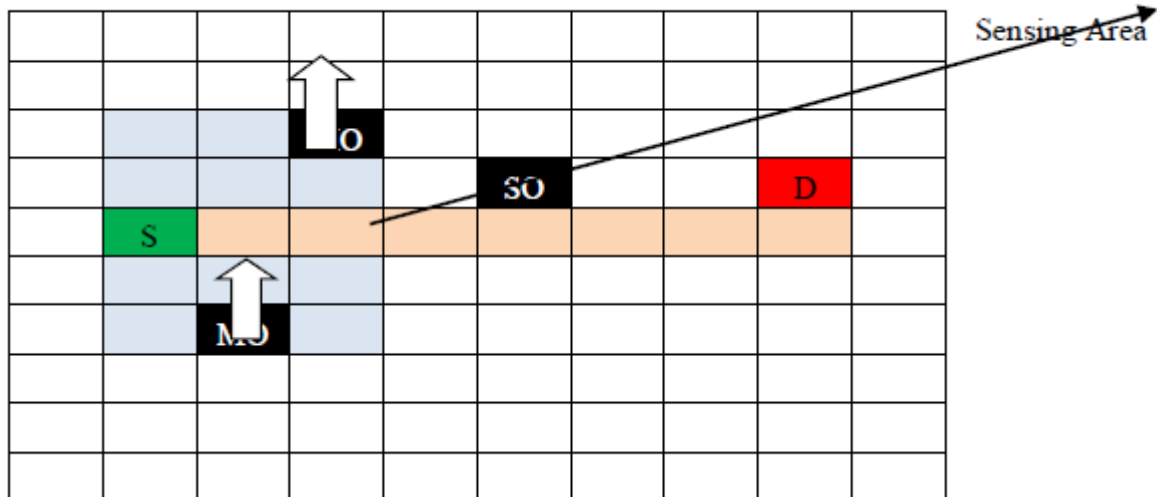


figure 2.1: Path Planning with static and Dynamic Obstacles

S-Source ,D-destination ,SO-Static Obstacle, MO-Moving Obstacle,

Whenever the robot is moving in dynamic environment it needs to identify the static obstacles and dynamic obstacles. if it's a static obstacle , the robot identifies these obstacles based on the distance from the robot to the obstacle. When the distance is constant then the robot assumes the obstacle is static.

If it is a moving obstacle then the robot will determine it as moving obstacle if the distance between the robot and the obstacle is varying. In this scenario the robot will differentiate obstacles as either coming towards the robot or moving away from robot. In the above figure 3.1 the robot will consider the obstacles as either static or dynamic. If the distance from robot to

obstacle is constant then it assumes as static and plans the path accordingly. If the obstacle is moving then we consider two cases) the obstacle is moving towards the robot. .ii) the obstacle is moving away from the robot .In the above two cases we in this paper are concerned with the obstacles that are moving towards the robot.

Approach:

Sensors	Obstacle detected(yes/No)	Initial distance from robot to obstacle	Distance after few seconds	Status
S1(Front)	Yes	D1	increased	dynamic
			Decreased	Static/dynamic
			constant	static
S2(behind)	No	Keep sensing	Keep sensing	Keep sensing
			Yes	D2
			increased	dynamic
S3(right)	Yes	D3	Decreased	Static/dynamic
			constant	static
			Keep sensing	Keep sensing
S4(Left)	No	Keep sensing	Keep sensing	Keep sensing
			Yes	D4
			increased	dynamic
S4(Left)	Yes	D4	Decreased	Static/dynamic
			constant	static
			Keep sensing	Keep sensing

Table: Sesors Data Processing for identifying the obstacles as Static and Dynamic

The robot sense the obstacle within the sensing area and then decide the obstacle as static or dynamic based on the following information. Moreover the sensed information is stored in the memory in the form of 4 sensors i.e. S1(front),S2(Behind),S3(right),S4(left).By default we consider the robot is in motion.

Case 1: S1(Front)

If the obstacle is sensed in the front direction and the distance from the robot to obstacle is increasing then it will be considered as dynamic obstacle. If the distance is decreased then it is either a static or dynamic obstacle. If the distance is constant then we consider it as static obstacle. If there is no obstacle in front then it keeps sensing for obstacles.

Case 2: S2(Behind)

If the obstacle is sensed in the behind direction and the distance from the robot to obstacle is increasing then it will be considered as dynamic obstacle. If the distance is decreased then it is either a static or dynamic obstacle. If the distance is constant then we consider it as static obstacle. If there is no obstacle behind then it keeps sensing for obstacles.

Case 3: S3(Right)

If the obstacle is sensed in the right direction of the robot and the distance from the robot to obstacle is increasing then it will be considered as dynamic obstacle. If the distance is decreased then it is either a static or dynamic obstacle. If the distance is constant then we consider it as static obstacle. If there is no obstacle to the right then it keeps sensing for obstacles.

Case 3: S4(Left)

If the obstacle is sensed in the Left direction of the robot and the distance from the robot to obstacle is increasing then it will be considered as dynamic obstacle. If the distance is decreased then it is either a static or dynamic obstacle. If the distance is constant then we consider it as static obstacle. If there is no obstacle to the Left then it keeps sensing for obstacles.

In this way we decide the obstacles as either static or dynamic and consider only those obstacles which are moving towards the robot. Then the robot will choose the path based on this information .

The above approach is an effective one in identifying the obstacles as we are not considering all the obstacles. Instead we consider only those which are in the sensory area and the one which is moving towards the robot which reduces the search area and improves the effectiveness of path planning.

$$\text{searched area} = \frac{\text{no.of cells searched}}{\text{total no.of cells}} \times 100$$

$$\text{efficiency} = \frac{\text{length of the final path}}{\text{number cells searched}} \times 100$$

4. CONCLUSION

In robotics path finding plays a major role to plan a collision free path by avoiding unnecessary obstacles in specified path. Many researchers are working on this issue and proposed different algorithms. But no method is giving optimal solution for this problem. So we proposed an idea of Detecting dynamic and static obstacles based on the sensory information that are optimal before finding optimal path which improves the effectiveness of path planning by reducing unnecessary obstacles. The proposed approach of identifying the obstacles is more effective and reduces the search space as it will not consider unnecessary obstacles and will consider only the obstacles which are in its path. In future we will extend how actual path planning of the robots will be done based on the information that is stored in the robot sensors.

5. REFERENCES

- [1] C.-K. Yap, "Algorithmic motion planning," in *Advances in Robotics vol. 1: Algorithmic and Geometric Aspects of Robotics*, J. T. Schwartz and C.-K. Yap, Eds. Hillsdale, New Jersey: Lawrence Erlbaum, 1987.
- [2] Y. Wang, D. J. Mulvaney and I. P. W. Sillitoe, "Genetic-based mobile robot path planning using vertex heuristics," 2006 IEEE Conf. Cybernetics and Intelligent Systems, pp.463-468.
- [3] D. W. Payton, J. K. Rosenblatt and D. M. Keirse, "Grid-based mapping for autonomous mobile robot," *Robotics and Autonomous Systems*, vol. 11, no. 1, pp. 13-21, 1993.
- [4] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime Dynamic A*: An Anytime, Replanning Algorithm," 2005 International Conference on Automated Planning and Scheduling, pp.
- [5] E. Horowitz and S. Sahni, "Fundamentals of computer algorithms," 1978.
- [6] M. Phillips and M. Likhachev, "Sipp: Safe interval path planning for dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011.
- [7] G. A. Mills-Tettey, A. Stentz, and M. B. Dias, "Dd* lite: Efficient incremental search with state dominance," in *Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, July 2006, pp. 1032-1038.
- [8] K. Bekris and L. Kavraki. Greedy but safe replanning under kinodynamic constraints. In *IEEE International Conference on Robotics and Automation*, 2007
- [9] Furuhashi T., Nakaoka K., and Uchikawa Y., A Study on Classifier System for Finding Control Knowledge of Multi-Input Systems, (F. Herrera and J.L. Verdegay, Editors), *Genetic Algorithms and Soft Computing*, Physica-Verlag, 1996.
- [10] M. Phillips and M. Likhachev. Sipp: Safe interval path planning for dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011