

## A MOVING OBSTACLES PLANNER ALGORITHM FOR A SIX-WHEELED SURVEYING ROBOT

S. Bazhan<sup>1</sup>, A. Hosseiniaveh<sup>2</sup>

<sup>1</sup> Dept. of Photogrammetry and Remote Sensing, Geodesy and Geomatics Engineering Faculty, K. N. Toosi University of Technology, Tehran, Iran - [s\\_bazhan@email.kntu.ac.ir](mailto:s_bazhan@email.kntu.ac.ir)

<sup>2</sup> Dept. of Photogrammetry and Remote Sensing, Geodesy and Geomatics Engineering Faculty, K. N. Toosi University of Technology, Tehran, Iran - [Hosseiniaveh@kntu.ac.ir](mailto:Hosseiniaveh@kntu.ac.ir)

**KEY WORDS:** Path Planning, MOOR, ROS, MOP, RRT, Dijkstra, A\*, DWA

### ABSTRACT:

Nowadays, robotic systems such as ground vehicle robots are mostly used in many industrial and military applications. Therefore, the path planning problem in the robotics domain is very important. Moving Obstacles Planner (MOP) algorithms have got the researchers interests in recent years and some of the most recent ones have been implemented in Robot Operating System (ROS) which is an open source middle wear to work with robots. This paper aims to compare the state-of-the-art MOP algorithms including Rapidly exploring Random Tree (RRT) and those implemented in the ROS navigation stack such as Dynamic Window Approach (DWA) local planner coupled with Dijkstra and A\* as global planners on a six-wheeled robot known as MOOR in simulation environment. The results reveal that all of these algorithms have been designed for a square shape footprint robot and thus have limitations for MOOR with a rectangular footprint shape.

### 1. INTRODUCTION

Path planning is a well-known problem in the areas of robotics (LaValle, 2006) and autonomous systems. Generally, path planning for a mobile robot refers to find the collision free, shortest, and smooth route from the specified start location to the desired goal location with optimal path cost. The operational environment of the robot is rarely static, and it often has many moving obstacles (Connell & La, 2017). This environment is called dynamic environment and is representative of the real world. The robot will need to decide how to proceed when one of these obstacles is obstructing its path (Connell & La, 2017) without a collision probability.

The algorithms for path planning are classified into different categories. In one case, they are global and local. Global algorithms use a priori knowledge e.g. a map of the environment to plan the path, and are thus applicable to planning in static environments. Local algorithms by the movement of the robot through the environment, revising the path based on environmental changes (Samadi & Othman, 2013).

This paper looks to an algorithm to path planning the surveying robot known as MOOR. A robot with six wheels, which includes the control boards, battery and various sensors such as cameras, GPS, IMU, laser scanner and odometer. The robot programs are developed under open source middle-ware tools known as ROS. ROS consists of a set of useful sources by which the robot is capable of moving in a known environment, unknown and less known. It has a package known as the navigation stack (Franklin et al., 2013) to path planning. The aim of this research is to compare the MOP algorithms including RRT and those implemented in the ROS navigation stack such as DWA local planner coupled with Dijkstra and A\* as global planners on the MOOR in simulation environment. Figure 1 shows MOOR.

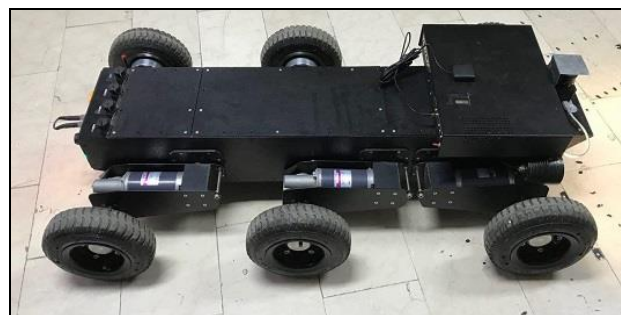


Figure 1. Six-wheeled surveying robot (MOOR)

The remaining sections of this paper are organized as follows. In the following section, the state-of-the-art MOP methods considered in this article are described and after that, in section 3 their implementation on MOOR is explained. The simulation results are shown and discussed in section 4. Finally, the concluding remarks are presented in section 5.

### 2. MATERIALS AND METHODS

Robot motion planning problem, as the name suggests, is to provide an efficient path between a given initial state and a target state, under the condition that the motion is constrained. After many years of development, motion planning has matured, and various algorithms and methods of determining a path have been researched. Path planning algorithms are mainly divided into the following categories: graph-based search methods, random sampling-based methods, potential field approaches and many other different planning methods.

In the following sections are provided the brief description of each kind of the path planning methods. A series of graph-based and sampling-based algorithms are discussed particularly includes Dijkstra and A\* as a global planner, DWA as a local planner and RRT algorithm.

## 2.1 Graph-Based Search Algorithms

Graph-based search methods are node-based algorithms (Yang et al., 2016); they are named by the reason that commonly deal with nodes and arcs weight information and is also called grid (network) algorithms (Zhan & Noon, 1998). These algorithms calculate the cost by exploring through the nodes in configuration space in order to find the minimal cost paths. Search algorithms are usually preferred to solve path planning problems of mobile robots with low-speed (Pivtoraiko, Knepper, & Kelly, 2009) in low-dimensional space (Daniel, Nash, Koenig, & Felner, 2010; Veeraswamy & Amavasai, 2006). Across the years, a number of classical graph-based algorithms have been developed. Recently some of grid-based methods treating nonholonomic constraints have been proposed (Pivtoraiko et al., 2009). The attention of this paper is on the Dijkstra and A\* algorithms.

### 2.1.1 Dijkstra Algorithm

Dijkstra (Dijkstra, 1959) is a simplest graph-based solution to find the less weight path from the starting point to the goal. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later. The weight of a path in a weighted graph is equal to the sum of the weights of its edges. This algorithm assumes that there are  $n$  vertices and the length of all pair of points that are connected together is known and there is at least one connection between the two vertices. The aim of this algorithm is to find a path with the shortest total length from the first point to the target point. In each stage, the algorithm chooses the shortest distance to the starting point and the distance of the other vertices are updated according to the distance  $x$ . For each vertex  $v$ :

$$dist(v) = \min(dist(v), dist(v) + w(x, v)) \quad (1)$$

where  $w(x, v)$  = weights of the arc between  $x$  and  $v$

### 2.1.2 A\* Algorithm

Peter Hart, Nils Nilsson and Bertram Raphael of Stanford Research Institute first described A\* algorithm in 1968. It is an extension of Edsger Dijkstra's algorithm. The A\* algorithm (Seo & Kim, 2013) is another graph-based path planning method to help the robot to find the optimal path in grid decomposed static grid maps. The environment with free space and obstacles is presented by a set of uniformed regular grids. The A\* uses a heuristic based Dijkstra algorithm to obtain the optimal result for the robot. The drawback is that the A\* uses uniformed grids representation which must allocate specific large amounts of memory for regions that may never be traversed or may not contain the obstacles. This drawback may affect the efficiency of the method. A\* algorithm shown in Figure 2. A\* tries to minimize the cost for the path by the formulized function as (Korkmaz & Durdu, 2018):

$$f(n) = g(n) + h(n) \quad (2)$$

where  $g(n)$  = cost of each move from the starting node

$h(n)$  = heuristic in which estimates future costs

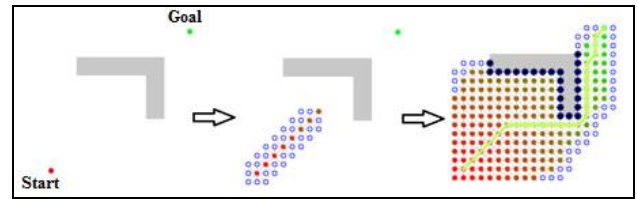


Figure 2. A\* algorithm

## 2.2 Dynamic Window Approach (DWA) algorithm

A velocity-based local path planning algorithm (Tomović) that first discretely sampled the control space and then simulated the robot movement in the path and chosen through the potential orders. The DWA algorithm is a more efficient algorithm because of smaller space sampling (Fox, Burgard, & Thrun, 1997). Figure 3 shows the DWA algorithm.

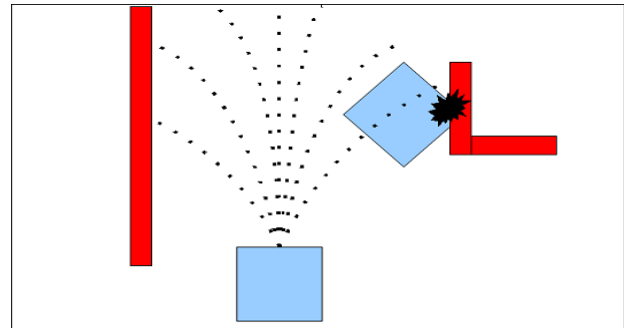


Figure 3. DWA algorithm

## 2.3 Sampling-Based Algorithms

Sampling-based algorithms all-share the merits of using an initial guess; they differ in how to do post processing to ensure completeness or optimum. The random initial guess ensures escaping of local minimum, and this kind of algorithms does not rely much on environmental representation. Sampling-based Planning (SBP) approaches are appropriate for planning in high dimensional spaces (Pivtoraiko et al., 2009). In these approaches, major limitation of is their slow convergence rate (Noreen, Khan, & Habib, 2016). LaValle et al. focused mainly on sampling based algorithms.

### 2.3.1 Rapidly Exploring Random Tree Algorithm

Rapidly Exploring Random Tree method is first proposed by LaValle (LaValle, 1998), that is a randomized path planning based on sampling algorithms that taking an initial point as the root node and incrementally generations a roadmap tree from samples (leaf nodes) drawn randomly and to quickly explore the a large area of the configuration space (Karova et al., 2015). When leaf nodes in the random tree contain the goal point or enter the goal region, a path from the initial point to the goal point can be found (Xinyu, Xiaojuan, Yong, Jiadong, & Rui, 2019). RRT is suitable algorithm to solve path planning problems under holonomic and non-holonomic constraints (Korkmaz & Durdu, 2018). Figure 4 shows the RRT algorithm.

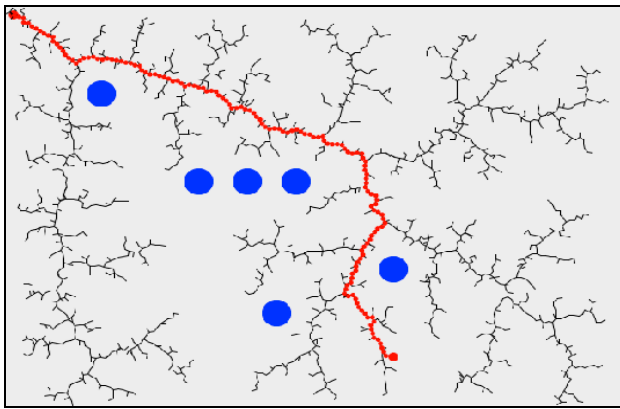


Figure 2. RRT algorithm

### 3. IMPLEMENTATION

The cited algorithms in the previous section were implemented on MOOR with respect to its characteristics in simulation environment in this section. This experiment was run in the VMware Workstations Ubuntu 14.04 on a computer with an Intel Core i5-720M CPU and 4GB RAM.

#### 3.1 Running on simulation

Simulation is provided for MOOR<sup>1</sup>, which provides a suitable environment for evaluating the performance of the MOP algorithms. The simulation is developed based on Gazebo on ROS and has all the robot sensors. In ROS, URDF tool used for modeling a robot and navigation stack used for moving that robot in the simulated environment (Gupta, Umrao, & Kumar). Gazebo has been used as a virtual world for the robot. Modeled MOOR with six motor actuated wheels on the sides and a Hokuyo 2D laser scanner in front of the robot to receive information about the environment during its motion as shown in Figure 5. Wheels are equipped with wheel encoders to obtain odometry information of the robot. Created 3D world in Gazebo simulator as shown in Figure 6. LiDAR<sup>2</sup> sensor is commonly used in robot navigation to create a map of environment (Gupta et al.). MOOR uses a 180° LiDAR sensor to get wide view of the environment. Figure 7 shows mapping in simulation environment with MOOR.

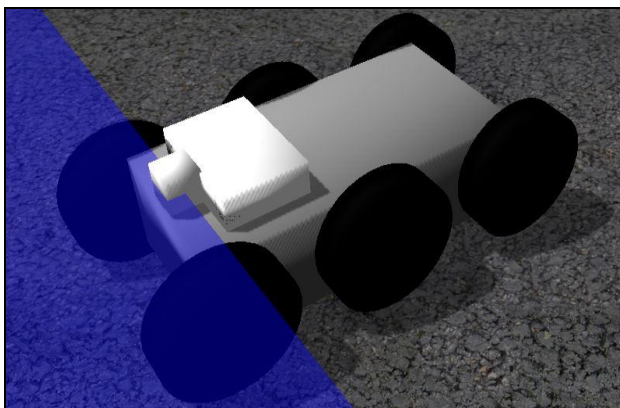


Figure 5. Robot modelled using ROS URDF



Figure 6. Simulated environment in Gazebo

ROS Navigation stack is a set of packages built with the aim of navigation of a mobile robot in an environment. It includes data of odometry, sensors and a goal position and outputs velocity commands<sup>3</sup> to move the robot. Navigation stack requires a map of the environment that the planning is performed (Gupta et al.). Different ready-made ROS packages are used in the implementation for mapping and path planning. The map of the environment illustrated in Figure 8 is obtained by implementing ROS gmapping package. Gmapping is a type of SLAM (Simultaneous Localization and Mapping) algorithm which is a powerful technique in order to mapping from the environment (Korkmaz & Durdu, 2018). Odometry and laser data are used as inputs of gmapping during 2D grid map creation. In the map, grey areas denote the obstacles detected by lidar sensor and white fields show free areas where the available path to move the robot.

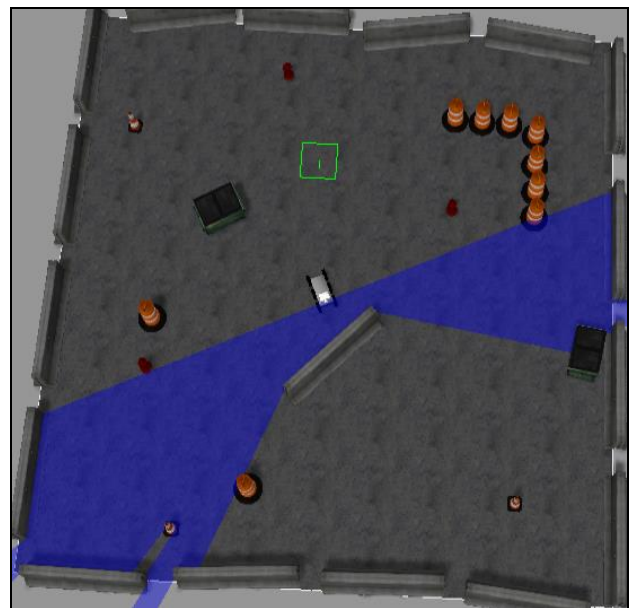


Figure 7. Mapping by MOOR in simulation environment

<sup>1</sup><https://github.com/hosseininaveh/Moor>

<sup>2</sup> Light Detection and Ranging

<sup>3</sup> Cmd\_vel



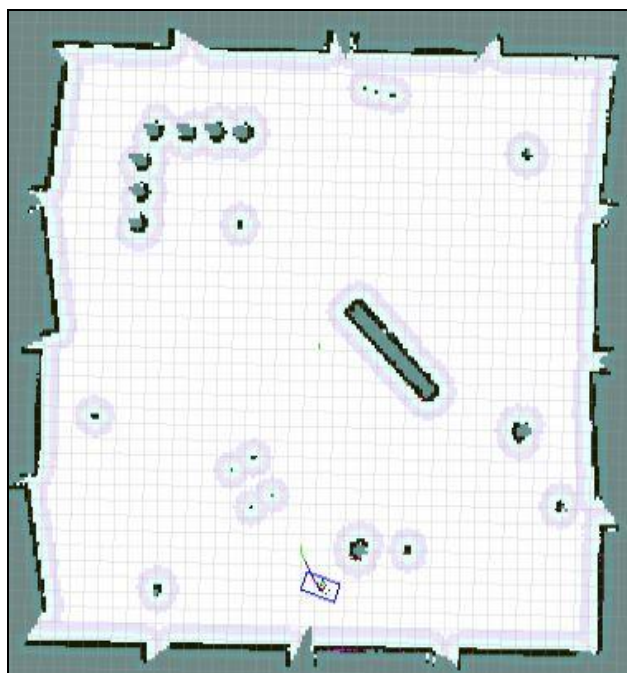


Figure 8. Mapping by gmapping algorithm

After mapping and localization are successfully done, move base package is used to accomplish navigation. This package provides move commands to the robot and also includes two costmap.yaml files for global and local planner (Gupta et al.). Nav\_core package maintains three global planner algorithms. The global planner is based on Navfn used the Dijkstra algorithm and is based on Global Planner used A\* algorithm. The local planner is also based on DWA.

For the implementation of RRT on MOOR, the rrt\_exploration package<sup>4</sup> has been used. The package has five different nodes including: global RRT frontier point detector node, local RRT frontier detector node, OpenCV-based frontier detector node, filter node and assigner node (Umari & Mukhopadhyay, 2017). The connection between these nodes is shown as a flowchart in Figure 9.

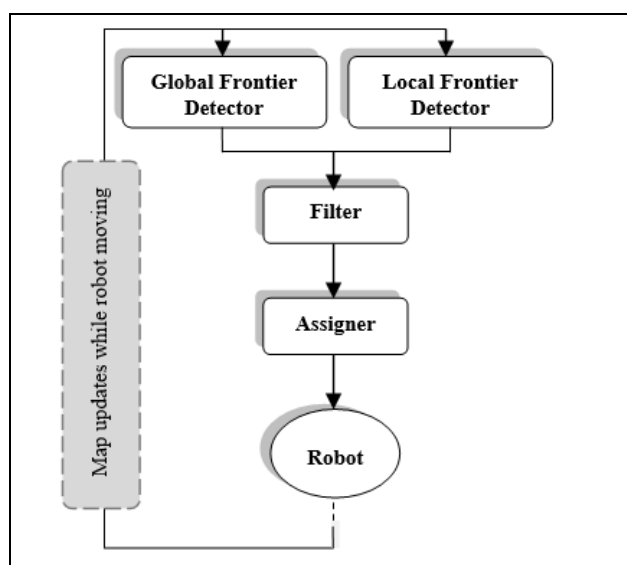


Figure 9. Connection between RRT package's nodes (Umari & Mukhopadhyay, 2017)

#### 4. RESULTS

Set of experiments has been conducted to evaluate the performance of the MOP algorithms. In order to implementation path planning algorithms on MOOR, we simulated robot in a virtual environment with zero-degree-of-freedom. Rviz (ROS Visualization Tool) used for visualization of robot's motion in the world. We built the simulation environment of the robot in Gazebo based on ROS and displayed it in Rviz. This section discusses the performance of the RRT, DWG, Dijkstra and A\* and compares the results. The ROS navigation stack is used to autonomously navigate MOOR in the simulation environment. Figure 10 and 11 show the simulation results of performance of Dijkstra and A\* algorithms as global planner with DWA as local planner. In Figure 12, RRT algorithm performance result has been represented.

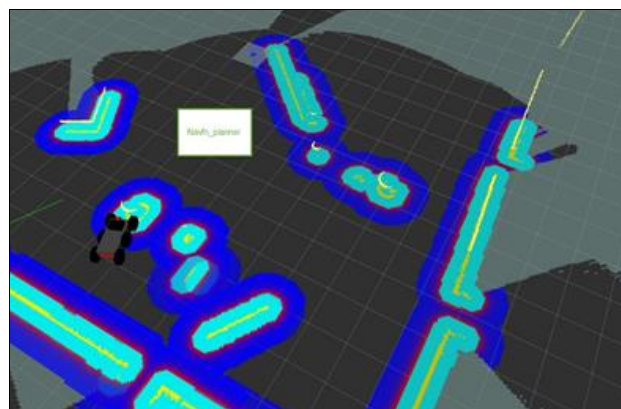


Figure 10. Performance of Dijkstra algorithm on a cost map

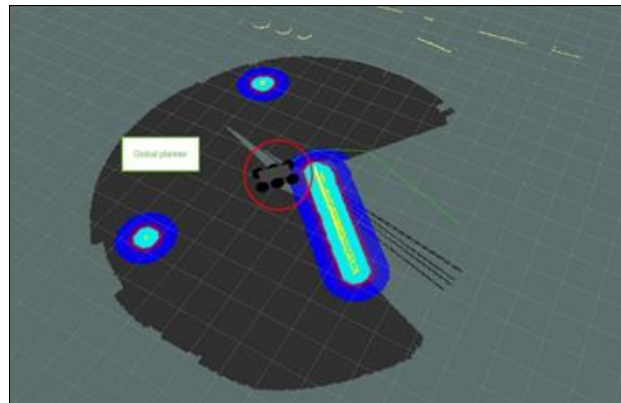


Figure 11. Performance of A\* algorithm on a cost map

<sup>4</sup> [https://github.com/hasauino/rrt\\_exploration](https://github.com/hasauino/rrt_exploration)

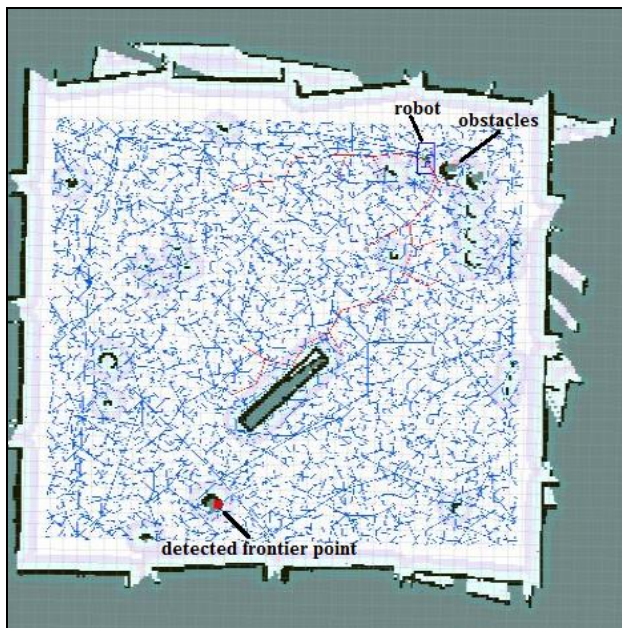


Figure 12. Performance of RRT on simulated MOOR

Simulation results show that surveying robot was unable to successfully pass the obstacles in the environment and couldn't change its path to reach the goal, i.e. these algorithms couldn't achieve good performance on rectangular robots.

## 5. CONCLUSION

With development of robotics, path planning is applicable to robots and many other fields. Therefore, it is necessary to study path planning algorithms. This paper presented the implementation and compare of path planning based on MOP algorithms and ROS navigation stack on the six-wheeled robots known as MOOR in simulation environment. The results reveal that all of these algorithms have been designed for a square shape footprint robot and thus have limitations for MOOR with a rectangular footprint shape.

## REFERENCES

- Connell, D., & La, H. M. (2017). Dynamic path planning and replanning for mobile robots using RRT. Paper presented at the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC).
- Daniel, K., Nash, A., Koenig, S., & Felner, A. (2010). Theta\*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, 39, 533-579.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269-271.
- Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1), 23-33.
- Gupta, A., Umrao, S., & Kumar, S. Optimal Path Planning in a Dynamic Environment.
- Karova, M., Zhelyazkov, D., Todorova, M., Penev, I., Nikolov, V., & PETKOV, V. (2015). Path planning algorithm for mobile robot. Paper presented at the Proc. Int. Conf. Appl. Comput. Sci.
- Korkmaz, M., & Durdu, A. (2018). Comparison of optimal path planning algorithms. Paper presented at the 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET).
- LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.
- LaValle, S. M. (2006). *Planning algorithms*: Cambridge university press.
- Noreen, I., Khan, A., & Habib, Z. (2016). Optimal path planning using RRT\* based approaches: a survey and future directions. *Int. J. Adv. Comput. Sci. Appl*, 7(11), 97-107.
- Pivtoraiko, M., Knepper, R. A., & Kelly, A. (2009). Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3), 308-333.
- Samadi, M., & Othman, M. F. (2013). Global path planning for autonomous mobile robot using genetic algorithm. Paper presented at the 2013 International Conference on Signal-Image Technology & Internet-Based Systems.
- Seo, D. J., & Kim, J. (2013). Development of autonomous navigation system for an indoor service robot application. Paper presented at the 2013 13th International Conference on Control, Automation and Systems (ICCAS 2013).
- Tomović, A. Path Planning Algorithms For The Robot Operating System.
- Umari, H., & Mukhopadhyay, S. (2017). Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. Paper presented at the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- Veeraswamy, A., & Amavasai, B. (2006). Optimal path planning using an improved A\* algorithm for homeland security applications. Paper presented at the Proceedings of the 24th IASTED international Conference on Artificial intelligence and Applications.
- Xinyu, W., Xiaojuan, L., Yong, G., Jiadong, S., & Rui, W. (2019). Bidirectional Potential Guided RRT\* for Motion Planning. *IEEE Access*, 7, 95034-95045.
- Yang, L., Qi, J., Song, D., Xiao, J., Han, J., & Xia, Y. (2016). Survey of robot 3D path planning algorithms. *Journal of Control Science and Engineering*, 2016, 5.
- Zhan, F. B., & Noon, C. E. (1998). Shortest path algorithms: an evaluation using real road networks. *Transportation science*, 32(1), 65-73.