A Proposed Trajectory Planning Algorithm for Mobile Robot Navigation Based on A* Algorithm

Chapter · January 2020

DOI: 10.4018/978-1-7998-0137-5.ch015

2 authors:

Sahin YILDIRIM
Erciyes Üniversitesi

Sertaç Savas
Erciyes Üniversitesi

Chapter 15
A Proposed Trajectory Planning Algorithm for Mobile Robot Navigation Based on A* Algorithm

Şahin Yıldırım
Erciyes University, Turkey

Sertaç Savaş
https://orcid.org/0000-0001-8096-1140
Erciyes University, Turkey

ABSTRACT

This chapter proposes a new trajectory planning approach by improving A algorithm, which is a widely-used, path-planning algorithm. This algorithm is a heuristic method used in maps such as the occupancy grid map. As the resolution increases in these maps, obstacles can be defined more precisely. However, the cell/grid size must be larger than the size of the mobile robot to prevent the robot from crashing into the borders of the working environment or obstacles. The second constraint of the algorithm is that it does not provide continuous headings. In this study, an avoidance area is calculated on the map for the mobile robot to avoid collisions. Then curve-fitting methods, general polynomial and b-spline, are applied to the path calculated by traditional A* algorithm to obtain smooth rotations and continuous headings by staying faithful to the original path calculated. Performance of the proposed trajectory planning method is compared to others for different target points on the grid map by using a software developed in Labview Environment.*

DOI: 10.4018/978-1-7998-0137-5.ch015

INTRODUCTION

Scientific studies on mobile robots have increased considerably in recent years. These studies can be summarized in four main sections; localization, mapping, path or trajectory planning and control. The path planning topic from these four main headings is the main theme of this paper. Over the years, many methods have been proposed for path planning. Calculation time, path length and path characteristics are the focus of improvements in path planning.

Many different algorithms have been developed for the mobile robot path planning task and the improvements for these algorithms are focused on factors such as calculation time, path length and path characteristics. In addition, these algorithms are mainly divided into classical methods and heuristic methods. A survey (Mac, Copot, Tran, & De Keyser, 2016) was concentrated on heuristic algorithms based on neural network, fuzzy logic, nature-inspired algorithms and hybrid algorithms. Also Buniyamin et al. (Buniyamin, Ngah, Sariff, & Mohamad, 2011) introduced a new approach called PointBug as a local path planning algorithm and compared the proposed approach with traditional Bug algorithms, Distbug and TangenBug, in terms of path length and reachability performance to the target. Čikeš et al. (Čikeš, Đakulović, & Petrović, 2011) presented a comparative study of D*, TWD* (Two Way D*) and E* algorithms on the occupancy grid map of the environment in terms of path characteristics, searching time and the number of iterations. Subramanian et al. (Subramanian, Sudhagar, & RajaRajeswari, 2014) presented Breadth First Search (BFS) algorithm as a path planning method to search for the shortest path faster than heuristic methods. Also in another study they (Subramanian, Sudhagar, & Rajarajeswari, 2015) presented a dynamic pathfinding approach in an unknown environment using vector field histogram for obstacle avoidance and A* algorithm for a heuristic path planning.

In addition, many studies have been made on A* algorithm which is a basic but powerful path planning approach and various improvements are presented to optimize the algorithm. Guruji et al. (Guruji, Agarwal, & Parsediya, 2016) proposed a method to provide efficiency in processing time of A* algorithm. Yin and Yang (2013) proposed a multi-path algorithm based A* comparing with a K-Dijkstra-based algorithm to use in route navigation systems. Pala et al. (Pala et al., 2013) proposed a path planning algorithm called HCTNav which is focused on minimum use of computing and energy resources and they compared their method with Dijkstra's algorithm.

The A* path planning algorithm is not only used for wheeled mobile robots. The algorithm can be also used for navigation planning of water and air vehicles. For this reason, studies on the development of the method have been made for various platforms and tasks. Masudur Rahman Al-Arif et al. (Al-Arif, Ferdous, & Nijami, 2012) proposed a path planning algorithm using artificial intelligence technique considering to be used for a water vehicle during a rescue operation and presented a comparative study of Dijkstra and A* algorithms. Tseng et al. (Tseng, Liang, Lee, Chou, & Chao, 2014) proposed A* algorithm to plan flight path of a civil unmanned aerial vehicle to provide a better quality on 3G communication during flight. Cheng et al. (Cheng, Liu, & Yan, 2014) presented an improved hierarchical A* algorithm that takes into account shortest distance and time to calculate optimal parking path in a large parking guidance system.

This paper is organized as follows: In Section 2, occupancy grid map and A* algorithm are introduced. In Section 3, calculating avoidance area on grid map and curve fitting process for calculated path points are explained. After that, simulation results for the proposed approach compared with other methods are given in Section 4. Finally, discussions on the performance of improvements and suggestions for future studies are made in Section 5.

OCCUPANCY GRID MAPS AND A* ALGORITHM

Occupancy Grid Maps

In mobile robot applications, it is very important to build a map that will be an abstract representation of the real environment. This representation should be defined as empty or non-empty areas in order to provide navigation of the mobile robot. Based on this basic idea occupancy grid maps (Gonzalez-Arjona, Sanchez, López-Colino, de Castro, & Garrido, 2013; Milstein, 2008; Satapathy & Kumar, 2016) are used to represent the working environment as a 2D array of cells/grids. Since each cell only provides a square representation, occupancy grid maps can not be absolutely accurate. But this problem could be overcomed insofar by choosing a small enough cell size.

Occupancy grid mapping algorithms are based on probabilistic approaches to determine the occupancy of each cell by using the depth data which can be collected from various sensors on mobile robot with the assumption that the instant position of the robot is known. So the map is generated and updated relative to the robot's estimated position. Occupancy grid maps are represented by using intensity graphs and the intensity value of a cell is 1 or 100, respectively white if the cell is free to move or black if it is not free because of an obstacle. An illustration of an occupancy grid map and the value of each cell is shown in Figure 1.

A* Algorithm

A* is a best-first algorithm widely used for mobile robot applications to search for a least-cost path through a static map from a start point to a **goal point**. The algorithm is invented in 1968 by the researchers working on path planning problem of "Shakey the Robot" which is the first general-purpose mobile robot (Brachman & Levesque, 2004). The method is a combination of Dijkstra and Herustic algorithm. The evaluation function (Duchoň et al., 2014) used in this algorithm is given by;

$$F\left(v\right) = G\left(v\right) + H\left(v\right) \tag{1}$$

Figure 1. An illustration of an occupancy grid map and the value of each cell

1	100	100	100	100	100	100	100	100
1	1	1	1	1	100	100	100	100
100	1	1	1	1	1	100	100	100
100	100	100	1	1	1	1	1	100
100	100	100	100	100	1	1	1	1

where $v = \left(x_v, y_v\right)$ is recent expanding cell, $F\left(v\right)$ is the estimated minimum cost of the cell, $G\left(v\right)$ is the actual cost of the length from the start cell to the current cell, $H\left(v\right)$ is the herustic estimation of minimum cost of a path form the current cell to the goal cell. The algorithm uses two list arrays called open list and closed list to calculate the cost values for the neighbors of the current cell and updates the lists during calculations. So the calculations are expanded for the following cell in the sequence with the lowest value of $F\left(v\right)$. A basic example for A* algorithm implementation to reach cell B from cell A is shown in Figure 2.

MODIFICATIONS ON GRID MAP AND CURVE FITTING PROCESS

Calculating Avoidance Area for Grid Map

A square platform, in dimensions of 4.5x4.5 meter, is considered as the working environment to generate the occupancy grid map for using to implement and compare proposed path planning algorithm with other methods. The grid map, in dimensions of 150x150 cells, is generated by manuel selection of each cell value by a software developed in Labview. Each cell in the grid map represents a square area of 3x3 cm. Also some square obstacles are considered to see the heading perfomance of algorithms.

After the grid map is built, a software is developed to calculate the avoidance area to prevent the mobile robot from crashing into environment borders and obstacles. The flowchart of the developed software is given in Figure 3 for good understanding. In this flowchart, $V\left(i, j\right)$ represents the value of each cell as 1 or 100. Each cell of the $M \times N$ size grid map is examined in terms of neighboring cells and the cells found to be near the obstacle or platform boundary are marked as full, and this marking is repeated *W* times for the entire map. *W* represents the width of the avoidance area, chosen as the width of the mobile robot. As a mobile robot, The "National Instruments Robotics Starter Kit 2.0" robot is chosen in order to determine the calculation parameters of proposed path planning to be used in simula-

Figure 2. A basic diagram of A algorithm implementation.*

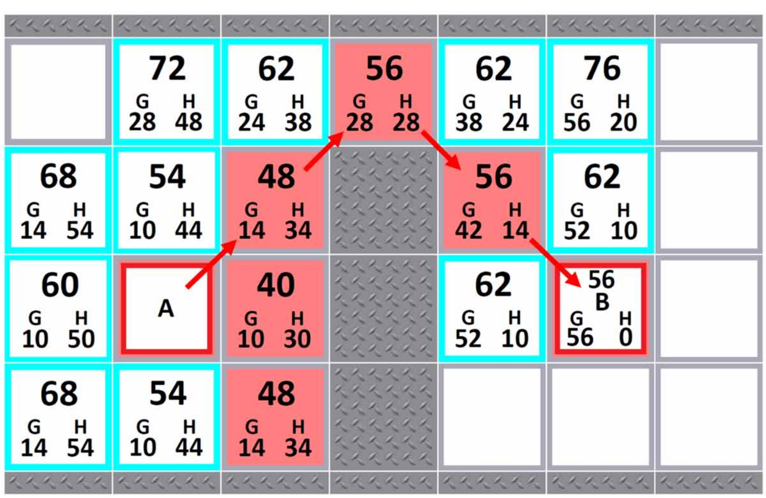
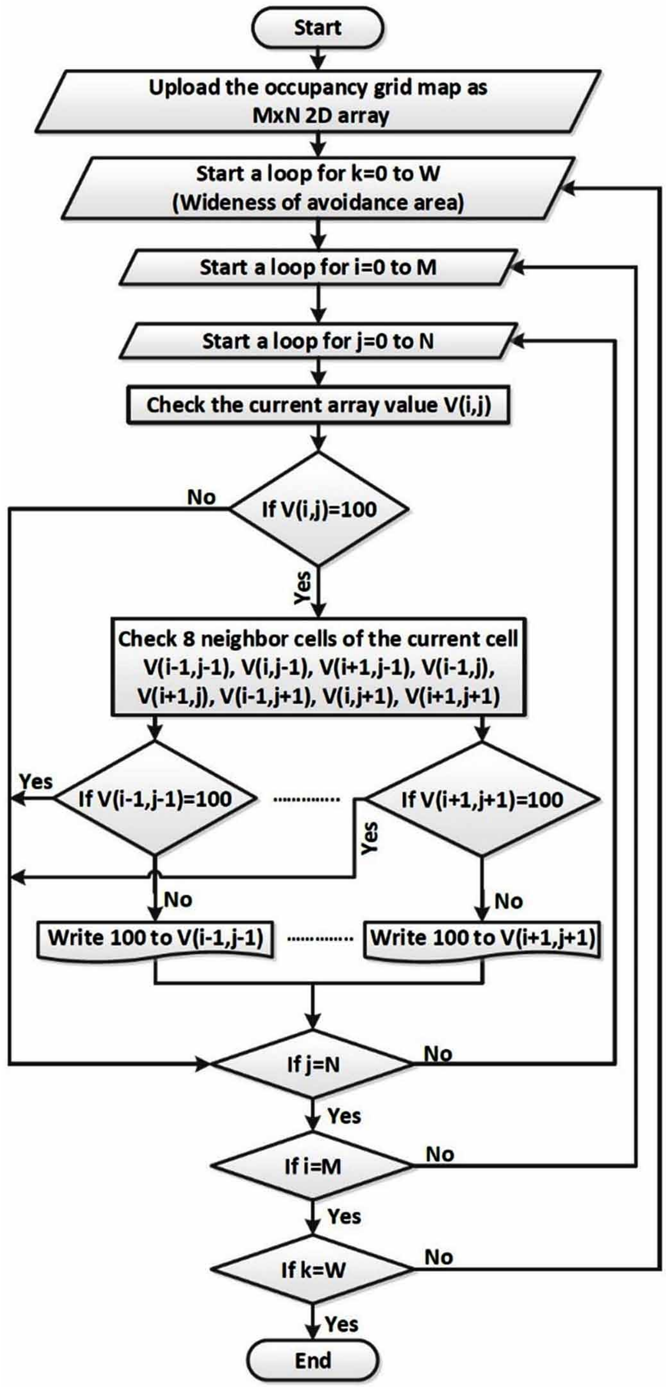


Figure 3. Flowchart of the software that calculates the avoidance area.



tion studies. The mobile robot is shown in Figure 4. Assuming that the width of the mobile robot (*L*) is 33 cm, the *W* value is chosen as 11 to provide an avoidance area as much as the width of the mobile robot. The occupancy grid map generated and the avoidance area calculated for this map are shown in Figure 5.

Curve Fitting for Calculated Path Points

In the study two curve fitting methods, general polynomial and b-spline, are applied to the path points *(X, Y)* calculated by A* algorithm on the occupancy grid map with avoidance area. General Polynomial Fitting process fits a data set to a polynomial function of the general form described by the following equation using the Least Square, Least Absolute Residual or Bisquare method;

$$f\left(i\right) = \sum_{j=0}^{m} a_j x_i^j \tag{2}$$

where f is the output array as best fit, x is the input array X, a_j is the polynomial coefficient for each portion of equation and m is the polynomial order ("General Polynomial Fit VI - LabVIEW 2014 Help," 2014).

Figure 4. "National Instruments Robotics Starter Kit 2.0" mobile robot.

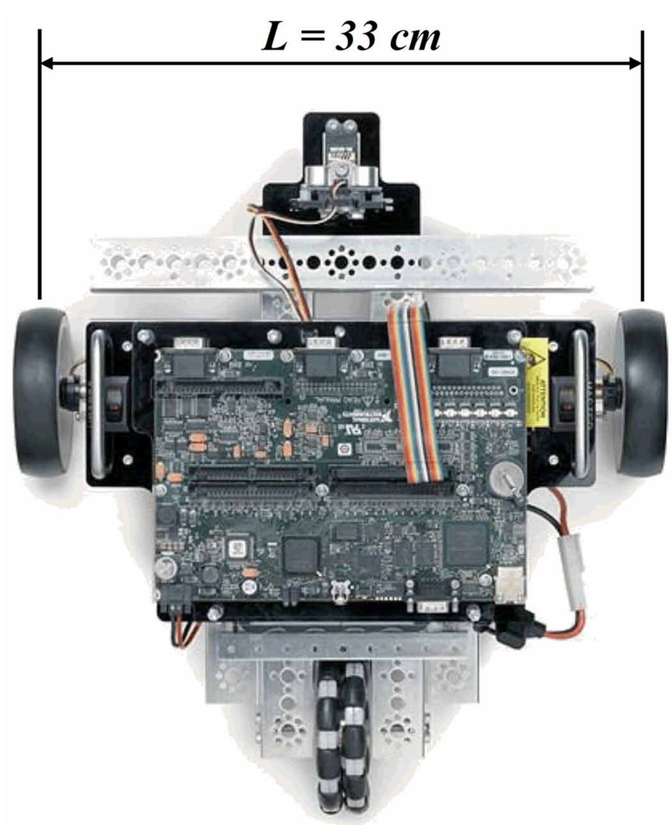
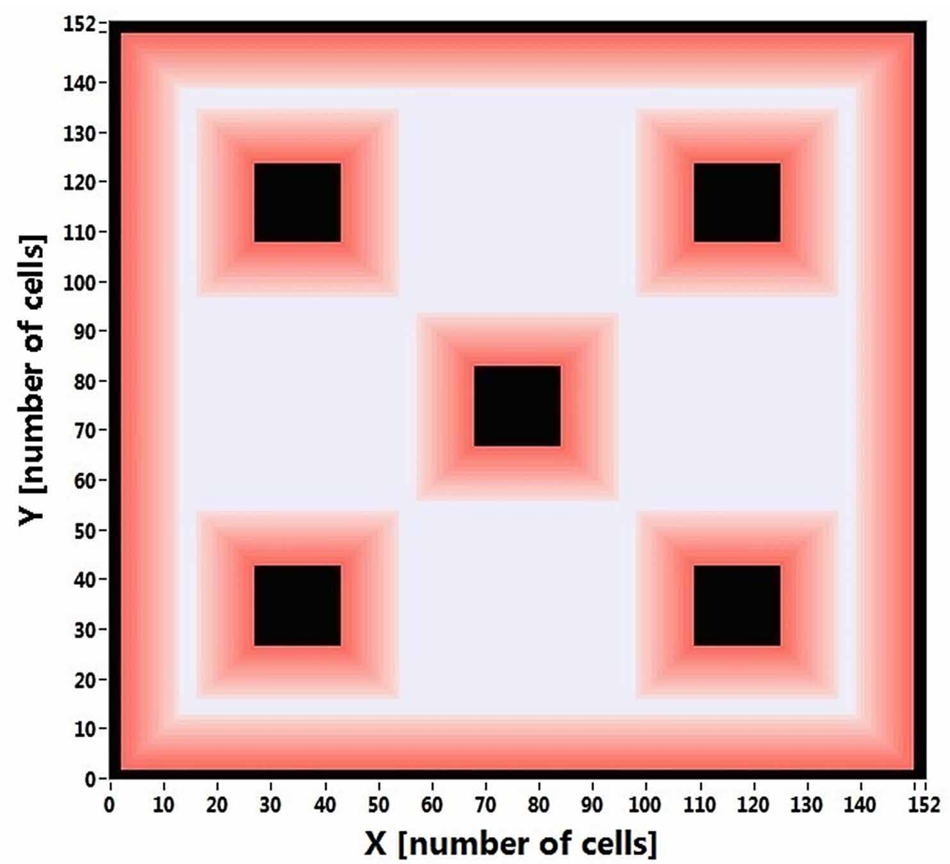


Figure 5. Occupancy grid map with avoidance area calculated.



B-Spline is the other method applied to the path points calculated. The B-Spline Fitting process fits a data set by minimizing the residue according to the following equation using the equally spaced, chord length or centripetal method which computes the interim knot vector;

$$\frac{1}{N}\sum_{i=0}^{N-1} w_i \cdot \left(x_i, y_i\right) - \left(x_i^{'}, y_i^{'}\right)^2 = \frac{1}{N}\sum_{i=0}^{N-1} w_i \cdot \left[\left(x_i - x_i^{'}\right)^2 + \left(y_i - y_i^{'}\right)^2\right] \tag{3}$$

where N is the length of Y, w_i is the i^{th} element of Weight, (x_i, y_i) is the i^{th} pair of the input sequences (X, Y), (x'_i, y'_i) is the i^{th} pair of best fit and the norm symbols (||) on both sides of the function compute the l_2 norm of a vector ("B-Spline Fit VI - LabVIEW 2011 Help," 2011).

SIMULATION RESULTS

A software is prepared to perform the simulations with Labview. This software is used to implement three tasks; calculating the avoidance area on the grid map, path planning by A* algorithm and applying curve fitting operations to the calculated paths. The interface of the developed software is shown in Figure 6.

After the avoidance area has been calculated for the grid map, a starting point (SP) and four target points marked A, B, C and D are identified on the map. The path is calculated from start point to target point marked A by using A* algorithm on the map. Then curve fitting methods are applied to the calculated path to determine the fitting parameters. Firstly, polynomial curve fitting is performed for 30 different combinations with different parameters and compared in terms of error values. The 6 combinations with the lowest error values are given in Table 1 and the parameters in row 4 with the lowest MSE value are chosen for the comparison. In the Labview, the equation of the polynomial function obtained when applying the curve fitting with these parameters to the x and y values of the points in the path calculated with A* is given below;

$$\begin{aligned} y &= -68{,}06 + 28{,}53 \cdot x - 2{,}75 \cdot x^2 + 0{,}14 \cdot x^3 - 0 \cdot x^4 \\ &+ 6{,}23 \cdot 10^{-5} \cdot x^5 - 5{,}16 \cdot 10^{-7} \cdot x^6 + 1{,}73 \cdot 10^{-9} \cdot x^7 \end{aligned} \tag{4}$$

Figure 6. The interface view of the developed software.

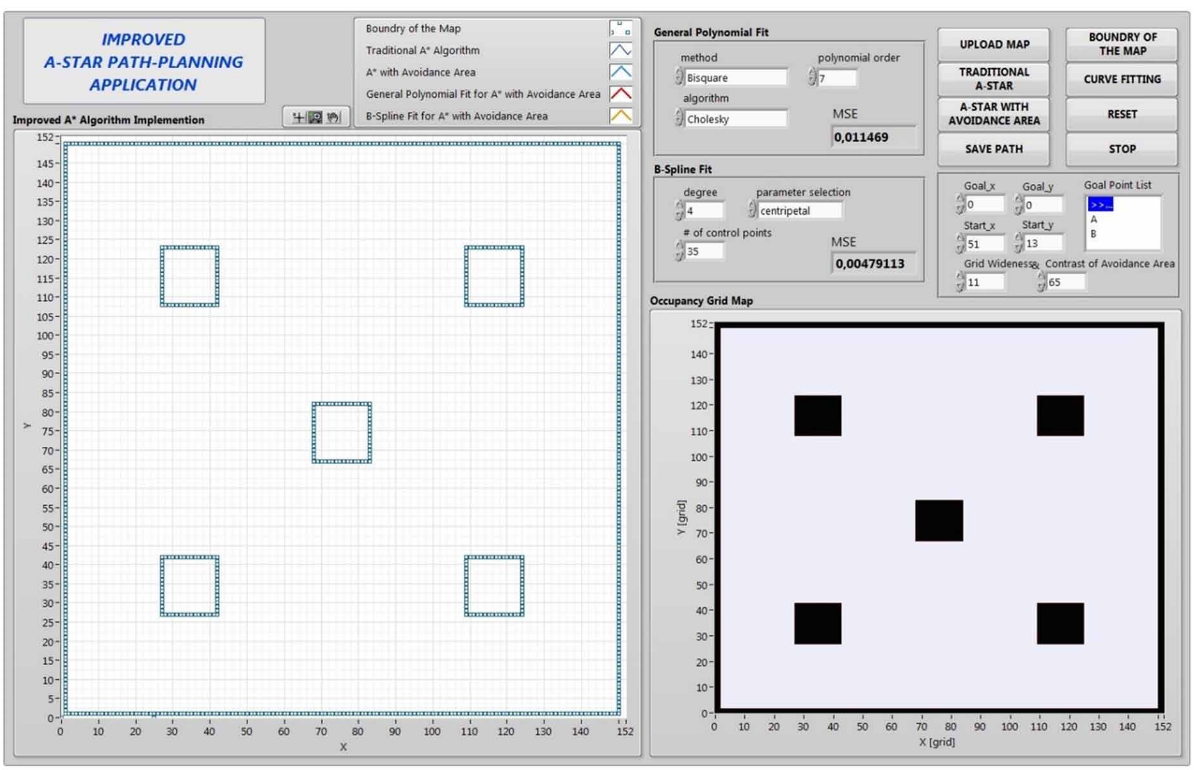


Table 1. Parameters and error values of general polynomial curve fitting for path points.

General Polynomial Fit							
	Fitting Method			Polynomial Order	MSE	Start Point Error (X-Axis)	End Point Error (X-Axis)
	Least Square	Least Absolute Residual	Bisquare				
1	X			8	0,213963	1,181257	-0,22097
2	X			9	0,150555	0,473597	0,486756
3		X		9	0,260558	0,659859	0,716724
4			**X**	**7**	**0,011469**	**1,450361**	**0,033156**
5			X	8	0,013018	1,509522	0,028266
6			X	9	0,017375	1,166607	0,189848

During the curve fitting process, it is selected 0.0001 as tolerance value, 1 as weight value for each input value and Cholesky as algorithm.

Then, B-spline curve fitting method is applied to the path planned by A* algorithm on the grid map with avoidance area. Curve fitting process is applied in 500 different combinations in terms of parameter, degree and control points, and during the curve fitting process, it is selected 1 as weight value for each input value. 6 combinations with the lowest error values are given in Table 2 and curve fitting parameters in row 5 with the lowest MSE value is chosen for comparison. The curve fitting parameters are determined for target point A and then these parameters are used for all other target points B, C, and D.

In the simulation study, four different path planning approaches have been compared with each other in terms of the paths calculated;

1. **Traditional A* Algorithm:** The path calculated by A* algorithm on the occupancy grid map without any avoidance area.
2. **A* with Avoidance Area:** The path calculated by A* algorithm on the occupancy grid map which has an avoidance area with the same width as the mobile robot.

Table 2. Parameters and error values of b-spline curve fitting for path points.

B-Spline Fit										
	Parameter Selection			Degree	Control Points	MSE	Start Point Error (X-Axis)	Start Point Error (Y-Axis)	End Point Error (X-Axis)	End Point Error (Y-Axis)
	Equally Spaced	Chord Length	Centripetal							
1	X			3	35	0,006377	0,004800	0,000000	0,063294	-0,000000
2		X		3	35	0,004740	0,001405	-0,000582	0,071522	-0,029625
3			X	3	35	0,005333	0,002642	-0,000500	0,067845	-0,012837
4	X			4	35	0,005495	0,000685	0,000000	0,021441	-0,000000
5		**X**		**4**	**35**	**0,004385**	**0,000161**	**-0,000067**	**0,025899**	**-0,010728**
6			X	4	35	0,004791	0,000338	-0,000064	0,023720	-0,004488

3. **General Polynomial Fit for A* with Avoidance Area:** The path obtained by applying general polynomial curve fitting method to the path which has calculated by A* algorithm on the occupancy grid map with avoidance area.
4. **B-Spline Fit for A* Avoidance Area:** The path obtained by applying b-spline curve fitting method to the path which has calculated by A* algorithm on the occupancy grid map with avoidance area.

The comparisons have been made for four target points on grid map. The performances of these four approaches for the target points marked A, B, C and D on the grid map are shown in Figure 7, Figure 8, Figure 9 and Figure 10, respectively.

As seen in Figure 7, if the target point is not closed too much with obstacles, the performance difference of the algorithms is not clear.However, the traditional A * algorithm could not provide a continuous trajectory by drawing a zigzag path in the last section.

Figure 7. Calculated paths for the target point marked A on the occupancy grid map

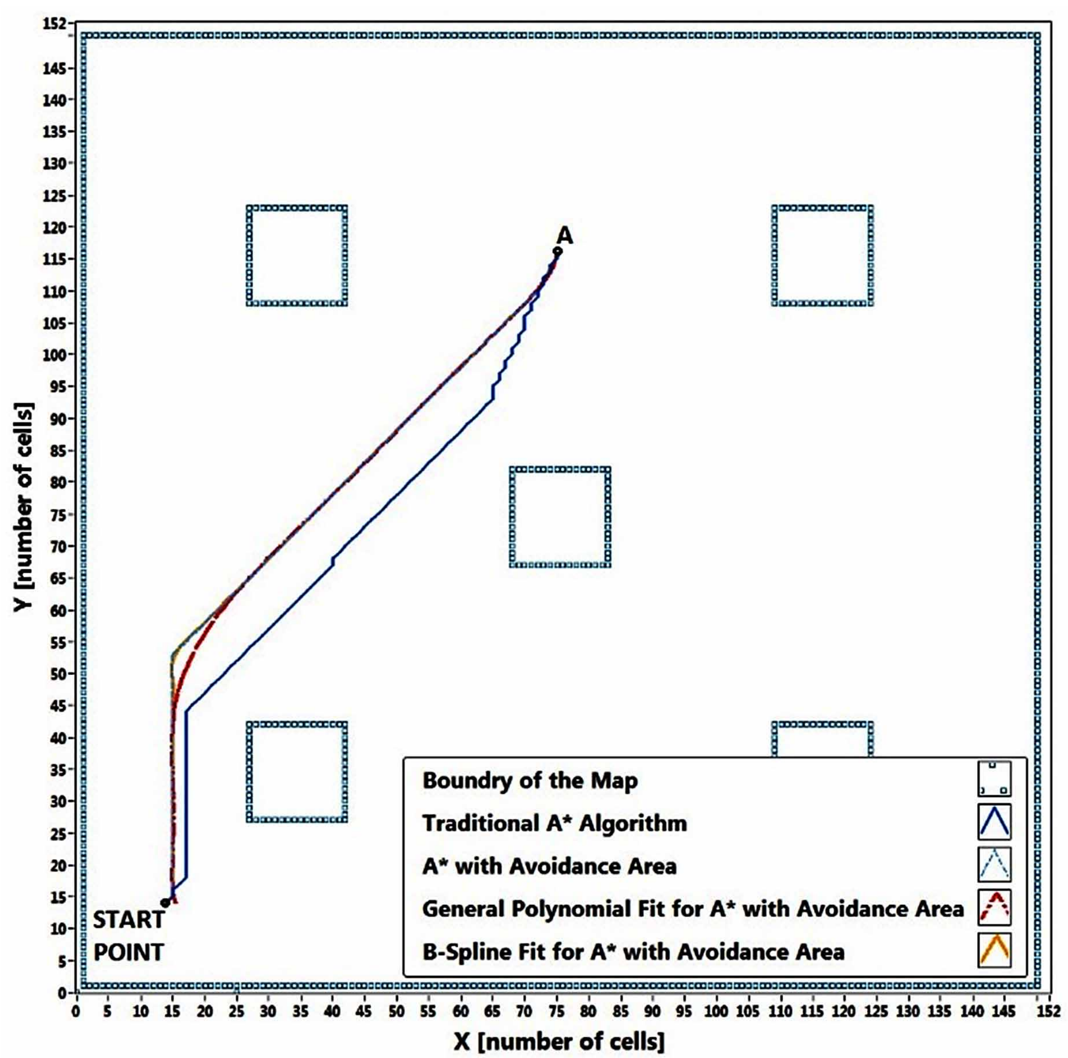
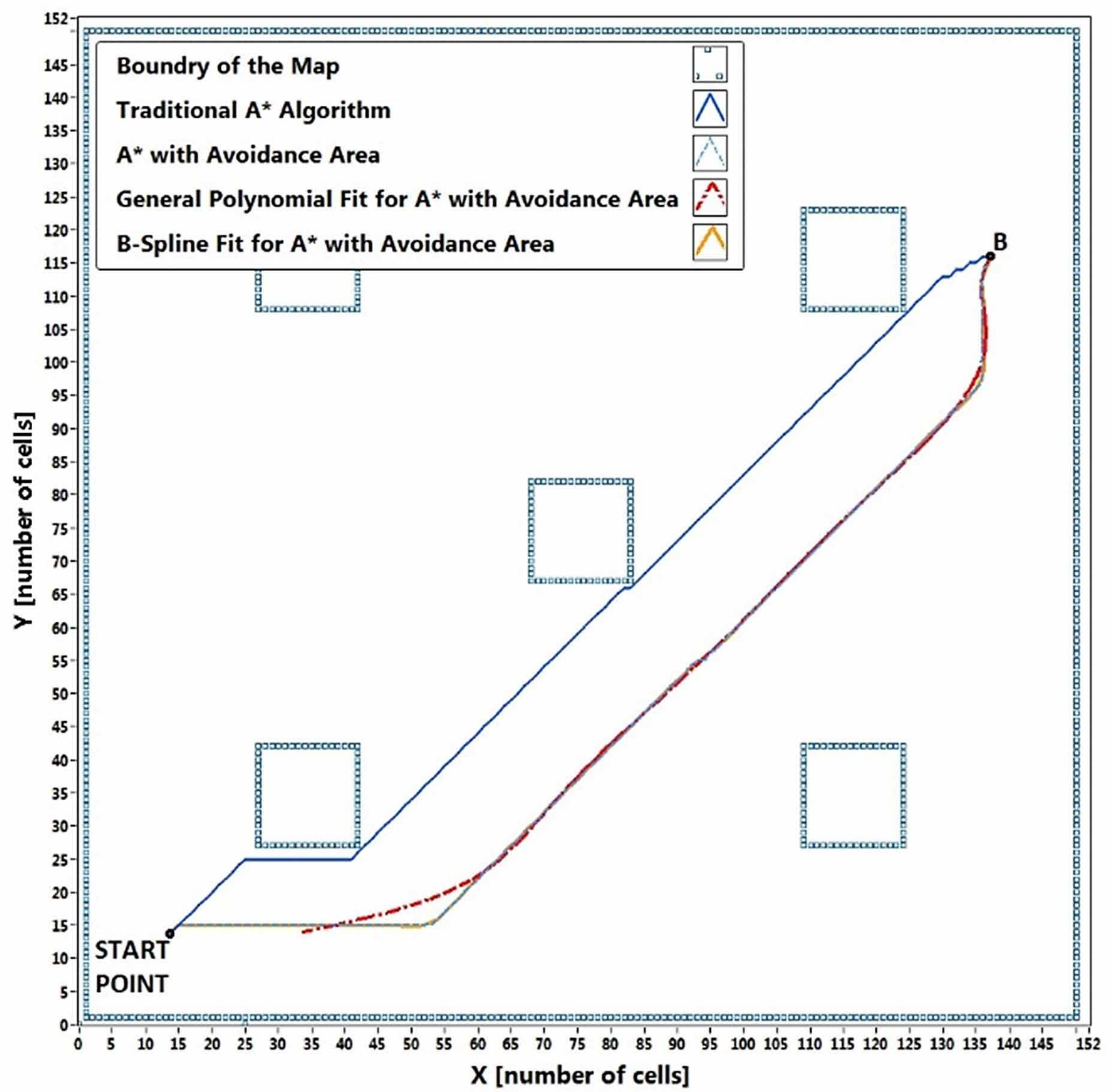
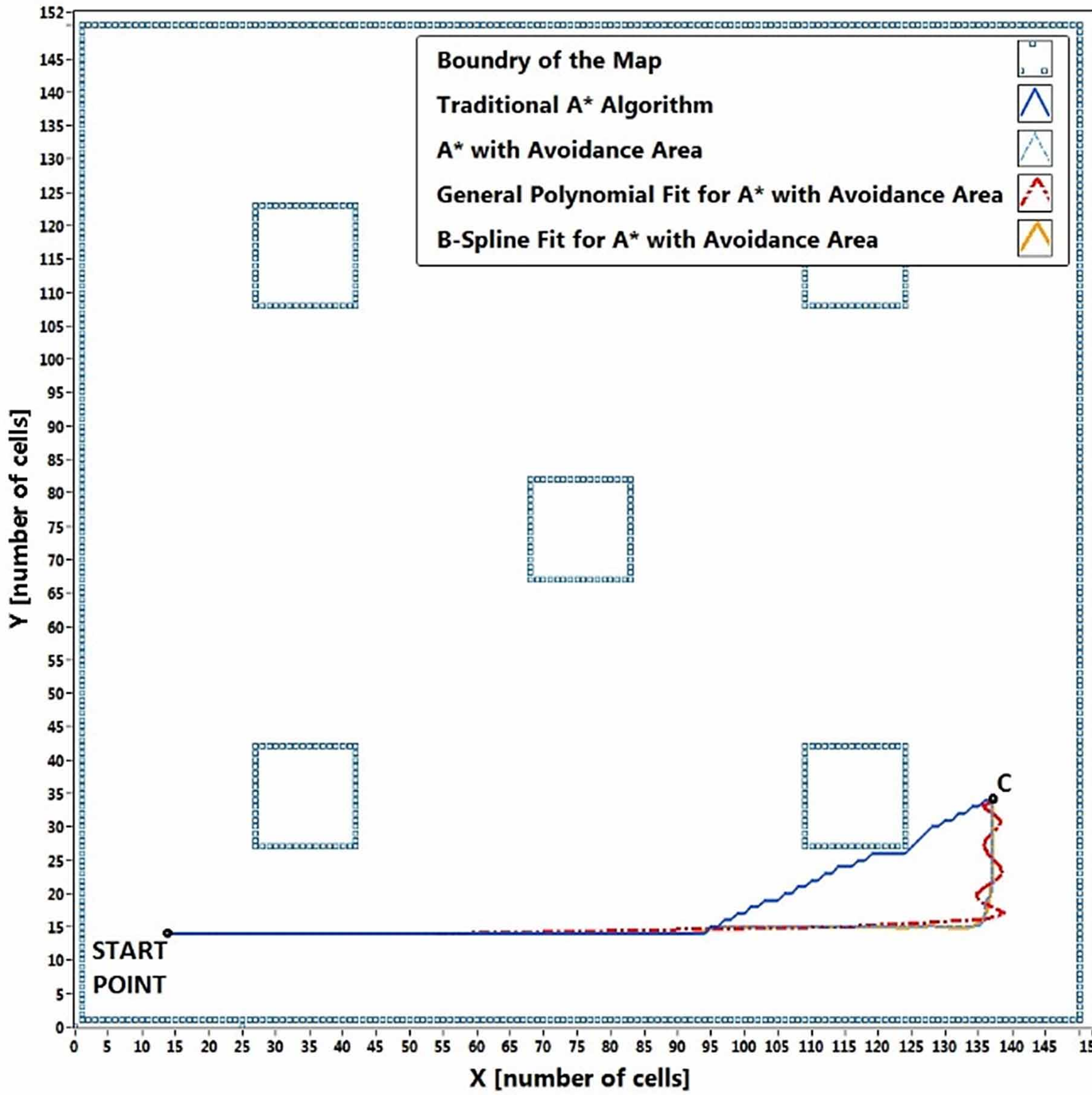


Figure 8. Calculated paths for the target point marked B on the occupancy grid map



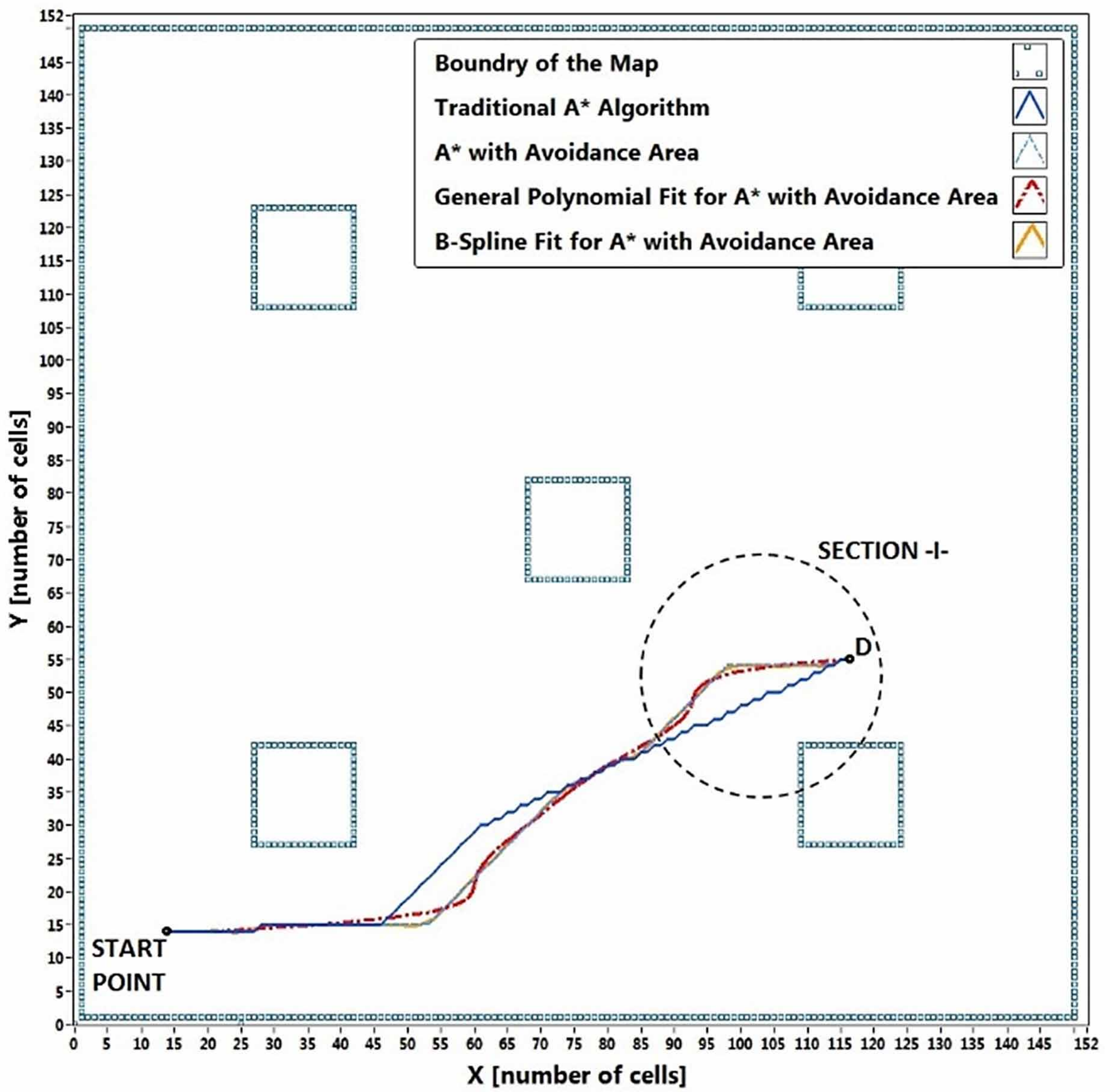
As can be seen in Figure 8 and Figure 9, the traditional A* algorithm plans a route very close to the obstacles, which can cause collisions without considering the width of the mobile robot. The trajectory obtained by applying the polynominal curve fitting performs well for the target point A in which the parameters are determined, but with these parameters a trajectory that can be used for other target points cannot be calculated.

Figure 9. Calculated paths for the target point marked C on the occupancy grid map



The performance of the comparative methods for rotation can be better seen in Figure 11 by referring to the Section-I in Figure 10. B-spline curve fitting applied trajectory shows the best performance. This performance can be defended by two criteria. The first criterion is that the calculated path does not hit any obstacles and map boundaries. The second criterion is that it plans a continuous trajectory with continuous headings in contrast to the traditional A* algorithm and others.

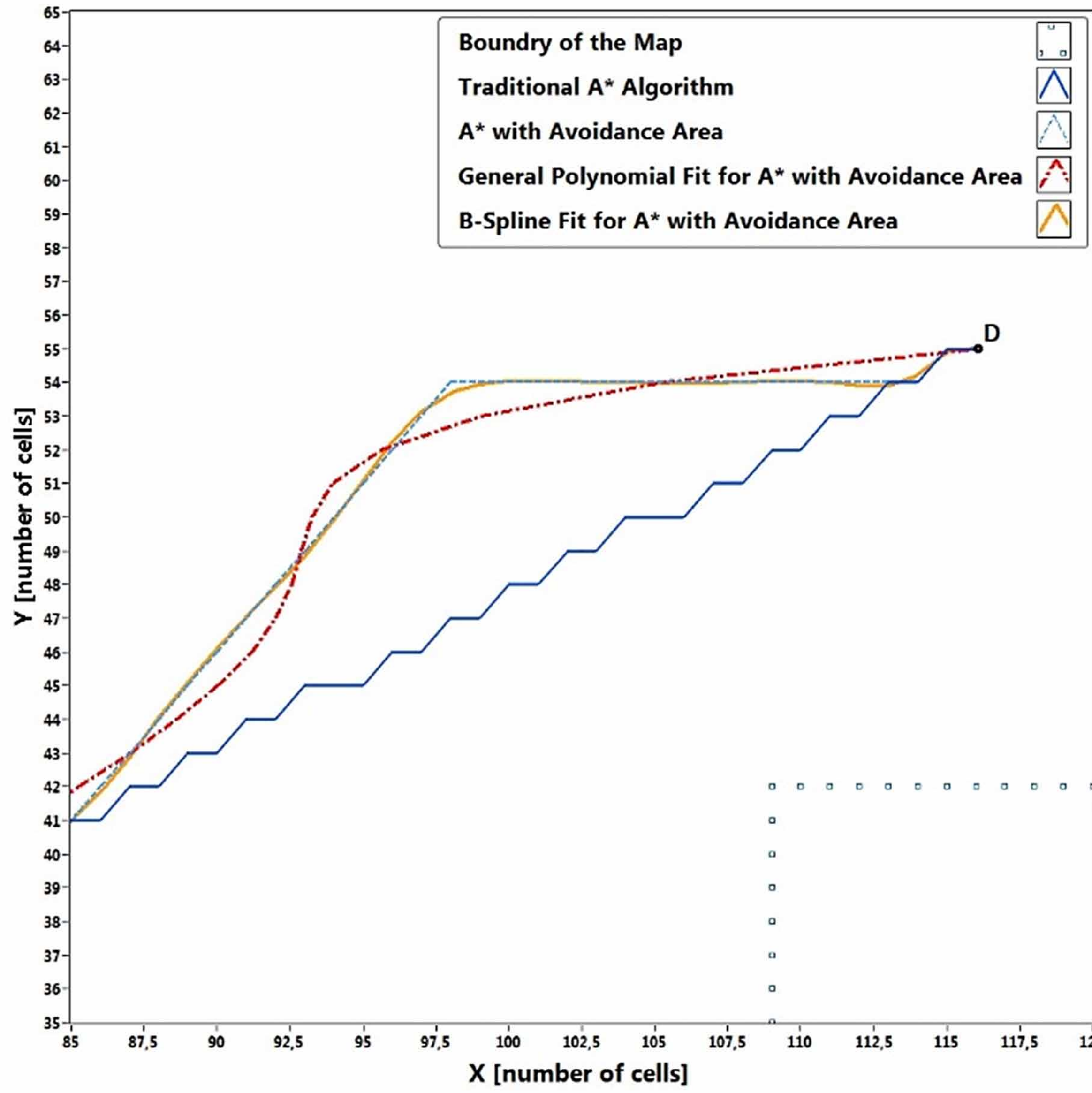
Figure 10. Calculated paths for the target point marked D on the occupancy grid map



CONCLUSION

Simulation studies have shown that the path calculated by traditional A* algorithm can not avoid collisions from borders of the obstacles or working environment. For this reason, an avoidance area is calculated in the width of the experimental mobile robot. But the traditional A* algorithm calculates a zigzag path without continuous headings even for the grid map with avoidance area. So curve-fitting methods, general polynomial and b-spline, are applied to the paths calculated by A* algorithm on this grid map with avoidance area. The path obtained by general polynomial curve fitting provides a suitable path on conditions which its parameters are determined, but it does not show the same performance under dif-

Figure 11. Rotational performance of the comparative methods - Section I in Figure 10



ferent conditions, especially for vertical movements. When B-spline curve fitting method is applied, it is possible to obtain smooth rotations by staying faithful to the path calculated by the A* algorithm on the grid map with avoidance area and thus continuous headings can be obtained.

Further research work will be an experimental investigation for real-time tracking performance of the mobile robot for trajectories calculated by the proposed algorithm. Also performance of the algorithm can be compared for different mobile platforms.

REFERENCES

Al-Arif, S. M. M. R., Ferdous, A. H. M. I., & Nijami, S. H. (2012). Comparative Study of Different Path Planning Algorithms: A Water based Rescue System. *International Journal of Computers and Applications*, *39*(5), 25–29. doi:10.5120/4817-7058

B-Spline Fit VI - LabVIEW 2011 Help. (2011). Retrieved from http://zone.ni.com/reference/en-XX/help/371361H-01/gmath/bspline_fit/

Brachman, R. J., & Levesque, H. J. (2004). *Planning Knowledge Representation and Reasoning* (pp. 305–325). San Francisco, CA: Morgan Kaufmann. doi:10.1016/B978-155860932-7/50100-X

Buniyamin, N., Ngah, W. A. J. W., Sariff, N., & Mohamad, Z. (2011). A Simple Local Path Planning Algorithm for Autonomous Mobile Robots. *International Journal of Systems Applications*, *Engineering & Development*, *5*(2), 151–159.

Cheng, L., Liu, C., & Yan, B. (2014, July). Improved Hierarchical A-star Algorithm for Optimal Parking Path Planning of the Large Parking Lot. Paper presented at the *2014 IEEE International Conference on Information and Automation (ICIA)*, Hailar, China. 10.1109/ICInfA.2014.6932742

Čikeš, M., Đakulović, M., & Petrović, I. (2011, October). *The Path Planning Algorithms for a Mobile Robot Based on the Occupancy Grid Map of the Environment - A Comparative Study.* Paper presented at the 2011 XXIII International Symposium on Information, Communication and Automation Technologies, Sarajevo, Bosnia.

Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., & Jurišica, L. (2014). Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Engineering*, *96*, 59–69. doi:10.1016/j.proeng.2014.12.098

General Polynomial Fit VI - LabVIEW 2014 Help. (2014). Retrieved from http://zone.ni.com/reference/en-XX/help/371361L-01/gmath/general_polynomial_fit/

Gonzalez-Arjona, D., Sanchez, A., López-Colino, F., de Castro, A., & Garrido, J. (2013). Simplified Occupancy Grid Indoor Mapping Optimized for Low-Cost Robots. *ISPRS International Journal of Geo-Information*, *2*(4), 959–977. doi:10.3390/ijgi2040959

Guruji, A. K., Agarwal, H., & Parsediya, D. K. (2016). Time-efficient A* Algorithm for Robot Path Planning. *Procedia Technology*, *23*, 144–149. doi:10.1016/j.protcy.2016.03.010

Mac, T. T., Copot, C., Tran, D. T., & De Keyser, R. (2016). Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, *86*, 13–28. doi:10.1016/j.robot.2016.08.001

Milstein, A. (2008). Occupancy Grid Maps for Localization and Mapping. In X.-J. Jing (Ed.), *Motion Planning*. InTech. doi:10.5772/6003

Pala, M., Osati, N., López-Colino, F., Sanchez, A., de Castro, A., & Garrido, J. (2013). HCTNav: A Path Planning Algorithm for Low-Cost Autonomous Robot Navigation in Indoor Environments. *ISPRS International Journal of Geo-Information*, *2*(3), 729–748. doi:10.3390/ijgi2030729

Satapathy, D., & Kumar, G. J. R. (2016). Path Planning Algorithm for a Mobile Robot – A Comparative Study in Different Environment. *IJCTA*, *9*(37), 453–459.

Subramanian, M. B., & Sudhagar, D. K., & RajaRajeswari, G. (2014). *Intelligent Path Planning Of Mobile Robot Agent By Using Breadth First Search Algorithm*. Paper presented at the 2014 IEEE International Conference on Innovations in Engineering and Technology (ICIET'14), Madurai, Tamil Nadu, India.

Subramanian, M. B., Sudhagar, D. K., & Rajarajeswari, G. (2015). Autonomous Mobile Robot Agent Path Planning Based on VFH Method Using Heuristic Based Optimal Algorithm. *International Journal of Applied Engineering Research*, *10*(6), 5189–5194.

Tseng, F. H., Liang, T. T., Lee, C. H., Chou, L. D., & Chao, H. C. (2014, August). *A Star Search Algorithm for Civil UAV Path Planning with 3G Communication.* Paper presented at the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan. 10.1109/IIH-MSP.2014.236

Yin, W., & Yang, X. (2013). A Totally Astar-based Multi-path Algorithm for the Recognition of Reasonable Route Sets in Vehicle Navigation Systems. *Procedia: Social and Behavioral Sciences*, *96*, 1069–1078. doi:10.1016/j.sbspro.2013.08.123