ITC 3/51
Information Technology and Control
Vol. 51 / No. 3 / 2022
pp. 499-514
DOI 10.5755/j01.itc.51.3.30016

A Fuzzy Logic Path Planning Algorithm Based on Geometric Landmarks and Kinetic Constraints

Received 2021/10/24 | Accepted after revision 2022/08/10

http://dx.doi.org/10.5755/j01.itc.51.3.30016

HOW TO CITE: Wang, J., Xu, Z., Zheng, X., Liu, Z. (2022). A Fuzzy Logic Path Planning Algorithm Based on Geometric Landmarks and Kinetic Constraints. *Information Technology and Control*, 51(3), 499-514. http://dx.doi.org/10.5755/j01.itc.51.3.30016

A Fuzzy Logic Path Planning Algorithm Based on Geometric Landmarks and Kinetic Constraints

Jinghua Wang

College of Mechanical and Electric Engineering, Changchun University of Science and Technology, No. 7089, Weixing Road, 130022, Changchun, China; e-mail: hit1920s@163.com

Ziyu Xu, Xiyu Zheng, and Ziwei Liu

College of Mechanical and Electric Engineering, Changchun University of Science and Technology, No. 7089, Weixing Road, 130022, Changchun, China; e-mail: zyxu1995@163.com

Corresponding author: hit1920s@163.com

This paper mainly focuses on the path planning of mobile robots in complex two-dimensional terrain. It proposes a fuzzy rule-based path planning algorithm for multiple guide points by changing the spatial point-taking method and combining Dijkstra's and fuzzy logic algorithms. In the process of path planning, the existing algorithms have completed the smoothing process of turning point, but when analyzing the angular acceleration and linear acceleration of mobile robot in the process of movement, it is found that these paths are still difficult to meet the motion law of mobile robot. Through the mobile robot Angle range, velocity, acceleration range such as constraints, can guarantee the absolute path is an excellent way to satisfy the requirement of the mobile robot motion, combined with the Dijkstra algorithm, by using a fuzzy logic system based on considering the environment status of path planning, motion is more suitable for mobile robot path. The simulation results show that this algorithm can solve the complex environment that traditional fuzzy inference algorithms cannot plan. In subsequent studies, this algorithm will extend to group path planning and dynamic environment planning.

KEYWORDS: path planning, fuzzy logic, multiple boot points, Dijkstra, mobile robot.

1. Introduction

Mobile robots are critical equipment for the development of logistics, medical, military, and many other aspects and are also effective vehicles to assist and replace human resources. Mobile robots based on intelligent algorithms and control systems can largely escape from human control to make decisions and plan how to accomplish tasks independently, which is why they are developing in various countries. Since mobile robots work in non-fixed environments and locations, they need to make decisions and plan their movements in new environments, so they have to face several questions: Where am I? Where do I want to go? How do I get there? [12] These three questions correspond to determining the start and goal (or task) points and how to connect the endpoints within the constraints. The start point coordinates are input by the navigation signal. The target point of the task is usually defined manually or by autonomous task decisions, and how to reach the target point is achieved by path planning.

The path planning problem is a combinatorial optimization problem [1], which belongs to operations research. The objective is to search among all possible solutions to find an optimal path function that meets the constraints. If this function exists, it will be found, but it will not be searched for indefinitely if it does not exist. In general, applying the search algorithm can obtain more than one path, but it must satisfy the constraints provided and must not intersect with obstacles. When multiple paths are found, the optimal path among these functions is decided.

Path planning tasks are generally composed of four parts [38]. The first is to obtain map data through external input or the robot's self-perception. The second step is to design the search space so that it fits the path planning algorithm. The third step requires the designed search algorithm to connect the starting point with the task point through a path and select the path with the least cost as the initial path according to the needs of the task. If the initial path is not enough to meet the needs of robot movement, the initial path needs to be optimized in the fourth step to make them more in line with the physical constraints of robot movement and eliminate the mutation of physical quantity. It can be seen that the main research direction of path planning task is in the second and third steps.

In terms of search space design, since there is an infinite number of possible paths in a continuous space, space needs to be discretized based on the needs of path planning algorithms. The standard methods for paths are road networks, raster maps, and Tyson polygons (Voronoi diagrams) [11]. Road networks and Tyson polygons require more complex calculations of the boundaries of obstacles within the environment.

Although raster maps can be obtained directly by discretizing the space, they are equivalent to the point marking the whole map, leading to too many points and affecting the planning speed. Generally speaking, the more points are selected, the more the path obtained tends to be the least costly, but the longer the search time. Therefore, it is necessary to design a suitable search space according to the algorithm. To simplify the complexity of space design, and at the same time reduce the amount of data for planning way finding points, this paper chooses to use the characteristics of obstacles themselves stored in the environment array, based on the idea of Tyson polygon graph and designs a search space design method.

The third step of the task - path planning algorithm design is usually divided into the following five categories: 1) reaction-based planning algorithms, 2) sampling-based search algorithms, 3) graph search algorithms, 4) heuristic algorithms, and 5) hybrid algorithms. Reaction-based search algorithms are usually derived from mathematical models and related physical ideas, such as the potential field method using a mechanical model and a simulated annealing algorithm using a thermodynamic model. Since calculating the impact of environmental factors on the mobile robot in real-time, dynamic constraints can be imposed. The resulting paths are more accessible for the robot to follow but may fall into local optimal points. For example, the potential field method [3, 18, 21]. The second type of sampling-based search algorithm makes the search tree grow randomly or convergently by random numbers to faster search speed. Because of its randomness can be better for particular terrain such as slits compared to reactive search algorithms. However, the paths generated usually have many unnecessary inflection points, so it is generally impossible to plan the optimal path. For example, PRM algorithm [6, 20, 25], RRT algorithm [4, 19, 23, 30, 33]. The

third type of graph search algorithm is usually based on the idea of breadth-first search, which can be regarded as fully traversed before searching to the target point to find the optimal path accurately under the algorithm constraints. However, the algorithm time complexity is high, and at the same time, because the search direction needs to be time-infinite, the corner is a fixed value, only four, eight, or 16 directions can be searched at any one time, and the time complexity of the algorithm increases dramatically as more directions are searched at the same point-lack of flexibility.

Such as Dijkstra's algorithm [8, 15, 22], A* algorithm [5, 9, 28, 31], and others. The fourth type of heuristic algorithm is a new type of algorithm that emerged in recent years. This algorithm is based on empirical construction, through multiple training to achieve convergence, but in general, whether the parameters set correctly for the map primarily affect the convergence speed of the algorithm. For example, ant colony algorithms [2, 13, 14, 16], particle swarm algorithms [10, 17, 26, 27], and others. The fifth category of hybrid algorithms, on the other hand, obtains the advantages of multiple algorithms at the same time by integrating the above four types of algorithms, but this is usually obtained by paying more time complexity.

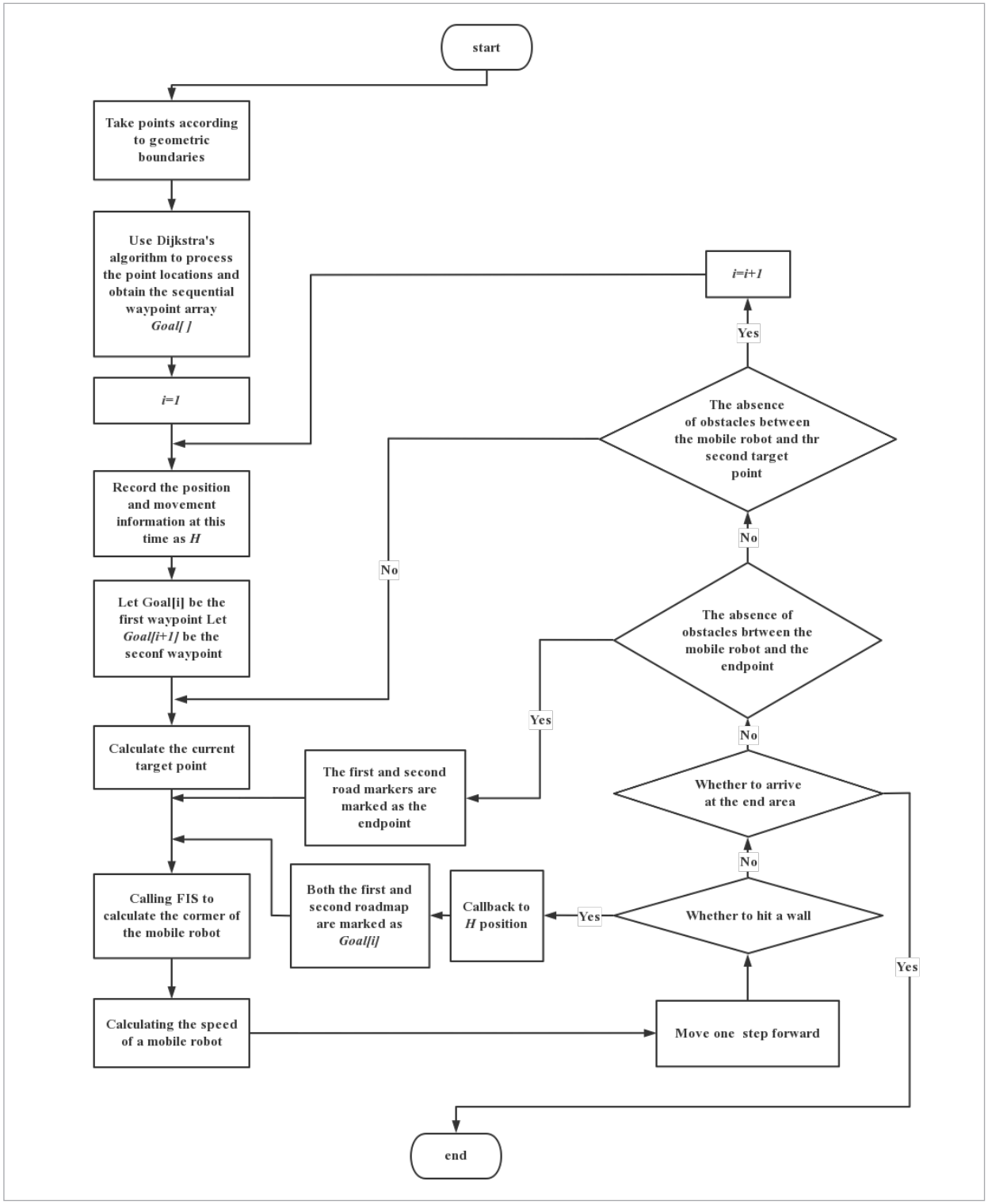
For example, in 2019, Yu combined the A* algorithm and the hybrid algorithm proposed by APF for solving the path planning problem in an unknown environment [37]. The values of the variables of the above algorithms are usually defined as fixed values in an array or as fixed functions, and the output of such fixed patterns is not always the most reasonable to handle when planning paths in a positional environment. Fuzzy logic provides an efficient way to obtain approximate and inexact properties in the real world, a simple way to handle large amounts of data [24]. Introducing fuzzy controllers into uncertain environments or considering path planning in dynamically constrained scenarios can lead to more flexible planning results. Traditional fuzzy logic algorithms use fuzzy logic systems in combination with shortest path algorithms [7], which can be used to define fuzzy numbers for the cost between uncertain coordinates and use them for path planning. Suppose the current state of the robot is considered to make a fuzzy judgment and guide the robot to the next state. In that case, a fuzzy controller is used to simulate the traditional reaction planning algorithm to give an approximately reasonable output [34-36]. Since the fuzzy controller is well integrated with the information to be considered, the feedback is more flexible. It can also consider the effects between the information simultaneously without creating severe local problems [29].

The traditional fuzzy logic algorithm takes the environmental information around the current position of the mobile robot as the affiliation function. It gets the next action instruction through fuzzy rules, but this also leads to the inability of this algorithm to consider the global map information. Thus, it loses forward-looking deep planning, which means that the obstacles faced will be judged as the better decision in the current situation. However, it may cost more or fall into a dead-end leading to planning failure in the future. Therefore, fuzzy logic systems are often used in combination with other algorithms. Montana and Keshari proposed a path planning algorithm based on fuzzy inference system and PRM algorithm in 2019 [20], using fuzzy inference system to re-optimize the path planned by PRM algorithm and smoothing at the fold to obtain a smooth path. This algorithm solves the problem that traditional fuzzy logic algorithms may fall into "dead ends" in complex environments while ensuring smooth paths. In 2020, Ning Wang proposed a hybrid algorithm combining fuzzy algorithm and potential field method for the path planning of unmanned surface vehicles. However, it did not eliminate the shortcomings of the traditional fuzzy algorithm in complex terrain planning [32].

The path planning algorithms combined with fuzzy logic in the previous paragraph plan a path from the starting point to the endpoint. However, there is a lack of correlation between each section of the path, while no indication is given to the speed and angle of the mobile robot in the path. When the robot follows the path, in reality, the discontinuity of the speed and angle may cause the actual trajectory to deviate too much from the planned path. Based on the problems of the above path planning algorithm, the proposed algorithm has the following innovations:

1. A new path search method is provided to search forward paths by constantly searching forward fuzzy logical order points.
2. In the process of path planning, factors such as maximum speed, maximum acceleration, and maximum rotation angle of the mobile robot are taken

Figure 1
Flow chart of Fuzzy-GK algorithm



into account to provide predetermined values of speed and heading for each step, ensuring the continuity of speed turning angle and making the final path more in line with the requirements of robot movement.

3 The fuzzy logic controller is used to determine the next movement according to the current state of the robot, which makes the movement selection more flexible and further improves the safety of the movement.

In order to better explain the specific process of the algorithm proposed in this paper and facilitate the understanding of the following contents, the specific process is shown in Figure 1.

The paper is structured as follows: Section 2 describes the information that needs to be considered by the mobile robot in path planning. Section 3 explains how the search space is handled and how the Dijkstra algorithm ranks the points. Section 4 describes how to use the ranking points to guide the fuzzy controller for path planning. Section 5 gives a comparison of the proposed method with other path planning methods. Section 6 summarizes the simulation results and offers the future directions of the algorithm.

2. Problem Definition

Before introducing the Fuzzy-GK algorithm, the problem faced by the mobile robot needs to be defined. This chapter will describe the problem to be solved by the path planning algorithm. The constraints it receives will be defined. The environmental parameters and mobile robot information identified in the algorithm during planning will also be determined.

The path planning algorithm aims to obtain a path that points from the starting point to the endpoint. The constraints come mainly from two aspects: the search space and the mobile robot. In terms of search space, the mobile robot must not touch or cross the obstacle zone and not drive away from the defined area during its action. In terms of the mobile robot, the size of the mobile robot itself and the effect of its speed on its turning ability is taken into account. In addition, how-to guide the mobile robot more precisely when it performs path tracking is also one issue.

Before solving the above problem, the following information needs to be defined in advance, the search space will be stored in an array, and the feasible and obstacle zones will be specified. Define the starting and ending coordinates and ensure that they are within the feasible zone. Since the search method is to converge to the endpoint, it is difficult for a robot to reach the predetermined point, so a circular area with the endpoint coordinates as the center and the diagonal length of the mobile robot as the radius is set as the endpoint area. The mobile robot needs to determine its length and width, initial speed and direction, maximum speed and maximum steering angle, and determine the correspondence between speed and turning angle.

3. Search Space Design

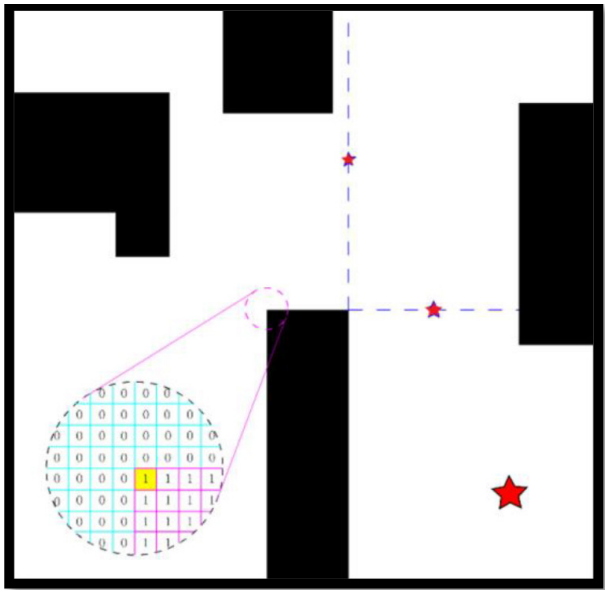
This section will introduce the method of search space design. As shown in Figure 1, the storage situation of two-dimensional map data is generally stored in a two-dimensional array. The number of rows and columns in which the points are located represents the coordinate values. The 0s and 1s within the array express whether this coordinate is passable or not, respectively.

Suppose the map's shape and distribution of obstacles are complicated or the map range is too extensive. In that case, the imported environment map is needed, for the search space is designed to reduce the search volume of the path planning algorithm and improve the search speed. This algorithm chooses to identify the obstacle boundary points and connects them with the next obstacle in horizontal and vertical directions, respectively. The midpoint of the line segment is taken as the marker point. The specific method is described below.

First of all, we need to determine the boundary points of the obstacles in the array. We can determine whether the coordinates around the point are obstacles to determine whether this coordinate is a boundary point. Generally speaking, a point can be judged as a boundary point when at least one of the four directions up, down, left, right, and center can touch the feasible area. However, to reduce the number of boundary points and speed up the search, the points that can touch the feasible area in at least two directions are selected as boundary points. After that, a line segment is made from the boundary point to explore the fea-

sible area until the next obstacle is encountered. The midpoint of the line segment of this exploration path is taken as a marker point. This method ensures that the convergence point of the path is not too close to the obstacle (as in Figure 2).

Figure 2
Schematic diagram of road marking points taking points



After having the marker points, sorting them and organizing the shortest sequence of marker points as the guide coordinates is necessary. This sequence requires no obstacles between the two marker points before and after, minimizing the total cost. Compared with the global map, the number of points has been dramatically reduced, so Dijkstra's algorithm is chosen for shortest path planning. The traditional Dijkstra algorithm calculates the cost using the displacement cost expressed as the Euclidean distance between two points, but this will quickly. The cost function of Dijkstra's algorithm is as follows:

$$diatanceCustjudge = VED + turningPrice \quad (1)$$

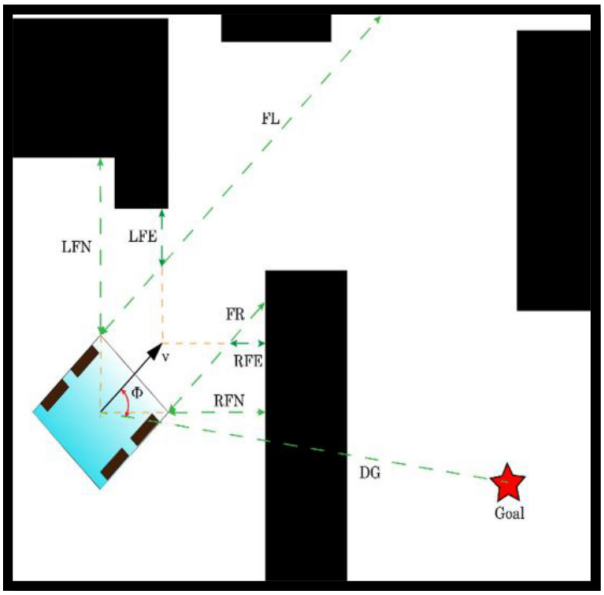
$$VED=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2} \quad (2)$$

$$turningPrice=2\pi r\frac{|\Delta\theta|}{\pi} \quad (3)$$

$$\Delta\theta=tan^{-1}\frac{y_2-y_1}{x_2-x_1}-tan^{-1}\frac{y_1-y_0}{x_1-x_0}, \quad (4)$$

where the previous point coordinates are (x_0, y_0), the current point coordinates are (x_1, y_1), the search point coordinates are (x_2, y_2), the minimum turning radius of the mobile robot is r, *VED* denotes the Euclidean distance, *turningPrice* denotes the turning cost, and *diatanceCostjudge* denotes the total cost (as in Figure 3).

Figure 3
Schematic diagram of distance parameters



4. Implementation of FUZZY-GK

This section describes the fuzzy logic path planning algorithm (FUZZY-GK) based on geometric signs and dynamic constraints. In general, the algorithm follows the model of asking for a path by asking for two subsequent guide coordinates to search for a path.

The first step is to set the information about the surroundings that the algorithm needs to move the robot when planning the path. The forward distance (F) will be selected as the minimum value of the distance from the left front end (FL) and the right front end (FR) of the mobile robot to the obstacles in front. In addition, the mobile robot needs to detect the obstacle in the oblique front. The mobile robot's left front

end detects the distance to the left front 45° direction of the obstacle (LFN) and detects the distance to the left front 45° after traveling forward at the current speed (LFE). The correct front obstacle measurement method is the same (RFN, RFE). In addition, the deviation angle between the current direction of the mobile robot and the direction of the target point, and the distance of the current position from the target point (DG).

Due to the large number of fuzzy sets considered, the traditional Mamdani inference method was chosen to simplify the definition of fuzzy relations, using the max-min operation to compose the algorithm. The fuzzy inference system integrates the obstacle distance (including front, left front, and right front), the angle between the direction of the target point and the direction of motion, the distance to the target point, and the previous direction turn. The output is the angle of the next turn. From the turn angle obtained in the previous step, the speed maintained in the next step can be calculated. The speed is calculated from the current speed, the size of the turn angle, and the distance to the left and suitable front obstacles after traveling forward at the current speed together. Thus, we obtain a method for step-by-step path planning in an austere environment using a fuzzy logic system.

Next, the guiding coordinates will be used to guide the mobile robot around more complex areas. In Chapter 3, we have used the boundary points of the obstacles and Dijkstra's algorithm to obtain a minimal sequence of guiding coordinates (*Goal* $[]$), which will serve as temporary goal points that will guide the mobile robot in order until it reaches the end of the exploration.

Using a sequence of guide coordinates to segment the overall path can effectively plan complex environments with multiple turns. However, it is easy to form large-angle corners when switching temporary target points, making the planned path lose coherence. Therefore, in this algorithm, except for the last segment toward the endpoint, the mobile robot will call the next two guide coordinates in the sequence to guide it together. As the current position approaches the next guidance coordinate, the temporary target point will gradually shift to the second guidance coordinate to make the path smoother.

It is not feasible to use a moving temporary target point in all cases. In a narrow, large-angle turn, the mobile robot controlled by the fuzzy logic system may not be able to bypass the obstacle altogether and hit it. In this case, this section of the path will be guided by the next guide coordinates to make a fixed temporary target point to pass this obstacle zone.

When no obstacle is detected between the current position and the endpoint, the target point of the robot will be locked at the endpoint. When the mobile robot enters the endpoint area, the planning is judged to be finished.

5. Simulation and Analysis

In this section, three different environments are designed to simulate the written path planning procedure using simulation software, and the simulation results are analyzed and compared with other algorithms. The map used for the simulation is a 500×500-pixel two-color map, with black representing the obstacle area and white representing the feasible area. The three environments are named Environment 1, Environment 2, and Environment 3. Environment 1 is a simple space consisting of ordinary rectangular obstacles, and the mobile robot only needs to perform simple obstacle avoidance operations to reach the target point. Environment 2 consists of lines that form an encircling obstacle, and the mobile robot needs to circumnavigate several times to find the target point. Environment 3 contains curved-edge obstacles and complex obstacle areas, and the mobile robot needs to carefully navigate the obstacles and find the most reasonable path to travel.

Environment 1 is a search space consisting of ordinary rectangular obstacles (as in Figure 4). The four corners of the rectangular map are marked to determine the range of coordinates. The coordinates of the upper left corner of the map are marked as (0, 0), the coordinates of the lower right corner are (500, 500), the coordinates of the start point are (10, 10), and the coordinates of the endpoint are (490, 490). In addition, set the size of the mobile robot to 10×10, the maximum speed to 10, the maximum rotation angle to 60°, and the maximum acceleration to 3. Set the start point to (10, 10); the start angle is 60°; the speed is 4; the endpoint to (490, 490). The termination area is a circular area of radius 20 with the termination point as the circle's center.

Figure 4
Environment 1 map

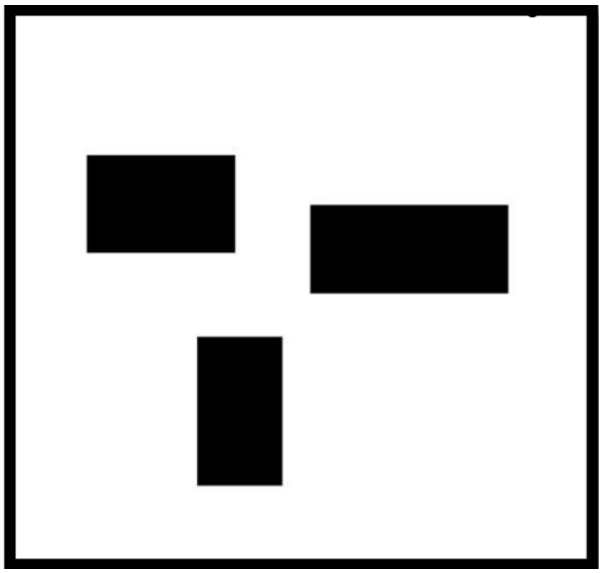


Figure 5
Simulation results of path planning in environment 1

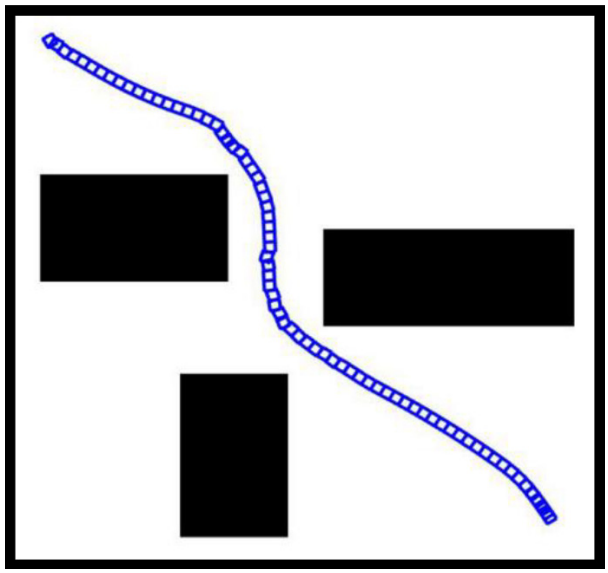


Table 1
Simulation results of this algorithm in environment 1

Step	X	Y	Angle	Speed	Price
1	10	10	1.047198	4	0
2	13.81369	15.8699	0.576166	7	7
3	19.69528	23.95736	0.628781	10	17
4	25.6664	31.97893	0.639896	10	27
5	31.61799	40.01502	0.637463	10	37
6	37.44347	48.14298	0.62186	10	47
7	43.05355	56.42108	0.595603	10	57
8	48.36121	64.89626	0.559505	10	67
9	53.45253	73.50315	0.534176	10	77
10	58.37982	82.20498	0.515223	10	87
11	63.18061	90.97723	0.500744	10	97
12	67.89871	99.79423	0.491343	10	107
13	72.51969	108.6625	0.48036	10	117
14	77.2064	117.4962	0.487786	10	127
15	82.22577	126.1453	0.525837	10	137
16	87.16255	134.8417	0.516313	10	147
17	91.6446	143.781	0.464757	10	157
18	95.93569	152.8136	0.443506	10	167
19	100.6097	161.5973	0.489019	9.949874	176.9499

Step	X	Y	Angle	Speed	Price
20	104.8177	169.553	0.486512	9	185.9499
21	112.7996	175.5769	0.924298	10	195.9499
22	118.0113	180.2502	0.839811	7	202.9499
23	125.9063	186.3876	0.910006	10	212.9499
24	134.1472	192.0522	0.968588	10	222.9499
25	142.6389	197.3334	1.014415	10	232.9499
26	151.2777	202.3703	1.042925	10	242.9499
27	160.8266	205.3399	1.269293	10	252.9499
28	170.5295	207.7596	1.326402	10	262.9499
29	180.3863	209.4458	1.401362	10	272.9499
30	190.3006	210.7525	1.439758	10	282.9499
31	200.2546	211.7098	1.47492	10	292.9499
32	210.2252	212.4767	1.494024	10	302.9499
33	220.098	210.8872	1.730428	10	312.9499
34	228.8244	213.0896	1.32357	9	321.9499
35	238.6313	215.0453	1.373962	10	331.9499
36	248.5118	216.5864	1.416068	10	341.9499
37	258.2129	219.013	1.325688	10	351.9499
38	267.9619	221.2397	1.346246	10	361.9499
39	276.4524	226.5228	1.014192	10	371.9499
40	284.5525	229.0503	1.26833	8.485281	380.4352
41	292.5402	235.0667	0.925239	10	390.4352
42	297.1451	243.5399	0.497815	9.643651	400.0788
43	302.4898	247.4861	0.934801	6.643651	406.7225
44	308.832	254.7508	0.717716	9.643651	416.3661
45	314.3737	263.0748	0.587359	10	426.3661
46	320.9954	270.5684	0.72371	10	436.3661
47	326.6545	278.813	0.601539	10	446.3661
48	332.8814	286.6377	0.672179	10	456.3661
49	338.0398	295.2046	0.541981	10	466.3661
50	343.1234	303.816	0.533287	10	476.3661
51	348.0599	312.5126	0.516274	10	486.3661
52	352.9828	321.217	0.514716	10	496.3661
53	357.977	329.8805	0.522933	10	506.3661
54	363.0673	338.488	0.534063	10	516.3661
55	368.2494	347.0406	0.544753	10	526.3661
56	373.5214	355.538	0.555299	10	536.3661
57	378.883	363.9791	0.565887	10	546.3661

Step	X	Y	Angle	Speed	Price
58	384.3343	372.3627	0.57654	10	556.3661
59	389.8737	380.6882	0.587095	10	566.3661
60	395.5016	388.9542	0.597753	10	576.3661
61	401.2275	397.1526	0.609666	10	586.3661
62	407.0609	405.2749	0.622831	10	596.3661
63	412.9987	413.3212	0.635747	10	606.3661
64	419.0174	421.3071	0.645844	10	616.3661
65	425.1091	429.2375	0.655008	10	626.3661
66	431.3523	437.0492	0.674266	10	636.3661
67	437.9462	444.5672	0.720008	10	646.3661
68	445.2357	451.413	0.816777	10	656.3661
69	453.314	457.2216	0.94741	9.949874	666.316
70	461.1476	461.9791	1.025006	9.165151	675.4811
71	468.5519	466.1237	1.060492	8.485281	683.9664
72	475.0818	469.4942	1.09431	7.348469	691.3149
73	480.2453	472.0117	1.117156	5.744563	697.0594
74	484.0904	473.8048	1.134436	4.242641	701.3021

Simulation using Matlab leads to the planned path shown in Figure 5. The planning time under this map is 11.69s, and the length of the planned path is 701.3. In planning the path, the algorithm records the position, velocity, direction angle, and the cost of the distance spent at each step, and the recorded values are shown in Table 1. The velocity and direction angle are plotted to obtain the line graph shown in Figures 6-7. The horizontal coordinate is the number of steps, the vertical coordinate of Figure 6 is the velocity, and the vertical coordinate of Figure 7 is the directional angle radian value. It can be seen that the velocity of the mobile robot increases uniformly from the initial velocity to the maximum velocity in the open area in the initial section (1 to 20 steps), and the directional angle is kept constant. In the middle section, from 21 to 45 steps, the mobile robot detects the surrounding obstacles. It makes the angle adjustment while decreasing the speed several times to adapt to the turning angle. In the end, the mobile robot drives into the open area

Figure 6
Statistical graph of speed variation

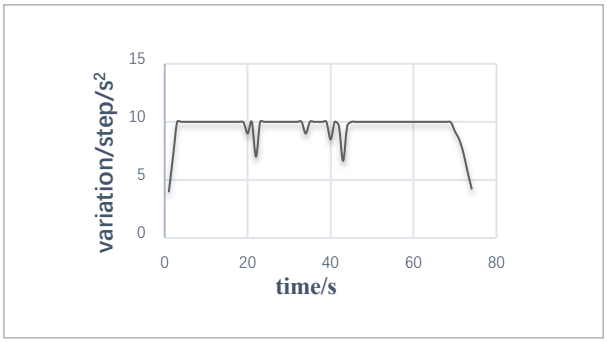
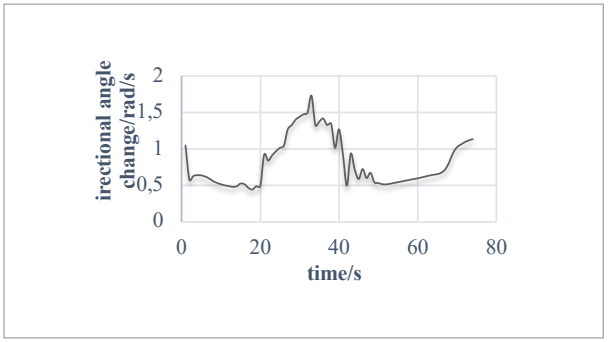


Figure 7
Schematic diagram of directional angle change



again. It makes only minor directional adjustments towards the endpoint, thus maintaining the maximum speed smoothly until it decelerates near the endpoint. The mobile robot strictly follows the predetermined speed, acceleration, and corner limits, and large corners only occur when the speed is reduced. In this way, it provides a good reference value for the mobile robot to follow the path.

Environment 2 is the obstacle space composed of loops (as in Figure 8). Let the starting position be the center point of the loop (250, 250), and the ending point be located at (490, 490). The path planning of environment 2 using this algorithm is also simulated using the A* algorithm, PRM algorithm, RRT algorithm, and traditional fuzzy logic algorithm under the same map, respectively.

The simulation results are shown in Figure 9. A is the Fuzzy-GK algorithm, b is the A* algorithm, c is the PRM combined with the Dijkstra algorithm, and d is the RRT algorithm. The simulation time and path length of several algorithms are summarized as shown in Table 2. The algorithm comprises three parts: determining boundary point coordinates, sorting boundary points, and finding the optimal path. Set the number of boundary point coordinates as n. The complexity of the three steps is $O(l)$,$O(m^2)$,$O(n)$, Where l is the number of grid points, m is the number of boundary points, n is the planned number of nodes, so the total complexity of this algorithm is $O(l+n+m^2)$. The complexity of the Dijkstra algorithm is $O(l^2)$, and the complexity of the A* algorithm is $O(llogl)$. In this algorithm, m is far less than l, so it has minor complexity than the Dijkstra algorithm and A*. On the other hand, although the complexity of the PRM algorithm and RRT algorithm is $O(l)$, it can be found by comparison that several different algorithms have folding angles close to obstacles, which may cause the robot to have to stop and turn or deviate from the planned path in practical applications. The PRM algorithm is prone to fail to find the path when there are not enough sampling points due to its highly random nature. The path planned by the RRT algorithm has a large number of unnecessary folds. Traditional fuzzy logic algorithms cannot plan in such loops. In contrast, the present algorithm considers the speed and corner constraints of the mobile robot, and the planned paths are more reasonable than the traditional algorithm.

Figure 8
Environment 2 map

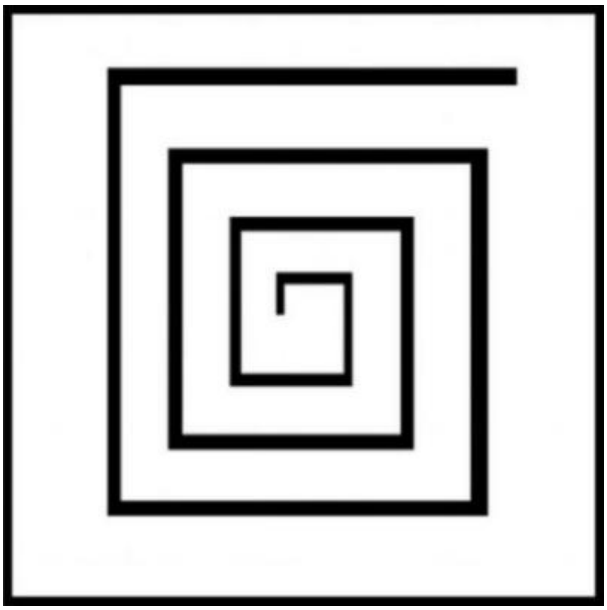
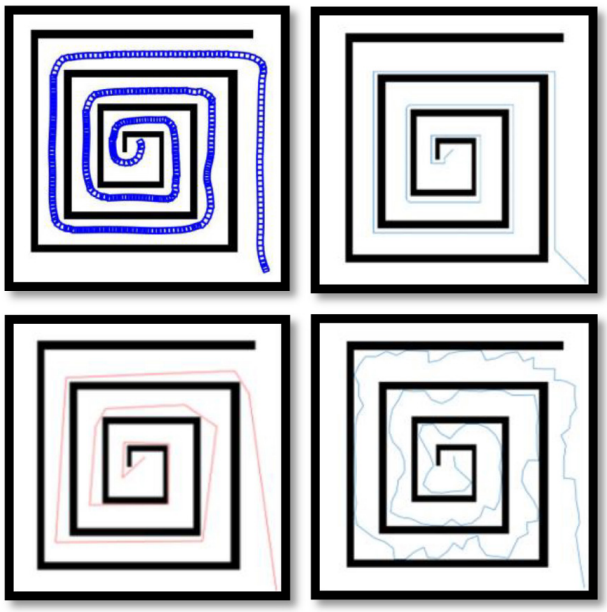


Figure 9
Simulation results of the four algorithms in environment 2: this algorithm (In the upper left), A* algorithm (In the upper right), PRM algorithm (In the lower-left), RRT algorithm (In the lower right)



In natural environments, obstacles are not always standard rectangles with a simple distribution as in

the first two. The natural environment has obstacles with curved edges, greater density, and more complex distribution. There is an optimal local region in the environment, which should be fully considered in the planning for the limitation of speed on the corner and whether the route of the mobile robot is attracted by the local optimal point and cannot get the planned path smoothly. For this complex situation, a map such as Environment 3 is built and simulated.

Table 2
The simulation time and path length of several algorithms in environment 2

Arithmetic	Simulation time/(s)	path length/(step)	kmeans++
Fuzzy-GD	57.695	2467.75	$O(l+n+m^2)$
A*	750.14	2311.56	$O(llogl)$
PRM	15.36	2447.32	$O(l)$
RRT	21.367	3265.91	$O(l)$

The map of Environment 3 and its simulation results are shown in Figure 10. The coordinates of the start point are (10, 10), and the coordinates of the endpoint are (490, 490). In addition, set the size of the mobile robot to 6×6, the maximum speed to 6, the maximum rotation angle to drop back below a certain threshold. When returning to the open area, the speed gradually returns to the maximum speed. Moreover, the mobile robot successfully avoids entering the wrong area and runs around the obstacle to the endpoint.60°, and the maximum acceleration to 2. Set the start point to (10, 10). As can be seen from the figure, when the mobile robot approaches an obstacle or the corner becomes large, its speed drops back below a certain threshold. When returning to the open area, the speed gradually returns to the maximum speed. Moreover, the mobile robot successfully avoids entering the wrong area and runs around the obstacle to the endpoint.

Finally, in order to more clearly show the differences of this algorithm compared to existing algorithms, we take the maps in the papers of Montana and Keshari as one of the maps of this algorithm. Through the planned paths, we can know that the path planned by this algorithm is smoother. No corner with small radius appears, as shown in Figure 11. At the same time, as shown in Figure 12, in the original algorithm, there were many instantaneous large Angle changes, which obviously did not conform to the actual motion law. However, in the path planning of this algorithm, Angle changes were continuous, which met the actual motion law. At the same time, it is worth noting that the path planned by this algorithm is farther away from obstacles and will not be too close to obstacles, which is of great benefit to the actual movement of mobile robots in most cases.

Figure 10
Environment 3 map (on the left), the simulation results in environment 3 (on the right)

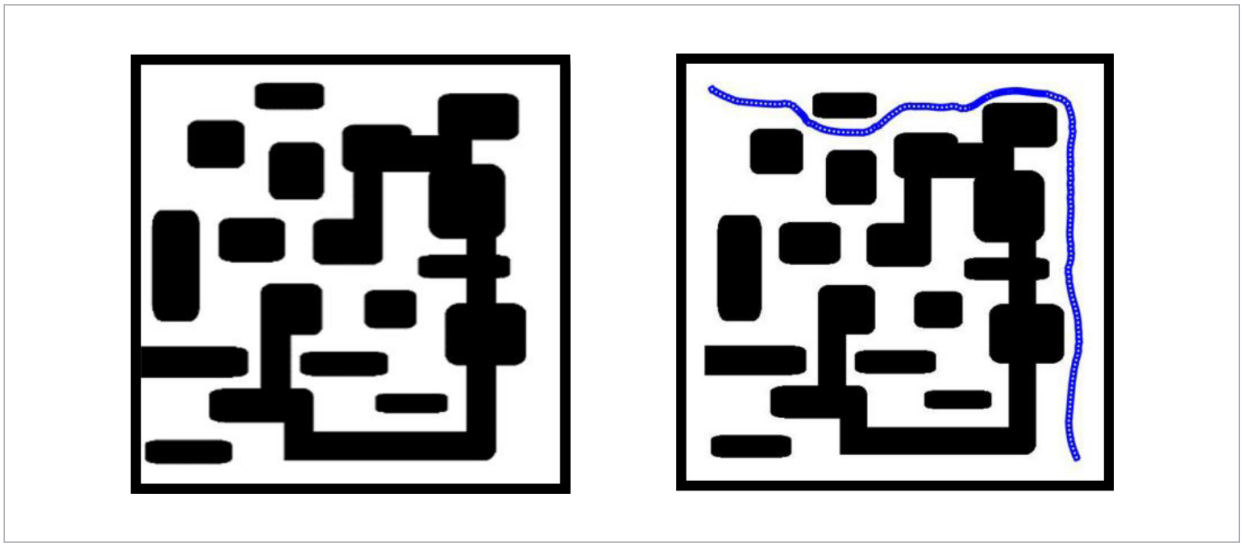


Figure 11
The path planned by the original algorithm (on the left) [20], the path planned by this algorithm (on the right)

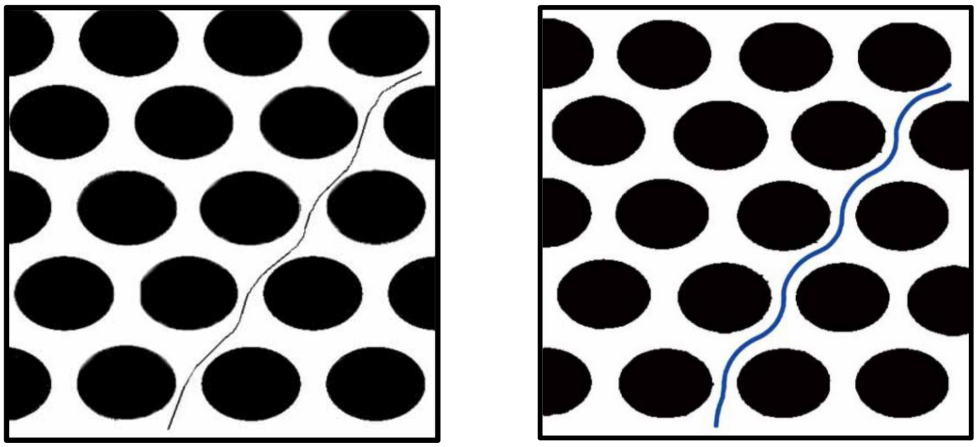
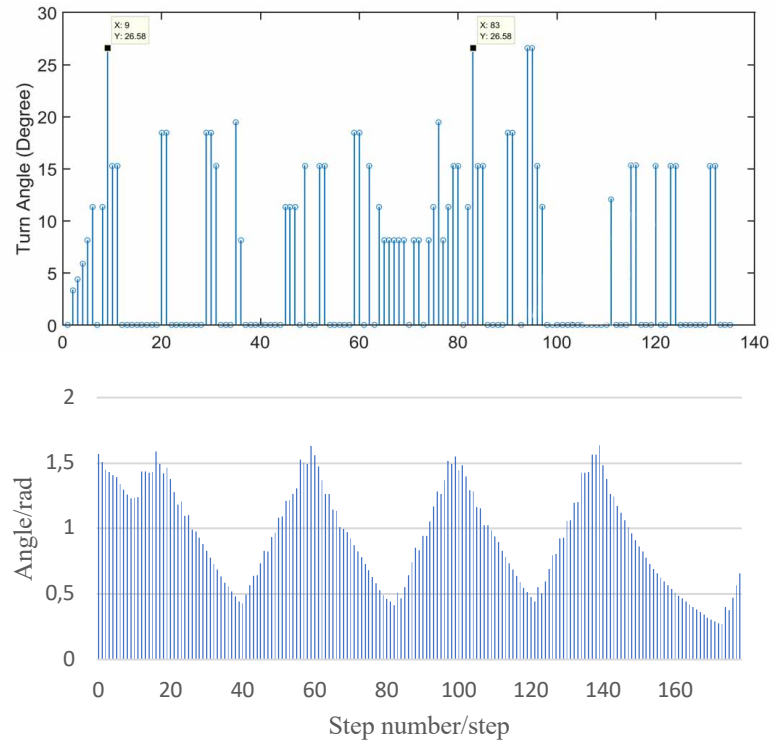


Figure 12
The angle change of the trajectory in the original algorithm (at the top) [20], The angle change of motion trajectory in this algorithm (at the bottom)



5. Conclusions

This paper proposes a method to handle the search space based on geometric methods and uses fuzzy logic algorithms for path planning. It solves the dimensional constraint of the mobile robot during path planning, considers the effect of the turning angle on the speed, and balances the planning time and path length. At the same time, the reference values of velocity and deflection angle are identified for each position point, which provides data support for the path tracking of the mobile robot in reality.

Compared with the original algorithm, the planned path turns less and does not have large angular acceleration, which is more in line with the motion law of mobile robots and has good value. At the same time, the algorithm has a good effect on multi-factor processing in a complex environment. However, when there are too many obstacles, it may be impossible to obtain the path that satisfies the movement of mobile robots, which is still one of the future research directions. At the same time, how to extend this algorithm to a high-dimensional environment to obtain application value is also worth studying, which is meant for planning the motion of underwater or aerial high-speed robots in complex environments.

Acknowledgement

This work is supported by Innovation Fundation of Luquan (LQ-2020-01) and the "111" project of China (D17017) grant.

References

1. Aggarwal, S., Kumar, N. Path Planning Techniques for Unmanned Aerial Vehicles: A Review, Solutions, and Challenges. Computer Communications, 2020, 149, 270-299. https://doi.org/10.1016/j.comcom.2019.10.014
2. Ali, H., Gong, D., Wang, M., Dai, X. Path Planning of Mobile Robot with Improved Ant Colony Algorithm and MDP to Produce Smooth Trajectory in Grid-based Environment. Frontiers in Neurorobotics, 2020, 14, 44. https://doi.org/10.3389/fnbot.2020.00044
3. Bayat, F., Najafinia, S., Aliyari, M. Mobile Robots Path Planning: Electrostatic Potential Field Approach. Expert Systems with Applications, 2018, 100, 68-78. https://doi.org/10.1016/j.eswa.2018.01.050
4. Chao, N., Liu, Y., Xia, H., Ayodeji, A., Bai, L. Grid-based RRT* for Minimum Dose Walking Path-Planning in Complex Radioactive Environments. Annals of Nuclear Energy, 2018, 115, 73-82. https://doi.org/10.1016/j.anucene.2018.01.007
5. Chen, J., Li, M., Yuan, Z., Gu, Q. An Improved Algorithm for UAV Path Planning Problems. 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC2020), Chongqing, China, June 12-14, 2020, 958-962. https://doi.org/10.1109/ITNEC48623.2020.9084806
6. Chen, J., Zhou, Y., Gong, J., Deng, Y. An Improved Probabilistic Roadmap Algorithm with Potential Field Function for Path Planning of Quadrotor. 2019 Chinese Control Conference (CCC2019), Nanchang, China, June 3-5, 2019, 3248-3253. https://doi.org/10.23919/ChiCC.2019.8865585
7. Deng, Y., Chen, Y., Zhang, Y., Mahadevan, S. Fuzzy Dijkstra Algorithm for Shortest Path Problem Under Uncertain Environment. Applied Soft Computing, 2012, 12(3), 1231-1237. https://doi.org/10.1016/j.asoc.2011.11.011
8. Dinitz, Y., Itzhak, R. Hybrid Bellman-Ford-Dijkstra Algorithm. Journal of Discrete Algorithms, 2017, 42, 35-44. https://doi.org/10.1016/j.jda.2017.01.001
9. Hart, P. E., Nilsson, N. J., Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2), 100-107. https://doi.org/10.1109/TSSC.1968.300136
10. Huang, C. A Novel Three-dimensional Path Planning Method for Fixed-wing UAV Using Improved Particle Swarm Optimization Algorithm. International Journal of Aerospace Engineering, 2021, 1-19. https://doi.org/10.1155/2021/7667173
11. Kelly, A., Wei, W., Weina, C. Mobile Robotics: Mathematics, Models, and Methods. China Machine Press, 2020, 10, 640-390.
12. Koubaa, A., Bennaceur, H., Chaari, I., Trigui, S., Ammar, A., Sriti, M., Alajlan, M., Cheikhrouhou, O., Javed, Y. Robot Path Planning and Cooperation: Foundations, Algorithms and Experimentations. Springer International Publishing AG, 2018, 1, 4-7. https://doi.org/10.1007/978-3-319-77042-0 https://doi.org/10.1007/978-3-319-77042-0

13. Lee, J. Heterogeneous-ants-based Path Planner for Global Path Planning of Mobile Robot Applications. International Journal of Control, Automation and Systems, 2017, 15(4), 1754-1769. https://doi.org/10.1007/s12555-016-0443-6

14. Liu, J., Yang, J., Liu, H., Tian, X., Gao, M. An Improved Ant Colony Algorithm for Robot Path Planning. Soft Computing, 2017, 21(19), 5829-5839. https://doi.org/10.1007/s00500-016-2161-7

15. Luo, M., Hou, X., Yang, J. Surface Optimal Path Planning Using an Extended Dijkstra Algorithm. IEEE Access, 2020, 8, 147827-147838. https://doi.org/10.1109/ACCESS.2020.3015976

16. Luo, Q., Wang, H., Zheng, Y., He, J. Research on Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm. Neural Computing and Applications, 2019, 32(6), 1555-1566. https://doi.org/10.1007/s00521-019-04172-2

17. Mac, T. T., Copot, C., Tran, D. T., Keyser, R. D. A Hierarchical Global Path Planning Approach for Mobile Robots Based on Multi-objective Particle Swarm Optimization. Applied Soft Computing, 2017, 59, 68-76. https://doi.org/10.1016/j.asoc.2017.05.012

18. Matoui, F., Boussaid, B., Metoui, B., Frej, G. B., Abdelkrim, M. N. Path Planning of a Group of Robots with Potential Field Approach: Decentralized Architecture. International Federation of Automatic Control (IFAC2017), Johannesburg, South Africa, December 7-8, 2017, 50(1), 11473-11478. https://doi.org/10.1016/j.ifacol.2017.08.1822

19. Mohammed, H., Romdhane, L., Jaradat, M. A. RRTN: An Efficient Approach to Path Planning in 3D for Static and Dynamic Environments. Advanced Robotics, 2021, 35(3-4), 168-180. https://doi.org/10.1080/01691864.2020.1850349

20. Mohanta, J. C., Keshari, A. A Knowledge Based Fuzzy-Probabilistic Roadmap Method for Mobile Robot Navigation. Applied Soft Computing, 2019, 79, 391-409. https://doi.org/10.1016/j.asoc.2019.03.055

21. Orozco-Rosas, U., Montiel, O., Sepúlveda, R. Mobile Robot Path Planning Using Membrane Evolutionary Artificial Potential Field. Applied Soft Computing, 2019, 77, 236-251. https://doi.org/10.1016/j.asoc.2019.01.036

22. Qu, Y., Zhang, Y., Zhang, Y. A global Path Planning Algorithm for Fixed-wing UAVs. Journal of Intelligent and Robotic Systems, 2018, 91(3), 691-707. https://doi.org/10.1007/s10846-017-0729-9

23. Qureshi, A. H., Ayaz, Y. Intelligent Bidirectional Rapidly-Exploring Random Trees for Optimal Motion Planning in Complex Cluttered Environments. Robotics and Autonomous Systems, 2017, 68, 1-11. https://doi.org/10.1016/j.robot.2015.02.007

24. Rath, A. K., Parhi, D. R., Das, H. C., Muni, M. K., Kumar, P. B. Analysis and Use of Fuzzy Intelligent Technique for Navigation of Humanoid Robot in Obstacle Prone Zone. Defence Technology, 2018, 14(6), 677-682. https://doi.org/10.1016/j.dt.2018.03.008

25. Santiago, R. M. C., Ocampo, A. L. D., Ubando, A. T., Bandala, A. A., Dadios, E. P. Path Planning for Mobile Robots Using Genetic Algorithm and Probabilistic Roadmap. 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM2017), Manila, Philippines, December 1-3, 2017, 1-5. https://doi.org/10.1109/HNICEM.2017.8269498

26. Song, B., Wang, Z., Zou, L. On Global Smooth Path Planning for Mobile Robots Using a Novel Multimodal Delayed PSO Algorithm. Cognitive Computation, 2017, 9(1), 5-17. https://doi.org/10.1007/s12559-016-9442-4

27. Song, B., Wang, Z., Zou, L., Xu, L., Alsaadi, F. E. A New Approach to Smooth Global Path Planning of Mobile Robots with Kinematic Constraints. International Journal of Machine Learning and Cybernetics, 2019, 10(1), 107-119. https://doi.org/10.1007/s13042-017-0703-7

28. Song, R., Liu, Y., Bucknall, R. Smoothed Algorithm for Practical Unmanned Surface Vehicle Path Planning. Applied Ocean Research, 2019, 83, 9-20. https://doi.org/10.1016/j.apor.2018.12.001

29. Sun, B., Zhu, D., Yang, S. X. An Optimized Fuzzy Control Algorithm for Three-dimensional AUV Path Planning. International Journal of Fuzzy Systems, 2018, 20(2), 597-610. https://doi.org/10.1007/s40815-017-0403-1

30. Tahir, Z., Qureshi, A. H., Ayaz, Y., Nawaz, R. Potentially Guided Bidirectionalized RRT for Fast Optimal Path Planning in Cluttered Environments. Robotics and Autonomous Systems, 2018, 108, 13-27. https://doi.org/10.1016/j.robot.2018.06.013

31. Thoresen, M., Nielsen, N. H., Mathiassen, K., Pettersen, K. Y. Path Planning for UGVs Based on Traversability Hybrid A. IEEE Robotics and Automation Letters, 2021, 6(2), 1216-1223. https://doi.org/10.1109/LRA.2021.3056028

32. Wang, N., Xu, H., Li, C., Yin, J. Hierarchical Path Planning of Unmanned Surface Vehicles: A fuzzy Artificial Potential Field Approach. International Journal of Fuzzy Systems, 2020. https://doi.org/10.1007/s40815-020-00912-y

33. Xu J., Tian Z., He W., Huang Y. A Fast Path Planning Algorithm Fusing PRM and P-Bi-RRT. 2020 11th International Conference on Prognostics and System Health Management (PHM2020), Jinan, China, June 8-10, 2020, 503-508. https://https://doi.org/10.1109/PHM-Jinan48558.2020.00098

34. Yang, G., Wang, H., Chen, J. Disturbance Compensation Based Asymptotic Tracking Control for Nonlinear Systems with Mismatched Modeling Uncertainties. International Journal of Robust and Nonlinear Control, 2021 31(8), 2993-3010. https://doi.org/10.1002/rnc.5436

35. Yang, G., Yao, J. Output Feedback Control of Electro-hydraulic Servo Actuators with Matched and Mismatched Disturbances Rejection. Journal of the Franklin Institute, 2019, 356(16), 9152-9179. https://doi.org/10.1016/j.jfranklin.2019.07.032

36. Yang, G., Yao, J., Ullah, N. Neuroadaptive Control of Saturated Nonlinear Systems with Disturbance Compensation. ISA Transactions, 2021. https://doi.org/10.1016/j.isatra.2021.04.017

37. Yu, J., Deng, W., Zhao, Z., Wang, X., Xu, J., Wang, L., Sun, Q., Shen, Z. A Hybrid Path Planning Method for an Unmanned Cruise Ship in Water Quality Sampling. IEEE Access, 2019, 7, 87127-87140. https://doi.org/10.1109/ACCESS.2019.2925894

38. Zhang, H., Lin, W., Chen, A. Path Planning for the Mobile Robot: A review. Symmetry, 2018, 10(10), 450. https://doi.org/10.3390/sym10100450