A Mixed Fast-RRT*-A* To Solve the Problem of Generating Nodes near Obstacles for Optimal Paths

Heru Suwoyo
Department of Electrical Engineering, Universitas Mercu Buana, Indonesia

Raudah Alfiani
Department of Electrical Engineering, Universitas Mercu Buana, Indonesia

Abstract: In the robot environment with static obstacles, robots designed to avoid obstacles and move from the initial position to the destination position, the consumption of the minimum value and the search for the shortest path as the current research, most of the studies are based on obstacle detection to search for more than one path planning. The path using path planning has two sampling methods based on which by working according to random nodes while searching based on using heuristics to find the path. Fast-RRT*-A* were to optimize a path with fast time intensity. From Fast-RRT* developed with improvement-RRT with optimal fast, namely fast optimal in an unreachable space from random trees introduced for speed and algorithm stability; (2) Random steering is used in expansion to solve performance problems in tight spaces; (3) Path fusion path adjustments are obtained quickly. The results of this study are in the form of a profit map comparing the three PRC algorithms* with a time of 48.6474, A* with a time of 38.7527, and FastRRT*-A* with a time of 10.1411, So these three steps make Fast-RRT*-A*.

Keywords: Path Planning, Fast-RRT*-A*, Improvement RRT, Mobile Robot, Path Optimization

Correspondents Author:
Heru Suwoyo, Department of Electrical Engineering, Universitas Mercu Buana, Indonesia
Email : heru.suwoyo@mercubuana.ac.id

Introduction

A robot is a tool that automatically solves human needs using programs embedded in the robotic system created, or with human-run controls. In recent years, the development of robots has received increasing attention since robots can increase productivity and provide various conveniences (Li et al., 2020). The development of robotics and research on automatic navigation technology has encouraged the expansion of robotic applications in various fields. Mapmaking and route planning are the core and focus of research on the development and construction of mobile robots (Patle et al., 2019). In the robot environment with static obstacles, the robot is designed to avoid obstacles and move from the initial position to the destination position, the consumption of the minimum value and the shortest path as a large part of the study on the invention for planning the path of more than one line. Mobile robots can be equipped with various types of sensors that can implement route planning, motion control, and data transmission. Mobile robots can turn the entire work environment into a two-dimensional map with the presence of a position sensor. According to the principle of geometry, the operating system considers the mobile robot as a point and performs the planning of the mobile robot from the starting point to the destination point. The goal is to find a feasible path that connects the starting point with the destination point. Research on path planning will become popular at a time when robotic productivity is on the rise. Currently path planning algorithms mainly include geometric algorithms, artificial terrain methods, grid-based search, and sampling-based algorithms (Li et al., 2020). However, each algorithm has advantages and disadvantages that cannot be overcome in every application. The author proposes an algorithm in which some nodes of the path are closer than the feasible path. Methods that serve this purpose, as well as tools that can address field challenges. RRT is one of the sampling-based algorithms that is widely used in the field of route planning because of its ability to rapidly expand in high-dimensional dense spaces and find routes to existing destinations (Al-Ansarry & Al-Darraji, 2021). There is a problem with the degree of convergence (Al-Ansarry & Al-Darraji, 2021; Buniyamin et al., 2011).

Background and Related Work

This chapter will discuss previous research as well as literature and explanations of each algorithm that will be used to determine the characteristics of the research so as to produce research with expected outputs (Al-Ansarry & Al-Darraji, 2021). Path planning is finding a continuous path that will propel the robot from start to destination configuration (Chaari et al., 2017). All paths must run to create paths in empty spaces (Martínez Novo et al., 2022). It

is therefore described as a sequence of actions that guides the robot from the start (state) through several intermediaries to configure the destination. Which action is chosen in the current state and which state will occur next depends on the planning algorithm used and the criteria used (Naderi et al., 2015). Rapidly Exploring Random Tree (RRT) is a random sampling algorithm derived from an algorithm in path planning that explores space by generating nodes randomly and uniformly to expand the tree relatively quickly and has a good solution in the search space that avoids collisions with its obstacles (Naderi et al., 2015). But the RRT algorithm has weaknesses, namely in time and there is no smoothing used in making the path by producing a bad road and it is not recommended to be recommended in the use of robots. RRT* introduces two operations in neighbor search and rewiring to achieve an optimal path with infinite sampling. However, the PRT* algorithm cannot generate low- cost assembly lines in a short time or even invalid assembly lines, when the number of sampling points is small. A* is a heuristic search algorithm that works efficiently.

Path planning is finding a continuous path that will propel the robot from start to destination configuration. All walking paths must create paths in free space. In the planning of the mobile system using a map of the known environment, which is stored in the Storage robot. The state or configuration gives the possible poses of the robot in the environment, and it can be represented as a point in the configuration space that includes all possible states of the robot (configuration) (Le et al., 2018). The matched path is therefore described as a sequence of actions that guides the robot from an initial configuration (status) through some intermediate to configure the destination. Which action is chosen in the current state and which state will occur next depends on the planning algorithm used and the criteria used. The algorithm selects the next most suitable state from the set of all possible states that can be visited from the current state. Sampling-Based Algorithm Sampling-based algorithms have become a popular choice because of their effectiveness and capability in high-dimensional continuous spaces. This algorithm explores the environment by randomly assigning waypoints from the state space and checking whether those points can be reached by the vehicle using the steering check routine. If successful, the sampling algorithm will find the right path. Rapidly Exploring Random Tree (RRT) is a random sampling algorithm derived from an algorithm in path planning that explores space by generating nodes randomly and uniformly to expand the tree relatively quickly and has a good solution in the search space that avoids collisions with its obstacles (Chen et al., 2021). The RRT algorithm reduces the computations of the spatial path planning algorithm, RRT also has weakness, namely:

- Search with wide randomness for slow convergence speed and low path generation efficiency.
- The resulting path is only a feasible path not to find the shortest or suboptimal path.
- No smoothing is used in road construction by producing bad roads and it is not recommended to use robots.
- Smoothing is not used in road construction by producing bad roads and it is not recommended to use robots.

RRT* introduces two operations in neighbor search and rewiring to achieve an optimal path with infinite sampling. However, the PRT* algorithm cannot generate low-cost assembly lines in a short time or even invalid assembly lines, when the number of sampling points is small. A* algorithm is a heuristic search algorithm that works efficiently to generate a good grid map, using estimated node costs and the magnitude of priority costs in the selection (Hasircioglu et al., 2008).

$$f(n) = g(n) + h(n) \quad (1)$$

where:

f(n) = The smallest estimated value for the path solution along the value of n g(n) = Coordinate distance from starting point to point n

h(n) = Heuristic value between the coordinates used to estimate the distance between the initial location to the destination location.

The algorithm is generally calculated using the Euclidean distance formula. The Euclidean Distance function calculates based on the coordinates of each node (Chen et al., 2021). A heuristic function can only be accepted if the function returns a quote that does not exceed the true value. If the heuristic function produces an over value that exceeds the actual value or overestimates, it means that the search process may be lost, and the heuristic may not be optimal. Heuristic functions are great if they can provide a price quote to approximate the actual price. Which is related to the principles of the Pythagorean theory the result of the square of the hypotenuse resulting from the sum of the squares of the other sides.

Research Method

This section describes the development of the F-RRT*-A*. We will discuss the method used in the analysis of combining the F-RRT* and A* algorithms to find the optimal path by generating nodes near obstacles. In this research, the stages in problem solving consist of several stages

from the research proposal of Fast Rapidly exploring Random Trees (Fast- RRT) (Drake et al., 2018).

Moving Robot Path Planning

a. Start Program.
b. Bring up the map in The Matlab.
c. The starting point starts looking for a path from a randomly distributed node
d. Find the nearest neighbor node to a random point by meeting the conditions for stepping.
e. If a node finds an obstacle, if yes, then searches for the previous point by not evaluating the area that has been explored, generates random points if not, adds nodes and connects them.
f. If the node is close to the radius of the goal point, otherwise it will return to step 4 by looking for the nearest node, if so, it will continue to step A*.
g. If the distance between the current node (n) and A* as the start node is less than the specified value, if yes, it will return to step 4 to search again, otherwise place the current node in the open-set and use it as the start point A. *.
h. The A* algorithm starts working by counting from the current node point f(n)=g(n)+h(n).
i. Remove the current node from open-set changed to closed-set, then save index (n) from the lowest result f(n).
j. Is the current node (n) already at the goal point? if yes the path optimization will read for more optimal paths otherwise the node will search again and will check the current node(n) around whether it is closed-set back to number 8.
k. Next the goal point, the optimal path, the Fast-RRT* path and the A* search have been completed.

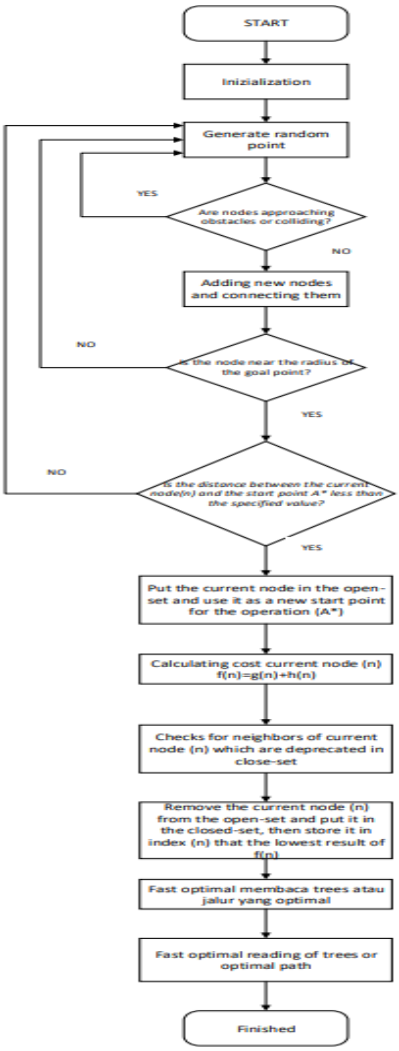

Figure 1 Flowchart F-RRT*-A*

Basic F-RRT*

In the F-RRT process there are two steps or processes to optimize the path search. First, Improved-RRT has the goal of finding a feasible path and the next step is Fast-optimal which combines the feasible path with the optimal path when approaching the optimal path position. Like Fast-sampling and Random steering and there is fast-optimal also has two parts Path fusion and path fine-tuning (Willms & Yang, 2006).

1. Improved-RRT

In this improved-RRT introduces a fast random expansion sampling method. By not evaluating the path already explored.

Algorithm 1. *Improved* RRT
Input: x_{init} , X_{goal} and *Map*
Output: A path P from x_{init} to x_{goal}
T.init()
For $i = 1 \ldots N$ do
    $x_{rand} \leftarrow$ FastSample (Map);
    $x_{near} \leftarrow$ Near(T , x_{rand});
    $x_{new} \leftarrow$ RandomSteer($x_{nearest}, x_{rand}$)
    $E_i \leftarrow$ Edge(x_{new}, x_{near})
    If ObstacleFree(Map, E_i) **Then**
        T.add Node (x_{new})
        T.add Node (E_i)
        If $x_{new} \in X_{goal}$ then

Fast-Sampling

The basic of the RRT algorithm is to take a random sample throughout the room. When a node is expanded by the RRT algorithm to calculate the distance between d and the target point. If d is less than the threshold set r, then yahoo will find the target point until the end of the search otherwise the search continues (Zhang et al., 2018). Therefore, every node that is expanded, the RRT algorithm detects in its area with the node as the center and distance from r. Then it is defined as an area that has been explored. Shows how the fast-sampling principle works, where a new node *xnew*is added in the surrounding area {x ∈ || x- $xgoal$ || < r} is an area that has already been explored Shown in Figure 2.

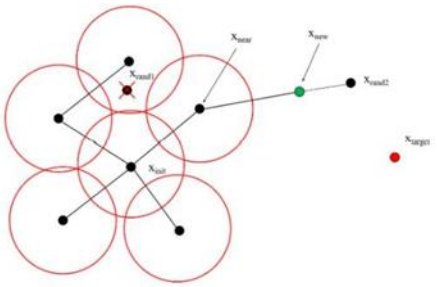

Figure 2 Illustration of Fast-Sampling

Algorithm 2. Fast Sampling
Input: x_{rand}
Let $x_{rand} \leftarrow$ UniformSample();
While $x_{rand} \leftarrow Xexplored$ ***do***
    $x_{rand} \in UniformSample()$

Fast-Optimal

This fast-optimal is very important from the Fast-RRT algorithm where as a result of searching for an algorithm with an optimal path, this fast-optimal is an asymptotic optimal which obtains a good path by combining the initial path with finding the optimal path.

Fusion Path

Path fusion has references in their respective sections on merging into a better path. In the evaluation index is the length of the path. Because path fusion is used to combine several initial paths into a shorter path. Each iteration P_{new} derives from an improved PRC due to the high efficiency of Fast-PRT. Then P_{new} obtained quickly, so that $P_{optimal}$ merges with the optimal path so that it becomes the optimal path. So, by generating a new path and combining it with the optimal path generated from the fusion path. In Figure 2 there is the first step by finding the intersection, and the second step selecting the sub-path at the first point, at the intersection point of the paths P_{athnew} and $P_{athbest}$ is calculated and used to divide the path into several sub-paths. The nodes represent two paths that do not completely overlap. If the distance between points1 and points2 is less than the specified threshold, the lines are considered to be overlapping and the midpoint is defined as the overlapping point. This step is considered to connect each intersection of the path. P_{athnew} and $P_{athbest}$. The sub-path P_{athnew} q_{start} and q_{near}. The shorter path of the Path $P_{athbest}$ is used to connect $qnear$ and q_{goal} because it shows a shorter path than P_{athnew}. With this previously there were two paths that could be combined into a better one. The current optimal path with newly generated random paths, along with rapidly acquiring near optimal paths. Shown in Figure 3.

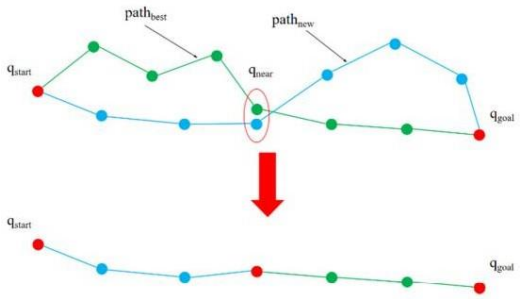

Figure 3. Illustration Path Fusion

Path fine-tuning

In optimizing a path, there is also a path alignment, which is a method that obtains a better path by creating a path obtained from a combination of paths. The path fine-tuning process can be seen in Figure 4.

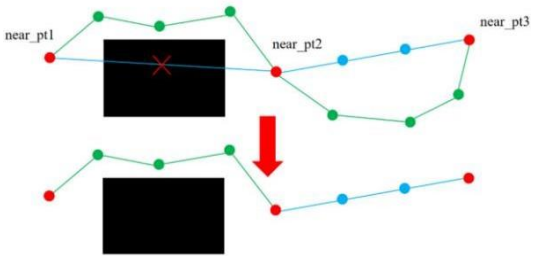

Figure 4. Illustration Path Fine-Tuning

Combining F-RRT*-A* from the results described above with the advantages of algorithms in order to produce nodes in obstacles for optimal paths by increasing node search efficiency and having good performance and accuracy.

Testing Environment

The map design in this study was made using Matlab software consisting of two maps measuring 70x70 through map 1 and map 2 with static obstacles. With parameter on map 1, Start point x, y= (8,25); Target point x, y= (35,35); Maximum nodes: 5000; Step size=1; Goal threshold=<10 (distance between A* points); Resolution = 1. In the second map, the difference is only at the start point and end point. Start point x, y= (15,0); Target point x, y= (40,25). The simulation environment built can be seen in Figure 5 and Figure 6.

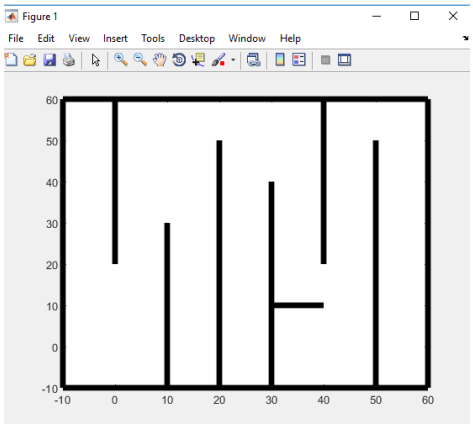

Figure 5 Map 1

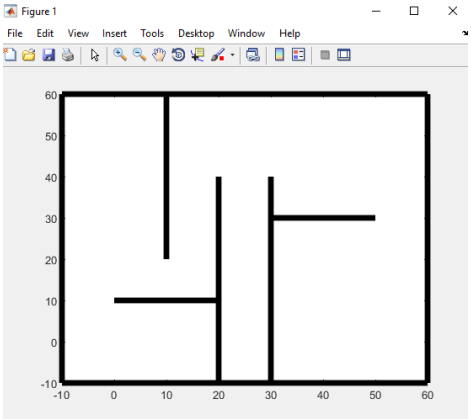

Figure 5 Map 2

Result and Discussion

The results of the research that the author has made are described in the following subsection.

A. The results of the design in this study can be seen in Figure 7, Figure 8, and Figure 9.

1. In Figure 7 it is explained that the RRT* algorithm produces an untidy path where there is no path optimization and the time taken is quite long

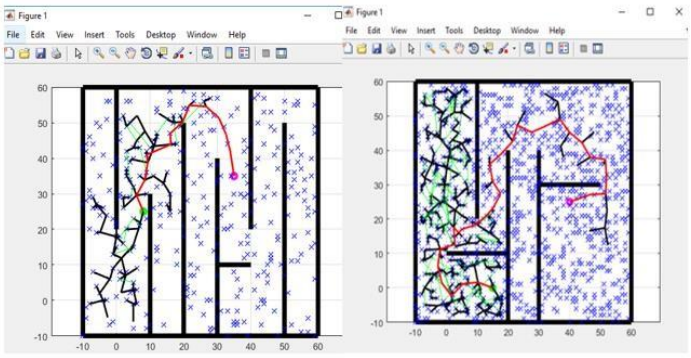

Figure 7 Graphic Result of RRT*

In Figure 8 it is clear that the A* algorithm, one of the samples taken almost all of them in timeliness, has consistency but takes a long process because the search is based on a graph or grid (searching based) where the neighbors are evaluated, each of which has 8 neighbors.

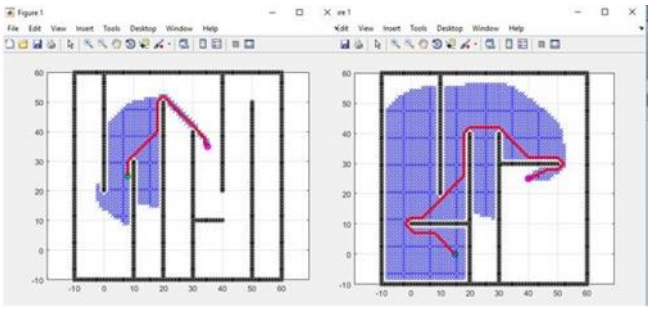

Figure 8 Graphic Result of A*

1. In Figure 9 it is explained that the Fast-RRT*-A* algorithm which is seen from the results of the path is neater because there are improvements to the PRC that cannot be found anymore if it has been explored so as to speed up the process of finding the path. After detecting the radius with the destination point, this process begins to make it easier to connect the last path that has the function used to control the A* search.

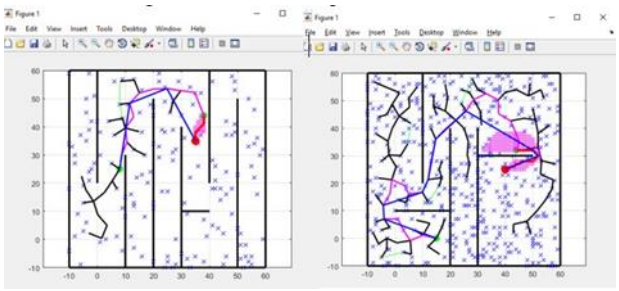

Figure 9 Graphic Result of F-RRT*-A*

The research of each group introduced a comparative analysis of the three methods which resulted in the optimization of the PRC repair path as well as the time used. Shown in the results of the RRT* algorithm in the sampling points that are distributed throughout the space, the RRT* uses more nodes to cover the whole world space, for A* because it is search based on this algorithm more to reach a destination point so it takes a long time, the three proposed algorithms Fast-RRT*-A* the sampling point of the Enhanced RRT algorithm is more densely distributed and the random tree is more efficient way to reach the state space. The mean and variance were used as two indicators for the characterization of the search time. Average time value. The search shows the average performance of the algorithm, while the variance shows proves it, i.e. both are very important for practical applications (eg, robots).

B. Research Results of Narrow Indoor Time (Map 1) Test results in the Time Test Table from Map 1.

Table 1 Time Test Table from Map 1

Method	Time (Sec)				
	Step1	Step2	Step3	Step4	Step5
RRT*	51.4191	58.6505	52.9095	48,6474	44.626
A*	45.6489	48.8553	38.7177	38.7527	40.001
F- RRT*-A*	25.1405	24.2561	20.9681	10.1411	15.9778

From the results of the time test from map 1, a comparison graph of the three algorithms can be made which can be seen in Figure 10.

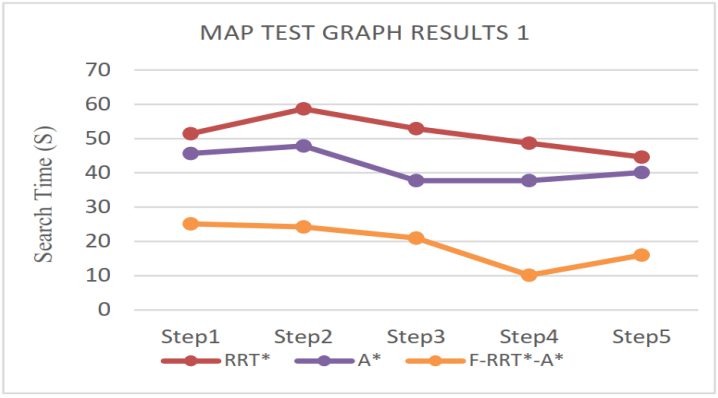

Figure 10. Time Test Graph from Map 1

In Figure 10, the results of the comparison of Fast-RRT*-A* time performance have a very low time value significantly compared to other algorithms. Therefore, the Fast-RRT*-A* algorithm can find a more feasible and faster path than the other two RRT* and A* algorithms.

C. Research Results in Narrow Indoor Time (Map 2)

The test results are in the Time Test Table from Map 2.

Table 2 Time Test Table from Map 1

Method	Time (Sec)				
	Step1	Step2	Step3	Step4	Step5
RRT*	203,8385	373,7892	475,4995	430,056	430,056
A*	100,3057	102,6468	99,755	101,94	101,94
F- RRT*-A*	87,0402	73,362	71,0764	68,512	68,512

From the results of the time test from map 2, a comparison graph of the three algorithms can be made that can be seen in Figure 11.

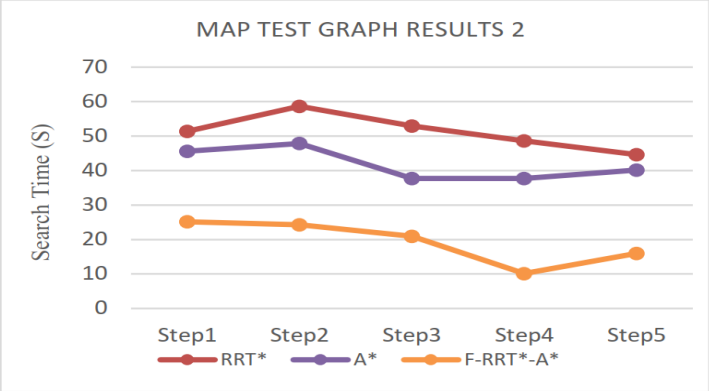

Figure 11 Time Test Graph from Map 2

In the second map test the results show that Fast-RRT*-A* is still superior among other algorithms, so the proposed Fast-RRT*-A* algorithm is able to produce an optimal path with a relatively fast time according to the data above.

Conclusions

The conclusions of this study are as follows: (1) The working principle of the algorithm is in accordance with the system design. (2) Fast-RRT* where to increase speed and stability in finding lanes. (3) Added a new algorithm to find a new optimal path to get a near optimal path by combining and adjusting several feasible paths. (4) By checking each Fast-RRT* node to check if it belongs to the destination region with a certain threshold. When reaching this region or radius then the A-star algorithm is activated. Suggestions for developing this final project is that the RRT Algorithm has great potential in practical motion planning applications as well as a broad scope to develop more optimal results.

References

Al-Ansarry, S., & Al-Darraji, S. (2021). Hybrid RRT-A*: An Improved Path Planning Method for an Autonomous Mobile Robots. *Iraqi Journal for Electrical & Electronic Engineering*, *17*(1).

Buniyamin, N., Ngah, W. 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.

Chaari, I., Koubaa, A., Bennaceur, H., Ammar, A., Alajlan, M., & Youssef, H. (2017). Design and performance analysis of global path planning techniques for autonomous mobile robots in grid environments. *International Journal of Advanced Robotic Systems*, *14*(2), 1729881416663663.

Chen, J., Tan, C., Mo, R., Zhang, H., Cai, G., & Li, H. (2021). Research on path planning of three-neighbor search A* algorithm combined with artificial potential field. *International Journal of Advanced Robotic Systems*, *18*(3), 17298814211026449.

Drake, D., Koziol, S., & Chabot, E. (2018). Mobile robot path planning with a moving goal. *IEEE Access*, *6*, 12800-12814.

Hasircioglu, I., Topcuoglu, H. R., & Ermis, M. (2008). 3-D path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms. Proceedings of the 10th annual conference on Genetic and evolutionary computation,

Le, A. T., Le, T. D., & Roka, R. (2018). Search-based planning and replanning in robotics and autonomous systems. *Advanced Path Planning for Mobile Entities*, 63-89.

Li, Y., Wei, W., Gao, Y., Wang, D., & Fan, Z. (2020). PQ-RRT*: An improved path planning algorithm for mobile robots. *Expert Systems with Applications*, *152*, 113425.

Martínez Novo, Á., Lu, L., & Campoy, P. (2022). Fast RRT* 3d-sliced planner for autonomous exploration using MAVs. *Unmanned Systems*, *10*(02), 175-186.

Naderi, K., Rajamäki, J., & Hämäläinen, P. (2015). RT-RRT* a real-time path planning algorithm based on RRT. Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games,

Patle, B., Pandey, A., Parhi, D., & Jagadeesh, A. (2019). A review: On path planning strategies for navigation of mobile robot. *Defence Technology*, *15*(4), 582-606.

Willms, A. R., & Yang, S. X. (2006). An efficient dynamic system for real-time robot-path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *36*(4), 755-766.

Zhang, H.-y., Lin, W.-m., & Chen, A.-x. (2018). Path planning for the mobile robot: A review. *Symmetry*, *10*(10), 450.