Journal of Basrah Researches ((Sciences)) Vol. (47). No. 1 (2021)

Available online at: https://www.iasj.net/iasj/journal/260/issues

ISSN –1817 –2695

Received date: 6-5-2021 Accepted date: 26-7-2021 Available online date: 31-12-2021

MT Hybrid RRT-A* Regression-based: An Enhanced Path Planning Method for an Autonomous Mobile Robots

Suhaib Al-Ansarry[1], Salah Al-Darraji[2]

[1,2]**Basrah University, College of Education for Pure Science, Computer Science Dept.**

E-mail : pgs2186@uobasrah.edu.iq, aldarraji@uobasrah.edu.iq

Abstract

In Robotics, especially the autonomous mobile robot, navigation is the crucial part. Path planning -as a navigation sub-field- is considered one of the most important research fields. It aims to find the shortest obstacle-free path that agrees with the dynamic constraints for the robots moving between the starting and goal points. Rapidly-exploring Random Tree (Basic RRT) is one of the widely sampling-based planners used in the field of path planning, due to its ability to plan the path within high dimensional environments efficiently. On the other hand, Basic RRT suffered from the high cost and convergence rate, besides the non-optimal path problem (i.e. long-length and oscillating). In this context, the method of Hybrid RRT-A is an improved variant of the Basic RRT which also has a problem with the path optimality that needs to be enhanced. In this paper, we proposed a method called (MT Hybrid RRT-A * regression-based) to overcome this problem and generate the near-optimal path. The length of the path that was remarkable in the worst scenario case for the MT Hybrid RRT-A-star RB was **(101** unit**)**, whereas for the Basic RRT and Hybrid RRT-A-star was about **(134** and **121)** respectively. Moreover, the concept of Multithreading programming was presented to achieve further -cost and convergence rate- enhancement. In the* worst scenario case, *the consuming time for (1000 execution) has been reduced for about **(12.5%)***, while the number of the lowest time cases of the Multithread version against Singlethread for **(**50 execution**)** was **(44:6)**.

KEYWORDS: Path planning, Mobile robot, Configuration space, Regression-based, Cost rate, Convergence rate, Grid-map.

I. Introduction

In robotics, navigation and path planning are the most popular issues. The main challenge of path planning is finding a feasible path that will take the robot from the point of start to a desired final state. In general, there are two classes of planners. First, the Online-planner, which does not have an overall view of the environment map. It just moves and collects the information depending on sensors, examples are the Potential Fields [1,2]. The main Online-planners issue is about falling in the local minimum region. Offline-planner is the second type, where the robot has the entire information about the environment map before starting its navigation. These planners can deal with the local minima problem, but it is computationally intense. Rapidly exploring Random Tree (RRT) [3] is one of the most efficient classical Offline-planners. This planner tries to explore the configuration space incrementally, then constructing the search tree. Even though Basic RRT is computationally intense, but it can find feasible routes also for those environments characterized by cluttered obstacles. Recently, many versions of Basic RRT have been introduced to improve the operation of search and path length.

Hybrid RRT-A* [4] is an improved extension of Basic RRT. This hybrid method has the same Basic RRT features, while it solved the problems of cost and convergence time. In this research, MT Hybrid RRT-A* RB is proposed to enhance equivalent the path problems (length and oscillating) of the previous method, and to further enhanced cost and searching time.

Our contributions can be summarized as follows:

1. Relying on the method of Multinomial Logistic Regression-based (RB), we proposed the method of (Hybrid RRT-A* RB) to improve and smooth the initial path of the Hybrid RRT-A*.
2. As a practical improvement, exploiting multi-cores to further enhancing the convergence rate using the concept of Multithreading (MT) by scanning the environment many times concurrently in (MT Hybrid RRT-A* RB).
3. Evaluate the performance of the Hybrid RRT-A* RB, and MT Hybrid RRT-A* RB by comparing the results of the cost, convergence, and path-length in each of these methods with the Hybrid RRT-A* under the same conditions (simulation and computer hardware).

The rest of this paper is organized as follows. Section II presents the related work of the mobile path planning methods. Section III describes our proposed methods (Hybrid RRT-A* RB, and MT Hybrid RRT-A* RB) within the same environmental settings. Section IV outlines the maps, setups, and results of experiments. Section V discusses the final results. Section VI presents the conclusions and future works.

II. Related work

Rapidly exploring Random Tree (Basic RRT), is developed by Steven M. LaValle and James J. Kuffner [4,5]. This method constructs simply a search tree by generating random samples (nodes) from the workspace depending on the Uniform distribution, then tries to link these nodes. If the connection is feasible, the joining will be made to the nearest node that passes completely through free space without any collision. Otherwise, that node is discarded. This process is repeated until the final state is reached.

1. Path Smoothing

Target Attractive Force (TAF-RRT) is an enhanced technique introduced by Zhang et al. [6]. It is used to deal with the problem of growing trees in multiple directions as a result of the widespread sampling by guiding the sampling operation toward targets away from colliding regions or non-relevant areas. This technique is used to shorten the path of Basic RRT and produce an optimal collision-free one. Finally, this technique is combined with the method of Dynamic Step-size (DS-RRT) to enhance the overall performance of the Basic RRT as in many previous studies that worked to develop the same idea within different ways [7,8].

Zhang et al. [9] proposed another improved form of Basic RRT basing on the use of the New Metric Function (NMF). This technique helps to produce a near-optimal path and solved the issue of sharp turn for Basic RRT efficiently. Its work looks like the Euclidean distance function of the Basic RRT, which computes the distance between a random node and the nearest node in each iteration until reaching the goal without considering the angle of rotation during sampling, which leads sometimes to difficult turns, then the inability of the robot to respond due to the kinetic restrictions. Therefore, NMF adds the degree of rotation angle beside the distance-cost through specific normalization to make the path as short and smooth as possible compared to Basic RRT.

In 2018 Wei et al. [10] proposed an improved version of Basic RRT called Smoothly RRT (S-RRT). The S-RRT method has two phases. First, the nodes enhancement strategy, which is used to guide the growth of the search tree towards the goal area efficiently. It helps to prune useless nodes using the (constraint of maximum-curvature) as a pruning algorithm which makes the initial path as straight and short as possible. In each iteration, the algorithm checks whether the random nodes are located within the obstacle region or not, to connect the last node of the tree that satisfies this condition with the first one using a straight line. Otherwise, links the node in the intersection area and makes it a new starting point for the next iteration. B-spline which has been shown in detail at [11], is used at the next phase by inserting the control points to the route if some hard turns are there to refine the final path and make it much smoother compared to Basic RRT.

Wang et al. [12] introduced an improved form of Basic RRT called (NC-RRT). Generally, the work of this method is similar to the Basic RRT, besides using two additional principles which greatly improve the performance of the resulting method. Initially, the method of gradually Changing the Sampling Area (CSA-RRT) is proposed to guide the rapid search tree to the goal point. The radius of the surrounding sampling area is represented by the guidance factor to help in determining the search area, through which the other tree nodes are expanded to reduce the number of nodes within this range, then increasing the guidance accuracy and speed in reaching the goal. The other principle is the coefficient of Node Control (NC), which controls the number of expanding branches from the original tree. It increases or decreases the branches depending on the local situation. Initially, the value of this parameter is (1) by default which helps to prevent the expansion towards undesirable directions. However, when facing a situation of local-minima, the area of search is expanded based on the previous technique (CSA) besides increasing the value of the control-factor to allow branching in many directions, which will address this problem finally.

2. Multithreading

Devaurs et al. [13] introduced the concept of multithreaded programming through two kinds of parallelism. First, is the "OR" operator, which is described simply by using a private copy of a search tree with every single thread. This kind gives good results with the bidirectional approaches such as RRT-connect. The second one is represented by the "AND" operator, which is divided into two sub-kind (Distributed-RRT and Manager-Worker RRT). The first kind is parallelizing every step of the operation of tree search and check at each iteration whether the main process (local tree) has received valid nodes from other processes or not. If the condition has been met, then the process uses these nodes to expand the local tree and broadcast this state. In the end, when the first process reaches the goal, a stop message will be broadcasted. In the Manager-Worker RRT, the process which has full access to the main tree is the "Manager". It performs the process of tree construction, besides delegating "Workers" to do the sub-search computations like the search for the nearest neighbor as in RRT* [14,15]. Basically, the expansion is the most costly phase of RRT creation. In each iteration, the “Manager” checks whether new nodes have been received from the “Workers” or not, then operates to determine the best neighbor one to be certified among those nodes. Finally, upon reaching the goal node, a termination message will be broadcast. Otherwise, the "Workers" will continue as long as they did not receive that message.

In 2019 Casalino et al. [16] proposed a multithreading-based form of Basic RRT called (MT-RRT), which aims to illustrate the abilities of the "General-Purpose Library" in presenting four separate multithreading techniques to speed up the exploration process of various RRT forms. First, the "Parallel Exploration on a Shared-tree", where each step of the algorithm is paralleled and then executed all at the same time.

It is preferred to apply this way for the Basic RRT form. The second is the "Parallelization of the Querying Activities" that can be represented in executing collectively among multiple threads for searching the nearest neighbor and select it. The best result of this implementation can be achieved in RRT*. The performance of this technique is poor, especially when it comes to close or re-open a new region every time, while it's efficient in running multiple threads at the start of algorithm work, by using a shared condition. Third, the "Parallel Expansions of Copied-trees", which presents a version preferred to use in the case of bidirectional schemes, where every thread has its own copy of the tree. Once a thread has been added a new node to its own tree, copies of (P – 1) shall be calculated and sent to other threads. The "Blinded Multiagent Exploration" is the last technique that exploiting large numbers of threads, besides decreasing the need for synchronization and memory sharing. On the other hand, a thread has only a small part of the tree knowledge. As a result, we will use several threads to concurrently explore an area for a specific number of iterations. In general, there is no particular technique that can be applied for all planning methods efficiently. Each strategy has its own features besides disadvantages. Therefore, relying on the planner used and environment nature, the most appropriate technique can be determined.

III. The Proposed Method

The version of Hybrid RRT-A* which is used for robot path planning is developed by Al-Ansarry, and Al-Darraji [4]. This method is a single-query scheme, takes a short convergence time, and can explore the entire configuration space with the lowest number of nodes. In this section, an enhanced method called (MT Hybrid RRT-A* RB) is proposed. This method consists of two stages. In the first stage, the principle of regression-based (RB) is presented by combining the method of Multinomial Logistic Regression (MLR) with the Hybrid RRT-A* planners. While Hybrid RRT-A* advantages are preserved, the MLR features are employed to solve the problem of bad Hybrid RRT-A* path length as well as decreasing the resulting path oscillating. In the second stage, the concept of Multithreading programming (MT) is applied to achieve further cost and time rate enhancement.

*A. Regression-based Hybrid RRT-A**

The issue of path non-optimality (in terms of length & smoothness) as shown in Figure 1, is a drawback that needs more research to make Hybrid RRT-A* be enhanced. The first stage of our proposed method attempts to shorten the path that is resulted from Hybrid RRT-A* and remove its oscillations. The principle that has been adopted to achieve this job is the method of Multinomial Logistic Regression. In statistics, this method is considered as a classification technique that generalizes the regression to multiclass problems. This means that regression is used as a model to predict the different potential results of a distributed dependent-variable classificationally (classification-based).

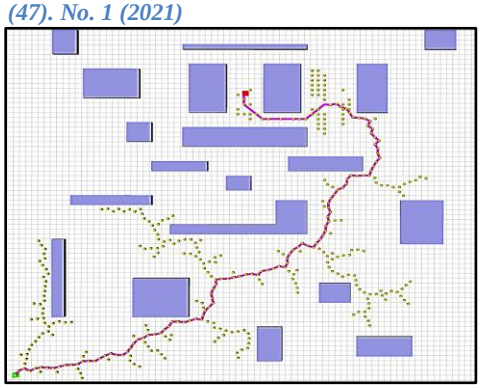

Figure 1: Path oscillations of Hybrid RRT-A* algorithm.

The main issue that faced most Basic RRT forms, as well as our proposed method is their inability to plan an optimal path within known environments. This kind of problem makes the use of these methods inefficient, especially with the robot's time and energy restriction. In this section, we introduce an enhanced Hybrid RRT-A* regression-based method to improve the initially planned path. The optimality is achieved by linking the points chain of the path produced initially by Hybrid RRT-A* as the way described in Figure 2.

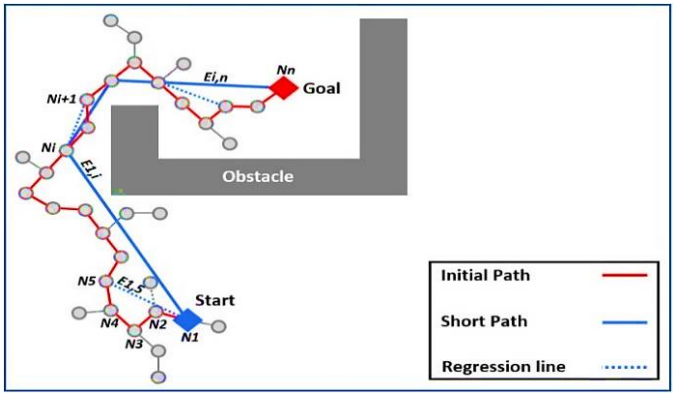

Figure 2: The process of regression search method.

All points from the start configuration to the goal are sequentially linked. The starting point is linked to the next one using a straight edge-line. If this line does not intersect any obstacle, then the point of start will be reconnected to the next-point repeatedly using a new edge-line at each time, and saving the last straight edge-line as a near-optimal path can reach the goal. But, if any edge-line facing an obstacle, the model will adopt the last known point as a new starting point for the regression search in the same way. Suppose that nodes $N_i \in \{N_1, N_2, N_3..., N_n\}$ are the points that are linked sequentially and represent the initially planned path by Hybrid RRT-A* that guides the robot to move along with it until reaching the goal safely. Depending on the method of regression search, the initial point N_1 as a starting point will be linked to the next point N_2 using the edge-line $E_{1,2}$. Then check whether $E_{1,2}$ facing an obstacle or not? If it does not, then the model will reconnect N_1 to N_3 using $E_{1,3}$, and repeat the step above in the same way, until checking the edge-line $E_{1,i+1}$ then stop. Because the $E_{1,i+1}$ facing an obstacle, and the possible near-optimal path is the $E_{1,i}$. This means that N_i is the

endpoint, and due to the node N_{i+1} is not the goal point, so the new starting point is N_i which will be linked to the next point N_{i+1} in the same way. Therefore, the near-optimal path is the edge-line $E_{1,i}$, and $E_{i,n}$. In this way, the robot will move between $E_{1,i}$, and $N_{i,n}$ smoothly which leads to a decrease in the required energy and time, due to the short distance between $E_{1,i}$, and $E_{i,n}$. Figure 3 shows through a block diagram the main regression search step.

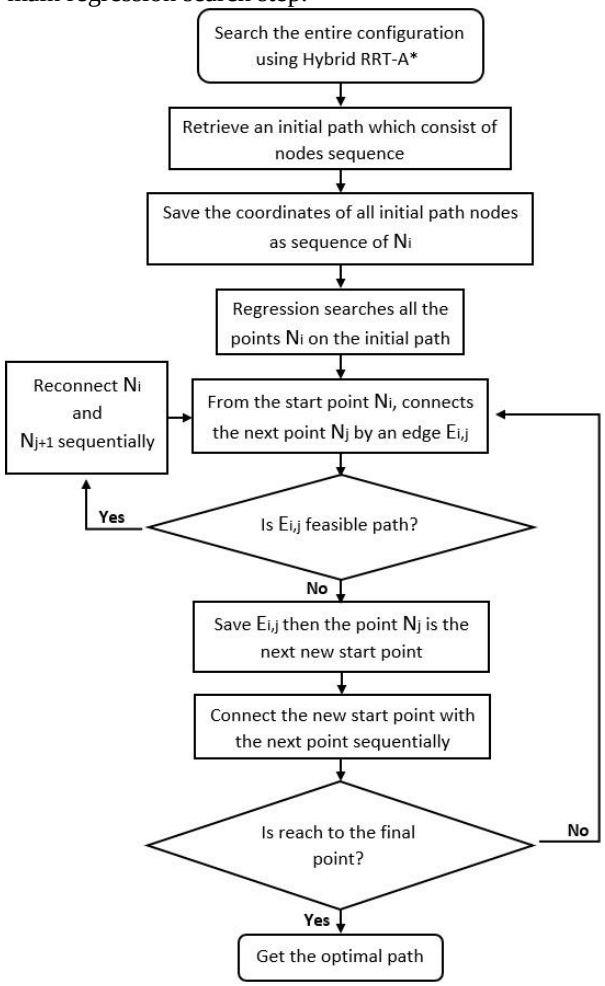

Figure 3: The algorithm of regression search.

*B. Multithreaded Regression-based Hybrid RRT-A**

In general, the version of the Hybrid RRT-A* planner takes a short convergence time with a lower computational cost compared to the most classical path planning methods. In this section, we introduce the (MT Hybrid RRT-A* RB) method, which represents the second stage of our proposed method, through applying the concept of Multithreading programming (MT) to achieve an extra decrease of convergence rate (time) as well as the cost for the Hybrid RRT-A* method.

Essentially, this concept may decrease the total computational time and cost by dividing or distributing the main program into two or more sub-part each called (thread), where every thread represents a distinct execution part. Then distributing those threads on multi-processors or multi-cores to run them concurrently, which reduces the total computational load on each processor and core finally. The basic idea is how to use the maximum number of available resources for processing purposes. When there is an available multi-core processor or more than one, the process of executing programs on a single processor or core is considered a waste of resources. Several manners can be adopted to represent any program depending on the multithreading concept [16]. The one that is adopted in this context is by subdividing the main program into several parts based on the number of main tasks (Objects), where each task is executed on a single core simultaneously to reduce the computational load for the main program. Another way is by distributing several complete copies of the main program - each one as a separate task- then assigning each copy to operate separately on a different core simultaneously to obtain the best execution result among all other cores. In this research, as shown in Figure 4 we prefer to use the second fashion of multithreading representation to produce an enhanced method that further increases the Hybrid RRT-A* RB cost and convergence rate significantly. As long as the behavior of this method is characterized by randomness, this means that the presence of more than one copy being executed at the same time will lead definitely to obtain a single execution which has a convergence rate considered the lowest among several synchronous executions, through applying the concept of the multithreading.

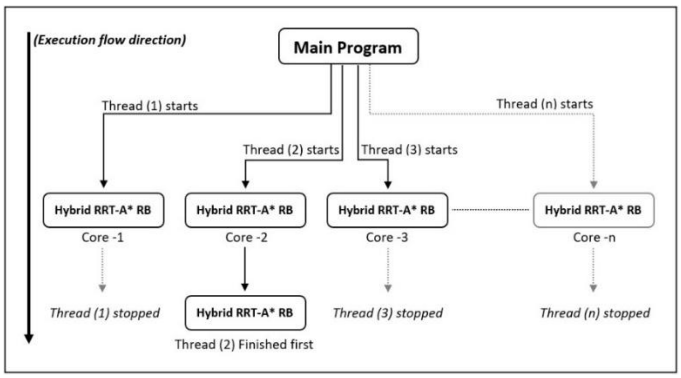

Figure 4: Enhanced Hybrid RRT-A* multithreading process.

That is means, each core executes its copy of the program sequentially, then the one that terminates first and finds the goal-point with a minimum time will stop the rest cores. In this way, we invest all available processing resources for further enhancement of Hybrid RRT-A* RB cost and convergence time.

IV. Experiments and Results

In this section, we evaluate the overall performance of the MT Hybrid RRT-A* RB method.

A. Setup and Maps

The experiments are carried out using Robot Operating System (ROS Kinetic). This framework runs under the operating system of Linux (64-bit), with (8Gb) RAM, and

processor of Intel Core-i7 8th Gen. The ROS visualization environment (Rviz).

The simulation of planning the path is divided into three groups. Group-1 consists of two experiments: First, for the (Basic RRT, Hybrid RRT-A*, and Hybrid RRT-A* RB) methods. Second, for the (Hybrid RRT-A* RB and MT Hybrid RRT-A* RB) methods. All experiments are conducted using a grid environment map of size 75*75 through Map-1. Group-2 and 3 repeat the same above methods, experiments are applied again on the same grid dimensions through Map-2 and Map-3. In all group experiments, the above-mentioned methods are examined by implementing them in different scenarios, then evaluated the performance results of those methods basing on several factors (cost, convergence, and path length). In these maps, the gray cells represent the obstacles region, while the red square denotes the goal point, the green square shape refers to the point of start. Finally, the rest grids are representing the region of free obstacles.

Many parameters have to be initialized: Start point= (74,74); Goal point= (37,15); Step size=1; Max iteration no.=30000; Obstacle padding=1; Robot frequency=0.05; Goal biasing=0.25; Goal threshold=13 (goal area radius); A* threshold=7 (minimum distance between two A* start points). The number of obstacles and their locations is varying from one to another (Map-1, Map-2, and Map-3) of all groups. Each of these maps will be used with the previous methods for planning the robot path. The results of search operations and resulted paths of all experiments will be discussed at the end of the next section.

B. Experiments

All experiments are applied on a grid environment-map of size 75*75 as shown in Figure 5 with a number of scattered obstacles to be used in the simulation process of finding the robot near-optimal path.

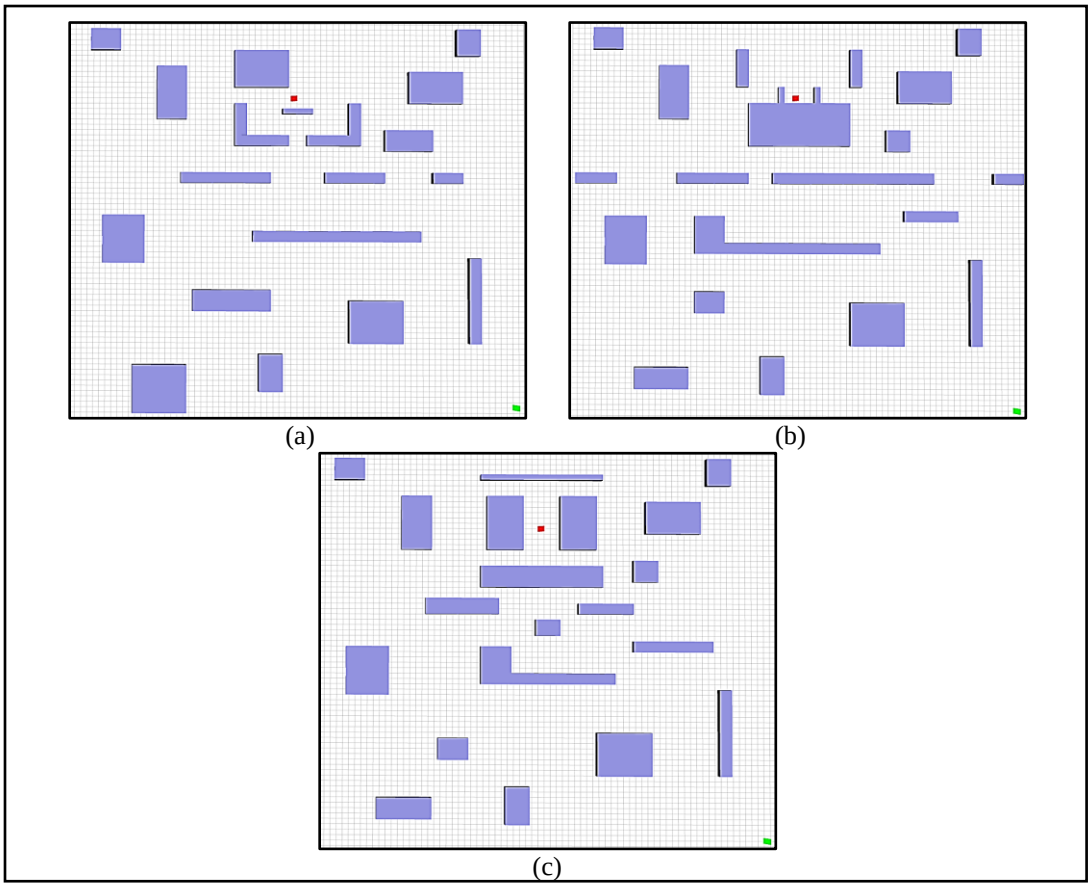

Figure 5: 75*75 grid environment (a) Map-1, (b) Map-2, and (c) Map-3.

In each group of experiments, four methods are applied (Basic RRT, Hybrid RRT-A*, Hybrid RRT-A* RB, and MT Hybrid RRT-A* RB) through two different experiments to explore the whole environment-map and reach the final state with a low cost and time rates, as well as trying to find the safe near-optimal path between source and destination.

Experiment-1 of all groups presents an analytical comparison among the results of the three methods depending on the path length for Basic RRT, Hybrid RRT-A*, Hybrid RRT-A* RB. Figure 6 illustrates the final path of these methods, and the results in Table 1 demonstrate that the process of regression-based helped to achieve the near-optimal path, by reducing the length of the Hybrid RRT-A* initial path and smoothen it. These results are plotted as shown in Figure 7.

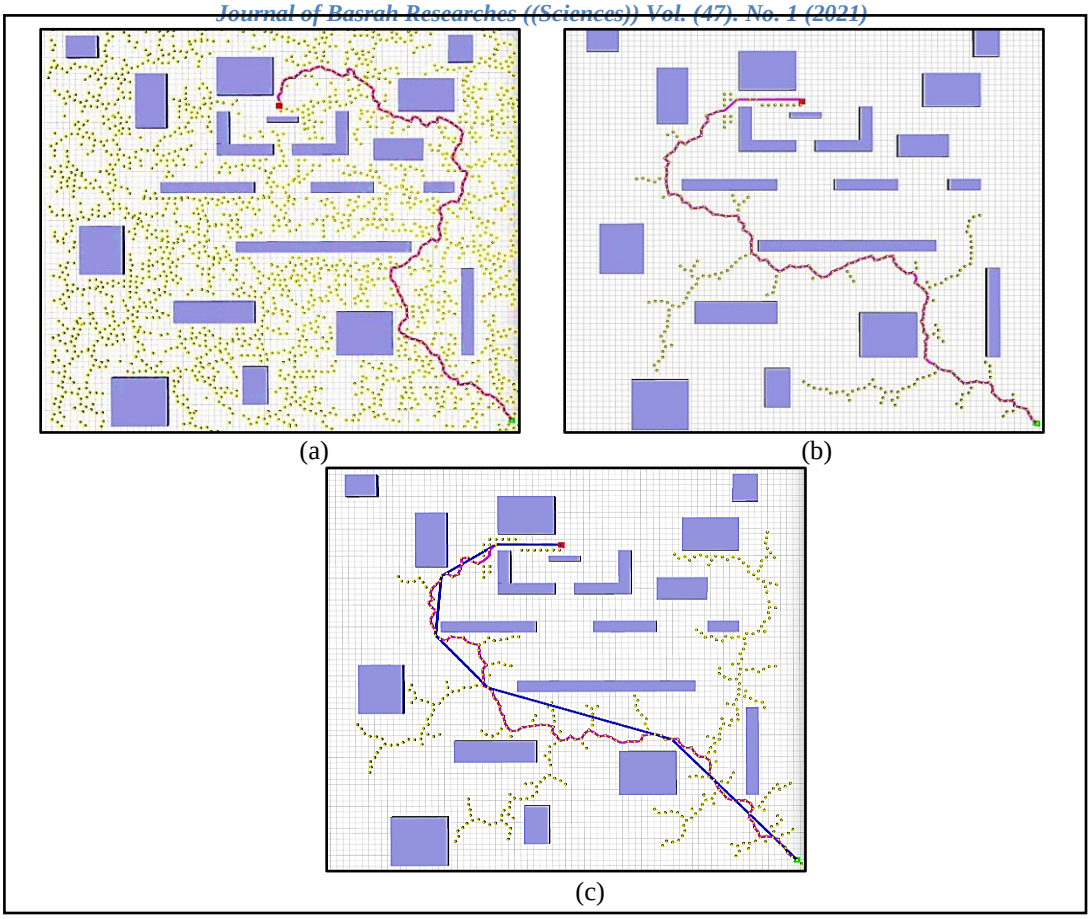

Figure 6: Final paths of (a) Basic RRT, (b) Hybrid RRT-A*, and (c) Hybrid RRT-A* RB in Map-1.

TABLE 1: Experiment results of Path-length rates.

Methods	Map-1	Map-2	Map-3
	Length (unit)	Length (unit)	Length (unit)
RRT	118.488	133.983	134.389
Hybrid RRT-A*	112.559	115.072	121.367
Hybrid RRT-A* RB	93.515	94.050	101.421

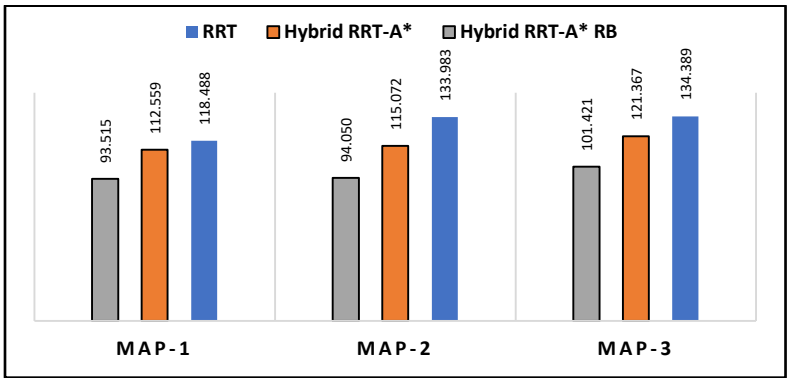

Figure 7: Summary graph for Path-length rates.

Experiment-2 of the three groups used the same previous settings and conditions with the Hybrid RRT-A* RB and MT Hybrid RRT-A* RB. The results in Table 2 show the time-consuming of (1000 execution) using one thread on a single core compared to (125 execution) on (8 cores) using (8) threads simultaneously, which is equivalent to (1000 executions) at the end. Obviously, using the multi-threaded version reduced the total time-consuming to (12.5%) when achieving (1000 execution). On the other hand, when a single-thread version (ST) of the Hybrid RRT-A* RB is executed sequentially along with a multi-threaded version (MT) at the same time for one execution, then the probability of finding an

optimal result to one thread on a specific core of the MT Hybrid RRT-A* would be large compared to the ST Hybrid RRT-A * RB result which operates sequentially.

TABLE 2: ST Hybrid RRT-A* & MT Hybrid RRT-A* time-consuming comparison.

Maps	ST Hybrid RRT-A*	MT Hybrid RRT-A*
	Time Consuming (min.)	Time Consuming (min.)
Map-1	64.058	13.168
Map-2	76.991	14.997
Map-3	82.076	15.935

V. Discussion

In general, Hybrid RRT-A* initially applies the obstacle avoidance method and if the path intersecting some obstacles, then it re-plans the path around the obstacles towards the available free spaces. This study used maps of size (75*75) with about (18-30) obstacles designed to simulate the worst cases and prevent the path from extending straight between the start and goal points to serve this purpose. The lengths of the paths are varying according to the scenario case and the regression method used which leads lastly to get the near-optimal path. The experiments of (1000 executions) are analyzed to compute the path length of each method. Eventually, the results proved that the path of our proposed method has been improved.

On the other hand, if the architecture of the robot's processor consisting of more than one core, then running Hybrid RRT-A* method sequentially is considered a waste of resources. Therefore, the multi-threaded version (MT Hybrid RRT-A*) reduced the total time-consuming to (12.5%) and helps to achieve a further enhancement in the cost and convergence rate. The results of (50 execution) show that this version achieved the lowest convergence through taking the advantage of multi-core architecture. In Map-1 which represents the best scenario, the results show that the number of the lowest time cases of MT against ST is (46:4). In Map-2 which represents the more complicated scenario, the number of the lowest time cases of MT against ST is (45:5). While in Map-3 which represents the worst scenario, the number of the lowest time cases of MT against ST is (44:6). In Table 3 below, we compared our method (MT Hybrid RRT-A* RB) with several earlier sampling-based algorithms depending on the characteristics of each one.

TABLE 3: The characteristics comparison of sampling-based algorithms.

Algorithm	Completeness	Fast Convergence	Near Optimality
RRT	Yes	No	No
RRT*	Yes	No	Yes
PRM [17]	Yes	No	No
PRM*[18]	Yes	No	Yes
RRG [14]	Yes	No	Yes
MT Hybrid RRT-A* RB	Yes	Yes	Yes

VI. Conclusion

In this paper, we proposed (MT Hybrid RRT-A* regression-based) as an enhanced hybrid path planning method. The efficiency of this planner is increased, so the present search of our proposed method has been improved. This is done by using the concept of Multithreading programming for further enhancement. Moreover, our proposed method has used the principle of the Multinomial Logistic Regression to enhance the initial path and make it as smooth and short as possible (near-optimal). We evaluate the performance of this method (MT Hybrid RRT-A* RB) with the Basic RRT, Hybrid RRT-A*, and Hybrid RRT-A* RB algorithms in three challenging environment maps. Finally, the Simulations results prove that (MT Hybrid RRT-A* RB) achieves the best cost and convergence compared to the Basic RRT and Hybrid RRT-A*.

In future work, we aim to check the validation of the proposed method with a real robot. We also aim to make the parameters of this method are tuned adaptively according to the environment size and the number of obstacles. This means, that the robot can tune the parameters automatically by analyzing the current situation as it moves based on reinforcement learning. Aiming also to optimize the shortest path and make it much smoother by using kind of curve methods like (B-spline).

References

1. Khatib, Oussama. "Real-time obstacle avoidance for manipulators and mobile robots." Proceedings. IEEE International Conference on Robotics and Automation. vol. 2, IEEE, 1985.
2. Khatib, Oussama. "Real-time obstacle avoidance for manipulators and mobile robots." Autonomous robot vehicles. Springer, New York, NY, pp. 396-404, 1986.
3. LaValle, Steven M. "Rapidly-exploring random trees: A new tool for path planning.", pp. 98-11, 1998.

4. Suhaib, A. and Salah, A. " Hybrid RRT-A*: An Improved Path Planning Method for an Autonomous Mobile Robots." Iraqi Journal for Electrical and Electronic Engineering, Iraq vol. 17, no. 1, pp. 107-115, 2021.
5. LaValle, Steven M., and James J. Kuffner Jr. "Randomized kinodynamic planning." The international journal of robotics research vol. 20, no. 5, pp. 378-400, 2001.
6. Zhang, Zhen, Defeng Wu, Jiadong Gu, and Fusheng Li. "A path-planning strategy for unmanned surface vehicles based on an adaptive hybrid dynamic stepsize and target attractive force-RRT algorithm." Journal of Marine Science and Engineering. vol. 7, no. 5, pp. 132, 2019.
7. Tang, Xiaoya, and Feng Chen. "Robot Path Planning Algorithm based on Bi-RRT and Potential Field." IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2020.
8. Yafei, Lu, Wu Anping, Chen Qingyang, and Wang Yujie. "An Improved UAV Path Planning method Based on RRT-APF Hybrid strategy." In 5th International Conference on Automation, Control and Robotics Engineering (CACRE), pp. 81-86. IEEE, 2020.
9. Zhang, Pengchao, Chao Xiong, Wenke Li, Xiaoxiong Du, and Chuan Zhao. "Path planning for mobile robot based on modified rapidly exploring random tree method and neural network." International Journal of Advanced Robotic Systems. vol. 15, no. 3, 2018.
10. Wei, Kun, and Bingyin Ren. "A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm." Sensors. vol. 18, no. 2, pp. 571, 2018.
11. Berglund, Tomas, Håkan Jonsson, and Inge Soderkvist. "An obstacle-avoiding minimum variation b-spline problem." International Conference on Geometric Modeling and Graphics. Proceedings. IEEE, 2003.
12. Wang, Xinda, Xiao Luo, Baoling Han, Yuhan Chen, Guanhao Liang, and Kailin Zheng. "Collision-free path planning method for robots based on an improved rapidly-exploring random tree algorithm." Applied Sciences. vol. 10, no. 4, pp. 1381, 2020.
13. Devaurs, Didier, Thierry Siméon, and Juan Cortés. "Parallelizing RRT on distributed-memory architectures." IEEE International Conference on Robotics and Automation. IEEE, 2011.
14. Karaman, Sertac, and Emilio Frazzoli. "Incremental sampling-based algorithms for optimal motion planning." Robotics Science and Systems vol. 104, no.2, 2010.
15. Karaman, Sertac, and Emilio Frazzoli. "Sampling-based algorithms for optimal motion planning." The international journal of robotics research. vol. 30, no. 7, pp. 846-894, 2011.
16. Casalino, Andrea, Andrea Maria Zanchettin, and Paolo Rocco. "MT-RRT: a general purpose multithreading library for path planning." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019.
17. Kavraki, Lydia E., Petr Svestka, J-C. Latombe, and Mark H. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." IEEE transactions on Robotics and Automation. vol. 12, no. 4, pp. 566-580, 1996.
18. Bohlin, Robert, and Lydia E. Kavraki. "Path planning using lazy PRM." In Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), vol. 1, pp. 521-528. IEEE, 2000.