Efficiency Considerations of an Offline Mobile Robot Path Planner

Pejman Kamkarian
Department of Electrical and Computer Engineering,
Southern Illinois University, Carbondale, IL 62901, USA

Henry Hexmoor
Department of Computer Science,
Southern Illinois University, Carbondale, IL 62901, USA

***Abstract*— The aim of disseminating this research article is to showcase a novel path planner method, which is shown to be an efficient offline path planner in terms of its capacity for analyzing workspace robot maneuvering skills and constructing collision free trajectories that yield the shortest possible path from initial point toward goal configuration. The determined route is considered to be adequately secure such that it enables the mobile robot to maneuver among obstacles in the workspace without dangers of encountering a near miss. In addition, this paper evaluates our novel path planner algorithm abilities and skills by examining it against different workspaces. We assess our novel path planner by comparing it to two of the most popular planners with the purpose of revealing its capability to route trajectories in regards to building optimal trajectory distances from initial to the goal configurations.**

***Keywords*— Path planning, Rapidly Optimizing Mapper, Robot path planning, Robot trajectory builder**

I. Introduction

A regular robot that is capable of performing an assigned task typically consists of many units that cooperate together for the sole purpose of enabling it to successfully achieve its missions. As an important functionality that plays a vital role for a robot to function appropriately, robust path planning is essential. The central task of a path planner is to analyze the robot's surrounding using equipped sensors and plan a secure and reasonable trajectory that guarantees a safe traversal for the robot from initial point to the goal configuration. The concept of security for the calculated path is commonly understood to be on a collision less trajectory that the path planner determines corresponding to the accuracy and reliability of detecting objects around the robot while moving toward its goal. The traversal distance of a path that is computed by the path planner component is directly related to the methodology employed by the planner when it composes the trajectory. Since the last few decades, we have witnessed several diverse methods proposed for path planning where each has its own advantages and disadvantages. The path planner is considered reliable in regards to planning a secure path when it is demonstrates that it maintains generous distances between the robot and every obstacle in the workspace. The objective for safety is often at odds with the objective for constructing an optimal path in terms of planning the shortest possible collision less trajectory from start point to the goal configuration.

Earliest reported work on robotic path planners has been the *Potential Field* planning method proposed in [2], and [13]. The *Potential Field* path planner employs the concept of virtual electromagnetic fields in the workspace modeled by a simulated attraction force towards desired points (i.e., goal/destination points) as well as repulsive forces from undesirable points (i.e., points occupied by obstacles). Each of these forces are simulated by a vector that captures the direction and magnitude of the force. Whereas the goal point vectors continuously exert pulling force for the robot, vectors corresponding to obstacles exert pushing away forces. At any point during path planning, the trajectory is adjusted to coincide with the result of reconciling the cumulative sum of applicable vector forces and directions. An impulse movement along the suggested path moves the robot to the next consecutive point along the trajectory. The magnitude of the impulse step is a parametric value corresponding to the path granularity. An overall trajectory for a pair of start and finish points is the accumulation of consecutive impulse moves. The process of path planning through *Potential Field* method guarantees a collision less trajectory from initial to goal configurations. However, due to the electromagnetic fields' constraints, this method performs poorly in certain scenarios. For example local minima is seen that causes when the robot becomes trapped in U-Shape obstacles (i.e., box canyons). There are other planner issues such as obstacles that leave narrow passage ways creating erratic trashing forces that can be either redundantly cyclic or contradictory forces. Latter problems often lead to impasse phenomenon: [5], [11], [14]. Many papers have proposed different solutions by updating the original algorithm or combining different method with the *Potential Field* planner construction to remedy specific problems: [3], [9], [7], [10], [19]. The *Rapidly-exploring Random Trees* is a sampling based mapping technique that solves non-holonomic constraints and it was introduced in [15]. *Information-rich Rapidly-exploring Random Trees* proposed in [16] is the extension of the *Rapidly-exploring Random Trees*, which is able to maneuver more efficiently to build the trajectory in workspaces with the presence of different constraint domains such as complex moving agent dynamics and moving robots sensor limitations in terms of resolution and narrow the detection view site. Several researchers have developed hybrid solutions, and hence, several studies and approaches related to the hybrid path planners are reported in [1], [4], [6], [12], [17], [18], and [20]. A hybrid path planner typically takes advantages from multiple path planning strategies, which are combined into a unique algorithm. Hybrid planners are promising to address more efficient path planning in order to elevate the quality and accuracy of the generated trajectories and also overcome various constraints and situations that can affect the traditional path planners in analyzing and determining optimal possible trajectories. In the next section, we illustrate the general concepts and methods that our planner employs to produce an optimal trajectory. In order to evaluate the performance of our planner, in subsequent section three, we will compare it with two other path planners by applying them on two sample workspaces. This paper further explores efficiency issues for our *Rapidly Optimizing Mapper* (ROM).

II. Foundations and Functions of Our ROM Framework

We have fabricated our planner on the premise of a multi-layer approach in the form of a unique algorithm such that each layer uses data provided from the prior layer and is responsible to generate the needed information for the following level. Each layer is also treated as a phase of a sequentially phased system that provides data for the next phase using information processes in the previous phase. Our ROM planner algorithm is constructed based on five general levels along with initial and final phases indicating as follows: *initial phase*, *workspace analyzer*, *graph builder*, and *shortest path calculation unit*. Each of the mentioned phase along with their objectives is detailed in the remainder of this section.

Initializing phase: This phase is achieved by adjusting values for variables that are salient constituents for building a trajectory. We considered the key feature of *path security*, which has a direct bearing on the robot maneuvering skills. Consideration for this element has to be determined at the initial phase as the primitive value which helps the planner to construct proper trajectories. The *standoff distance*, (SD) is our main path security parametric variable that is defined to be the width of a virtual buffer zone around perimeters of obstacles in order to specify a safety area for pathways within which enables the robot to navigate without involving collisions. The security channel width is determined based on robot sensors accuracy specifications. The more sensitive obstacle detection equipment the robot possesses, the lower security consideration required for the width of the safety channel. Each robot, based on the type of mission and the terrain specifications, is equipped with different capabilities such as actuators and sensors that equip it to move around and detect objects in the environment and thereby adjusts its path toward the determined trajectory instructed by the path planner.

Workspace analyzer: This phase of ROM is responsible for analysis of the workspace obstacles to determine roadblock obstacles as well as roadblock obstacles side edge nodes generation. The roadblock obstacles will be recognized by considering virtual straight lines from valid accessible nodes (i.e., safe obstacle boundary points) toward goal configuration. The valid nodes are in the form of a group of certain nodes starting from the initial configuration and ending with the goal location along with the group of roadblock side edge nodes. As it is illustrated in figure 1, the workspace analyzer phase obtains the first group of roadblock side edge nodes by considering straight rays from candid nodes toward goal configuration. Any obstacle that shares intersecting points with the start-goal line in at least one hit point is classified as a *roadblock obstacle*. The number of detected roadblock obstacles will vary based on the number of obstacles intersecting with the start-goal straight line. The best scenario occurs when there are no roadblock obstacles in the workspace. In such a situation, the optimal trajectory will be considered to be the straight line from start point to the goal configuration.

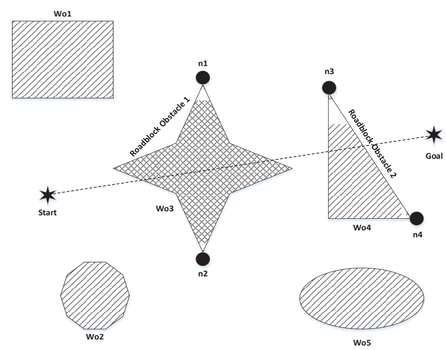

Figure 1. A sample roadblock obstacles side edge nodes generated from the workspace analyzer unit

Graph builder: The main objective of this phase of ROM is to form a complete graph consisting of roadblock side edge nodes combined with the start and goal configuration points. In order to achieve this goal, ROM enforces multiple steps. As the first step, the planner connects roadblock side edge nodes together to form a primitive graph of trajectories so as to enable maneuvering around the perimeter of the obstacle. The next step is to examine roadblock side edge nodes in order to recognize *uncompleted nodes* and to adjust them. Uncompleted nodes will be considered based on roadblock side edge nodes belonging to a single obstacle that are not yet connected to one another that do not go across the surface of the obstacle and remain entirely at one possible contiguous side of the obstacle. In other words, the path planner at this processing stage examines side edge nodes of each single roadblock obstacle to assure that they trail each other contiguously and steer clear of the surface of the obstacle. We use this technique to enable our planner to consider all possible paths crossing from roadblock side edge nodes with the purpose of elevating the planner ability to consider all possible paths toward goal and increasing the accuracy in determination of the shortest possible trajectory toward goal configuration. To obtain the best results in terms of refining the shortest possible trajectories, the path planner, as the next step, simplifies the paths via removing nodes that are located in between pairs of *visible nodes*. Visible nodes will be recognized if there is a possibility to connect two nodes through a straight path without intersecting any obstacles in the workspace. The last step of this phase of the planner consists of adjusting Euclidean distances for the remaining pairs of nodes that are already processed. The final result of this phase of ROM is a completed graph including start and goal points. Depending on the specifications of elements in the workspace, such as the number, size, shape and locations of obstacles, the pattern of the lattice and hence, the graph that forms through the graph builder phase will vary. Different scenarios result in having different numbers of paths from start to goal configurations.

Shortest path calculation unit: This phase of our planner adopts the *Dijkstra* algorithm, [8], in order to refine the shortest trajectory from start point toward goal configuration. Our planner uses the graph, which is generated at the previous phase of the planner as input data. The mentioned graph consists of all possible pathways that are optimized through the path planner optimization steps and eventually constructed as a form of graph. The predominant task of this phase is to examine all available paths in the graph and subsequently produce the best possible trajectory consisting of roadblock side edge nodes as the optimal path. The output of this phase is the final result of the planner in generating the trajectory, which is the optimal single collision less path from start point to the goal configuration.

III. EXPERIMENTS AND EVALUATION OF ROM

In order to assess our planner performance, we compare it with the two other path planners including *Potential Field* and *Rapidly-exploring Random Trees* path planners. The process of evaluation is conducted by applying our planner as well as the mentioned path planners on two exemplar workspaces that are both illustrated in figure 2.

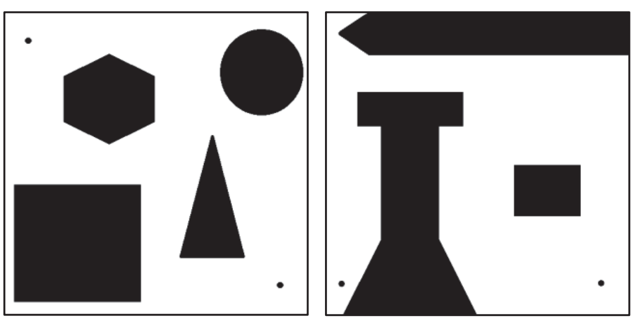

Figure 2. Left: The first candidate environment for applying path planners. Right: The second candidate workspace

The first map illustrated in figure 2 (left), consists of four obstacles with different polygonal geometric shapes and locations whereas figure 2 (right) consists of three obstacles. As seen in the figure 2, we considered a narrow distance between obstacles with the purpose of evaluating the skills of path planner algorithms to analyze and determine the optimal trajectories in terms of distance from start to goal configurations and the security of the constructed path. Both workspaces are considered to have the same dimensions of 500 by 500. The vertical axis spans from up to down and the horizontal axis emanates from left to right. The initial point for the first workspace is considered at the point (50, 50) and the goal location is at (450, 450). The start location for the second environment is at (20, 450). The goal for the second workspace is also located at (450, 450). The following figures 3 illustrate the resultant trajectory from applying out path planner on the sample workspaces.

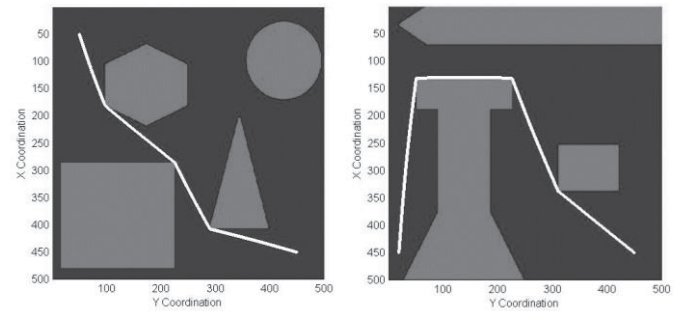

Figure 3. Left: The resultant trajectory from applying ROM on the first workspace. Right: The resultant path from applying ROM on the second environment

The optimal path, which is constructed using our planner algorithm is shown as a bright path trajectory starting from initial point to the goal configuration. The Euclidean distance from start location to the goal configuration is calculated to be 709.12 units for the first environment and 1018.43 units long for the second workspace. The trajectory resultant of applying our novel path planner shown in both workspaces in figure 3 indicates that our planner is able to route a collision less path, successfully. Moreover, neither obstacle complexity in terms of shapes nor the distances between obstacles could permit the planner to route the best possible trajectory in terms of the safety and the length toward goal. This is because our planner considers all possible valid directions around each roadblock obstacle to achieve the best results in determining the optimal trajectory. In addition, our planner benefits from using nontrivial strategies to reconstruct the generated graph in the early stages of its algorithm with the purpose of recognizing the best candidates among all possibilities of different routes and generating worthwhile trajectories, regardless of constraints posed by different scenarios in workspaces.

In order to compare the performance of ROM path planner skills with other planners, we employed two path planners in offline mode, specifically *Potential Field* path planner and *Rapidly-exploring Random Trees* algorithm. Each planner is applied on the sample workspaces in the form of a different case study and the results of each scenario are discussed in detail. Through case study I, the process of planning trajectories using *Potential Filed* algorithm will be analyzed.

Case study 1:

The *Potential Field* algorithm performs the trajectory based on considering start and goal points as well as obstacles as electromagnetic charges and fields. The goal point has the most attractive power (i.e., exerting attraction force) among other objects in the workspace, whereas obstacles repel (i.e., exert pushing way force) the planner path finder away from them. The *Potential Field* planner method benefits this strategy to build a collision less trajectory. The following figure 4 shows the result of the path generated by applying the *Potential Field* algorithm on the sample workspaces.

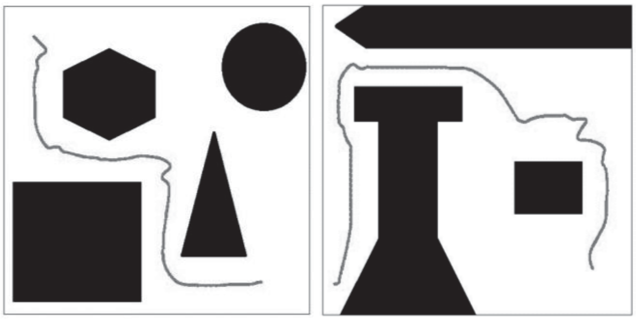

Figure 4. Left: The resultant trajectory from applying the *Potential Field* path planner on the first workspace. Right: The resultant path from applying the *Potential Field* on the second environment

The constructed trajectory resultant from applying the *Potential Field* method on sample workspaces are considered in a dark path starting from start into goal configuration. The trajectory length from start point to the goal configuration for the first map is equal 984.16 units based on Euclidean distance measurement. The *Potential Field* planner is also computed the path length for the second environment to be 1984.31 long in Euclidean distance measurement. As it is evident in both maps in figure 4, the final generated trajectories consist of several curvatures. This event can be explained according to the similarity of simulated charges between the trajectory and obstacles. In addition, the rate of severe curves increases when the path is crossing from sharp obstacle edges nearby. This is because of forming electromagnetic fields with different intensities around sharp edges of obstacles that cause the *Potential Field* planner to continuously adjust the path based on different amounts of repulsion forces around the mentioned areas. Also, in the second workspace illustrated in the figure 4, (right), the planner was not able to consider the trajectory with shorter route. This is because there exists many adjacent sharp edges between two T-Shape and square obstacle that push the planner to stay out of the pathway crossing from the shorter side toward goal configuration. Comparing constructed trajectories in both workspaces through the *Potential Field* planner reveals that the planner is able to route a collision less trajectory in both environments. It is, however, suffering from the side effects of electromagnetic fields forming around nearby obstacles, especially around obstacles sharp edges surroundings to consider the optimal paths in terms of length and hence reduces the performance of the *Potential Field* planner.

The next case study is dedicated to examining the *Rapidly-exploring Random Trees* path planner on the sample workspaces to evaluate its performances on determining the optimal trajectories.

Case study 2:
The *Rapidly-exploring Random Trees* method works based on forming random trees consisting of a group of arbitrary sample points located outside of the obstacles in the workspace. The planner algorithm will then examines all possibilities of branches that form randomly around the main stem of the generated trees at each moment during the path generation process and gradually selects the best collision less matches in terms of the length and security, as sub-optimal trajectories among them. The planner constructs the final path from considering all optimal sub-trajectories obtained in the previous level. The following figure 5 demonstrates the resultant trajectory determined from the application of the *Rapidly-exploring Random Trees* on two sample workspaces.

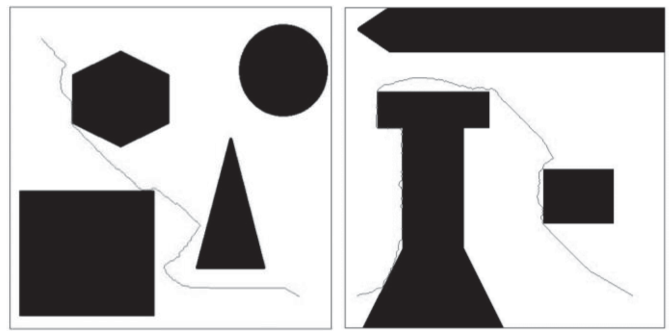

Figure 5. Left: The resultant trajectory from applying the *Rapidly-exploring Random Trees* planner on the first workspace. Right: The resultant path from applying the *Rapidly-exploring Random Trees* algorithm on the second environment

The Euclidean distance from the start configuration for the trajectory length resultant from applying the *Rapidly-exploring Random Trees* on the first workspace is calculated to be 865.52 units for the first scenario. The *Rapidly-exploring Random Trees* is also determined to be 1252.89 units long Euclidean distance from start point for the second workspace. Because the nature of the RRT planner algorithm in forming random nodes to explore the workspace surroundings and to refine the best matches, we observe that the planner constructs trajectories that are slightly different from each other at every run. Benefiting the techniques of using random nodes leads the planner to achieve collision less trajectories that are close to the optimal trajectories in most cases. It is, however, still is not able to reach the shortest possible collision less trajectories due to lack of existing proper methods to shortened the final generated path in the planner construction. As it can be examined in figure 4, (right), the issue that is addressed above exhibits a larger effect in accuracy reduction to determining the shortest possible path in the area between the T-Shaped and the square obstacles.

In order to clarify the best results in trajectory lengths, we collected all results that our ROM planner as well as other path planner candidates achieved in a single trajectory length chart as shown in the following figure 6:

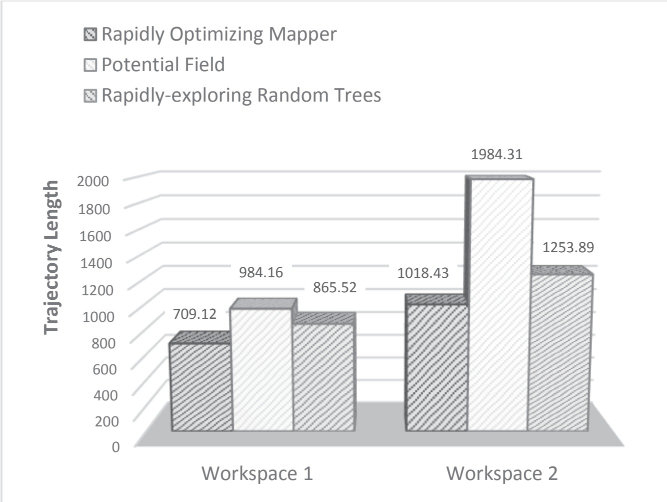

Figure 6. The trajectory distance chart of computer path for the *Rapidly Optimizing Mapper* as well as *Potential Field* and *Rapidly-exploring Random Trees* path planners

Comparing trajectory distances illustrated in the figure 6 reveals that our planner is able to route the most efficient trajectories in terms of shortest paths among other candidates. In other words, *Rapidly Optimizing Mapper* exhibits a higher performance for building collision less trajectories in terms of distances from start points to the goal configurations. Employing the *Potential Field* algorithm to route trajectory in workspaces with close obstacles and sharp edges reduces the performance of the planner dramatically as it illustrated in the second workspace trajectory rate in figure 6. In order to overcome the mentioned environmental constraints, our planner benefits from specific techniques to optimize the calculated paths during the process of planning helps our planner to take all possible paths into account for the planning operation.

IV. CONCLUSION AND PERSPECTIVES

A novel path planner termed ROM has been elaborated within this research article. We demonstrated our novel path planner specifications and abilities by illustrating its constituent components. In order to validate the performance of our planner, we considered sample workspaces with complex constraints that contain a variety of different shapes and locations to apply and evaluate the planner performances for building trajectory skills on different scenarios. In order to clarify our novel path planner's strength for determining the ultimate trajectories, we compared it with two of the best known path planners through applying them on the same sample workspaces. Based on the obtained results, we conclude that ROM is able to compute the optimal trajectories in terms of path length more efficiently. This is because we adopted techniques to furnish our planner with the intention of elevating its abilities to operate overcoming a variety of different constraints on workspaces elements specifications. Our future target is to examine our novel path planner on workspaces with more constraints along with upgrading its structure to heighten its efficiency to act limitlessly in any types of environment, flawlessly.

REFERENCES

[1] Abinaya, S., Hemanth Kumar, V., Srinivasa Karthic, P., Tamilselvi, D., and Mercy Shalinie, S., "Hybrid Genetic Algorithm Approach for Mobile Robot Path Planning" in Journal of Advances in Natural and Applied Sciences, Vol.8, No.17, Special 2014, pp.41-47, 2014.

[2] Andrews, J.R. and Hogan, N., "Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator." Control of Manufacturing Processes and Robotic Systems, Eds. Hardt, D. E. and Book, W., ASME, Boston, pp.243-251, 1983.

[3] Arámbula Cosío F. and Padilla Castañeda M.A., "Autonomous Robot Navigation using Adaptive Potential Fields" in Mathematical and Computer Modelling, Vol.40, Issue 9-10, pp.1141-1156, 2004.

[4] Bashra K., and Oleiwi and Hubert Roth, "Modified Genetic Algorithm Based on A* Algorithm of Multi Objective Optimization for Path Planning" in Journal of Automation and Control Engineering, Vol.2, No.4, pp.357-362, 2014.

[5] Borenstein J. and Koren Y., "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots" in IEEE Transactions on Robotics and Automation 7, pp.278-288, 1991.

[6] Chaari, I., Koubaa, A., Trigui, S., Bennaceur, H., Ammar, A., and Al-Shalfan, K., "SmartPATH: An Efficient Hybrid ACO-GA Algorithm for Solving the Global Path Planning Problem of Mobile Robots" in International Jounal of Advanced Robotics Systems, pp.1-15, 2014.

[7] Connolly C.I., Burns J.B., and Weiss R., "Path Planning using Laplace's Equation" in Proceedings of the lEEE Conference On Robotics and Automation, Cincinnati, OH, pp.2102-2106, 1990.

[8] Dijkstra, E.W., "A Note on Two Problems in Connexion with Graphs" in Numerische Mathematik 1, pp.269–271, 1959.

[9] Ge S.S. and Cui Y.J., "New Potential Functions for Mobile Robot Path Planning" in IEEE Transactions on Robotics and Automation, Vol.16, No.5, pp.615-620, 2000.

[10] Guldner J., Utkin V., and Hashimoto H., "Robot Obstacle Avoidance in N-Dimensional Space using Planar Harmonic Artificial Potential Fields" in Journal of Dynamic Systems, Measurement, and Control, Vol.119, pp.160-166, 1997.

[11] Grefenstette J. and Schultz A.C., "An Evolutionary Approach to Learning in Robots" in Proceedings of the Machine Learning Workshop on Robot Learning, International Conference on Robot Learning, New Brunswick, N.J., pp.65-72, 1994.

[12] Ju, M., Wang, S., and Guo, J., "Path Planning using a Hybrid Evolutionary Algorithm Based on Tree Structure Encoding" in The Scientific World Journal, Vol.2014, Article ID 746260, 8 pages, 2014.

[13] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots" in IEEE International Conference on Robotics and Automation, St. Louis, Missouri, pp.500-505, 1985.

[14] Koren Y., and Borenstein J., "Potential Field Methods and Their Inherent Limitations for mobile robot navigation" in Proceedings of the IEEE International Conference on Robotics and Automation, pp.1398-1404, 1991.

[15] LaValle, S.M., "Rapidly-Exploring Random Trees: A New Tool for Path Planning", Computer Science Department, Iowa State University, TR 98-11, 1998.

[16] Levine, D., Luders, and B. How, J.P., ""Information-rich Path Planning with General Constraints using Rapidly-exploring Random Trees." in American Institute of Aeronautics and Astronautics (AIAA), Atlanta, Georgia, AIAA 2010-3360, 2010.

[17] Loo, C.K., Rajeswari, M., Wong, E.K., and Rao, M.V.C., "Mobile Robot Path Planning using Hybrid Genetic Algorithm and Traversability Vectors Method" in Intelligent Automation and Soft Computing, Vol.10, No.1, pp.51-64, 2004.

[18] McFetridge, L. and Ibrahim, M.Y., "New Technique of Mobile Robot Navigation using a Hybrid Adaptive Fuzzy-Potential Field Approach" in Computers and Industrial Engineering, Vol.35, Issues (3–4), pp.471–474, 1998.

[19] Utkin V.I., Drakunov S., Hashimoto H., and Harashima F., "Robot Path Obstacle Avoidance Control via Sliding Mode Approach" in Proceedings of the lEEE/RSJ International Workshop on Intelligent Robots and Systems, Osaka, Japan, pp.1287-1290, 1991.

[20] Yao., Z. and Ren, Z., "Path Planning for Coalmine Rescue Robot based on Hybrid Adaptive Artificial Fish Swarm Algorithm" in International Journal of Control and Automation, Vol.7, No.8, pp.1-12, 2014.