Journal of Theoretical and Applied Information Technology
10th December 2014. Vol.70 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A CRITICAL EVALUATION OF LITERATURE ON ROBOT PATH PLANNING IN DYNAMIC ENVIRONMENT

[1]MOHAMMAD NAIM RASTGOO, [2]BAHAREH NAKISA, [3]MOHAMMAD FAIDZUL NASRUDIN, [4]MOHD ZAKREE AHMAD NAZRI

Department of Computer science, Faculty of Information Science and Technology, University Kebangsaan Malaysia, Bangi, Malaysia.

Email: [1]Naim.rastgoo@gmail.com, [2]Bahareh.nakisa@gmail.com, [3]mfn@ukm.edu.my [4]Zakree@ukm.edu.my

ABSTRACT

Robot Path Planning (RPP) in dynamic environments is a search problem based on the examination of collision-free paths in the presence of dynamic and static obstacles. Many techniques have been developed to solve this problem. Trapping in a local minima and maintaining a Real-Time performance are known as the two most important challenges that these techniques face to solve such problem. This study presents a comprehensive survey of the various techniques that have been proposed in this domain. As part of this survey, we include a classification of the approaches and identify their methods.

Keywords: *Path Planning, Dynamic Environment, Real-Time Performance, Local Minima.*

1. INTRODUCTION

There are several research areas in robotic such as cognitive robotics, multi-robot systems, robot path planning. One of the major areas in robotics is Robot Path Planning (RPP) that is studied by many researchers until now because this problem has been applied in several robotic applications such as autonomy [1], robotic surgery [2] and automation [3]. RPP plans the behaviors of a robot to be able to do a task. Therefore, if current status of the robot is considered as the start state and the status of the robot after finishing the task as the goal state, RPP must plan on finding a possible path and through this path; the robot will be able to achieve the goal state.

RPP has been studying in different environments such as static environments [4],[5],[6],[7],[8],static environment contain movable obstacles [9],[10],[11],[12],[13],[14] dynamic environments contain static and dynamic obstacles[15],[16],[17]. In these environments the path of the dynamic obstacles is unknown and is known as the hardest search problem in compare with the other environments. This study reviews various RPP approaches that have been applied on dynamic environments.
In dynamic environments autonomous robots must move reliably among dynamic and static obstacles. Two challengeable problems for RPP have been proposed in dynamic environments. These are trapping in local minima [18] and maintaining real-time performance [19].

Local Minima problem refers to Some regions in the state-space have inaccurate heuristic values and they look promising and algorithms attract to continue its search toward the goal from these regions, therefore trapping inside these regions increase the search time [Ishida 2003]. In addition, the most of exciting methods cannot guarantee Real-Time performance [Cannon 2012], which enhance the average solution costs. This means the robot cannot correctly track the environment changes; therefore the cost of finding near optimal path increases.

[20] is a general survey about robot motion planning in dynamic environment that reviews several approaches which are applied to solve the problem. [21] is a review paper in robot motion planning and compare several applied approaches in solve trajectory planning with and without deferential constraints.[22], [23] are another research reviews that classify the approaches in the domain of robot motion planning and coverage path planning. None of these approaches don't refer to above challenges specifically and doesn't review the related approach.

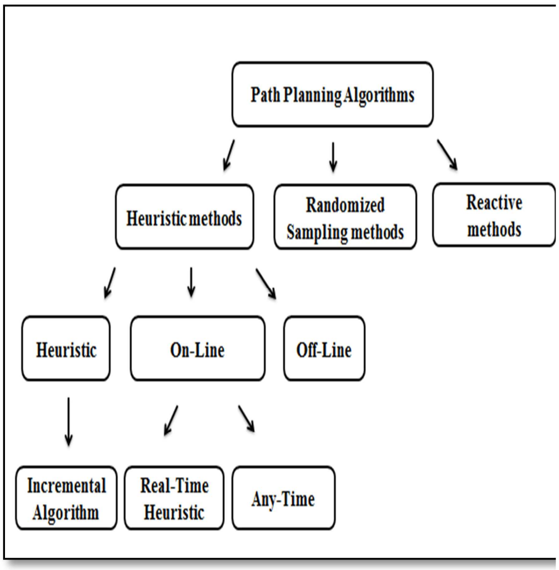
Journal of Theoretical and Applied Information Technology
10th December 2014. Vol.70 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In this survey the performance of various algorithms that have been proposed in this domain in face with these challenges are compare with each other.

2. APPIED ALGORITHMS ON PATH PLANNING PROBLEM

Several approaches have been used to solve path-planning problem in dynamic environment, which is shown in Figure 3.1. Based on this figure, the techniques and algorithms classify into 3 categories; namely, Heuristic methods, Randomized sampling methods and reactive methods.



2.1 Heuristic Methods

The heuristic methods are considered by many researchers and heavily by many researches. A simple idea for finding the path is to apply a basic search algorithm with uniform cost [18][24]. Although this search method is complete and optimal, it expands a lot of nodes, which are not really near to the solution path. A heuristic function is an alternative method that is able to estimate the cheapest cost path from any node in the search space to the goal. Heuristic function is designed based on the problem constraints. In addition, Heuristic function guides the search toward the goal without expanding many of the nodes visited by uniform cost search. Therefore Heuristic function is able to decrease the search time and use as a Real-Time search algorithm. This category is divided into three classes; namely, heuristic online and off-line methods. Online methods divide to real-time heuristic search and any time algorithms classes. In following we explain these classes and review some of their famous algorithms and also report their advantage and disadvantage of these algorithms to solve RPP in dynamic environments.

2.1.1 Heuristic

Simple straight-line distance heuristic and the static 2D Dijkstra heuristic [19][25] are the two Heuristic functions that are used for robot path planning. Although, these methods estimate the distance of any state to the goal, which are admissible, they suffer from the local minima that appeared due to the static obstacles. A* algorithm is proposed by [20], which is the most well-known heuristic search algorithm. There are two lists in this algorithm, called open-list and close-list. The open-list stores and orders all states searched and generated by A*. Open-list is implemented by a priority queue (mean heap) list. The heuristic function that is used for A* is f(n), this function for a node n is as follow:

$$f(n) = g(n) + h(n)$$

Where g(n) (cost-thus-far) is calculated cost to move from the start node to node n and h(n) (cost-to-go) is estimated cost to reach the goal from node n. By adding each state to open list, this list reorders its records by f-values incrementally. Thus, the top of the list refers to lowest f-value. It should be mentioned that the records with equal f-values are sorted by higher g_s. After finishes the search A* pops up the state, which is in the top of the open-list and then, stores it in the close-list. Therefore the close-list contains selected states by A-star. This algorithm is not able to maintain the real time performance of robot and easily trap in local minima. The Weighted A* is a version of A* [21]. This algorithm uses a weighted heuristic function:

$$f(n) = g(n) + w.h(n)$$

Although the higher value of w makes the search greedier that reduce the search time, this method has the same problems like as A* for dynamic environments.

A. Incremental heuristic search

The incremental heuristic search algorithm finds the presumed unblocked path from its current cell to the goal cell. This search is done iteratively until the robot reaches the goal or faces with the block cell [22]. The incremental algorithms that are used in path planning domain classifies into 3 groups [23].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The first group contains algorithms, which are incremental versions of A-star. FSA* [24] is one of the important algorithms in this class. The mechanism of this method is to find a shortest path from start node to goal with using A-star. The algorithm reuses preceded A* search graph that is identical to the current A* graph search to find another short path between start and goal state if the some staged is marked as static obstacles.

The second group learns the h-value of nodes from the previous search and makes the estimation better. By upgrading the h-value, the search algorithm is able to escape from local minima. Adaptive A* is an incremental algorithm [25], which is placed in the second group. The third group modifies g-values based on the previous search. This group is more efficient group in compare with other groups [23], Differential A* [26], Focused Dynamic (D-star) [27] and D* Lite [28]. In this group D* and D* Lite are more sophisticated than Differential A-star. In Mars rovers and tactical mobile robot projects, D* is used for the mobile robots path planning [29], [30]. These algorithms have poor performance in face of local minima and they cannot guarantee real-time performance in dynamic environments.

2.1.2 On-Line Methods

Real-Time search algorithms have a constant communication with their environments and they are able to track the environment's changes, therefore Real-time search algorithms are suitable methods to solve search problems when the robot initially has incomplete information about its environment. Several real-time algorithms are presented for solving path-planning problem ([31], [32], [22],[33],[34],[35],[36], [37]).

There is two phases in real-time search algorithms [38]: i) the planning phase, and ii) the action phase. In the planning phase, the robot selects one or several possible actions (selection step) and updates the heuristic values of selected states (learning step), and then it executes them in the action phase. This process repeats continuously until it reaches the goal. In the action phase, the robot is able to observe the environment and update its information about such environment.

A. Real time heuristic search algorithms

The first real-time algorithm is learning real-time A*(LRTA*)[19]. The several modified versions of LRTA* have been introduced. In the phase of planning, LRTA* initially generates the successors of the current robot position and by using A* algorithm [20] selects the best successors among all successors. In the learning step it updates the heuristic values of the selected successors. This method can find the near optimal path faster than off-line methods. The mechanism of LRTA* is very poor when faced with dynamic environments and it has a high-cost solution. Although the structure of LRTA*(k) [40] is like as LRTA*, this algorithm uses a different approach than LRTA* to update heuristic values. By performing more learning, LRTA*(k) is able to escape from local minima. This algorithm has very poor performance in face with dynamic environments.

$LRTA_{LS}(k, d)$ is a modified version of LRTA*(k), which is proposed by [35]. The learning mechanism of this algorithm is improved by increasing Lookahead and adding more learning in each planning phase. Based on the performance of this method it can plan high-speed motion but this algorithm cannot guarantee real-time performance.

Local Search Space Learning Real-Time A* (LSS-LRTA*)[23] is known as a state-of-the-art algorithm. In selection part, the robot selects a limit number of states (Lookahead) and this set of nodes is called the local search space. After selecting the states and storing them in the close list LSS-LRTA* selects a local goal from states that are stored in the open list. The state with the lowest f-value is algorithm's candidate for local goal. In the learning part, algorithm uses Dijkstra algorithm to modify heuristic values of close list. By using this learning mechanism, LSS-LRTA* is able to escape from local minima. The drawback of this algorithm is that it cannot guarantee real-time performance. RTD* [41] is a state-of-the-art algorithm for dynamic environments. This method combines D* Lite [28] and LSS-LRTA* [23]. The planning phase in this method is divided into two parts: 1) D* Lite. 2) LSS-LRTA*. In the first part, RTD* come back from the goal state to the current state of robot and if it can find a complete path then returns the optimal path; Otherwise LSS-LRTA* runs to find a suitable action in the time remaining from planning time. The drawback of this algorithm returns to the D* lite because this algorithm cannot plan high-speed actions and it easily fails in the high dimensional state space.

PLRTA*[42] is a new version of LSS-LRTA*. This method by separating the costs to static and dynamic tries to learn them separately. This algorithm has better performance in comparison with

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

other real-time heuristic search algorithms in face with dynamic environments. In addition this algorithm guarantees real-time performance.

As it stated before trapping in local minima are one of the important challenge in this domain. This problem happens when the robots trap in Heuristic depression regions. Heuristic depression regions are called to the bounded areas of search space which heuristic function is inaccurate and real-time heuristic functions easily become trapped in those regions since the heuristic values of their states may need to be updated multiple times, which results in costly solutions[43].

The early definition of a heuristic depression region was presented by [18], is a bounded region for which the heuristic value of states on the boundary of region is greater than or equal to the heuristic value of states inside of the region. This definition modified by [43] present two new techniques for real-time search algorithms. These techniques are Mark-and Avoid and Move-to-Border, they applied these techniques on LSS-LRTA* and RTAA*and present 4 new modified versions, the best one is called daLSS-LRTS*. The mechanism of these new algorithms in face with heuristic depression regions is marking the states inside the heuristic depression regions and then avoiding them in the following iterations. Based on their results Move-to-Border is a better technique and the reason is that this technique guides the search to move to the border or to states that are closer to the border. Although these new versions avoid from local minima, which improves the performances of the algorithm in compare with the other real-time search algorithms, these new versions like the original version cannot guarantee real-time performance.

B. Any-Time algorithms

Anytime algorithms try to make their solutions better as the time progress. This kind of algorithms can even return a partial solution before any interruption at any time. The quality of the answers that are generated by these algorithms has a direct way with the computation time. It means the generated solution by Anytime algorithm is an approximation of an optimal or best solution and the quality of the solution increase by adding a computation time. [45] introduced Anytime algorithm. This method is useful for solving time-dependent problem. [19] proposed Anytime algorithm. They reach a great success by utilizing an Anytime Repairing A* (ARA*) that is introduced by [46]. One of the applications of the Anytime algorithm is robot motion planning and ARA* is one of the important path planning algorithms.

2.1.3 Off-line algorithms

The off-line algorithms generate all paths from the start position to the goal position at each time step ([38],[47]). In other words, unlike the other methods that generate an action at each time step, this kind of algorithm has the ability to generate a complete path from the start to the goal at each time step.

[15] proposed a new method for dynamic environment. It means the planning phase in this method has to predict the location of the moving objects in a noisy environment. Also, the computation of the planning is expensive due to the time that is added as a variable of state-space. Therefore, for predicting the trajectory of the moving object, the re-planning is a need. In this method, the time planning is divided into two time steps. In the first time step each state is represented by 5 tuples (x, y, θ, v, t). In this step the algorithm predicts the dynamic obstacle's movement until T_b^{max}, which the prediction is reliable. The author found that if they want to achieve an acceptable behavior without collision, they have to limit the T_b^{max}. This limitation for time planning allows it to re-plan quickly, which results in fast reaction to the recent changes in the environment then it goes to the second time step. In this method, the algorithm does the planning until reaching the goal. For finding the optimal path, the Weighted A* search is used. Furthermore, when the world space becomes larger and the start position and the goal position are placed further from each other, the amount of time will increase. If the number of dynamic obstacles in the environment grows, the time of planning rises .The important thing about this method is that when the time bound is finished and then the search switches to 2D Dijkstra search, this technique will be entrapped into local minima.

[48] introduced a new version of Accelerated A*(AA*) which is called IAA*. This algorithm that is based on the adaptive sampling, removes the balance between the speed and precision. IAA* with a fast planning method is a suitable technique for large environments. There is an adaptive parameterization, which is a set of acceptable motion actions of the vehicle. At each generation, this adaptive parameterization is considered during the expansion of the child states. It means when the robot is far from the obstacle, the algorithm can

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

make a bigger step and make a smaller step when the robot is near the obstacle. In this technique multiple search iterations run until the path is found. Thus, it is not clear how the algorithm allocated the time bound to the planning phase and also if a solution is not found before the time bound the algorithm would not know what should be done for the next action.

2.2 Randomized Sampling Methods

The randomized approach has used to decrease the significant number of explored states during the search. This approach is able to handle easily the situations that search become impossible due to increasing the state space and too many explored states. Rapidly-Exploring Random Tree (RRT) [49] is a randomized sampling algorithm. The mechanism of this algorithm for search builds and extends outwards randomly a tree from the initial state.

Although extension of the tree is randomly, unexplored regions have outstanding weight than other regions and the search algorithm has biased toward them. RRT is one of the most famous randomized sampling algorithms in robotic community and this algorithm is used in various platforms of robotic [50]. RRT cannot guarantee real time performance and it has demonstrated the best solution that is relearned by this algorithm almost always is a non-optimal solution [51]. In addition, RRT does not consider cost in expanding the tree; therefore the founded path may be placed in high cost areas. The real time version of RRT is ERRT [52]. At each iteration of search, ERRT records some information and uses the recorded information from pervious iterations to find a better path. The Metric Adaptive RRT (MA-RRT) algorithm [53] is a modified version of RRT. This algorithm addresses to the RRT problems.

MA-RRT like as ERRT uses information of pervious iteration and by using previous information; MA-RRT is able to guide the search (tree) toward the areas that contain better and near optimal paths. RRT* [54] has different approach to extend the tree. According to RRT*, the tree just grows randomly through one vertex. Also RRT*, first grows the tree with one vertex and evaluate this extension and if the extension from that vertex is counted as a successful extension then RRT* tends simultaneously to examine and grows the tree from all other vertexes where their distances to the proposed vertex don't exceed from specific threshold. Although this algorithm finds a better solution than RRT especially if search time increases, it has high computational cost.

2.3 Reactive Methods

In general, the reactive methods do not do any search to find their paths. Instead of searching they use differentiable function. The mechanism of this method based on this assumption that the surface of state space can be map to a differentiable function whose gradient slopes away from obstacles and towards the goal. The robot's actions should be converged to the gradient. Potential field [55] is one of the algorithms in this category. The main drawback of this category is trapping in local minima.

3. CURRENT ISSUES AND CHALLENGES IN THE FIELD OF ROBOT PATH PLANNING

In this paper different approaches in the field of robot path planning are compared. Table shows the performance of the most applicable reviewed approach in face with two challenges. These two challenges are trapping into the local minima and maintaining Real-Time performance. As discussed before, trapping in local minima cause increasing search time and if the mechanism of approach doesn't maintain real-time performance then the average solution cost increase.

It is shown that some algorithms like Incremental Heuristic search and Real-Time heuristic search algorithms in this field are more useful and therefore has better opportunities to do some research about them. In addition, these algorithms have better performance in compare with other algorithms in face with mentioned challenges. On the other hands, it seems that each approach, especially the reactive methods ones, suffer from many drawbacks.
After analyzing the existing researches in the field of Path Planning, the following new intellectual challenges, new applications, and emerging issues for Path Planning are proposed:

- One of the open issues is what are the possible mechanisms to detect and avoid the local minima in complex environments?
- Another open issue in heuristic search algorithms is researching about partitioning heuristic costs (static and dynamic) in the environment?
- When the environment is complex how can the robot tracks the changes in the environment and maintain real-time performance?

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.CONCLUSION

In this overview, we have studied researched conducted on robot path planning in dynamic environment contains static and dynamic obstacles between 1968 and 2012. Some of the main contributions in this area such as trapping in local minima and maintaining real-time performance have been selected and studied. All the studied papers in this area have been classified into three categories; namely, Heuristic, Randomized Sampling and Reactive methods. The algorithms belong to each category are reviewed. Among these categories heuristic methods contains several algorithms, which has good mechanism in face with the mentioned challenges in dynamic environments. The most applicable of these algorithms belong to the Heuristic Real-Time Search group. These algorithms uses escape and avoid mechanism when they trap in local minima and also apply learning mechanism to have a real-time performance.

REFERENCES:

[1]. Kuffner, J., et al. Motion planning for humanoid robots. Robotics Research, Springer: 365-374.2005.
[2]. Tombropoulos, R. Z., et al. "Carabeamer: A treatment planner for a robotic radiosurgical system with general kinematics." Medical Image Analysis 3(3): 237-264.1999.
[3]. Lengyel, J., et al. Real-time robot motion planning using rasterizing computer graphics hardware, ACM.1990.
[4]. Arkin, R. C. (1989). "Motor schema—based mobile robot navigation." The International Journal of Robotics Research 8(4): 92-112.
[5]. Latombe, J.-C. ROBOT MOTION PLANNING.: Edition en anglais, Springer.1990.
[6]. Glasius, R., et al. "Neural network dynamics for path planning and obstacle avoidance." Neural Networks 8(1): 125-133.1995.
[7]. Van Den Berg, J., et al. Anytime path planning and replanning in dynamic environments. Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, IEEE.2006.
[8]. Ismail, A.-T., et al. "A mobile robot path planning using genetic algorithm in static environment." Journal of Computer Science 4(4): 341.2008.
[9]. Wilfong, G. Motion planning in the presence of movable obstacles. Proceedings of the fourth annual symposium on Computational geometry, ACM.1988.
[10]. Chen, P. C. and Y. K. Hwang. Practical path planning among movable obstacles. Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on, IEEE.1991.
[11]. Stilman, M., et al. "Planning and executing navigation among movable obstacles." Advanced Robotics 21(14): 1617-1634.2007.
[12]. Stilman, M., et al. Manipulation planning among movable obstacles. Robotics and Automation, 2007 IEEE International Conference on, IEEE.2007.
[13]. Stilman, M. and J. Kuffner. "Planning among movable obstacles with artificial constraints." The International Journal of Robotics Research 27(11-12): 1295-1307.2008.
[14]. Van Den Berg, J., et al. Path planning among movable obstacles: a probabilistically complete approach. Algorithmic Foundation of Robotics VIII, Springer: 599-614.2009.
[15]. Kushleyev, A. and M. Likhachev. Time bounded lattice for efficient planning in dynamic environments. Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE.2009.
[16]. Snape, J., et al. Independent navigation of multiple robots and virtual agents. Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems.2010.
[17]. Phillips, M. and M. Likhachev. Sipp: Safe interval path planning for dynamic environments. Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE.2011.
[18]. Ishida, T. Moving target search with intelligence. In Proceedings of the 10th National Conference on Artificial Intelligence (AAAI), pp. 525-532.1992.
[19]. Korf, R. E. "Real-time heuristic search." Artificial intelligence 42(2): 189-211.1990.
[20]. Galceran Yebenes, E., & Carreras Pérez, M. A survey on coverage path planning for robotics. © Robotics and Autonomous Systems, 2013, vol. 61, núm. 12, p. 1258-1276.2013.
[21]. Keshmiri, S., & Payandeh, S. An overview of mobile robotic agents motion planning in dynamic environments. Paper presented at the Proceedings of the 14th IASTED International Conference, Robotics and Applications (RA20), MA, Boston.2009.
[22]. Tang, S., Khaksar, W., Ismail, N., & Ariffin, M. A review on robot motion planning

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

approaches. Pertanika Journal of Science and Technology, 20(1), 15-29.2012.
[23]. Goerzen, C., Kong, Z., & Mettler, B. A survey of motion planning algorithms from the perspective of autonomous uav guidance. Journal of Intelligent and Robotic Systems, 57(1-4), 65-100.2010.
[24]. Nils J. Nilsson. Problem-Solving Methods in Artificial Intelligence. McGraw-Hill Publishing Co., 1971.
[25]. Maxim Likhachev and David Ferguson. Planning long dynamically feasible maneuvers for autonomous vehicles. International Journal of Robotic Research, 28:933–945, 2009.
[26]. P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2): 100–107, July 1968.
[27]. Ira Pohl. Heuristic search viewed as path finding in a graph. Artificial Intelligence,1(3-4):193–204, 1970.
[28]. S. Koenig, M. Likhachev, Y. Liu, and D. Furcy. Incremental heuristic search in Artificial Intelligence. Artificial Intelligence Magazine, 25(2): 99–112, 2004.
[29]. Koenig, S. and X. Sun. "Comparing real-time and incremental heuristic search for real-time situated agents."Autonomous Agents and Multi-Agent Systems 18(3): 313-341.2009.
[30]. X. Sun and S. Koenig. The Fringe-Saving A* search algorithm - a feasibility study. In Proceedings of the International Joint Conference on Artificial Intelli-gence, pages 2391–2397, 2007.
[31]. S. Koenig and M. Likhachev. A new principle for incremental heuristic search: Theoretical results. In Proceedings of the International Conference on Autonomous Planning and Scheduling, pages 402–405, 2006.
[32]. K. Trovato. Differential A*: An adaptive search method illustrated with robot path planning for moving obstacles and goals, and an uncertain environment. Journal of Pattern Recognition and Artificial Intelligence, 4(2):245–268, 1990.
[33]. Anthony Stentz. The focussed d* algorithm for real-time replanning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pages 1652–1659, 1995.
[34]. S. Koenig and M. Likhachev. D* Lite. In Proceedings of the National Conference on Artificial Intelligence, pages 476–483, 2002.
[35]. M. Hebert, R. McLachlan, and P. Chang. Experiments with driving modes forurban robots. In Proceedings of the SPIE Mobile Robots, 1999.
[36]. S. Thayer, B. Digney, M. Diaz, A. Stentz, B. Nabbe, and M. Hebert. Distributed robotic mapping of extreme environments. In Proceedings of the SPIE: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII, volume 4195, pages 84–95, 2000.
[37]. Furcy, D. and S. Koenig. Speeding up the convergence of real-time search. AAAI/IAAI.2000.
[38]. Shimbo, M. and T. Ishida. Towards real-time search with inadmissible heuristics. ECAI, Citeseer.2000.
[39]. Bulitko, V. and G. Lee. "Learning in real-time search: a unifying framework." J. Artif. Intell. Res.(JAIR) 25: 119-157.2006.
[40]. Bulitko, V., et al. Dynamic Control in Path-Planning with Real-Time Heuristic Search. ICAPS.2007.
[41]. Hernández, C. and P. Meseguer. Improving real-time heuristic search on initially unknown maps. Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS), Workshop on Planning in Games, Citeseer.2007.
[42]. Rayner, D. C., et al. Real-Time Heuristic Search with a Priority Queue. IJCAI.2007.
[43]. Sturtevant, N. R., et al. On learning in agent-centered search. Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems.2010.
[44].Koenig, S. "Agent-centered search." AI Magazine 22(4): 109.2001.
[45]. Korf, R. E. "Real-time heuristic search." Artificial intelligence 42(2): 189-211.1990.
[46]. Hernández, C. and P. Meseguer. LRTA*(k). Proceedings of the 19th international joint conference on Artificial intelligence, Morgan Kaufmann Publishers Inc.2005
[47]. Bond, D., Widger, N.A., Ruml, W., & Sun, X. Real-time search in dynamic worlds. Paper presented at the Third Annual Symposium on Combinatorial Search.2010
[48]. Cannon, J., Rose, K., & Ruml, W. Real-time motion planning with dynamic obstacles. Paper presented at the SOCS.2012
[49]. Hernández, C. and J. A. Baier. "Avoiding and Escaping Depressions in Real-Time Heuristic Search." J. Artif.Intell. Res.(JAIR) 43: 523-570.2012

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[50]. Ishida, T. Moving target search with intelligence. In Proceedings of the 10th National Conference on Artificial Intelligence (AAAI), pp. 525-532.1992
[51]. Dean, T. L. and M. S. Boddy. An Analysis of Time-Dependent Planning. AAAI.1998
[52]. Likhachev, M., et al. ARA*: Anytime A* with provable bounds on sub-optimality. Advances in Neural Information Processing Systems.2003
[53]. Chiddarwar, S. S. and N. R. Babu. "Offline decoupled path planning approach for effective coordination of multiple robots." Robotica 28(4): 477-491. 2010
[54]. Kopriva, S., et al. Iterative accelerated A* path planning. Decision and Control (CDC), 2010 49th IEEE Conference on, IEEE. 2010
[55]. Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning.84,Technical Report TR 98-11, Computer Science Department at Iowa State University, 1998.
[56]. Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J.P. How. Realtimemotion planning with applications to autonomous urban driving.IEEE Transactions on Control Systems, 17(5):1105–1118, 2009.
[57]. S. Karaman and E. Frazzoli. Incremental sampling-based algorithms for optimal motion planning. Robotics: Science and Systems (RSS), 2010.
[58]. James Bruce and Manuela Veloso. Real-time randomized path planning for robot navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), volume 3, pages 2383–2388, 2002.
[59]. Jinhan Lee, C. Pippin, and T. Balch. Cost based planning with rrt in outdoor environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 684–689, September 2008.
[60]. S. Karaman, M.R. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime motion planning using the RRT*. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 1478–1483, May 2011.
[61]. E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. IEEE Transactions on Robotics and Automation, 8(5):501–518, October 1992.

Journal of Theoretical and Applied Information Technology
10th December 2014. Vol.70 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table: Compares The Reviewed Approaches In The Field Of Robot Path Planning In Dynamic Environments Based On Search Time And Average Cost.

Method	**Algorithm**	**Limitation**					
		Search time			**Average cost**		
		low	medium	high	low	medium	high
Heuristic	2D Dijkstra heuristic			✓			✓
Heuristic	A*			✓			✓
Incremental heuristic search	FSA*			✓			✓
Incremental heuristic search	Adaptive A*		✓				✓
Incremental heuristic search	Differential A*			✓		✓	
Incremental heuristic search	Focused Dynamic (D*)			✓		✓	
Incremental heuristic search	D* Lite			✓		✓	
Real time heuristic search	LRTA*			✓			✓
Real time heuristic search	LRTA*(k)		✓				✓
Real time heuristic search	$LRTA_{LS}(k, d)$		✓			✓	
Real time heuristic search	LSS-LRTA*		✓			✓	
Real time heuristic search	RTD*		✓			✓	
Real time heuristic search	PLRTA*		✓		✓		
Real time heuristic search	daLSS_LRTA*	✓				✓	
Real time heuristic search	ERRT			✓		✓	
Any-time	ARA*		✓			✓	
off-line algorithms	Time bounded lattice			✓			✓
off-line algorithms	A*(AA*)			✓			✓
off-line algorithms	IAA*			✓			✓
Randomized Sampling Methods	RRT			✓			✓
Randomized Sampling Methods	MA-RRT			✓		✓	
Randomized Sampling Methods	RRT*			✓		✓	
Reactive Methods	Potential field			✓			✓