Robot Navigation: Review of Techniques and Research Challenges

Dr. Swati Aggarwal*
Department of Computer Engineering, NSIT, Delhi University, New Delhi, India
*Corresponding Author
Email Id: swati1178@gmail.com

Kushagra Sharma
Department of Computer Engineering, NSIT, Delhi University, New Delhi, India
Email Id: sharma.kushagra7@gmail.com

Manisha Priyadarshini
Department of Computer Engineering, NSIT, Delhi University, New Delhi, India
Email Id: manisha9735@gmail.com

Abstract – Robot navigation has multi-spectral applications across a plethora of industries. Being one of the fastest evolving field of Artificial intelligence, it has garnered a lot of speculation and the problem of efficient Robot navigation has become a talking point in the "Artificial Intelligence research community". Robot Navigation can be thought of as a collection of sub-problems that when solved efficiently will (most likely) give the complete and the most efficient solution. While the main Navigation problem can be divided into numerous smaller ones some of the prevalent (sub)-tasks associated with Robot Navigation are Path Planning, Collision Prevention, Search Algorithm and an appropriate pictorial (graphical) representation of the given environment. Path Planning gives us the most optimal path finding approach while keeping the dynamism of the environment and the objects in mind. Collision Prevention ensures the Robot reaches its goal without colliding with any of the objects present in the environment. Collision Prevention techniques' vitality increases considerably when the number of objects in the input space is large (>10, approximately). Search Algorithms are the functional units of this problem as they determine the path the Robot follows while traveling to its destination. Since the environment greatly affects the choice of technique for all the other tasks it is vital to have a clear, concise and a non-ambiguous representation of the same. The main aim of the paper is to summarise the development of various techniques in the multiple domains of the Robot Navigation problem in the last fifteen years (2000-2015) while stating the scope and the restriction of each of them.

Keywords– Robot Navigation; Robot Path Planning; Robot Collision prevention; Robot Search Algorithms

I. INTRODUCTION

Robot Navigation finds extensive use in many real time domains. Research areas like Nano-bots**[1]** and robotic Arm**[2]** have picked up speed and carry the potential of revolutionizing the field of surgery and cancer treatment. Other fields that have benefited from the growth in Robot Navigation are Data Mining**[3]**, Automation in Manufacturing**[4]** and CAD**[5].** Robotics in Artificial Intelligence has come a long way but a lot still needs to be done. While independent works have been done on each of the problem a hybrid- modernistic approach is yet to be formed. A lot of existing papers under this domain have presented numerous solutions to the sub-tasks of the Navigation Problem but very few have been able to merge that into a complete, wholesome and a restraint free solution. Most of the completed work in this field focuses primarily on the hardware component of the problem and treats the Robot as a physical entity. This instills a prominent restraint in the problem- that of uncertainty in the change in the environment the robot is subjected to. While the ultimate goal of Robot Navigation is to implement it in the real world, artificial simulation would enable us to optimise the solution before its real implementation.

The first step to attaining a concrete (complete) solution would be to put together the current techniques of all the sub-problems underlining their advantages and anomalies.

II. LITERATURE REVIEW

The authors found the following to be the core sub-problems of the Robot Navigation problem-

1. Path Planning
2. Collision Prevention
3. Search Algorithms
4. State-Space representation of the Environment

Some other important work areas in the domain of the given problem were - Image analysis**[6]** for the efficient working of the sensors (receptor area), Switching technique**[7]** -used to switch between multiple path finding algorithms, and Scope of actions of the Robot - the kind of actions a Robot can be made to perform after the all the other aspects have been put to use. Image analysis not only plays an important role in the visual reception of the robot**[6]** (further aiding its navigation) but it also helps the user to get a clear picture of the spatial representation of the obstacles and the path. Since, Robot Navigation is a complex problem with inherent intricacies of uncertainties and dynamism it is pragmatic to use multiple search algorithms for a given environment. The choice of a particular algorithm at a specific instant time is determined by an objective function and the transition from one technique to another is governed by a proper switching technique**[7].**

Since Path Planning is widely accepted as the first step in the solution building process the paper also begins with the same. The techniques reviewed under this are Deterministic Path

Planning and Metric Path Planning. While deterministic path planning deals with concepts like associative probability**[8]** and backtracking**[9]** Metric Path Planning employs techniques such as Configuration Space**[10]** and Degrees of Freedom**[11].** Deterministic Path Planning is one of the oldest Path planning techniques and its fundamentals of "heuristic**[22]**" and "fuzzy logic**[23]**" have been employed in subsequent techniques. Metric Path Planning is an advanced method and divides the input environment into a configuration space. It also assigns the Robot some specific degrees of freedom. The restraints of the configuration space and the allotted degrees of freedom make the basics of this Planning technique.

Dynamic window approach**[12]** uses a velocity- distance**[13]** and/or a time- distance**[14]** parameter to prevent collisions and has been explained in details in further sections of the paper. A velocity-distance method fixes a threshold velocity which must be maintained to avoid collision give the robot enough time to change course) while a time-distance method fixes a minimum distance that must be maintained to avoid any possible collision. Since velocity, time and distance are mathematically related it is practically impossible to optimise all three. Hence, the choice of the parameter is both user and space oriented.

Under pictorial representation of the environment the authors have reviewed Meadow Maps**[15]** and Voronoi graphs**[16]**. Meadow maps treat the input space as a set of convex polygons and the method assumes that if the robot travels along the edge of any of the given polygons it will not "fall off the grid" (remain on course to the destination). Voronoi Graphs use topological method of representation**[17].** They convert the space into a set of nodes and create Voronoi edges which are equidistant from a set of points The competences and drawbacks of each have been talked in the latter sections of the paper.

Search algorithms such as A*, D*, D*Lite and LPA* have been critiqued by the authors. A* search algorithm uses heuristic and best-first search to determine the path. D* allows the user to dynamically re-adjust the graph after every iteration of the algorithm. LPA* stands for lifelong Planning A* algorithm. It keeps a track of all the previously visited nodes and once the current path is found to be futile it changes the course in an efficient time. D* Lite traverses the entire space in the reverse order beginning from the leaf nodes to the goal node. This makes the path-cost adjustment faster.

III. RESEARCH CHALLENGES

Some of the research challenges in this domain are –

- Coming up with an optimal switching technique that continuously transitions from one searching method to other as per the current configuration of the environment.**[24]**
- A proper objective function that lays a concrete foundation for the probability mapping of the position of the obstacles (and other robots) to the real time domain.
- A more efficient collision prevention technique- one that does not require the user to compromise on one parameter to optimise the others.
- A multiple Robot-multiple obstacle dynamic environment. **[25]**

Inculcating an observable EQ (emotional quotient) in the robot so that it is more aware of the sentimental aspect of the environment.**[26]**

IV. PATH PLANNING

Path Planning is used to form the basic and initial foundation to the navigation problem. It forms a navigation policy for the subsequent steps of the navigation.

Associative Probability in Deterministic Planning is used to determine the possibility of occurrence of the position of an object/robot at any future instant of time. As briefly mentioned in *table I*, this probability is used to define the likelihood and desirability of the said object position at any time in the future. In case of the occurrence of an unprecedented event backtracking is used to retrace the previous paths of the algorithm and find another path. This backtracking is supported with the reconstruction of graph/pictorial representation of the environment. The final step in Deterministic Planning is the implementation of a suitable search algorithm. Deterministic Planning is one of the oldest Planning techniques *(table I)* and has been rendered obsolete with the numerous advancements in the said domain. However, some if its concepts such as associative probability and backtracking still form the backbone newer Planning techniques such as Metric Path Planning.

Metric Path Planning treats the input environment as Configuration Space and allocates certain degrees of freedom to the Robot. Configuration Space along with the specified degrees of freedom forms the basis of the navigation technique. This technique is a definitive improvement over the Deterministic method but suffers from drawbacks of its own. The choice of the number of degrees of freedom is ambiguous and no mathematical model clearly states the optimal number of degrees that should be assigned to the robot. Also, a user might end up allocating more than necessary number of degrees of freedom thereby decreasing the efficiency of the technique.

TABLE I. PATH PLANNING IN ROBOT NAVIGATION

METHOD	DESCRIPTION	LIMITATIONS	FUTURE SCOPE
Deterministic Path Planning [1996] **[7]**	The main concept of this technique has been sub-divided into four points- 1. Associative probability 2. Backtracking 3. Reconstruction of the graph 4. Implementation of search algorithm	The method uses landmarks and checkpoints to guide the robot towards its goal It assumes the checkpoints to be static and objective in nature Doesn't allow the re-planning of the input space once the	The technique has been rendered obsolete and has been replaced by more efficient methods

		robot has begun the navigation	
Metric Path Planning 2000 **[10]**	The main concept behind this approach is categorisation of the input space as Configurations Space and the allocation of certain degrees of freedom to the Robot.	Assumes the robot to be holonomic[]- the robot can turn in any direction Works efficiently only when the degree of freedom is 2 or less	Anomalies of metric path planning are being worked on and RRT algorithm[] is one of the solutions

V. COLLISION PREVENTION

Dynamic Window approach is used to ensure a collision free movement of the robot. It specifies a threshold velocity (for the robot) which allows the user to maintain a minimum distance between the robot and the obstacles**[14].** It is assumed that if at all times the robot maintains its velocity lower than the threshold velocity (*table II*) it will have enough time to change its course or the current velocity to avoid a collision. Since velocity, time and distance have a close mathematical relation this approach is more like a trade-off between the three. In other words, all three parameters cannot be optimized simultaneously. Maintaining a larger threshold velocity would mean a smaller time allowance for robot to alter its course. Similarly, keeping a large time allowance would mean a smaller permissible range of velocities which would greatly reduce the efficiency of the robot. The decision to fix one parameter and change the others is situation-dependent. This approach, however, does not consider the situation where the obstacles have uncontrolled acceleration as in such a case it would be nearly impossible for the robot to avoid collision. It would also suffer in a scenario where the mean velocities of the obstacles are much greater than that of the robot(s)

As can be seen by *table II*, dynamic window approach has some pertinent issues. The choice of the threshold velocity is subjective and is yet to be supported by a universal mathematical function. Moreover, the approach makes the user compromise on one (or more) parameter to optimise the other one. Having said that, the authors feel this technique is the most promising one and can be applied to multiple situation

TABLE II-COLLISION PREVENTION IN ROBOT NAVIGAITON

METHOD	DESCRIPTION	LIMITATIONS	FUTURE SCOPE
Dynamic Window Approach	**1.** A velocity-distance/Time-distance method.	Uncontrolled acceleration of the obstacle	Approximation of objective function to overcome the restricted range of velocities is the way forward
[1997]	**2.** Estimates the time to possible collision and changes the path (Time-distance)**[13]**		
[12]	**3.** Fixes a threshold distance (and velocity) to avoid collisions.	Trade-offs between distance and time and velocity and time.	

with the aid of a mathematical model and appropriate heuristics.

VI. PICTORIAL REPRESENTATION

Meadow maps are used to divide the entire environment into a set of convex polygons (*table III*)and work on the principle that if a robot keeps walking on the edge of any of those polygons it will not fall off (collide or go off-track). As mentioned in *table III*, The major drawback with this approach is that it assumes the input space to be simple enough to be divided into convex polygons. This approach would suffer badly when the environment is extremely dynamic as a major chunk of the space will be left out of the representation.

Topological representation works around landmarks (*table III*) or checkpoints and navigates the robot on the basis of the checkpoints reached at and the landmarks ahead**[27]**. The choice of the same is subjective and thus an objective mathematical function cannot be formed in this case. This method overlooks the fact that landmarks do not give a detailed picture of the environment and that the chosen landmarks may not remain there all the time. Hence, this technique is now obsolete and is used as a sub-solution for other approaches.

Voronoi graphs are the latest development and make use of topological representation while giving the freedom of dynamic reconstruction. They create a topological map of the environment and divide the same into spots. After, that Voronoi edges**[28]**, which are equidistant from surrounding vertices are drawn and the robot is made to navigate along these edges. This representation would not work well in cases where no visible/prominent checkpoints exist.

The main task in this sub-problem would to be come up with a more efficient way of categorising the input space and then binding that with a suitable dynamic reconstruction technique.

TABLE III- PICTORIAL REPRESENTATION (ROBOT NAVIGAITON)

METHOD	DESCRIPTION	LIMITATIONS	FUTURE ASPECT
Meadow Maps [2000] **[15]**	They convert the entire configuration space into a set of convex polygons and work on the principle that if a Robot travels along the edge of a polygon it won't fall off.	Not every environment is simple enough to be divided into regular convex polygons.	Finding a way to make this technique work for more complex scenarios, inculcating concave polygons.
Topological Representation [1999] **[17]**	Divides the space in accordance with the presence of landmarks which are then chosen to determine the position of the Robot and also to navigate it further	The choice of landmarks is a subjective one and a landmark's presence is not permanent	The technique has been rendered obsolete with the development of more advanced approaches
Vornoi Representation [2010]	Creates a topological map of the input	Would not be efficient in a case where the	Is currently the most suitable technique (in the

[16]	space. Vornoi edges are created by making them equidistant from the nearby landmark points (indicating the important physical checkpoints).The point where two Vornoi edges meet is called a Vornoi Vertex.	environment does not have any visible checkpoints.	authors' opinion)

VII. SEARCH ALGORITHMS

Search algorithms form the fundamental functional unit of the Robot Navigation problem. All other sub-problems' efficient solution aid the searching technique. An efficient search technique requires all the previous methods to be optimised as this is the final step of the main task. While a large number of searching techniques have been developed and used in multiple related domains over the years the authors have reviewed four.

A* search algorithm was given in 1968 and has proven to be the stepping stone in the "search-technique" domain. It combines the advantages of Breadth-First Search**[30]** and Depth First search**[31]** and gives a hybrid Best-First search (*table IV*). It sets a list of nodes that sum up the input space and applies heuristic technique to move further. Once the current path is found to be futile it backtracks and chooses the next best node.

LPA* is an out-branch of A* (*table IV*) search technique and uses two parameters to aid its heuristic**[32].** These parameters are called keys and are fed to a heuristic function that calculates the desirability of the next node. It also offers a faster retracing in case of a dead path. Both A* and LPA* suffer from the limitation of static environment and cost calculation. These algorithms do not update the environment structure or the cost of the paths after iterations.

D* algorithm **[33]**provides a solution to this anomaly by enabling the automatic recorrection of the path in the occurrence of rare

TABLE 1V- SEARCH ALGORITHMS IN ROBOT NAVIGAITON

METHOD	**DESCRIPTION**	**LIMITATIONS**	**FUTURE ASPECT**
A* [1968] **[30][31]**	Divides the state-space into a set of nodes and creates an initial map. These nodes are then subsequently subjected to a heuristic fashion.This algorithm is also called Best First search as it incorporates the advantages of both Depth First search (DFS)**[18]** and breadth First search (BFS**[19]**). Once the current path is seen to be no longer viable the algorithm backtracks and chooses the next best path. This sequence of iterations is repeated until a final solution is arrived at.	Not efficient in a discrete environment. Choice of nodes for the construction of the graph is ambiguous. Does not allow the input space to be updated after each iteration.	This technique can be used as a part solution along with other search algorithms.
LPA* algorithm [2004] **[32]**	An improvement over the A* algorithm as it uses two parameters to estimate the distance from a prospective node to the destination. These two parameters are called keys and are used to determine a Priority Queue**[20]**. This queue used to update the algorithm once a path is termed futile or too "expensive" and the provides the next best node in a more efficient time than its predecessor	Presence of two keys for determination of the likelihood makes the objective function quite complex. No reconstruction of the graph.	Can work well if it has a provision for the dynamic correction of the input space.
D* Algorithm [1995] **[33]**	Enables the correction (updation) of the path when a "rare" event occurs. This algorithm not only updates the environment map but it also continually updates the cost of each node, edge and path with each iterative step. This greatly speeds up the switching between	Time space complexity is high- impacts the efficiency of the algorithm	Currently the most suitable technique (in the authors' opinion)

	multiple paths and cost-correction while doing so.		
D* Lite Algorithm [2005] **[33]**	An improvement over D* algorithm Works in the reverse order. Starts from the goal state and woks towards the start goal updating the costs to the goal state after every step	Would not work well in cases where the goal state is uncertain or there are multiple goals	Can be used in accordance with D*. The space can be traversed from both directions.

events (*table IV*). It also updates the pat costs and the entire input structure after iteration. This largely eliminates the need to backtrack and consequently saves time. D* Lite **[33]** algorithm moves in the reverse order, starting from the goal state towards the start state. It treats the start state to be the root of the tree and the goal state(s) to be the leaf nodes. It starts from the leaf nodes and traverses to the root node. This algorithm does not automatically place the robot at the goal state. Only the calculative steps are reversed. The reconstruction and recorrection advantages of D* algorithm are sustained.

The main challenge here would be to design a way for the robot to switch between two or more search techniques as per the requirement of the current state and use that to create a universal switching model.

VIII. CONCLUSION

While Robot Navigation has found extensive use across multiple domains it is yet to have a consistent, concrete and a complete solution. Various sub-problems of Robot Navigation have been reviewed and worked on but the said work is yet to provide a clear-cut solution to the main task at hand. Numerous techniques, some of which the authors have reviewed, have evolved to tackle the underlying problems and each has presented with its distinct advantages and limitations. Being a part of one of the newest sciences (Artificial Intelligence) Robot Navigation has made a visible impact on industries such as medical and manufacturing. Having said that, there are many I(s) to dot and many T(s) to cross before we start to realize the full potential of this amazing concept.

REFERENCES

[1]. Nerlich, B. (2008). Powered by imagination: nanobots at the Science Photo Library. Science as Culture, 17(3), 269-292

[2]. Geis, W. P., Kim, H. C., Brennan, E. J., McAfee, P. C., & Wang, Y. (1996). Robotic arm enhancement to accommodate improved efficiency and decreased resource utilization in complex minimally invasive surgical procedures. Studies in health technology and informatics, 471-481.

[3]. Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and techniques: concepts and techniques. Elsevier.

[4]. Asfahl, C. (1992). Robots and manufacturing automation. John Wiley & Sons, Inc..

[5]. Tzu, S. D., Tu, S. C., & Lin, C. H. (1999). U.S. Patent No. 5,994,009. Washington, DC: U.S. Patent and Trademark Office.

[6]. Serra, J. (1983). Image analysis and mathematical morphology. Academic Press, Inc..

[7]. Gal, A., Vladareanu, L., Smarandache, F., Yu, H., & Deng, M. (2013). Neutrosophic Logic Approaches Applied to" RABOT" Real Time Control.Neutrosophic Theory and Its Applications., 55.

[8]. Timcenko, A., & Allen, P. (1994, May). Probability-driven motion planning for mobile robots. In Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on (pp. 2784-2789). IEEE.

[9]. Ginsberg, M. L. (1993). Dynamic backtracking. Journal of Artificial Intelligence Research, 25-46.

[10]. Lozano-Perez, T. (1983). Spatial planning: A configuration space approach.Computers, IEEE Transactions on, 100(2), 108-120.

[11]. Campbell, D. T. (1975). III."Degrees of freedom" and the case Study.Comparative political studies, 8(2), 178-193.

[12]. Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), 23-33.

[13]. Simmons, R. (1996, April). The curvature-velocity method for local obstacle avoidance. In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on (Vol. 4, pp. 3375-3382). IEEE.

[14]. Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research, 5(1), 90-98.

[15]. Arkin, R. C. (1987, March). Motor schema based navigation for a mobile robot: An approach to programming by behavior. In Robotics and Automation. Proceedings. 1987 IEEE International Conference on (Vol. 4, pp. 264-271). IEEE.

[16]. Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence, 99(1), 21-71.

[17]. Friedman, S., Pasula, H., & Fox, D. (2007, January). Voronoi Random Fields: Extracting Topological Structure of Indoor Environments via Place Labeling. InIJCAI (Vol. 7, pp. 2109-2114).

[18]. Tarjan, R. (1972). Depth-first search and linear graph algorithms. SIAM journal on computing, 1(2), 146-160.

[19]. Zhou, R., & Hansen, E. A. (2006). Breadth-first heuristic search. Artificial Intelligence, 170(4), 385-408.

[20]. Rönngren, R., & Ayani, R. (1997). A comparative study of parallel and sequential priority queue algorithms. ACM Transactions on Modeling and Computer Simulation (TOMACS), 7(2), 157-209.

[21]. Song, K. T., & Sheen, L. H. (2000). Heuristic fuzzy-neuro network and its application to reactive navigation of a mobile robot. Fuzzy Sets and Systems,110(3), 331-340.

[22]. Wang, M., & Liu, J. N. (2005, August). Fuzzy logic based robot path planning in unknown environment. In Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on (Vol. 2, pp. 813-818). IEEE.

[23]. Xu, W. L., & Tso, S. K. (1999). Sensor-based fuzzy reactive navigation of a mobile robot through local target switching. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 29(3), 451-459.

[24]. Meyer, J. A., & Filliat, D. (2003). Map-based navigation in mobile robots:: Ii. a review of map-learning and path-planning strategies. Cognitive Systems Research, 4(4), 283-317.

[25]. Balch, T., & Hybinette, M. (2000). Social potentials for scalable multi-robot formations. In Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on (Vol. 1, pp. 73-80). IEEE.

[26]. Fellous, J. M. (2004). From human emotions to robot emotions. Architectures for Modeling Emotion: Cross-Disciplinary Foundations, American Association for Artificial Intelligence, 39-46.

[27]. Mataric, M. J. (1992). Integration of representation into goal-driven behavior-based robots. Robotics and Automation, IEEE Transactions on, 8(3), 304-312.

[28]. Kuipers, B., Modayil, J., Beeson, P., MacMahon, M., & Savelli, F. (2004, April). Local metrical and global topological maps in the

hybrid spatial semantic hierarchy. In Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on (Vol. 5, pp. 4845-4851). IEEE.

[29]. Hansen, E. A., & Zilberstein, S. (2001). LAO∗: A heuristic search algorithm that finds solutions with loops. Artificial Intelligence, 129(1), 35-62.

[30]. Korf, R. E. (1993). Linear-space best-first search. Artificial Intelligence, 62(1), 41-78.

[31]. Korf, R. E. (1985). Depth-first iterative-deepening: An optimal admissible tree search. Artificial intelligence, 27(1), 97-109.

[32]. Manley, K. (2003). Pathfinding: from A* to LPA. University of Minnesota.

[33]. Koenig, S., & Likhachev, M. (2002, July). D* Lite. In AAAI/IAAI (pp. 476-483).