World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:12, No:6, 2018

A Review on Comparative Analysis of Path Planning and Collision Avoidance Algorithms

Divya Agarwal, Pushpendra S. Bharti

***Abstract*—**Autonomous mobile robots (AMR) are expected as smart tools for operations in every automation industry. Path planning and obstacle avoidance is the backbone of AMR as robots have to reach their goal location avoiding obstacles while traversing through optimized path defined according to some criteria such as distance, time or energy. Path planning can be classified into global and local path planning where environmental information is known and unknown/partially known, respectively. A number of sensors are used for data collection. A number of algorithms such as artificial potential field (APF), rapidly exploring random trees (RRT), bidirectional RRT, Fuzzy approach, Purepursuit, A* algorithm, vector field histogram (VFH) and modified local path planning algorithm, etc. have been used in the last three decades for path planning and obstacle avoidance for AMR. This paper makes an attempt to review some of the path planning and obstacle avoidance algorithms used in the field of AMR. The review includes comparative analysis of simulation and mathematical computations of path planning and obstacle avoidance algorithms using MATLAB 2018a. From the review, it could be concluded that different algorithms may complete the same task (i.e. with a different set of instructions) in less or more time, space, effort, etc.

***Keywords*—**Autonomous mobile robots, obstacle avoidance, path planning, and processing time

I. Introduction

AMR are employed in widespread application domains such as manufacturing sector, nuclear power plants, space exploration operations, under water exploration, medical assistance etc., thereby considered as smart operational tools [1], [2]. For a robot to work autonomously, path planning and obstacle avoidance are two important control parameters [3]. Path planning can be deliberated as an optimization problem whose objective is to find a path from start location to destination through various waypoints in a cluttered environment so that the robots does not get collided or get struck in a local minima/artificial roadblock in accordance to some constraints such as distance, time, cost or energy [4], [5]. Path planning and obstacle avoidance algorithm in a robot must be able to build a map, propose optimum paths, and maneuver the robot autonomously avoiding obstacles within the path [5], [6]. Information of static obstacles must be given to the robot while building the map, whereas sensors are used to sense the real-time environmental conditions to avoid getting struck and maneuver easily for dynamic obstacles [6].

D. Agarwal (Research Scholar) is with the Guru Gobind Singh Indraprastha University, Dwarka Sector 16-C, Delhi- 110078, India (phone: +91 9968362010, e-mail: er.divya7@gmail.com).

P. S. Bharti (Associate Professor) is with the Guru Gobind Singh Indraprastha University, Dwarka Sector 16-C, Delhi- 110078, India (corresponding author, phone: +91 9810478041, e-mail: psbharti@ipu.ac.in).

Ideally the path planning and obstacle avoidance algorithm must handle with real-time environmental uncertainties, to minimalize collision with obstacles and find optimum path in least time. Complexity of the algorithm rises exponentially with rise in the floor space [1], [3]. Path planning problem is composed of two sub-problems- Findspace (in which floor space or map is constructed with static obstacles) and Findpath (in which collision free path is designed and a robot is desired to maneuver from start to destination point) [4]. There are two types of Path planning problem [5] – Global Path Planning in which complete environmental information (steady state) is known to the system to reach destination, such as RRT [7], [8], A* Algorithm [9], [10], Pure Pursuit [11], Particle Swarm Optimization (PSO) [12], [13] etc., [14] and Local Path Planning in which real-time environmental information is required such as Artificial Neural Networks (ANN) [15], [16], APF algorithm, Genetic Algorithm (GA) [17] and Fuzzy Logic Algorithm [12], [18]-[21]. Obstacle avoidance algorithm basically deals with designing path with anticipated and unanticipated obstacles and how to overcome it without collision [22], [23]. The objective of obstacle avoidance algorithms is to alter real-time robot's trajectory to avoid collisions along the path and maneuver with the help of feedback data from sensors and real-time environment [24], [25]. Every algorithm starts with choosing an object and initializing it with an initial location and orientation, destination point, its orientation is also initialized to object, and certain obstacles are initialized in workspace [18]-[24]. The problem is to arduously maneuver along the way continuously from start point to destination without getting collided with an obstacle.

A*algorithm does not need complete information about the environment and uses a grid-based map [9], [10], [26], while RRT considers starting location as a root and extends it to branches to optimally plan a collision free path [7], [8]. Conventional algorithms such as visibility/Voronoi graphs, potential fields and cell decomposition, etc. get struck into local minima and require more time for complex configuration space for planning a collision free path. Soft computing algorithms such as PSO, GA and fuzzy logic [27], etc. are used for this problem. Modified A*path planning algorithm plans an obstacle-free path with optimal computational time considering issues such as real-time obstacle avoidance and smooth path maneuvering [28]. Challenging issue in path planning and obstacle avoidance algorithm is to find an obstacle free path in real-time situations in low computational time. Certain algorithms which can pass through this challenge are A* algorithm [27], APF algorithm [29], heuristic algorithms [30], [31].

Objective of this review article is to present an overview and discuss strength and weakness of path planning and obstacle avoidance algorithms developed and used by previous and current researchers. A brief overview of some of the selected algorithms is presented along with discussion of their effectiveness in Section II. Algorithms are simulated and mathematically computed using MATLAB 2018a giving an insight into strength and weakness of reviewed algorithms in Section II. Simulation and experimental results show that algorithms play an important role to produce an optimal path (short, smooth and robust) for AMR and simultaneously it proves that appropriate algorithms can run fast enough to be used practically in the shortest possible time.

II. Path Planning and Obstacle Avoidance Algorithms

Numerous algorithms have been developed over the last few years to create real time path planning and obstacle avoidance algorithm system for autonomous robots [18], [19], [21], [32]. There are three things or activities that must be followed by an AMR to enable the execution of the task of robot maneuvering. These activities are mapping the environment, path planning and obstacle avoidance [2], [5], [18]-[21]. Selection of an appropriate algorithm is important to ensure that the maneuvering process will run smoothly. The following algorithms were simulated and mathematically computed using MATLAB 2018a.

A. Probabilistic Roadmap (PRM)

PRM is a sampling-based algorithm which comprises of networks of connected nodes in a given map based on free and occupied spaces to find an obstacle free path from start to end point [18], [33]. PRM generates nodes to create connections among them based on algorithm parameters. Number of nodes can be customized to fit complexity of map to find the most optimal collision free path [33]. It uses network of connected nodes to find an obstacle-free path from a start to an end location. Algorithm begins road map with a node at the initial configuration which is then expanded randomly by adding edges and nodes further [14], [34]. This algorithm was coded and simulated in MATLAB 2018a for known environment with static unknown obstacles. In the MATLAB code, simple (Figs. 1-3) and complex (Figs. 4-6) binary occupancy maps were used, and then, the start and goal positions along with obstacles were mentioned for the robot to navigate autonomously through the path.

Path Traversed

2.0000	1.0000	7.5974	8.3888
2.4822	0.4653	11.0693	9.4455
2.6791	3.5829	11.0000	10.0000
4.6539	7.6582		

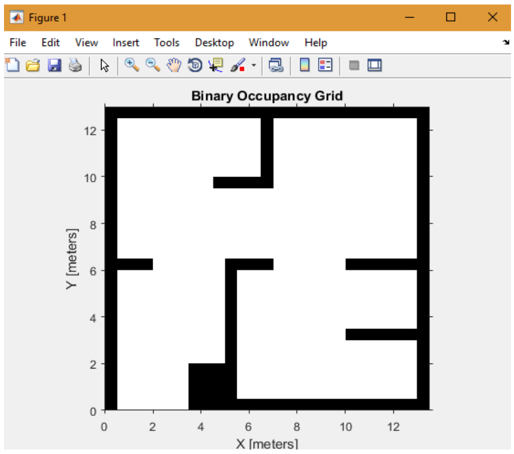


Fig. 1 Simple Map for planning path

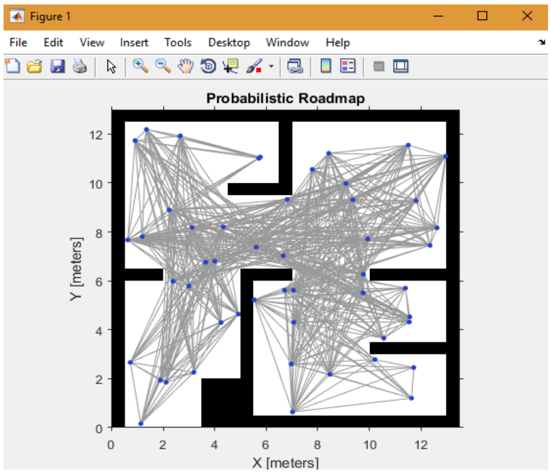


Fig. 2 Probabilistic roadmap

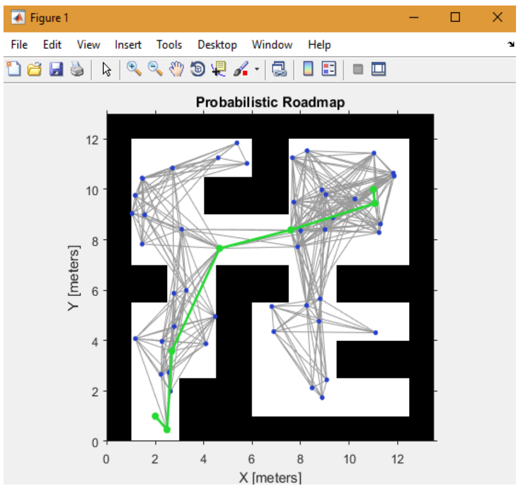


Fig. 3 Feasible path on constructed PRM

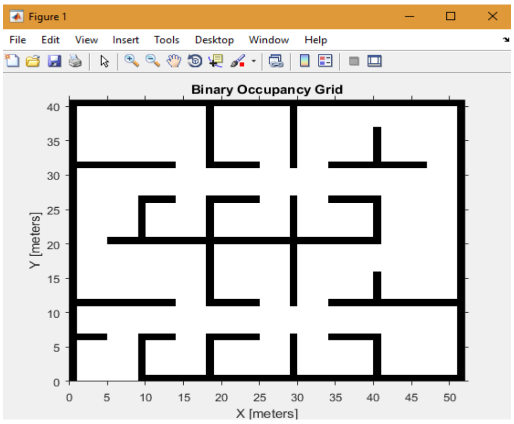


Fig. 4 Complex map for planning path

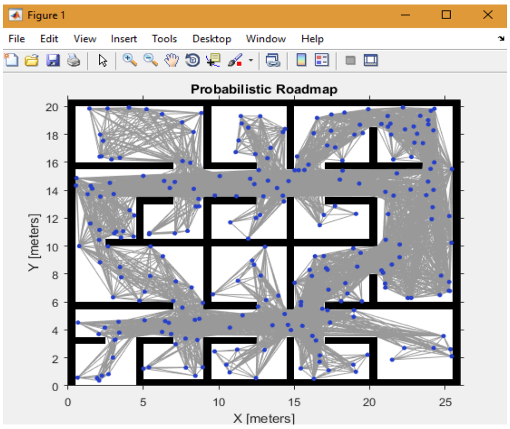


Fig. 5. Complex probabilistic roadmap

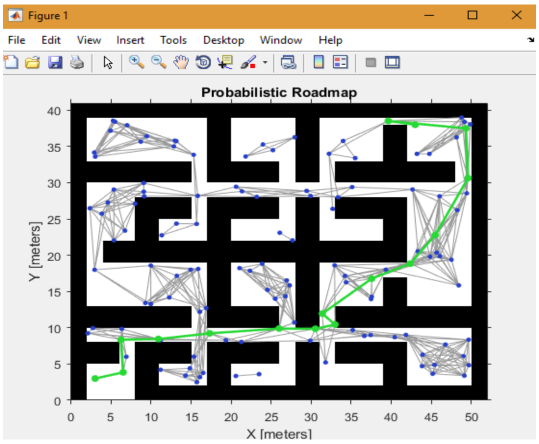


Fig. 6 Feasible path on constructed PRM

Path Traversed =

3.0000	3.0000	31.3932	11.9517
6.5095	3.8493	37.5704	16.7749
6.2334	8.3465	42.3880	18.8434
10.9156	8.4485	45.5038	22.7893
17.3253	9.2285	49.6089	30.6386
25.9983	9.8913	49.3373	37.4945
30.5245	9.8553	39.6426	38.5265
32.9894	10.4466	43.0	8.0000

B. Modified PRM

The objective of this algorithm is to reduce number of collisions along the path, thereby decreasing computation time. Initially roadmap is created, then nodes are created arbitrarily [34]-[36]. Neighboring nodes are connected using edges to represent as path. Unlike PRM, this algorithm considers collision free path existing in between edges and nodes followed by shortest path search [34], [36]. In searching process, collisions are checked, if there is any obstacle in path, then the edges comprising that path are dropped from the roadmap. Later on, roadmap is updated with new nodes and edges and the process of shortest path search starts again [36].

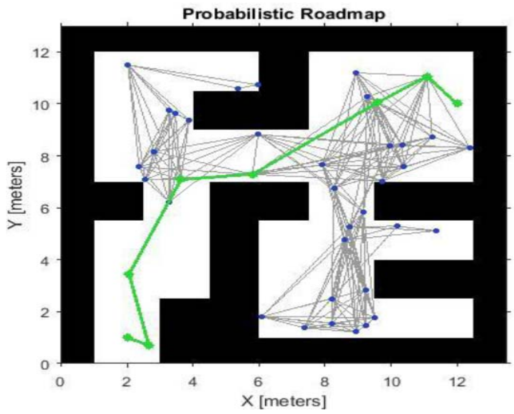


Fig. 7 Final traversed path for simple map

Path Traversed =

2.0000	1.0000	5.8039	7.2873
2.6602	0.7147	9.5753	10.0357
2.0617	3.4421	11.0932	11.0593
3.6323	7.0813	12.0000	10.0000

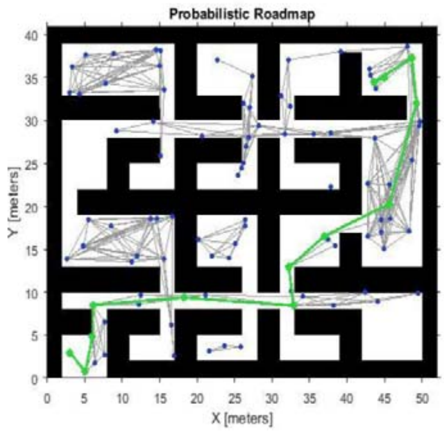


Fig. 8 Final traversed path for complex map

Path Traversed =

3.0000	3.0000	32.2508	12.9018
5.0746	0.7264	36.9467	16.5280
6.0223	4.8307	45.6330	20.2881
6.0848	8.5292	49.2606	32.0227
18.2287	9.4102	48.6877	37.3380
32.8779	8.4651	43.6344	34.4697

The difference between PRM and Modified PRM is that the former builds the roadmap using feasible paths, while the latter builds the roadmap using randomly selected collision free paths [14], [18], [34], [36]. Modified PRM algorithm for path planning and obstacle avoidance in mobile robot was coded, mathematically computed, and simulated using MATLAB 2018a for known environment with static unknown obstacles. The code was written using simple (Fig. 1) and complex (Fig. 4) binary occupancy maps, then the start position and the goal position were mentioned in the code for the robot to navigate autonomously avoiding static obstacles through path. Final traversed path is shown in Figs. 7 and 8 for simple and complex maps, respectively.

III. Artificial Potential Field

Potential field approach was first proposed by Khatib [37]. It is based on principle of imaginary forces acting on a robot termed as Potential field [5]. APF algorithm is a reactive algorithm technique, where immediate distances from obstacles are considered to compute the immediate move, without much bothering about the future [37], [38], leading to goal. All obstacles repel the robot with a magnitude inversely proportional to the distance [39]. Resultant potential, accounting for the attractive and repulsive components, is measured and used to move the robot [39]. Potential field for a sample scenario is shown in Fig. 9. Directions indicate the direction of the potential vector [40], [41]. The distance of the obstacles at all angles from the robot is measured. In this progress report, only five distances at specific angles are measured to compute the repulsive potential (as shown in Fig. 10). In APF, robot is reflected as a particle which is engrossed in artificial positive and negative forces/potential are created by obstacles and goal, in order to move along the way collision free [41]. Goal generates positive or attractive potential such that it never deflects the desired path, while the obstacles produce negative or repulsive potential such that it repels desired path for a collision free way for the robot [40].

Advantages of APF are [37]-[41]:

- For static obstacles with well-known atmosphere, potential is offline estimated such that the robot's velocity is within the energy field from start to destination point.
- Applicable for online or real-time environment as well with added obstacle avoidance component.

Drawbacks of APF [37]-[41]:

- Gets trapped into local minima for the symmetric environment and bowl-shaped obstacles,
- Oscillatory behavior in narrow lanes.
- Computationally extensive.

Coding for APF algorithm was done in MATLAB 2018a for path planning and obstacle avoidance of mobile robot for known environment with static unknown obstacles. Here maps were designed using paint file, and each map was used to get the shortest path using APF algorithm by specifying start position and goal position along with obstacles in code to autonomously navigate through the path. Final traversed path for all three maps is shown in Figs. 11-13 along with their processing time and path length.

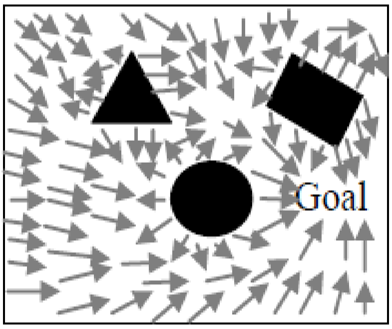


Fig. 9 APF [39]

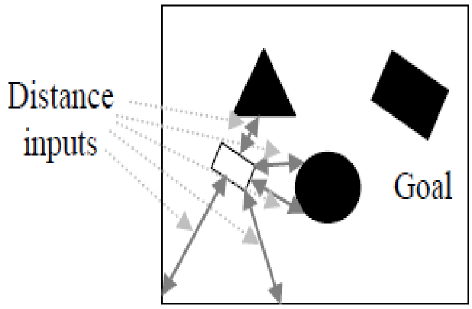


Fig. 10 Measurement of repulsive potential [39]

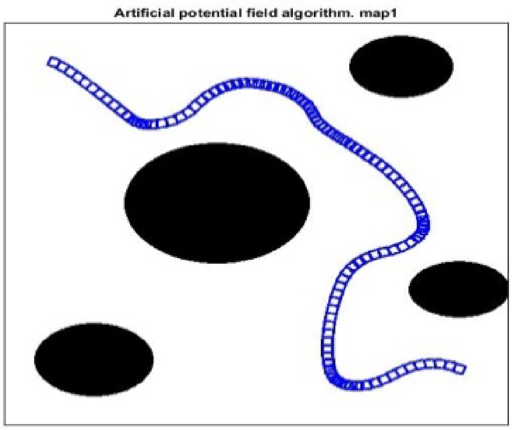


Fig. 11 APF. Map1

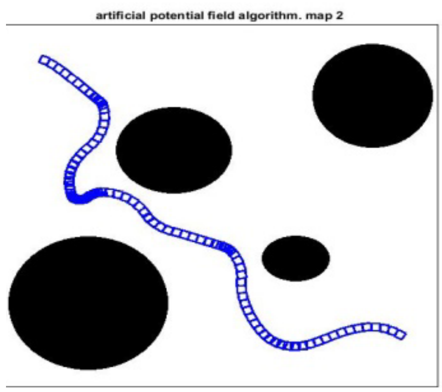


Fig. 12 APF. Map2

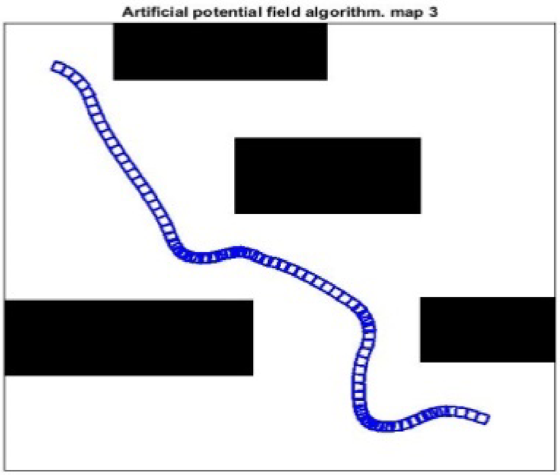


Fig. 13 APF. Map3

Readings:
APF.Map1
Processing time=9.182928e+00
Path Length=8.393897e+02
APF.Map2
Processing time=6.109794e+00
Path Length=7.847881e+02
APF.Map3
Processing time=5.838579e+00
Path Length=7.754157e+02

IV. Rapidly-Exploring Random Trees(RRT)

RRT is an alternative to PRM for non-holonomic vehicle [42]. In RRT, initially a source node or tree is created at the start point which expands or extends randomly, further to branches of the tree within the workspace, ultimately leading to optimal collision free path which is near to goal or destination node [43], [45]. It does not guarantee good performance for this problem. [43]-[47]. Tree extension might be biased towards goal's direction by selecting it as a random state with some probability [44], [46] (as shown in Figs. 14 (a) and (b)). The experimental results of path computed are shown in Figs. 15-17.

Readings:
RRT.Map1
Processing time=7.236812e+00
Path Length=8.725659e+02
RRT.Map2
Processing time=1.940650e+01
Path Length=1.320163e+03
RRT.Map3
Processing time=1.788054e+01
Path Length=1.143509e+03

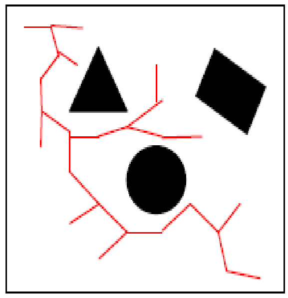


Fig. 14 (a) RRT Tree Roadmap

Fig. 14 (b) Path computed [46], [47]

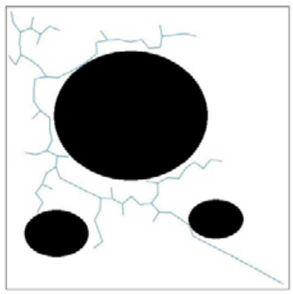


Fig. 15 (a) Map1 Roadmap

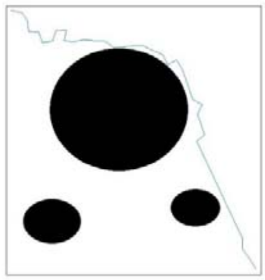


Fig. 15 (b) Map1 Path Computed

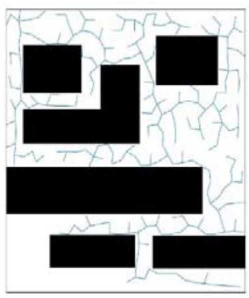


Fig. 16 (a) Map2 Roadmap

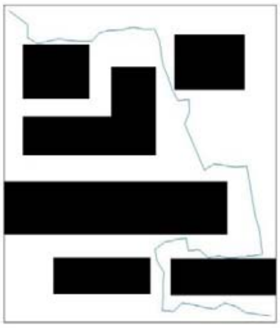


Fig. 16 (b) Map2 Path Computed

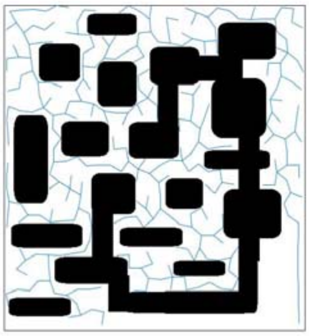


Fig. 17 (a) Map3 Roadmap

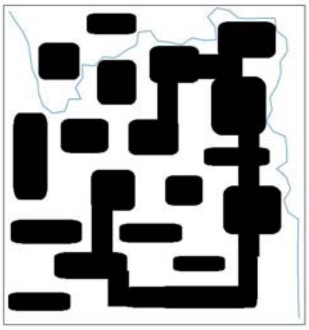


Fig. 17 (b) Map3 Path Computed

V. Fuzzy Approach

Fuzzy logic was developed by Zadeh in 1965. In this algorithm, whole logic is divided into simpler blocks composed of set of fuzzy logic rule statements intended to achieve desired objective. It is reactive planning technique in which instant position, orientation, and distances from obstacles are used to evaluate instant motion, without considering future parameters [48]. Decision of motion is prepared only on the basis of input parameters not real-time situation. Kumar et al. [49] evaluated problem using six inputs such as distances from obstacle in-front, distances from obstacle at front-left diagonal, distance from obstacle at front-right diagonal, angle between robot's heading direction and goal, distances from goal and preferred turn [49]. The different inputs are summarized in Fig. 18.

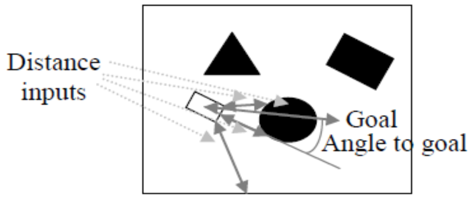


Fig. 18 Different inputs for the fuzzy planning [49]

Fuzzy rules are written in order to avoid obstacles along the waypoints and reach towards goal [49]. Maps given in the input can be captured from camera and converted into .bmp file or can be designed using paint application. Experimental results of path computed are shown in Figs. 19-21.

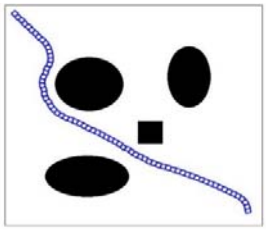


Fig. 19 Fuzzy map 1

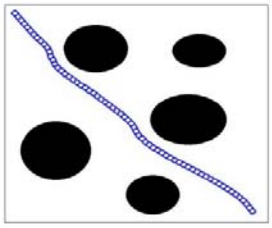


Fig. 20 Fuzzy Map 2

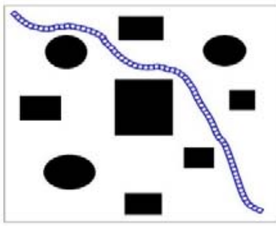


Fig. 21 Fuzzy Map 3

Readings:
Fuzzy map 1
Processing time=2.902649e+00
Path Length=9.039876e+02
Fuzzy map 2
Processing time=3.110466e+00
Path Length=8.985088e+02
Fuzzy map 3
Processing time=3.217180e+00
Path Length=9.000459e+02

VI. Genetic Algorithm

GA was introduced by John Holland [20], [50]. It starts with no prior knowledge of solution and depends entirely on real-time responses to achieve best optimal, thereby good for dynamic environments [51]-[54]. In path planning and obstacle avoidance algorithm, GA is used to move in a dynamic environment with predictable as well as unpredictable obstacles [55]. All points from source to goal location, considered one by one style genetic individual for optimization [54] (as shown in Fig. 22). Steps employed for path planning and obstacle avoidance algorithm using GA are [54]:

1. Initially grid graph is constructed out of search environment which is then maneuverer by robot in step fashion.
2. Start and stop point are specified.
3. Predictable obstacles are defined on each step of grid.

Every point in the path marks a turn, also total points in the algorithm correspond to maximum number of turns that robot can take within the workspace [54], [55]. Large workspace means high computational time and energy thereby rendering random results sometimes [20], [54]. The experimental results of path computed in MATLAB 2018a are shown in Figs. 23-25.

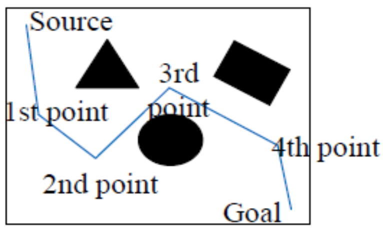


Fig. 22 GA individual representation [20]

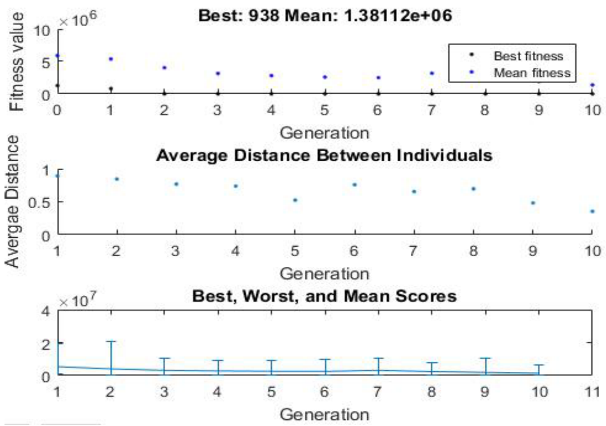


Fig. 23 (a) GA map 1 Convergence Map

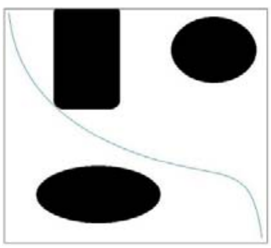


Fig. 23 (b) GA map 1 Path Computed

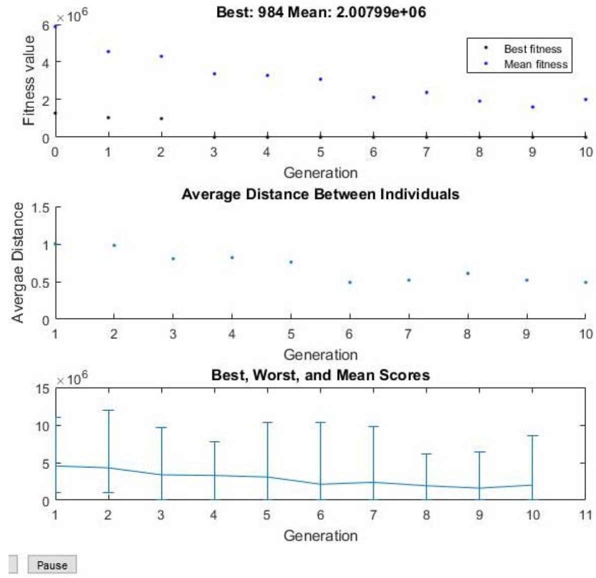


Fig. 24 (a) GA map 2 Convergence Map

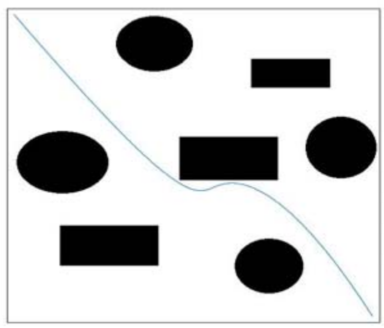


Fig. 24 (b) GA map 2 Path Computed

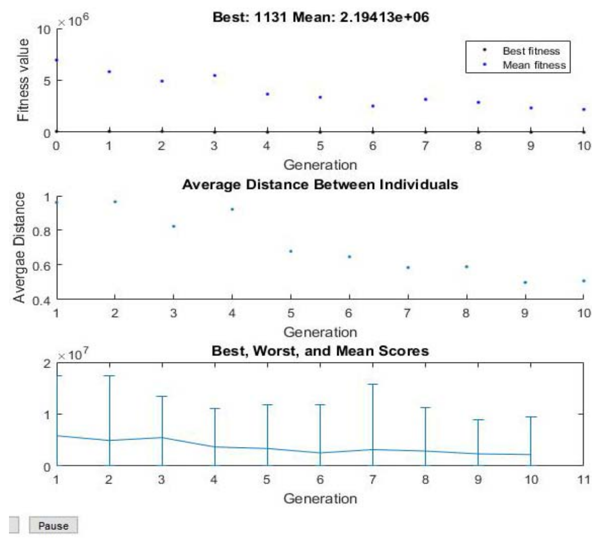


Fig. 25 (a) GA map 3 Convergence Map

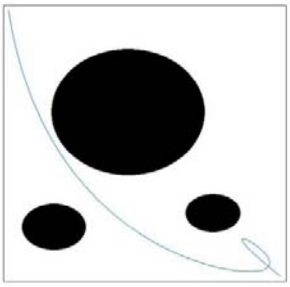


Fig. 25 (b) GA map 3 Path Computed

Readings:
GA map 1
Processing time=7.919825e+01
Path Length=938
GA map 2
Processing time=6.078662e+01
Path Length=936
GA map 3
Processing time=7.135095e+01
Path Length=1131

VII. Purepursuit

It is a path tracking algorithm which computes angular velocity command to maneuver robot from its current position to look-ahead point until the desired position is reached [11] while linear velocity component is presumed constant. In MATLAB's Robotics System Toolbox, PRM.controller is made, and waypoints are defined. If position and orientation of mobile robot is specified as input, then controller can be used to evaluate linear and angular velocities for mobile robot. In this algorithm, look-ahead point is the property of robot to demonstrate how far robot will travel along the path [11]. Experimental result of path planning and obstacle avoidance of a differential drive robot is computed using PRM algorithm in MATLAB 2018a and is shown in Figs. 26-29.

Disadvantages of Purepursuit algorithm are: [11]

- Controller lacks in following straight line paths between way-points successfully.

Robot, which is not stable at one point, requires a distance threshold to be smeared to stop robot near desired goal.

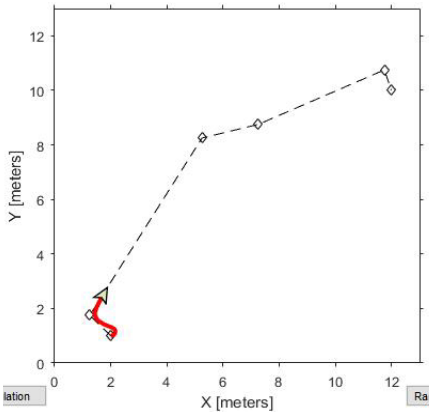


Fig. 26 (a) Purepursuit Visualize

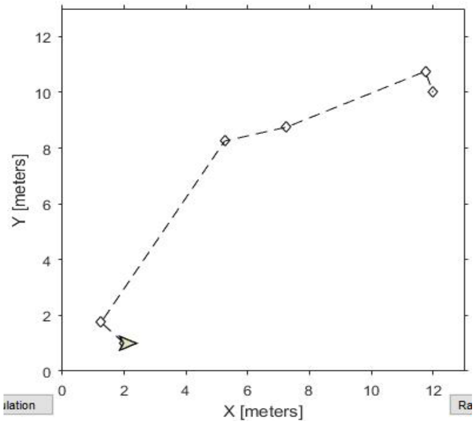
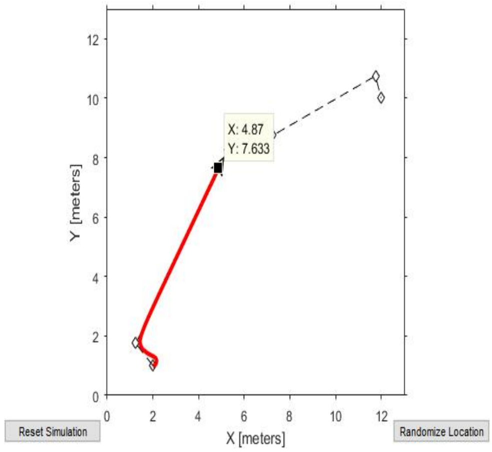
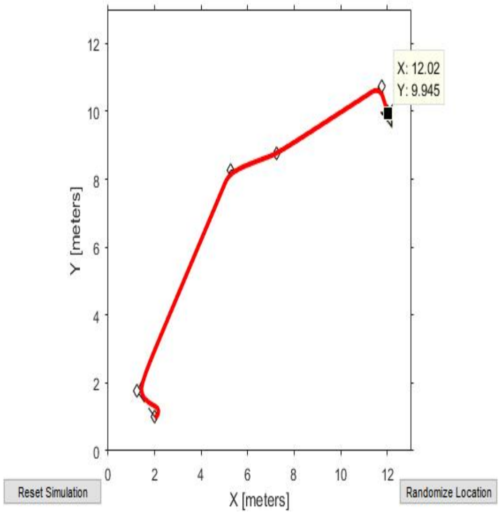


Fig. 26 (b) Robot traversing on the path desired waypoints



(a)



(b)

Fig. 27 (a) and (b) Robot traversing over desired waypoints

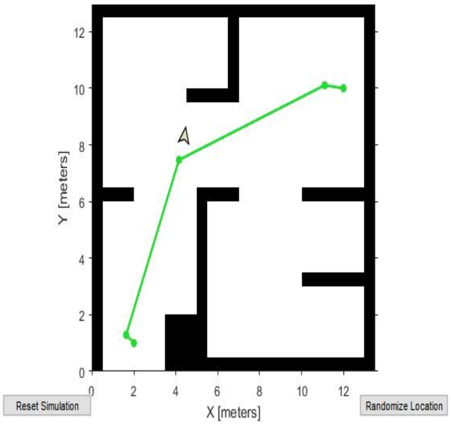


Fig. 28(a) Path computed using path planner along PRM

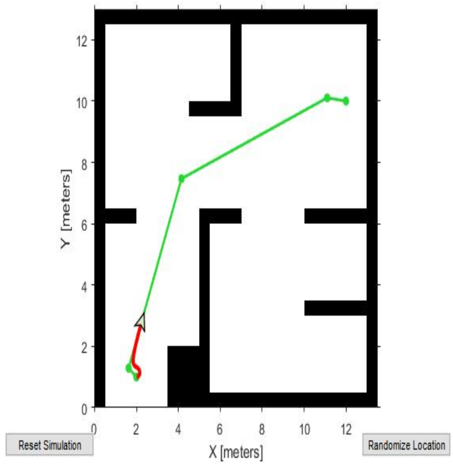
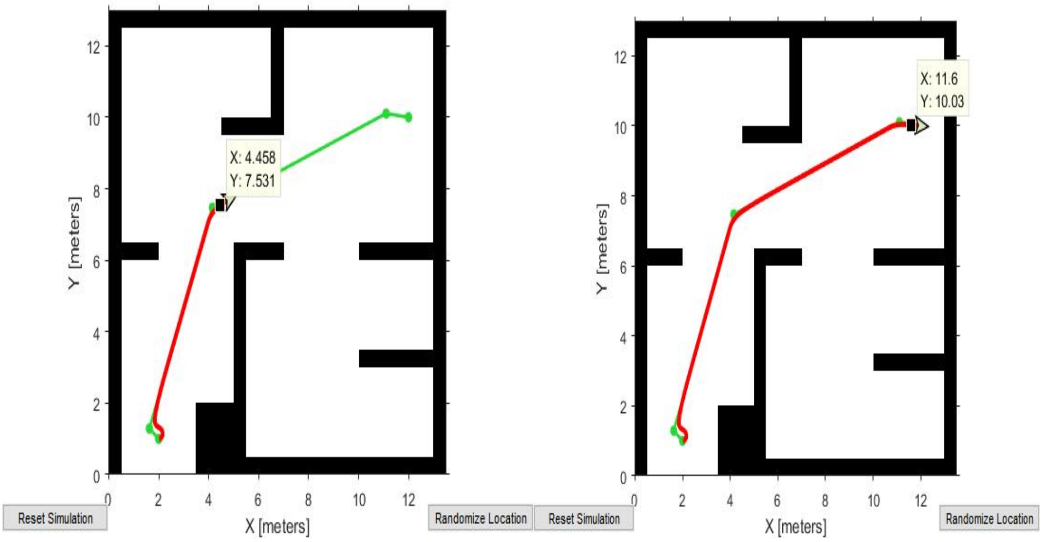


Fig. 28(b) Path following robot



(a) (b)

Fig. 29 (a) and (b) Path following robot along PRM

Readings:
Waypoints:
MaxAngularVelocity: 1
LookaheadDistance: 1
DesiredLinearVelocity: 0.1000
Waypoints for desired Path =

2.0000	1.0000
2.5280	0.8824
3.9162	6.9955
11.6860	9.8786
12.0000	10.0000

VIII. A* Algorithm

It is most efficient free-space searching algorithm for path planning and obstacle avoidance [20], [39]. Initially, graph I is taken as an input which is then explored step by step till all the nodes are evaluated to find the shortest obstacle free path [56]. With the increase in the complexity of map, computation time is increased [56], [57]. Guruji et al. [57], designed a path using this algorithm and defined position of robot (marked as 2in matrix) as shown in Fig. 30 and coded its position and orientation in a matrix which is used as an input. Possible and impossible movements are labeled as 1 and 0, respectively [57]. Higher the number of 1 is, higher are the chances of optimal path with more number of flexible turns. Experimental results for this algorithm are computed in MATLAB 2018a and are shown in Figs. 31-33.

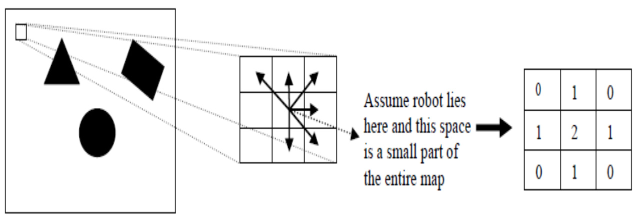


Fig. 30 Connection Matrix [57]

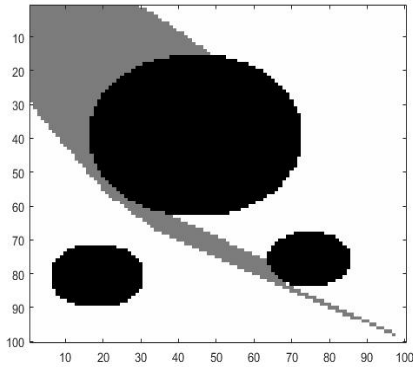


Fig. 31 (a) A* Map1 Possible path or connections

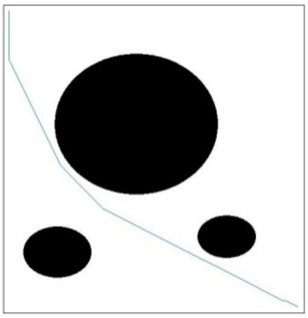


Fig. 31 (b) Final Computed path

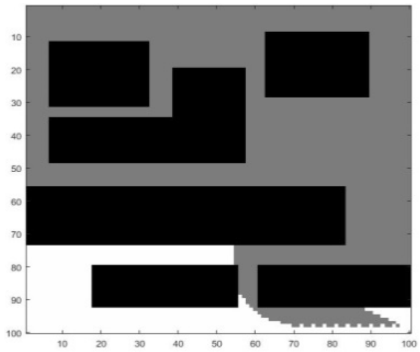


Fig. 32 (a) A* Map2 Possible path or connections

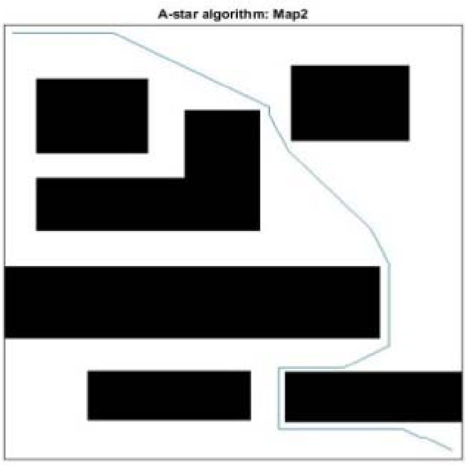


Fig. 32 (b) Final Computed path

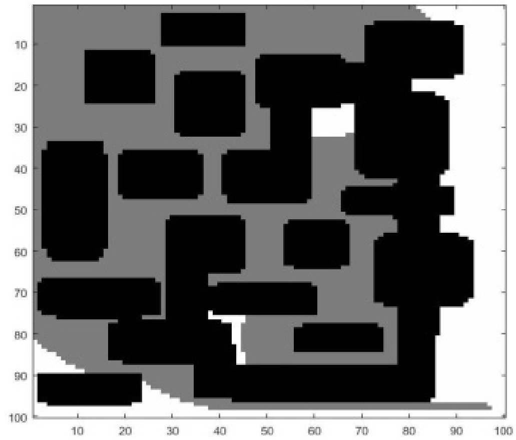


Fig. 33 (a) A* Map3 Possible path or connections

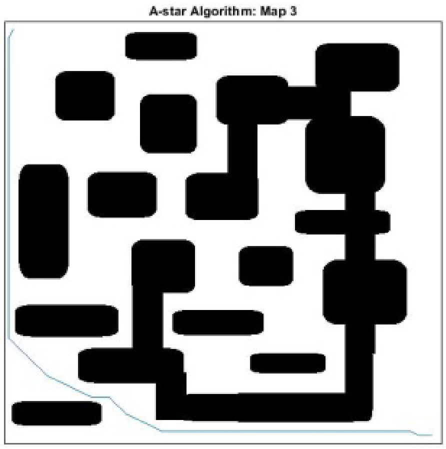


Fig 33 (b): Final Computed path

Readings:
Astar Algorithm: Map 1
Processing Time=1.003114e+02
Path Length=7.318316e+02
Astar Algorithm: Map 2
Processing Time=2.649525e+02
Path Length=1.013591e+03
Astar Algorithm: Map 3
Processing Time=1.470752e+02
Path Length=8.899076e+02

IX. Particle Swarm Optimization

It is a population-based optimization algorithm (motivated by behavior of bird flocks), where each member is called a particle having a probable solution to problem [58]-[61]. In PSO, every particle has a memory which remembers the last best position, leading it to reach desired result in less time [59], [62]. Lu and Gong [62]-[64] have recommended sensor based PSO-fuzzy type-2 model for path planning and obstacle avoidance problem in mobile robots. The path planning problem is solved using PSO in MATLAB 2018a, and the simulation results are shown in Figs. 34-36.

Steps to be followed for PSO algorithm are:

1. Initialize each particle with random numbers within the workspace [59].
2. Loop is implemented till a particle reaches stopping criteria or maximum no. of iterations.
3. The loop value of fitness parameter is decided and pbest is determined [59]
4. After analyzing all particles, global best is calculated and its position and velocity are evaluated to achieve the objective of the problem.

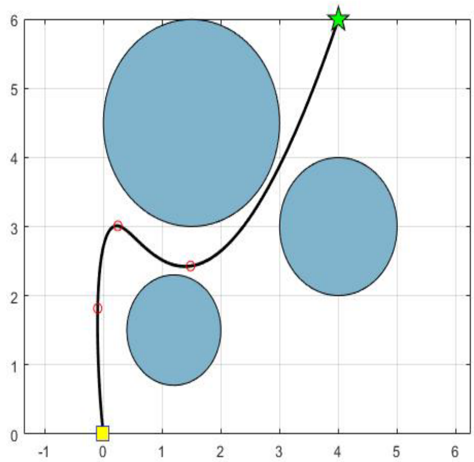


Fig. 34 (a) PSO iteration 4

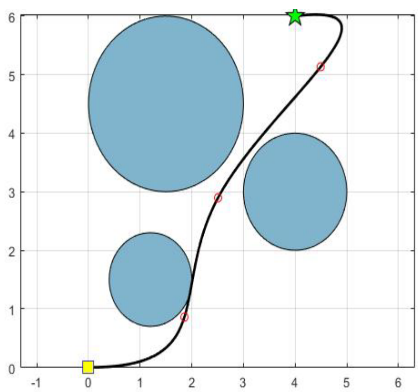


Fig. 34 (b) PSO iteration 5

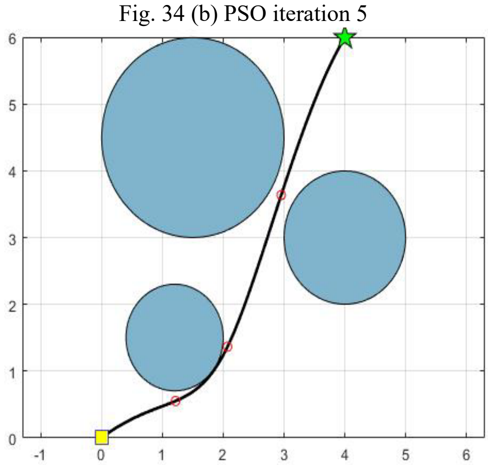


Fig. 35 (a) PSO iteration 29

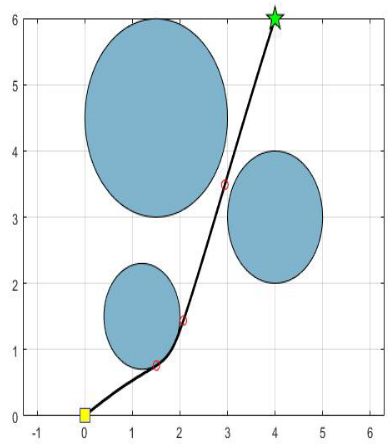


Fig. 35(b) PSO iteration 225

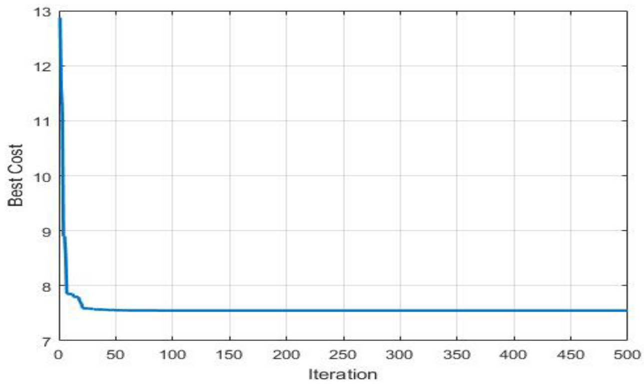


Fig. 36 Final Graph

Readings:
Iteration 4: Best Cost = 11.5143 *
Iteration 5: Best Cost = 11.5143 *
Iteration 9: Best Cost = 9.1895, Violation = 3.57e-05
Iteration 29: Best Cost = 7.6773 *
Iteration 225: Best Cost = 7.6007 *
Iteration 500: Best Cost = 7.6007 *

X. Bidirectional RRT (BRRT)

BRRT is an extension of RRT algorithm, it explores optimal collision free path using search space method using trees. Difference in RRT and BRRT is former approach which uses single trees to explore the path from source to destination point, while the later uses two trees [65]. In BRRT, one tree starts from source location and extends towards destination locations, while the second tree starts from destination location and extends towards source location. The point at which they meet is the optimal collision free path achieved [66]. Concept is simulated and computed for path planning and obstacle avoidance problem in mobile robot using MATLAB 2018a platform and is shown in Figs. 37-39 for different paths passing through different obstacles.

Readings:
BRRT Algorithm: Map 1
Processing time=2.210911e+01

Path Length=9.2720082490e+02
BRRT Algorithm: Map 2
Processing time=1.962829e+01
Path Length=1.1838272008e+03
BRRT Algorithm: Map 3
Processing time=2.984834e+01
Path Length=1.168694e+03

Fig. 37 (a) BRRT Map1 Roadmap

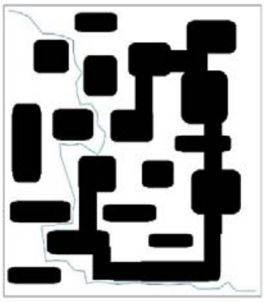


Fig. 37 (b) Final computed path

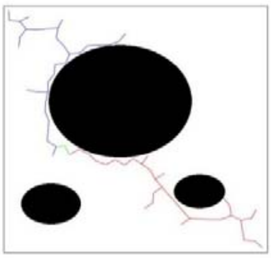


Fig. 38 (a) BRRTMap2 Roadmap

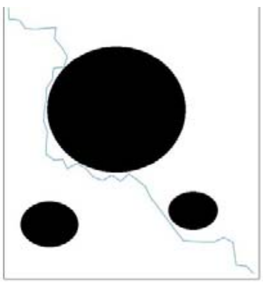


Fig. 38 (b) Final computed path

Fig. 39 (a) BRRT Map3 Roadmap

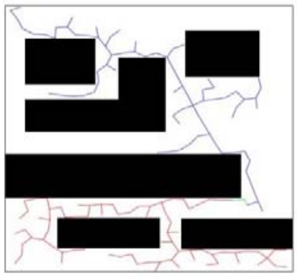


Fig. 39 (b) Final computed path

XI. Vector Field Histogram

Borenstein and Koren [38] developed VFH which is a real-time path planning and obstacle avoidance algorithm permitting obstacle detection and obstacle avoidance simultaneously maneuvering towards goal location [5], [18], [67]. Steps to be followed for VFH algorithm are [33]:

1. Constructing a 2D Cartesian histogram grid signifying obstacles
2. From the 2D grid, 1D polar histogram is built considering active window around robot.
3. For optimization procedure, 1D polar histogram is used to evaluate steering angle and velocity controls.

Advantages of VFH algorithm are:

- Poor sensor measurements and its impact are minimized.
- Instability in maneuvering is eliminated because there are slight variations in sonar readings.
- The robot never gets trapped in its local minima as unlike APF there is no repulsive or attractive forces.

Disadvantages of AFH algorithm are:

- This algorithm might lead robot away from goal position.

For evaluating steering angle, AFH requires four input parameters such as robot radius, safety distance, minimum turning radius, and distance limits [67]. Path planning and obstacle avoidance problem for mobile robots in computed using VFH algorithm and results are simulated using MATLAB 2018a as shown in Fig. 40.

Readings: steering Direction = -0.8014

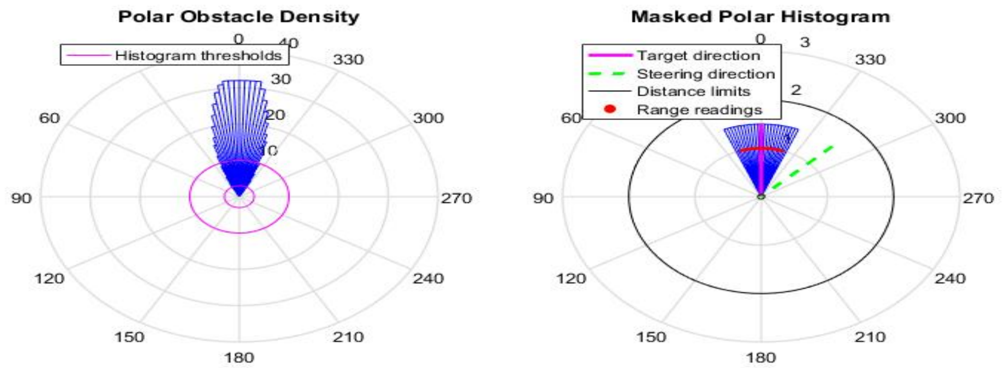


Fig. 40 Path planning and obstacle avoidance using VFH algorithm based on range sensor data

XII. Modified Local Path Planning Algorithm

This algorithm is evaluated for path planning and obstacle avoidance problem in mobile robots in unknown environment [68]. This algorithm permits mobile robot to maneuver along static obstacles to obtain a collision free path till it reaches goal location. Mobile robot traverses from source to target point through unknown environment avoiding obstacles and hurdles coming along its way [68]. The algorithm is implemented and tested through simulation in MATLAB 2018a. In this algorithm, every obstacle is considered as a charge particle having repulsive potential, while the goal is considered as a charge particle having attractive potential. MATLAB GUI program is created to accomplish this problem. Fig. 42 shows GUI interface, this algorithm is simulated for three maps, and the final map is obtained and is shown in Figs. 43-46.

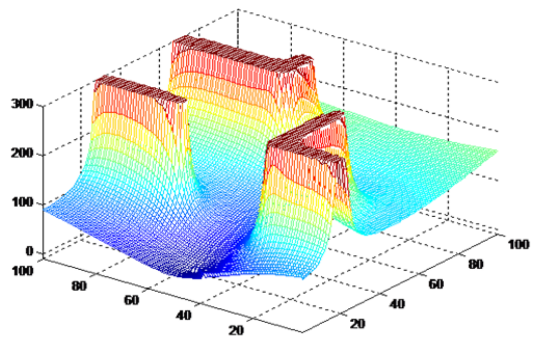


Fig. 41 Path planning of a mobile robot [68]

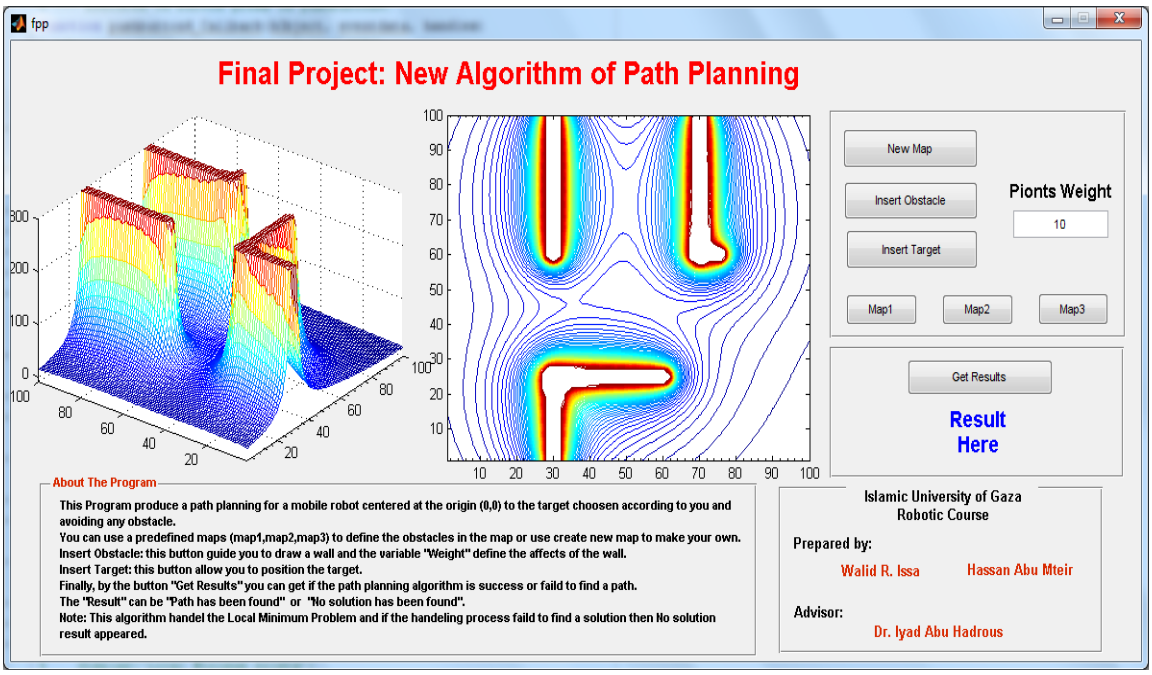


Fig. 42 GUI Interface [68]

World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:12, No:6, 2018

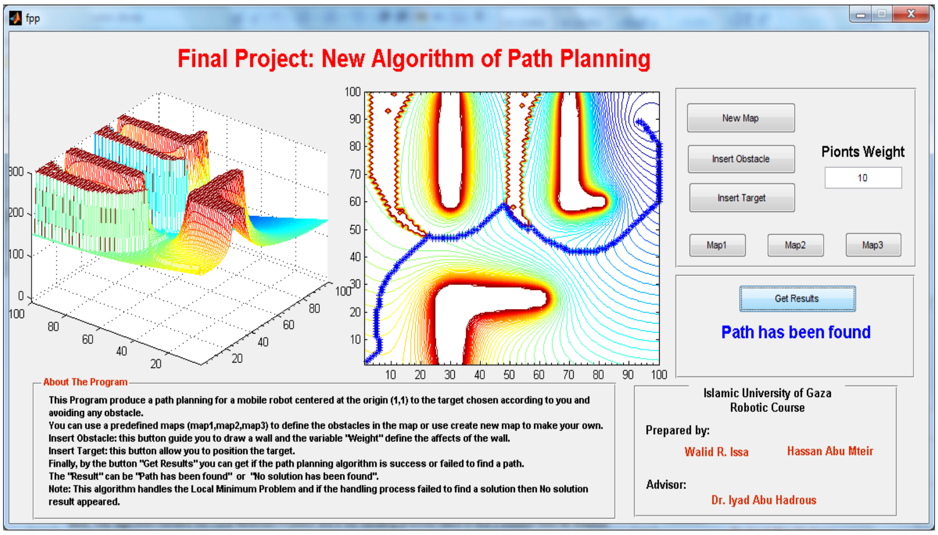


Fig. 43 Map 1 Computed path

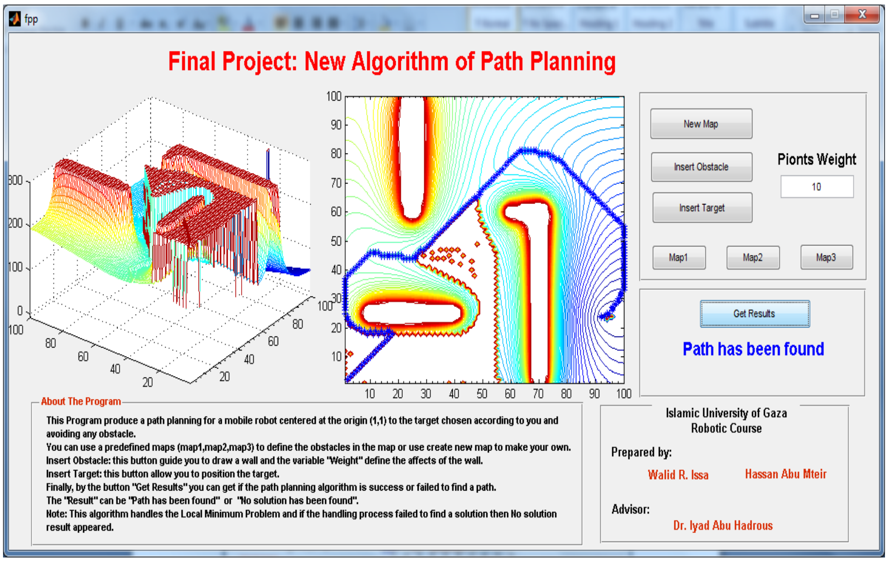


Fig. 44 Map 2 Computed path

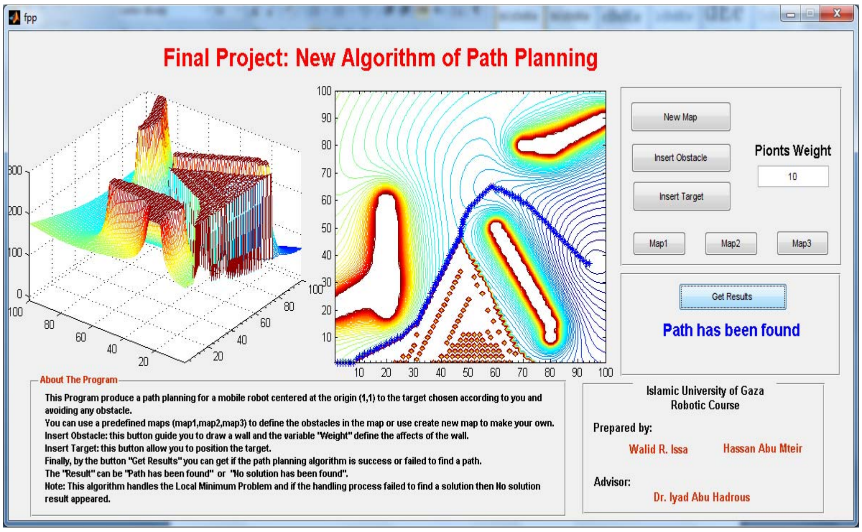


Fig. 45 Map 3 Computed path

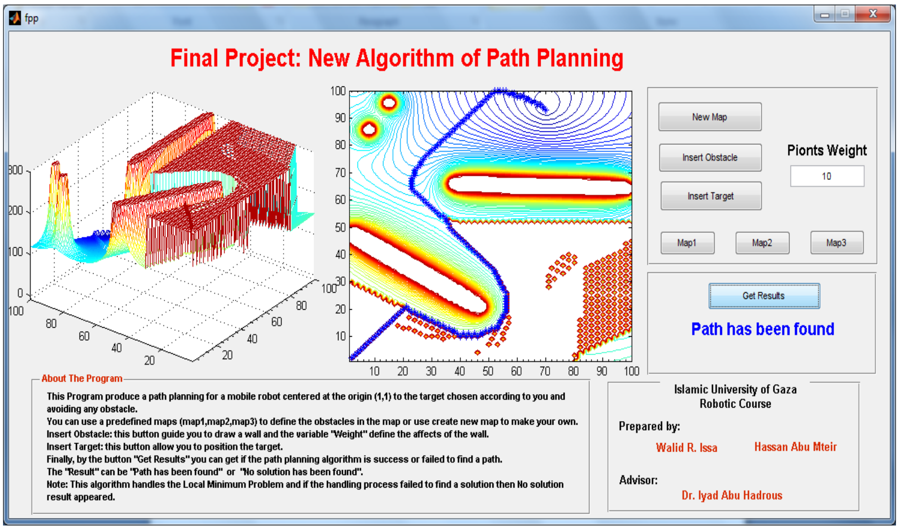


Fig. 46 Final computed path

XIII. Conclusion

An overview of path planning and obstacle avoidance algorithms for AMR, their strengths and weakness are presented and discussed. This review paper discusses the robot path planning and driving systems designed by various researchers, and their simulation results are also shown in this paper giving an insight into the positive and negative points of every algorithm while comparing them. From the literature, it can be concluded that the algorithm should be generic in respect to different maps and it should be capable of resulting a collision free path in less computation time thereby saving cost and energy. From the literature, it can be concluded that there are many different path planning and obstacle avoidance algorithms employed to successfully maneuver through the obstacle free path. The objective of an algorithm should be to use sensor data to result into superior performance in achieving autonomous movement. Problems of classical techniques are such that they are computationally time taking and expensive, requiring large memory space and result into paths which are tangent to the obstacles.

References

[1] A. Yufka and O. Parlaktuna, "Performance Comparison of Bug Algorithms for Mobile Robots", 5th International Advanced Technologies Symposium, pp. 1-5, 2009

[2] M.S. Ganeshmurthy, and G.R. Suresh, "Path planning algorithm for autonomous mobile robot in dynamic environment", *International Conference on Signal Processing, Communication and Networking*, vol. 3, pp. 1-6, 2015

[3] K. M. Hasan, A. Al-Nahid, and K.J. Reza, "Path planning algorithm development for autonomous vacuum cleaner robots," *In Proceedings International Conference on Informatics, Electronics & Vision*, pp. 1-6, 2014

[4] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving", *AAAI Workshop - Technical Report,* 2008.

[5] L. Huang, "Velocity Planning for a Mobile Robot to Track a Moving Target - A Potential Field Approach", *Robotics and Autonomous Systems*, vol. 57, pp. 55-63, 2009

[6] Y.W. Chen, and W.Y. Chiu, "Optimal Robot Path Planning System by Using a Neural Network-Based Approach", *Proceedings of IEEE International Automatic Control Conference (CACS)*, pp. 85-90, 2015

[7] E.H. Shan, B. Dai, J. Song, and Z.P. Sun, "A dynamic RRT path planning algorithm based on B-spline," In *Proc. International Symposium on Computational Intelligence and Design*, pp. 25-29, 2009

[8] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," *in Proc. IEEE International Conf. on Robotics and Automation*, pp. 995-1001, 2000

[9] A. Stentz, "Optimal and efficient path planning for partially-known environments," In Proc. IEEE International Conf. on Robotics and Automation, pp. 3310-3317, 1994

[10] Z.J. Du, D.K. Qu, F. Xu, and D.G. Xu, "A hybrid approach for mobile robot path planning in dynamic environments," *In Proc. IEEE International Conf. on Robotics and Biomimetics*, pp. 1058-1063, 2012

[11] M. Samuel, M. Hussein and M.B. Mohamad, "A Review of some Pure-Pursuit based Path Tracking Techniques for Control of Autonomous Vehicle", *International Journal of Computer Applications*, vol. 135, no. 1, pp. 35- 38, 2016

[12] C. Wong, H. Wang, and S. Li, "PSO-based Motion Fuzzy Controller Design for Mobile Robots", *International Journal of Fuzzy Systems*, vol. 10, issue no. 1, pp. 284-292, 2008

[13] H.Y. Chung, C.C. Hou and S.C. Liu, "Automatic Navigation of a Wheeled Mobile Robot using Particle Swarm Optimization and Fuzzy Control", *IEEE International Symposium on Industrial Electronics, Taiwan*, pp. 1-6, 2013

[14] Y. Zhang, D.W. Gong, J.H. Zhang, "Robot Path Planning in Uncertain Environment using Multi-Objective Particle Swarm Optimization", *Elsevier Neurocomputing*, vol. 103, pp. 172-185, 2013

[15] L.E. Zarate, M. Becker, B.D.M. Garrido, and H.S.C. Rocha, "An artificial neural network structure able to obstacle avoidance behaviour used in mobile robots". *IEEE Annual Conference of the Industrial Electronics Society*, vol. 28, pp. 2457-246, 2002

[16] S.X. Yang and C. Luo, "A neural network approach to complete coverage path planning", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 34, issue 1, pp. 718-724, 2004

[17] K. Sedighi, K. Ashenayi, R. Wainwright and H. Tai, "Autonomous local path planning for a mobile robot using a genetic algorithm", *IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1338–1345, 2004

[18] P.O. Petterssonand P. Doherty, "Probabilistic roadmap based path planning for an autonomous unmanned helicopter'", *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 17, no. 4, pp. 395-405, 2005

[19] H. Kristo, "Path Planning and Learning Strategies for Mobile Robots in Dynamic Partially Unknown Environments'", *Faculty of Mathematics and Computer Science*, pp. 19-32, 2006

[20] R. Kala, A. Shukla, R. Tiwari, S. Roongta and R.R. Janghel, "Mobile Robot Navigation Control in Moving Obstacle Environment using Genetic Algorithm, Artificial Neural Networks and A* Algorithm", *Proceedings of the IEEE World Congress on Computer Science and Information Engineering*, pp. 705-713, 2009

[21] L. Yang, J. Qi, D. Song, J. Xiao, J. Han and Y. Xia, "Survey of Robot 3D Path Planning Algorithms", *Journal of Control Science and*

Engineering, vol. 2016, Article ID 7426913, 22 pages, 2016

[22] D. Nguyen and T. Nguyen, "A solution of obstacle collision avoidance for robotic fish based on fuzzy systems" IEEE *International Conference on Robotics and Biomimetics (ROBIO)*, pp. 7-11 2011

[23] I. Ullah, F. Ullah, Q. Ullah and S. Shin, "Integrated tracking and accident avoidance system for mobile robots", *International Journal of Control, Automation and Systems*, pp. 1253-1265, 2013

[24] J. A. Oroko and G. N. Nyakoe, "Obstacle Avoidance and Path Planning Schemes for Autonomous Navigation of a Mobile Robot: A Review", *Proceedings of Mechanical Engineering Conference on Sustainable Research and Innovation*, Vol. 4, pp. 314-318, 2012

[25] L. Zeng and G.M. Bone, "Mobile Robot Collision Avoidance in Human Environments", International Journal of Advanced Robotic Systems, vol. 10, pp. 1-14, 2013,

[26] A. Mohammadi, M. Rahimi, and A.A. Suratgar, "A New Path Planning and Obstacle Avoidance Algorithm in Dynamic Environment", *IEEE Iranian Conference on Electrical Engineering (ICEE 2014)*, vol. 22, pp. 1301-1306, 2014

[27] K.H. Su, F.L. Lian and C.Y. Yang, "Navigation design with SVM path planning and fuzzy-based path tracking for wheeled agent", *International Conference onFuzzy Theory and it's Applications (iFUZZY)*, pp. 273,278, 2012

[28] M.Y. Ju and C.W. Cheng, "Smooth path planning using genetic algorithms", *World Congress onIntelligent Control and Automation (WCICA),* vol. 9, pp.1103- 1107, 2011

[29] J.A. Goldman, "Path planning problems and solutions", *Proceedings of the IEEE Aerospace and Electronics Conference*, pp. 105-108, 1994

[30] H.C. Huang "FPGA-Based Parallel Metaheuristic PSO Algorithm and its Application to Global Path Planning for Autonomous Robot Navigation", *Journal of Intelligent & Robotic Systems*, vol. 76, issue no. (3-4), pp. 475- 488, 2014

31] K. Chu, M. Lee and M. Sunwoo, "Local Path Planning for Off-Road Autonomous Driving With Avoidance of Static Obstacles", *IEEE Transactions on Intelligent Transportation Systems,* vol.13, no.4, pp.1599-1616, 2012

32] M. Aizzat, H. Zamzuri, R. Mamat and S. Amri, "A Path Tracking Algorithm Using Future Prediction Control with Spike Detection for an Autonomous Vehicle Robot", *International Journal of Advanced Robotics Systems*, vol. 10, pp. 1-10, 2013.

33] Y. Zhang, N. Fattahi and W. Li, "Probabilistic roadmap with self-learning for path planning of a mobile robot in a dynamic and unstructured environment", *IEEE International Conference on Mechatronics and Automation*, pp. 1074-1079, 2013

34] L. Gang, and J. Wang, "PRM path planning optimization algorithm research", *Wseas Transactions on Systems and control*, vol. 11, pp. 81-86, 2016

35] M. Zohaib, M. Pasha, R.A. Riaz, N. Javaid, M. Ilahi and R.D. Khan, "Control Strategies for Mobile Robot With Obstacle Avoidance", *Journal of basic and Applied Scientific Research*, Vol. 3, issue no. 4, pp. 1027-1036, 2013

36] P.O. Petterssonand P. Doherty, "Probabilistic Roadmap Based Path Planning for an Autonomous Unmanned Aerial Vehicle", *In Proceedings of the Workshop on Connecting Planning and Theory with Practice,* pp. 1-8, 2004

37] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *The International Journal of Robotics Research*, pp. 90-98, 1998

38] J. Borenstein and Y. Koren, "The vector field histogram – fast obstacle avoidance for mobile robots", *IEEE Journal of Robotics and Automation,* vol. 7, issue no. 3, pp. 278–288, 1991.

[39] M.C. Lee, and M.G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept", *Proceedings in IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, vol. 2, pp. 735-740, 2003

[40] L. Xie, H. Chen, and G. Xie, "Artificial Potential Field Based Path Planning for Mobile Robots Using Virtual Water-Flow Method", *Advanced Intelligent Computing Theories and Applications, with Aspects of Contemporary Intelligent Computing Techniques*, vol. 2, pp. 588-595, 2007

[41] A.A. Ahmed, T.Y. Abdalla, and A.A. Abed, "Path Planning of Mobile Robot by using Modified Optimized Potential Field Method", *International Journal of Computer Applications*, vol. 113, no. 4, pp. 6-10, 2015

[42] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning", *International Journal of Robotics Research*, vol. 30, issue no. 7, pp. 846–894, 2011

[43] H. Yu, R. Sharma, R.W. Beard, and C.N. Taylor, "Observability-based local path planning and obstacle avoidance using bearing-only measurements", *Robotics and Autonomous Systems*, vol. 61, pp. 1392–1405, 2013

[44] G. Zhang, S. Ferrari and M. Qian, "An Information Roadmap Method for Robotic Sensor Path Planning", *Journal of Intelligent and Robotic Systems*, vol. 56, Issue no. 1–2, pp. 69–98, 2009

[45] C. Fragkopoulos and A. Graeser, "A RRT based path planning algorithm for Rehabilitation robots", *41st International Symposium on Robotics and ROBOTIK, vol. 6*, pp. 1-8, 2010

[46] L. Garrote, C. Premebida, M. Silva and U. Nunes, "An RRT-based navigation approach for mobile robots and automated vehicles", *In: 2014 IEEE International Conference on Industrial Informatics*, pp. 326–331, 2014

[47] R. Seif and M.A. Oskoei, "Mobile Robot Path Planning by RRT* in Dynamic Environments", *International journal of intelligent systems and applications,* vol. 5, pp. 24-30, 2015

[48] M. Wang, and J.N.K. Liu, (2005) "Fuzzy logic based robot path planning in unknown environment", *International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 813-818, 2005

[49] N. Kumar, M. Takács and Z. Vámossy, "Robot navigation in unknown environment using fuzzy logic", *IEEE International Symposium on Applied Machine Intelligence and Informatics*, vol. 15, pp. 279-284, 2017

[50] W.G. Han, S.M. Baek, and T.Y. Kuc, "GA based online path planning of mobile robots playing soccer games", *Proceedings of the 40th Midwest Symposium* on *Circuits and Systems*, vol.1, pp. 2-15 , 1997

[51] A. Elshamli, H. A. Abdullah and S. Areibi, "Genetic algorithm for dynamic path planning", *IEEE In: Proc. Canadian Conf. Elect. andComput. Eng.*, Vol.2, pp: 677-680, 2004

[52] C. E. Thomas, M. A. C. Pacheco, M.M. and B.R. Vellasco, "Mobile Robot Path Planning Using Genetic Algorithms", *In foundations and tools for neural modeling,* vol. 1606/1999, Springer, pp. 671- 679, 1999

[53] J. Brank, "Evolutionary approaches to dynamic optimization problems-introduction and recent trends," In Proc. *GECCO Workshop on Evol. Algorithms for Dynamic Optimization Problems*, pp. 2-4, 2003

[54] I.A. Taharwa, A.Sheta and M.A. Weshah, "A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment", *Journal of Computer Science*, vol. 4, issue 4, pp. 341-344, 2008

[55] J. Lu and D. Yang, "Path planning based on double-layer genetic algorithm," *In 3rd International Conference on Natural Computation*, Vol. 4, pp: 357-361, 2007

[56] Z. Weiteng, H. Baoming, L. Dewei, and Z. Bin, "Improved Reversely A star Path Search Algorithm based on the Comparison in Valuation of Shared Neighbour Nodes", *Fourth International Conference on Intelligent Control and Information Processing*, pp. 9 – 16, 2013

[57] A.K. Guruji, H. Agarwal, D.K. Parsediya, and A. Madhav, A. "Time-Efficient A* Algorithm for Robot Path Planning", *International Conference on Innovations in Automation and Mechatronics Engineering,* vol. 3, pp. 144 – 149, 2016

[58] Q. Zhang and S. Li, "A Global Path Planning Approach based on Particle Swarm Optimization for a Mobile Robot", *International Conference on Robotics, Control & Manufacturing Technology World Scientific and Engineering Academy and Society (WSEAS)*, pp. 263-267, 2007

[59] E.V. Kumar, M. Aneja and D. Deodhare, "Solving A Path Planning Problem In A Partially Known Environment Using A Swaim Algorithm", *IEEE International Symposium on Measurements and Control in Robotics,* pp. 15-26, 2008

[60] P. Raja and S. Pugazhenthi, "Path Planning for Mobile Robots in Dynamic Environments using Particle Swarm Optimization", *IEEE International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 401-405, 2009

[61] Masehian and D. Sedighizadeh, " A Multi-Objective PSO-based Algorithm for Robot Path Planning", *IEEE International Conference on Industrial Technology (ICIT)*, pp. 465-470, 2010

[62] L. Lu and D. Gong, "Robot Path Planning in Unknown Environments using Particle Swarm Optimization". *IEEE International Conference on Natural Computation (ICNC)*, pp. 422-426, 2008

[63] Q. Li, Y. Tang, L. Wang, C. Zhang and Y. Yin, "A Specialized Particle Swarm Optimization for Global Path Planning of Mobile Robots",. *IEEE International Workshop on Advanced Computational Intelligence (IWACI)*, pp. 271-276, 2010

[64] Z.T. AllawiandT.Y.A. Abdalla, "PSO-Optimized Type-2 Fuzzy Logic

Controller for Navigation of Multiple Mobile Robots", *IEEE International Conference on Methods and Models in Automation and Robotics*, pp. 33-39, 2015

[65] M. Jordan, and A. Perez, "Optimal bidirectional rapidly-exploring random trees", *Technical Report MIT-CSAIL-TR,* pp. 97-101, 2013

[66] A.H. Qureshi and Y. Ayaz, "Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments", *Robotics and Autonomous Systems*, vol. 68, pp. 1-11, 2015

[67] J.S. Kumar, and R. Kaleeswari, "Implementation of Vector Field Histogram based obstacle avoidance wheeled robot", *International Conference on Green Engineering and Technologies,* pp. 1-6, 2016

[68] O. Hachour, "Path planning of Autonomous Mobile robot", *International Journal Of Systems Applications, Engineering & Development*, vol. 2, no. 4, pp. 178-190, 2008

Ms. Divya Agarwal was born in Delhi, India in 1990. She is Research Scholar in Robotics and Automation Department of Guru Gobind Singh Indraprastha University, Delhi. She did her B. Tech in Electronics and Communication Engineering in 2012. She completed her M. Tech in Robotics and Automation Engineering from Indira Gandhi Delhi Technical University, Delhi, India. Her current research area includes Robotics & Automation, Facility Layout Planning Problem, Multiple Criteria Decision Making Approaches, Fuzzy Approach, Automated Guided Vehicle, Attribute and Criteria Selection Problem, and Path Planning Problem. She has 2 year of teaching experience and 6 months of industry experience. She has 7 publications in International journals and conferences of repute.

Dr. Pushpendra S. Bharti is working as Associate Professor in University School of Information, Communication and Technology, Guru Gobind Singh Indraprastha University, New Delhi. He received B.E. (Mechanical Engineering) from Dayalbagh Educational Institute, Agra, M.E. (Production Engineering) from Motilal Nehru National Institute of Technology, Allahabad. He received Silver Medal from AllahabadUniversity. He received his Ph.D. from GGSIPU, New Delhi. He has written more than 25 research papers in various journals and conferences. His research interests include non-conventional manufacturing, robotics and manufacturing automation.