Comparison of Temporal Difference Learning Algorithm and Dijkstra's Algorithm for Robotic Path Planning

Devika S.Nair, Dr. Supriya P
Department of Electrical and Electronics Engineering
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham, India
devikaprayag@gmail.com, P_supriya@cb.amrita.edu

***Abstract*— Robotic Navigation is a crucial issue for any robotic based automation. In order to implement a seamless robotic navigation, path planning is a key challenge in robotic navigation. Several algorithms exist in literature on robotic path planning of static and dynamic obstacles. In this work, an approach called Modified Temporal Difference Learning for path planning and obstacle avoidance is proposed for static obstacles. The algorithm is developed in MATLAB software and path planning is implemented in a 4 ×4 grid environment. A GUI for the same is created, which access the user inputs like obstacle number, positions and type. The developed algorithm is compared with the conventional path planning Dijkstra's algorithm in the same environment. It is observed that computational complexity is less in the proposed approach compared to the conventional method.**

Keywords—Modified Temporal Difference Learning, Dijkstra's algorithm, Path planning, Obstacle avoidance.

I. Introduction

The world is moving towards automation and Robots are replacing humans in every area. Development of artificial intelligence and soft computing techniques play a key role in automating the human done activities. For robots to replace humans, they should be able to move autonomously. So path planning and obstacle avoidance are vital part in robotics. There are several existing path planning algorithms like Dijkstra's, Astar, Bio inspired planning algorithms… etc. In this paper, Temporal difference learning algorithm which is a known prediction-based machine learning approach, is enhanced and used for implementing path planning and obstacle avoidance in a known environment with static obstacles. Modified TD Learning based path planning method is compared with the Dijkstra's algorithm, which is a widely used approach for finding the short distance path from start to target node in a weighted graph. The entire paper divided into different sections. The section one deals with the introduction to robotic path planning and section two deals with the related works in path planning. The actual problem is formulated in section three and simulation results are shown under section four. The final section concerns with the conclusions and future scope.

II. Related work

Path planning with obstacle avoidance is a commonly explored domain in robotics. In order to perform path planning with obstacle avoidance numerous algorithms are formulated to attain the optimal path. Many soft computing methods in the domain of artificial intelligence like neural networks, ant colony methods, fuzzy logic, genetic algorithm etc. are used in path planning. Fuzzy logic approaches are attempted [1] to avoid obstacles in an unknown environment by considering the sensor inputs of distance to the nearest obstacle from robot. Here the speed of robot is controlled by applying fuzzy rules. Ant colony optimization algorithm [2] is developed and implemented on a mobile robot that can trace the optimum path from start point to destination point without collisions. Remodeled A* algorithm [3] based on nodal weight upgradation method is used for robotic path planning and avoidance of obstacles which are stationary in nature, optimized the distance to be travelled from source to target.

Q learning is used to choose an optimal action-value pair that finally provides the expected utility of executing a particular action in a specified current state and followed by the optimal policy. Reinforcement learning and TD Learning are two categories of Q-learning. Reinforcement learning is a kind of machine learning method that assists the agent to carry out an action in an environment. TD learning is basically a prediction based machine learning type. In paper [4], a path is planned using reinforcement learning and implemented on a small two wheeled robot. This method provided an easier solution for a robot with non-holonomic constraints. BharadwajSrinivasan attempted to avail robot navigation in a domain with partial observability [5]. The domain was a grid-world with intersecting corridors, where the agent learns an optimal policy for navigation by making use of a hierarchical memory-based learning algorithm. The Partially Observable Markov Decision Process (POMDP) is used to model the problem and the State-Action-Reward-State-Action (SARSA) algorithm is implemented to achieve the solution. The SARSA algorithm utilizes the Temporal Difference learning method. The short-term memory is used by the agent to extract over very minute details to enable it to scale up to the wide partially observable

978-1-5386-2842-3/18/$31.00 ©2018 IEEE

domains. A real robot ‘Koolio’ [6] is implemented which is an autonomous delivery robot that has the ability to select the optimal policy of Reinforcement learning or Q- learning to reach its destination using sensor inputs. An efficient learning rate rule is suggested by Rongkuan Tang and Hongliang Yuan [7] to implement reinforcement learning. An error-sensitive learning rate mechanism is tried out for Q-learning algorithm to obtain improved mitigation as well as to quicker the process of learning. This method is simulated and experimented in an indoor static grid world environment for robotic navigation purpose. In paper [8], an adaptive fire fly algorithm for non-holonomic motion planning of a car like system was proposed. Such a technique concludes that adaptation of Q-learning in Firefly Algorithm improved the corresponding algorithm and is found more effective in terms of runtime, efficiency and accuracy. Adaptive Firefly Algorithm is analysed by combining the positive facts of Firefly Algorithm for having global exploration and Temporal Difference Q Learning for implementing local tuning of the parameter called absorption coefficient. AbdulrahmanAltahhan [9] proposed a goal aware robotic navigation technique which utilizes gradient and conjugate gradient Temporal Difference approaches to obtain faster learning. A home aware robot navigation is realized using fast learning variable lambda TD method. The thresholding technique using von Mises distribution is utilized to decrease the sensitivity of the orientation of goal and this model was able to achieve a convergence in a very less amount of episodes of twenty. Temporal Difference learning incorporating Fuzzy state [10] is employed for navigation of robot in a multiple obstacle environment. Here the Artificial Potential Field (APF) is used to implement global optimal path planning. The APF is calculated by using TD learning method and they suggested that such a learned APF will be able to trace a global optimal path capable of avoiding obstacles from different source point. Incorporation of fuzzy state enhanced the learning performances. The article [11] concerns to forecast the upcoming behavior with a not fully known system by learning the past experiences of it. Richard S.Sutton said that, for most of the problem of predictions in world, temporal difference approaches consume less memory and less peak computation compared to the commonly used practices and they will be able to output predictions which are more accurate. They suggested that application of temporal difference learning on problems which are currently solved by supervised learning will be advantageous since most of them are prediction cases. The initial results in the area of temporal difference approach were introduced and provided along with incremental learning methodology intended for solving the issues of prediction. The output of the commonly used predictive learning approaches are based on the error measurement between actual and predicted values whereas error measurement between temporally successive predictions is utilized by TD approaches. In such systems, whenever a change occurs in prediction over the time, learning happens. The two most benefits showcased by temporal difference approaches over common prediction learning approaches are they are incremental in nature so that its more easier to calculate and they utilize their experiences more effectively in order to converge quicker and output predictions which are more reliable and better. An algorithm called least-squares temporal difference [12], which is based on Extreme learning machin, an algorithm used for training single hidden layer feed forward network is proposed. The new method used global approximator which resulted in more scalability and has the ability to deal problems with high dimension. In this work, path planning is attempted with varying number of obstacles and at varied positions and the behavior of the algorithm is analysed with a conventional path planning algorithm like *Dijkstara’s* algorithm.

III. Problem formulation

Machine learning is used very often to train the mobile robots about its environment. Supervised learning and Reinforcement learning are the two subcategories of machine learning. The supervised learning methods teach a system how to behave in an environment whereas Reinforcement learning teaches a system to take suitable action based on punishments and reward concepts.

Temporal difference learning is a subset of Reinforcement learning method. TD learning shows the way to predict a value which has a dependency on future quantity. It is an example of unsupervised technique. Here expected quantity revealed at the end of series of states will be predicted by a learning agent, which is trained to learn. In TD learning, the predicted value of present time is updated and used to predict the same quantity in upcoming steps. TD learning is a combination of Monte Carlo and dynamic programming approaches. TD learning is said to be similar to Monte because it carries out the learning of the environment by implementing a sampling process using some known policy, and to dynamic programming methods since its present estimate is approximated using learned estimates obtained in past experiences. It will look farther into a future of a move and update Q function only after looking farther ahead and speeds up the learning process. Fig.1. shows the structure of a Modified TD Learning based path planning. TD Learning algorithm calculates the Q factor value for each possible movement from each state for the given environment and inputs like starting point, destination point and obstacle position. A Reward function assigns reward for each possible action from current state based on the priority of movement and obstacle positions. The action selector will provide the action to be carried next from the current state by analyzing both Q-factor and Reward function.

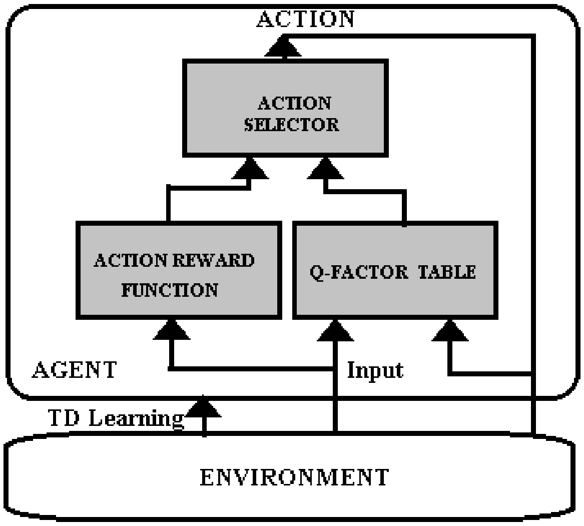


Fig.1. Structure of Modified TD Learning based path planning

A. *Modified Temporal Difference Learning (MTDL) Algorithm*

Some of the major assumptions in applying modified TD learning to path planning are as follows:

1. A 4 x 4 grid with 16 cells is considered for path planning environment.
2. Obstacles are located at the centre of each cell.
3. Target position is altering in the environment.
4. Robot is forbidden from moving backwards.
5. Utility Factor is zero for a cell housing the obstacle.

The algorithm for the modified TD learning is described below:

1) Start the trajectory from source point.

2) Calculate the utility factor for each cell in a 4 x 4 grid using Q Learning equation

$$Q(S,a) = r(s,a) + \gamma \max\left(Q(S',a')\right) \tag{1}$$

Where *Q(S, a)* is the estimated utility function, *r(s, a)* is the immediate reward, ϒ is the relative value of delayed versus immediate reward, is utility factor for the resulting state *S'* after the action *a*.

3) A reward function is assigned for all possible movements. First preference assigned to diagonally forward movement, next priority given to vertically forward movement and last precedence assigned to horizontally forward.

4) Add the utility factor and corresponding reward function to evaluate the modified utility factor, which will give the path for a particular movement from the current position.

$$Q(S,a)_{modified} = R(a') + r(s,a) + \gamma \max\left(Q(S',a')\right) \tag{2}$$

Where *R(a')* is the reward function assigned for action *a'*.

5) Compare the modified utility factor of all movements, which are immediately possible and calculate the difference in the utility factor for the current state and the next state. Follow the path having higher value.

6) Repeat the steps 1-5 until the target position is reached

Fig.2. shows the utility factor $Q(S,a)$ calculated for each movement while fixing the starting point at *s1,* destination point at *s16,* immediate reward function *r(s, a)* as 400,and ϒ- the relative value of delayed versus immediate reward as 0.5 after running the TD learning algorithm .

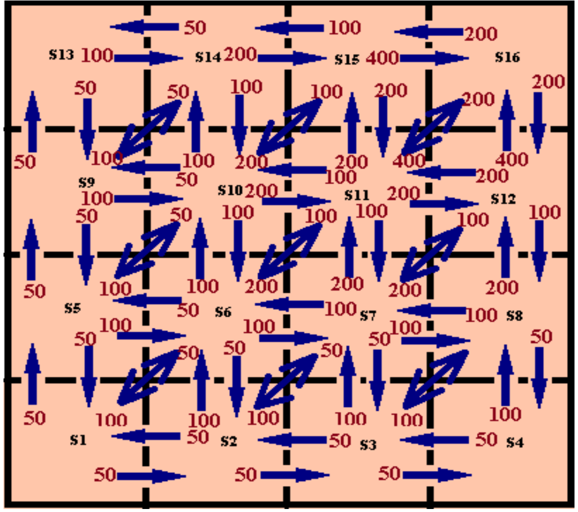


Fig.2. Utility factor for each movement

IV. Simulation Results

Modified TD Learning algorithm is implemented in MATLAB. A path planning is implemented using the same algorithm and it is tested in a 4x4 grid environment by varying the obstacle number and positions. Fig.3 and Fig.4 show the trajectory given by the MTD Learning algorithm for different number and position of static obstacles. In all cases destination point is fixed at (3, 3) and source point can be varied. In Fig.3 and Fig.4 source point is fixed at (0, 0). The Fig.3 (Figure 1) shows the trajectory obtained for single obstacle placed at (2, 2). In Fig.3 (Figure 2), two obstacles are considered which are placed at (1, 1) and (2, 2).

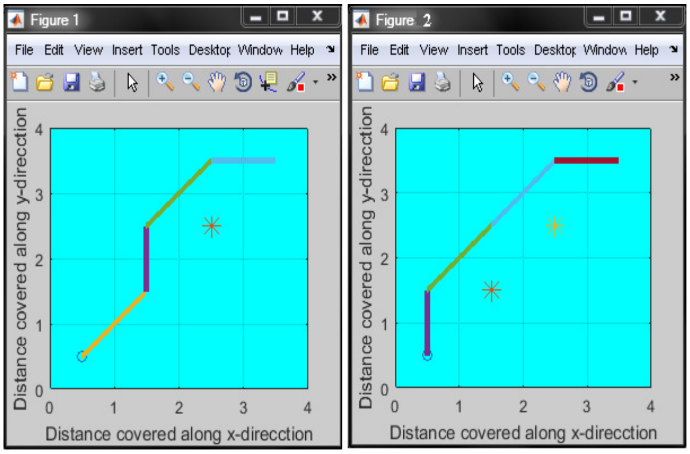


Figure 1
Figure 2
File Edit View Insert Tools Desktop Window Help
Distance covered along y-direcction
Distance covered along x-direcction
0
1
2
3
4

Fig.3.Path planning for one and two obstacles

In Fig.4 (Figure 1), path planning is carried out for three obstacles which are at positions (0, 1), (1, 1) and (3, 2) and Fig.4 (Figure 2) shows the four obstacle case which are placed at (1, 1), (2, 1), (1, 2) and (2, 2) respectively.

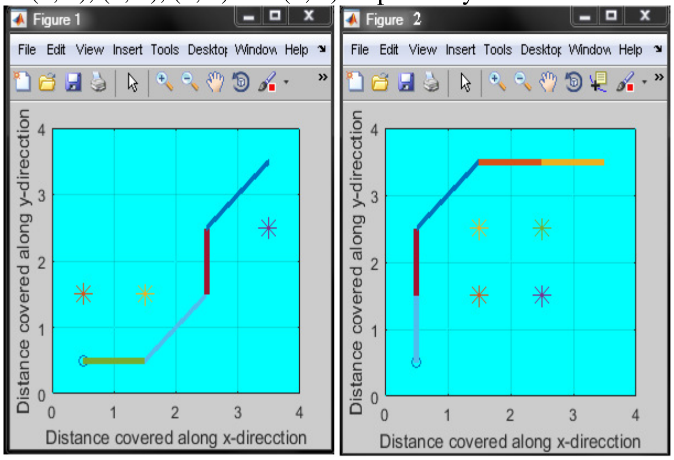


Figure 1
Figure 2
File Edit View Insert Tools Desktop Window Help
Distance covered along y-direcction
Distance covered along x-direcction
0
1
2
3
4

Fig.4. Path planning for three and four obstacles

The algorithm worked satisfactorily when different source points were considered. The Fig.5 and Fig.6 show the trajectory output given by the MTD algorithm for variable starting points. The destination point is fixed at (3, 3). In Fig.5 (Figure 1), the starting point is at (0, 1) and it investigates three obstacles placed at (0, 2), (1, 2) and (2, 1). Fig.5 (Figure 2) shows the case of starting point at (0, 3) with single obstacle at (1, 3) and supports the vertical backward movement of the robot to avoid the obstacle.

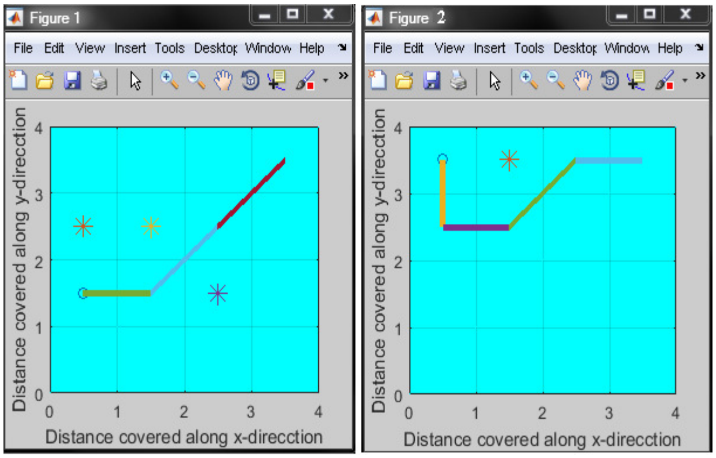


Figure 1
Figure 2
File Edit View Insert Tools Desktop Window Help
Distance covered along y-direcction
Distance covered along x-direcction
0
1
2
3
4

Fig.5. Path planned for different source points

Fig.6 (Figure 1) examines a case with the starting point at (1, 1) and two obstacles placed at (1, 2) and (2, 2). In Fig.6(Figure 2) starting point is at (1, 0) and three obstacles are taken into account at (3,0),(3,1) and (3,2).

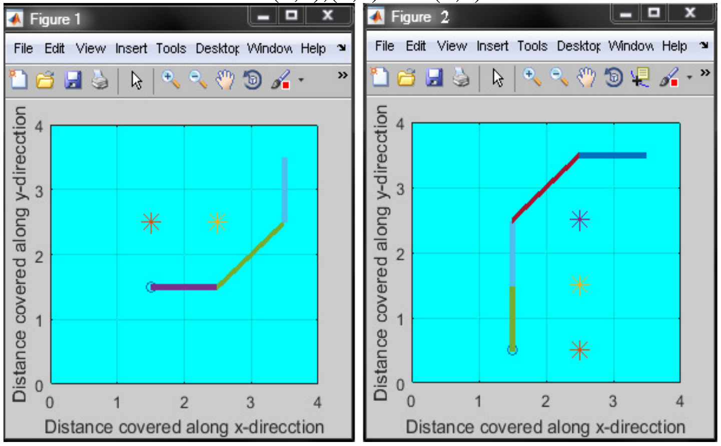


Figure 1
Figure 2
File Edit View Insert Tools Desktop Window Help
Distance covered along y-direcction
Distance covered along x-direcction
0
1
2
3
4

Fig.6. Path planned for different source points

There are certain cases where MTD Learning algorithm fails to give output. Fig.7. shows two cases where MTD Learning fails to find the optimum path to destination. In Fig.7 (Figure 1) starting point is at (0, 0) and three obstacles are placed at (2, 2), (2, 3) and (3, 2). The algorithm fails to find the next move after reaching at (1, 3) since there is no way to reach destination by avoiding obstacle. Similar is the case of Fig.7 (Figure 2) where the starting point is again at (0, 0) and three obstacles are placed at (0, 1), (1, 1) and (1, 0). Since all forward movements are blocked by obstacles at starting point, the algorithm take a backward movement initially but fails to move afterwards.

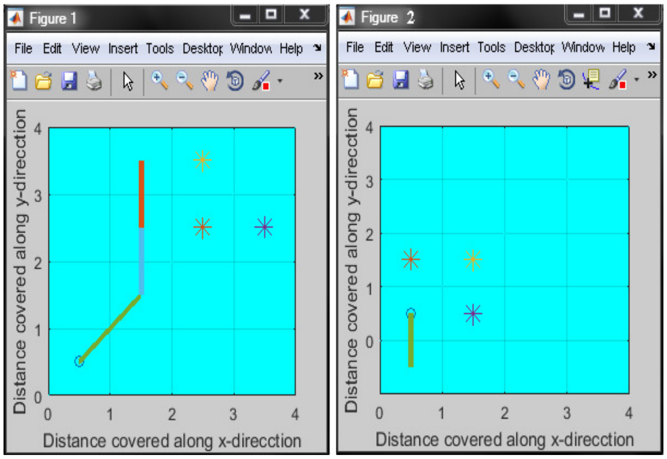
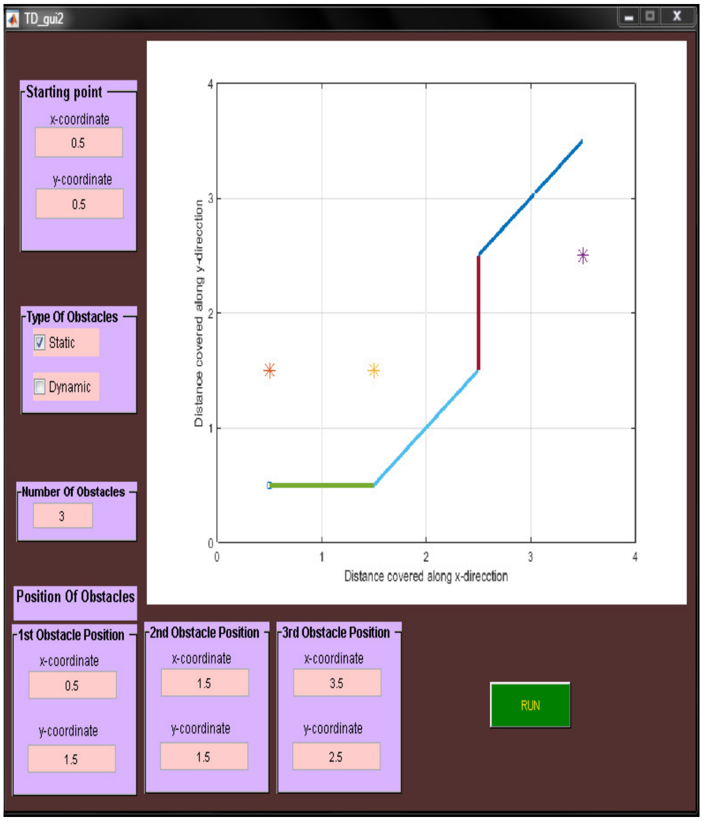


Figure 1
File Edit View Insert Tools Desktop Window Help
Distance covered along y-direcction
Distance covered along x-direcction
0
1
2
3
4
Figure 2
File Edit View Insert Tools Desktop Window Help
Distance covered along y-direcction
Distance covered along x-direcction
0
1
2
3
4

Fig.7. Cases where MTD Learning fails

A GUI for the path planning is also developed in MATLAB. Fig.8. shows the GUI developed for MTD Learning based path planning. The starting point, type of obstacle, number and position of obstacles are given as user inputs and after pressing the RUN button, the algorithm will run for the given inputs and obtained path will be shown in the GUI.


TD_gui2
Distance covered along y-direcction
Distance covered along x-direcction
0
1
2
3
4
RUN

Starting point
x-coordinate 0.5
y-coordinate 0.5

Type Of Obstacles
☑ Static
□ Dynamic

Number Of Obstacles 3

Position Of Obstacles

1st Obstacle Position
x-coordinate 0.5
y-coordinate 1.5

2nd Obstacle Position
x-coordinate 1.5
y-coordinate 1.5

3rd Obstacle Position
x-coordinate 3.5
y-coordinate 2.5

Fig.8. GUI for the MTD Learning based path planning

The developed algorithm is compared in terms of execution time of algorithm in various cases with the conventional path planning algorithm Dijkstra's algorithm. Table.1 shows the comparison study of time of execution of MTD Learning and Dijkstra's algorithm based path planning. It is found that Modified TD Learning is faster in computation compared to Dijkstra.

TABLE I SIMULATION RESULTS OF EXECUTION TIME OF MODIFIED TD LEARNING AND DIJKSTRA'S ALGORITHM

SL NO.	Number Of Obstacles	Position Of Obstacles (x-coordinate, y-coordinate)	Time of Execution of algorithm (in seconds)	
			TD Learning algorithm	Dijkstra's algorithm
1	0	-	0.00476	0.043874
2	1	(2.5,2.5)	0.018919	0.045596
3	2	(1.5,1.5),(2.5,2.5)	0.01967	0.048541
4	3	(0.5,1.5),(1.5,1.5),(3.5,2.5)	0.020014	0.051879
5	4	(1.5,1.5),(1.5,2.5),(2.5,1.5), (2.5,2.5)	0.035182	0.055570
6	6	(0.5,1.5), (1.5,1.5), (1.5,2.5), (2.5,0.5), (2.5,2.5), (3.5,1.5)	0.039984	0.056944
7	8	(0.5,1.5), (1.5,0.5), (1.5,3.5), (2.5,0.5), (2.5,1.5), (2.5,2.5), (3.5,1.5), (3.5,2.5)	0.042093	0.061727

V. CONCLUSIONS AND FUTURE SCOPE

The robot capable of finding the best trace of path by executing the Modified TD Learning algorithm for an unknown grid environment with static obstacles is simulated successfully using MATLAB software. A GUI for the same is developed. The developed algorithm is compared with Dijkstra's algorithm in terms of execution time of algorithm and found MTD Learning is faster. It is proposed to implement the same on an iRobot platform.

REFERENCES

[1] Divya Davis and Supriya.P, "Implementation Of Fuzzy Based Robotic Path Planning", Proceedings of the Second International Conference on Computer and Communication Technologies IC3T 2015, Volume 2 ,Springer, pp 383-390,2015J.

[2] ParvathyJoshy, Supriya.P, "Implementation of robot path planning using Ant Colony Optimisation", International Conference on Inventive Computation Technologies (ICICT), Vol.3,pp.163-168,August 26-27,2016.

[3] Ms.Jasna.S.B, Dr. Supriya P and Dr. T N P Nambiar, "Remodeled A* Algorithm for Mobile Robot agents with Obstacle Positioning.",IEEE International Conference on Computational Intelligence and Computing Research, 2016.

[4] Dennis Barrios Aranibar and Pablo Javier Alsina "Reinforcement Learning-Based Path Planning for Autonomous Robots,"Laborat´orio de SistemasInteligentesDepartamento de Engenharia de Computac¸ .ao e

Automac¸ .aoUniversidade Federal do Rio Grande do Norte Campus Universit´ario .Lagoa Nova 59.072-970 - Natal - RN –Brasil.

[5] BharadwajSrinivasan “Robot Navigation in Partially Observable Domains using Hierarchical Memory-Based Reinforcement Learning,”The 2nd International Conference on Ubiquitous Robots and Ambient Intelligence.

[6] Lavi M. Zamstein, Dr. A. Antonio Arroyo and Sara Keen “Koolio: Path planning usingreinforcement learning on a real robot platform,”Florida Conference on Recent Advances in Robotics, FCRAR 2006.

[7] Rongkuan Tang, Hongliang Yuan “An Error-Sensitive Q-learning Approach for Robot Navigation,” Proceedings of the 34th Chinese Control Conference, Hangzhou, China, July 28-30, 2015.SDGG.

[8] AbhishekGhosh Roy, PratyushaRakshit, AmitKonar, Samar Bhattacharya and Eunjin Kim “Adaptive Firefly Algorithm for Nonholonomic Motion Planning of Car-like System,” 2013 IEEE Congress on Evolutionary Computation,Cancún, México, June 20-23.

[9] AbdulrahmanAltahhan “A Fast Learning Variable Lambda TD ModelUsed to Realize Home Aware Robot Navigation,” 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, July 6-11, 2014.

[10] XiaodongZhuang ,QingchunMeng , and Bo Yin Hanping Wang “Robot Path Planning by Artificial Potential Field Optimization Based on Reinforcement Learning with Fuzzy Slate,” Proceedings of the 4th World Congress on Intelligent Control and Automation, Shanghai, P.R.China, june 10-14, 2002.

[11] RICHARD S. SUTTON “Learning to Predict by the Methods of Temporal Differences”, Machine Learning 3: 9-44,1988 © 1988 Kluwer Academic Publishers, Boston - Manufactured in The Netherlands.

[12] Pablo Escandell-Montero, Jos´e M. Mart´ınez-Mart´ınez, Jos’e D.Mart in-Guerrero, Emilio Sonia-Olivas and Juan G’omez-Ssnchis “Least-Squares Temporal Difference Learning based on Extreme Learning Machine”, ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.