

Comparison of SARSA algorithm and Temporal Difference Learning Algorithm for Robotic Path Planning for Static Obstacles

Laya Harwin

Department of Electrical and Electronics Engineering
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham, India
layaamy.harwin@gmail.com

Supriya P

Department of Electrical and Electronics Engineering
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham, India
p_supriya@cb.amrita.edu

Abstract—For any mobile device, the ability to navigate by its own in its environment is a very important requirement. Avoiding dangerous situations such as accidents and conditions that are not safe comes primarily and if the robot is designed to find certain places according to its application then the robot should find those places. Path planning is an important requirement for such mobile robots. For doing this the robots needs to be aware of their environment. This is possible only if robots are provided with information about the environment. In this paper, the different obstacle occurring cases are studied and simulated for SARSA and temporal difference algorithm for a 6x6 grid in MATLAB and it is found that the success rate of SARSA is much better than that of TD. It is proposed to implement the SARSA algorithm in iRobot with a mobile interface for interaction with the user.

Keywords—SARSA; TD; Path Plannig; Success rate

I. INTRODUCTION

Each and every task that humans are involved with will be automated with time and this is only possible with robots. These kinds of robots should be employed with various capabilities like path planning, localization, mapping and so on. Robotics is a field which is flexible and adaptive to various areas like, they can move around in places where humans fear to tread like dull, dirty, dangerous and difficult areas provided the robots are trained to do so.

Path planning is a key domain of research in robotics and the core idea is to permit the robot to find the optimal path between two locations in an environment. The optimal paths could be paths which minimize the turning rate, the braking rate or whatsoever a specific hypothesis requires. All these have a lot of application in other fields like computer games and solving mazes.

In this paper, two algorithms namely SARSA (state, action, reward, state, action) and Temporal Difference Learning (TD) is compared based on their success rate with an aim of successfully reaching the destination point from

the start point avoiding all the obstacles present in its path. The algorithm that fails to reach the destination avoiding the obstacles is considered to have less success rate compared to the other.

The paper is organized into five sections. The first portion introduces to the importance of robotics and path planning. The second section deals with the related work on path planning, reinforcement learning and SARSA algorithm. Problem formulation of the algorithms SARSA and TD comes in the third portion. Simulation results are highlighted in the fourth section and the final section contains the conclusions and future scope in the related work.

II. RELATED WORK

Path planning is one of the main requirements of autonomous robots. There are different algorithms that are developed for path planning [2]. This is an area which has huge research scope. Ant colony optimization is one of the path finding method where the robot takes the path followed the most which is similar to how ants find their food [9]. Path discovering can also be found using fuzzy logic where it does actions to control the wheels of the robot [10]. In temporal difference algorithm which is an RL(reinforcement learning) algorithm is used in path detection by assigning priority for each action and accordingly rewarding them [4] [11].

Reinforcement Learning is a recent technique for robotic path planning. Reinforcement learning is an online learning method which interacts with environment if required to take real time decisions [1]. It is mainly categorized into two. They are model free and model-based algorithms. Model based algorithms are those algorithms that have pre-information of the environment whereas model free algorithm does not have any information about its environment [2].

One of the main RL algorithm focused in this paper is SARSA. It is an on-policy learning algorithm. It has proved its contribution in many fields like swarm RL algorithm where

multiple agents communicate with each other and exchange information [5]. It is used in self-adaptive traffic lights control systems in urban areas and it controls the traffic signals automatically [6]. SARSA is also used in providing optimal power in power systems [8] and in cellular networks for channel allocation which also enables blocking calls [7].

The concepts of reinforcement learning can be implemented in robots only when they can do specific tasks and also compute the path to be travelled from the start point to the destination point [1]. Robots can be provided with a topological path so that they can track their path [3]. Path tracing can also be done using Markovian decision process where each action is taken based on reward and punishment [1][4].

III. PROBLEM FORMULATION

Reinforcement learning comes under the domain of Machine Learning. It takes the right or suitable action to maximize reward in a particular situation. It is employed by various software and machines to determine the best possible solution it should apply in a specific situation. There are different categories of reinforcement learning. SARSA algorithm is one of them which is discussed here.

The expansion of the word SARSA is state-action-reward-next state-next action. This signifies that the final function for modifying the Q-value is dependent on the agent's current state " S_1 ", the action chosen by the agent " A_1 ", the reward given for choosing this action " R ", the next state the agent reaches after the action is taken " S_2 ", and finally the next action the agent chooses in the new state " A_2 ". The acronym for the quintuple ($s_t, a_t, r_t, s_{t+1}, a_{t+1}$) is SARSA.

The steps involved in SARSA algorithm is described below:

- 1) First, arbitrarily initialise $Q[s, a]$. Then the current state- s is observed and then an action is selected on Q based on a policy.
- 2) Then action a is done and then the reward and state- r, s' is observed.
- 3) Next action- a' is selected on Q by a policy.
- 4) Update the Q -value according to the equation:
$$Q[s, a] \leftarrow Q[s, a] + \alpha (r + \gamma Q[s', a'] - Q[s, a]) \quad (1)$$
In "(1)", $Q[s, a]$ is the current state of Q -value, $Q[s', a']$ is the next state of Q -value, r is the function for reward decided based on the policy, α and γ are constants.
- 5) Then take the next state and next action.
- 6) Repeat the steps 2-5 until the destination grid is reached.

In this paper SARSA algorithm is compared with TD algorithm. TD algorithm is an OFF-policy algorithm whereas SARSA is an ON-policy algorithm. TD learning takes each action using greedy policy and there is no interaction with the environment but in SARSA algorithm each action is taken according to the current policy which can also be greedy

policy. It gathers information from environment to find the right path.

The algorithm of TD algorithm is explained below:

- 1) Q -value for each grid is calculated according to the equation:
$$Q[s, a] \leftarrow Q[s, a] + \alpha (r + \gamma \max_{a'} Q[s', a'] - Q[s, a]) \quad (2)$$
In "(2)", $Q[s, a]$ is the Q -factor for each state, $r[s, a]$ is the given for the reward function, γ and α are constants and $Q[s', a']$ is the next state Q -factor.
- 2) Reward is given according to the movement. Diagonal movement has the maximum reward, then for vertical movement and least reward for horizontal movement.
- 3) Then compare the Q -factor value for each direction and the one having maximum Q -value is taken as the next state.
- 4) Repeat these steps until it reaches the destination grid.

The system layout of a 6x6 grid is as shown in fig 1. Each grid has its x and y value in the form (x, y) . This information is provided to understand the position of obstacles in the grid. The starting point will be $(0.5, 0.5)$ and the destination point will be $(5.5, 5.5)$.

(0.5, 5.5)	(1.5, 5.5)	(2.5, 5.5)	(3.5, 5.5)	(4.5, 5.5)	(5.5, 5.5)
(0.5, 4.5)	(1.5, 4.5)	(2.5, 4.5)	(3.5, 4.5)	(4.5, 4.5)	(5.5, 4.5)
(0.5, 3.5)	(1.5, 3.5)	(2.5, 3.5)	(3.5, 3.5)	(4.5, 3.5)	(5.5, 3.5)
(0.5, 2.5)	(1.5, 2.5)	(2.5, 2.5)	(3.5, 2.5)	(4.5, 2.5)	(5.5, 2.5)
(0.5, 1.5)	(1.5, 1.5)	(2.5, 1.5)	(3.5, 1.5)	(4.5, 1.5)	(5.5, 1.5)
(0.5, 0.5)	(1.5, 0.5)	(2.5, 0.5)	(3.5, 0.5)	(4.5, 0.5)	(5.5, 0.5)

Fig1: System Layout

IV. SIMULATION RESULTS

Some of the assumptions made while implementing both SARSA and TD algorithm are mentioned below:

- 1) A 6x6 grid containing 36 grids is used for path planning
- 2) Obstacles are assumed to be placed at the centre of the grid.
- 3) The start and destination node are fixed.

Different obstacle occurrence scenarios for SARSA and TD algorithm for a 6x6 grid was tested. Both the algorithms were coded in MATLAB according to the algorithm explained in the problem formulation section. There are cases where TD and SARSA algorithm works properly. When there are no

obstacles both the algorithm works similarly as given in fig 2. The thick black marking shows the shortest route from source to destination.

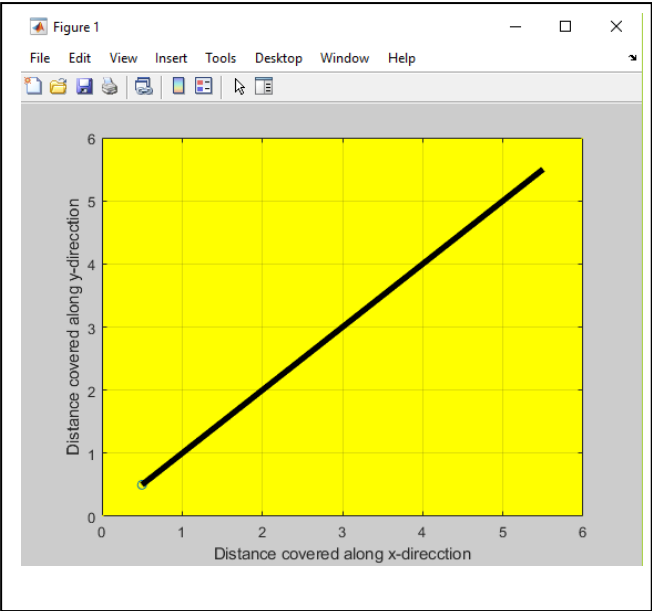


Fig 2: Path planning for 0 obstacle for SARSA and TD algorithm

In fig 2, (0.5,0.5) is the start point and (5.5,5.5) is the destination point. When there is no obstacle the path taken is according to the diagonal movement.

Compared to TD algorithm, SARSA algorithm has got more success rate. There are cases where TD algorithm fails and SARSA works perfectly. Three such cases are highlighted here. In fig 3, TD fails for four obstacles and their positions are (3.5,5.5), (3.5,4.5), (3.5,3.5) and (3.5,2.5) which is represented as pink square boxes. It fails to reach the destination point (5.5, 5.5). According to TD algorithm it gives maximum priority for diagonal movement and moves forward, Due to the presence of obstacles in the positions (3.5,2.5) and (3.5, 3.5), it takes a vertical up movement and reaches the top and cannot proceed further. Hence TD failed in this condition. However, SARSA works perfectly for this condition which is shown in fig 4. The specialty of SARSA algorithm is the interaction with its environment. Here it randomly selects grids and find out the possible routes from the grid till it reaches its destination. This helps it to tackle situations where TD fails but SARSA works properly.

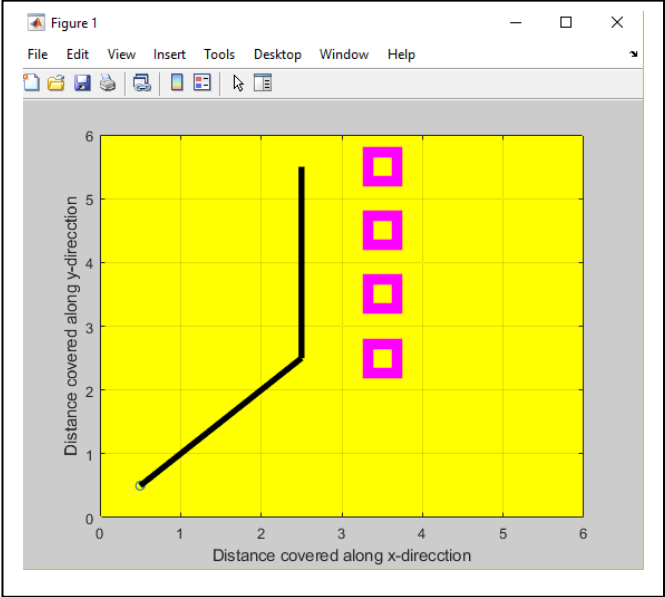


Fig 3: Case 1 when TD fails

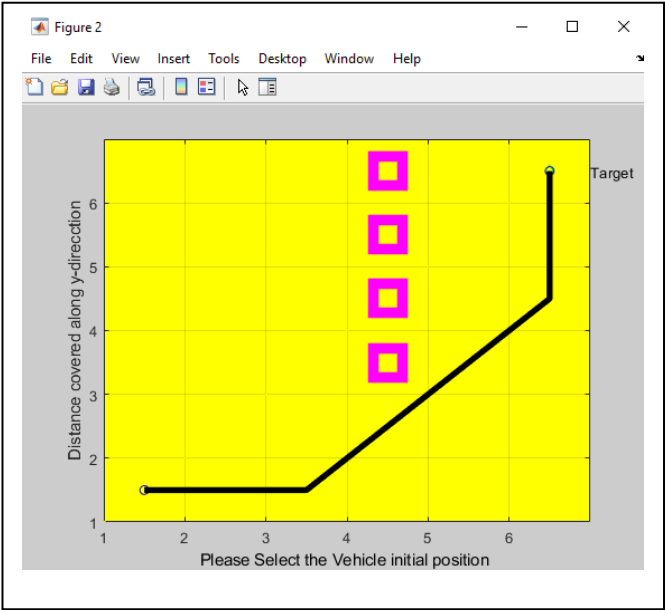


Fig 4: Case 1 when SARSA works

In fig 5, case 2 where TD fails is shown for four obstacles with their positions (2.5,3.5), (3.5,3.5), (4.5,3.5) and (5.5,3.5). Here also it fails to reach its destination grid. The path cannot proceed more when continuous horizontal right movements are taken after diagonal movements. For this particular case SARSA works and reaches the destination without any collision with the obstacles which is illustrated in fig 6.

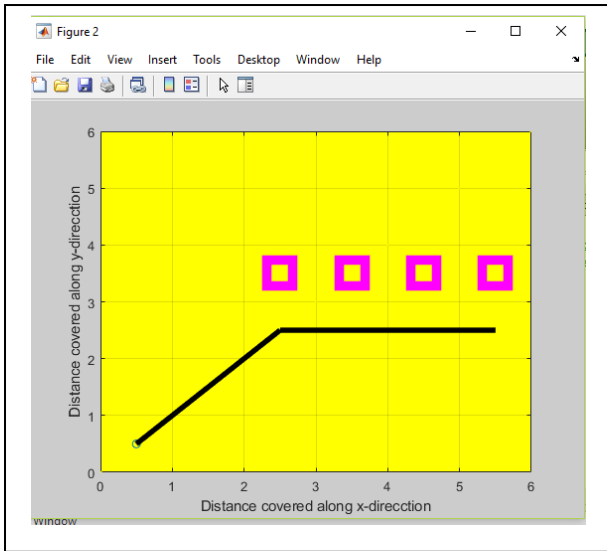


Fig 5: Case 2 where TD fails

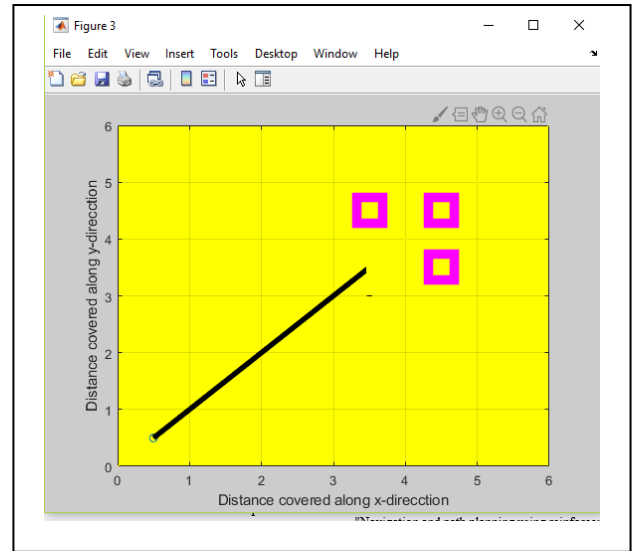


Fig 7: Case 3 when TD fails

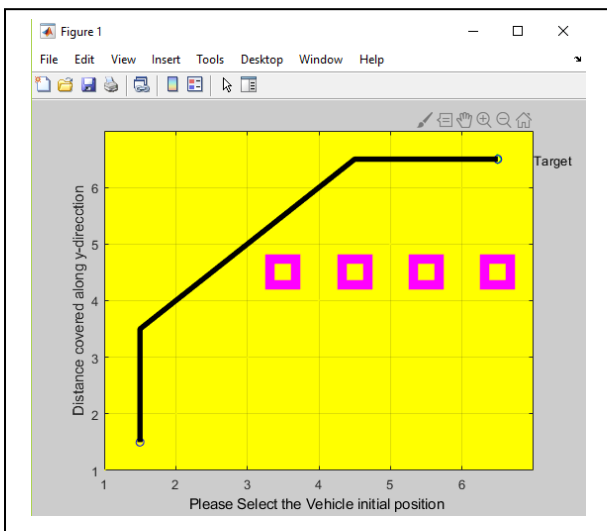


Fig 6: Case 2 where SARSA works

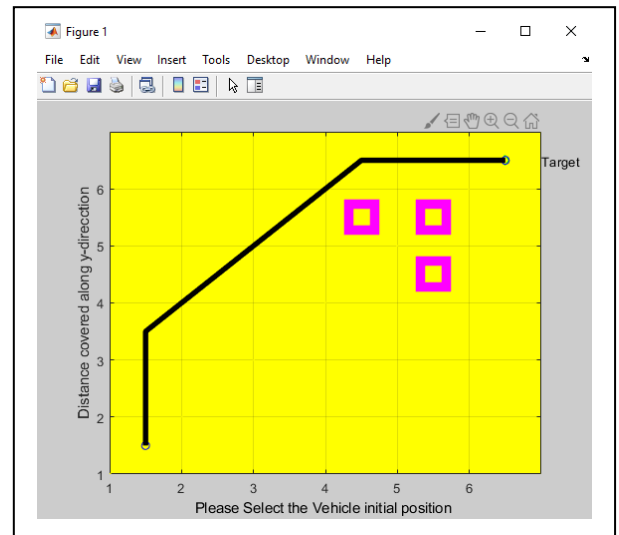


















Fig 8: Case 3 where SARSA works

Fig 7, illustrates the third case where TD fails for three obstacles and their positions are (3.5,4.5), (4.5,4.5) and (4.5,3.5). Here the path cannot proceed because all the direction of movement i.e. diagonal, horizontal and vertical movements is obstructed. The path taken by SARSA is shown for the above case in fig 8 which successfully reaches the end point.

TABLE I. CASES WHERE TD FAILS AND SARSA WORKS FOR DIFFERENT NUMBER OF OBSTACLES

SL No	No. of Obstacles	Special Cases	TD	SARSA
1	3	a. (4,5,5,5),(4,5,4,5),(4,5,3,5) b. (3,5,4,5),(4,5,4,5),(4,5,3,5)	 	 
2	4	a. (3,5,5,5),(3,5,4,5),(3,5,3,5),(3,5,2,5) b. (2,5,3,5),(3,5,3,5),(4,5,3,5),(5,5,3,5)	 	 
3	5	a. (2,5,4,5),(3,5,4,5),(4,5,4,5),(4,5,3,5),(4,5,2,5) b. (0,5,5,5),(1,5,4,5),(2,5,3,5),(3,5,2,5),(4,5,1,5)	 	 
4	6	a. (2,5,4,5),(3,5,4,5),(3,5,3,5),(3,5,2,5),(3,5,1,5),(3,5,0,5) b.(0,5,3,5),(1,5,3,5),(2,5,3,5),(3,5,3,5),(3,5,2,5),(3,5,1,5)	 	 

Several cases of 3,4,5 and 6 obstacles in a 6×6 grid are analyzed for both the TD and SARSA algorithm. A few such case are listed in Table 1, where TD fails and SARSA works. From these cases its evident that SARSA is better than TD when it comes to better path planning. Its success rate is much higher than TD. TD fails in all the above cases due to greedy policy where priority is given for each direction of movement and not for the optimal path planning. SARSA learns about the environment first and then does the path planning which helps the algorithm to determine the optimal path.

V. CONCLUSION AND FUTURE SCOPE

Path Planning is simulated with various number of obstacles using two algorithms- TD and SARSA for a 6×6 grid. Different cases of obstacle positions are simulated and tested for both SARSA and TD. In majority of the cases, both the algorithms work similarly but there are few cases where TD fails and SARSA works precisely. This proves that the success rate is for SARSA is higher than TD. This is due to the interaction with the environment in SARSA algorithm

which helps it to solve the problem in spite of the presence of many obstacles.

SARSA algorithm can be used to solve complex mazes with a greater number of obstacles and with increased size of grid. The same algorithm can be implemented for static and dynamic obstacles in iRobot interfaced with a mbed controller. Such a work in in progress and development of a mobile application to communicate with the robotics in progress.

REFERENCES

- [1] Zhang, Qian et al. "Reinforcement Learning in Robot Path Optimization." , Journal of Software(JSW 7) (2012):657-662.
- [2] Ravishankar, N.R, Vijayakumar and M.V, "Reinforcement Learning Algorithms: Survey and Classification", **Indian Journal of Science and Technology**, [S.I.],2017, doi:10.17485/ /2017/v10i1/109385.
- [3] D. P. Romero-Martí, J. I. Núñez-Varela, C. Soubervielle - Montalvo and A. Orozco-de-la- Paz , "Navigation and path planning using reinforcement learning for a Roomba robot," *2016 XVIII Congreso Mexicano de Robotica*, Sinaloa, 2016, pp. 15.
- [4] Shreyas J and Sandeep J. " Modern Machine Learning Approaches For Robotic Path Planning", *IJCSIT*(2016):256-259
- [5] H. Iima and Y. Kuroe, "Swarm reinforcement learning algorithms based on Sarsa method", *2008 SICE Annual Conference*, Tokyo, 2008, pp. 2045-2049
- [6] C. Li, M. Wang, S. Yang and Z. Zhang, "Urban Traffic Signal Learning Control Using SARSA Algorithm Based on Adaptive RBF Network", *2009 International Conference on Measuring Technology and Mechatronics Automation*, Zhangjiajie, Hunan, 2009,pp.658-661.doi:10.1109/ICMTMA . 2009.445
- [7] N. Lilith and K. Dogancay, "Distributed reduced-state SARSA algorithm for dynamic channel allocation in cellular networks featuring traffic mobility," *IEEE International Conference on Communications*, 2005. *ICC*, Seoul, 2005, pp. 860-865 Vol. 2. doi: 10.1109/ICC.2005.1494473
- [8] M. R. Tousi, S. H. Hosseini, A. H. Jadidinejad and M. B. Menhaj, "Application of SARSA learning algorithm for reactive power control in power system," *2008 IEEE 2nd International Power and Energy Conference*, Johor Bahru, 2008, pp. 1198-1202. doi: 10.1109/PECON.2008.4762658
- [9] P. Joshy and P. Supriya, "Implementation of robotic path planning using Ant Colony Optimization Algorithm," *2016 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, pp. 1-6, 2016.
- [10] D. Davis and P. Supriya, "Implementation of Fuzzy-Based Robotic Path Planning", *3rd IEEE International Conference on Engineering and technology (ICETECH)*, 2016.
- [11] Devika S. Nair and P. Supriya, " Comparison of Temporal Difference Learning Algorithm and Dijkstra's Algorithm for Robotic Path Planning", *International Conference on Intelligent Computing and Control Systems*, Madurai. 2018. (to be published)
- [12] Ravishankar, N. R; Vijayakumar, M.V." Reinforcement Learning Algorithms: Survey and Classification". *Indian Journal of Science and Technology*, ISSN: 0974-5645, 2017