Curiginal

Document Information

Analyzed document	RAPIG-D PROJECT REPORT.pdf (D140555420)
Submitted	2022-06-16 23:46:00
Submitted by	Dr. V. Sathiesh Kumar
Submitter email	sathiieesh@gmail.com
Similarity	36%
Analysis address	sathiieesh.annauniv@analysis.urkund.com

Sources included in the report

SA	**Anna University, Chennai / RAPIG-D.pdf** Document RAPIG-D.pdf (D138452381) Submitted by: sathiieesh@gmail.com Receiver: sathiieesh.annauniv@analysis.urkund.com	26
W	URL: https://www.finsliqblog.com/ai-and-machine-learning/what-are-the-types-of-reinforcement-learning-algorithms/ Fetched: 2021-04-22 14:49:16	5
W	URL: https://www.datacamp.com/tutorial/introduction-reinforcement-learning Fetched: 2022-06-16 23:47:40	1
W	URL: https://www.guru99.com/reinforcement-learning-tutorial.html Fetched: 2019-09-24 16:36:09	2
SA	**Université Virtuelle de Tunis / Rapport_PFE_Nour_Bejaoui.pdf** Document Rapport_PFE_Nour_Bejaoui.pdf (D110250487) Submitted by: khaled.khnissi@uvt.tn Receiver: khaled.khnissi.uvt@analyse.urkund.com	1
W	URL: https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning Fetched: 2021-12-18 05:46:48	1

Entire Document

1

In partial fulfilment for the award of the degree of BACHELOR OF ENGINEERING in ELECTRONICS AND COMMUNICATION DEPARTMENT OF ELECTRONICS ENGINEERING MADRAS INSTITUTE OF TECHNOLOGY ANNA UNIVERSITY: CHENNAI 600044 JUNE 2022

2 ACKNOWLEDGEMENT We consider it as a privilege and our primary duty to express our gratitude and respect to all those who guided and inspired us in the successful completion of the project. We owe solemn gratitude to Dr J. PRAKASH , Dean, Madras Institute of Technology, for having given consent to carry out the project work at MIT Campus, Anna University. We wish to express our sincere appreciation and gratitude to Dr. M. GANESH MADHAN, Professor and Head of the Department of Electronics Engineering, who has encouraged and motivated us in our endeavors. We are extremely grateful to our project guide Dr. V. SATHIESH KUMAR , Assistant Professor, Department of Electronics Engineering, for his timely and thoughtful guidance and encouragement for the completion of the project. We sincerely thank all our panel members Dr. S. P. JOY VASANTHA RANI, Dr. A. VIJI, and MR. N. JAGADEESH KUMAR for their valuable suggestions. We also thank all the teaching and non-teaching staff members of the Department of Electronics Engineering for their support in all aspects. SATHIYA MURTHI S (2018504604) PRANAV BALAKRISHNAN (2018504581) C ROSHAN ABRAHAM (2018504591)

3 BONAFIDE CERTIFICATE Certified that this Project Report titled “

who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate. Dr. M. GANESH MADHAN Professor and Head Department of Electronics Engineering Madras Institute of Technology Anna University Chennai - 600044

This work validates the hardware implementation of Reinforcement learning based path planning and the use of depth estimation for obstacle detection.

6 TABLE OF CONTENTS CHAPTER NO. TITLE PAGE NO. ABSTRACT 5 LIST OF TABLES 8 LIST OF FIGURES 9 LIST OF ABBREVIATIONS 10 1 INTRODUCTION 11 1.1 OVERVIEW 1.2 REINFORCEMENT LEARNING 1.2.1 ALGORITHMS IN REINFORCEMENT LEARNING 1.3 STEREO CAMERA 1.4 DEPTH MAP 1.5 OPENCV 1.6 MPU 6050 1.7 NVIDIA JETSON NANO 1.8 PYTHON 11 11 12 14 15 15 16 16 16 2 LITERATURE SURVEY 18 3 MOTIVATION AND OBJECTIVES 21 3.1 MOTIVATIONS 3.2 OBJECTIVES 21 21 4 PROPOSED WORK 22 4.1 REINFORCEMENT LEARNING 4.1.1 SARSA ALGORITHM 4.2 SOFTWARE IMPLEMENTATION 22 22 24

7 4.3 HARDWARE IMPLEMENTATION 4.3.1 COMPONENTS USED 4.3.2 TESTING SPACE 4.3.3 FLOW CHART 4.3.4 ALGORITHM 4.3.5 PATH PLANNING 4.3.6 OBSTACLE DETECTION 4.3.7 CALIBRATED MOVEMENT 24 25 27 28 29 30 31 31 5 RESULTS AND DISCUSSIONS 32 5.1 SOFTWARE IMPLEMENTATION 5.2 SOFTWARE IMPLEMENTATION 5.2.1 OBSTACLE DETECTION 5.2.2 PATH PLANNING 32 34 34 34 6 CONCLUSION AND FUTURE WORKS 38 REFRENCES 39

8 LIST OF TABLES TABLE NO. TITLE PAGE NO. 4.1 Components used in hardware implementation 25 5.1 Final Q-Table – Optimal Path 36 4.3 Full Q-Table 36

9 LIST OF FIGURES FIGURE NO. TITLE PAGE NO. 1.1 Xbox Kinect Stereo camera 14 1.2 Depth estimation 15 4.1 Flowchart of SARSA Algorithm 22 4.2 NVIDIA Jetson Nano board 25 4.3 MPU6050 gyro sensor 26 4.4 L298N Motor driver 26 4.5 Robot designed using the mentioned components – Front and Side view 26 4.6 Testing Space 27 4.7 Flow diagram for each episode of the algorithm 28 4.8 Flow diagram for decision making at each cell 28 4.9 Path Planning Algorithm 29 4.10 Angular velocity data integration 31 5.1 Map generated in software implementation 33 5.2 Plot of number of steps in each episode - software 33 5.3 Obstacle detection output 34 5.4 Map generated in hardware implementation 35 5.5 Plot of number of steps in each episode - hardware 35 5.6 Shortest route 37

10 LIST OF ABBREIVIATIONS TERM EXPLANATION SARSA State action state reward UAV Unmanned Aerial Vehicle RL Reinforcement Learning 3D Three Dimensional RoI Region of Interest SSH Secure Shell DMP Digital Motion Processor CV Computer Vision GPU Graphics Processing Unit MEMS Micro Electro-mechanical system AI Artificial Intelligence

11 CHAPTER 1 INTRODUCTION 1.1 OVERVIEW Reinforcement Learning is an adaptable and

of Reinforcement Learning in Autonomous Robots,

REINFORCEMENT LEARNING

12 Reinforcement Learning is a type of machine learning

and error using feedback from its own actions and experiences. Though both supervised and

the case of

the total cumulative reward of the agent. The figure below illustrates the action-reward feedback loop of a generic RL model. Some

the

Policy — Method to map agent’s state to actions Value — Future reward that an agent would receive by taking an action in a particular state 1.1.1 ALGORITHMS

IN

Policy-based • Model-based Value-based: Optimum value function can be found using this approach and it will be the maximum value at a state. Hence, at any state a long-term return will be expected by an agent under the
algorithm, there are two main policies for a learning agent. They are • On Policy • Off policy On policy: In this, the learning agent learns the value function according to the current action derived from the policy which is in use. SARSA technique comes under On Policy and uses the action performed by the current policy to learn the Q-value.
14 Off policy: In this, the learning agent learns the value function according to the action derived from another policy. Q-Learning technique comes under Off policy and uses the greedy approach to learn the Q-value. 1.2 STEREO CAMERA A stereo camera is a type of camera with two or more lenses with a separate image sensor or film frame for each lens. This allows the camera to simulate human binocular vision, and therefore gives it the ability to capture three-dimensional images, a process known as stereo photography. Stereo cameras may be used for making stereo-views and 3D pictures for movies, or for range imaging. The distance between the lenses in a typical stereo camera (the intra-axial distance) is about the distance between one's eyes (known as the intra-ocular distance) and is about 6.35 cm, though a longer base line (greater inter-camera distance) produces more extreme 3-dimensionality. Stereo cameras are mounted in cars to detect the lane's width and the proximity of an object on the road. Figure 1.1: Xbox Kinect Stereo camera
15 1.3 DEPTH ESTIMATION Depth Estimation is the task of measuring the distance of each pixel relative to the camera. Depth is extracted from either monocular (single) or stereo (multiple views of a scene) images. Traditional methods use multi-view geometry to find the relationship between the images. Newer methods can directly estimate depth by minimizing the regression loss, or by learning to generate a novel view from a sequence. Figure 1.2: Depth Estimation 1.4 OPENCV OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it is integrated with various libraries, such as NumPy, Python is capable of processing the OpenCV array structure for analysis .
16 1.5 MPU 6050 MPU6050 is a Micro Electro-mechanical system (MEMS), it consists of three- axis accelerometer and three-axis gyroscope. It helps us to measure velocity, orientation, acceleration, displacement and other motion like features. MPU6050 consists of Digital Motion Processor (DMP), which has property to solve complex calculations. MPU6050 consists of a 16-bit analog to digital converter hardware. Due to this feature, it captures three-dimension motion at the same time. 1.9
micro-controller that is generally used to prototype projects. Due to its GPU, it is commonly used for inference of neural networks and deep learning models on the edge. This makes it a suitable option for tasks such as Object detection, Image processing, Image classification. It can be used to interface with sensors, cameras, run python scripts and implement Reinforcement learning algorithms. 1.10 PYTHON Python is a programming language that supports the creation of a wide range of applications. Developers regard it as a great choice for Artificial Intelligence (AI), Machine Learning, and Deep Learning projects.
17 The Python language comes with many libraries and frameworks that make coding easy. This also saves a significant amount of time. Python projects can be integrated with other systems coded in different programming languages. This means that it is much easier to blend it with other AI projects written in other languages. Also, since it is extensible and portable, Python can be used to perform cross languages tasks. The adaptability of Python makes it easy for data scientists and developers to train machine learning models. It can be used to implement the RL SARSA algorithm as well
18 CHAPTER 2 LITERATURE SURVEY [7]
is proposed. First a deep Q-network is designed and it is trained which will evaluate the state action values function. Hence the q-value corresponding to each action is determined by this network. The actions used are left, right, forward. The input is captured from the environment and this RGB image is given to the DQN. The action selection strategy selects the suitable action to be taken for the robot. Hence the robot finally reaches the goal without colliding with any obstacles. Real time experiments have been done and proved this method as effective.

A modified SARSA (state-action-reward-state-action) algorithm has been proposed. Difference between Q-learning and SARSA algorithms have been analyzed and SARSA algorithm has been modified. In modified algorithm, after the robot reaches the final destination, the start and the end points are interchanged and the algorithm is made to run again. After all the iterations are over the shortest path among all the paths found is highlighted. This algorithm has high time complexity but it provides the safe path. It is found out that Q- learning algorithm is the fastest and also it gives the optimum path. It has been concluded that Q-learning algorithm is the best when executional risks are not taken into account. SARSA algorithm can be used when the path length doesn't matters and with low execution time. Iterative SARSA algorithm can be used when safe optimum path is required irrespective of the time complexity. [2] RL based planning algorithms provides a new resolution by integrating a high- level artificial intelligence. This work investigates the application of deep reinforcement learning algorithms for USV and USV formation path planning
20 with specific focus on a reliable obstacle avoidance in constrained maritime environments.
CHAPTER 3 MOTIVATION AND OBJECTIVES 3.1 MOTIVATION ? A learning based approach can be used to navigate and map an unknown environment and trace the best possible path between any two points. ? Stereo camera based depth mapping would be better suited to identify any type of obstacle and is a robust approach towards obstacle detection. ? Current methods are limited by their inability to adapt to new environments and varying obstacles. 3.2 OBJECTIVES ? To perform hardware implementation of reinforcement learning based path planning to generate a map of an unknown environment by employing depth estimation based obstacle detection and analyze its performance. ? To estimate the best possible path to reach a given destination
22 CHAPTER 4 PROPOSED WORK It is proposed to design a robot equipped with a stereo camera capable of generating a depth map which is integrated with SARSA algorithm - a Reinforcement Learning technique to find the optimal path from start to destination in a cell-based testing space while mapping the unknown environment by detecting and avoiding obstacles. 4.1
It shows adaptability and the ability to account for dynamic environments with limited to no parameters specified as opposed to other learning methods such as Supervised or Unsupervised learning. This is favorable when attempting to map an unknown environment, giving more flexibility. 4.1.1 SARSA ALGORITHM Figure 4.1: Flowchart of SARSA Algorithm
23 SARSA algorithm is a passive Reinforcement learning algorithm that can be applied to fully observable environments. It is an On policy algorithm. Q-value is to determine how favorable a specific action is in a specific state. To update the Q-values, an action is chosen at each state using a policy. The action is chosen based on this policy and in turn leads to the reward for the next state. The name of the algorithm represents the sequence of its operation. The current state S is given to the agent by the environment, which choses an action A based on the policy function and then this is fed to the environment. The environment determines the corresponding reward R for the action taken in the current state and the net state S which is given to the agent to take the next action A. (,) ← (,) + [+ + (+ , +) − Q (,)] Equation 4.1: SARSA Q-value update rule Equation 4.1 tells how to calculate the Q-value for each state-action pair given by the SARSA algorithm. Some parameters used in SARSA are ? Learning Rate ? Discount factor ? Initial conditions Learning Rate: It determines the extent to which the newly acquired information can override the old information. A factor value 0 indicates the agent doesn't learn anything while a factor value of 1 indicates the agent should consider only the most recent information.
24 Discount factor: It determines the importance of the future rewards. Value of 0 indicates that the agent is opportunistic and a value of 1 indicates that the agent is striving or struggling for long high term reward. If the value exceeds 1, it indicates that the Q-values may diverge. The policy function taken up for this implementation is as follows: Ninety percent of the actions are chosen based on the most favorable choice using the determined Q-values and Ten percent of the actions are chosen at random. 4.2 SOFTWARE IMPLEMENTATION There are many algorithms for path planning. In this work, SARSA (State – Action – Reward – State – Action) algorithm is used. The entire coding is done in Python Language. The validity of this algorithm is first estimated in a software implementation with a virtual agent and specified obstacles in a fixed environment. 4.3 HARDWARE IMPLEMENTATION
Figure 4.2)
XBOX 360 Kinect camera which is shown in Figure 1.1 and the
The Kinect camera provides the required depth map by utilizing a stereo camera setup along with an IR camera. An MPU6050 sensor (Figure 4.3)
which are controlled by the L298N motor driver (Figure 4.4).
25

COMPONENTS USED Table 4.1: Components used in hardware implementation COMPONENTS SPECIFICATION NUMBER USED NVIDIA Jetson Nano Developer Kit – 4GB 1 Robot Chassis Dimensions: 250 x 200 x 46 mm 1 DC motors 12 V, 100 rpm 4 Wheels Diameter – 130 mm Width – 60 mm 4 XBOX Kinect v1 Camera Stereo camera 1 Motor Driver L298N 1 Gyro sensor MPU6050 – 6 axis motion sensor 1 Battery 11.1 V 4200mah Li-Po 1 Figure 4.2: NVIDIA Jetson Nano board

26 Figure 4.3: MPU6050 gyro sensor Figure 4.4: L298N Motor driver Figure 4.5: Robot designed using the mentioned components – Front and Side view

27 Figures 4.5 shows the robot that had been designed for estimating the validity of the hardware estimation. The components mentioned above have been put together and integrated to perform their respective functions in the task. The MPU6050 sensor is utilized to accurately turn 90 degrees as movement accuracy is a concern in a cell-based environment. The Kinect camera shall provide the depth map for obstacle detection and the NVIDIA Jetson Nano is the micro- controller used to integrate the modules and perform the Reinforcement learning algorithm. It can be remotely operated through SSH. 4.3.2 TESTING SPACE A 3x3 testing space with cells of 2 feet length and 2 feet width is utilized to test this implementation. Figure 4.6 shows the testing space with the robot at the starting point (0,0). The diagonal bottom right point (2,2) is taken as the destination and boxes have been placed to represent obstacles at (0,1) and (2,1). Figure 4.6: Testing space

28 4.3.3 FLOW CHART Figure 4.7: Flow diagram for each episode of the algorithm Figure 4.8: Flow diagram for decision making at each cell

29 Figure 4.9: Path Planning Algorithm 4.3.4 ALGORITHM

Figure 4.7 depicts the workflow of the algorithm implemented for each episode and Figure 4.8 shows the decision-making process at each cell. Figure 4.9 shows the Path planning algorithm. Step 1: Reset environment at start of each episode Step 2: Choose action at each state based on policy function Step 3: Check validity of next state Step 4: Turn to required direction based on action chosen Step 5: Check next state for presence of obstacle Step 6: If no obstacle, move to next state and continue

30 Step 7: Check if new state is the destination given Step 8: If destination, current route is checked for shortest path from start to destination Step 9: Each episode ends when either an obstacle is detected or destination is reached and the obstacles are populated in the map Step 10: A specified number of episodes are run and the agent learns the optimal path and location of obstacles 4.3.5 PATH PLANNING At each state i.e., at each cell the

agent – the robot chooses an action and then takes the step if it is valid based on the action by turning to the required direction – up, down, left or right.

OBSTACLE DETECTION This module is performed by using the depth map provided by the Kinect v1 camera. A threshold is applied on the depth map to detect objects in the immediate next cell i.e., around two feet away from the robot. A region of interest is also applied in the middle sixty percent width and length of the frame to increase the accuracy of obstacle placement. When an object is detected covering more than ten percent of this area, an obstacle is said to be detected. 4.3.7 CALIBRATED MOVEMENT The task requires accurate four directional movement to enable repetitive traversal in the testing space in each episode of the learning process. The forward and backward movements are controlled using the time taken to move from one cell to the next (1.32 seconds). For turns, the MPU6050 sensor is utilized. The angular velocity of the robot is extracted from the sensor which is then integrated over time (Figure 4.10) to identify the angular displacement. This is then used to control the turning angle. Figure 4.10: Angular velocity data integration

32 CHAPTER 5 RESULTS AND DISCUSSIONS 5.1

The start point is (0,0) and the destination is (9,9). Figure 5.1 shows the output of the SARSA algorithm – the map of environment with the detected obstacles and the optimal path to reach the destination from the starting point.

Figure 5.2 shows a plot of Number of steps taken as each episode progresses. As the agent identifies obstacles and starts reaching the destination, the Q-table gets updated and the agent starts favoring the optimal path to reach the destination in the shortest path identified. Thus, as the episodes progress, the number of steps taken in each episode reduces. This is indicative of the learning process of the agent, allowing it to adapt to the unknown environment after exploring and mapping obstacles and identifying the shortest path to the given destination.

33 Figure 5.1: Map generated in software implementation Figure 5.2: Plot of number of steps in each episode - software

34 5.2 HARDWARE IMPLEMENTATION 5.2.1 OBSTACLE DETECTION Figure 5.3 shows the output of the Obstacle detection module. The depth map from the stereo camera is visualized. The red box is the region of interest taken for accurate object placement and the red outline of the box indicates the presence of an obstacle as the size of the object within the specified threshold of 50 cm is greater than 10 percent of the area of the RoI. Figure 5.3: Obstacle detection output 5.2.2 PATH PLANNING The robot agent was allowed to explore a 3 x 3 testing space (Figure 4.6) for 10 episodes. The starting point is at (0,0) and the destination is at (2,2). The agent successfully detected obstacles in adjacent cells and populated the detected obstacles as expected. The movement was also accurate, controlled by the MPU6050 gyro sensor. It was observed that after reaching the destination a

35 couple times in the first 4 episodes, the agent began traversing the optimal path to reach the destination from the 5 th episode. At the end of 10 episodes the algorithm generates an exact map of the environment as shown in Figure 5.4 including the obstacles placed at (0,1), (2,1). The map also shows the optimal path to reach the destination as estimated by the algorithm. Figure 5.5 shows the plot of number of steps taken by the agent in each episode in the hardware implementation. Figure 5.4: Map generated in hardware implementation Figure 5.5: Plot of number of steps in each episode - hardware

36 Table 5.1: Final Q-Table – Optimal Path Length of final q-table = 3 0 1 2 3 [40.0,0.0] 0.0 0.000008 0.00000 0.00 [40.0,40.0] 0.0 0.000882 -0.01000 -0.01000 [40.0,80.0] 0.0 0.000000 0.04901 0.00 Table 5.2: Full Q-Table Length of full q-table = 8 0 1 2 3 [0.0,0.0] 0.00 -0.010000 3.615840e-08 0.00 Obstacle 0.00 0.000000 0.000000e+00 0.00 [40.0,0.0] 0.00 0.000008 0.000000e+00 0.00 [80.0,0.0] 0.00 -0.010000 0.000000e+00 0.00 [40.0,40.0] 0.00 0.000882 -1.000000e- 02 -0.01 [40.0,80.0] 0.00 0.000000 4.900995e-02 0.00 [0.0,80.0] -0.01 0.000000 2.673090e-04 0.00 Goal 0.00 0.000000 0.000000e+00 0.00

37 Figure 5.6: Shortest route Figure 5.6 shows the shortest path determined by the algorithm. This is as shown in Figure 5.4 based on the pixel values. (Each cell is shown by 40 pixels in the map). Table 5.1 shows the final Q-table showing the Q-values of the states in the final optimal route to the destination. Table 5.2 shows the full Q-table with the Q-values for all the states explored by the agent. It is observed that at any given state the Q-value corresponding to an action that would lead to an obstacle in the next state is negative and the action that would allow the agent to move closer to the destination or to the destination itself is positive. By this way, the agent chooses which action to take at each state and is able to identify the optimal path to reach the destination while avoiding obstacles.

38 CHAPTER 6 CONCLUSION AND FUTURE WORKS

its obstacles

the 3

or

REFRENCES 1.

65%	MATCHING BLOCK 1/36	SA	RAPIG-D.pdf (D138452381)

REINFORCEMENT LEARNING BASED AUTOMATED PATH PLANNING IN GARDEN ENVIRONMENT USING DEPTH – RAPIG-D A Project Report Submitted By: SATHIYA MURTHI S (2018504604) PRANAV BALAKRISHNAN (2018504581) C ROSHAN ABRAHAM (2018504591)

62%	MATCHING BLOCK 2/36	SA	RAPIG-D.pdf (D138452381)

Reinforcement Learning based Automated Path Planning in Garden environment using Depth” is the bonafide work of SATHIYA MURTHI S (2018504604), PRANAV BALAKRISHNAN (2018504581) & C ROSHAN ABRAHAM (2018504591)

100%	MATCHING BLOCK 3/36	SA	RAPIG-D.pdf (D138452381)

learning based method that can allow an agent to learn and determine the optimal solution to a problem on its own based on the rewards and penalties offered by the environment. This allows for the agent to handle any state that it might encounter while learning to achieve the required results. SARSA or State-Action-Reward-State Action is a basic Reinforcement Learning Algorithm, an On Policy technique. With increasing innovations and implementations

	MATCHING BLOCK 4/36	SA	RAPIG-D.pdf (D138452381)

Self-driving vehicles, UAVs there rises a requirement of adaptive mapping of unknown environments and the ability to determine the shortest path to any given destination in order to handle unpredictable and dynamic environments. Reinforcement Learning is an effective method that can be used to satisfy the requirements of such an environment by learning from repeated iterations of trials. In this paper, this implementation has been employed to plan routes and identify obstacles in a garden environment which can be used to automate management and care taking of gardens and plants. This technique can further be extended to exploration and searching tasks such as Space exploration and mapping, Air crash and accident investigations. 1.1

87%	MATCHING BLOCK 5/36	SA	RAPIG-D.pdf (D138452381)

Dr.V. SATHIESH KUMAR Assistant Professor and Guide Department of Electronics Engineering Madras Institute of Technology Anna University Chennai - 600044 4 5 ABSTRACT Path planning by employing Reinforcement Learning is a versatile implementation that can account for the ability of a robot to autonomously map any unknown environment. In this paper, such a hardware implementation is proposed and tested by making use of the SARSA algorithm for path planning and by utilizing stereovision for depth estimation based obstacle detection. The robot is tested in cell-based environments – 3x3 with 2 obstacles. The goal is to map the environment by detecting and mapping the obstacles and finding the ideal route to the destination. The robot starts at one end of the environment runs through it for a specified number of episodes and it is observed that the robot can accurately identify and map obstacles and find the shortest path to the destination in under 10 episodes. Currently the destination is a fixed point and is taken as the other diagonal end of the environment.

84%	MATCHING BLOCK 6/36	W	https://www.finsliqblog.com/ai-and-machine-lea ...

technique that enables an agent to learn in an interactive environment by trial

68%	MATCHING BLOCK 7/36	W	https://www.finsliqblog.com/ai-and-machine-lea ...

reinforcement learning use mapping between input and output, unlike supervised learning where the feedback provided to the agent is correct set of actions for performing a task, reinforcement learning uses rewards and punishments as signals for positive and negative behavior. As compared to unsupervised learning, reinforcement learning is different in terms of goals. While the goal in unsupervised learning is to find similarities and differences between data points, in

61%	MATCHING BLOCK 8/36	W	https://www.finsliqblog.com/ai-and-machine-lea ...

key terms that describe the basic elements of an RL problem are: Environment — Physical world in which

88%	MATCHING BLOCK 9/36	W	https://www.finsliqblog.com/ai-and-machine-lea ...

agent operates State — Current situation of the agent Reward — Feedback from the environment

71%	MATCHING BLOCK 10/36	W	https://www.finsliqblog.com/ai-and-machine-lea ...

reinforcement learning the goal is to find a suitable action model that would maximize

85%	MATCHING BLOCK 11/36	W	https://www.datacamp.com/tutorial/introduction ...

REINFORCEMENT LEARNING There are three primary approaches to implement a Reinforcement learning algorithm. They are 13 • Value-based •

100%	MATCHING BLOCK 12/36	W	https://www.guru99.com/reinforcement-learning- ...

policy π. Policy-based: In a policy-based RL method, you try to come up with such a policy that the action performed in every state helps you to gain maximum reward in the future.

90%	MATCHING BLOCK 13/36	W	https://www.guru99.com/reinforcement-learning- ...

Model-Based: In this Reinforcement Learning method, you need to create a virtual model for each environment. The agent learns to perform in that specific environment. For any reinforcement learning

95%	MATCHING BLOCK 14/36	SA	RAPIG-D.pdf (D138452381)

To discover the path between source and destination, Konor et al. reported on an enhanced Q-learning approach. The step distance (from one state to the next) and the eventual destination are assumed here. It is used to update the entries in the Q-table. Unlike the traditional Q-learning approach, where the values are continually updated, the values are only entered once. At each state, the Q-value derived for the best action is saved. In terms of traversal time and the number of states traversed, performance tends to increase. [8] An end-to-end mobile robot path planning using deep reinforcement learning

97%	MATCHING BLOCK 15/36	SA	RAPIG-D.pdf (D138452381)

Path planning is done out using a Q-learning algorithm based on the Markov Decision Process, according to Sichkar et al. [11] It is challenging to find an optimal path in complicated situations using the traditional Q-learning approach. The robot determined/identified an optimal path from the source to the destination by avoiding collisions with impediments in its propagation path, according to the authors. The shortest path between the source and destination is determined using Q-learning and SARSA algorithms. The method has been tested in a simulated environment with preset barriers. Different 19 learning periods are included in the two algorithms used. It also fluctuates in the number of steps it takes to get to its objective by avoiding collisions with objects along the route. [10] The shortest path between the source and destination cannot be found using traditional Breadth First Search (BFS) or Rapidly Exploring Random Trees (RRT) techniques. As a result, the authors designed and showed a path planning algorithm based on reinforcement learning. To begin, a random route graph is chosen. If the chosen path has barriers, it is not taken into account. A collision- free route is found using the Q-learning approach. When compared to RRT and BFS algorithms, the suggested approach provided a smooth and quickest path, according to the authors. [12]

100%	MATCHING BLOCK 16/36	SA	RAPIG-D.pdf (D138452381)

Based on a thorough review of the literature, it has been determined that path planning Reinforcement learning is still in its infancy. The algorithm may be fine- tuned or improved further so that it can be used in real-time situations. The use of the Reinforcement learning algorithm in connection to path planning in an unknown environment is investigated in this work. In the process of picking a suitable action by the robot in an unfamiliar environment, the SARSA algorithm is applied. 21

100%	MATCHING BLOCK 17/36	SA	Rapport_PFE_Nour_Bejaoui.pdf (D110250487)

NVIDIA JETSON NANO NVIDIA Jetson Nano Developer Kit is a

100%	MATCHING BLOCK 18/36	SA	RAPIG-D.pdf (D138452381)

To perform hardware implementation, the NVIDIA Jetson Nano Developer Kit (

100%	MATCHING BLOCK 19/36	SA	RAPIG-D.pdf (D138452381)

is used as the Micro-controller to run the developed algorithm and interface with the

100%	MATCHING BLOCK 20/36	SA	RAPIG-D.pdf (D138452381)

required sensors for calibrated movement from one cell to the next in the environment.

100%	MATCHING BLOCK 21/36	SA	RAPIG-D.pdf (D138452381)

is used to accurately turn by integrating the angular velocity output of the sensor to obtain angular displacement. A Li-Po battery is used to power four 12V 100 RPM DC motors

100%	MATCHING BLOCK 22/36	SA	RAPIG-D.pdf (D138452381)

The Jetson Nano and the Kinect camera are currently externally powered using 5V4A and 12V2A AC adapters for testing purposes. 4.3.1

87%	MATCHING BLOCK 23/36	SA	RAPIG-D.pdf (D138452381)

Both the software simulation and the hardware implementation follow the same algorithm except for the additional implementation of Obstacle detection using depth estimation in hardware implementation as opposed to feeding obstacle co-ordinates in the simulation.

100%	MATCHING BLOCK 24/36	SA	RAPIG-D.pdf (D138452381)

The agent then checks if the next state is an empty space, an obstacle or the destination. This is then used to populate the map of the environment. A Q-table is initialized with zeros at the start of the first episode which is used to determine how valuable it is to take a particular action at a particular state so as to maximize reward received. The Q-table is updated according to the reward offered for the next state for each action at the current state. A reward of 1 is offered if the next state is the destination, 0 if the next state is an empty space and -1 if it is an obstacle. In each episode, the agent goes around the environment, exploring until it reaches either an obstacle or the destination. If the agent reaches the destination, the path from start point is recorded to check if it is the shortest path it has taken to the destination so far. This is used to identify the optimal path to the destination. The optimal path can then be used to traverse to the destination as and when required. 31 4.3.6

90%	MATCHING BLOCK 25/36	SA	RAPIG-D.pdf (D138452381)

SOFTWARE IMPLEMENTATION Software implementation was performed on a larger 10 x 10 environment with 10 obstacles and was run for 500 episodes. Software implementation takes less time per episode.

100%	MATCHING BLOCK 26/36	SA	RAPIG-D.pdf (D138452381)

The obstacle co-ordinates are pre-determined and fed to the algorithm. The Reinforcement Learning agent moves through environment based on the policy and checks if each co-ordinate is an obstacle, empty space or the destination. By this way, every time it reaches the destination the shortest path is updated allowing the agent to identify the optimal path to the destination while avoiding obstacles.

78%	MATCHING BLOCK 27/36	SA	RAPIG-D.pdf (D138452381)

Thus, it is seen that the robot agent successfully performs Reinforcement Learning based path planning to accurately map the environment and

100%	MATCHING BLOCK 28/36	SA	RAPIG-D.pdf (D138452381)

and identify the optimal path to reach a given destination. In

96%	MATCHING BLOCK 29/36	SA	RAPIG-D.pdf (D138452381)

x3 environments, the robot performs accurate obstacle detection using depth estimation and by employing the SARSA algorithm, finds the optimal path to the destination within 10 episodes. As the obstacles and the size of the environment increases the time required to identify the optimal path would also increase.

100%	MATCHING BLOCK 30/36	SA	RAPIG-D.pdf (D138452381)

Thus, a simple Reinforcement Learning algorithm such as SARSA is used to identify a suitable path and is able to allow the agent to map an unknown environment.

86%	MATCHING BLOCK 31/36	SA	RAPIG-D.pdf (D138452381)

This implementation can further be extended to environments of larger scale and even dynamic environments. Future works may involve increasing the degrees of freedom of the environment states and going beyond cell-based environments. It can also be employed in diversified applications ranging from garden

100%	MATCHING BLOCK 32/36	SA	RAPIG-D.pdf (D138452381)

warehouse management, serving food at restaurants, Air crash investigations, Search and Rescue operations and even Space exploration. 39

100%	MATCHING BLOCK 33/36	SA	RAPIG-D.pdf (D138452381)

S. Zheng and H. Liu, “Improved Multi-Agent Deep Deterministic Policy Gradient for Path Planning-Based Crowd Simulation” in IEEE Access, vol. 7, pp. 147755-147770, 2019. 2. Xinyuan Zhou, Peng Wu, Haifeng Zhang, Weihong Guo and Yuanchang Liu, “Learn to Navigate: Cooperative Path Planning for Unmanned Surface Vehicles Using Deep Reinforcement Learning”, IEEE Access, vol. 7, 2019 3. Chen Chen, Jiange Jiang, Ning Lv and Siyu Li, “An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge”, IEEE Access, vol. 8, 2020. 4.

100%	MATCHING BLOCK 34/36	SA	RAPIG-D.pdf (D138452381)

Chao Wang, Jian Wang, SeYuan Shen, and Xudong Zhang,” Autonomous Navigation of UAVs in Large-Scale Complex Environments: A Deep Reinforcement Learning Approach”, IEEE Transactions On Vehicular Technology, vol. 68, pp. 2124-2136, March 2019. 5. D. Ebrahimi, S. Sharafeddine, P. -H. Ho, and C. Assi, “Autonomous UAV Trajectory for Localizing Ground Objects: A Reinforcement Learning Approach,” in IEEE Transactions on Mobile Computing, vol. 20, pp. 1312- 1324, 1 April 2021. 6.

97%	MATCHING BLOCK 35/36	SA	RAPIG-D.pdf (D138452381)

B. Wang, Z. Liu, Q. Li, and A. Prorok, “Mobile Robot Path Planning in Dynamic Environments Through Globally Guided Reinforcement Learning,” in IEEE Robotics and Automation Letters, vol. 5, pp. 6932- 6939, Oct. 2020. 7. A. Konar, I. Goswami Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, “A Deterministic Improved Q-Learning for Path Planning of a Mobile Robot,” in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, pp. 1141-1153, Sept. 2013 8. J. Xin, H. Zhao, D. Liu, and M. Li, “Application of deep reinforcement learning in mobile robot path planning,” In: Chinese Automation Congress 40 (CAC), 2017. 9. V. N. Sichkar, “Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot,” In: International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 2019, pp. 1-5 10. P. Gao, Z. Liu, Z. Wu, and D. Wang, “A Global Path Planning Algorithm for Robots Using Reinforcement Learning,” In: IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019. 11. Y. Long, and H. He, “Robot path planning based on deep reinforcement learning,” In: IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), 2020 12. P. Mohan, L. Sharma and P. Narayan, “Optimal Path Finding using Iterative SARSA,” In: 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 811-817. 13. Valentyn N. Sichkar, “Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot,” In: International Conference on Industrial Engineering, Applications and Manufacturing, 2019

95%	MATCHING BLOCK 36/36	W	https://www.techtarget.com/searchenterpriseai/ ...

REINFORCEMENT LEARNING Reinforcement learning is a machine learning technique based on rewarding desired behaviors and/or punishing undesired ones. In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trial and error.

Hit and source - focused comparison, Side by Side

Submitted text As student entered the text in the submitted document.

Matching text As the text appears in the source.

1/36	SUBMITTED TEXT	35 WORDS	65% MATCHING TEXT	35 WORDS

Submitted text	Matching text
REINFORCEMENT LEARNING BASED AUTOMATED PATH PLANNING IN GARDEN ENVIRONMENT USING DEPTH – RAPIG-D A Project Report Submitted By: SATHIYA MURTHI S (2018504604) PRANAV BALAKRISHNAN (2018504581) C ROSHAN ABRAHAM (2018504591)	Reinforcement Learning based Automated Path Planning in Garden environment using Depth RAPiG-D S. Sathiya Murthi, Pranav Balakrishnan, C Roshan Abraham,

SA RAPIG-D.pdf (D138452381)

2/36	SUBMITTED TEXT	29 WORDS	62% MATCHING TEXT	29 WORDS

Submitted text	Matching text
Reinforcement Learning based Automated Path Planning in Garden environment using Depth" is the bonafide work of SATHIYA MURTHI S (2018504604), PRANAV BALAKRISHNAN (2018504581) & C ROSHAN ABRAHAM (2018504591)	Reinforcement Learning based Automated Path Planning in Garden environment using Depth RAPiG-D S. Sathiya Murthi, Pranav Balakrishnan, C Roshan Abraham,

SA RAPIG-D.pdf (D138452381)

3/36	SUBMITTED TEXT	72 WORDS	100% MATCHING TEXT	72 WORDS

Submitted text	Matching text
learning based method that can allow an agent to learn and determine the optimal solution to a problem on its own based on the rewards and penalties offered by the environment. This allows for the agent to handle any state that it might encounter while learning to achieve the required results. SARSA or State-Action-Reward-State Action is a basic Reinforcement Learning Algorithm, an On Policy technique. With increasing innovations and implementations	learning based method that can allow an agent to learn and determine the optimal solution to a problem on its own based on the rewards and penalties offered by the environment. This allows for the agent to handle any state that it might encounter while learning to achieve the required results. SARSA or State-Action-Reward-State-Action is a basic Reinforcement Learning Algorithm, an On Policy technique. With increasing innovations and implementations

SA RAPIG-D.pdf (D138452381)

4/36	SUBMITTED TEXT	118 WORDS	98% MATCHING TEXT	118 WORDS
	Self-driving vehicles, UAVs there rises a requirement of adaptive mapping of unknown environments and the ability to determine the shortest path to any given destination in order to handle unpredictable and dynamic environments. Reinforcement Learning is an effective method that can be used to satisfy the requirements of such an environment by learning from repeated iterations of trials. In this paper, this implementation has been employed to plan routes and identify obstacles in a garden environment which can be used to automate management and care taking of gardens and plants. This technique can further be extended to exploration and searching tasks such as Space exploration and mapping, Air crash and accident investigations. 1.1		Self- driving vehicles, there rises a requirement of adaptive mapping of unknown environments and also the ability to determine the shortest path to any given destination in order to handle unpredictable and dynamic environments. Reinforcement Learning is an effective method that can be used to satisfy the requirements of such an environment by learning from repeated iterations of trials. In this paper, this implementation has been employed to plan routes and identify obstacles in a garden environment which can be used to automate management and care taking of gardens and plants. This technique can further be extended to exploration and searching tasks such as Space exploration and mapping, Air crash and accident investigations.	

SA RAPIG-D.pdf (D138452381)

5/36	SUBMITTED TEXT	181 WORDS	87% MATCHING TEXT	181 WORDS
	Dr.V. SATHIESH KUMAR Assistant Professor and Guide Department of Electronics Engineering Madras Institute of Technology Anna University Chennai - 600044 4 5 ABSTRACT Path planning by employing Reinforcement Learning is a versatile implementation that can account for the ability of a robot to autonomously map any unknown environment. In this paper, such a hardware implementation is proposed and tested by making use of the SARSA algorithm for path planning and by utilizing stereovision for depth estimation based obstacle detection. The robot is tested in cell-based environments – 3x3 with 2 obstacles. The goal is to map the environment by detecting and mapping the obstacles and finding the ideal route to the destination. The robot starts at one end of the environment runs through it for a specified number of episodes and it is observed that the robot can accurately identify and map obstacles and find the shortest path to the destination in under 10 episodes. Currently the destination is a fixed point and is taken as the other diagonal end of the environment.		Dr. V. Sathiesh Kumar. Department of Electronics Engineering, MIT Campus, Anna University, Chennai- 600044. gmail.com, bk@outlook.charlroshan@gmail.Abstract — Path planning by employing Reinforcement Learning is a versatile implementation that can account for the ability of a robot to autonomously map any unknown environment. In this paper, such a hardware implementation is proposed and tested by making use of the SARSA algorithm for path planning and by utilizing stereo-vision for depth estimation based obstacle detection. The robot is tested in cell-based environments – 3x3 and 4x4 with 2 and 3 obstacles respectively. The goal is to map the environment by detecting and mapping the obstacles and finding the ideal route to the destination. The robot starts at one end of the environment runs through it for a specified number of episodes and it is observed that the robot can accurately identify and map obstacles and find the shortest path to the destination in under 20 episodes. Currently the destination is a fixed point and is taken as the other diagonal end of the environment.	

SA RAPIG-D.pdf (D138452381)

6/36	SUBMITTED TEXT	14 WORDS	84% MATCHING TEXT	14 WORDS
	technique that enables an agent to learn in an interactive environment by trial		technique that enables an agent to learn in a competitive & interactive environment by trial &	

W https://www.finsliqblog.com/ai-and-machine-learning/what-are-the-types-of-reinforcement-learning- ...

7/36	SUBMITTED TEXT	70 WORDS	68% MATCHING TEXT	70 WORDS
	reinforcement learning use mapping between input and output, unlike supervised learning where the feedback provided to the agent is correct set of actions for performing a task, reinforcement learning uses rewards and punishments as signals for positive and negative behavior. As compared to unsupervised learning, reinforcement learning is different in terms of goals. While the goal in unsupervised learning is to find similarities and differences between data points, in		Reinforcement & supervised learning methods use mapping between input & output, unlike supervised learning, where feedback provided to the agent is the correct set of actions for completing a task, reinforcement learning uses rewards & punishments as signals for positive & negative behavior. As compared to unsupervised learning – reinforcement learning is quite different in terms of goals. While the goal in supervised learning is to find differences & similarities between data points, in	

W https://www.finsliqblog.com/ai-and-machine-learning/what-are-the-types-of-reinforcement-learning- ...

8/36	SUBMITTED TEXT	19 WORDS	61% MATCHING TEXT	19 WORDS
	key terms that describe the basic elements of an RL problem are: Environment — Physical world in which		key terms that best describe the elements of Reinforcement Learning problems are: • Environment: Physical world in which	

W https://www.finsliqblog.com/ai-and-machine-learning/what-are-the-types-of-reinforcement-learning- ...

9/36	SUBMITTED TEXT	16 WORDS	88% MATCHING TEXT	16 WORDS
	agent operates State — Current situation of the agent Reward — Feedback from the environment		agent operates. • State: It represents the current situation of the agent. • Reward: Feedback from the environment. •	

W https://www.finsliqblog.com/ai-and-machine-learning/what-are-the-types-of-reinforcement-learning- ...

10/36	SUBMITTED TEXT	15 WORDS	71% MATCHING TEXT	15 WORDS
	reinforcement learning the goal is to find a suitable action model that would maximize		reinforcement learning, the main goal is to find an appropriate action model that would maximize	

W https://www.finsliqblog.com/ai-and-machine-learning/what-are-the-types-of-reinforcement-learning- ...

Curiginal

11/36	SUBMITTED TEXT 23 WORDS	85% MATCHING TEXT 23 WORDS
	REINFORCEMENT LEARNING There are three primary approaches to implement a Reinforcement learning algorithm. They are 13 • Value-based •	Reinforcement Learning Works There are majorly three approaches to implement a reinforcement learning algorithm. They are - • Value Based:

W https://www.datacamp.com/tutorial/introduction-reinforcement-learning

12/36	SUBMITTED TEXT 36 WORDS	100% MATCHING TEXT 36 WORDS
	policy π. Policy-based: In a policy-based RL method, you try to come up with such a policy that the action performed in every state helps you to gain maximum reward in the future.	policy π. Policy-based: In a policy-based RL method, you try to come up with such a policy that the action performed in every state helps you to gain maximum reward in the future.

W https://www.guru99.com/reinforcement-learning-tutorial.html

13/36	SUBMITTED TEXT 30 WORDS	90% MATCHING TEXT 30 WORDS
	Model-Based: In this Reinforcement Learning method, you need to create a virtual model for each environment. The agent learns to perform in that specific environment. For any reinforcement learning	Model-Based: In this Reinforcement Learning method, you need to create a virtual model for each environment. The agent learns to perform in that specific environment. Characteristics of Learning

W https://www.guru99.com/reinforcement-learning-tutorial.html

14/36	SUBMITTED TEXT 101 WORDS	95% MATCHING TEXT 101 WORDS
	To discover the path between source and destination, Konor et al. reported on an enhanced Q-learning approach. The step distance (from one state to the next) and the eventual destination are assumed here. It is used to update the entries in the Q-table. Unlike the traditional Q-learning approach, where the values are continually updated, the values are only entered once. At each state, the Q-value derived for the best action is saved. In terms of traversal time and the number of states traversed, performance tends to increase. [8] An end-to-end mobile robot path planning using deep reinforcement learning	To discover the path between source and destination, Konor et al. reported on an enhanced Q- learning approach [12]. The step distance (from one state to the next) and the eventual destination are assumed here. It is used to update the entries in the Q- table. Unlike the traditional Q-learning approach, where the values are continually updated, the values are only entered once. At each state, the Q-value derived for the best action is saved. In terms of traversal time and the number of states traversed, performance tends to increase. [13] describes end-to-end path planning using Deep Reinforcement Learning.

SA RAPIG-D.pdf (D138452381)

15/36	SUBMITTED TEXT	220 WORDS	97%	MATCHING TEXT	220 WORDS

Path planning is done out using a Q-learning algorithm based on the Markov Decision Process, according to Sichkar et al. [11] It is challenging to find an optimal path in complicated situations using the traditional Q-learning approach. The robot determined/identified an optimal path from the source to the destination by avoiding collisions with impediments in its propagation path, according to the authors. The shortest path between the source and destination is determined using Q-learning and SARSA algorithms. The method has been tested in a simulated environment with preset barriers. Different 19 learning periods are included in the two algorithms used. It also fluctuates in the number of steps it takes to get to its objective by avoiding collisions with objects along the route. [10] The shortest path between the source and destination cannot be found using traditional Breadth First Search (BFS) or Rapidly Exploring Random Trees (RRT) techniques. As a result, the authors designed and showed a path planning algorithm based on reinforcement learning. To begin, a random route graph is chosen. If the chosen path has barriers, it is not taken into account. A collision- free route is found using the Q-learning approach. When compared to RRT and BFS algorithms, the suggested approach provided a smooth and quickest path, according to the authors. [12]

Path planning is done out using a Q-learning algorithm based on the Markov Decision Process [14], according to Sichkar et al. It is challenging to find an optimal path in complicated situations using the traditional Q-learning approach. The robot determined/identified an optimal path from the source to the destination by avoiding collisions with impediments in its propagation path, according to the authors. The shortest path between the XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE source and destination is determined using Q-learning and SARSA algorithms [15]. The method has been tested in a simulated environment with preset barriers. Different learning periods are included in the two algorithms used. It also fluctuates in the number of steps it takes to get to its objective by avoiding collisions with objects along the route. The shortest path between the source and destination cannot be found using traditional Breadth First Search (BFS) or Rapidly Exploring Random Trees (RRT) techniques. As a result, the authors designed and showed a path planning algorithm based on reinforcement learning [16]. To begin, a random route graph is chosen. If the chosen path has barriers, it is not taken into account. A collision-free route is found using the Q-learning approach. When compared to RRT and BFS algorithms, the suggested approach provided a smooth and quickest path, according to the authors.

SA RAPIG-D.pdf (D138452381)

16/36	SUBMITTED TEXT	103 WORDS	100%	MATCHING TEXT	103 WORDS

Based on a thorough review of the literature, it has been determined that path planning Reinforcement learning is still in its infancy. The algorithm may be fine- tuned or improved further so that it can be used in real-time situations. The use of the Reinforcement learning algorithm in connection to path planning in an unknown environment is investigated in this work. In the process of picking a suitable action by the robot in an unfamiliar environment, the SARSA algorithm is applied. 21

Based on a thorough review of the literature, it has been determined that path planning Reinforcement learning is still in its infancy. The algorithm may be fine-tuned or improved further so that it can be used in real-time situations. The use of the Reinforcement learning algorithm in connection to path planning in an unknown environment is investigated in this work. In the process of picking a suitable action by the robot in an unfamiliar environment, the SARSA algorithm is applied.

SA RAPIG-D.pdf (D138452381)

17/36	SUBMITTED TEXT	12 WORDS	100% MATCHING TEXT	12 WORDS
	NVIDIA JETSON NANO NVIDIA Jetson Nano Developer Kit is a		Nvidia Jetson Nano NVIDIA® Jetson Nano™ Developer Kit is a	

SA Rapport_PFE_Nour_Bejaoui.pdf (D110250487)

18/36	SUBMITTED TEXT	11 WORDS	100% MATCHING TEXT	11 WORDS
	To perform hardware implementation, the NVIDIA Jetson Nano Developer Kit (		To perform hardware implementation, the NVIDIA Jetson Nano Developer Kit	

SA RAPIG-D.pdf (D138452381)

19/36	SUBMITTED TEXT	15 WORDS	100% MATCHING TEXT	15 WORDS
	is used as the Micro-controller to run the developed algorithm and interface with the		is used as the Micro-controller to run the developed algorithm and interface with the	

SA RAPIG-D.pdf (D138452381)

20/36	SUBMITTED TEXT	15 WORDS	100% MATCHING TEXT	15 WORDS
	required sensors for calibrated movement from one cell to the next in the environment.		required sensors for calibrated movement from one cell to the next in the environment.	

SA RAPIG-D.pdf (D138452381)

21/36	SUBMITTED TEXT	32 WORDS	100% MATCHING TEXT	32 WORDS
	is used to accurately turn by integrating the angular velocity output of the sensor to obtain angular displacement. A Li-Po battery is used to power four 12V 100 RPM DC motors		is used to accurately turn by integrating the angular velocity output of the sensor to obtain angular displacement. A Li-Po battery is used to power four 12V 100 RPM DC motors.	

SA RAPIG-D.pdf (D138452381)

22/36	SUBMITTED TEXT	24 WORDS	100% MATCHING TEXT	24 WORDS

The Jetson Nano and the Kinect camera are currently externally powered using 5V4A and 12V2A AC adapters for testing purposes. 4.3.1

The Jetson Nano and the Kinect camera are currently externally powered using 5V4A and 12V2A AC adapters for testing purposes.

SA RAPIG-D.pdf (D138452381)

23/36	SUBMITTED TEXT	36 WORDS	87% MATCHING TEXT	36 WORDS

Both the software simulation and the hardware implementation follow the same algorithm except for the additional implementation of Obstacle detection using depth estimation in hardware implementation as opposed to feeding obstacle co-ordinates in the simulation.

Both the software simulation and the hardware implementation follow the same algorithm except for XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE the additional implementation of Obstacle detection using depth estimation in hardware implementation as opposed to feeding obstacle co-ordinates in the simulation.

SA RAPIG-D.pdf (D138452381)

24/36	SUBMITTED TEXT	198 WORDS	100% MATCHING TEXT	198 WORDS

The agent then checks if the next state is an empty space, an obstacle or the destination. This is then used to populate the map of the environment. A Q-table is initialized with zeros at the start of the first episode which is used to determine how valuable it is to take a particular action at a particular state so as to maximize reward received. The Q-table is updated according to the reward offered for the next state for each action at the current state. A reward of 1 is offered if the next state is the destination, 0 if the next state is an empty space and -1 if it is an obstacle. In each episode, the agent goes around the environment, exploring until it reaches either an obstacle or the destination. If the agent reaches the destination, the path from start point is recorded to check if it is the shortest path it has taken to the destination so far. This is used to identify the optimal path to the destination. The optimal path can then be used to traverse to the destination as and when required. 31 4.3.6

The agent then checks if the next state is an empty space, an obstacle or the destination. This is then used to populate the map of the environment. A Q-table is initialized with zeros at the start of the first episode which is used to determine how valuable it is to take a particular action at a particular state so as to maximize reward received. The Q-table is updated according to the reward offered for the next state for each action at the current state. A reward of 1 is offered if the next state is the destination, 0 if the next state is an empty space and -1 if it is an obstacle. In each episode, the agent goes around the environment, exploring until it reaches either an obstacle or the destination. If the agent reaches the destination, the path from start point is recorded to check if it is the shortest path it has taken to the destination so far. This is used to identify the optimal path to the destination. The optimal path can then be used to traverse to the destination as and when required.

SA RAPIG-D.pdf (D138452381)

25/36	SUBMITTED TEXT	31 WORDS	90% MATCHING TEXT	31 WORDS

SOFTWARE IMPLEMENTATION Software implementation was performed on a larger 10 x 10 environment with 10 obstacles and was run for 500 episodes. Software implementation takes less time per episode.

Software implementation Software implementation was performed on a larger 9x9 environment with a larger number of obstacles and was run for 500 episodes as software implementation takes less time per episode.

SA RAPIG-D.pdf (D138452381)

26/36	SUBMITTED TEXT	64 WORDS	100% MATCHING TEXT	64 WORDS

The obstacle co-ordinates are pre-determined and fed to the algorithm. The Reinforcement Learning agent moves through environment based on the policy and checks if each co-ordinate is an obstacle, empty space or the destination. By this way, every time it reaches the destination the shortest path is updated allowing the agent to identify the optimal path to the destination while avoiding obstacles.

The obstacle co-ordinates are pre- determined and fed to the algorithm. The Reinforcement Learning agent moves through environment based on the policy and checks if each co- ordinate is an obstacle, empty space or the destination. By this way, every time it reaches the destination the shortest path is updated allowing the agent to identify the optimal path to the destination while avoiding obstacles.

SA RAPIG-D.pdf (D138452381)

27/36	SUBMITTED TEXT	22 WORDS	78% MATCHING TEXT	22 WORDS

Thus, it is seen that the robot agent successfully performs Reinforcement Learning based path planning to accurately map the environment and

Thus, it is seen that the robot agent successfully performs Reinforcement Learning based path planning XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE to accurately map the environment and

SA RAPIG-D.pdf (D138452381)

28/36	SUBMITTED TEXT	12 WORDS	100% MATCHING TEXT	12 WORDS

and identify the optimal path to reach a given destination. In

and identify the optimal path to reach a given destination. In

SA RAPIG-D.pdf (D138452381)

29/36	SUBMITTED TEXT	50 WORDS	96% MATCHING TEXT	50 WORDS

x3 environments, the robot performs accurate obstacle detection using depth estimation and by employing the SARSA algorithm, finds the optimal path to the destination within 10 episodes. As the obstacles and the size of the environment increases the time required to identify the optimal path would also increase.

x4 environments, the robot performs accurate obstacle detection using depth estimation and by employing the SARSA algorithm, finds the optimal path to the destination within 10 episodes and 20 episodes respectively. As the obstacles and the size of the environment increases the time required to identify the optimal path would also increase.

SA RAPIG-D.pdf (D138452381)

30/36	SUBMITTED TEXT	29 WORDS	100% MATCHING TEXT	29 WORDS

Thus, a simple Reinforcement Learning algorithm such as SARSA is used to identify a suitable path and is able to allow the agent to map an unknown environment.

Thus, a simple Reinforcement Learning algorithm such as SARSA is used to identify a suitable path and is able to allow the agent to map an unknown environment.

SA RAPIG-D.pdf (D138452381)

31/36	SUBMITTED TEXT	45 WORDS	86% MATCHING TEXT	45 WORDS

This implementation can further be extended to environments of larger scale and even dynamic environments. Future works may involve increasing the degrees of freedom of the environment states and going beyond cell-based environments. It can also be employed in diversified applications ranging from garden

This can further be extended to environments of larger scale and even dynamic environments. Future works may involve increasing the degrees of freedom of the environment states and going beyond cell-based environments. This path planning implementation can be employed in diversified applications ranging from garden

SA RAPIG-D.pdf (D138452381)

32/36	SUBMITTED TEXT	30 WORDS	100% MATCHING TEXT	30 WORDS

warehouse management, serving food at restaurants, Air crash investigations, Search and Rescue operations and even Space exploration. 39

warehouse management, serving food at restaurants, Air crash investigations, Search and Rescue operations and even Space Exploration

SA RAPIG-D.pdf (D138452381)

33/36	SUBMITTED TEXT	89 WORDS	100%	MATCHING TEXT	89 WORDS

S. Zheng and H. Liu, “Improved Multi-Agent Deep Deterministic Policy Gradient for Path Planning-Based Crowd Simulation” in IEEE Access, vol. 7, pp. 147755-147770, 2019. 2. Xinyuan Zhou, Peng Wu, Haifeng Zhang, Weihong Guo and Yuanchang Liu, “Learn to Navigate: Cooperative Path Planning for Unmanned Surface Vehicles Using Deep Reinforcement Learning”, IEEE Access, vol. 7, 2019 3. Chen Chen, Jiange Jiang, Ning Lv and Siyu Li, “An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge”, IEEE Access, vol. 8, 2020. 4.

S. Zheng and H. Liu, “Improved Multi-Agent Deep Deterministic Policy Gradient for Path Planning-Based Crowd Simulation” in IEEE Access, vol. 7, pp. 147755-147770, 2019. [2] Xinyuan Zhou, Peng Wu, Haifeng Zhang, Weihong Guo and Yuanchang Liu, “Learn to Navigate: Cooperative Path Planning for Unmanned Surface Vehicles Using Deep Reinforcement Learning”, IEEE Access, vol. 7, 2019 [3] Chen Chen, Jiange Jiang , Ning Lv and Siyu Li, “An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge”, IEEE Access, vol. 8, 2020. [4]

SA RAPIG-D.pdf (D138452381)

34/36	SUBMITTED TEXT	70 WORDS	100%	MATCHING TEXT	70 WORDS

Chao Wang, Jian Wang, SeYuan Shen, and Xudong Zhang,” Autonomous Navigation of UAVs in Large-Scale Complex Environments: A Deep Reinforcement Learning Approach”, IEEE Transactions On Vehicular Technology, vol. 68, pp. 2124-2136, March 2019. 5. D. Ebrahimi, S. Sharafeddine, P. -H. Ho, and C. Assi, “Autonomous UAV Trajectory for Localizing Ground Objects: A Reinforcement Learning Approach,” in IEEE Transactions on Mobile Computing, vol. 20, pp. 1312- 1324, 1 April 2021. 6.

Chao Wang, Jian Wang, SeYuan Shen , and Xudong Zhang,” Autonomous Navigation of UAVs in Large- Scale Complex Environments: A Deep Reinforcement Learning Approach”, IEEE Transactions On Vehicular Technology, vol. 68, pp. 2124-2136, March 2019. [8] D. Ebrahimi, S. Sharafeddine, P. -H. Ho, and C. Assi, “Autonomous UAV Trajectory for Localizing Ground Objects: A Reinforcement Learning Approach,” in IEEE Transactions on Mobile Computing, vol. 20, pp. 1312-1324, 1 April 2021. [9]

SA RAPIG-D.pdf (D138452381)

35/36	SUBMITTED TEXT	244 WORDS	97% MATCHING TEXT	244 WORDS

B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile Robot Path Planning in Dynamic Environments Through Globally Guided Reinforcement Learning," in IEEE Robotics and Automation Letters, vol. 5, pp. 6932- 6939, Oct. 2020. 7. A. Konar, I. Goswami Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, "A Deterministic Improved Q-Learning for Path Planning of a Mobile Robot," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, pp. 1141-1153, Sept. 2013 8. J. Xin, H. Zhao, D. Liu, and M. Li, "Application of deep reinforcement learning in mobile robot path planning," In: Chinese Automation Congress 40 (CAC), 2017. 9. V. N. Sichkar, "Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot," In: International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 2019, pp. 1-5 10. P. Gao, Z. Liu, Z. Wu, and D. Wang, "A Global Path Planning Algorithm for Robots Using Reinforcement Learning," In: IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019. 11. Y. Long, and H. He, "Robot path planning based on deep reinforcement learning," In: IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), 2020 12. P. Mohan, L. Sharma and P. Narayan, "Optimal Path Finding using Iterative SARSA," In: 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 811-817. 13. Valentyn N. Sichkar, "Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot," In: International Conference on Industrial Engineering, Applications and Manufacturing, 2019

B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile Robot Path Planning in Dynamic Environments Through Globally Guided Reinforcement Learning," in IEEE Robotics and Automation Letters, vol. 5, pp. 6932-6939, Oct. 2020. [12] A. Konar, I. Goswami Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, "A Deterministic Improved Q-Learning for Path Planning of a Mobile Robot," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, pp. 1141-1153, Sept. 2013 [13] J. Xin, H. Zhao, D. Liu, and M. Li, "Application of deep reinforcement learning in mobile robot path planning," In: Chinese Automation Congress (CAC), 2017. XX.00 ©20XX V. N. Sichkar, "Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot," In: International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 2019, pp. 1-5 [15] P. Gao, Z. Liu, Z. Wu, and D. Wang, "A Global Path Planning Algorithm for Robots Using Reinforcement Learning," In: IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019. [16] Y. Long, and H. He, "Robot path planning based on deep reinforcement learning," In: IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), 2020 [17] P. Mohan, L. Sharma and P. Narayan, "Optimal Path Finding using Iterative SARSA," In: 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 811-817. [18] Valentyn N. Sichkar, "Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot," In: International Conference on Industrial Engineering, Applications and Manufacturing, 2019.

SA RAPIG-D.pdf (D138452381)

36/36	SUBMITTED TEXT	42 WORDS	95% MATCHING TEXT	42 WORDS
	REINFORCEMENT LEARNING Reinforcement learning is a machine learning technique based on rewarding desired behaviors and/or punishing undesired ones. In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trial and error.		reinforcement learning? Reinforcement learning is a machine learning training method based on rewarding desired behaviors and/or punishing undesired ones. In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trial and error.	

W https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning