REINFORCEMENT LEARNING BASED AUTOMATED PATH PLANNING IN GARDEN ENVIRONMENT USING DEPTH - 'RAPiG-D'

ZEROTH REVIEW

- S SATHIYA MURTHI - 2018504604
- PRANAV BALAKRISHNAN - 2018504581
- C ROSHAN ABRAHAM - 2018504591

BE ELECTRONICS AND COMMUNICATION ENGINEERING

GUIDED BY:
DR. V. SATHIESH KUMAR,
ASSISTANT PROFESSOR,
DEPARTMENT OF ELECTRONICS ENGINEERING
MIT CAMPUS, ANNA UNIVERSITY

TABLE OF CONTENTS

INTRODUCTION

- Over the years, robots have been implemented in a wide range of applications and environments.

- In turn, this has led to dealing with a host of environments that are increasingly dynamic and unknown.

- Path planning with reinforcement learning would aid the automation of multiple tasks with minimal human intervention.

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.
- Current methods are limited by their inability to adapt to new environments and varying obstacles.

OBJECTIVES

- To develop a path finding algorithm that finds the best possible path between two paths in an unknown garden environment.
- To perform hardware implementation of reinforcement learning based obstacle mapping and analyze its performance.
- To perform detection of plant species using deep learning.
- To develop an algorithm to explore all possible paths and determine the most efficient one.
- To ensure that the robot avoids all obstacles and unnecessary detours.

LITERATURE SURVEY

AUTHOR NAME	TITLE	YEAR OF PUBLICATION	JOURNAL	DESCRIPTION
Qingbiao Li , Fernando Gama , Alejandro Ribeiro , Amanda Prorok	Graph Neural Networks for Decentralized Multi-Robot Path Planning	2020	IEEE/RSJ International Conference on Intelligent Robots and Systems	• This paper proposed a convolutional neural network (CNN) that extracts adequate features from local observations. • A graph neural network (GNN) that communicates these features among robots. • Model is trained to imitate an expert algorithm and evaluate the method in simulations

LITERATURE SURVEY

AUTHOR NAME	TITLE	YEAR OF PUBLICATION	JOURNAL	DESCRIPTION
Martin Gromniak, Jonas Stenzel	Deep Reinforcement Learning for Mobile Robot Navigation	2019	Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)	• Developed training procedure, set of actions available, suitable state representation, and a reward function. • Evaluated using a simulated real-time environment • The experimental evaluation showed that DRL can be applied successfully to robot navigation.

LITERATURE SURVEY

AUTHOR NAME	TITLE	YEAR OF PUBLICATION	JOURNAL	DESCRIPTION
Jing Xin, Huan Zhao, Ding Liu	Application of Deep Reinforcement Learning in Mobile Robot Path Planning	2019	IEEE robotics and automation letters	• A Deep Q-network (DQN) is designed and trained to approximate the mobile robot state-action value function and Q value corresponding to each possible mobile robot action is determined by the well trained DQN. • The current optimal mobile robot action is selected by the action selection strategy to the goal point while avoiding obstacles ultimately

METHODOLOGY

- **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.

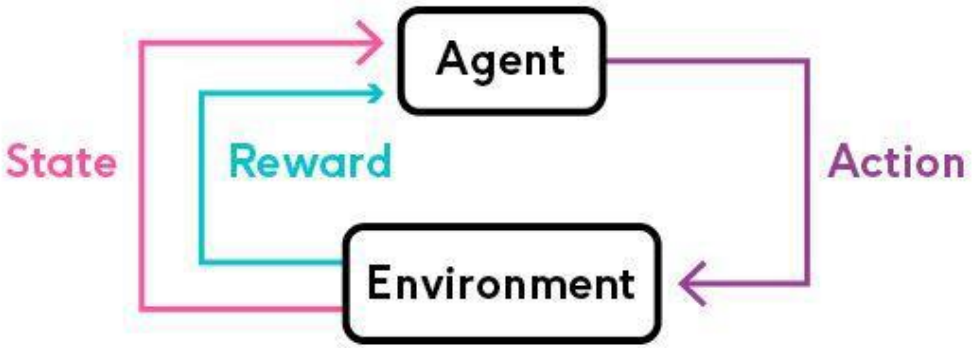


Figure1: SARSA Illustration
Reference: https://medium.com/@vishnuvijayanpv/what-is-reinforcement-learning-e5dc827c8564

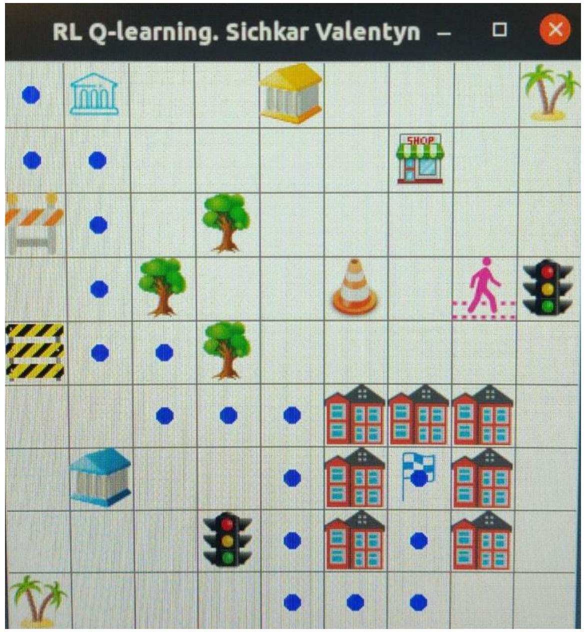


Figure2: SARSA – Software Implementation

METHODOLOGY

- **Stereo vision** is the computation of depth based on the binocular disparity between the images of an object in left and right eyes

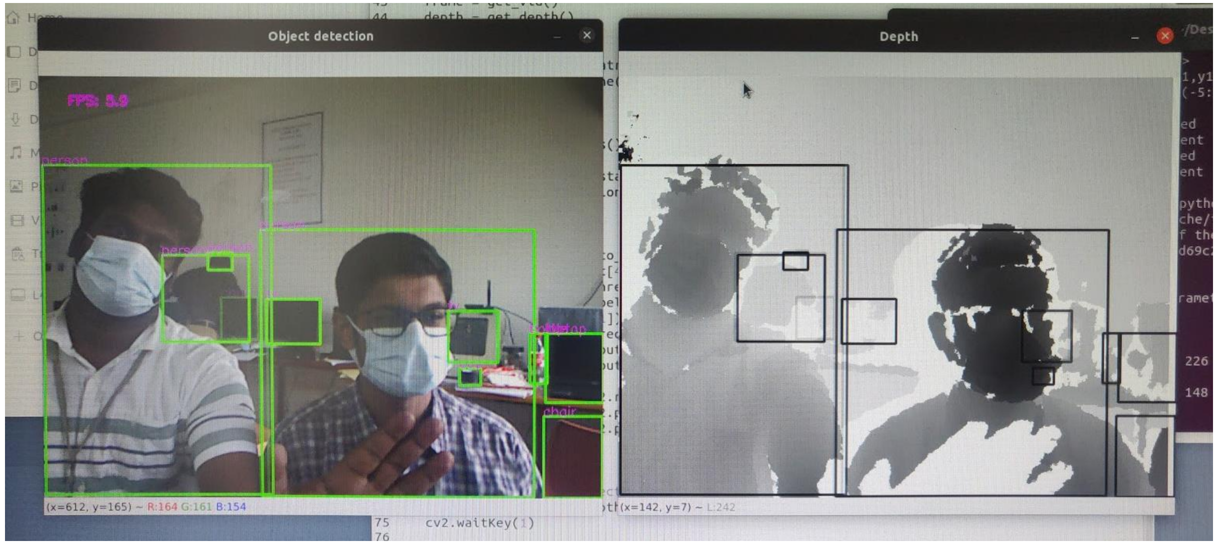
- The **Kinect** camera is a motion sensing input device produced by Microsoft and first released in 2010. The device generally contains RGB cameras, and infrared projectors and detectors that map depth through either structured light or time of flight calculations, which can in turn be used to perform real-time gesture recognition and body skeletal detection, among other capabilities.

- **Image processing** techniques are used to extract and utilize the depth map generated.

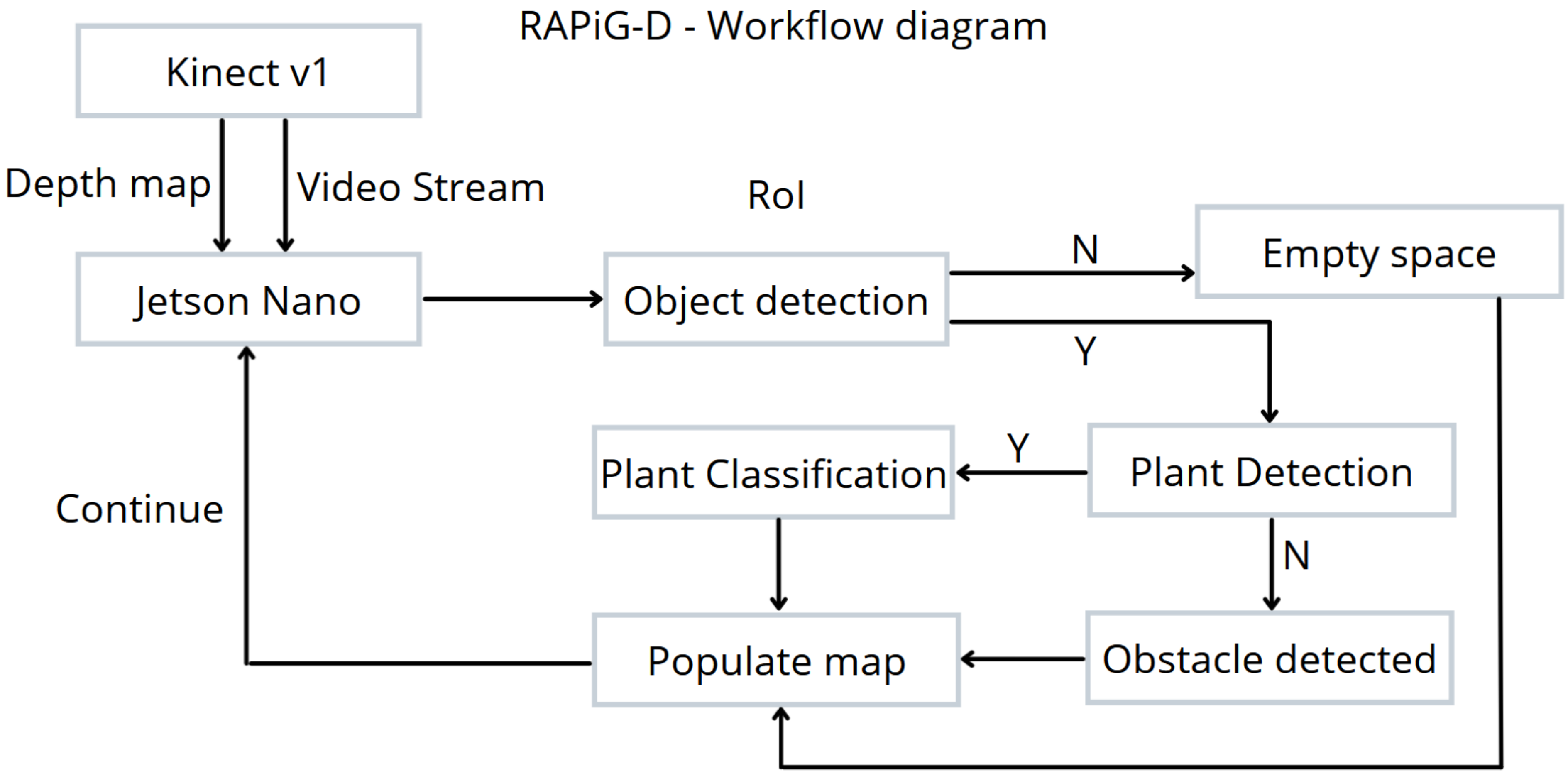
Figure1: XBOX 360 Kinect camera
Reference: https://en.wikipedia.org/wiki/Kinect

METHODOLOGY

- Convolutional neural network (**CNN**) is a class of deep learning neural networks. They're most commonly used to analyze visual imagery and are frequently used in image classification.
- CNNs can therefore be used to detect and classify plants after collection of suitable dataset.
- Models like **YOLOv5** can be trained with custom data for the task



METHODOLOGY

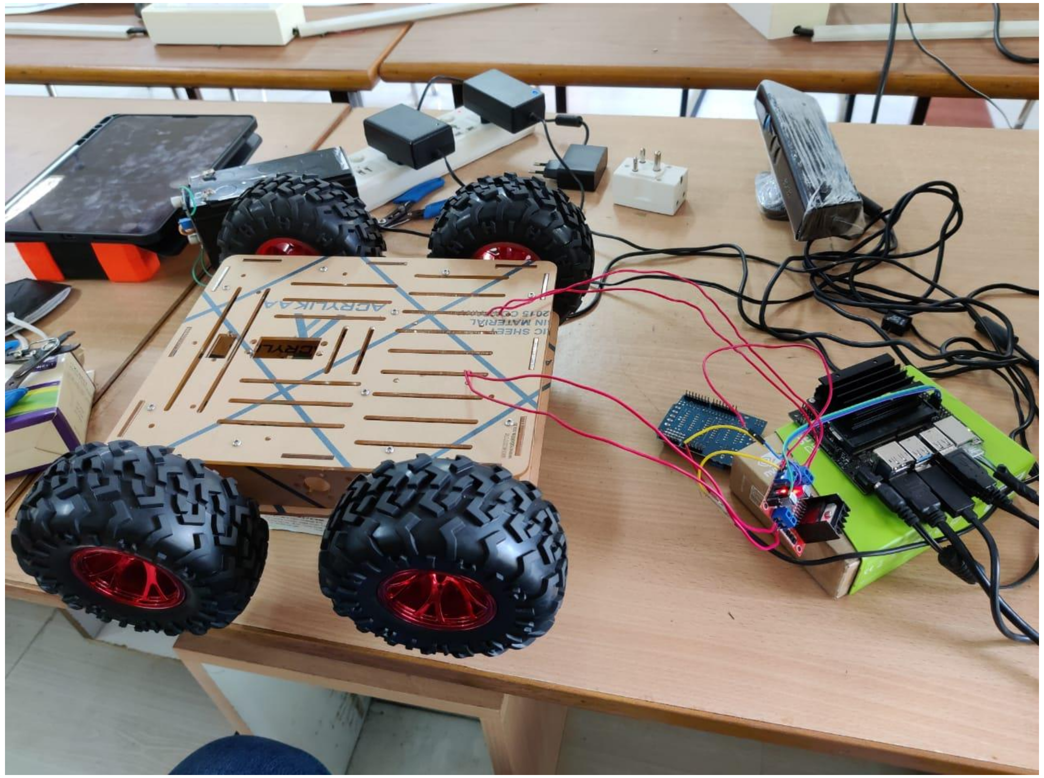


COMPONENTS USED

COMPONENTS	SPECIFICATION	NUMBER USED
Jetson Nano	Developer Kit - 4GB	1
Robot Chassis	Dimensions: 250 x 200 x 46 mm	1
DC motor	100 rpm 12V	2
Wheels	Diameter-130mm Width-60mm	4
Kinect v1 sensor		1
Motor Driver IC	L298N	1
IR sensor	8 channel array	1
Battery	12V7A	1

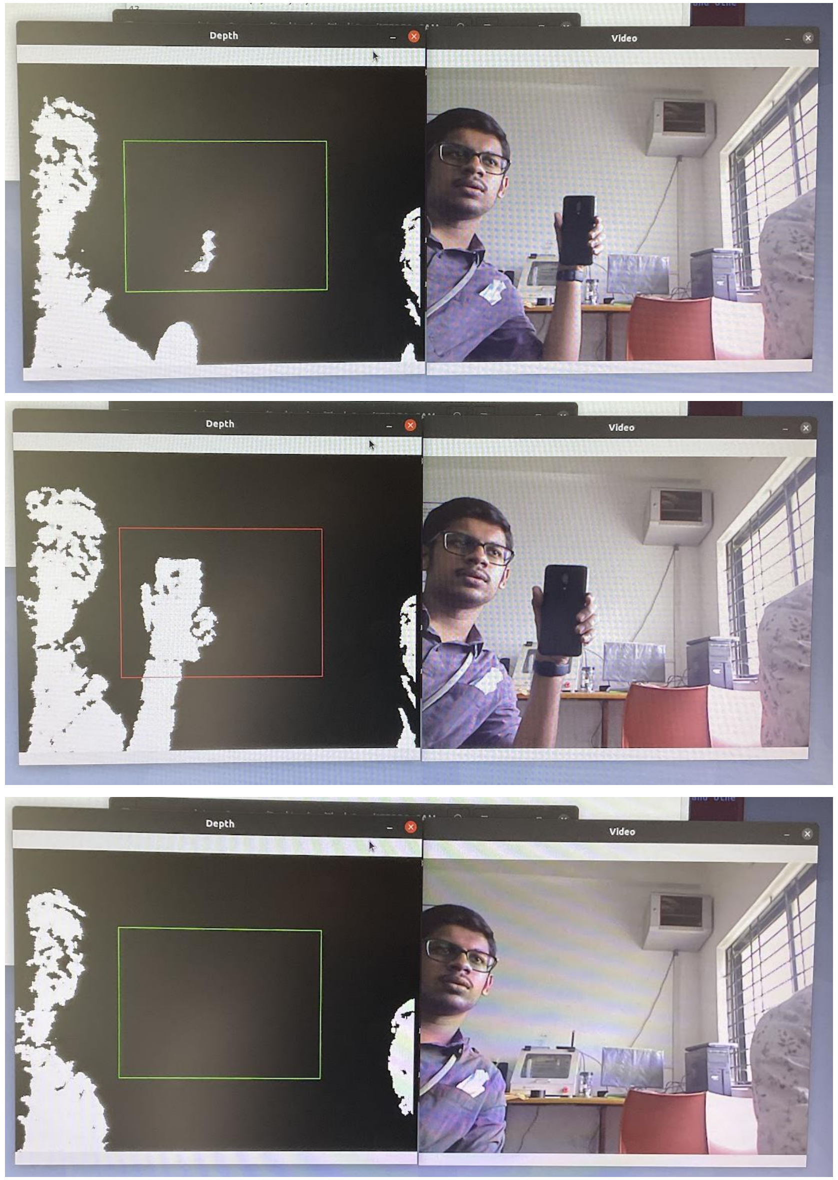
CURRENT PROGRESS

- Tested software implementation of RL SARSA in PC.
- Explored stereo camera and microcontroller options.
- Initialized Jetson NANO and set up bot.



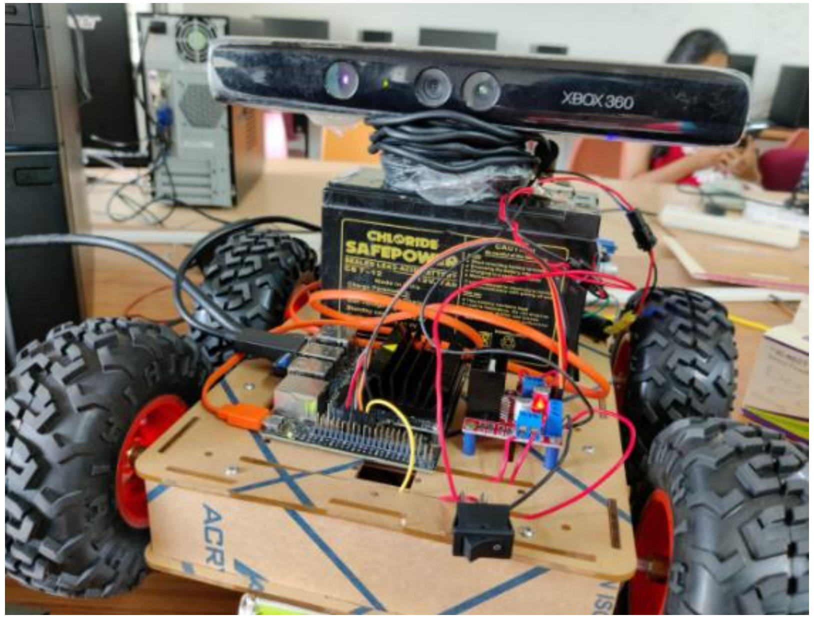
CURRENT PROGRESS

- Interfaced with stereo camera (KINECT v1) on Jetson NANO.
- Obtained depth map and applied depth threshold to identify nearby objects.



CURRENT PROGRESS

- Interfaced with L298N motor driver and enabled four directional movement.
- Set up remote access for Jetson NANO using SSH and VNC.
- Set up suitable power supply for bot portability.



WORK PLAN

First review:	Second Review:	Third review:
• Set up testing space. • Study available and suitable RL algorithms. • Test the performance of the algorithms.	• Collect data set for plant detection model. • Train plant detection model.	• Setup interface for mapping the environment. • Testing and optimizing.

REFERENCES

- Arga Dwi Pambudi,Trihastuti Agustinah and Rusdhianto Effendi,"Reinforcement Point and Fuzzy Input Design of Fuzzy Q-Learning for Mobile Robot Navigation System",International Conference of Artificial Intelligence and Information Technology (ICAIIT),September 2019

- Martin Gromniak and Jonas Stenzel ,"Deep Reinforcement Learning for Mobile Robot Navigation", Asia-Pacific Conference on Intelligent Robot Systems (ACIRS),July 2019

- Guillaume Sartoretti , Justin Kerr , Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset,“PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning”, Ieee robotics and automation letters, , Vol. 4, no. 3, July 2019.

- Hyansu Bae , Gidong Kim , Jonguk Kim , Dianwei Qian 4 and Sukgyu Lee , “Multi-Robot Path Planning Method Using Reinforcement Learning”, Multidisciplinary Digital Publishing Institute, Vol 3,No 4, May 2019.

- Jing Xin,Huan Zhao,Ding Liu,”Application of Deep Reinforcement Learning in Mobile Robot Path Planning”, Ieee robotics and automation letters, Vol 2, No 3 , January 2019.

END OF PRESENTATION

REINFORCEMENT LEARNING BASED AUTOMATED PATH PLANNING IN GARDEN ENVIRONMENT USING DEPTH - 'RAPiG-D'

FIRST REVIEW

- S SATHIYA MURTHI - 2018504604
- PRANAV BALAKRISHNAN - 2018504581
- C ROSHAN ABRAHAM - 2018504591

BE ELECTRONICS AND COMMUNICATION ENGINEERING

GUIDED BY:
DR. V. SATHIESH KUMAR,
ASSISTANT PROFESSOR,
DEPARTMENT OF ELECTRONICS ENGINEERING
MIT CAMPUS, ANNA UNIVERSITY

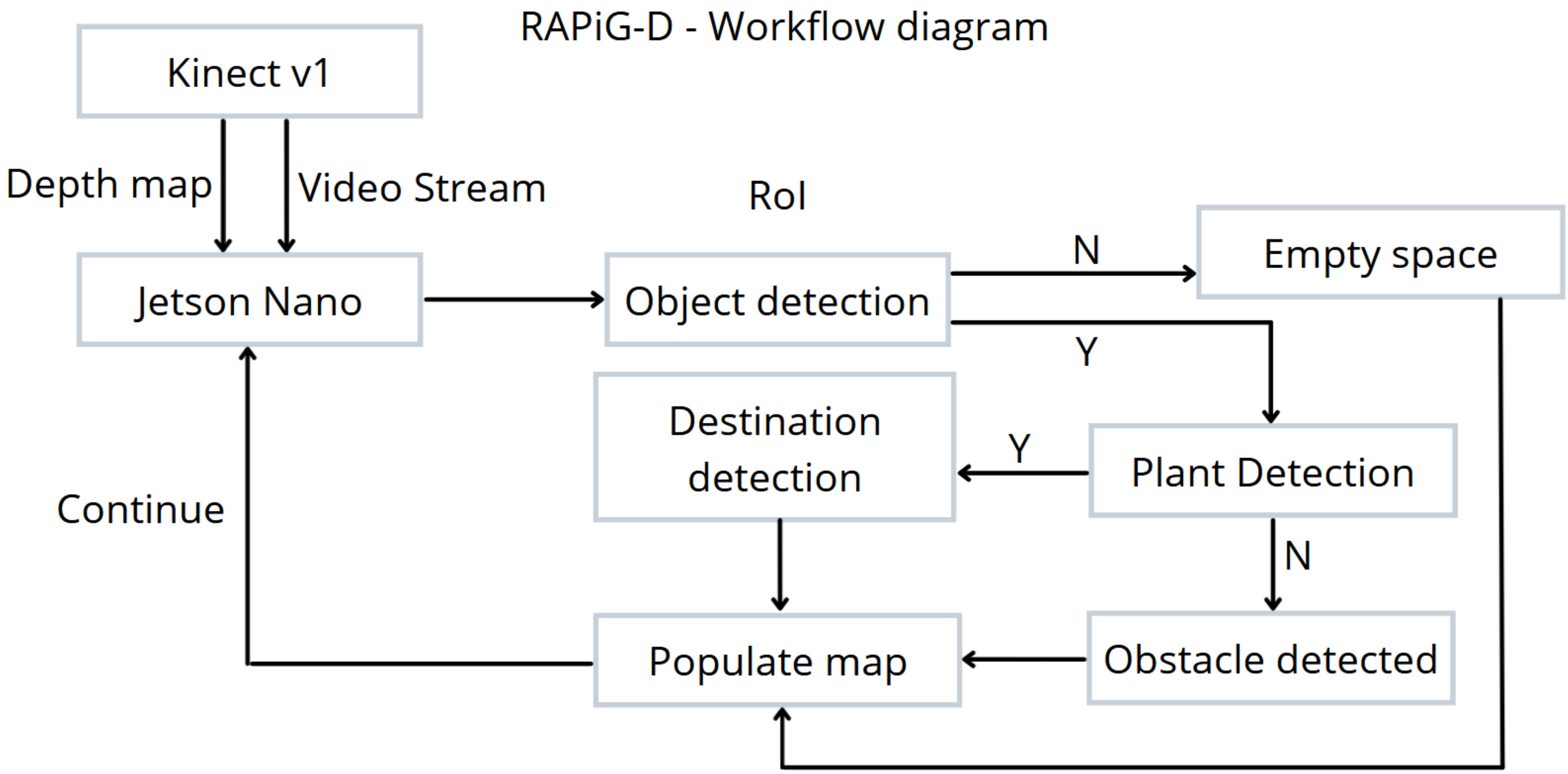
TABLE OF CONTENTS

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.

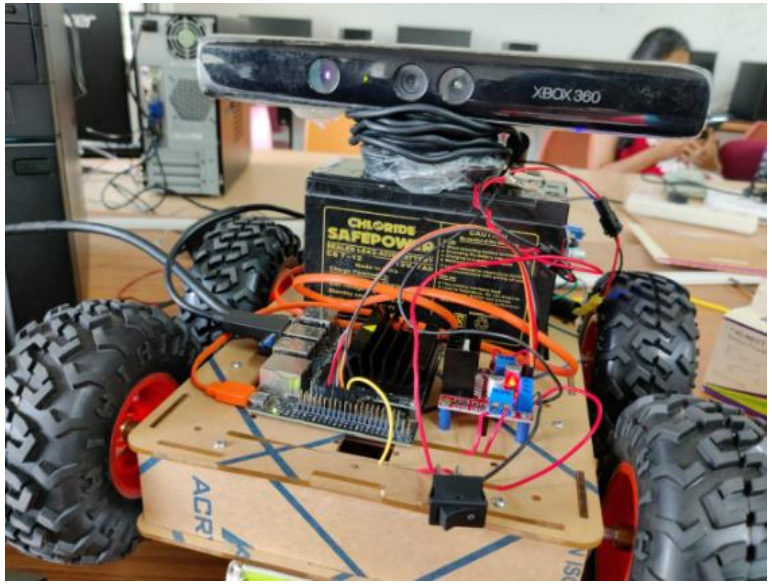
- To estimate the best possible path to reach a given destination.

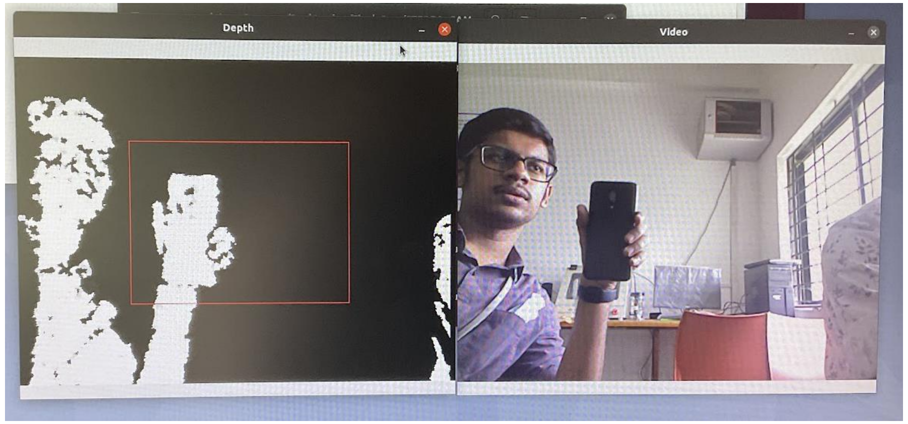
METHODOLOGY



PREVIOUS PROGRESS

- Tested software implementation of RL SARSA in PC.
- Initialised Jetson nano and interfaced with stereo camera (KINECT v1) .
- Obtained depth map and applied depth threshold.
- Interfaced with L298N motor driver and enabled four directional movement.
- Set up basic design of robot.



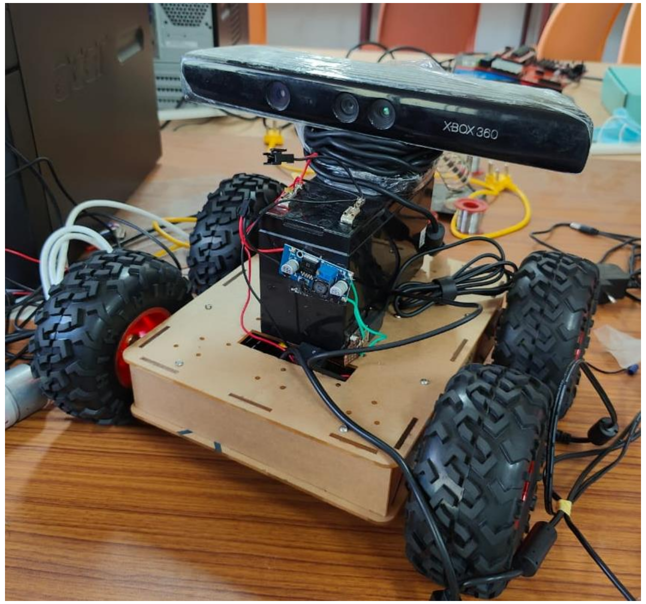


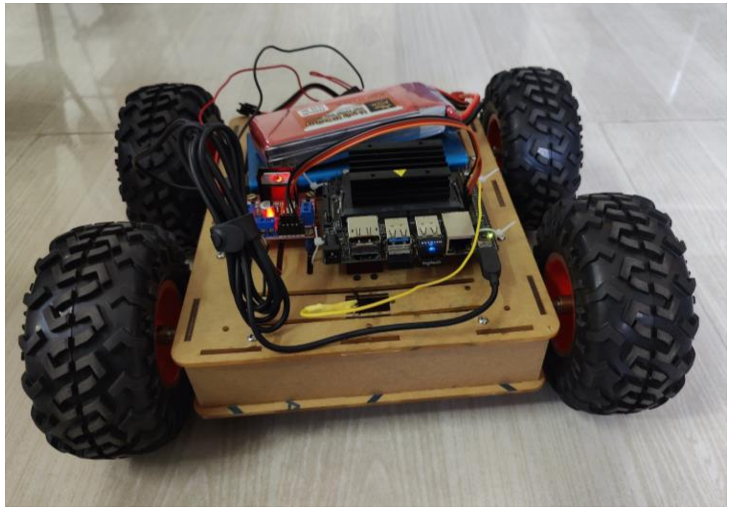
COMPONENTS USED

COMPONENTS	SPECIFICATION	NUMBER USED
Jetson Nano	Developer Kit - 4GB	1
Robot Chassis	Dimensions: 250 x 200 x 46 mm	1
DC motor	100 rpm 12V	2
Wheels	Diameter-130mm Width-60mm	4
Kinect v1 Camera	Stereo camera, IR camera	1
Motor Driver IC	L298N	1
IR sensor	8 channel array	1
Battery	11.1 V 4200mah Li-Po	1

CURRENT PROGRESS

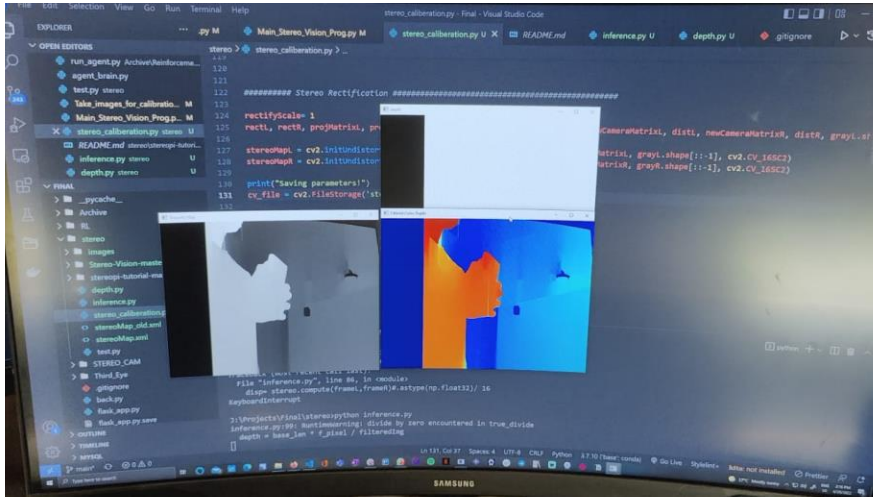
- Tried repositioning Jetson nano and motor driver for better weight distribution.
- Decided to switch to Li-po battery instead of lead acid 12v battery to power the motor driver.
- Set up web output for obstacle detection for remote inference
- Tested mono-camera depth estimation using MIDAS model

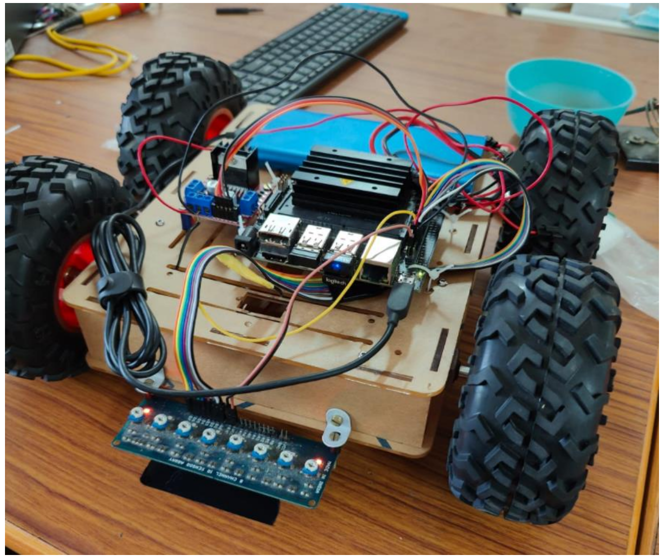




CURRENT PROGRESS

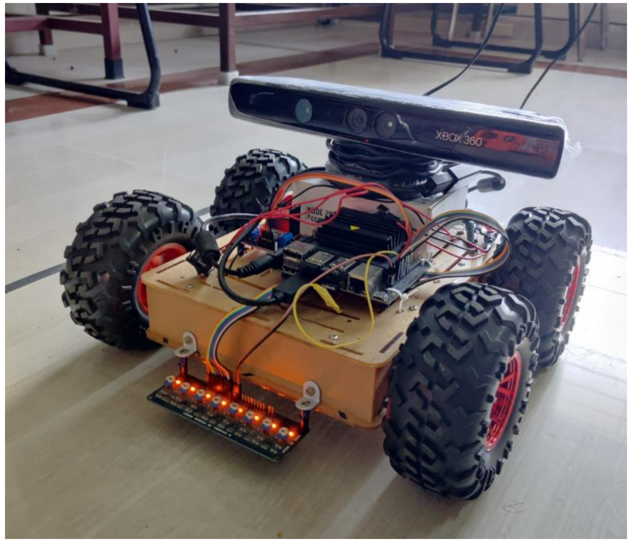
- Mounted and used IR sensor array to enable line following for movement calibration.
- Tested speed control of motor by voltage control using DC-DC converter to improve intersection detection
- Implemented stereo camera calibration and depth estimation with two webcams. Performance was not adequate.





CURRENT PROGRESS

- Switched back to Kinect camera
- Tested lane detection for movement calibration.
- Tried to calibrate turn using line following, timing and detecting line crossing
- Optimised object detection to detect obstacles in next cell then integrated it with line following linear movement to set up obstacle mapping in linear space



PROBLEMS FACED

- Bot movement irregular due to weight and loose shaft
- Unable to VNC remotely without display connected. Requires a Dummy HDMI connection.
- Jetson nano and Kinect camera not functioning properly when used in battery mode – 12V 7A Lead Acid battery
- Stability of bot and calibration is an issue.
- Bot doesn’t turn due to the weight of the 12V lead acid battery
- Jetson Nano turns off when using Kinect camera when powered by power bank
- Bot turning is irregular – Wheels get stuck, skid. Turning inaccurate.

WORK PLAN

Second Review:	Third review:
• Set up RL algorithm to map 2D space • Improve bot stability and calibrate turning	• Setup interface for mapping the environment remotely. • Testing and optimizing.

REFERENCES

- Arga Dwi Pambudi, Trihastuti Agustinah and Rusdhianto Effendi, "Reinforcement Point and Fuzzy Input Design of Fuzzy Q-Learning for Mobile Robot Navigation System",International Conference of Artificial Intelligence and Information Technology (ICAIIT),September 2019

- Martin Gromniak and Jonas Stenzel ,"Deep Reinforcement Learning for Mobile Robot Navigation", Asia-Pacific Conference on Intelligent Robot Systems (ACIRS),July 2019

- Guillaume Sartoretti , Justin Kerr , Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset,“PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning”, Ieee robotics and automation letters, , Vol. 4, no. 3, July 2019.

- Hyansu Bae , Gidong Kim , Jonguk Kim , Dianwei Qian 4 and Sukgyu Lee , “Multi-Robot Path Planning Method Using Reinforcement Learning”, Multidisciplinary Digital Publishing Institute, Vol 3,No 4, May 2019.

- Jing Xin,Huan Zhao,Ding Liu,”Application of Deep Reinforcement Learning in Mobile Robot Path Planning”, Ieee robotics and automation letters, Vol 2, No 3 , January 2019.

END OF PRESENTATION

REINFORCEMENT LEARNING BASED AUTOMATED PATH PLANNING IN GARDEN ENVIRONMENT USING DEPTH - 'RAPiG-D'

SECOND REVIEW

- S SATHIYA MURTHI - 2018504604
- PRANAV BALAKRISHNAN - 2018504581
- C ROSHAN ABRAHAM - 2018504591

BE ELECTRONICS AND COMMUNICATION ENGINEERING

GUIDED BY:
DR. V. SATHIESH KUMAR,
ASSISTANT PROFESSOR,
DEPARTMENT OF ELECTRONICS ENGINEERING
MIT CAMPUS, ANNA UNIVERSITY

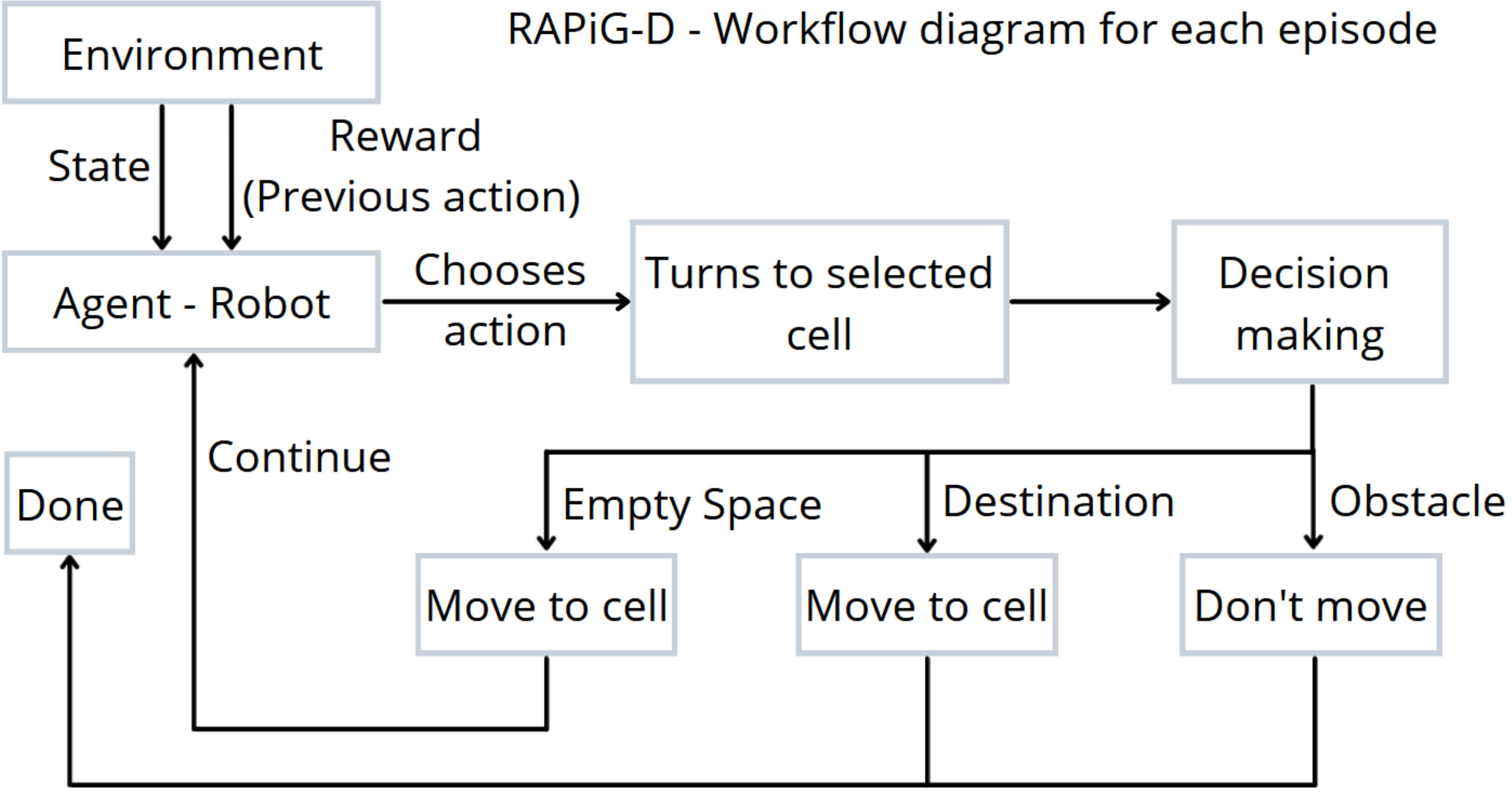
TABLE OF CONTENTS

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.

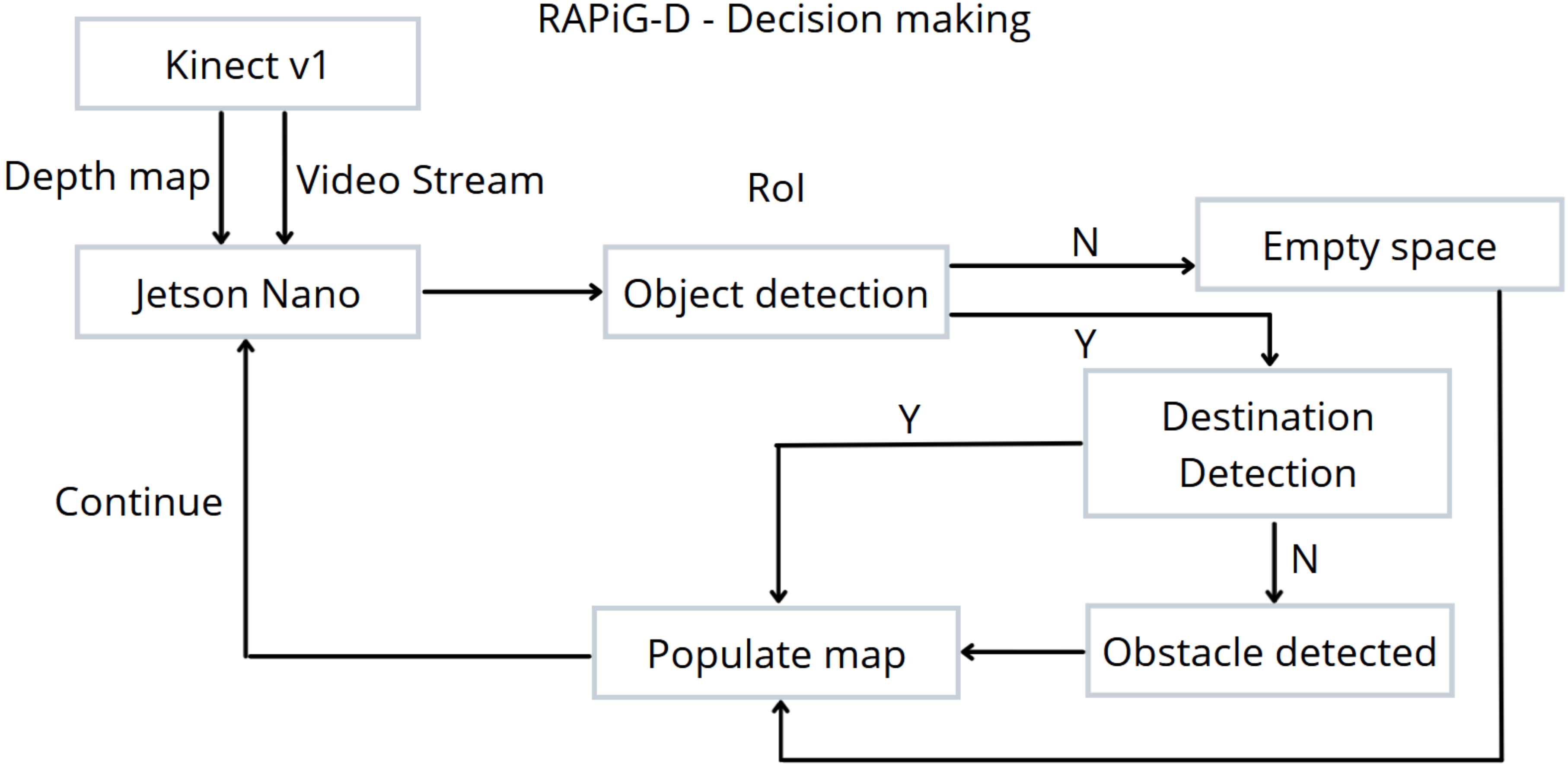
- To estimate the best possible path to reach a given destination.

METHODOLOGY



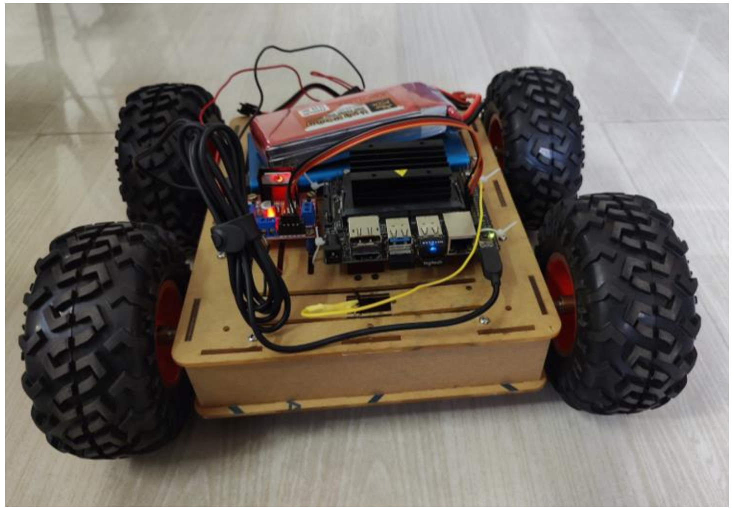
METHODOLOGY

RAPiG-D - Decision making

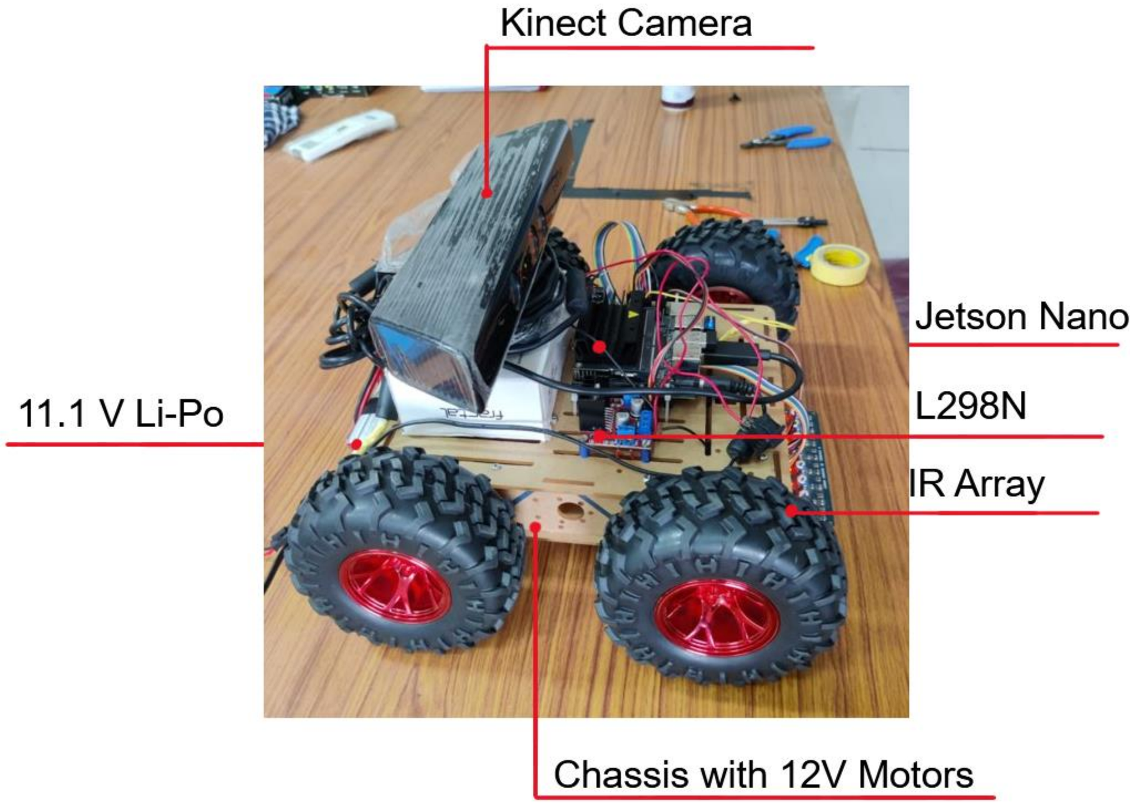


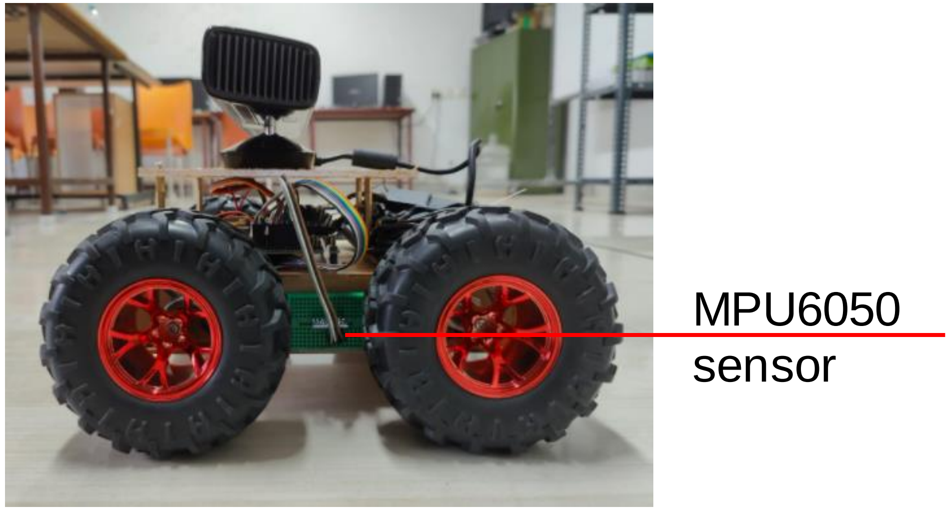
PREVIOUS PROGRESS

- Tried options for better weight distribution.
- Switched to Li-po battery instead of lead acid 12v battery.
- Set up web output for obstacle detection for remote inference
- Mounted and used IR sensor array to enable line following for movement calibration.
- Tried to calibrate turn using line following, timing and detecting line crossing
- Performed mapping of linear space with accurate obstacle detection



CURRENT BOT


Kinect Camera
Jetson Nano
11.1 V Li-Po
L298N
IR Array
Chassis with 12V Motors

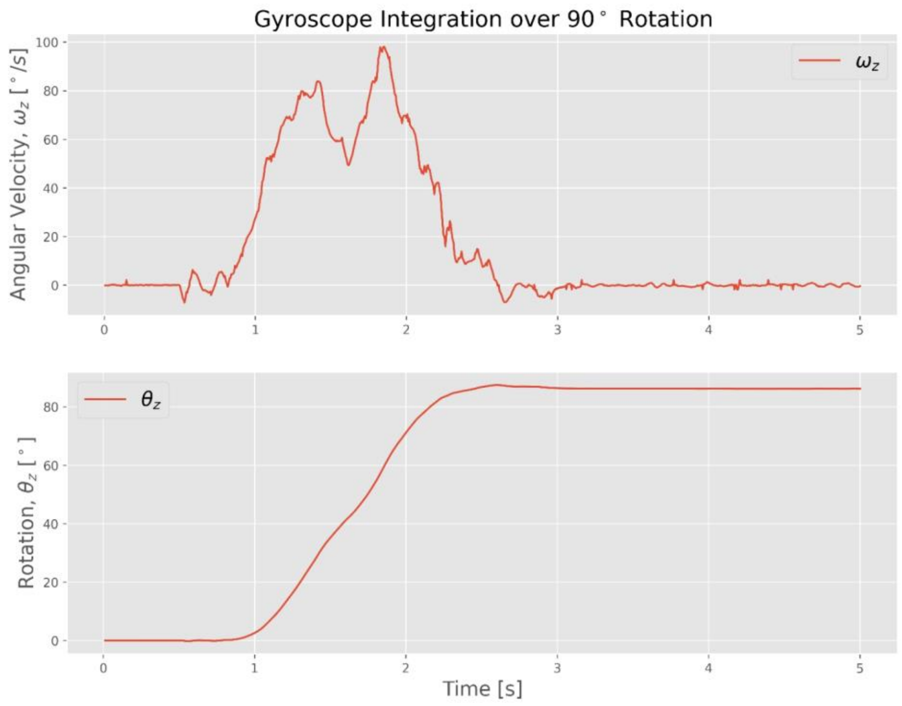

MPU6050
sensor

COMPONENTS USED

COMPONENTS	SPECIFICATION	NUMBER USED
Jetson Nano	Developer Kit - 4GB	1
Robot Chassis	Dimensions: 250 x 200 x 46 mm	1
DC motor	100 rpm 12V	2
Wheels	Diameter-130mm Width-60mm	4
Kinect v1 Camera	Stereo camera, IR camera	1
Motor Driver IC	L298N	1
MPU-6050	6 Axis motion tracking	1
Battery	11.1 V 4200mah Li-Po	1

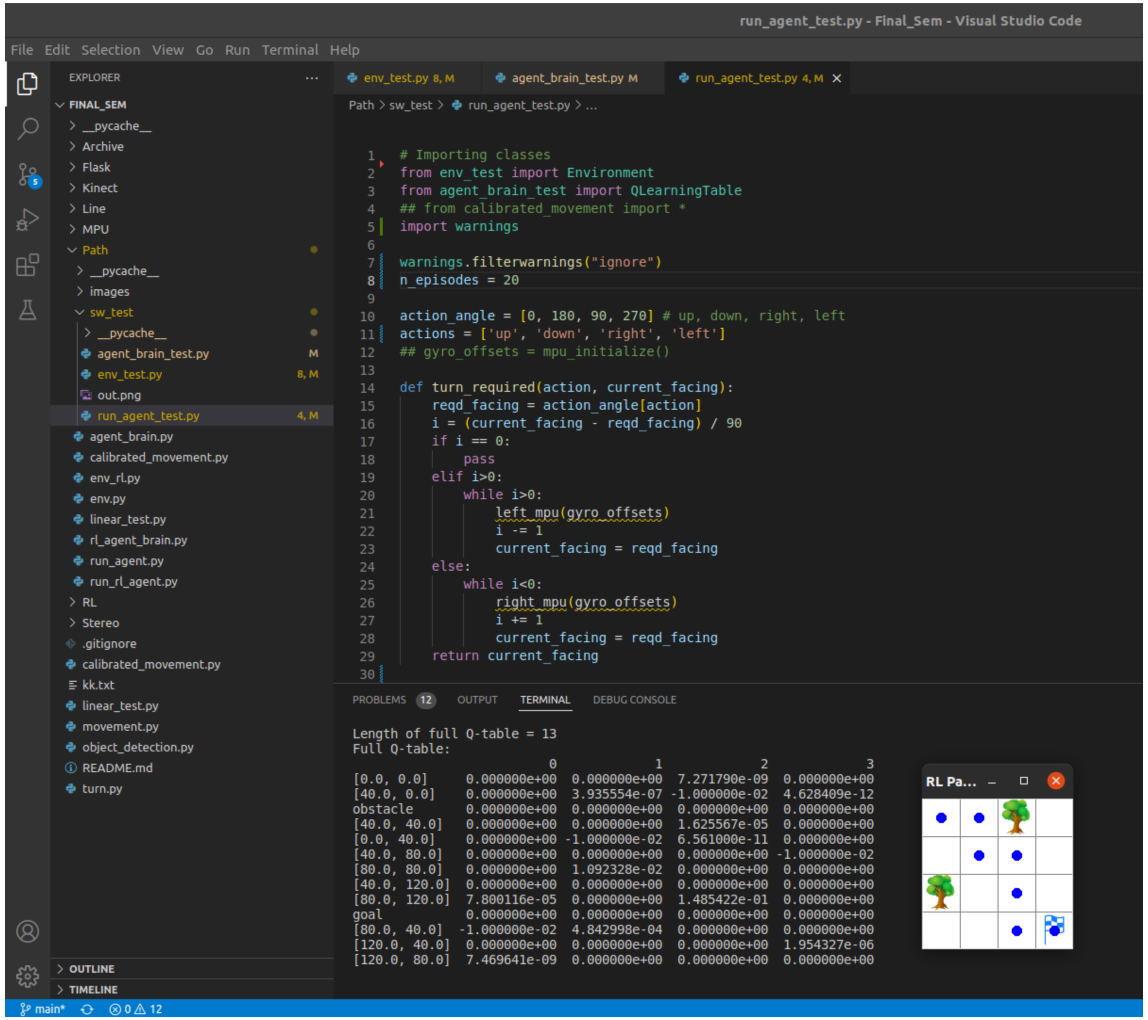
CURRENT PROGRESS

- Mounted stage for Kinect camera.
- Connected 100RPM 12V motors on all four wheels for better stability.
- Interfaced MPU-6050 with jetson nano.
- Integrated gyro output (Angular velocity) to get Angular displacement along z-axis which is suitable to identify bot turning.
- Used angular displacement destination to accurately turn left and right.

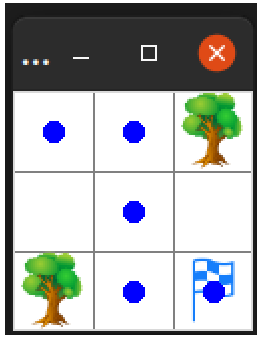


CURRENT PROGRESS

- Modified RL SARSA algorithm to make it suitable for hardware implementation by integrating the different modules of obstacle detection, calibrated movement and RL algorithm for path planning.
- Tested software simulation of the modified algorithm
- Tested the algorithm on a stage atop table with temporary obstacles.



CURRENT PROGRESS



Full Q-table:

	0	1	2	3
[0.0, 0.0]	0.00	0.000000	2.770883e-07	6.561000e-11
[40.0, 0.0]	0.00	0.000028	-1.000000e-02	3.254256e-10
[40.0, 40.0]	0.00	0.001828	0.000000e+00	0.000000e+00
[80.0, 40.0]	-0.01	0.000000	0.000000e+00	0.000000e+00
obstacle	0.00	0.000000	0.000000e+00	0.000000e+00
[0.0, 40.0]	0.00	-0.010000	0.000000e+00	0.000000e+00
[40.0, 80.0]	0.00	0.000000	6.793465e-02	0.000000e+00
goal	0.00	0.000000	0.000000e+00	0.000000e+00

Output of hardware implementation when tested atop table with temporary obstacles.

PROBLEMS FACED

- Current RL SARSA algorithm was not suitable for hardware implementation
- Compatibility issues when integrating the obstacle detection, calibrated movement and RL algorithm for path planning.
- Faced errors with processing MPU6050 raw data
- Issues with detecting and populating obstacles in map in hardware implementation

WORK PLAN

Third review:

- Test the validity of hardware implementation in proper testing space.
- Setup interface for mapping the environment remotely.
- Testing and optimizing.

QUESTIONS FROM PREVIOUS REVIEW:

1) Briefly explain your object detection methodology.

Ans: The bot detects objects using stereo vision and depth mapping. The bot detects objects at a distance of 45cm based on the threshold we have set. In order to check whether an object is an actual obstacle, we check if the pixels occupied by an object takes up more than 15% of the area of the pre-defined ROI.

2) Will the bot detect any object in its path?

Ans: Yes, the bot will detect any object in its path as an obstacle. Obstacle detection using depth mapping is very robust as it can detect objects of any shape and size that are at a particular distance from the camera.

3) Explain how you are powering the bot.

Ans: As of now, the four motors are powered by the Li-Poly battery. The Jetson NANO and the Kinect v1 are connected to an AC power source through adapters

QUESTIONS FROM PREVIOUS REVIEW:

4) What are the specifications of the motor you are using?
Ans: We are using four motors each rated for 12V and 100rpm.

5) Can the bot be connected to a wifi network?
Ans: Yes, the bot is capable of connecting to a network. We already pass commands to run the bot remotely using SSH.

CONFERENCE SUBMITTED

AIKP'22: Second International Conference On Artificial Intelligence and Knowledge Processing
Woxsen University
Hyderabad, India, July 22-23, 2022

REFERENCES

- Martin Gromniak and Jonas Stenzel ,"Deep Reinforcement Learning for Mobile Robot Navigation", Asia-Pacific Conference on Intelligent Robot Systems (ACIRS),July 2019
- Guillaume Sartoretti , Justin Kerr , Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset,“PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning”, Ieee robotics and automation letters, , Vol. 4, no. 3, July 2019.
- Hyansu Bae , Gidong Kim , Jonguk Kim , Dianwei Qian 4 and Sukgyu Lee , “Multi-Robot Path Planning Method Using Reinforcement Learning”, Multidisciplinary Digital Publishing Institute, Vol 3,No 4, May 2019.
- Jing Xin,Huan Zhao,Ding Liu,”Application of Deep Reinforcement Learning in Mobile Robot Path Planning”, Ieee robotics and automation letters, Vol 2, No 3 , January 2019.

END OF PRESENTATION

REINFORCEMENT LEARNING BASED AUTOMATED PATH PLANNING IN GARDEN ENVIRONMENT USING DEPTH - 'RAPiG-D'

THIRD REVIEW

- S SATHIYA MURTHI - 2018504604
- PRANAV BALAKRISHNAN - 2018504581
- C ROSHAN ABRAHAM - 2018504591

BE ELECTRONICS AND COMMUNICATION ENGINEERING

GUIDED BY:
DR. V. SATHIESH KUMAR,
ASSISTANT PROFESSOR,
DEPARTMENT OF ELECTRONICS ENGINEERING
MIT CAMPUS, ANNA UNIVERSITY

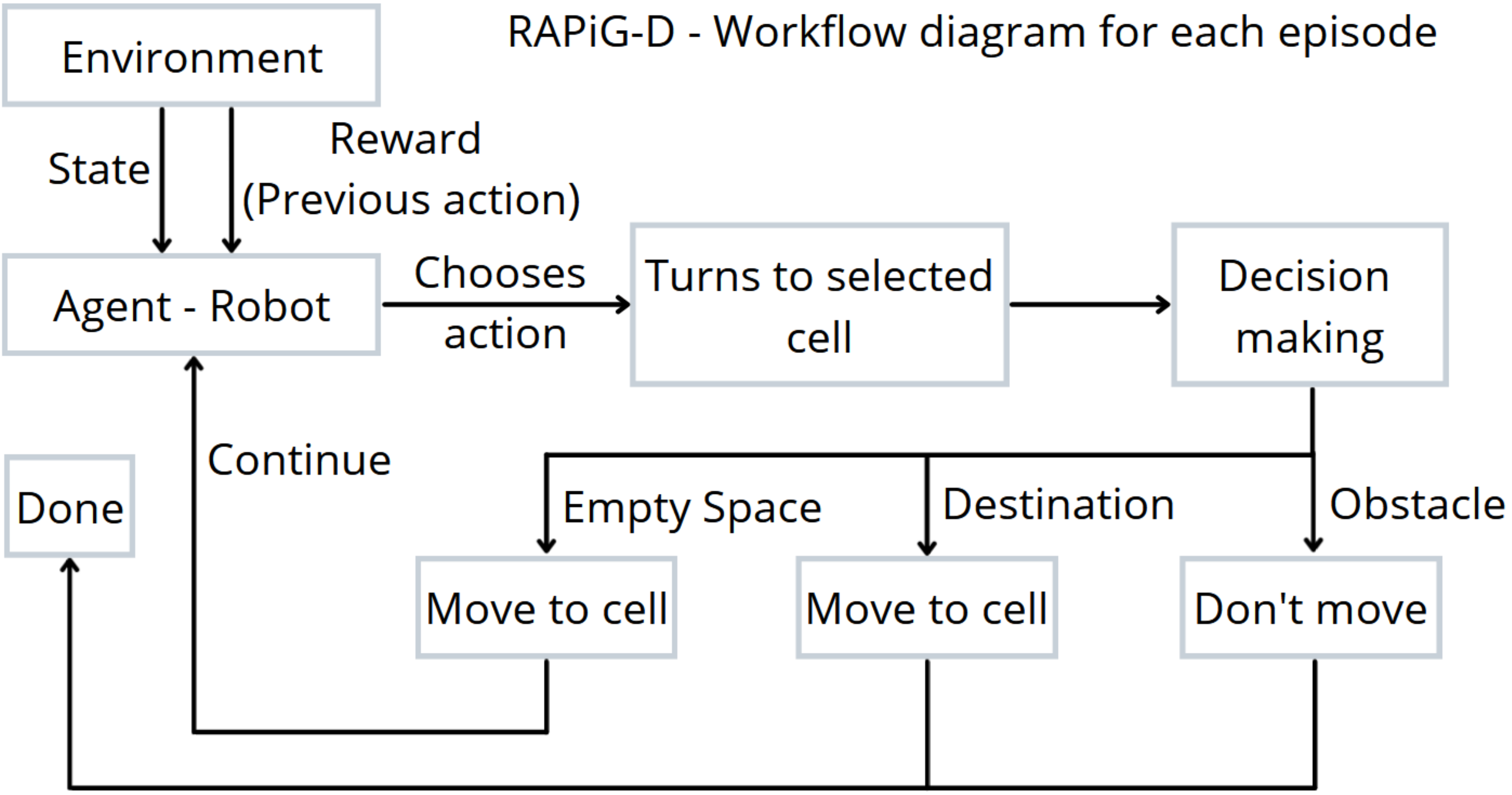
TABLE OF CONTENTS

OBJECTIVES

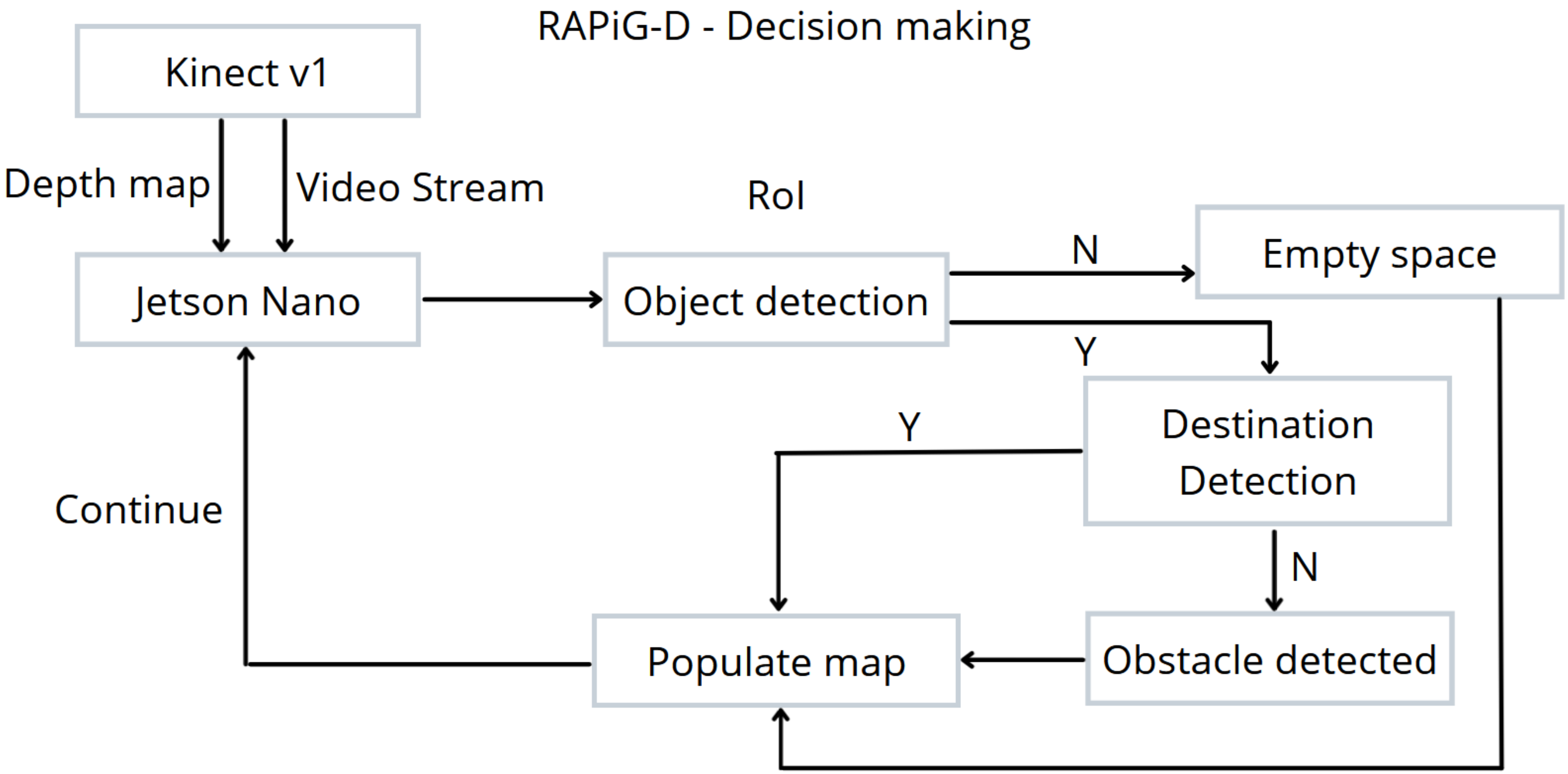
- To perform hardware implementation of reinforcement learning based path planning to generate a map of an unknown environment by employing depth estimation based obstacle detection and analyze its performance.

- To estimate the best possible path to reach a given destination.

METHODOLOGY



METHODOLOGY

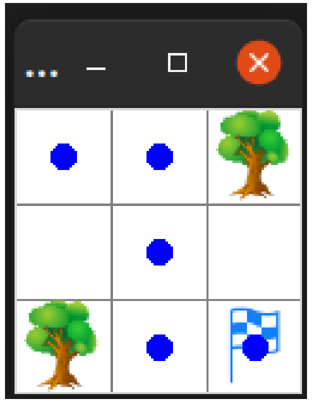


PREVIOUS PROGRESS

- Calibrated bot movement using MPU 6050.
- Modified RL SARSA algorithm to make it suitable for hardware implementation.
- Tested software simulation of the modified algorithm
- Tested the algorithm on a stage atop table with temporary obstacles.

Full Q-table:

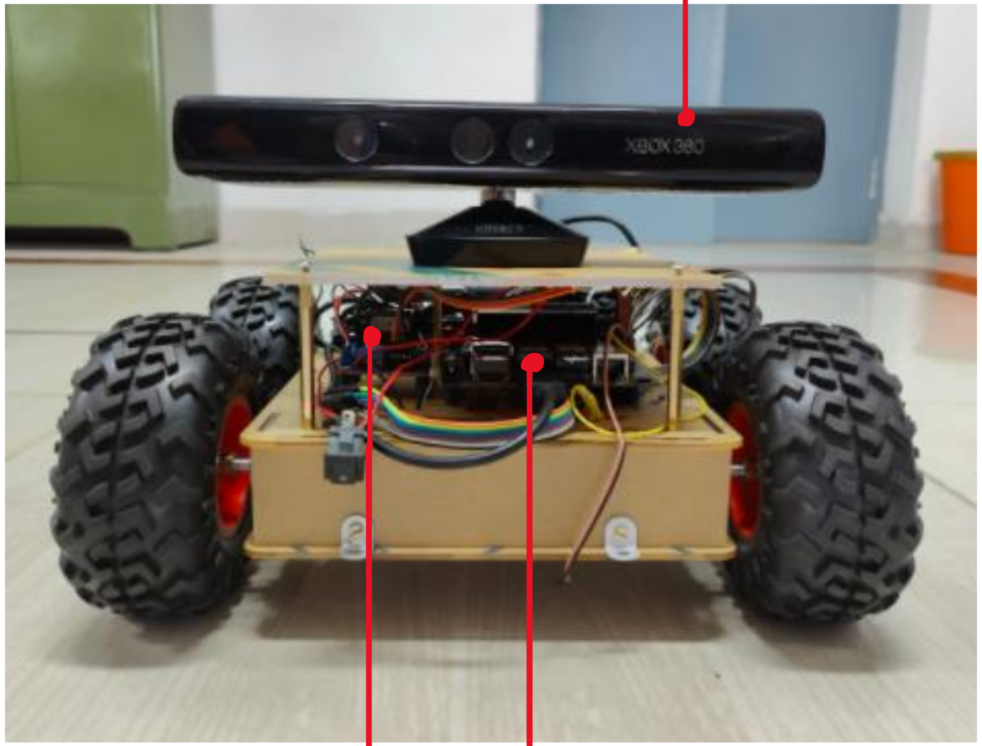
	0	1	2	3
[0.0, 0.0]	0.00	0.000000	2.770883e-07	6.561000e-11
[40.0, 0.0]	0.00	0.000028	-1.000000e-02	3.254256e-10
[40.0, 40.0]	0.00	0.001828	0.000000e+00	0.000000e+00
[80.0, 40.0]	-0.01	0.000000	0.000000e+00	0.000000e+00
obstacle	0.00	0.000000	0.000000e+00	0.000000e+00
[0.0, 40.0]	0.00	-0.010000	0.000000e+00	0.000000e+00
[40.0, 80.0]	0.00	0.000000	6.793465e-02	0.000000e+00
goal	0.00	0.000000	0.000000e+00	0.000000e+00

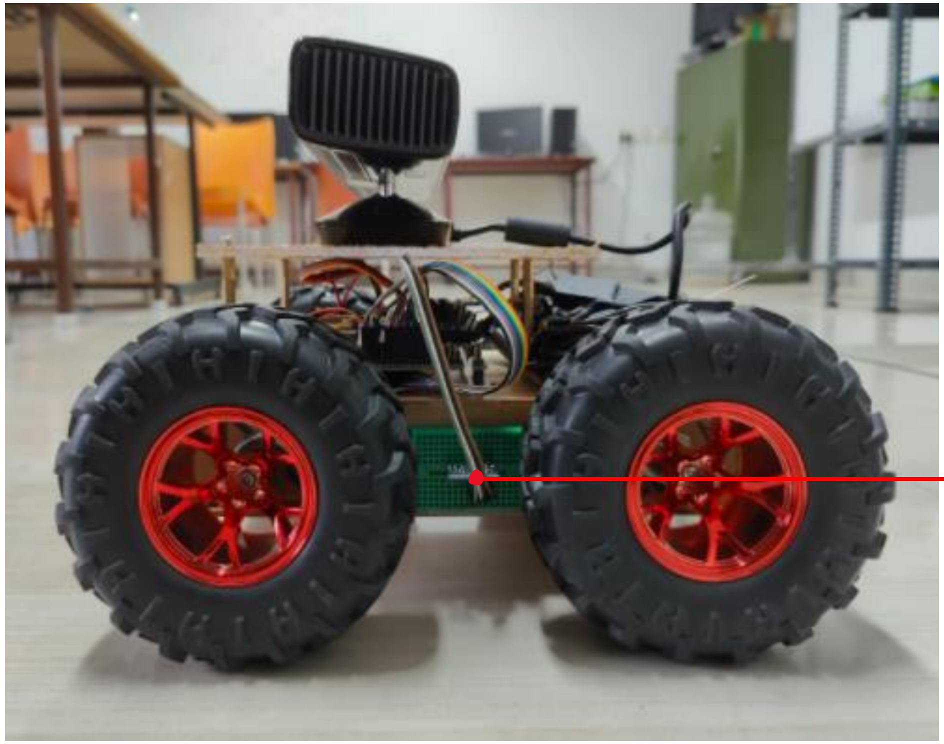


Output of hardware implementation when tested atop table with temporary obstacles.

CURRENT BOT

Kinect v1 Camera





MPU6050 sensor

NVIDIA Jetson Nano

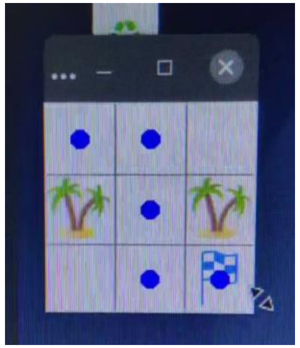
L298N Driver

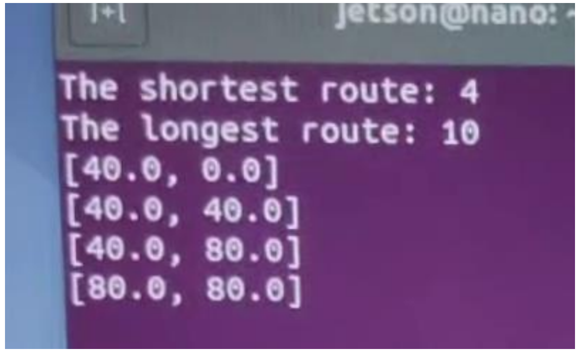
COMPONENTS USED

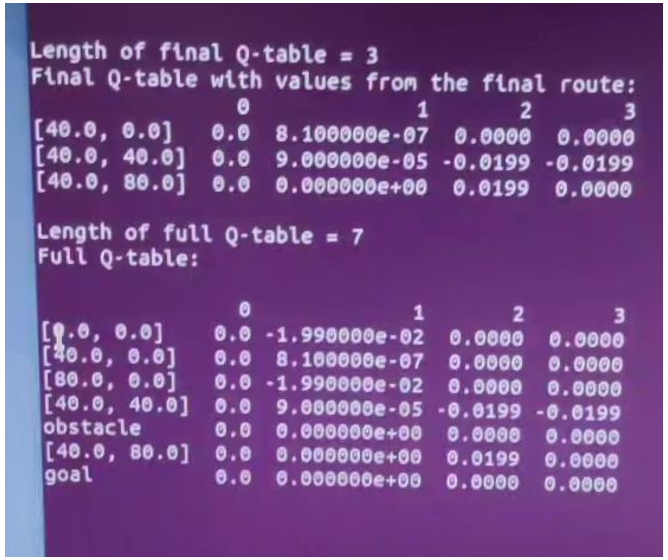
COMPONENTS	SPECIFICATION	NUMBER USED
Jetson Nano	Developer Kit - 4GB	1
Robot Chassis	Dimensions: 250 x 200 x 46 mm	1
DC motor	100 rpm 12V	2
Wheels	Diameter-130mm Width-60mm	4
Kinect v1 Camera	Stereo camera, IR camera	1
Motor Driver IC	L298N	1
MPU-6050	6 Axis motion tracking	1
Battery	11.1 V 4200mah Li-Po	1

CURRENT PROGRESS

- Optimised bot movement.
- Performed hardware test in testing space
- Set up web app for passing information regarding action chosen in each step.
- Performed 2D space mapping in testing space successfully.
- Ran 10 episodes on 3x3 space and determined the optimal path.







CODE / DEMO – OBSTACLE DETECTION

```python
def show_depth(): ## 640x480
    global threshold
    global current_depth

    depth, timestamp = freenect.sync_get_depth()
    depth = 255 * np.logical_and(depth >= 0,
                                 depth <= current_depth + threshold)
    depth = depth.astype(np.uint8)
    depth = cv2.cvtColor(depth, cv2.COLOR_GRAY2RGB)

    is_obj = check_if_object(depth)
    if is_obj:
        color = (0, 0, 255)
        print('Object')
        try:
            send_to_flask(True)
        except Exception as e:
            pass
        time.sleep(1)
    else:
        color = (0, 255, 0)
        try:
            send_to_flask(False)
        except Exception as e:
            pass

    frame = cv2.rectangle(depth, ROI[0], ROI[1], color, 1)


    cv2.imshow('Depth', frame)
```

```python
def check_if_object(frame):
    area = int((1 - 2*roi_val) * height * (1 - 2*roi_val) * width)
    pixels = int(np.sum(frame[ROI[0][1]:ROI[1][1], ROI[0][0]:ROI[1][0], 0]) / 255)
    return pixels > (area * alert_thresh)
```

Libraries used:

- Freenect
- OpenCV

- freenect.sync_get_depth() - Function used to generate depth map

CODE / DEMO – CALIBRATED MOVEMENT

```python
def left_mpu(gyro_offsets):
    stop()
    angle = 0
    rot_axis = 2 # axis being rotated (2 = z-axis)
    data,t_vec = [],[]
    t0 = time.time()
    while angle < angle_thresh:
        left()
        data.append(get_gyro())
        t_vec.append(time.time()-t0)
        data_offseted = np.array(data)[:,rot_axis]-gyro_offsets[rot_axis]
        integ1_array = cumtrapz(data_offseted,x=t_vec)
        try:
            angle = integ1_array[-1]
        except:
            pass
    stop()
```

- Turn calibrated using MPU6050 gyro sensor data integration
- Linear movement controlled by timing
- Motor control using GPIO pins

```python
def move_one_f():
    fwd()
    time.sleep(1.32)
    stop()
    time.sleep(1)


def move_one_b():
    back()
    time.sleep(1.32)
    stop()
    time.sleep(1)
```

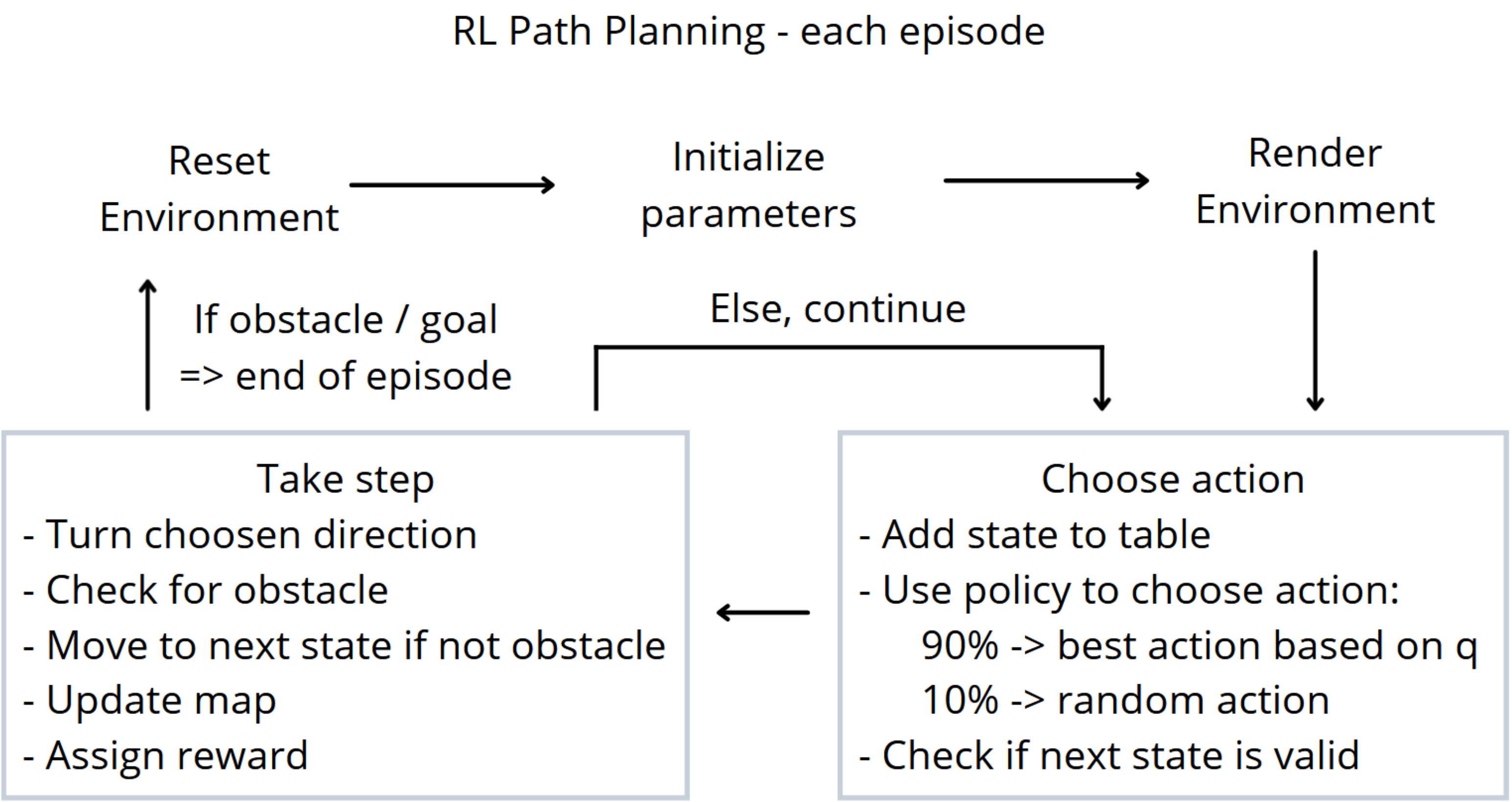
```python
def fwd():
    stop()
    GPIO.output(motor_pin_d, GPIO.HIGH)
    GPIO.output(motor_pin_c, GPIO.HIGH)
    GPIO.output(motor_pin_a, GPIO.LOW)
    GPIO.output(motor_pin_b, GPIO.LOW)


def back():
    stop()
    GPIO.output(motor_pin_a, GPIO.HIGH)
    GPIO.output(motor_pin_b, GPIO.HIGH)
    GPIO.output(motor_pin_c, GPIO.LOW)
    GPIO.output(motor_pin_d, GPIO.LOW)
```

```python
def left():
    stop()
    GPIO.output(motor_pin_c, GPIO.HIGH)
    GPIO.output(motor_pin_b, GPIO.HIGH)
    GPIO.output(motor_pin_a, GPIO.LOW)
    GPIO.output(motor_pin_d, GPIO.LOW)

def right():
    stop()
    GPIO.output(motor_pin_d,GPIO.HIGH)
    GPIO.output(motor_pin_a,GPIO.HIGH)
    GPIO.output(motor_pin_b,GPIO.LOW)
    GPIO.output(motor_pin_c,GPIO.LOW)
```

CODE / DEMO – RL PATH PLANNING



CONFERENCE ACCEPTANCE

AIKP’22
Second International Conference On
Artificial Intelligence and Knowledge Processing
Woxsen University
Hyderabad, India, July 22-23, 2022

AKIP’22 notification for paper 8079 Inbox ×

AKIP’22 <akip22@easychair.org>
to me

Kindly submit the revised paper with the registration fee on or before 20th June 2022 in easy-chair itself.

SUBMISSION: 8079
TITLE: Reinforcement Learning based Automated Path Planning in Garden environment using Depth - RAPiG-D

----------------------- REVIEW 1 ---------------------
SUBMISSION: 8079
TITLE: Reinforcement Learning based Automated Path Planning in Garden environment using Depth - RAPiG-D
AUTHORS: Sathiya Murthi S, Pranav Balakrishnan, Roshan Abraham C and Sathiesh Kumar V

----------- Overall evaluation -----------
SCORE: 2 (accept)
----- TEXT:
Dear Authors,

Congratulations!

Your paper has been accepted to the AKIP'22 Conference.

REFERENCES

- Martin Gromniak and Jonas Stenzel ,"Deep Reinforcement Learning for Mobile Robot Navigation", Asia-Pacific Conference on Intelligent Robot Systems (ACIRS),July 2019

- Guillaume Sartoretti , Justin Kerr , Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset,“PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning”, Ieee robotics and automation letters, , Vol. 4, no. 3, July 2019.

- Hyansu Bae , Gidong Kim , Jonguk Kim , Dianwei Qian 4 and Sukgyu Lee , “Multi-Robot Path Planning Method Using Reinforcement Learning”, Multidisciplinary Digital Publishing Institute, Vol 3,No 4, May 2019.

- Jing Xin,Huan Zhao,Ding Liu,”Application of Deep Reinforcement Learning in Mobile Robot Path Planning”, Ieee robotics and automation letters, Vol 2, No 3 , January 2019.

END OF PRESENTATION

REINFORCEMENT LEARNING BASED AUTOMATED PATH PLANNING IN GARDEN ENVIRONMENT USING DEPTH - 'RAPiG-D'

- S SATHIYA MURTHI - 2018504604
- PRANAV BALAKRISHNAN - 2018504581
- C ROSHAN ABRAHAM - 2018504591

BE ELECTRONICS AND COMMUNICATION ENGINEERING

GUIDED BY:
DR. V. SATHIESH KUMAR,
ASSISTANT PROFESSOR,
DEPARTMENT OF ELECTRONICS ENGINEERING
MIT CAMPUS, ANNA UNIVERSITY

TABLE OF CONTENTS

INTRODUCTION

- Over the years, robots have been implemented in a wide range of applications and environments.
- In turn, this has led to dealing with a host of environments that are increasingly dynamic and unknown.
- Path planning with reinforcement learning would aid the automation of multiple tasks with minimal human intervention.

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.
- Current methods are limited by their inability to adapt to new environments.

OBJECTIVES

- To perform hardware implementation of reinforcement learning based path planning to generate a map of an unknown environment by performing depth estimation based obstacle detection and analyze its performance.

- To estimate the best possible path to reach a given destination.

LITERATURE SURVEY

AUTHOR NAME	TITLE	YEAR OF PUBLICATION	JOURNAL	DESCRIPTION
Valentyn N. Sichkar	Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot	2019	International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)	• This paper implemented two Reinforcement learning algorithms – Q-learning and SARSA in a software simulated virtual environment. • It analyzed their performance. • The algorithm learnt the optimal path to get maximum payoff and reached goal while avoiding obstacles.

LITERATURE SURVEY

AUTHOR NAME	TITLE	YEAR OF PUBLICATION	JOURNAL	DESCRIPTION
Martin Gromniak, Jonas Stenzel	Deep Reinforcement Learning for Mobile Robot Navigation	2019	Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)	• Developed training procedure, set of actions available, suitable state representation, and a reward function. • Evaluated using a simulated real-time environment • The experimental evaluation showed that DRL can be applied successfully to robot navigation.

LITERATURE SURVEY

AUTHOR NAME	TITLE	YEAR OF PUBLICATION	JOURNAL	DESCRIPTION
Jing Xin, Huan Zhao, Ding Liu	Application of Deep Reinforcement Learning in Mobile Robot Path Planning	2019	IEEE robotics and automation letters	• A Deep Q-network (DQN) is designed and trained to approximate the mobile robot state-action value function and Q value corresponding to each possible mobile robot action is determined by the well trained DQN. • The current optimal mobile robot action is selected by the action selection strategy to the goal point while avoiding obstacles ultimately

METHODOLOGY - SARSA

- **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.

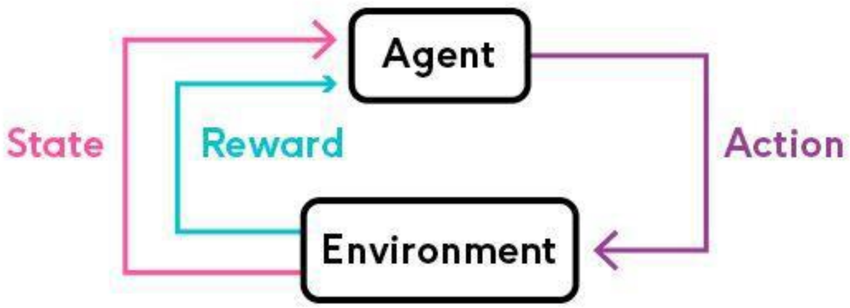


Figure 1: SARSA Illustration
Reference: https://medium.com/@vishnuvijayanpv/what-is-reinforcement-learning-e5dc827c8564

- The SARSA algorithm involves learning the environment by choosing actions at each state using a policy function.
- At each state, an action is chosen and then the environment provides the reward for that action along with the next state. Based on the reward, the Q-values are updated in the Q-table.

METHODOLOGY - OBSTACLE DETECTION

- **Stereo vision** is used for the computation of depth based on the binocular disparity between the images of an object in left and right eyes.
- Stereo Camera based depth estimation is used for detecting obstacles in the testing region
- Generated depth map is processed by thresholding to identify objects within a certain distance.
- This is then used to detect the presence of obstacles that are large enough in the immediate next cell of the testing space.

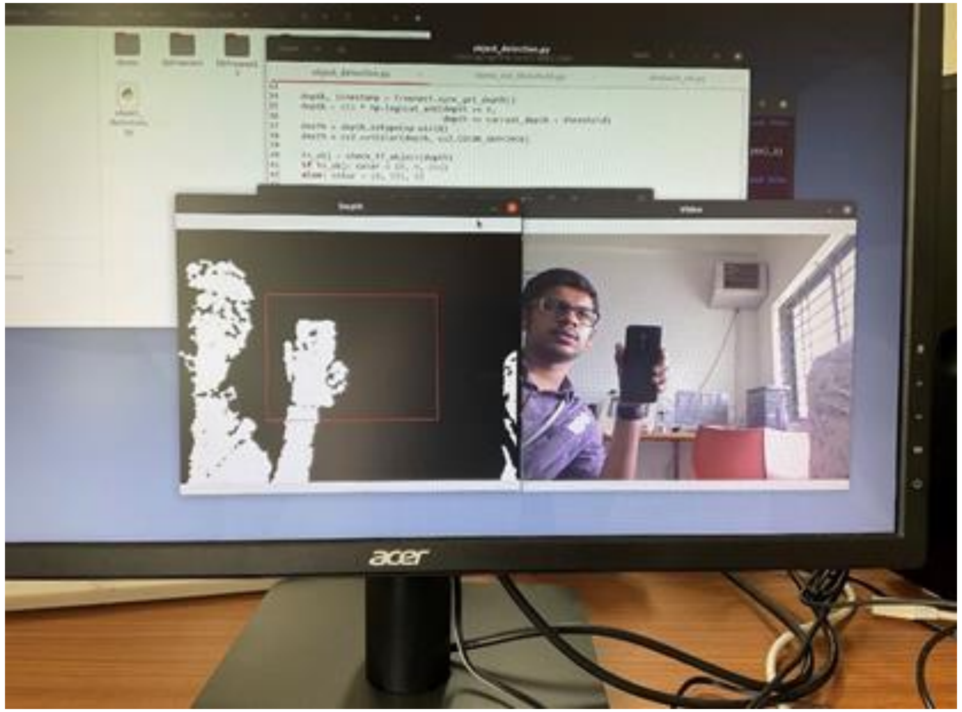


Figure 2: Obstacle detection output using stereo camera depth estimation

METHODOLOGY – HARDWARE IMPLEMENTATION

- To perform hardware implementation, a robot is designed using NVIDIA Jetson Nano Developer Kit as the Micro-controller and XBOX 360 Kinect camera.
- The Kinect camera provides the required depth map by utilizing a stereo camera setup along with an IR camera.
- An MPU6050 sensor 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 which are controlled by the L298N motor driver.
- The Jetson Nano and the Kinect camera are currently externally powered using a 5V4A AC.

Figure 3: Testing space with robot and obstacles

ROBOT DESIGNED

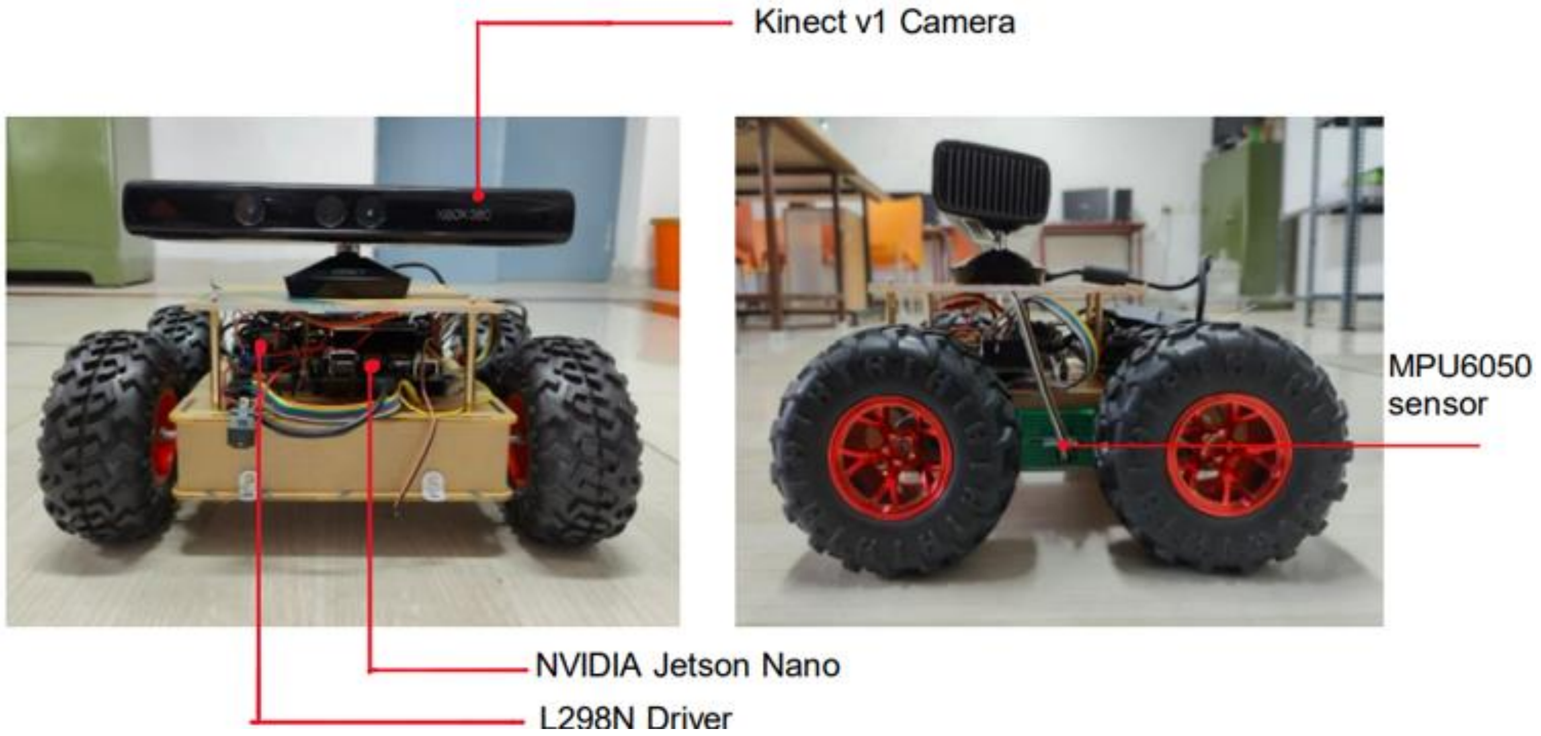
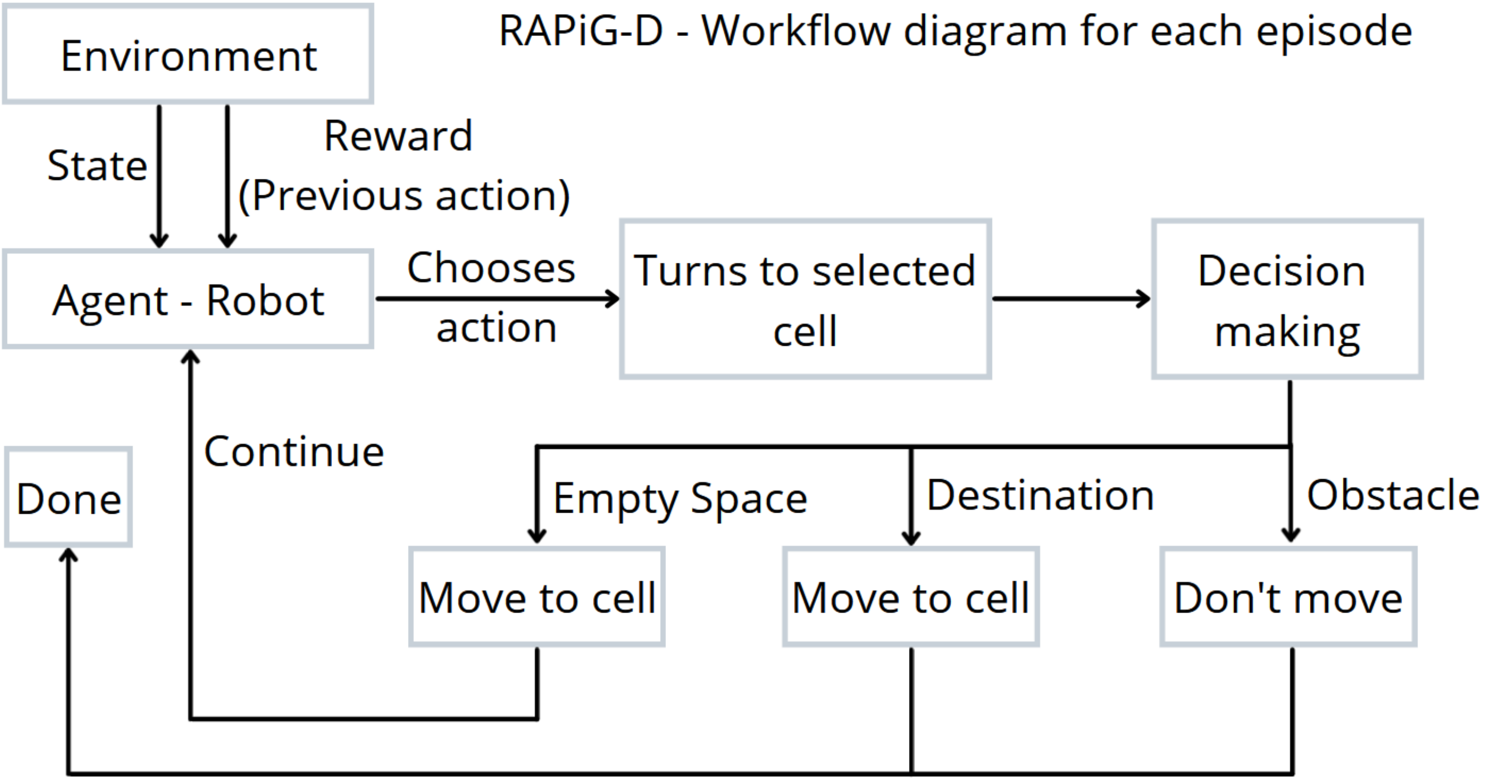


Figure 4: Robot designed for hardware implementation

COMPONENTS USED

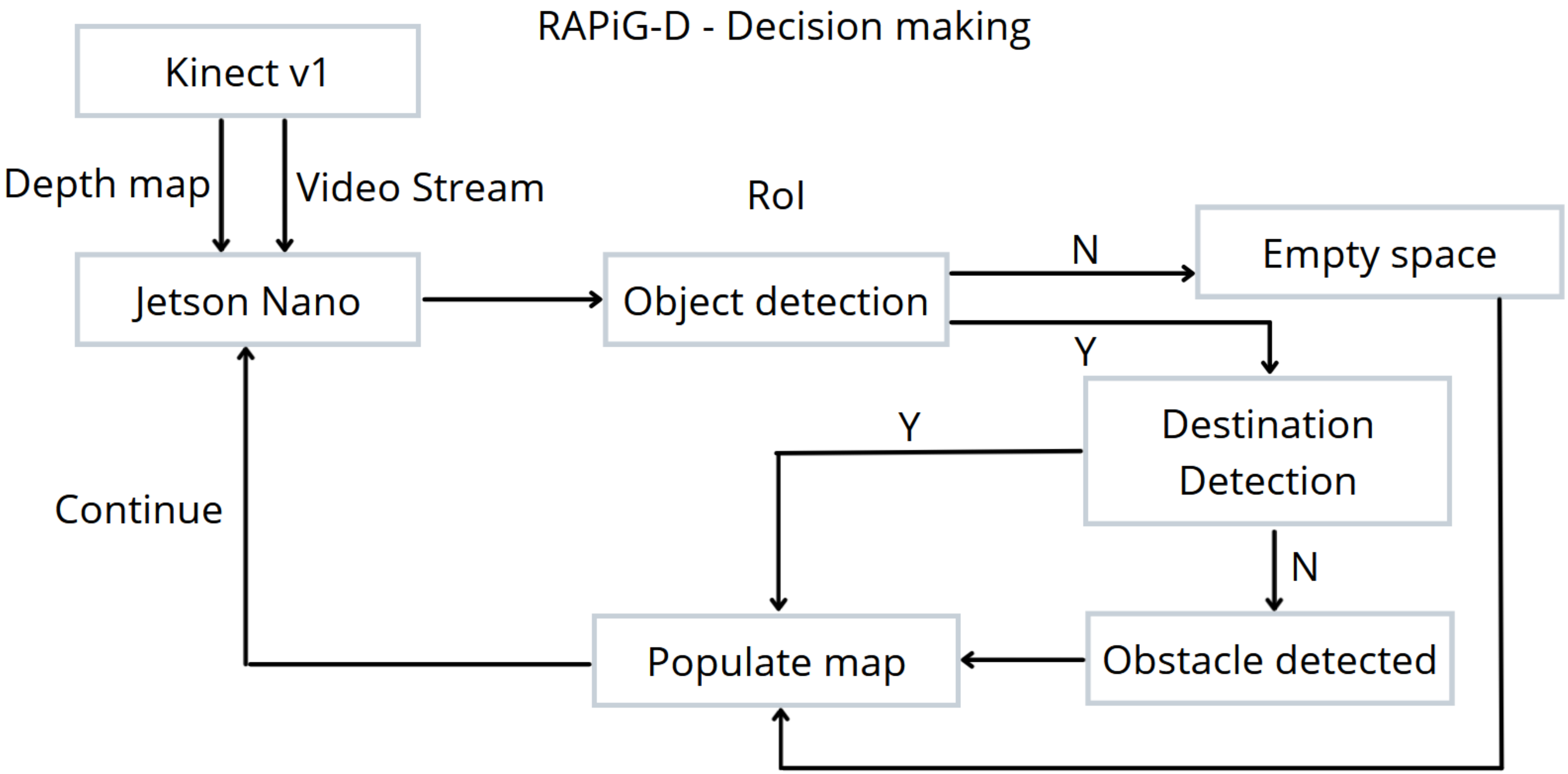
COMPONENTS	SPECIFICATION	NUMBER USED
Jetson Nano	Developer Kit - 4GB	1
Robot Chassis	Dimensions: 250 x 200 x 46 mm	1
DC motor	100 rpm 12V	2
Wheels	Diameter-130mm Width-60mm	4
Kinect v1 Camera	Stereo camera, IR camera	1
Motor Driver IC	L298N	1
MPU-6050	6 Axis motion tracking	1
Battery	11.1 V 4200mah Li-Po	1

FLOWCHART



RAPiG-D - Workflow diagram for each episode
Environment
State
Reward
(Previous action)
Agent - Robot
Chooses
action
Turns to selected
cell
Decision
making
Continue
Done
Empty Space
Destination
Obstacle
Move to cell
Move to cell
Don't move

FLOWCHART


RAPiG-D - Decision making
Kinect v1
Depth map
Video Stream
RoI
Jetson Nano
Object detection
N
Empty space
Y
Destination Detection
Y
Continue
N
Populate map
Obstacle detected

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
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

RESULTS AND DISCUSSIONS

- To check the validity of the path planning algorithm, software simulation was performed on a larger 10 x 10 environment with 10 obstacles and the agent ran 1000 episodes.
- Software implementation takes less time per episode. The start point is (0,0) and the destination is (9,9).
- The obstacle coordinates are manually set and fed to the algorithm.
- Based on the policy, the RL agent goes around the environment, checking each coordinate to see if it is an obstacle, an empty space, or the destination.
- 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.

RESULTS AND DISCUSSIONS

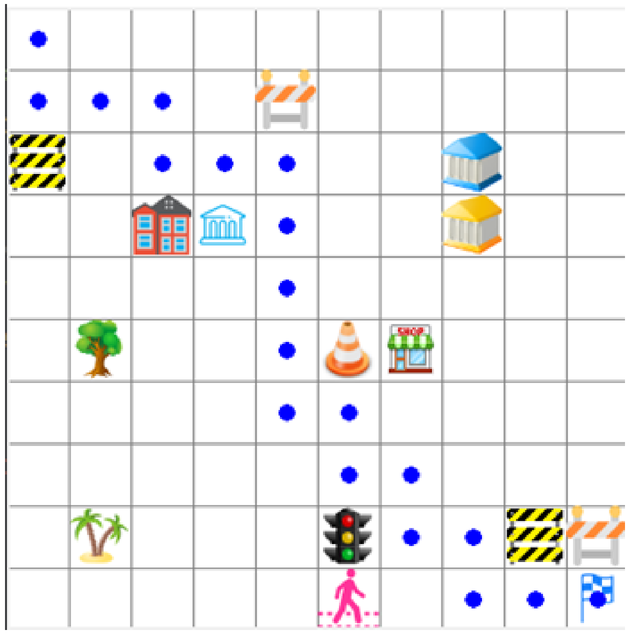


Figure 5: Map generated in software implementation

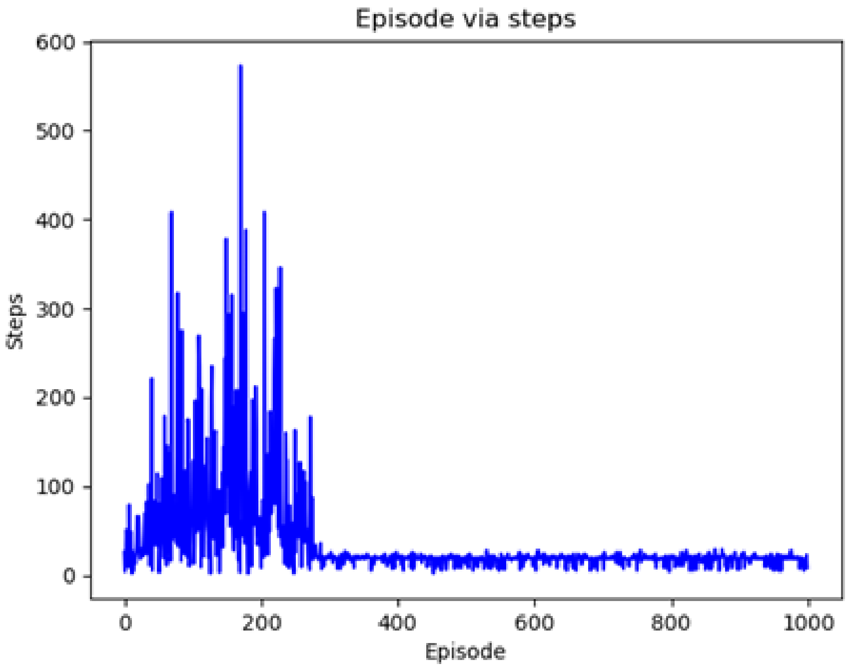


Figure 6: Plot of number of steps in each episode – Software implementation

RESULTS AND DISCUSSIONS

- The robot agent was allowed to explore a 3 x 3 testing space 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 is also accurate, controlled by the MPU6050 gyro sensor.
- After 10 episodes, a map is generated along with the optimal path (Figure 7)

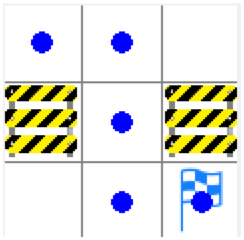


Figure 7: Map generated in hardware implementation

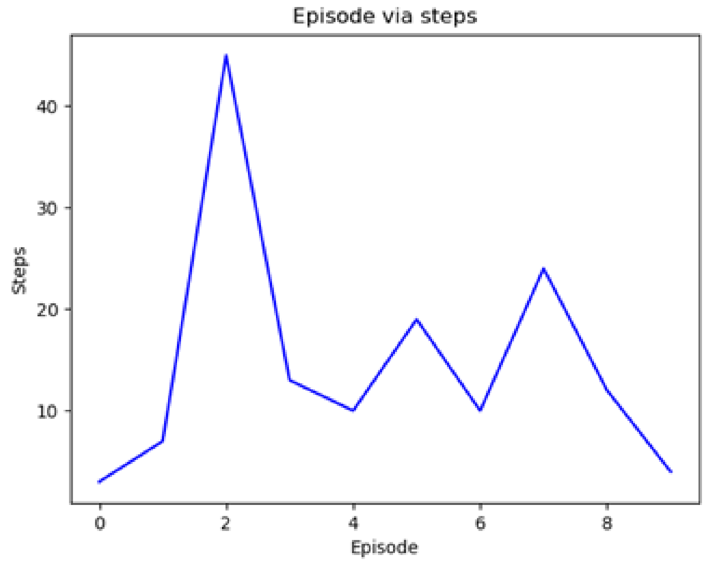
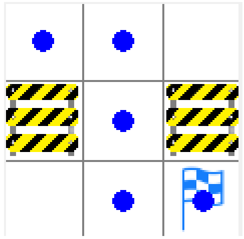


Figure 8: Plot of number of steps in each episode – Hardware implementation

RESULTS AND DISCUSSIONS

	Up	Down	Right	Left
[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



Final Q-Table generated by the RL SARSA algorithm

RESULTS AND DISCUSSIONS

- Figure 9 shows the shortest path determined by the algorithm.
- 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.
- The agent then chooses which action to take at each state and is able to identify the optimal path to reach the destination while avoiding obstacles.

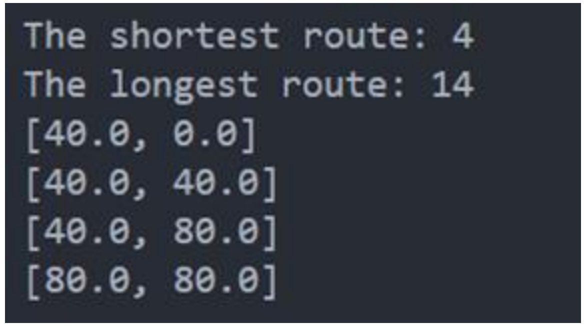


Figure 9: Shorted route and longest route

CONCLUSION

- The robot as an RL agent, successfully performs RL based path planning to accurately map the environment and its obstacles and determine the shortest path to reach a given destination.
- In the 3x3 environment, the robot agent performs accurate obstacle detection implementing depth estimation and by using the SARSA algorithm, identifies the optimal path to the destination within 10 episodes.
- As the number of obstacles and the size of the environment increases the time taken to identify the shortest path would also increase.
- Thus, a simple RL algorithm such as SARSA is used to identify an optimal path and is able to allow the robot agent to map an unknown environment.

FUTURE WORKS

- Can be further extended to larger environments and even dynamic environments.
- Future implementations may involve increasing the degrees of freedom of the agent and the environment states and even going beyond cell-based environments.
- It can also be implemented in diversified applications ranging from garden or warehouse management, Air crash investigations, Search and Rescue operations, serving food at restaurants, and even Space exploration.

REFERENCES

- Valentyn N. Sichkar, “Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot,” In: International Conference on Industrial Engineering, Applications and Manufacturing, 2019

- Martin Gromniak and Jonas Stenzel ,"Deep Reinforcement Learning for Mobile Robot Navigation", Asia-Pacific Conference on Intelligent Robot Systems (ACIRS),July 2019

- Guillaume Sartoretti , Justin Kerr , Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset,“PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning”, Ieee robotics and automation letters, , Vol. 4, no. 3, July 2019.

- Hyansu Bae , Gidong Kim , Jonguk Kim , Dianwei Qian 4 and Sukgyu Lee , “Multi-Robot Path Planning Method Using Reinforcement Learning”, Multidisciplinary Digital Publishing Institute, Vol 3,No 4, May 2019.

- Jing Xin,Huan Zhao,Ding Liu,”Application of Deep Reinforcement Learning in Mobile Robot Path Planning”, Ieee robotics and automation letters, Vol 2, No 3 , January 2019.

CONFERENCE ACCEPTANCE

AIKP'22 - ID: 8079
Second International Conference On
Artificial Intelligence and Knowledge Processing
Woxsen University
Hyderabad, India, July 22-23, 2022

Paper will get published in CRC Press Taylor & Francis Book

END OF PRESENTATION