PROJECT REPORT

Title

“DEVELOPMENT OF JOYSTICK CONTROL INTERFACE FOR AUTONOMOUS ROBOTIC NAVIGATION”

By

NARAYANAN PP

(Research Intern)

JNResearch Labs LLP

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that **Mr. NARAYANAN P P**, 2nd year student of **BTech, Mechanical Engineering at Maulana Azad National Institute of Technology, Bhopal** has successfully completed the summer Internship training at our Company **JNResearch Labs LLP, Bangalore** from 10th May 2022 to 15th July 2022.

Project Title: Development of Joystick Control Interface for Autonomous Robotic Navigation

He successfully completed the project, meeting all the objectives with the implementation on a real world prototype robotic system.

His conduct during the internship training is **GOOD**

Anantharaman P N

Chief Executive Officer

Place: Bangalore

Date: 18th July 2022

Registered Office:

JNResearch Labs LLP
435/A, 'Ananth',9th Main
2A Cross, Ideal Homes Township
RR Nagar, Bengaluru 560098
Phone: +91-80-9632177446

Reg No: AAS-5183
PAN: AAQFJ2021A
GSTIN: 29AAQFJ2021A1ZQ

ACKNOWLEDGEMENT

The internship opportunity I had with JN Research Labs LLP was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to **Mr. Anantharaman PN (CEO, JNRL)** who in spite of being extraordinarily busy, took time out to hear, guide and keep me on the correct path and allow me to carry out my project at their esteemed organization and extend it during the training.

I express my deepest thanks to **Mr. Parameswaran PN (CFO, JNRL)** for taking part in useful decisions & giving necessary advice and guidance. I would like to extend my gratitude to my internal guides at NIT Bhopal**, Professor R.K. Dwivedi** and **Professor Ravi Kumar Mandava** for their mentorship and support.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future,

Sincerely,

Narayanan PP

2nd Year,

B.Tech - Mechanical Engineering

MANIT, Bhopal

CONTENTS

ABSTRACT

An Autonomous robot is one that can perceive its environment, make decisions based on what it perceives and has been programmed to recognize, and then actuate a movement or manipulation within that environment. Several vital functions of robots such as Autonomous Navigation, Object Recognition, Speech Recognition & Synthesis, etc. are increasingly AI driven. This technical report prepared as an outcome of the internship, describes the methodologies to create a general purpose user-interface that helps to acquire labeled datasets for performing several of these AI features by navigating a real service robot in an indoor environment.

BACKGROUND

An Autonomous Robot is a machine that has the capability to sense, make decisions and perform actions in the real world environment in which it operates. Robotics applications vary greatly and a robot can be designed to carry out various tasks desired for its application. Broadly robots can be grouped into 6 categories:

1. Autonomous Mobile Robots (AMRs)

An Autonomous mobile robot can function completely independently and can understand and make decisions in real time as it moves through its environment. It is equipped with sensors and cameras which provides the robot with data needed for decision making and navigation. AI makes an important impact in decision making. Advanced AI algorithms enable the robot to perceive and navigate through its environment. AMR robots are capable of constructing a floor plan or a map for navigation and localisation purposes with respect to their environment. They can operate in any environment and are not restricted to pre-defined tracks or paths. A self-driving car is an example of AMR. It can autonomously navigate through its environment and is capable of detecting and avoiding obstacles on its way. A delivery robot which is used for delivering food or other items is also an example of AMR.

Fig 1: Robot in a factory automation

2. **Automated Guided Vehicles(AGVs)**

AGVs operate under a controlled environment specified by tracks or pre-defined paths and hence differ from AMRs. They can obey only simple programming instructions and are guided by magnetic strips or sensors for their navigation. Therefore, they require extensive facilities for their operation and are more expensive as compared to AMRs. They are usually used for restricted applications like in a warehouse or factory floor for moving items and delivering materials. They can sense obstacles but are unable to move past them and hence stop when they encounter obstacles.

Fig 2: Robot in a factory automation – Compact

3. Pre-Programmed Robots

As the name suggests, a pre-programmed robot is programmed to do a specific task and it simply executes that task. It cannot do other tasks and does not require human intervention. Such robots find extensive usage in automobile industry and medicine. Pre-programmed robotic arms are used in industries today to handle entire automobiles and perform certain operations on them such as spraying paint or making small welds, ext. They have very high accuracy and precision and reduce human effort greatly. In medicine, presently these robots are used for performing high accuracy and precise operations such as delivering radiations in order to kill tumors in cancer patients.

Fig 3: Robot with arm movement

4. Humanoid Robots

A humanoid robot is a robot which resembles a human body in shape or even a part of human body and can mimic human behavior and actions. Since the goal of a humanoid robot is to closely resemble a human, it needs to take decisions and perform certain actions or tasks like a human. Therefore, humanoid robots may also fall under the category of AMRs. A humanoid robot can perform human-like actions like walking, lifting loads and speaking as well. They can be used for interacting with humans such as guiding or helping old people, for entertainment purposes and also for teaching and medical use cases like performing a surgery. Typically a humanoid robot has a head, torso, two arms and two legs. A popular example of a humanoid robot is Hanson Robotics’ Sophia.

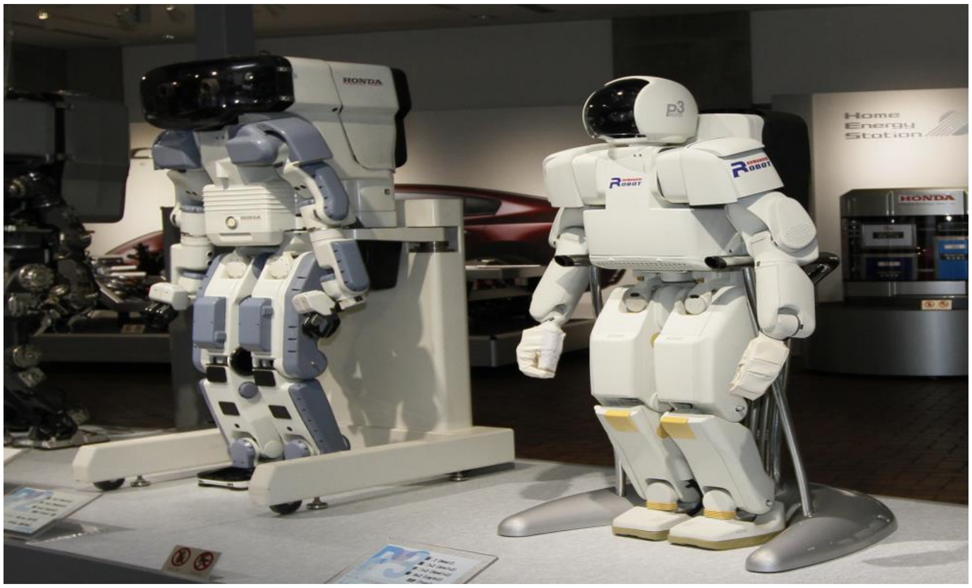

Fig 4: Humanoids

5. Augmenting Robots

Augmenting robots, also known as cobots, are designed to work alongside humans. These robots are not completely autonomous or independent and are remotely controlled by humans.They also learn from human behavior and can be used to perform manual or strenuous day to day tasks. Presently, cobots are for lifting heavy weights, diffusing bombs, performing surgery, stopping gas leaks, ext.

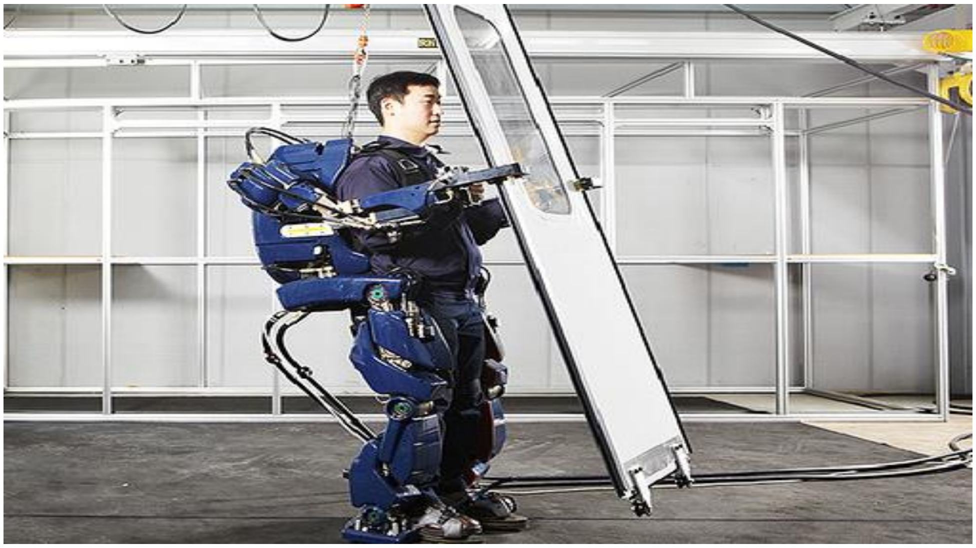

Fig 5: Cobots

6. Hybrid Robots

The various types of robots which are already discussed, can be combined together to design a hybrid robot. For example, pre-programmed robots or AMRs can be combined with humanoid robots to create a hybrid robot. Since the functions of various robotic types are combined into a single unit, hybrid robots can perform many functions which single robotic types are unable to. General AI can be realized through hybrid robots and therefore hybrid robots can be seen as future prospects.

Fig 6: Custom Robot Prototype

INTRODUCTION

Autonomous vehicles (AVs) are starting to become a real possibility in some parts of industry. Agriculture, transportation and the military are some of the examples. Many of the operations that vehicles have to perform are based on sensor information and artificial intelligence (AI) algorithms. Vehicles need to collect data, plan their trajectory and execute the trajectory. These tasks, especially the last two require non-traditional programming approaches and rely on machine learning techniques, which are part of AI.

AI algorithms used in autonomous vehicles

A. Route Planning and Control Algorithms

Traditional algorithms from computer science that are heuristic in nature can be used for this task. These are algorithms like A* Algorithm, D* Algorithm, etc. For these algorithms to work we need to have localization of the vehicle during the whole time. Localization is accomplished through sensors such as GPS as well as simultaneous localization and mapping (SLAM) techniques.

SLAM is used when there is no GPS availability such as underground or enclosed spaces for example. SLAM generates a map of the environment and at the same time estimates the state of a vehicle. The map is composed of landmarks or obstacles in order to represent the environment. SLAM is used in applications where the map is not available and needs to be constructed. It uses sensors and special algorithms that create models of the data in order to produce the map.

B. *Object Detection Algorithms*

Object detection is one of the most important tasks that AI has to handle in a moving vehicle. These algorithms are an area of active research and they rely on different sensor data. Object detection can be based on cameras or lidars, radars and other types of sensors. The algorithms used are normally deep learning algorithms which use some type of a neural network to do the job.

Challenges in Traditional AI Techniques

AI Path Planning algorithms such as A* algorithm generates the path for the robot's navigation from the initial starting position to the target position in the form of a set of coordinate points. However, this approach fails to consider the stochastic nature of the real world where a dynamic environment is prevalent. For example; navigating a robot over a surface with higher friction coefficient requires operating at a higher speed as compared to navigating a robot over a smooth surface with a lesser friction coefficient over the same time interval and path distance. Providing a set of path coordinates alone in this case is insufficient for performing autonomous navigation as the velocity to be supplied to the robot varies according to the surface characteristics. Another flaw in this technique is that the path planning algorithms only generate a set of coordinate poses but the velocity to be supplied to the robot for its navigation has to be determined by implementing a different algorithmic strategy. Such deterministic approaches make it harder to implement Autonomous Navigation in real life that yields high accurate results.

Deep Learning Approach

We propose a Deep Learning approach to tackle the shortcomings of path planning algorithms by implementing Supervised Learning and/or Reinforcement Learning Techniques. Such Deep Learning techniques require large amounts of labeled datasets gathered by navigating the robot over multiple time steps several times for training and learning. Therefore, Acquisition of dataset is of foremost importance towards solving any Machine Learning problem.

We intend to create datasets for Supervised Learning tasks using a custom built robotic platform. This requires the control of different actuators and sensors, operated by an easy-to-use mechanism. We came up with a Joystick based user interface for this purpose.

SYSTEM DESIGN AND IMPLEMENTATION

List of Sensors

Navigation Control	Vision	Audio
LiDAR	Fixed cameras	6 Channel Mic
Accelerometer	Servo cameras (Pan and Tilt): Front and Back	Speaker
Magnetometer	Depth camera (RGB-D and IR)	
Gyroscope		

Robotic Platform

Our robot is a lab version of a service robot. It is built as a general platform for experimentation. It can be customized for a number of applications. It is equipped with a variety of sensors and actuators which produce multimedia data and permit development of different AI Algorithms.

Sensors and actuators belong to the following categories:

- Navigation
- Vision
- Audio

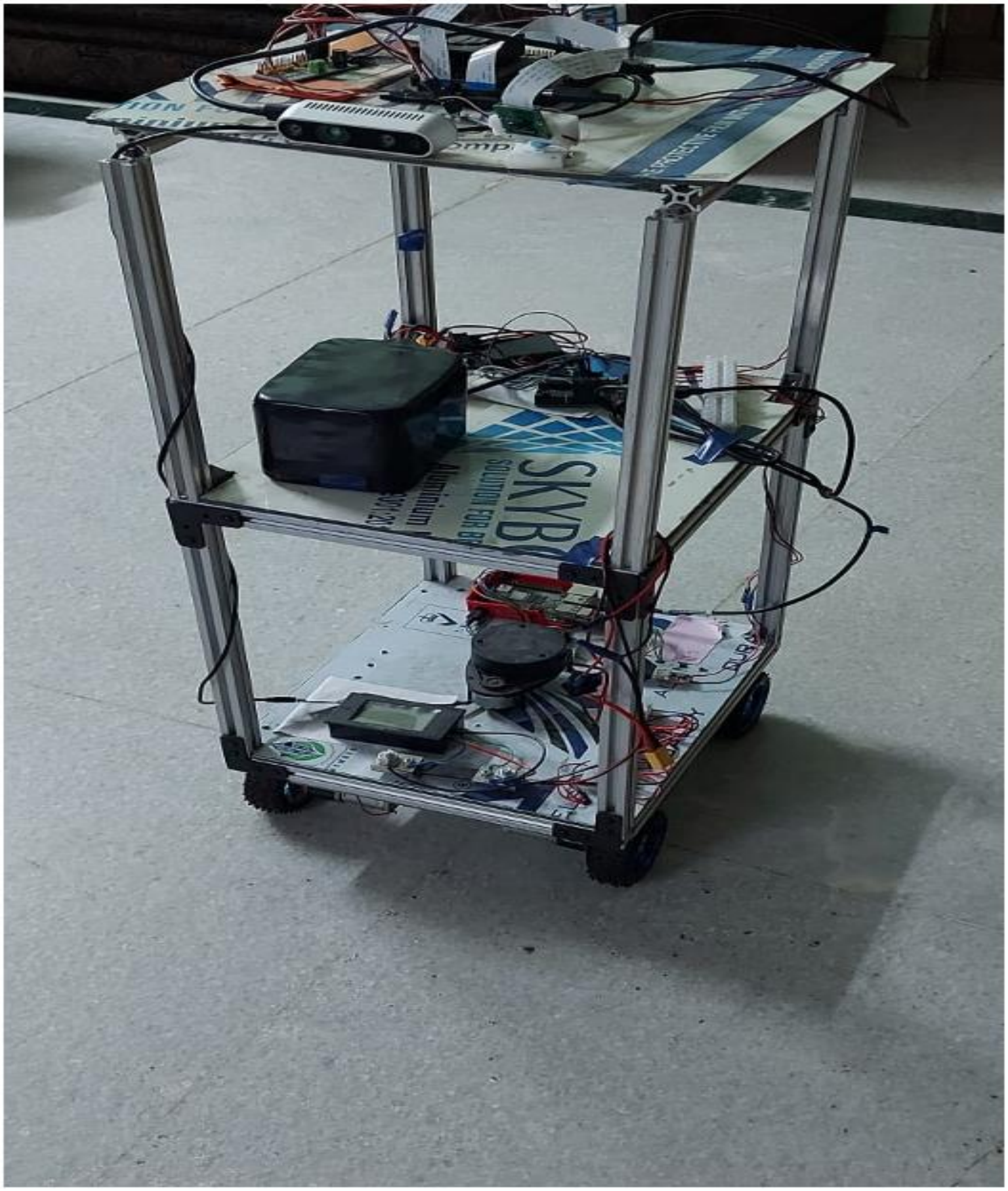

Fig 7: Our service robot

Joystick Control Interface

A joystick controller acts as a general purpose user-interface that facilitates in navigating real robots and helps in acquisition of data to enable development of various AI Algorithms.

It has 2 main functionalities:

1. Provides high flexibility and easy-to-use user interface for controlling / navigating the robot.
2. Acts as a single super controller enabling control of various sensors and actuators present in the robot, not just limiting to navigation but also audio(mic, speaker) and vision(cameras) control.

Fig 8: Joystick Interface

The following keys of the joystick control specialized robotic features.

Navigation Control

L3: Velocity control

R3: Direction control

Start: Start recording

Select: Stop recording

Vision Control

R1: Enable servo-camera(rear)

R2: Enable servo-camera(front)

L1: Click Photo

X: Tilt down

Triangle: Tilt up

Circle: Pan right

Square: Pan left

Architecture

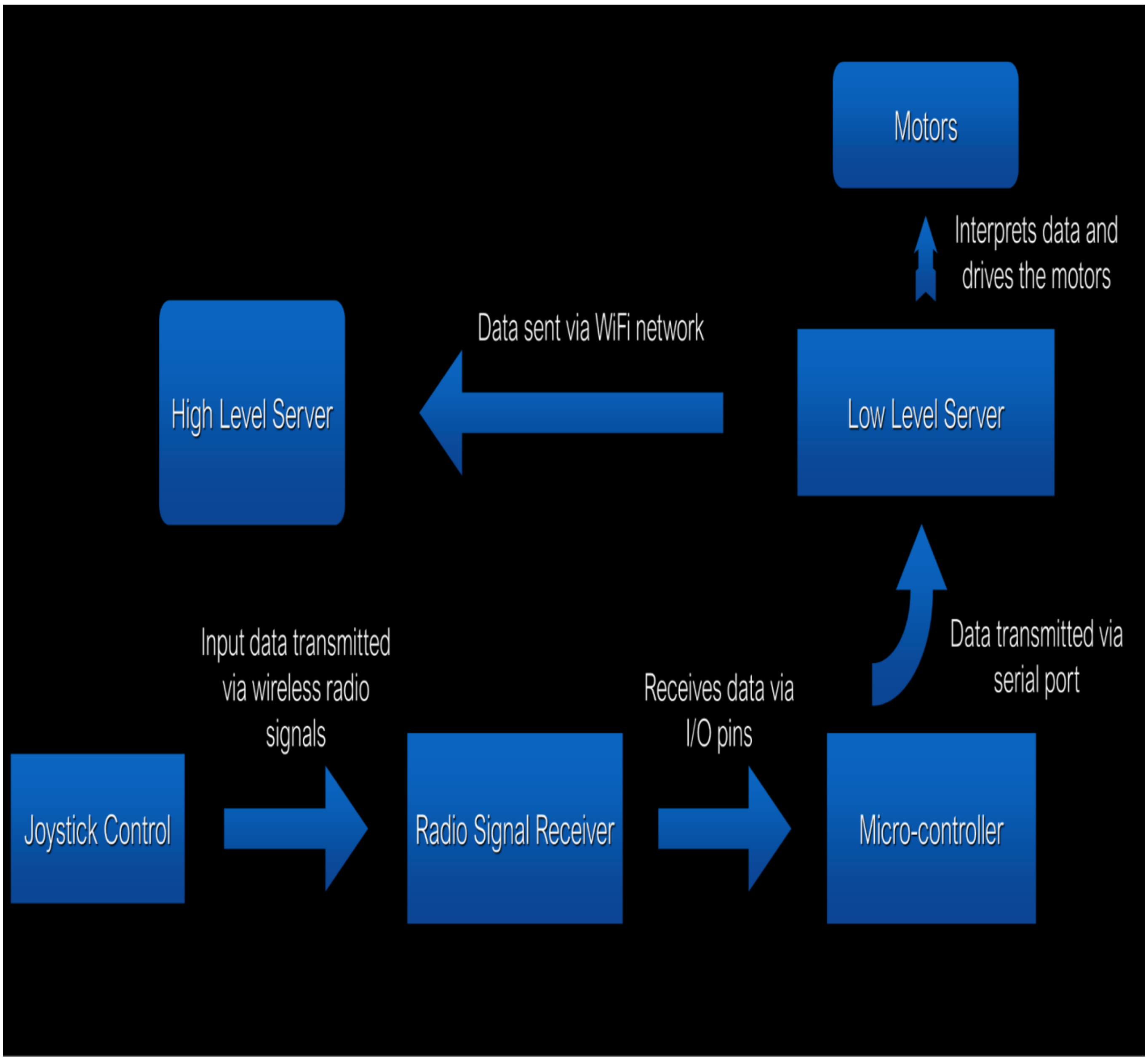

Fig 9: High Level Architecture

Client Design

The key elements in the client design are listed as follows:

1. Joystick Control
2. Radio Signal Receiver
3. Microcontroller

As the user starts controlling the joystick, the joystick interface interprets the logic and transmits the input data to the radio signal receiver board present in the robot. The data is transmitted to the radio signal receiver in the form of radio signals. The design architecture can be visualized using the above diagram.

As the signals are being transmitted in real time, the radio signal receiver board receives these input signals and transmits these signals to the microcontroller through I/O pins.

The microcontroller reads the I/O pins and interprets the data. It converts the input data into a Python Data Structure and sends the data to the server.

Server Design

The key elements of the server design are as follows:

1. Low Level Server
2. High Level Server

The Low Level Server receives the input data from the microcontroller through the serial port. The Low Level Server interprets and manipulates the logic and performs the navigation.

The data is transmitted to a High Level Server through WIFI Network.

The High Level Server is responsible for performing high level functions such as controlling Vision and Audio features. The High Level Server stores all the data into a database and runs the Deep Learning model. The model, after training through all the data, performs the Autonomous Navigation.

RESULTS

Indoor Robot Navigation Dataset(IRND)

Data was collected by operating the robot using a Joystick control over 2 surfaces with varying surface characteristics and the results are compiled as below:

Each file of the dataset is a *json file* that stores a sequence data recorded from one episode of robotic control, i.e, data collected from robotic sensors and actuators while controlling the robot from an initial position at rest to the target position. The data collected at each time step of an episode contains the following records:

- **num_records**: no. of records in an episode
- **direction**: clockwise/counter-clockwise depending upon the direction of movement of robot
- **pose**: current location/position of robot
- **brake**: 1 if brake is applied, 0 otherwise
- **angles**: obtained from LiDAR. Ranges from -180 degrees to +180 degrees
- **dists**: obstacle distances obtained from LiDAR corresponding to respective LiDAR angles
- **horn**: 1 if pressed, otherwise 0
- **counts_left**: speed of left wheel. Max speed = 2000 counts
- **counts_right**: speed of right wheel. Max speed = 2000 counts

Experimentally it was found that approximately 800 counts corresponds to 1 ft/s speed

Data:

{"num_records": 47, "data": [{"direction": "cw", "pose": {"seq": 2723, "stamp": 1.6624635950197248e+18, "theta": 0.05912135153023863, "y": 0.3183, "x": 1.3248, "heading": null}, "brake": 1, "dists": [6.072, 6.072, 6.072, 6.072, 6.072, 6.04, 6.04, 6.04, 6.04, 6.04, 3.652, 3.652, 3.652, 3.644, 3.644, 3.628, 3.616, 3.62, 3.632, 3.644, 3.664, 4.104, 4.104, 4.104, 4.104, 4.112, 4.104, 4.056, 3.876, 3.884, 3.884, 3.884, 3.888, 3.896, 3.9, 3.9, 3.904, 3.912, 3.912, 3.916, 3.92, 3.932, 3.936, 3.936, 3.944, 3.948, 3.952, 3.96, 3.964, 3.964, 3.972, 3.98, 3.988, 4.0, 4.008, 4.016, 4.02, 4.032, 4.044, 4.056, 4.056, 4.06, 4.072, 4.08, 4.096, 4.096, 4.112, 4.12, 4.12, 4.128, 4.128, 4.144, 4.152, 4.16, 4.168, 4.176, 4.192, 4.2, 4.208, 4.216, 4.232, 4.232, 4.248, 4.256, 4.272, 4.288, 4.296, 4.32, 4.344, 4.368, 4.384, 4.4, 4.4, 4.408, 4.416, 4.432, 4.44, 4.448, 4.464, 4.48, 4.488, 4.504, 4.52, 4.536, 4.536, 4.552, 4.576, 4.6, 4.624, 4.656, 4.696, 4.728, 4.744, 4.744, 4.736, 4.736, 4.736, 4.688, 0.202, 0.202, 0.202, 0.202, 0.202, 0.202, 0.202, 0.202, 0.202, 0.192, 0.191, 7.808, 7.808, 7.792, 0.193, 0.197, 3.776, 3.776, 3.752, 3.732, 4.384, 4.384, 4.384, 4.384, 4.432, 4.48, 4.488, 4.48, 4.472, 4.48, 4.44, 4.44, 4.44, 4.368, 4.368, 4.368, 4.368, 4.352, 4.352, 4.352, 4.336, 4.32, 4.304, 4.288, 4.264, 4.248, 4.248, 4.232, 4.224, 4.24, 4.248, 4.2, 4.184, 4.216, 4.192, 4.168, 4.152, 4.152, 4.144, 4.136, 4.128, 4.128, 4.12, 4.112, 4.104, 4.096, 4.096, 4.084, 4.072, 4.072, 4.064, 4.064, 4.048, 4.04, 4.04, 4.028, 4.02, 4.004, 3.992, 3.988, 3.988, 3.98, 3.98, 3.972, 3.968, 3.968, 3.96, 3.956, 3.952, 3.948, 3.944, 3.94, 3.94, 3.936, 3.936, 3.928, 3.924, 3.92, 3.916, 3.916, 3.912, 3.912, 3.904, 3.904, 3.904, 3.904, 3.9, 3.9, 3.896, 3.892, 3.888, 3.888, 3.888, 3.888, 3.88, 3.888, 3.888, 3.888, 3.884, 3.88, 3.876, 3.88,

3.88, 3.876, 3.88, 3.88, 3.88, 3.872, 3.872, 3.872, 3.88, 3.88, 3.88, 3.884, 3.884, 3.888, 3.888, 3.88, 3.884, 3.892, 3.892, 3.896, 3.896, 3.892, 3.896, 3.9, 2.46, 2.46, 2.46, 2.46, 3.92, 3.92, 3.92, 3.924, 3.928, 3.936, 3.94, 3.94, 3.94, 3.948, 3.952, 3.956, 2.516, 2.516, 2.516, 2.516, 3.06, 3.06, 3.06, 2.932, 2.932, 2.932, 2.92, 2.92, 2.908, 2.908, 2.884, 2.864, 2.848, 2.836, 2.82, 2.816, 2.816, 2.812, 2.804, 2.804, 2.812, 2.812, 2.812, 2.816, 2.824, 2.816, 2.816, 2.812, 2.804, 2.804, 2.816, 2.828, 2.828, 2.836, 2.856, 2.888, 2.908, 2.96, 4.272, 4.272, 4.224, 4.224, 4.224, 4.16, 4.108, 4.068, 4.028, 3.992, 3.948, 3.892, 3.848, 3.812, 3.812, 3.768, 3.736, 3.712, 1.378, 1.378, 1.378, 1.378, 1.342, 1.312, 1.288, 1.276, 1.276, 1.264, 1.248, 1.236, 1.228, 1.22, 1.212, 1.202, 1.196, 1.188, 1.182, 1.178, 1.176, 1.17, 1.168, 1.168, 1.172, 1.176, 1.134, 1.134, 1.134, 1.134, 1.136, 1.138, 1.142, 1.146, 1.056, 1.056, 1.062, 1.062, 1.062, 1.062, 1.07, 1.12, 1.12, 1.122, 1.134, 1.154, 1.154, 1.168, 0.23, 0.23, 0.23, 0.23, 0.232, 1.178, 1.178, 1.178, 1.178, 0.22, 0.218, 0.218, 0.218, 0.227, 0.227, 0.227, 0.228, 0.228, 0.227, 1.634, 1.634, 1.634, 1.634, 1.634, 1.63, 1.63, 1.632, 1.65, 1.666, 1.666, 1.662, 1.658, 1.654, 1.65, 1.646, 1.638, 1.634, 1.628, 1.626, 1.622, 1.62, 1.616, 1.616, 1.612, 1.61, 1.604, 1.602, 1.598, 1.594, 1.59, 1.588, 1.584, 1.58, 1.578, 1.578, 1.578, 1.574, 1.57, 1.566, 1.564, 1.562, 1.558, 1.556, 1.556, 1.556, 1.552, 1.552, 1.55, 1.548, 1.544, 1.542, 1.542, 1.54, 1.536, 1.532, 1.532, 1.528, 1.524, 1.524, 1.524, 1.522, 1.522, 1.522, 1.52, 1.518, 1.518, 1.518, 1.516, 1.514, 1.514, 1.514, 1.514, 1.514, 1.514, 1.512, 1.51, 1.508, 1.506, 1.51, 1.51, 1.508, 1.508, 1.506, 1.506, 1.506, 1.508, 1.506, 1.504, 1.504, 1.504, 1.502, 1.504, 1.506, 1.502, 1.502, 1.502, 1.502, 1.502, 1.504, 1.506, 1.506, 1.504, 1.972, 1.972, 1.972, 1.974, 1.976, 1.976, 1.982, 1.984, 1.984, 1.986, 1.862, 1.862, 1.852, 1.846, 1.84, 1.836, 1.834, 1.834, 1.834, 1.834, 1.838, 1.844, 1.844, 1.848, 1.85, 1.854, 1.858, 1.86, 1.86, 1.864, 1.87, 1.878, 1.882, 1.882, 1.886, 1.892, 1.894, 1.898, 1.902, 1.906, 1.78, 1.78, 1.774, 1.774, 1.774, 1.774, 1.768, 1.764, 1.77, 1.772, 1.754, 1.754, 1.768, 1.77, 1.778, 2.136, 2.136, 2.136, 2.136, 2.136, 2.136, 2.156, 2.176, 2.184, 2.184, 2.164, 2.14, 2.144, 2.164, 2.172, 2.18, 2.188, 2.192, 2.204, 2.216, 2.22, 2.22, 2.228, 2.236, 2.244, 2.248, 2.252, 2.256, 2.264, 2.272, 2.284, 2.284, 2.296, 2.3, 2.316, 2.344, 2.38, 0.229, 0.229, 0.229, 0.229, 0.228, 0.228, 0.229, 0.229, 0.218, 0.218, 2.48, 2.48, 0.216, 0.215, 2.564, 2.56, 0.223, 0.224, 2.58, 2.58, 0.225, 0.225, 0.225, 0.225, 2.512, 2.512, 2.512, 2.512, 2.496, 2.496, 2.496, 2.488, 2.476, 2.46, 2.448, 2.44, 2.424, 2.412, 2.404, 2.404, 2.392, 2.384, 2.372, 2.364, 2.352, 2.34, 2.332, 2.32, 2.308, 2.3, 2.292, 2.292, 2.288, 2.28, 2.268, 2.26, 2.252, 2.24, 2.232, 2.224, 2.212, 2.208, 2.204, 2.196, 2.196, 2.192, 2.204, 2.228, 2.252, 2.28, 2.312, 2.336, 2.364, 2.396, 2.396, 2.424, 2.452, 2.484, 2.508, 2.532, 2.56, 6.568, 6.568, 6.568, 6.568, 6.568, 6.544, 6.52, 6.504, 6.488, 6.48, 6.456, 6.424, 6.408, 6.392, 6.392, 6.296, 6.208, 6.184, 6.16, 6.136, 6.128, 6.112, 6.104, 6.096, 6.072, 6.064, 6.064, 6.048, 6.048, 6.04, 6.04, 6.032, 6.024, 6.024, 6.024, 6.016, 6.016, 6.008, 6.0, 6.0, 6.0, 5.992, 5.976, 5.48, 5.48, 5.48, 5.456, 5.456, 5.448, 5.448, 5.448, 5.448, 5.448, 5.448, 5.44, 5.424, 5.424, 5.432, 5.424, 5.424, 5.424, 5.456, 5.456, 5.472, 5.496, 5.52, 5.456, 5.456, 5.456, 5.384, 5.36, 2.608, 2.608, 2.608, 2.608, 2.604, 2.6, 2.612, 2.636, 2.656, 5.52, 5.52, 5.888, 5.888, 5.888, 5.888, 5.888, 5.888, 5.888, 5.888, 5.912, 2.572, 2.572, 2.572, 2.572, 2.552, 2.552, 2.548, 2.548, 2.56, 2.572, 2.58, 2.988, 2.988, 2.988, 2.992, 2.992, 2.996, 2.996, 3.0, 3.004, 3.004, 3.004, 3.008, 3.016, 3.024, 3.028, 3.032, 3.036, 3.04, 3.048, 3.056, 3.06, 3.06, 3.064, 3.072, 3.08, 3.092, 3.096, 3.1, 3.108, 3.116, 3.116, 3.124, 3.128, 3.132, 3.14, 3.144, 3.144, 3.148, 3.156, 3.16, 3.164, 3.164, 3.168, 3.176, 3.192, 3.208, 3.22, 3.228, 3.236, 3.24, 3.248, 3.248, 3.596, 3.6, 3.6, 3.6, 3.612, 3.632, 3.64, 3.648, 3.66, 3.66, 3.672, 3.684, 3.696, 3.712, 3.724, 3.736, 3.752, 3.768, 3.784, 3.784, 3.796, 3.808, 3.824, 3.836, 3.848, 3.868, 3.884, 3.9, 3.908, 3.92, 3.92, 3.94, 3.956, 3.968, 3.984, 4.024, 4.068, 4.084, 4.104, 4.104, 0.196, 0.196, 0.196, 0.196, 0.198, 0.198, 0.198, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.191, 0.191, 0.191, 0.191, 0.191, 0.19, 0.19, 0.188, 0.191, 0.191, 0.196, 0.196, 0.196, 0.196, 0.197, 0.197, 0.197, 0.19, 0.19, 0.19, 0.189, 0.189, 0.189, 0.19, 4.416, 4.416, 4.416, 4.416, 4.384, 4.384, 4.384, 4.384, 4.072, 4.072, 4.052, 4.04, 4.032, 4.032, 4.048, 4.064, 5.016, 5.016, 5.04, 5.04, 5.104, 5.192, 5.288, 5.384, 5.44, 5.456, 5.52, 5.52, 5.616, 5.744, 5.856, 5.912,

5.992, 6.096, 6.192, 6.424, 6.424, 6.424, 6.4, 6.4, 6.384, 6.376, 6.368, 6.36, 6.352,
6.336, 6.328, 6.328, 6.32, 6.304, 6.288, 6.272, 6.264, 6.24, 6.224, 6.208, 6.192,
6.192, 6.184, 6.184, 6.184, 6.168, 6.16, 6.152, 6.152, 6.152, 6.144, 6.136, 6.136,
6.128, 6.12, 6.12, 6.12, 6.12, 6.112, 6.112, 6.104, 6.096, 6.096, 6.096, 6.088,
6.088, 6.088, 6.08, 6.08, 6.08, 6.08], "angles": [-3.14159, -3.13611, -3.13063, -
3.12514, -3.11966, -3.11418, -3.0429, -3.03742, -3.03194, -3.02646, -3.02097, -
3.01549, -3.01001, -3.00452, -2.99904, -2.99356, -2.98808, -2.98259, -2.97711, -
2.97163, -2.96615, -2.96066, -2.95518, -2.9497, -2.94422, -2.93873, -2.93325, -
2.92777, -2.92228, -2.9168, -2.91132, -2.90584, -2.90035, -2.89487, -2.88939, -
2.88391, -2.87842, -2.87294, -2.86746, -2.86197, -2.85649, -2.85101, -2.84553, -
2.84004, -2.83456, -2.82908, -2.8236, -2.81811, -2.81263, -2.80715, -2.80166, -
2.79618, -2.7907, -2.78522, -2.77973, -2.77425, -2.76877, -2.76329, -2.7578, -
2.75232, -2.74684, -2.74135, -2.73587, -2.73039, -2.72491, -2.71942, -2.71394, -
2.70846, -2.70298, -2.69749, -2.69201, -2.68653, -2.68105, -2.67556, -2.67008, -
2.6646, -2.65911, -2.65363, -2.64815, -2.64267, -2.63718, -2.6317, -2.62622, -
2.62074, -2.61525, -2.60977, -2.60429, -2.5988, -2.59332, -2.58784, -2.58236, -
2.57687, -2.57139, -2.56591, -2.56043, -2.55494, -2.54946, -2.54398, -2.53849, -
2.53301, -2.52753, -2.52205, -2.51656, -2.51108, -2.5056, -2.50012, -2.49463, -
2.48915, -2.48367, -2.47818, -2.4727, -2.46722, -2.46174, -2.45625, -2.45077, -
2.44529, -2.43981, -2.43432, -2.35208, -2.3466, -2.34112, -2.33563, -2.33015, -
2.32467, -2.31919, -2.3137, -2.30822, -2.28629, -2.28081, -2.27532, -2.26984, -
2.26436, -2.25888, -2.25339, -2.24791, -2.24243, -2.23695, -2.23146, -2.08891, -
2.08343, -2.07795, -2.07246, -2.06698, -2.0615, -2.05602, -2.05053, -2.04505, -
2.03957, -2.03409, -2.0286, -2.02312, -2.01764, -2.01215, -2.00667, -2.00119, -
1.99571, -1.99022, -1.98474, -1.97926, -1.97378, -1.96829, -1.96281, -1.95733, -
1.95184, -1.94636, -1.94088, -1.9354, -1.92991, -1.92443, -1.91895, -1.91347, -
1.90798, -1.9025, -1.89702, -1.89153, -1.88605, -1.88057, -1.87509, -1.8696, -
1.86412, -1.85864, -1.85316, -1.84767, -1.84219, -1.83671, -1.83123, -1.82574, -
1.82026, -1.81478, -1.80929, -1.80381, -1.79833, -1.79285, -1.78736, -1.78188, -

1.7764, -1.77092, -1.76543, -1.75995, -1.75447, -1.74898, -1.7435, -1.73802, -1.73254, -1.72705, -1.72157, -1.71609, -1.71061, -1.70512, -1.69964, -1.69416, -1.68867, -1.68319, -1.67771, -1.67223, -1.66674, -1.66126, -1.65578, -1.6503, -1.64481, -1.63933, -1.63385, -1.62836, -1.62288, -1.6174, -1.61192, -1.60643, -1.60095, -1.59547, -1.58999, -1.5845, -1.57902, -1.57354, -1.56805, -1.56257, -1.55709, -1.55161, -1.54612, -1.54064, -1.53516, -1.52968, -1.52419, -1.51871, -1.51323, -1.50775, -1.50226, -1.49678, -1.4913, -1.48581, -1.48033, -1.47485, -1.46937, -1.46388, -1.4584, -1.45292, -1.44744, -1.44195, -1.43647, -1.43099, -1.4255, -1.42002, -1.41454, -1.40906, -1.40357, -1.39261, -1.38713, -1.38164, -1.37616, -1.37068, -1.36519, -1.35971, -1.35423, -1.34875, -1.34326, -1.33778, -1.3323, -1.32682, -1.32133, -1.31585, -1.31037, -1.28295, -1.27747, -1.27199, -1.26651, -1.26102, -1.25554, -1.25006, -1.24458, -1.23909, -1.23361, -1.22813, -1.22264, -1.21716, -1.21168, -1.2062, -1.20071, -1.19523, -1.18975, -1.18427, -1.17878, -1.1733, -1.16782, -1.16233, -1.15685, -1.15137, -1.14589, -1.1404, -1.13492, -1.12944, -1.12396, -1.11847, -1.11299, -1.10751, -1.10202, -1.09654, -1.09106, -1.08558, -1.08009, -1.07461, -1.06913, -1.06365, -1.05816, -1.05268, -1.0472, -1.04171, -1.03623, -1.03075, -1.02527, -1.01978, -1.0143, -1.00882, -1.00334, -0.99785, -0.99237, -0.98689, -0.98141, -0.97592, -0.97044, -0.96496, -0.95947, -0.95399, -0.94851, -0.94303, -0.93754, -0.93206, -0.92658, -0.9211, -0.91561, -0.91013, -0.90465, -0.89916, -0.89368, -0.8882, -0.88272, -0.87723, -0.87175, -0.86627, -0.86079, -0.8553, -0.84982, -0.84434, -0.83885, -0.83337, -0.82789, -0.82241, -0.81692, -0.81144, -0.80596, -0.80048, -0.79499, -0.78951, -0.78403, -0.77854, -0.77306, -0.76758, -0.7621, -0.75661, -0.75113, -0.74565, -0.74017, -0.73468, -0.7292, -0.72372, -0.71823, -0.71275, -0.70727, -0.70179, -0.6963, -0.64148, -0.63599, -0.63051, -0.62503, -0.61955, -0.61406, -0.60858, -0.6031, -0.59762, -0.59213, -0.58665, -0.58117, -0.57568, -0.5702, -0.56472, -0.55924, -0.55375, -0.54827, -0.54279, -0.4441, -0.43862, -0.43313, -0.42765, -0.42217, -0.41669, -0.4112, -0.40572, -0.40024, -0.39476, -0.38927, -0.38379, -0.37831, -0.37282, -0.36734, -0.36186, -0.35638, -0.35089, -0.34541, -0.33993, -

0.33445, -0.32896, -0.32348, -0.318, -0.31251, -0.30703, -0.30155, -0.29607, -0.29058, -0.2851, -0.27962, -0.27414, -0.26865, -0.26317, -0.25769, -0.2522, -0.24672, -0.24124, -0.23576, -0.23027, -0.22479, -0.21931, -0.21383, -0.20834, -0.20286, -0.19738, -0.19189, -0.18641, -0.18093, -0.17545, -0.16996, -0.16448, -0.159, -0.15352, -0.14803, -0.14255, -0.13707, -0.13158, -0.1261, -0.12062, -0.11514, -0.10965, -0.10417, -0.09869, -0.09321, -0.08772, -0.08224, -0.07676, -0.07128, -0.06579, -0.06031, -0.05483, -0.04934, -0.04386, -0.03838, -0.0329, -0.02741, -0.02193, -0.01645, -0.01097, -0.00548, 0.0, 0.00548, 0.01097, 0.01645, 0.02193, 0.02741, 0.0329, 0.03838, 0.04386, 0.04934, 0.05483, 0.06031, 0.06579, 0.07128, 0.07676, 0.08224, 0.08772, 0.09321, 0.09869, 0.10417, 0.10965, 0.11514, 0.12062, 0.1261, 0.13159, 0.13707, 0.14255, 0.14803, 0.15352, 0.159, 0.16448, 0.16996, 0.17545, 0.18093, 0.18641, 0.19189, 0.19738, 0.20286, 0.20834, 0.21383, 0.21931, 0.22479, 0.23027, 0.23576, 0.24124, 0.24672, 0.2522, 0.25769, 0.26317, 0.26865, 0.27414, 0.27962, 0.2851, 0.29058, 0.29607, 0.30155, 0.30703, 0.31251, 0.318, 0.32348, 0.32896, 0.33445, 0.33993, 0.34541, 0.35089, 0.35638, 0.36186, 0.36734, 0.37282, 0.37831, 0.38379, 0.38927, 0.39476, 0.40024, 0.40572, 0.4112, 0.41669, 0.42217, 0.42765, 0.43313, 0.43862, 0.4441, 0.44958, 0.45506, 0.46055, 0.46603, 0.47151, 0.477, 0.48248, 0.48796, 0.49344, 0.49893, 0.50441, 0.50989, 0.51537, 0.52086, 0.52634, 0.53182, 0.53731, 0.54279, 0.54827, 0.55375, 0.55924, 0.56472, 0.5702, 0.57568, 0.58117, 0.58665, 0.59213, 0.59762, 0.6031, 0.60858, 0.61406, 0.61955, 0.66889, 0.67437, 0.67986, 0.68534, 0.69082, 0.6963, 0.70179, 0.70727, 0.71275, 0.71824, 0.72372, 0.7292, 0.73468, 0.74017, 0.74565, 0.75113, 0.75661, 0.7621, 0.76758, 0.77306, 0.77854, 0.78403, 0.78951, 0.79499, 0.89368, 0.89916, 0.90465, 0.91013, 0.91561, 0.9211, 0.92658, 0.93206, 0.93754, 0.94303, 0.94851, 0.95399, 0.95947, 0.96496, 0.97044, 0.97592, 0.98141, 0.98689, 0.99237, 0.99785, 1.00334, 1.00882, 1.0143, 1.01978, 1.02527, 1.03075, 1.03623, 1.04171, 1.0472, 1.05268, 1.05816, 1.06365, 1.06913, 1.07461, 1.08009, 1.08558, 1.09106, 1.09654, 1.10202, 1.10751, 1.11299, 1.11847, 1.12396, 1.12944, 1.13492, 1.1404, 1.14589, 1.15137, 1.15685, 1.16233, 1.16782, 1.1733, 1.17878, 1.18427, 1.18975, 1.19523, 1.20071, 1.2062, 1.21168, 1.21716, 1.22264, 1.22813, 1.23361, 1.23909, 1.24458, 1.25006, 1.25554, 1.26102, 1.26651, 1.27199, 1.27747, 1.28295, 1.28844, 1.29392, 1.2994, 1.30489, 1.31037, 1.31585, 1.32133, 1.32682, 1.3323, 1.33778, 1.34326, 1.34875, 1.35423, 1.35971, 1.36519, 1.37068, 1.37616, 1.38164, 1.38713, 1.39261, 1.39809, 1.40357, 1.40906, 1.41454, 1.42002, 1.4255, 1.43099, 1.43647, 1.44195, 1.44744, 1.45292, 1.4584, 1.46388, 1.46937, 1.47485, 1.48033, 1.48581, 1.4913, 1.49678, 1.50226, 1.50775, 1.51323, 1.51871, 1.52419, 1.52968, 1.53516, 1.54064, 1.54612, 1.55161, 1.55709, 1.56257, 1.56806, 1.57354, 1.57902, 1.5845, 1.58999, 1.59547, 1.60095, 1.60643, 1.61192, 1.6174, 1.62288, 1.62836, 1.63385, 1.63933, 1.64481, 1.6503, 1.65578, 1.66126, 1.66674, 1.67223, 1.67771, 1.68319, 1.68867, 1.69416, 1.69964, 1.72157, 1.72705, 1.73254, 1.73802, 1.7435, 1.74898, 1.75447, 1.75995, 1.76543, 1.77092, 1.7764, 1.78188, 1.78736, 1.79285, 1.79833, 1.80381, 1.80929, 1.81478, 1.82026, 1.82574, 1.83123, 1.83671, 1.84219, 1.84767, 1.85316, 1.85864, 1.86412, 1.8696, 1.87509, 1.88057, 1.88605, 1.89154, 1.89702, 1.9025, 1.90798, 1.91347, 1.91895, 1.92443, 1.92991, 1.9354, 1.94088, 1.94636, 1.95184, 1.95733, 1.96281, 1.96829, 1.97378, 1.97926, 1.98474, 1.99022, 1.99571, 2.00119, 2.00667, 2.01215, 2.01764, 2.02312, 2.0286, 2.03409, 2.03957, 2.04505, 2.05053, 2.05602, 2.0615, 2.06698, 2.07246, 2.07795, 2.08343, 2.08891, 2.0944, 2.09988, 2.10536, 2.11084, 2.11633, 2.12181, 2.12729, 2.13277, 2.13826, 2.14374, 2.14922, 2.15471, 2.16019, 2.16567, 2.17115, 2.17664, 2.18212, 2.1876, 2.19308, 2.19857, 2.20405, 2.20953, 2.21501, 2.2205, 2.22598, 2.23146, 2.23695, 2.24243, 2.24791, 2.25339, 2.25888, 2.26436, 2.26984, 2.27532, 2.28081, 2.35208, 2.35757, 2.36305, 2.36853, 2.37401, 2.3795, 2.38498, 2.39046, 2.39594, 2.40143, 2.40691, 2.41239, 2.41788, 2.42336, 2.42884, 2.43432, 2.43981, 2.44529, 2.45077, 2.45625, 2.46174, 2.46722, 2.4727,

2.47818, 2.48367, 2.48915, 2.49463, 2.50012, 2.5056, 2.51108, 2.51656, 2.52205, 2.52753, 2.53301, 2.53849, 2.54398, 2.54946, 2.6646, 2.67008, 2.67556, 2.68105, 2.68653, 2.69201, 2.69749, 2.70298, 2.70846, 2.71394, 2.71942, 2.72491, 2.73039, 2.73587, 2.74136, 2.74684, 2.75232, 2.7578, 2.76329, 2.76877, 2.77425, 2.77973, 2.78522, 2.7907, 2.79618, 2.80166, 2.80715, 2.81263, 2.81811, 2.8236, 2.82908, 2.83456, 2.84004, 2.84553, 2.85101, 2.85649, 2.86197, 2.86746, 2.87294, 2.87842, 2.88391, 2.88939, 2.89487, 2.90035, 2.90584, 2.91132, 2.9168, 2.92228, 2.92777, 2.93325, 2.93873, 2.94422, 2.9497, 2.95518, 2.96066, 2.96615, 2.97163, 2.97711, 2.98259, 2.98808, 2.99356, 2.99904, 3.00453, 3.01001, 3.01549, 3.02097, 3.02646, 3.03194, 3.03742, 3.0429, 3.04839, 3.05387, 3.05935, 3.06483, 3.07032, 3.0758, 3.08128, 3.08677, 3.09225, 3.09773, 3.10321, 3.1087, 3.11418, 3.11966, 3.12514, 3.13063, 3.13611], "counts_left": 0, "horn": 0, "counts_right": 0},....], }

CONCLUSION

In this project, our main goal was to design, develop and productize a Joystick controller interface and use this to gather datasets that aid advanced research in Robotics using a real world robot. We have implemented a novel multi-tiered architecture and tested it completely on a physical robot. Further we developed a rich dataset that aids experiments on Autonomous Navigation, where given a current pose and a target, we generate the speed values for the differential drive robot to control the wheels for left, right, forward, reverse motions at the desired angles. The Joystick provides an elegant method not only to generate the motion based data but also is a natural way to interact with the machine.

Our next steps will be to use this platform to research on Artificial Intelligence (AI) algorithms pertaining to computer vision, natural language processing, navigation, etc. using this system.

References

1. S Russel and Peter Norvig, “Artificial Intelligence: A Modern Approach”, 4th Edition
2. Jonatan E. Salinas-Avila et al, “Assistant Delivery Robot for Nursing Home using ROS”
3. Aniketh Gujarathi et al, “Design and Development of Autonomous Delivery Robot”, Btech project report, Visvesvaraya National Institute of Technology (VNIT), Nagpur India