AI Based Self Driving Robot Using Hybrid Energy Sources

Final Year Design Project Report

(EE491 & EE492)

Submitted By

Zain Ul Abideen	2020523
Muhammad Shahabuddin	2020352
Muhammad Jalil Arish	2020317
Fawad Arif	2020132

Advisor: Dr. Waleed Tariq Sethi Co-Advisor: Dr. Muhammad Hanif

Faculty of Electrical Engineering
GIK Institute of Engineering Sciences and Technology
Topi 23640, District Swabi, Khyber Pakhtunkhwa, Pakistan
www.giki.edu.pk

April 2024

Certificate of Ownership

It is certified that the work presented in this report, regarding our senior year design project titled,

AI Based Self Driving Robot Using Hybrid Energy Sources

has been carried out through the collective efforts of all the undersigned team members. This work is carried out under the supervision of the undersigned faculty members, using number of available resources including research articles, books, and expert opinion.

Zain Ul Abideen 2020523	Muhammad Jalil Arish 2020317
Muhammad Shahabuddin 2020352	Fawad Arif 2020132

Advisor Dr. Waleed Tariq Sethi Assistant Professor – FEE	Co-Advisor Dr. Muhammad Hanif Assistant Professor – FCSE

Dr. Arbab Abdur Rahim

Dean, Faculty of Electrical Engineering,

Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

Topi 23640, District Swabi, Khyber Pakhtunkhwa, Pakistan

Submission Date: April 2024

Final Year Design Project as a Complex Engineering Problem

It is to certify here that the final year design project (FYDP) entitled *AI based self driving robot using hybrid energy sources* is categorized as a complex engineering problem (CEP) based on the preamble (in-depth engineering knowledge) and involvement of the following attributes.

1. Depth of knowledge required.
2. Depth of analysis required.
3. Familiarity of issues
4. Interdependence

The above listed attributes are thoroughly assessed after conducing meeting on 14th September 2023 with the following final year students, who proposed the idea of the titled FYDP.

1. Fawad Arif (2020132)
2. Muhammad Jalil Arish (2020317)
3. Muhammad Shahabuddin (2020352)
4. Zain-ul-Abideen (2020523)

This project is going to be conducted in fall semester *2023* and spring semester *2024*. Further, it is submitted that the proposed idea is worthy, and the required efforts are up to the level of a final year design project.

FYDP Advisor
Dr. Waleed Tariq Sethi
Assistant Professor – FEE
Date

FYDP Co-Advisor
Dr. Muhammad Hanif
Assistant Professor – FCSE
Date

Acknowledgements

We would like to thank the university for providing us with the necessary resources and opportunities to complete our undergraduate Final Year Project. Also, for letting us use advanced software and testing tools to conduct in-depth research and analysis. We are particularly grateful for the guidance and support of our advisors, whose criticism and support were invaluable throughout the project. Encouraging comments and suggestions from the faculty also helped us improve our knowledge and skills in our field of research.

We would also like to thank our friends for their support and encouragement throughout this project. Their encouragement and support were instrumental in keeping us motivated and focused during difficult times. We would like to thank them for their criticism and suggestions that helped us improve the quality of our design project. We would also like to thank our families for their understanding and support during this challenging time. Their support and love gave us the strength to follow our dreams and achieve our goals.

We would also like to especially thank Engr. Afaq Hussain (UTS Australia) for helping us with our FYP. He has dedicated a lot of time and effort for us that our FYP gets completed. Without him our project would not have even come close to being complete. He has guided us in every aspect of our project and if he did not guide us then half of our project related to Solar energy and power converters would not have been completed. We are very grateful for all that he has done for us, and it was an honour for us to work with him.

Contents

List of Figures

List of Tables

Abstract

The project titled “AI based self driving robot using hybrid energy sources” is developed after doing research on previous projects conducted in GIKI and looking at the state of the electrical vehicle (EV) and autonomous vehicle related projects and discovering there is still research and innovation to be done in this field. The rationale behind the project is to cater to the SDGs introduced by United Nations (UN) and drive the goals of innovation and sustainability forward. The project consists of two modes of operation, one is manual mode, and the other is an autonomous mode. In manual mode we control the robot using an ESP 32 microcontroller. The ESP 32 microcontroller is connected to a Sony PS3 gaming controller via Bluetooth that transfer the commands it receives from the PS3 controller to an Arduino Mega which executes those commands. In autonomous mode the project focuses on developing a new AI-driven robot that utilizes hybrid power sources and integrates state-of-the-art technology for optimal performance and sustainability. The robot utilizes Jetson Nano for implementing YOLOv7 for object detection and YOLOP for road segmentation. The input camera stream comes from a depth camera which utilizes triangulation for depth perception which is used for finding the distance of objects. Additionally, the project includes the design and implementation of a buck converter with a Maximum Power Point Tracking (MPPT) algorithm. This converter enables the robot to charge its battery efficiently from a solar panel and conventional sources such as the high voltage/main electrical power grid. This increases the duration of the battery charge for maximum efficiency.

The goal of this endeavour has been to deliver an efficient, adaptable, and environmentally conscious autonomous robot that could revolutionize various industries with its cutting-edge capabilities and eco-friendly energy usage. The project overall is in working condition except the implementation of the fully autonomous features. This happened primarily due to lack of financing resources and hardware constraints when working with jetson nano. The part related to road understanding still needs a lot of research before it can be in useable condition on an edge computing device such as a Jetson nano. Therefore, we have left road understanding as future work for students attempting such a project after attempting two different techniques ourselves.

1 Introduction

In modern times the world has become more conscious about the affects of pollution on our environment due to numerous awareness campaigns run by different organizations and in general due to the global increase in literacy rate, so the world is slowly moving towards more and more sustainable solutions that are good for the environment hence solar energy has become a solid choice for more sustainable electricity generation options. Not only this but due to technological innovations the world is moving towards a new renaissance for Artificial Intelligence. Due to better hardware available to run different more complex models, AI is being incorporated into most facets of human life such as the education industry, the beauty industry and one such industry is the transport industry, some examples include Tesla, BMW and Waymo, therefore, our project aims to incorporate both autonomous robots as well as sustainable solar energy to create a proof-of-concept model that can use both. The detail about this project will be shared in the proceeding chapters.

1.1 Background and Motivation

Robots have gained great importance in the modern era due to advancements in chip manufacturing and design which has given rise to self-driving cars and robots which demonstrate the potential of AI to transform both the transportation and robotics industry. Other advancements that have been made are in sustainable power generation which use hybrid energy sources, including but not limited to solar power and gas batteries which offer alternatives to traditional fuels, making them ideal for sustainable energy use. The integration of artificial intelligence and hybrid power is something that has not been explored in detail, the use of which will make the robot or car not only smart but also environmentally friendly and energy efficient.

The motivation behind the project is to work on autonomous vehicular technology in general and to focus on open air warehouse delivery robots especially. The project attempts to solve these problems and contribute to the solution of urban transportation problems by developing self-propelled smart robots that use integrated electronics. The current model aims to make goods transportation robots in industries and warehouses more sustainable and efficient by using AI based technology with plans to solve the problems faced in the transportation sector as mentioned above.

The project aims to support the development of autonomous systems while solving environmental and technical problems in transportation by developing self-driving robots based on artificial intelligence using hybrid power sources. The project focuses on the integration of artificial intelligence and hybrid energy and holds promise for the creation of new technologies useful for the future. Through this project, we hope to support further research and development in robotics and artificial intelligence to create sustainable and efficient robotics.

The project in current form cannot be used as an autonomous vehicle due to many missing features and safety concerns so the current model can be used in a large industry with many buildings. The project can transport goods from one building to another autonomously. Once inside the building then the project can be controlled manually to take the goods to the exact location where it is to be unloaded. So current applications include open air goods transportation and delivery robots such as robots employed by Amazon.

1.2 Problem statements

The specific objective of this project is to develop an autonomous robot that moves from point A to point B with a 50kg weight carrying capacity using an AC source and solar panel to charge the battery to demonstrate the feasibility of hybrid energy sources in the powering of robots as well as transportation vehicles. To achieve autonomy the robot will use AI models for obstacle avoidance and road segmentation. For autonomous navigation, the robot will use GPS. The project aims to gauge the feasibility of these approaches for autonomous robots. The project will not use an AI model for decision making but will use traditional approaches such as Finite State Machine (FSM) neither will the project use any Lidar technology but will instead rely on a single depth camera for all its functionality and feedback. The project will also include its own solar system including the required buck converter and MPPT algorithm.

1.3 Scope of work and expected outcome

The scope of the project is to design, develop, and evaluate an autonomous robot, which can move from point A to point B with a 50 kg payload. Apart from an electric power supply from AC and a solar panel, the robot must have an advanced energy management system. Moreover, an AI algorithm to avoid obstacles and to detect the road is used.

The anticipated result of the project is a working model which efficiently navigates, avoids obstacles, understands its surroundings, and manages its weight. The robot, which will perform all the specified tasks, will demonstrate the capabilities of AI and hybrid sources of energy in the robotics sector and find potential applications in numerous industries. The project will further the knowledge of autonomous systems and robotics in the research field, offering critical information on the application of AI and the integration of hybrid power in the robotics sector using edge computing devices such as a jetson nano it will detail which algorithms are usable in the given context and which are not. The project will not create an autonomous vehicle with all the rigorous safety requirements that are needed in order to operate an autonomous vehicle, but it will be a proof-of-concept model which will aim to move autonomously in a controlled environment. The project will also not use any lanes for road understanding as that has already been done and proven before countless times but will aim to use road segmentation algorithms which could be useful in areas where lane lines are not available.

Overall, the project will tackle the urban transportation and logistics problems by developing an autonomous robot that is resilient and robust. The autonomous movement and the large load carriage potential of the robot means that innovation will have positive impacts in various applications, including warehouse automation, safe delivery services and enhanced industrial logistics. Through displaying the potential of AI and hybrid energy in robotics, the project will significantly contribute to the development of sustainable and efficient robotic systems. The project could also be considered a general robot that could be used in many different applications with hardware changes such as the project could be utilized in delivery services, in cleaning of roads, in lawn mowing, in an agriculture industry and many more.

Through this work, we aim not only to develop a successful model but also to contribute to the dissemination of robotics and artificial intelligence research. The knowledge gained from this project will be shared with the global robotics community, fostering collaboration and further advancements in the field. Overall, this project represents an important step toward creating intelligent and sustainable robotic systems for future applications.

1.4 Sustainable Development Goals

The development of robotics using integrated electronics aligns with multiple Sustainable Development Goals (SDGs) as defined by the United Nations. Firstly, the project contributes to the goal of ensuring affordable and clean energy by highlighting the potential of renewable energy sources such as solar power in robotics. By using AC power and solar panels, the robot demonstrates sustainable energy consumption, which is crucial for reducing greenhouse gas emissions and mitigating climate change. So, the project caters to the goal of affordable and clean energy.

Secondly, the project contributes to the goal of promoting economic growth, innovation, and infrastructure development by fostering innovation in robotics and artificial intelligence technology. The integration of artificial intelligence algorithms for navigation and obstacle avoidance, as well as the development of robust energy management systems, represents advancement in the field of autonomous systems. This innovation can lead to improved efficiency and productivity across industries, stimulating economic growth and development. Which caters to the goal of industry, innovation, and infrastructure.

Furthermore, the project aligns with the goal of creating sustainable cities and communities by addressing urban transportation challenges and using clean energy to do so. The ability of autonomous robots to efficiently navigate and transport goods in urban environments can help reduce traffic congestion and improve air quality, contributing to the development of sustainable cities.

The project by attempting to fix traffic problems and air pollution problems is also helping with the climate as less cars on the road means less CO_2 which means less global warming overall.

Overall, the development of autonomous robots using hybrid power sources demonstrates the potential of robotics and artificial intelligence technologies to drive innovation and contribute to various sustainability goals. The project supports the global development agenda for a sustainable and inclusive future by providing new solutions to transportation and energy challenges.

Table 1.1 SGD table of the project

Sr. No	Title	Compliance (Y/N)	Remarks/Justification
1	No poverty	N	
2	Zero hunger	N	
3	Good health/wellbeing	Y	By attempting to reduce air pollution the project contributes to good health and wellbeing of people living in cities.
4	Quality education	N	
5	Gender equality	N	
6	Clean water and sanitation	N	
7	Affordable and clean energy	Y	By using solar energy to power the robot the project aims to make use of solar panels more widespread which will drive investments into more clean energy therefore creating more affordable and clean energy.
8	Decent work and economic growth	N	
9	Industry, innovation and infrastructure	Y	By attempting to create an autonomous robot the project aims to innovate some industries which will benefit society over the long run.
10	Reduced Inequalities	N	
11	Sustainable Cities and	Y	By using Solar energy, the project aims to

	Communities		make cities and communities more sustainable.
12	Responsible consumption and production	N	
13	Climate action	N	
14	Life below water	N	
15	Life on land	N	
16	Peace, Justice and strong institutions	N	
17	Partnerships for the goals	N	

1.5 Complex Engineering Problem

This final year project displays the characteristics of a sophisticated engineering challenge. It necessitates extensive knowledge in robotics, artificial intelligence, edge computing, embedded systems, electric machines, power electronics, programming, and energy management systems. Multiple conflicting aspects are being considered, such as efficient navigation, obstacle avoidance, road segmentation and payload management, while also prioritizes energy efficiency and sustainability.

A comprehensive analysis is required for the successful completion of the project. This includes designing and implementing AI algorithms, integrating intricate electronic components, and optimizing energy consumption. Profound familiarity with robotics, AI, programming, and energy systems is crucial, along with an understanding of issues pertaining to autonomous systems and renewable energy sources. The project involves extensive engagement with engineers, faculty members, industry experts and foreign collaborators. Such involvement can introduce conflicting requirements and competing priorities, further intensifying the complexity of the project. Failure or suboptimal performance in the project could have far-reaching consequences.

Table 1.2 CEP attributes mapping

Sr. No	**Attribute**	**Justification**
1	Preamble - In-depth engineering knowledge	The project requires in-depth engineering knowledge in numerous fields such as Programming, AI, computer vision, microcontrollers, embedded systems, electric machines, robotics, power electronics, mechanical design etc.
2	Range of conflicting requirements	The project must balance road detection and moving on the road while avoiding any obstacles as well as navigating autonomously. Not only that but it must also manage power effectively.
3	Depth of analysis required	Extensive research and analysis are required for choosing the best AI model for all the above-mentioned requirements.
4	Depth of knowledge required	Extensive knowledge is required in the field of power electronics for converter design and creation and in AI for choosing the best model for the project requirements.
5	Familiarity of issues	Issues that can arise are related to electrical in the hardware section and in the software section issues that can arise are related to programming.
6	Extent of applicable codes	Most codes that are used in industries are applicable how ever there are exceptions as the project works with some hardware that uses old firmware so some issues with dependencies will arise.
7	Extent of stakeholder involvement and level of conflicting requirement	Stakeholders that are involved include advisors with conflicting requirements, international collaborators,

		and group members with conflicting interests.
8	Consequences	The project if not executed with care could have consequences on human health as autonomous vehicles are rigorously tested before deployment but same can not be said about robots so extensive care is needed.
9	Interdependence	The project consists of many parts some software and some hardware that are interdependent on each other for proper working.

1.6 Contribution

This project requires the design and use of a Buck converter equipped with an MPPT algorithm to provide hybrid power to the robot. The converter effectively charges the robot's battery from the solar panel and AC power, allowing the energy to be converted and used by the MPPT algorithm. This combination increases the robot's strength and stability, making it ideal for working in a variety of environments for long periods of time.

In addition, in the project, multiple algorithms such as YOLOv7 and YOLO-P algorithm based on artificial intelligence, are integrated into the robot. The algorithm allows the robot to detect and avoid obstacles, calculate distance, and navigate around itself, improving safety and efficiency.

In addition, the project is multidisciplinary and combines different expertise, and knowledge in the electrical industry as well as in the software programming industry. The aim of the project is to create solutions for robots that not only improve their performance, but also support the advancement of hybrid systems and artificial intelligence algorithms in the working of robots.

The design and implementation of this project includes careful analysis and optimization to ensure reliability and efficiency. The team validated the effectiveness of hybrid power and AI algorithms through simulation and real-life experiments. This extensive testing process ensures that the robot can operate effectively in different situations and environments.

Additionally, the results and findings of this project have broad implications for the field of robotics and renewable energy systems. They provide insight into the design and optimization of hybrid systems for autonomous robots, as well as the integration of artificial intelligence algorithms to enhance autonomy. These findings pave the way for the development of more efficient and effective tools in the future. Finally, studies on the creation of integrated power systems and integrated artificial intelligence algorithms show that these can also be used in the real world. The results of the program can be used in many sectors where robots play an important role, such as agriculture, transportation, and surveillance. Ultimately, the project represents a breakthrough in the development of autonomous robots, increasing usefulness and independence. This projects contribution lies in testing algorithms in environments they have not been tested in before using hardware in certain circumstances where it has not been used before. This will contribute to the overall scientific knowledge base and pave way to future advancements using these findings.

2 Literature Review

2.1 Literature Review

Hybrid autonomous robots could be used in many sectors such as manufacturing, transportation, search and rescue operations. For example, in manufacturing, autonomous robots can easily move around the factory floor and handle heavy objects. They can also be used in transportation, such as trucks that can travel in harsh environments and use solar power for more efficient power usage and move longer distances before requiring recharge. During search and rescue operations, autonomous robots can traverse difficult terrain and provide real-time information to rescue teams.

Autonomous robots can recognize and interpret their environment, make decisions and act independently. This allows robots to perform tasks that were once performed by humans, such as recognizing and identifying objects and making decisions based on this, such as planning routes and avoiding interference.

However, many difficulties need to be overcome before autonomous robots could be used widely. Ensuring that autonomous robots work well in an unpredictable environment is a major concern and therefore requires a specific approach. Additionally, the high cost of autonomous robot technology prevents its adoption, especially in resource-limited environments.

2.2 Inference from Literature

There have been many examples of autonomous vehicles and robots that have been deployed at different levels of automation. The levels of automation can be found on the BMW website [3]. One example of an autonomous vehicle is in Las Vegas which launched the first electric autonomous shuttle on U.S. public roads in 2017 [4]. Another example is a Self-driving shuttle running at NUS [5]. Tesla is also famous for its autonomous driving cars [6]. Different universities are also working on autonomous vehicles such as MIT [7] and Stanford [8].

Keeping the industry in mind a literature review was conducted amongst the projects done by GIKI students in the past. It was discovered that very little amount of work was attempted on autonomous robots and autonomous vehicles in general. The work that was done includes the following projects.

Table 2.1: Results of previous FYPs undertaken

YEAR	FACULTY	FYDP TITLE	WORK DONE	SUPERVISORS
2022	Electrical	Self Driving car [9]	Use small RC car Old algorithms used	Dr. Arbab Dr. Zaiwar Ali
2021	Electrical	A localized IOT Based Autonomous Delivery Robot for fast Moving Goods [10]	Commands through mobile and GPS Focused	Dr. Ahmed Kamal Dr. Zaiwar
2021	FCSE	Autonomous Driving with Toy car [11]	Line following robot.	Dr. Sajid Anwar
2019	FCSE	Autonomous Shopping Cart [12]	Cart following the person and used YOLO	Dr. Ghulam Abbas

Autonomous robots have also been attempted in different papers, some of these are listed below.

1) Complete Deep Learning model utilization: In this paper the self-driving was done only using DL models which resulted in an over-fitting problem. But performed better with steering controlling. [13]

2) LiDAR Degradation Quantification for Autonomous Driving in Rain: The autonomous car was run using lidar, but for low-reflectivity surfaces and for object detection the method was resulting in insufficient fps i.e 10fps. [14][15]

3) Self-Driving Car Implementation using Computer Vision for Detection and Navigation: an autonomous car model in simulation using YOLOv3 and Stereo vision-based camera. The model was able to detect lanes, traffic cars, obstacles, signals etc. The paper mainly provides proof of concept. [16]

The literature that has been conducted is not specific but is general meaning it has been conducted regarding autonomous vehicles and cars rather than robots targeting the industrial goods transportation. As the principle remains the same just the requirements are different.

2.3 Chapter Summary

In this chapter a basic literature review has been conducted. The review consists of three parts. One is regarding the existing implementations of autonomous vehicles done by companies and cities. Second is autonomous research that has been conducted in GIKI and other universities and finally some implementations that have been conducted by researchers and they have written research papers. The chapter also talks about the pros and cons of all these implementations.

3 Design and Methodology

This chapter will describe the project in extreme detail and will mention the reason for choosing all the necessary hardware and software.

3.1 Hardware:

The project requires motors, a motor driver for the motors, a microcontroller for controlling the motors, a power source, a DC-DC converter, a microcontroller for image processing, and a sensor for detection.

3.1.1 Motor selection:

For motor selection calculations must be done.[17][18][19]

3.1.1.1 Weight of the robot:

First, we need to assume and calculate the total weight of the robot that we will be using. The weights used here are estimates. The expected weight of the robot is 20 kg and the payload of the robot is 50 kg so overall weight is 70 kg. Now we know that the total weight that the motors must bear is 70 kg. Here by 70 kg, we essentially mean 700 N as the actual unit of weight is newton but for one motor, we will assume the weight to be 175 N.

3.1.1.2 Dimensions of the robot:

Length of the robot is 3 ft or 0.9144 m and the width of the robot is 2.5 ft or 0.762 m and the area of the robot in meters is 0.9144 m x 0.762 m which is equal to 0.697 m^2 (7.5 ft^2) so the estimated area of the robot is 0.7 m^2.

3.1.1.3 Dimensions of the wheel:

The radius of the wheel of the robot is 0.0572 m.

3.1.1.4 RPM:

We can assume the max speed of the robot to be 5 km/hr or 1.389 m/s and we can assume transmission efficiency to 85 %. Transmission efficiency in an electric vehicle or robot is the efficiency of transmission system in transmitting power from the electric motors to the wheels. The formula is also given below.

$$TransmissionEfficiency\% = \left(\frac{UsefulOutputpower}{InputPower}\right) * 100$$

Transmission efficiency = 85 %

The circumference of the wheel can be calculated from the estimated radius of the wheel.

Linear wheel travel (circumference) = 2 * π * radius of wheel = 2 * 3.14 * 0.0572 m = 0.359 m

For the required RPM to achieve a speed of 5 km/hr

$$WheelTravel = \frac{Wheel\ Circumference}{No.\,of Rotations} = \frac{0.359m}{1} = 0.359m$$

$$RPMrequired = \frac{Speed}{WheelTravel} = \frac{1.389\ m/s * 60s}{0.359m} = 232.15\ RPM$$

To achieve a speed of 5 km/hr the robot requires 232.15 rpm.

3.1.1.5 Power:

Power (P) = [Weight of robot x Speed of robot x Rolling Resistance] + [Air Density x Coefficient of drag x Area of robot x (Speed of robot)^3] + [Weight of robot x sin (Angle of inclination)]

So assuming angle of inclination as 15 ^{o}, rolling resistance is 0.02 N , air density at 1 atm and 20 ^{o} C is 1.204 kg/m^3 and the coefficient of drag is 0.82.

Power (P) = [175 N x 1.389 m/s x 0.02 N] + [1.204 kg/m^3 x 0.82 x 0.697 m^2 x (1.389 m/s)^3] + [175 N * sin(15)]

Power is found to be 51.9985 W. This is the power that is required by each motor for it to carry a 70 kg weight and be able to go up a 15 ^{o} incline.

3.1.1.6 Torque:

$$Torque = \frac{Power * 60}{2 * pi * N} = \frac{51.9985W * 60}{2 * pi * 232.15RPM} = 2.14Nm$$

This is the torque required to overcome the speed of 5 km/hr

3.1.1.7 Current:

P = V * I

V = 12 volts

$$I = \frac{51.9985W}{12V} = 4.333A$$

But as a precaution the motor that will be used in the project is going to be 7 A. These calculations have been conducted to find the current of motors. To prove that these calculations are linearly scalable a matlab code has been written. A graph is given below.

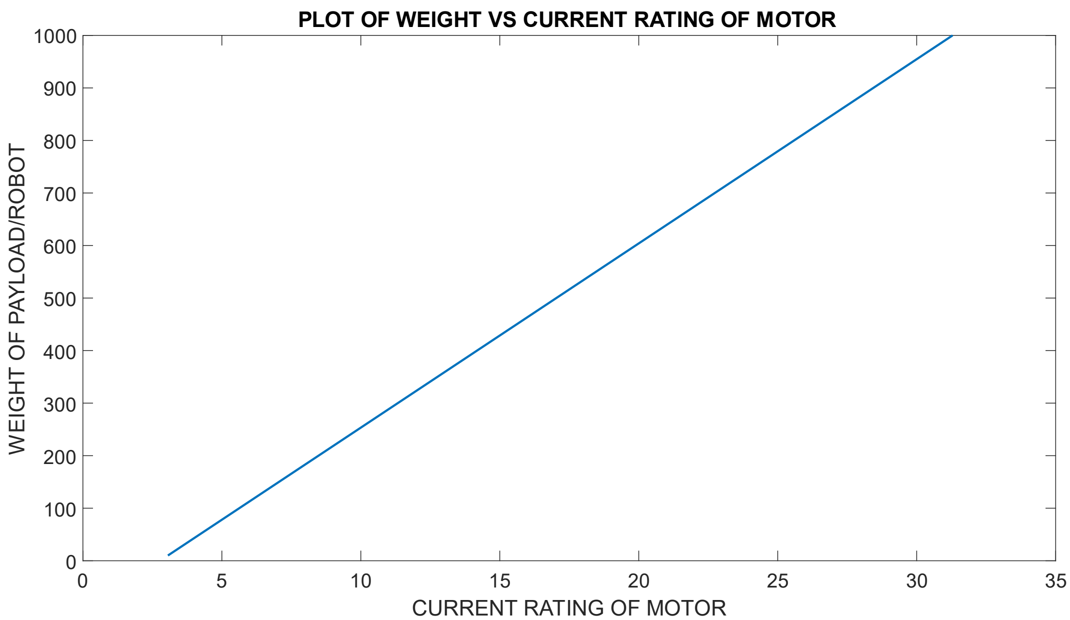


Figure 3.1: A graph of weight vs current

Now we must decide which type of motor to use. Since the system we are using is DC and we don't want to introduce extra complexity of converting to AC so DC motors will be used. Two main options are available, one is BLDC motor, and the other is PMDC motor. A table comparing both is given as under.

Table 3.1: Difference between the two types of motors

PMDC	BLDC
Speed controlling is simpler.	Speed controlling is very precise.
Control mechanism is simple as voltage and polarity adjusting is easy.	Its control mechanism is complex as it uses sensor feedback for commutation purposes [to restrict dc current to one direction].
It has low cost	It has high initial cost
It has low speed	It has high speed

We are going to choose PMDC motors as they have simpler control and are also cheaper as compared to BLDC motors.

3.1.2 Motor Driver selection:

Due to the presence of inrush current which is 2 to 3 times higher than the rated current and the rated current being 7A we will need a motor driver that has a rating of at least 21 A. So, a table of three different motor drivers is given below.[25]

Table 3.2: Table comparing three different motor drivers.

DBH 12 H-Bridge	IBT2 H-Bridge	MD25HV H-Bridge
V=6-15V	V=6-27V	V=40V
I=10-20A	I=43A	I=60A
Less thermal and current protection in it	Good high current response	Good thermal and current protection in it
Less expensive	Less expensive	Expensive

The H-bridge we will use is IBT2 due to it being easily available and less expensive

3.1.3 Body Design:

After the motors have been chosen then comes the chassis and robot base design. To keep the robot simple and give us as much space as possible the design of the base we are going with is rectangular with 3 ft by 2.5 ft dimensions. The CAD model as well as dimensions are given below.

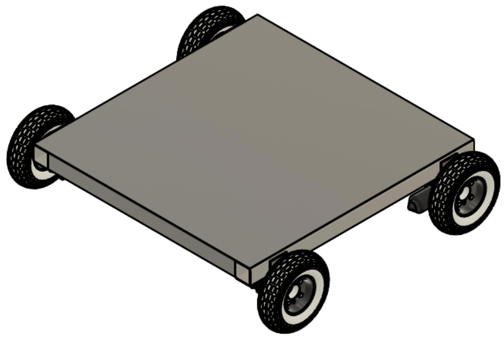


Figure 3.2: Base of the robot (CAD)

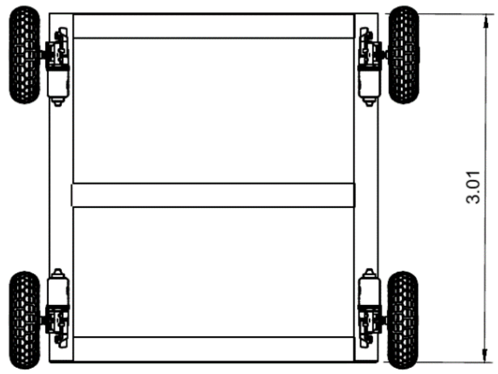


Figure 3.3: Bottom view (CAD)

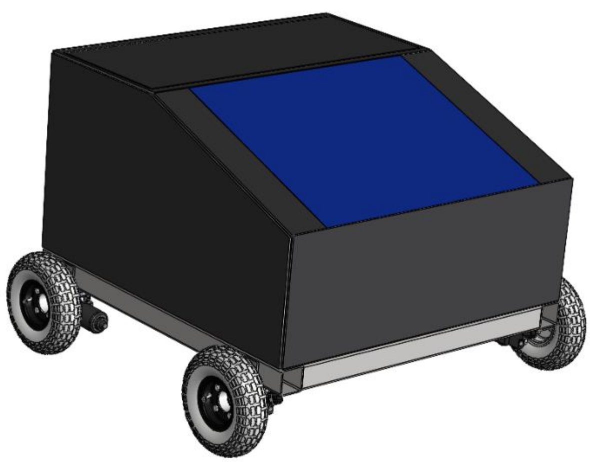


Figure 3.5: Final CAD model

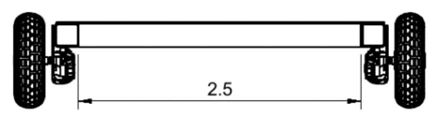


Figure 3.4: Front view (CAD)

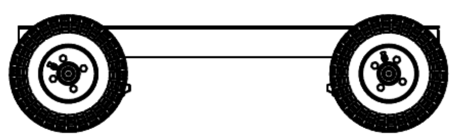


Figure 3.6: Side view (CAD)

The height of the robot is 2 ft and the blue part is the solar panel. The body is made with 16-gauge (1.6 mm) sheet metal. The base uses hollow rectangular pipes. The part where motors are connected uses sheet metal as well. The tyres in the model are slightly larger than the actual tyres.

3.1.4 Solar panel selection:

The solar panel that has been selected is a 50 W panel made by LIDO. The Current at maximum power (Imp) is 2.86 A, the Voltage at maximum power (Vmp) is 17.5 V, the short circuit current (Isc) is 3.23 A and the open circuit voltage is 21.6 V. The maximum weight is 4.5 kg and the dimensions are 540 mm x 670 mm x 30 mm.

3.1.5 Main sensor selection:

The two main sensors that we must select from include a Depth camera or a Lidar sensor. So after research here is a table comparing both of them.

Table 3.3: Table comparing stereo camera and lidar.

STEREO CAMERA	LIDAR
Affordable [20]	Very expensive [21][22]
Easy interface [23]	Complex 3D mapping [23]
Can be used independently [23]	Needs additional sensors [23]
Comparatively low accuracy [24]	Highly accurate [24]
Comparatively short range [24]	Long range [23]

Due to limitations of resources and time we have chosen the stereo camera as our main sensor.

3.1.6 Microcontroller selection:

Due to the complex nature of the robot, we will be using two different microcontrollers for control. One microcontroller is a jetson nano which will be used for processing the data received from the stereo camera and the second is the Arduino mega microcontroller which will be used for motor and other peripheral control. Here jetson nano has been chosen instead of raspberri pi because jetson nano has a built in GPU as compared to raspberri pi which relies only on a CPU. This GPU is essential for getting high FPS so that AI can be used in real time. Arduino mega has been chosen due to its high number of input output pins, as a lot of pins are required for a project of such a complex nature. On the side an ESP32 has also been used for manual mode of operation, this is due to the ESP32 having a built in Bluetooth system which allows it to connect with a PS3 controller directly. An Arduino nano has been used separately to connect with the buck converter so that it can run take in input from the current and voltage sensor and adjust the duty cycle accordingly.

3.1.7 Battery Selection:

Battery that is being used is a lead acid battery with 48 Ah rating. Other options included a dry cell, a lithium-ion battery and a Li-Po battery but all the other options are very expensive though they are better overall when it comes to battery life and battery health.

3.1.8 Overall Block diagram:

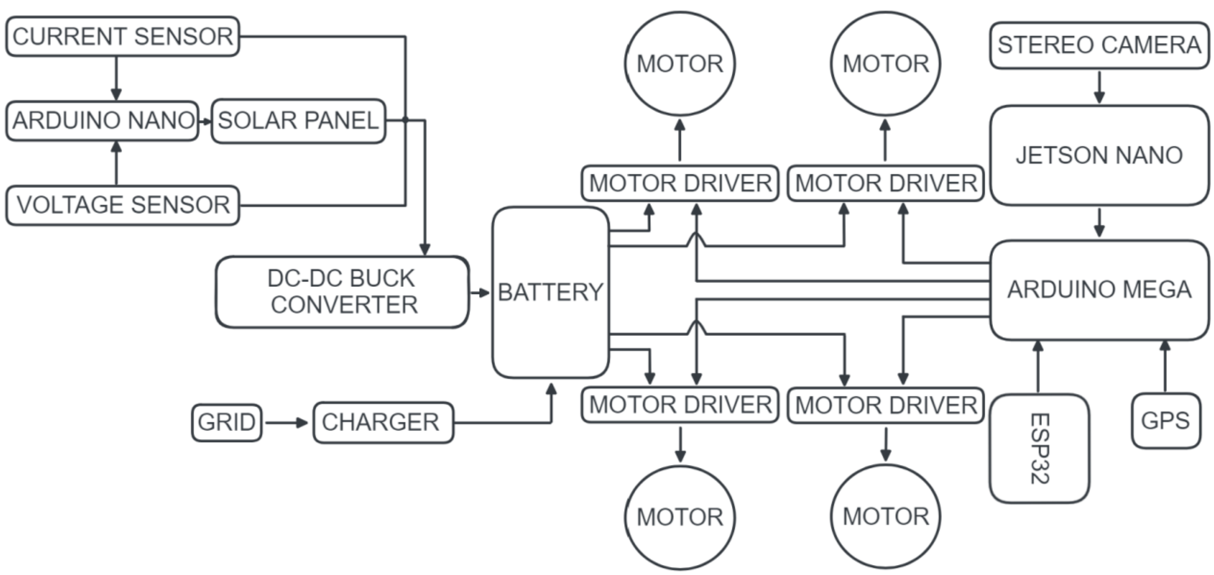


Figure 3.7: Block diagram of the project.

Jetson nano will get its input from the stereo camera and the GPS module. It will then process this input and make the best possible decision based on these inputs. It will then send its decision to the Arduino mega which will then execute the decision if it is in autonomous mode. If it is in manual mode, then it will take in input from the ESP32 only which will get its input from a PS3 controller. The Arduino mega will control the IBT2 motor drivers to change the speed of the motors based on requirement. The battery will be used to power the motors and it will be charged via two methods. Either using Solar panel or from the grid. If it is being charged from the solar panel, then it will have to be charges via a converter using an MPPT algorithm.

3.1.9 Designing of Power Converter:

For this final year project, we decided to design a unique buck-boost converter capable of both boosting and bucking the voltage simultaneously. This means that the converter can increase or decrease the voltage as needed, making it very versatile for various applications. To enhance its efficiency, MPPT (Maximum Power Point Tracking) algorithm was implemented. This algorithm helps track and utilize the maximum power output from the solar panel, ensuring optimal performance and energy efficiency by varying the duty cycle of the converter.

The decision to incorporate both boosting and bucking capabilities into our converter was driven by the need for a flexible power solution. By being able to adjust the voltage in either direction, our converter can adapt to different power requirements, making it suitable for a wide range of scenarios. Additionally, the MPPT algorithm allows us to extract the maximum power from our converter, ensuring that we can make the most of the available energy sources. Moreover, the power input into the converter will come from a solar panel, making our system environmentally friendly and sustainable. Overall, we believe that this combination of features will make our buck-boost converter a valuable and efficient component in various applications.

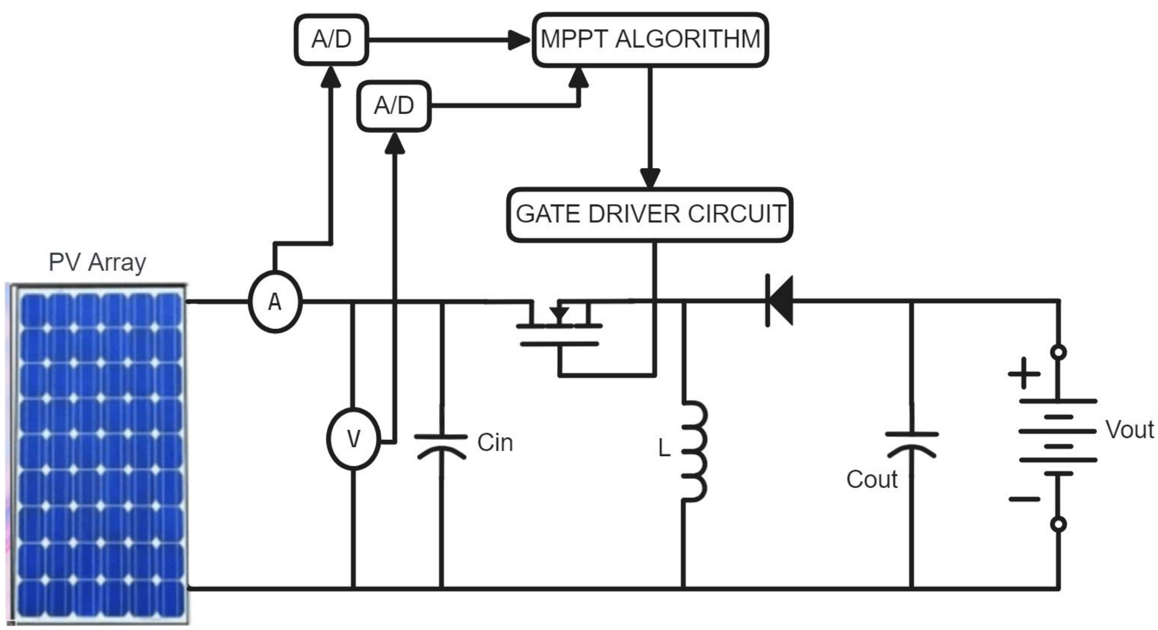


Figure 3.8: Circuit schematic of a buck boost circuit.

3.1.9.1 Simulation of buck boost Converter:

For our final year project, we conducted simulations using MATLAB Simulink to analyse the performance of our design, which includes the MPPT (Maximum Power Point Tracking) algorithm and the buck-boost converter. The MPPT algorithm plays a crucial role in our project as it helps optimize the power output from our buck-boost converter by continuously tracking and adjusting to the maximum power point of the solar panel. This ensures that our system operates at peak efficiency, maximizing the energy harvested from the solar panel.

After running the simulations, we analysed the results and generated graphs to visualize the performance of our system. These graphs illustrate how the MPPT algorithm effectively tracks the maximum power point, allowing our converter to operate efficiently under different conditions. By simulating our design, we were able to validate its functionality and performance, providing valuable insights for further optimization and implementation.

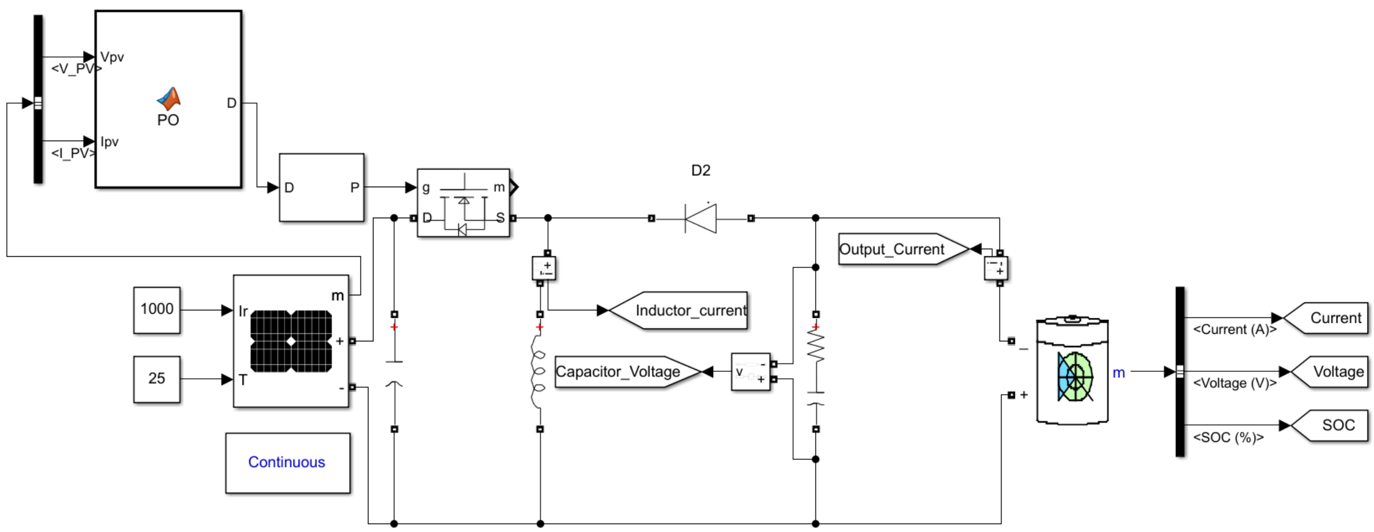


Figure 3.9: Buck-Boost converter

Following are the results of the simulation.

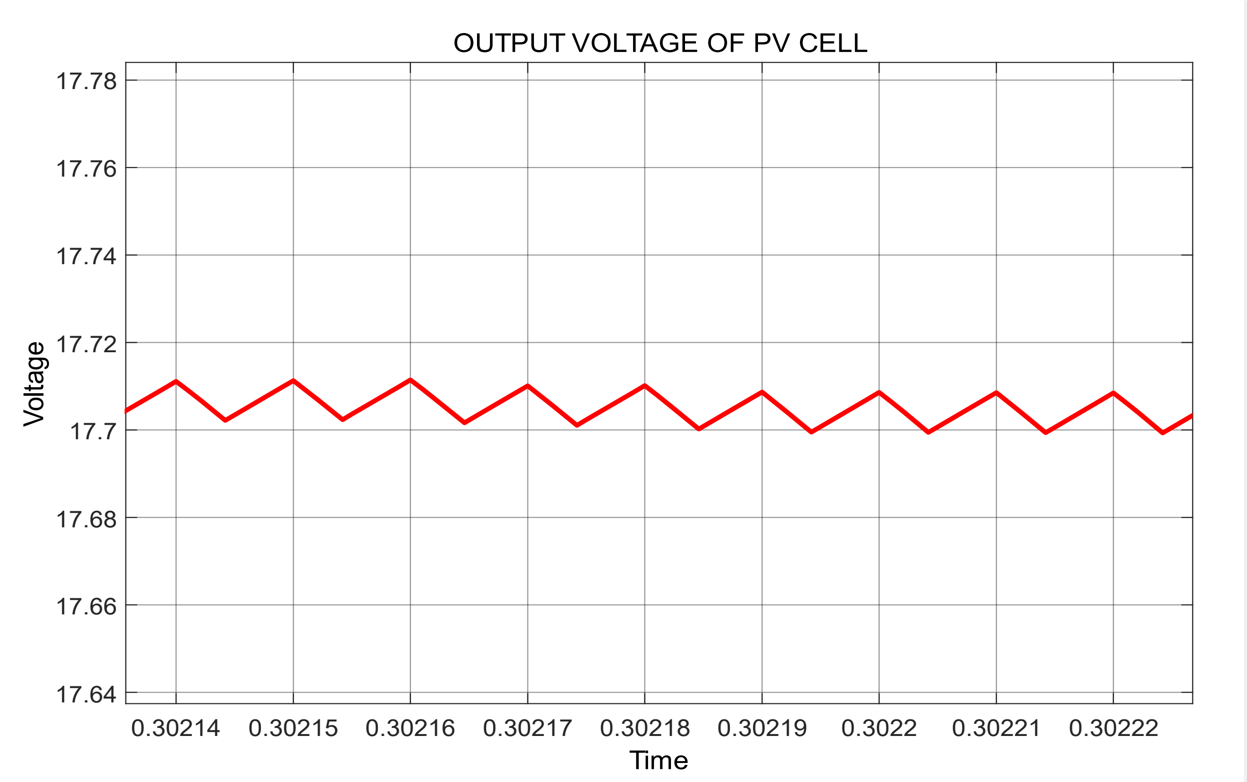


Figure 3.10: Output voltage of the PV cell (simulation)

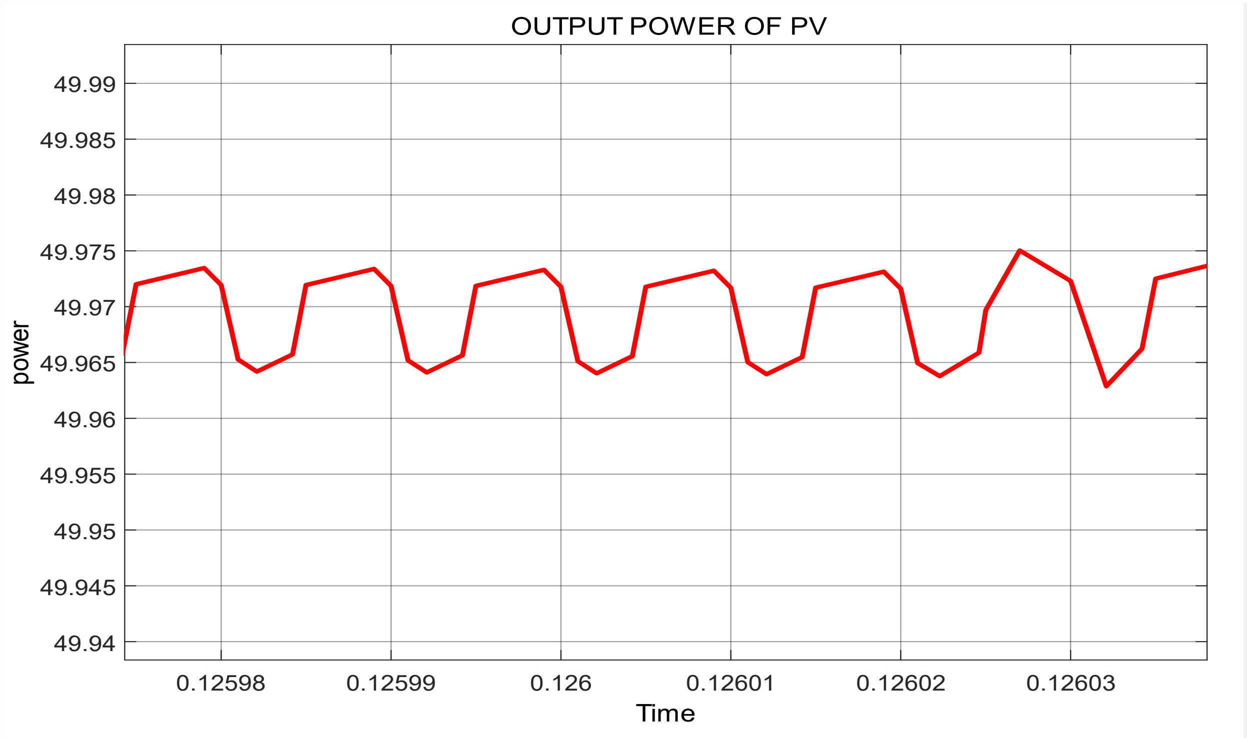


Figure 3.11: Graph of output power of the solar panel (simulation)

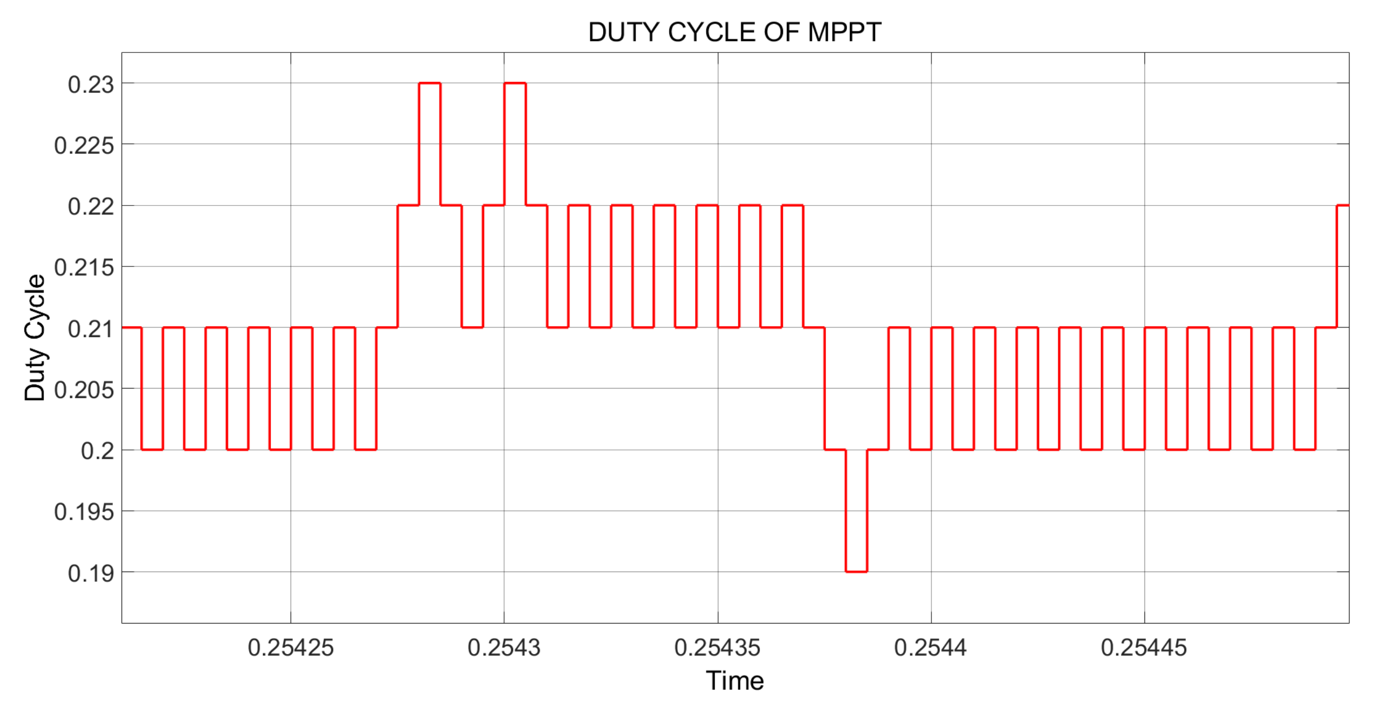


Figure 3.12: Graph of duty cycle of MPPT (simulation)

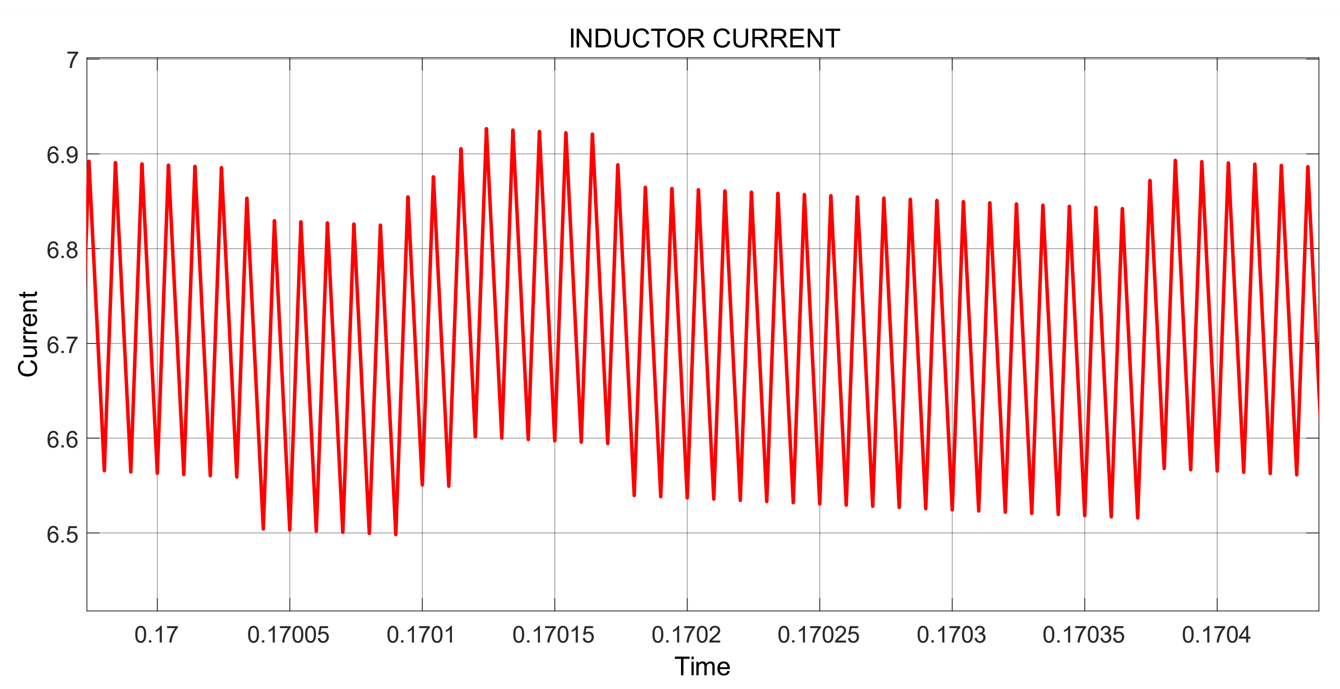


Figure 3.13: Graph of inductor current with fluctuations of 4.5 percent (simulation)

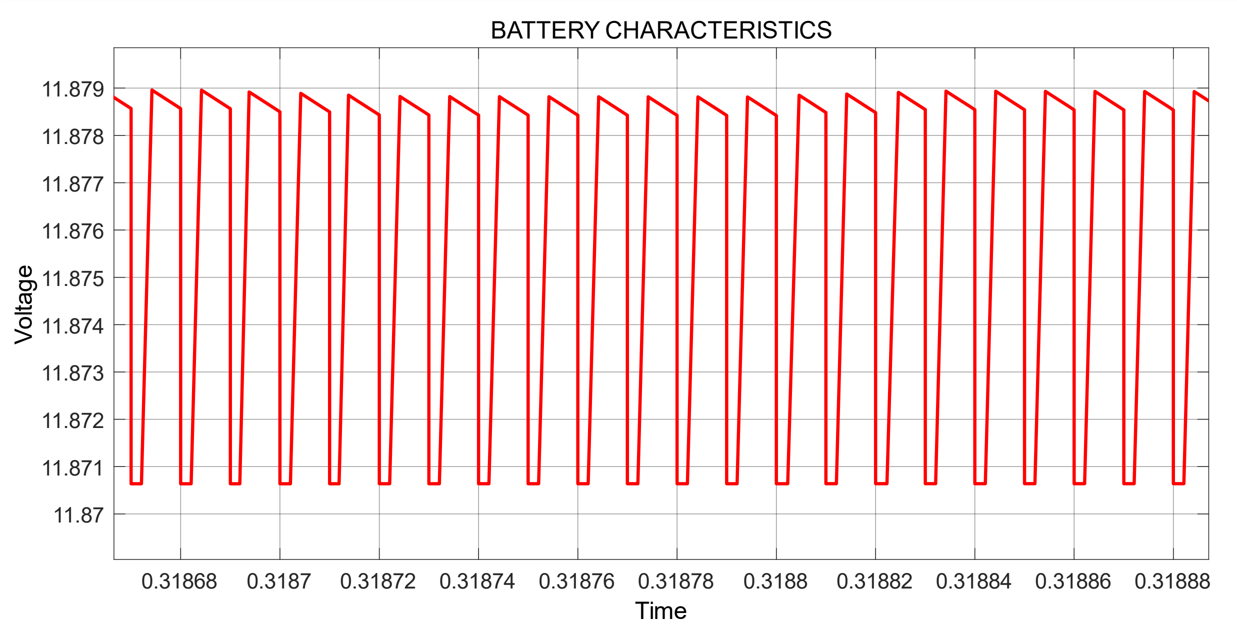


Figure 3.14: The battery voltage with fluctuations 0.085% (simulations)

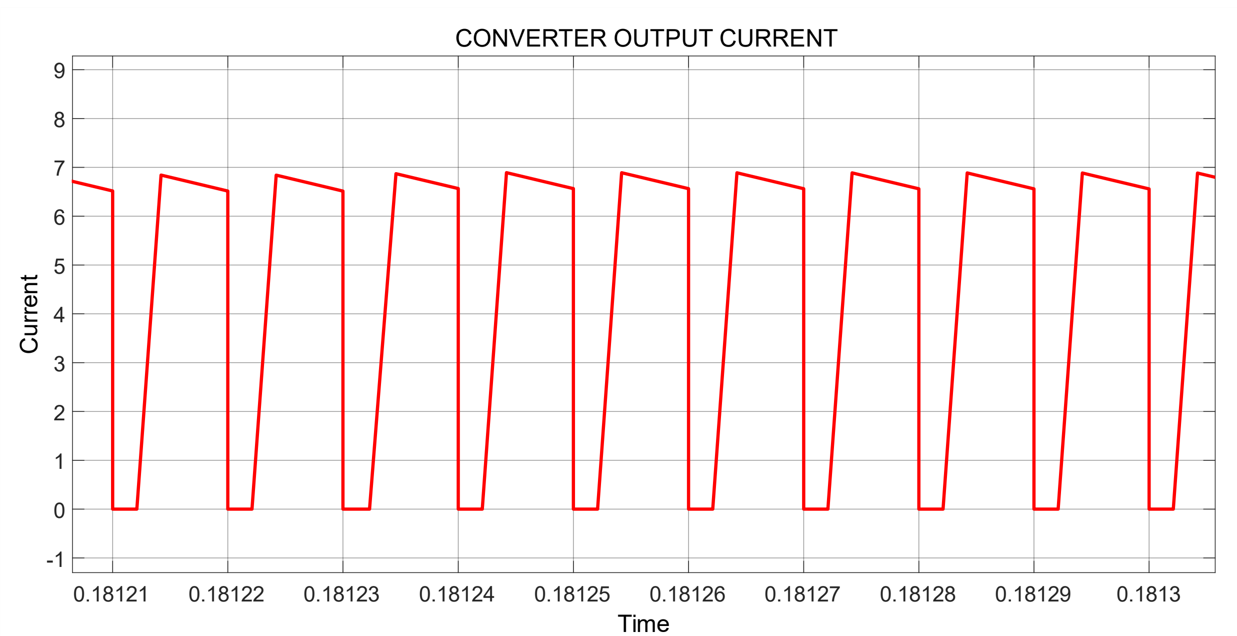


Figure 3.15: Graph of output current (simulation)

3.1.9.2 Hardware Implementation of buck boost Converter:

In the hardware implementation of the buck-boost converter, we used the IR2104 high side MOSFET driver IC and implemented bootstrapping in the circuit to drive the high side MOSFET effectively. Additionally, we utilized an N-channel MOSFET, specifically the IRF540N, as the main switching element, and a Schottky diode, the RHRP3120, for diode-based energy transfer during switching transitions and the switching frequency for driving MOSFET was 100khz but during testing we used 50khz sometimes.

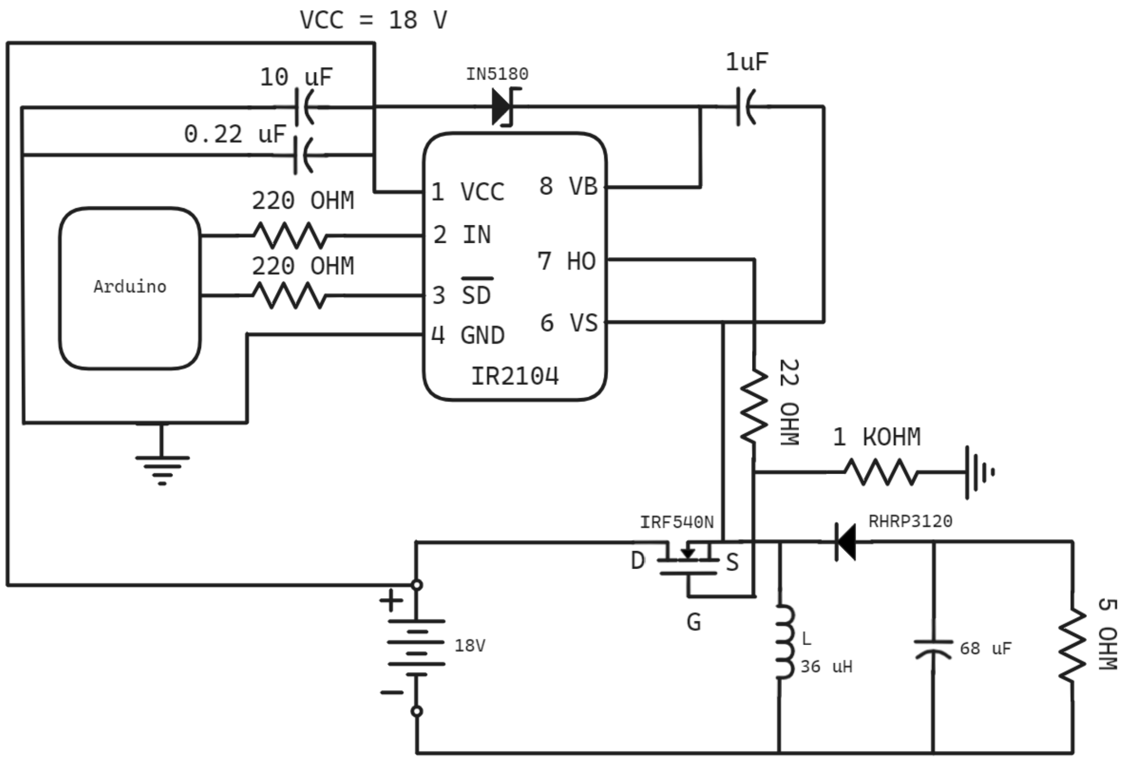


Figure 3.16: Diagram of Buck-Boost converter

During testing, I observed the high side gate waveform results on the oscilloscope. These observations provided crucial insights into the switching behavior and performance of the buck-boost converter, allowing for adjustments to optimize its efficiency and functionality in various voltage regulation applications.

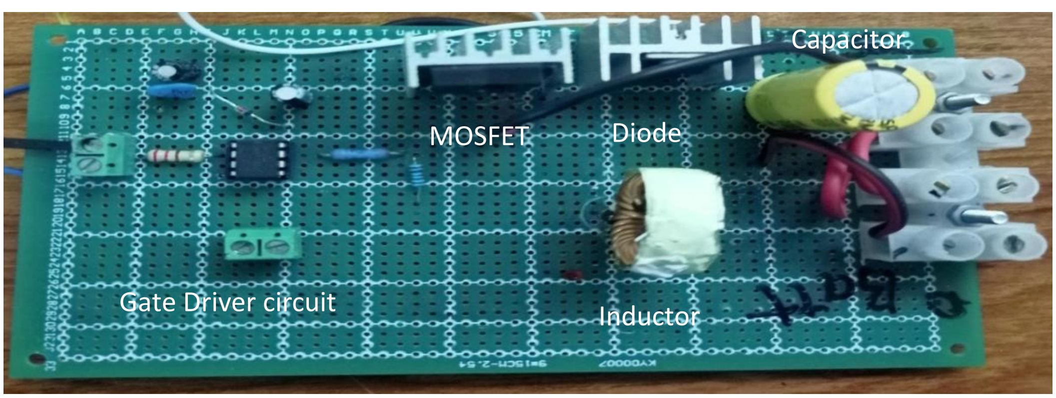


Figure 3.17: Hardware of Buck-Boost converter

3.1.9.3 Result of the hardware implementation of buck boost Converter:

The waveform above displays the high side gate results, while the waveform below shows the low side gate results here the one small box equal to 5 volts so high side gate has approximately 11 volts which is enough to drive high side MOSFET.

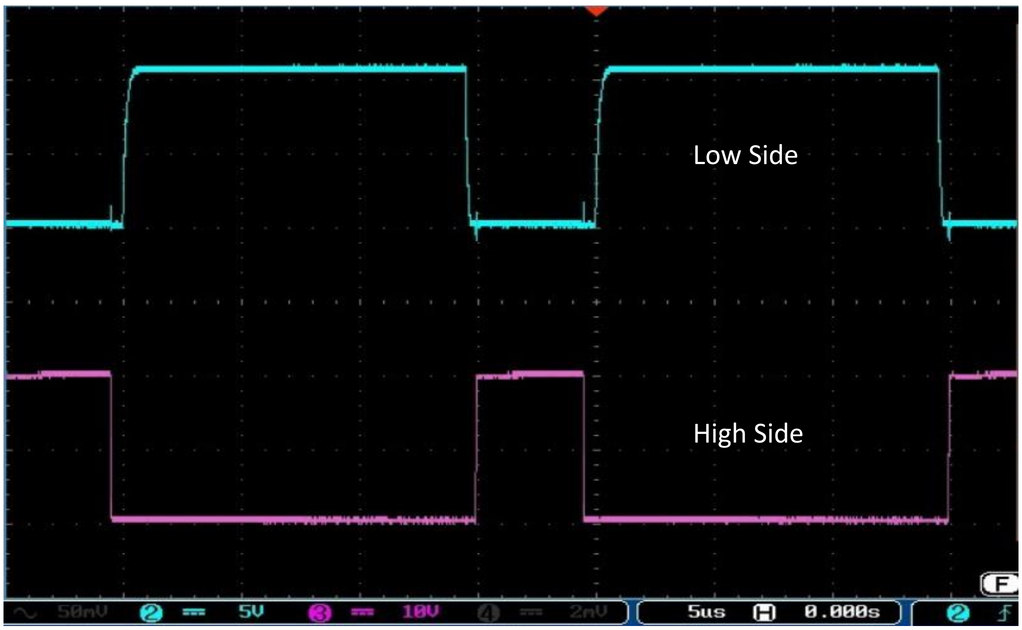


Figure 3.18: Hardware results of Gate driver where above is low side and below is high side voltage

3.1.9.4 Hardware problem with buck boost Converter and result discussion:

After analysing the practical results of the converter, we discovered that the buck-boost converter was not efficiently boosting or bucking the voltage. The issue stemmed from the inductor, specifically related to its charging, and discharging characteristics. To mitigate this problem, we decided to transition to a buck converter topology, which did not exhibit similar issues. We then proceeded with the hardware implementation of the buck converter to achieve the desired voltage regulation without encountering the previous challenges.

3.1.9.5 Hardware implementation of buck Converter:

In the revised hardware implementation of the buck converter, the IRF9540P P-channel MOSFET was chosen for the high side switching element. This selection was made to address the previous issues encountered with the high side MOSFET, ensuring more reliable and efficient operation. Additionally, the RHRP Schottky diode was carefully evaluated to ensure it could handle the required current and voltage levels in the circuit.

After recalculating the circuit parameters and conducting thorough simulations, it was determined that the hardware implementation could proceed without any major modifications. The updated design was then translated into a physical PCB layout, taking into consideration factors such as component placement, trace routing, and signal integrity. Careful attention was paid to ensure that the layout adhered to best practices for high-speed switching circuits, minimizing parasitic effects and maximizing efficiency.

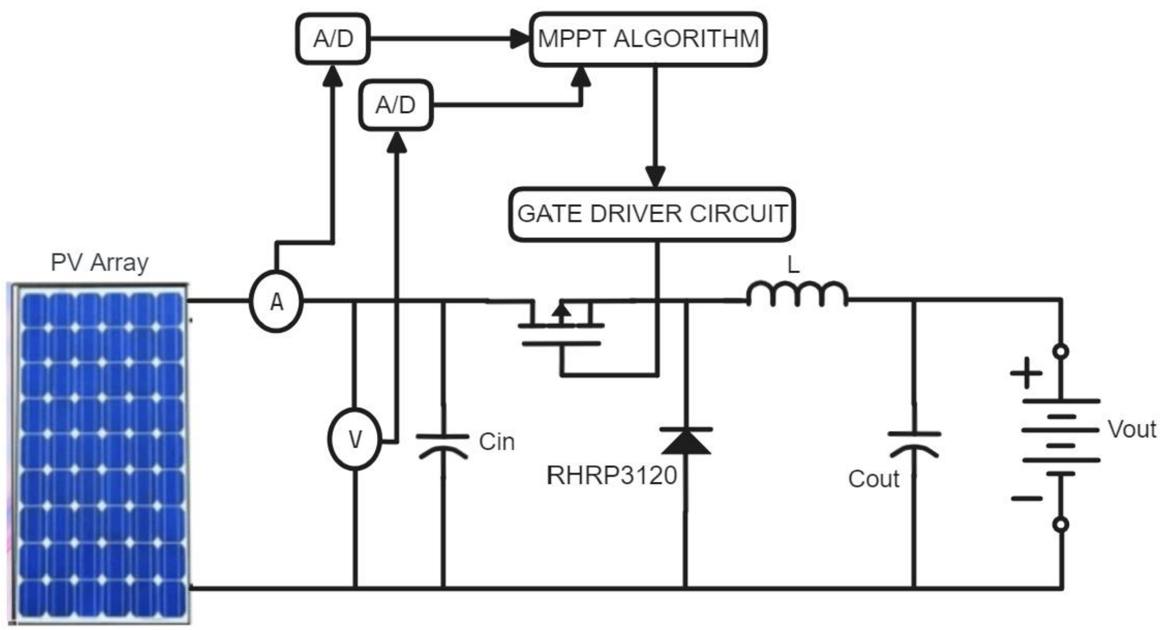


Figure 3.19: Buck converter schematic

Following the completion of the PCB layout, the hardware was assembled and tested to verify its performance. Various tests, including voltage regulation, efficiency measurement, and thermal analysis, were conducted to ensure that the buck converter operated within the specified parameters. The results of these tests confirmed that the hardware implementation was successful, and the buck converter was able to efficiently regulate the voltage as intended.

Figure 3.20: PCB of Buck converter

3.1.9.6 Hardware result of buck converter:

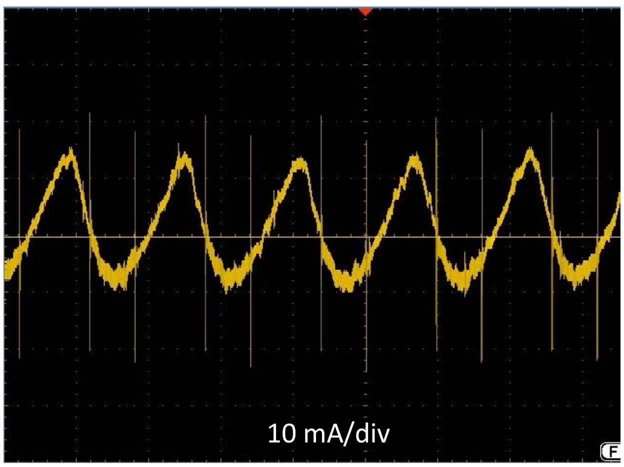


Figure 3.21: Output current graph of converter

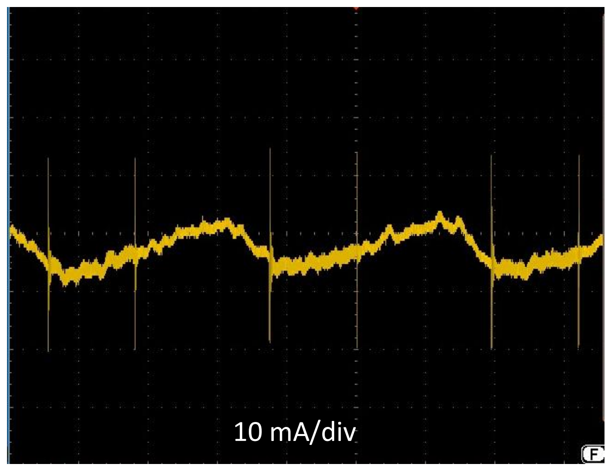


Figure 3.22: Input current graph

3.1.10 Grid connection:

The battery is also connected to the grid via a simple battery charger that was bought. The goal of the charger is to act as the main source of charging but once the charging is done then the solar panel will be used to essentially extend the time before which the battery will need to be recharged. It is also possible to charge the battery by just using the solar panel, this will be done by just placing the robot outside in the sun.

3.2 Software

3.2.1 Stereo Vision:

Utilizing the RealSense depth camera, the project aims to enable the robot to perceive its surroundings in 3. This involves constructing a 3D image and estimating distances of objects using the depth camera. Implementation is carried out on the Jetson Nano platform with the assistance of the LibreSense and pyrealsense2 libraries.

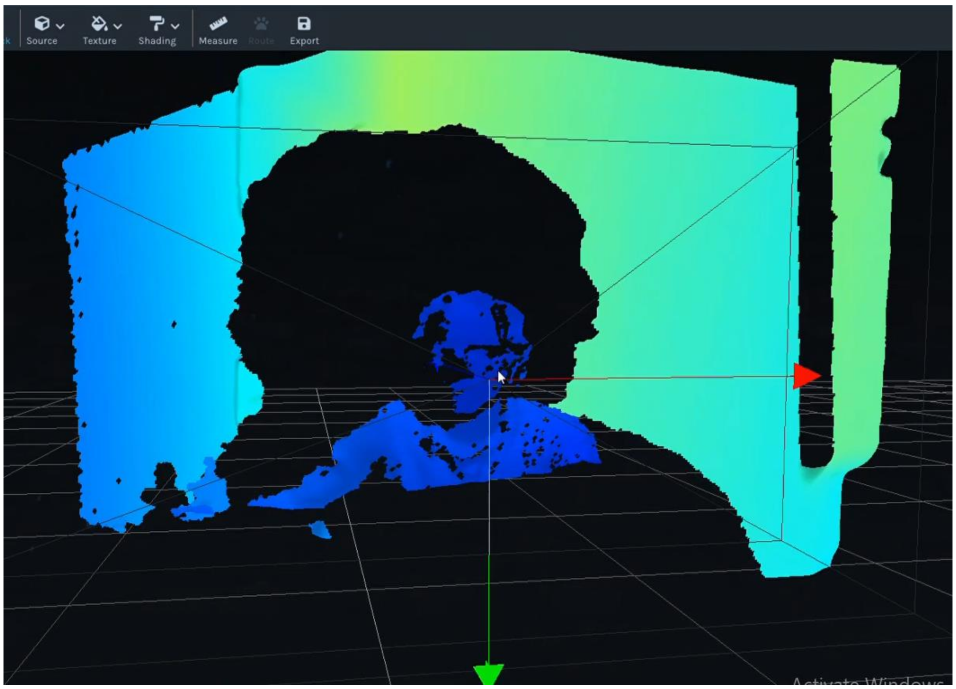


Figure 3.23: Depth camera image 3D construction

3.2.1.1 Sensor Selection and Rationale:

The selection of the RealSense depth camera for stereo vision is based on its ability to capture depth information with high accuracy and resolution. The reason for choosing the depth camera over Lidar has been mentioned above in the hardware section.

3.2.1.2 Image Acquisition and Processing:

Standard process of acquiring stereo image pairs from the RealSense depth camera and preprocessing them for depth estimation was followed. Pyrealsense2 library was used for on image rectification, disparity calculation, and for camera calibration. Pyrealsense2 is a python wrapper for librealsense which is a software package released by intel for working with stereo cameras using edge computing devices.

3.2.1.3 Depth Estimation and 3D Reconstruction:

The depth estimation and 3D reconstruction process involves utilizing the stereo image pairs to calculate the disparity map and subsequently reconstructing the scene in three dimensions. Stereo inaging and triangulation techniques were employed for depth estimation. Z represents the distance of object from camera.

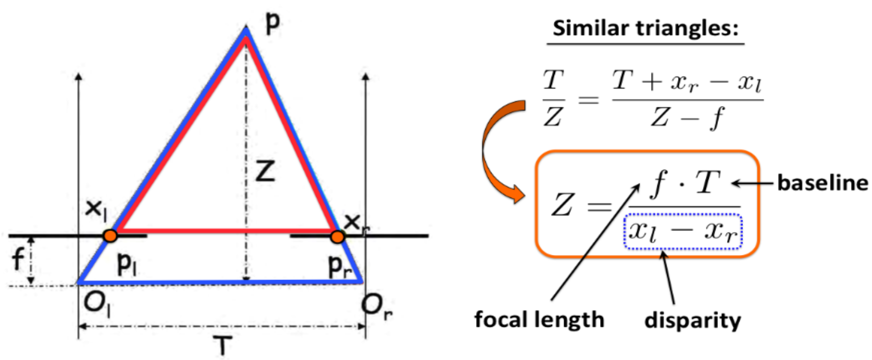


Figure 3.24: Triangulation process.

3.2.1.4 Integration with Perception Module:

The integration of stereo vision with the perception module involves incorporating the depth information obtained from stereo vision into the overall perception framework. Intel released the package librealsense and its python wrapper pyrealsense2 to help with the smooth working and calculation of distance and stereo camera operation. The Intel RealSense D455 camera that is used in this project provides 98% accuracy for meter object distance.

Table 3.4: Depth camera comparison

Metric	D400/D410/D415 (up to 2 Meters and 80% ROI, HD Resolution)	D420/D430/D435/ D435i (up to 2 Meters and 80% ROI, HD Resolution)	D450/D455 (up to 4 Meters and 80% ROI, HD Resolution)
Z-accuracy (or absolute error)	± 2%	± 2%	± 2%
Fill rate	≥ 99%	≥ 99%	≥ 99%
RMS Error (or Spatial Noise)	≤ 2%	≤ 2%	≤ 2%
Temporal Noise	≤ 1%	≤ 1%	≤ 1%

3.2.2 Scene Understanding:

This stage involves the robot comprehending the data received from the camera frames, essentially categorizing, and understanding its environment. It helps the model to develop a pre intelligence before all the information is introduced to the main decision-making algorithm. It consists of two main parts one is object detection and the second is road understanding.

3.2.2.1 Object Detection model selection:

A thorough analysis of the literature was carried out and many models were compared. The object detection model that we need is a model that has relatively high accuracy but is very fast meaning it has very fast FPS so here is a table comparing some of the object detection models available today.

Table 3.5: Comparison of AI models

AI Models	FPS	mAP
Fast R-CNN	70	0.5
Faster R-CNN VGG-16	73.2	7
SSD300	74.3	46
YOLOv2	76.8	67
PPYOLOE-S	208	60.5
EfficientDet-D7x	6.5	72.4
YOLOv7-tiny-SiLU	273	56.7

After comparing all the object detection models we found that YOLOv7 was the best model for what we were trying to do.

3.2.2.2 Object Detection Using YOLOv7:

The adoption of YOLOv7 for object detection is motivated by its real-time performance and high accuracy, making it well-suited for deployment in resource-constrained environments. A pre-trained version of YOLOv7 was used with MS COCO dataset, this was done to take advantage of the already trained model instead of training it ourselves which would have taken a lot of time and resources. Only the classes relevant to the road were used by the decision-making algorithm. If we compare the yolov7 model with previous models of this class, then it outperforms all of them. Yolov8 has also come out at this point in time, but its research paper has not come out therefore we have no way of confirming if it is the fastest model or not therefore YOLOv7 has been used. Yolov7 was not used directly but was instead used with TensorRT instead which is an optimization framework for built for edge computing devices having GPUs. Without TensorRT YOLOv7 achieved an FPS of 2 to 3 which is unusable for real time applications but with TensorRT Yolov7 achieved an FPS of 15 which is substantially better and is enough for our application.

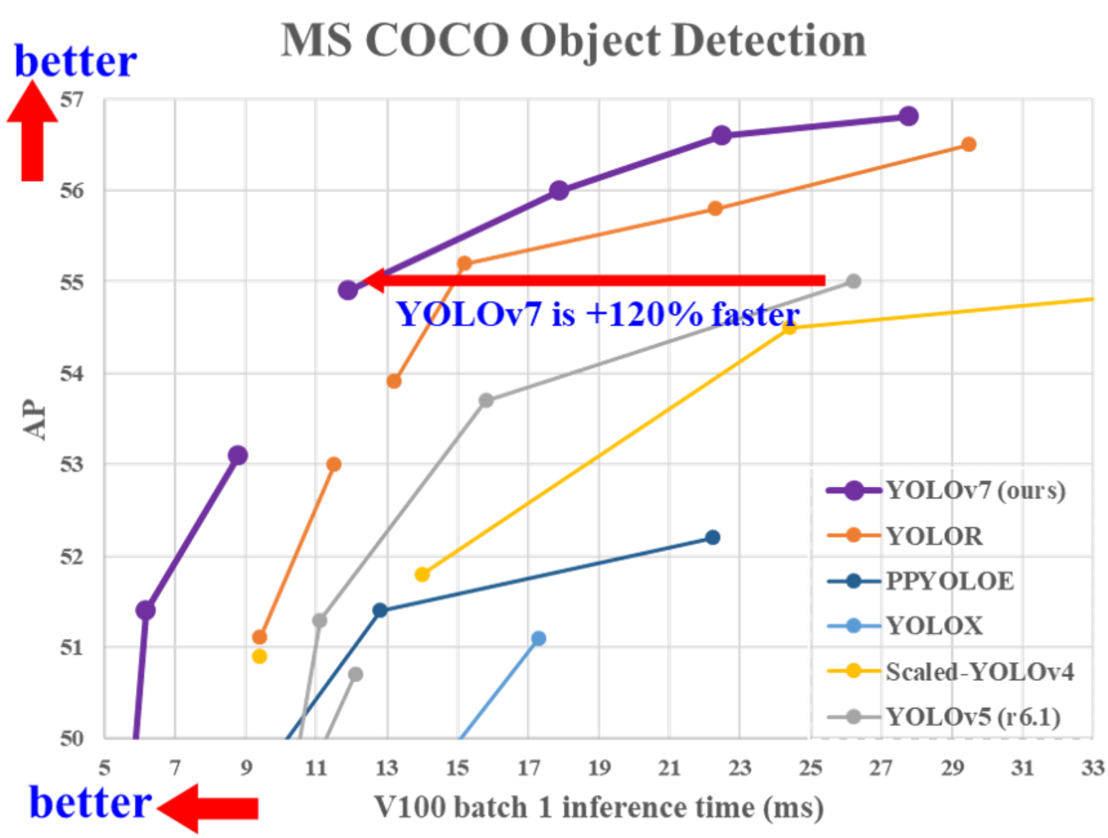


Figure 3.25: Graph comparing YOLOv7 with previous AI models

Figure 3.26: Object detection using YOLOv7

3.2.2.3 Road Understanding:

For road understanding we needed a model that could detect the road in real time meaning it would have high FPS as well as accuracy but we could not find a specific model that could be considered the best in segmentation for our application. The method od segmentation we went with is called instance segmentation instead of semantic segmentation. Instance segmentation is used to detect and then mask the part of the image we want to detect. The rest of the image remains untouched but in semantic segmentation all the pixels in the image must be assigned a class, so it is used to detect everything that an image contains and then separate it. Semantic segmentation is also a method for road understanding, but it is far more complex to train and use and does not have a lot of resources available on the internet for our specific use case on an edge computing device. So instead, we chose semantic segmentation. For our first attempt we chose YOLOv7 segmentation algorithm because we assumed that since YOLOv7 detection is one of the fastest algorithms so YOLOv7 segmentation must also be the fastest. But after training it on 3500 images of roads we found that its accuracy was very low, so we needed more training data and also more time and resources to use it which we did not have. Another problem was that

Jetson nano being an edge computing device does not have enough computing power to run two models at once.

Figure 3.27: Road segmentation using YOLOv7 segmentation

Keeping these constraints in mind we chose a model that could do both detection and road segmentation and was already pre trained with a large dataset. So, we found YOLOP or YOLO Panoptic which can do object detection, road segmentation and lane detection at once. Not only that but it is also trained on BDD100K dataset which consists of 100K videos meaning 120 Million images therefore we assumed it was bound to be accurate but when we tried to implement it on Jetson nano we found that it lacked any extensive documentation and running it directly on a jetson nano proved to be a significant challenge however even after completing that challenge and successfully running YOLOP on Jetson nano it was found that the model lacked accuracy and had very low FPS. Running with TensorRT did not prove any more useful. It could have been due to mistakes made by the author while implementing TensorRT or it could have been mistakes on our part while trying to run it. In a nutshell it was also unusable since it had very low accuracy with both road detection and object detection. Therefore road segmentation was successfully completed but it seemed that we could not use it therefore the robot could not detect the road.

Figure 3.28: Road segmentation using YOLOP

3.2.3 Location Navigation:

Location navigation is essential for guiding the self-driven robot from its starting point to its destination autonomously. It involves tasks such as route planning, localization, and path following, enabling the robot to navigate safely and efficiently in dynamic environments.

Integration of Google Maps API or Direction API enables the robot to determine its route from a starting point to a desired destination.

By providing the robot's current GPS coordinates and extracting relevant information from the API's JSON file, the robot can navigate effectively, akin to systems used in taxi platforms like Uber.

3.2.3.1 Localization Techniques:

Localization is critical for determining the robot's position relative to its surroundings accurately. In this project we used Neo-6m GPS module and used localization techniques. The accuracy of the GPS is 2.5m. The GPS module sends its data to the Jetson nano serially so we can simply read this serial data and extract the necessary latitude and longitude coordinates. One problem that was discovered was that GPS modules cannot work indoors therefore it is not possible to autonomously navigate from one place to another while inside a building by using the GPS method. Therefore, other techniques such as mapping using SLAM are required.

3.2.3.2 Integration of Google Maps API:

The integration of Google Maps API facilitates route planning and navigation by providing access to real-time map data and driving directions. The Google API can be used only if you are registered to google cloud platform. When the API key is acquired, we can only use it to a certain limit without having to buy the subscription. Now the only thing that is left is to constantly communicate with Directions API and navigate through the route. The first step is to enter the starting GPS coordinates of the and the desired destination coordinates. After this we have two choices, one is to keep communicating with the google maps api in real time but that requires internet connection to be present at all times meaning if the internet signals become weak then the robot will fail and secondly it requires a subscription to the google cloud platform therefore to get around that we simply pre-generated all the necessary routes and got their JSON files. Now we can simply store all the files on the Jetson nano and choose which ever file we want based on the route we want to follow.

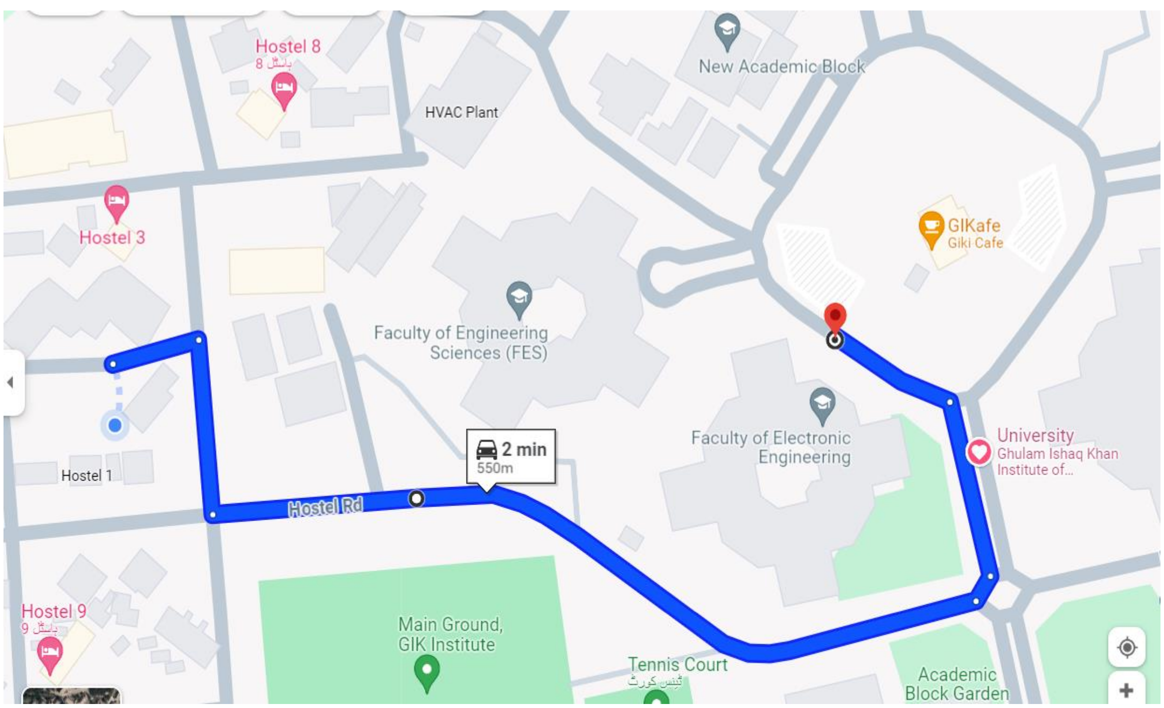
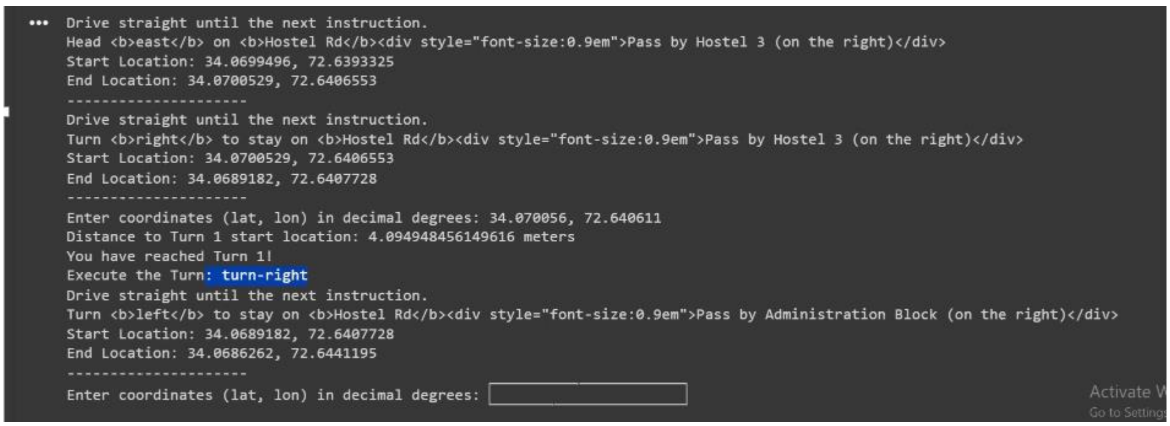


Figure 3.29: Google maps API

3.2.3.3 Path Following and Trajectory Planning:

Path following and trajectory planning enable the robot to navigate along the planned route while adhering to traffic rules. The Google Directions API has officially released the documentation which is needed by developers to extract the relevant information that they need. Therefore, with the help of a JSON file we can navigate through the route that we chose, this is done by sharing the current GPS location that we acquired with the GPS module and get the direction that we should go for our current GPS position. Therefore, when the vehicle is near a roundabout with a distance less than 5 meters the turning direction is generated according to route that the vehicle is following. The distance is configurable.



```
Drive straight until the next instruction.
Head <b>east</b> on <b>Hostel Rd</b><div style="font-size:0.9em">Pass by Hostel 3 (on the right)</div>
Start Location: 34.0699496, 72.6393325
End Location: 34.0700529, 72.6406553
--------------------
Drive straight until the next instruction.
Turn <b>right</b> to stay on <b>Hostel Rd</b><div style="font-size:0.9em">Pass by Hostel 3 (on the right)</div>
Start Location: 34.0700529, 72.6406553
End Location: 34.0689182, 72.6407728
--------------------
Enter coordinates (lat, lon) in decimal degrees: 34.070056, 72.640611
Distance to Turn 1 start location: 4.094948456149616 meters
You have reached Turn 1!
Execute the Turn: turn-right
Drive straight until the next instruction.
Turn <b>left</b> to stay on <b>Hostel Rd</b><div style="font-size:0.9em">Pass by Administration Block (on the right)</div>
Start Location: 34.0689182, 72.6407728
End Location: 34.0686262, 72.6441195
--------------------
Enter coordinates (lat, lon) in decimal degrees:
```

Figure 3.30: Path detection algorithm

3.2.4 Decision Making:

Decision making is the cognitive process through which the self-driving robot analyses its perception data, evaluates its current state, and selects appropriate actions to achieve its objectives. It involves tasks such as obstacle avoidance, path planning, and adaptive behavior generation, enabling the robot to navigate safely and effectively in complex environments.

This is a customized algorithm written from scratch to model the best behavioral decision for this specific robot. In this algorithm FSM approach was used as this was best suited in case of our project as opposed to deep leaning based approach which is a better approach overall but due to its complexity and the resources required to train the model this approach was not used

3.2.4.1 Finite State Machine (FSM) Approach:

The adoption of the Finite State Machine (FSM) approach for decision making provides a structured framework for modeling the robot's behavior and transitioning between different states based on environmental stimuli. This model outlines the design and implementation of the FSM, including state definition, transition rules, and action selection criteria, emphasizing its flexibility and scalability for handling diverse scenarios. There were given sets of some initial and final states. Almost 8 initial states have 5 total outcomes and it can decided by following any transition path.

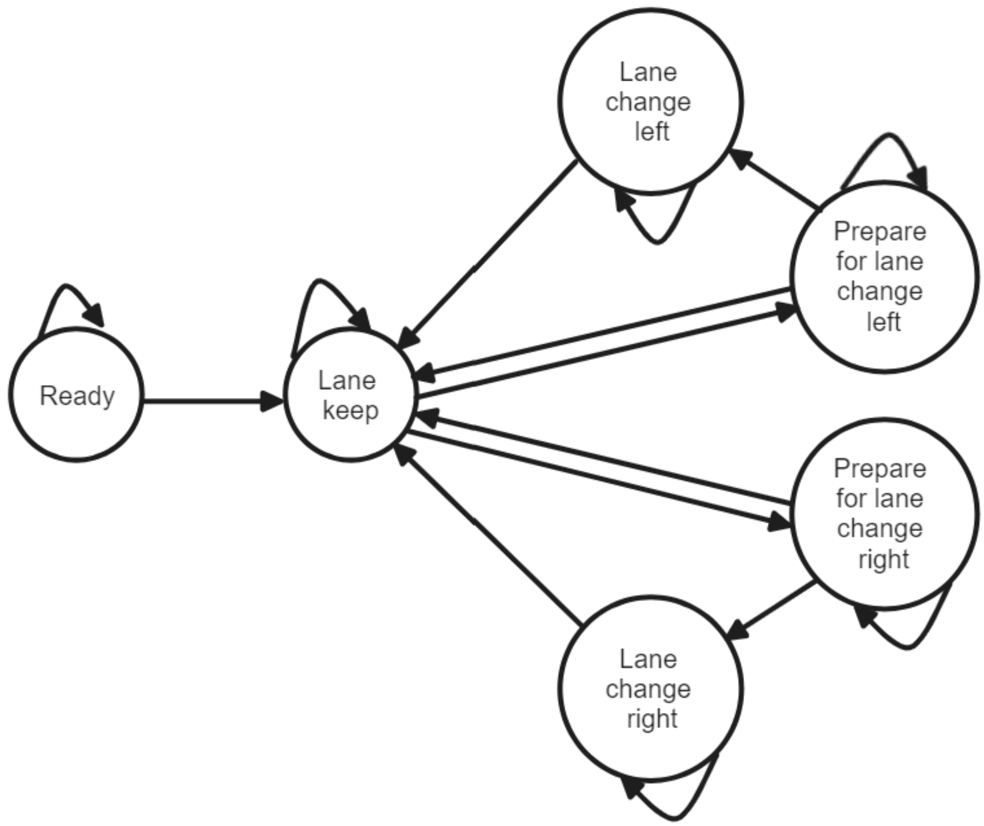


Figure 3.31: FSM decision

Decisions range from slowing down, changing direction, overtaking obstacles, or even stopping in response to unexpected events.

3.2.4.2 Adaptive Behaviour Generation:

Adaptive behaviour generation enables the robot to respond intelligently to dynamic environmental changes, such as unexpected obstacles or traffic congestion. Mainly transition matrix can be defined as adaptive behavior generation for this project. Remember FSM is not a deep learning model it is a user specific technique of writing a certain algorithm to make best possible decision. With the help of adaptive behaviour generation, we can intricately modify the behavioral decision of the robot of our own choice. This helps the robot make the best real-world decision based on all the data provided. We have implemented the code for this but have only been able to test only two decisions i.e. stopping of the robot when a person or a car comes within a certain distance which is configurable.

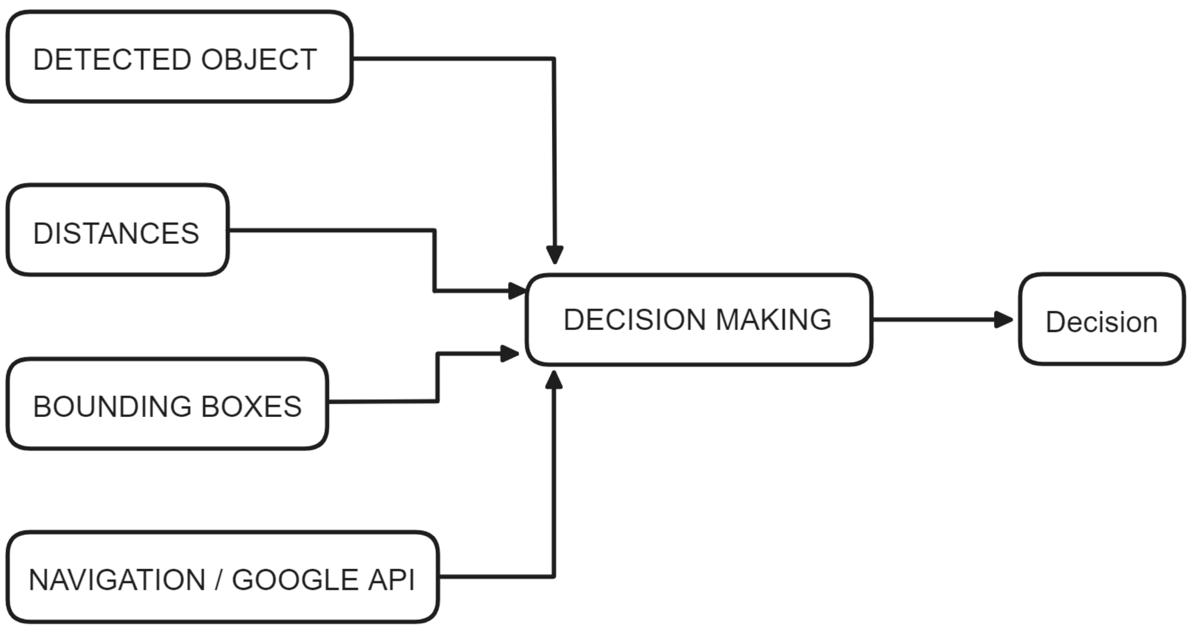


Figure 3.32: Decision flow

3.3 Chapter Summary

In this chapter we have discussed all the hardware as well the software that was required in the project as well as discussed, the reason for choosing each hardware or software. Some setbacks were considered as well but in the proceeding chapter the results, the problems faced and troubleshooting of these problems will be discussed.

4 Results, Discussion, and Troubleshooting

4.1 Problems faced:

There were many problems faced though out the problem and many problems were successfully solved after troubleshooting. These problems will be discussed in the proceeding pages.

4.1.1 YOLOv7 on Jetson nano:

Running YOLOv7 on google collab or on a modern operating systems is a breeze as the python libraries are well optimized to handle current operating systems, but Jetson nano uses an ubuntu based distro called jetpack which essentially uses GNU/Linux as its operating system. The main issue is that Jetpack is based on version 18 of ubuntu while the current version is 24.04 therefore we are forced to use older versions of python libraries which have their own set of problems while working together, not only that but Jetson nano uses an arm based architecture instead of a much more common x86 or amd64 based architecture therefore all libraries that are used must be specifically built for this especially if we are not compiling a library from source but are instead using a pre bult wheel file. Therefore, many library problems were faced while installing YOLOv7 on jetson nano but in the end the problems were fixed, and we were able to get it consistently running in a few hours each time after that.

4.1.2 YOLOP on Jetson nano:

Yolov7 though difficult to run on a jetson nano still has extensive documentation on the internet but YOLOP on the other hand has almost no documentation. Therefore, getting YOLOP to run was an even bigger challenge than running YOLOv7. In the end we were successfully able to run YOLOP on a jetson nano though running it with TensorRT proved even more difficult. Not only that but even after successfully running it with TensorRT we did not get the results that we hoped for therefore we had no choice but to abandon it.

4.1.3 Configuring the Realsense camera on Jetson nano:

Though it was easy to use the stereo camera on Jetson nano if all that was required was a webcam and detection, but our application required that distance of the detected objects also be found so we needed a special software package built by Intel called Librealsense. It used to be that you needed to go through a very extensive process that had you changing configuration settings in the Linux kernel but now people have built scripts that just need to be run and everything can be done for you. The script also builds the pyrealsnse2 library which is a python wrapper for librealsense meaning it allows the use of the depth camera inside python. There is documentation for how to use the pyrealsense2 library on the internet so that did not prove very difficult.

4.1.4 Changing of ports:

In Linux there is a problem that whenever a device is plugged out and plugged back in then it gets assigned a different name. Linux has a philosophy that everything will be stored as a file, so all devices or ports are also stored as files. These can be found in the /dev/ directory. An example is that of an Arduino mega, whenever an Arduino mega is connected to a jetson nano it would get assigned any name from ttyACM0 to ttyACMN where N is any number. This also occurred for every other device as well so what we did was that we identified the product ID of each device and assigned it a specific name so whenever it got detected then a file of that name would automatically be created which fixed the problem.

4.1.5 Problems with Serial Communication:

To communicate between the Jetson nano and Arduino mega we used a library called pyserial the goal of which is to make serial communication between two devices possible using python. The main problem that was faced was that a lot of data go sent together and Arduino could not sperate the data e.g. straightstraightstaright would be sent and Arduino mega could not separate it so a standard industry technique was used which uses special characters in front of and at the end of the message e.g. in our case we used < and >. We had Arduino detect these characters and interpret everything within these triangular brackets as a single message.

4.1.6 Problems with TensorRT:

One of the biggest problems we faced software wise was getting TensorRT to run specifically the generation of a engine file that TensorRT can run. Generating the engine file wasn’t the biggest problem itself the bigger problem was generating an ONNX file which was needed to get an engine file. The biggest problem that we faced was getting the libraries related to ONNX working in the end we could not do this and had to instead rely on prebuilt implementations of TensorRT and prebuilt ONNX files as ONNX files are not hardware specific they could be generated anywhere and used in a Jetson nano abut engine files are hardware specific and need to be generated exactly where they are to be used. One problem that was encountered frequently was the use of Cmake in the projects since most libraries in python are actually made with C++ therefore, they required the use of Cmake to compile them before they could be used effectively.

4.1.7 Jetson nano heating problem:

One the software side when we used Jetpack, we did not face that many heating problems but while trying to run YOLOP we had to switch to a custom and prebuilt Ubuntu operating system for Jetson nano which was based on Ubuntu 20 which made the process of installing libraries far quicker. But the issue with this version of Ubuntu was that we ran out of memory far more often as the Jetson nano only has 2GB of ram, so we faced a lot of heating problems, so we had to install DC 12 V fans on the robot circuitry to keep the Jetson nano cool. In fact, we would recommend using this newer OS as compared to Jetpack OS for those who are not doing complex image processing.

4.1.8 Problems with IBT2:

When working with IBT2 motor drivers using Arduino mega the biggest problem we faced was that when we tried to control the motors directly by giving signals to the pins, we faced problems with consistency, sometimes the drivers worked perfectly and sometimes they did not, so we switched to a library called TomIBT2 which gave us consistent results till the end of the project.

4.1.9 Problems with GPS:

When dealing with GPS initially we used an approach where we used our phone as a GPS but the problem that arose with that was the accuracy was so low that it was essentially unusable so we had to switch to a specific GPS module but that also only has accuracy of 2.5 m, so we have not been able to get 100 percent consistent results on turns. A better approach would be to use dGNSS or RTK based GPS, but both require a base station as well as a GPS on board the robot but as they are expensive and not available in Pakistan, we were not able to use them.

4.1.10 Tyre Problems:

The biggest problems we faced were tyre problems, the tyres that we used were Teflon based instead of steel tyres as we were not able to find steel tyres so these tyres gave us problems all through out the projects lifecycle therefore we recommend spending as much time as possible trying to find good steel tyres and then getting good coupling with the motors instead of just welding the together.

4.2 Final Result and Discussion:

Overall, the project was successfully completed but there were some parts that were not completed due to hardware constraints and a general lack of time and resources. These parts include complete road understanding and a lot more decisions. But as a proof-of-concept model we came very close. Though not a true autonomous vehicle or robot it is still close to an autonomous robot. All the hardware has been successfully implemented only the software has work to be doe. With some work done on road segmentation this project can become an autonomous robot in the truest sense of the word. But this will require extensive research, coding, AI model training and testing. Following is our final product.

Figure 4.1: Final prototype

5 Conclusion and Future Work

5.1 Conclusion

The project "AI Based Self-Driven Robot Using Hybrid Energy Sources" represents a significant milestone in the development of autonomous robotics systems. By integrating advanced perception, navigation, and decision-making modules, the project has demonstrated the feasibility of creating a self-driven robot capable of navigating its environment autonomously while harnessing renewable energy sources for sustainability.

The achievements of the project include:

Successful implementation of stereo vision for environmental perception, enabling the robot to perceive and understand its surroundings in three dimensions.

Integration of object detection and scene understanding algorithms, enhancing the robot's ability to identify and classify objects in its path.

Development of location navigation functionalities using Google Maps API, facilitating route planning and navigation from a starting point to a desired destination.

Implementation of decision-making algorithms based on Finite State Machines, enabling the robot to make intelligent decisions in real-time scenarios.

Implementation of a buck converter with an MPPT algorithm to use solar energy effectively.

The project's outcomes underscore its potential applications in various domains, including transportation, surveillance, and environmental monitoring. Furthermore, the project serves as a testament to interdisciplinary collaboration and innovation in robotics research.

5.2 Future Work

While the project has achieved significant milestones, several avenues for future research and development remain to be explored. Potential areas for future work include:

Enhancement of Perception and Navigation Systems:

Further optimization of perception algorithms and sensor fusion techniques to improve the accuracy and robustness of environmental perception.

Integration of advanced localization and mapping methodologies to enhance the robot's spatial awareness and navigation capabilities in complex environments.

Integration of Advanced Control and Planning Techniques:

Exploration of advanced control and planning algorithms, such as reinforcement learning and model predictive control, to enable more intelligent and adaptive behavior in the self-driven robot.

Investigation of swarm intelligence approaches for collaborative decision-making and task allocation in multi-robot systems.

Sustainability and Energy Efficiency Optimization:

Optimization of energy harvesting mechanisms, such as solar panels and battery systems, to maximize energy efficiency and prolong operational autonomy.

Development of predictive maintenance strategies to ensure the long-term reliability and sustainability of the robot's hardware components.

Real-World Deployment and Validation:

Conducting field trials and pilot studies to validate the performance and usability of the self-driven robot in real-world scenarios.

Collaboration with industry partners, stakeholders, and end-users to identify practical applications and address emerging challenges in autonomous robotics.

By addressing these future research directions, the project aims to further advance the state-of-the-art in autonomous robotics and contribute to the development of innovative solutions for sustainable and intelligent transportation systems.

6 Appendix:

6.1 Code for linear scalability of motors:

```
c=1;
for r = 10:10:1000
    % Constants and Parameters
    airDensity = 1.204;            % Air density (kg/m^3)
    dragCoefficient = 0.82;        % Coefficient of drag
    frontalArea = 0.465;           % Frontal area (m^2)
    rollingResistanceCoefficient = 0.02; % Rolling resistance coefficient
    velocity = 4.17;               % Velocity (m/s)
    angleOfInclination = 15;       % Angle of inclination (degrees)
    sin15 = sind(angleOfInclination); % Sine of the angle of inclination
    weight = r;
    batteryVoltage = 12;           % Battery voltage (V)
    % Calculate power required (in watts) for the EV
    powerRequired = (weight * velocity * rollingResistanceCoefficient) + ...
        (airDensity * dragCoefficient * frontalArea * velocity^3) + ...
        (weight * sin15);
    % Calculate torque required (in Nm) for the PMDC motor
    torque = (powerRequired * 60) / (2 * pi);
    % Calculate current required (in Amperes) for the PMDC motor
    currentRequired = powerRequired / batteryVoltage;
    % Display results
    fprintf('Power Required for the Motor: %.2f Watts\n', powerRequired);
```

```
    fprintf('Torque Required for the Motor: %.2f Nm\n', torque);
    fprintf('Current Rating of PMDC Motor: %.2f Amperes\n', currentRequired);
    fprintf('Weight: %.2f Kg\n',weight);
    x(c)=currentRequired;
    y(c)=weight;
    c=c+1;
end
plot(x,y,'LineWidth',2)
title("PLOT OF WEIGHT VS CURRENT RATING OF MOTOR");
xlabel("CURRENT RATING OF MOTOR")
ylabel("WEIGHT OF PAYLOAD/ROBOT")
```

6.2 Code for ESP32:

```
#include <Ps3Controller.h>
const int voltagePin = 25;
int battery = 0;
void setup()
{  Serial.begin(115200);
   Serial2.begin(9600);
   Ps3.begin("1a:2b:3c:4d:5e:6f");
   Serial.println("Ready.");

   pinMode(14,OUTPUT);
   pinMode(voltagePin, INPUT);
}
void BatteryCharge()
{
     if( battery != Ps3.data.status.battery ){
       battery = Ps3.data.status.battery;
```

```
        if( battery == ps3_status_battery_full )     Ps3.setPlayer(4);
        else if( battery == ps3_status_battery_high )     Ps3.setPlayer(3);
        else if( battery == ps3_status_battery_low)       Ps3.setPlayer(2);
        else if( battery == ps3_status_battery_dying )    Ps3.setPlayer(1);
        else if( battery == ps3_status_battery_shutdown ) Ps3.setPlayer(0); }}
int light=0;
void loop(){
if(Ps3.isConnected()){
        BatteryCharge();
        if( Ps3.data.button.down ){
            Serial2.print("d");}
        if( Ps3.data.button.left ){
            Serial2.print("l");}
        if( Ps3.data.button.up ){
            Serial2.print("u");}
        if( Ps3.data.button.right ){
            Serial2.print("r");}
        if( Ps3.data.button.square){
            Serial2.print("q"); }
        if( Ps3.data.button.circle ){
            Serial2.print("m");}
        if( Ps3.data.button.triangle ){
            Serial2.print("t");}
        if(Ps3.data.button.cross){
            Serial2.print("w");}
        if( Ps3.data.button.l1 ){
            Serial2.print("k");}
        if( Ps3.data.button.r1 ){
            Serial2.print("h")}
        if(Ps3.data.button.l2) {
            Serial2.print("b");}
        if(Ps3.data.button.r2 ){
            Serial2.print("a"); }
```

```
    if(Ps3.data.button.start ){
        Serial2.print("p");}
    if(Ps3.data.button.select ){
        Serial2.print("i");}
    if(Ps3.data.button.triangle && Ps3.data.button.r2)
    {Serial2.print("g");}}}
```

6.2 Code for Arduino:

```
#include <TomIBT2.h>
#include <NewPing.h>

#define man 48
#define RPWM1 12
#define LPWM1 13
#define R_EN1 52
#define L_EN1 53

#define RPWM2 4
#define LPWM2 5
#define R_EN2 34
#define L_EN2 35

#define RPWM3 7
#define LPWM3 6
#define R_EN3 32
#define L_EN3 33

#define RPWM4 11
#define LPWM4 10
#define R_EN4 25
#define L_EN4 24
```

```
#define RPWM5 9
#define LPWM5 8
#define R_EN5 27
#define L_EN5 26

#define FTrig 38
#define FEcho 39
#define RTrig 23
#define REcho 22
#define LTrig 38
#define LEcho 37

#define In1 29
#define In2 31
#define In3 30
#define In4 28

TomIBT2 motor1(R_EN1, L_EN1, RPWM1, LPWM1);
TomIBT2 motor2(R_EN2, L_EN2, RPWM2, LPWM2);
TomIBT2 motor3(R_EN3, L_EN3, RPWM3, LPWM3);
TomIBT2 motor4(R_EN4, L_EN4, RPWM4, LPWM4);
TomIBT2 motor5(R_EN5, L_EN5, RPWM5, LPWM5);

NewPing
Sonic[3]={NewPing(FTrig,FEcho,100),NewPing(RTrig,REcho,100),NewPing(LTrig,LEcho,100
)};
int FrontDistance=40 ;//IN CM
int RightDistance=30;
int LeftDistance=20;
int FrontUltra=0;
int RightUltra=1;
int LeftUltra=2;
```

```
void setup() {
  motor1.begin();
  motor2.begin();
  motor3.begin();
  motor4.begin();
  motor5.begin();
  Serial.begin(9600);
  Serial3.begin(9600);
  pinMode(In1,OUTPUT);
  pinMode(In2,OUTPUT);
  pinMode(In3,OUTPUT);
  pinMode(In4,OUTPUT);
  digitalWrite(In1,HIGH);}

void StopMotors(){
  motor1.stop();
  motor2.stop();
  motor3.stop();
  motor4.stop();}

void Straight(int speed1=255,int speed2=255){
  motor1.rotate(speed1, TomIBT2::CW);
  motor2.rotate(speed2, TomIBT2::CW);
  motor3.rotate(speed2, TomIBT2::CW);
  motor4.rotate(speed1, TomIBT2::CW);}

void Left(int speed1=255,int speed2=0){
  motor1.rotate(speed1, TomIBT2::CW);
  motor2.rotate(speed2,TomIBT2::CW);
  motor3.rotate(speed2,TomIBT2::CW);
  motor4.rotate(speed1, TomIBT2::CW);
  digitalWrite(In2,HIGH);}
```

```
void Right(int speed1=255,int speed2=0){
  motor1.rotate(speed2, TomIBT2::CW);
  motor2.rotate(speed1, TomIBT2::CW);
  motor3.rotate(speed1, TomIBT2::CW);
  motor4.rotate(speed2, TomIBT2::CW);
  digitalWrite(In3,HIGH);}

void Back(int speed = 255){
  motor1.rotate(speed, TomIBT2::CCW);
  motor2.rotate(speed, TomIBT2::CCW);
  motor3.rotate(speed, TomIBT2::CCW);
  motor4.rotate(speed, TomIBT2::CCW);
  digitalWrite(In2,HIGH);
  digitalWrite(In3,HIGH);}

void AntennaUp()
{motor5.rotate(255, TomIBT2::CCW);}

void AntennaDown()
{motor5.rotate(255, TomIBT2::CW);}

void UltrasonicFunc()
{
  if (Sonic[FrontUltra].ping_cm() < FrontDistance)
  {Straight(127,127);
  delay(10000);}
  else if(Sonic[RightUltra].ping_cm() < LeftDistance)
  {Left(255,150);
  delay(3000);}
  else if(Sonic[LeftUltra].ping_cm() < RightDistance)
  {Right(255,150);
  Serial.println("GOing Right");}
}
```

```
String RecieveData(){
 while(1){
   if(Serial.available()>0){
     String recievedData = Serial.readString();
     return recievedData; }}}
void JetsonCommand(String JetsonData="DEFAULT"){
 if(JetsonData=="Straight"){
   Straight();}
 else if(JetsonData=="Back"){
   Back();}
 else if(JetsonData=="right"){
   Right();}
 else if (JetsonData == "left"){
   Left();}
 else if (JetsonData == "Stop"){
   StopMotors();}}

void Ps3Command(char Ps3Flag)
{if (Ps3Flag == 117)
{   Straight(255,175);
    Serial.print('u');}
 else if(Ps3Flag == 100)
 { Back();
    Serial.print('d');}
else if (Ps3Flag == 114)
{ Right(255,100);
   Serial.print('r');}
else if (Ps3Flag == 108)
{ Left(255,100);
   Serial.print('l');}
else if (Ps3Flag == 119)
{  StopMotors();
```

```
   Serial.print("Stop");}}

int mFlag=0;
void loop() {
  int Ps3Flag=Serial3.read();
  Serial.println(Ps3Flag);
  if (Ps3Flag == 109)
  { mFlag=1;}
  if (Ps3Flag == 113)
  { mFlag =0;}
  if(mFlag == 1)
    {Ps3Command(Ps3Flag);}
  else{
   digitalWrite(man,HIGH);
   String JetsonData=RecieveData();
   Serial.println(JetsonData);
   JetsonCommand(JetsonData);
  }
  //UltrasonicFunc();
  //Serial.println("Front: "+String(Sonic[FrontUltra].ping_cm()));
  //Serial.println("Right: "+String(Sonic[RightUltra].ping_cm()));
  //Serial.println("Left: "+String(Sonic[LeftUltra].ping_cm()));
  }
```

References

[1] “ANNEXURES TO NHA CODE,” 1999. https://nha.gov.pk/uploads/topics/16389470316461.pdf

[2] “CLIMATE CHANGE PROFILE OF PAKISTAN.” https://www.adb.org/sites/default/files/publication/357876/climate-change-profile-pakistan.pdf

[3] BMW, “Autonomous driving – 5 steps to the self-driving car,” *Bmw.com*, Oct. 21, 2020. https://www.bmw.com/en/automotive-life/autonomous-driving.html

[4] D. Etherington, “Las Vegas launches the first electric autonomous shuttle on U.S. public roads,” *TechCrunch*, Jan. 11, 2017. https://techcrunch.com/2017/01/11/las-vegas-launches-the-first-electric-autonomous-shuttle-on-u-s-public-roads/

[5] “Self-driving shuttle now runs at NUS, paves the way for driverless cars,” *Online Car Marketplace for Used & New Cars*, Jul. 29, 2019. https://carbuyer.com.sg/autonomous-vehicle-shuttle-av-nusmart-inchcape-comfortdelgro-easymile-singapore-2019/

[6] Tesla, “Autopilot and Full Self-Driving Capability,” *Tesla.com*, 2023. https://www.tesla.com/support/autopilot

[7] “Autonomous vehicles | MIT News | Massachusetts Institute of Technology,” *news.mit.edu*. https://news.mit.edu/topic/autonomous-vehicles?type=2

[8] K. Miller, “Designing Ethical Self-Driving Cars,” *Stanford HAI*, Jan. 23, 2023. https://hai.stanford.edu/news/designing-ethical-self-driving-cars

[9] GIKI FEE FYDP Thesis 2022 “Self Driving Car”

[10] GIKI FEE FYDP Thesis 2021 “A localized IOT Based Autonomous Delivery Robot for fast Moving Goods”

[11] GIKI FEE FYDP Thesis 2021 “Autonomous Driving with Toy car”

[12] GIKI FEE FYDP Thesis 2019 “Autonomous Shopping Cart”

[13] Y. Yin, "Design of Deep Learning Based Autonomous Driving Control Algorithm," 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 2022, pp. 423-426, doi: 10.1109/ICCECE54139.2022.9712792.

[14] C. Zhang, Z. Huang, M. H. Ang and D. Rus, "LiDAR Degradation Quantification for Autonomous Driving in Rain," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021, pp. 3458-3464, doi: 10.1109/IROS51168.2021.9636694.

[15] S. McCrae and A. Zakhor, "3d Object Detection For Autonomous Driving Using Temporal Lidar Data, " 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 2661-2665, doi: 10.1109/ICIP40778.2020.9191134.

[16] B. Barua, C. Gomes, S. Baghe and J. Sisodia, "A Self-Driving Car Implementation using Computer Vision for Detection and Navigation," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 271-274, doi: 10.1109/ICCS45141.2019.9065627.

[17] Warner, S. (2008). Build Your Own Electric Vehicle. Tab Books.

[18] “Motor Selection for Electric vehicle” |motor selection calculation | bldc motor | ev, “www.youtube.com. https:// www.youtube.com/watc?v=cyEj90LM1SQ (accessed Sep. 24, 2023).

[19] GIKI FEE FYDP Thesis 2023 “Conversion of Combustion Engine Car into Electric Car with Safety Features”.

[20] “Amazon.com: Vision Engineering MEH-001 Mantis Elite Head : Electronics,” www.amazon.com. https://www.amazon.com/Vision-Engineering-MEH-001-Mantis-Elite/dp/ B00EZJJFYY (accessed Oct. 19, 2023).

[21] N. Inc, “LiDAR Price for Cars,” Neuvition | solid-state lidar,lidar sensor suppliers, lidar\ technology, lidar sensor, Feb. 25, 2022. https://cdn.neuvition.com/media/blog/lidar- price.html#:~:text=Velodyne (accessed Oct. 19, 2023).

[22] “7004.0C$ 11% OFF|American 3d 16 Line Lidar Puck Vlp-16 Lidar Sensor - Automation Kits - AliExpress,” aliexpress.com, 2018. https://www.aliexpress.com/item/4000206764459.html (accessed Oct. 19, 2023).

[23] B. Barua, C. Gomes, S. Baghe and J. Sisodia, "A Self-Driving Car Implementation using Computer Vision for Detection and Navigation," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 271-274, doi: 10.1109/ICCS45141.2019.9065627.

[24] P. Wang, “Research on Comparison of LiDAR and Camera in Autonomous Driving,” Journal of Physics: Conference Series, vol. 2093, no. 1, p. 012032, Nov. 2021, doi: https://doi.org/10.1088/1742-6596/2093/1/012032.

[25] “DroneBot Workshop,” DroneBot Workshop, 2016. https://dronebotworkshop.com