DESIGN AND DEVELOPMENT OF UNAMANNED GROUND VEHICLE

SUBMITTED By

MUHAMMAD HAMZA JAVED	**CIIT/FA19-BME-013/WAH**
HAMZA SHAFIQUE	**CIIT/ FA19-BME-010/WAH**
KHAYYAM US SALAM	**CIIT/ FA19-BME-018/WAH**
AHMED USMAN	**CIIT/ FA19-BME-011/WAH**
MUHAMMAD JAMSHAID KHAN	**CIIT/ FA19-BME-026/WAH**

Final Year Design Project Report

Bachelor of Science in mechanical engineering

Bachelor of Science in Mechanical Engineering

COMSATS UNIVERSITY ISLAMABAD
Wah Campus – Pakistan
Spring 2023

COMSATS University Islamabad
Wah Campus
Department of Mechanical Engineering

PROJECT ID	FA2019-FYDP-	**NUMBER OF MEMBERS**	**5**

TITLE	**DESIGN AND DEVELOPMENT OF UNMANNED GROUND VEHICLE**
SUPERVISOR NAME	**DR MUHAMMAD SHOAIB NASEEM**

MEMBER NAME	REG. NO.	EMAIL ADDRESS
MUHAMMAD HAMZA JAVED	FA19-BME-013	muhammadhamzajaved.786@gmail.com
HAMZA SHAFIQUE	Fa19-BME-010	hamzagrt55@gmail.com
KHAYYAM US SALAM	FA19-BME-018	khayyamussalam@gmail.com
AHMED USMAN	FA19-BME-011	ahmedusman4288@gmail.com
MUHAMMAD JAMSHAID KHAN	FA19-BME-026	muhammadjamshaidkhan2212@gmail.com

CHECKLIST:

Number of pages attached with this FYDP report	☐
I/We have submitted the complete Project to the Supervisor	**YES / NO**
I/We have enclosed the soft copy of this document along with supporting material in CD.	**YES / NO**
My/Our supervisor has attested the attached document	**YES / NO**
I/We confirm to state that this project is free from any type of plagiarism and misuse of copyrighted material	**YES / NO**

MEMBERS' SIGNATURES

____

____

____

____

Supervisor's Signature

Declaration

“No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university/institute or other institution of learning”.

MEMBERS’ SIGNATURES

COMSATS University Islamabad
Wah Campus
Department of Mechanical Engineering

Plagiarism Certificate

This is to certify that FYDP Title **Design and development of unmanned ground vehicle** of BS (Mechanical Engineering) students at Mechanical Engineering Department COMSATS University Islamabad, Wah Campus, declare that their BS thesis is checked by the Department and the similarity index is **___15__ %** that is an acceptable limit as set by HEC.

Name: MUHAMMAD HAMZA JAVED Registration No. FA19-BME-013

Name: HAMZA SHAFIQUE Registration No. FA19-BME-010

Name: KHAYYAM US SALAM Registration No. FA19-BME-018

Name: AHMED USMAN Registration No. FA19-BME-011

Name: MUHAMMAD JAMSHAID KHAN Registration No. FA19-BME-026

Date: ______________________

Supervisor Signature: ____

Secretary FYDP Committee Signature: ______________________________

ABSTRACT

The design and development of unmanned ground vehicles (UGVs) have become a focal point of research and innovation in the field of robotics. This thesis provides an overview of the comprehensive approach employed in the design and development of a UGV, encompassing key aspects such as mechanical design, sensor integration, control systems, and software development. The primary objective of this research is to create an efficient and versatile UGV capable of performing complex tasks in a variety of environments. The mechanical design phase focuses on developing a robust chassis and drivetrain, considering factors such as weight, structural integrity, and manoeuvrability.

A systematic selection of motors, wheels, and suspension components is carried out to optimize the UGV's performance and ensure its adaptability to different terrains. Sensor integration plays a crucial role in enhancing the UGV's perception and situational awareness. Various sensors, including infrared, cameras, and inertial measurement units (IMUs) and GPS are integrated into the UGV to provide accurate real-time data about its surroundings. The fusion of sensor data enables reliable obstacle detection, terrain mapping, and localization, facilitating safe and efficient navigation. The control system architecture encompasses both hardware and software components.

An embedded computing platform is utilized to handle sensor data processing, decision-making algorithms, and actuator control. Advanced control algorithms, such as path planning and motion control, are developed and implemented to enable autonomous navigation and precise manoeuvring. Software programming plays a critical role in the UGV's overall functionality and performance. The software architecture incorporates modules for perception, decision-making, and control. Machine learning algorithms are employed to train the UGV to recognize and classify objects, improving its ability to make informed decisions in dynamic environments.

The design and development process includes thorough testing and evaluation to validate the UGV's performance and functionality. Real-world scenarios and simulated environments are utilized to assess the UGV's navigation, obstacle avoidance, and task execution capabilities. The results of these tests are analysed, and necessary refinements are made to optimize the UGV's performance. In conclusion, this thesis presents a comprehensive approach to the design and development of a UGV, encompassing mechanical design, sensor integration, control systems, and software development.

Sustainable Development Goals (SDGs) of FYDP

The Sustainable Development Goals (SDGs) are a set of 17 global goals established by the United Nations to address various social, economic, and environmental challenges facing the world. When designing and developing unmanned ground vehicles (UGVs) or any technology, it's important to consider how these advancements can contribute to sustainable development. Here are some relevant SDGs and their connections to the design and development of UGVs:

SDG 9: INDUSTRY, INNOVATION, AND INFRASTRUCTURE

UGVs can play a crucial role in advancing innovation in the transportation and logistics sectors, improving infrastructure monitoring, and enhancing industrial processes.

SDG 13: CLIMATE ACTION

UGVs can be designed to reduce carbon emissions by facilitating the transition to electric and autonomous vehicles, thereby contributing to climate change mitigation.

SDG 15: LIFE ON LAND

UGVs can be employed for land management, such as monitoring and protecting ecosystems, conducting research, or aiding in wildlife conservation efforts.

SDG 16: PEACE, JUSTICE, AND STRONG INSTITUTIONS

UGVs can assist in maintaining security and peace by supporting law enforcement and defense agencies in surveillance, reconnaissance, and other tasks.

SDG 17: PARTNERSHIPS FOR THE GOALS

Collaborative efforts among governments, industry, and research institutions are essential for the responsible development and deployment of UGVs, ensuring they align with sustainable development objectives.

EDUCATION AND CAPACITY BUILDING (SDG 4)

Promoting education and capacity building in robotics and automation can help countries and communities harness the benefits of UGV technology while addressing potential challenges.

When designing and developing UGVs, it's crucial to conduct a thorough impact assessment to ensure that they contribute positively to sustainable development goals while mitigating potential negative consequences. Collaboration, transparency, and responsible innovation are key principles to follow in this process.

RESEARCH MOTIVATION

When a process' performance is assessed in industrial and manufacturing automation, safety, productivity, and accuracy are crucial considerations. The uses for UGVs have grown dramatically, and they are now a common form of transportation as well as a key component in the advancement of industrial automation. The study to address trajectory tracking and navigation issues for non-holonomic robotic systems is provided in the state-of-the-art literature, which serves as the source of inspiration. These issues are particularly difficult because they provide both theoretical and practical hurdles.

There are several difficulties with how non-holonomic constraints behave theoretically. For example, such systems are underactuated, meaning that there are fewer control inputs than there are states or variables in the system that need to be regulated. Motion planning hence indicates that the systems may be fully controlled with a smaller number of actuators, enhancing the system's total cost-effectiveness. Additionally, under actuation might offer backup control strategies for a system that is fully actuated.

Unmanned ground vehicle trajectory tracking and navigation issues have recently grown in popularity and research vigour. Such issues are also among the major obstacles to the development of sophisticated UGV. Utilizing intelligent controllers and optimization techniques to address trajectory tracking and navigation issues is the main goal of this study. The development of completely autonomous navigation involves a thorough study of trajectory tracking and navigation based on many circumstances.

To enhance the industrial settings, these two distinct issues must be addressed. The tracking of the trajectory is the first issue. A UGV must adhere to certain trajectories in this issue, including continuous gradient trajectories and non-continuous gradient trajectories. The navigation is the second issue. The goal is to create a UGV that can move safely and effectively between an initial location and a target position without running into any blocking impediments, including both dynamic and static barriers.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to all my group members who have contributed to the successful completion of my final year project, titled **"Design and Development of Unmanned Ground Vehicle."** This project has been a significant milestone in our academic journey, and it would not have been possible without the support, guidance, and encouragement of several individuals and organizations.

First and foremost, I extend my sincere appreciation to my project supervisor, **Dr Muhammad Shoaib Naseem**, for their unwavering support, invaluable insights, and continuous guidance throughout the project. Their expertise in robotics and autonomous systems has been instrumental in shaping this endeavor and has broadened my understanding of the field.

I am also deeply thankful to the faculty members of the **Mechanical Department** for their commitment to excellence in education and for providing me with the resources and facilities necessary for the successful execution of this project.

I would like to acknowledge the technical staff and lab assistants who assisted me with the equipment, materials, and technical expertise required for the development and testing of the unmanned ground vehicle.

I owe a debt of gratitude to my fellow classmates and friends who were a constant source of motivation and encouragement during the challenging phases of this project. Their willingness to brainstorm ideas, troubleshoot issues, and provide moral support was invaluable.

I am grateful to my family for their unwavering support, understanding, and patience during my academic journey. Their belief in my abilities has been a driving force behind my dedication to this project.

Last but not least, I would like to thank the anonymous reviewers and experts in the field who provided constructive feedback on the project design and implementation, contributing to its overall improvement.

This project has not only enriched our knowledge but also provided a practical experience in the field of unmanned ground vehicles. I hope that the work presented in this project contributes in some way to the advancement of autonomous systems and robotics.

Thank you all for being a part of this journey and for helping me realize our vision. Your contributions are deeply appreciated.

Table Of Contents

CHAPTER 1 1

1.1 INTRODUCTION 1

1.2 AIMS AND OBJECTIVES 4

CHAPTER 2 5

2.1 LITERATURE REVIEW 5

2.1.1 HISTORICAL DEVELOPMENT OF UGVS 5

2.1.2 KEY MILESTONES IN UGV TECHNOLOGY 5

2.1.3 Stanford Cart (1960s) 5

2.1.4 Shakey the Robot (1970s) 5

2.1.5 UGVs in Space (1990s) 6

2.1.6 DARPA Grand Challenge (2004) 6

2.2 EXAMPLES OF HISTORICAL UGV PROJECTS AND THEIR CONTRIBUTIONSTALON: 7

2.2.1 The TALON 7

2.2.2 Curiosity Rover 7

2.2.3 UGVs in Agriculture 7

2.2.4 UGVs in Mining 7

2.3 UGV APPLICATIONS THROUGH TIME 7

2.4 MILITARY APPLICATIONS 8

2.4.1 HISTORICAL CASE STUDY: REMOTEC ANDROS F6A (1980S) 8

IMPACT ON MILITARY OPERATIONS 8

2.4.2 AGRICULTURAL APPLICATIONS 8

2.4.3 HEALTHCARE APPLICATIONS 9

2.4.4 ENVIRONMENTAL MONITORING APPLICATIONS 9

2.5 EVOLUTION OF UGV TECHNOLOGIES 10

2.5.1 NAVIGATION SYSTEMS 10

2.5.2 SENSORS 11

2.6 EVOLUTION AND IMPROVEMENT OF UGV CAPABILITIES 12

2.6.1 Autonomy 12

2.6.2 Efficiency 12

2.6.3 Versatility 12

2.6.4 Remote Operation 12

2.7 HISTORY 13

2.8 REMOTE CONTROL CONNECTIONS 15

2.8.1 THE CURRENT USED DESIGN ... 15

2.8.2 THE SUGGESTED DESIGN ... 16

2.9 OBSTACLE AVOIDANCE ... 17

2.9.1 INFRARED IR SENSOR ... 17

2.9.2 ULTRASONIC SENOR ... 17

2.9.3 LASER RANGE FINDER SENSOR ... 18

2.9.4 VISION SENSOR ... 18

2.10 CHALLENGES AND LIMITATIONS OVER TIME ... 26

2.11 HISTORICAL CHALLENGES ... 27

2.11.1 Limited Autonomy ... 27

2.11.2 Sensory Limitations ... 27

2.11.3 Communication Range ... 27

2.12 CONTEMPORARY CHALLENGES ... 27

2.12.1 Autonomy and Decision-Making ... 27

2.12.2 Navigation in Complex Environments ... 27

2.12.3 Safety Concerns ... 27

2.12.4 Human-Robot Interaction ... 28

2.12.5 Data Security ... 28

2.13 FUTURE TRENDS AND PROSPECTS OF UGVS ... 28

2.14 EMERGING TRENDS IN UGVS ... 28

2.14.1 Increased Autonomy ... 28

2.14.2 Sensor Fusion ... 28

2.14.3 Human-Machine Collaboration ... 28

2.14.4 Edge Computing ... 29

2.14.5 Energy Efficiency ... 29

2.15 POTENTIAL ADVANCEMENTS IN UGV TECHNOLOGY ... 29

2.15.1 Swarm Robotics ... 29

2.15.2 Biomechanics-Inspired Design ... 29

2.15.3 Robotic Vision ... 29

2.15.4 Agriculture ... 30

2.15.5 Logistics and Delivery ... 30

2.15.6 Healthcare ... 30

2.15.7 Mining and Construction ... 30

2.15.8 Defense and Security ... 30

2.15.9 Environmental Monitoring ... 30

2.15.10 Transportation ... 30

2.16 AREAS FOR FURTHER RESEARCH ... 31

2.16.1 Advanced AI and Machine Learning 31
2.16.2 Simulated Environments 31
2.16.3 Ethical and Legal Frameworks 31
2.16.4 Interoperability and Standard 31
2.16.5 Human-Machine Trust 31
2.16.6 Safety and Redundancy 31

CHAPTER 3 32

3.1.1 METHODOLOGY 32
3.2 COMAPRISON OUR UGV WITH OTHER UGVs 32
3.2.1 PERCEPTION AND SENSING 32
3.2.2 AUTONOMY 33
3.2.3 OBJECT DETECTION AND RECOGNITION 33
3.2.4 OBSTACLE AVOIDANCE 33
3.2.5 MISSION FLEXIBILITY 33
3.2.6 COMPLEXITY AND COST 33
3.2.7 DATA PROCESSING AND BANDWIDTH 34
3.2.8 RELIABILITY AND ROBUSTNESS 34
3.2.9 SCALABILITY 34
3.3 FLOWSHEET DIAGRAM OF OUR UGV METHODOLOGY 35
3.4 METHODOLOGY EXPLANATION 36
3.5 2D DRAWINGS OF UNMANNED GROUND VEHICLE 38
CHAPTER 4 44
4.1 DEVELOPMENT 44
4.1.1 HARDWARE DESCRIPTION 44
4.1.2 SERVO MOTORS 44
4.1.3 RASPBERRY PI 3 45
4.1.4 4-CHANNEL 5V RELAY MODULE (BRIDGE MOTOR DRIVER) 45
4.1.5 REES52 12V TO 5V 5A CONVERTER, BUCK REGULATOR DC 9V-36V STEP DOWN TO DC 5V 46
4.1.6 UBLOX NEO-6M GPS MODULE 46
4.1.7 GY-271 3-AXIS MAGNETIC ELECTRONIC COMPASS MODULE 46
4.1.8 MPU 6500 SIX-AXIS (GYRO + ACCELEROMETER) 47
4.1.9 SHARP GP2Y0A41SK 4-30CM INFRARED PROXIMITY SENSOR SHORT RANGE 47
4.1.10 ARDUINO MEGA ATMEGA2560 47
4.1.11 SG90 SERVO 48

4.1.12 CONVERTER WITH 7 SEGMENT DISPLAY LM2596......................................48
4.1.13 XBEE S2C MODULE AND SHIELD 48
4.2 EXPLANATION OF OBSTACLE AVOIDANCE CODE 49
4.2.1 SUMMARY OF THE CODE..50
4.3 EXPLANATION OF IMAGE PROCESSING CODE 51
4.3.1 SUMMARY OF THE CODE..51

CHAPTER 5 52

5.1 *CONCLUSION*……………………………………………………………….52

6 REFERENCES 54

7 APPENDIXES.. 56

7.1 SOFTWARE DESCRIPTION...59
7.1.1 UGV OBSTACLE AVOIDANCE CODE..59
7.1.2 IMAGE PROCESSING CODE 74

LIST OF FIGURES

Figure 1 REMOTE CONTROLLED VEHICLE………………………………………….……….. 12

Figure 2 GENERAL CONNECTION FOR CONTROLLING A 4 MOTORS.....................................13

Figure 3 GENERAL CONNECTION FOR CONTROLLING A 4 MOTORS.....................................13

Figure 4 DISTRIBUTION OF DUTIES FOR EACH CHANNEL …………………………..……..14

Figure 5 DISTRIBUTION OF DUTIES FOR EACH CHANNEL …………………………..……..14

Figure 6 INFRARED SENSOR……………………………….………..………………………15

Figure 7 ULTRSONIC SENSOR……………………………………………………………………15

Figure 8 LASER RANGE F SENSOR……………………………………..………………...…16

Figure 9 VISION SENSOR……………………………………….……………………………...16

Figure 10 CHAIN DRAWING……………………………………………………………………….36

Figure 11 SUPPORTING WHEEL DRAWING…………………………………………..……36

Figure 12 SPROCKET DRAWING…………………………………………………………………37

Figure 13 SPRING DRAWING……………………………………………………………………..37

Figure 14 IDLER DRAWING…………………………………………………………………….38

Figure 15 SPROCKET DRAWING…………………………………………………………………38

Figure 16 CENTRAL BODY PART 1 DRAWING…………………………………………………39

Figure 17 CENTRAL BODY PART 2 DRAWING…………………………………………………39

Figure 18 TANK DRAWING 1……………………………………………………………………..40

Figure 19 TANK DRAWING 2…………………………………………………………………..40

Figure 20 UGV ………………………………………………………………………………..41

Figure 21 SERVO MOTOR…………………………………………………………………..43

Figure 22 UBLOX NEO-6M ……………………………………..………………………44

Figure 23 GY-271 3-AXIS ………………………………...……………………………..…45

Figure 24 ARDUINO MEGA …………………………...…………………………………..…45

Figure 25 XBEE S2C …………………………………………………………………………..46

Figure 26 RASPBERRY PI 3...……………………………………………………………..46

LIST OF TABLES

Table 1: 4-CHANNEL 5V RELAY MODULE (BRIDGE MOTOR DRIVER)…..………..… 54

Table 2: REES52 12V TO 5V 5A CONVERTER, BUCK REGULATOR DC 9V-36V STEP DOWN TO DC 5……….……………………………………………………………..…. 54

Table 3: UBLOX NEO-6M GPS MODULE..………..…………………….………....………… 54

Table 4: GY-271 3-AXIS MAGNETIC ELECTRONIC COMPASS MODULE…..………… 55

Table 5: MPU 6500 SIX-AXIS (GYRO + ACCELEROMETER)……..……………………..55

Table 6: SHARP GP2Y0A41SK 4-30CM INFRARED PROXIMITY SENSOR SHORT RANGE………………………..……………………………………………………......…55

Table 7: ARDUINO MEGA ATMEGA2560……….…………………………………..…56

Table 8: SG90 SERVO …………………………..………….…………………………..56

Table 9: CONVERTER WITH 7 SEGMENT DISPLAY LM2596…..………………………56

Table 10: XBEE S2C MODULE AND SHIELD……………..……………………………..57

CHAPTER 1

1.1 INTRODUCTION

Unmanned ground vehicles (UGVs) are programmable, multifunctional robots that can acquire information from their environment utilizing sensors and control systems to plan and carry out their task autonomously in their environment. Unmanned vehicles may be divided into three categories based on their operating environment: airborne, terrestrial, and aquatic vehicles (A.A. Nayak, 2012.). The operational environment of an unmanned ground vehicle (UGV) is just the air. Unmanned ground vehicles (UGVs) are those that operate on the water's surface, and unmanned underwater vehicles (UGVs) are those that operate below the water's surface (Anderson, 2019). Terrestrial vehicles make up the final category and can include any type of unmanned ground vehicle with wheels.

These days, there is a clear increase in the use of such vehicles in a variety of everyday life and industrial automation applications. Based on the operational environment, the uses of such vehicles may be summed up. For example, in a hostile or military setting, it would be crucial in some situations to have a UGV for carrying out specified predetermined duties when it is hazardous or difficult for humans to carry out such operations (B. Hendrik, 2013.). Additionally, for mining reasons in some regions that may have dangerous radioactivity or toxic gases. Therefore, the primary method for such workspaces will be the application of a UGV.

The UGV may be used in a variety of various settings, such as those that aim to boost output, cut expenses, and enhance quality of life. They might also be used in transportation networks and industrial manufacturing lines (B. Hendrik, 2013.) (B.J. Yoon). They can be used in a transportation system and for specialized tasks during production operations, such loading and unloading machinery and goods. They have recently been used for surveillance in high-tech security applications.

Mechanical, electrical, and computer engineering are just a few of the technical disciplines that must be understood in order to design a UGV. Sensors, intelligent controllers, and actuation systems are the essential parts of a UGV. These elements are crucial to gaining autonomy and enable a UGV to function and complete its responsibilities without human intervention [(Bonne, 2018),(Borenstein, 1991)]. A UGV therefore functions independently based on perception, intellect, and action. Such vehicles' effective navigation is made possible by constant exchanges between perception, intellect, and action. A UGV is able to comprehend and interact with its environment in a condensed manner to seamlessly attain its specialized goals.

To prevent collisions and function safely, a UGV has to be able to perceive and comprehend things, which may be accomplished by using sensing devices. The main factor in determining decisions is artificial intelligence. This will enable a UGV to function intelligently in its workspace and complete the necessary duties (C. R. Terwelp, 2003). To create an intelligent approach that allows a UGV to learn and infer inside the workspace surroundings, many controllers and algorithms may be applied.

Unmanned ground vehicle is very important for the civilian and military uses. Generally, an unmanned ground vehicle is the vehicle that is used on ground without the presence of the man in it (D. S. Samosir) (Dudek, 2010). These Vehicles can be used at many places where it is very harmful or impossible for the mankind to sit in it. UGVs are essential in current military operations. They can be utilized for reconnaissance, surveillance, EOD, and logistics support.

The capacity of UGVs to work in hazardous or dangerous areas without compromising human life is one of its key advantages. UGVs can be used in situations where humans would be too risky, such as looking for disaster regions, handling hazardous chemicals, or executing military operations in conflict zones. UGVs can deliver significant cost savings in a variety of situations. Organizations can minimize labour expenses, insurance costs, and workers' compensation claims by substituting human workers with UGVs in duties like as monitoring, inspection, and transportation [10]. Furthermore, UGVs can work without breaks or overtime compensation, resulting in greater operational efficiency.

UGVs can conduct repetitive tasks with great precision and accuracy, boosting efficiency and productivity. They can walk in predetermined pathways, continually accomplish jobs, and operate faster than people. This makes them very valuable in transportation, warehousing, agriculture, and manufacturing (Siciliano, 2008). UGVs can be controlled remotely or operated autonomously, enabling for remote operations in a variety of fields. This function is useful in situations where physical presence is difficult or impossible, such as remote surveillance, monitoring, or looking for unsafe places. Experts can also drive the UGV remotely from a safe place, offering real-time information and decision help.

UGVs have the potential to lessen the dangers associated with human mistake and weariness. Its powerful sensors, cameras, and algorithms are capable of detecting and responding to environmental changes or possible risks better than people. UGVs contribute to better risk management by minimizing risk, assuring safer operations, and lowering the possibility of accidents or costly mistakes. UGVs can be customized and outfitted with a variety of tools and payloads to suit a variety of activities and conditions. They are easily reprogrammable or reconfigurable to match changing requirements, making them adaptable to a wide range of applications. UGVs can work in a wide range of terrains, climates, and weather situations, giving them flexibility and adaptability in industries like agricultural, mining, construction, and search and rescue.

UGVs can acquire and transmit huge amounts of data using their onboard sensors and cameras. These statistics can be analysed, tracked, and used to make decisions. In agriculture, for example, UGVs may collect data on soil conditions, crop health, and yield forecasts, allowing farmers to optimize operations and resource allocation. Unmanned Ground Vehicles (UGVs) provide several major benefits across a wide range of industries and applications. For starters, unmanned aerial vehicles (UGVs) improve safety by reducing the need for humans to operate them in risky or dangerous settings (King, 2014). They can be used in situations where human life is at risk, such as areas of disaster, dangerous material handling, or battle zones. UGVs are capable of handling these settings precisely, lowering the risk of human deaths and injuries.

UGVs have improved efficiency and production. They can constantly do repeated jobs and function continuously without breaks or relaxation. In industries such as transportation, agriculture, manufacturing, and warehousing, this results in improved operational effectiveness, cost savings, and increased production. UGVs also provide enhanced accuracy and precision in operations including surveillance, assessment, and data collecting (H. Yu, 2010). Because of their sensors and clever algorithms, they can gather and analyse massive volumes of data, offering significant insights for making choices and process optimization.

1.2AIMS AND OBJECTIVES

AIMS

The project's objective is to develop a UGV with terrain exploration and landscape mapping capabilities. The UGV also includes a number of other features, including the ability to map the landscape using a method called Simultaneous Localization and Mapping (SLAM), which can then display the map on a desktop computer. The autonomous robot will also be able to avoid obstacles when moving through the environment. Third, the unmanned vehicle will have a power control system that can evenly transfer power to motors to prevent course deviations. Last but not least, the vehicle will have a positioning adjustment mechanism that can reduce the distance between the robot's course and the desired path to the destination.

OBJECTIVES

- Creating a dependable and durable UGV platform with autonomous mobility and navigation capabilities.
- Applying real-time image processing algorithms to the visual data gathered by the UGV's sensors, providing perceptual intelligence and deliberative capability.
- Improving the UGV's capacity to detect and distinguish between various objects, patterns, and anomalies to help with duties like monitoring, inspection, and surveillance.
- Including communication tools to enable easy contact between UGVs and human operators, facilitating data sharing and remote control.
- Thorough testing and evaluation will be used to validate and improve the UGV system, ensuring its practical application and scalability for a variety of industries, including defense, public safety, and industrial automation.

Chapter 2

2.1 LITERATURE REVIEW

2.1.1 HISTORICAL DEVELOPMENT OF UGVS

ORIGINS AND EARLY DEVELOPMENTS OF UGVS

The origins of Unmanned Ground Vehicles (UGVs) can be traced back to the mid-20th century, with roots in both military and industrial applications.

Military Origins: The earliest UGVs were primarily developed for military purposes. During World War II, the German military deployed the "Goliath tracked mine," a remotely operated, tracked vehicle filled with explosives used to destroy tanks and fortifications. This can be considered one of the earliest precursors to modern UGVs.

Industrial and Research Beginnings: In the post-war years, UGV research and development expanded beyond military applications. Universities and research institutions began exploring autonomous and remotely controlled vehicles for various purposes, including materials handling and exploration of hazardous environments. This marked the beginning of UGVs as tools for scientific research and industrial automation.

2.1.2 KEY MILESTONES IN UGV TECHNOLOGY

UGV technology has seen significant advancements over the decades, marked by key milestones:

2.1.3 Stanford Cart (1960s)

The Stanford Cart, developed in the 1960s, was one of the earliest autonomous vehicles. It could navigate its environment using sensors and stored maps, marking a milestone in autonomous robotics.

2.1.4 Shakey the Robot (1970s)

Shakey, developed at the Stanford Research Institute in the 1970s, was a groundbreaking UGV equipped with sensors for perception and problem-solving capabilities. It laid the foundation for modern UGVs' ability to plan and execute tasks in dynamic environments.

2.1.5 UGVs in Space (1990s)

In the 1990s, UGVs were deployed for extraterrestrial exploration. The Sojourner rover, part of the Mars Pathfinder mission (1997), marked a historic moment as it became the first UGV to explore another planet, contributing to our understanding of Mars.

2.1.6 DARPA Grand Challenge (2004)

The Defense Advanced Research Projects Agency (DARPA) Grand Challenge in 2004 was a pivotal event in UGV development. It challenged teams to build UGVs capable of navigating a desert route autonomously. The success of this challenge showcased the progress in autonomous vehicle technology.

2.2 EXAMPLES OF HISTORICAL UGV PROJECTS AND THEIR CONTRIBUTIONSTALON:

2.2.1 **The TALON**

UGV, developed by Foster-Miller in the early 2000s, gained recognition for its versatility. It was initially designed for bomb disposal but found applications in search and rescue missions, as well as hazardous materials handling. TALON demonstrated the adaptability of UGVs in various domains.

2.2.2 Curiosity Rover

Launched by NASA in 2011, the Curiosity rover exemplifies UGVs' contributions to space exploration. It conducted scientific experiments on Mars, collected valuable data about the planet's geology, and searched for signs of past microbial life, expanding our knowledge of the Red Planet.

2.2.3 UGVs in Agriculture

UGVs have played a crucial role in modern agriculture. Robots like the Lely Astronaut automated milking system and autonomous tractors have revolutionized farming practices by improving efficiency and precision in tasks such as planting, harvesting, and livestock management. **(USA, 2013)**

2.2.4 UGVs in Mining

UGVs have been employed in the mining industry for tasks such as exploration, mapping, and even autonomous hauling of materials. These machines enhance safety by reducing the need for human presence in hazardous mining environments.

2.3 UGV APPLICATIONS THROUGH TIME

Over the years, Unmanned Ground Vehicles (UGVs) have undergone a remarkable transformation, diversifying their range of applications across various industries. This section will explore the evolving landscape of UGV applications, including their roles in military operations, agriculture, healthcare, and more. We will delve into historical case studies that demonstrate how UGVs have been used in these domains and discuss their significant impact on these industries over time.

2.4 MILITARY APPLICATIONS

2.4.1 HISTORICAL CASE STUDY: REMOTEC ANDROS F6A (1980S)

UGVs have a rich history in military applications, dating back to their early use for bomb disposal and reconnaissance. One notable case is the Remotec ANDROS F6A, developed in the 1980s. It was one of the first UGVs designed for explosive ordnance disposal (EOD) tasks. The ANDROS F6A allowed military personnel to remotely handle and dispose of dangerous explosives, reducing the risk to human life.

IMPACT ON MILITARY OPERATIONS

UGVs have significantly impacted military operations by providing a safer means of handling explosives, scouting hostile environments, and conducting surveillance. They are also used in logistical support, transporting supplies in challenging terrains, and as reconnaissance tools in conflict zones. UGVs enhance military efficiency and reduce the exposure of soldiers to dangerous situations.

2.4.2 AGRICULTURAL APPLICATIONS

HISTORICAL CASE STUDY: CARNEGIE MELLON UNIVERSITY'S "ROVER" (1980S)

UGVs have played a crucial role in modernizing agriculture. In the 1980s, Carnegie Mellon University developed a UGV known as the "Rover" for agricultural tasks. This early example demonstrated the potential for automation in farming, as the Rover could navigate fields, monitor crops, and collect data.

IMPACT ON AGRICULTURE

UGVs have revolutionized agriculture by enhancing precision, efficiency, and sustainability. They are used for tasks such as planting, harvesting, soil analysis, and weed control. UGVs equipped with sensors and GPS technology enable precise application of resources like water and fertilizers, reducing waste and environmental impact. They have also addressed labor shortages in agriculture, particularly in regions where manual labor is scarce.

2.4.3 HEALTHCARE APPLICATIONS

HISTORICAL CASE STUDY: CYBERKNIFE ROBOTIC RADIOSURGERY SYSTEM (1990S)

UGVs have found applications in healthcare, particularly in the field of medical robotics. The CyberKnife Robotic Radiosurgery System, introduced in the 1990s, is a prime example. This UGV is used for precise, non-invasive cancer treatments. It delivers high doses of radiation to tumors with pinpoint accuracy, minimizing damage to surrounding healthy tissue.

IMPACT ON HEALTHCARE

UGVs in healthcare have transformed the way complex medical procedures are conducted. They enable surgeons to perform minimally invasive surgeries with greater precision, reducing patient recovery times and complications. UGVs also facilitate telemedicine by allowing remote specialists to participate in procedures, improving access to healthcare in remote areas.

Search and Rescue Applications

HISTORICAL CASE STUDY: PACKBOT (2000S)

The PackBot, developed by iRobot in the early 2000s, has been instrumental in search and rescue operations. It was notably deployed in the aftermath of the 9/11 attacks and during disaster responses, such as Hurricane Katrina. The PackBot's mobility and sensors enabled it to navigate hazardous environments, locate survivors, and assess damage.

IMPACT ON SEARCH AND RESCUE

UGVs have proven invaluable in search and rescue efforts by accessing areas too dangerous for human responders. They can provide real-time data and communication capabilities to rescue teams, improving situational awareness. UGVs save lives by expediting search and rescue missions and increasing the chances of finding survivors.

2.4.4 ENVIRONMENTAL MONITORING APPLICATIONS

HISTORICAL CASE STUDY: GAVIA AUTONOMOUS UNDERWATER VEHICLE (2000S)

UGVs are not limited to terrestrial applications; they also extend to underwater environments. The Gavia Autonomous Underwater Vehicle, developed in the 2000s, is used for environmental monitoring and underwater exploration. It collects data on oceanography, marine biology, and geology.

IMPACT ON ENVIRONMENTAL MONITORING

UGVs have revolutionized environmental research by enabling data collection in remote or hazardous environments. They contribute to our understanding of ecosystems, climate change, and geological phenomena. UGVs in underwater applications have been vital for exploring the depths of oceans, gathering data on marine life, and mapping the seafloor.

UGVs have typically been created for military purposes since the 2000s. The USSR created Trletanks (A.A. Nayak, 2012.). in the 2000s, a machine gun-equipped tank that could be remotely operated by radio from another tank. At the time, robots were operated manually. Today, however, UGVs are employed extensively for autonomous transportation and navigation across many industries. The first autonomous vehicle was displayed at the New York World's Fair in 2000s, which was sponsored by "General Motor" (Anderson, 2019). Following then, other improvements have been taken into account in order to create a completely autonomous vehicle (B. Hendrik, 2013.) (A.A. Nayak, 2012.).

2.5 EVOLUTION OF UGV TECHNOLOGIES

Unmanned Ground Vehicles (UGVs) have undergone a remarkable evolution in terms of their technologies, which have played a pivotal role in shaping their capabilities. This section will delve into the historical and current technologies employed in UGVs, including navigation systems, sensors (such as LIDAR, cameras, and IMUs), and communication systems. We will explore how these technologies have evolved over time, enabling UGVs to become more autonomous, versatile, and efficient.

2.5.1 NAVIGATION SYSTEMS

Navigation is one of the fundamental aspects of UGV technology, and it has evolved significantly from basic remote control to sophisticated autonomous systems.

Early Navigation (Remote Control): In the early days of UGVs, navigation relied heavily on manual remote control. Operators would control UGVs using joysticks or other input devices, which limited their range and autonomy. These systems lacked the ability to navigate complex environments independently.

GPS Integration: The integration of Global Positioning System (GPS) technology in the 1980s marked a significant advancement. GPS enabled UGVs to determine their precise location, making it possible for them to follow predefined routes, navigate large areas, and perform tasks with greater accuracy. (B.J. Yoon)

SLAM (Simultaneous Localization and Mapping): In the 2000s, UGVs started to incorporate SLAM technology. SLAM algorithms allow UGVs to create maps of their surroundings while simultaneously determining their position within those maps. This development greatly improved their ability to navigate in unfamiliar and dynamic environments.

Machine Learning and AI: Modern UGVs are increasingly equipped with machine learning and artificial intelligence (AI) algorithms. These technologies enable UGVs to learn from their experiences, adapt to changing conditions, and make real-time decisions, enhancing their autonomy and adaptability.

2.5.2 SENSORS

Sensors are the eyes and ears of UGVs, enabling them to perceive and interact with their environment. Several types of sensors have played crucial roles in UGV technology.

Cameras: Cameras have been an integral part of UGVs from the beginning. Early UGVs used basic cameras for visual inspection and remote operation. However, advancements in camera technology, including higher resolution and real-time image processing, have greatly improved perception capabilities.

LIDAR (Light Detection and Ranging): LIDAR technology, which uses laser beams to measure distances and create 3D maps of the environment, has revolutionized UGV perception. It provides highly accurate spatial data, allowing UGVs to navigate complex terrains and avoid obstacles effectively.

IMUs (Inertial Measurement Units): IMUs, consisting of accelerometers and gyroscopes, have improved UGVs' ability to maintain orientation and stability. They are crucial for tasks that require precise positioning and motion control, such as terrain mapping and vehicle stabilization.

Radar and Sonar: Radar and sonar sensors are used in UGVs for obstacle detection and navigation in challenging conditions. These sensors can operate in adverse weather conditions, such as fog or rain, making them valuable for various applications.

Communication Systems: Communication systems enable UGVs to receive instructions, transmit data, and collaborate with human operators or other UGVs.

Radio Control: Early UGVs relied on radio control for communication. While effective for short-range operations, it limited the UGV's range and autonomy.

Wireless Networks: The integration of wireless networks, such as Wi-Fi and cellular connectivity, expanded the communication range and enabled UGVs to send and receive data in real-time. This allowed for remote monitoring and control over long distances.

Mesh Networks: Mesh networks, introduced in the 2000s, allowed UGVs to create ad-hoc networks among themselves. This technology enhances UGVs' ability to communicate in remote or challenging environments where traditional networks may not be available.

2.6 EVOLUTION AND IMPROVEMENT OF UGV CAPABILITIES

The evolution of UGV technologies has led to significant improvements in their capabilities:

2.6.1 Autonomy

UGVs have become increasingly autonomous, thanks to advanced navigation and sensor technologies. They can now operate in complex and dynamic environments, plan their routes, and make decisions without constant human intervention.

Safety: Improved perception through sensors like LIDAR and cameras has enhanced UGV safety. They can detect obstacles more accurately and react swiftly to avoid collisions, reducing the risk of accidents.

2.6.2 Efficiency

UGVs equipped with advanced sensors and navigation systems are more efficient in carrying out tasks. They can optimize routes, conserve resources, and reduce operational costs in various applications, such as agriculture, logistics, and mining.

2.6.3 Versatility

The flexibility of modern UGVs, driven by machine learning and AI, allows them to adapt to different tasks and environments. They can switch between applications with minimal reprogramming.

2.6.4 Remote Operation

Enhanced communication systems have extended the range of UGVs and enabled remote operation from command centers. This capability is critical in military, search and rescue, and exploration applications.

2.7 HISTORY

Google has already created an autonomous vehicle that can navigate itself between locations without human assistance. The Taifun-M UGV was introduced by the Russian Army in April 2014 as a remote sentry to guard the RS-24 Yars and SS-27 Topol-M missile sites. The Taifun-M has a single cannon and laser aiming to perform reconnaissance and patrol missions, find and eliminate fixed or moving targets, and support security personnel at guard sites with fire. Currently controlled remotely, they intend to eventually integrate an autonomous artificial intelligence system to make them totally autonomous.

Unmanned ground vehicle research has become more prevalent, with the majority of the focus being on navigation, mapping, and obstacle avoidance. These difficulties have been taken into account and documented in several research articles. There are several articles that look at ways to get around them. The next pages will discuss several strategies that may be utilized to address these problems, and this portion of the dissertation will analyse existing procedures in case they serve as a baseline for future study into increasing the performance of the ground vehicle.

Obstacle detection and avoidance must be possible for the unmanned vehicle. It needs sensors that can measure the separation between the sensor and the obstacle in order to avoid obstacles. To recognize an obstacle, however, requires a complex configuration system that takes into account the obstruction's looks and how apparent the obstacle is to the sensor because certain obstacles may have varying forms or sizes [7-8]. As a result, we will examine the sensors used to identify obstacles, including vision sensors, ultrasonic sensors, infrared sensors, and laser range finders.

Path planning and obstacle avoidance for both static and moving objects make up the navigation challenge. Environments. Numerous methods and approaches, including artificial potential fields, visibility graphs, genetic algorithms, simulated annealing, and particle swarm optimization, can be used to accomplish this. Swarm optimization and an optimization method using ant colonies. It has been noted that the majority of the work done has been applied to static surroundings, which does not mean that the UGV would behave in a smart manner to avoid obstacles and get where it is going. (Wikipedia, accessed on Oct – 2014.)

Although several approaches of the previous studies have demonstrated the use of UGV in dynamic situations, various restrictions and limits were assumed to simplify the navigation process, such as the assumption that obstacles move at a constant pace and are the same size and form. The author uses sophisticated controllers like artificial intelligence in a dynamic environment where barriers move arbitrarily at different rates and directions. The barriers are also thought to be built using a range of sizes and forms based on real-world events. Such considerations will address every obstacle that the UGV could run against while navigating such technologically advanced workplaces. The UGV will be given intellectual capabilities by the intelligent controllers to carry out its manoeuvrability effectively and safely.

One of the ready-to-use transmitter-receiver (transceiver) modules supplied on the market is a remote control with four to six channels that is frequently used in model aircraft. The use of remote control (RC) is widespread and can be used to run a model car or a number of other specific devices. Unmanned Maritime Systems (UMSs), Unmanned Ground Vehicles (UGVs), and Unmanned Aircraft Systems (UASs) can all be operated remotely [8]. A vehicle that can be operated remotely is shown in Fig. 1[8–10]. In general, UGVs have four to eight independent rotating wheels in addition to manipulators, coupling tracks, pan-tilt camera mechanisms, and switch modes.

Figure 2 REMOTE CONTROLLED VEHICLE (a) Excavator (b) Unmanned ground vehicle

In Hendrik's paper [12], the omnidirectional robot is steered using a remote control. The microcontroller transforms the six channel data received by the receiver module into 4 bit data to drive a DC motor either clockwise (CW) or counterclockwise (CCW). Only four DC motors can be driven with the remote control. The transceiver module can be used to transmit the numbers 0 to 9 thanks to changes made by Samo sir (O. Mahendra, 2012,). The microcontroller reads keypad input before converting it to binary code and sending it through a transmitter module. After being reconverted by the microcontroller, the data received by the receiver module is displayed through seven segments (Howe, 1999) (Wikipedia, accessed on Oct – 2014.). The used remote-controlled automobile has four bits (forward, backward, and left/right). It has a high "1" value and a low "0." Because forward-backward and right-left cannot both have high ("1") values simultaneously, some data cannot be delivered. Signal identification method is crucial for remote controls with PWM outputs. The 16-bit timer on the microcontroller is utilized as an internal counter in Mahendra's PWM signal identification approach (Howe, 1999) (J.-S. Valois, presented at the Unmanned System Technology X, 2008) to determine the pulse width of the PWM with a resolution of 1/crystal frequency.

2.8 REMOTE CONTROL CONNECTIONS

2.8.1 THE CURRENT USED DESIGN

The remote control that is available on the market is highly useful and can generally be utilized without modification. Fig. 2(a) illustrates the connections that are typically utilized on remote controls. It is not sufficient to use servo motors as actuators for gadgets that need high speed or more torque. It calls for modifying the receiver output module by including further modules, like a microcontroller (see Fig. 2(b)).

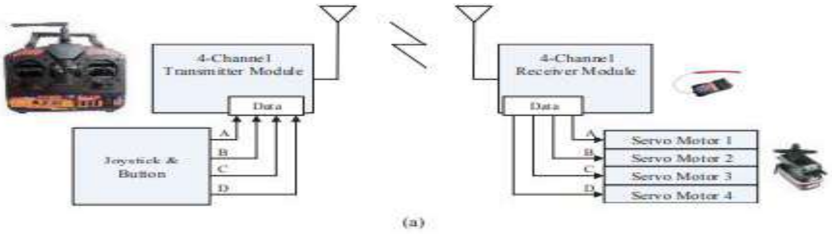


Figure 3 GENERAL CONNECTION FOR CONTROLLING A 4 MOTORS

(a) Without microcontroller

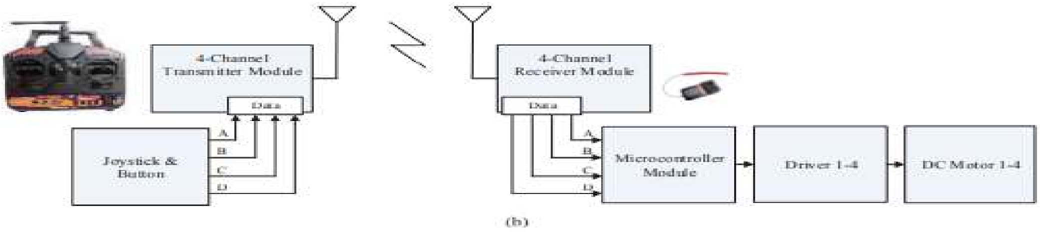


Fig. 2. General connections for controlling 4 motor: (a) without microcontroller; (b) with microcontroller.

Figure 4 GENERAL CONNECTION FOR CONTROLLING A 4 MOTORS

(b) With Microcontroller

As indicated in Fig. 2, the general connections for remote control of four motors can be made with or without a microcontroller. The connection seen in Fig. 2(a) is present in every model of aircraft (helicopter, plane, etc.) that employs servo motors as actuators. A microcontroller has been added to Fig. 2(b) to process the signal from the receiver module and drive various DC motor types.

2.8.2 THE SUGGESTED DESIGN

The configuration in Fig. 2 has two drawbacks: (1) data communication is only possible through analog voltage input and PWM output; and (2) due to the configuration's four channels, a maximum of four motors can be used. The connection diagram in Fig. 3 can be utilized to make the remote control more dynamic when sending data. Depending on the message's aim, the data in Fig. 3 may consist of a collection of letters or numbers. The sole purpose of a transmitter and receiver module was to send data in hexadecimal code (R. C. Detmer, 2014.). Two channels are utilized as the address and command for the slot array's two identifiers, while the other two channels are used for data. A 0–5 analogue signal. In order to guarantee the precision and correctness of the data, each channel that serves as the transmitter module's input is divided into 16 sections. Fig. 4 illustrates each channel's function.

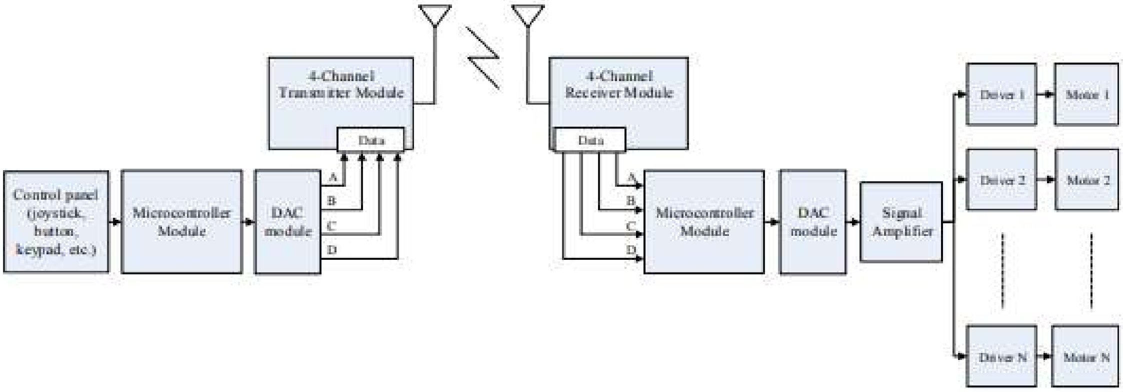


Figure 5 DISTRIBUTION OF DUTIES FOR EACH CHANNEL

Proposed connection for controlling UGV.

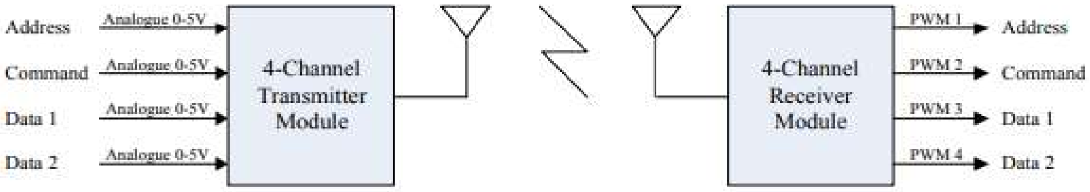


Figure 6 DISTRIBUTION OF DUTIES FOR EACH CHANNEL

Distribution of duties of each command

Additionally, the unmanned ground vehicle has to be able to localize itself while simultaneously mapping the environment. The goal of this task is to locate the vehicle's position and orientation on the map while also gathering a number of nearby unidentified landmarks. In addition, it must to map the area around the car based on its location. Simultaneous Localization and Mapping (SLAM), which has numerous varieties as Hector SLAM, G mapping, and Karto SLAM.

2.9 OBSTACLE AVOIDANCE

An integrated computer in the autonomous vehicle's distance sensor may translate incoming input (such as a laser beam, sound wave, or infrared ray) into an electrical signal, record the information, and then carry out a predetermined action. This signal may be employed to map the surroundings and be used to detect the surrounding environment. Different sensor types can be applied to unmanned ground vehicles.

2.9.1 INFRARED IR SENSOR

Infrared sensors work on the basis of emitting infrared rays with wavelengths between 700 nm and 1 m, and then detecting changes in the wavelength of the reflected light as a result of an object's absorption property. The signal pin is activated when the wavelength falls within a particular range, indicating that the ray was reflected by the obstruction.

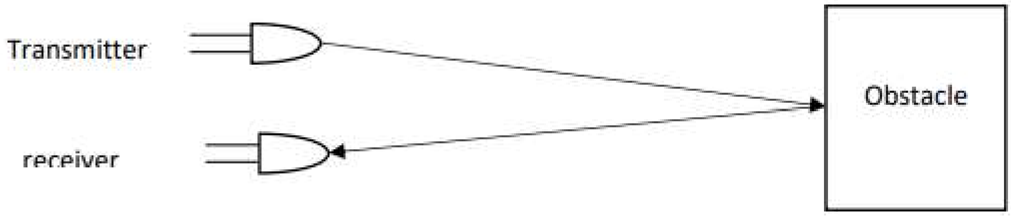


Figure 7 INFRARED SENSOR

2.9.2 ULTRASONIC SENOR

A technique used by ultrasonic sensors is called Time of Flight (TOF), in which the transmitter sends out an acoustic wave in the direction of the obstacle, and the receiver picks up the wave that is reflected back. The distance between the sensor and the obstruction is determined by the amount of time between transmission and reception.

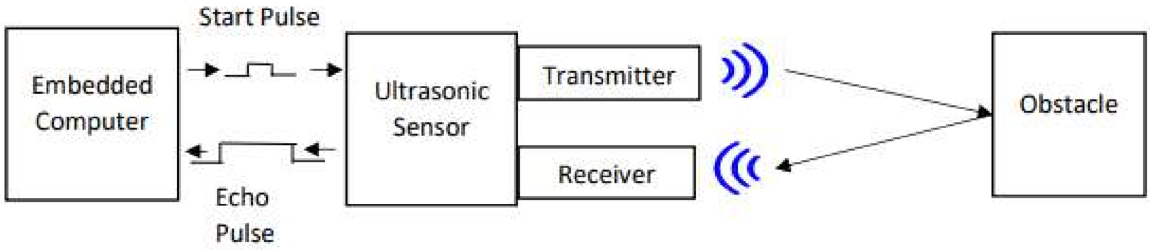


Figure 8 ULTRSONIC SENSOR

2.9.3 LASER RANGE FINDER SENSOR

Similar to an ultrasonic sensor in concept, a laser rangefinder uses laser beams.

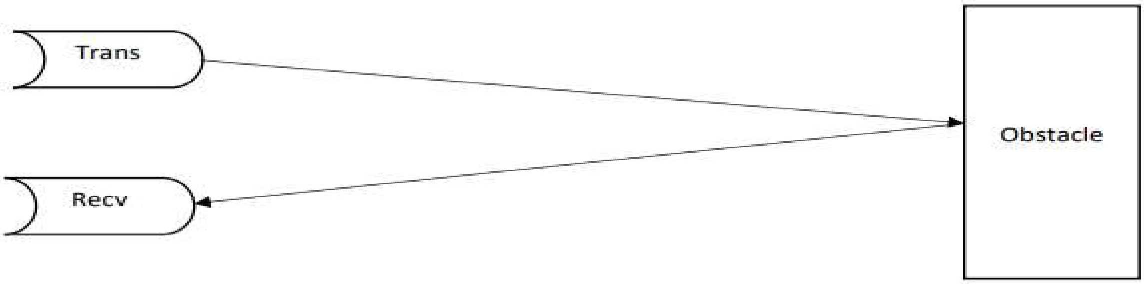


Figure 9 LASER RANGE F SENSOR

2.9.4 VISION SENSOR

The monocular vision approach and the stereo vision approach are the two main hypotheses behind the vision sensor. Monocular vision is able to identify obstacles by obtaining their appearance and identifying their features. The stereo vision technique examines an obstruction from, say, two places and produces a 3D depth map.

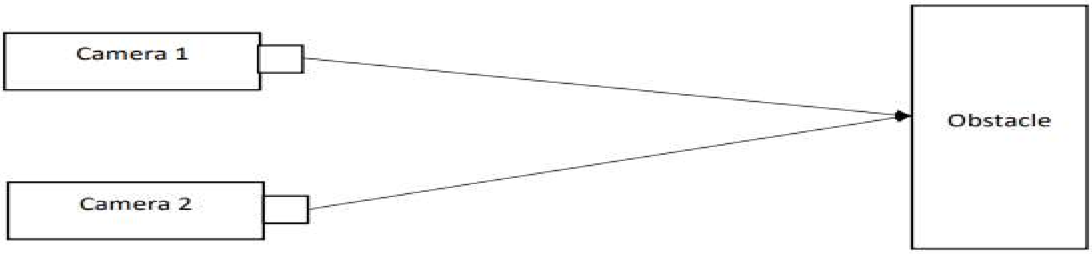


Figure 10 VISION SENSOR

As a result, it is crucial to pick the right sensor since its use in mapping will have an impact on the data about the area around the vehicle. (Thrun, 2005)

Adarsh et al. (2016) looked at the demonstration of the infrared and ultrasonic sensors in obstacle ID with obstructions made of diverse materials. These sensor data are influenced by a number of factors. For instance, infrared sensor data may alter depending on the colour of the obstruction material (Mohamed Kassim et al., 2016). However, because the ultrasonic sensor is sensitive to changes in environmental factors like temperature and clamminess, the information it provides can be altered.

According to Shrivastava, Verma, and Singh (2010), the ultrasonic sensor is also sensitive to objects with somewhat sharp surfaces. The research looked at how well sensors made of cardboard, wood, flexible plastic, paper, wipes, and tile performed at a distance. The assessment's delayed result revealed that the infrared sensor is appropriate for detecting obstacle materials like paper and wipe, while the ultrasonic sensor is appropriate for detecting get materials like wood, tile, plastic, wipe, and cardboard.

In his study, Panich (2010) considered how far an ultrasonic sensor may be from a sound structure vision. The review examined the effects of the sensor investigation taking into account its update rate, where the rate depends on the robot's continuous movement. The analysis revealed that the sound system vision distance error is 36.9 cm, which is different from the ultrasonic distance error of about 3.6 cm because the sound structure vision requires two-fold managing speculation because it compares two images using the same calculation. (NYWF., accessed on Oct-2014.)

The study that Mohamed Kassim et al. (2016) appropriated has provided an analysis that looks at a few sensors, including ultrasonic, infrared, and laser range locater sensors. The experts used a condition to choose which sensor should be used taking into account its features including reach, accuracy, size, cost, weight, and energy use. According to the assessment, the ultrasonic sensor has accuracy, range, size, weight, cost, and energy use, whereas the laser range locater has exactness and show up at, giving it a weighting score of 2.666. The infrared sensor, on the other hand, has four characteristics—size, weight, cost, and energy consumption—and scored 1.466 for weight.

An article by Chong et al. (2015) that takes into account the sensor degrees of advancement used in Sledge has been widely disseminated. The evaluation looked at sensors like laser range finders, acoustic sensors (ultrasonic sensor), sound structure vision, and researched each aspect of the sensors. It revealed that ultrasonic sensors are most frequently used in robots because they are unobtrusive and can function with most surface materials that have good acoustics.

The sensor, however, is sensitive to any change in climate, has a low spatial target, recognizes a limited range, and responds slowly. While the laser range locater is a well-known option for Hammer because it can provide an astounding result of the weather whether it was placed indoors or outside. The arrival locater is additionally renowned for its speed and accuracy, which enable it to do a clear distance calculation.

In his article, Xiaobin Zhou, proposed a strategy for the planning and tracking of the UGV's while the obstacles are there. As moving to its destination more precisely and more effectively is the most important parameter for an unmanned ground vehicle, so this tracking strategy provides a valuable information for the given work. With the optimal environment, the destination for the given UGV is also defined previously using the artificial fish swarm algorithm (AFSA) so that it could be connected at a later time. A trial-based forward search (TFS) algorithm is also proposed on the chain so that the local trajectory module could start working and at the same time for the collision prediction, it is assumed as the heuristic information. Finally, the outcomes of both simulations and tests show that the suggested method works well in the context of dynamic as well as static barriers. (Jennifer Carlson, 2004.)

Jianwei Gong, in his article proposes an interesting technique for the navigation and motion of the unmanned ground vehicles, which also gave a significant modernization for the purpose. In his paper, the author discusses a Lasor Radar technique for the tracking of the unmanned ground vehicle. The local minimum problem is minimized using move-to-goal and wall following techniques. With these techniques, navigation and obstacle avoidance within the unknown conditions or environment is made possible. With these effective parameters, two more important things are also covered like position prediction and state memory so that the coordination could be made possible only at the right time. In the paper, the author further utilized his previous paper which was discussing the navigation and obstacle avoidance using the VPH+. Local minimum problem and the position finding problems were improved using these. Contrastive simulations and real-world UGV vehicle testing revealed that the suggested technique is resilient, stable, and efficient in complex situations.

Abhik Singla, in his article (2019), proposed a very new approach for the Unmanned Ground Vehicle to avoid the obstacles in its path. When the vehicle has very limited knowledge about the surroundings, then using the memory based deep reinforcement learning is something which can help for better results. A camera has been used to see the path of the vehicle when the environment is not structured and the vehicle is also not familiar with the path. The current proposed technique used the monocular pictures or images to get information about the path. The deep learning-based technique for this avoidance can do the exact work. This approach employs recurrent neural networks using temporal concentration and outperforms previous research in terms of range travelled without collisions. Furthermore, our approach has a high reasoning rate and decreases power waste by minimizing the oscillatory motion of the UAV.

Planning to get the most optimal path for the Unmanned Ground Vehicle has always been on board. So, Mohammad Rzea Jabbarpour, discussed this problem in her article. This article is based on the problem that while going for the most optimized path for the UGV's, they consume a lot of energy using the battery provided. There must be solution to this so that the energy consumption or power loss in this scenario could be minimized and the Unmanned Ground Vehicle could work for a longer span. For this purpose, Green Ant based algorithm has been proposed. The G-ant methods is integrated with ant-based algorithms so that the shortest path could be found or evaluated which is the main goal of the article. Then the results of the G-ant baes algorithm are compared with colony optimization, genetic algorithm, and particle swarm optimization approaches. Multiple circumstances were simulated to assess G-Ant effectiveness in terms of UGV trip time, journey length, and computing time while accounting for varied numbers of cycles, obstacles, and numbers of people. The outcomes showed that G-ant based algorithm is the best of all in terms of travel length and number of iterations.

Sangwoo Moon, also discussed a real time study for the tracking of the trajectory of the Unmanned Ground vehicle. By this strategy, the obstacles in the very unknown and unfamiliar environment are discussed. This method for the avoidance of the obstacles is based on the potential field approach and some modifications are made for the previous algorithms because the navigation function can measure not only the path but the position and the velocity of the UGV's while moving on that path. The costs are calculated using finite future states, which are solved using dynamical equations and vehicle limitations. Although this approach is basic, it can efficiently discover the path since it employs simple equations to generate an acceptable route and provides high dependability for the outcome.

A manoeuvre control strategy for an unmanned aerial vehicle (UGV) with obstacle avoidance is described by Cao Lei in his article. This work offers a lateral and longitudinal control system for an unmanned ground vehicle (UGV) for obstacle avoidance. A local path is initially suggested to achieve UGV obstacle avoidance. The direct collocation point technique and NMPC are paired to achieve avoiding obstructions. The simulation results show that when an obstruction is present, UGV may track both a prescribed velocity curve and a locally new route. This improves the agility and flexibility of the UGV's operation.

Hongyan Guo, in his research paper also discussed an algorithm for the obstacle avoidance. In this article, the avoidance is made possible with the Moving Horizon Optimization on the autonomous ground vehicle. The braking method is addressed utilizing moving horizon optimization based on differentiated flatness, with a 3 DOF vehicles model following the planned path for UGV to avoid obstacles. The optimum problem of eliminating obstacles is defined as minimizing the bias of the lateral movement while taking lateral acceleration into account. The simulation in a multi-vehicle setting is used to validate the successful implementation of the suggested strategy.

An actual time collision prevention approach for unmanned ground vehicles (UGV) is proposed in this study given by TokSon Choe. Virtual force fields are computed in three dimensions to ensure real-time application. The vehicle's steering field of forces is created by the steering command, which is either communicated or computed in the UGV's autonomous navigation system (ANS). Range information collected by a laser length finder (LRF) installed on the UGV is used to produce an impediment force field. To circumvent obstacles and continue a planned course, adjusted steering, velocity, and emergency stop orders are made using the combined force field overlapping these driving and obstacle stress fields. The proposed technique is incorporated as a component of the experimental autonomous vehicle (XAV) system. The efficacy and practicability of the suggested technique are validated by several real-world experiments and testing utilizing the XAV.

Autonomous Ground Vehicles Obstacle Identification System Using Laser Scanner and Vision has been proposed by Hee-Chang Moon. This study describes an unmanned ground vehicle obstacle detection system. The UGV's obstacle detection system is critical for operating stability and security. Our obstacle detection system is made up of two parts. The first is a laser system component, while the second is a vision system component. The laser scanner is incapable of determining precise details of an obstruction. It is difficult to recognize some types of impediments. It can, however, obtain a precise location of the obstruction. The vision system collects a large amount of data. As a result, data processing is challenging. As a result, it obtains the position of the obstruction from the Laser scanning system component. And the vision sensor component can detect various types of obstructions on the ground. These systems are built on the JAUS platform. This study introduces a UGV system that can configure and explain a system for identifying obstacles utilizing an optical scanner and a video camera.

Optimal route planning is a critical challenge in independent unmanned ground vehicle control. In the static circumstances, particle swarm optimization was employed to tackle the optimal problem; nevertheless, optimal route selection for UGV units in a dynamic context has not been completely studied. As a result, dynamic obstacle-avoidance route planning for an autonomous ground vehicle group was chosen as the best solution for the shortest path considering formation limitations. The task was defined in Cartesian space, with the observable vehicle and barrier velocity. The fitness value was determined by minimizing the group's trajectory while maintaining the group's V-shape configuration. Analysing the convergence of the PSO algorithm determines the stable zone of the parameters in the paper by Yunji Wang.

Se, Lowe, and Little (2001, 2002, 2005) used EKF because it reduced the uncertainties of the environment features that were currently stored and compared the stored features with new features in the event that the robot revisited the same environment, according to Chen, Samarabandu, and Rodrigo's (2007) investigation of this filter.

The robot was able to localize itself, thus Madhavan, Fregene, and Parker (2004) used EKF to create a local map using the robot's posture, elevation gradient, and vision-based range. However, when the Extended Knowledge Filter (EKF) and Sparse Extended Information Filter (SEIF) are compared, EKF is slower and uses more memory while SEIF creates more errors.

Aulinas et al. examined a number of alternative filters in their 2008 study, including KF, PF, Information Filter (IF), Expectation Maximization (EM), and Compressed Extended FK (CEKF). The analysis demonstrates the advantages of employing KF/EKF because it has strong convergence and can handle uncertainty. The filter has drawbacks, though, including the usage of Gaussian noise and the development of high-dimensional maps that is delayed.

A couple of the filtering methods to be applied in underwater robotic navigation were evaluated by Yuan et al. in 2017. In their analysis, several of the benefits and drawbacks of using KF/EKF and PF were highlighted. Since KF/EKF can tolerate uncertainty when calculating the robot's posture and the filter can determine if the environment's geometry is approximately identical to the real world, using KF/EKF over PF offers the advantage of high convergence. (R. C. Detmer, 2014.)

The usage of the Gaussian noise assumption and the delayed creation of high-dimensional maps are drawbacks of the KF/EKF filter. PF has an advantage over EKF/KF because it does not rely on the Gaussian assumption and can handle models of nonlinear systems. The PF's drawback, on the other hand, is that as the map environment gets more complicated, the filter's computing cost rises.

Unmanned ground vehicles (UGVs) use particle filter SLAM to do real-time mapping and localization in uncertain and dynamic settings. In Particle Filter SLAM, UGVs use a collection of particles to represent the robot's posture and the map, with each particle representing a plausible hypothesis about the robot's position and the structure of the environment. The particles are modified and resampled over time to derive the real robot position and map by blending sensor readings with motion models. This method allows UGVs to deal with non-linearities, uncertainties, and complicated scenarios like loop closures and landmark identification. Particle Filter SLAM enables UGVs to estimate their position and build a map in real time and with high accuracy, making it appropriate for real-time and robust mapping.

The review by Aulinas et al. (2008) demonstrated that this filter is similar to Fast SLAM and what the fact that this SLAM struggles with data association while attempting to match several scans of the landmarks picked out by the robot sensor makes it notable.

This issue occurs for a variety of reasons, including the difficulty in seeing the same landmark again and mixing up previously scanned landmark with the recently scanned landmark (Riisgaard and Blas, 2004). The robot may even infer that there was a landmark at a certain position when there is actually none. This issue is resolved by the Fast SLAM approach, which divides it into two components: robot localization and landmark estimation. It helps the robot examine the path by using PF, and one benefit of this filter is that it uses less memory and processing resources than a regular filter.

In comparison to Kalman Filter, using this filter offers the advantages of not assuming Gaussian noise, not linearizing the motion of the vehicle, and being able to handle nonlinear systems, according to Zamora and Yu's (2013) assessment. The filter's consistency will decrease over time, making it unsuitable for prolonged use. Another drawback is that because it tries to re-observe the landmarks when it enters a high-dimensional area, it requires a lot of computing resources (Bailey, Nieto, and Nebot, 2006).

Point-Based SLAM is a popular mapping and localization approach in unmanned ground vehicles (UGVs). The environment is represented by the UGV in Point-Based

SLAM as a collection of distinct 3D or 2D point markers. The UGV progressively develops a map of the surroundings while concurrently calculating its own position within that map by exploiting sensor measurements such as lidar or laser range finder data. Based on feature extraction and matching algorithms, the UGV correlates new observations with existing landmarks or produces new landmarks. The UGV may enhance the map and optimize its self-localization estimate through this iterative approach. Point-Based SLAM is adaptable, robust, and extensively applicable, making it appropriate for a wide range of UGV applications such as navigation, research, and object detection in complex and dynamic situations.

Graph-based implementation of constraints between the landmark and the robot's posture in SLAM uses a graphical network. This method makes a number of assumptions, such as employing Gaussian noise, which is akin to EKF, but in reality, the map is only as accurate as the noise (Zamora and Yu, 2013). Additionally, the method presents a static view of the world, which may be helpful while dealing with a dynamic environment.

The graph-based process, according to Zamora and Yu (2013), uses optimization techniques that help with environmental condition prediction.

1. **Gradient descent:** The graph map is broken up into many map elements that show the action in real time.
2. **Conjugate gradient:** Since it reduces the computing load, the gradient examines matrix inversion.
3. **Stochastic gradient descent:** This technique has an advantage over the preceding two since it seeks out the global minimum posture by hopping between local minima. Additionally, a node reduction strategy is applied to hasten map convergence.

The particle filter method is used in particle-based SLAM to locate the robot on a map. In their 2007 study, Grisetti, Stachniss, and Burgard used this SLAM and the scan matching technique to estimate the location and pose of the robot using the resampling method, which aids in the retention of an appropriate number of particles.

Santos, Portugal, and Rocha (2013) assessed each package in simulation and in a real arena using a Hokuyo URG-04LX-UG01 laser sensor. MRL arena and lr5map are the simulation maps, with lr5map being a huge map with few distinguishing landmarks, while MRL arena is the real-world arena. The following points will go through the outcomes.

The scientific community is actively researching the problem of autonomous driving, and over the past ten years, the popularity of self-driving cars has rapidly increased. Recent advances in control techniques, path planning, and safety have resulted from study in this area of robots (Levinson et al. 2011, Hoang et al. 2014, Jo et al. 2014). In order to automate a vehicle, precise localization, optimal trajectory planning, and the execution of appropriate control signals are all necessary. Traditional algorithms for motion planning and obstacle avoidance rely on simple rule-based decision-making processes that respond to outside stimuli (Levinson et al. 2011). This work suggests a clever substitute employing an AI model trained as a Deep Reinforcement Learning (DRL) agent, given no prior knowledge of environment characteristics, for trajectory planning and obstacle avoidance for conventional 4-wheeled autonomous vehicles. The agent is demonstrated to intelligently converge to optimal policies in two different test scenarios with the same tuned network architecture and hyperparameters at the conclusion of the training phase.

Path planning is a crucial part of autonomous vehicle navigation, and creating an unobstructed trajectory is crucial to an Unmanned Ground Vehicle's (UGV) ability to complete its missions. Numerous research teams interested in various trajectory creation algorithms have expressed interest in this subject. These include the creation of waypoints, fuzzy logic algorithms, genetic algorithms, artificial potential field methods, convolution neural networks, and reinforcement learning algorithms (Watiasih et al. 2017, Bakdi et al. 2017, Zhu et al. 2010, and Liu et al. 2019). To be successful in real-world circumstances where computational resources may be restricted, the vehicle must demonstrate intelligent optimal behavior oriented towards reducing time and energy in addition to addressing the motion planning problem (Gómez et al. 2012).

2.10 CHALLENGES AND LIMITATIONS OVER TIME

While Unmanned Ground Vehicles (UGVs) have made significant advancements in terms of technology and applications, they continue to face challenges and limitations that have evolved over time. In this section, we will discuss both historical and contemporary challenges associated with UGVs, including autonomy, navigation in complex environments, safety concerns, and the need for further research to address these issues. (USA, 2013)

2.11 HISTORICAL CHALLENGES

2.11.1 Limited Autonomy

In their early stages, UGVs had limited autonomy and relied heavily on human operators for navigation and decision-making. This restricted their range of applications and effectiveness in dynamic environments.

2.11.2 Sensory Limitations

Early UGVs lacked advanced sensors, making it challenging for them to perceive their surroundings accurately. This led to difficulties in obstacle detection and navigation in complex terrains.

2.11.3 Communication Range

Communication technology was limited in the past, constraining UGVs' ability to operate over long distances. Remote operation and real-time data transmission were often hindered by communication restrictions.

2.12 CONTEMPORARY CHALLENGES

2.12.1 Autonomy and Decision-Making

While UGVs have become more autonomous, they still face challenges in decision-making, especially in complex and unstructured environments. Making real-time decisions in scenarios with rapidly changing conditions remains a significant challenge.

2.12.2 Navigation in Complex Environments

Despite advancements in sensors like LIDAR and cameras, navigating complex and cluttered environments, including urban settings or disaster zones, remains a challenge. Obstacle avoidance and path planning in these conditions require further improvements.

2.12.3 Safety Concerns

Safety is a paramount concern, particularly in applications like autonomous vehicles and robotics for healthcare. Ensuring that UGVs can operate safely alongside humans without causing harm or accidents is an ongoing challenge.

2.12.4 Human-Robot Interaction

Integrating UGVs into environments where humans are present, such as urban areas or medical facilities, requires effective human-robot interaction. Developing natural and intuitive interfaces for communication and collaboration remains a challenge. (Thrun, 2005)

2.12.5 Data Security

UGVs often collect and transmit sensitive data. Ensuring the security of data, as well as protecting UGVs from cyberattacks, is a growing concern that needs ongoing attention.

2.13 FUTURE TRENDS AND PROSPECTS OF UGVS

The future of Unmanned Ground Vehicles (UGVs) is both exciting and transformative, with emerging trends and advancements poised to reshape various industries. In this section, we will predict the future of UGVs, highlighting key trends and potential technological advancements, and discuss how UGVs are likely to shape industries in the coming years.

2.14 EMERGING TRENDS IN UGVS

2.14.1 Increased Autonomy

UGVs will continue to become more autonomous, with advancements in artificial intelligence (AI) and machine learning enabling them to make complex decisions in real-time. This trend will lead to greater flexibility and adaptability in a wide range of applications.

2.14.2 Sensor Fusion

The integration of multiple sensors, including LIDAR, cameras, radar, and IMUs, will become more sophisticated. Sensor fusion techniques will enhance UGV perception, enabling them to navigate diverse and dynamic environments with higher accuracy.

2.14.3 Human-Machine Collaboration

UGVs will increasingly collaborate with humans in shared workspaces, such as warehouses, hospitals, and construction sites. Natural language interfaces and intuitive control systems will facilitate seamless interaction between humans and UGVs.

2.14.4 Edge Computing

Edge computing will play a pivotal role in UGVs' decision-making processes. UGVs will process data locally, reducing latency and enabling faster responses to environmental changes and challenges.

2.14.5 Energy Efficiency

UGVs will see improvements in energy efficiency, enabling longer operational durations. This will be especially crucial for applications like agriculture, where UGVs need to cover large areas without frequent recharging.

2.15 POTENTIAL ADVANCEMENTS IN UGV TECHNOLOGY

Advanced AI and Machine Learning: UGVs will leverage advanced AI algorithms to enhance their ability to understand and navigate complex environments. Machine learning will enable UGVs to adapt to changing conditions and learn from their experiences.

2.15.1 Swarm Robotics

The concept of UGV swarms, where multiple UGVs collaborate autonomously to achieve tasks, will become more prevalent. This approach can be particularly useful in scenarios like search and rescue missions and environmental monitoring.

2.15.2 Biomechanics-Inspired Design

Bio-inspired UGVs, mimicking the locomotion and behaviors of animals and insects, may become more common. These designs can improve UGVs' mobility and adaptability in challenging terrains.

2.15.3 Robotic Vision

Advancements in computer vision and image processing will enable UGVs to recognize and interpret their environment with greater precision. This will enhance their object recognition, path planning, and overall situational awareness.

UGVs Shaping Various Industries

2.15.4 Agriculture

UGVs will continue to revolutionize agriculture by automating tasks like planting, harvesting, and pest control. Precision agriculture techniques, enabled by UGVs, will optimize resource usage and increase crop yields.

2.15.5 Logistics and Delivery

UGVs will play a vital role in logistics and last-mile delivery. Autonomous delivery robots will become a common sight in urban environments, reducing delivery times and costs.

2.15.6 Healthcare

UGVs will find broader applications in healthcare, including telemedicine, patient care, and medical logistics. Robotic assistants will aid healthcare professionals in tasks such as patient monitoring and medication delivery.

2.15.7 Mining and Construction

UGVs will continue to be essential in mining and construction operations. They will be used for exploration, excavation, and materials handling in challenging and hazardous environments.

2.15.8 Defense and Security

UGVs will see increased deployment in military and security operations. Their role will expand beyond reconnaissance and EOD tasks to include surveillance, logistics support, and even combat applications.

2.15.9 Environmental Monitoring

UGVs will be at the forefront of environmental research and monitoring. They will help gather data on climate change, wildlife conservation, and natural disaster management in remote or inaccessible areas. (R. C. Detmer, 2014.)

2.15.10 Transportation

UGVs will contribute to the development of autonomous vehicles, improving transportation efficiency and safety. Self-driving cars and trucks will become more prevalent on our roads, transforming the automotive industry.

2.16 AREAS FOR FURTHER RESEARCH

To address these challenges and limitations, further research and development efforts are essential:

2.16.1 Advanced AI and Machine Learning

Continued research in AI and machine learning algorithms is crucial for improving UGV decision-making capabilities. Developing algorithms that can handle complex and dynamic environments will enhance UGV autonomy.

2.16.2 Simulated Environments

Creating realistic simulated environments for UGV testing and training can help accelerate development and address safety concerns without the need for physical experimentation.

2.16.3 Ethical and Legal Frameworks

Developing ethical and legal frameworks for UGV deployment and operation, including liability and privacy considerations, is essential as UGVs become more integrated into society.

2.16.4 Interoperability and Standard

Establishing industry standards for UGV communication protocols and data formats will facilitate collaboration between different UGV platforms and enhance their overall capabilities. (USA, 2013)

2.16.5 Human-Machine Trust

Research on building trust between humans and UGVs is critical. Understanding how to convey UGV intentions and decisions transparently to human operators is essential for safe and effective collaboration.

2.16.6 Safety and Redundancy

Enhancing safety mechanisms, including redundancy in critical systems, and implementing fail-safe modes can mitigate the risks associated with UGV operations.

CHAPTER 3

3.1 METHODOLOGY

Our project has two part UGV and base station. In this project base station is used as the destination point means wherever the base station will be placed the UGV will go there through GPS connection to the satellite. In the project base station is also act as a transmitter which can transmit signal to UGV by there connection with the satellite through GPS. When base station is in contact with the satellite it can give a coordinate values (latitude and longitude). When it gives a high value it is stored in the external memory (i.e. raspberry pi 3) and when will give a small value then it is stored in the arduino mega and then transmitted signal further.

Our methodology for this project is that base station transmit data to xbee from where it is connected. We take a log of this data and also we calculate the base station values by taking log of it. Now we get a coordinates values for UGV. We can calculate the UGV values and base station values by taking their difference. Now we can put these values into the haversine formula which state that uses the longitude and latitude of two places on a sphere to calculate the great-circle distance between them. The law of haversines, which relates the sides and angles of spherical triangles, is a more broad formula in spherical trigonometry that is significant in navigation. Now the distance can be calculated by using the haversine formulae and the Arduino which is connected to the UGV send data to the IMU (Inertial Measurement Unit). UGV starts on the path and tracking its base station location. If there is some obstacle on the path UGV avoid it through IR or Proximity sensor and move it on path to reach its destination. But if the target is not achieved then the loop start working again and again until it reached its destination. (B. Hendrik, 2013.) (Wikipedia, accessed on Oct – 2014.)

3.2COMAPRISON OUR UGV WITH OTHER UGVs

3.2.1 PERCEPTION AND SENSING

UGV-IP: UGVs with image processing can leverage cameras and sensors to perceive their surroundings, identify objects, and analyze data in real-time.

Standard UGVs: Traditional UGVs may rely on simpler sensors like LiDAR or ultrasonic sensors, which may not provide as detailed or versatile perception capabilities.

3.2.2 AUTONOMY

UGV-IP: Image processing enables advanced autonomy features, such as object recognition, obstacle avoidance, and path planning based on visual data.

Standard UGVs: May have limited autonomy and rely more on manual control or basic sensor data for navigation.

3.2.3 OBJECT DETECTION AND RECOGNITION

UGV-IP: Can detect and classify objects, which is valuable for tasks like target identification, mapping, and navigation in complex environments.

Standard UGVs: May struggle to identify objects beyond basic obstacles.

Navigation and Mapping:

UGV-IP: Image processing can be used for simultaneous localization and mapping (SLAM), allowing UGVs to create detailed maps of their environment.

Standard UGVs: Mapping capabilities are typically less advanced.

3.2.4 OBSTACLE AVOIDANCE

UGV-IP: Image processing enables real-time obstacle detection and avoidance, making UGVs safer in dynamic environments.

Standard UGVs: May rely on pre-mapped obstacles or basic sensor data for avoidance.

3.2.5 MISSION FLEXIBILITY

UGV-IP: UGVs with image processing can adapt to a wider range of tasks and environments, making them more versatile.

Standard UGVs: Often designed for specific, limited tasks and environments.

3.2.6 COMPLEXITY AND COST

UGV-IP: Implementing image processing capabilities can increase the complexity and cost of the UGV due to the need for cameras, processing hardware, and software development.

Standard UGVs: Typically simpler and may be more cost-effective for specific applications.

3.2.7 DATA PROCESSING AND BANDWIDTH

UGV-IP: Requires substantial computational power and sufficient bandwidth for data transfer when processing high-resolution images, which can be a challenge in some scenarios. (USA, 2013)

Standard UGVs: May have lower computational and bandwidth requirements.

3.2.8 RELIABILITY AND ROBUSTNESS

UGV-IP: Image processing systems can be sensitive to lighting conditions, weather, and environmental factors, potentially affecting reliability.

Standard UGVs: Simpler sensor systems may be more robust in adverse conditions.

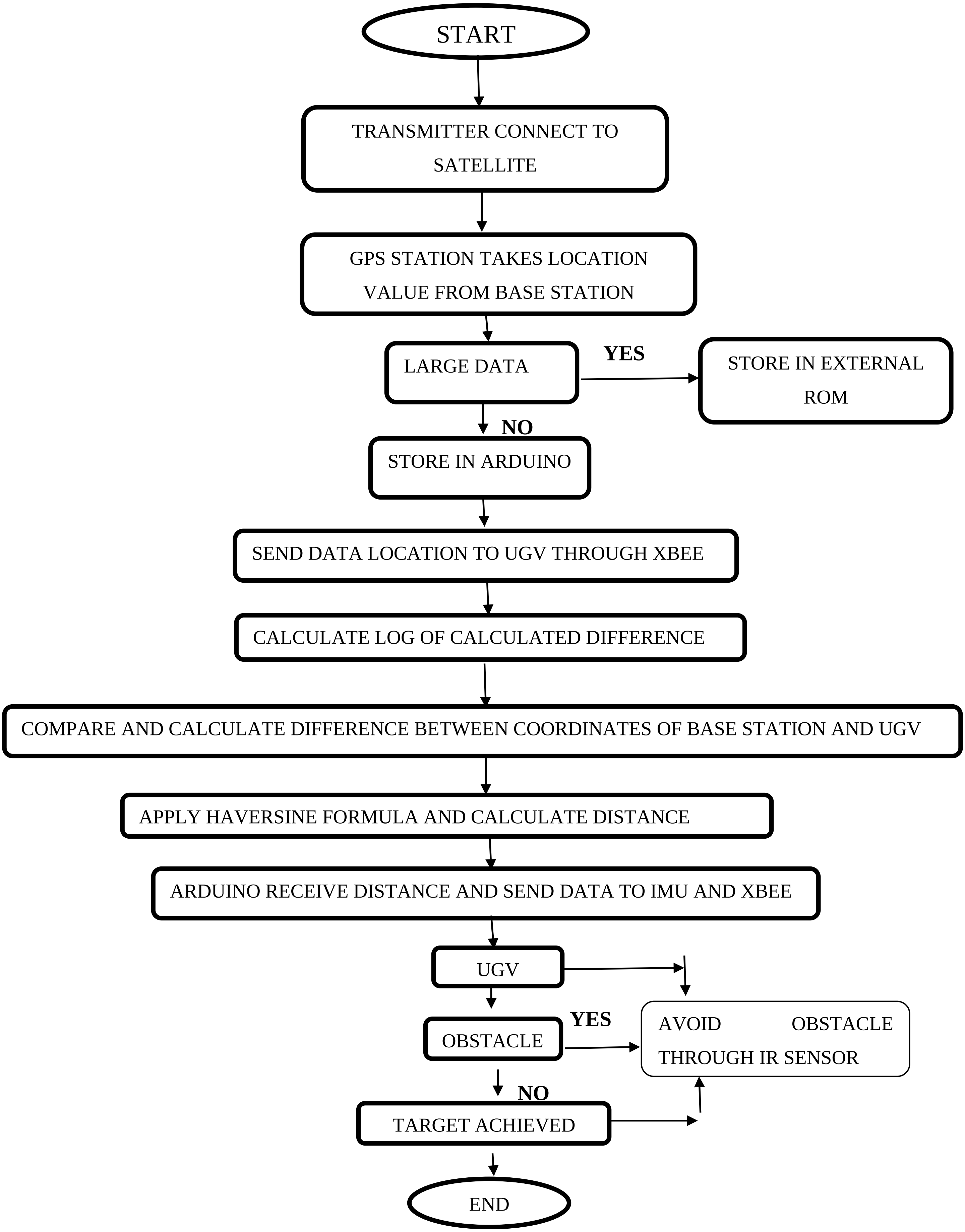
3.2.9 SCALABILITY

UGV-IP: Image processing capabilities can be scaled by upgrading hardware and software, allowing for future enhancements.

Standard UGVs: Limited scalability in terms of perception and processing capabilities.

In conclusion, UGVs with image processing techniques offer significant advantages in terms of perception, autonomy, and adaptability, making them suitable for a wide range of applications. However, they come with increased complexity and cost considerations. The choice between UGV-IP and standard UGVs depends on the specific requirements of the mission and available resources.

3.3 FLOWSHEET DIAGRAM OF OUR UGV METHODOLOGY



3.4 METHODOLOGY EXPLANATION

RASPBERRY PI 3

The Raspberry Pi serves as the central processing unit and may run higher-level tasks, such as navigation algorithms, communication, and high-level decision-making.

METHODOLOGY

- We can install a compatible operating system (e.g., Raspberry Pi OS) on an SD card.
- We can connect peripherals like a keyboard, mouse, and monitor during initial setup.
- We Can Set up SSH for remote access.
- We can install necessary software libraries and packages for GPIO control, wireless communication, and sensor data processing.
- We can write Python or another preferred language code to interface with sensors, control motors, and manage communication.

ARDUINO MEGA

The Arduino Mega acts as a microcontroller responsible for low-level control of hardware components, such as motors and sensors.

METHODOLOGY

- Write code in the Arduino IDE using the Arduino programming language (based on C/C++).
- Interface with servo motors for steering control.
- Interface with motor drivers (L298N or L293D) to control the UGV's movement.
- Connect and read data from sensors like infrared sensors, GPS, and compass modules.
- Implement safety features and emergency stop mechanisms in the code.

SERVO MOTORS

Servo motors are used for precise control of the UGV's steering mechanism.

METHODOLOGY

- Connect servo motors to the Arduino Mega.
- Write code to control the servo motors, allowing precise steering adjustments based on sensor input and navigation algorithms.
- Calibrate the servo motors to ensure accurate and proportional movement.

MOTOR DRIVERS (E.G., L298N OR L293D)

Motor drivers control the UGV's movement by driving its DC motors.

METHODOLOGY

- Connect motor drivers to the Arduino Mega.
- Configure motor driver pins for controlling direction and speed.
- Write code to control the motors, implementing forward, backward, left, and right movements.
- Ensure proper power supply to the motor drivers and motors to avoid damage.

POWER REGULATORS (E.G., VOLTAGE REGULATORS OR BUCK CONVERTERS)

Power regulators provide stable voltage levels to various components to prevent voltage fluctuations and ensure proper operation.

METHODOLOGY

- Select suitable power regulators or buck converters based on component voltage requirements.
- Connect them to the power source and distribute regulated power to the Raspberry Pi, Arduino, and sensors.
- Ensure voltage levels are within the specified ranges for each component.

GPS MODULE (E.G., NEO-6M)

The GPS module provides location data for navigation and mapping.

METHODOLOGY

- Connect the GPS module to the Raspberry Pi (usually via UART).
- Implement code to read GPS data and extract latitude and longitude coordinates.
- Use GPS data to determine the UGV's position and plan routes.

COMPASS MODULE (E.G., HMC5883L)

The compass module provides orientation data for navigation.

METHODOLOGY

- Connect the compass module to the Arduino Mega (typically via I2C or SPI).
- Write code to read compass data and determine the UGV's heading.
- Use compass data for navigation and direction control.

INFRARED SENSORS (FOR OBSTACLE DETECTION)

Infrared sensors help the UGV detect and avoid obstacles.

METHODOLOGY

- Connect the infrared sensors to the Arduino Mega's analog or digital pins.
- Write code to read sensor data and detect obstacles.
- Implement obstacle avoidance algorithms to adjust the UGV's path.

XBEE S2C MODULE AND SHIELD (FOR WIRELESS COMMUNICATION)

XBee modules enable wireless communication between the UGV and base stations or remote controllers.

METHODOLOGY

- Connect the XBee module and shield to the Arduino Mega.
- Write code to establish communication protocols.
- Implement data transmission/reception for remote control or telemetry.

UNMANNED GROUND VEHICLE (UGV) CHASSIS AND BASE STATIONS

The UGV chassis provides the physical structure for mounting components, while base stations may serve as control centers or data collection points.

METHODOLOGY

- Assemble and secure all components on the UGV chassis as previously explained.
- Set up base stations with appropriate hardware and software for monitoring and controlling the UGV remotely.

3.5 2D DRAWINGS OF UNMANNED GROUND VEHICLE

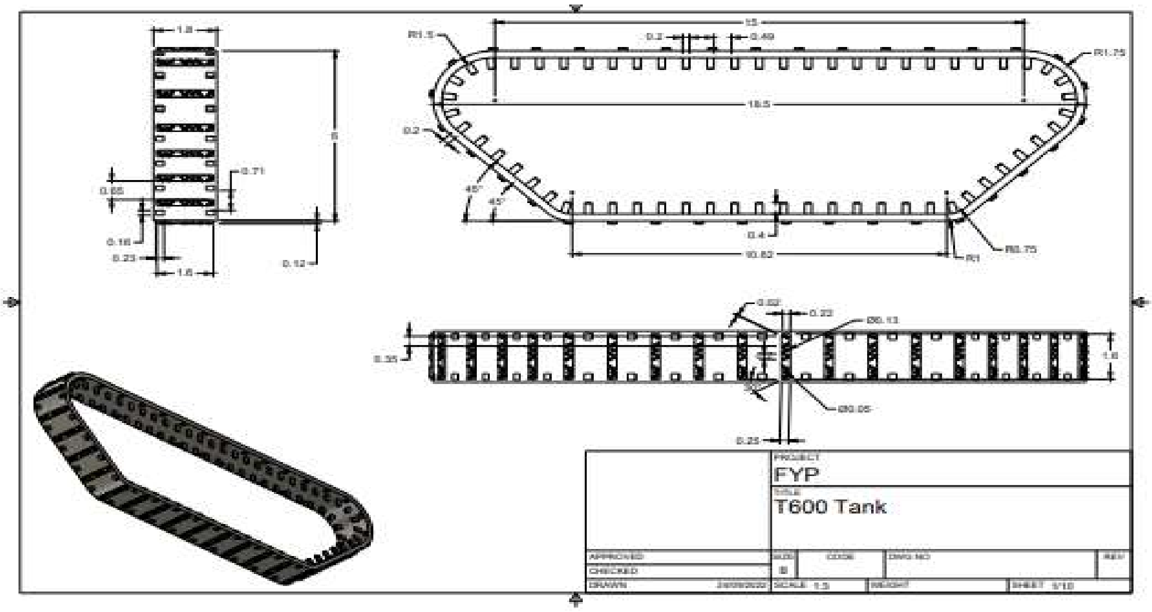


Figure 11 CHAIN DRAWING

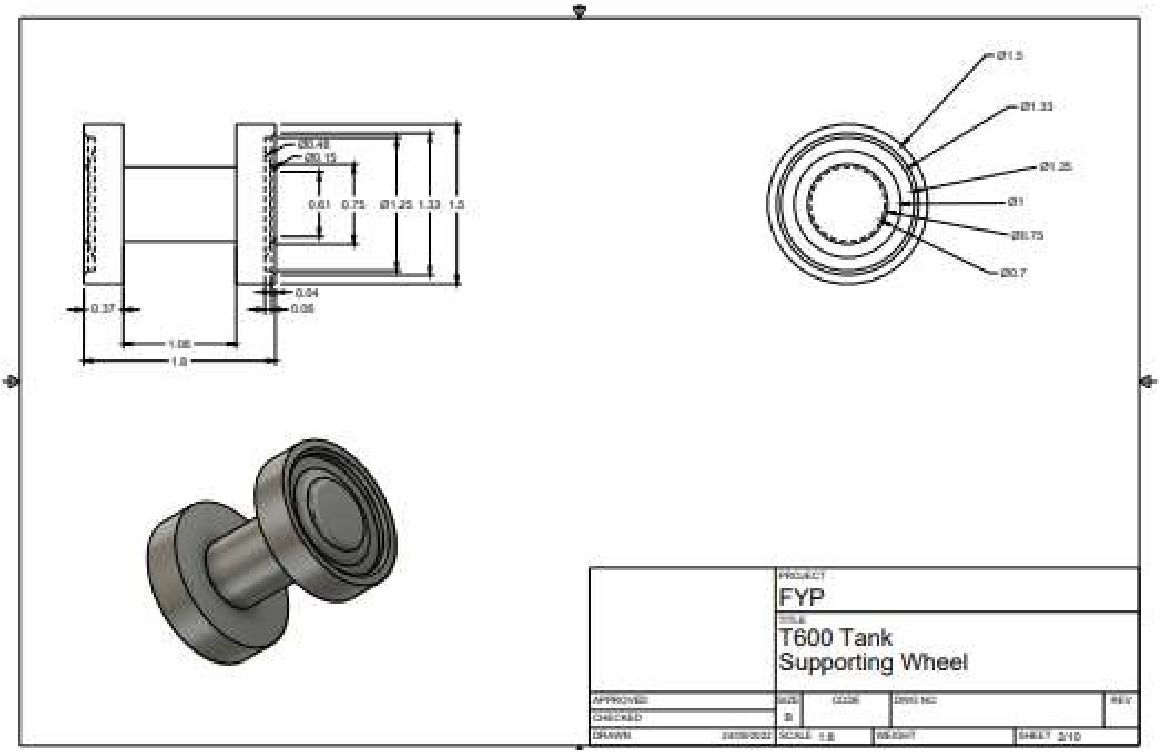


Figure 12 SUPPORTING WHEEL DRAWING

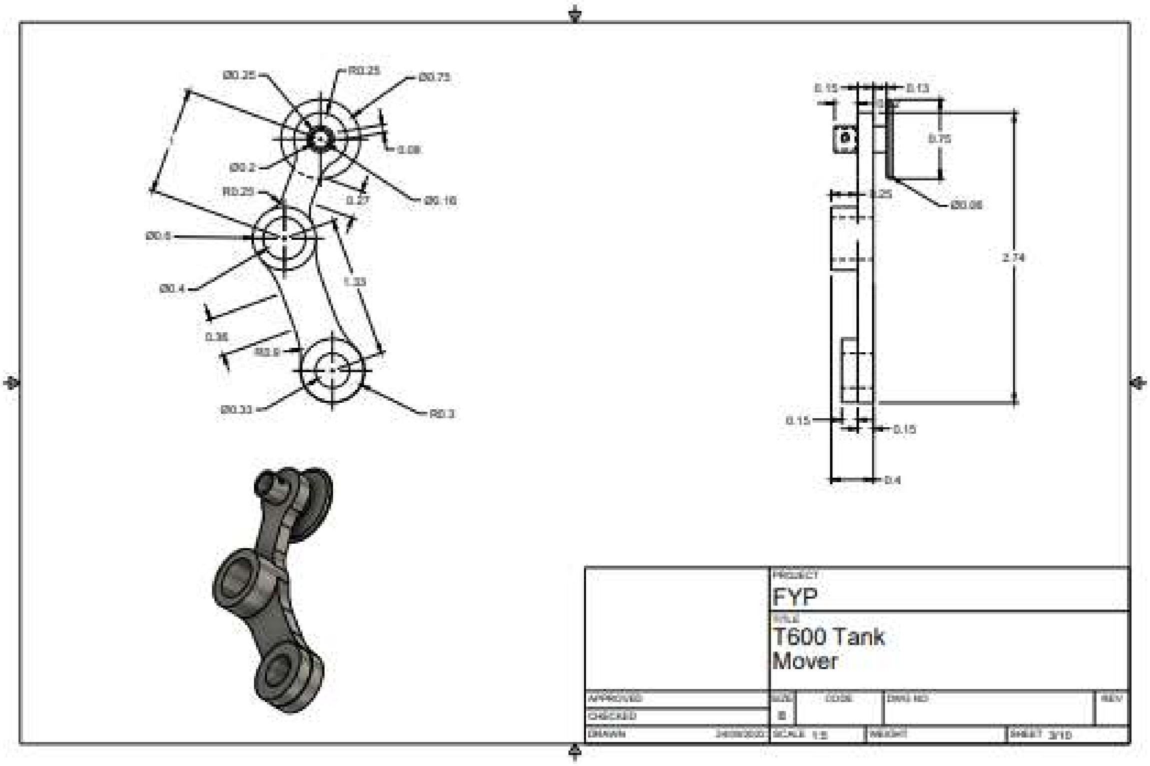


Figure 13 SPROCKET DRAWING

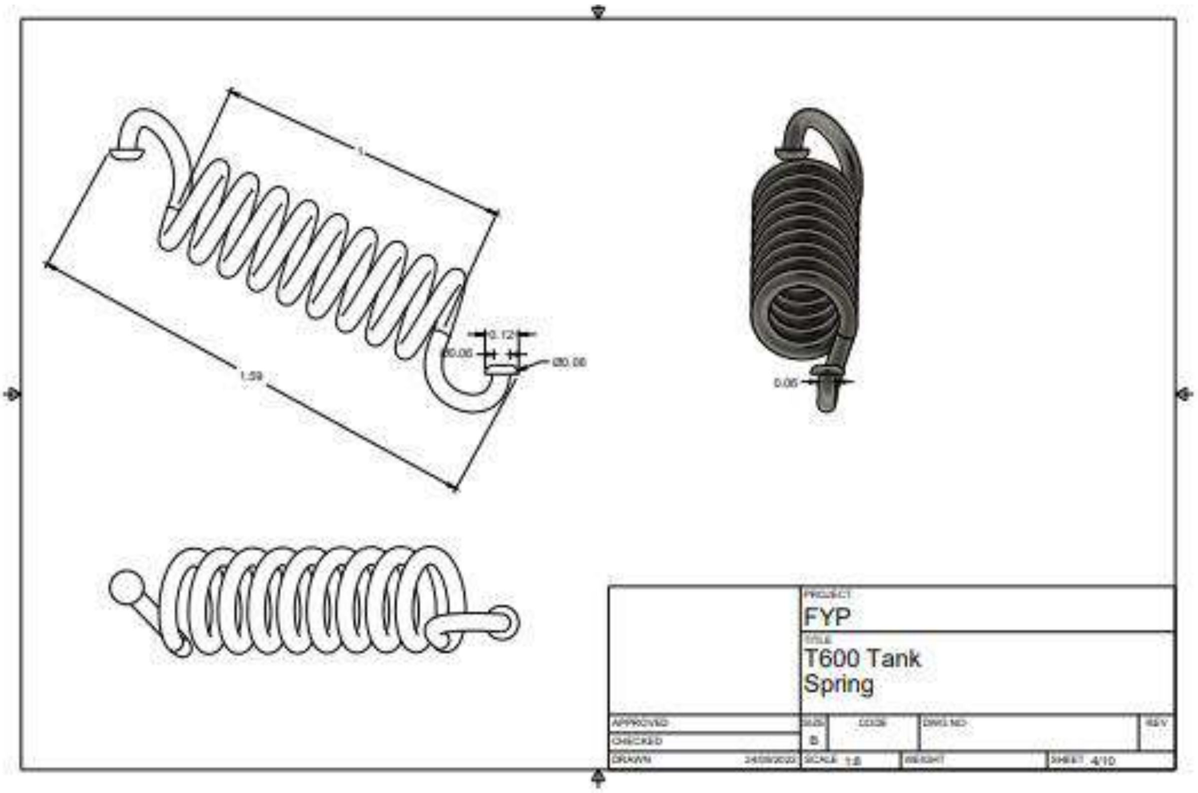


Figure 14 SPRING DRAWING

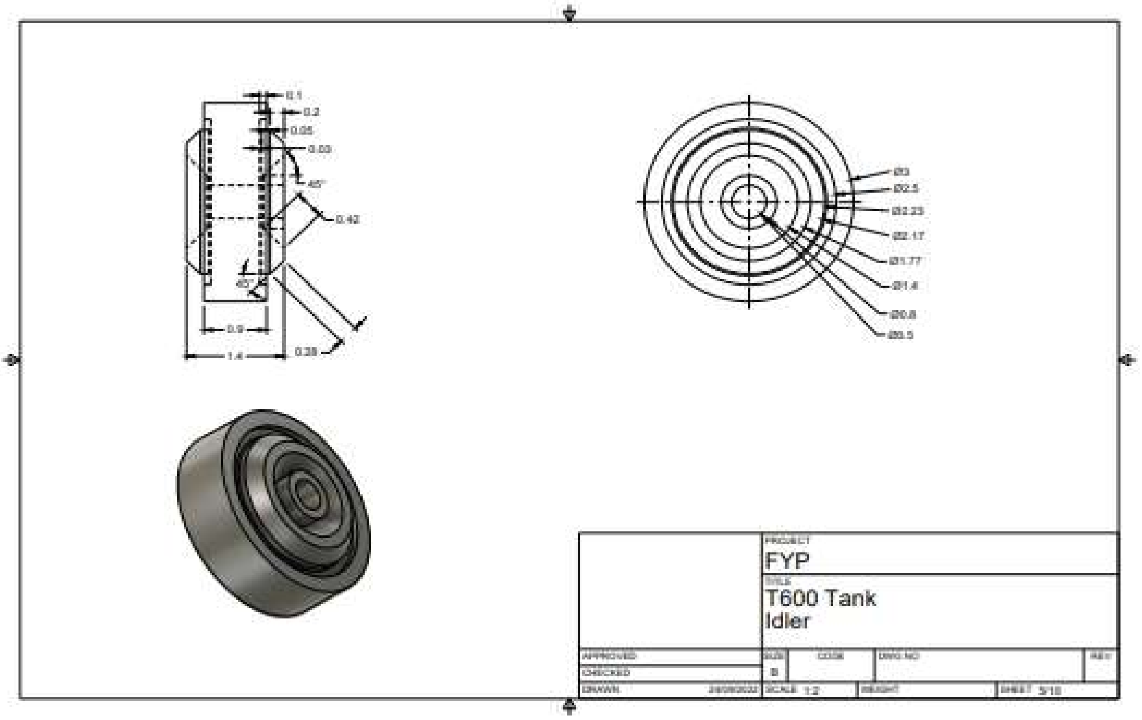


Figure 15 IDLER DRAWING

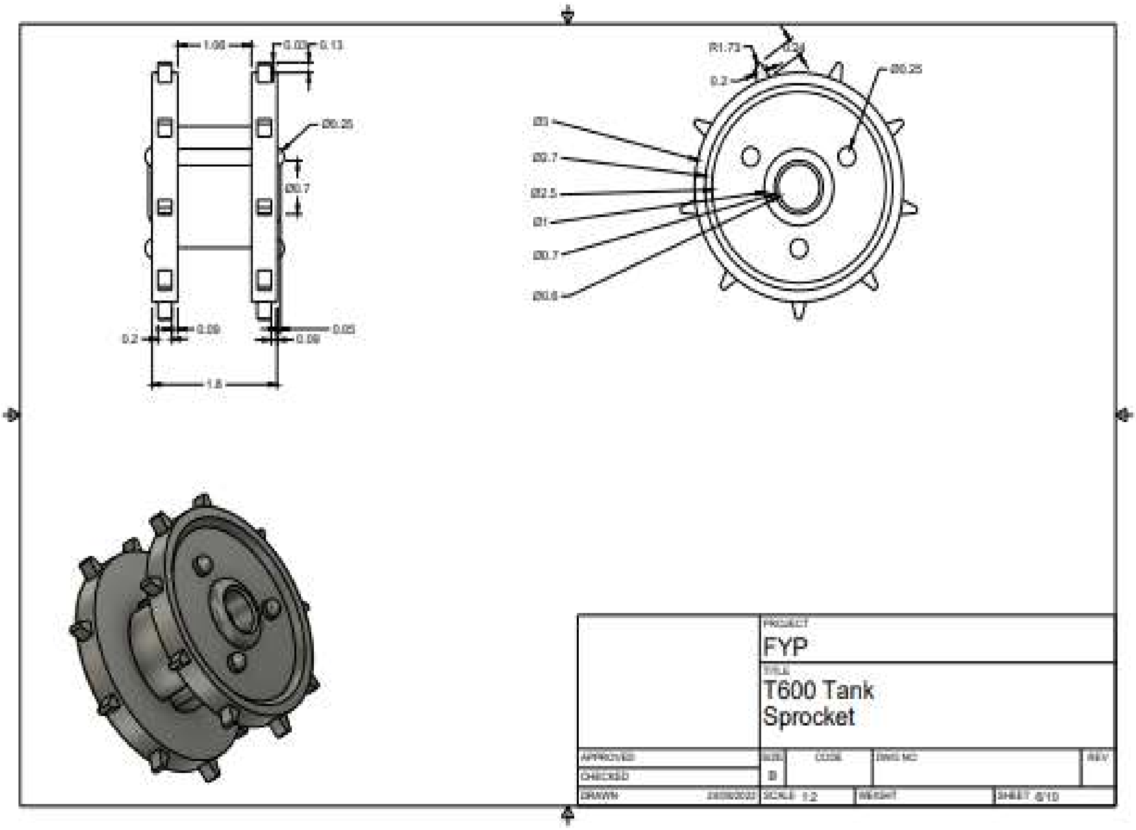


Figure 16 SPROCKET DRAWING

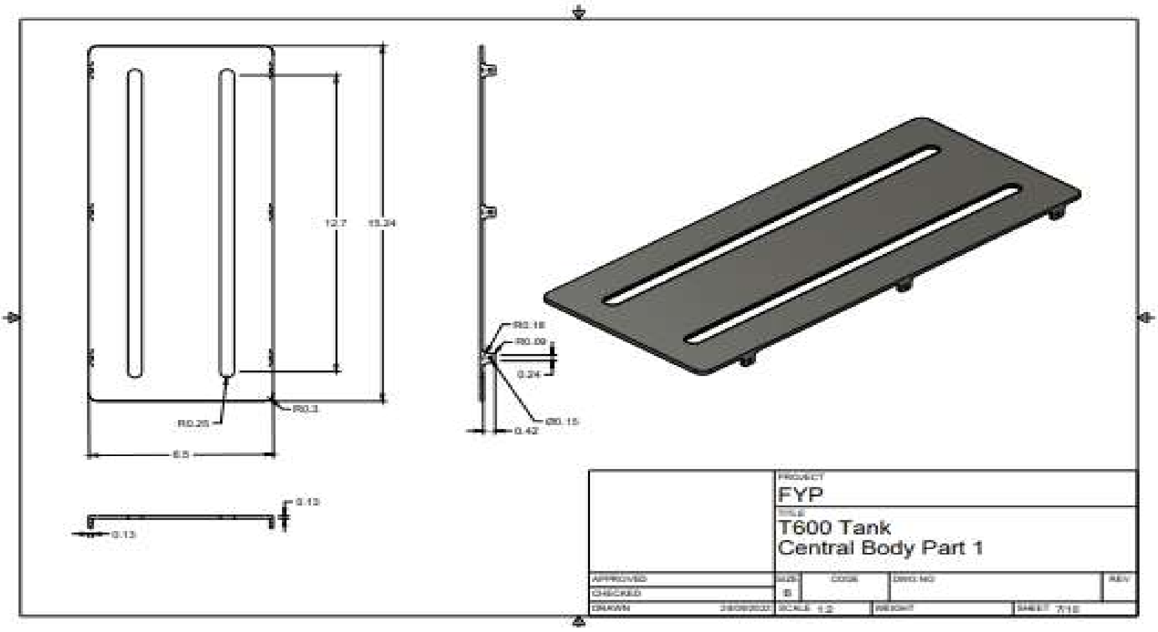


Figure 17 CENTRAL BODY PART 1 DRAWING

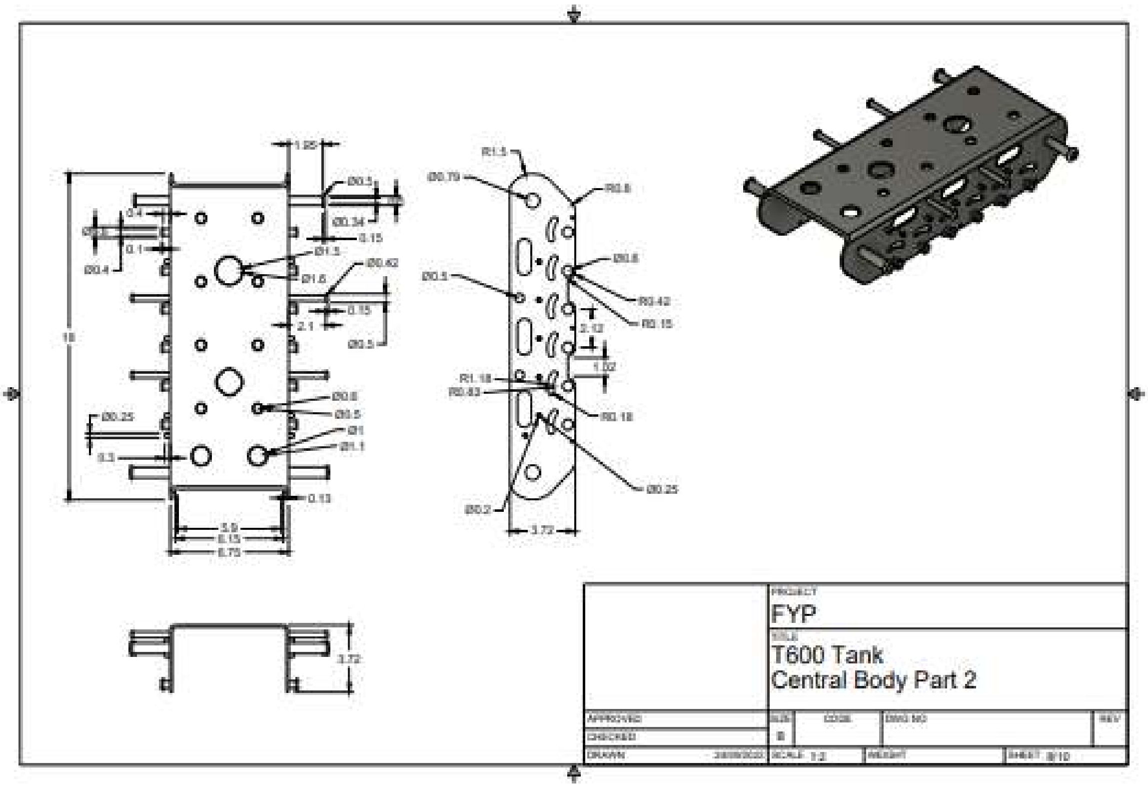


Figure 18 CENTRAL BODY PART 2 DRAWING

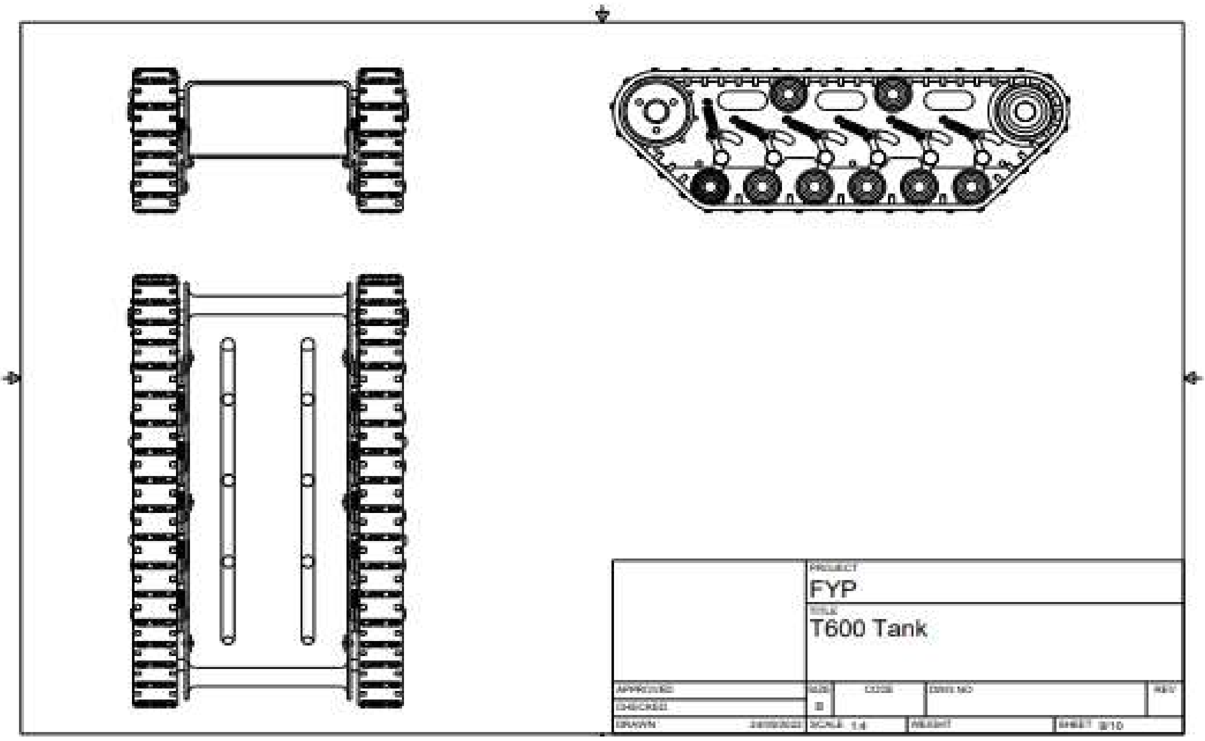


Figure 19 TANK DRAWING 1

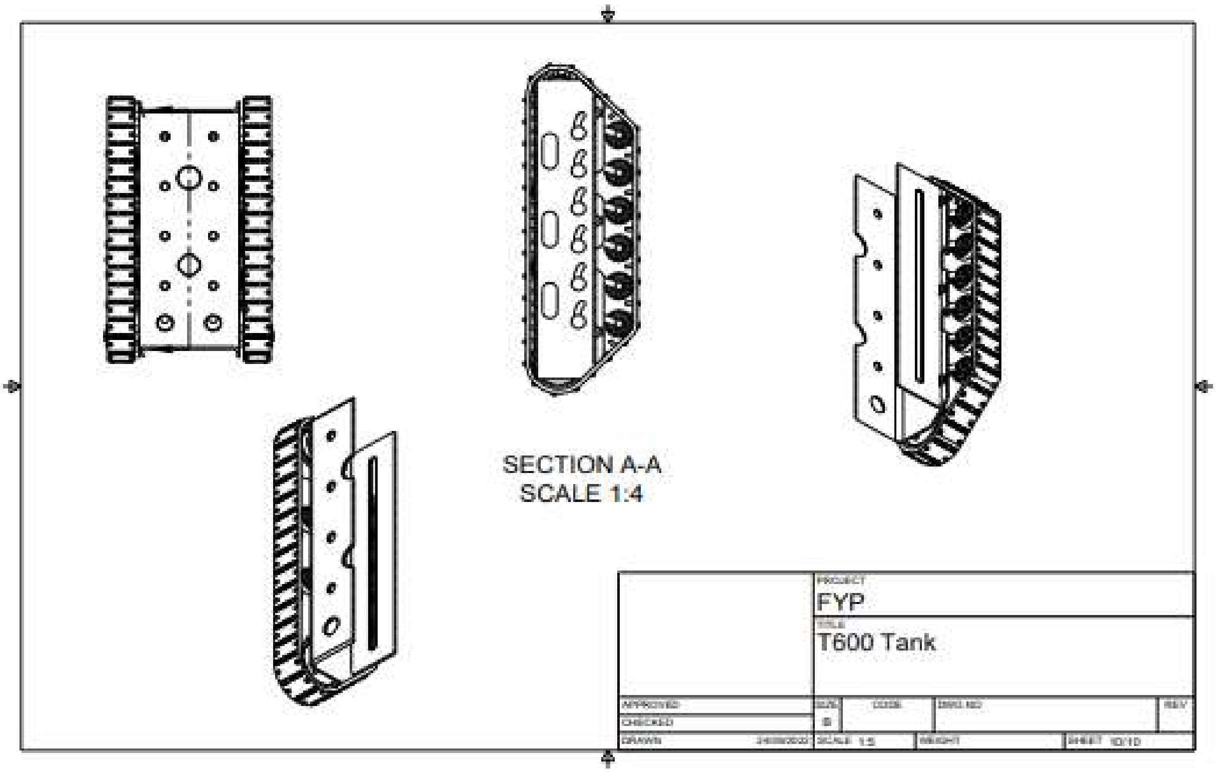


Figure 20 TANK DRAWING 2

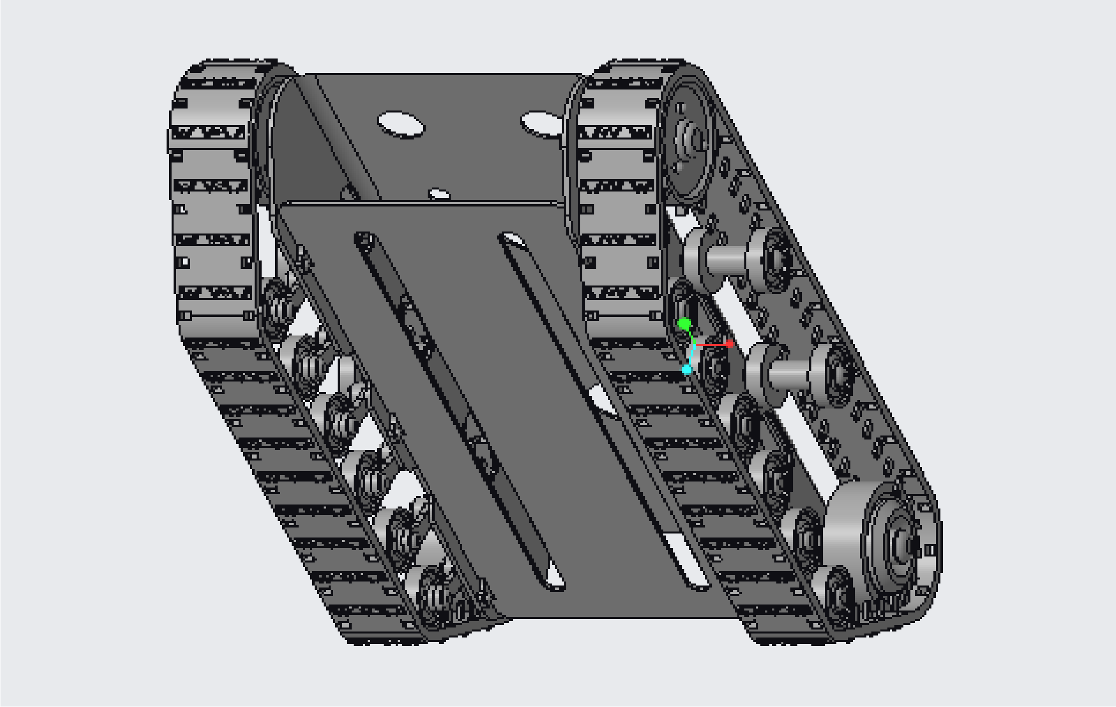


Figure 21 UGV

CHAPTER 4

DEVELOPMENT

4.1 HARDWARE DESCRIPTION

A microprocessor is built into each sensor module. The microcontroller gathers all the data, processes it, and then transmits the appropriate instructions to the motor driver circuit, which in turn runs the UGV's motor for propulsion. The controller will use wireless PC-based connection to send the command to the UGV during manual operation. Two wireless trans-receiver XBee modules are used for wireless communication; one is connected to the PC and the other to the robot.

The UGV's orientation and distance traveled must be known in order to position the UGV on a planar surface. Therefore, distance measuring sensors are used, and a comparison study is done to determine which sensor is best suited for this function. A sensitive GPS module is also utilized to determine the robot's absolute position in each potential location. By using that position as a benchmark, the relative positioning of the UGV may subsequently be predicted in the near future. If a GPS signal is not present, the UGV will use its current location as a reference point and forecast its future positioning. (ors, n.d.)

4.1.1 SERVO MOTORS

An angular or linear actuator known as a servo motor allows for precise control of position, velocity, and acceleration. A suitable motor and a position feedback sensor make up its construction. Applications for servo motors include robotics, CNC machinery, and automated manufacturing. Additionally, a complicated controller is required, typically a module designed specifically for servo motor use. Despite the fact that the word "servo motor" is widely used to describe a motor suitable for use in a closed-loop control system, servo motors are not a specific type of motor. It is essential that you use the servo motor's controller, which was designed specifically for this use and is its most important part. The closed-loop systems used by servo motors to control rotational or linear speed and position incorporate position feedback.

Figure 22 SERVO MOTOR

4.1.2 RASPBERRY PI 3

A quad-core, 1.2GHz Broadcom 64-bit CPU powers the Raspberry Pi 3's single computing board. The board also has the following items:

- HDMI port
- Wireless LAN and onboard low energy Bluetooth modules
- MicroSD card port
- CSI camera port
- DSI display port

4.1.3 4-CHANNEL 5V RELAY MODULE (BRIDGE MOTOR DRIVER)

The 4-Channel 5V Relay Module functions as a bridge motor driver, controlling and distributing power to up to four channels. It efficiently controls the link between the control message and the motor load when powered by a 5V input voltage. (vehicle:) It offers dependable switching along with the management of motor-driven devices due to its electromechanical relay type. The module receives logic-level control signals through GPIO pins and provides TTL-compatible trigger signal voltage, making it simple to integrate with microcontrollers and digital circuitry. It is versatile, with screw terminals or header pins for easy connection and the ability to handle particular maximum load currents as well as voltages per channel (values set by the individual module variation). Specifications are given in Table.

4.1.4 REES52 12V TO 5V 5A CONVERTER, BUCK REGULATOR DC 9V-36V STEP DOWN TO DC 5V

The REES52 12V to 5V 5A Converter is a buck regulator that steps down a DC input voltage that ranges from 9V to 36V to a steady 5V output. This converter is designed to manage a maximum current output that is 5A, making it suited for applications requiring a reliable power supply. Its high efficiency and minimal heat production enable dependable and steady power conversion for a wide range of electronic devices and systems. Overcurrent security, over-temperature safeguarding, and short-circuit security are included in the converter, assuring the safety and endurance of connected devices. Because of its small size and simple screw connections, it is ideal for installation and incorporation into an extensive variety of applications and tasks. Specifications are given in Table.

4.1.5 UBLOX NEO-6M GPS MODULE

The Ublox Neo-6m GPS Module is a small and dependable GPS receiver. It makes use of powerful Ublox 6 positioning technology to give precise and quick positional information. It is useful for applications such as navigation, tracking, and timing in a variety of sectors because to its high sensitivity and low power consumption. Specifications are given in Table.

Figure 23 UBLOX NEO-6M

4.1.6 GY-271 3-AXIS MAGNETIC ELECTRONIC COMPASS MODULE

The GY-271 3-Axis Magnetic Electronic Compass Module is a small and multifunctional magnetic field measurement module. It has a tri-axis magnetometer sensor that allows it to detect and analyze magnetic fields in three dimensions. It is excellent for applications like as navigation, orientation, and heading determination in robots, drones, and other electronic devices because to its great sensitivity and precision. (King, 2014)Specifications are given in Table.

Figure 24 GY-271 3-AXIS

4.1.7 MPU 6500 SIX-AXIS (GYRO + ACCELEROMETER)

The MPU 6500 Six-Axis module combines a gyroscope and accelerometer into a single compact package. It provides precise measurement and sensing capabilities for both rotational and linear motion. With its six-axis functionality, it enables accurate tracking of orientation, motion, and gesture recognition in applications such as robotics, drones, and virtual reality systems. Specifications are given in Table.

4.1.8 SHARP GP2Y0A41SK 4-30CM INFRARED PROXIMITY SENSOR SHORT RANGE

The Sharp GP2Y0A41SK Infrared Proximity Sensor is a short-range sensor that detects things within a 4-30cm range. It precisely measures the closeness of objects by using infrared light. It is extensively utilized in applications such as obstacle recognition, robotics, and automation systems wherever close-range object identification are required because to its small design and reliable efficiency. Specifications are given in Table.

4.1.9 ARDUINO MEGA ATMEGA2560

The ATmega2560-based Arduino Mega Atmega2560 is a powerful microcontroller board. It has a wide range of digital and analog input/output pins, giving it versatility for a variety of tasks. It is appropriate for complicated applications needing several sensors, actuators, and interfaces for interaction due to its massive flash memory and extensive I/O capabilities. Specifications are given in Table.

Figure 25 ARDUINO MEGA

4.1.10 SG90 SERVO

The SG90 servo motor is a tiny and lightweight motor that is widely utilized in robotics and remote-control applications. It has a 0 to 180-degree range of motion and provides accurate control and placement. The SG90 servo is perfect for projects that demand smooth and precise motions, such as robotic arms, RC cars, and tiny systems, due to its compact shape and low power use. Specifications are given in Table.

4.1.11 CONVERTER WITH 7 SEGMENT DISPLAY LM2596

The LM2596 Converter with 7 Segment Display is a flexible voltage regulator module that includes a voltage converter and a 7-segment display in a single package. It enables efficient and customizable step-down voltage conversion, resulting in a steady output voltage for a wide range of electrical equipment. The inbuilt 7-segment display provides visual feedback of output voltage, making it easy to monitor and change voltage levels in applications that operate immediately. Specifications are given in Table.

4.1.12 XBEE S2C MODULE AND SHIELD

The XBee S2C module and shield combo offers a dependable wireless communication solution for a variety of applications. The XBee S2C module provides strong and low-power Zigbee mesh networking features, allowing for smooth device connection. The shield, which is designed to interact with the XBee S2C module, allows for simple integration into microcontroller-based applications while providing wireless communication and data transfer over a long distance. Specifications are given in Table.

Figure 26 XBEE S2C

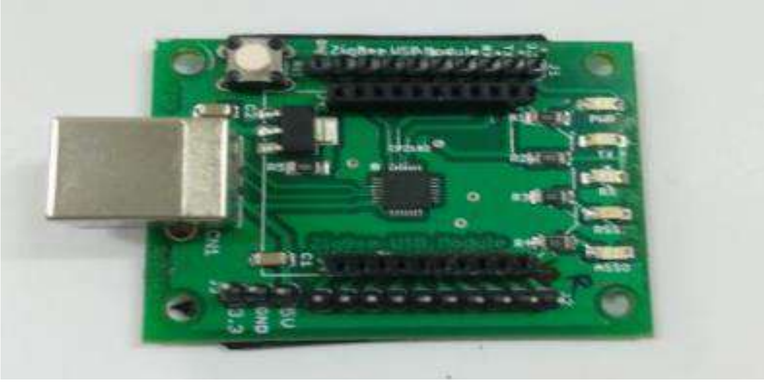


Figure 27 Raspberry pi 3

4.2 EXPLANATION OF OBSTACLE AVOIDANCE CODE

LIBRARIES AND GLOBAL VARIABLES:

The code starts by including necessary libraries such as TinyGPS, Servo, Wire, and HMC5883L. It declares global variables for sensor data, GPS coordinates, motor control, and other parameters. (Wikipedia, accessed on Oct – 2014.)

SETUP FUNCTION:

In the setup function, the code initializes serial communication, configures motor control pins, and sets up the HMC5883L magnetometer sensor.

LOOP FUNCTION

OBSTACLE AVOIDANCE:

The Obstacle function is called in the loop to handle obstacle avoidance. It uses an ultrasonic sensor (analog input from pin A1) to detect obstacles and adjust the robot's direction accordingly.

GPS DATA RETRIEVAL:

The gps.get_position function retrieves latitude and longitude data from the GPS module, and the getGPS function is called to format and calculate GPS data.

HEADING CALCULATION:

The magnetometer sensor is used to calculate the heading (direction) of the robot relative to magnetic north. The heading is adjusted for declination and stored in the bearing variable.

MOTOR CONTROL BASED ON HEADING:

Depending on the difference between the calculated heading and the desired heading (bearingp and bearings), the code adjusts the motor control to steer the robot in the correct direction.

SERIAL COMMUNICATION:

The code also listens for commands received via serial communication, likely for remote control of the robot.

FUNCTIONS

GETGPS:

This function waits for GPS data for up to 200 milliseconds and then formats and stores the latitude and longitude data.

FEEDGPS: This function reads GPS data from Serial3 and passes it to the gps object for processing.

GPSDUMP:

This function further processes GPS data, calculates distance and bearing between two GPS coordinates, and stores the results in variables.

LEFT, RIGHT, FORWARD, STOP:

These functions control the robot's motor movements to move it forward, stop, turn left, or turn right.

MICROSECONDSTOINCHES, MICROSECONDSTOCENTIMETERS:

These functions convert the duration of a pulse received from an ultrasonic sensor into inches or centimeters, respectively. (Siciliano, 2008)

4.2.1 SUMMARY OF THE CODE

The code is designed for a robotic platform that uses GPS and a magnetometer to navigate based on a desired heading (bearing) while avoiding obstacles using an ultrasonic sensor. It listens for commands via serial communication and adjusts the robot's motor control to execute movements and respond to input. The code appears to be a combination of sensor interfacing, control logic, and obstacle avoidance routines for a mobile robot.

4.3 EXPLANATION OF IMAGE PROCESSING CODE

1. Import necessary libraries, including numpy and cv2 (OpenCV). Load the pre-trained Haar Cascade classifier for face detection.
2. Open the default camera (usually the built-in webcam) using cv2.VideoCapture.
3. Set the desired width and height for the video frame using cap.set().
4. Start an infinite loop for capturing and processing frames from the webcam.
5. Read a frame from the camera using cap.read().
6. Convert the frame to grayscale to perform face detection, as it's more efficient.
7. Detect faces in the grayscale frame using faceCascade.detectMultiScale. This function returns the coordinates and dimensions of detected faces.
8. Draw rectangles around the detected faces using cv2.rectangle.
9. Display the frame with face detection using cv2.imshow.
10. Check for the 'ESC' key press (k == 27) to exit the loop.
11. Release the camera and close the OpenCV window when exiting the loop.

4.3.1 SUMMARY OF THE CODE

The code uses OpenCV to perform real-time face detection from a webcam feed. It loads a pre-trained Haar Cascade classifier for face detection, captures video frames from the webcam, converts them to grayscale, detects faces, and draws rectangles around the detected faces. The code continuously displays the video feed with face detection and exits when the 'ESC' key is pressed. This script provides a basic example of how to perform real-time face detection using OpenCV in Python.

CHAPTER 5

5.1 CONCLUSION

In conclusion, the integration of image processing techniques into Unmanned Ground Vehicles (UGV-IP) represents a significant advancement in the field of robotics and autonomous systems. This technology brings a host of advantages compared to standard UGVs, but it also comes with its own set of challenges and considerations.

Firstly, UGVs equipped with image processing capabilities excel in perception and sensing. They can leverage cameras and sensors to gain a more detailed and versatile understanding of their surroundings, enabling them to identify objects, analyze data in real-time, and adapt to dynamic environments. In contrast, standard UGVs often rely on simpler sensors like LiDAR or ultrasonic sensors, which may limit their perception capabilities.

UGV-IP systems also greatly enhance autonomy. With the ability to process visual data, these vehicles can perform advanced tasks such as object recognition, obstacle avoidance, and path planning based on real-time visual input. This leads to a higher level of autonomy compared to standard UGVs, which may depend more on manual control or basic sensor data for navigation.

Object detection and recognition are crucial in various applications, ranging from target identification in military operations to mapping and navigation in complex environments. UGV-IP excels in this regard by being capable of detecting and classifying objects effectively. Standard UGVs may struggle to identify objects beyond basic obstacles.

Moreover, UGV-IP systems can employ image processing for simultaneous localization and mapping (SLAM), allowing them to create detailed and up-to-date maps of their environment. This capability is invaluable for missions requiring precise navigation and situational awareness. Standard UGVs typically have less advanced mapping capabilities.

One of the most significant advantages of UGV-IP is real-time obstacle avoidance, which enhances safety in dynamic environments. These vehicles can quickly detect and respond to obstacles, reducing the risk of collisions. Standard UGVs may rely on pre-mapped obstacles or basic sensor data for avoidance, which may not be as effective in dynamic scenarios.

UGV-IP systems also offer unparalleled mission flexibility. They can adapt to a wider range of tasks and environments, making them versatile tools for various applications. In contrast, standard UGVs are often designed for specific, limited tasks and environments, limiting their adaptability. (NYWF., accessed on Oct-2014.)(Bonne, 2018)

However, it's essential to consider the trade-offs. Implementing image processing capabilities increases the complexity and cost of UGV-IP systems due to the need for cameras, processing hardware, and software development. Additionally, UGV-IP systems require substantial computational power and sufficient bandwidth for processing high-resolution images, which may pose challenges in some scenarios. Standard UGVs are typically simpler and may be more cost-effective for specific applications.

Furthermore, UGV-IP systems can be sensitive to lighting conditions, weather, and environmental factors, potentially affecting their reliability. In contrast, standard UGVs with simpler sensor systems may be more robust in adverse conditions.

Lastly, in terms of scalability, UGV-IP systems have an advantage as their image processing capabilities can be scaled by upgrading hardware and software. This allows for future enhancements and adaptability to evolving mission requirements. Standard UGVs have limited scalability in terms of perception and processing capabilities. (NYWF., accessed on Oct-2014.)(B.J. Yoon)

In summary, UGVs with image processing techniques offer a significant leap in capability, autonomy, and adaptability compared to standard UGVs. These systems are well-suited for a wide range of applications, particularly those requiring advanced perception and real-time decision-making. However, the decision to adopt UGV-IP technology should be made carefully, considering factors such as complexity, cost, reliability, and the specific mission requirements. Ultimately, the choice between UGV-IP and standard UGVs should align with the objectives and resources of the mission at hand.

6 References

1. E. Rijanto, e. a. (2013). "Positioning Control for the Mobile Robot LIPI Articulated Robot Arm Based on PD Control Approach,". *International Journal of Applied Engineering Research, vol. 8, pp. 423-433*.
2. A.A. Nayak, D. P. (2012.). Robotics navigation in the presence of static and dynamic obstacles, B.Tech. Thesis, NIT Rourkela,.
3. Anderson, J. M. (2019). *Designing autonomous vehicles for future mobility. Science,.*
4. B. Hendrik, e. a. (2013.). "Penerapan Mini Robot dengan Konsep Omni Directional untuk mendapatkan Informasi pada Daerah Berbahaya," . *Teknologi Informasi dan Pendidikan, vol. 6*.
5. B.J. Yoon, M. P. (n.d.). UGV(Unmanned Ground Vehicle) Navigation Method using GPS and Compass, SICE-ICASE International Joint Conference,.
6. Bonne, A. &. (2018). Unmanned ground vehicle navigation in outdoor environments: A review. Journal of Field Robotics, .
7. Borenstein, J. &. (1991). *The vector field histogram—fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation.*
8. C. R. Terwelp. (2003). Virginia Polytechnic Institute and State University Blacksburg, Virginia, Master Thesis, May 28 2003. *"Remote Control of Hydraulic Equipment for Unexploded Ordnance Remediation "* .
9. D. S. Samosir. (n.d.). "Komunikasi Digital Menggunakan Gelombang Radio FM berbasis Mikrokontroler AT89S51," . *Universitas Sumatera Utara, Medan, Tugas Akhir2007.*
10. D.W. Gage, U. H. (1995). A brief history of unmanned ground vehicle (UGV) development efforts. Unmanned Systems, 13:9–32,.
11. Douglas W. Gage, “. b. (1995.). Special Issue on Unmanned Ground Vehicles, Unmanned Systems Magazine, .
12. Dudek, G. &. (2010). Computational principles of mobile robotics. Cambridge University Press.
13. H. Yu, e. a. (2010). "Review of modelling and remote control for excavators ". *International Journal Advanced Mechatronic Systems, vol. 2, pp. 68-80,.*
14. Howe, R. D. (1999). *Robotics for surgery. Annual Review of Biomedical Engineering,.*
15. J.-S. Valois, e. a. (presented at the Unmanned System Technology X, 2008). "Remote Operation of the Black Knight Unmanned Ground Combat Vehicle,".
16. Jennifer Carlson. (2004.). “Analysis of How Mobile Robots Fail in the Field”,. *Master thesis, Department of Computer Science and Engineering, University of South Florida, .*

17. King, H. K. (2014). Retrieved from http://www.hobbyking.com/hobbyking/store/__8337__hobby_king_2_4ghz_4ch_tx_rx_v2_mode_1_.html
18. Lin, P. &. (2017). *Robot ethics 2.0: From autonomous cars to artificial intelligence. Oxford University Press.*
19. M. Peckham, T. a. (2014, October). Retrieved from http://techland.time.com/2012/05/08/googles driverless-cars-now-officially-licensed-in-nevada/
20. NYWF. (accessed on Oct-2014.). *The 1939 New York World's fair.* . Retrieved from http://www.1939nyworldsfair.com/worlds fair/dex.htm
21. O. Mahendra. (2012,). "Metoda Pembacaan Lebar Pulsa Sinyal PWM Motor Servo dengan Timer 16 bit pada Mikrokontroler AVR,".
22. ors. (n.d.). Retrieved from https://ieeexplore.ieee.org/document/5723511/auth
23. Ponce., D. F. (Nov – 2011.). *Computer vision: a modern approach. Pearson; 2 edition, .*
24. R. C. Detmer, J. &. (2014.). *Introduction to 80x86 Assembly Language and Computer Architecture, Third ed.: .*
25. Siciliano, B. &. (2008). Springer Handbook of Robotics. Springer.
26. Thrun, S. B. (2005). Probabilistic Robotics. MIT Press.
27. USA, D. o. (2013). DoD, Unmanned Systems Integrated Roadmap FY2011-2036,.
28. vehicle:, A. n.-w. (n.d.). design and implementation (uvic.ca).
29. Wikipedia. (accessed on Oct – 2014.). *Wikipedia article on Unmanned Ground Vehicle, .* Retrieved from http://en.wikipedia.org/wiki/Unman ned_ground_vehicle

5 APPENDIXES

TABLE 1 4-CHANNEL 5V RELAY MODULE (BRIDGE MOTOR DRIVER)

Specifications	**Description**
Channels	04
Input Voltage	5V
Relay Type	Electromechanical
Control Interface	GPIO pins
Control Signal	Logic level (5V)
Operating Temperature	-20°C to +70°C

TABLE 2 REES52 12V TO 5V 5A CONVERTER, BUCK REGULATOR DC 9V-36V STEP

Specifications	**Description**
Input Voltage Range	DC 9V-36V
Output Voltage	DC 5V
Maximum Output Current	5A
Conversion Type	Buck (Step-Down)
Protection Features	Overcurrent, Overheating, Short Circuit
Mounting	PCB Mount or Standalone

TABLE 3 UBLOX NEO-6M GPS MODULE

Specifications	**Description**
Receiver Type	GPS (Global Positioning System)
Number of Channels	50
Positioning Accuracy	<2.5 meters (under optimal conditions)
Communication Interface	Serial (UART)
Operating Voltage	3.3V
Antenna	External active antenna required

TABLE 4 GY-271 3-AXIS MAGNETIC ELECTRONIC COMPASS MODULE

Specifications	**Description**
Measuring Range	±1.3 to ±8.1 Gauss (G)
Resolution	0.92 milligauss (mg.)
Output Data Rate	Up to 75 Hz
Sensitivity	1090 LSB/Gauss
Communication Protocol	I2C (Inter-Integrated Circuit)
Operating Voltage	3.3V or 5V (depending on module variant)

TABLE 5 MPU 6500 SIX-AXIS (GYRO + ACCELEROMETER)

Specifications	**Description**
Gyroscope Range	±250, ±500, ±1000, or ±2000 degrees per second
Accelerometer Range	±2g, ±4g, ±8g, or ±16g
Communication Interface	I2C (Inter-Integrated Circuit)
Digital Resolution	16 bits
Operating Voltage	**2.375V to 3.46V**
Operating Current	4.1 **mA (typical)**

TABLE 6 SHARP GP2Y0A41SK 4-30CM INFRARED PROXIMITY SENSOR SHORT

Specifications	**Description**
Sensing Distance	4 cm to 30 cm
Output Type	Analog voltage output
Supply Voltage	4.5V to 5.5V (Typical: 5V)
Average Current Consumption	**33 mA**
Operating Temperature Range	**-10°C to +60°C**

TABLE 7 ARDUINO MEGA ATMEGA2560

Specifications	**Description**
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20mA

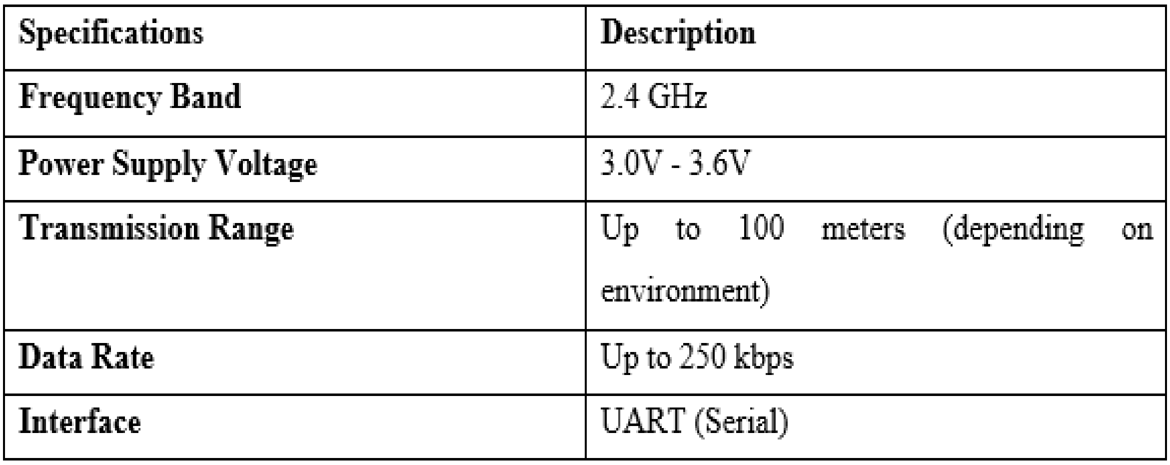
TABLE 8 SG90 SERVO

Specifications	**Description**
Operating Voltage	4.8V - 6.0V
Stall Torque	1.8 kg/cm (4.8V), 2.2 kg/cm (6.0V)
Operating Speed	0.12 sec/60° (4.8V), 0.10 sec/60° (6.0V)
Control Signal	PWM (Pulse Width Modulation)
Control Angle	180°
Dimensions	23mm x 12.2mm x 29mm

TABLE 9 CONVERTER WITH 7 SEGMENT DISPLAY LM2596

Specifications	**Description**
Display Model	LM2596
Input Voltage	4-35V.
Adjustable Output Voltage	1.25-30V
Output Current	3A
PCB Size	68x35mm
Control Interface	Potentiometer or Buttons

TABLE 10 XBEE S2C MODULE AND SHIELD



Specifications	Description
Frequency Band	2.4 GHz
Power Supply Voltage	3.0V - 3.6V
Transmission Range	Up to 100 meters (depending on environment)
Data Rate	Up to 250 kbps
Interface	UART (Serial)

5.1SOFTWARE DESCRIPTION

7.1.1 UGV OBSTACLE AVOIDANCE CODE

AUTONOMOUS NAVIGATION THROUGH SHORTEST PATH (PATH PLANNING) WITH CM LEVEL ACCURACY WITH OBSTACLE DETECTION RECOGNITION AND REAL TIME DATA CORRECTION TRANSMISSION THROUGH DIFFERENTIAL GLOBAL POSITIONING

```
#include <TinyGPS.h>
unsigned long fix_age;
TinyGPS gps;
void gpsdump(TinyGPS &gps);
bool feedgps();
void getGPS();
long lat, lon;
double LAT,LON;
void setup(){
Serial3.begin(9600);
Serial.begin(9600);
}
 void loop(){
long lat, lon;
unsigned long fix_age, time, date, speed, course;
unsigned long chars;
```

```
unsigned short sentences, failed_checksum;
// retrieves +/- lat/long in 100000ths of a degree
 gps.get_position(&lat, &lon, &fix_age);
getGPS();
Serial.println(LON/1000000,7);
Serial.println(LAT/1000000,7);
}
void getGPS(){
bool newdata = false;
unsigned long start = millis();
while (millis() - start < 200)
{
 if (feedgps ()){
 newdata = true;
 }
}
if (newdata)
{
 gpsdump(gps);
}
}
 bool feedgps(){
while (Serial3.available())
{ if (gps.encode(Serial3.read()))
 return true;
}
return 0;
 }
void gpsdump(TinyGPS &gps)
 {
//byte month, day, hour, minute, second, hundredths;
 gps.get_position(&lat, &lon);
 LAT = lat;
 LON = lon;
```

```
 struct dataStruct
 {
double latitude;
double longitude;
unsigned long date;
unsigned long time;
 }
 gpsData;
 Serial.begin(115200);
 ss.begin(9600); while (ss.available() > 0)
{
 if (gps.encode(ss.read())){
 getInfo();
 printResults();
}
void getInfo(){
 if (gps.location.isValid()){
 gpsData.latitude = gps.location.lat();
 gpsData.longitude = gps.location.lng();
}
 else
 {
Serial.println("Invalid location");
 }
if (gps.date.isValid()){
 gpsData.date = gps.date.value();
}
else
{
 Serial.println("Invalid date");
 }
If (gps.time.isValid()){
 gpsData.time = gps.time.value();
 }
```

```
 else
{
Serial.println("Invalid time");
}
}
void printResults(){
Serial.print("Location: ");
Serial.print(gpsData.latitude, 6); Serial.print(", ");
 Serial.print(gpsData.longitude, 6);
 Serial.print(" Date: ");
Serial.print(gpsData.date);
 Serial.print(" Time: ");
 Serial.print(gpsData.time);
 Serial.println();
}
{
 feedgps();
}
}
#include <Servo.h>
#include<TinyGPS.h>
 #include<math.h>
 #include<Wire.h>
 #include <HMC5883L.h>
Servo myservo;
int sensorValue=0;
 int pulse,val;
int LF=9;
 int RB=8;
int LB=10;
 int RF=11;
double bearingp;
double bearings;
double bearing;
```

```
 HMC5883L compass;
 int error = 0;
int head;
TinyGPS gps;
 void gpsdump(TinyGPS &gps);
bool feedgps();
 void getGPS();
long lat, lon;
double LAT,LON;
 double LAT1D, LON1D;
 double LAT2D, LON2D;
double LAT1R, LON1R;
 double LAT2R, LON2R;
double DLAT, DLON;
unsigned long fix_age, time, date, speed, course;
 unsigned long chars;
 unsigned short sentences, failed_checksum;
 int var=0;
String inString1 = "";
double x[2];
int y=0;
void setup(){
 myservo.attach(7);
 Serial.begin(9600);
 Serial1.begin(9600);
Serial3.begin(9600);
 pinMode(LF, OUTPUT);
 pinMode(LB, OUTPUT);
 pinMode(RF, OUTPUT);
pinMode(RB, OUTPUT);
Wire.begin();
 compass = HMC5883L();
 error = compass.SetScale(1.3);
if(error != 0){
```

```
 Serial.println(compass.GetErrorText(error));
error =0;
}
error =
 compass.SetMeasurementMode(Measurement_Continuous);
 if(error != 0) {
 Serial.println(compass.GetErrorText(error));
 error=0;
}
void loop(){
Obstacle();
gps.get_position(&lat, &lon, &fix_age);
 getGPS();
 Serial.print("Latitude1 : ");
 Serial.print(LAT1D,7);
Serial.print(" :: Longitude1 : ");
 Serial.println(LON1D,7);
 Serial.print("Latitude2 : ");
Serial.print(LAT2D,7);
 Serial.print(" :: Longitude2 : ");
 Serial.println(LON2D,7);
MagnetometerRaw raw = compass.ReadRawAxis();
 MagnetometerScaled scaled = compass.ReadScaledAxis();
int MilliGauss_OnThe_XAxis = scaled.XAxis;
float heading = atan2(scaled.YAxis, scaled.XAxis);
float declinationAngle = 0.0404;
 heading += declinationAngle;
if(heading < 0)
 heading += 2*PI;
if(heading > 2*PI)
heading -= 2*PI;
float headingDegrees = heading * 180/M_PI;
bearingp=bearing+10;
 bearings=bearing-10 ;
```

```
while(headingDegrees<bearings)
{
digitalWrite(LB, HIGH);
 digitalWrite(RF, HIGH);
digitalWrite(LF, LOW);
 digitalWrite(RB, LOW);
MagnetometerRaw raw = compass.ReadRawAxis();
MagnetometerScaled scaled = compass.ReadScaledAxis();
 int MilliGauss_OnThe_XAxis = scaled.XAxis;
float heading = atan2(scaled.YAxis, scaled.XAxis);
 float declinationAngle = 0.0404;
heading += declinationAngle;
if(heading < 0)
heading += 2*PI;
 if(heading > 2*PI)
 heading -= 2*PI;
float headingDegrees = heading * 180/M_PI;
Serial.print(headingDegrees);
 Serial.print(" ");
Serial.println(bearing);
 if(headingDegrees>bearings){
 digitalWrite(LF, HIGH);
 digitalWrite(LB, HIGH);
 digitalWrite(RF, HIGH);
digitalWrite(RB, HIGH);
break;}
 }
digitalWrite(LF, LOW);
 digitalWrite(RF, LOW);
 while(headingDegrees>bearingp)
 {
 digitalWrite(RB, HIGH);
 digitalWrite(LF, HIGH);
digitalWrite(LB, LOW);
```

```
digitalWrite(RF, LOW);
MagnetometerRaw raw = compass.ReadRawAxis();
 MagnetometerScaled scaled = compass.ReadScaledAxis();
int MilliGauss_OnThe_XAxis = scaled.XAxis;
float heading = atan2(scaled.YAxis, scaled.XAxis);
float declinationAngle = 0.0404;
 heading += declinationAngle;
if(heading < 0)
heading += 2*PI;
if(heading > 2*PI)
 heading -= 2*PI;
 float headingDegrees = heading * 180/M_PI;
Serial.print(headingDegrees);
Serial.print(" ");
 Serial.println(bearing);
 if(headingDegrees<bearingp){
digitalWrite(LF, HIGH);
 digitalWrite(LB, HIGH);
digitalWrite(RF, HIGH);
 digitalWrite(RB, HIGH);
break;}
 }
 Serial.print(headingDegrees);
digitalWrite(LF, LOW);
 digitalWrite(RF, LOW);
 }
void getGPS(){
 bool newdata = false;
unsigned long start = millis();
 // Every 1 seconds we print an update
 while (millis() - start < 100)
{
 if (feedgps ()){
 newdata = true;
```

```
}
 }
if (newdata)
 {
gpsdump(gps);
}
 }
 bool feedgps(){
while (Serial3.available())
 {
if (gps.encode(Serial3.read())) return true;
 }
return 0;
 }
void gpsdump(TinyGPS &gps)
 {
//byte month, day, hour, minute, second, hundredths;
 gps.get_position(&lat, &lon);
 LAT = lat;
 LON = lon;
LAT1D=LAT/1000000;
LON1D=LON/1000000;
 LAT1R=LAT1D *(PI/180);
 LON1R=LON1D *(PI/180);
while (LAT2D==0 || LON2D==0){
 while (Serial1.available() > 0)
 { int inChar1 = Serial1.read();
 if (inChar1 != '\n') {inString1 += (char)inChar1;}
 else {
 x[y]=inString1.toFloat();
inString1 = "";
if(y==1)
 {y=-1;}
 y++;
```

```
 if(x[0]<50)
{LON2D=x[0];}
if(x[1]<50)
{LAT2D=x[1];}
 if(x[1]>50)
{LON2D=x[1];}
 var++;
 }
}
 }
 LAT2R=(LAT2D)*(PI/180);
LON2R=(LON2D)*(PI/180);
double DLON=LON2R-LON1R;
 double DLAT=LAT2R-LAT1R;
double a= ( ( sin (DLAT/2)* sin (DLAT/2) )+ (
cos (LAT2R)*cos(LAT1R) * sin (DLON/2)* sin (DLON/2) ) ) ;
 double c=2*( atan2 ( (sqrt (a)),(sqrt (1-a)) ) );
 double d=c*6371000*39.3701;
double flon1=LON1D;
double x2lon=LON2D;
 double flat1=LAT1D;
double x2lat=LAT2D;
flon1 = radians(flon1);
 x2lon = radians(x2lon);
flat1 = radians(flat1);
x2lat = radians(x2lat);
double-heading=atan2(sin(x2lon-flon1)*cos(x2lat),cos(flat1)*sin(x2lat)-
sin(flat1)*cos(x2lat)*cos(x2lon-flon1));
heading = heading*180/3.1415926535;
 int head =heading; //make it a integer now
if(head<0){heading+=360;}
bearing=heading;
 Serial.print("heading:");
 Serial.println(heading); // print the heading.
```

```
Serial.print(" Distance=");
Serial.print(d,4);
 Serial.println("inch");
 Serial.print(" Bearing=");
 Serial.print(bearing,4);
 Serial.println("Degree");
if (Serial.available() > 0)
{
 bt = Serial.read();
 digitalWrite(frontled, 1);
long duration, inches, cm;
pinMode(pingPin, OUTPUT);
 digitalWrite(pingPin, LOW);
 delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
 delayMicroseconds(10);
digitalWrite(pingPin, LOW);
 pinMode(echoPin, INPUT);
 duration = pulseIn(echoPin, HIGH);
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);
 if (cm > 10 ){
if(bt == 'F') //move forwards
 {
 digitalWrite(outPin1,HIGH);
digitalWrite(outPin2,LOW);
 digitalWrite(outPin3,HIGH);
 digitalWrite(outPin4,LOW);
}
else if (bt == 'B' {
digitalWrite(outPin1,LOW);
digitalWrite(outPin2,HIGH);
 digitalWrite(outPin3,LOW);
 digitalWrite(outPin4,HIGH);
```

```
 }
else if (bt == 'S')
 {
digitalWrite(outPin1,LOW);
digitalWrite(outPin2,LOW);
 digitalWrite(outPin3,LOW);
digitalWrite(outPin4,LOW);
}
else if (bt == 'R')
{
digitalWrite(outPin1,HIGH);
digitalWrite(outPin2,LOW);
 digitalWrite(outPin3,LOW);
 digitalWrite(outPin4,LOW);
 }
else if (bt == 'L')
{
 digitalWrite(outPin1,LOW);
 digitalWrite(outPin2,LOW);
 digitalWrite(outPin3,HIGH);
 digitalWrite(outPin4,LOW);
 }
else if (bt == 'I')
{
 digitalWrite(outPin1,HIGH);
 digitalWrite(outPin2,LOW);
digitalWrite(outPin3,LOW);
 digitalWrite(outPin4,HIGH);
 }
else if (bt == 'G')
{
digitalWrite(outPin1,LOW);
 digitalWrite(outPin2,HIGH);
 digitalWrite(outPin3,HIGH);
```

```
digitalWrite(outPin4,LOW);
 }
}
else
{
 digitalWrite (buzzerPin, HIGH);
 delay (500);
digitalWrite (buzzerPin, LOW);
delay (500);
 }
 }
 }
 long microsecondsToInches(long microseconds) {
return microseconds / 74 / 2;
 }
long microsecondsToCentimeters(long microseconds) {
 return microseconds / 29 / 2;
 }
 {
 feedgps();
}
 }
void Obstacle()
{
 myservo.write(20); //////Servo Turn Right
 delay(200);
 sensorValue = analogRead(A1);
if(sensorValue>100)
{
 STOP();
 delay(2);
LEFT();
delay(2000);
STOP();
```

```
 FORWARD();
delay(2000);
STOP();
 }
myservo.write(65); ////// Servo Turn Front
delay(200);
sensorValue = analogRead(A1);
if(sensorValue>100)
{
 STOP();
delay(2);
LEFT();
delay(2000);
STOP();
FORWARD();
 delay(2000);
STOP();
 }
myservo.write(110); ////// Servo Turn Left
 delay(200);
sensorValue = analogRead(A1);
 if(sensorValue>100)
 {
 STOP();
 delay(2);
 RIGHT();
 delay(2000);
STOP();
FORWARD();
 delay(2000);
STOP();
}
 myservo.write(65); //////Servo Turn Front
 delay(200);
```

```
 sensorValue = analogRead(A1);
if(sensorValue>100)
{
STOP();
 delay(2);
 LEFT();
 delay(2000);
STOP();
FORWARD();
 delay(2000);
 STOP();
}
FORWARD();
}
void FORWARD()
 {digitalWrite(LF, LOW);
 digitalWrite(RF, LOW);
}
void STOP()
{digitalWrite(LF, HIGH);
 digitalWrite(LB, HIGH);
digitalWrite(RF, HIGH);
 digitalWrite(RB, HIGH);}
void LEFT()
 {digitalWrite(LB, LOW);
digitalWrite(RF, LOW);}
void RIGHT()
{digitalWrite(LF, LOW); digitalWrite(RB, LOW);}
```

6.1.2 IMAGE PROCESSING CODE

```
import numpy as np
import cv2
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
cap.set(3,320) # set Width
cap.set(4,240) # set Height
while True:
    ret, img = cap.read()
    #img = cv2.flip(img, -1)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.2,
        minNeighbors=5,
        minSize=(20, 20)
    )
 for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
   cv2.imshow('detection',img)

    k = cv2.waitKey(30) & 0xff
    if k == 27: # press 'ESC' to quit
        break
cap.release()
cv2.destroyAllWindows()
```