BORDER SWARM FORCE

A PROJECT REPORT

Submitted by

AM.EN.U4ECE13007	**APARNA RAM S S**
AM.EN.U4ECE13020	**GOKUL KRISHNA V**
AM.EN.U4ECE13049	**RAGESH R**
AM.EN.U4ECE13058	**SOORAJ MURALIDHARAN**

In partial fulfillment for award of the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the guidance of

Mr. Meher Madhu D

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

AMRITAPURI -690525, KERALA

MAY 2017

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in text.

Signature of the Student:
Place:
Date:

Signature of the Student:
Place:
Date:

Signature of the Student:
Place:
Date:

Signature of the Student:
Place:
Date:

Signature of the Guide:
Place:
Date:

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

AMRITA SCHOOL OF ENGINEERING, AMRITAPURI

CERTIFICATE

This is to certify that the project report entitled **"*BORDER SWARM FORCE*"** submitted by ***Ms. APARNA RAM S S, Mr. GOKUL KRISHNA V, Mr. RAGESH RAMACHANDRAN and Mr. SOORAJ MURALIDHARAN.*** to ***Amrita Vishwa Vidyapeetham University*** towards partial fulfillment of the requirements for the award of the ***Degree of Bachelor of Technology*** *in* ***Electronics and Communication Engineering*** is a bonafide record of the work carried out by his/her under my supervision and guidance.

Mr. Mehr Madhu D
Assistant Professor
Department of EEE
Amrita School of Engineering
Amritapuri Campus

Head of the Departmment
Dept of ECE
Amrita School of Engineerng
Amritapuri Campus

ACKNOWLEDGEMENT

We humbly dedicate this project at the lotus feet of our beloved AMMA, Satguru Shri Mata Amritanandamayi Devi, our guiding force and inspiration.

Our humble salute to the Indian Armed Force for providing the inspiration and motivation for taking up this project.

We are deeply obliged to our teachers, Dr. Sundararaman Gopalan (Chairperson, ECE Dept.) & Dr. Purushothaman (Vice- Chairperson, ECE Dept).

We thank the ECE Dept. for the opportunity given to us to do this project. We extend our sincere thanks to all the faculties of the department for the continuous support both technical and material all throughout the project.

We express our sincere gratitude to our guide Mr. Meher Madhu D (Asst. Professor, EEE Dept) without whose guidance we wouldn't be able to complete the project and present in an orderly manner during reviews and demonstrations.

Special thanks to Mrs. Aswathy K N (Asst. Prof, ECE), Ms Anu Chalil (Asst. Prof, ECE), Mrs. Geethu R S (Asst. Prof, ECE) and Mrs. Baby Sreeja (Asst. Prof, ECE) for the smooth scheduling of the project and for the guidance in completing the project.

We would like to thank all our professors and laboratory staffs, for lending us their heart-felt cooperation and support whenever we required.

We also express our sincere gratitude to this Institution and for providing all the facilities required for the completion of this project.

We extend our deepest gratitude to our parents, friends and classmates for their whole hearted support and constructive criticism.

TABLE OF CONTENTS

ABSTRACT

Inadequate sleep, harsh, abusive and sadistic seniors and a constant fear that even genuine errors would be treated as negligence of duty and invite punishment — such is the lot of our Border Security Force (BSF) soldiers. This, from 'Emotional Intelligence and Occupational Stress' a first of its kind study conducted by the government on BSF jawans and officers, who're often deployed in inhospitable and risky locations. According to a study report, over 70 per cent of personnel interviewed had a complaint that they were not getting adequate rest and sleep. The number is larger for the other ranks (jawans and constables). Many complained that they were getting as little as four hours of sleep on a regular basis. Such physical exhaustion and sleep deprivation leads to chronic stress and affects performance badly.

We plan to develop a network of swarm UGV and UAV collaborated together for surveillance, targeting, and improved communications. We integrate newly available technologies into computational, vision, communications and sensing payloads and develop sensing algorithms to support vision-based target tracking and alert the soldiers about enemy infiltration, which reduces the occupational stress on soldiers.

The swarm ground robots is used for continuous surveillance throughout day and night and whenever any motion is detected or any disturbance is sensed the entire ground robots becomes vigilant and passes this message along with its location to all robots in the network and makes them alert about the incoming threat. The UAV which was initially at rest would be deployed to the target location sent by the ground robot. The entire swarm of ground robots moves towards the target location covering the enemy from all directions thus acting as a first line of defense. This will give enough time for the soldiers to react to the situation reducing threat to the life of soldiers. All these robots communicate with each other exchanging their locations and other information throughout the mission.

LIST OF FIGURES

ABBREVIATIONS

UGV - Unmanned Ground Vehicle

UAV – Unmanned Aerial Vehicle

IP - Internet Protocol

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

MTU – Maximum Transmission Unit

GPS – Global Positioning System

IMU – Inertial Measurement Unit

IR – Infra Red

CHAPTER 1

INTRODUCTION

Robots become ever more useful in today's society, the demand for them to do more continue to grow. Some of the potential applications of robots do not always require a single, incredibly complex robot. Instead, sometimes, a task can be best achieved by a team of simpler robots working together. These teams working together in the real world present several new and interesting challenges. Working together requires communication and coordination that can be difficult in real environments. Because of the challenges and potential benefits, control of teams of robots is an active area of research. As new theories are developed for coordination and control of teams of robots, experiments need to be done to evaluate their effectiveness.

The future battle space will see an increased number of semi-autonomous and fully autonomous systems. By 2030 it is expected that fully autonomous systems will be sufficiently robust to permit them to vastly outnumber humans on the battlefield. To advance to the lofty goals of 2030 we are currently researching multi robot systems and collective robotics.

1.1 INTRODUCTION TO SWARM

A swarm robot system or a multi robot system is a team of robots, either heterogeneous or homogeneous, having the following characteristics: cooperation, awareness, communication and coordination. Each of these terms, as it pertains to the qualification of multi robot systems, is defined below:

Cooperation is defined as joint operation or action amongst a group of individuals.

Awareness is defined as the knowledge of the existence of other individuals in a system.

Communication is defined as the act of relaying information from one individual, either directly or indirectly, to another individual.

Coordination is defined as cooperation in which the actions of the group are performed as a reaction to the previous actions of the group. Coordination can be either centralized or decentralized.

The swarm robots can be controlled by two ways: manual and automatic. Normally it is controlled in automatic mode. But they can be controlled manually by accessing the IP address.

Advantages and Disadvantages

There are many applications where cooperating mobile robots could be a good solution. A few of these have been discussed in the literature. One example presented in the paper uses a team of robotic highway safety cones to mark out construction sites. Each cone contains a simple robot which must work with the others to close down a lane for highway maintenance or other activities. Papers such as and others examine the problem of controlling formations of spacecraft such as satellites. Another area of application involves distributed sensor networks where there are many agents each equipped with sensors. The paper is an example of this, where a formation of robots tracks the maximum or minimum

of some sensed variable in the environment. Another example of a distributed sensing application is the paper which looks into the problem of distributing optimally agents with sensors in an environment. There are many other potential applications.

Currently fielded UGVs and UAVs act as standalone tele operated systems with one operator per asset. Integrating UGV and UAV into a single collaborative system has the potential to create dramatic improvements in situational awareness and coordinated control providing force multiplication in intelligence, reconnaissance, and overall mission performance. Most of the UGV's uses sensors to interact with the environment. Sensors like IR sensors, imu's along with the cameras and an efficient GPS system.

So we included a UGV and UAV based collaborative system for surveillance, target locating and tracking along with human detection.

The swarm ground robots is used for continuous surveillance throughout day and night and whenever any motion is detected or any disturbance is sensed the entire ground robots becomes vigilant and passes this message along with its location to all robots in the network and makes them alert about the incoming threat. The UAV which was initially at rest would be deployed to the target location sent by the ground robot. The entire swarm of ground robots moves towards the target location covering the enemy from all directions thus acting as a first line of defense. This will give enough time for the soldiers to react to the situation reducing threat to the life of soldiers. All these robots communicate with each other exchanging their locations and other information throughout the mission.

The aerial robot will be a quad copter equipped with Raspberry Pi3, GPS, IMU, ultrasonic sensors. The swarm of ground robots is also equipped with all the payloads carried by UAV with the only difference of the means of travel. The ground robot would be a four wheel drive all terrain vehicles capable of maneuvering over the difficult terrains and withstanding a large range of temperature which is required to work along Indian Borders.

The major advantages of this multi-robot system compared to existing models will be its speed since it does not involve map building and uses behavioral based localization instead of map based localizations. This project can reduce occupational stress and provides adequate rest and sleep to soldiers, reduce threat to the life of soldiers Thereby Improves the overall performance of the border security force

CHAPTER 2
SYSTEM DESIGN

The system consists of 3 UGV and a quadcopter communicating with each other and the base station, their position and identified target location thereby making them aware of the situation.

2.1 UGV

UGV is a ground robot which is designed as an all-terrain chassis. It uses an 8 bit Microcontroller to collect the sensor values and to navigate the robot. The different sensors are ultrasonic sensor, a compass and a GPS. It uses a Raspberry pi embedded system. It also includes a Raspberry pi camera which is used for human detection. The pi camera is adjusted by 2 servo motors making it a pan and tilt camera. Raspberry pi has its own communication system with 2.4GHz IEEE 802.11n with 140mbps data rate.

2.2 NAVIGATION

With navigation, we refer to the task of finding a collision-free path for a robotic system to travel from one place to another. Algorithms in swarm robotics mostly rely on cooperation and simple interactions between robots, rather than on complex individual behaviors that require powerful sensory capabilities. Concretely, in the context of navigation, this means that the focus is on cooperative navigation, where robots guide each other, rather than on the use of maps or map-building strategies.

2.2.1 Path planning for UGV

Path planning is one of the most important elements for mobile robot. Path planning is the determination of a path that a robot must take in order to pass over each point in an environment and path is a plan of geometric locus of the points in a given space where the robot has to pass through. Generally, the problem of path planning is about finding paths by connecting different locations in an environment such as graph, maze and road. Path planning “enables” mobile robots to see the obstacle and generate an optimum path so as to avoid them. In general path planning gives an idea to the robots for travelling from an initial location A to a final location B. There are two types of path planning.

2.2.1.1 Global Path Planning

Global path planning is a path planning that requires robot to move with priori information of environment. The information about the environment first loaded into the robot path planning program before determining the path to take from starting point to a target point.

2.2.1.2 Local Path Planning

Local path planning is path planning that requires robot to move in unknown environment or dynamic environment where the algorithm is used for the path planning will response to the obstacle and the change of environment.

As our robots should operate in a dynamic environment, we chose Local Path Planning. Usually Bug algorithms are used for path planning.

2.2.1.2.1 Bug Algorithms

Bug algorithms are well known mobile robot navigation method for local path planning with minimum sensor and simple algorithm.

There are different types of bug algorithms

2.2.1.2.1.1 Bug 1 algorithm

Bug1 moves from start point toward target point by hitting and circumnavigating the obstacle then leaving the leave point. Bug1 is considered overcautious and having coverage more than the full perimeter of the obstacle.

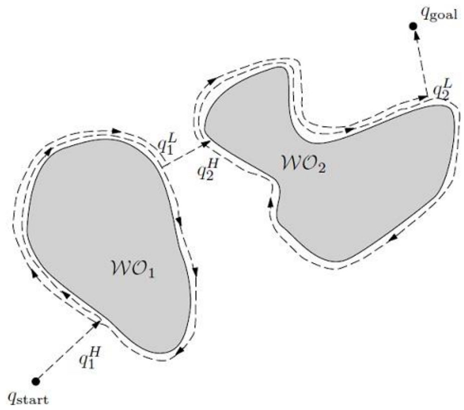


Figure 2.1: Bug 1 algorithm

2.2.1.2.1.2Bug 2 algorithm

Bug 2 has similar behavior as Bug 1except it is guided by the tangent line where tangent line is used as leaving point and hitting point.

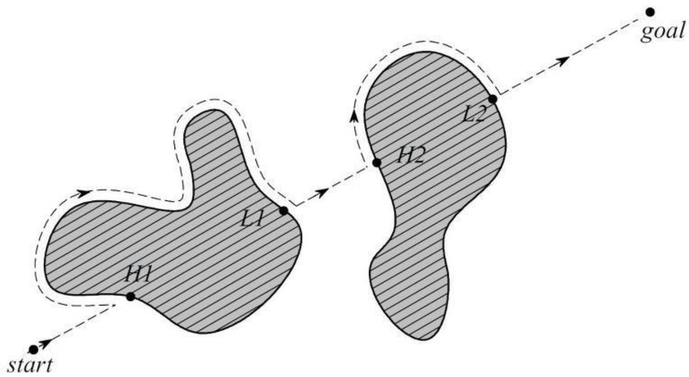


Figure 2.2: Bug 2 algorithm

2.2.1.2.1.3Intelligent Bug

The proposed IBA algorithm is based on two behaviors: move to goal and obstacle avoidance. The behaviors in IBA also depend on the present sensorial information of environment i.e. whether obstacles are sensed or not. Initially, in move to goal behavior, a reference path is generated from source to goal position and the robot is forced to follow it until an obstacle is encountered or destination is reached.

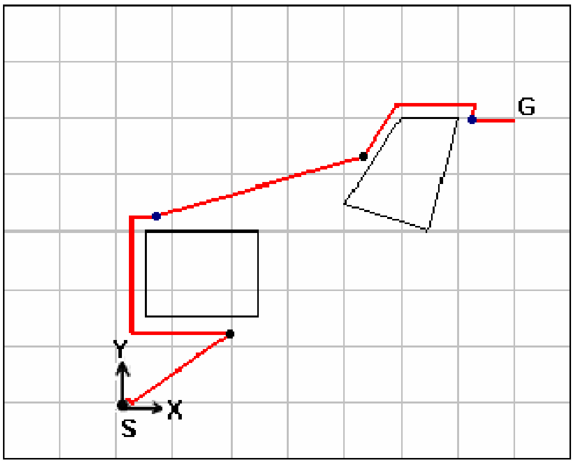


Figure 2.3: Intelligent Bug

2.2.1.2.1.4Intelligent bug algorithm (IBA)

The behavior of the robot is changed to obstacle avoidance when an obstacle is sensed and the robot is commanded to follow the edges of the obstacle until leaving point is reached. In IBA, leaving point by taking the goal position into account is selected on the basis of free path toward the destination. The robot monitors the obstacles in the path towards destination while detecting edge in obstacle avoidance behavior.

The condition dictates that in IBA, the leaving point is not taken on the basis of minimum distance to destination. The obstacle-free path towards goal is also considered. This ensures that the robot does not have to wait for the point having minimum distance to goal. The robot changes its behavior to move to goal in order to generate new reference path, in case an obstacles-free path is sensed (just like a human as they follow the straight path after avoiding hurdles).

We are using intelligent Bug Algorithm as it is less time consuming and more accurate.

2.2.2Localization

Localization involves one question: Where is the robot now? It identifies the current position of the robot itself with reference to some landmark with the help of sensors.

Localization technique that works fine for one robot in one environment may not work well or at all in another environment.

All localization techniques generally provide two pieces of information:

What is the current location of the robot in some environment?

What is the robot's current orientation in that same environment?

GPS mechanism was used for localizing the UGV

2.2.2.1Global Positioning System (GPS)

GPS is a highly useful utility that provides users with positioning, navigation and timing services. It is used for a variety of applications like in electronic equipments, military uses etc.

2.2.2.2 Localization using GPS

The GPS value of the location of the UGV is generated with the help of a mobile application and is fed back to the Arduino using Bluetooth. The mobile phone uses an android application named GPS developed by us.

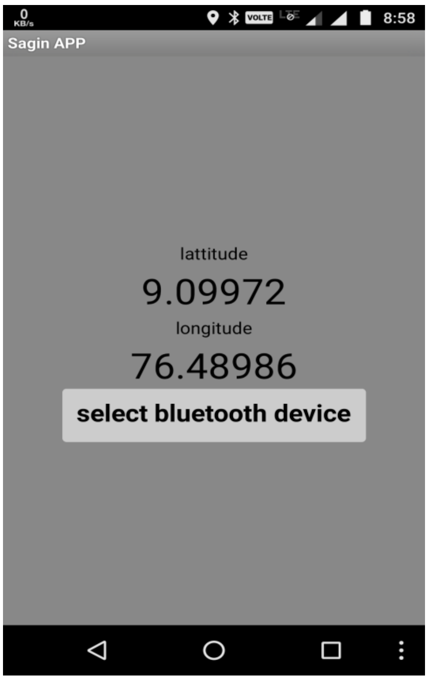


Figure 2.4 : Android application to receive GPS values

The distance between two GPS points is calculated using Haversine formula. Haversine formula is an equation important in navigation giving great circle distances between two points on a sphere from their longitudes and latitudes.

2.2.2.3 Compass

Compass module is used in the UGVs to give the orientation of the robots. The angle which we obtain from the compass module is in reference with the magnetic north pole of Earth.

2.2.3 Obstacle Avoidance

Human beings and other animals move through a terrain avoiding obstacles by utilizing their external sense organs. Similarly, in the UGVs we use ultrasonic sensors (HC-SR04) for detecting the obstacles and avoiding it. Three ultrasonic sensors are arranged strategically at definite angles so that the blind spots are minimized. The sensor data determines the direction of travel of the UGV.

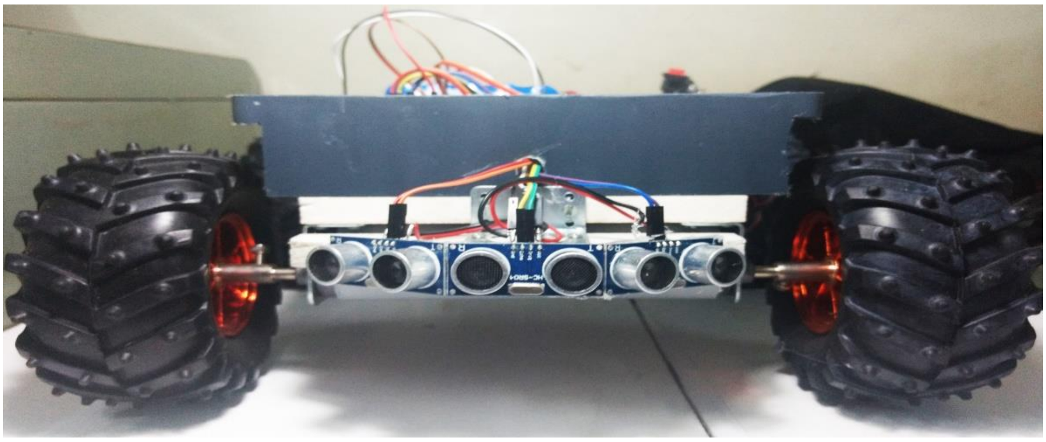


Figure 2.5 : Ultrasonic sensor arrangement

2.3 HUMAN DETECTION

Human sensing (also called human detection or human presence detection) encompasses a range of technologies for detecting the presence of a human body in an area of space, typically without the intentional participation of the detected person. Common applications include search and rescue, surveillance, and customer analytics. The problem of human detection is to automatically locate people in an image or video sequence. Keeping continuous track of person will allow identifying person at any time. The system consists of two parts first human detection and secondly tracking. Human detection step is split into face detection and eye detection. Face is a vital part of human being represents most important information about the individual. Eyes are the important biometric feature used in person identification. Face detection is done using skin color based methods. Detecting humans in images is a challenging task owing to their variable appearance and the wide range of poses that they can adopt. The first need is a robust feature set that allows the human form to be discriminated cleanly, even in cluttered backgrounds under difficult illumination. Two methods which came into discussion were the Histogram of Oriented Gradients (H.O.G) mechanism and the HAAR cascade classifier mechanism.

2.3.1 HAAR vs. HOG

HAAR mechanism considers various features of interest from the human face as of eyes and the face itself. The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. HOG mechanism requires a static background for the identification. But as the project demand if of a dynamic background, Haar cascade classifier was used.

2.3.2 HAAR cascade classifiers

A Haar-like feature considers neighboring rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. In case of human faces,the areas around the eyes are darker than the areas on the cheeks. One example of a Haar-like feature for face detection is therefore a set of two neighbouring rectangular areas above the eye and cheek regions

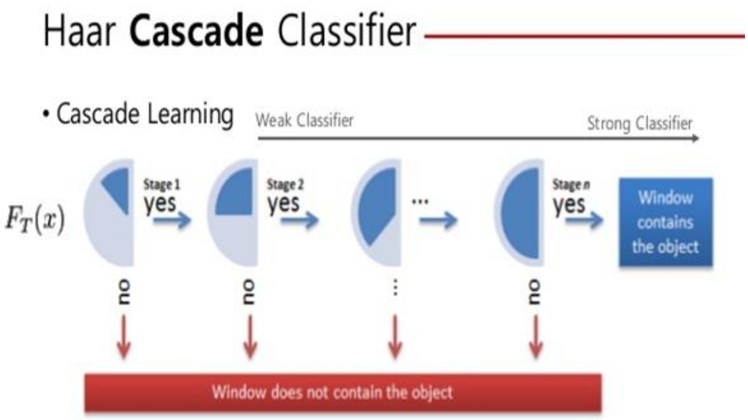


Figure 2.6 : Haar cascade classifier

The cascade classifier consists of a list of stages, where each stage consists of a list of weak learners. The system detects objects in question by moving a window over the image. Each stage of the classifier labels the specific region defined by the current location of the window as either positive or negative – positive meaning that an object was found or negative means that the specified object was not found in the image. If the labeling yields a negative result, then the classification of this specific region is hereby complete and the location of the window is moved to the next location. If the labeling gives a positive result, then the region moves of to the next stage of classification. The classifier yields a final verdict of positive, when all the stages, including the last one, yield a result, saying that the object is found in the image. OpenCV comes with a trainer as well as detector. First the required XML classifiers are loaded into the OpenCV. Input image is loaded (or video) in grayscale mode. The faces are detected.

2.3.3 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeiterathat make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, C, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision application sand takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There is over 500 algorithms and about 10 times as many functions that composeor support those algorithms. OpenCV is written natively in C++ and has a template interface that works seamlessly with STL containers.

2.3.4 Training a classifier

Haar-cascade is an object detection algorithm used to locate faces, pedestrians, objects and facial expressions in an image and mainly used for face detection. We train a classifier using a few hundred sample views of a particular object. System is provided with several numbers of positive images and negative images. Scale down these images to small dimensions. The cascading classifier is trained such that it detects a specified region that bounds the object of interest. The training is done in Open CV.

For training we need a set of samples. There are two types of samples: negative and positive.

Negative samples are taken from arbitrary images. These images must not contain detected objects. Negative samples are enumerated in a special file. It is a text file in which each line contains an image filename of negative sample image. This file must be created manually. Positive samples are created by opencv_crearesamples utility. They may be created from a single image with object with object or from a collection of previously marked up images. It should be noted that number of negative images should be greater than the number of positive images as the classifier need more proof that the given image does not have an object of interest

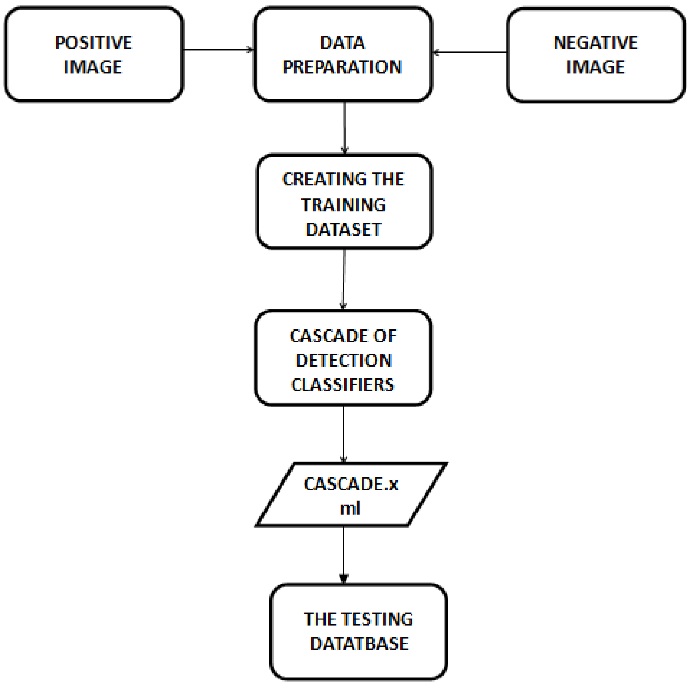


Figure 2.7: Training a classifier flowchart

2.4 COMMUNICATION

The robots need to communicate with each other all the time. As the robots are in a dynamic environment, the communication should be faster. For the communicat-ion between the robots, we use internet protocol as the medium. Internet protocol is a set of rules governing the format of data sent over the Internet or other network. There are two types of Internet Protocol (IP) traffic. They are TCP or Transmission Control Protocol and UDP or User Datagram Protocol. TCP is connection oriented – once a connection is established, data can be sent bidirectional. UDP is a simpler, connectionless Internet protocol. Multiple messages are sent as packets in chunks using UDP.

2.4.1 TCP vs UDP

TCP	UDP
Transmission Control Protocol	User Datagram Protocol or Universal Datagram Protocol
TCP is a connection-oriented protocol.	UDP is a connectionless protocol.
As a message makes its way across the internet from one computer to another. This is connection based.	UDP is also a protocol used in message transport or transfer. This is not connection based which means that one program can send a load of packets to another and that would be the end of the relationship.
TCP is suited for applications that require high reliability, and transmission time is relatively less critical.	UDP is suitable for applications that need fast, efficient transmission, such as games. UDP's stateless nature is also useful for servers that answer small queries from huge numbers of clients.

As we need faster communication network we are using UDP Protocol.

2.4.2 User Datagram Protocol (UDP)

UDP (User Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss tolerating connections between applications on the Internet. Both UDP and TCP run on top of the Internet Protocol (IP) and are sometimes referred to as UDP/IP or TCP/IP. Both protocols send short packets of data, called datagrams.

UDP provides two services not provided by the IP layer. It provides port numbers to help distinguish different user requests and, optionally, a checksum capability to verify that the data arrived intact. In contrast, UDP just sends the packets, which means that it has much lower bandwidth overhead and latency. But packets can be lost or received out of order as a result, owing to the different paths individual packets traverse between sender and receiver. UDP is an ideal protocol for network applications in which perceived latency is critical such as gaming, voice and video communications, which can suffer some data loss without adversely affecting perceived quality. In some cases, forward error correction techniques are used to improve audio and video quality in spite of some loss.

UDP can also be used in applications that require lossless data transmission when the application is configured to manage the process of retransmitting lost packets and correctly arranging received packets. This approach can help to improve the data transfer rate of large files compared with TCP.

2.4.3 Multicasting

Multicast is a kind of UDP traffic similar to BROADCAST, but only hosts that have explicitly requested to receive this kind of traffic will get it. This means that you have to JOIN a multicast group if you want to receive traffic that belongs to that group. Multicast is useful when you have to transmit the SAME message to more than one host.

Usually the client that sends multicast does not know how many servers will really receive his packets. When talking about client-server in network, the client sends the request; the server receives the request and might send back an answer. Since multicast is based UDP, the transmission is by default not reliable.

The advantage of using multicast instead of broadcast is that only interested hosts will get the message, and the message should be transmitted only once for many clients (saves a lot of bandwidth). Another advantage is the possibility of sending packets larger than interface MTU.

2.5 UAV

We are using a quadcopter as UAV. A quadcopter is an aerial vehicle that uses four rotors for lift, steering, and stabilization. Unlike other aerial vehicles, the quadcopter can achieve vertical flight in a more stable condition. The quadcopter is not affected by the torque issues that a helicopter experiences due to the main rotor. It uses an 8 bit Microcontroller to collect the sensor values and to navigate the robot. The different sensors are a compass and a GPS. It uses a Raspberry pi embedded system. It also includes a Raspberry pi camera which is used for human detection. The pi camera is fixed. Raspberry pi has its own communication system with 2.4GHz IEEE 802.11n with 140mbps data rate.

A quadcopter is an aerial vehicle that uses four rotors for lift, steering, and stabilization. Unlike other aerial vehicles, the quadcopter can achieve vertical flight in a more stable condition. The quadcopter is not affected by the torque issues that a helicopter experiences due to the main rotor. Furthermore, due to the quadcopters cyclic design, it is easier to construct and maintain .As the technology becomes more advanced and more accessible to the public, many engineers and researchers have started designing and implementing quadcopters for different uses.

Various groups such as the military, engineers, researchers, and hobbyists have been developing quadcopters to understand different technical areas. For example, quadcopters can be used for re-connaissance and collecting data. This could range from searching for survival victims in a disaster area to checking the state of electrical power lines. Some quadcopters in production today can hold light payloads, such as food and medical supplies, and deliver them to areas where normal planes cannot reach.

2.5.1 Flight control

A quadcopter consists of four motors evenly distributed along the quadcopter frame as can be seen in figure 2.8 below. The circles represent the spinning rotors of the quadcopter and the arrows represent the rotation direction. Motors one and three rotate in a clockwise direction using pusher rotors. Motor two and four rotate in a counter-clockwise direction using puller rotors. Each motor produces a thrust and torque about the center of the quadcopter. Due to the opposite spinning directions of the motors, the net torque about the center of the quadcopter is ideally zero, producing zero angular acceleration. This eliminates the need for yaw stabilization.

A vertical force is created by increasing the speed of all the motors by the same amount of throttle. As the vertical forces overcome the gravitational forces of the earth, the quadcopter begins to rise in altitude. Figure 2.9 shows the vertical movement of the quadcopter. As above, the circles represent the spinning rotors, the larger arrows represent the direction the rotors are spinning, and the black arrows represent the forces caused by the spinning rotors.

Pitch is provided by increasing (or decreasing) the speed of the front or rear motors. This causes the quadcopter to turn along the x axis. The overall vertical thrust is the same as hovering due to the left and right motors; hence only pitch angle acceleration is changed. Figure 2.10 shows an example of pitch movement of a quadcopter. As the front motor slows down, the forces created by the corresponding rotor are less than the forces created by the back rotor.

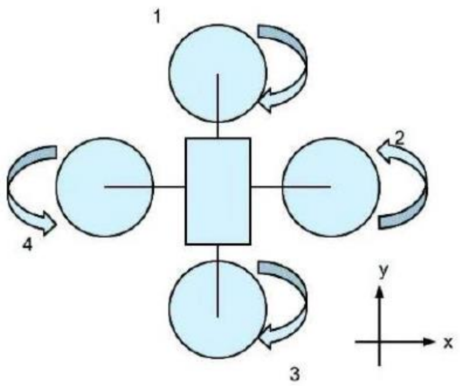


Figure 2.8: Quadcopter: Motor rotation directions.

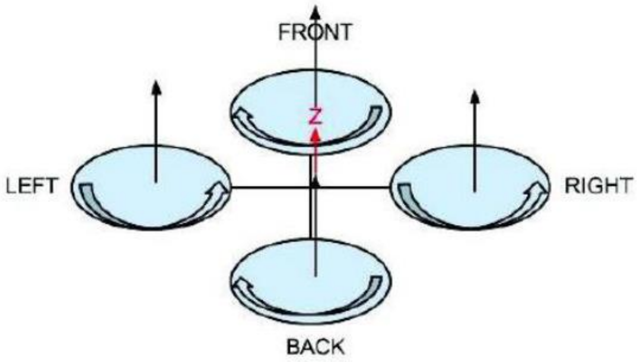


Figure 2.9 : Quadcopter: Vertical thrust movement.

These forces are represented by the blue arrows. These forces cause the quadcopter to tip forward and this movement is represented by the red arrow.

Roll is provided by increasing (or decreasing) the speed of the left rotor speed and right motors. This causes the quadcopter to turn along the y axis. The overall vertical thrust is the same as hovering due to the front and back motors; hence only roll angle acceleration is changed. Figure 2.11 shows an example of roll movement of a quadcopter. As the right motor slows down, the forces created by the corresponding rotor are less than the forces created by the left rotor. These forces are represented by the blue arrows. This causes the quadcopter to tip to the right and this movement is represented by the red arrow.

Yaw is provided by increasing (or decreasing) the speed of the front and rear motors or by increasing (or decreasing) the speed of the left and right motors. This causes the Quadcopter to turn along its vertical axis in the direction of the stronger spinning rotors.

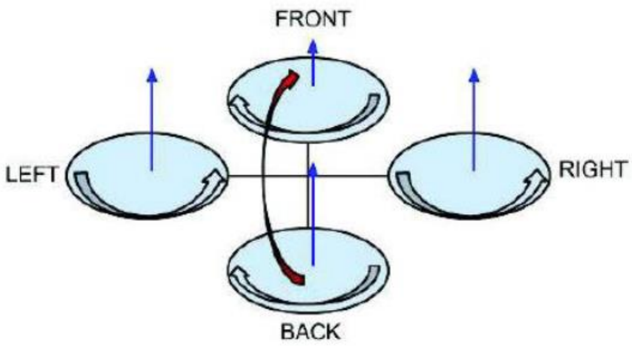


Figure 2.10: Quadcopter: Pitch movement

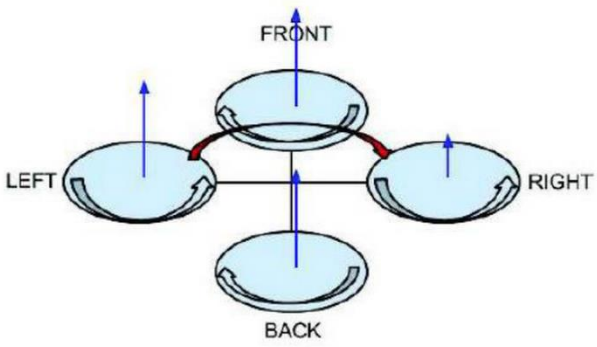


Figure 2.11: Quadcopter: Roll movement.

Figure 2.12 shows an example of yaw movement of a quadcopter. As the front and back motor slows down, the forces created by the corresponding rotors are less than the forces created by the left and right rotors. The quadcopter will begin to rotate in the same direction as the faster spinning rotors due to the difference in torque forces. This movement is represented by the red arrow.

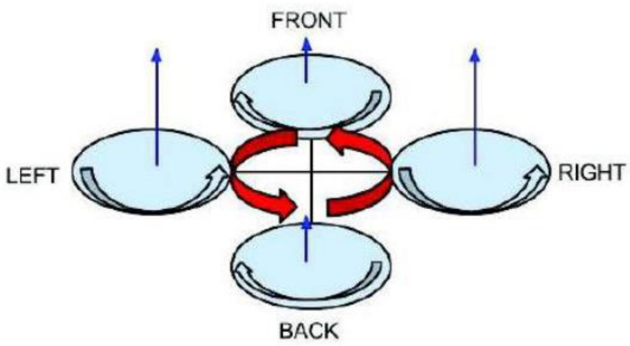


Figure 2.12: Quadcopter: Yaw movement

2.5.2 Advantages of a Quadcopter

There are many advantages to quadcopters compared to other aircrafts. A quadcopter does not require a large area to obtain lift, like a fixed wing aircraft does. The quadcopter creates thrust with four evenly distributed motors along its frame. A helicopter suffers from torque issue due to its main rotor. The design of the quadcopter does not suffer from the same torque issues as the helicopter. The counter balancing forces of the spinning motors cancel out the torque forces caused by each motor causing the quadcopter to balance itself. Because the quadcopter uses four rotors instead of one main rotor, it requires less kinetic energy per rotor for the same amount of thrust when compared to the helicopter. Due to this and its symmetrical design, a quadcopter maintenance and manufacturing costs are relatively lower than other aircrafts.

CHAPTER 3
COMPONENTS

3.1 Unmanned Ground Vehicle (UGV)

Our experimental platform is three four wheeled differential drive autonomous robots that will function as a swarm force. Four wheels are powered by high torque geared motors of 60 RPM. Three HC-SR04 ultrasound sensors are attached in the front of each robot which gives information about obstacles in 4m range. A digital compass - HMC5883L – is used to find the direction of robot's movement. The core elements of the embedded system of this robot are (i) 8 bit ATmega328 microcontroller based Arduino board (ii) A Raspberry pie. Arduino collects data from all the sensors used in the platform and also controls the motion of motors through a motor driver.

3.1.1 Raspberry pi

Figure 3.1 : Raspberry pi 3 Development Board

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries.The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi. Some accessories however have been included in several official and unofficial bundles.

3.1.1.1 Overview

All models feature a Broadcom system on a chip (SoC), which includes an ARM compatible central processing unit (CPU) and an on-chip graphics processing unit (GPU, a VideoCore IV). CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either the SDHC or MicroSDHC sizes. Most boards have between one and four USB slots, HDMI and composite video output, and a 3.5 mm phone jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I^2C. The B-models have an 8P8CEthernet port and the Pi 3 and Pi Zero W have on board Wi-Fi 802.11n and Bluetooth.

The Foundation provides Raspbian, a Debian-based Linux distribution for download, as well as third party Ubuntu, Windows 10 IOT Core, RISC OS, and specialized media center distributions. It promotes Python and Scratch as the main programming language, with support for many other languages. The default firmware is closed source, while an unofficial open source is available

3.1.1.2 Hardware

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support. The USB/Ethernet chip contains a five-point USB hub, of which four ports are available

3.1.1.2.1 Processor

The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

3.1.1.2.2 Performance

The Raspberry Pi 3, with a quad-core Cortex-A53 processor, is described as 10 times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelized tasks.

3.1.1.2.3 RAM

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero and Zero W have 512 MB of RAM.

3.1.1.2.4 Networking

The Model A, A+ and Pi Zero have no Ethernet circuitry and are commonly connected to a network using an external user-supplied USB Ethernet or Wi-Fi adapter. On the Model B and B+ the Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip. The Raspberry Pi 3 and Pi Zero W (wireless) are equipped with 2.4 GHz Wi-Fi802.11n(150 Mbit/s) and Bluetooth 4.1(24 Mbit/s) based on Broadcom BCM43438 FullMAC chip with no official support for Monitor mode but implemented through unofficial firmware patching and the Pi 3 also has a 10/100 Ethernet port.

3.1.1.2.5 Peripherals

The Raspberry Pi may be operated with any generic USB computer keyboard and mouse. It may also be used with USB storage, USB to MIDI converters, and virtually any other device/component with USB capabilities. Other peripherals can be attached through the various pins and connectors on the surface of the Raspberry Pi.

3.1.1.2.6 Video

The video controller can emit standard modern TV resolutions, such as HD and Full HD, and higher or lower monitor resolutions and older standard CRT TV resolutions. Allowing the highest resolutions does not imply that the GPU can decode video formats at those; in fact, the Pis are known to not work reliably for H.265 (at those high resolutions), commonly used for very high resolutions (most formats, commonly used, up to Full HD, do work).Although the Raspberry Pi 3 does not have H.265 decoding hardware, the CPU is more powerful than its predecessors, potentially fast enough to allow the decoding of H.265-encoded videos in software. The GPU in the Raspberry Pi 3 runs at a higher clock frequencies of 300 MHz or 400 MHz, compared to previous versions which ran at 250MHz.

The Raspberry Pi's can also generate 576i and 480icomposite video signals, used on old-style (CRT) TV screens and less-expensive monitors through standard connectors – either RCA or 3.5 mm phone connector depending on models. The television signal standards supported are PAL-BGHID, PAL-M, PAL-N, NTSC and NTSC-J.

3.1.1.2.7 Real-time clock

None of the current Raspberry Pi models have a built-in real-time clock, so they are unable to keep track of the time of day independently. As a workaround, a program running on the Pi can retrieve the time from a network time server or from user input at boot time, thus knowing the time while powered on. To provide consistency of time for the file system, the Pi does automatically save the time it has on shutdown, and re-installs that time at boot. A real-time hardware clock with battery backup, such as the DS1307, which is fully binary coded, may be added (often via the I²C interface).

3.1.2 ARDUINO MEGA

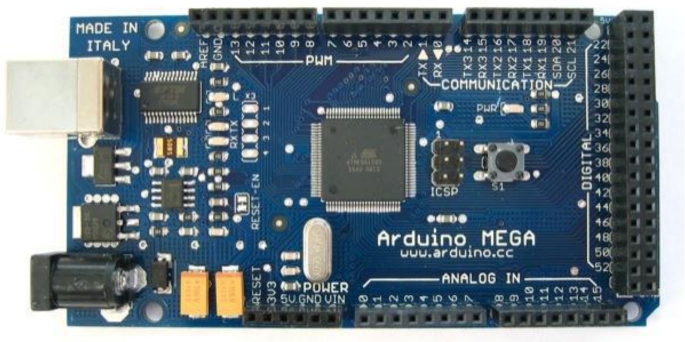


Figure 3.2 : ARDUINO MEGA Development Board

The Arduino Mega is a microcontroller board based on the ATmega 2560. It has 54 digital inputs/output pins (of which 14 can be used as PWM outputs), 16 analog pins 4UART's (hardware serial ports), a 16MHz crystal oscillator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

3.1.2.1 Summary

Microcontroller ATmega2560

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 54 (of which 15 provide PWM output)

Analog Input Pins 16

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 128 KB of which 4 KB used by boot loader SRAM 8 KB

EEPROM 4 KB

Clock Speed 16 MHz

3.1.2.2 Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

3.1.2.3 The power pins

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- 5V- the regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3. A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.
- GND. Ground pins.

3.1.2.4 Memory

The ATmega1280 has 128 KB of flash memory for storing code (of which 4 KB is used for the boot loader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms.

The Mega has 16 analog inputs, each of which provides 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analog Reference function.

3.1.3 Ultrasound Sensor - HC-SR04

Figure 3.3: HC-SR04 Ultrasound Sensor

Ultrasonic sensors work on a principle similar to sonar which evaluates distance of a target by interpreting the echoes from ultrasonic sound waves. This ultrasonic module measures the distance accurately which provides 0cm - 400cm with a gross error of 3cm. Its compact size, higher range and easy usability make it a handy sensor for distance measurement and mapping. The module can easily be interfaced to micro controllers where the triggering and measurement can be done using two pin.

The sensor transmits an ultrasonic wave and produces an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.

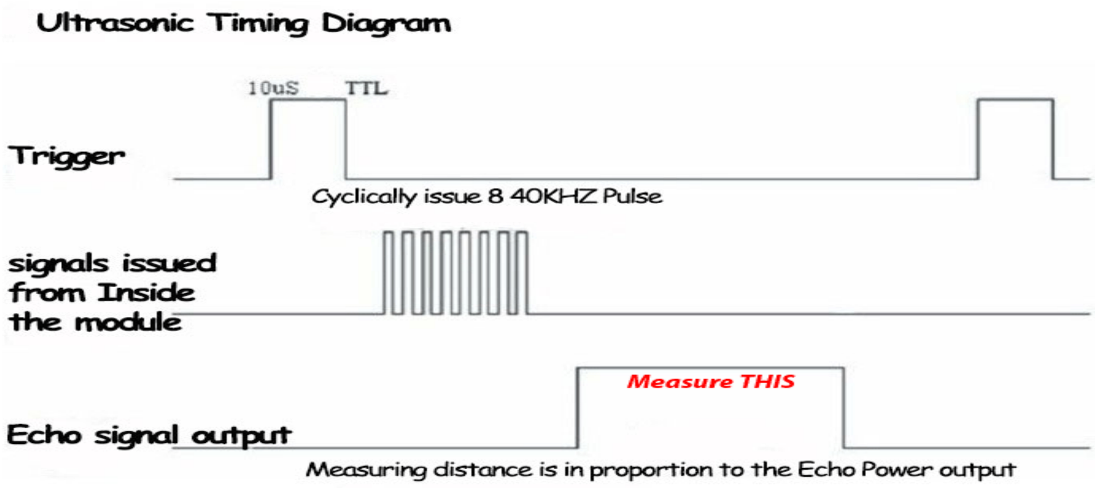


Figure 3.4: Timing diagram of the operation of HC-SR04

A short 10uS pulse is supplied to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at and raise its echo. The distance is calculated by finding the duration of the received reflected pulse as follows:

Distance = high level time * velocity (340m/s) / 2

This comes in an easy to use 4-pin breakout -Vcc, Gnd, Trigger and Echo. The coverage angle of this sensor is in both clockwise and anticlockwise direction. This sensor operates in voltage range of 4.8V to 5.5V and sends signal in 40 KHz ultrasonic frequency range.

3.1.4 Digital Compass Module - HMC5883L

Figure 3.5: Digital Compass Module – HMC5883L

The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancelation, and a 12-bit. ADC that enables 1 to 2 compass heading accuracy. The I2C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include mobile phones, consumer electronics, Auto Navigation Systems, and Personal Navigation Devices.

3.1.5 Sabertooth dual 60A motor driver

Figure 3.6 :Sabertooth dual 60A motor driver

The Sabertooth 2X60 is one of the most versatile, efficient and easy to use dual motor drivers on the market. It is suitable for high powered robots - up to 120lbs in combat or up to 1000lbs for general purpose robotics. Out of the box, the Sabertooth can supply two DC brushed motors with up to 60A each. Peak currents of 120A per channel are achievable for a few seconds. Overcurrent and thermal protection means you'll never have to worry about killing the driver with accidental stalls or by hooking up too big a motor.

Sabertooth allows you to control two motors with: analog voltage, radio control, serial and packetized serial. You can build many different robots of increasing complexity for years to come with a Sabertooth. Sabertooth has independent and speed+direction operating modes, making it the ideal driver for differential drive (tank style) robots and more. The operating mode is set with the onboard DIP switches so there are no jumpers to lose. Sabertooth features screw terminal connectors - making it possible for you to build a robot without even soldering.

Sabertooth is the first synchronous regenerative motor driver in its class. The regenerative topology means that your batteries get recharged whenever you command your robot to slow down or reverse. Sabertooth also allows you to make very fast stops and reverses - giving your robot a quick and nimble edge.

Sabertooth has a built in 5V 1A Switch-mode BEC that can supply power for your receiver or microcontroller, as well as 3-4 standard analog servos. The lithium cut-off mode allows Sabertooth to operate safely with lithium ion and lithium polymer battery packs - the highest energy density batteries available.

Sabertooth's transistors are switched at ultrasonic speeds (24 kHz) for silent operation. Sabertooth 2X60 uses 0.8 milliohm MOSFETs in its bridge. Use of our motor drivers with cheap AC adapters is not recommended. Use a battery or at least put a battery in parallel with a DC supply.

3.1.6 High Torque DC Geared Motor

Figure 3.7 : High Torque DC Geared Motor

In this project the robot has to carry consider amount of payload which includes, a battery, processing unit, stereo camera and other essential circuitry which controls the robot. To carry this load of 2kg, a considerable amount of torque is needed for the motor, so for this application the best suited motor is high torque DC geared motor which have a torque of 10Kgcm.The weight is an another attractive feature that a high torque DC geared motor possess, the weight of this motor is 300gm and weight to torque ratio of this motor is less. Since the weight of this motor is less, this will not contribute much to the weight of the robotic platform.

The base motor is having an RPM of 1800 and the attached metal gear box reduces the output RPM to 60 and corresponding increase in torque is observed. Since it is metal geared, wear and tear problems are reduced. It draws a maximum current of5A in full load and 800mA at no load. The motor is having a diameter of 28.5mmand its length is 63mm without shaft. It is having a shaft of length of 15mm and diameter of 6mm.

3.1.7 Wheels

Figure 3.8: The free rotating wheels

The robot platform is six wheeled differential steering rover among which four wheels are powered by DC motors and other two are free rotating wheels. Standard high quality plastic wheels of 70mm diameter are used for powered wheels. These wheels have a width of 40mm and come with a metal bush with 6mm hole for inserting motor shaft. These wheels also possess a good quality rubber grip which gives enough traction to smooth indoor surfaces.

The free rotating wheels used for position encoding is having same specification as the powered wheels except the width is 20mm, which is specifically chosen to reduce the friction between these wheels and ground. These wheels are fixed on a dummy shaft, and placed along the geometrical central line of the robot platform.

3.1.8 Buck convertor

Figure 3.9: Buck Converter

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage (while stepping up current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor, although modern buck converters frequently replace the diode with a second transistor used for synchronous rectification) and at least one energy storage element, a capacitor, inductor, or the two in combination.

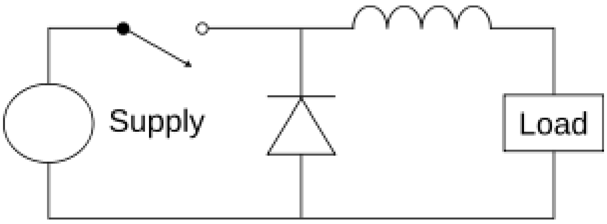


Figure 3.10: Buck Converter circuit diagram

To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output (load-side filter) and input (supply-side filter).

Switching converters (such as buck converters) provide much greater power efficiency as DC-to-DC converters than linear regulators, which are simpler circuits that lower voltages by dissipating power as heat, but do not step up output current. Buck converters can be remarkably efficient (often higher than 90%), making them useful for tasks such as converting a computer's main (bulk) supply voltage (often 12V) down to lower voltages needed by USB, DRAM, the CPU (1.8V or less), etc.

3.1.9 Raspberry Pi Camera Module

Figure 3.11 : Rasperry pi camera

The Raspberry Pi camera module can be used to take high-definition video, as well as stills photographs. It’s easy to use for beginners, but has plenty to offer advanced users if you’re looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion and other video cleverness. You can also use the libraries we bundle with the camera to create effects.

If you’re interested in the nitty-gritty, you’ll want to know that the module has a five mega pixel fixed-focus camera that supports 1080p30, 720p60 and VGA90 video modes, as well as stills capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Pi camera Python library.

The camera module is very popular in home security applications, and in wildlife camera traps. You can also use it to take snapshots.

3.1.9.1 Features

- 5MP sensor
- Wider image, capable of 2592x1944 stills, 1080p30 video
- 1080p video supported
- CSI
- Size: 25 x 20 x 9 mm

3.1.9.2 Camera Details

The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. The camera's image sensor has a native resolution of five megapixels and has a fixed focus lens. The software for the camera supports full resolution still images up to 2592x1944 and video resolutions of 1080p30, 720p60 and 640x480p60/90.

3.1.10 MICRO SERVO

Figure 3.12: Micro Servo

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

A servomotor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft.

The motor is paired with some type of encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

3.1.10.1 Specifications

Required Pulse	:	3-5 Volt Peak to Peak Square Wave
Operating Voltage	:	4.8-6.0 Volts
Operating Temperature Range:		-10 to +60 Degree C
Operating Speed (4.8V)	:	0.12sec/60 degrees at no load
Operating Speed (6.0V)	:	0.10sec/60 degrees at no load
Stall Torque (4.8V)	:	1.8kg/cm
Stall Torque (6.0V)	:	2.4kg/cm
360 Modifiable	:	Yes
Bearing Type	:	Ball Bearing
Gear Type	:	Nylon Gears

3.1.11 BLUETOOTH MODULE HC- 05

Figure 3.13: Bluetooth module HC-05

This HC-06 Bluetooth module is the most economical and easiest way to go wireless. It can be accessed by pairing the Bluetooth via a mobile or any other devices. The module can be configured for baud rates 1200 to 115200 bps.

3.1.11.1 Specifications

- Support Master & Slave Mode
- Serial communications: 9600-115200bps
- SPP (Serial Port Profile) support
- Support UART,USB,PCM interface to host system
- Easy Configuration through AT Commands
- Encrypted connection
- Frequency: 2.4~2.524 GHz

- Bluetooth core V2.0 compliant
- Built-in Chip antenna
- Power Supply: 3.7-5V

3.1.12 Battery

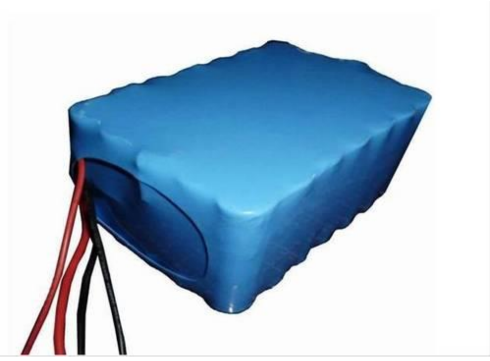


Figure 3.14: Lithium-ion battery

It is a rechargeable Lithium-ion battery pack. Just as with other lithium-ion cells, LiPos work on the principle of intercalation and de-intercalation of lithium ions from a positive electrode material and a negative electrode material, with the liquid electrolyte providing a conductive medium. To prevent the electrodes from touching each other directly, a microporous separator is in between which allows only the ions and not the electrode particles to migrate from one side to the other.

3.1.12.1 Specifications

Voltage: 11.1V

Current: 13Ah

3.1.13 Laser Pointer

Figure 3.15: Laser pointer

A laser pointer or laser pen is a small handheld device with a power source (usually a battery) and a laser diode emitting a very narrow coherent low-powered laser beam of visible light, intended to be used to highlight something of interest by illuminating it with a small bright spot of colored light. Power is restricted in most jurisdictions not to exceed 5mW. It has a power rating upto 1000mW.

3.2 Unmanned Aerial Vehicle (UAV)

Quadcopters generally use two pairs of identical fixed pitched propellers; two clockwise (CW) and two counterclockwise (CCW). These use independent variation of the speed of each rotor to achieve control. By changing the speed of each rotor it is possible to specifically generate a desired total thrust; to locate for the centre of thrust both laterally and longitudinally; and to create a desired total torque, or turning force.

3.2.1 KK V5.5 MultiCopter Flight Control Controller Board

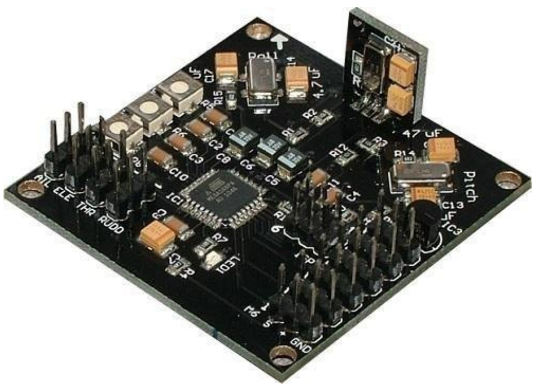


Figure 3.16: KK V5.5 flight control board

The KKmulticopter v5.5 board is a flight control board for remote control multicopters with 2, 3, 4 and 6 rotors. Its purpose is to stabilize the aircraft during flight. To do this it takes the signal from the three gyros on the board (roll, pitch and yaw) and feeds the information into the Integrated Circuit (Atmega IC). This then processes the information according the KKmulticopter flash tool software and sends out a control signal to the Electronic Speed Controllers (ESC’s) which are plugged onto the board and also connected to the motors. Depending upon the signal from the IC the ESC's will either speed up or slow down the motors in order to establish level flight. It is easily interfaceable. Auto tuning is done during connection.

3.2.1.1 KK Pin Out

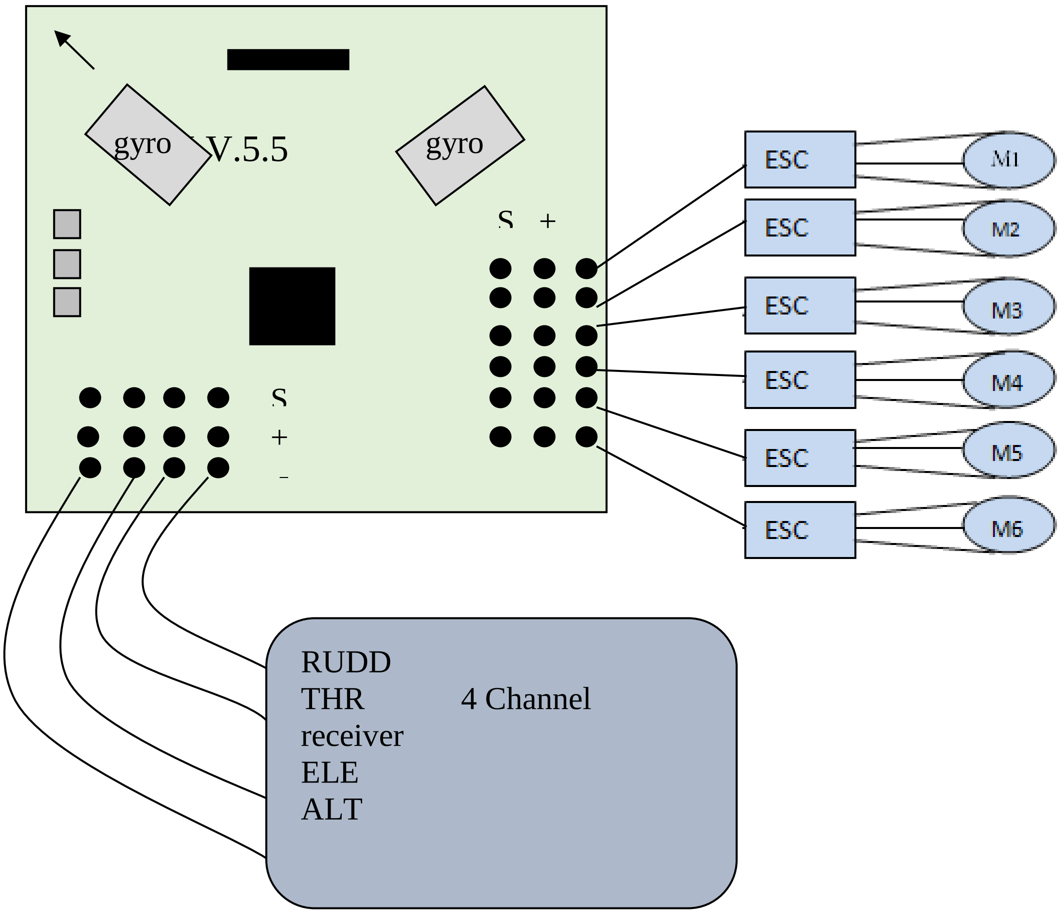


Figure 3.17: KK pin out block diagram

3.2.1.2 Motors/ESC's

Down in the corner there are 6 motor outputs (M1 through M6) On a Quadcopter the ESC's are plugged in as such:

M1 - Front motor CW

M2 - Left motor CCW

M3 - Right motor CCW

M4 - Back motor CW

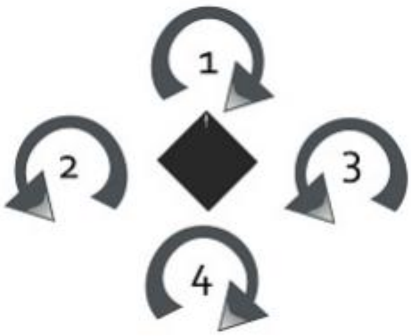


Figure 3.18: KKQuadcopter configuration

3.2.2 R450 Quadrotor Frame with integrated PCB

Figure 3.19 : R450 Quadcopter frame with integrated PCB

The R450 is a well thought out 450mm quad frame built from quality materials. The main frame is glass fiber while the arms are constructed from ultra durable polyamide nylon. This version of the R450 features integrated PCB connections for direct soldering of your ESCs. This eliminates the need for a power distribution board or messy multi-connectors keeping your electronics layout very tidy. Q450 also comes with stronger moulded arms, so no more arm breakage at the motor mount on a hard landing.

Assembly is a breeze with pre-threaded brass sleeves for all of the frame bolts, so no lock-nuts are required. It utilizes one size of bolt for the entire build, making the hardware very easy to keep in order and only requiring one size of hex wrench to assemble.

A great feature of this frame is the large mounting tabs at the front and rear of the main frame bottom plate for mounting cameras or other accessories. This makes for a great way to take aerial video or fly FPV without the need to add any additional mounting brackets. The Q450 also features colored arms (2 white and 2 red) which are great for orientation. It helps to keep you flying the right direction without the need for different colored props.

3.2.3 Self-Locking Counter Rotating Propeller

Figure 3.20: Self looking counter rotating propeller

The DJI Phantom Self-Tightening Propellers are a set of two 9" propellers compatible with the DJI Phantom 2 Quadcopter. Both propellers are rotating props while one is clockwise (CW) and the other is counter-clockwise (CCW). These DJI style Self-Tightening Propellers are specially designed to be used with the DJI Phantom 2. They have built-in threads to allow self tightening while in flight and require a special spanner to tighten before flight. The self-tightening design avoids prop becoming loose during flight. These props will only work with the right and left handed threaded motors.

3.2.4 APM uBlox Neo7M GPS with Compass

Figure 3.21: APM uBlox Neo7M GPS with Compass

This new design incorporates the HMC5883L digital compass, providing a convenient method of mounting the compass away from sources of interference that may be present in the confines of the vehicle. This new design incorporates the HMC5883L digital compass, providing a convenient method of mounting the compass away from sources of interference that may be present in the confines of the vehicle. An excellent solution for multicopters and rovers is where GPS accuracy is paramount. It features active circuitry for the ceramic patch antenna, rechargeable backup battery for warm starts, and I2C EEPROM for configuration storage. It has 10Hz update rate. It has 3V Lithium backup battery. And a low noise 3.3V regulator. It also has a water resistant casing.

3.2.5 RC Brushless Motor 2212 1000KV

Figure 3.22: RC Brushless Motor 2212 1000KV

Extremely Powerful RC Brushless motor in small size, specialized for multi-rotors. Includes prop adaptor to suite variety of props, supports pusher or puller props (Counter rotating props). To be used directly with propellers. It has 980 RPM/V no load speed with free load current of 2.5A.

3.2.6 30A Brushless ESC with SimonK firmware

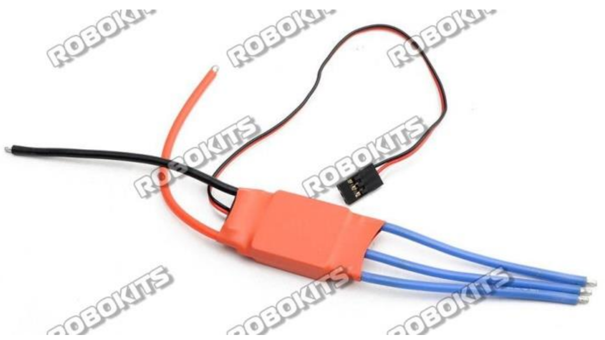


Figure 3.23: 30A Brushless ESC with SimonK firmware

Brushless Electronics Speed Controller with SimonK firmware flashed for high speed control of brushless motors. This is recommended ESC for using with multirotor aircrafts like quadcopter, hexcopter and octocopter. It can be used with 3 or 4 Cell Li-Po, also programmable for throttle range. It uses an input voltage of 6-16.8V. It has a battery eliminator circuit to reduce the voltage down to 5V 2A. The running current of the device is 30A.

3.2.7 6 ch 2.4GHz transmitter & receiver 1Km range

Figure 3.24: 6 ch 2.4GHz transmitter & receiver 1Km range

This is FS-CT6B 6ch 2.4GHz transmitter & receiver. It has 0.8W transmitter with range up to 1km line of sight. It has 6 Channels and works at 2.4GHz frequency. It is a type 2 remote control as the throttle is on the left side. The transmitter has a power of 0.8W. It is computer programmable.

3.2.8 Lithium Polymer (Li-Po) Rechargeable Battery 11.1V 2200mAH 25C

Figure 3.25: Lithium Polymer (Li-Po) Rechargeable Battery 11.1V

Lithium-Polymer (Li-Po) Battery is used for robotic applications. It can give great instantaneous discharge current up to 55A. Very light weight and small size compared to Ni-Cd, Ni-MH and Lead acid batteries. It has a very long life without losing charging capacity. It weighs just 167 gm. Full charge will be completed in180 minutes with special charger and the maximum charging current is 1A. It has long life capacity for up to 1000 charge cycles.

Li-Ion batteries are very sensitive and can get damaged easily and permanently if not used properly. Charging batteries with non-standard chargers will ensure reduction in battery life and efficiency. If the batteries are drained beyond their discharge capacity they will get heated and will get damaged permanently.

CHAPTER 4

EXPERIMENTS AND RESULTS

4.1 OBSTACLE AVOIDANCE

Ultrasonic sensors were used for obstacle avoidance**.** An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

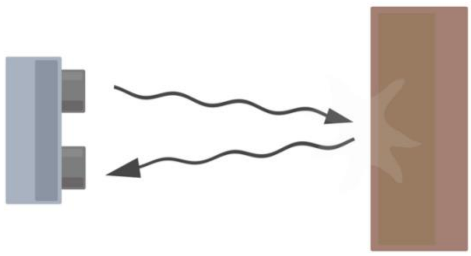


Figure 4.1 :*Diagram of the basic ultrasonic sensor operation*

$$distance = \frac{speed\ of\ sound\ \times time\ taken}{2}$$

Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave. Round-trip means that the sound wave travelled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object

AND the 'trip' from the object to the Ultrasonic sensor (after the sound wave bounced off the object). To find the distance to the object, simply divide the round-trip distance in half.

4.1.1 Experiment 1

In the initial phase of this project an ultrasound sensor where placed on top of a servo motor and made to sweep from left to right stopping at every 30 degrees for taking measurements and used stop search go method for navigation. The main role of the sensor was to avoid the collisions with dynamic obstacles coming into the path such as walking humans.

Figure 4.2: Prototype used for testing

After the implementation of complete sensor based navigation, robot was deployed to different environments. But it was seen that the navigation of the robot fails when it approaches the corners of rooms at certain angles. Also, it was found that robot fails to avoid dynamic obstacle approaching form front corners. This problem is carefully studied later and found out that the coverage area of front sensor, which is 30-degree cone is inadequate for detecting dynamic obstacles coming in front of the robot.

Another problem faced was that the robot was slow in taking decisions as it was using stop search go method. After a research on the propagation pattern and reflection pattern of the ultrasound produced from this particular sensor, it is found that the proper reflection of waves which reaches the sensor only happened if the obstacle orientation is perpendicular to the sensor. Most of the cases this condition is not attained in the indoor navigation.

Several solutions for eliminating above mentioned problems had found out, which includes increasing the number of sensors, changing the positioning of sensors, reducing the velocity of the robot and stop-search-go method of navigation. Increasing the number of sensor is an easy way out for this problem, but it makes the robot expensive and the algorithm complexity increases. Also the pin constraints of Arduino also make this option difficult to implement. Reducing the velocity of robot will allow sensor to take multiple values at same point which decreases the probability of error detections, but at the expense of increase in operation time of the robot for mapping the environment. Stop-search-go method is a widely used technique where robot travels a specific distance, then rotates in clockwise and anticlockwise directions to find the most obstacle free path around and continuing navigation in that path, but it also increases the time of operation.

4.1.2 Experiment 2

Finally the choice of rearranging the sensor orientation is adopted and sensor in left and right sides were moved to front-left and front-right corners of the robot. It increased the coverage area to 120 degrees around the front side of the robot. This setup made navigation error free in all kind of environments. Robots ability to solve situations like corners and narrow passages and avoiding dynamic obstacles are increased considerably.

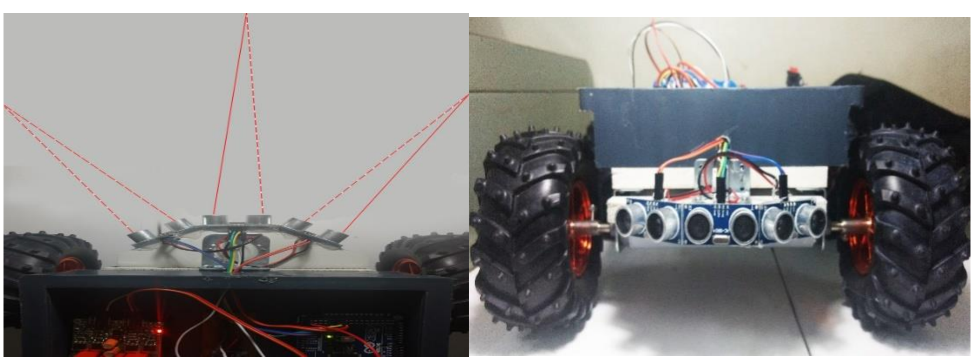


Figure 4.3: Robotic platform after sensor rearrangement

Table 4.1: Rules for obstacle avoidance

Rule	Left	Center	Right	Speed	Rotation
1	Near	Near	Near	Slow	Right big
2	Near	Near	Far	Medium	Right small
3	Near	Far	Near	Slow	No turn
4	Near	Far	Far	Fast	Right small
5	Far	Near	Near	Medium	Left small
6	Far	Near	Far	Slow	Right big
7	Far	Far	Near	Fast	Left small
8	Far	Far	Far	Fast	No turn

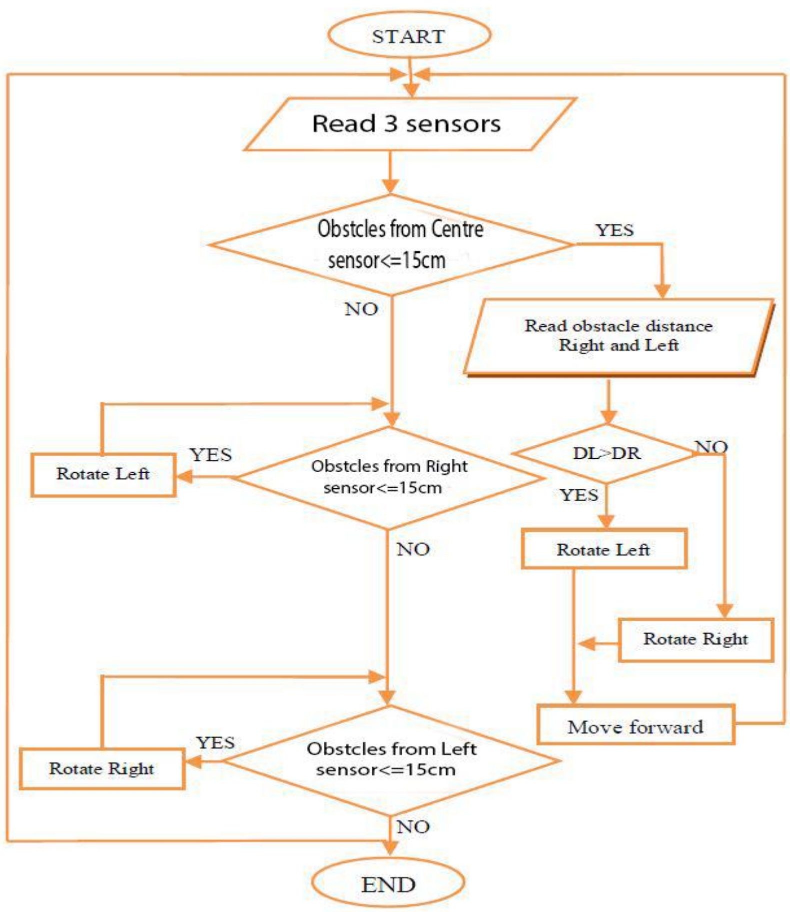


Figure 4.4 : Obstacle avoidance algorithm

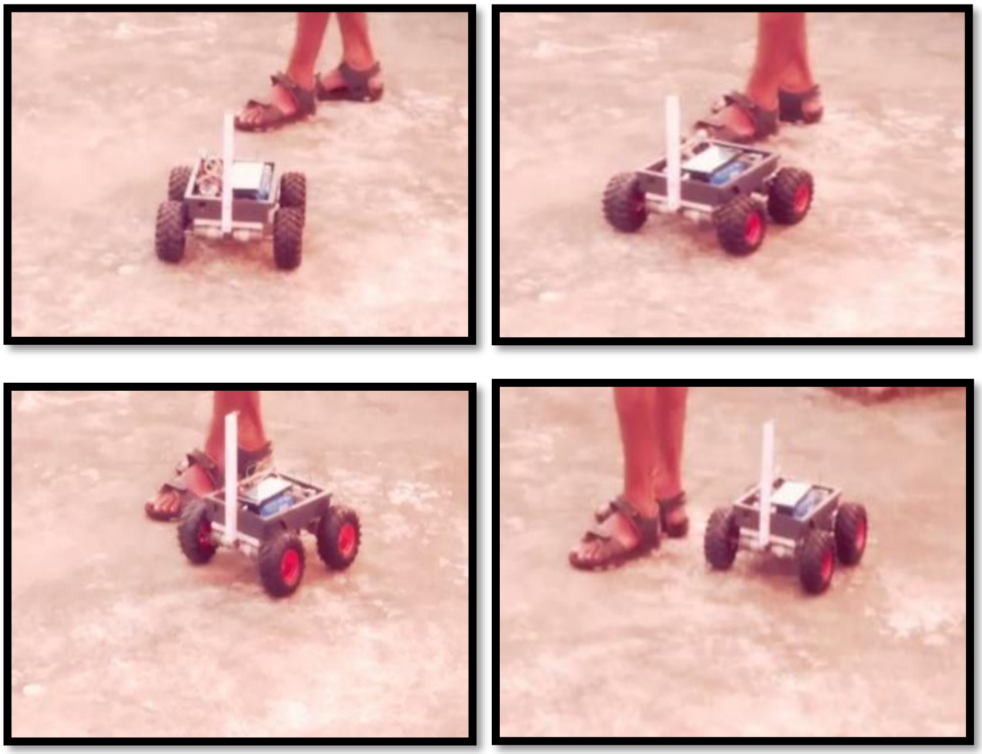


Figure 4.5: UGV avoiding obstacle

4.2 PATH PLANNING

4.2.1 GPS data acquisition

The GPS data was taken from a cell phone. The smart phones use another method for navigation, which is called A-GPS. “A” stands for assisted. As the name implies, A-GPS is assisted to provide accurate and fast feedback to the user on his location. A-GPS uses cellular network to communicate with satellites. A-GPS service is considered to be more reliable than regular GPS, especially when the environment is not quite navigation-friendly. These makes getting GPS data from smart phones more accurate and reliable.

The GPS data from the smart phone is transmitted to the Microcontroller via HC 05 Bluetooth module using a custom made android mobile application. The mobile application transmitted the latitude and longitude of the current location with a precision of about 5 decimal places. The transmitted data from the phone reached the micro controller as a string and it was parsed to extract the values from it.

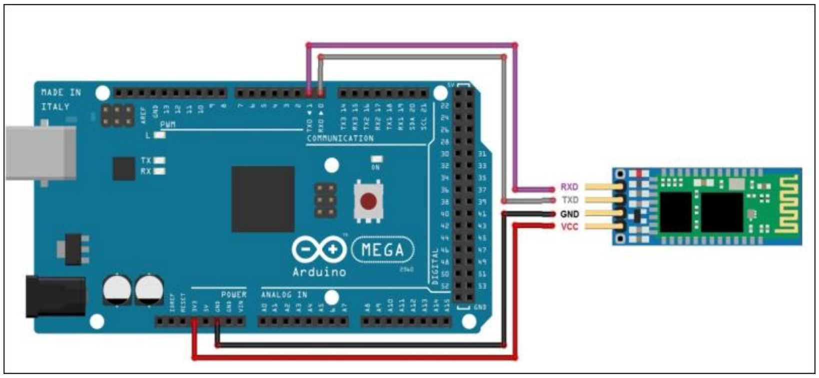


Figure 4.6: Using TX0 and RX0 for communication

The received data from the mobile was through the serial port of the micro controller. This serial port is required to communicate with the raspberry pi which acts as a master controller in the UGV. Using the same port for reading the data from Bluetooth and master slave communication was a problem as it made the communication of the Arduino with the raspberry pi difficult. So another communication port of Arduino mega was used since it offered a total of four communication channels. This port enabled the Arduino mega to communicate properly with the raspberry pi.

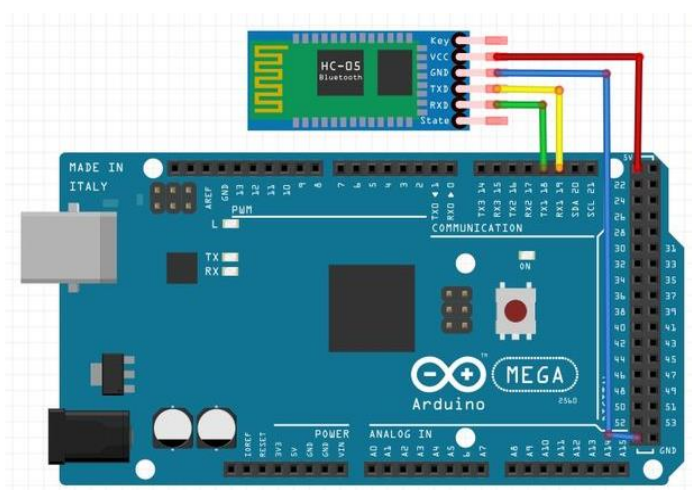
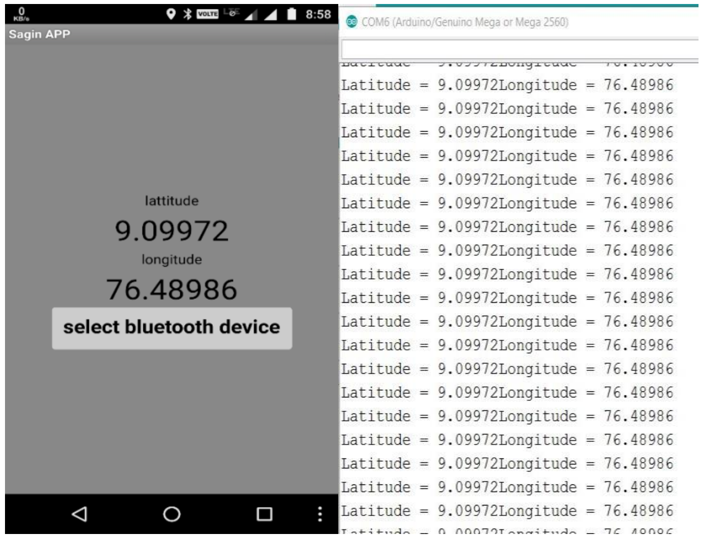


Figure 4.7: Using TX1 and RX1 for communication



(a) (b)

Figure 4.8 : GPS data (a) mobile app (b) received in arduino

The distance and heading towards the target from the current position of the UGV is calculated using Haversine formula. The formula calculates the great-circle distance between two points – that is, the shortest distance over the earth's surface

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \mathrm{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

This formula is for the initial bearing (sometimes referred to as forward azimuth) which if followed in a straight line along a great-circle arc will take you from the start point to the end point:1

$$\theta = \mathrm{atan2}(\sin\Delta\lambda \cdot \cos\varphi_2 , \cos\varphi_1 \cdot \sin\varphi_2 - \sin\varphi_1 \cdot \cos\varphi_2 \cdot \cos\Delta\lambda)$$

φ_1,λ_1 is the start point, φ_2,λ_2 the end point ($\Delta\lambda$ is the difference in longitude)

4.2.2 Compass Data

The GPS works great for providing accurate location data, but the scale on which this project operates is too small for it to provide accurate heading information. So a digital magnetometer was use for calculating the heading for the robot. The "compass" provides a super-fast readout of the robots current heading. With the current location from the GPS and the current heading from the compass the course to our destination and which way to turn (left/right) to intercept the target course was calculated.

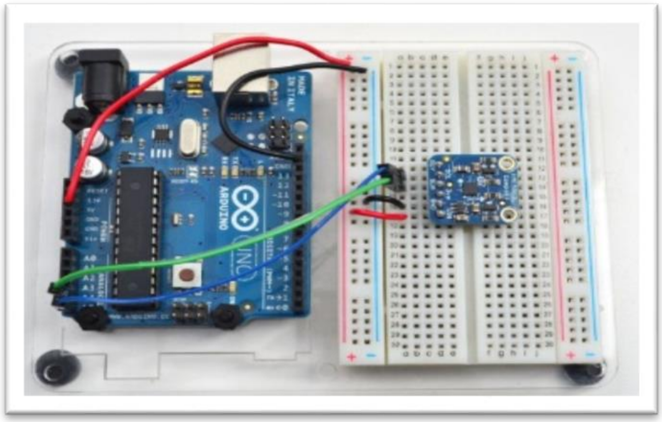


Figure 4.9 : Mangenetometer

The magnetometer is very sensitive to electrical interference, so originally it was mounted on a breadboard as far from the DC motors as possible to avoid the main culprits of interference. Unfortunately, there was still too much interference, leading to inaccurate and random compass readings. Finally the magnetometer was mounted on a pole sitting about 6" above the car; that seemed to work well.

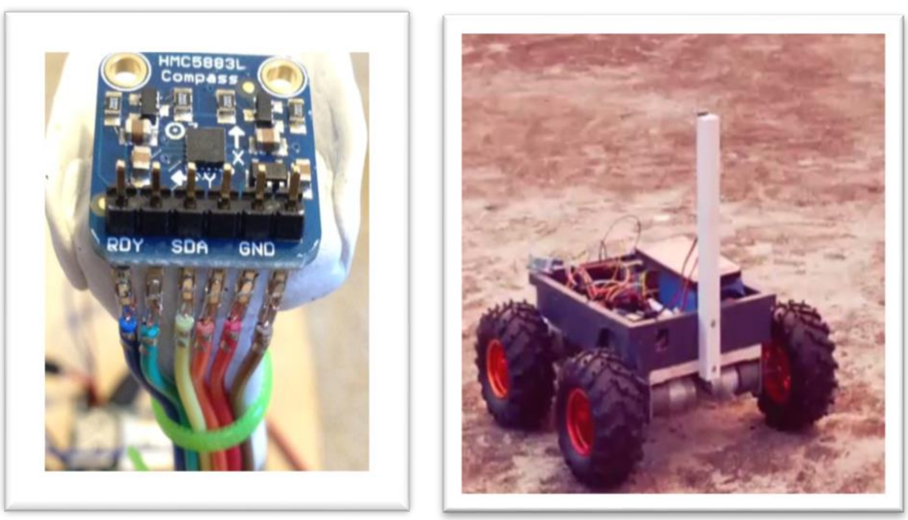


Figure 4.10 : Compass mounted on a pole of 6 inches high

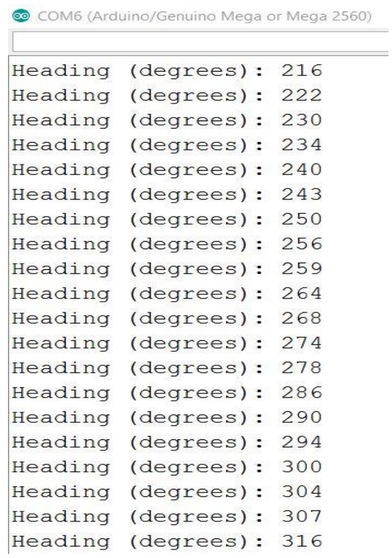


Figure 4.11 : Magnetometer data received in arduino

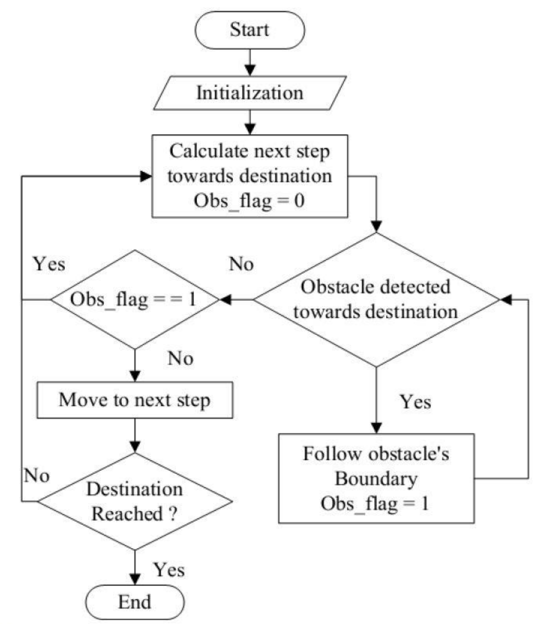


Figure 4.12 : Intelligent Bug Algorithm

4.2.2.1 Result

The UGV was successful in travelling from its current GPS location to its target GPS location using intelligent bug algorithms and navigated through an environment with static as well as dynamic obstacles.

Figure 4.13: Navigation from point A to B

4.3 Human Detection using Haar Cascade Classification

Haar cascade procedure is developed from the concept proposed by voila johns paper on rapid object detection using haar like features. In this procedure, we train a classifier using a few hundred sample views of a particular object called positive samples. For efficiency and to increase the speed of training, images are scaled down to small dimensions. Here training is done with 50x70 size images. This is one factor that decides the training time the machine would take. And we also need a few hundred negative image to test and verify computation procedure, we started with around 40-50 positive images and a few hundred negative images. The cascading (multiple stages of classifying an ROI) classifier is trained such that it detects a specified region that bounds the object of interest. The classifier is designed such that it can be easily resized to detect features similar to those marked in the samples provided for training. So, to find the object of interest, the image has to be scanned for different scales of classifier.

4.3.1 Procedure for training classifier

4.3.1.1 Step1: Collecting Positive and Negative samples.

I had to take about 50 snaps of faces at slightly changed angles and different background and saved to a folder. Around 100 samples of negative images were also captures and saved to a separate folder.

4.3.1.2 Step 2: cropping and scaling images

Positive images: We cropped each image carefully to have a smaller image containing face. Make sure to keep the dimensions of the cropped images constant, or maintain a constant ratio for every image, because every image will be scaled down to much smaller dimensions of size 50x70. Save the final images in positive_images folder.

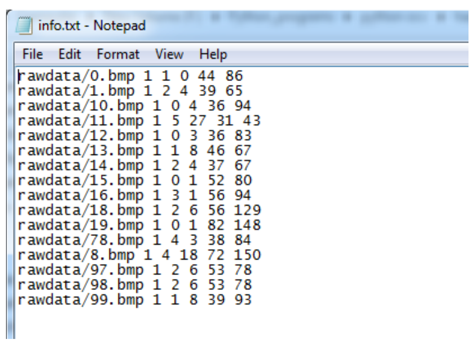
Negative Images: We need to simply scale these images down to same size as positive images. And store them in a folder called negative_images

4.3.1.3 Step 3: Converting to BMP

One of the tools we use needs the images to be in bmp format). The images we captured using camera were in jpeg format, so a fast way of converting them all to BMP was using this free software called "Total Image Convertor" . Both positive and negative images should be converted to BMP format.

4.3.1.4 Step 4: Selecting ROI

We use the "objectmaker" tool to mark the object of interest. This will help us creating a text file which contains the image path, and the will be used to make a vector file.


info.txt - Notepad

File Edit Format View Help

```
rawdata/0.bmp 1 1 0 44 86
rawdata/1.bmp 1 2 4 39 65
rawdata/10.bmp 1 0 4 36 94
rawdata/11.bmp 1 5 27 31 43
rawdata/12.bmp 1 0 3 36 83
rawdata/13.bmp 1 1 8 46 67
rawdata/14.bmp 1 2 4 37 67
rawdata/15.bmp 1 0 1 52 80
rawdata/16.bmp 1 3 1 56 94
rawdata/18.bmp 1 2 6 56 129
rawdata/19.bmp 1 0 1 82 148
rawdata/78.bmp 1 4 3 38 84
rawdata/8.bmp 1 4 18 72 150
rawdata/97.bmp 1 2 6 53 78
rawdata/98.bmp 1 2 6 53 78
rawdata/99.bmp 1 1 8 39 93
```

Figure 4.14: A vector file

4.3.1.5 Step 5: Creating a bat file to construct a vector file from POSITIVES.txt

Open a new notepad file and type this:

F:\Python_programs\opencv\build\x64\vc10\bin\opencv_createsamples.exe -info POSITIVES.txt -vecsamples1.vec -num 100 -w 50 -h 80 PAUSE

And store this as "create_vec_from_data_file.bat"

The file should be pointing to an exe file called opencv_createsamples.exe available in the “build>x64>vc10>bin” path inside the extracted OpenCV folder. Adjust –w and –h parameters according to the max width and height of our samples. Running this file should create a .vec file named “samples1.vec”

4.3.1.6 Step6: Combining the samples1 and negative file to train a cascade

We use the samples1.vec file as reference to how the object of interest is looks, in the positive image collection and train the cascade to learn the features of the mentioned object in all the positive images. We have 3 stages of training of this cascade, and at the end of each stage, the progress will be stored in a folder named classifier, so we have to create a folder and name it classifier.

Once we are done with this preparation, run the bat file from command prompt preferably. Ensure that we are not running any other application that would take significant memory and power from CPU. We used Intel i5 processor overclocked at 3 GHz with a RAM of 8 GB to perform the training task. It took a total of 27 hours to finish the training of classifier.

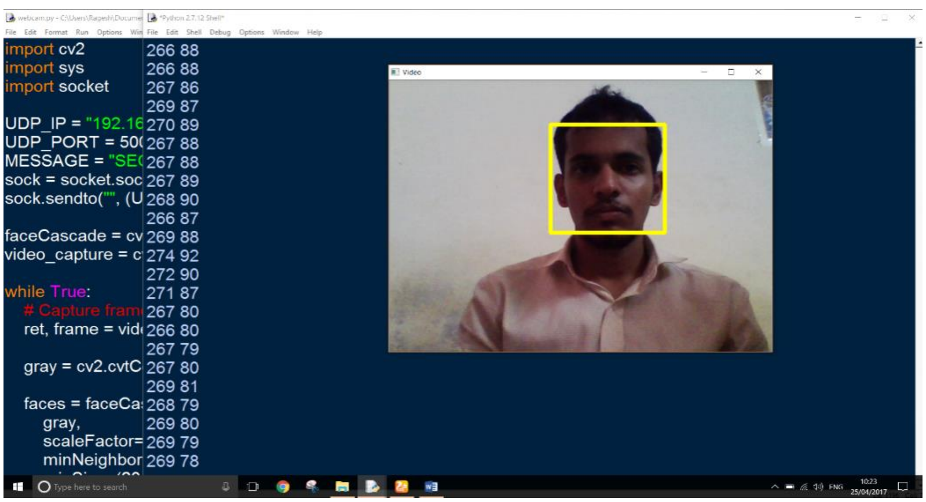


Figure 4.15: Target at a distance less than 1 metre

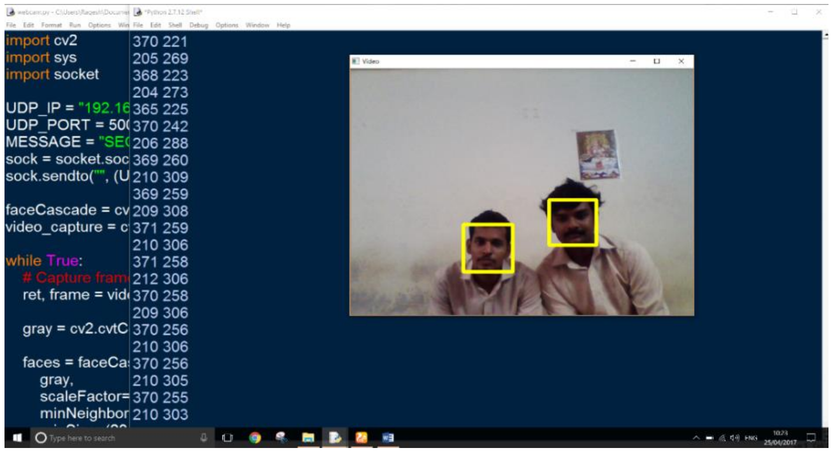


Figure 4.16: Multiple targets at 1 metre

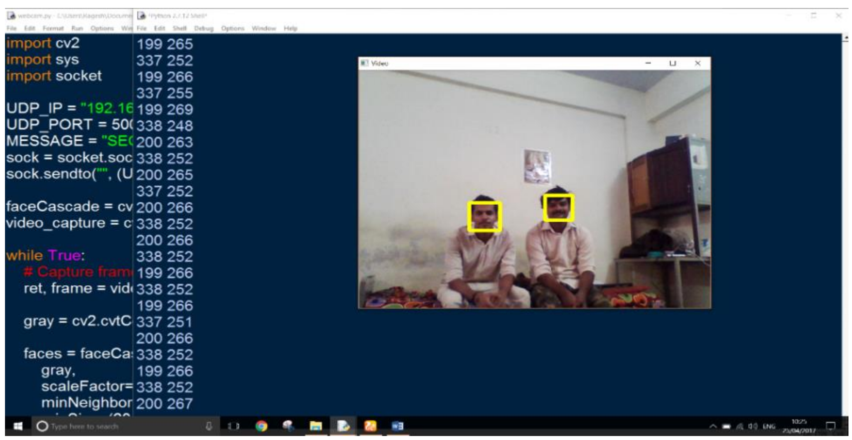


Figure 4.17: Multiple targets at 3 metres

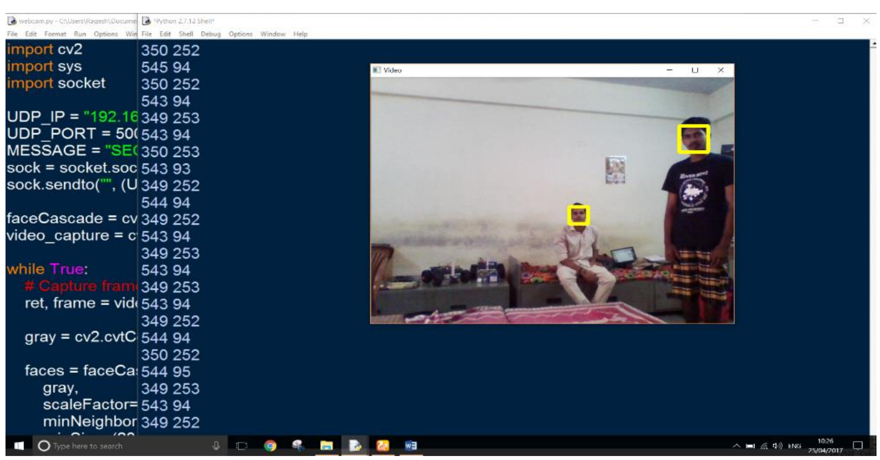


Figure 4.18 : Multiple targets at 5 metres and 3 metres

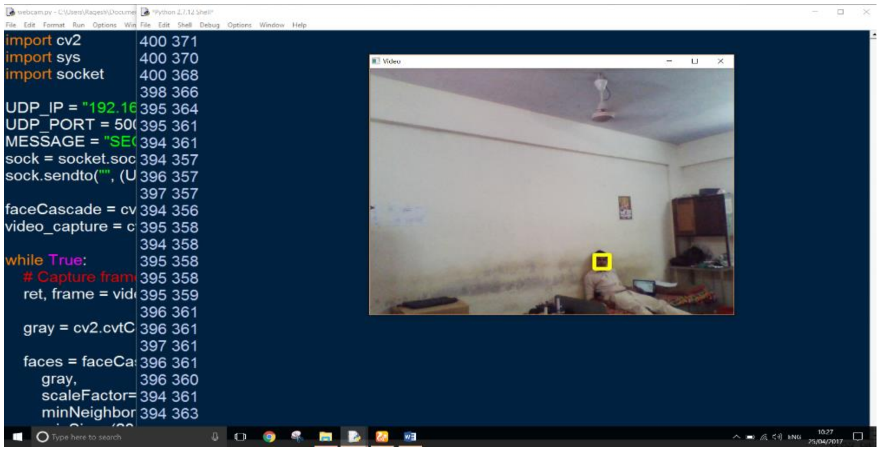
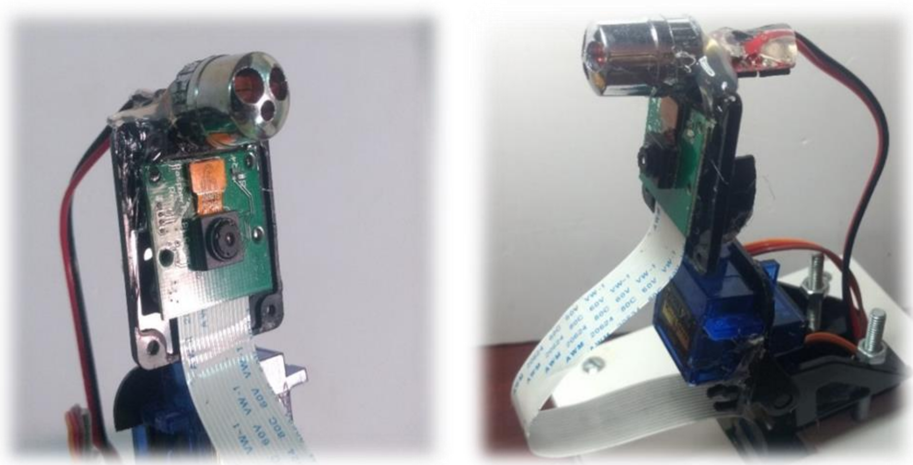


Figure 4.19: Target at 7 metres

Once a target is detected the target is tracked using pan and tilt mechanism over which the Pi camera is fitted. The target is aimed with a laser pointer that lights up once a target is detected and make sure that the target is tracked properly.



(a) (b)

Figure 4.20 : (a) Laser pointer for target tracking (b) Pan and tilt mechanism for camera

4.4 Communication

UGV and UAV have to communicate with each other and exchange information about its GPS location, detection of target, location and direction towards the target. Wi-Fi modems inside the Raspberry Pi were used for communication. So in order to communicate with each other all the UGV and UAV is connected to the same Wi-Fi network through an external modem. This modem is then connected to a laptop in the base station which is used to alert the soldiers about the potential threat. UDP protocol is used for the transmission of data. Since the same data has to be transmitted to more than one host UDP Multicasting was used. The soldiers can also view the aerial footage of the threat scene shot by the UAV, which will be transmitted to the laptop via Wi-Fi. All this information provided by the UAV as well as the UGV gives the soldier a clear picture of the entire situation and could make him aware about the situation.

4.4.1 Multicasting

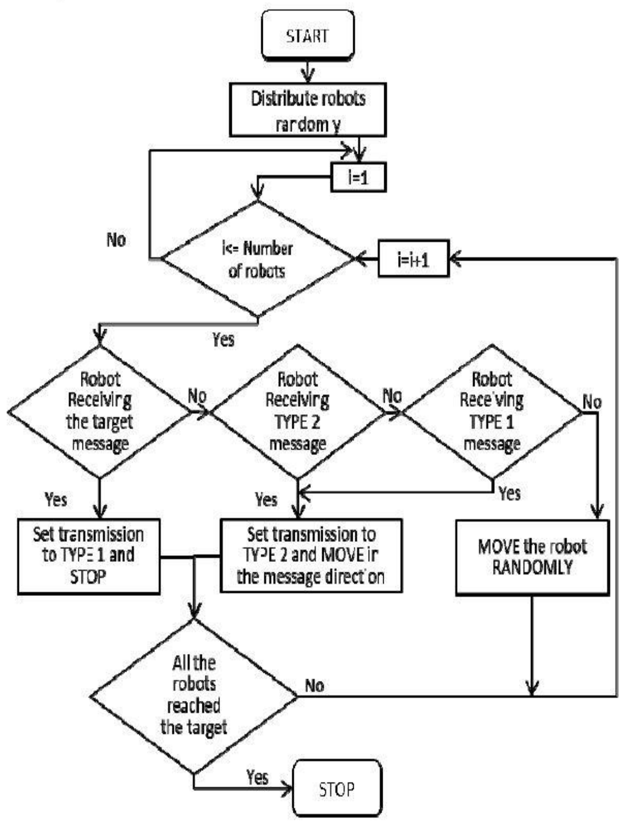


Figure 4.21: Flowchart for foraging algorithm of swarm robots

Multicast is a kind of UDP traffic similar to broadcast, but only hosts that have explicitly requested to receive this kind of traffic will get it. This means that you have to join a multicast group if you want to receive traffic that belongs to that group. Multicast is useful when you have to transmit the same message to more than one host. Usually the client that sends multicast does not know how many servers will really receive his packets. When talking about client-server in network, the client sends the request; the server receives the request and might send back an answer.

Since multicast is based UDP, the transmission is by default not reliable. The advantage of using multicast instead of broadcast is that only interested hosts will get the message, and the message should be transmitted only once for many clients (saves a lot of bandwidth). Another advantage is the possibility of sending packets larger than interface MTU.

The robots in the swarm communicate with each other by a predefined message protocol of assigning priorities**.** It is necessary for the robots in the swarm to communicate for foraging, which is assumed to be a combination of collective exploration and gathering. Each robot is strategized to transmit any of the three messages, also called message protocols hitherto.

Type 1: ‘Target found.’

Type 2: ‘Penultimate Target found.’

Type 3: ‘Nothing to say.’

4.4.1.1 Method of operation

The UGV that first detects the target multicast its GPS location and the location as well as the direction towards the target. Other UGV receives this data and navigate towards the desired GPS location using their onboard GPS and magnetometer. The UGV’s transmit three types of messages and based on the priorities of the message type actions are excecuted.

In order to make the UDP multicasting reliable we ended up transmitting same messages five times and then performing UNION operation to the five messages to extract error free message. This method helps resolve the reliability issue of the UDP communication to an extent.

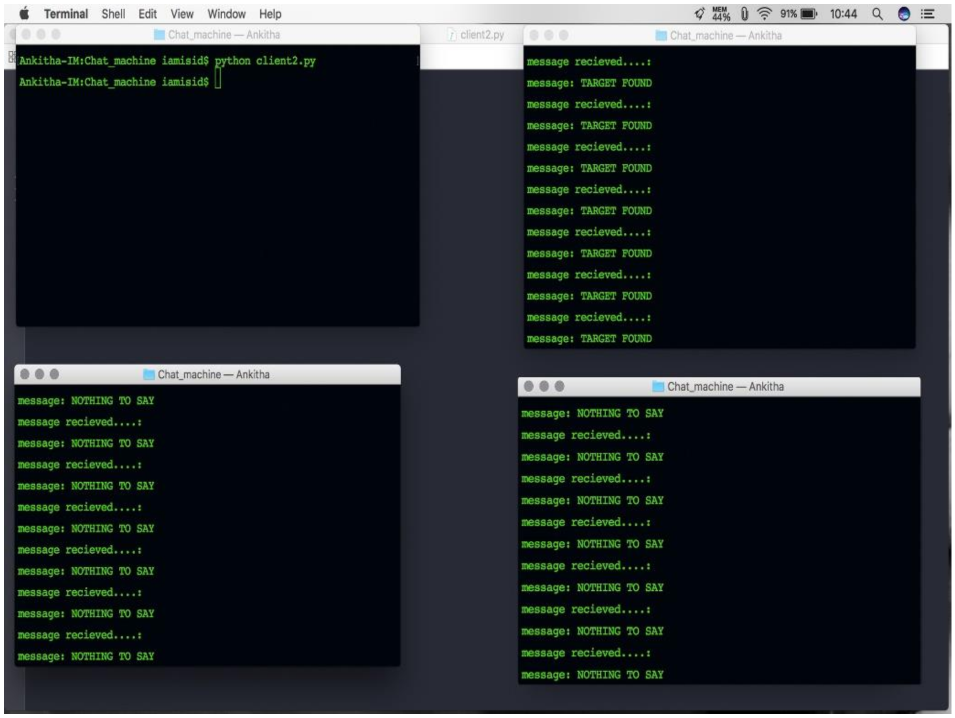


Figure 4.21: Target detected by only 1 UGV

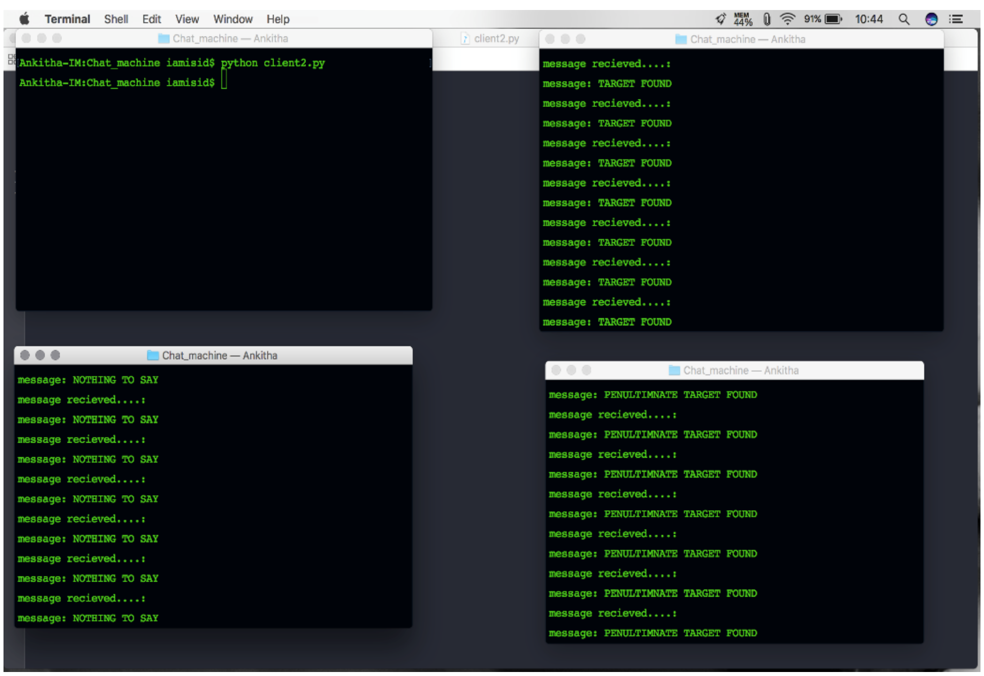


Figure 4.22 :Target detected by 2 UGV's

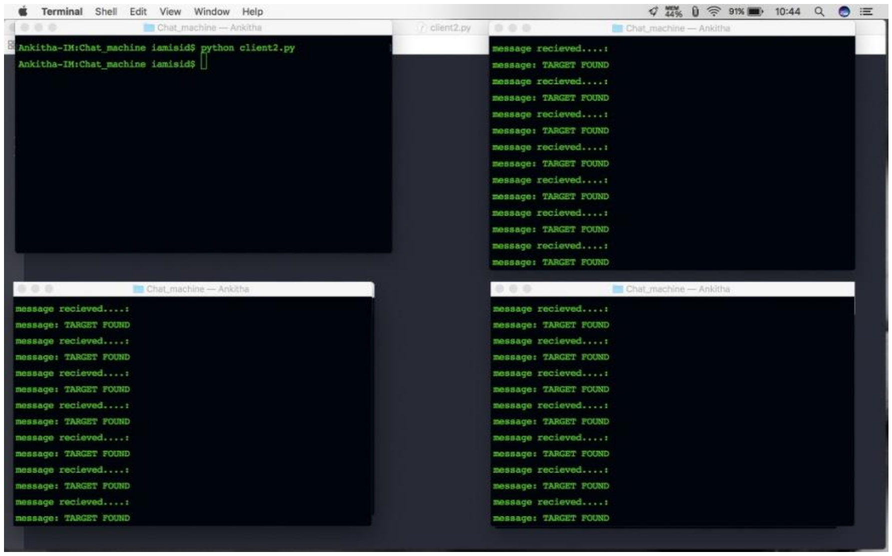


Figure 4.23: Target detected by all three UGV’s

4.5 BASE STATION

The soldiers in the base station will be alerted by flashing a security alert message along with a siren. The GIF image of the message was loaded to the system along with the mp3 file of siren sound. This will be activated once the target is detected by any one of the UGV. This gives them ample time to respond to the situation thereby reducing the need to stay awake throughout the day. This could possibly reduce the stress faced by the soldiers in the borders and also reduce the threat to their life. Thus this project could act as a First line of Defence.

Figure 4.23: The alert message flashed in the base station

4.6 The final prototype

Figure 4.24: Swarm of Unmanned Ground Vehicles

Figure 4.25 : Unmanned Ground Vehicle

Figure 4.26: Unmanned Aerial Vehicle

CHAPTER 5
CONCLUSION

We were able to successfully address all the problems stated in the problem statement through this project. The project was meant to assist the Border Swarm Force in detecting intruders entering the secured areas. This project could eliminate the need for the soldiers to stay vigilant throughout the time which is an impossible task for humans. It is this reason that motivated us to take up this project. A robot could do what a human cannot do, so we came up with the idea of using robots that can assist the soldiers. But still it was found out that a single robot cannot assist the soldiers to a great extent. So we came up of with the idea of a multi robot system for addressing the issue. The multirobot system we developed consist of three unmanned ground vehicles for detection and tracking of intruders and a Unmanned Aerial Vehicle which is capable of transmitting high definition video to the soldiers base station via Wi-Fi network.

The prototype we developed was able to detect the presence of intruders and was successful in tracking the movement of intruders. The communication network worked great but had little issues with the operational range. The target detection cameras performed poorly under low light conditions however this could be improved by the usage of IR night vision enabled cameras. The detector also failed when the intruders were camouflaged. These situations could only be avoided by the usage of thermal cameras. The unmanned aerial vehicle was used from a previous year project which was properly stabilized and could take a payload of maximum 1kg without loosing its stability. Raspberry Pi along with a Pi camera was added to the UAV in order to communicate with the multirobot system as well as transmission of high definition aerial images during a mission.

In the navigation which only uses ultrasound sensors, narrow obstacles such as legs of tables and chairs, will not be properly detected hence leading to collision. Use of LIDAR systems could improve the accuracy and speed but it is less impressive for large scale implementation due to its high cost.

Future Works

1. This prototype can be implemented directly under the Border Surveillance of BSF if it is furnished in a basic level
2. Dedicating this project to the Border Security Force and Central Industrial Security Force after some advancements for implementing in real time applications.
3. Developing a better method for secured and more reliable communication between the mobile agents.
4. Using thermal cameras for target detection as well as to check whether the intruder is armed.

APPENDIX

Python code for human detection and tracking

```
import cv2
import sys
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import serial
from optparse import OptionParser
import math
import datetime
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)
GPIO.setup(11,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
servo1=GPIO.PWM(11,50)
servo2=GPIO.PWM(13,50)
servo1N=45
servo2N=40
def angle(dutyCycle):
_angle=float(dutyCycle)/10+2.5
return _angle

pos1=angle(servo1N)
pos2=angle(servo1N)

faceCascade = cv2.CascadeClassifier("haarcascade_frontalface.xml")
profileCascade = cv2.CascadeClassifier("haarcascade_profileface.xml")
camera = PiCamera()
camera.resolution = (320, 240)
camera.framerate = 60
rawCapture = PiRGBArray(camera, size=(320, 240))
screenx=160
screeny=120
servo1.start(angle(servo1N))
```

```
servo2.start(angle(servo2N))

for frame in camera.capture_continuous(rawCapture, format="bgr",
use_video_port=True):
    image = frame.array

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags=cv2.cv.CV_HAAR_SCALE_IMAGE
    )
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 255), 2)
        pt1 = (int(x), int(y))
        pt2 = (int((x + w)), int((y + h)))
        x1 = pt1[0]
        x2 = pt2[0]
        y1 = pt1[1]
        y2 = pt2[1]
        normal=pt2
        centrex = x1+((x2-x1)/2)
        centrey = y1+((y2-y1)/2)
        centre = (centrex, centrey)
        print centre
        if(pos1>0 and pos2>0):
if(centrex<screenx and centrey<screeny):
move1=pos1 + .2
servo1.start(move1)
time.sleep(.025)
pos1=move1
move2=pos2 - .2
servo2.start(move2)
time.sleep(.025)
pos2=move2
if(centrex>screenx and centrey>screeny):
move1=pos1 - .2
```

```
servo1.start(move1)
time.sleep(.025)
pos1=move1
move2=pos2 + .2
servo2.start(move2)
time.sleep(.025)
pos2=move2
if(centrex<screenx and centrey>screeny):
move1=pos1 + .2
servo1.start(move1)
time.sleep(.025)
pos1=move1
move2=pos2 + .2
servo2.start(move2)
time.sleep(.025)
pos2=move2
if(centrex>screenx and centrey<screeny):
move1=pos1 - .2
servo1.start(move1)
time.sleep(.025)
pos1=move1
move2=pos2 - .2
servo2.start(move2)
time.sleep(.025)
pos2=move2
    cv2.imshow('Video', image)
    cv2.moveWindow('Video' ,0, 0)
    rawCapture.truncate(0)

    if cv2.waitKey(1) & 0xFF == ord('q'):
GPIO.cleanup()
break

cv2.destroyAllWindows()
```

BIBLIOGRAPHY

1) Shilum and C. Thomasas, “Face detection using haar features,” *IEEE*, vol. CVPR94, pp. 01–07, 1994.

2) M.P, “Outdoor mobile robot localization,” *USCE*, vol. 02, pp.714–719, 2010.

3) R. Talix and S. Glaser, “Accuracy in obstacle localization,” *IEEE*, vol. 02, pp. 18–21, 2002.

4) A Novel Strategy to Navigate Mobile Robots Autonomously International Journal Of Systems Applications, Engineering & Development Issue 2, Volume 5, 2011

5) International Journal of Emerging Trends in Engineering Research, Volume 3, No.4, April 2015, GPS-based Navigated Autonomous Robot

6) G. Bradaski and A. Kachler, *Learning OpenCV*. OReilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2008.

7) Daniel and L. Baggio, *Mastering OpenCV with Practical Computer Vision Projects*. Livery Place 35 Livery Street Birmingham B3 2PB, UK., 2012.

8) M. Peris. (2011, Jun.) Computer vison@ONLINE. [Online]. Available:

9) Fritsl. (2008, Jun.) Lets make robot@ONLINE. [Online]. Available: