SMART ASSISTIVE CANE FOR THE BLIND

A Major Project submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL ENGINEERING

By

Abhishek Sharma	(2K11/EE/003)
Abhishek Thakur	(2K11/EE/004)
Amit Singh	(2K11/EE/008)
Ankur Taparia	(2K11/EE/013)

Under the guidance of

Mr Ram Bhagat

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DELHI-110042

MAY 2015

CERTIFICATE

This is to certify that the project titled **'SMART ASSISTIVE CANE FOR THE BLIND'** which is being submitted by **ABHISHEK SHARMA (Roll No - 2K11/EE/003), ABHISHEK THAKUR (Roll No - 2K11/EE/004),AMIT SINGH (Roll No - 2K11/EE/008) and ANKUR TAPARIA (2K11/EE/013)** in partial fulfilment of the requirements for the award of degree – Bachelor of Technology in Electrical Engineering is a bonafide record of work carried out by them under my supervision. The matter embodied in this project has not been submitted elsewhere for the award of any other degree or diploma.

(**Mr. Ram Bhagat**)

Assistant Professor

Electrical Engineering

Delhi Technological University

ACKNOWLEDGEMENT

First and foremost, we express our sense of gratitude to our supervisor Shri Ram Bhagat, Assistant Professor, Department of Electrical Engineering, Delhi Technological University for his constant supervision and valuable suggestions for our thesis work entitled " SMART ASSISTIVE CANE FOR THE BLIND ".

We wish to take this opportunity to express our gratitude to Professor Madhusudan Singh, Head of Department of Electrical Engineering for his constant encouragement during the conduct of the project work. We express our gratitude to all the faculty members of Electrical Engineering Department for their motivation.

We also thank all the non-teaching staff of the Electrical Engineering Department for their fullest cooperation. We would like to thank all those who have directly and indirectly helped us in completion of thesis well in time.

Finally, we wish to thanks our family members for their moral and financial support during making of the project and thesis.

Abhishek Sharma (2K11/EE/003)

Abhishek Thakur (2K11/EE/004)

Amit Singh (2K11/EE/008)

Delhi, May 2015

Ankur Taparia (2K11/EE/013)

ABSTRACT

In the past fifty years there has been pathbreaking inventions in the medical arena to treat many diseases and in the field of medical/assistive technologies assisting the healthy and not so healthy in keeping track of vital body parameters like blood pressure, sugar etc. But there has been a dearth of cost effective assistive technologies that bring the visually impaired people at par with sighted persons and make them self dependent. What if a blind person no longer requires a sighted person to assist him navigate the city, board the right bus to his destination, cross the road without any assistance from others, recognize a known acquaintance in a crowd and call out his name pointing in his direction andthen move towards him and to say hello.

We intend to provide the visually impaired with a smart cane that comes as complete package sloution to the person to navigate, travel and socialise without any assistance from others. We will achieve this through a mix of various technologies like GPS navigation through streets assisted by a camera and a radar to avoid obstacles, face recognition through opencv when in community place like college, social gatherings , internet of things to assist in using public transport and other applications. The Beaglebone Black is employed to handle complex image processing algorithm to detect lane markings , detect and recognize faces in real time using opencv libraries running on Ubuntu OS. The beaglebone is employed as a Master EVM to perform control funtions for GPS, IOT ,Radar and Speech to text conversion.

Keywords—BeagleboneBlack ; GPS; image processing; visually impaired

CONTENTS

CERTIFICATE II

ACKNOWLEDGEMENT III

ABSTRACT IV

LIST OF FIGURES VIII

CHAPTER-1: INTRODUCTION 1

- 1.1: General 1
- 1.2: Objectives of Smart Assistive cane 1
- 1.3: Market analysis 2

CHAPTER-2 : LITERATURE REVIEW 4

- 2.1: GLOBAL POSITIONING SYSTEM (GPS) 4
  - 2.1.1 HISTORY OF GPS 4
  - 2.1.2 FUNDAMENTALS OF GPS 4
  - 2.1.3 COMMUNICATIONS 5
  - 2.1.4 Message Format 6
  - 2.1.5 STANDARD INTERFACING SENTENCES 7
  - 2.1.6 DECODING OF SOME POSITION SENTENCES 11
  - 2.1.7 Decode of some Navigation Sentences 12
  - 2.1.8 NEO-6 u-BLOX 6 GPS MODULE 15
- 2.2: Radar 19
  - 2.2.1 Introduction 19
  - 2.2.2 Principle of Radar 20
  - 2.2.3 Radar Observables. 21
  - 2.2.4 Radar Block Diagram 21

2.2.5 Radar Frequency bands 22

2.2.6 Radar range equation 22

2.2.7 Detection of radar pulses in noise 23

2.2.8 Propagation effects on radar performance 24

2.2.9 Doppler shift concept 24

2.3: Speech user interface 26

2.3.1 speech recognition 26

2.3.2 Text to speech 31

2.4: Digital Image Processing 35

2.4.1 Open CV – An intro 35

2.4.2 Canny edge detection 36

2.4.3 Hough transform 37

2.4.4 Bird eye view 41

2.4.5 object detection using classifier 42

CHAPTER-3: GPS and Mapping systems 44

3.1: Technology and Basic Concepts 44

3.2: Data format for exact mapping 45

3.2.1 Almanac 45

3.2.2 Emphemeris 45

3.3: Trilateration 45

3.4: Description of navigation code 46

CHAPTER-4: RADAR (HB 100 Microwave Motion Sensor Module) 48

4.1 Introduction 48

4.2 Features of a HB100 module 49

4.3 Mounting of radar and its components 49

4.4 Radiation pattern observed 49

4.5 Amplifier circuit for radar 50

4.6 Calculation of frequency using Doppler equations 50

4.7 2 Pulse MTI canceller 51

4.8 Algorithm 52

CHAPTER-5: Speech User Interface Implementation 54

5.1 Voice recognition 54

5.2 Text to Speech 56

5.3 Google Map API 57

CHAPTER-6: Image Processing Subsystem 59

6.1 General 59

6.2 Description of Algorithm 59

6.3 Software Implementation 60

6.4 Vehicle Detection Program 60

6.5 Lane Detection and mapping program 61

CHAPTER-7: RESULTS AND DISCUSSION 63

7.1 Simulation and results of radar 63

7.2 Discussion of simulation results 66

7.3 Software implementation of voice based user interface 67

7.4 Discussion of software results of voice user interface 69

7.5 Software implementation and results of image processing 69

CHAPTER-8: CONCLUSION AND SCOPE FOR FUTURE WORK 71

8.1 Conclusions 71

8.2 Scope of future work 71

REFERENCES 73

LIST OF FIGURES

S.No.	Figure No. / Description	Page No.
1	**Figure 1.1:** Tabular representation of visually impaired	3
2	**Figure 2.1.1:** Schematic of concept for Detection of position by GPS	5
3	**Figure 2.1.2:** Subparts of Standard GPS message Format	6
4	**Figure 2.1.3:** Table representing definition of GGA format.	11
5	**Figure 2.1.4:** Table representing definition of RMC format.	12
6	**Figure 2.1.5:** Table representing definition of WPL format.	13
7	**Figure 2.1.6:** Table representing definition of AAM format.	13
8	**Figure 2.1.7:** Table representing definition of BOD format.	14
9	**Figure 2.1.8:** Table representing definition of RMB format.	14
10	**Figure 2.1.9 :** Actual photo of NEO 6m Ublox GPS	16
11	**Figure 2.1.10 :** Available Protocols in GPS for communication	17
12	**Figure 2.1.11 :** Table representing various parameters of Neo 6m Ublox GPS	18
13	**Figure 2.2.1:** Principle of Radar	20
14	**Figure 2.2.2:** Block diagram	21
15	**Figure 2.2.3:** Frequency band of radar	22
16	**Figure 2.2.4:** Concept of Doppler Shift	25
17	**Figure 2.4.1:** Canny edge detection	37
18	**Figure 2.4.2:** Hough Standard transform	38
19	**Figure 2.4.3:** Plot for standard transform	39
20	**Figure 2.4.4:** Plot for Hough Transform	39
21	**Figure 2.4.5:** Detection of hough lines in an image	40
22	**Figure 2.4.6:** Bird eye view	41
23	**Figure 3.1:** Schematic explaining the Triangulation Method	46

24	**Figure 4.1:** HB100 Microwave Motion sensor module	48
25	**Figure 4.2:** Radar Mounting	49
26	**Figure 4.3:** PCB of Radar	49
27	**Figure 4.4:** Radiation pattern of radar	50
28	**Figure 4.5:** Amplifier Circuit	50
29	**Figure 4.6:** Block diagram for multiple target detection	52
30	**Figure 4.7:** Flowchart of whole algorithm	53
31	**Figure 6.1:** Flowchart of Lane Detection Program	62
32	**Figure 7.1:** Output of radar module when no object is in the field	63
33	**Figure 7.2:** Output of radar module when an object is in the field	63
34	**Figure 7.3:** Output of radar module for different frequency	64
35	**Figure 7.4:** Laboratory setup of radar module depicting frequency shift	64
36	**Figure 7.5:** Design of radar amplifier circuit on multisim	65
37	**Figure 7.6:** Simulation of radar amplifier circuit on multisim for high gain	65
38	**Figure 7.7:** Snapshot of terminal window depicting navigation command	67
39	**Figure 7.8:** Snapshot depicting on board voice recognition	67
40	**Figure 7.9:** Real time location tracking on Google Map	68
41	**Figure 7.10:** GPS coordinates to online database	68
42	**Figure 7.11:** Output of lane detection and mapping program	69
43	**Figure 7.12:** Output of Vehicle detection program	70
44	**Figure 7.13:** Output of Vehicle detection program	70

Chapter 1

Introduction

1.1 General

With the advent of cheaper computing devices and the ability to install faster processors power on smaller chips with a multitude of added functionalities like Wifi, Bluetooth it has become very easier to design Internet of Things based devices that can utilise internet's capabilities to serve wider domains and make existing devices smarter. A Smart Cane for the Blind is our effort to improve upon the existing canes that assist a Blind person in moving around and turn it into a product that provides a complete solution for mobility, navigation and emergency assistance. Existing canes provide are able to sense only the deformations in the path through touch. It basically acts as an extension of the person's body thus, widening his reach and informing him of obstacles in his path and changes in path elevation beforehand. Earlier implementations of Smart Canes have employed ultrasonic sensors to inform the person through vibration based indications.

1.2 Objectives of Smart Assistive Cane for Blind

We intend to provide the visually impaired with a smart cane that comes as complete package solution to the person to navigate, travel and socialise without any assistance from others.

- We will achieve this through a mix of various technologies like GPS navigation through streets by utilising Google Maps application programming interface.
- Maps based navigation will be assisted by a camera complex image processing algorithm to detect lane markings, detect and recognize faces in real time to assist him in navigating.
- In order to avoid obstacles, an obstacle detection subsystem is designed that employs a Doppler radar and an Ultrasonic Sensor.

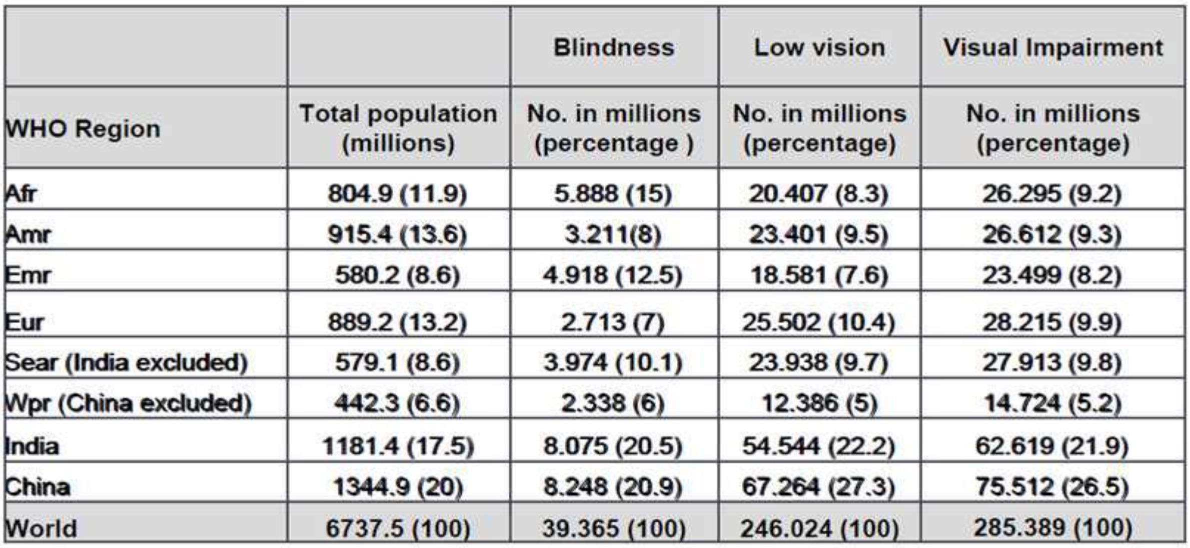
- The Blind person will interface with the cane through voice based commands using text to speech and speech to text conversions that are done on chip.
- The complete system will be powered by a set of lithium ion batteries that are rechargeable.
- This system also has a safety feature through which any authorised person like a family member can track the blind person and see his location on a map.

All these objectives serve the primary objective of the Smart Cane project that is to bring visually impaired people at par with sighted persons and make them independent.

1.3 Market Analysis

According to the World Health Organization (WHO), there are approximately 285 million people who are visually impaired worldwide of which 39 million are blind and 246 million have low vision restricting their abilities to function as a normal human being. India is home to the largest population of blind people in the world. Though our innovation will target the market of assistive technologies for the visually impaired all over the world we would make our device with a special emphasis on India because the total size of addressable market is very large owing to largest number of visually impaired in the world and there is a dearth of innovations addressing concerns other than navigation that visually impaired people face.

The existing solutions provide navigation assistance using GPS and Radar but our solution will also have an image processing subsystem that would make the innovation more accurate in navigation and provide other functionalities like recognizing known acquaintance among crowd and facilitate the movement of user towards the person. Besides we will provide another improved functionality of locating the user on a map and we can gather the real time location of the user.



		Blindness	Low vision	Visual Impairment
WHO Region	Total population (millions)	No. in millions (percentage)	No. in millions (percentage)	No. in millions (percentage)
Afr	804.9 (11.9)	5.888 (15)	20.407 (8.3)	26.295 (9.2)
Amr	915.4 (13.6)	3.211(8)	23.401 (9.5)	26.612 (9.3)
Emr	580.2 (8.6)	4.918 (12.5)	18.581 (7.6)	23.499 (8.2)
Eur	889.2 (13.2)	2.713 (7)	25.502 (10.4)	28.215 (9.9)
Sear (India excluded)	579.1 (8.6)	3.974 (10.1)	23.938 (9.7)	27.913 (9.8)
Wpr (China excluded)	442.3 (6.6)	2.338 (6)	12.386 (5)	14.724 (5.2)
India	1181.4 (17.5)	8.075 (20.5)	54.544 (22.2)	62.619 (21.9)
China	1344.9 (20)	8.248 (20.9)	67.264 (27.3)	75.512 (26.5)
World	6737.5 (100)	39.365 (100)	246.024 (100)	285.389 (100)

Figure 1.1: Tabular representation of visually impaired population in the world

Chapter-2

Literature Review

2.1 Global Positioning System (GPS)

GPS refers to Global positioning system used to provide navigation, position, time under the condition of unobstructed line of sight to four or more GPS satellite from a total of 2.14 satellites [1].

2.1.1 History of GPS

Initially GPS was created by the Department of Defence which used 2.14 satellites to overcome the limitations of previous navigation systems but later it was made accessible to anyone with a GPS receiver [2]. Similar to this different Navigation or positioning system have been developed by different countries e.g.; The Russian Global Navigation Satellite System (GLONASS) was developed contemporaneously with GPS, Indian Regional Navigation Satellite System etc.

2.1.2 Fundamentals of GPS

The satellites and receiver both consists of clocks but satellites consists of Stable atomic clock which are synchronized with true time on the other hand receivers consists of clocks which are not that much stable and are not synchronized with the true time [3].

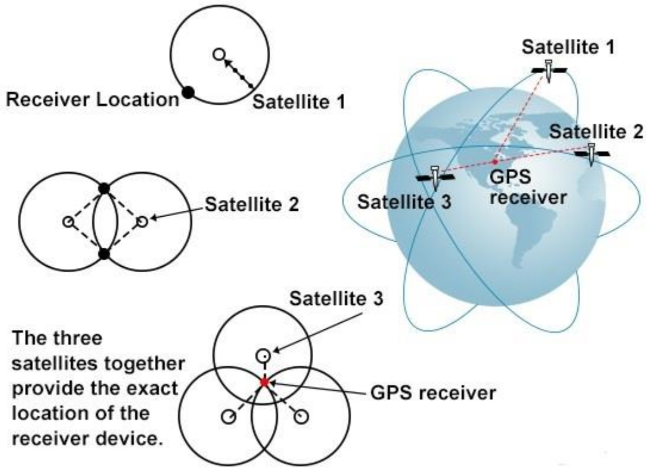


Figure 2.1.1: Schematic of concept for Detection of position by GPS

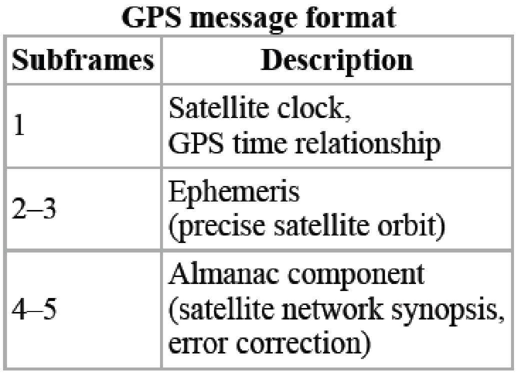
A GPS receiver continuously looks for data transmitted by the satellites (position and time) and solves the equation for determining the exact position and its deviation from true time . For the calculation of position and time a minimum of four satellites must be in communication with the receiver (three position coordinates and clock deviation from satellite time). The receiver measures the TOAs (time of arrival according to its own clock) of four satellite signals. From the TOAs and the TOTs(time of transmission), the receiver forms four time of flight (TOF) values, which are (given the speed of light) approximately equivalent to receiver-satellite range differences [4]. The receiver then computes its three-dimensional position and clock deviation from the four TOFs. The receiver's data is usually converted to latitude, longitude, and height relative to an ellipsoidal earth model.

2.1.3 Communications

The navigational signals transmitted by GPS satellites encode a variety of information including satellite positions, the state of the internal clocks, and the health of the network.

2.1.4 Message Format

Each GPS satellite continuously broadcasts a navigation message on L1 C/A and L2.1 P/Y frequencies at a rate of 50 bits per second. Each complete message takes 750 seconds (12.1 1/2.1 minutes) to complete [5]. The message structure has a basic format of a 1500-bit-long frame made up of five sub frames, each sub frame being 300 bits (6 seconds) long. Sub frames 4 and 5 are sub commutated 2.15 times each, so that a complete data message requires the transmission of 2.15 full frames. Each sub frame consists of ten words, each 30 bits long [5]. Thus, with 300 bits in a sub frame times 5 sub frames in a frame times 2.15 frames in a message, each message is 37,500 bits long.



GPS message format

Subframes	Description
1	Satellite clock, GPS time relationship
2–3	Ephemeris (precise satellite orbit)
4–5	Almanac component (satellite network synopsis, error correction)

Figure 2.1.2: Subparts of Standard GPS message Format

2.1.5 Standard Interfacing Sentences

2.1.5.1 Introduction

The National Marine Electronics Association (NMEA) has developed a specification or rather a standard that defines the interface between various electronic equipment. GPS receiver communication is defined within this specification. Mostly all the receivers and transmitters receive and transmit the data (position, velocity, and time) in the NEMA standard.

The basic concept of NEMA standard is to send a sentence which contains all the information and doesn't depend on other sentences. Each company can have their own proprietary sentences for use by the company.

2.1.5.2 Technical Specifications Of Sentences

All the standard sentences consists of two letter prefix that defines the types of device which wants to use it (For GPS receivers the prefix is GP.) which is followed by a three letter sequence that defines the sentence contents. All proprietary sentences begin with the letter P and are followed with 3 letters that identifies the manufacturer controlling that sentence.

Each sentence begins with a '$' and ends with a carriage return/line feed sequence and can be no longer than 80 characters of visible text (plus the line terminators). The data is contained within this single line with data items separated by commas. Programs that read the data should only use the commas to determine the field boundaries and not depend on column positions [6].

There is a provision for a checksum at the end of each sentence which may or may not be checked by the unit that reads the data. The checksum field consists of a '*' and two hex digits representing an 8 bit exclusive OR of all characters between, but not including, the '$' and '*'. A checksum is required on some sentences [7].

Most GPS receivers generally work with Serial to USB adapters and serial ports attached via the pcmcia (pc card) adapter. For general NMEA use with a GPS receiver only two wires in the cable will be sufficient, data out from the GPS and ground.

A third wire, Data in, will be needed if the receiver is expected to accept data on this cable such as to upload waypoints or send DGPS data to the receiver. The hardware interface for GPS units with most computer serial ports using RS2.132.1 protocols is compatible. The interface speed can be adjusted on some models but the NMEA standard is 4800 b/s (bit per second rate) with 8 bits of data, no parity, and one stop bit.

2.1.5.3 NMEA Sentences

NMEA consists of sentences, the first word of which, called a data type, defines the interpretation of the rest of the sentence. Each Data type would have its own unique interpretation and is defined in the NMEA standard. Whatever device or program that reads the data can watch for the data sentence that it is interested in and simply ignore other sentences that is doesn't care about [5]. In the NMEA standard there are no commands to indicate that the GPS should do something different.

Instead each receiver just sends all of the data and expects much of it to be ignored [6]. Some receivers have commands inside the unit that can select a subset of all the sentences or, in some cases, even the individual sentences to send.

There is no way to indicate anything back to the unit as to whether the sentence is being read correctly or to request a re-send of some data you didn't get. Instead the receiving unit just checks the checksum and ignores the data if the checksum is bad figuring the data will be sent again sometime later.

There are many sentences in the NMEA standard for all kinds of devices that may be used and have applicability to GPS receivers are listed below :

AAM - Waypoint Arrival Alarm

ALM - Almanac data

APA - Auto Pilot A sentence

APB - Auto Pilot B sentence

BOD - Bearing Origin to Destination

BWC - Bearing using Great Circle route

DTM - Datum being used.

GGA - Fix information

GLL - Lat/Lon data

GRS - GPS Range Residuals

GSA - Overall Satellite data

GPS- Pseudorange Noise Statistics

GSV - Detailed Satellite data

MSK - send control for a beacon receiver

MSS - Beacon receiver status information.

RMA - recommended Loran data

RMB - recommended navigation data for GPS

RMC - recommended minimum data for GPS

RTE - route message

TRF - Transit Fix Data

STN - Multiple Data ID

VBW - dual Ground / Water Speed

VTG - Vector track an Speed over the Ground

WCV - Waypoint closure velocity (Velocity Made Good)

WPL - Waypoint Location information

XTC - cross track error

XTE - measured cross track error

ZTG - Zulu (UTC) time and time to go (to destination)

ZDA - Date and Time

In interfacing a GPS unit to another device, including a computer program, ensurity should be there that the receiving unit is given all of the sentences that it needs. If it needs a sentence that GPS does not send then the interface to that unit is likely to malfunction. On NMEA input the receiver stores information based on interpreting the sentence itself.

While some receivers accept standard NMEA input this can only be used to update a waypoint or similar task and not to send a command to the unit.

2.1.6 Decoding Of Some Position Sentences

The most important NMEA sentences include the GGA which provides the current Fix data, the RMC which provides the minimum GPS sentences information, and the GSA which provides the Satellite status data.

2.1.6.1 GGA - essential fix data which provide 3D location and accuracy data [7]. $GPGGA,12.10145,4304.064,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

GGA	Global Positioning System Fix Data
12.10145	Fix taken at 12.1:35:19 UTC
4304.064,N	Latitude 43 deg 04.064' N
01131.000,E	Longitude 11 deg 31.000' E
Fix quality:	
0	Invalid
1	GPS fix (SPS)
2	DGPS fix
3	PPS fix
4	Real Time Kinematic
5	Float RTK
6	estimated (dead reckoning)
7	Manual input mode
8	Simulation mode
08	Number of satellites being tracked
0.9	Horizontal dilution of position
545.4,M	Altitude, Meters, above mean sea level
46.9,M	Height of geoid (mean sea level) above WGS84

Figure 2.1.3: Table representing definition of GGA format.

2.1.6.2 RMC - NMEA has its own version of essential GPS pvt (position, velocity, time) data. It is called RMC [6], The Recommended Minimum, which will look similar to:

$GPRMC, 12.13519, A, 4807.038, N, 01131.000, E, 02.12.1.4, 084.4, 2.130394, 003.1, W*6A

Where:

RMC	**Recommended Minimum sentence C**
12.13519	Fix taken at 12.1:35:19 UTC
A	Status A=active or V=Void.
4807.038,N	Latitude 48 deg 07.038' N
01131.000,E	Longitude 11 deg 31.000' E
02.12.1.4	Speed over the ground in knots
084.4	Track angle in degrees True
2.130394	Date - 2.13rd of March 1994
003.1,W	Magnetic Variation
***6A**	The checksum data, always begins with *

Figure 2.1.4: Table representing definition of RMC format.

2.1.7 Decode of some Navigation Sentences

2.1.7.1 WPL - Waypoint Location data provides essential waypoint data. It is output when navigating to indicate data about the destination and is sometimes supported on input to redefine a waypoint location [6]. Waypoint data does not mean or defines altitude, comments, or icon data. When a route is active, this sentence is sent once for each waypoint in the route, in sequence. When all waypoints have been reported, the RTE sentence is sent in the next data set. In any group of sentences, only one WPL sentence, or an RTE sentence, will be sent [7].

$GPWPL, 4807.038, N, 01131.000, E, WPTNME*5C

With an interpretation of:

WPL	Waypoint Location
4807.038,N	Latitude
01131.000,E	Longitude
WPTNME	Waypoint Name
***5C**	The checksum data, always begins with *

Figure 2.1.5: Table representing definition of WPL format.

2.1.7.2 AAM - Waypoint Arrival Alarm is generated by some units to indicate the Status of arrival (entering the arrival circle, or passing the perpendicular of the course line) at the destination waypoint [7].

$GPAAM, A, A, 0.10, N, WPTNME*32.1

AAM	Arrival Alarm
A	Arrival circle entered
A	Perpendicular passed
0.10	Circle radius
N	Nautical miles
WPTNME	Waypoint name
***32.1**	Checksum data

Figure 2.1.6: Table representing definition of AAM format.

2.1.7.3 BOD - Bearing - Origin to Destination shows the bearing angle of the line, calculated at the origin waypoint, extending to the destination waypoint from the origin waypoint for the active navigation leg of the journey [7].

$GPBOD, 045, T, 02.13., M, DEST, START*01 where:

BOD	Bearing - origin to destination waypoint
045., T	bearing 045 True from "START" to "DEST"
02.13., M	bearing 02.13 Magnetic from "START" to "DEST"
DEST	Destination waypoint ID
START	Origin waypoint ID
***01**	Checksum

Figure 2.1.7: Table representing definition of BOD format.

2.1.7.4 RMB - The recommended minimum navigation sentence is sent whenever a route or a goto is active. On some systems it is sent all of the time with null data. The Arrival alarm flag is similar to the arrival alarm inside the unit and can be decoded to drive an external alarm.

$GPRMB, A, 0.66, L, 003, 004, 4917.2.14, N, 12.1309.57, W, 001.3, 052.1.5, 000.5, V*2.10

Where :

RMB	Recommended minimum navigation information
A	Data status A = OK, V = Void (warning)
0.66L	Cross-track error (nautical miles, 9.99 max),

	Steer Left to correct (or R = right)
003	Origin waypoint ID
004	Destination waypoint ID
4917.2.14,N	Destination waypoint latitude 49 deg. 17.2.14 min. N
12.1309.57,W	Destination waypoint longitude 12.13 deg. 09.57 min. W
001.3	Range to destination, nautical miles (999.9 max)

Figure 2.1.8: Table representing definition of RMB format.

2.1.8 NEO-6 u-BLOX 6 GPS Module

It is the most cost effective, high-performance u-blox 6 based NEO-6 series GPS module that brings the high performance of the u-blox 6 positioning engine to the miniature NEO form factor. These receivers combine a high level of integration capability with flexible connectivity options in a small package [9]. This makes them perfectly suited for mass-market end products with strict size and cost requirements

2.1.8.1 Introduction

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature package.

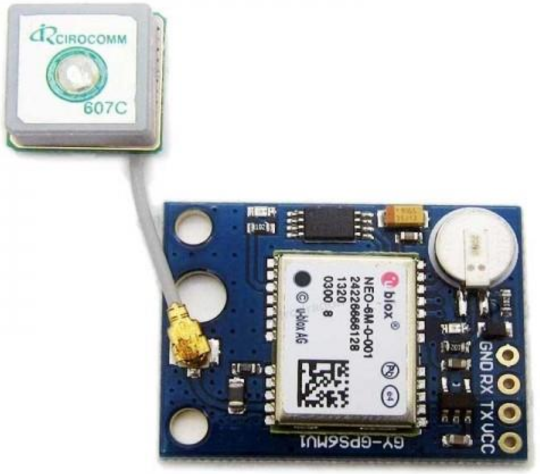


Figure 2.1.9: Actual photo of NEO 6m Ublox GPS

2.1.8.2 Communication

For communication between the GPS module and other devices some protocols or standard rules which have been defined are as follows

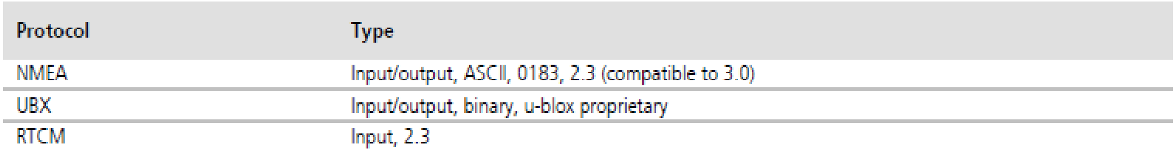
2.1.8.2.1 Precise Point Positioning

u-blox industry proven PPP algorithm provides extremely high levels of position accuracy in static and slow moving applications, and makes the NEO-6P an ideal solution for a variety of high precision applications such as surveying, mapping, marine.

2.1.8.2.2 Oscillators

NEO-6 GPS modules are available in Crystal and TCXO versions. The TCXO allows accelerated weak signal acquisition, enabling faster start and reacquisition times.

2.1.8.3 Protocols and interfaces



Protocol	Type
NMEA	Input/output, ASCII, 0183, 2.3 (compatible to 3.0)
UBX	Input/output, binary, u-blox proprietary
RTCM	Input, 2.3

Figure 2.1.10: Available Protocols in GPS for communication

2.1.8.3.1 UART

NEO-6 modules include one configurable UART interface for serial communication.

2.1.8.3.2 USB

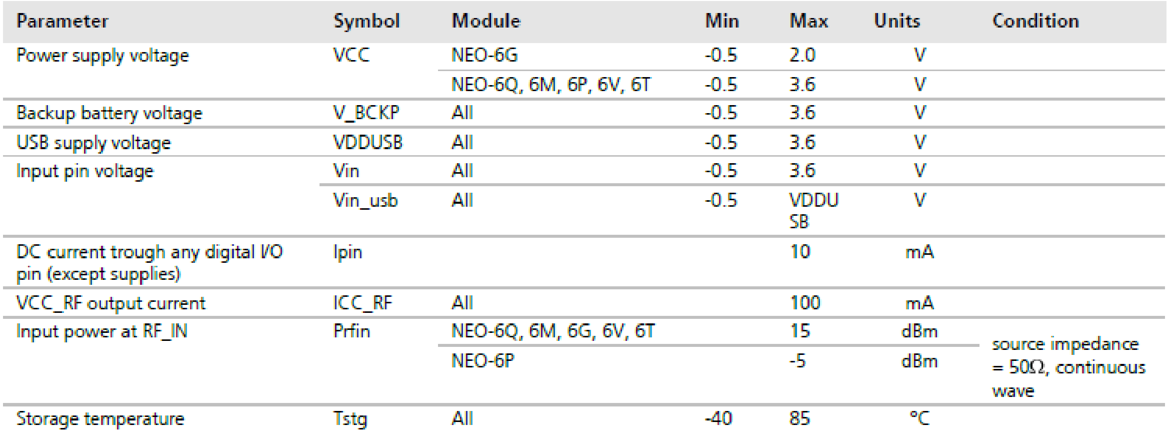
NEO-6 modules provide a USB version 2.1.0 FS (Full Speed, 12.1Mbit/s) interface as an alternative to the UART [6]. The pull-up resistor on USB_DP is integrated to signal a full-speed device to the host. The VDDUSB pin supplies the USB interface [8]. U-blox provides a Microsoft® certified USB driver for Windows XP, Windows Vista and Windows 7 operating systems.

2.1.8.3.3 Serial Peripheral Interface (SPI)

The SPI interface allows for the connection of external devices with a serial interface, e.g. serial flash to save configuration and Assist Now Offline A-GPS data or to interface to a host CPU. The interface can be operated in master or slave mode [9]. In master mode, one chip select signal is available to select external slaves. In slave mode a single chip select signal enables communication with the host.

2.1.8.4 Electrical specifications

2.1.8.4.1 Absolute maximum ratings



Parameter	Symbol	Module	Min	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	-0.5	2.0	V	
		NEO-6Q, 6M, 6P, 6V, 6T	-0.5	3.6	V	
Backup battery voltage	V_BCKP	All	-0.5	3.6	V	
USB supply voltage	VDDUSB	All	-0.5	3.6	V	
Input pin voltage	Vin	All	-0.5	3.6	V	
	Vin_usb	All	-0.5	VDDUSB	V	
DC current trough any digital I/O pin (except supplies)	Ipin			10	mA	
VCC_RF output current	ICC_RF	All		100	mA	
Input power at RF_IN	Prfin	NEO-6Q, 6M, 6G, 6V, 6T		15	dBm	source impedance = 50Ω, continuous wave
		NEO-6P		-5	dBm	
Storage temperature	Tstg	All	-40	85	°C	

Figure 2.1.11: Table representing various parameters of Neo 6m Ublox GPS

2.2 RADAR

2.2.1 Introduction

RADAR means **Radio Detection And Ranging.** Radar detects objects using radio waves to determine the range, altitude, direction, or speed of objects. The application of radar can be to detect aircraft, ships, spacecraft, guided missiles, motor vehicles to weather formations, terrain. The radar transmitter transmits radio waves through the radar transmitting antenna which are electromagnetic in nature. These waves travel through the medium incurring losses in the way to reach the object. The object generally metal interacts with the radio waves, produces electromagnetic field of its own so that no net electric field is formed inside the metal. This electromagnetic field produced to oppose the incident electromagnetic field travels to the receiver antenna. The losses are incurred in every portion of the journey from the transmitter to receiver.

Radars are used in diverse fields ranging from air traffic control, radar astronomy, air-defence systems, antimissile systems to marine radars to locate landmarks and other ships; aircraft anti-collision systems; ocean surveillance systems, outer space surveillance and rendezvous systems; meteorological precipitation monitoring; altimetry and flight control systems; guided missile target locating systems; and ground-penetrating radar for geological observations. High tech radar systems use digital signal processing which helps in extracting useful information from very high noise levels.

The information provided by radar includes the bearing and range (and therefore position) of the object from the radar scanner. It is thus used in many different fields where the need for such positioning is crucial. The first use of radar was for military purposes: to locate air, ground and sea targets

2.2.2 Principle of RADAR

A radar system has a transmitter that emits radio waves called *radar signals* in predetermined directions. When these come into contact with an object they are usually reflected or scattered in many directions. Radar signals are reflected especially well by materials of considerable electrical conductivity—especially by most metals, by seawater and by wet ground. The radar signals that are reflected back towards the transmitter are the desirable ones that make radar work. If the object is *moving* either toward or away from the transmitter, there is a slight equivalent change in the frequency of the radio waves, caused by the Doppler effect. Although the reflected radar signals captured by the receiving antenna are usually very weak, they can be strengthened by electronic amplifiers.

The weak absorption of radio waves by the medium through which it passes is what enables radar sets to detect objects at relatively long ranges—ranges at which other electromagnetic wavelengths, such as visible light, infrared light, and ultraviolet light, are too strongly attenuated.

Below is a figure showing the general principle of RADAR.

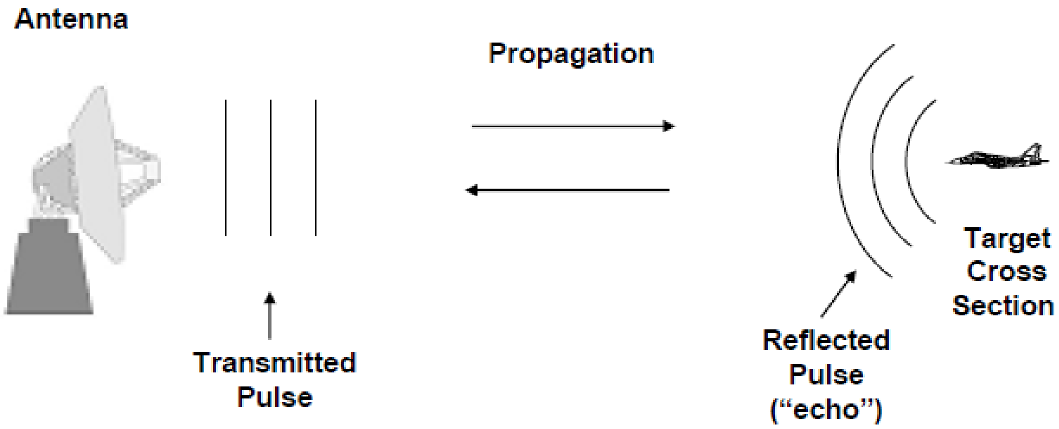


Figure 2.2.1 Principle of RADAR

2.2.3 Radar observables

- Target Range
- Target angles (Azimuth and Elevation)
- Target size (Radar cross section)
- Target Speed (Doppler)
- Target features (Imaging)

2.2.4 Radar Block Diagram

A typical Block Diagram of Radar and its execution is shown below:

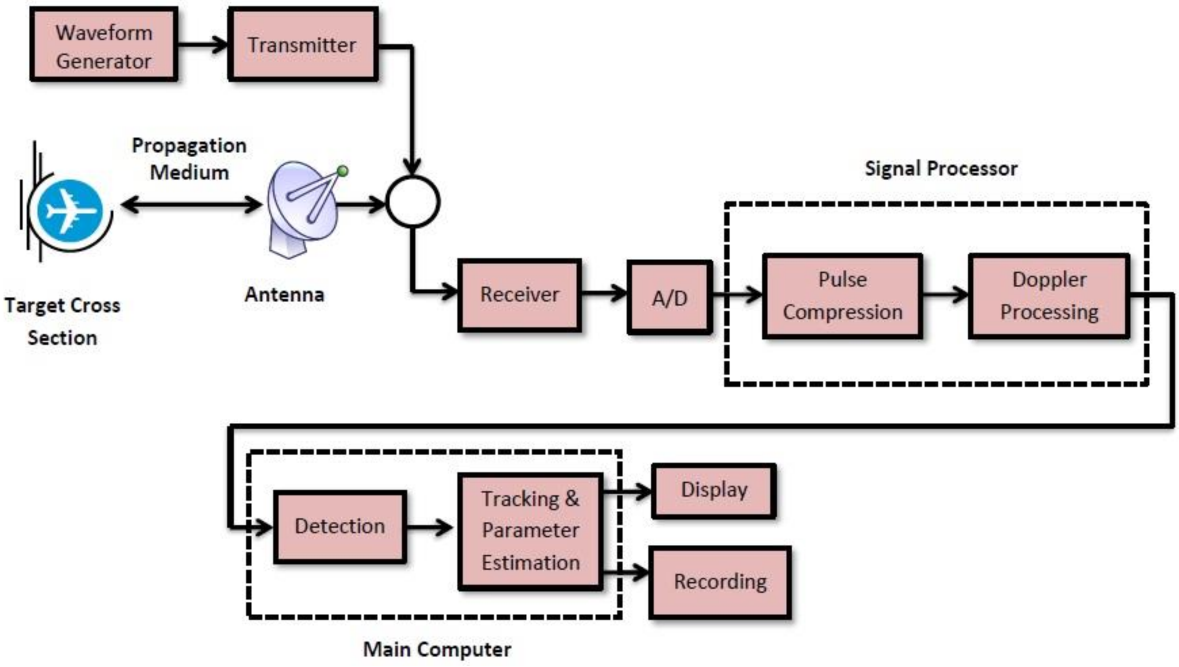


Figure 2.2.2 Block Diagram

2.2.5 Radar Frequency Bands

There are many different kinds of radars based on the criteria of classification:

Based on frequency of operation the radars can be classified into:

(i) HF (ii) VHF (iii) UHF (iv) L band (v) S band (vi) C band (vii) X band (viii) Ku band (ix) K band (x)Ka band (xi) V band (xii) W band Figu.

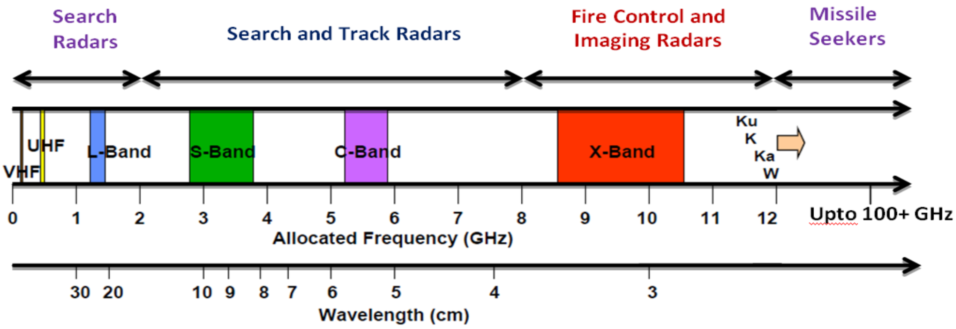


Figure 2.2.3 Frequency Band of Radar

2.2.6 Radar Range Equation

The received signal energy from the radar is given by the equation:

$$\text{Received Signal Energy} = \overset{\text{Transmit Power}}{\left[\,P_T\,\right]} \overset{\text{Transmit Gain}}{\left[\frac{4\pi A}{\lambda^2}\right]} \overset{\text{Spread Factor}}{\left[\frac{1}{4\pi R^2}\right]} \overset{\text{Losses}}{\left[\frac{1}{L}\right]} \overset{\text{Target RCS}}{\left[\,\sigma\,\right]} \overset{\text{Spread Factor}}{\left[\frac{1}{4\pi R^2}\right]} \overset{\text{Receive Aperture}}{\left[\,A\,\right]} \overset{\text{Dwell Time}}{\left[\,\tau\,\right]}$$

This received energy at the receiver is mixed with noise. Noise are of following types galactic, solar, manmade interference noise, atmospheric, ground, transmitter, receiver, waveguide and duplexer noise.

2.2.7 Detection of Radar pulses in noise

The detection of radar pulses involves two parameters namely probability of detection and probability of false alarm. On a fixed threshold, higher the Signal to Noise Ratio higher would be the probability of detection. The Signal to Noise Ratio of Radar is given by following equation.

SNR = 10 log10 [signal power/noise power]

Signal Power reflected from target and received by radar $\quad P_r = \frac{P_t G_t}{4\pi R^2} \frac{\sigma Ae}{4\pi R^2}$

Average Noise Power $\quad N = k\, T_s\, B_n$

Signal to Noise Ratio $\quad \frac{S}{N} = \frac{P_r}{N}$

$$\boxed{S / N = \frac{P_t\, G^2 \lambda^2 \sigma}{(4\,\pi)^3 R^4 k\, T_s\, B_n L}}$$

The System Noise Temperature Ts, is divided into 3 components:

Ts = Ta + Tr + LrTe

- Ta is the contribution from the antenna

- Apparent temperature of sky (from graph)

- Loss within antenna

- Tr is the contribution from the RF components between the antenna and the receiver

- Temperature of RF components

- Lr is the loss of input RF components

- Te is the temperature of the receiver

- Noise factor of receiver

The detection of target echoes in noise involves integration of pulses, fluctuating target issues and adaptive thresholding techniques. The integration of pulses can be done using coherent and non-coherent techniques. Coherent integration involves addition of in phase and quadrature quantities of complex radar return signal. These voltages are then computed, averaged and matched with a threshold. In general non-coherent integration, the pulse magnitude is calculated which is then averaged so that it can be compared with a threshold. In coherent techniques no information is lost while in non-coherent techniques phase information is lost. Coherent techniques are more efficient than non-coherent integration techniques.

2.2.8 Propagation effects on Radar Performance

- Atmospheric Attenuation
- Reflection off of earth's surface
- Over the Horizon Diffraction
- Atmospheric Refraction

Radar beams can be attenuated, reflected and bent bye the environment.

2.2.9 Doppler Shift Concept

To find speed from the output signal of the module the equation is used, where c is the s peed of light, fo is the signal frequency, and v is the speed of value to the application.

$$\Delta f = \frac{2 * v}{\lambda}$$

$$\lambda = \frac{c}{f_o}$$

The value obtained from this can then be manipulated using Doppler equations to find the

Speed of the target object. Using speed and time we can also find the distance from the obstacle.

Doppler frequency is related to velocity of motion through:

$$F_d = 2V\left(\frac{F_t}{c}\right)\text{Cos}\theta$$

F_d= Doppler Frequency

V = Velocity of the target

F_t = Transmit frequency

C = Speed of light ($3x10^8$ m/sec)

Θ = The angle between the target moving direction and the axis of the module.

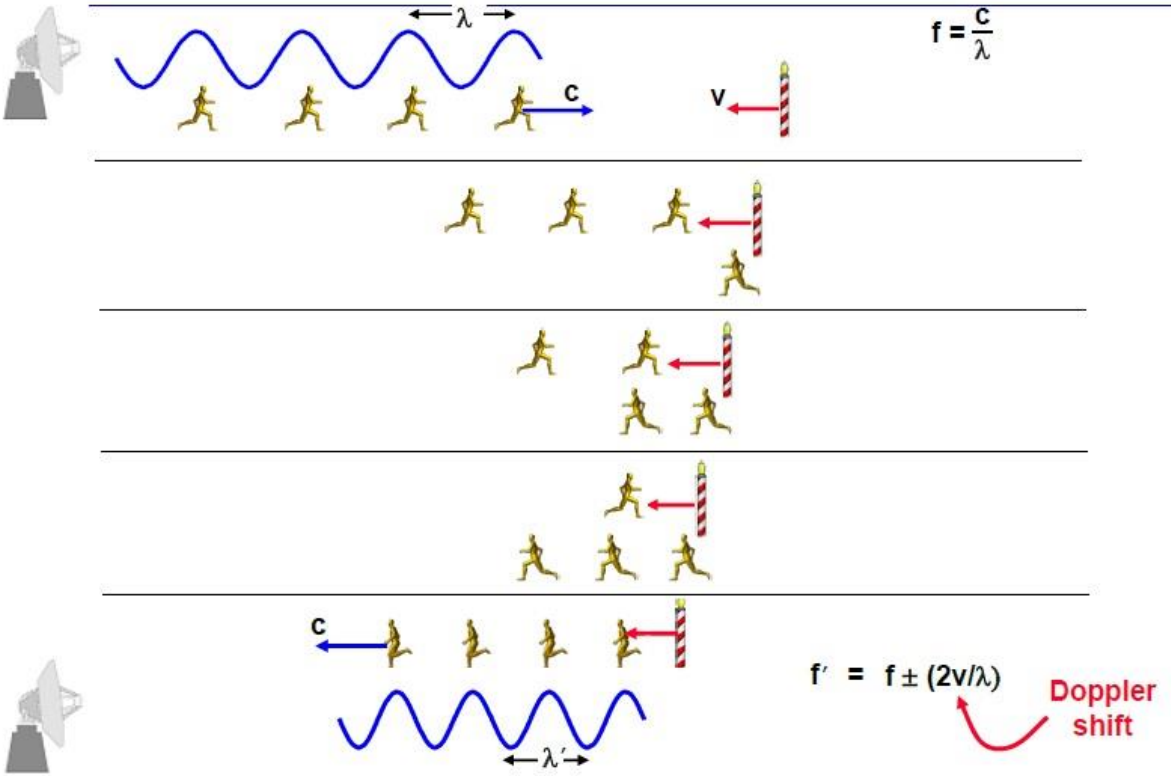


Figure 2.2.4 Concept of Doppler Shift

Doppler lets you separate things that are moving from things that aren't.

2.3. Speech User Interface

2.3.1. Speech recognition

Speech recognition is the translation of vocal words into text for recognition of the word by the computer. Speech recognition may be speaker independent which doesn't need training to adapt to the user's voice and speaker dependent systems requiring training.

Speech being a complex phenomenon is hard to understand how is it produced and perceived. One perception is that speech is built with words, and each word consists of phones. But it’s not so. Speech is a dynamic process without distinguishable parts.

All modern descriptions of speech are to some degree probabilistic which are why there are no certain boundaries between units, or between words. This is the major reason why Speech to text translation and other applications of speech are never 100% accurate. This idea is probably quite different for software developers, who usually work with deterministic systems which is why it creates a lot of issues specific only to speech technology.

2.3.1.1. Structure of speech

In current practice, speech structure is understood as follows:

Speech is a continuous sequence of states where rather stable states mix with dynamically changed states. In this, one can define more or less similar classes of sounds, or phones. The acoustic properties of a phonetic waveform varies according to - phone context, speaker, style of speech and so on. Coarticulation makes phones sound very different from their “canonical” representation. Since transitions between words are more informative than stable regions, developers often talk about diphones - which can be referred to as parts of phones between two consecutive phones. Sometimes its better to deal with subphonetic units which are different substates of a phone.

Three subphonetic units for a speech recognition engine are easily perceivable. First part of the phone depends on the preceding phone, the middle part is stable one, and the next part depends on the subsequent phone. That's why three states are mainly used in a phone, selected for speech recognition.

Sometimes phones are considered in context of the other parts. Such phones are called triphones or quinphones. For example “u with left phone b and right phone d” in the word “bad”. And it sounds a bit different from the same phone “u” with left phone b and right phone n“in word “ban”. Unlike diphones, they are matched with the same range in waveform as phones. Diphones and triphones just differ by name because they describe slightly different sounds.

For computational purpose detecting parts of triphones instead of triphones as a whole is much more helpful, for example, to create a detector for a beginning of triphone and share it across many triphones. The whole variety of sound detectors can be recognised by a small amount of distinct short sound detectors. These detectors are called senones. A senone's dependence on context is much more complex than just left and right context. It can be a rather complex function for example be defined by a decision tree.

Next, phones build subword entities, like syllables which are also defined as “reduction-stable entities”. For example, when speech becomes fast, phones change, but syllables do not. Syllables are related to intonational contour. There are other ways to build subwords - morphologically-based in morphology-rich languages or phonetically-based. Subwords are used in open vocabulary speech recognition techniques.

Collection of subwords forms words. Words are important as they restrict combinations of phones significantly. Words and other non-linguistic sounds, which we call as fillers (breath, um, uh, cough), form utterances. They are separate chunks of audio stream between pauses.

2.3.1.2. Recognition process

To recognize speech we do the following: we take the audio waveform, split it on utterances by silences and then try to recognize what's being said in each utterance. To accomplish it, we take all possible combinations of words and try to match them with the audio waveform we are processing. The best matching combination is chosen for further processing and implementation. There are few important things in this match.

First is the concept of features. Owing to the huge number of parameters dealt with, we are trying to optimize it. Numbers are calculated from speech usually by dividing speech on frames. Then for every frame of length typically 10 milliseconds we extract 39 numbers that represent the speech numerically. This is called feature vector. There are many ways to generate and code the signal into numbers but the one used more frequently is to code the numbers by their derivatives.

Second is the concept of model. Model describes a mathematical object that gathers common attributes of the spoken word. In practice, audio model of senone is a gaussian mixture of its three states or it's a most probable feature vector. From concept of the model many problems are raised - how good does model fits practice, can model be made better of its internal model problems and how adaptive model is to the changed conditions.

The model of speech is called Hidden Markov Model or HMM which is a generic model that describes sequential process like speech. In this model the process is described as a sequence of states which change each other with certain probability. It has been proven to be really useful for speech decoding.

Third, is the matching process itself. Since it would take a long time to compare all feature vectors with all models, the search has to be optimized by using tricks. At any point, we maintain the best matching variants and extend them as time goes producing best matching variants for the subsequent frame.

2.3.1.3. Models

According to the speech structure, three models are used in speech recognition to do the match:

An acoustic model contains acoustic properties for each senone. There are context-independent models that contain properties (most probable feature vectors for each phone) and context-dependent ones (built from senones with context).

A phonetic dictionary contains a mapping from words to phones. The dictionary is not the only variant of mapper from words to phones. It could be done with function learned with a machine learning algorithm.

A language model is used to restrict word search to optimize the process. It defines which word could follow previously recognized words (remember that matching is a sequential process) and helps to significantly restrict the matching process by stripping words that are not probable. Most common language models used are n-gram language models-these contain statistics of word sequences-and finite state language models-these define speech sequences by finite state automation, sometimes with weights. To reach a good accuracy rate, your language model must be very successful in search space restriction. This means it should be very good at predicting the next word. A language model usually restricts the vocabulary considered to the words it contains. That's an issue for name recognition. To deal with this, a language model can contain smaller chunks like subwords or even phones.

Those three entities are combined together in an engine to recognize speech.

A Lattice is a directed graph that represents variants of the recognition. Often, getting the best match is not practical; in that case, lattices are good intermediate formats to represent the recognition result. N-best lists of variants are like lattices, though their representations are not as dense as the lattice ones. Word confusion networks (sausages) are lattices where the strict order of nodes is taken from lattice edges.

Speech database: a set of typical recordings from the task database. If we develop dialog system it might be dialogs recorded from users. For dictation system it might be reading recordings. Speech databases are used to train, tune and test the decoding systems.

Text databases - sample texts collected for language model training and so on. Usually, databases of texts are collected in sample text form. The issue with collection is to put present documents (PDFs, web pages, scans) into spoken text form. That is, you need to remove tags and headings, to expand numbers to their spoken form, and to expand abbreviations.

2.3.1.4. Optimization

When speech recognition done, the most complex issue is to make search precise (consider as many variants to match as possible) and to make it fast enough. There are issues with making the model match the speech since models aren't perfect.

Usually the system is tested on a test database that is meant to represent the target task correctly.

The following characteristics are used:

Word error rate: Let we have original text and recognition text of length of N words. From them the I words were inserted D words were deleted and S words were substituted Word error rate is

WER = (I + D + S) / N

WER is usually measured in percent.

Accuracy: It is almost the same thing as word error rate, but it doesn't count insertions.

Accuracy = (N - D - S) / N

Accuracy is actually a worse measure for most tasks, since insertions are also important in final results. But for some tasks, accuracy is a reasonable measure of the decoder performance.

Speed: Suppose the audio file was 2 hours and the decoding took 6 hours. Then speed is counted as 3xRT.

ROC curves. When we talk about detection tasks, there are false alarms and hits/misses; ROC curves are used. A curve is a graphic that describes the number of false alarms vs. number of hits, and tries to find optimal point where the number of false alarms is small and number of hits matches 100%.

2.3.2. Text to speech

The goal of speech synthesis or text-to-speech (TTS) is to automatically generate speech (acoustic waveforms) from text. In other words, a text-to-speech synthesizer is a computer-based system that used to read any text aloud. There is a fundamental difference between text-to-speech synthesizer and any other talking machine in the sense that we are interested in the automatic production of new sentences. Speech synthesis performs this mapping in two phases. The first one is text analysis, where the input text is transcribed into a phonetic representation, and the second one is the generation of speech waveforms, where the acoustic output is produced from this phonetic and prosodic information. These two phases are usually called as high- and low-level synthesis. There are three main approaches to speech synthesis: articulatory synthesis, formant synthesis, and concatenative synthesis. Articulatory synthesis generates speech by direct modelling of human articulator behaviour. Formant synthesis models the pole frequencies of speech signal. Formants are the resonance frequencies of the vocal tract. Since the formants constitute the main frequencies that make sounds distinct, speech is synthesized using these estimated frequencies. On the other hand, concatenative speech synthesis produces speech by concatenating small, pre-recorded units of speech, such as phonemes, diphones, and triphones to construct the utterance. The following figure gives a high-level block diagram of the concatenative TTS synthesis process.

2.3.2.1 Text normalization

The first task of all text-to-speech systems is to pre-process or normalize the input text in a variety of ways. We will need to break the input text into sentences. For each sentence, we divide it into a sequence of tokens (such as words, numbers, dates and other types). Non natural language tokens such as acronyms and abbreviations must be converted to natural language tokens. In the following subsections, the steps of text normalization are explained in more details. Sentence Tokenization The first task in text normalization is sentence tokenization. This step has some difficulties because sentence boundaries are not always indicated by periods and can sometimes be indicated by other punctuations like colons. To determine sentence boundaries, the input text is divided into tokens separated by whitespaces and then any token containing one of these characters ! , . , or ? is selected and a machine learning classifier can be used to determine whether each of these characters inside these tokens indicate an end-of-sentence or not.

2.3.2.2. Pronunciation

The next stage after normalizing the input text is to find a pronunciation for each word. The main component in this stage is a large pronunciation lexicon. The pronunciation lexicon alone is not enough, because the input text can contain words such as names that cannot be found in the lexicon. For this reason, many text-to-speech systems use a name pronunciation lexicon in addition to the principal pronunciation lexicon. The name pronunciation lexicon needn't be very large, since the pronunciation of many names can be produced by analogy. For example, if the name-pronunciation lexicon contains the pronunciation of the name Trotsky, but not the name Plotsky, the initial /tr/ from Trotsky can be replaced with the initial /pl/ to generate a pronunciation for Plotsky. The pronunciation of unknown words that are not found in the pronunciation lexicon can be produced via the grapheme-to-phoneme conversion methods.

2.3.2.3. Prosodic Analysis

The final stage of text analysis is prosodic analysis. Prosody refers to the features that make sentences flow naturally. Without these features, speech would sound like a reading of a list of words. The three main components of prosody are phrasing, prominence, and intonation. For unit selection synthesis, an abstract representation of these features is all what is needed. For diphone and Hidden Markov Model (HMM) synthesis, a further step is needed which is to predict the fundamental frequency (F0) and the duration values. Phrasing has many effects on speech synthesis; the final vowel of a phrase is longer than the previous vowels and there is often a drop in the fundamental frequency from the start of a phrase to its end. Phrasing prediction can be based on deterministic rules. Modern techniques for phrasing prediction are data driven techniques. Wang and Hirschberg introduced the use of decision trees for phrase break prediction. A wide variety of machine learning algorithms have been applied for phrasing prediction such as memory based learning and neural networks. Prominence is used to indicate the strength of a word, syllable or phrase when it is used in a sentence. A word is made more prominent by saying it louder, saying it slower, or by varying the fundamental frequency during the word. Prominent words are generally associated with pitch accent. A sentence can be said with a final rise in F0 to indicate a yes-no question. In the following sections, DSP component is explored. Two rule-based synthesis techniques (formant synthesis and articulatory synthesis) are explained, and then concatenative synthesis is introduced, after that unit selection synthesis is explored and finally, HMM synthesis is introduced.

2.4 Digital Image Processing

The field of digital image processing refers to processing digital images by means of a digital computer. A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, and pixels[].

In digital image processing various operations like enhancement, segmentation, filtering and restoration are applied on images to extract useful information from them that are relevant to our field of application. There are various types of application programs or more appropriately said “frameworks” that can be employed to implement these operations on images on a computer, like MATLAB and OpenCV based vis. Visual Studios, Aforge, Numpy being a few prominent ones. In this project we have employed OpenCV libraries for image processing. Pre-built OpenCV libraries in Visual Studios were used by us in visual studio framework for testing and the same libraries were installed in our Ubuntu based system for real time implementation.

2.4.1 OpenCV- An Introduction

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel Russia research centre in Nizhny Novgorod, and now supported by Willow Garage and Itseez.[1] It is free for use under the open source BSD license. The library is cross-platform. It focuses mainly on real-time image processing. If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are now full interfaces in Python, Java and MATLAB/OCTAVE (as of version 3.0).

One of OpenCV’s goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. The OpenCV library contains over

500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics. Because computer vision and machine learning often go hand-in hand, OpenCV also contains a full, general-purpose Machine Learning Library (MLL).[]

Various OpenCV functions, applications and concepts were deployed to achieve different objectives during the development of program. These concepts are explained in detail in the following articles.

2.4.2 Canny Edge Detection

The Canny Edge detector was developed by John F. Canny in 1986. Also known to many as the optimal detector, Canny algorithm aims to satisfy three main criteria:

- Low error rate: Meaning a good detection of only existent edges.
- Good localization: The distance between edge pixels detected and real edge pixels have to be minimized.
- Minimal response: Only one detector response per edge

2.4.2.1 Implementation:

1) Filter out any noise. The Gaussian filter is used for this purpose. An example of a Gaussian kernel of size = 5 that might be used is shown below:

$$K = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

2) Find the intensity gradient of the image: A) Apply a pair of convolution masks (in X and Y directions) and Find the gradient strength and direction with:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \begin{matrix} G = \sqrt{G_x^2 + G_y^2} \\ \theta = \arctan(\frac{G_y}{G_x}) \end{matrix}$$

The direction is rounded to one of four possible angles (namely 0, 45, 90 or 135)

3) Non-maximum suppression is applied. This removes pixels that are not considered to be part of an edge. Hence, only thin lines (candidate edges) will remain.

4) Hysteresis: The final step. Canny does use two thresholds (upper and lower):

If a pixel gradient is higher than the upper threshold, the pixel is accepted as an edge. If a pixel gradient value is below the lower threshold, then it is rejected. If the pixel gradient is between the two thresholds, then it will be accepted only if it is connected to a pixel that is above the upper threshold.

Canny recommended a upper: lower ratio between 2:1 and 3:1.

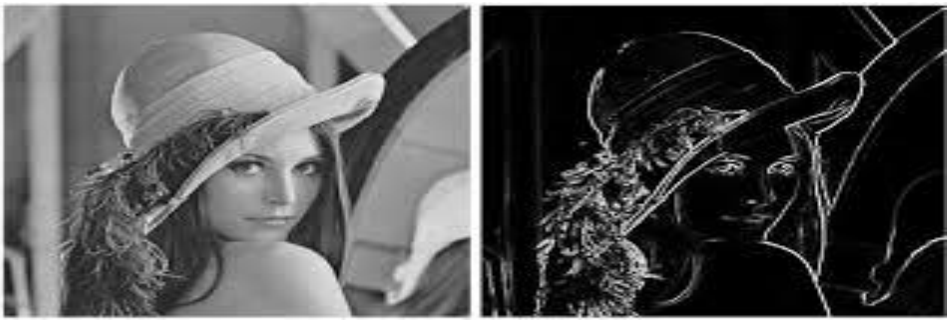


Figure2.4.1: Canny edge detection

2.4.3 Hough Transform

Hough transform is a function available in OpenCV which helps in shape detection of some standard geometrical shapes like circle, line, ellipse. The Hough line transform is used for lane detection. There are two types of Hough line transform explained in the subsequent articles

2.4.3.1 Hough Standard Transform

When we take image as matrix of pixel then a line in this image matrix can be represented in two basic forms:

1) Cartesian coordinate system: Parameters: (m,b).

2) Polar coordinate system: Parameters: (r,theta).

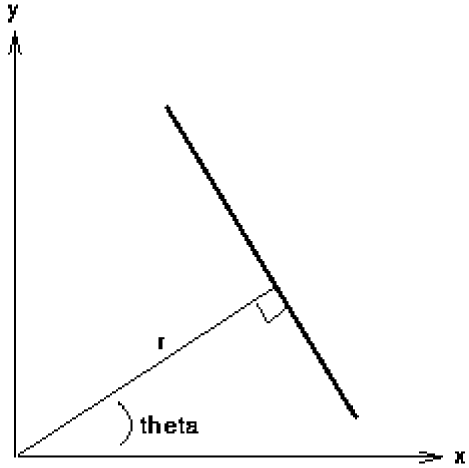


Figure 2.4.2: Hough Standard transform

In Hough standard Transform, we will express lines in the Polar system. Hence, a line equation can be written as:

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

OR

$$r = x\cos\theta + y\sin\theta$$

each pair (r) represents each line that passes by (x_0, y_0)

- In general for each point (r_θ, θ) we can define the family of lines that goes through that point as: $r_\theta = x_0 \cdot \cos\theta + y_0 \cdot \sin\theta$.

- If for a given (r_θ, θ) we plot the family of lines that goes through it, we get a sinusoid. For instance, for x=8, y=6, we get the following plot (in a plane θ - r):

  We consider only points such that $r > 0$ and. $0 < \theta < 2\pi$

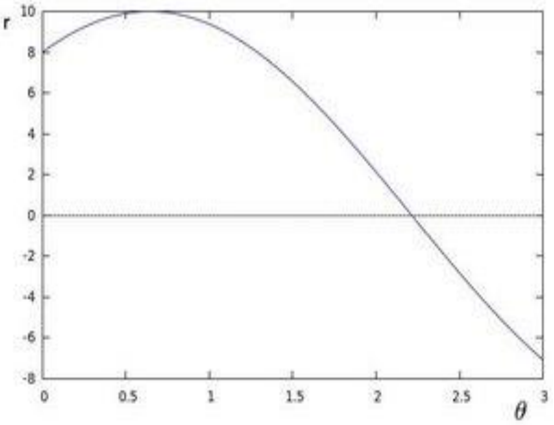


Figure 2.4.3: Plot for standard transform

3) We can do the same operation above for all the points in an image. If the curves of two different points intersect in the plane θ - r, that means that both points belong to a same line. For instance, following with the example above and drawing the plot for two more points: $x_1 = 9$, $y_1 = 4$ and $x_2 = 12$, $y_2 = 3$, we get:

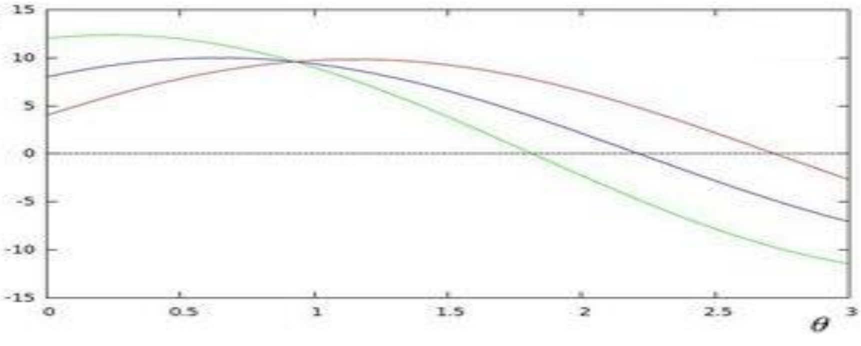


Figure 2.4.4: plot for hough tranform

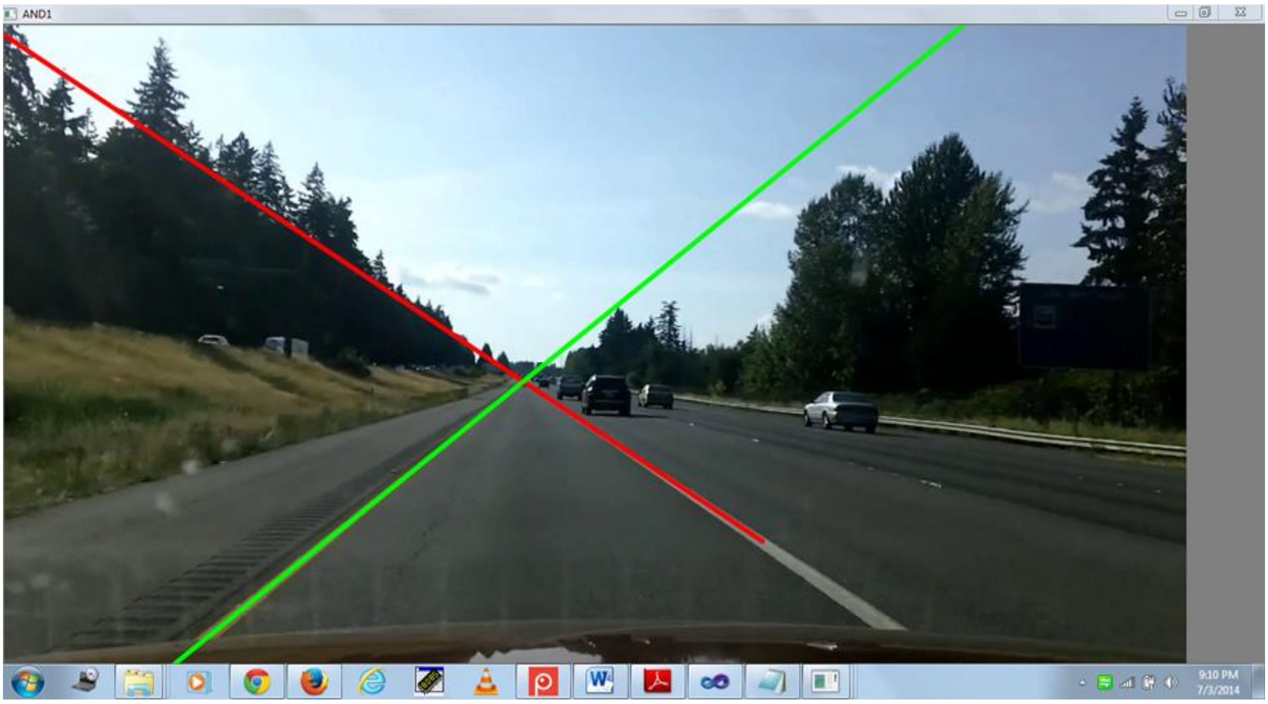


Figure 2.4.5: Detection of Hough lines in an image

The three plots intersect in one single point (0.925,0.96), these coordinates are the parameters (θ, r) or the line in which (x_0, y_0), (x_1, y_1) and (x_2, y_2) lay. It means that in general, a line can be *detected* by finding the number of intersections between curves. The more the curves are intersecting means that the line represented by that intersection has more points. In general, we can define a *threshold* of the minimum number of intersections needed to *detect* a line.

2.4.3.1Hough Probabilistic Transform

It is much more efficient and accurate way to detect lines then Hough transform because instead of returning lines in polar coordinates it directly gives two Cartesian coordinates of the detected lines. So it is easy to interpret the data returned by the function

- In Hough probabilistic, there is a parameter: minLine Length, it is used to set the Minimum line length. Line segments shorter than that are rejected; this is not present in Hough standard transform.

- Another parameter present in Hough probabilistic which is not in standard transform: maxLine Gap – Maximum allowed gap between points on the same line to link them.
- Third and most important advantage is that Hough probabilistic directly returns Cartesian coordinates not polar.

2.4.4 Bird Eye View / Perspective View:

When camera takes image then it covers area in shape of trapezium, ie as vertical distance increases, camera covers more area, and near camera it covers less area and hence covers the area in shape of trapezium, it is quite obvious. As a result it can be seen in above image that lanes which are parallel seem to be converging and intersecting at some point. So camera sees lanes as non-parallel lines but we want it to be real (as if bird is viewing the road from top).So for getting lanes parallel I remapped the pixels of trapezium into rectangle by calculating the relation between real world distance and pixels (example 1cm =10 pixel) and forming two matrix to adjust point ‘2’ at point ‘C’ and similarly point3 at point D and then use remapping function available in OpenCV. This will give me parallel lanes as output.

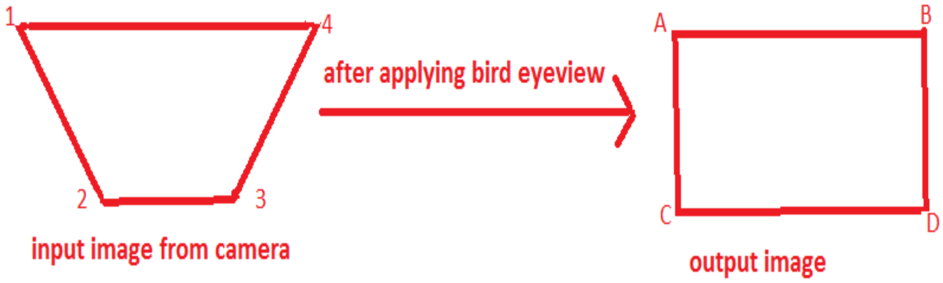


Figure 2.4.6: Bird eye prospective

2.4.5 Object Detection Using Classifier

The classifier is a set of APIs that allow you to define classes, or categories of nodes. By running samples of classes through the classifier to train it on what constitutes a given class, you can then run that trained classifier on unknown documents or nodes to determine to which classes each belongs. There are many classifiers available on internet like HAAR, LBP, etc. With these classifiers one can not only detect colour but also it gives good results for some complex tasks which are not possible with contour detection. Classifiers can be used for face detection, character and text recognition and many more. Classifier works on feature extraction. It involves following steps:

- Sampling: Sampling means to collect sample images of the object which is to be detected. This is very important step, and for good results sampling should be done accurately. Generally for good face detection program more than 1000 samples are to be taken. Suppose I want detect a traffic sign, for that I have to gather sample images of the sign from all possible angles and brightness conditions. More are the samples gathered more is the accuracy. In order to train our own classifier we need samples, which means we need a lot of images that show the object we want to detect (positive sample) and even more images without the object (negative sample).
- POSITIVE IMAGES: It means images of object to be detected, take photos of the object you want to detect, look for them on the internet, extract them from a video or take some Polaroid pictures generate positive samples for OpenCV to work with. It's also important that they should differ in lighting and background.
- NEGATIVE IMAGES: Now negative images are needed, the ones that don't show a object to be detected. In the best case, if one wants to train a highly accurate classifier, he should have a lot of negative images that look exactly like the positive ones, except that they don't contain the object we want to recognize. To detect stop signs on walls,

the negative images would ideally be a lot of pictures of walls. Maybe even with other signs. Keep an eye on the ratios of the cropped images, they shouldn't differ that much. The best results come from positive images that look exactly like the ones in which the object to be detected is present , except that they are cropped so only the object is visible.

CHAPTER 3

GPS AND MAPPING SYSTEMS

3.1 Technology and Basic concept

The basic concept behind the working of GLOBAL POSITIONING SYSTEM is the interaction of GPS receiver with a minimum of 4 satellites. The GPS system currently has 31 active satellites in orbits inclined 55 degrees to the equator. The satellites orbit about 2.10,000 km from the earth's surface and make two orbits per day. The orbits are designed so that there are always 6 satellites in view, from most places on the earth.

The GPS receiver gets a signal from each GPS satellite. The satellites transmit the exact time the signals are sent. By subtracting the time the signal was transmitted from the time it was received, the GPS can tell how far it is from each satellite.

The GPS receiver also knows the exact position in the sky of the satellites, at the moment they sent their signals. So given the travel time of the GPS signals from three satellites and their exact Position in the sky, the GPS receiver can determine position in three dimensions - east, north and altitude. To calculate the time the GPS signals took to arrive, the GPS receiver needs to know the time very accurately. The GPS satellites have atomic clocks that keep very precise time, but it's not feasible to equip a GPS receiver with an atomic clock. However, if the GPS receiver uses the signal from a fourth satellite it can solve an equation that lets it determine the exact time, without needing an atomic clock.

If the GPS receiver is only able to get signals from 3 satellites, we can still get our position, but it will be less accurate. As we noted above, the GPS receiver needs 4 satellites to work out our position in 3-dimensions. If only 3 satellites are available, the GPS receiver can get an approximate position by making the assumption that we are at mean sea level.

3.2 Data Format For Exact Mapping

To determine the location of the GPS satellites two types of data are required by the GPS receiver: the almanac and the ephemeris. This data is continuously transmitted by the GPS satellites and your GPS receiver collects and stores this data.

3.2.1 Almanac

The almanac contains information about the status of the satellites and approximate orbital information. The GPS receiver uses the almanac to calculate which satellites are currently visible. The almanac is not accurate enough to let the GPS receiver gives the co-ordinates.

3.2.2 Ephemeris

To get co-ordinates, GPS receiver requires additional data for each satellite, called the ephemeris. This data gives very precise information about the orbit of each satellite. GPS receiver can use the ephemeris data to calculate the location of a satellite to with a metre or two. The ephemeris is updated every 2.1 hours and is usually valid for 4 hours.

3.3 Trilateration Method

When GPS receivers interacts with a minimum of 4 satellites it can provide us with data (position) by using the method of TRILATERATION

Explanation of TRILATERATION method by an example:

Imagine we are standing somewhere on Earth with three satellites in the sky above us. If we know how far away we are from satellite A, then we know that we must be located somewhere on the red circle. If you do the same for satellites B and C, we can work out our location by seeing where the three circles intersect. This is just what our GPS receiver does, although it uses overlapping spheres rather than circles.

The more satellites there are above the horizon the more accurately our GPS unit can determine where we are.

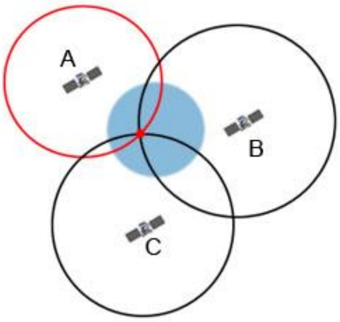


Figure 3.1: Schematic explaining the Triangulation method.

3.4 Description Of Navigation Code Or Program

For calculating the latitude and longitude of the GPS receiver using a software program, Beaglebone black is used for the execution of code .Firstly UART 4 and UART 2 (Communication protocol) are enabled, followed by opening the port and setting the baud rate of the port to 9600. After setting the baud rate beaglebone black is instructed to read the incoming data from GPS which is in the form of a string containing a huge amount of characters in the form of strings in different format. Out of all the data the first 500 bits of incoming data are received by the beaglebone black and the ports are closed. After that the required formats of data (GPvtg and GPrmc) are extracted from the string by using a compare command in the python language. Basically the word $GPrmc and $GPvtg are searched and the location of the data just after starting of GPrmc and before GPvtg are marked using Index register .The location of index register is used to extract the useful or required data from the received data. After extraction of data between GPrmc and Gpvtg the latitude data and longitude data is extracted by using the fact that the latitude and longitude are at a fix number of distance from the start of $GPrmc. The latitude and longitude after extraction are divided by 100 because the data that comes are already multiplied by 100. The hour unit of latitude is calculated by storing the data in other variable in the form of integer, Similarly the minute unit of latitude is calculated by subtracting the value of data in integer format from the value of data in float format In a similar way the hour, minute, second unit of longitude are extracted and stored in different variables.

The values of latitude and longitude so far obtained are in degree format so it is converted into radians by using suitable formula. By using a mathematical formula the distance between the GPS reading of receiver and the co-ordinates of destination to be defined by the user is calculated.

After some time the readings are taken and distance is again calculated and if it is found to be increased then the user will be notified that he is moving in the wrong direction. After getting the co-ordinated of destination, the co-ordinates are feed into the Google API server and the directions are provided by the Google maps and the user is navigated accordingly.

Chapter-4

RADAR (HB100 Microwave Motion Sensor Module)

4.1 Introduction

HB Series of microwave motion sensor module are X-Band Mono-static DRO Doppler transceiver front-end module. These modules are designed for movement detection, like intruder alarms, occupancy modules and other innovative ideas. The module consists of Dielectric Resonator Oscillator (DRO), microwave mixer and patch antenna (see Figure 4.1).

The radar system is designed using the HB-100 pulsed microwave Doppler sensor module. The range of the module is up to 20 meters which is enough for braking systems.

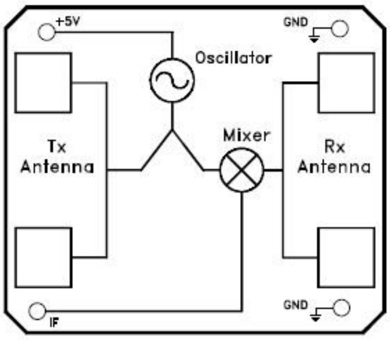


Figure 4.1: HB100 Microwave Motion Sensor Module

Doppler shift output is observed from IF terminal when movement is detected in the field of detection. The magnitude of the Doppler Shift is in proportions to reflection of transmitted energy and is in microvolts. A high gain low frequency bandwidth amplifier is connected to the IF pin to amplify the Doppler shift to a level that can be read by development platform like Arduino Uno. Frequency of Doppler shift is calculated using the algorithm to determine the velocity of targets.

4.2 Features of a HB100 Module

- Low current consumption
- CW or pulse operation
- Flat Profile
- Long Detection Range

4.3 Mounting of Radar and its components

Header pins can be used to connect the terminals (IF,+5V,Ground) to the amplifier circuit as well as mounting support.

The Module operates at +5V DC for continuous wave operation. It can also be powered by +5V low duty cycle pulsed trains to reduce its power consumption.

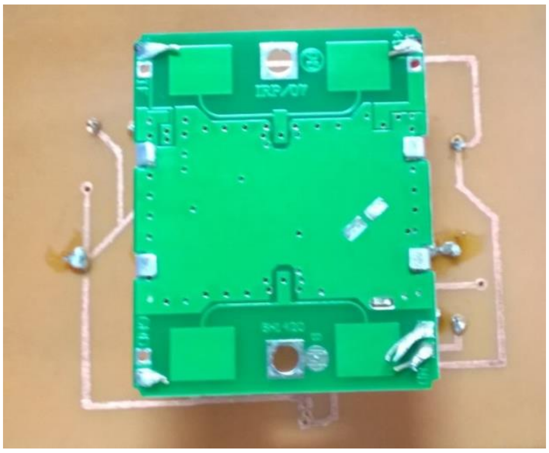


Figure 4.2 Radar mounting

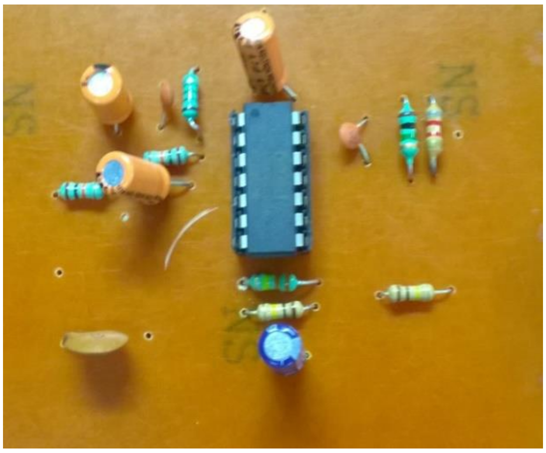


Figure 4.3 PCB of Radar

4.4 Radiation Pattern Observed

The module to be mounted with the antenna patches facing to the desired detection zones. The user may vary the orientation of module to get the best coverage. The radiation pattern are shown below.

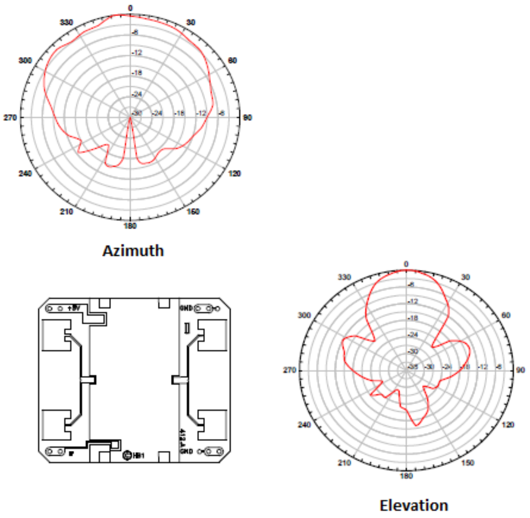


Figure 4.4: Radiation Pattern of Radar

4.5 Amplifier Circuit for Radar

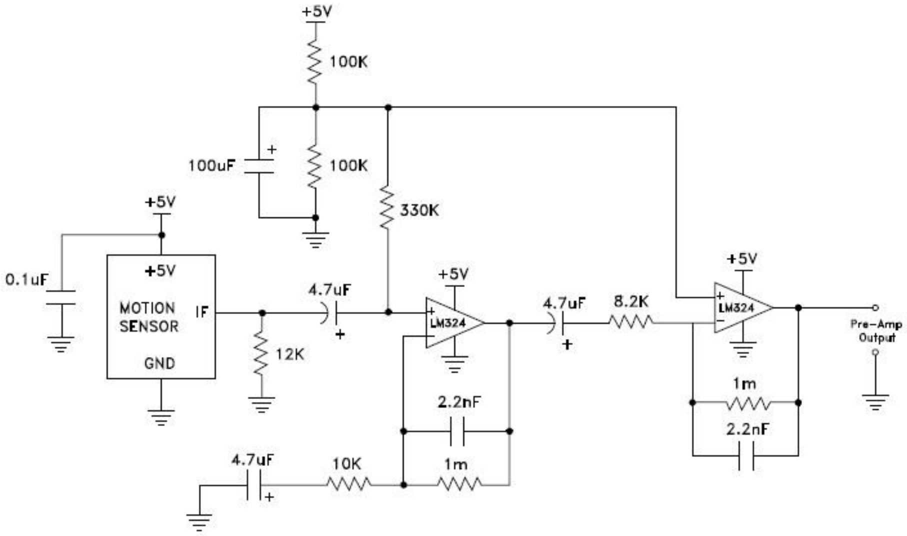


Figure 4.5: Amplifier Circuit

4.6 Calculation of frequency using Doppler equations

The output voltage from the intermediate frequency pin is roughly 20mVpp, so our first order of business was to amplify the signal to a level where a comparator could easily detect zero crossings of the sinusoidal signal.

A single sided op-amp with a gain of roughly 50 can modify the Doppler signal to a 1Vpp level. The frequency to voltage converter is fed with this signal and is designed to map 0 – 400 Hz to output voltages of 0 – 4 VDC. The ADC is then fed by our DC signal from the converter and sampled to reproduce an 8bit value, with a comparison point of 4V to increase accuracy range.

To find speed from the output signal of the module the equation is used, where c is the speed of light, fo is the signal frequency, and v is the speed of value to the application.

$$\Delta f = \frac{2 * v}{\lambda}$$

$$\lambda = \frac{c}{f_o}$$

Doppler frequency is related to velocity through the given equation:

$$F_d = 2V\left(\frac{F_t}{c}\right)\text{Cos}\theta$$

4.7 2 Pulse MTI canceller

Targets can be distinguished from the clutter using Moving Target Indicator technique and Pulse Doppler technique. Moving target Indicator techniques use low pulse repetition frequency and short waveforms to separate targets and clutter. Pulse Doppler technique classifies targets into different velocity regimes providing velocity data along with separation of targets from clutter. This technique uses long waveforms for their operation.

A two pulse MTI canceller can be used to illustrate the principal of multiple object detection. The amplitude of return signal from the moving objects changes in consecutive pulses owing to the Doppler shift while the fixed clutter returns fixed echoes. The subtraction of subsequent pulses eliminates the clutter from the return signal and gives the multiple moving objects at different range cells. The following block diagram depicts method deployed for multiple target detection.

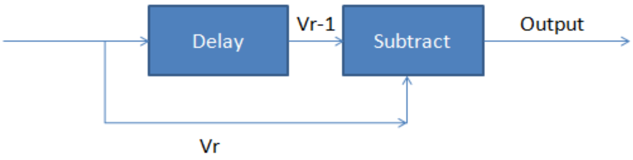


Figure 4.6: Block diagram for multiple target detection

For 2 pulse MTI canceller,

$$\mathbf{V_{output} = V_{i+1} - V_i}$$

For 3 pulse MTI canceller,

$$\mathbf{V_{output} = V_i - 2V_{i-1} + V_{i-2}}$$

4.8 Algorithm

The Arduino Uno is initialized with Pin 11 as Pulse generator. A PWM of 2 kHz pulse repetition frequency is generated with a pulse width of 10 microseconds high time using the delay Microseconds() function and giving 240 microseconds low time for the digital pin. The pulse received is converted from analog to digital and this is sent to non-coherent integrator wherein the consecutive pulses are simply added and averaged to increase SNR. These integrated pulses are then differenced from subsequent pulses to get the number of objects. The objects in the path are detected using Greatest of Mean level CFAR technique adaptive thresholding with the Arduino storing the range gates in an array and then thresholding each range point with the greater of the mean of the next and previous ten range points. The non-coherently integrated pulses are passed through a zero crossing detector [4] or comparator to get a square wave. The frequency counter library of Arduino is then used to compute the frequency and hence the speed.

Flowchart of whole algorithm is shown

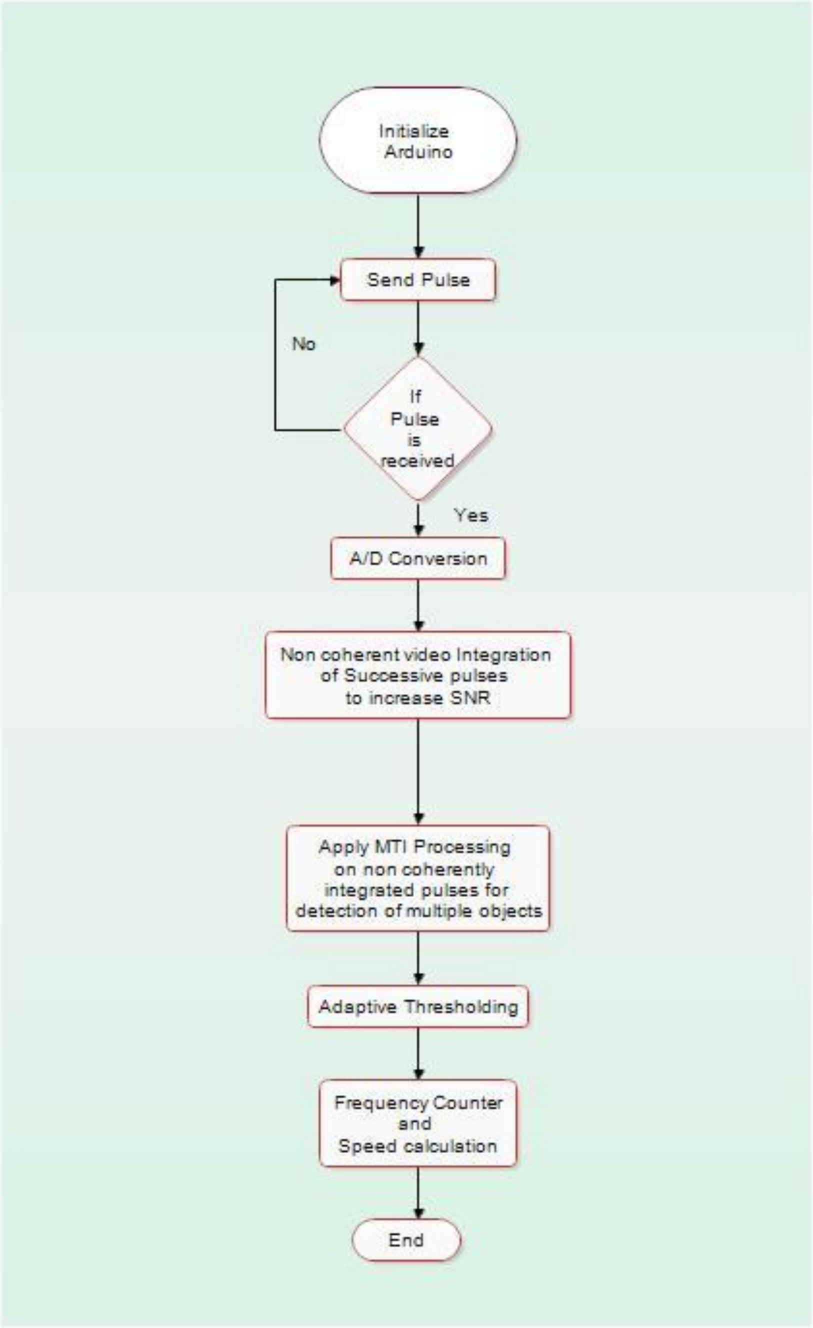

Initialize Arduino
Send Pulse
No
If Pulse is received
Yes
A/D Conversion
Non coherent video Integration of Successive pulses to increase SNR
Apply MTI Processing on non coherently integrated pulses for detection of multiple objects
Adaptive Thresholding
Frequency Counter and Speed calculation
End

Figure 4.7 Flowchart of whole Algorithm

Chapter 5

Speech User Interface Implementation

In this chapter we will be discussing about the user interface built to assist the blind in its navigation. The user interface relies on the speech and hearing capabilities to convey or acquire any message to or from the user. The implementation of user interface is described.

5.1. Voice Recognition

The project tries to implement the user interface for the blind in a speech format. The blind can give speech commands to the cane as to where he wants to visit. This gives the destination coordinates in terms of latitude and longitude for the user. We are giving the user a full variety of instructions as the whole system will be implemented on the Google speech engine which is highly efficient in its operation. Owing to the use of a speech recognition engine it gets highly convenient for the blind to move freely in its surroundings. The speech recognition is used to give the location of the blind for further processing by the Google map engine. There are major advantages for this system over the conventional mobile application. First it is integrated in the cane itself which the blind will take wherever he goes. Secondly, there are not many mobile applications that cater to this problem.

For implementing the algorithm for speech recognition we are using the python os libraries to get the mic configured. Firstly installing the drivers and configuring the beaglebone black for operation on an usb PnP sound card is done through bash terminal commands. The default sound card is changed from the normal HDMI sound card to usb sound card so that the mic can be interfaced with the system. The sound card being a cheap one that’s only available in the market has no internal audio amplification. A class D amplifier would have been considered as an alternative for amplification. To rectify this we considered building our own amplifier circuit but were not able to do so due to time and cost constraints. Owing to the low cost of the sound card the audio quality of the mic is quite noisy making it unfit for the product. So it was decided to replace the usb sound card in all future revisions of the product with a better quality one.

The mic is interfaced with the system to take voice commands and store the audio in wav format. The aforementioned task is trivial in case of a Linux machine using terminal commands. Using arecord and saving the file to the place you desire is all that you need. The file can be then further used for processing by other programs in the system. It can be used for a variety of purposes as in real time voice recognition to saving for future reference. The recorded file is stored in wav format. The wav format has to be converted into binary data so that it can be read and recognised by the recogniser. This is accomplished using the python speech recognition library for Google speech recognition engine. The wav file is sent as an input source to the recognizer to recognize the audio. The recognizer firstly uses the wav file extract audio data from the wav file. The audio data is stored in a separate file which is sent to the Google speech recognition engine for processing.

The Google speech API transforms the input data into valid text and gives back the transcription to the beaglebone black. If the voice volume level is low or if noise constitutes a large part of the input data, the transcription is not possible in which case the Google speech recognition engine replies with an answer that it could not understand the audio. There are several limitations to this system with the Google speech recognition engine. The engine only allows 50 calls per day which can be easily used up if you don’t give enough consideration in your code and learning process. While it’s highly accurate, these limitations are quite depressing in terms of making the product continue with the speech API in its future revisions. There are alternative speech recognition software that we are looking forward to using to make the product see the light of the day.

5.2. Text to Speech

The speech recognition engine is complemented with the text to speech engine that we are using so that the blind can hear the directions where he should be visiting along with hindrances and obstacles in the path. While this can be done using vibration motors also, but the main purpose of our project was to make the blind get as much information about its surroundings as possible. While it would have been highly convenient for us to use the former way, we chose the other one. The text to speech engine is provided offline using the Espeak libraries for python. There are various advantages of this in real time. Instead of just getting to know that there is an object in front using ultrasonic sensor with a vibration motor, we here can implement the text to speech engine to read out loud the distance from the various objects in the different directions using the ultrasonic sensors. This is very beneficial for the blind to choose his future course of action. Along with this usual distance from the obstacles we are using the text to speech engine to get the directions to the destination read out loud to the user. The user will be getting the directions from the Google map API and these will be converted to the format so as to further process the readings. The Google map API sends the data in a format in which it gives the distance to be travelled in a particular direction along with the duration it will take to reach the waypoint. The duration of the reading or more specifically the waypoint can be used to repeat the directions to the user. After the user gives it preferred destination through voice commands the system gives direction commands to the user in terms of speech. These speech commands are repeated at definite intervals according to the duration specified by the Google map engine. In this way as the person reaches the waypoint the Google map API gives the next waypoint direction at that point only. This leads to a feedback type control structure for the system.

The whole system is implemented on the Espeak text to speech engine. One of the main advantages of the Espeak library is its offline nature. Being very compact and low in memory, it can be deployed on any major single board computer for its effective implementation. Espeak has variety of commands for various kinds of voices and parameters. The volume level and speed level of the speech can be changed with simple commands from the command line. This can be useful to get the blind his desired speed level and volume level for a comfortable usage.

Espeak has provisions for different languages along with different versions of the voice one being male and other female. Although the Espeak library is still in development it can be used for basic text to speech conversion as is the case for our system. Espeak supports a variety of languages ranging from English, Spanish, and French to Tamil, Hindi and other Indian languages. Owing to its incomplete nature many of the languages are still available in either male or female versions of the language.

In our system we have first imported the Espeak libraries to the python programming language. The Espeak libraries are then used with to synthesize the full sentence from the text that we are providing it. The text for the text to speech engine comes from Google map API. After the concatenation of the different strings the text is converted into speech for the user. The various parameters like the speed, gender and language can be passed as a parameter to the command for changing the corresponding values.

5.3. Google Map API

The Google speech recognition engine uses the input wav file and sends it to the Google server for audio extraction. This audio is send to the map API as the destination of the user. The Google map API is used along with the direction and distance API. The GPS readings of the user are taken using the GPS installed in the system. This GPS reading is put into the map API as initial location of the user. The direction is then calculated using the python library “Googlemaps”. This API gives us many functions such as geo coding, reverse geo coding, directions and distance. The geo coding gives us the name of the place according to their respective latitude and longitude. The reverse geo coding gives us the latitude and longitude as output and takes the name of the place as the input. The direction of the place can be acquired using this function. The parameters going into this function are the source, destination and the mode of transport. In the case of blind, we are taking the mode as walking.

Firstly the text extracted from the voice is translated into text. The final destination is then reversing geo coded to get the latitude and longitude of the final destination. The text extracted then goes directly to the map API to get the directions. The latitude and longitude is sent to the distance function to get the distance between the initial GPS location (latitude, longitude) and the final destination location that is derived from the reverse geo coded function of the extracted audio.

The direction function is used to get the direction and this is acquired by the python program. The format of the return string is changed to get the required information from the return string. This is then sent out to the blind as speech instructions.

The format of the return output is an array with a lot of information. The output of the Google map API contains legs and steps. The output can be parsed for distance, duration and the directions of the different waypoints to reach the final destination.

The distance calculated is used to guide the blind as in a feedback type control system. The GPS location of the person is continuously monitored and compared with the destination coordinates. If the distance calculated using the aforementioned method gives an increase in distance the user is alerted that he is going in the wrong direction. This distance estimation requires the use of the haversine formula which gives the distance between two points. The haversine formula is an equation important in navigation, giving great-circle distances between two points on a sphere from their longitudes and latitudes.

Chapter 6

Image Processing Subsystem

6.1 General

The image processing subsystem is employed in our project so as to detect lane markings on the road and guide the blind person accordingly. This subsystem woks in tandem with the GPS subsystem in navigating turns and keeping the blind on a set path as found out by the Google Maps application. This subsystem also includes programs for vehicle detection so as to inform the blind person of any approaching fast vehicle that may not be detected well in time by the radar subsystem.

6.2 Description of Algorithm

The algorithm for image processing is designed in a way such that two objectives are served simultaneously namely detection of any oncoming vehicle and the other being guiding the blind person on a set path by detecting the lane markings on the adjacent road so that the user does not stray away on the road. In a situation where the user is required to cross the road this program will detect any oncoming vehicle and the program will branch out to the main program that will inform the user through audio commands that a car is approaching and he should stop. Once the car has passed then the user will be told to cross the road safely. While the user is moving alongside the road following the navigation path told to him by the navigation subsystem the camera will continuously monitor the distance from the lane markings thus informing him about the correct and doing course correction in case he stays away from the set path. This subsystem ensures that the blind person keeps walking only on the footpath thus safely guiding the user to his destination.

6.3 Software Implementation

The software implementation of the algorithm mentioned in the previous article is accomplished by installing OpenCV libraries on our control unit that is Beaglebone Black that runs on Ubuntu OS. Using these libraries we have developed the programs for vehicle detection and lane extraction and mapping. Vehicle detection and lane mapping programs are explained in the following subsections.

6.4 Vehicle Detection Program

The method that we applied for detecting approaching vehicles is vehicle detection using Cascades with Haar like features. It is one of the best methods available to detect objects. In this we can train the classifier according to our needs that is according to the object that we want to detect along with any particular surroundings that we want.

Training is the process of taking content that is known to belong to specified classes and creating a classifier on the basis of that known content. Classification is the process of taking a classifier built with such a training content set and running it on unknown content to determine class membership for the unknown content. Training is an iterative process whereby you build the best classifier possible, and classification is a one-time process designed to run on unknown content.

To train using Haar Cascades we have to make an ".xml" file with positive and negative image samples of objects to track and relevant surroundings. Making the .xml file is a long, time consuming process which involves cropping positive samples and negative samples from a very large sample of pictures. These pictures were extracted using ffmpeg from videos of our surroundings and objects. Finally after training, a 'xml' file is generated which is loaded in the main program to match the features and detect whether object is present or not.

This method basically uses Viola Jones Facial Detector. In this detector the haar-features are located in a particular frame by running small rectangular detector over the image. By comparing the relative gradient in pixels the detector is able to identify vehicles in the image. These vehicles are marked by blue circles in each frame.

6.5 Lane Detection and Mapping Program

In this lane detection program the camera is used to acquire frames continuously at thirty frames per second. Then Gaussian smoothing and filtering is applied on this frame to remove noise. The frame shows lanes in a form that lanes which are parallel seem to be converging and intersecting at some point. Extracted lanes from this frame will show incorrect data about the length of the lanes and the distance between them as they are parallel actually. In order to remove this erroneous data we will transform this trapezoidal frame into rectangular shape by applying bird's eye perspective. The bird's eye perspective is applied using Mean Value Theorem and the pixels are remapped into a rectangle using a mapping relation. The image is then converted into grayscale and canny edge detection is applied on it. Canny edge detection extracts edges from the from the grayscale image. As we know that a line can be formed between any two points on a plane, infinite number of lines can be formed between the edges drawn out from canny edge detection. Then on this image probabilistic Hough transform is applied to extract lines that resemble actual lanes on the road. After this process using the mapping parameters the distance between the lanes is calculated along with the distance of the user from the center of the road. This distance is then continuously monitored to guide the user along the road. The following flow chart explains the flow of control in the lane detection program.

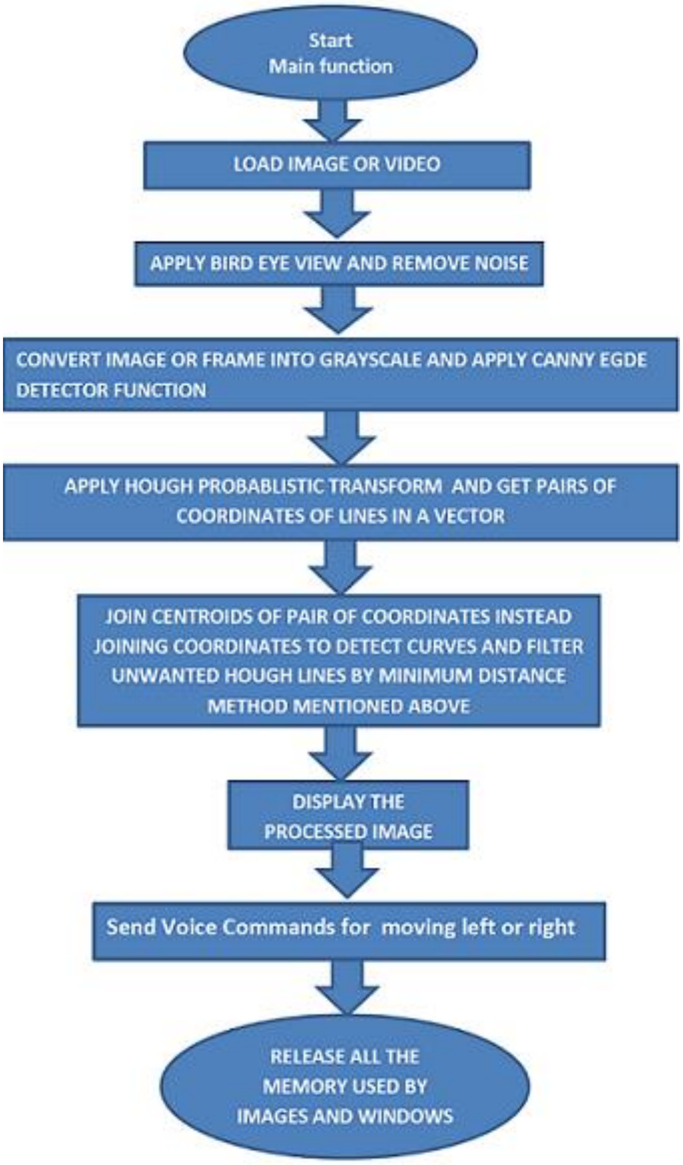
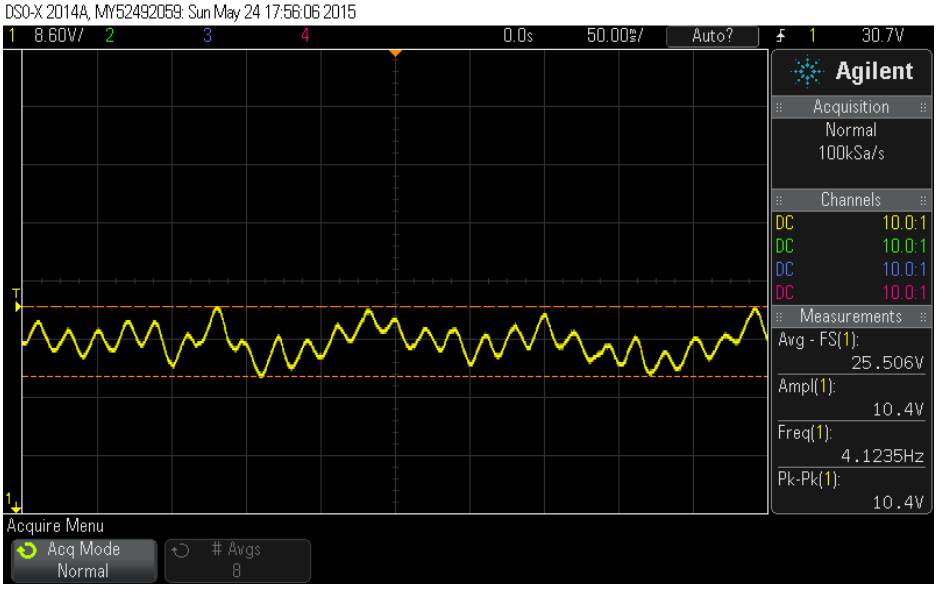

Start
Main function
LOAD IMAGE OR VIDEO
APPLY BIRD EYE VIEW AND REMOVE NOISE
CONVERT IMAGE OR FRAME INTO GRAYSCALE AND APPLY CANNY EGDE DETECTOR FUNCTION
APPLY HOUGH PROBABLISTIC TRANSFORM AND GET PAIRS OF COORDINATES OF LINES IN A VECTOR
JOIN CENTROIDS OF PAIR OF COORDINATES INSTEAD JOINING COORDINATES TO DETECT CURVES AND FILTER UNWANTED HOUGH LINES BY MINIMUM DISTANCE METHOD MENTIONED ABOVE
DISPLAY THE PROCESSED IMAGE
Send Voice Commands for moving left or right
RELEASE ALL THE MEMORY USED BY IMAGES AND WINDOWS

Figure 6.1: Flowchart of Lane Detection Program

Chapter 7

Results and Discussions

7.1 Simulation and Results of Radar



DSO-X 2014A, MY52492059: Sun May 24 17:56:06 2015

1 8.60V/ 2 3 4 0.0s 50.00s/ Auto? 1 30.7V

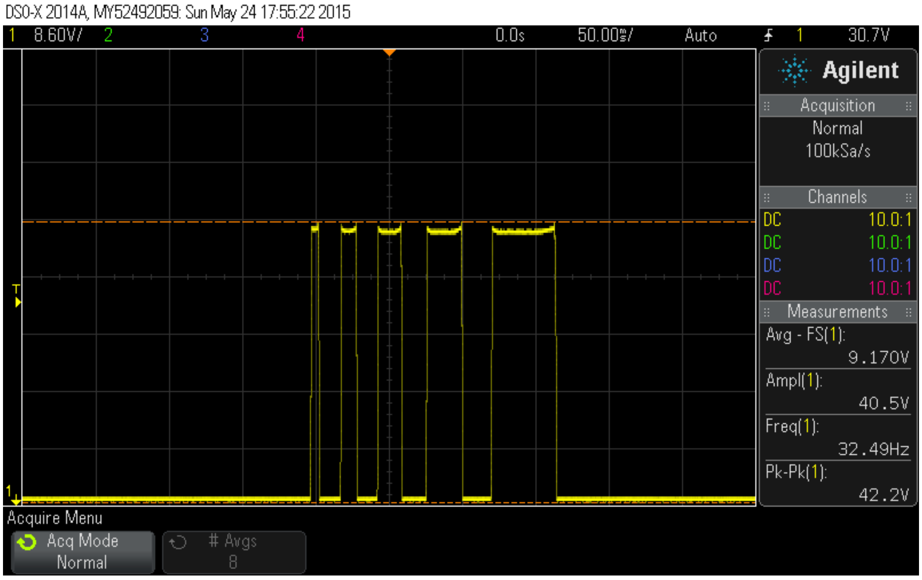
Acquisition: Normal 100kSa/s

Channels: DC 10.0:1, DC 10.0:1, DC 10.0:1, DC 10.0:1

Measurements:
Avg - FS(1): 25.506V
Ampl(1): 10.4V
Freq(1): 4.1235Hz
Pk-Pk(1): 10.4V

Acquire Menu: Acq Mode Normal, # Avgs 8

Figure 7.1: Output of Radar Module when no object is in the field



DSO-X 2014A, MY52492059: Sun May 24 17:55:22 2015

1 8.60V/ 2 3 4 0.0s 50.00s/ Auto 1 30.7V

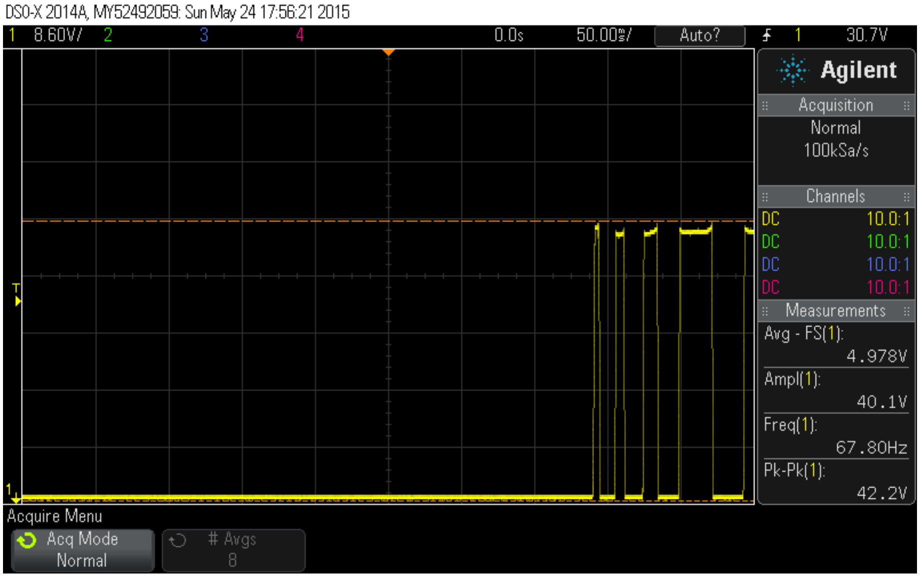
Acquisition: Normal 100kSa/s

Channels: DC 10.0:1, DC 10.0:1, DC 10.0:1, DC 10.0:1

Measurements:
Avg - FS(1): 9.170V
Ampl(1): 40.5V
Freq(1): 32.49Hz
Pk-Pk(1): 42.2V

Acquire Menu: Acq Mode Normal, # Avgs 8

Figure 7.2 Output of Radar Module when an object is in the field frequency shift of 32.49 Hz


DSO-X 2014A, MY52492059: Sun May 24 17:56:21 2015

1 8.60V/ 2 3 4 0.0s 50.00ms/ Auto? 1 30.7V

Acquisition: Normal 100kSa/s

Channels: DC 10.0:1, DC 10.0:1, DC 10.0:1, DC 10.0:1

Measurements:
Avg - FS(1): 4.978V
Ampl(1): 40.1V
Freq(1): 67.80Hz
Pk-Pk(1): 42.2V

Acquire Menu: Acq Mode Normal, # Avgs 8

Figure 7.3 Output of Radar Module when an object is in the field frequency shift of 67.80 Hz

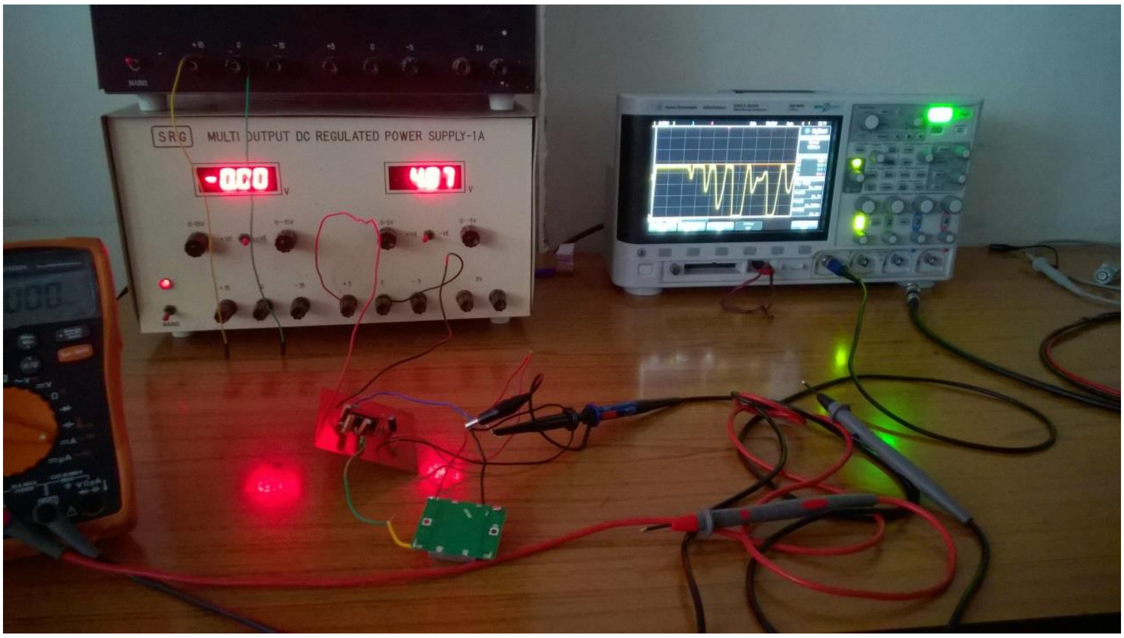


Figure 7.4 Laboratory setup of Radar Module depicting frequency shift object is in the field.

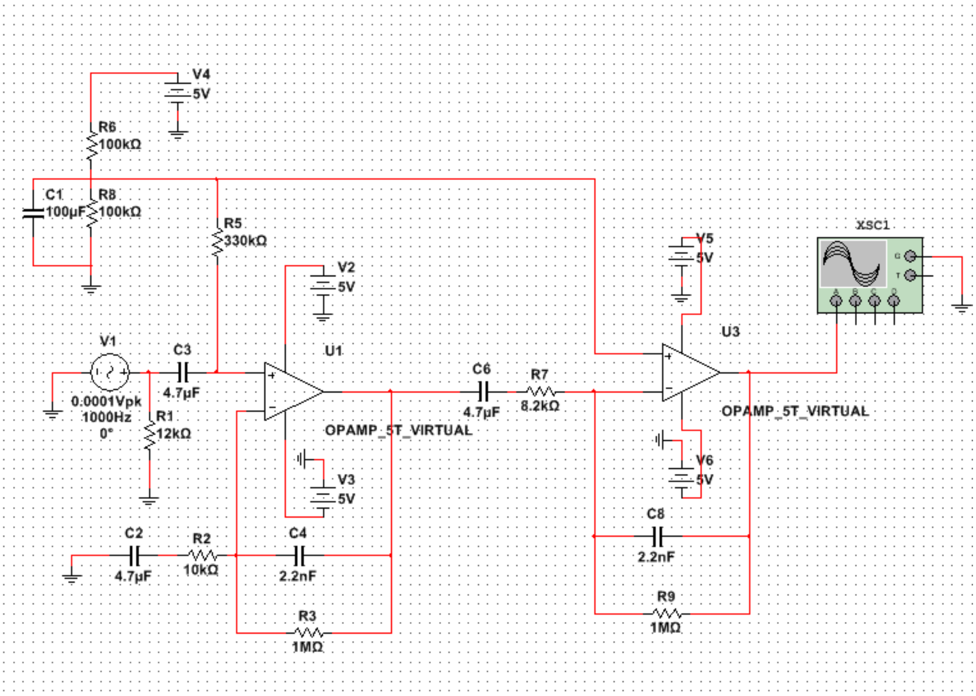
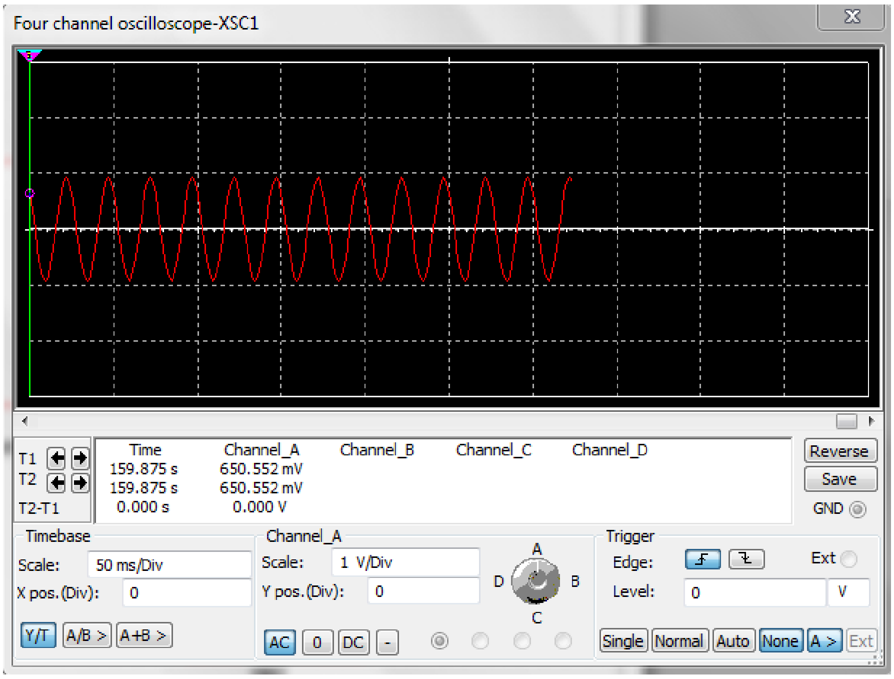


Figure 7.5 : Design of Radar Amplifier Circuit on MultiSIM


Four channel oscilloscope-XSC1

	Time	Channel_A	Channel_B	Channel_C	Channel_D
T1	159.875 s	650.552 mV			
T2	159.875 s	650.552 mV			
T2-T1	0.000 s	0.000 V			

Timebase
Scale: 50 ms/Div
X pos.(Div): 0

Channel_A
Scale: 1 V/Div
Y pos.(Div): 0

Trigger
Edge:
Level: 0 V

Figure 7.6: Simulation of Radar Amplifier Circuit on MultiSIM for high gain and high frequency bandwidth

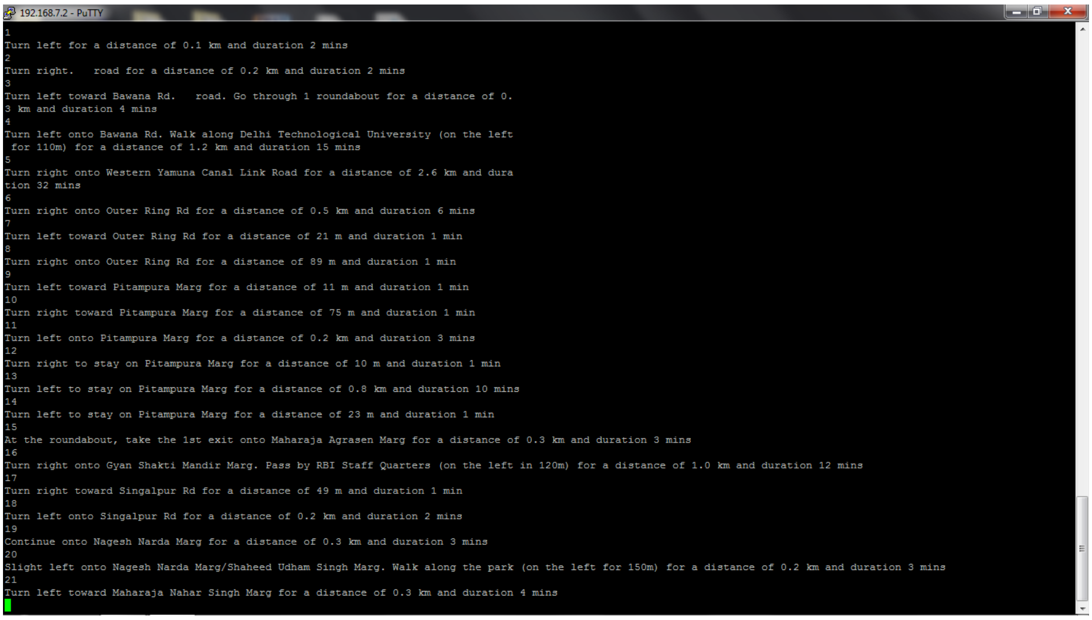
7.2 Discussion of Simulations results

The radar module generates a frequency shift of about 4.8Hz when the module is idle that is there is no object in the field of detection of Radar. This frequency shift is basically an error signal which is automatically removed whenever a moving object is present in field of detection of the radar. The observed frequency shift is then mapped using the frequency to speed conversion to provide the speed of moving objects. Various observed speeds are listed below in the following table.

S.No	Doppler Frequency (Hz)	Speed (km/hr)
1.	20.0	1.026
2.	52.08	2.672
3.	66.2	3.396
4.	101.53	5.209
5.	305.78	15.688

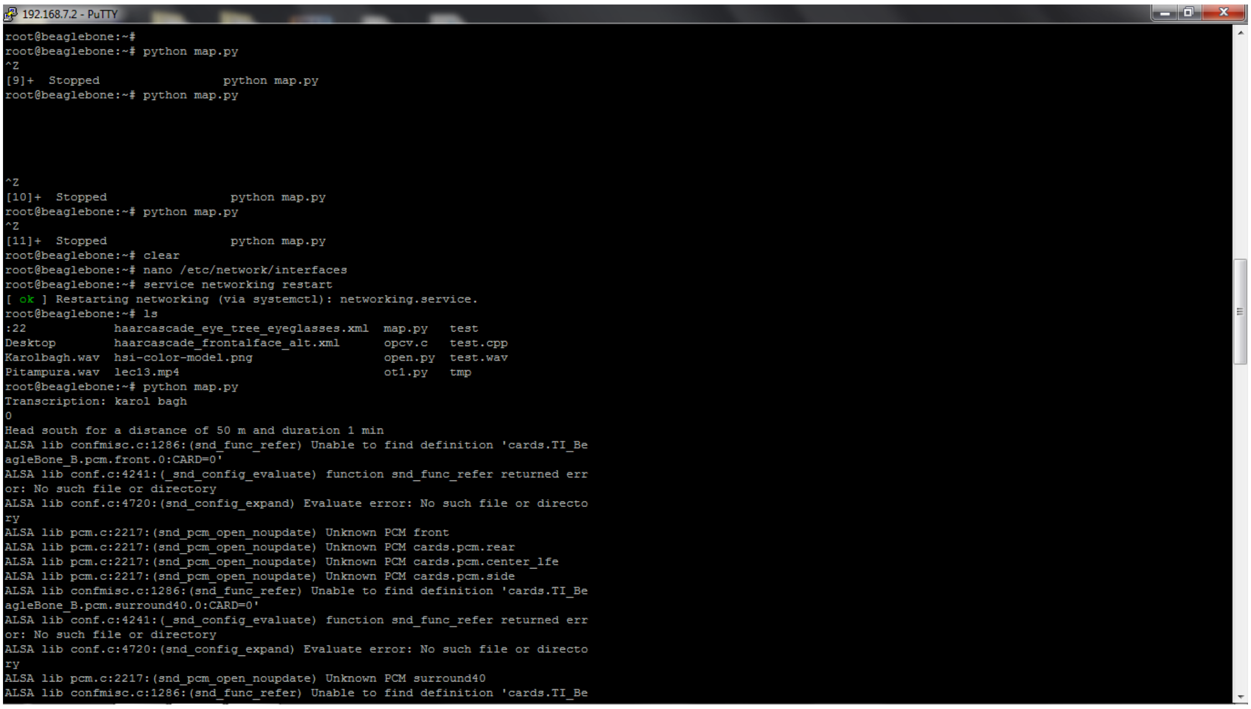
Table 7.1: Speed of objects that are detected corresponding to the frequency shifts generated.

7.3 Software Implementation of Voice based User Interface



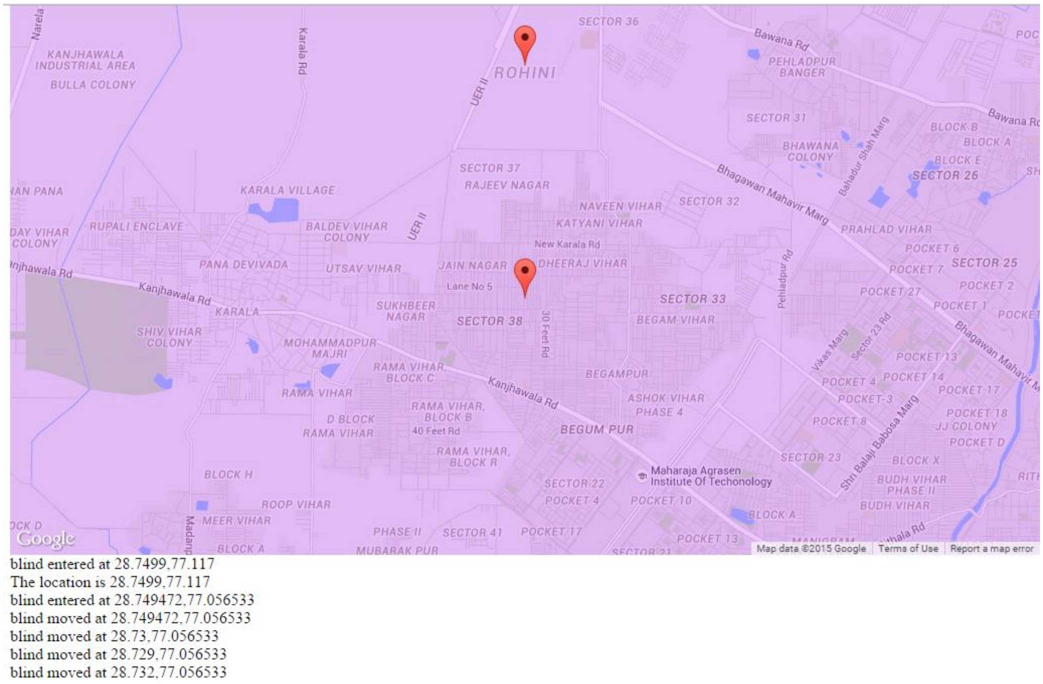
```
1
Turn left for a distance of 0.1 km and duration 2 mins
2
Turn right.   road for a distance of 0.2 km and duration 2 mins
3
Turn left toward Bawana Rd.   road. Go through 1 roundabout for a distance of 0.
3 km and duration 4 mins
4
Turn left onto Bawana Rd. Walk along Delhi Technological University (on the left
 for 110m) for a distance of 1.2 km and duration 15 mins
5
Turn right onto Western Yamuna Canal Link Road for a distance of 2.6 km and dura
tion 32 mins
6
Turn right onto Outer Ring Rd for a distance of 0.5 km and duration 6 mins
7
Turn left toward Outer Ring Rd for a distance of 21 m and duration 1 min
8
Turn right onto Outer Ring Rd for a distance of 89 m and duration 1 min
9
Turn left toward Pitampura Marg for a distance of 11 m and duration 1 min
10
Turn right toward Pitampura Marg for a distance of 75 m and duration 1 min
11
Turn left onto Pitampura Marg for a distance of 0.2 km and duration 3 mins
12
Turn right to stay on Pitampura Marg for a distance of 10 m and duration 1 min
13
Turn left to stay on Pitampura Marg for a distance of 0.8 km and duration 10 mins
14
Turn left to stay on Pitampura Marg for a distance of 23 m and duration 1 min
15
At the roundabout, take the 1st exit onto Maharaja Agrasen Marg for a distance of 0.3 km and duration 3 mins
16
Turn right onto Gyan Shakti Mandir Marg. Pass by RBI Staff Quarters (on the left in 120m) for a distance of 1.0 km and duration 12 mins
17
Turn right toward Singalpur Rd for a distance of 49 m and duration 1 min
18
Turn left onto Singalpur Rd for a distance of 0.2 km and duration 2 mins
19
Continue onto Nagesh Narda Marg for a distance of 0.3 km and duration 3 mins
20
Slight left onto Nagesh Narda Marg/Shaheed Udham Singh Marg. Walk along the park (on the left for 150m) for a distance of 0.2 km and duration 3 mins
21
Turn left toward Maharaja Nahar Singh Marg for a distance of 0.3 km and duration 4 mins
```

Figure 7.7: Snapshot of Terminal Window depicting the navigation commands received by the user



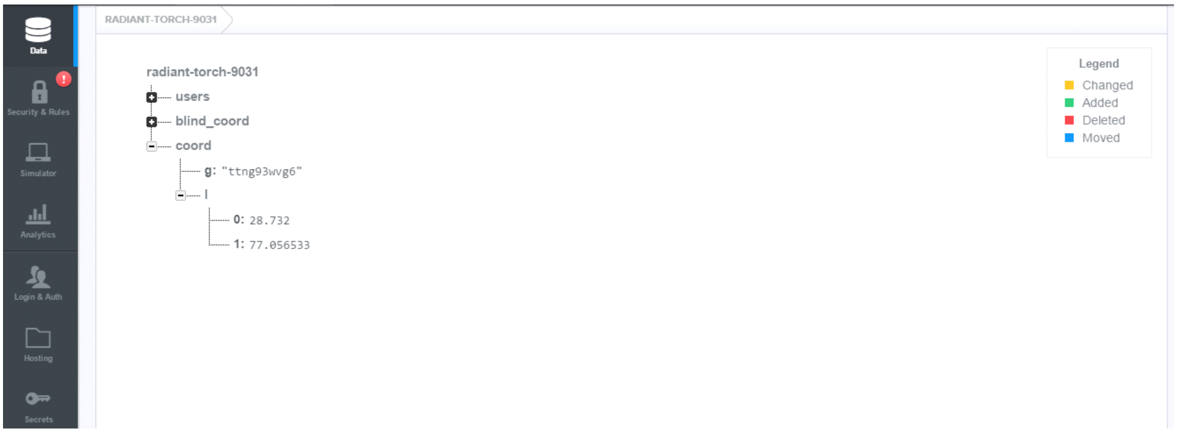
```
root@beaglebone:~#
root@beaglebone:~# python map.py
^Z
[9]+  Stopped                 python map.py
root@beaglebone:~# python map.py





^Z
[10]+  Stopped                 python map.py
root@beaglebone:~# python map.py
^Z
[11]+  Stopped                 python map.py
root@beaglebone:~# clear
root@beaglebone:~# nano /etc/network/interfaces
root@beaglebone:~# service networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
root@beaglebone:~# ls
:22            haarcascade_eye_tree_eyeglasses.xml  map.py   test
Desktop        haarcascade_frontalface_alt.xml      opcv.c   test.cpp
Karolbagh.wav  hsi-color-model.png                  open.py  test.wav
Pitampura.wav  lec13.mp4                            ot1.py   tmp
root@beaglebone:~# python map.py
Transcription: karol bagh
0
Head south for a distance of 50 m and duration 1 min
ALSA lib confmisc.c:1286:(snd_func_refer) Unable to find definition 'cards.TI_Be
agleBone_B.pcm.front.0:CARD=0'
ALSA lib conf.c:4241:(_snd_config_evaluate) function snd_func_refer returned err
or: No such file or directory
ALSA lib conf.c:4720:(snd_config_expand) Evaluate error: No such file or directo
ry
ALSA lib pcm.c:2217:(snd_pcm_open_noupdate) Unknown PCM front
ALSA lib pcm.c:2217:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2217:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2217:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib confmisc.c:1286:(snd_func_refer) Unable to find definition 'cards.TI_Be
agleBone_B.pcm.surround40.0:CARD=0'
ALSA lib conf.c:4241:(_snd_config_evaluate) function snd_func_refer returned err
or: No such file or directory
ALSA lib conf.c:4720:(snd_config_expand) Evaluate error: No such file or directo
ry
ALSA lib pcm.c:2217:(snd_pcm_open_noupdate) Unknown PCM surround40
ALSA lib confmisc.c:1286:(snd_func_refer) Unable to find definition 'cards.TI_Be
```

Figure 7.8: Snapshot of Terminal Window depicting the on board voice recognition



```
blind entered at 28.7499,77.117
The location is 28.7499,77.117
blind entered at 28.749472,77.056533
blind moved at 28.749472,77.056533
blind moved at 28.73,77.056533
blind moved at 28.729,77.056533
blind moved at 28.732,77.056533
```

Figure 7.9: Real time location tracking on Google Map Webpage using Firebase



```
radiant-torch-9031
  users
  blind_coord
  coord
    g: "ttng93wvg6"
    l
      0: 28.732
      1: 77.056533
```

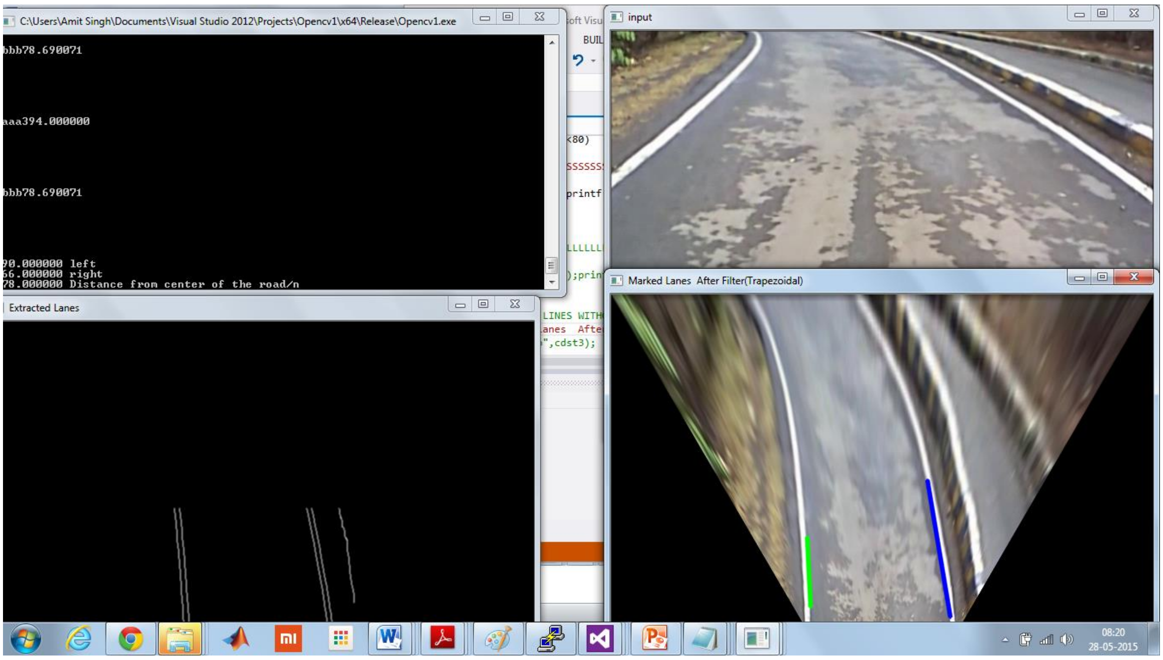
Figure 7.10: GPS coordinate sent to the online database

7.4 Discussion of Software Results of Voice User Interface

The terminal window shows the transcription that we got after passing the audio for KarolBagh to the system. We can clearly see the transcription in the picture. The next terminal window picture shows the directions to the destination being printed on the screen so that it can be demonstrated as a result.

The online repository where GPS coordinates are received and are mapped to the Google Maps is depicted in the next set of figures. It also shows the real time location of the user on a Map.

7.5 Software Implementation and results of image Processing


C:\Users\Amit Singh\Documents\Visual Studio 2012\Projects\Opencv1\x64\Release\Opencv1.exe
input
Extracted Lanes
Marked Lanes After Filter(Trapezoidal)
08:20
28-05-2015

```
bbb78.690071


aaa394.000000


bbb78.690071


90.000000 left
66.000000 right
78.000000 Distance from center of the road/n
```

Figure7.11: Output of Lane Detection and Mapping Program

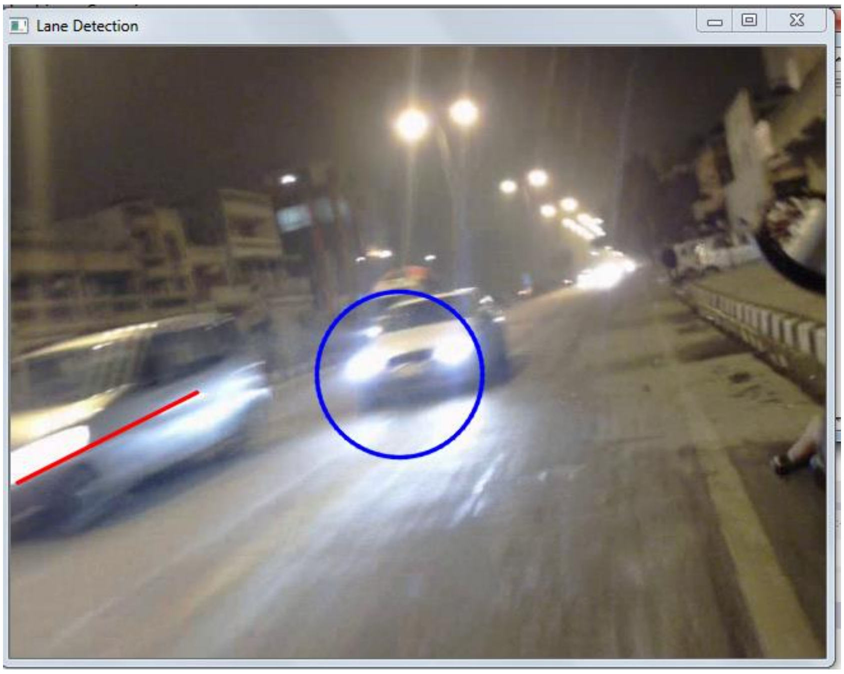


Figure7.12: Output of Vehicle detection Program

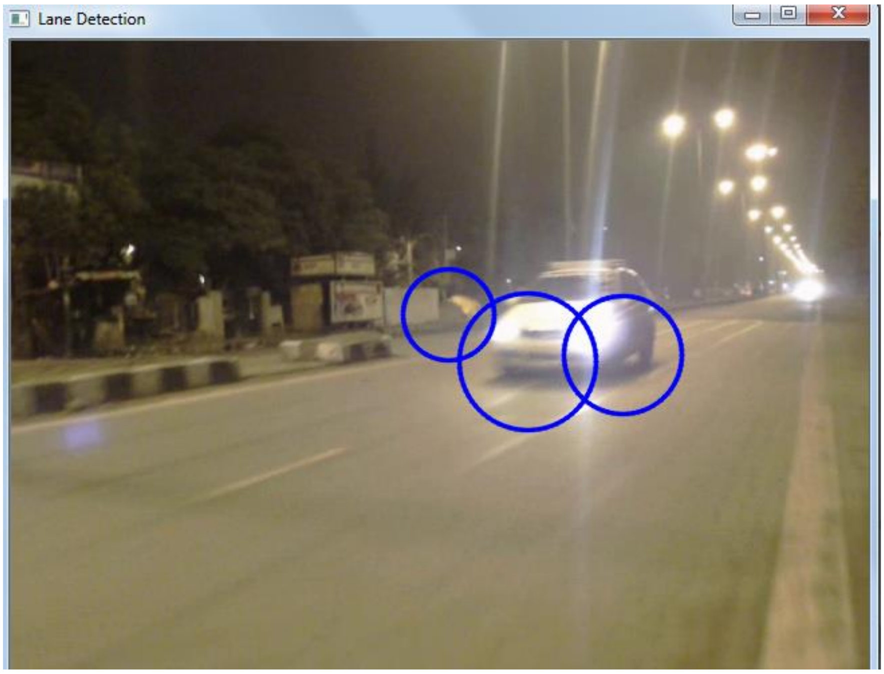


Figure7.13: Output of Vehicle detection Program

Chapter 8

Conclusions and scope of future work

8.1 Conclusions

This project creates a new product for the visually impaired that helps them navigate throughout any particular establishment by using the radar and ultrasonic sensor for avoiding obstacles. It provides them with low cost device that serves the multiple functionality of helping him navigate to his destination by following the voice commands from the headset connected to the cane. This project is a novel implementation of Internet of Things concept for a social cause. It utilises the vast capabilities that an active internet connection provides a discrete device by implementing complex navigation algorithms, voice based commands and text to speech conversion that run on powerful internet servers thus keeping the cost of device low as no such powerful system is required on board the cane.

This device also provides a solution to the crucial safety aspect of the visually impaired person by incorporating a feature that enables any authorised family member or guardian to track the location of the person. This feature also serves as a mechanism to locate the blind person in any emergency situation.

The prototype is very to use and has a flat learning curve and interfaces with user using audio commands and a set of push buttons. This prototype augments all the crucial needs of a visually impaired person at a very low cost.

8.2 Scope of future work

The future work in the our project is to design a complete ready to use Smart Cane with a proper battery charger system and online support through a server so as to provide guidance and directions over the phone to the user in case of any malfunction of the cane or is the user is out of internet coverage. We also intend to deploy speech conversion libraries that adds Hindi and other local languages for receiving and sending voice based commands so that are product has a wider addressable market. Another valuable addition to our product will be implementing multiple target detection and ranging using Pulse Doppler Techniques on the radar module.

References

[1].NMEA Sentences,“http://en.wikipedia.org/wiki/NMEA_0183”retrieved on 20/10/2014

[2]. Trilateration, “http://www.physics.org/article-questions.asp?id=55” retrieved on 25/1/2015

[3]. Recommended Minimum Information, http://www.GPSinformation.org/nmearetrieved on 25/3/2015

[4]. Stefan van der Spek ,Jeroen van Schaick , Peter de Bois , Remco de Haan . Sensing Human Activity: GPS Tracking 2009.

[5]. GPS Working Principle, http://www.pocketGPSworld.com/howGPSworks.phpretrieved on 19/4/2015

[6]. NMEA Format, http://www.nmea.orgretrieved on 3/9/2014.

[7]. Haversine Formula, http://vancouver-webpages.com/peter/retrieved on 17/2/2015.

[8]. GPS Format, http://GPSinformation.net/retrieved on 2/10/2014.

[9].Neo-6 Datasheet, https://www.u-blox.com/images/downloads/Product_Docs/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdfretrieved on 25/10/2014

[10]. Mohammed Altawim, Ahmed Alahmadi , Mohammed Bonais , Ben Soh , Fahad Algarni(2013),”Radar Vehicle Detector Mote”,[Online] http://www.erpublications.com/uploaded_files/download/download_06_04_2013_01_12_49.pdf

[11]”Introduction to Radar Systems”, “https://www.ll.mit.edu/workshops/education/videocourses/introradar/lecture1/lecture.pdf” retrieved on 25/10/2014

[12]. Agilsense, “HB100 Microwave Sensor Application Note”, Agilsense

[13]. Asha G Hagargund, Udayshankar R, Rashmi.N (2013),”Radar based cost effective vehicle speed detection using zero cross detection”,[Online] http://iraj.in/journal/journal_file/journal_pdf/1-9-139029869701-05.pdf, retrieved on 12/1/2015

[14]. Skolnik, M., Introduction to Radar Systems, New York, McGraw-Hill, 3rd Edition, 2001

[15]. Barton, D. K., Modern radar System Analysis, Norwood, Mass., Artech House, 1988

[16] Speech Recognition, "http://en.wikipedia.org/wiki/Speech_recognition" retrieved on 28/5/15

[17].Speech Synthesis, "http://en.wikipedia.org/wiki/Speech_synthesis" retrieved on 28/5/15

[18]. Trajectory Modelling, "https://wiki.inf.ed.ac.uk/twiki/pub/CSTR/TrajectoryModelling/HTS-Introduction.pdf" retrieved on 28/5/15

[19] Speech Recognition software for language learning, "http://scholarcommons.usf.edu/cgi/viewcontent.cgi?article=2908&context=etd" retrieved on 28/5/15

[20].Digital Image Processing,Gonzalez, Rafael C / Richard E.Woods, Prentice Hall

[21] Learning OpenCVby Gary Bradski and Adrian Kaehler, O’Reilly Publishing

[22] Robust Real-Time Face Detection, Paul Viola and Michael J. Jones, International Journal of Computer Vision 57(2), 137–154, 2004

[23] Canny Edge detector “docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html” retrieved on 28/5/2015.