IRIS Orientation Based Mobile Robot Steering Using Regional Division of Human Eye

Author

FAZAL UR REHMAN

Registration Number

00000172705

Supervisor

DR. YASAR AYAZ

DEPARTMENT OF ROBOTICS AND INTELLIGENT MACHINES ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

September 2020

IRIS Orientation Based Mobile Robot Steering Using Regional Division of Human Eye

Author

Fazal ur Rehman

Registration Number

00000172705

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics and Intelligent Machine Engineering

Thesis Supervisor:

DR. YASAR AYAZ

Thesis Supervisor's Signature: ____________________

DEPARTMENT OF ROBOTICS AND INTELLIGENT MACHINES ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

September, 2020

<u>National University of Sciences and Technology</u>

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: **Mr. Fazal ur Rehman Regn # 00000172705** Titled: **““IRIS Orientation Based Mobile Robot Steering Using Regional Division of Human Eye”** be accepted in partial fulfillment of the requirements for the award of **<u>MS Robotics & Intelligent Machine Engineering</u>** degree. **(Grade B)**

<u>Examination Committee Members</u>

1. Name: <u>Miss Sara Baber Sial</u> Signature: ________________

2. Name: <u>Dr. Muhammad Jawad Khan</u> Signature: ________________

Supervisor’s name: <u>Dr. Yasar Ayaz (HoD R & AI)</u> Signature: ________________

Date: ____________________

____
Head of Department

____
Date

<u>COUNTERSIGNED</u>

Date:__________

____
Principal

Thesis Acceptance Certificate

It is certified that the final copy of MS Thesis written by ***Fazal ur Rehman*** (Registration No. ***00000172705***), of Department of Robotics and Intelligent Machine Engineering (SMME) has been vetted by undersigned, found complete in all respects as per NUST statutes / regulations, is free from plagiarism, errors and mistakes and is accepted as a partial fulfilment for award of MS Degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in this dissertation.

Signature:______________________

Name of Supervisor: Dr. Yasar Ayaz

Date: _____12 October 2020_

Signature (HOD): ___________________

Date: ___________________

Signature (Principal): ___________________

Date: ___________________

Declaration

I certify that this research work titled *"IRIS Orientation Based Mobile Robot Steering Using Regional Division of Human Eye"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

FAZAL UR REHMAN

2016-NUST-MS-RIME-00000172705

Certificate for Plagiarism

It is certified that MS Thesis Titled: "Iris Orientation Based Mobile Robot Steering Using Regional Division of Human Eye**"** by **Mr Fazal ur Rehman** has been examined by us. We undertake the follows:

a. Thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph, or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.

b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results, or words of others have been presented as Author own work.

c. There is no fabrication of data or results which have been compiled /analyzed.

d. There is no falsification by manipulating research materials, equipment, or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.

e. The thesis has been checked using TURNITIN (copy of originality report attached) and found within limits as per HEC plagiarism Policy and instructions issued from time to time.

<u>Name & Signature of Supervisor</u>

Signature**:** ____________________

(Dr. Yasar Ayaz)

Copyright Statement

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which I setup in my mind to improve it. Indeed, I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express special thanks to my supervisor Dr. Yasar Ayaz for his help throughout my thesis and for teaching me different courses. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to thank Dr. Jawad Khan and Miss Sara Baber Sial for being on my thesis guidance and evaluation committee.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional parents whose tremendous support and cooperation led me to this accomplishment.

Abstract

Paralysis in human being is a major curse. Quadriplegia or Tetraplegia is a form of paralysis caused from injuries or illness results in partial or complete loss of all four limbs in this condition a patient can only move his/her eyes. Different types of interfaces with wheelchair had been introduce for paralyzed patients i.e. joystick control, chin control and head control. But many of the patients cannot even utilize these devices due to limited movement. Eye controlled wheelchair is also introduced but most of them are difficult to operate during steering due to IRIS positioning as well as camera positioning. In this work a solution for IRIS tracking and mobile robot steering is developed. For which first step is face detection, second eye extraction, third IRIS detection and regional division and fourth mobile robot control. Regional division of human eye is based on IRIS orientation for which we use individualized IRIS localization, in which first step is to find the natural IRIS orientation of patient. Initially subject is told to look straight, left, and right which are then used to mark the eye left positions in three regions, eye right positions in three regions and straight up to a certain limit depending on eye natural straight orientation. Then the marked regions are used to control steering angle of mobile robot. All the image processing was done on raspberry pi using open cv. Images were captured using Pi-cam. Raspberry pi after taking decision of IRIS localization in the eye frame send commands to the mobile robot using Bluetooth module. Arduino controlled mobile robot is developed which receives commands of movement from Raspberry Pi via Bluetooth which are then decoded and used to operate four-wheel robot.

Key Words: *paralysis, quadriplegia, human eye, IRIS tracking, regional division, mobile robot, wheelchair, steering angle, steering control*

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

EMG	Electromyogram
EEG	Electroencephalogram
MEMS	Micro-electro-mechanical systems
FOV	Field of view
HFOV	Horizontal field of view
VFOV	Vertical field of view
V-R	Vertical region
H-R	Horizontal region
BCI	Brain computer interface
ROI	Region of interest
HOG	histogram of orient gradient

CHAPTER 1

INTRODUCTION

Life of human beings is full of challenges and obstacles. For which we human tend to find solutions. Some people suffer from poverty, some from daily life challenges and others have health issues such as disability of their different body parts to operate for example weaken eye sight, cannot speak or hear, one of the disability kind is paralysis where human being is unable to move one or more parts of his/her body and dependent on others to help him/her in daily life challenges. Multiple solutions are designed for paralyzed people to help them in their daily life activity which can be mechanical or robotic. One of that solutions is development of Iris controlled wheelchair to help them move from one place to another which otherwise can be a real hassle.

1.1 Background, Scope and Motivation

Paralysis due to spinal cord injury is mostly divided into four parts which includes monoplegia, hemiplegia, paraplegia, and quadriplegia. Researchers have designed multiple working solutions for first three forms of paralysis but solutions for quadriplegia patients is still in research phase and number of people suffering from quadriplegia is increasing with time such as USA have 1.7% people [1] paralyzed one way or another. Tetraplegia, also called as **quadriplegia**, is a form of paralysis which is result of injury or illness of spinal cord due to which a person is unable to move his/her body because of the loss of all four limbs rather can only have control on his/her head movements. In tetraplegia both sensation and control of muscle are lost.

There are different ways that are used to help tetraplegia or quadriplegia patients in movement which includes Iris [2] and EEG [3] electroencephalography-controlled wheelchair designed for people who have complete loss of muscle control are suffering from extreme form of paralysis. Other than that Chin, EMG, and joystick-controlled wheelchair [4] are used for partially paralyzed people. For IRIS controlled wheelchair mostly, a camera is attached in front of eye [2] with the glasses or head mount that is used to capture images of eye. Then after image processing IRIS orientation is detected which is left, right, stop, forward and commands are transferred to motor controllers [2]. In EEG a head mount is attached on patients' head which receives signal based on brain activity and decides what commands signals to the motor controller are needed to be passed

[3]. Chin and Joystick control have same kind of operational system where in chin control a paralyzed person uses his chin [5] to operate wheelchair for which a joystick near chin is attached. While for simple joystick control system a small joystick is attached near the hand of patient that helps them generate control signal for controller [4].

In this work Iris orientation detection and command generation for steering control of mobile robot/wheelchair is targeted. Our scope is develop an Iris tracking algorithm that can help paralyzed people to steer mobile robot easily and steering control of robot should be dependent on regional division of eye so that an exact turn according to pupil direction can be possible. Eye region should be divided into multiple parts depending on Iris orientations and each region to be used to steering control or other control commands of mobile robot.

1.2 Objective and Problem Statement

Iris orientation (left, right, straight) changes with person to person and for paralyzed people it is difficult to steer wheelchair in left or right direction due to time-based turning [2]. One more issue that paralyzed people face during turning of wheelchair is unable to look in exact position where they want to turn due to presence of camera attached in front of one eye and predefined left or right turning Iris position which is extreme left or right in eye region. For that problem, a solution should be developed that can focus on steering control using Iris orientation in eye region. A turning angle should be decided based on pupil direction or looking angle instead of time we want to turn. Camera for Iris tracking should be contact free for ease of patient and to widen the angle of view.

1.3 Aims

To develop an Iris tracking algorithm that can help in bringing accuracy for steering angle of mobile robot. Give multiple turning angles options such as 30, 60,90 degree to paralyzed patients based on horizontal regional division of human eye as well as vertical regional division for stop and start. Develop a system which can do real-time image processing and minimum time lapse for eye localization to robot control.

1.4 Hypothesis

i. Regional division of human eye by using gaze tracking algorithm will enable patient to generate exact steering command in direction of gaze

ii. Based on above hypothesis steering control of mobile robot (wheelchair) will be possible in multiple angles.

1.5 Thesis Outline

Chapter 2 of this thesis reflects on literature review related wheelchair control.

Chapter 3 discuss about designed system architecture which includes IRIS detection, IRIS localization and decision-making process

Chapter 4 Hardware used and its specifications

Chapter 5 Experimental results

Chapter 6 Conclusion and future works

CHAPTER 2

Literature Review

This chapter thoroughly explains techniques from literature used to help paralyzed people, move from one place to another and architecture which are being used to take operational commands from patients results in helping them in movement.

2.1 Hand gesture technique for wheelchair movement

To control a wheel chair various technology has been used previously, for example hand gesture technique is used for wheel chair movement [6] in which researchers have used flex sensors to determine different gestures formed by person on wheelchair and actions are determined according to it, their developed system consisted to wheelchair equipped with multiple sensor for obstacle avoidance. Another work done on hand gesture-based wheelchair control is based on determining hand shapes contours [7]. Researchers used image processing to capture pre-defined shapes of hand which are for forward, right, left, reverse and stop.

2.2 Flex sensor-based wheelchair

In flex sensor-controlled wheelchair, a flex sensor is attached to finger of paralyzed person and different orientations are defined to resistance change when finger moves. A recent research work published in International Journal of Advance Research is gesture controlled wheelchair using flex sensor [8] in which inputs for wheelchair are captured using flex sensor and then commands are transferred to motor controller. One more work done is based on gyro sensor mounted to head using specially designed head mount [9]. Controls movement of wheelchair in forward, left, right and stop which takes input in the form of ASL codes which are decoded, and motors are controlled. Another paper published hand(accelerometer) and finger-based wheelchair [10] to operate wheelchair it uses Flex sensor and accelerometer, but main drawback is some human effort is required which is difficult for paralyzed patient Specially Quadriplegia.

2.3 Voice operated wheelchair

To minimize the use of body voice operated wheelchair are introduced such as [11] have developed a wheelchair which uses Arduino to receive voice commands from patient and move wheelchair accordingly. More advance work done in Japan which is tested on real patient with moving wheelchair [12]. They had a recognition of moving commands of 98.3 percent and 97 percent of verification commands. Voice commands were recognized on laptop and signal was transferred to motor controller microcontroller. There are several drawbacks as well for this technology i.e. accuracy, background noise speaking style etc.

2.4 EEG Based wheelchair

For quadriplegia patient one more solution is developed using EEG in which brain signals are decoded to generate movement commands such as [3] in which they used brain computer interface (BCI) with neuro sky EEG headset. Different command signals were generated based on patient attention level and for pause command two times eye blink was used. One more review study on BCI based wheelchair control [13] used one hundred articles which consisted of different papers relevant to BCI and wheelchair. They suggested multiple outputs at the end of their studies which are development of low cost, more accurate EEG headset. Wheelchair operation in dynamic environment. Improvement in results by increasing samples for testing.

2.5 Joystick Controlled Wheelchair

Joystick controlled wheelchair is designed for partially paralyzed people when their lower body is paralyzed while they have control over their upper body especially arms. Such as [14] developed joystick-controlled wheelchair to assist movement in which they used Arduino and dummy wheelchair to present idea. More developed solution is presented in [15], in which they presented idea of visual joystick in which they removed real hardware instead used image processing to decode different hand gestures for movement of wheelchair. Another research published [4] developed advance wheelchair in which they incorporated different environment scanning sensors to avoid dynamic obstacles as well while taking commands from joystick.

2.6 Eye movement based electronic wheelchair

For patients who are completely paralyzed or effected by quadriplegia and unable to move, require advance systems to help them move for which IRIS movement based electronic wheelchairs are developed. Such as researcher in paper [16] used MATLAB for image processing on laptop and then generated control signals are transferred to wheelchair for movement operations. Another research published in scientific and technology journal in 2014 [17] have used camera attached in front of eye using glasses which sends input to laptop where all the image processing is taking place and commands are generated for wheelchair control. Recently a research published [18] uses matlab for image processing and labview for taking decision for wheelchair control for which they have used a four-wheel robot as a model with two tires powered while other two are free. Which Main difficulty for user is that every time they must carry their laptops along with wheelchairs which difficult as well as costly. Similar technique is done using raspberry pi [2]. Image processing is rapidly increasing day by day and eye tracking in image processing is a technique where eye movements are determined. Another research published [17] uses IRIS tracking for left, right, forward and stop of wheelchair incorporated with environment sensing using IR sensors. Eye movement are used as control signals, there are several techniques used to track eye movements. In [8] wheelchair is controlled using Eye and a switch. Camera is mounted in front of eye to capture images. Researchers have also used raspberry pi with pi cam such as [2] have controlled wheelchair using raspberry pi in which image processing was done using opencv.

2.7 IRIS Tracking Algorithm

For quadriplegia patients, most researchers are focusing on IRIS tracking based algorithm because it is mostly noninvasive and have high accuracy [19]. The method used for recording eye position or movements is called oculography [20]. There are four main techniques which are widely used for eye tracking [19]. First one is electro-oculography in which small sensors are attached around eye which measure electric field when eye rotates. This is system is very effective to measure eye movement when person is sleeping, and it is widely used for that purpose. Second widely used system is scleral search coils in which coils are embedded in contact lenses and eye movement is measured using change in electric field. Third method which is widely used for eye position and movement tracking is infrared oculography which measures light intensity reflected from sclera illuminated by infrared light [19]. But due to advancement of technology and processing power of computers fourth system video-based eye tracking have increased drastically as a result accuracy and cost is getting effective day by day [19]. Researchers have developed multiple solution focusing on video-based eye tracking which consists of single, multiple camera-based eye tracking, in these kind of research mostly if single camera is used it is attached in front of one eye using head mount and if it is multiple cameras then they are present in front of person to track eyes. Invasive and non-invasive system, visible light or infrared light depends in researcher needs from output, environment variables and lightening conditions. There are two main gaze tracking techniques [17] which are widely used first one is feature-based gaze estimation in which distinctive features set of human eyes such as contours (pupil contours), eye corners, cornea reflection. This system consists of model-based approach which uses 3D gaze direction vector and interpolation-based approach which uses mapping to find gaze coordinates from image features. Second method which is widely used is gaze estimation based on appearance. In this type of system photometric appearance are directly used to detect and track eyes [16] .

2.8 Human Eye

Human eye has three visible parts which are pupil, sclera, and Iris [21]. Pupil is responsible for the light entering eye. Sclera is protective layer of the eye that consists of white region which provides protection to optic nerve binding[ref] and IRIS is responsible for controlling the diameter and size of pupil, thus controls light reaching retina [22]. Eyeball movement is controlled by six muscles inside eye which are four recti muscle and two oblique muscle [23]. Field of view (FOV) of human eye is dependent on movement of recti and oblique muscles. FOV is total area where a person can observe environment using his or her eyes. FOV helps us to cover certain area instead of single point. In humans FOV consists of two monocular FOV that is two eyes, which are joined by our brain to form one single binocular FOV. For human eye total horizontal angle of view of both eyes from extreme left to extreme right is 190 degree [24] and for vertical view from top to bottom is 150 degree.

Chapter 3

Methodology

3.1 Concept

Using the literature discussed above human eye horizontal binocular FOV for human eye is divided into multiple regions which are seven different orientations of iris in eye region that is left 90 degree, left 60 degree, left 30 degree for extreme left, moderate left and minimal left respectively. Same is the case for right 90 degree, right 60 degree, right 30 degree for extreme right, moderate right and minimal right respectively. A specific region in straight direction is considered forward orientation of iris or gaze. As shown in figure no.3.2.

Vertical FOV of eyeball in eye region is between 130 degree to 150 degree for which two positions can easily be considered which are top and forward based on IRIS positioning as shown in figure no. 3.1.

Based on concept discussed above if iris position can be found in the eye, we can divide that region into multiple small regions to control steering of robot in exact gaze direction. This system will enable quadriplegia patients to generate steering commands exactly for the direction in which they want their wheelchair to turn.

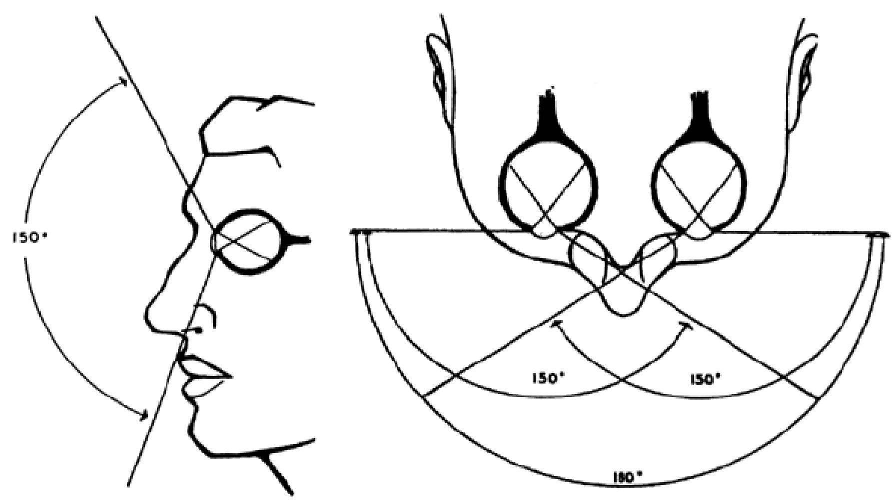


Figure 3.1: Horizontal and Vertical Field of View of Human Eye

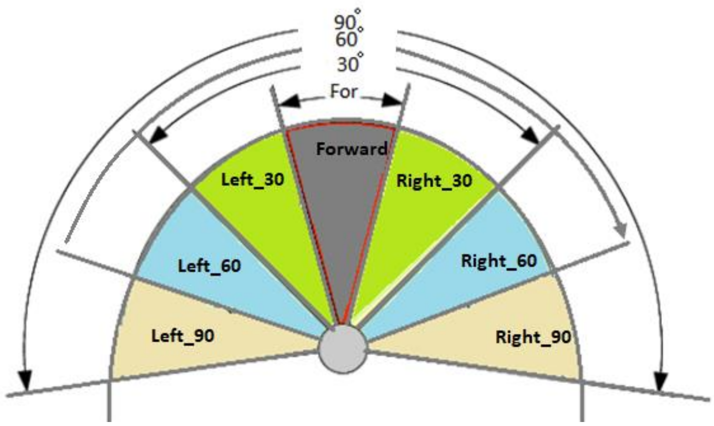


Figure 3.2: Regional Division of Human Eye

3.2 Iris Orientation Detection

For iris orientation detection in developed system following are the key features.

- Noninvasive technique for camera mount is used
- Camera is placed in front of patient face instead of in front of eye

There are multiple steps which are used till detection of orientation of iris, as discussed below.

1. Face detection and facial landmarks
2. Eye detection and extraction
3. Iris detection
4. Iris localization

3.2.1 Face Detection and facial landmarks

First step for gaze tracking is face detection in which histogram of orient gradient (HOG) is used. HOG descriptor key feature is to detect object in computer vision and image processing. This descriptor is used to find region of interest (ROI). For our case we used OpenCV with already face detector algorithm working based on HOG descriptor.

Once face is detected from image next step is to draw facial landmarks on face which represent different facial features which are face boundary, eyes, nose, mouth. For that purpose, pre-trained facial landmark detector library dlib is used. Dlib works on the principle of machine learning where training dataset consists of faces with manually input of dataset of faces with marked landmarks. Dataset used is IBUG 300w. Dlib estimates 68(x,y) coordinates which results in mapping of facial structure. Below is the figure no.3.3 representing output from two algorithms which are HOG and facial landmarks detector dlib.

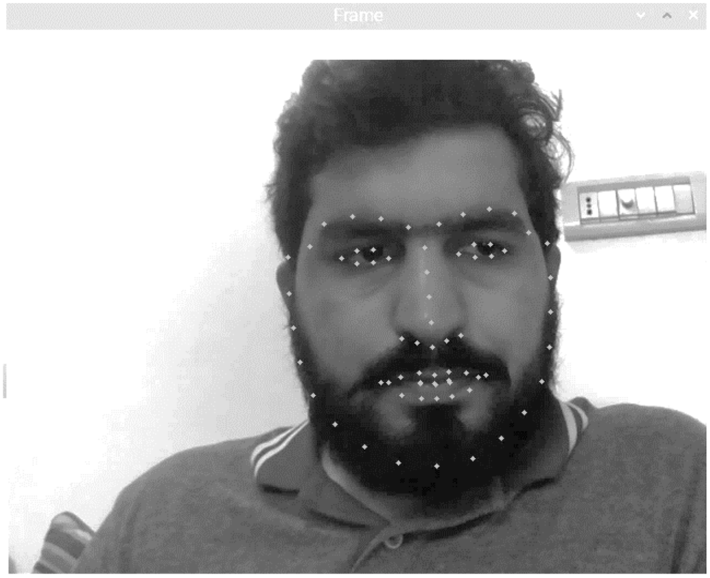


Figure 3.3: Face and Facial Landmarks Detection

3.2.2 Eye detection and Extraction

a) Eye-Detection

Each facial landmarks drawn represent separate (x,y) coordinates in the image. To extract any patch from the image we used landmarks bounding around specific region. For our case which were eyes. Coordinates for every landmark is predefined so extracted frame will also have landmarks representing same points which is

Left_eye_points = [36,37,38,39,40,41]

Right_eye_points = [42,43,44,45,46,47]

Once we get those points bounding around left and right eye then to determine that region equation (1) to determine that region.

- $\boldsymbol{region}$=$\boldsymbol{array[facial\_landmarks.partpoint.x,faciallandmarks}$ $region = array[facial\_landmarks.part(point).x, facial_{landmarks}.part(point).y]$ **(1)**

To extract region an array is formed in which input is points which are left eye points and right eye points. Using those marked points x coordinates and y coordinates of those specific points is find out and left and right eye is marked in the image.

b) Eye Extraction

Once the region containing eye is marked and determined next step is to extract eyes from the image so that processing speed of the system can be increased and instead of working on whole image we can work on small patch of image.

Step – 1 : Apply Mask to Get Only Eye

Mask applied is fill poly mask which enables us to draw a boundary around any region which can be in the form of black or white pixels.

First step is to take frame of black pixels of same size as original frame for that purpose height and width of real time frame is found out using pre-defined in OpenCV which is below equation (2)

$$h,w = frame.shape \quad (2)$$

Then black frame is taken using equation (3)

$$blackframe=numpy.zeros h,w,pixelintesity h=frameheight \; black_{frame} = numpy.zeros\left((h,w), pixel_{intesity}\right) \quad (3)$$

$where$

$pixel_{intesity} = 255$

$h = frame_{height}$

$w = frame_{width}$

$w = framewidth$

Once we get black frame of same image frame size then we apply that mask on frame captured which results in black frame overlapping original frame using equation (4).

$$mask = numpy.full(h, w, boundry_{color}, pixel_{intensity}) \quad (4)$$

Next step is applying fillpoly mask around eye region for that purpose opencv function fillypoly is used as shown in equation (5).

$$cv2.fillpoly(mask, [region], (color)) \quad (5)$$

Where

Region is area around eye which is marked using eye points

While color is the color of fillpoly

Once fillpoly is applied next step is to invert every pixel to white except eye mask so that we can have a clear output for visualization as shown in equation (6). Output of single eye is shown in which result of fillpoly mask is shown in figure no.3.4.

$$cv2.eyebitwise_{not}(black_{frame}, frame_{copy}(\quad), mask = mask) \qquad \textbf{(6)}$$

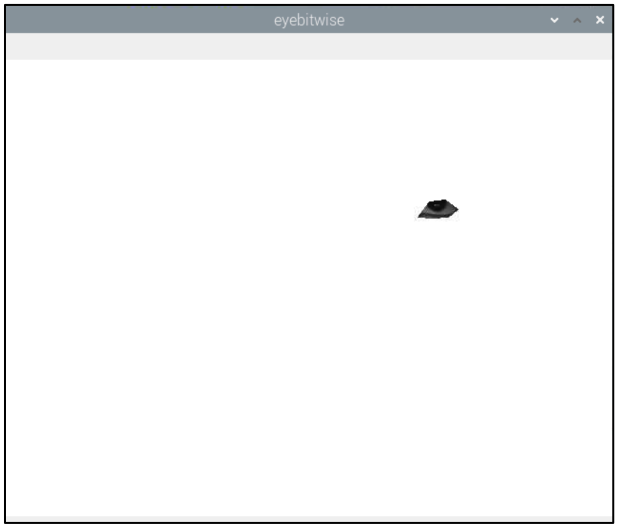


Figure 3.4: Eye Region After Fill Poly Mask

Step-2: Crop Eye

After applying mask and doing eye bitwise not operation we have output shown in figure no.3.4. Next step is to extract eye from frame. For that purpose, we extract an eye patch from our figure no. using x and y pixels value. To mark the region, we used a minimum and maximum of x and y which will cover complete eye as shown in equation no (7).

$$eye = frame[minimumy : maximumy, minimumx : maximumx] \qquad \textbf{(7)}$$

where

- $minimumx = minimumregion:, \mathbf{0}$ // 0 for x
- $maximumx = maximumregion:, \mathbf{0}$
- $minimumy = minimumregion:, \mathbf{1}$ // 1 for y
- $maximumy = maximumregion:, \mathbf{1}$

From figure no.3.5 show an eye is extracted from original frame while removing all other facial features. Both eyes are extracted separately using the same equation but different landmark points.

Figure 3.5: Extracted Eye

3.2.3 IRIS Detection

a) Binarization of Eye Frame

To binarize an image there are multiple ways, but most effective way is to find best threshold that can give better results without extruding image properties. To find best single intensity threshold that separate image in two classes foreground and background. Using this system we try to find sweet spot from which if pixel value is greater than it, it is set to maximum value and if pixel value is smaller than it, pixel value is set to zero. OTSU thresholding continuously searches for minimizing of intra-class variance. Using equation (8), first input is gray scale eye frame which we extracted previously, second input is threshold, third maximum value of pixel if above defined threshold. Fourth value define method we want to apply for thresholding.

$$\boldsymbol{thr = cv2.threshold(eye_{frame}, threshold, 255, CV2.THRESH_{OTSU})} \qquad \textbf{(8)}$$

Once we find best possible threshold with minimum intra-class variance then we binarize our eye frame using that thresholding value found in equation (9).

$$\boldsymbol{iris_{frame} = image_{processing}(eye_{frame}, thr)} \qquad \textbf{(9)}$$

b) **Draw Contours**

Visualization of required contour is shown in figure no.3.6. Which will be extracted based on its size and color. After binarization our next step is to draw contours which are joining of continuous points to form a curve surrounding different objects with same pixel properties. By using equation (10) inputs are IRIS frame in binarized formed, second variable is contour retrieval mode, and third input is contour approximation method.

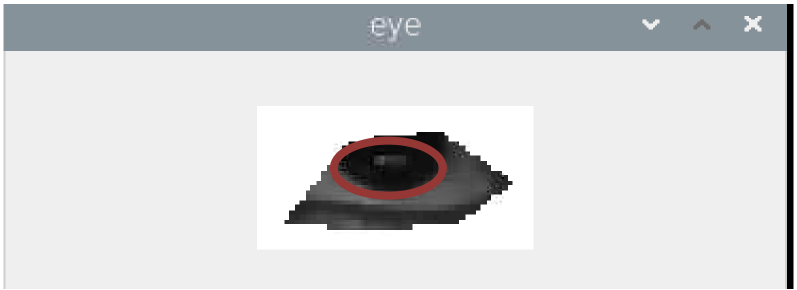


Figure 3.6: Required Output

$$\boldsymbol{contours_{,=cv2}\,.\,findcontours(iris_{frame}, cv2.RETR_{TREE}, CV2.CHAIN_{APPROX_{NONE}})} \quad \textbf{(10)}$$

Contours drawn are of different shapes and sizes which consists of multiple curves around any continuous region. To extract IRIS, we use a simple equation (11). Which sorts contour according to their size. In our binarized frame, IRIS is biggest continuous region which is on first position of sorted contours based on its size and approximate size. It is extracted while removing all other small contours as shown in figure no. 3.7 and 3.8.

$$\boldsymbol{contours = sorted(contours, key = cv = contour_{area})} \qquad \textbf{(11)}$$

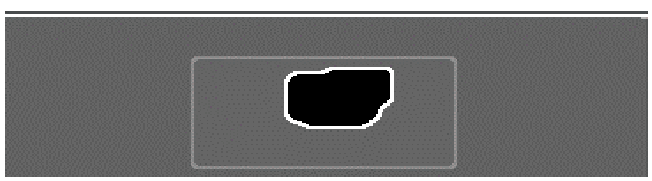


Figure 3.7: Extracted IRIS from Contours

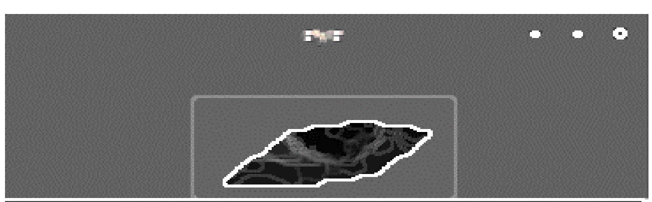


Figure 3.8: Human Eye with Before Contours

In figure no.3.9 results of IRIS extracted at multiple positions is shown for which center, left and right regional direction is observed.

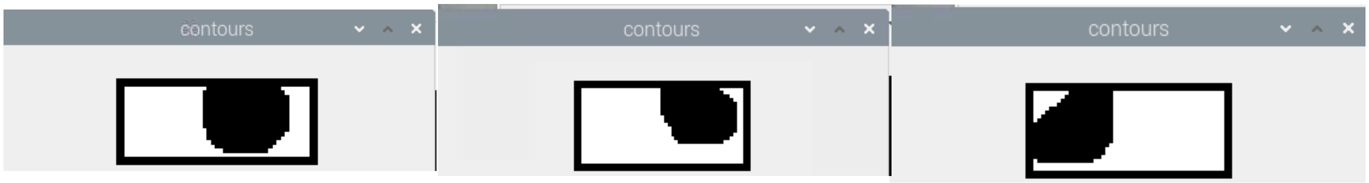


Figure 3.9: Eye Frame with Contour on Multiple Positions of IRIS

To find an average image pixel intensity we use image moments, or image moments are used to find interesting properties in an image. In images one of the main use of moments is to describe or find properties of contours for example center of mass, object area, centroid. For that purpose, we applied moments on contours extracted in equation (12).

$$\boldsymbol{moments} = \boldsymbol{cv2.moments}(\boldsymbol{contours}) \quad \textbf{(12)}$$

After finding moments next step is to extract specific information using moments in our case it was to find centroid of contour drawn around IRIS. For that purpose, we used maximum of moments for x and y divided by origin of moments as shown in equation no (13), (14).

$$\boldsymbol{x_{centroid}} = \boldsymbol{int}(\boldsymbol{moments}['\boldsymbol{m10}']/\boldsymbol{moments}['\boldsymbol{m00}']) \quad \textbf{(13)}$$

$$\boldsymbol{y_{centroid}} = \boldsymbol{int}(\boldsymbol{moments}['\boldsymbol{m01}']/\boldsymbol{moments}['\boldsymbol{m00}']) \quad \textbf{(14)}$$

Centroid points of IRIS are then added with the origin value to remove any error. Our output from equation (15) is x centroid of IRIS with reference to origin value of frame and from equation (16) output is y centroid with reference to origin of frame.

$$\boldsymbol{x} = \boldsymbol{Right}|\boldsymbol{left}_{\boldsymbol{eye}_{\boldsymbol{origin}_x}} + \boldsymbol{x}_{\boldsymbol{centroid}_{\boldsymbol{contour}}} \quad \textbf{(15)}$$

$$\boldsymbol{y} = \boldsymbol{Right}|\boldsymbol{left}_{\boldsymbol{eye}_{\boldsymbol{origin}_y}} + \boldsymbol{y}_{\boldsymbol{centroid}_{\boldsymbol{contour}}} \quad \textbf{(16)}$$

3.2.6.2 IRIS Centroid Horizontal Ratio

Horizontal ratio will generate left, right and forward commands. IRIS frame is used to find horizontal ratio of IRIS centroid. That ratio tells us exact position of IRIS with reference to origin of frame. To get a better visualization of results we normalized (scale data into smaller range) the range between 0 and 1 for both eyes IRIS separately.

For that purpose, we used three different values of x which are centroid of IRIS, other than that start, and end point of x is taken as range to be normalized. As shown in equation (17) and equation (18) for normalization of right eye

$$\boldsymbol{left_{normalized}} = \frac{(\boldsymbol{x} - \boldsymbol{x_{minimum}})}{(\boldsymbol{x_{maximum}} - \boldsymbol{x_{minimum}})} \quad \textbf{(17)}$$

$$\boldsymbol{right_{normalized}} = \frac{(\boldsymbol{x} - \boldsymbol{x_{minimum}})}{(\boldsymbol{x_{maximum}} - \boldsymbol{x_{minimum}})} \quad \textbf{(18)}$$

$$left_{normalized} = normalized\ value\ of\ left\ eye\ on\ horizontal\ axis$$

$$right_{normalized} = normalized\ value\ of\ right\ eye\ on\ horizontal\ axis$$

$$x = iris\ centroid$$

$$x_{min} = frame\ origin$$

$$x_{max} = maximum\ IRIS\ frame\ value$$

Find mean of both pupil

For a normal person with straight eyes or no squint we find a combined mean of eyes as a result it gives more accuracy and enable us to get better results using equation (19)

$$Horizontal_{mean} = \frac{(lefnormalized + right_{normalized})}{2} \quad \textbf{(19)}$$

3.2.6.3 IRIS Centroid Vertical Ratio

Vertical ratio will generate start, pause and stop commands. IRIS frame is used to find vertical ratio of IRIS centroid. That ratio tells us exact position of IRIS with reference to origin of frame. To get a better visualization of results we normalized (scale data into smaller range) the range between 0 and 1 for both eyes IRIS separately. For that purpose, we used three different values of y which are centroid of IRIS, other than that start, and end point of y is taken as range to be normalized. As shown in equation (20) for left and equation (21) for right.

$$left_{normalized_{vertical}} = \frac{(y - y_{minimum})}{(y_{maximum} - y_{minimum})} \quad \textbf{(20)}$$

$$right_{normalized_{vertical}} = \frac{(y - y_{minimum})}{(y_{maximum} - y_{minimum})} \quad \textbf{(21)}$$

$$left_{normalized_{vertical}} = normalized\ value\ of\ left\ eye\ on\ vertical\ axis$$

$$right_{normalized_{vertical}} = normalized\ value\ of\ right\ eye\ on\ vertical\ axis$$

$$y = iris\ centroid$$

$$y_{min} = frame\ origin$$

$$y_{max} = maximum\ IRIS\ frame\ value$$

Mean of both centroid(pupil)

For a normal person with straight eyes or no squint we find a combined mean of eyes as a result it gives more accuracy and enable us to get better results by using equation (22).

$$Vertical_{mean} = \frac{(left_{normalized} + right_{normalized})}{2} \quad (22)$$

Centroid point of eye frame

$$eye_{center} = \frac{w}{2}, \frac{h}{2} \quad (23)$$

3.3 Blinking Ratio

Blinking ratio is horizontal ratio divided by vertical ratio of human eye. For calculation we use gray scale eye frame in which six points are marked P(0,1,2,3,4,5).

For horizontal distance four points are used which are value of x and y pixel at P(0) and value of x and y pixel at P(3) shown in figure (3.10). Equation (24) and (25) gives us output of left and right points of each eye separately.

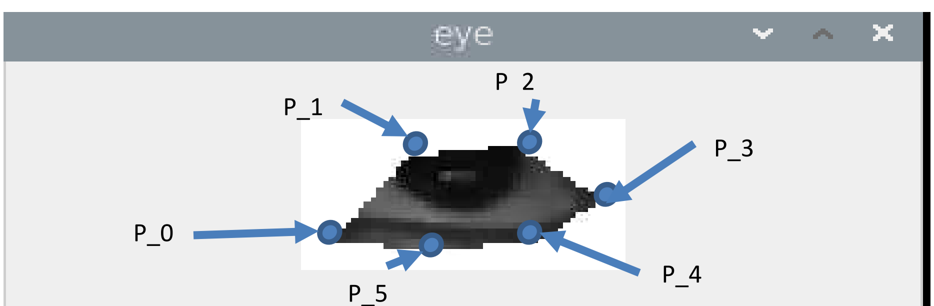


Figure 3.10: Eye Points Marking for Visualization

$$eye_{left} = landmarks.part(point[0]).x, landmarks.part(point[0]).y \quad (24)$$

$$eye_{right} = landmarks.part(point[3]).x, landmarks.part(point[3]).y \quad (25)$$

For vertical distance two points are used which are value of top P(1,2) and value of bottom P(4,5) shown in figure (3.10), then their centroid is of top and bottom points is found out which is used to find vertical distance. Equation (26) and (27) gives us output of top and bottom vertical points of each eye separately.

$$\boldsymbol{eye_{top} = landmarks.part(point[1]), landmarks.part(point[2])} \quad \textbf{(26)}$$

$$\boldsymbol{eye_{bottom} = landmarks.part(point[5]), landmarks.part(point[4])} \quad \textbf{(27)}$$

Eye horizontal distance is found out using equation (28) & (29).

$$\boldsymbol{x = eye_{left}[0] - eye_{right}[0]} \quad \textbf{(28)}$$

$$\boldsymbol{y = eye_{left}[1] - eye_{right}[1]} \quad \textbf{(29)}$$

Hypot function in python returns Euclidian norm which is refer to as distance between origin and coordinates. To find horizontal distance between points which as often refer as hypotheses of right-angle triangle we use equation (30).

$$\boldsymbol{eye_{width} = hypot(x, y)} \quad \textbf{(30)}$$

Same procedure is followed to find vertical distance from origin using equation (31),(32).

$$\boldsymbol{x1 = eye_{top}[0] - eye_{bottom}[0]} \quad \textbf{(31)}$$

$$\boldsymbol{y1 = eye_{top}[1] - eye_{bottom}[1]} \quad \textbf{(32)}$$

Vertical distance is also found using hypot with distance between top and bottom using equation (33).

$$\boldsymbol{eye_{height} = hypot(x1, y1)} \quad \textbf{(33)}$$

Then after finding out horizontal and vertical distance last step is to find eye blinking ratio which helps is in determining whether eye is closed or open for which we divided eye width by its height using equation (34),

$$\boldsymbol{ratio = \frac{eye_{width}}{eye_{height}}} \quad \textbf{(34)}$$

Figure 3.11: Eye Blink Detection Decrease in Eye Height

3.4 Decisions

All the decisions are taken based on horizontal, vertical, and blinking ratio for move forward, turn left, and turn right, pause, stop. Horizontal ratio is used for moving forward and steering while vertical ratio and blinking ratio is used for stop, pause and start. Table (3-1) shows the regional division range for IRIS in human eye on horizontal axis for generation of different commands according to subject (patient) IRIS orientation.

Table 3-1: Horizontal Regional Division Based Decision Range

Decisions	Regional Division Range	Command
Move Forward	0..46 <= H-R <= 0.55	Move forward
Turn Left	0.3 <= H-R <= 0.45	Turn 30° Left
	0.18 <= H-R <= 0.29	Turn 60° Left
	0 <= H-R <= 0.17	Turn 90° Left
Turn Right	0.56 <= H-R <= 0.70	Turn 30° Right
	0.71 <= H-R <= 0.85	Turn 60° Right
	0.86 <= H-R <= 1	Turn 90° Right

For start, stop and pause commands blinking ratio and vertical ratio is being used. For generation of stop command vertical ratio is defined based on patient (controller) IRIS orientation which can be changed according to person requirement. Decision taken requirement are shown in table (3-2).

Table 3-2: Vertical Regional Division & Blink Detection Range

Decisions	Regional Division Range	Requirement for Command Generation	Command Generated
Start	Blinking Ratio >= 0.45	Blink = 2 times	Enable Robot Movement
Stop	0 <= V-R <= 0.30		Disable Robot Movement
Pause	Blinking Ratio >=	Blink = 2 times	Pause Robot Movement

a) **Start**

To start generation of mobile robot control commands, paralyzed patient is prompted to blink eyes two times as shown in figure (). For which blinking ratio is found using equation (). Blinking ratio is adjustable according to person eyes size, environmental conditions such as light, camera placement, dark area around eyes. For every person, his individualized blinking ratio is found where they feel easiness for command generation. Based on patient easiness, trail, and error specific blinking ratio is defined for test subject.

b) **Forward**

Based on horizontal ratio region for generation of command for move forward is defined in IRIS frame. Which varies from person to person. For our test subject, forward region is between (0.46 - 0.55) so after enabling robot movement if patient brings his/her IRIS in that defined specific region command for forward moving will be generated.

c) **Stop**

For generation of stop command and to terminate code processing/ disable the system we have used vertical ratio which is found using equation (). Specific ratio for every subject is different which depends on his/her eye size, camera positioning, lightening, head angle to camera. The cut off point for command generation is marked by trial and error and saved in the code that is to be executed. Whenever subject brings his IRIS above certain region system disable the movement.

d) **Pause**

If a paralyzed person wants to pause movement of wheelchair for some time, he needs to blink his eyes two times which will pause all the system if already system is not paused. Blinking ratio is calculated using equation () which gives us the value between certain region. This ratio will be smaller if person is not blinking and increases when person blinks. Based on that it is determined whether person is blinking or not and command is generated.

e) **Left**

Using the horizontal ratio found by equation () turn left command is generated. Specific region is defined on left orientation of IRIS. There are three left directions are defined which are turn left 30, 60, 90 degree based on region identified

f) Right

Using the horizontal ratio found by equation () turn right command is generated. Specific region is defined on right region of IRIS. There are three left directions defined which are turn right 30, 60, 90 degree based on region identified.

3.5 Communication between raspberry pi and 4-wheel robot

When Raspberry Pi is powered on it continuously search for HC-05 mac address to connect as soon as mobile robot is powered up and raspberry pi connects with mobile robot Bluetooth module it enables eye tracking algorithm.

HC-05 is set to work as slave mode where it can only receive commands and connect with raspberry pi. Baud rate for communication is set to 9600. Following are the bits which are set for specific commands, generated by subject (paralyzed patient).

Bit 0 – Enable robot

Bit 1- Move Forward

Bit 2 – Turn Left 30 Degree

Bit 3 – Turn Left 60 Degree

Bit 4 – Turn Left 90 Degree

Bit 5 – Turn Right 30 Degree

Bit 6 – Turn Right 60 Degree

Bit 7 – Turn Right 90 Degree

Bit 8 – Pause

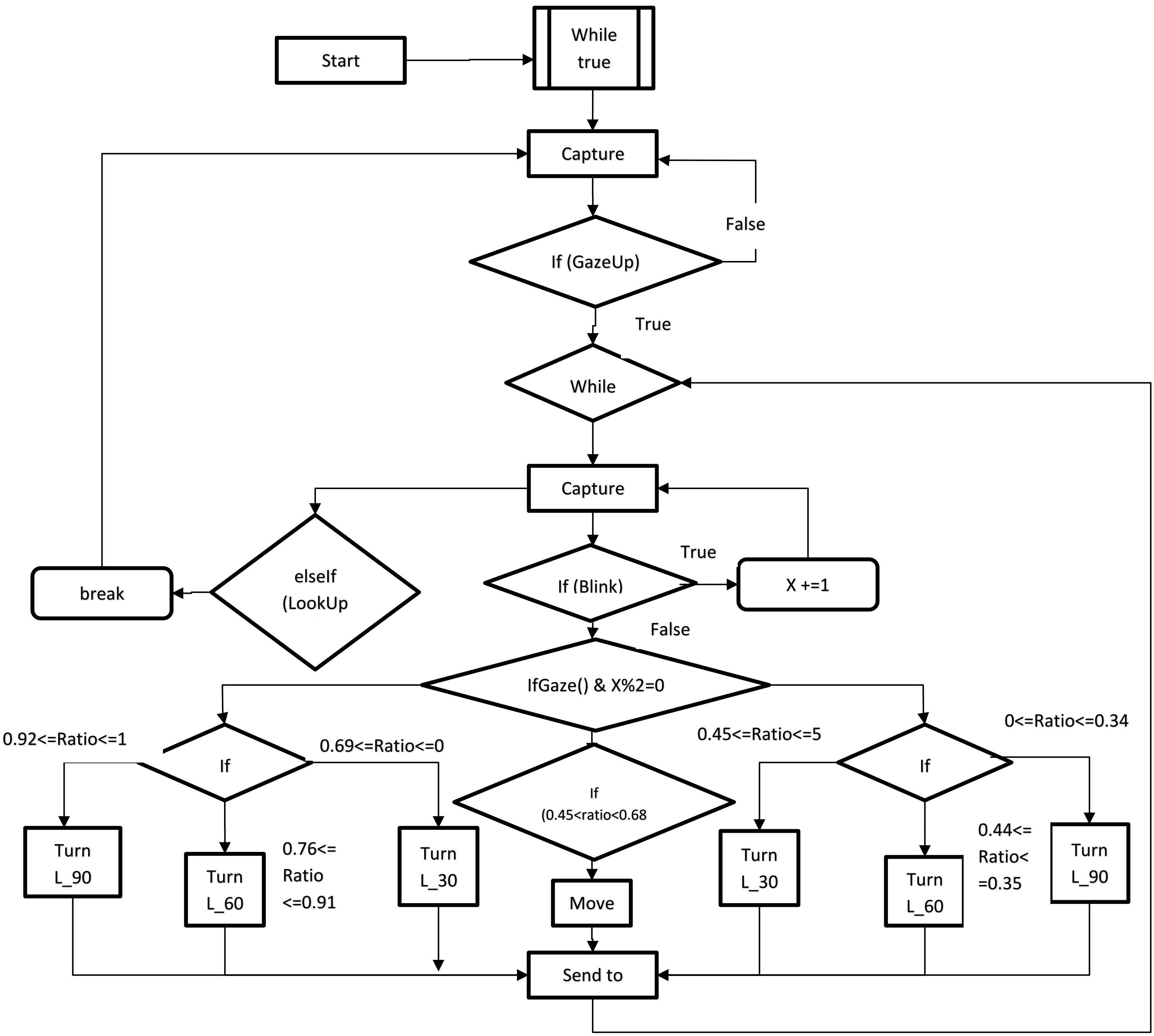
Bit 9 – Stop (Terminate)

3.6 Regional Division of Human Eye Decision Chart

For regional division algorithm decision chart-1 is shown in which algorithm enables as soon as subject looks upwards beyond defined gaze up region. Next step is to continuously scan IRIS if subject is not blinking or if blinked twice then move to next step which is movement control based on regional division.

For steering left three regions are defined and as soon as subject shifts his gaze in specific region commands of steer left are generated. Same is the case for steering right. For forward movement based on subject natural straight orientation region between 0.46-0.68. If at any time subject blinks system pause and if looks in up region algorithm execution stops.

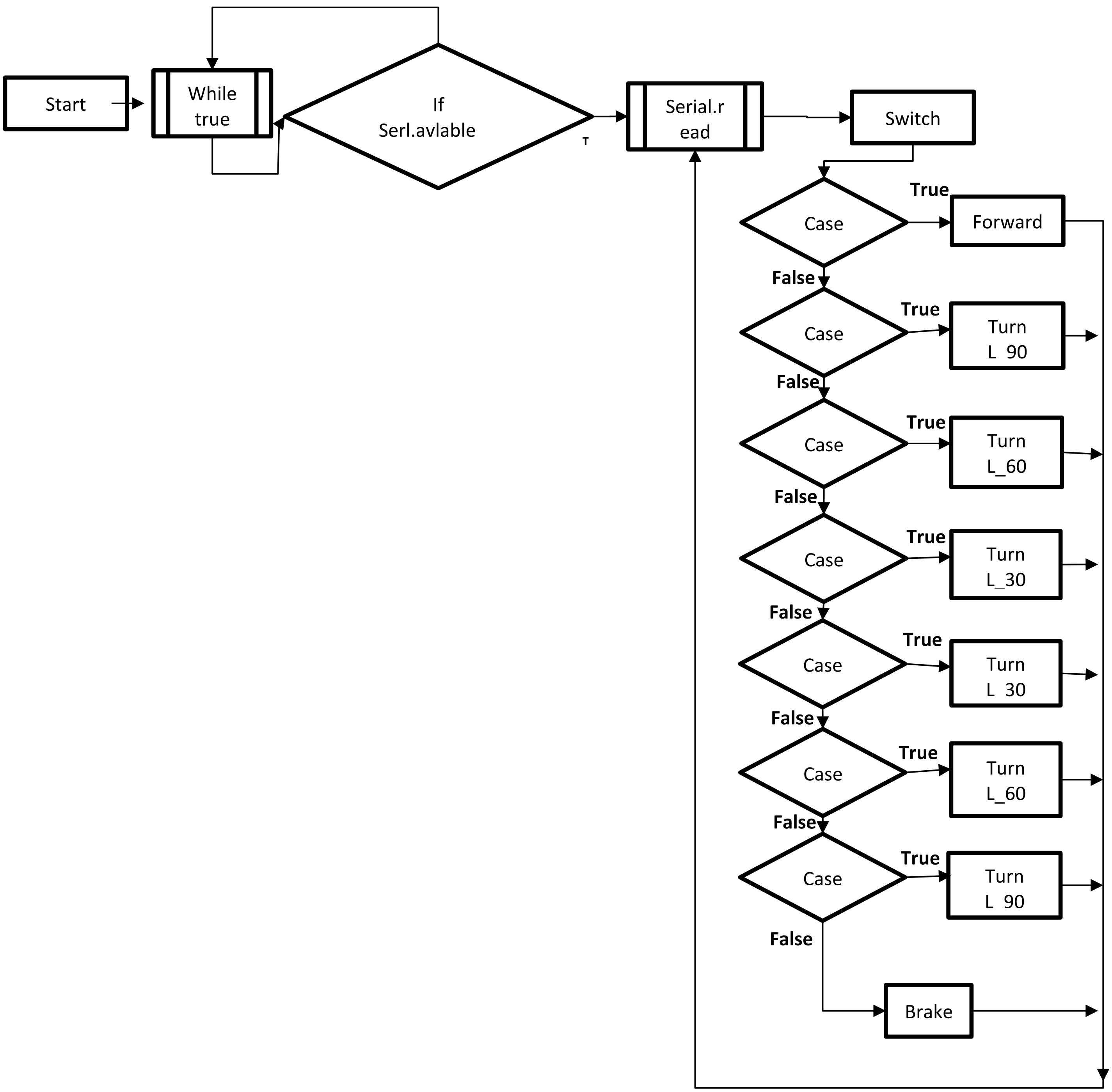
Flow Chart-1: Regional Division of Human Eye Decision



3.7 Mobile Robot Operational Decision Chart

As soon as Bluetooth connects with raspberry pi and data starts receiving on Bluetooth switch cases is made with separate function for each input bit. Depending on input case is executed and if none of the input is in the cases automatically brakes are applied to mobile robot.

Flow Chart-2: Mobile Robot Operational Decision



CHAPTER 04

HARDWARE

4.1 Raspberry Pi

Raspberry pi model 4b is used with ram of 6 GB and storge memory of 32 GB shown in figure no 4.1. This model can directly communicate with external LCD, LED or computer. We can also have a virtual display using wifi communication. It is small, low power consumption and easy to handle or attachment with wheelchair. This module consists of wifi, Bluetooth, usb input, ethernet, serial, multiple power input options.

Figure 4.1: Raspberry Pi Model 4b

4.2 Pi cam

5 MP camera is used to capture images using raspberry pi which is attached via flex cable with raspberry pi. All the communication take place using this cable no extra power input is needed.

Figure 4.2: Pi Cam 5mp

4.3 4-wheel robot chassis

To demonstrate experimental results four-wheel robot chassis is used with back wheel powered via DC motors.

Figure 4.3: Mobile Robot Chassis

4.4 Arduino Uno

For steering control of mobile robot Arduino Uno is used, which have multiple inputs and outputs such as digital, analogue, power, ground, UART, Serial.

Figure 4.4: Arduino Uno

4.5 Bluetooth module HC-05

Raspberry pi communicates with Bluetooth module HC-05 which receives data and transfer it to Arduino for further processing. Its input power is 5v with built in 3.3v regulator. TX and RX pins are used for serial communication with Arduino.

Figure 4.5: Bluetooth Module HC-05

4.6 IR encoder sensor

IR digital encoder sensor is used to control steering of mobile robot. Two encoders are attached each with back wheels with encoder disk having twenty high and low pulses generation. Based on those high pulses steering angle is determined.

Figure 4.6: Digital IR Encoder Sensor

4.7 Robot Control Circuitry

Robot circuitry is made shown in figure 4.7 using following points
12v DC rechargeable power source
Two +12v outputs one connected with motor controller and second connected with variable voltage regulator
Voltage regulator
Connects with Arduino uno at +5v, IR digital encoder sensor using +3.3v, Bluetooth module via +5v.
Arduino Uno
Input from voltage regulator +5v
Tx (voltage divider 3.3v), Rx pins connected with Bluetooth module HC-05
Six digital pins connected with motor controller (3 for each motor)
Two digital pins connected with IR sensor to receive input from them.
Motor Controller
Eight inputs six for control two for power (+12v, gnd)
Four outputs two for each motor connected with tires of robot.

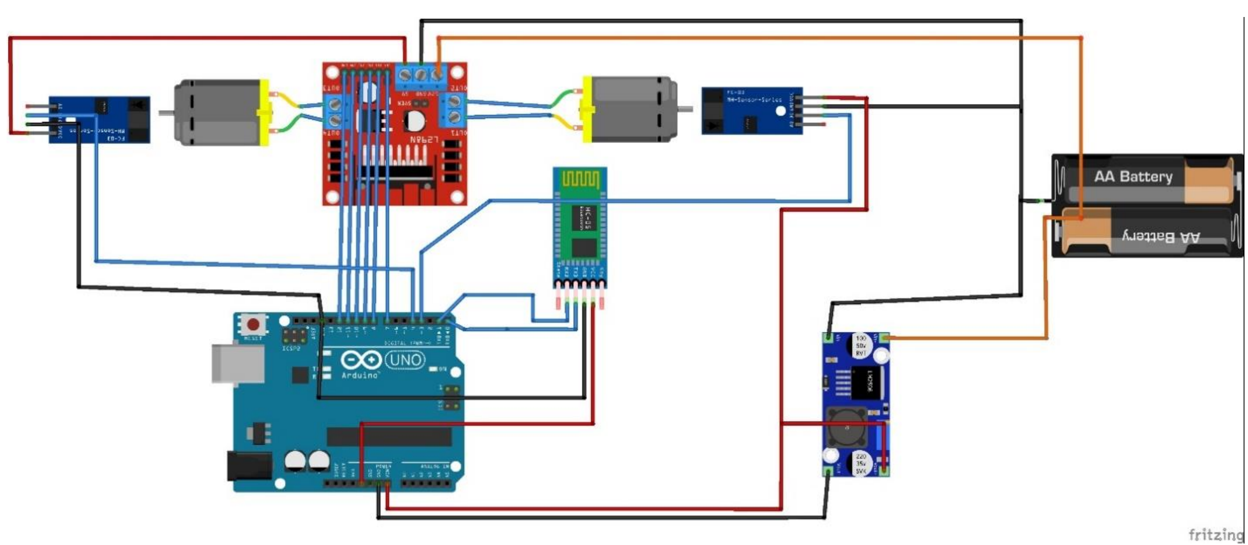


Figure 4.7: Electronic Circuitry

CHAPTER 05

EXPERIMENTAL RESULTS

Results section of thesis have been divided into two different parts. First part describes different results related to IRIS orientation detection, i.e., IRIS regional division which is left, right, forward and stop, second part describes system accuracy under different conditions. Third part describes results related to operation of 4-wheel robot according to command generated.

5.1 Regional Division Based IRIS Orientation Detection

For every orientation IRIS accuracy is found using real time samples for each command and their results are shown in table (). Sample for each command were recorded manually and separately to generate steering and control decision. To find out results only one command was executed multiple times. For horizontal regional division move forward, steer left 30, 60,90 and steer right 30, 60, 90 were selected. Vertical regional division was used to enable stop (disable). Eye blink was used to enable start or pause IRIS tracking. Results show accuracy of 92.3% while only 3.8% of the commands were inaccurately and 3.8% of the time given command was not recognized.

Table 5-1: Regional Division Based IRIS Tracking Accuracy

Decision	Movement	No of Samples	Correct	Incorrect	Undecided
Start		30	27	0	3
Forward	1 Meter	20	19	1	0
Left	30 Degree	20	20	0	0
	60 Degree	20	18	2	0
	90 Degree	20	17	1	2
Right	30 Degree	20	19	1	0
	60 Degree	20	17	3	0
	90 Degree	20	18	0	2
Pause		20	19	0	1
Stop		20	20	0	0
Total		210	194	8	8
Accuracy			92.3%	3.8%	3.8%

5.2 Start and Pause Command

As shown in table () start and pause command have zero percent incorrect detection rate. While only four times algorithm was unable to detect blink of an eye due to multiple reasons which are below

Fast blinking speed

Not opening eyes properly before blinking second time

Head orientation not in line with camera attached

Sudden change in lightening conditions

Figure 5.1: Eye Blink Detection with Pause Command

5.3 Stop

Horizontal regional division was used to detect only one command which was stop(disable) IRIS tracking due to which had enough region to divide our horizontal ratio into two parts as shown in figure () as a result we got hundred percent accuracy in results. For stop command algorithm was tested under different head postures and lightening conditions.

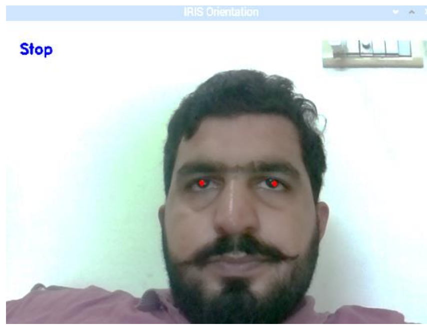


Figure 5.2: Stopped Command Generation When Patient Looks Up

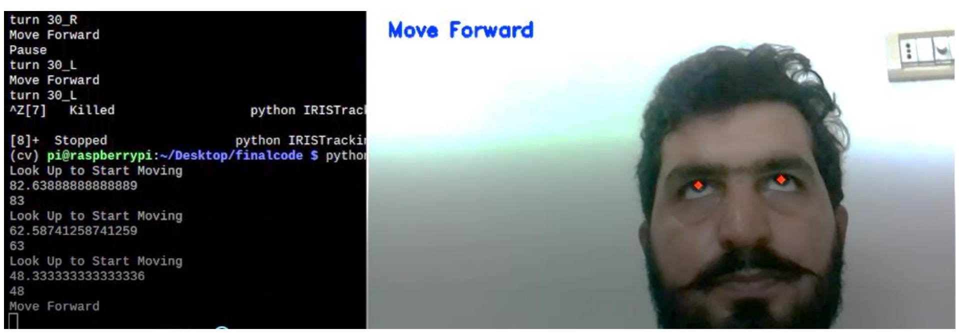


Figure 5.3: IRIS Tracking Enable Once Patient Looks Up

5.4 Steer Right

Three regions were identified based on horizontal regional division for steering right of mobile robot. Out of sixty different commands algorithm detected 54 correct IRIS orientation while only 3 commands were detected wrong due to following reasons.

Change in head posture

Change in lightening conditions

Incorrect system calibration

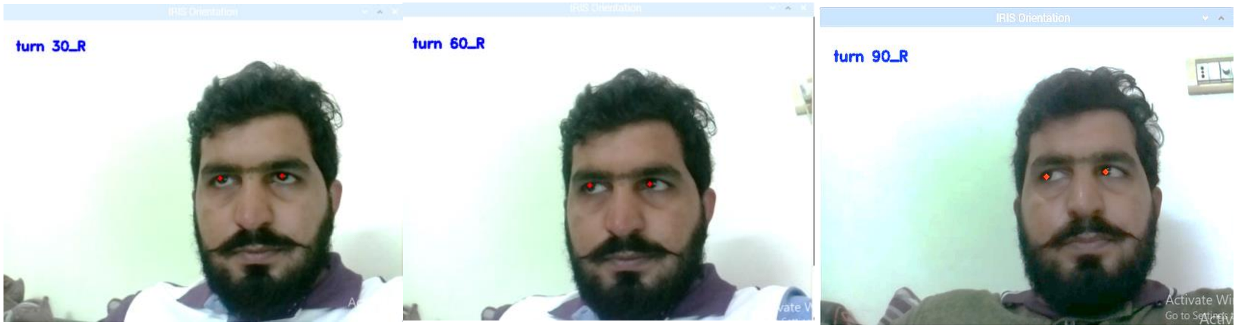


Figure 5.4: Steering Right Regional Position (Right 30 Degree, 60 Degree and 90

5.5 Steer Left

For steering left three regions were identified spam betweeen 0-0.55 divided according to subject (patient) ease of use. Only three IRIS regions were detected incorrectly due to reasons discussed previously. Results are shown in figure ().

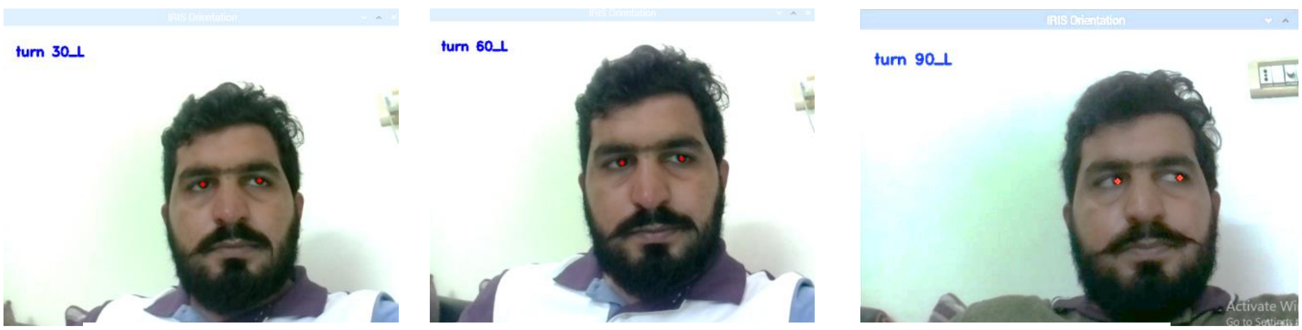
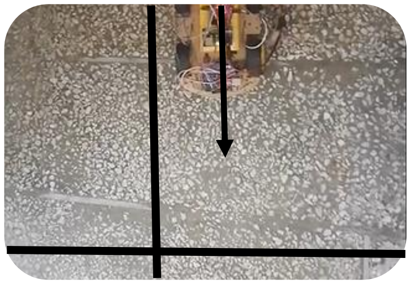


Figure 5.5: Steering Left Regional Position (Left 30 Degree, 60 Degree and 90

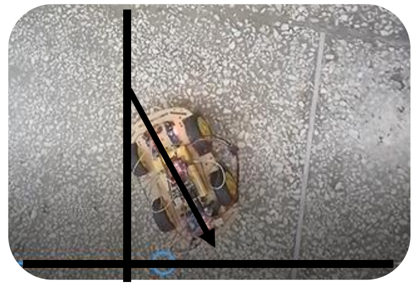
5.6 Real Time Robot Control

All the steering, move, pause, and stop commands generated were transferred to mobile robot using Bluetooth. Each generated command has separate digit representing it which was decoded correctly. Microcontroller was successful in decoding all the commands and generation of control signals which were passed to motor controller. Encoders were able to successfully detect steering angle and stop the robot.

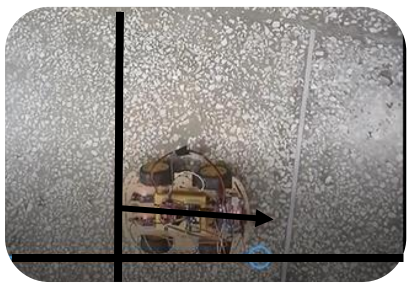
A small error in steering of robot occurred due to inertia and slippage of tires, it was observe only when control signal for robot was steer 90 degree in either direction because the more turning more inertia was created while for 30 degree and 60 degree negligible error occur.



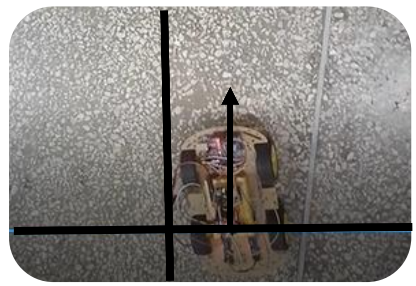
Robot In Straight Orientation



Robot After Turning 30 Degree



Robot After Turning 60 Degree



Robot After Turning 90 Degree

Figure 5.6: Robot Steering Control

CHAPTER 06

CONCLUSION AND FUTURE WORK

Quadriplegia patient suffering from all four limbs control loss can control wheelchair using their IRIS movement for that purpose multiple solutions are developed but recently researchers did not focused on steering control for that purpose solution was proposed which consisted of regional division of human eye focusing on turning in the direction towards which person wants. Eye regional was successfully divided into horizontal seven regions based on individualized IRIS positioning and vertical region was divided into two regions, both of those regional divisions can be changed according to patient natural IRIS orientation in eye. Proposed system successfully generated steering commands with accuracy of 94 percent. Mounted camera for IRIS tracking was non-invasive which reduced the hassle of irritation and broadening eye movement. Results suggest IRIS was successfully tracked by turning head up to 5 degrees which can further be increased. Mobile robot was successfully steered with few errors due to inertia and slippage of tires.

In future this developed algorithm should be tested on wheelchair incorporated with environment sensors on quadriplegia patient. Steering angle should be decreased below minimum 30 degree by refining developed algorithm. Undecided and inaccurate commands generated should be minimize by incorporating IRIS tracking with head turn. Speed of wheelchair should be controlled using the vertical regional division of human eye.

References

[1] E. A. P. Brian S. Armour, "Prevalence and Causes of Paralysis—United States,," *Research and Practice,* pp. 1855-1857, 2013.

[2] M. A. C. C. a. M. S. Jatin Sharma, "Iris movement based wheel chair control using raspberry pi," in *IOP Conference Series: Materials Science and Engineering, Volume 263, Issue 5*, 2017.

[3] M. A. u. R. N. M. Antora Dev, "Design of an EEG-Based Brain Controlled Wheelchair for Quadriplegic Patients," in *IEEE 3rd International Conference for Convergence in Technology (I2CT), 6-8 April 2018, Pune, India,* 2018.

[4] J. B. M. C. B. Trinayan Saharia, "JOYSTICK CONTROLLED WHEELCHAIR," *International Research Journal of Engineering and Technology (IRJET),* vol. 4, no. 7, 2017.

[5] M. &. Z. C. &. I. T. &. T. M. &. P. J. Shibata, "Improvement of a Joystick Controller for Electric Wheelchair User," *Modern Mechanical Engineering,* vol. 5, pp. 132-138, 2015.

[6] S. S. K. v, "MEMS based Hand Gesture Wheel Chair Movement Control for Disable Persons," *International Journal of Current Engineering and Technology* , vol. 5, no. 3, pp. 1774-1776, 2015.

[7] N. T.-T. D. C. Tien Kim Nguyen, "A Method for Controlling Wheelchair Using Hand Gesture Recognition," in *Robot Intelligence Technology and Applications 2012*, 2012.

[8] L. H. K. S. K. G. Narendra Kumar Sura, "A FLEX SENSOR BASED GESTURE CONTROLLED WHEELCHAIR," *International Journal of Advanced Research,* vol. 8, no. 2, pp. 1175-1177, 2020.

[9] S. M. A. Auns Al-Neami, "Controlled Wheelchair System Based on Gyroscope Sensor for Disabled Patients," *Biosciences Biotechnology Research Asia* , vol. 15, no. 4, pp. 921-927, 2018.

[10] V. Shah, "Advanced Wheel chair," in *2nd National Conference on "Emerging Trends in Engineering, Technology & Management*, 2015.

[11] S. G. K. Khyati Meena, "Voice Controlled Wheelchair," *International Journal of Electronics, Electrical and Computational System,* vol. 6, no. 4, 2017.

[12] T. S. K. Masato Nishimori, "Voice controlled intelligent wheelchair," in *SICE, Annual Conference*, 2007.

[13] 1. N. E. I. a. RahibH.Abiyev, "Brain-Computer Interface for Control of Wheelchair Using Fuzzy Neural Networks," *BioMed Research International,* p. 9, 2016.

[14] J. B. M. C. B. Trinayan Saharia1, "JOYSTICK CONTROLLED WHEELCHAIR," *International Research Journal of Engineering and Technology (IRJET),* vol. 4, no. 7, 2017.

[15] 1. M. M. a. F. F. Yassine Rabhi, "Intelligent Control Wheelchair Using a New Visual Joystick," *Journal of Healthcare Engineering,* 2018.

[16] N. A. N. G. K. Chaudhary, "Iris controlled wheel chair in real time," *Journal of Engineering and Applied Sciences,* pp. 75-79, 2018.

[17] G. S. K. K. C. S. S. D. Gunda Gautam, "Eye Movement Based Electronic Wheel Chair For Physically Challenged Persons," *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH,* vol. 3, no. 2, 2014.

[18] P. S. S. S. P. S. G. K. Ashwini A. Nashte1, "Automation of Wheelchair Using Iris Movement," *International Journal of Computational Engineering Research (IJCER)* , vol. 6, no. 3, 2016.

[19] X. Y. H.R. Chennamma, "A SURVEY ON EYE-GAZE TRACKING TECHNIQUES," *Indian Journal of Computer Science and Engineering (IJCSE),* vol. 4, no. 5, 2013.

[20] A. l. o. o. panelJ.S.Stahl, "Eye Movement Recording," in *Encyclopedia of the Neurological Sciences (Second Edition)*, 2014, pp. 245-247.

[21] A. Dey, "Perceptual Characteristics of Visualizations for Occluded Objects in Handheld Augmented Reality," University of South Australia, 2013.

[22] B. Adithya, B. N. P. Kumar, H. Lee, J. Y. Kim, J. C. Moon and Y. H. Chai, "An experimental study on relationship between foveal range and FoV of a human eye using eye tracking devices," in *International Conference on Electronics, Information, and Communication (ICEIC)*, 2018.

[23] N. J. Wade, "Pioneers of eye movement research," *Iperception*, vol. 1, no. 2, pp. 33-68, 2010.

[24] 1. D. C. D. L. Y. W. WEITAO SONG, "Design and assessment of a wide FOV and high-resolution optical tiled head-mounted display," *Applied Optics*, vol. 54, no. 28, 2015.