Head Gesture Control Based Assistive Mobile Robot

FINAL PROJECT REPORT

Submitted by

KOTA TOMARU (N12935319)

SARVESH SIVAPRAKASAM (N11059298)

Under the guidance of

VIKRAM KHAPILA

(Professor, Department of Mechanical & Aerospace Engineering)

in partial fulfilment for the course

of

ROBOTS FOR DISABILITY

for

MECHATRONICS AND ROBOTICS ENGINEERING

of

TANDON SCHOOL OF ENGINEERING

11201, 6 MetroTech Center, Brooklyn, NY 11201

1. INTRODUCTION

Paralysis is the inability—whether temporary or permanent to move a part of the body. In almost all cases, paralysis is due to nerve damage, not to an injury to the affected region. The recent statistics regarding the paralytic patients is alarming. Approximately 1.7 percent of the U.S. population, roughly 5.3 million people reported they are living with some form of paralysis and every year 11,000 new cases are being reported. There are many causes for a paralysis but the most common cause for it is Stroke and Spinal cord injuries and they contribute to 33.7 percentage and 27.3 percentage respectively. Among the Spinal cord injured patients more than 50 percentage people are paraplegic and remaining are quadriplegic.

Quadriplegia is a total paralysis of the arms and legs requires damage to the limbs, but most quadriplegics have perfectly healthy legs and arms. This form of paralysis is inevitably the product of damage high in the spinal cord, usually in the cervical spine between C1-C7. The higher the injury is, the more extensive the damage will be, and very high spinal cord injuries are often immediately fatal. Paraplegia occurs in spinal cord injuries below the first thoracic spinal levels (T1-L5). Paraplegics can fully use their arms and hands, but the degree to which their legs are disabled depends on the injury. Some paraplegics are completely paralyzed from the waist down.

2. Existing Assistive Technology

One of the most common assistive technology for these kind of patients is Brain Controlled interface (BCI) to control the robots to assist in their daily activities. A BCI is a collaboration in which a brain accepts and controls a mechanical device as a natural part of its representation of the body. The main advantage of this kind of interface is that it can identify the electrical neural patterns as a thought - before the pattern has fully manifested in to a conscious feeling/command. Though this one of the promising future technology brain is incredibly complex and its difficult to process the neural signals. Additionally, the user needs to purchase costly equipment’s to control the robots.

Controlling Robots using User’s Voice is low cost and comparatively easy technology to BCI. The main disadvantage these voice recognition is that the system needs to be trained to recognize the individual voice for these kinds of application. Another major disadvantage is that most of the voice recognition system is in English and sizable population in world doesn’t speak English.

3. System Concept

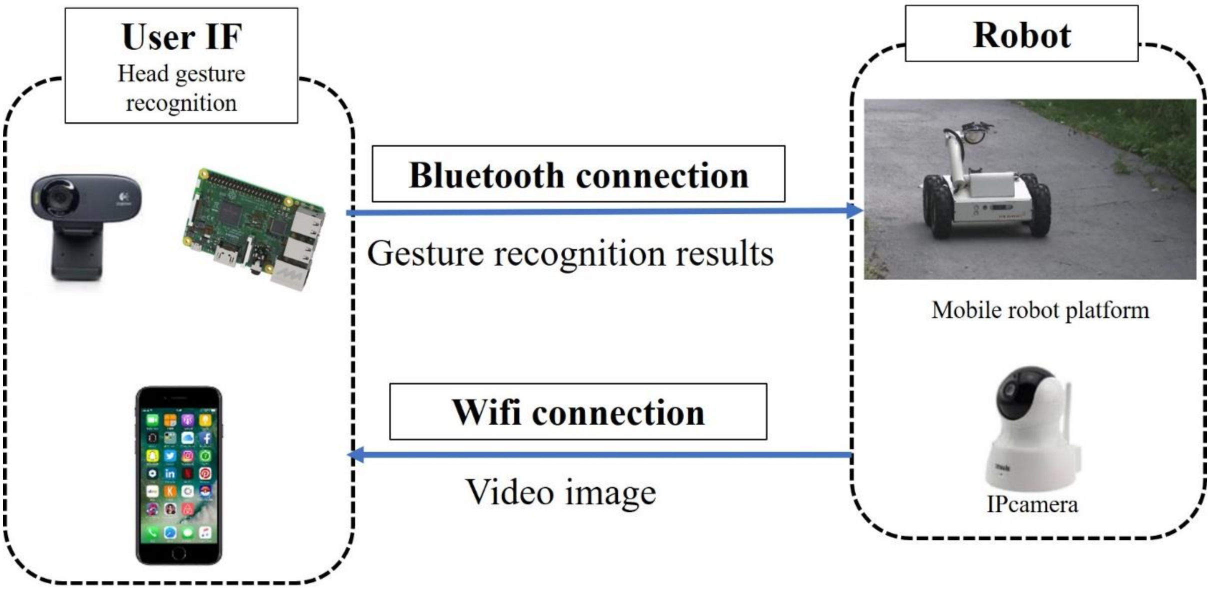
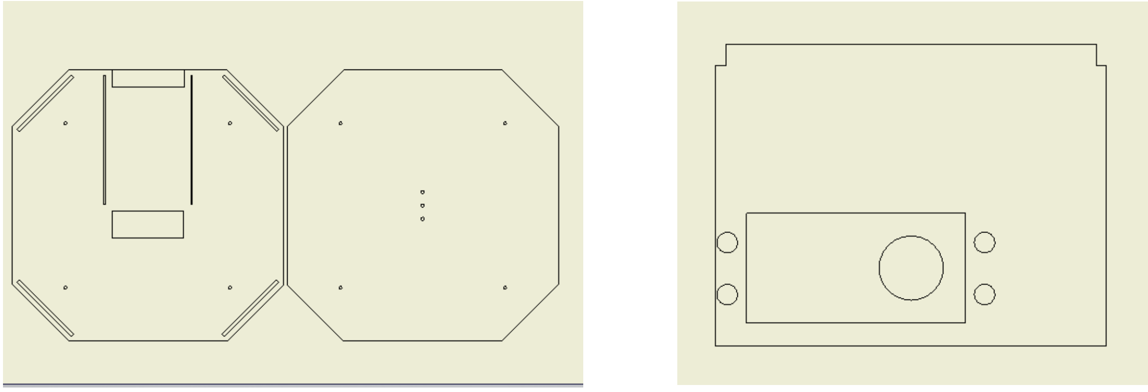


Fig. 1: System concept for assistive robot

Our project goal is to develop a hands-free interface to control the robot that the users don't need to equip any tools. In our project, user's Head Gestures is detected using computer vision and with these detected gestures, the control for the mobile robot platform is mapped. For gesture detection, a USB camera connected with a Raspberry pi 3 is used in this project. So, this kind of system can be used by any kind of paralytic patients. This requires a mobile platform equipped with IP camera and a Bluetooth hardware to communicate with the user interface. At the user end, the user just has a phone where he can visualize the robot's movement and raspberry pi with a USB camera to detect the head gestures.

4. Hardware

The robot chassis was designed in Autodesk Inventor and was laser cut from plywood, the camera & servo mount from Plexiglass. Parallax continuous servo motor is used for the movement the robot. For the end user visualization IP camera is mounted on the robot. A 3000 mAh LiPo Battery is used to power the motors of the robot. Arduino Uno with a Servo motor shield and Bluetooth module is placed onboard to control the movement of the motors. Arduino Uno is powered by a battery pack. The IP camera is powered by a Powerbank.



(a) Chassis mount (b) Servo mount

Fig 2 Autodesk Inventor Design

Fig. 3 Actual Robot

Holonomic drive is used in this robot and it allows an operator to translate in any direction, independent of rotation. There are several ways to achieve this mechanically, such as using omni-wheels. Holonomic refers to the relationship between controllable and total degrees of freedom of a robot. If the controllable degree of freedom is equal to total degrees of freedom, then the robot is said to be Holonomic. A robot built on castor wheels or Omni-wheels is a good example of Holonomic drive as it can freely move in any direction and the controllable degrees of freedom is equal to total degrees of freedom. The Fig 4 shows a omni wheel which can rotate in both X-axis and Y-axis making it move in both the directions.

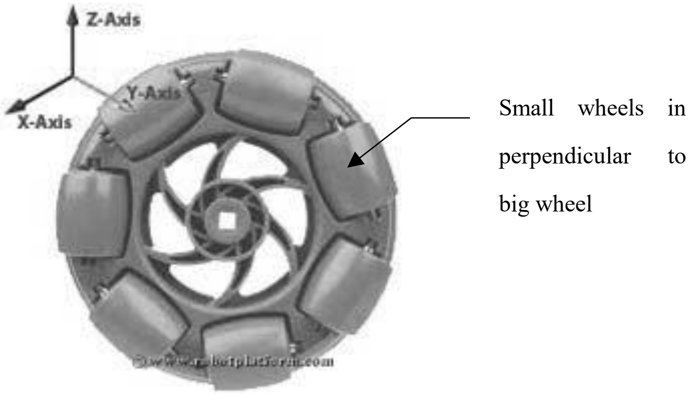


Fig 4: omni direction wheel

The Fig 5, 6 & table 1 explains the details motor movement to achieve tractions in different directions

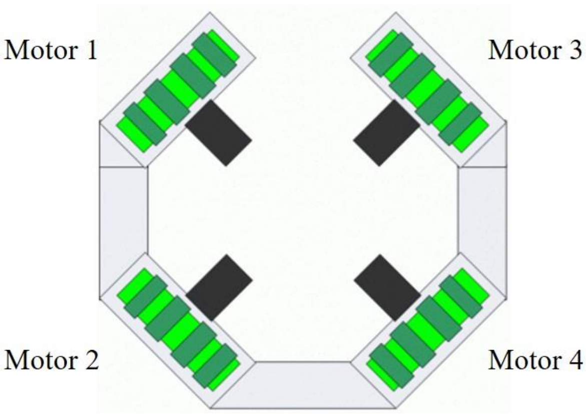


Fig 5: motor orientation (45 deg) in chassis

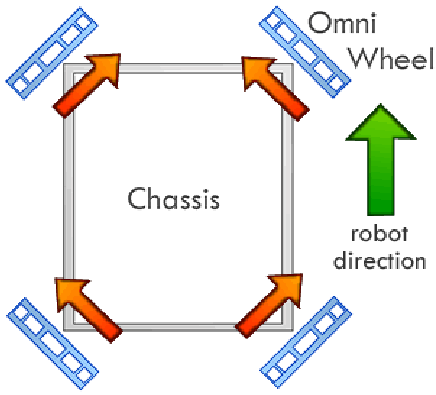


Fig 6 motor rotation

DIRECTION	MOTOR 1	MOTOR 2	MOTOR 3	MOTOR 4
Forward	CW	CW	CCW	CCW
Reverse	CCW	CCW	CW	CW
Right	CW	CCW	CW	CCW
Left	CCW	CW	CCW	CW
Turn Left	CCW	CCW	CCW	CCW
Turn Right	CW	CW	CW	CW

Table 1: Different Robot Directions using motor rotation
(CW – clockwise direction, CCW – counterclockwise direction)

5. Detection of user's head gestures

There are 2 phases to detect user's head gesture in our project. One is face preparing phase and the second one is command input phase. In the first phase, face detection works to find user's face. The first phase is for preparing for the second phase to detect user's gesture correctly. In the second phase, there are two functions, pose estimation and detecting smile face. Pose estimation is for commands to make the mobile robot move to desired directions. Detecting smile face is for command to let the mobile robot when the robot should grasp and release objects.

Figure.7 shows whole sequence of detection of user's head gestures. As the figure shows, after detecting the user's head gestures, the system sends the results of detection to the mobile robot through Bluetooth connection. And the system goes back to the preparing phase to detect next user's gesture correctly.

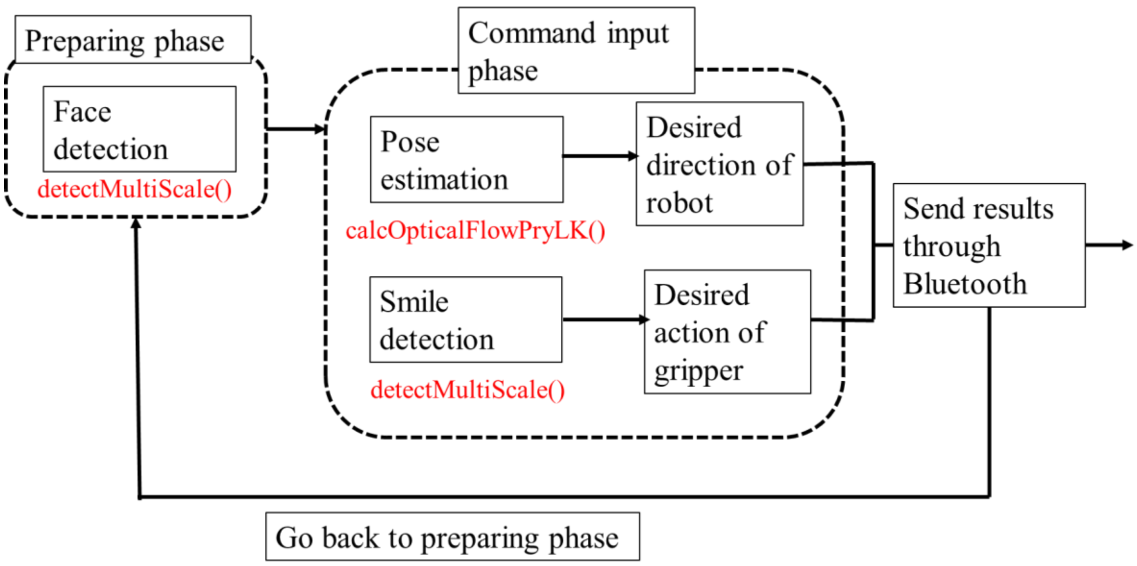


Fig 7 System sequence of gesture detection

5.1. OpenCV methods

In our project, OpenCV methods are used to detect the user's head gestures.

5.1.1. Face detection

The face detection is done by OpenCV's method detectMultiScale().This method implements the Viola-Jones algorithm that basically applies an ensemble cascade of weak classifiers over each frame to detect a face. If a face is successfully detected, three points that represent the location of the eyes and the nose are calculated over a 2D plan. An example of the result is shown in Figure.7.

5.1.2. Pose estimation

The gesture recognition is achieved through the estimation of the head pose. The tracking of the face points is performed by calculating the optical flow, thanks to the Lukas-Kanade algorithm implemented on the OpenCV's method calcOpticalFlowPyrLK(). This algorithm processes only two frames and basically tries to define where, in a certain neighborhood, a specific-intensity pixel will appear on a frame (treated here as the "current frame") based on the immediately previous frame. Figure 8 shows whole concept of pose estimation.

First of all, three subjects of features in previous frame are defined as following equations.

$$\mathbf{p}_{LE} = \begin{bmatrix} x_{LE} & y_{LE} \end{bmatrix}$$
$$\mathbf{p}_{RE} = \begin{bmatrix} x_{RE} & y_{RE} \end{bmatrix}$$
$$\mathbf{p}_{NE} = \begin{bmatrix} x_{N} & y_{N} \end{bmatrix}$$

And same as the above equations, three subjects of features in current frame are defined as following equations

$$\mathbf{P}_{LE} = \begin{bmatrix} X_{LE} & Y_{LE} \end{bmatrix}$$
$$\mathbf{P}_{RE} = \begin{bmatrix} X_{RE} & Y_{RE} \end{bmatrix}$$
$$\mathbf{P}_{NE} = \begin{bmatrix} X_{N} & Y_{N} \end{bmatrix}$$

By using these definitions, roll, pitch, and yaw are defined as following

$$\text{Roll} = \arctan\left(\frac{Y_{LE} - Y_{RE}}{X_{LE} - X_{RE}}\right)$$
$$\text{Pitch} = Y_N \text{-} \ y_N$$
$$\text{Yaw} = x_N \text{-} \ X_N$$

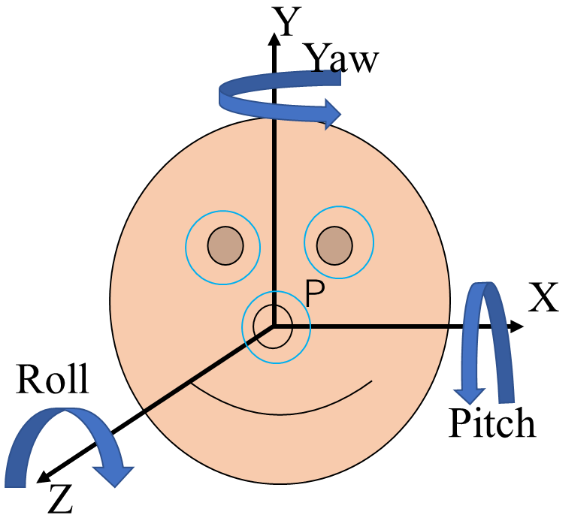


Figure 8: Concept of pose estimation

User's gestures are detected through the results of yaw, roll, and pitch. By combining these results, the system estimates user's head pose like fig9. Each pose is corresponding to the command to make the robot move forward desired direction. Figure 9 shows the relationships between user's head pose and commands.

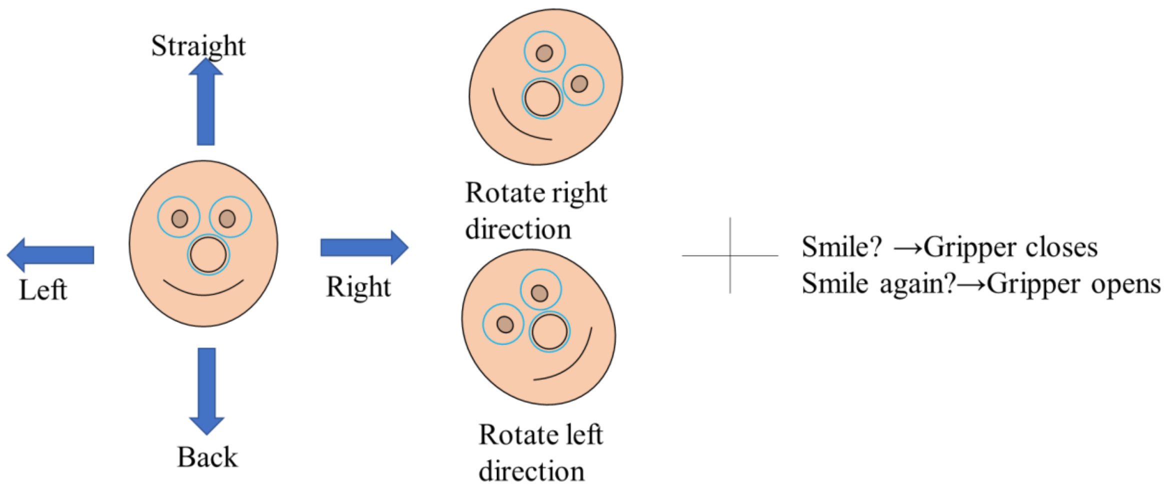


Figure 9: Relationships between gestures and commands for the robot

5.1.3. Smile detection

The smile detection is the similar method to one described in 5.1.1. The difference point is the kind of library code. In the smile detection, haarcascade_smile.xml is used to detect user's smile face. In our system, when smile is detected continuously 10 times, the system outputs the command for "grasping".

6. Interfaces

6.1. IF between Raspberry pi 3 and Arduino

In our system, Raspberry pi3 and Arduino communicate through Bluetooth connection. For the Bluetooth connection, the robot equips Bluetooth module HC-05(figure 10). The Bluetooth module communicates with Bluetooth function that is already equipped to Raspberry pi3.

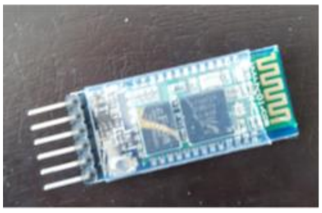


Figure 10: Bluetooth module Hc-05

6.2. IF between IP camera and users

The users can observe the situation in front of the robot though IP camera equipped on the robot. The IP camera is connected Wifi connection(2.4Ghz). Then, through the wifi connection, the IP camera sends the video captured image to user's smartphone. Thanks to this function, the users can watch what kind of objects or obstacles exist in front of the robot.

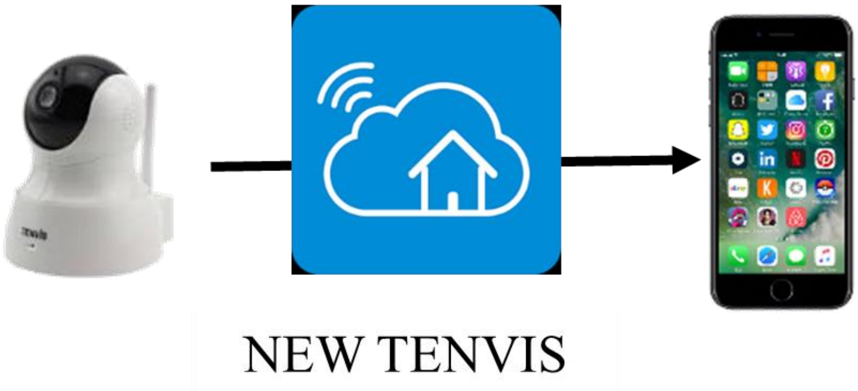


Figure 11: Connection between IP camera and smartphone

7. Demonstration

As proposed a hand free control of the mobile robot is achieved and it can carry and deliver objects like water bottle, small boxes to the user defined location by the commands created from user's gestures. The fig 12 (a) and (b) gives a pictorial depiction of robot carrying an object and delivering it in the goal position and Fig 12(c) depicts the users head movement which in turn moves the robot to turns right.

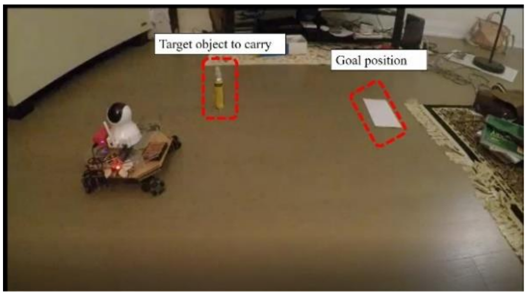


Fig 12 (a) Initial position of robot

Fig 12 (b) Robot in target position with object

Fig 12 user's head gestures

8. Conclusion

Our project proposed hands-free interface to control robot though OpecCV. This method does not require users to equip any tools like a head gear. In a demonstration, the robot could deliver the object to desired goal position only though user's head gestures.

As a future development of this project we would like to implement machine learning methods for faster and more precise user's gestures detection. Moreover, we would like to implement path planning to improve the maneuverability of the robot.

9. Reference

1. http://www.who.int/mediacentre/factsheets/fs384/en/
2. http://www.sci-info-pages.com/facts.html
3. https://www.christopherreeve.org/living-with-paralysis/costs-and-insurance/costs-of-living-with-spinal-cord-injury
4. https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwjr1pqrsIrYAhVDkeAKHU_nAw8QjxwIAw&url=https%3A%2F%2Fwww.chiefdelphi.com%2Fmedia%2Fphoto%2F22353&psig=AOvVaw2a3gGPsWqpj8YBPNuEbuMD&ust=1513370639012162
5. https://www.youtube.com/watch?v=1-Ju_VqYLAU