VIRA-Virtual Reality Telepresence Robot

Author
Sarosh Siddique (210755)
Zunaira Khan (222162)
Usama Chaudhary (236942)
Saqib Ullah (143512)

Supervisor
Dr. Mir Yasir Umair

Submitted to the faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology,
in partial fulfillment for the requirements of B.E Degree in Electrical Engineering
(JUNE), 2021

VIRA-Virtual Reality Telepresence Robot

By
Sarosh Siddique
Zunaira Khan
Usama Chaudhary
Saqib Ullah

Supervisor
Dr. Mir Yasir Umair

Submitted to the faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology,
in partial fulfillment for the requirements of B.E Degree in Electrical Engineering
(JUNE), 2021

CERTIFICATE OF CORRECTIONS & APPROVAL

Certified that work contained in this thesis titled "VIRA-Virtual Reality Telepresence Robot", carried out by NC Sarosh Siddique, NC Zunaira Khan, NC Usama Chaudhary and NC Saqib Ullah under the supervision of Dr. Mir Yasir Umair for partial fulfillment of Degree of Bachelor of sElectrical Engineering, in Military College of Signals, National University of Sciences and Technology, Islamabad during the academic year 2020-2021 is correct and approved. The material that has been used from other sources it has been properly acknowledged / referred.

Approved by Supervisor
Asst. Prof Dr. Mir Yasir Umair
Department of EE, MCS

Date:__________________

DECLARATION

No portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

Sarosh Siddique
210755

Zunaira Khan
222162

Usama Chaudhary
236942

Saqib Ullah
143512

Signature of Supervisor

Acknowledgements

We would like to pay gratitude towards Almighty Allah to guide us throughout this work at every step, opening new doorways of success for us, enlightening us with the knowledge and showering upon us His blessings which enabled us to achieve this success.

We are also extremely thankful to our supervisor Asst. Prof. Dr. Mir Yasir Umair for his help during the project, his constant guidance and encouragement. The project would not have been possible without his expert advice, unfailing patience, and invaluable efforts.
We would also like to mention Mr. Shoaib Ahmed for his incredible provision and assistance. We appreciate his guidance throughout the whole project.

Finally, we would also thank the department of EE and our project coordinators who walked us through all the administration related affairs.

Dedicated to our exceptional parents and respectful teachers whose tremendous support and cooperation led us to this wonderful accomplishment.

Abstract

VIRA is a virtual reality telepresence robot which utilizes state-of-the- art virtual reality combined with live **360°** video. It is specifically designed for healthcare providers and allow them to have fully immersive 'virtual presence' from any remote location. Navigation system for VIRA is developed and can be implemented for real-time navigation of mobile robot.

In our work we developed system that supports live video streaming from any remote location as well as video & audio transmission, the video is configured in dual-screen and viewed in Virtual Reality environment. The user's head motions are recorded, and the robot camera moves in response, creating an immersive experience. A gesture-based navigation system using Gyro-scope glove controls the robot mobility. Furthermore, in VR, a pop-up module shows important clinical information about the discovered patient in a pop-up form in front of the doctor.

Contents

List of Figures

List of Tables

Abbreviations

VR Virtual Reality

USB Universal Serial Bus

RPI Raspberry Pi

UDP User Datagram Protocol

IMU Inertial Measurement Unit

IDE Integrated Development Environment

HTTP Hypertext Transfer Protocol

CSI Camera Serial Interface

QR Quick Response Code

URL Uniform Resource Locator

HTML Hypertext Markup Language

Chapter 1

1 Introduction

This chapter includes the basic introduction of project, the purpose, scope and the background.

1.1 Background

In today's era when everyone prefers to do things in an easy way and from their own private space, the concept of telepresence plays an important role in many places where human beings can do their tasks in an efficient way and their presence is not achievable. Such tasks can be performed by using virtual reality telepresence robot that can be controlled by a person remotely. A Telepresence robot gives the user experience of remote location as being present in that location. This type of technology is exceedingly useful for medical purpose as it enhances patient healthcare. In our work we have proposed a telepresence robot combined with VR which is controlled by hand gesture movement of user. [1]

As there is lack of qualified medical staff in remote areas of our country and there are some old age people who have mobility issue, VIRA allows them to see the doctor from home. Doctors cannot physically access these locations regularly and require a way to remotely consult staff and examine patients. As a result, doctors can discuss treatment options and connect with patients remotely. During an emergency, this helps to increase the efficiency of medical diagnostic and treatment plans. This project has specifically made for health care providers where doctors and patient can communicate from separate location.

1.2 Problem Description

In some countries, there is no such facility that could enable doctors to provide treatment to the people who are located in remote places especially in rural areas where patients may have to travel for hours for a relatively simple appointment. Many patients may fail to show up for appointments or postpone them. Missed or delayed appointments can cause conditions to worsen, as the health of the patients is compromised. Even after surgery, many of the pre- and post-operation appointments can be handled with care-at-a-distance through VIRA, lowering the expense of those visits for the patient. Often times there are medical emergencies in remote places and there is no one to provide immediate medical guidance due to a lack of qualified medical staff in these areas. The people in these areas suffer due to the lack of medical attention. Doctors cannot access these locations regularly and require a way to remotely consult staff and examine patient in these areas. VIRA provides a way to cater to all these problems efficiently. It provides a solution that falls into remote patient monitoring allowing healthcare providers to track a patient's condition from a distance. This makes it easy for a doctor to monitor and provide consultation to patients who are at health-risk or are recovering from recent surgery. Moreover, VIRA reduces the number of times that frontline healthcare workers come into contact with patients to reduce the risk of infectious disease especially during COVID-19.

1.3 Purpose

The purpose of our project aims to build a telepresence robot that combine with state of art VR technology to design an integrated robotics system especially for health care providers. This includes building a robot with gesture-controlled navigation system that display patient vital signs to doctor in form of pop up. By using this technology doctor can be able to feel the remote location as being present there in a way that customary video failed to do so.

1.4 Scope

The project finds its scope in every remote location. It will make the working of frontline healthcare workers easier as now they can treat patients from their own personal space and guide the healthcare staff present in remote location. It is an efficient solution for patient monitoring as doctor and patient communicate easily through VIRA.

1.5 Deliverables

VIRA has five main deliverables: Mobile Robot, Integrated audio-video communication, Gesture controlled navigation system, Patient data pop-up module and lastly Head-tracking for 360-degree view. The combination of these deliverables makes VIRA an effective and immersive medium for telepresence.

1.5.1 Mobile robot

This is the main body of robot that is capable of moving anywhere in any remote location and allow doctors to see that location as being there. It is being controlled by hand gesture that user has made.

1.5.2 Integrated audio-video communication

This will be combined with our moving robot that will allow doctors and patients to communicate with each other easily.

1.5.3 Gesture controlled navigation system

This will consist of gyroscope-based glove, and it will enable the user to control the motion of robot with hand gesture.

1.5.4 Patient data pop up module

The purpose of this is to provide the doctor with the basic information of patient so when robot moves in a specific room and detects a patient, doctor will see the pop-up text in VR that will contain all of basic data of that specific patient including his name blood pressure, sugar level etc.

1.5.5 Head-Tracking for 360° view

The sensor data from user will go in raspberry pi and raspberry pi will use that data to give propriate command to servo motor to move pi camera on patient's side. This data basically represents the movement of user which gives the user immersive experience of VR.

1.6 Limitations

As it is a prototype, next big step would be to test it on different networks. Real time testing will further validate its importance and significance.

Chapter 2

2 Background and related work

This chapter presents the basic theories, literature review that we studied for our project and provides an overview of hardware and software used.

2.1 Virtual reality

Virtual Reality is a computer-generated environment that gives the user a simulated experience that is similar to the real world. Virtual Reality combines immersive display with 3D graphics, creating an environment where the displayed content matches the user's viewpoint. VIRA uses this technology as an inclusive relationship between the virtual content and the user(doctor). Currently, VIRA uses standard virtual reality system i.e., Virtual Reality Headset to simulate a user's physical presence in a virtual environment. The user(doctor) will be able to look around the artificial world, move around it and interact with the patients and the medical staff.

2.2 Telepresence

Telepresence technology allows the user to feel as if they were present, to have an effect at a place other than their true location. Information travels in both directions between the user and the remote location. Telepresence deploys improved fidelity of both sight and sound. VIRA focus on the type of telepresence that uses remotely controlled robots. In this type of communication VIRA's surrounding is presented to a user that controls the robot remotely.

2.3 Hardware Overview

The project consists of following hardware components which are described below.

2.3.1 Raspberry pi

Raspberry pi is a micro-processor board, and it requires operating system to run. it consists of high-speed processor, 4GB memory and connectivity. It uses ethernet adapter for internet connectivity. Raspberry pi is mounted on upper shaped body of robot along with pi camera, and live video will be streaming towards the cell phone of user. Servo motors, microphone and speakers are also attached with raspberry pi for audio/video communication between robot and user.

Figure 1: Raspberry pi [2]

2.3.2 Pi camera

The pi camera v2 module has a Sony IMX219 8-megapixel sensor and it supports 1080p30, 720p60 and VGA90 video modes. This module is directly attached with raspberry pi. It is used to capture images, recording videos and for live videos. It is mounted on upper shaped body of robot so that it can provide live video streaming to user and on detecting the number plate of patient bed, it captures images so that patient information will be display in form of pop-up to the doctor.

s

Figure 2: Pi camera [3]

2.3.3 VR headset

A mobile phone is inserted into VR headset, and it shows virtual reality to user. User can be able to see virtual environment after wearing it. VR headset is used for VIRA so that user may be able to see virtual environment of remote location

Figure 3: VR headset [4]

2.3.4 Arduino nano atmega328

Arduino nano atmega328 is a micro-controller used for some applications. It has 14 digitals pins, 8 analog pins, 2 reset pins and 6 power pins. Robotics, embedded systems, automation, and electronics projects are used for Arduino boards. The Arduino Nano has a crystal oscillator with a frequency of 16 MHz. It's used to produce a clock of precise frequency by using a continuous voltage source.

Figure 4: Arduino nano atmega328 [5]

Arduino nano in VIRA is used for navigation system and it will be on glove side. The communication between Arduino nano and UNO will take place through nRF24l01.

2.3.5 Arduino UNO

Arduino uno is a micro controller board based on the ATmega328P. The board has digital and analogue input/output pins that can be used to connect to numerous expansion boards and other devices. The board features 14 digital I/O pins (six of which may produce PWM), as well as 6 analogue I/O pins, and can be programmed using the Arduino IDE. In VIRA Arduino UNO will be on robot side, it is connected with L298 motor driver IC and used to receive data from glove to give propriate command to motor driver IC to move the robot.

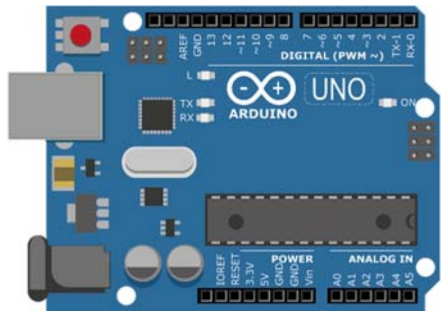

Figure 5: Arduino UNO atmega328P [6]

2.3.6 MPU6050

A gyro sensor consists of 3 axis accelerometer and 3 axis gyroscope data to measure motion related variable of objects. MPU6050 is used in VIRA for gesture recognition of glove, and it is attached with Arduino. Arduino of glove used that sensor data and send that data to Arduino of robot.

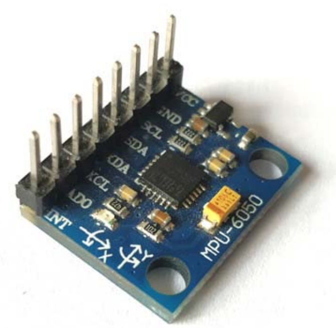

Figure 6: MPU6050 [7]

2.3.7 nRF24Lo1 + PA/LNA

nRF24L01+PA/LNA is a wireless transceiver where PA stands for power signal it raises the signal being transmitted from nRF24L01 + chip and LNA stands for Low-noise amplifier, it amplifies a very weak and unreliable signal from an antenna to a useful level. It is suitable and easy module for wireless communication range up to 1km. It operates at a voltage of 3.3 V. Two modules are used in VIRA one as a transmitter and other as a receiver.

It is used for communication between two Arduino’s.

Figure 7: nRF24L01 + PA/LNA [8]

2.3.8 Motor Driver IC

The DC motors are driven by the LM 298N Motor Driver module. This L298N Motor Driver Module is a high-performance motor driver for DC and Stepper Motors. An L298 motor driver IC and a 78M05 5V regulator make up this module. Up to four DC motors can be controlled by the L298N Module, or two DC motors with directional and speed control.

Figure 8: L298N motor driver module [9]

2.3.9 Servo Motors

Servo motors are high-torque motors that are extensively employed in robotics and other applications because their rotation is easy to control. A geared output shaft on a servo motor can be electrically controlled to rotate one (1) degree at a time. In VIRA these motors are used in the camera mount. This kind of mount is often referred to as “pan and tilt mount”. It has two motors to control the pan (rotation on horizontal plane) and tilt (rotation around y-axis).

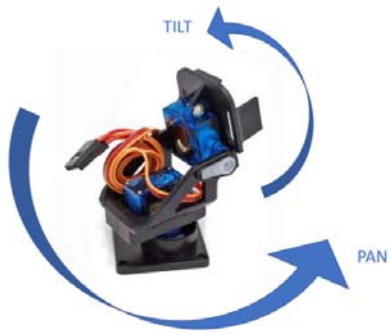

Figure 9: Sg90 servo motor [10]

2.3.10 DC motors

The motor used in mobile platform of VIRA are geared DC motors. The geared DC motors are basically DC motors with a gearhead or gearbox attached to it. RPM refers to rotations per minute, which is how the motor's speed is measured. The motor's gear assembly has been added to generate more torque. The gear assembly often known as the speed reducer has two gears, the input gear and the output gear. The output gear has more teeth than the input. The output gear rotates more slowly thus decreasing the speed of the input gear and increasing the torque.

Figure 10: DC motor [11]

2.3.11 Microphone + speakers

Microphone and speakers are attached with raspberry pi for audio/video communication between robot and user.

Figure 11: Microphone + speakers [12]

2.3.12 Dry lead acid batteries

VIRA is equipped with a 12V battery to power the bot. The battery supplies the power to the bot for functioning. The battery used is a kind of lead Acid battery. It is a rechargeable and high-power battery. It has a long life and lasts for about 2000 charge and discharge cycle; therefore, it is a good choice for the robot. These batteries charge up in around two hours using a 438-watt charger which is about four times faster than the other available alternatives. Thus, the robot will be charged and back at job in no time.

Figure 12: Dry lead acid battery [13]

2.4 Software Overview

The project requires the following software related tasks:

2.4.1 PyCharm

Python is a high-level language that is easy to learn and use and has a fairly simple syntax. It is used to develop GUI, web apps and websites. It makes programming easier with the wide variety of built-in libraries that it provides. Pycharm is an integrated environment used for python. For this project, we used Pycharm version 3.3. In VIRA, python is used to code the pop-up data feature.

Figure 13: PyCharm logo [14]

2.4.2 OpenCV

The OpenCV-Python library is a set of Python bindings for solving computer vision issues. It focuses primarily on image processing, video recording, and analysis, including capabilities such as face and object detection. In VIRA it is used to focus and detect the QR code of the patient bed which in result will show up a pop up containing the essentials of the patient.

Figure 14: OpenCV logo [15]

2.4.3 Arduino IDE

ARDUINO IDE (Integrated Development Environment) is software that is used to code Arduino board. It is easy-to-use software that runs on different operation systems. It an open-source software and the coding environment is Java based. It can be used with all Arduino boards and is used to program boards for multiple tasks. In VIRA, Arduino board is used in glove for navigation and Arduino IDE is used to program Arduino UNO and NANO for movement of robot. The Arduino board is coded to enable and assist the robot movement.

Figure 15: Arduino IDE logo [16]

2.4.4 Shairport

Shairport Sync is an AirPlay audio player that can stream audio from iOS, Apple TV, and macOS devices, Shairport will be install on raspberry pi and it capture audio from AirMusic trial app and play it on speakers.

2.4.5 SoundWire

SoundWire allows you to stream any music or audio from your Windows or Linux computer to your Android smartphone or tablet. SoundWire has a minimal latency, so it can be used to listen to the soundtrack of any video while watching it. SoundWire can be used as a listening device with a computer that has a built-in microphone, such as a netbook. Before using this app on android phone, first install the SoundWire server on windows and run it. It is easy to use and provide variety of features like Live audio capture and streaming, Excellent sound quality and it runs on all android versions. In VIRA it is used to send audio from raspberry pi to cell phone of a user.

Figure 16: SoundWire logo [17]

2.4.6 AirMusic

This is widely used to run stream all audio from any app to every receiver. AirMusic is compatible with all of the current streaming technologies, including Apple AirPlay, DLNA, SONOS, Google Cast, AllPlay, Amazon Fire TV, Denon HEOS, and Roku. It runs in the background and recognize immediately if receiver is available. In VIRA it is used to capture from cellphone of a user and send that to raspberry pi.

Figure 17: AirMusic logo [18]

2.5 Related works and technology

This section will go through existing solution and research papers related to this technology.

2.5.1 Existing solution and their drawbacks

There are some companies who are already into this market, named SPOT, the AVA 500 and Ohmni SuperCam.

2.5.1.1 SPOT

Spot is an agile mobile robot that navigates ground with unprecedented mobility, allowing you to automate routine inspection tasks and data capture safely, accurately, and frequently. [19] It is also customized for healthcare and provides audio-video communication but on a downside of it, they are not providing any virtual reality and user-controlled navigation which make VIRA unique.

2.5.1.2 The AVA 500

The Ava 500 gives a seriously instructing presence when you can't go to a far off gathering, meeting, or address, or to meet with laborers at an assembling plant, distribution center, or preparing office. Telepresence empowers you to save a huge number of dollars and many hours of the year that would some way or another be spent on flights, inns, and related travel costs. [20] But comparing to VIRA it is not design for healthcare not it has a feature of user-controlled navigation.

2.5.1.3 Ohmni SuperCam

The Ohmni Supercam is a telepresence robot, giving users the power to ascertain content on whiteboards and printed documents far more clearly than other systems. [21] It provides audio video communication same as VIRA but the downside of it, it does not have virtual reality concept and nor its movement can be controlled by user. Moreover, it is not customized for healthcare.

	The AVA 500	**Ohmni SuperCam**	**SPOT**	**VIRA**
Audio/video communication	✓	✓	✓	✓
Virtual Reality	✗	✗	✗	✓
User controlled navigation	✗	✗	✗	✓
Head tracking	✗	✗	✗	✓
Customized for healthcare	✗	✗	✓	✓

Table 1: Comparison of existing technologies with VIRA

2.5.2 Literature Review

An overview of the literature studied to develop the final methodology for our project is given below:

Firstly, we viewed the master's thesis by Gabriella Rydenfors, called "telepresence and remote communication through virtual reality" this thesis presents the implementation of a telepresence system which is combined with state-of-the-art virtual reality technology. The thesis explores different navigation systems for the telepresence system and evaluates them through a user study. The paper also evaluates the performance of telepresence as a communication media and compares it to the performance of traditional video communication. The results show that telepresence is a better communication media than video communication. [22]

The second research paper is 'A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors' this paper basically presents a technique for continuous Hang gesture recognition using a 3-axis accelerometer and gyroscope sensors by comparing gesture codes with gesture database. It also introduces a solution to reduce the unstableness of the hand making the gesture. [23]

Lastly, we reviewed the paper titled “live video streaming using raspberry pi in IOT devices” this paper presents the concept of using Raspberry pi camera to send and receive video data via the internet as a live stream. [24]

Chapter 3

3 Design and development

This chapter present the project design and features that includes live video transmission as well as audio communication, head tracking, and pop-up data.

3.1 Chassis design and build

VIRA's chassis has been designed and built while keeping the requirements of the application in mind. The chassis has 2 main parts that are discussed in detail in this section.

3.1.1 Iron Chassis

The chassis of the robot is a very important part of the entire robot. It ties all the parts together and is designed meticulously to perform the required tasks. The chassis for VIRA is made out of Iron. The reason to use Iron frame is that it is strong and durable to keep the robot stable and well-balanced while moving. The chassis has two main parts. The Upper part is the Rod-Shaped body with a camera mount. The height of the rod is adjusted comparable to actual human height, enable the doctor to see the patient easily giving a more realistic telepresence experience to the doctor. The camera mount consists of a Pi camera and is directly attached with raspberry pi. It provides live video streaming to user (doctor) and detects the number/name plate of patient bed; it captures images so that patient information will be display in form of pop-up to the doctor.

3.1.2 Mount

Mount is the upper part of the chassis body. It is a rod-shaped mount with a camera mount on the top. It used to keep the RPi camera comparable to the height of the patient bed to give doctor a complete view of patient, so that the doctor could easily communicate with the patient. The camera mount consists of the RPi camera mounted on it, together with the SG-90 servo motors to rotate the camera according to the doctor's head rotation. This kind of mount is often referred to as "pan and tilt mount". It has two SG-90 motors to control the pan (rotation on horizontal plane) and tilt (rotation around y-axis). The rod-shaped mount also consists of a rubber plank to hold Raspberry pi and the power bank. Raspberry pi is powered by a power bank of 10000 mAh, 5V and 3.0A output. This mount is used for head tracking feature. The accelerometer and orientation sensor is sent to raspberry pi by using the MPU+GPS stream application. This data is then mapped to servo motor angles. By using this concept, we developed a pan-Tilt moving camera system which mirrors the movement of the user's head.

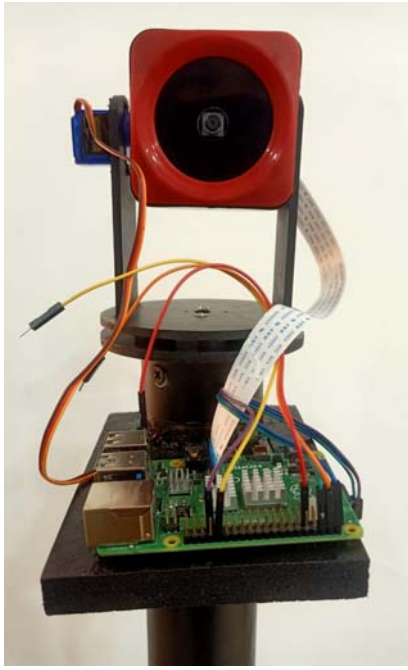

Figure 18: Camera Mount used in the design

3.1.3 Motors

The motor used in mobile platform of VIRA are geared DC motors. The geared DC motors are basically DC motors with a gearhead or gearbox attached to it. The speed of the motor is measured in terms of rotations of shaft per minute which is termed as RPM. The gear assembly in the motor is added to provide a higher torque.

The gear assembly often known as the speed reducer has two gears: the input gear and the output gear. The output gear has more teeth than the input. The output gear rotates more slowly thus decreasing the speed of the input gear and increasing the torque.

Figure 19: Mechanism of gearing down [25]

Geared DC motors are used to lower the speed, hence enhancing the torque. Using the correct gear combination, the speed can be reduced to any desired figure so that the robot will be able to move smoothly.

The motor being used in VIRA is a 12 V motor with a metal geared box, 100 to 1 reduction ratio and an integrated quadrature encoder. The motor provides a resolution of 120 shaft rotations per minute. It has an 8 mm D shaped output shaft. Motor should be used at 12V, but it starts rotating at voltage as low as 1 V. The motor is able to change the direction of shaft rotation depending on positive and negative wiring changes. The motor will reduce the speed to a desirable value so that the robot is able to swift smoothly. It will generate a torque around 35 kgf-cm.

3.1.4 Wheels

For any mobile robot, wheels are necessary part of the entire design. As an important feature of VIRA is its ability to move and navigate its environment on its own, the wheels suitable for its design and functionality are an important part of its structure. Wheels are preferred over legs in a robot as they provide a smoother, unobstructed tread and are much easier to control and program. Wheels of various sizes and materials are already available in the market.

Figure 20: Robot base with wheels and motors

A total of four wheels are employed in a robot. There are four drive wheels which control the forward and backward motion and the turns in left and right direction. For VIRA, we used a 15 cm diameter, nylon and rubber wheel. As the motor being used is a geared DC Motor with a shaft of 8mm, it is able to support a 15 cm diameter wheel. Nylon is being used to make wheels for a long time now and is very popular for industrial applications. Nylon and rubber wheels are strong and have high load capacity. Nylon and rubber wheels are well-suited for this as they are cost efficient and have strong tear and wear resistance thus, they have a long-life span. These wheels add to the strength and reliability of VIRA.

Figure 21: Robot body

3.2 Live video transmission in VR

The Live video transmission in VR is done using the Raspberry pi and the Pi camera module v2.1.

3.2.1 Enabling the Raspberry Pi Camera Module

The camera is connected to the Raspberry pi's CSI port using a 30 cm ribbon cable [26]. The 30 cm cable is used to allow enough space for the camera to rotate freely when tracking head movements of the user, as will be explained in the coming chapters. In order to capture pictures and video using the camera module, we first enabled the camera software in the Raspberry pi by using the desktop environment. The camera interfaces were enabled from the Raspberry pi configuration preferences. [27]

3.2.2 Web Streaming

For Streaming live video from Raspberry pi camera to local network, we built the code using the official PiCamera package documentation. The video is streamed in MJPEG format. The *StreamingOutput* class is used to construct a custom output called "*output*" for the video stream and the video is captured to a file-like *io.BytesIO* object.

```
class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()
```

Figure 22 Video output object

To interface with the Raspberry Pi camera module, the class *picamera.PiCamera* is used. When this class is constructed, it initializes the camera and is also used to adjust the attributes of the piCamera such as resolution, framerate, brightness e.t.c. According to our requirement, we have set the resolution to be 640x480 and the framerate as 24. The output object is then created using the *StreamingOutput* class as described above and the *start_recording()* method is used to start recording the video. The script uses Python's built-in module "*http.server*" to establish a basic video streaming server. For the server, the address host and port are set to the Raspberry pi's IP address and port 8000 respectively. To initialize the camera and video stream and create and run the server the following code is written:

```python
with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    camera.start_recording(output, format='mjpeg')


    try:
        address = ('', 8000)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()

    finally:
        camera.stop_recording()
```

Figure 23 http streaming server and camera initialization

The live video stream can be accessed by visiting http://192.168.8.130:8000/ using any web browser, where 192.168.8.130 is the Raspberry pi's IP address. The stream can be accessed using any device on the same network as the Raspberry pi.

The *BaseHTTPRequestHandler* class is used to handle any http request that arrives at the server. The class itself does not respond to any actual requests, instead the subclass *do_GET()* is added to handle get requests. The requests and headers are parsed by the handler and the handler calls the specific method corresponding to the request. So when the get request method is received, the *do_GET()* method is called with no arguments.

The *send_header()* and *send_response()* methods of the *BaseHTTPRequestHandler* class are used to add the HTTP and response headers to their corresponding buffers i.e. the headers buffer and the internal buffer [28]. The class is defined using the following code:

```python
class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            self.send_header('Content-Type', 'multipart/x-mixed-replace; boundary=FRAME')
            self.end_headers()
```

Figure 24 http response class

3.2.3 Dual Screen Configuration

After streaming the live video to the local web host, it must be configured to be viewed in Virtual Reality. This is done by using Android's built-in split screen feature. The video stream is accessed using chrome and split-screen is activated to view two browsers side-by-side in a Dual-Screen or stereoscopic view where the right part of the screen is for the right eye and the left screen is for the left eye. Once the stereoscopic configuration of the video stream is achieved, the Android phone is placed inside the VR headset and the live video stream is ready to be viewed by the user in Virtual Reality. This stereoscopic VR is immersive and uses the concept of depth perception to give the user a feeling of "being there" [29]. The stereoscopic view is shown in Fig 24.

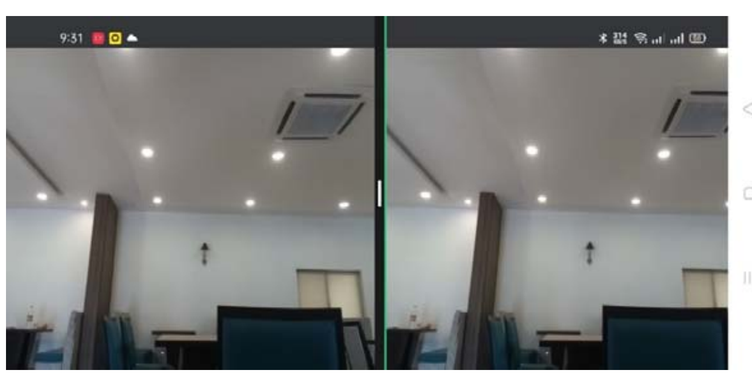

Figure 25 Stereoscopic view

3.3 Head tracking

Head tracking is done by using two SG-90 servo motors. In our case, we have one servo for *panning* left and right and second servo for *tilting* up and down.

- Tilt motion tracking
- Pan motion tracking

3.3.1 Tilt motion tracking

For tilt motion tracking we used accelerometer sensor data that is sent to raspberry pi by using the MPU+GPS stream application. This app reads the sensors that we configure and simply sends them to a host and port that we specify, using UDP protocol. we just have to set the host IP and port that we want to send the data to. In our case we want to send to the Raspberry pi (192.168.8.130) and to port 5555. Just set those values and press the switch stream ON button. The Raspberry receives sensor data from the Android phone and Raspberry pi is connected to Android phone via Wi-Fi through ethernet cable. Raspberry pi on receiving the data from MPU+GPS stream application will use that data to give propriate command to servo motor to move pi camera either in upward or downward direction. It will work in such a way that this data basically represents the movement of user head, so if user move the head in upward direction the servo motor also move the pi camera to upward with appropriate same speed, this gives the user immersive experience of virtual reality.

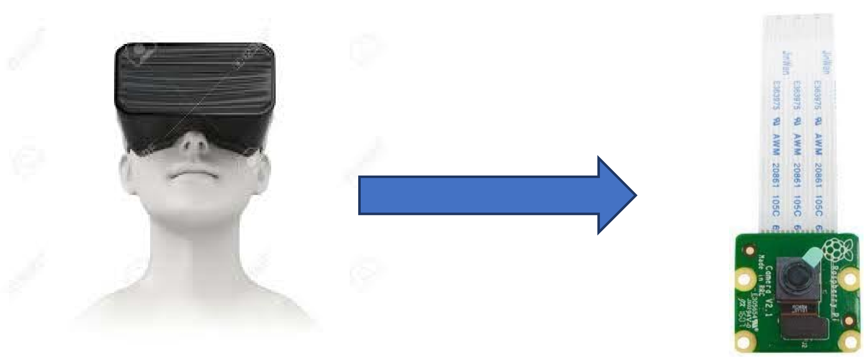

Figure 26: Tilt motion

3.3.2 Pan motion tracking

For pan motion tracking, orientation sensor is sent to raspberry pi by using the MPU+GPS stream application. This data is then mapped to servo motor angles. Sensor data is sent through UDP stream to raspberry pi. Raspberry pi on receiving the data from MPU+GPS stream application will use that data to give propriate command to servo motor to move pi camera either in left or right direction. This data basically represents the movement of user head, so if user move the head in left or right direction the servo motor also move the pi camera to left or right with appropriate same speed, this gives the user immersive experience of virtual reality. By using this concept, we developed a pan-tilt moving camera system which mirrors the movement of the user's head.

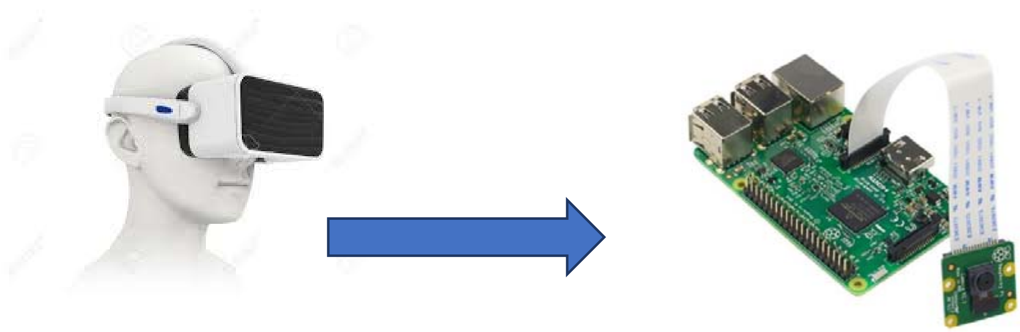

Figure 27: Pan motion

3.4 Audio communication

This subsection will cover the two-way audio communication solution used in the project. Two-way audio communication is completed using AirMusic, shairport and SoundWire software tools. The video is transmitted from the robot to the doctor and is configured in dual-screen mode and viewed in virtual reality.

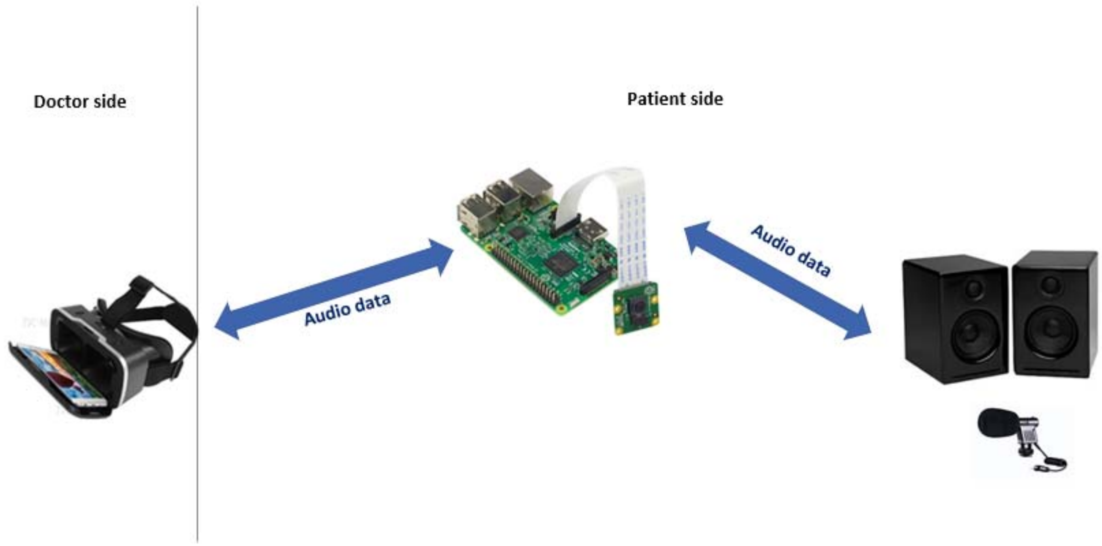

Figure 28: Two-way audio communication

3.4.1 Raspberry pi to android phone

We first download the SoundWire server on RPI, SoundWire Server downloads and installs the audio device's essential modules on a RPi, then broadcasts audio via a local area network. Secondly install its app on android phone. SoundWire allows you to stream any audio from windows to android phone. We activate the server on pi, when server is running on RPi, start the app on android phone. Press the connect button on app to connect to the server and start listening. When connection is already established, the SoundWire will automatically locate the server and server take data from pi through mic and play it through app. [30]

3.4.2 Android phone to raspberry pi

First, we install shairport on raspberry pi after installing it suceesfuly, enable and start the service. After that Shairport Sync AirMusic service appears on the network with a service name made from the Pi's hostname (in our case hostname is ViraSoundReceiver). AirMusic capture the audio from cellphone of user and send that to raspberry pi. Shairport will receive data from AirMusic app and play it on speakers. [31]

Our project uses the technology of Wi-Fi for communication between android device and RPi to control devices provided that both are connected to same network. The AirMusic app allow the user to send voice to raspberry so the user can send his voice through pi and patient can easily hear it.

3.5 Pop-up data

The pop-up data module can be split into 4 main parts as listed below:

- Image capture from live stream
- QR code detection
- Excel to image conversion
- Image infiltration in video stream

These 4 functions are implemented in the code to develop a system that detects a patient based on the pre-assigned QR codes and displays the data of the patient in Virtual Reality for a fixed period of time so that the doctor can look over it and then continues with the live video stream. The 4 parts of this module and their coding are discussed in detail in the coming sections.

3.5.1 Image Capture from Live Stream

OpenCV is used to capture frames from the live stream every alternate second using *cv2.VideoCapture()*; Where the URL of the stream is given as an argument. The frame is then converted to an image using *cv2.imwrite()*. This image is stored and updated on the Pi's disk every alternate second as "*test.jpg*" [32].

```
while True:

    cap = cv2.VideoCapture('http://192.168.8.130:8000/stream.mjpg')
    ret, frame = cap.read()
    cv2.imwrite("test.jpg",frame)
```

Figure 29 Image capture from Live Stream

3.5.2 QR Code Detection

OpenCV is used to detect the QR code. The image *test.jpg* is passed to the

QR_code_reader() to check if there is a QR code in the frame captured and if there is, it is decoded and the decoded string is stored in a local variable for comparison. This is done using the *detectandDecode()* method of OpenCV. The function *QR_code_reader()* returns true if a QR code is detected and false if not.

```
def QR_code_reader(imge):
    count = 0
    imgBlur = cv2.GaussianBlur(imge, (7, 7), 1)
    data, bbox, _ = detector.detectAndDecode(imgBlur)
```

Figure 30 QR code detection and decoding

3.5.3 Excel to Image Conversion

The next step is to import the excel sheet of the detected patient and convert it into an image so that it can be streamed to the local host. The string variable *data* contains the id of the patient as extracted from the QR code, for example: PATIENT1, PATIENT2, PATIENT3, and so on. In the designed prototype we have considered a sample database of 4 patients. Each of these patients has a designated excel sheet containing their basic information such as the name, D.O.B, medication, and disease history of the patient. A sample sheet is shown in Fig. 30. These excel sheets have the same names as the string encoded in their QR code. So, the variable *data* as described in section 3.5.2 is compared to a list of strings having the names of all patient files. When one of the filenames matches with the decoded *data*, that file is imported, read and converted into panda's data-frame using the *openpyxl* and *pandas* library. Now, the *DataFrame_to_image()* function is called for further processing of the data-frame into the final image. The code segment as describe is shown in Fig. 31.

A	B	C	D	E	F	G
NAME	PATIENT ID	D.O.B	GENDER	MEDICATIONS	HISTORY	
Zunaira	23-115-678	02/05/1999	Female	Prinivil Tabs	Diabetes	
				Humulin	Hypertension	

Figure 31 Sample patient data sheet

```python
Excel_sheet_name = ['PATIENT1.xlsx','PATIENT2.xlsx','PATIENT3.xlsx','PATIENT4.xlsx']
stList = ['PATIENT1','PATIENT2','PATIENT3','PATIENT4']

if data=="":
    return False
else:
    for str1 in stList:
        if str1==data:
            Excel_name=Excel_sheet_name[count]
            print(Excel_name)
            df = pd.read_excel(Excel_name, engine='openpyxl')
            DataFrame_to_image(df, css, outputfile="out.jpg", format="jpg")
        count = count+1
    return True
```

Figure 32 Excel to data-frame conversion

The next step is to convert this data-frame to an image. This is done using the function *DataFrame_to_image()* as defined in Fig. 32. The arguments given to this function specify the format and name of the output image file that will be generated. We have named the output file as out.jpg. The "jpg" format is chosen so that the image can easily infiltrate the existing video stream which is also in mjpg format. In this function, firstly a css file is defined which contains the styling rules for the output table. The data-frame is then converted to an HTML-ized table using the styling specified in the css file. Lastly, the *imgkit* library is used to convert this HTML table to an image. The imgkit options are used to define the attributes of the generated image. We have utilized these options to set the width and height of the image in accordance with our live video stream frame so as to not disturb the existing stereoscopic view.

```python
def DataFrame_to_image(data, css, outputfile="out.jpg", format="jpg"):

    fn = str(random.random() * 100000000).split(".")[0] + ".html"
    try:
        os.remove(fn)
    except:
        None
    text_file = open(fn, "a")
    text_file.write(css)
    text_file.write(data.to_html())
    text_file.close()
    imgkitoptions = {'width': 600,'height': 520, 'disable-smart-width': ''}

    imgkit.from_file(fn, outputfile, options=imgkitoptions)
    os.remove(fn)
```

Figure 33 Data-frame to image conversion

3.5.4 Image Infiltration in Video Stream

As explained in section 3.5.2, the function *QR_code_reader()* returns either true or false depending upon whether a QR code is detected or not. The output of this function is saved in a local variable *check* and is used to decide whether the image "*out.jpg*" should be streamed or not.

The *self.frame* attribute of the output object *stream* (defined in section 3.2.2) contains the frame that is to be streamed to the local host. This attribute changes every time a new video frame is captured from the camera and is then streamed. So, if the patient is detected i.e., *check==True*, the *out.jpg* file is imported, read and converted into a byte array. *Self.frame* is the altered to contain this byte array frame instead of the frame captured from the camera. At this instant, the user can see the image containing the patient's data instead of the live stream as show in Fig. 34. Alternatively, while the patient is not detected, the variable *check* continues to be false and *self.frame* is kept in its normal condition containing the frame captured from the camera and thus the live stream continues.

```
check=QR_code_reader(imge)
if check==False:
    self.frame = self.buffer.getvalue()
else:
    with open("out.jpg", "rb") as image:
        f = image.read()
        self.frame= bytearray(f)
```

Figure 34 Image infiltration in stream

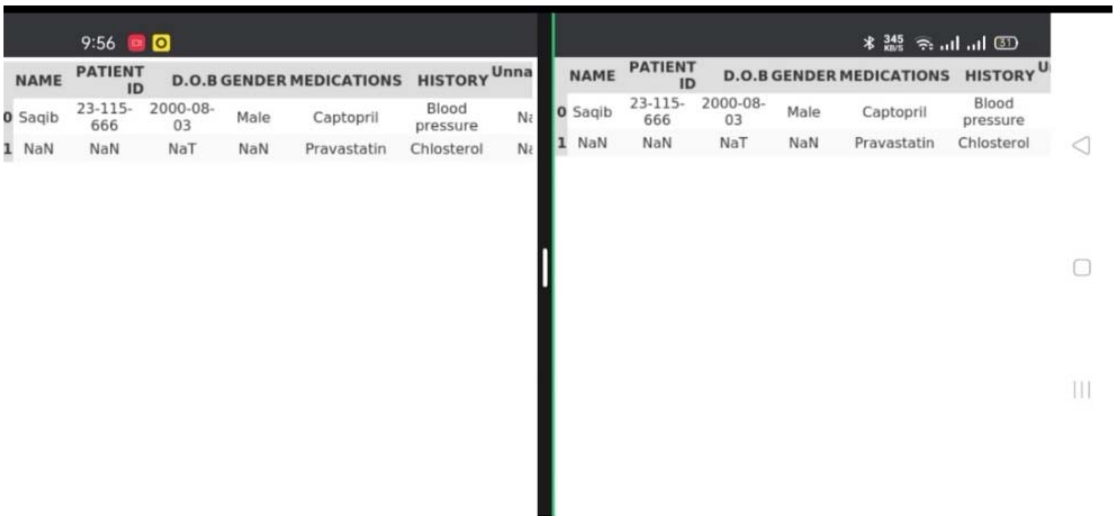

Figure 35 pop-up data in stereoscopic view

Lastly, the *check* variable is also used to set a timer to specify how long the image with patient data should be displayed before continuing the stream. *Time.sleep()* is used to pause the code for 7 seconds if the patient data is sent. This gives the user (doctor) enough time to look over the data, however, the timing can be adjusted according to preference. In addition, the data that is displayed in Virtual Reality cam easily be modified by simply editing the excel sheets containing the patient data. The process of the pop-up data from patient detection to data display is summarized in the block diagram of Fig 35.

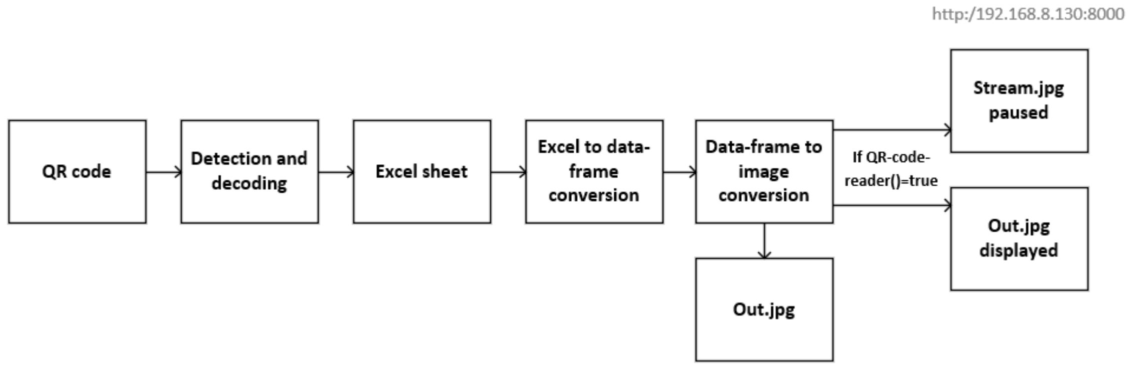

Figure 36 Pop-up data methodology

Chapter 4

4 Navigation system

This chapter presents the navigation system for our robot.

4.1 Communication between Arduinos

In this project we used two Arduino for communication, one is nano which is on the glove side and other is uno which is on robot side. The communication between two Arduino take place through nRF24L01 + PA/LNA module. The transmitter sends data at a regular interval to the receiver which receives the received data on the receiving side.

4.1.1 nRF24L01 Automatic Packet Handling

There are three scenarios in which NRF24L01 module transact with each other

Figure 37: transaction with acknowledgement and interrupt [33]

First, case is a positive scenario transaction with acknowledgement and interrupt, the transmitter initiates communication with the receiver by transmitting a data packet. It waits (about 130 µs) for the acknowledgement packet (ACK packet) to arrive after the entire packet has been broadcast. The receiver sends an ACK packet to the transmitter when it receives the packet. The transmitter asserts the interrupt (IRQ) signal when it receives the ACK packet, indicating that additional data is available. [33]

Figure 38: Transaction with data packet lost [33]

Transaction with data packet lost, in this case scenario is negative where a retransmission is required due to loss of the packet transmitted. The transmitter waits for the ACK packet to arrive after transmitting the packet. The packet is retransmitted if the transmitter does not receive it within the Auto-Retransmit-Delay duration. When the receiver receives the retransmitted packet, the ACK packet is sent, which causes an interrupt at the transmitter. [33]

Figure 39: Transaction with acknowledgement lost [33]

Transaction with acknowledgement lost, in this case again a retransmission is required due to loss of the ACK packet. Even if the receiver receives the packet on the first attempt, the transmitter believes the recipient has not received the packet at all due to the loss of the ACK packet. It retransmits the packet once the Auto-Retransmit-Delay (ARD) period has passed. When the receiver receives a packet with the same packet ID as the previous one, it discards it and sends another ACK packet.

The nRF24L01+ chip handles complete packet processing without the microcontroller's involvement. [33]

4.1.2 Connecting nRF24L01 module with Arduino

First, connect the VCC pin on the module to 3.3V on the Arduino and GND pin to ground. The pins CSN and CE can be connected to any digital pin on the Arduino. In our case, it’s connected to digital pin 9 and 10 respectively. The pins that are used for SPI communications are 11 (MOSI) ,12(MISO) and 13(SCK). As there is a lot of data transfer need in nRF24L01 module so we used SPI pins on Arduino to provide best results. The hardware SPI pins are substantially faster than using another set of pins to ‘bit-bang' the interface code. [33]

Arduino nano and nRF24L01 connection on transmitter side are as follows, and same connection need to be on receiver side but with Arduino uno.

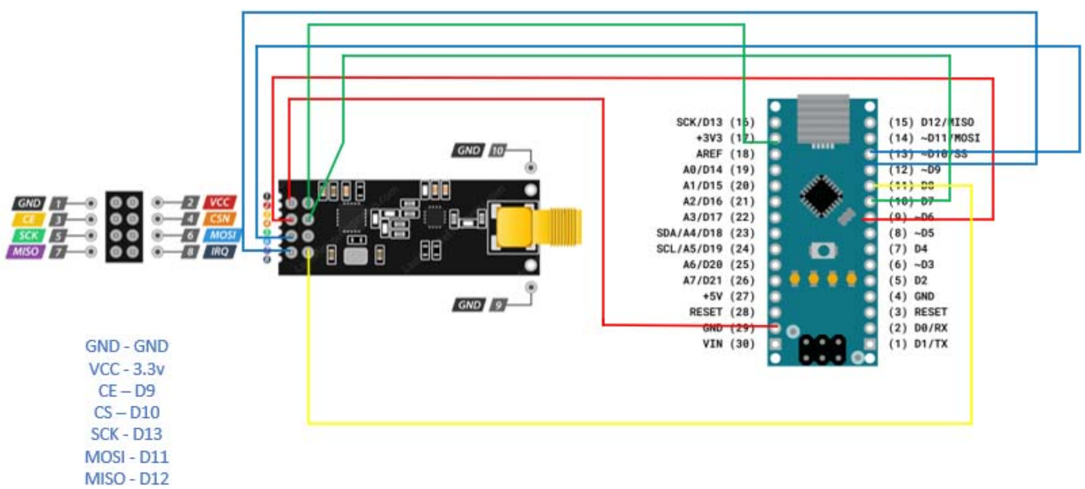

Figure 40: Transmitting side

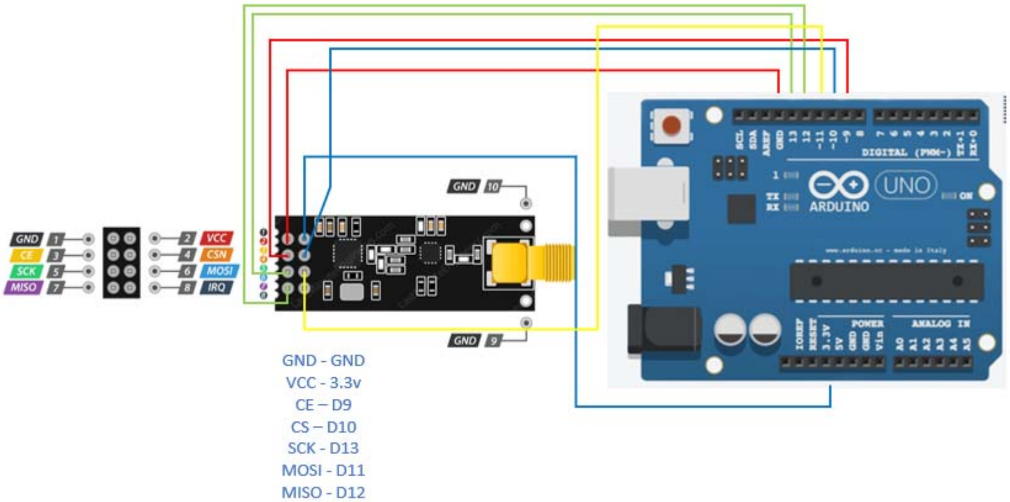

Figure 41: Receiving side

After setting up the connection, we will set the code for transmitter and receiver in which we set the address in which two modules communicate with each other. To check whether data is sent and receive correctly, we can also open serial monitor on both sides.

4.2 Gesture recognition

Hand gesture recognition is made to control the movement of robot. In our project we used mpu6050 for gesture recognition of glove. It consists of 3-axis accelerometer and 3-axis gyro sensor data accelerated on chip. This chip uses I2C protocol for communication. The rotational velocity, or rate of change of angular position over time, is measured by the gyroscope along the X, Y, and Z axes. On other side, MPU6050 accelerometer measures acceleration along the 3 axes. [34]

Tgk900000

The on-chip accelerometer on the MPU6050 can measure acceleration with four programmable full-scale ranges of 2g, 4g, 8g, and 16g (where g is accel full scale range). Three 16-bit analog-to-digital converters sample the three axes of movement simultaneously in the MPU6050 along X, Y and Z axis. [35]

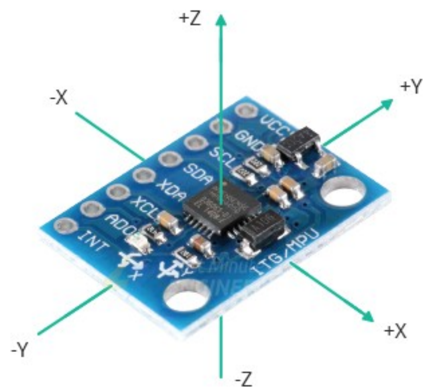

Figure 42: Measuring acceleration [35]

The MPU6050 has an on-chip gyroscope that can monitor angular rotation with four programmed full-scale ranges of 250°/s, 500°/s, 1000°/s, and 2000°/s. It has another three 16-bit analog-to-digital converters sample the three axes of rotation along X, Y and Z axis. From 3.9 to 8000 samples per second, the sampling rate can be adjusted. [35]

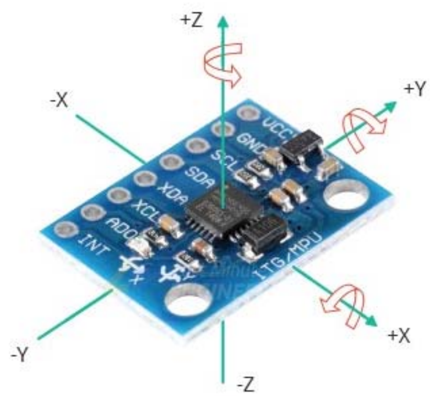

Figure 43: Measuring Rotation [35]

4.2.1 Arduino and gyroscope

The MPU 6050 communicates with the Arduino through the I2C protocol. It supports two different I2C addresses: $0x68_{HEX}$ and $0x69_{HEX}$. This enables the usage of two MPU6050s on the same bus or to avoid address conflicts with another device on the bus. The ADO pin determines the I2C address of the module. A 4.7K pull-down resistor is integrated into this pin. When the ADO pin is not connected, the default I2C address is 0x68HEX; however, when the pin is connected to 3.3V, the line is pulled HIGH, and the I2C address changes to 0x69HEX. [35]

Connection of Arduino with MPU6050 can be made by connecting VCC pin to the 5V output on the Arduino and connect GND to ground. Now connect the pins that are used for I2C communication. Each Arduino Board has different I2C pins which should be connected accordingly. In our case for Arduino nano the SDA (data line) and SCL (clock line) are on the pin headers close to the AREF pin. They are also known as A5 (SCL) and A4 (SDA). Lastly connect the INT pin of MPU6050 to digital pin 2 of Arduino. [35]

After making the connection add the libraries, navigate to sketch> include library> manage library. After installing the libraries correctly MPU6050 add in examples. [35] Next, upload the code in Arduino. Open up the serial monitor after uploading the code and set the baud rate to the value specified. Next, check "Initializing I2C devices ..." on the serial monitor that says, "Send any character to begin DMP programming and demo." Just type in any character on the serial monitor and send it, and values of yaw, pitch, and roll coming in from the MPU 6050 which shows that MPU6050 is successfully interface with Arduino. The MPU 6050 is connected to Arduino as shown in the following diagram below. [36]

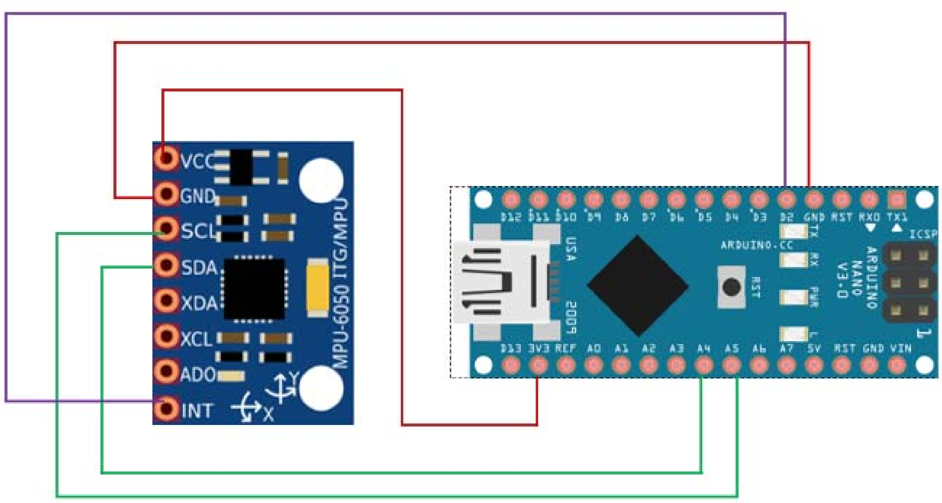

Figure 44: Arduino nano interfacing with MPU6050

4.3 Motor control

Every motor driver has 4 outputs output 1, output 2, output 3, and output 4. One motor is connected with two outputs. Output 1 is connected with positive terminal and output 2 is connected with negative terminal for motor to move in forward direction. Output 3 is connected with negative and output 4 is connected with positive terminal to move the motor in backward direction. In this way two motors are connected with one motor driver IC.

To move the robot the robot in forward direction, all outputs 1 and 3 of motor driver ICs are HIGH and all outputs of 2 and 4 are LOW. In this way all motors move in forward direction.

```
// forward condition
if (data[0] < 310 && data[0] > 200) {
    Serial.print("foractivate");
  //   put your main code here, to run repeatedly:
  digitalWrite(Amotor1pin1, HIGH);
  digitalWrite(Amotor1pin2, LOW);
  digitalWrite(Bmotor1pin1, HIGH);
  digitalWrite(Bmotor1pin2, LOW);

  ////
  digitalWrite(Bmotor2pin1, HIGH);
  digitalWrite(Bmotor2pin2, LOW);
  digitalWrite(Amotor2pin1, HIGH);
  digitalWrite(Amotor2pin2, LOW);
```

Figure 45: Code for forward motion of robot

To move the robot the robot in backward direction, all outputs 1 and 3 of motor driver ICs are LOW and all outputs of 2 and 4 are HIGH. In this way all motors move in backward direction.

```
// backward condition
if (data[0] > 360) {
  Serial.print("backactivate");
  //  put your main code here, to run repeatedly:
  digitalWrite(Amotor1pin1, LOW);
  digitalWrite(Amotor1pin2, HIGH);
  digitalWrite(Bmotor1pin1, LOW);
  digitalWrite(Bmotor1pin2, HIGH);

  ////
  digitalWrite(Bmotor2pin1, LOW);
  digitalWrite(Bmotor2pin2, HIGH);
  digitalWrite(Amotor2pin1, LOW);
  digitalWrite(Amotor2pin2, HIGH);
```

Figure 46: Code for backward motion of robot

To move the robot in left direction, right side of robot wheels are move in backward direction and left side of wheels are move in forward direction. To do this we connect the left side of wheels with one motor driver IC and right side of wheels with other motor driver IC. When we give left command, the outputs 1 and 3 of first motor driver IC are HIGH and outputs of 2 and 4 are LOW and the outputs 1 and 3 of second motor driver IC are LOW and outputs of 2 and 4 are HIGH.

```
 // left
 if (data[1] < 115) {
    Serial.print("activate");
   //  put your main code here, to run repeatedly:
   digitalWrite(Amotor1pin1, LOW);
   digitalWrite(Amotor1pin2, HIGH);
   digitalWrite(Bmotor1pin1, HIGH);
   digitalWrite(Bmotor1pin2, LOW);

   ////
   digitalWrite(Bmotor2pin1, HIGH);
   digitalWrite(Bmotor2pin2, LOW);
   digitalWrite(Amotor2pin1, LOW);
   digitalWrite(Amotor2pin2, HIGH);
//
```

Figure 47: Code for left motion of robot

To move the robot in right direction, right side of robot are move in backward direction and left side of wheels are move in forward direction. The connections are reverse in this case. When giving right command, the outputs 1 and 3 of first motor driver IC are LOW and outputs of 2 and 4 are HIGH and the outputs 1 and 3 of second motor driver IC are HIGH and outputs of 2 and 4 are LOW.

```
// right
if (data[1] > 180) {
  Serial.print("activate");
  //  put your main code here, to run repeatedly:
  digitalWrite(Amotor1pin1, HIGH);
  digitalWrite(Amotor1pin2, LOW);
  digitalWrite(Bmotor1pin1, LOW);
  digitalWrite(Bmotor1pin2, HIGH);

  ////
  digitalWrite(Bmotor2pin1, LOW);
  digitalWrite(Bmotor2pin2, HIGH);
  digitalWrite(Amotor2pin1, HIGH);
  digitalWrite(Amotor2pin2, LOW);
//
```

Figure 48: Code for right motion of robot

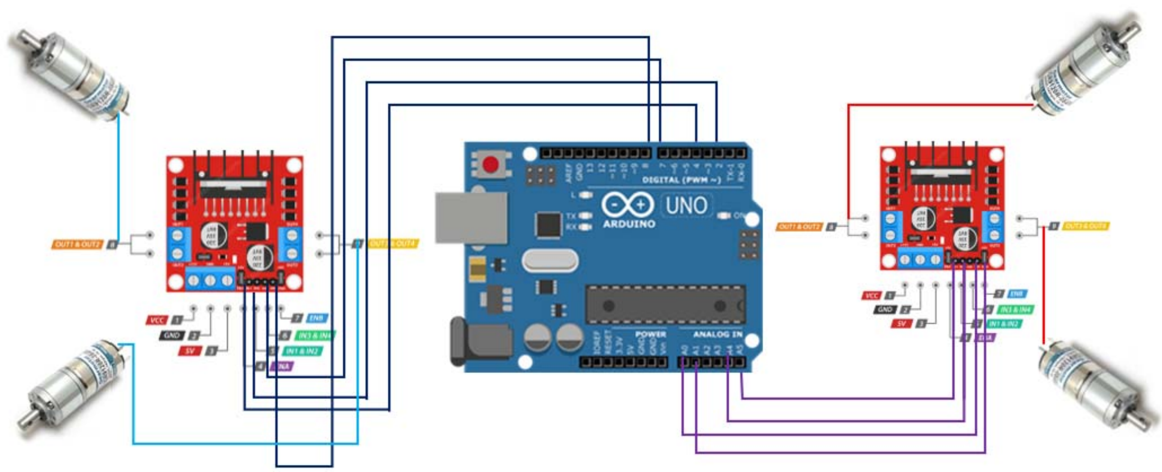

Figure 49: DC Motors with L298N and Arduino UNO

In our project we used 4 DC motors with two L298N motor driver IC's so digitals pins of Arduino D4, D2, D7 and D8 are connected to IN1, IN2, IN3 and IN4 of one motor driver IC and analogue pins A5, A2, A0 and A1 are connected to IN1, IN2, IN3 and IN4 of second motor driver IC. Supply the power of 5V and 12 to motor driver IC by 12 V and 5V by regulator. [37]

The below truth table shows the various mode of operation for motors.

In1	In2	In3	In4	Description
1	**0**	**1**	**0**	**Motor moves forward**
0	**1**	**0**	**1**	**Motor moves backward**
0	**0**	**0**	**0**	**Motors are stopped**

Table 2: Truth table of motors working

4.4 Sensor data mapping

This subsection will cover the control of robot by using gyro-based glove. We used the MPU-6050 Accelerometer sensor for direction control and nRF24L01 + PA/LNA module to provide remote communication.

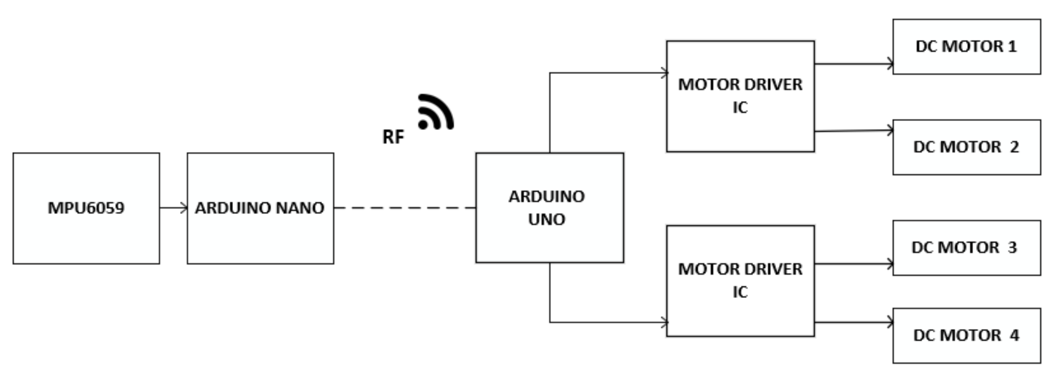

Figure 50: Proposed block diagram for Motor Control

In receiving side, we have raspberry pi along with pi camera, servo motors, nRF24L01+ PA/LNA receiver, Arduino uno that is attached with two motor driver IC, that are further connected with 4 DC motors. The motor driver IC is used to supply the appropriate current to the robot's motors in order for it to move. either in forward, backward, left and right direction. [38]

The transmitter section of the robot, which comprises of Arduino, MPU6050de, and

RF Transmitter, will continually monitor the MPU6050 sensor after it is turned on. [38] The accelerometer sensor value of x axis and y axis are coming from MPU6050. If the values lie in the range of 200-300 then we give forward command, and if the values are greater than 360 then we give backward command. To give left command values are less than 115 and to give right command values lies above than 180.

Based on the direction of the MPU6050 Sensor, this data is recorded by the Arduino, and send that to Arduino on robot. [38] The communication between two Arduino take place through RF transceiver.

The RF Receiver accepts serial data from sensor in the receiver portion. The serial data will give propriate command to the motor driver IC to move the robot. The movement of the motors, and hence the movement of the robot, is defined based on the data. [38]

Figure 51: Find construction of Glove

Chapter 5

5 Analysis and Evaluation

This chapter includes the analysis and evaluation of project.

5.1 Project Block Diagram

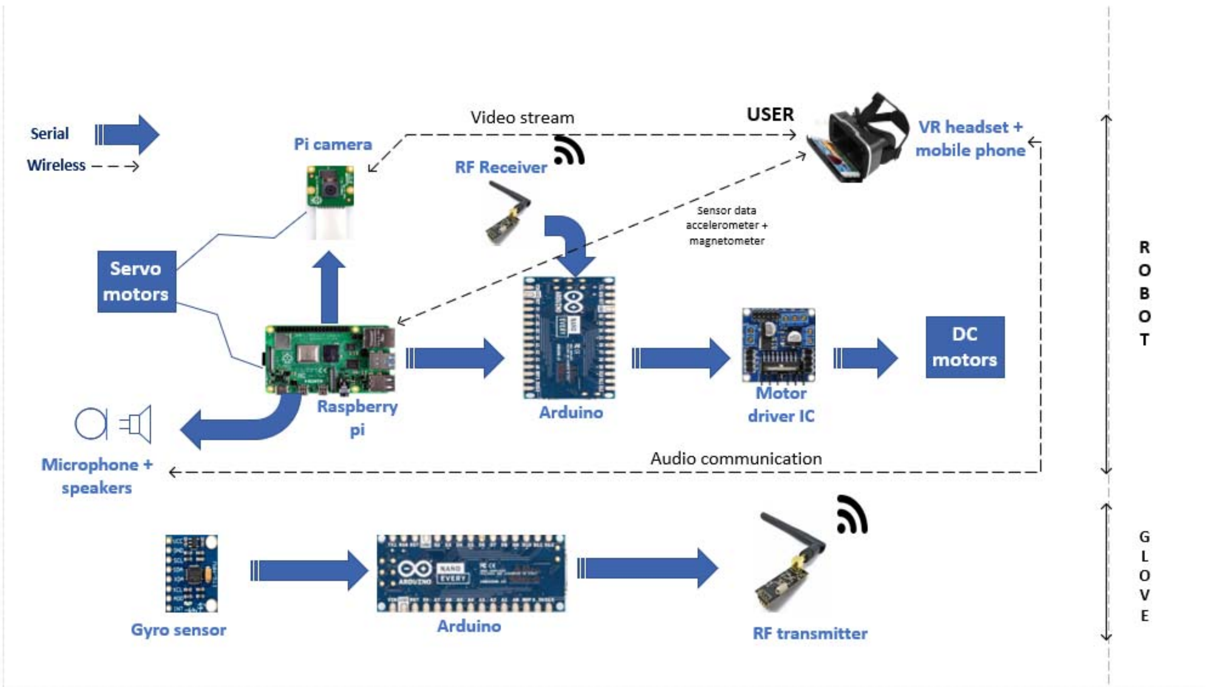

Figure 52: Methodology

The overall system diagram of our system is shown in Figure 11. The system has two parts, the first one is of robot side that contains various hardware that will be on patient's side. The second part is of glove that will be on doctor's side which control the movement of robot in any remote location.

Project consists of real-time video transmission from the raspberry pi to mobile phone, the video is configured in dual-screen and viewed in Virtual Reality environment. The two-way audio transmission is achieved using microphones and speakers attached with raspberry pi and software tools. Head movements of user are tracked using MPU+GPS stream application and the robot camera is moved accordingly, giving an immersive experience. A gesture-based navigation system is developed using gyro-scope glove to control the motion of robot. The gyroscope data from the glove is sent to the robot via the nRF24L01 communication module and that data is used to give navigation commands to the motor driver IC. A pop-up module displays important clinical information of the detected patient in front of doctor in a pop-up form in VR.

5.2 Controlling of VIRA with gyro Glove

The navigation of VIRA is based on the gyro-based gesture recognition glove. VIRA motion is controlled via a gesture-based navigation system using Gyro-scope glove. The gyroscope data from the glove is sent to the robot via the RF Transceiver communication module and that data is used to give navigation commands to the motor driver IC. Arduino Nano is interfaced with gyro to send data to robot according to the hand gesture of doctor.

The glove is mounted with gyro-sensor that change their values when hand bend. When hand bent downward VIRA moves backward, when its bend upward, VIRA moves forward and when hand bent left or right, VIRA moves left or right respectively.

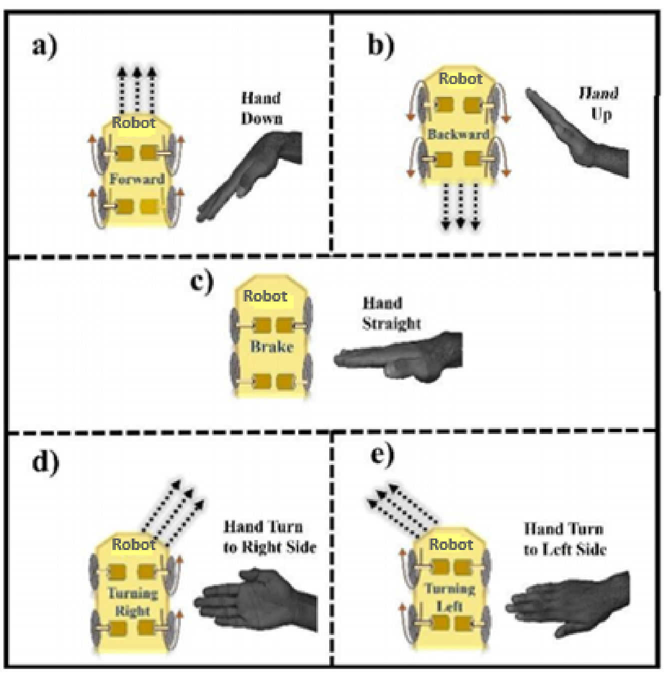

Figure 53: Gesture recognition theme [38]

5.3 Objectives Achieved

In our work we have developed a telepresence robot specifically for medical environment where doctors can monitor and examine a patient from any part of world. It provides live video streaming and audio communication, so doctor can see the patient side in form of VR and can communicate with the patient easily in virtual reality and feel the presence of remote location as being present there.

Moreover, VIRA detect the QR code reader of patient's bed which will show the patient's history to the doctor in form of pop-up. The movement of robot will be controlled by hand gesture of the user by using Gyroscope glove. It will provide the doctor ease to move the robot according to his own will, he can examine patient, he can give diagnosis whether the patient is having a stroke, epilepsy and give consultant to medical staff how to deal with patient present in remote location.

Chapter 6

6 Future Work and Conclusion

Future work and conclusion drawn from this project are discussed in this chapter.

6.1 Future work

Presented in this thesis is the prototype development of VIRA. There are many factors that can be improved to further enhance the experience of telepresence. Some of these factors are discussed in detail in this section.

6.1.1 Global network

To test the concept of telepresence with virtual reality, this project used a local network that restrict the communication between doctor and user to a local network. However, in order to use this project for long distance communication a global network solution is required. The use of a global network is likely to increase communication between doctor and patient from any part of the world. [22]

6.1.2 Adjustable height

This project's height is fixed and cannot be changed; however, in the future, for greater communication and ease, the robot's height can be changed so that it adjusts to the patient's height.

6.1.3 Docking station

This project is working with batteries and batteries needs charging when robot stop working. To remove the battery and plug it with power supply is not a very essential method for sufficient duration. A docking station for automatic battery exchanging and charging is required for robots. Using a docking station for autonomous robots is a viable way to enhance robot activity over time. In future it can be design in such a way that robot can go to docking station when indicated batteries are less than actual requirement and automatically leave the docking station on full charging. [39]

6.1.4 360° Camera

In this project, we are using 2D camera with servo motors but in the future to make the head tracking more efficient instead of using servo motors with 2D camera, we can use the 360◦ camera.

6.2 Conclusion

Healthcare services are essential for human being. Within increase in technology clearly point that robot, sooner or later, will be an indispensable part of the medical industry. Subsequently, automation is the key for maintaining the quality and sustainability of healthcare. This thesis summarizes the efforts to overcome some key technical problems arise in remote location and also focuses on solution to reduce the risk of infectious disease transmission. The people living in remote/rural areas of country need Vira to gain access to proper healthcare. The idea of this project will benefit those who do not have access to basic primary healthcare services.

In our work we have developed a simple and cost-effective virtual reality telepresence robot. Mainly it is designed for remote environments but can be extended to other applicable areas as well. It has an effective architecture while being cost effective especially in under-developed countries. It provides a video streaming capability, audio communication, pop-up feature which shows the history of patient, and it can be controlled remotely through gyro-based glove. A gyro-based glove is also developed for this robot, to control the robot remotely. As future enhancement, the robot can also be navigated through different network as well.

7 Bibliography

[1]	R. S. R. J. a. I. M. H. Zaka-ul-Haque, "Tele-presence Based Patient Monitoring Robot," *International Journal of Advanced Science and Technology Vol. 42,* p. 17, May 2012.
[2]	[Online]. Available: https://www.google.com/search?q=raspberry+pi+pic&sxsrf=ALeKk01rb-M0PFbyYQ_xChU53sXeMNs9jA:1623514744341&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjduf7cv5LxAhXQYMAKHf7zCjIQ_AUoAXoECAEQAw#imgrc=xfXOJl1VMvBldM.
[3]	[Online]. Available: https://www.google.com/search?q=raspberry+pi+camera+pic&tbm=isch&ved=2ahUKEwj3z5fev5LxAhWP0oUKHeUWDhQQ2-cCegQIABAA&oq=raspberry+pi+camera+pic&gs_lcp=CgNpbWcQAzoECCMQJzoCCAA6BggAEAgQHlDNmAVY7qgFYMWqBWgAcAB4AIABiQKIAfIKkgEDMi02mAEAoAEBqgELZ3dzLXdpei1pbWfAAQ.
[4]	[Online]. Available: https://www.google.com/search?q=vr+headset+pic++with+mobile+phone+&tbm=isch&ved=2ahUKEwj71Pe6wJLxAhUChRoKHf6yAoIQ2-cCegQIABAA&oq=vr+headset+pic++with+mobile+phone+&gs_lcp=CgNpbWcQAzoECCMQJzoGCAAQBxAeOgIIADoGCAAQBRAeOgYIABAIEB46BAgAEBg6BAgAEB5Q3YUDWJG-A2Dx.
[5]	[Online]. Available: https://www.google.com/search?q=arduino+nano+atmega+328&tbm=isch&ved=2ahUKEwikrtXXwJLxAhXQwIUKHTgyCZ0Q2-cCegQIABAA&oq=arduino+nano+atmega+328&gs_lcp=CgNpbWcQAzIGCAAQChAYMgYIABAKEBg6BAgjECc6BQgAELEDOggIABCxAxCDAToCCAA6BAgAEEM6BwgAELEDEEM6BAgAEB5QiL4EWKXyBG.
[6]	[Online]. Available: https://www.google.com/search?q=arduino+uno+atmega328p&tbm=isch&hl=en&sa=X&ved=2ahUKEwi9vv_IwZLxAhUPohoKHRd8ChUQBXoECAEQPQ&biw=1226&bih=516#imgrc=rP_QBumzYED0CM.
[7]	[Online]. Available: https://www.google.com/search?q=mpu6050&tbm=isch&ved=2ahUKEwjmtMHKwZLxAhUL-4UKHR3GBD4Q2-cCegQIABAA&oq=mpu6050&gs_lcp=CgNpbWcQAzIECCMQJzIECCMQJzIECAAQQzIECAAQQzIECAAQQzIECAAQQzICCAAyAggAMgIIADICCAA6BwgjEOoCECc6BQgAELEDOggIABCxAxCDAVCT1QFYwfABYM_0AWgBcAB4AY.
[8]	[Online]. Available: https://www.google.com/search?q=nrf24l01+la%2Fpna&tbm=isch&ved=2ahUKEwjxht6LwpLxAhUKmRoKHePSAtsQ2-cCegQIABAA&oq=nrf24l01+la%2Fpna&gs_lcp=CgNpbWcQAzoECCMQJzoECAAQQzoCCAA6BggAEAgQHjoECAAQGDoECAAQHlDlGVjZQGDHRWgDcAB4AIABkwWIAcQUkgEJMi00LjIuMS4xmAEAoAEBqgELZ3.
[9]	[Online]. Available: https://www.google.com/search?q=motor+driver+ic+l298n&sxsrf=ALeKk03wbY9hiW4btzuDsh2MFZ69BPXaAg:1623515489814&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiv0LrAwpLxAhWEZMAKHRbsDvgQ_AUoAXoECAEQAw&biw=1242&bih=516#imgrc=1OCpGJ0f3Mc9IM.

[10]	[Online]. Available: https://www.google.com/search?q=two+servo+motor+sg90+for+pan+tilt&tbm=isch&ved=2ahUKEwjM84n5w5LxAhVNiRoKHXVVC_YQ2-cCegQIABAA&oq=two+servo+motor+sg90+for+pan+tilt&gs_lcp=CgNpbWcQA1DqOljYWWDsW2gAcAB4AIABswKIAb8YkgEGMi0xMi4xmAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sc.
[11]	[Online]. Available: https://www.google.com/search?q=dc+motors+12v&tbm=isch&ved=2ahUKEwjR8LCmxJLxAhUBgc4BHdJXAIsQ2-cCegQIABAA&oq=dc+motors+12v&gs_lcp=CgNpbWcQAzIGCAAQCBAeOgQIABBDOgIIAFChQVj5SGDhTGgAcAB4AIABiQKIAfoHkgEDMi00mAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=Q-PEYJH.
[12]	[Online]. Available: https://www.google.com/search?q=speakers+&tbm=isch&ved=2ahUKEwiPq7PPxJLxAhUB1IUKHeMrBHEQ2-cCegQIABAA&oq=speakers+&gs_lcp=CgNpbWcQAzIECCMQJzICCAAyAggAMgIIADICCAAyAggAMgIIADICCAAyAggAMgIIAFDsiQFY0IsBYNeQAWgAcAB4AIAB9wGIAeEDkgEDMi0ymAEAoAEBqgELZ3dzLXdpei1pbW.
[13]	[Online]. Available: https://www.google.com/search?q=dry+lead+acid+battery&tbm=isch&ved=2ahUKEwi3irfZxJLxAhWHBhoKHRR7BHkQ2-cCegQIABAA&oq=dry+lead+acid+battery&gs_lcp=CgNpbWcQAzIECCMQJzIECAAQGDoHCCMQ6gIQJzoECAAQQzoHCAAQsQMQQzoFCAAQsQM6AggAOgYIABAIEB46BggAEAoQGFCZqwJYxdkCYJDdAm.
[14]	[Online]. Available: https://www.google.com/search?q=pycharm+logo&tbm=isch&ved=2ahUKEwiNgu7vxJLxAhUNwYUKHXtmBz4Q2-cCegQIABAA&oq=pycharm+logo&gs_lcp=CgNpbWcQAzICCAAyAggAMgIIADICCAAyAggAMgIIADIGCAAQBRAeMgYIABAFEB4yBggAEAUQHjIGCAAQCBAeOgcIIxDqAhAnOgQIIxAnOgUIABCxAzoICAAQsQMQgwE6.
[15]	[Online]. Available: https://www.google.com/search?q=opencv++logo&tbm=isch&ved=2ahUKEwjEzcb_xJLxAhUF4oUKHawwAr8Q2-cCegQIABAA&oq=opencv++logo&gs_lcp=CgNpbWcQAzICCAAyAggAMgIIADICCAAyAggAMgIIADICCAAyBggAEAcQHjIGCAAQBRAeMgQIABAYOgQIABBDUMqoAVjtuAFg570BaABwAHgAgAG9A4gB0BCSAQcyLTcu.
[16]	[Online]. Available: https://www.google.com/search?q=arduino+ide+logo&tbm=isch&ved=2ahUKEwji29ibxZLxAhWY44UKHbW5Be4Q2-cCegQIABAA&oq=arduino+ide+logo&gs_lcp=CgNpbWcQAzIECAAQQzIECAAQQzIGCAAQCBAeOgIIADoECAAQGFCqI1icK2CHLmgAcAB4AIABuwKIAcUKkgEFMi00LjGYAQCgAQGqAQtnd3Mtd2l6LWltZ8AB.
[17]	[Online]. Available: https://www.google.com/search?q=soundwire+logo&tbm=isch&ved=2ahUKEwjI79C3xZLxAhW-gM4BHWwGCRAQ2-cCegQIABAA&oq=soundwire+logo&gs_lcp=CgNpbWcQAzoECCMQJzoHCAAQsQMQQzoCCAA6BQgAELEDOgQIABBDOgYIABAHEB46CAgAEAcQBRAeUK5oWOqJAWCbiwFoAXAAeACAAf8BiAHBEpIBBDItMTCYAQCg.
[18]	[Online]. Available: https://www.google.com/search?q=airmusic+logo&sxsrf=ALeKk02pYzOlQ_SJ_bNKIdHYeXuW2cAUCA:1623516408152&tbm=isch&source=iu&ictx=1&fir=c-F0u4T4yAhKqM%252CXyToEK1Eodu6qM%252C_&vet=1&usg=AI4_-kQC6mr2CMeJC5sIRgzpMrAKykiFGw&sa=X&ved=2ahUKEwjqr632xZLxAhWMY8AKHX_GC.
[19]	Boston Dynamics, [Online]. Available: https://www.bostondynamics.com/spot.

[20]	"Ava 500," Telepresence Robots, [Online]. Available: https://telepresencerobots.com/robots/ava-robotics-ava-500.
[21]	"OHMNI TELEPRESENCE ROBOT," OHMNILABS, [Online]. Available: https://ohmnilabs.com/products/ohmnirobot/.
[22]	G. Rydenfors, "Telepresence and remote communication through virtual reality," *Linkoping University Electronic Press,* p. 54, 09 June 2017.
[23]	H. S. C. S. M. T. D. a. K. S. Hari Prabhat Gupta, "A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction using Accelerometer and Gyroscope sensors," *DOI 10.1109/JSEN.2016.2581023, IEEE Sensors,* p. 10, 15 August 2016.
[24]	a. P. G. J. Dr. G . G Sivasankari, "Live Video Streaming using Raspberry Pi in IOT Devices," *International Journal of Engineering Research & Technology (IJERT) Volume 5, Issue 19,* p. 5, 2017.
[25]	[Online]. Available: https://www.google.com/search?q=mechanism+of+gearing+down+pic&hl=en&sxsrf=ALeKk02EoYYNt28MzIetI3bS-el13Twfjg:1623516560777&source=lnms&tbm=isch&sa=X&ved=2ahUKEwih-pC_xpLxAhUKiFwKHZcSB_0Q_AUoAXoECAEQAw&biw=1242&bih=516#imgrc=7DVOtwlLlvY8PM.
[26]	"Getting Started With the Camera Module," Raspberry pi, [Online]. Available: https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/3.
[27]	"VIdeo Streaming with Raspberry pi Camera," 01 October 2017. [Online]. Available: https://randomnerdtutorials.com/video-streaming-with-raspberry-pi-camera/. [Accessed 01 December 2020].
[28]	"picamera," ReadTheDocs, [Online]. Available: https://picamera.readthedocs.io/en/latest/index.html. [Accessed 06 06 2021].
[29]	"Monoscopic vs Stereoscopic 360 VR," Immersion VR, [Online]. Available: https://immersionvr.co.uk/blog/monoscopic-vs-stereoscopic-360-vr/.
[30]	"SoundWire Audio Communication System," [Online]. Available: https://georgielabs.altervista.org/SoundWireHelp.html.
[31]	"How to set up a Raspberry Pi AirPlay receiver," [Online]. Available: https://thepi.io/how-to-set-up-a-raspberry-pi-airplay-receiver/.
[32]	S. Labs, "SDL_Pi_MJPEGStream," GItHub, 20 07 2019. [Online]. Available: https://github.com/switchdoclabs/SDL_Pi_MJPEGStream.
[33]	"How nRF24L01+ Wireless Module Works & Interface with Arduino," [Online]. Available: https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/.
[34]	Dejan, "Arduino and MPU6050 Accelerometer and Gyroscope Tutorial," 2019. [Online]. Available: https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/.
[35]	"Interface MPU6050 Accelerometer and Gyroscope Sensor with Arduino," [Online]. Available: https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/.
[36]	A. Sanjeev, "How to Interface Arduino and the MPU 6050 Sensor," 21 March 2018. [Online]. Available: https://maker.pro/arduino/tutorial/how-to-interface-arduino-and-the-mpu-6050-sensor.
[37]	"Wiring & Driving the L298n H Bridge with 2 to 4 DC Motors," [Online]. Available: https://www.14core.com/wiring-driving-the-l298n-h-bridge-on-2-to-4-dc-motors/.
[38]	"Hand Gesture Controlled Robot using Arduino," [Online]. Available: https://www.electronicshub.org/hand-gesture-controlled-robot/.
[39]	M.-C. T. a. Y.-J. T. Yi-Cheng Wu, "Robot Docking Station for Automatic Battery Exchanging and Charging," *International Conference on Robotics and Biomimetics,* p. 4, February 2009.
[40]	,. Z. M. L. A. M. A. M. Saleem Ullah, "An Automated Robot-Car Control System with HandGestures and Mobile Application Using Arduino," p. 17, January 2019.

VIRA

ORIGINALITY REPORT

11%	8%	1%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	fics.nust.edu.pk Internet Source	1%
2	lastminuteengineers.com Internet Source	1%
3	Submitted to RMIT University Student Paper	1%
4	www.hackster.io Internet Source	1%
5	Submitted to University of Leicester Student Paper	<1%
6	Submitted to Embry Riddle Aeronautical University Student Paper	<1%
7	acoptex.com Internet Source	<1%
8	Submitted to Higher Education Commission Pakistan Student Paper	<1%

9	Submitted to University of Northumbria at Newcastle Student Paper	<1 %
10	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
11	protosupplies.com Internet Source	<1 %
12	chit-chattinggeneralcomputing.blogspot.com Internet Source	<1 %
13	maker.pro Internet Source	<1 %
14	Submitted to Madanapalle Institute of Technology and Science Student Paper	<1 %
15	Submitted to Institute of Technology, Nirma University Student Paper	<1 %
16	Submitted to University of Wolverhampton Student Paper	<1 %
17	V. Indragandhi, A. Chitra, R. Raja Singh, Aishvardhan Bajiya, Yash Tilak, V. Subramaniyaswamy. "Optimal design, control and implementation of multi-drones for commercial utility", International Journal of Intelligent Unmanned Systems, 2020	<1 %

18	Submitted to University of Balamand Student Paper	<1%
19	apkappsstores.com Internet Source	<1%
20	Submitted to Queensland University of Technology Student Paper	<1%
21	Submitted to University of Leeds Student Paper	<1%
22	journalarticle.ukm.my Internet Source	<1%
23	hdl.handle.net Internet Source	<1%
24	Geoffroy Moucheboeuf, Romain Griffier, David Gasq, Bertrand Glize, Laurent Bouyer, Patrick Dehail, Helene Cassoudesalle. "Effects of robotic gait training after stroke: A meta-analysis", Annals of Physical and Rehabilitation Medicine, 2020 Publication	<1%
25	Submitted to Institute of Technology Blanchardstown Student Paper	<1%
26	Submitted to School of Business & Computer Science Limited	<1%

27	Submitted to JK Lakshmipat University Student Paper	<1%
28	Submitted to Softwarica College Of IT & E-Commerce Student Paper	<1%
29	Submitted to Strayer University Student Paper	<1%
30	Submitted to iGroup Student Paper	<1%
31	www.answers.com Internet Source	<1%
32	www.electronicslovers.com Internet Source	<1%
33	Submitted to AUT University Student Paper	<1%
34	Submitted to Heriot-Watt University Student Paper	<1%
35	Submitted to University of Brighton Student Paper	<1%
36	www.modelsoundsinc.com Internet Source	<1%
37	www.mymindsi.com Internet Source	<1%

38	Mahdi Jampour, Saeid Abbaasi, Malihe Javidi. "CapsNet Regularization and its Conjugation with ResNet for Signature Identification", Pattern Recognition, 2021 Publication	<1 %
39	Submitted to Universiti Malaysia Perlis Student Paper	<1 %
40	Submitted to Universiti Tenaga Nasional Student Paper	<1 %
41	Hari Prabhat Gupta, Haresh S. Chudgar, Siddhartha Mukherjee, Tanima Dutta, Kulwant Sharma. "A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors", IEEE Sensors Journal, 2016 Publication	<1 %

Exclude quotes Off

Exclude bibliography Off

Exclude matches Off