VR Surveillance Bot

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Electrical and Electronics Engineering

by

Unnikrishnan R. Menon

17BEE0047

Atharva Hudlikar

17BEE0100

Under the guidance of

Dr. P. Mahalakshmi

School of Electrical Engineering,

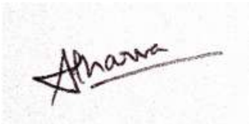
VIT, Vellore.

June, 2021

DECLARATION

I hereby declare that the thesis entitled VR Surveillance Bot submitted by me, for the award of the degree of *Bachelor of Technology in Electrical and Electronics Engineering* to VIT is a record of bonafide work carried out by me under the supervision of Dr. P. Mahalakshmi.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 2nd June 2021

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled “VR Surveillance Bot” submitted by **Unnikrishnan Menon (17BEE0047) & Atharva Hudlikar (17BEE0100)**, School of Electrical Engineering, VIT, Vellore, for the award of the degree of *Bachelor of Technology in Electrical and Electronics Engineering*, is a record of bonafide work carried out by him under my supervision, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 2nd June 2021

P. Mahalakshmi

Signature of the Guide

(Dr. P. Mahalakshmi)

The thesis is satisfactory / unsatisfactory

Approved by

Head of the Department

EEE

Dean, SELECT

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to our Project Guide Dr. P. Mahalakshmi, Professor, Department of Instrumentation, SELECT, for her guidance and constant support throughout this project. I would also like to offer special thanks to the Head of Department and Dean of SELECT School for timely communications and assistance. I am also deeply grateful to all the panel members present during our reviews for their insightful comments and suggestions, without which, the project would not have reached completion.

Atharva Hudlikar

Unnikrishnan Menon

Executive Summary

Numerous operations and tasks are too perilous for humans to perform in person. These risks may arise due to various reasons: alien environments, radioactive conditions, unmapped terrain and so on. More recently, with the Covid-19 pandemic, a new hazard of social contact has come into play. In such cases, it is safer and much more viable to have a robot that can be teleoperated from a remote location to avoid human presence. This task necessitates the establishment of an immersive telepresence that allows the user to react to situations more effectively as well as provides a better understanding of the environment. This thesis presents a thorough examination of the technology and design for such a robot. The proposed design employs a virtual reality controller developed for smartphones that allows the user to immerse into the robot's environment through its real-time video stream. The prototype developed during this research can traverse environments and returning relevant data to the user; a technology that can easily be adapted to different types of robots with varying degrees of sophistication.

CONTENTS

List of Figures

List of Abbreviations

VR	Virtual Reality
IoT	Internet of Things
TCP/IP	Transmission Control Protocol/ Internet Protocol
GSM	Global System for Mobile Communications
CPU	Central Processing Unit
Wi-fi	Wireless-Fidelity
GPU	Graphics Processing Unit
GPIO	General Purpose Input Output
PWM	Pulse Width Modulation

1. INTRODUCTION

1.1. OBJECTIVES

- To build an autonomous robot capable of traversing alien environments and send relevant data retrieved back to the user.
- To develop a VR based controller that will allow the user to take control of the bot and use the real-time video feed from the bot to manipulate various aspects of the bot.

The controller and VR headset will be connected to the bot via a Flask based server that will act as a medium for transmitting sensor data as well as the video stream from the bot to the user in control.

1.2. MOTIVATION

Remotely operated robots find applications in many operations such as search and rescue, mapping an alien environment, remotely operating robotic arms in nuclear reactors, underwater or even space stations. For such robots, situational awareness and control are integral to its design and implementation, especially as the robot is teleoperated. The better the human controllers can visualize themselves in place of the robot, the better they can react and respond to sudden changes in the environment around the robot and relay more informed instructions to the robot. Thus, creating a telepresence of the robot's environment around the controller becomes paramount.

During the Covid-19 pandemic, it has become imperative to ensure personal protection and safety. The high rate of infections has put the lives of healthcare workers and the safety of essential services personnel largely at risk [1]. In such scenarios, using robots [2] as a means of maintaining social distancing as well as enabling telemedicine technologies [3] has caught the eye of many [4]. Introducing VR Control systems for such robots not only allow more complex robots to be used in the medical sector, but due to the enabling of remote operations, the same human touch can be given while ensuring the safety of all workers. These tasks, ranging from basic diagnosis to minor checkups can be performed remotely by doctors, without having to be physically present, controlling the robots; and due to the VR control, the accurate and real time visual feedback will allow the doctors to perform tasks efficiently.

With the rise of social trends and substantial technological innovations, there are now a plethora of scalable channels that allow live broadcasting of different types of data. Big buffers, powerful compression algorithms, and more efficient transcoding methods, along with scalability to serve a larger audience, have resulted in a noticeable delay on these live streaming platforms. Since most of these platforms serve as a medium to broadcast data to non-interactive audiences, the delay renders no negative impact. However, since the user's activities are mirrored in the data stream, this delay may trigger a slew of issues in interactive applications [5].

In remotely controlled robots, the robot's actions in an environment are controlled by the user. The observations of the environment from the robot's perspective define the current state of the robot. These observations act as a feedback for the user so that they can analyze the scenario and make decisions to control the robot. Introducing significant latency in such interactive systems leads to uncertainty and instability. Model predictive controller-based techniques have previously been designed to compensate for large time delays caused by the internet [6].

Due to the significant advancements in communication and IoT technologies, it is now possible to send and receive video data on various devices across a network with minimal latency and provide an interactive VR presence to the user, of the environment around the robot. Video monitoring has evolved considerably over the years and is now a critical tool for many organizations in terms of safety and protection [7]. These devices are constantly being used in a variety of settings, from the home to the workplace. Existing structures are being enhanced and expanded with new elements and functionalities [8].

The proposed design employs a virtual reality controller developed for Android platform that allows the user to take control of the bot and modify different aspects of it using the bot's real-time video stream. The robot holds a raspberry pi model 4 that hosts a server as the backend for seamless duplex communication between the user's virtual reality headset and the onboard electronics.

The main goal of this project is to introduce a virtual reality interface-based control on a real-time mobile robot using a smartphone, as well as to provide an overview of which practical scenarios this system is better suited for.

1.3. BACKGROUND

Many researchers have conducted studies on mobile robots for applying them in the defence sector in the past. Yoichi Shimosasa *et al.*, created an Autonomous Guard Robot that combines the defence and service systems into a self-contained guard robot. During the day, the robot will lead tourists and patrol at night [9]. Recently, cell phone-based robotic systems have gotten a lot of press. Ren C.Luo created a protection robot device that can be operated by a cellular phone using GSM without the need for image data [10].

Various technological advances in video monitoring are made possible using Closed-Circuit Cameras [11]. Many felony scenes have been investigated with this tool, but the crime rate has not decreased due to the immobility of detection devices.

Several global ventures have attempted to build mobile security platforms in recent years. The Mobile Detection Assessment and Response System (MDARS), developed by HR Everett and DW Gage, is one significant example. The goal of this project was to create a multi-robot device capable of monitoring warehouses and storage facilities, detecting anomalous conditions such as flooding and fire, detecting trespassers, and evaluating the status of inventoried items using sophisticated RF transponders [12].

A security robot created at the University of Waikato in Hamilton, New Zealand, is another example. It's called MARVIN (Mobile Autonomous Robotic Vehicle for Indoor Navigation) and it's meant to serve as a protective agent in enclosed spaces [13].

Sung Wook Moon developed a phone-controlled monitoring device that enabled video images to be transmitted from a spider robot to a smart phone. However, the 200ms time interval between the camera and the cell phone made remote control ineffective [14].

Yong-Ho Seo *et al.*, created a Touch-Based Control of a Robot Device that could remotely collect information from the robot and control the robot using a mobile phone touch event [15].

Another notable project includes a Wi-Fi monitoring bot developed by Diksha Singh *et al.*, where an entire new approach for controlling the robot through an IoT software called Blynk has been proposed. Arduino Uno R3 Based Robot Control Board was used to design the robot [16]. G. Anandravisekar *et al.,* developed a robot to perform continuous surveillance in domestic areas. It has the potential to minimize both human labour and human error. The robot is operated by a mobile device or a laptop via the Internet of Things using Cayenne software, which sends commands to the robotic system [7].

Reconnaissance systems have played a crucial role in monitoring human activity in the war zone and frontier areas, to reduce the risk of human life by allowing soldiers of armed forces to ascertain the state of the area before approaching it. J. Patolia *et al.,* have developed a war field robot using an Arduino uno board with an L293D motor controller, an HC-05 Bluetooth module, and a night vision wireless camera. An android application was created through MIT app inventor and was used for the total navigation control of the robot [17]. Shuddha Chowdhury *et al.*, also announced a design of a robot that was intended to serve during natural disasters such as cyclones, earthquakes, landslides, and so on. The robot detects humans or animals stuck under fallen buildings using an infrared camera. The proposed robot's main disadvantage was the inability to detect the robot's real-time position [18].

Security devices have been developed for remote discovery and monitoring, but they also lack the ease of use needed by the elderly and disabled. Research has been conducted to emphasize the importance of functional prototypes that consider the cognitive shortcomings of this category of people. This issue has been addressed by the creation of a novel vehicular Remote Exploration Surveillance Robot (RESBot) capable of tracking the world in real time in response to incidents. Natural language commands are used to communicate with the machine, which improves accessibility over conventional approaches [19].

2. PROJECT DESCRIPTION AND GOALS

During the Covid-19 pandemic, it has been critical to ensure personal safety and security. The high prevalence of infections has placed healthcare workers' lives and the wellbeing of essential services employees in jeopardy. Introducing VR Control systems for such robotics not only allows more sophisticated robots to be used in the medical field, but also allows for the same human interaction to be provided thus maintaining the safety of all staff due to the ability to perform remote operations. This thesis examines the technology and architecture for such a robot in great detail. The proposed design uses a smartphone-based virtual reality controller that allows the user to immerse themselves in the robot's environment through a real-time video stream. This research's prototype can traverse environments and returning relevant data to the user.

Our main goals were to create an autonomous robot capable of traversing alien environments and returning relevant data to the user, as well as to create a VR-based controller that would enable the user to take control of the bot and modify various aspects of the bot using the bot's real-time video feed.

The controller and VR headset will be linked to the bot through a Flask-based server, which will function as a conduit for sensor data and the bot's video stream to the user in command.

3. TECHNICAL SPECIFICATIONS

The proposed robot design consists of a wheeled platform which is driven by 2 DC geared motors (BO motor L type) that can provide up to 300 RPM. These motors, which weigh about 40 gm, can operate between $3V$ and $9V$. They have a torque of 4.2 $Kgf.cm$, a no-load current of 60 mA, and a stall current of 700 mA. These drive motors derive power from two series-connected $9V$ batteries via an L298N motor driver circuit. A freely rotating castor wheel, placed under the front section of the base platform prevents the whole robot from tipping over as well as provides for free movement. A smaller secondary platform is screwed on top of the base panel using 25 mm long hex spacers to provide additional space to accommodate the onboard electronic components.

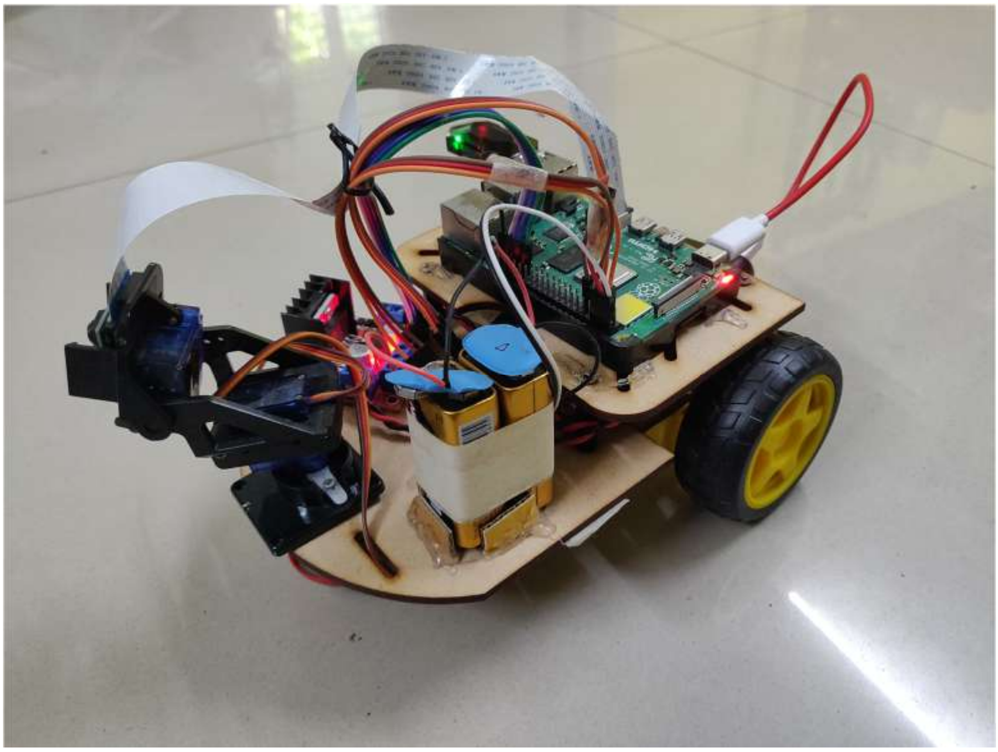


Figure 1. Working Prototype of VR Surveillance Bot

Both platforms are made of medium density fiberboard. This wood fiber-based material, when pushed onto a crafting board creates a very solid board that is robust enough to not bend while carrying the weight of all onboard electronic components. The top platform is oriented such that it leaves some extra space on the base panel to place a 2 degree of freedom pan/tilt camera mounting bracket towards the front of the robot. This camera mounting setup makes use of two

$SG-90$ servo motors (weighing $9g$ each) to enable motion across the horizontal and vertical axes. The pan/tilt servo motors and the L298N motor driver circuit receive signals from a Raspberry Pi 4 model B (8 GB RAM variant) which is placed on the top platform towards the back side of the robot.

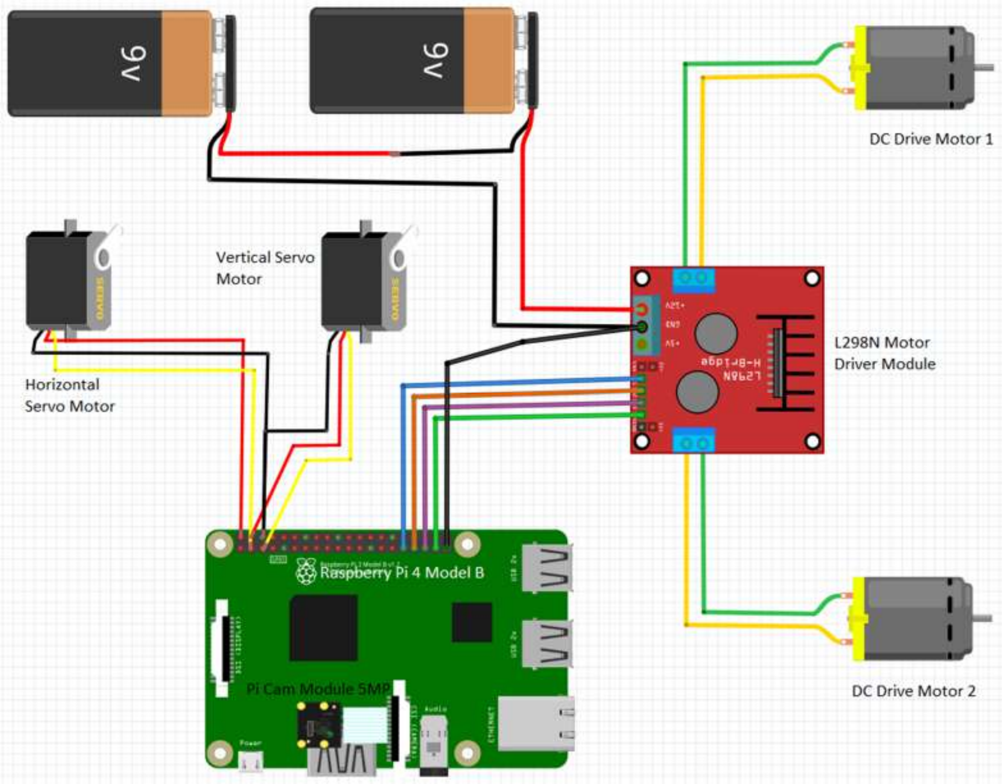


Figure 2. Circuit schematic simulated on Fritzing

The Raspberry Pi is used by the robot to process data, transmit and receive data, and monitor the robot. The Raspberry Pi was selected because it can handle several tasks without overburdening the CPU and can easily communicate with our preferred peripherals. This single-board computer is the main onboard data processing unit which also hosts the server backend. The raspberry pi's built-in wifi module enables a stable wireless connection to the network without any auxiliary requirements. For enabling remote video streaming, the pan/tilt mount houses a picam which is connected to the raspberry pi via a REES52 camera cable. The picam module is a 5 MP camera capable of full HD video capture in $h264$ format and taking still images while recording a video. The Raspberry pi derives it's power from a compact Lithium Polymer rechargeable power bank unit (18 W Power Delivery; 10000 mAh capacity), strategically sandwiched in between the base and top panels to optimize space utilization.

4. DESIGN APPROACH AND DETAILS

4.1. DESIGN APPROACH

The VR bot's locomotion is controlled by a WiFi-enabled gaming controller. For concept testing, we intend to use a dual joystick controller. To reduce latency, the controller's signals are then sent via Wi-Fi to the Raspberry Pi for parsing at a satisfactory refresh rate. The popular python game development package called Pygame parses the joystick axis commands and converts them to the appropriate GPIO on-off signals on the Raspberry Pi, which are then sent to the motor driver circuit. The raspberry pi has a picam module attached onboard which streams the live video.

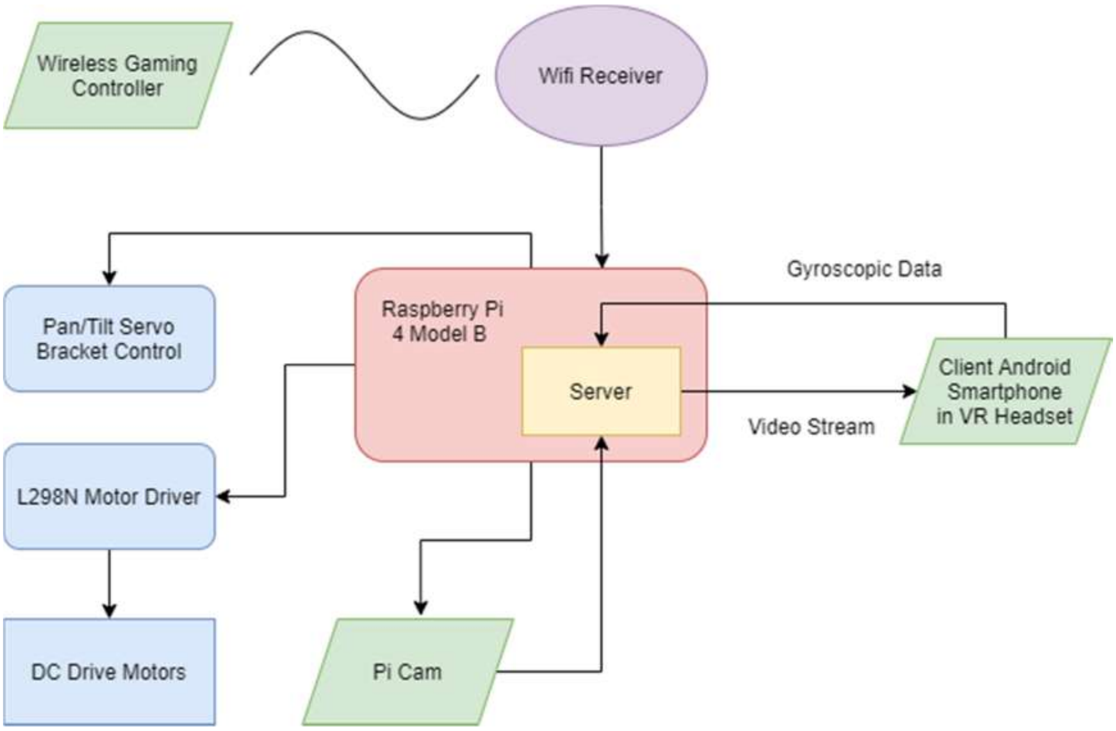


Figure 3. System Architecture of Robot

The picam has been placed on a servo bracket that allows 2 degrees of freedom. The video streamed by the picam is hosted on a server on the Raspberry Pi itself. When viewed on a handheld device, the website/android app displays the live feed in a two-windowed VR format. The gyroscopic values of the client system is also tracked by the server code. These x-y axis readings are sent back to the raspberry pi's server, where they are converted into PWM based servo commands that move the servo bracket in real time to change the camera view.

A motor driver is a small Current Amplifier that converts a low-current control signal into a higher-current signal capable of driving a motor. The L298N is a popular Motor Driver that can drive two DC motors at the same time. The module can power DC motors with voltages ranging from $5V$ to $35V$ and peak currents of up to $2A$. Connecting the DC drive motors directly with the GPIO pins of the raspberry pi is not sufficient. Most microprocessors (including raspberry pi) run at low voltages and require a small amount of current, while motors generally require higher voltages and current. As a result, the microprocessor is unable to supply enough current to the motors. This is the motor driver IC's primary function.

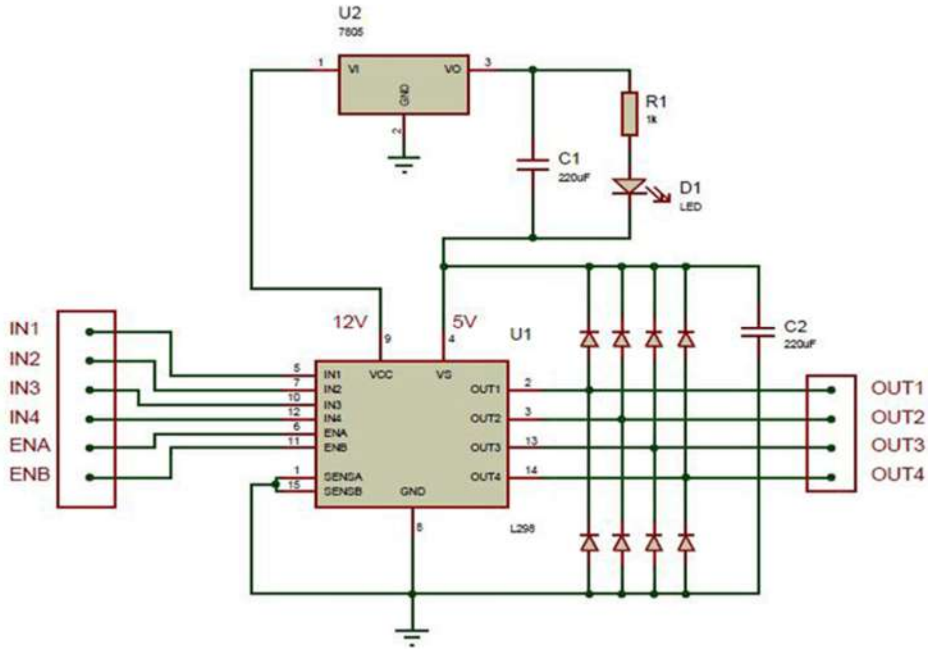


Figure 4. L298N Motor Driver Module

4.2. CODES AND STANDARDS

The IEEE Standards relevant to our project are:

- IEEE 802.11 (Wireless Networking - WiFi)
- IEEE P1003.1 (Portable Operating System Interface - POSIX)

4.2.1. Android Interface

The Android application acts as a client to the Socket Server hosted on the Raspberry Pi. The app asks for a login to provide a basic layer of security. The application collects the gyroscopic sensor data from the device and sends it to the

Raspberry Pi via Socket Server in real time. This is then used to turn the Servo motors (holding the Pi Camera) connected to the Raspberry Pi accordingly.

Consequently, the Raspberry Pi retrieves the video feed from the Pi Camera and returns it to the Android application where it is converted to a VR Video Stream. The mobile device can be put in a VR Headset and one can view the environment around the bot from its perspective.

The android device connects to the Raspberry Pi via WebSocket. The WebSocket Protocol empowers two-route correspondence between a client running code in a controlled environment to a remote host that has allowed correspondence from that code. The security model utilized for this is the origin-based security model regularly utilized by internet browsers. The protocol consists of an initial handshake followed by basic message outlining, layered over TCP [20].

The way this works is that the android device collects the gyroscope sensor calibrated data, which is in rads/sec and transmitted to the server hosted on the Raspberry Pi, every $100\,ms$. The Raspberry Pi receives the data from the server in the form of bytes. This data is then processed to obtain the motion of the android device about the x and y axes in rads/sec. As the data is transmitted every $100\,ms$, the exact degrees by which the android device has moved since the last transmission of sensor data ($100\,ms$ ago) is obtained and is fed to the servo motors to move the camera accordingly.

The delay of $100\,ms$ was considered as it was found experimentally that the processing of data as well as the time taken for the data to be transmitted was just around $100\,ms$ and this delay was short enough so that there is a fluid motion of the servo motors, shadowing the android device. Any excess data received due to the processing exceeding $100\,ms$ or more is considered too, allowing a smooth motion of the camera mounted on the servo motors. This allows the user to view the video stream smoothly and without any unnecessary delays and lagging of the servo-mounted camera system.

The video stream is sent from the picam to the android device via the same server connection, where it is simulated as a VR Stream on device. This enables the user to wear a VR headset in conjunction with the android device and experience the telepresence of the bot in real time.

This implementation involves the use of multi-processing on both, the Raspberry Pi, as well as the Android device. The Raspberry Pi has 3 processes running in parallel:

- Hosting Server: This process deals with hosting the server on the Raspberry Pi to allow the transmission of data. It interacts with the other processes by mutating the values of the shared variables (accessed by all relevant processes) according to the input values from the client via the server. All inputs and outputs to the server are dealt with by this process.
- Controlling the Servos: The servos that are present on the camera mount are controlled by this process. It takes the values from the shared variable to set the servos in the right orientation.
- Video Stream: This process merely retrieves the video stream from the picam and sends it back to the Server process via a shared variable to send to the device.

The Android device has 2 background processes. One that keeps retrieving and sending the gyroscope sensor data to the Raspberry Pi Server, and one that updates the UI and allows other tasks to work on top.

4.2.2. Web App Interface

To deliver the robot's video stream directly to the user, we also developed a webapp using JavaScript and HTML. The webapp can be accessed easily on an Android smartphone by opening the IP address of the Flask Server that is hosted on the Raspberry Pi. Once the web app is opened, it presents the user with 2 options - Connect and Calibrate.

Pressing the connect button establishes a connection with the flask server and renders the video stream coming from the onboard picam. Simultaneously the smartphone's gyroscope data is fetched and transmitted back to the server where it is processed further.

JavaScript makes use of the alpha parameter of device orientation event to get a number representing the motion of the device around the z axis. This is expressed in degrees with values ranging from 0 (inclusive) to 360 (exclusive). The value of this parameter is likely to change depending on where the smartphone is pointing when the website is opened. The calibrate button on the web app allows the user to configure the value of the alpha parameter so that there is no lateral shift when the servos are provided the angle. To avoid out of bound exceptions, the range of this variable was clipped in between $0°$ to $180°$. This makes it easier to align the onboard servos. Calibration is performed by first pointing the smartphone (in landscape) in the same direction as the robot itself then pressing the calibrate button on the webapp. Now the user can load the smartphone into the VR headset and operate the robot.

The gamma parameter of device orientation event returns the rotation of the device around the Y axis. This denotes an angle (in degrees), ranging between $-90°$ and $90°$, by which the device is tilted left or right. Since the smartphone will be placed in landscape orientation inside the VR headset, the y-axis data practically denotes the vertical angle. When the user looks straight up towards the sky, it returns $180°$. On the other hand, it returns $0°$ when the user looks down at the ground.

These sensor values encode the horizontal and vertical orientation data of the smartphone in real time. The website itself was developed using HTML. It simply reads the sensor value of the device and periodically transmits the updated data to the Flask Python script once the user presses the connect button. The sensor data can be updated hundreds of times per second but since the onboard servo motors require sufficient time to adjust their position, the JavaScript code sends the request at a set interval timeout. This avoids data overhead on the backend and allows the servos to get sufficient time to readjust to a new position.

Flask is a popular micro web framework written in Python. It is classified as a microframework because it does not require specific tools or libraries. Primarily, it is used for developing the backend of web applications. It is intended to be simple and fast to get started, with the potential to scale up to complex applications. Flask makes recommendations but does not impose any dependencies or project layout. It is up to the creator to pick the software and libraries they want to use. These features make it an excellent choice for developing the backend server of the webapp.

Pressing the connect button on the client's end sets up two simultaneous endpoints with the flask server. Endpoints are essentially identifiers that are used to determine which logical unit of the code will handle the request. It is generally only the name of a view function. One endpoint takes care of streaming the video frames from the raspberry pi to the android phone's browser(client). The other endpoint deals with the transmission of gyroscopic data from the android's browser back to the flask server. The incoming data comprises the vertical and horizontal angle of the smartphone (which is placed in landscape orientation inside the VR headset). This endpoint simply parses the incoming data and adjusts the duty cycle of PWM signals that are then conveyed to the two servo motors via GPIO pins. In return, the servos move to the appropriate angles such that the entire pan/tilt mount moves and points the camera in the correct direction.

4.2.3. Asynchronous Processes

Upon receiving a signal of a certain frequency, servo motors can rotate to any angle from $0°$ to $180°$. The $90°$ position is generally referred to as 'neutral' position because it can rotate equally in either direction from that point. In our case, the neutral position vertically and horizontally means that the picam is facing straight towards the forward direction.

A servo motor reads the information that is sent to it by using an electrical signal known as PWM which stands for "Pulse Width Modulation". This simply entails sending ON electrical signals for a set period, followed by an OFF period, hundreds of times per second. The angle at which the servo motor rotates is determined by the length of time the signal is active. For this application, the expected frequency is $50\ Hz$, or 3000 cycles per minute. Servos will set to $0°$ if given a signal of $0.5\,ms$, $90°$ when given $1.5\,ms$, and $180°$ when given $2.5\,ms$ pulses. This translates to about $2.5 - 12.5\%$ duty in a $50\,Hz$ PWM cycle.

4.3. CONSTRAINTS, ALTERNATIVES AND TRADEOFFS

One of the major constraints faced in this project was the conversion from a 2D video stream to a VR enabled stream as our video is not a 3D video stream. It is 2D but it works in conjunction with a movable camera mount to simulate a VR environment. This poses an unforeseen problem as all conventional conversion methods aim at converting 2D video streams to VR streams assuming 3D data to be considered, which is not the case in our project. We have done minor stereoscopic revisions to the stream to simulate a Virtual Environment. However, this leaves a scope for further research in this domain.

5. SCHEDULE, TASKS AND MILESTONES

For the Zeroth review held on 16 February 2021, the task was to finalize the idea and conduct some background research on the communications protocols and IoT (Internet of Things) technologies to be used.

For the First review held on 24 March 2021, the task was to establish connection between the Raspberry Pi microprocessor and the wireless dual joystick controller. This milestone was successfully achieved by using pygame module commands to establish a stable connection and send signals to the motor driver. Additionally, the hardware aspect of the robot was finished during this time.

For the Second review held on 22 April 2021, the task was to build a fully functional Android app using Java (using socket server) as well as a webapp using JavaScript and HTML (running on Flask server). This milestone was successfully achieved and integrated with the hardware for testing. Demonstrations of the robot were presented during this review.

The Figure below shows the Gantt Chart for the project timeline:

Figure 5: Gantt Chart for Work Timeline

6. PROJECT DEMONSTRATION

Figure 6. Hardware model including controller and VR Headset with Android smartphone

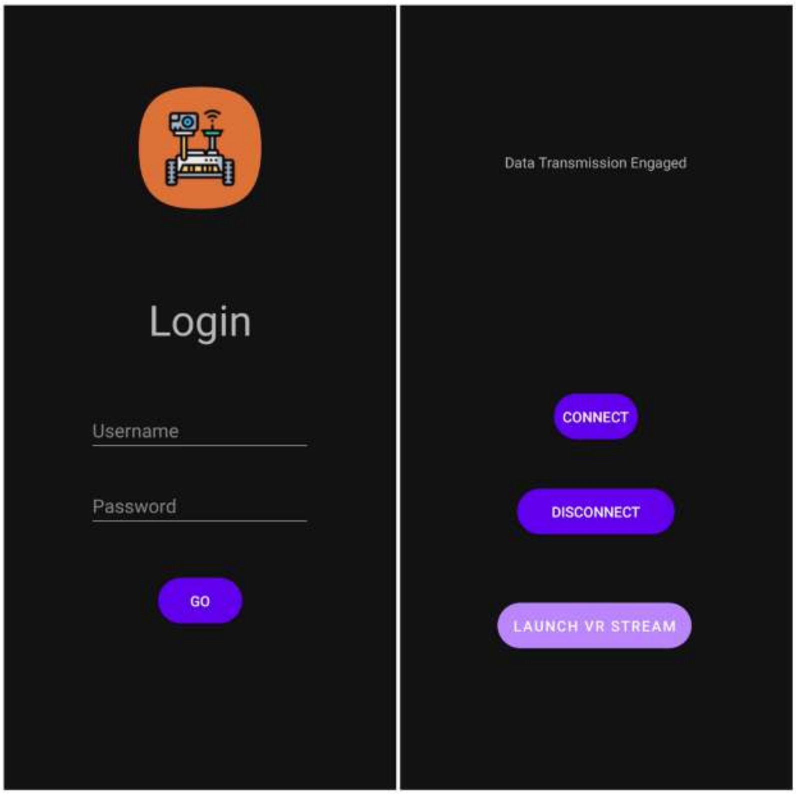


Figure 7. Android Application

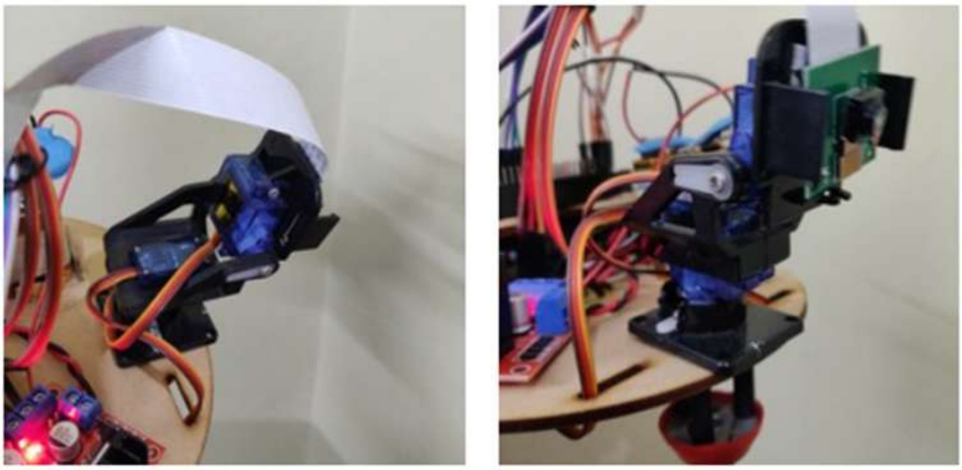


Figure 8. Pan/Tilt mount for picam with two servo motors

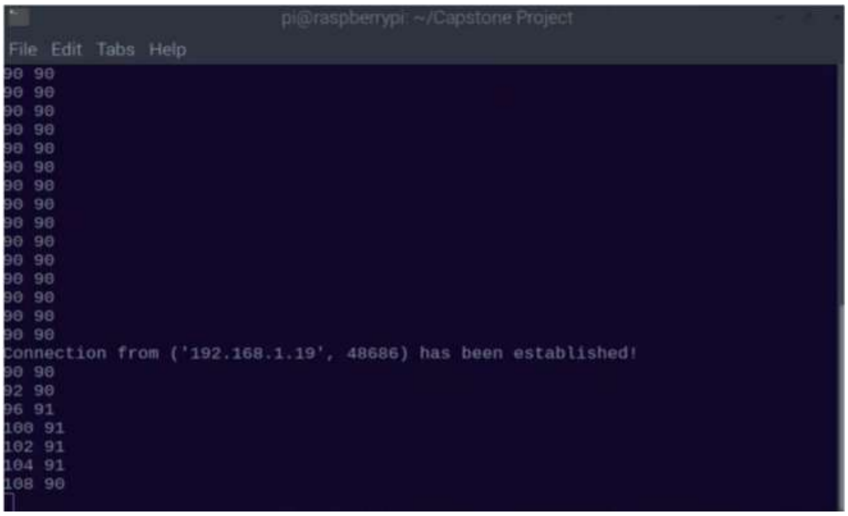


Figure 9. Output Display for Raspberry Pi

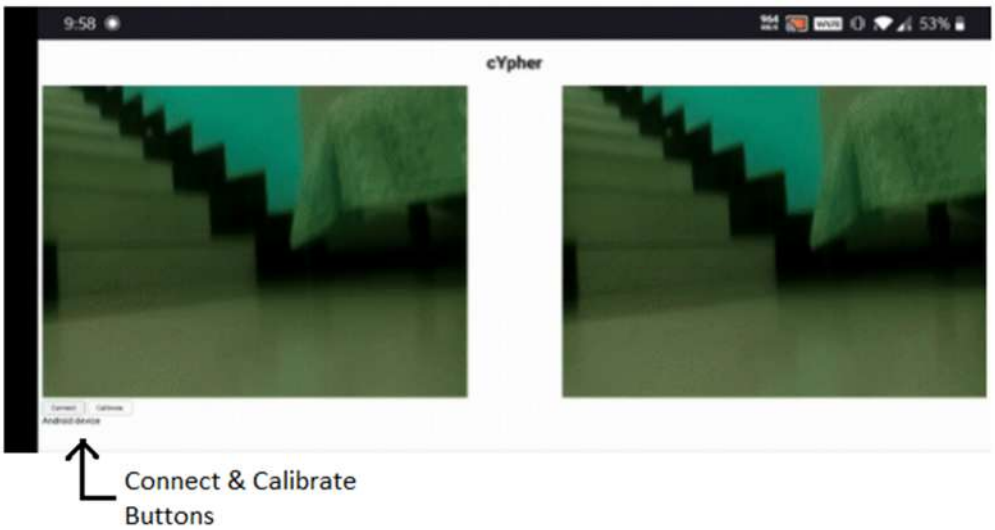


Figure 10. Application from VR Headset perspective

7. RESULTS AND DISCUSSION

7.1. List of Components used in the project:

- Raspberry Pi 4 Model B (8 GB RAM): ₹6,748.42
- Servo Motors + Stand: ₹699
- Chassis + Geared DC Motors + Wheels: ₹675
- WiFi Enabled Controller: ₹1599
- PiCam (5 Megapixels): ₹390
- SD Card (32 GB): ₹459
- Jumper Cables: ₹215
- L298N Motor Driver: ₹205
- VR Headset: ₹229
- Batteries (9V): ₹40
- Power Bank (10,000 mAh): ₹999

Total Cost: ₹12,258.42

7.2. Discussion

The objective of incorporating a versatile user interface to robotic instruments, without interfering with the primary task or the environment, drove the creation of this project. The bot developed is capable of traversing environments using the simple drive system, while being observed and controlled via the VR System as discussed in detail in this thesis.

The servo mounted camera system moves along with the head movement of the controller allowing an immersive control of the bot. The entire process happens with minimum latency which does not hinder with the user experience. Figure 7 shows the pan and tilt servo mounted camera system, that allows the camera to look around with 2 degrees of freedom. This contraption is located at the anterior of the robot.

The terminal output of the raspberry pi is shown in Figure 8. The two values shown are the orientation values (in degrees) of the two servo motors to set the camera at the appropriate angle. The output "Connection from ('192.168.1.30', 48686) has been established!" is printed by the server when the android device establishes a connection successfully. It can be observed that the orientation of the camera mount is (90,90), which is default, till the android device is connected. On connecting, the gyroscope data allows the device to manipulate the servos mounted in the camera mount as seen subsequently.

Our robot harbors a precise modus operandi for handling the back-and-forth transmission of gyroscopic and video data which been successfully implemented on hardware. It has shown promising results in our testing laboratory. Since the entire backend and link establishment over Wi-Fi network has been hosted on a single raspberry pi microprocessor, the entire hardware aspect of this project requires minimal space. The proposed paradigm makes it easier to integrate VR control interfaces into much more sophisticated robots with a minimal set of hardware components that take up nominal amounts of space onboard. The work done so far along with the demonstrations of this robot have been compiled into a GitHub repository available here [21].

8. SUMMARY

The idea of this project is to introduce and establish Virtual Reality based control as a significant approach to operating mobile robots in scenarios where human lives cannot be compromised due to hazardous circumstances; however, a human approach is deemed necessary. In this project we have implemented a VR interface which allows the user to immerse themselves in the robot's environment through the system's real-time video stream. This thesis investigates two distinct client interfaces (Android and Web Application) which were custom designed and integrated to the backend server with an intuitive calibration system.

Further work can be put into this direction of research by optimizing the speed of data transmission between the server hosted on the bot and the client at the user's end. There is also scope for further research in the domains of 2D video stream being converted to a VR video stream more optimally as well as in reducing latency and implementing more sophisticated VR technologies to work in conjunction with the bot.

References

[1] Lancet, T.: Covid-19: protecting health-care workers. Lancet (London, England) 395(10228), 922 (2020)

[2] Yang, Guang-Zhong, *et al.*, "Combating COVID-19—The role of robotics in managing public health and infectious diseases." (2020).

[3] Yang, Geng, *et al.*, "Homecare robotic systems for healthcare 4.0: visions and enabling technologies." IEEE journal of biomedical and health informatics 24.9 (2020): 2535-2549.

[4] Yang, Geng, *et al.*, "Keep healthcare workers safe: application of teleoperated robot in isolation ward for COVID-19 prevention and control." Chinese Journal of Mechanical Engineering 33.1 (2020): 1-4.

[5] Rodriguez-Gil, Luis, *et al.*, "Interactive live-streaming technologies and approaches for web-based applications." Multimedia Tools and Applications 77.6 (2018): 6471-6502.

[6] Liu, Letian, Xiaorui Zhu, and Yi Tang. "Indoor surveillance robot controlled by a smart phone." 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2013.

[7] Anandravisekar, G., *et al.*, "IOT based surveillance robot." International Journal of Engineering Research & Technology (IJERT) 7.03 (2018): 84-87.

[8] Suárez-Armas, Jonay, Pino Caballero-Gil, and Cándido Caballero-Gil. "Video surveillance robot powered by raspberry pi." Proceedings of the 1st International Conference on Internet of Things and Machine Learning. 2017.

[9] Shimosasa, Yoichi, *et al.*, "Some results of the test operation of a security service system with autonomous guard robot." 2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies. Vol. 1. IEEE, 2000.

[10] Luo, Ren C., Tung Y. Lin, and Kuo L. Su. "Multisensor based security robot system for intelligent building." Robotics and autonomous systems 57.3 (2009): 330-338.

[11] Ghute, Minal S., Kanchan P. Kamble, and Mridul Korde. "Design of military surveillance robot." 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). IEEE, 2018.

[12] Everett, Hobart R., and Douglas W. Gage. "From laboratory to warehouse: Security robots meet the real world." The International Journal of Robotics Research 18.7 (1999): 760-768.

[13] Carnegie, Dale A., *et al.*, "A human-like semi autonomous mobile security robot." 2nd International Conference on Autonomous Robots and Agents. 2004.

[14] Moon, Sung Wook, *et al.*, "Implementation of smartphone environment remote control and monitoring system for Android operating system-based robot platform." 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). IEEE, 2011.

[15] Seo, Yong-Ho, Seong-Sin Kwak, and Tae-kyu Yang. "Mobile robot control using smart phone and its performance evaluation." International Conference on Advanced Communication and Networking. Springer, Berlin, Heidelberg, 2011.

[16] Singh, Diksha, Pooja Zaware, and Anil Nandgaonkar. "Wi-Fi surveillance bot with real time audio & video streaming through Android mobile." 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, 2017.

[17] Patoliya, Jignesh, Haard Mehta, and Hitesh Patel. "Arduino controlled war field spy robot using night vision wireless camera and Android application." 2015 5th Nirma University International Conference on Engineering (NUiCONE). IEEE, 2015.

[18] Chowdhury, Shuddha, and Mahmud Rafiq. "A proposal of user friendly alive human detection robot to tackle crisis situation." 2012 12th International Conference on Control, Automation and Systems. IEEE, 2012.

[19] Gregoriades, Andreas, *et al.*, "A robotic system for home security enhancement." International Conference on Smart Homes and Health Telematics. Springer, Berlin, Heidelberg, 2010.

[20] Fette, Ian, and Alexey Melnikov. "The websocket protocol." (2011).

[21] Menon, U. R. & Hudlikar, A. cypher_droid, URL: https://github.com/7enTropy7/cypher_droid

List of publications

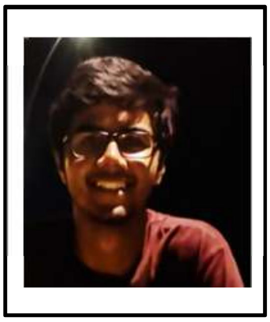
1. Communicated as a research article entitled “Design and Implementation of a Virtual Reality Interface for Teleoperated Robots” in “International Journal of Intelligent Robotics and Applications” under Springer.

International Journal of Intelligent Robotics and Applications
Design and Implementation of a Virtual Reality Interface for Teleoperated Robots
--Manuscript Draft--

Manuscript Number:	
Full Title:	Design and Implementation of a Virtual Reality Interface for Teleoperated Robots
Article Type:	Regular Paper
Manuscript Classifications:	200.040: Exploration and Security Robotics; 200.050: Healthcare, Medical robotics and Assistive; 300.010: Novel and Emerging Applications
Funding Information:	
Abstract:	Numerous operations and tasks are too perilous for humans to perform in person. These risks may arise due to various reasons: alien environments, radioactive conditions, unmapped terrain and so on. More recently, with the Covid-19 pandemic, a new hazard of social contact has come into play. In such cases, it is safer and much more viable to have a robot that can be teleoperated from a remote location to avoid human presence. This task necessitates the establishment of an immersive telepresence that allows the user to react to situations more effectively as well as provides a better understanding of the environment. This paper presents a thorough examination of the technology and design for such a robot. The pro-posed design employs a virtual reality controller developed for smartphones that allows the user to immerse into the robot's environment through the it's real-time video stream. The prototype developed during this research is capable of traversing environments and returning relevant data to the user; a technology that can easily be adapted to different types of robots with varying degrees of sophistication.
Corresponding Author:	Mahalakshmi P, Ph.D Vellore Institute of Technology: VIT University VELLORE, TAMIL NADU INDIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Vellore Institute of Technology: VIT University
Corresponding Author's Secondary Institution:	
First Author:	Unnikrishnan Menon, (B.Tech.,)
First Author Secondary Information:	
Order of Authors:	Unnikrishnan Menon, (B.Tech.,)
	Atharva Hudlikar, (B.Tech.,)
	Mahalakshmi P, Ph.D
Order of Authors Secondary Information:	
Author Comments:	Respected Sir, This is a real time work carried out under my supervision. Would like to receive your valuable feedback on this work. Thanks.
Suggested Reviewers:	Dr. VPS Naidu Vadlamudi, Ph.D Scientist, NAL: National Aerospace Laboratories CSIR vpsnaidu@gmail.com Expert

2. Unnikrishnan Menon, Anirudh Rajiv Menon and Atharva Hudlikar, “A Novel Chaotic System for Text Encryption Optimized with Genetic Algorithm” International Journal of Advanced Computer Science and Applications(IJACSA), 11(10), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0111005
3. Unnikrishnan Menon, Atharva Hudlikar, and Divyani Panda. "Scytale--An Evolutionary Cryptosystem." *arXiv preprint arXiv:2008.05290* (2020).

Curriculum vitae



Name	:	Unnikrishnan R. Menon
Father's name	:	Rajendran Menon
Date of Birth	:	04 November 1998
Nationality	:	Indian
Sex	:	Male
Company placed	:	Matelabs Innovations Pvt. Ltd.
Permanent Address	:	D-304, SuperTech Ecociti, Sector-137, Noida, Uttar Pradesh 201035
Phone Number	:	+91-8376048185
Mobile	:	+91-8376048185
Email Id	:	unnikrishnanr.menon2017@vitstudent.ac.in

CGPA: 8.46

Examinations taken:

1. GRE: N/A
2. TOEFL: N/A
3. CAT: N/A

Placement Details:

External Offer

Curriculum vitae

Name	:	Atharva Hudlikar
Father’s name	:	Rajendra Hudlikar
Date of Birth	:	03 April 1999
Nationality	:	Indian
Sex	:	Male
Company placed	:	Tata Consultancy Services
Permanent Address	:	904, Mahavir Ornate, Sec-11, Koparkhairane, Navi Mumbai - 400709
Phone Number	:	+91-9003358600
Mobile	:	+91-9167244778
Email Id	:	rajendra.hudlikar2017@vitstudent.ac.in

CGPA: 8.65

Examinations taken:

1. GRE: N/A
2. TOEFL: N/A
3. CAT: N/A

Placement Details: TCS, Placed

Capstone Project

Project Title	VR Surveillance Bot
Team Members (Names and Reg Nos.)	Unnikrishnan Menon (17BEE0047) Atharva Hudlikar (17BEE0100)
Faculty Guide	Dr. P. Mahalakshmi
Semester / Year	VIII / IV year
Project Abstract (not more than 200 words)	Numerous operations and tasks are too perilous for humans to perform in person. These risks may arise due to various reasons: alien environments, radioactive conditions, unmapped terrain and so on. More recently, with the Covid-19 pandemic, a new hazard of social contact has come into play. In such cases, it is safer and much more viable to have a robot that can be teleoperated from a remote location to avoid human presence. This task necessitates the establishment of an immersive telepresence that allows the user to react to situations more effectively as well as provides a better understanding of the environment. This thesis presents a thorough examination of the technology and design for such a robot. The proposed design employs a virtual reality controller developed for smartphones that allows the user to immerse into the robot's environment through its real-time video stream. The prototype developed during this research can traverse environments and returning relevant data to the user; a technology that can easily be adapted to different types of robots with varying degrees of sophistication.
Project Title	**VR Surveillance Bot**
List **codes** and **standards** that significantly affect your project. **(Must)**	1. IEEE 802.11 (Wireless Networking - WiFi) 2. IEEE P1003.1 (Portable Operating System Interface - POSIX) 3. TCP/IP (websockets) 4. Flask web framework 5. Android OS 4.4 and above compatible. 6. Firefox browser for flask client.

List at least two significant **realistic design constraints** that are applied to your project. **(Must)**	1. The refresh rate of the VR stream is limited by the RAM specifications of the Raspberry Pi. 2. In case of poor network connection, full duplex data transmission on the flask endpoints might get significantly affected and cause delays. 3. The number of parallel processes running on the Server were limited by the number of threads present on the Raspberry Pi.
Briefly explain two **significant trade-offs** considered in your design, including options considered and the solution chosen **(Must)**	1. The server backend was hosted publicly on the internet instead of private hosting due to cost and resource limitations. We considered other frameworks such as CherryPy and Tornado. However, websocket and flask were cheaper, computationally lighter on hardware and easier to work with. 2. We had also considered using an Nvidia Jetson Nano as the onboard microprocessor since it has a GPU available with extensive multithreading capabilities. However, since connecting a Jetson nano to a network requires the use of extra hardware, we instead went ahead with a Raspberry Pi 4 Model B.
Describe the **computing aspects, if any,** of your project. Specifically identifying **hardware-software** trade-offs, interfaces, and/or interactions	1. A time delay was introduced in the transmission of gyroscopic data from the Android client to the onboard flask server. The servo motors inside the pan/tilt camera mount require a few milliseconds to adjust to the appropriate angle. Experimentally, this delay was set to 100 ms. 2. The raspberry pi implements multiprocessing to enable balance between seamless transmission of data as well as driving the hardware components connected to it (such as servos and DC motors). 3. The conversion from a 2D video stream to a VR enabled stream as our video is not a 3D video stream. It is 2D but it works in conjunction with a movable camera mount to simulate a VR environment. We have done minor stereoscopic revisions to the stream to simulate a Virtual Environment.