Robotic Teleoperation Using a Virtual Reality Interface

By Zak Graham Rackham UP899802

School of Computing

Final Year Engineering Project PJE40

February 8, 2023

Abstract

With the expansion of robotics into new fields yields new applications, the problem of human robot interaction has become increasingly significant. Robots face complex unpredictable manipulation tasks in hazardous environments. Teleoperation approaches show potential in assisting the operator to complete complex tasks that require human decision making and intuition. Achieving humanoid actuation of robotic manipulators is a challenge, creating an autonomous robot to achieve a dynamic set of operations becomes convoluted very quickly. Teleoperation implemented in virtual reality presents a natural paradigm for controlling a mobile robot remotely. This paper details the design of a mobile robot and the implementation of a distributed system to teleoperate the robot using virtual reality. This system converts motion data from the headset and controllers to joint states and velocity commands to remote control a collocated robot called Mai.

Consent to share

I consent for this project to be archived by the University Library and potentially used as an example project for future students.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms

API Application Programming Interface. 4, 60, 71, 72

CMOS Complementary Metal Oxide Semiconductor. 8

DC Direct Current. 8, 38, 43, 49, 53, 69, 79, 92, 98

DoF Degree of Freedom. 38

FoV Field of View. 10

GPIO General Purpose Input Output. 13, 20, 59, 79

HMD Head Mounted Display. 4, 75

HTTP Hyper Text Transfer Protocol. 44

IMU Inertia measurement unit. 98

IoT Internet of Things. 97

Mai Mobile Arm Interaction. 97

MCU Micro Controller Unit. 91, 96

PWM Pulse Width Modulation. 13, 38, 43, 68–70, 82

ROS Robot Operating System. ix, 5, 7–9, 11, 60, 70, 72, 73, 75–78, 91, 92, 96

RPM Revolutions Per Minute. 55, 68, 72, 82, 98

SBC Single Board Computer. 11, 13

SDK Software Development Kit. 4, 60, 71

TCP Transmission Control Protocol. 45

UI User Interface. 97

USB Universal Serial Bus. 99

VR Virtual Reality. 4, 5, 9, 10, 41, 60, 70, 71, 73, 75, 85, 86, 92, 98

WAN Wide Area Network. 45

XML Extensible Markup Language. ix, 9, 77

Chapter 1

Introduction

1.1 Overview

Traditional programming methods in robotics try to solve sets of manipulation tasks based on sensory information to achieve an understanding of an environment and to apply an understanding of these environments to reach a specified goal. Within these scenarios there are assumptions that the robot is to follow a set method of control in a constant and unchanging environment or scenario. If conditions are subject to unforeseeable change the system will need consistent development and changes to adapt. In a dynamic environment development becomes a problem, with rapidly changing requirements the development life cycle cannot match demand. Recently developed robot learning methods are an intersection of machine learning and robotics these methods are still limited and as of yet cannot provide a complete solution. Everyday tasks, while seemingly simple, contain many variations and complexities that pose insurmountable challenges for today's machine learning algorithms. Due to these challenges human intuition remains valuable and depending on the application vital to achieve a range of objectives.

This report details the design, development, implementation, testing, and evaluation of a wireless remote controlled mobile robot and a VR application to remotely operate the robot. To enable humans to operate by proxy in a remote location. The functions of the system and design of both hardware and software are to be derived from specified requirements that remain static.

1.2 Project Goal

To improve domestic life new areas of technology have emerged, internet of things devices are more prominent in the home environment than ever, each is specialised and there are ways to integrate various IoT devices from different developers so that they coordinate to create effective automation. Amazon's cloud web services and Alexa frameworks are a great example of this, However whilst automation and IoT systems are great there is yet to be a device that allows remote presence that would also allow the user to complete a greater verity of physical tasks that can be more dynamic and to integrate this system into various home environments effectively. This leads directly to my project goal to create a dynamic remote presence system that allows users to complete a verity of physical tasks remotely.

1.3 Project Aim

This project aims to achieve teleoperation of a mobile robot over a local area network using ROS a Oculus VR system and the Unity game engine and development environment. To achieve a set of domestic pick and place tasks; picking up and moving an object, feeding a pet and anything else the user would like to achieve (Hardware constrained).

1.4 Objectives

Research and design

- Structured PID

- Comprehensive research
- Answer literature review question
- Requirements elicitation
- Hardware design
- Software design

Development

- Hardware development
- 3D printing stack
- Motor control
- Robot kinematics
- Virtual reality interface development

Testing

- Robot/Hardware testing
- Unit testing
- System testing
- User testing
- Evaluation and improvements

1.5 Constraints

Constraints on the project are considered and presented to review their respective affects in order to mitigate potential problems within development.

Cost

- The projects direction is heavily influenced by costs.
- Costs need to be kept low due to limited self-funding.
- Unforeseen costs may become a problem if hardware is damaged or broken.
- Impact of Covid-19 Pandemic, increasing prices, limited logistics.

Time

- Time needs to be managed effectively.
- This project runs concurrently with various modules.
- Time needs to be allocated according to estimated development time for each component, estimations are based on research and analysis of the researchers individual ability and skills.

Experience

- The researcher has broad experience with various computing devices, computer software, hardware and firmware.
- The researcher is an undergraduate student.
- Due to the researchers lack of experience time needs to be considered against experience to ensure each phase of development is successful and on schedule.
- The researcher is not proficient in every sub-discipline within the project, lots of time will be allocated to learning and personal development to ensure research is scoped appropriately and justified.

1.6 Contribution

In order to design and implement a system that achieves the project goal i will make use of existing software solutions APIs and SDKs and existing hardware solutions for VR. The Oclous rift s will be used as the HMD along with the Unity game engine for the VR environment running within Windows 11. ROS will be implemented within Ubuntu server on the raspberry pi. The web-socket protocol will be implemented for communication between the desktop environment and the raspberry pi.

Chapter 2

Literature Review & Research

2.1 Introduction to review

This rapid review aims to answer the question of how can robotics be integrated effectively into the home environment? And more specifically how can we use modern robotics to aid our domestic life?

2.2 Robotic systems

Robotics is the intersection of three core disciplines, computer science, electronics and mechanics each representing the brain, nervous system and body respectively. A typical industrial robotic manipulator is composed of a combination of links, linear structural body elements and joints motor enabled rotational axes. During the last 45 years, robotics research has been aimed at finding solutions to the technical necessities of applied robotics. New trends in robotics research focus more on human robot interaction (Garcia et al., 2007). This observation has been substantiated by (Doelling et al., 2014). Service robots have evolved with technology, enabling robots to operate in a less structured environment and interact more naturally with humans. Doelling et al. (2014) conducted a survey to identify service robots commercially available today for in home use. The majority of commercially available robots they identified for in home use were entertainment, toys, and cleaning robots. Cleaning robots are identified as the most numerous and have a high adoption rate with the market growing 60% each year (Doelling et al., 2014). This shows that consumers are interested in home service robot solutions. Lipton et al. (2018) believe the future of manufacturing is to require a combination of robots and humans this further emphasises the importance of human-robot interaction.

2.3 Software frameworks

Software is key in enabling robotic systems to function. There are many paradigms for applying software within robotics. The most developed being ROS (Robot Operating System), ROS is defined by its creators as a meta operating system (Robotics, 2018), similar to an operating system implemented as a middle ware, a type of software that acts as an intermediary and can operate within a distributed system. Due to the proprietary nature of robotics most manufacturers of robot hardware also provide their own software, this makes evaluating the current software approaches in industry very challenging and results in the lack of standardization of programming methods for robots, this leaves a gap in the industry for better standardisation that could lead to improved clarity and faster development.

2.4 Sensory technology

For a robot to understand its environment sensory technology has to be applied. Lipton et al. (2018) Details a robotic system in which a stereo camera is used to provide visual feedback to the user, a stereoscopic camera is used here as depth needs to be considered for the VR application. Smoot

(2022) Compares currently available sensory technology each were identified to have their respective advantages and disadvantages. Evaluating what sensory technology to use would depend on the specifics of the application. When considering a 3D environment stereoscopy is key in achieving creating and enhancing the illusion of depth in an image by means of stereopsis for binocular vision. The Kinect RGB-D sensor contains a monochrome CMOS sensor that captures image depth using computation by calculating the distortion of a known infrared light pattern which is projected into the scene. Whilst the underlying technology is different, both approaches achieve the illusion of depth capture.

2.5 Actuation approaches

For a robot to move in the physical world actuation approaches need to be applied. Electro-mechanical is identified as being the most appropriate approach. Software again becomes fundamental in the control of robotic actuators by applying the appropriate mathematics, kinematics a branch of geometry and sub-field of physics, developed in classical mechanics that describes the motion of points, bodies and systems of bodies without considering the forces that cause them to move. Jerk a sub-field of physics concerned with an objects acceleration in respect to time is another considerable factor they are both key to achieve smooth motion. Servo motors, stepper motors and standard brushless DC motors are all viable for robotic applications, servo and stepper motors are best used due to the ability to hold a static position to remain stationary.

2.6 Teleoperation methods

Teleoperation (or remote operation) is a method of controlling a system or machine from a distance. Reachy is an open source robotics project by pollen-robotics (2020) Reachy has been implemented within ROS using the python programming language. ROS provides a universal robot description format abbreviated URDF based on XML that is used to define the kinematic structure of a robot. The team developing reachy also implement gRPC an open source remote procedure call system developed by google as a server to send data from the ROS system to a client. The client can be used to program the robot using the custom SDK and to integrate the software with Unity. Unity is a game engine where VR applications for reachy can also be developed.

VR provides a compelling interface for robots as it allows fluid interactions in the real world, Whitney et al. (2018) detail a "robocentric" model where the human and robot both share the same virtual space they use both ROS and Unity connected via the ROS bridge WebSocket connection. WebSocket is located at layer seven in the OSI model and depends on TCP at layer four.

2.7 Human-robot interaction

Human robot interaction is often explored in research questions often considering the robot and human to be separate entities. Lipton et al. (2018) embed the user in a VR control room to teleoperate a collocated robot. Teleoperation systems map objects between the user's space U to the robot's space R (Lipton et al., 2018). This presents a new paradigm for human-robot interaction where the robot and human are better interconnected allowing for precise intuitive control.

2.8 Virtual Reality

Virtual reality is a technology that allows the immersion of a user in a multi-sensory computer generated three dimensional virtual representation of real or fictional environments. VR has many applications in the real world the most notable being gaming, virtual conferencing and training. VR provides an interactive graphical user interface supplemented by non visual modalities such as audio and haptics. Angelov et al. (2020) examine the classification of VR head mounted displays, two types are identified tethered and standalone the former referring to head mounted displays that require a computer to operate and the latter where computation is integrated within the headset. There is also an intersection of the two that can operate with or without an external computer.

The screen-door artifact is identified as a main problem with current VR hardware it refers to how the user sees a grid that outlines the contours around the pixels, this distracts from the sense of immersion in the environment. This artefact is typical of multimedia displays in general but in this instance it's much more critical to the users experience, increases in display resolution and pixel density help to mitigate this effect. Field of view is another important parameter for optimal immersion, the human's average binocular FoV reaches up to 190° (Angelov et al., 2020). All consumer level VR head mounted displays are yet to achieve full FoV immersion yet partial solutions do exist at research level. StarVR One is an example of this with a 210° FoV (StarVR-Corp, 2020).

A key component of any modern VR set is its tracking system, tracking refers to the process of determining users' viewpoint position and orientation there are multiple methods for tracking. Angelov et al. (2020) state that the Oculus Rift S has one key difference, an additional AI algorithm "Insight" is utilized this algorithm is used in the case when the user controllers are out of the sensor visibility range. This algorithm makes an informed assumption on where the controllers are located. Inside out tracking refers to the use of computer vision within a camera array placed at strategic positions across the HMD to gain a 3D representation of the users external environment and to sense the location of controllers through IR emitters whereas lighthouse refers to the tracking using constellations of infrared LED's built into the HMD and controllers and lighthouse sensors external to the headset.

2.9 Conclusion

It is evident form this review of literature that robotics can be integrated into the home environment, it is also apparent that there already exist applications for intercommunication between the identified Unity game engine and ROS. Existing systems that place the user within the robots perspective have been considered to identify their respective findings to influence the development of the system detailed in the continuation of this paper.

2.10 Hardware research

2.10.1 Single board computer comparison

To evaluate the most suitable SBC a comparison of various single board computers is presented below. Alternatively any computing device could be used although due to size constraints on the project a SBC would be more feasible.

ID	Board	Specifications	Suitability
SBC0	Odroid-C4	ARM-Cortex S905X3, Quad-core Cortex-A55 cluster, Mali-G31 GPU, 4 GB DDR4 RAM, HDMI 2.0, 4x USB 3.0, micro-USB, IR receiver, UART, 47-pin GPIO	Not suitable no on-board WIFI (Wifi could be added using a serial connection this is however less feasible)
SBC1	Rock Pi 4 Model C	Rockchip RK3399 (2x 1.4Ghz Cortex-A72, 4x 1.4GHz Cortex A53), ARM, 64-bit, Mali T860MP4, 4 GB LPDDR4-3200, 2.4/5 GHz Wi-Fi, Bluetooth 5.0, Gigabit Ethernet, Micro HDMI 2.0, USB 3.0 OTG, USB 3.0 host, 2x USB 2.0 host, 3.5mm audio/mic, DSI, CSI, RTC, 40-pin GPIO	Yes

Table 2.1: Single board computer comparison

ID	Board	Specifications	Suitability
SBC2	Raspberry Pi 4 Model B	Broadcom BCM2711B0, (4× 1.5GHz Cortex-A72), ARM, 32-bit, Broadcom VideoCore VI, Memory: 2, 4, 8 GB DDR4, Dual-band 802.11ac/b/g/n Wi-Fi, Bluetooth 5.0, Bluetooth Low Energy, Gigabit Ethernet I/O: 2× micro-HDMI 2.0, 2× USB 3.0, 2× USB 2.0, 3.5mm audio/video, CSI, DSI, 40-pin GPIO	Yes, with all specifications considered this was identified as the most appropriate board to use.

Table 2.2: Single board computer comparison continuation

SBC2 is identified as the most appropriate board and has been selected due to its built-in ac Wi-Fi, GPIO with PWM and more than sufficient RAM at 4GB and microprocessor speed at 1.8 GHz.

2.10.2 Raspberry Pi

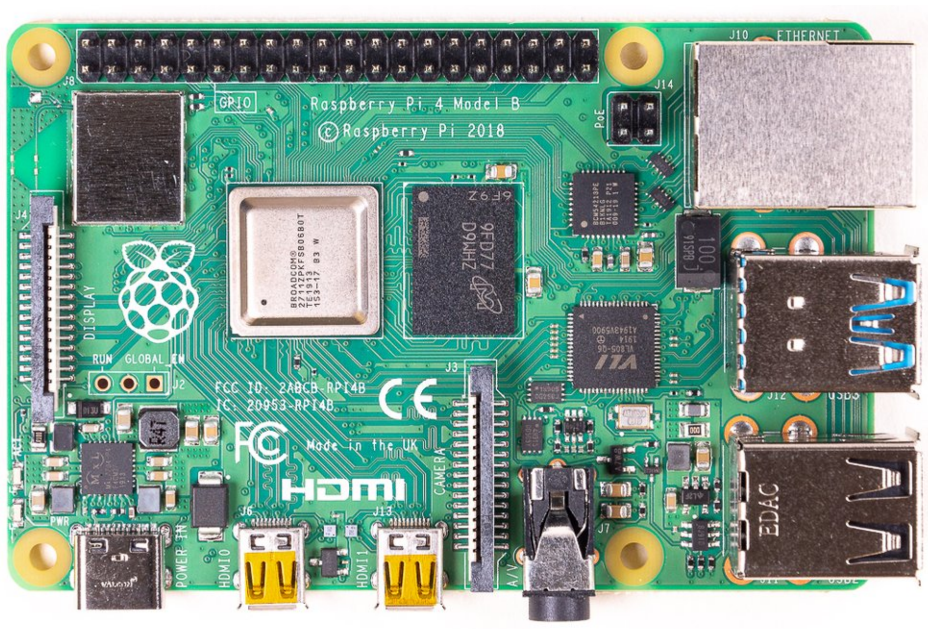


Figure 2.1: Raspberry Pi 4b

The raspberry pi is a popular single board computer, this board was selected due to it meeting the requirements of the project all specifications considered. Furthermore the Raspberry Pi Foundation is a UK-based charity that works to put the power of computing and digital making into the hands of people all over the world through education. There is consequently a vast community of coders and hardware engineers working with this small platform for a verity of different projects. This means more comprehensive documentation and support that may aid the project. The raspberry pi can run many different operating systems including but not limited to windows 10/11, the raspberry pi OS (formerly Raspbian) and various Linux distributions.

2.10.3 Arduino

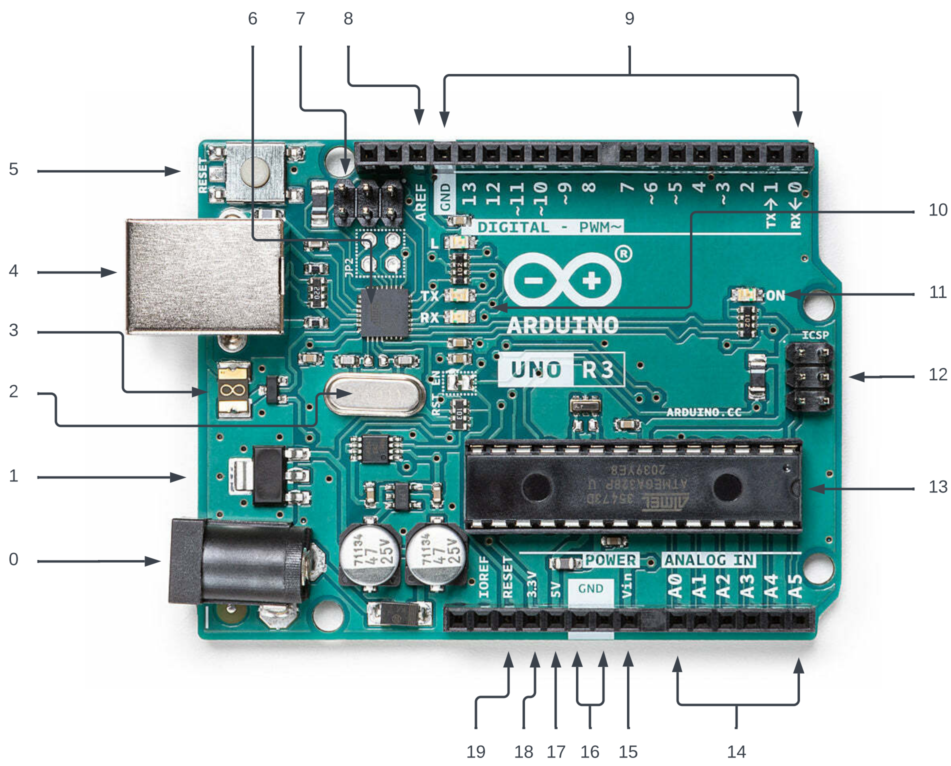


Figure 2.2: Arduino uno rev3

Arduino is an Italian organisation who designs, manufactures, and supports electronic devices and software, the platform was created for education and therefore has comprehensive documentation from various sources across the web. More specifically the uno meaning one in Italian, is a single board microcontroller, a single integrated circuit (IC) that is typically used for a specific application and designed to implement certain tasks. The Arduino Uno R3 is an open source hardware computing platform it integrates the ATmega328 microcontroller and the ATmega16u2 as an onboard USB to serial converter(“Arduino Uno R3”, 2019).

ID	Component	Specification	Discription
C0	Power DC	2.1mm inner diameter, 5.5mm outer diameter with Positive pin to 12V DC, Recommended 9v	Provides power to the board.
C1	Voltage Regulator	AMS1117 5.0V SMD	Stabilises DC Voltage.
C2	Crystal Oscillator	16.000H9H	16 MHz, used to calculate time.
C3	Polyfuse	Unknown	A resettable fuse or polymeric positive temperature coefficient device is a passive electronic component used to protect against overcurrent faults in electronic circuits.
C4	USB-B 2.0	Speed: 480 Mbps	Serial bus uses an onboard ATmega16U2 to connect the serial TX and RX pins on the ATmega 328.

Table 2.3: Arduino board components

ID	Component	Specification	Description
C5	Arduino Reset	test	Reset's the arduino, starts the program from the start.
C6	Atmega 16u2	speed: 16MHz	bridge between the computer's USB port and the main processor's serial port.
C7	ICSP Header	6 pin	in-circuit serial programming, post component installation programming for USB bridge.
C8	AREF	test	Analog Reference used to set an external reference voltage as the upper limit for the analog input pins.

Table 2.4: Arduino board components continuation

ID	Component	Specification	Discription
C9	Digital I/O	6 provide PWM	pins labeled ~ can be used to generate PWM.
C10	TX and RX	LED's	Display when serial port is active, TX (transmit) and RX (receive).
C11	Power LED	n/a	Indicator for power to board.
C12	MCU(ICSP)	6-pin	in-circuit serial programming for ATmega328 MCU.
C13	ATmega328 microcontroller	20MHz, 32Kb Flash, I/O: 23	Low power, RISC architecture.
C14	Analog I/O	6-pin A0 – A5	Used to provide analog input in the range of 0-5V.
C15	Vin	n/a	Alternative power input voltage to Arduino when using an external power source.
C16	Ground	n/a	Auxiliary ground for external devices.

Table 2.5: Arduino board components continuation

ID	Component	Specification	Discription
C17	Auxiliary I/O	5v	5v power for auxiliary devices
C18	Auxiliary I/O	3.3v	3.3v power for auxiliary devices
C19	Auxiliary reset	n/a	LOW will reset the ATMega328 micro controller, used to integrate with shields.

Table 2.6: Arduino board components continuation

2.11 Motors and actuation

DC Motor considerable specifications

- Voltage
- Speed
- Torque
- Reduction Ratio
- Current
- Encoding

2.11.1 Linear actuation

Linear actuation is considered due to the identification of a need for a lift function for the mobile robot.

Linear actuation can be achieved by the translation of a rotational axis via a lead screw and a nut where the screw thread is in direct contact with the nut thread. They are typically not used to transmit high power, but rather for intermittent use in low power drive and position mechanisms.

2.11.2 Motor drivers

In order to move the wheels a motor driver is required. The driver is connected to a power supply, single board computer via GPIO and motor.

There exist many drivers that are suitable for this project the L298n driver has been identified as appropriate due to its simplicity to implement and low cost.

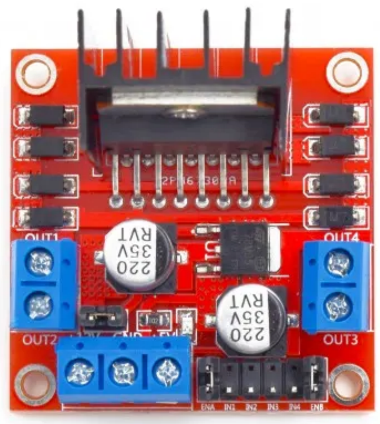


Figure 2.3: L298n motor driver

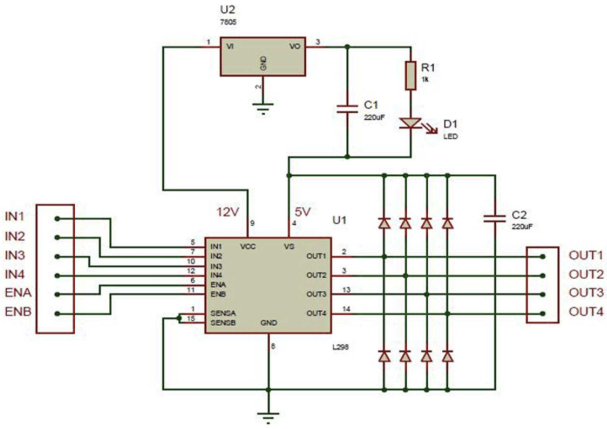


Figure 2.4: L298n internal circuit schematic (Components101, 2021).

The internal circuit of the L298n driver is presented. This integrated circuit utilises a H-bridge that allows a DC motor to be driven in a clockwise or anti-clockwise direction. Two motors can be driven in either direction simultaneously, this is essential for the forward kinematic structure of the robot base.

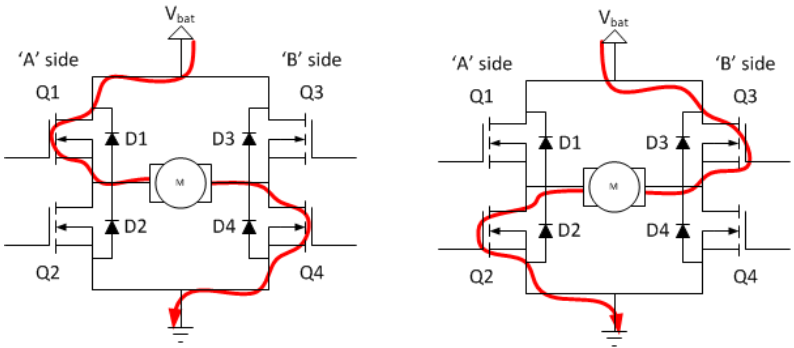


Figure 2.5: H-bridge explanation

The H-bridge allows current to be inverted to drive the DC motors in either direction.

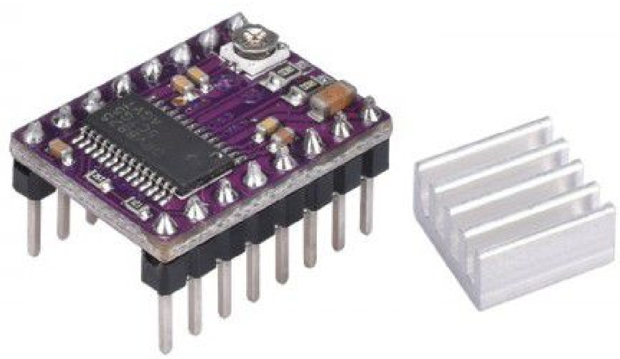


Figure 2.6: DRV8825 stepper motor driver

The DRV8825 is a micro stepping bipolar stepper motor driver that features adjustable current limiting, over-current and over-temperature protection, and six microstep resolutions. It operates from 8.2V to 45V and can deliver up to approximately 1.5A per phase without a heat sink or forced air flow Pololu-Corporation (2022).

Chapter 3

Methodology

3.1 Comparison of formal approaches

Software development methodologies provide a framework for planning, executing, and managing the process of developing software systems (Vijayasarathy & Butler, 2016). Although formal approaches have been developed, sometimes the choice of methodology may be based on marketing and literature bias that supports new or industry-supported practices shown to produce desirable results rather than applying a more specific or custom methodology that may be more appropriate. To avoid this bias only academic literature will be considered in the evaluation of approaches. The most traditional method is the waterfall model, this model was shown by Vijayasarathy and Butler (2016) to be the most common amongst methodologies from the survey sample see figure 3.1. Companies with high employee counts were the dominant group for traditional approaches this further substantiates the identification of a bias for industry supported practices.

Agile approaches are the second most used, companies with high employee counts were the dominant group for traditional approaches, whereas companies with low employee counts were dominant for agile and iterative approaches (Vijayasarathy & Butler, 2016).

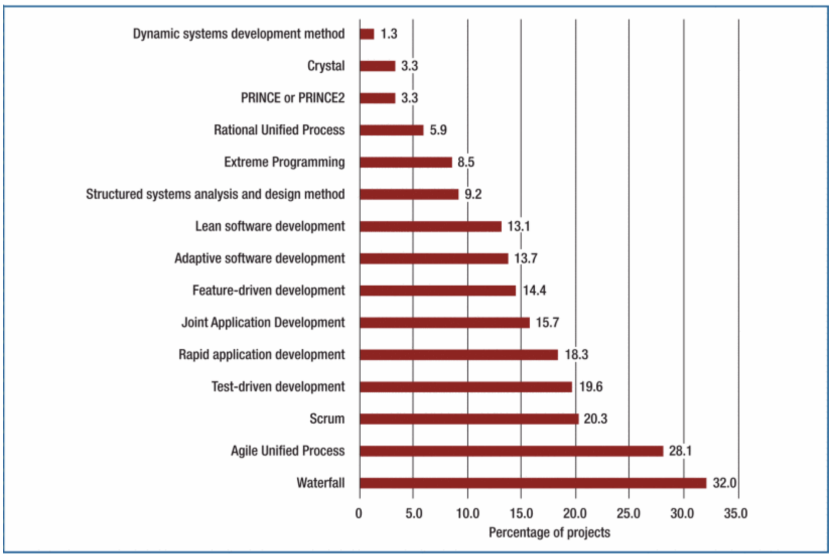


Figure 3.1: Software development methodologies used, survey sample by (Vijayasarathy & Butler, 2016).

To evaluate what method would be the most appropriate for this project a comparison between the waterfall and agile approaches is conducted.

Attribute	**Agile**	**Waterfall**
Scope	Many small projects	Complete one single project
Division	Separates into sprints	Divides into phases
Approach	Incremental & iterative	Linear & sequential
Focus	Customer satisfaction	Successful project delivery
Requirements	Prepared everyday	Prepared once at start
Requirements change	Any time	Avoids changes
Testing	Concurrent	After build
Project manager	No	Yes

Table 3.1: Comparison table of methodology attributes

3.1.1 Benefits of waterfall

- Straightforward planning and designing due to the agreement on deliverables at the start of the project.
- Better design with a whole system approach.
- Defined scope of work.
- Easier costing.
- Clear measurements of progress.
- Defined team roles.
- Dedicated resources can work in parallel for their specific tasks.

3.1.2 Disadvantages of waterfall

- Rigid structure to allow necessary changes.
- No allowance for uncertainty.
- Limited customer engagement, resulting in poor satisfaction.
- Sequential approach is not ideal for a large-sized project where the end result is too far in the future.
- Testing is done only at the latter phases of the project.

3.1.3 Benefits of agile

- Faster software development life cycle.
- Predictable schedule in sprints.
- Customer-focused approach, resulting in increased customer satisfaction.
- Flexible in accepting changes.

- Empowers teams to manage projects.
- Promotes efficient communications.
- Ideal for projects with non-fixed funding.

3.1.4 Disadvantages of agile

- Agile requires a high degree of customer involvement, which not all customers are comfortable with or prefer to give.
- Agile assumes every project team member is completely dedicated, without which weakens the principle of self-management.
- A time-boxed approach may not be enough to accommodate each deliverable, which will require changes in priority and additional sprints that can bring up cost.
- Agile recommends co-location for efficient communication, which is not always possible.

3.2 Chosen approach

The agile methodology is the most appropriate choice for this project. The key driving factor for this decision is that due to the project complexity an agile approach will offer the best flexibility in accepting changes, its faster life cycle is appropriate in this instance due to the limited development time, agile also relies on the principle of self-management that is key to this projects success. Within this project the researcher will also act as the product owner and therefore will review each iteration concurrently.

Sinha and Das (2021) conclude from their case study 4 major points that further substantiate my findings on agile methodology as follows; Divide and conquer, changes always welcome, Time and cost estimation and finally customer satisfaction. It may be appropriate to make adjustments and al-

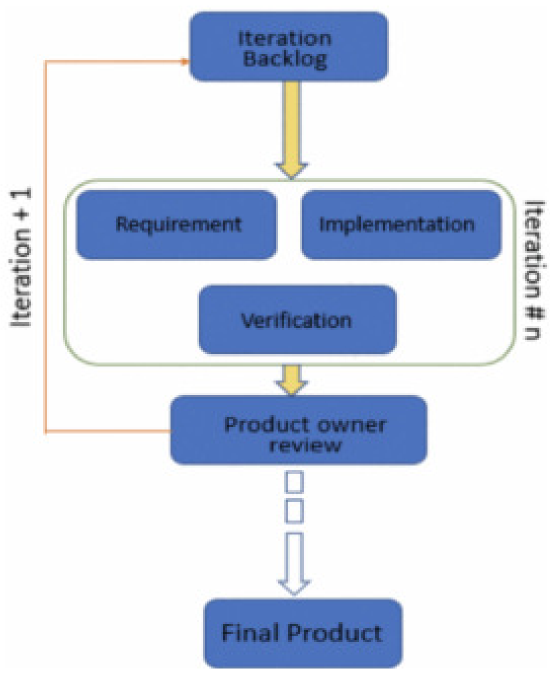


Figure 3.2: Agile illustration by (Sinha & Das, 2021).

terations to the formal definition of agile to better suit the projects specific requirements during its life cycle.

Each iteration will build on the previous the iteration review and retrospective will help to evaluate what areas can be improved. What worked well what didn't and what will the focus be within the next iteration. As this project incorporates both hardware and software solutions each will be reviewed independently, hardware is to be designed using the same agile methodology using a modular chassis for iterative hardware development.

Chapter 4

Requirements

4.1 Pre-Elicitation

It is important to consider the project goals and objectives before the requirements elicitation refer to chapter 1 for the statement of goals and objectives. Iqbal et al. (2020) discuss pre-elicitation in global software development projects below are the most appropriate points scoped in relation to this project.

- Domain understanding
- Organisational understanding
- Project contextual analysis
- Elicitation session preparation

4.1.1 Domain Understanding

Domain understanding is the process of compiling knowledge of a specific, specialized discipline or field to develop a better understanding of the domain to better focus elicitation procedure.

The domain of this project falls within system integration since various software and hardware products will be integrated together to create the complete system this also substantially reduces the development time needed for the project this is important due to the limited time constraints and large scope of the project.

4.1.2 Organisational Understanding

This project will be organised using various management principles, the project is managed over its life-cycle using a time based gantt chart to track progress for different set objectives. Both soft and hard deadlines are used to organise a development timescale and to prevent the overlap of tasks, if a soft deadline is missed focus is shifted to the time critical task. Within requirements elicitation organisation is managed using a table to track the various steps within the process.

4.1.3 Contextual Analysis

Contextual analysis describes the circumstances in which an event occurs, in this context the internal and external environment of the project are considered and the effect of this environment on the project is presented to develop appropriate strategies to achieve the project aim. Mai is to be implemented within a home (domestic) environment. Requirements must consider this environment.

The bulk of hardware development is completed in a home environment with software development being independent to the hardware location yet dependant on the hardware being powered. Hardware will be developed without access to the desirable tools and workshop environment due to the limited funding for the project and considerations and impact of the covid-19 pandemic. Due to this lack of resources the hardware needs to be designed with this considerable in mind to ensure a smooth development process and not to impact other aspects of development.

4.1.4 Elicitation Preparation

Preparation for elicitation is completed by first clarifying the scope and setting appropriate boundaries to focus more directly on the most relevant information. Users will be surveyed to gain appropriate insight on what feature's users are most interested in and to gain further insight to the appropriate control methods, range of appropriate tasks and concerns of the general public.

4.2 Elicitation

Requirements are crucial when considering engineering any system or prototype. There are many approaches to requirement gathering. Requirement gathering is completed in this project by the method of surveying the targeted user base and specific market segment due to the user being the primary target within this project the elicitation process will focus closely on the user. Technical requirements were gathered though the study of academic literature and appropriate web articles from hardware and software suppliers including open source documentation.

4.2.1 Survey

The survey was disseminated internally within the school of computing, externally to the general public though social media posts and word of mouth using email, instant messaging and physical modalities. The survey was intended to quantify users opinions on such as system to achieve a more comprehensive analysis on how to scope requirements.

4.2.2 Questions

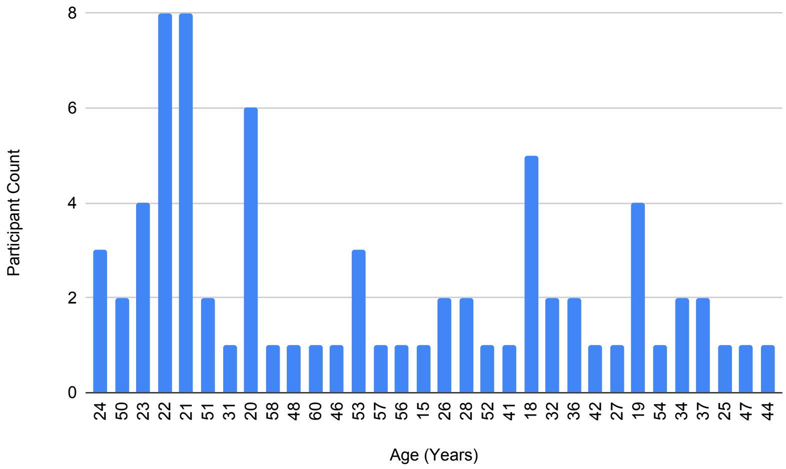


Figure 4.1: Q1

The age of the participant is an important measurable variable in discerning whether the survey was successful in engaging participants from a variety of ages. Since the project targets a domestic environment interaction people from all ages can be assumed. Therefore it is important that a variety of ages are represented within the survey to make results representative of the artefacts user base and to influence requirements accordingly.

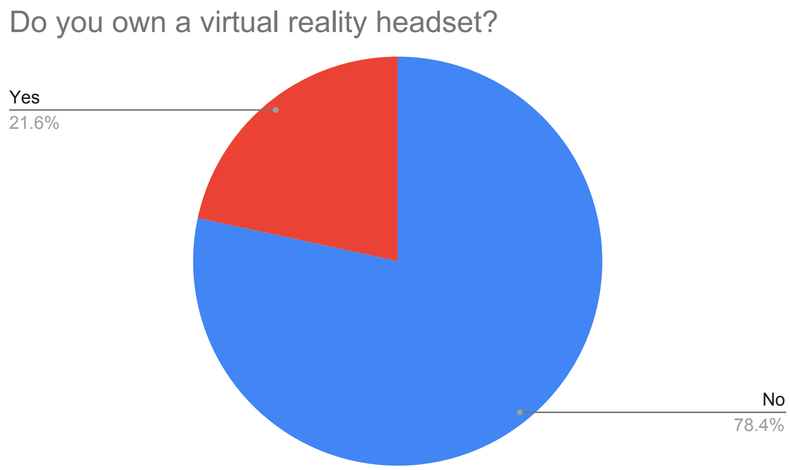


Figure 4.2: Q2

The question of ownership of VR is considered to generate a metric to show how many users could implement the software used within and written for this project.

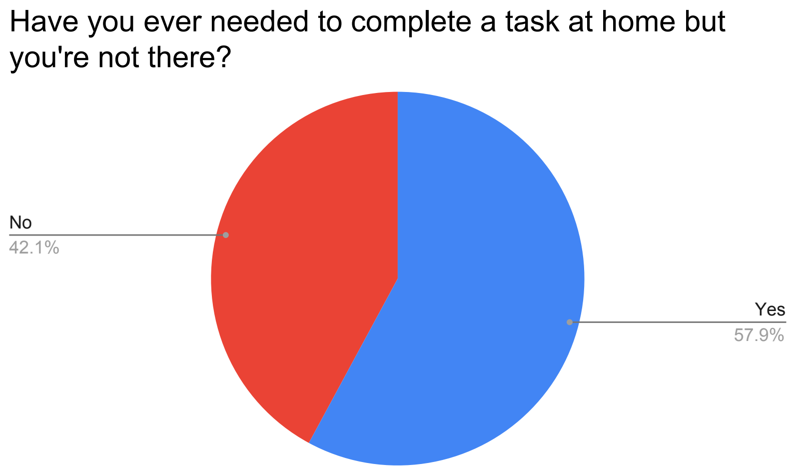


Figure 4.3: Q3

A substantial considerable is to gather data on whether users would be interested in completing domestic tasks remotely. This was asked to gather an understanding for the potential market for such a device.

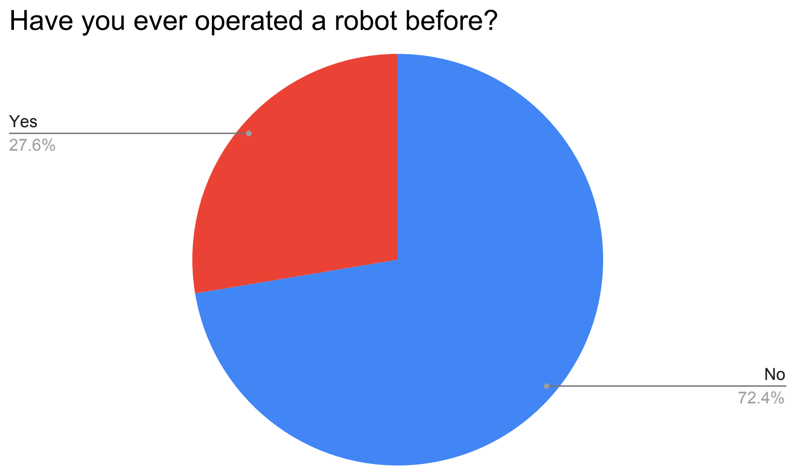


Figure 4.4: Q4

Participants were asked if they had ever operated a robot before 27.6% answered yes this is much higher than first anticipated. This further substantiates findings within the literature review in an increased interaction with robots.

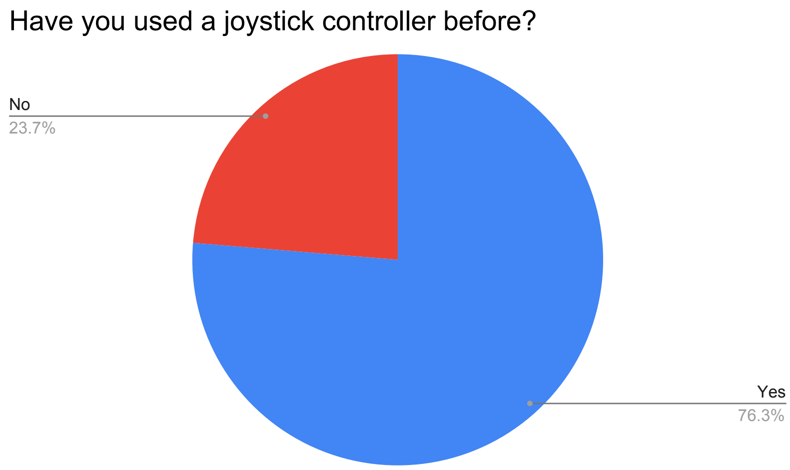


Figure 4.5: Q5

Participants were asked if they have used a joystick controller before to understand if this control method would be familiar.

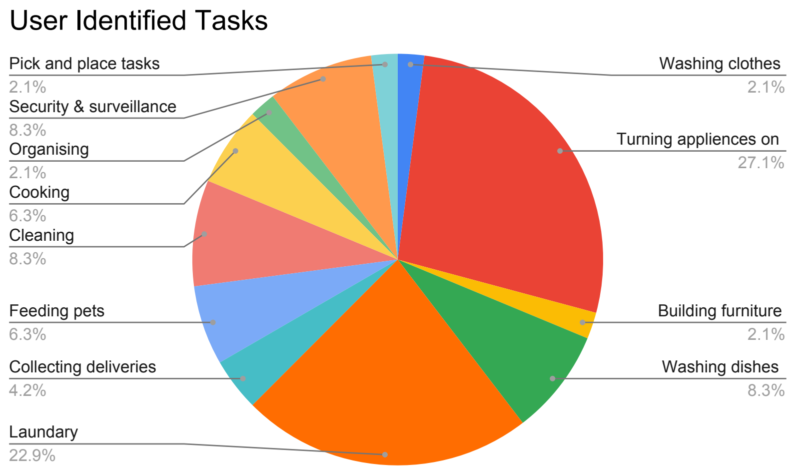


Figure 4.6: User identified tasks

Participants were asked to identify tasks that they would be interested in completing to scope requirements accordingly.

4.3 Non-functional (user requirements)

ID	Requirement	Description	Priority
NFR0	User interface	The UI is to be implemented in VR using the touch controllers.	yes
NFR1	Security	The artefact must Validate and authenticate users, so no unauthorised individual can operate the robot.	no
NFR2	Reliability	Mai must be be reliable in operation and function as the user expects.	yes
NFR3	Latency	Latency is critical in this project latency must be minimised where possible at each component of the system.	yes
NFR4	Portable/Mobile	The complete artefact must be mobile in terms of hardware.	yes
NFR5	Battery life	The battery life of the complete artefact must be substantial enough to provide a reasonable amount of operation time.	no

Table 4.1: Non-Functional requirements

ID	Requirement	Description	Priority
NFR6	Localisation	The artefact must fit within the context of the local environment (Home environment).	no
NFR7	Maintainability	The artefact must be maintainable and adjustable in terms of software architecture hardware needs to be easily replaceable.	yes
NFR8	Usability	The artefact must be easily usable for laymen.	Yes

Table 4.2: Non-Functional requirements continuation

4.4 Functional Requirements(technical requirements)

ID	Requirement	Description	Priority
FR0	DC Motor & stepper control via LAN	Robotic actuators need to be controlled over a wireless network.	Yes
FR1	Servo control via LAN	Robotic actuators need to be controlled over a wireless network with low latency.	Yes
FR2	Camera stream	The stream must be clear with low latency	Yes
FR3	Motion control	Motion control implemented within VR must translate real world arm movement into mirrored actuation of the robotic manipulator.	Yes
FR4	Portable/Mobile	The complete artefact must be mobile in terms of hardware and portable in software as a package.	Yes
FR5	Bidirectional LAN communication	Bidirectional communications are needed to send data from the raspberry pi to the PC hosting the VR environment.	Yes

Table 4.3: Functional requirements

ID	Requirement	Description	Priority
FR6	PWM	PWM is required for DC motor and Stepper motor control, 8 pins are required by the DC motor drivers 2 for each direction.	Yes
FR7	DC Power supply	As a portable power solution a rechargeable 12v battery is required to supply power to each component of the system, a 10Ah LiFePo4 battery will be used.	Yes
FR8	Wiring	Wiring at various gauges and lengths will be used to supply power to each component, higher gauges are used where load is highest.	Yes

Table 4.4: Functional requirements extended

4.5 Cost estimation

The overall monetary cost of the project is considered before the design phase. The budget for this project is £400. This is to be split into £300 working budget and a £100 reserve budget in anticipation for errors in design, damage and hardware faults. The estimated cost of £400 is based on the current market price considering the most expensive components first, the five DoF robotic arm, battery and raspberry pi at £200, £50 and £50 respectively.

The researcher does not consider the cost of more trivial components at this stage.

Chapter 5

Design

5.1 Design constraints

The design of the mobile robot has been greatly influenced by constraints. The robot is to be designed as a prototype therefore the robot must be saleable and modular to accept changes in design as the project progresses.

Considerable Factors

- Environment.
- Resources, tools and equipment.
- Logistics.
- Availability.
- Networking hardware and software configuration.
- Budget.

5.2 System Design

The system has been designed using auto desk fusion 360 software for hardware abstraction. The design is approached with the agile methodology that focuses more on adaptive design rather than being predictive. Agile has the flexibly respond to the changing needs within a fixed time and cost agile uses an iterative design process instead of the linear and successive design process.

5.3 VR

Theoretically any VR system will meet the requirements for this project as long as it has a tracking system capable of tracking head and hand movement accurate to 5cm, the more accurate the tracking the more accurate the movement due to the direct approach where tracking position from the headset coordinate system in meters is mapped to a position defined in millimeters within the cartesian coordinate system of the robotic arm relative to the center of the arms base. Monica R (2022) state from their findings a data set that produces the average tracking error of 1.83cm for the Oculus rift s.

Figure 5.1: VR system Rift S

The Oculus rift s and touch controllers will be used due to usability considerations for the control of the mobile robot. The touch controllers include analogue triggers located on the side of each controller, this makes it possible to allow the user to actuate the gripper with precise control from 10° (open) to 73° (closed).

5.3.1 User Interface

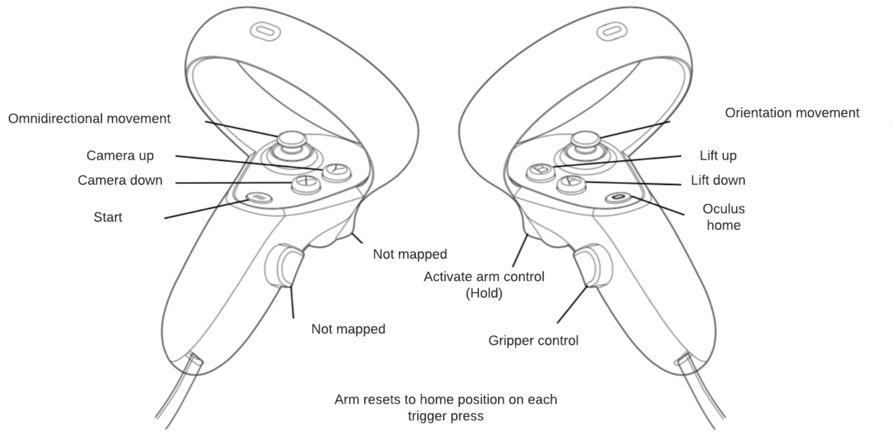


Figure 5.2: Controller mapping

The user interface was designed to be similar to gaming applications that are available to the every day consumer. Two analogue input joysticks are used, the x-y position from the left joystick is used to control the omni-directional movement of the base and the right to control angular movement in the Z axis for the orientation of the mobile base. For omni-directional control whatever direction the user points the thumb stick is translated to the velocity of the robot in the x-y plane.

5.4 System architecture

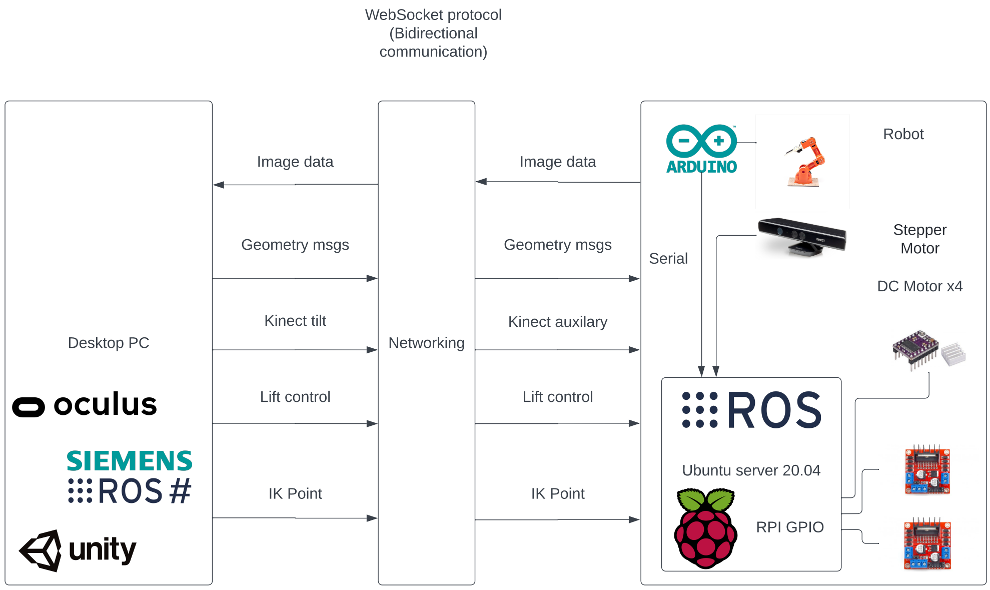


Figure 5.3: High level design

The proposed architecture of the system is shown in figure 5.3. The desktop PC is running windows 11, the raspberry pi is running Ubuntu server in headless operation for remote programming. The desktop PC is connected to an AC power source, the robots raspberry pi is powered using an 12v DC battery. The Braccio arm servos are connected to the Arduino motor shield this shield connects to the Arduino to allow processing via the ATmega328 micro controller and to allow serial communication to external devices. The Arduino and Kinect are connected to the raspberry pi via two separate USB ports on the pi. Each L298n motor driver is connected to four PWM I/O pins on the pi, the pi is also grounded to the negative power distribution block. Each motor driver is connected to a 12v supply. Power is supplied to each component at the required voltage using DC-DC converters.

5.5 Networking

5.5.1 WebSocket protocol

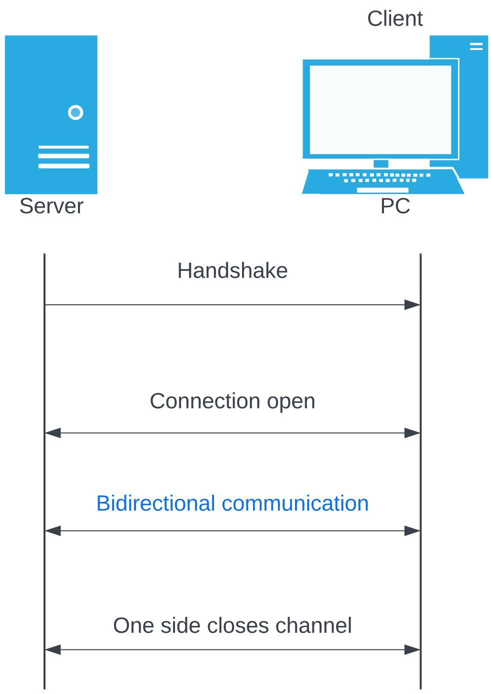


Figure 5.4: WebSocket Protocol

WebSocket is a protocol for full-duplex communication over a single TCP connection, illustrated in figure 5.4, HTTP is used at the application layer Hong et al. (2019) state that WebSocket makes the exchange of data between clients and servers simpler.

- Web client (Unity) sends a link request to the WebSocket server.
- WebSocket server parses the request and generates a response to the client.
- A bidirectional persistent connection is established between the client and server.

- Both the client and server can simultaneously transmit data. If there is no new data packets, probes will be sent to keep the connection open.
- Either the client or server can close the conection.
- The protocol server remains active on port 8080, the client can re-establish a conection at any time.

5.5.2 Port forwarding

Port forwarding can be used to connect to the robot from a WAN. Port Forwarding is a popular concept that is used over networks specifically for network address translation. This means that the Unity application is still able to connect to the robot on a separate network. By setting a specific port for the robot, the router is configured to always accept requests for those ports and forward data to a device's private IP address. Since WebSocket protocol is being used here port forwarding needs to use TCP.

5.6 Hardware design

5.6.1 Chassis

To better accommodate requirement changes the chassis of Mai has been designed to be modular and salable. 2020 V-slot Aluminium extrusion was identified as a building block for enabling changes. T-nuts can be used to affix hardware to any of the 4 faces of the extrusion this enables fast prototyping that is able to accept changes.

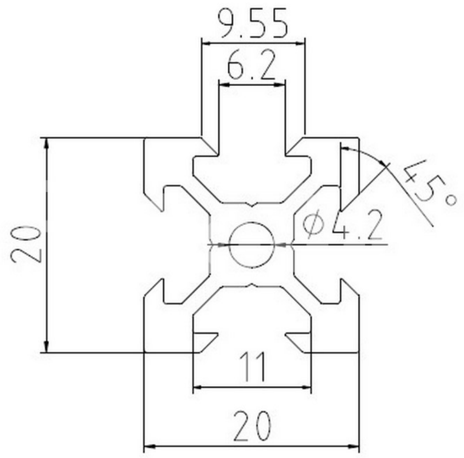


Figure 5.5: V-slot profile

This extrusion has been converted into a chassis of the robot and to implement a lead screw based lift platform for the robots' arm to move in the Z-axis.

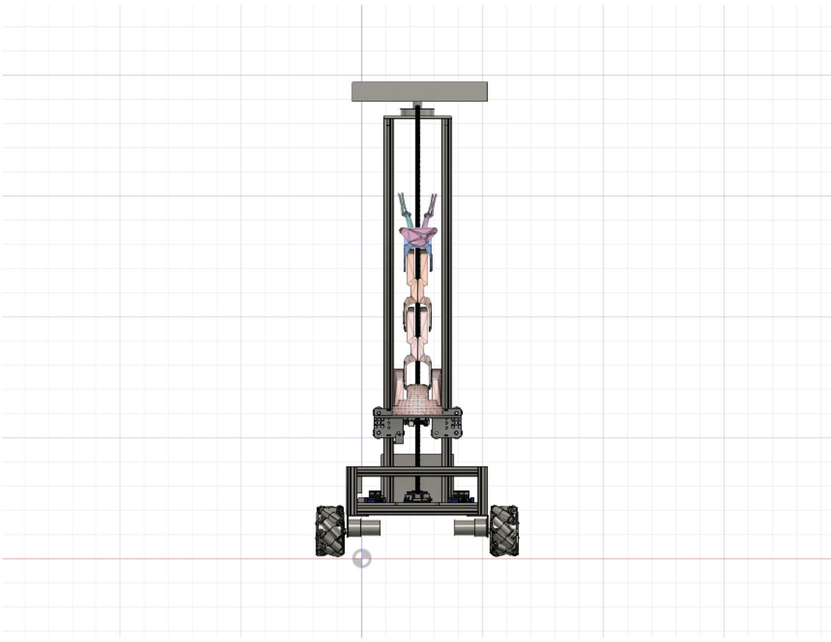


Figure 5.6: Robot schematic front view

The hardware schematic was modeled within fusion 360 to provide a virtual 3D representation of the hardware design before building the prototype in the implementation phase.

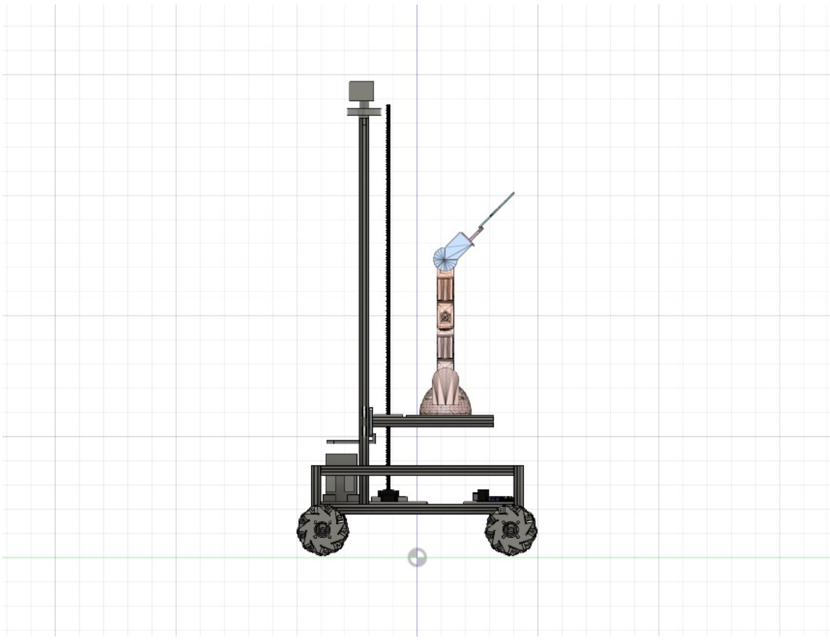


Figure 5.7: Robot schematic left view

5.6.2 Robotic arm

The arm used in this project is the Arduino braccio, the arm has 5 degrees of freedom. The included Braccio shield allows the control of servos directly from the Arduino board.

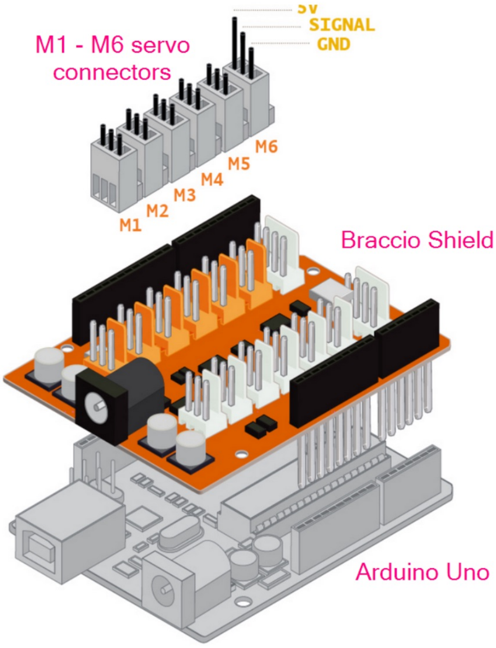


Figure 5.8: Motor shield

5.6.3 Battery

In order to meet the design requirements of being mobile the robot needs a portable power solution. A 12V 10Ah battery is to be used to supply power to all components of the system. DC to DC converters are used step down the voltage for components requiring less than 12V within the system. The battery used has its own built-in battery management system.

Figure 5.9: LiFePO4 Battery

The battery is to be connected to a DC terminal that supplies power to DC to DC converters to step down voltage where required. Each component is connected in parallel.

Power Consumption

- Battery output 120W.
- 4 × DC motor 0.5A 25W.
- Arduino Braccio robotic arm 5A maximum consumption 25W.
- Raspberry Pi 4B max consumption 7.28W at a worst case synthetic load (Bate, 2019).
- Bipolar stepper motor 1.5A per phase, 36W constant.
- Total maximum power consumption: 93.28W.

The complete system is therefore capable of operating at max load for approximately one hour. Practical battery life will be much higher since each component will not draw its maximum amperage continuously except the stepper motor as phases consume constant power when energised to hold the lift.

5.7 3D Printing

In order to rapid prototype Mai 3D Printing was identified as the most appropriate method. By creating custom 3D models, structural hardware components can be designed with relative ease.

5.7.1 Hardware

The 3D printer used was the Creality Ender 3 V2 with polylactic acid filament. It is made from renewable products, it is also a very easy material to work with, The only drawback is that the PLA filament has a higher viscosity that may lead to a clogged nozzle. PLA is biodegradable and can degrade rapidly when subjected to industrial composting. However under atmospheric conditions degradation is slowed dramatically to up to 80 years. Considering the environment designs should only be printed when complete and final to avoid excess waste.

5.7.2 Software

The Cura slicer was used to "slice" 3D models into layers that are converted to Geometry Code that is interpreted by the printer to actuate motors so the print head follows a specified path. Marlin is open source firmware primarily used by 3D-printers using the Arduino platform and can be customised. Firmware used to print the final models was not altered and used the default parameters for the Ender 3 due to satisfactory print quality.

5.7.3 Final 3D Printed Components

The best way to design the structural hardware of the system was to use 3D printing. This was identified as a quick way of rapid prototyping by being able to add modules to the robot though the use of 3D printed parts. Cable management was completed using custom 3D printed clips. By the method of 3D Printing changes in design can be implemented with relative ease.

Figure 5.8 shows the custom model used to mount the Kinect to the aluminum extrusion.

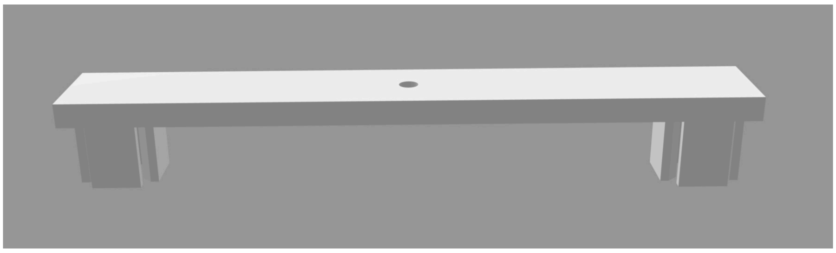


Figure 5.10: Kinect mount custom 3D model

Figure 5.9 shows the custom model used to mount the battery to the chassis.

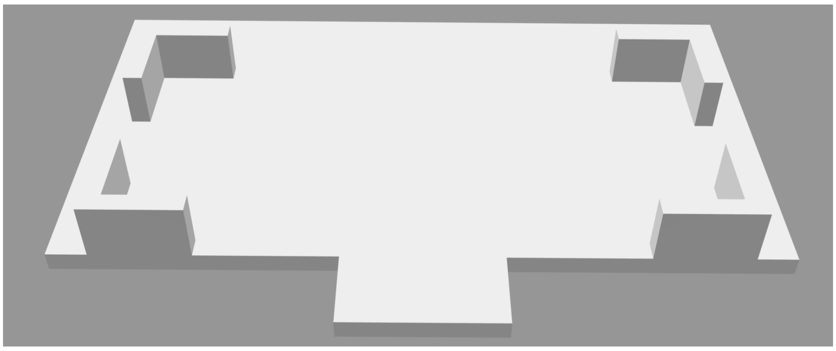


Figure 5.11: Battery mount custom 3D model

Simple rectangular planes were used to affix the motor drivers and raspberry pi to the chassis.

5.8 Actuation

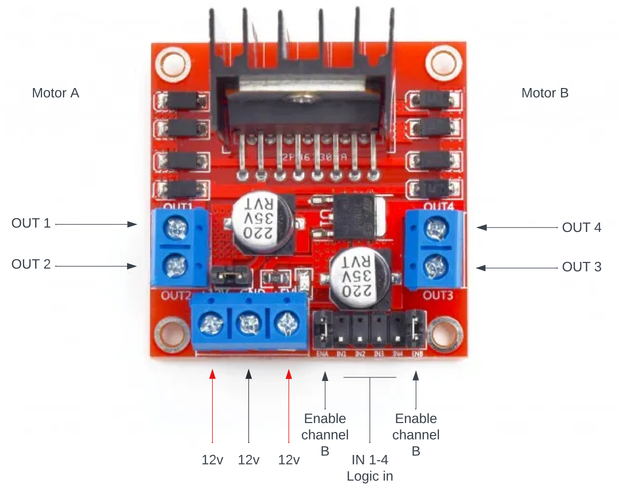


Figure 5.12: L298n motor driver

A DC motor is connected to OUT1 & OUT2 and a second for OUT3 & OUT4, the left motor driver drives the front left wheel and the rear left wheel, whilst the right is connected to the front and rear right wheels.

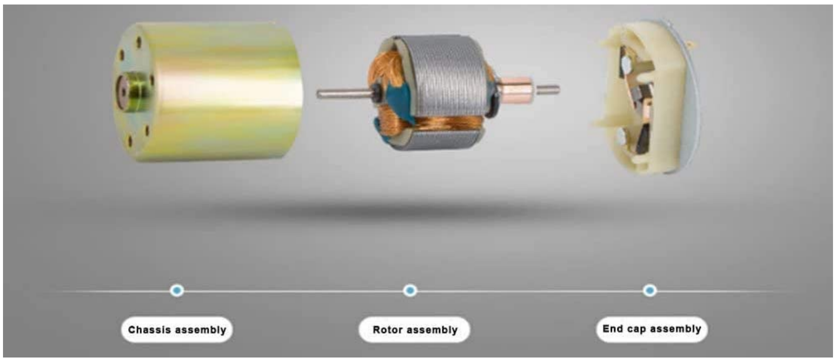


Figure 5.13: DC Motor

Nominal DC Motor Specifications

- Rated Voltage: 12V
- Speed: 200RPM

- Rated Torque: 2.2Kg.cm
- Reduction Ratio: 1:24
- Rated Current: 0.5A
- Encoding: None

Due to financial constraints on the project price was very influential when selecting motors for actuation, the motor detailed in figure 5.13 is a standard brushless motor. Tourqe and speed are the most considerable specifications. Tourqe needs to be considered to overcome the load from the robot hardware to move the omni-direcitonal base. Speed needs to be considered to actuate the robot to speeds that are appropriate for the application. Speed needs to mirror the capabilities of human movement within a domestic environment.

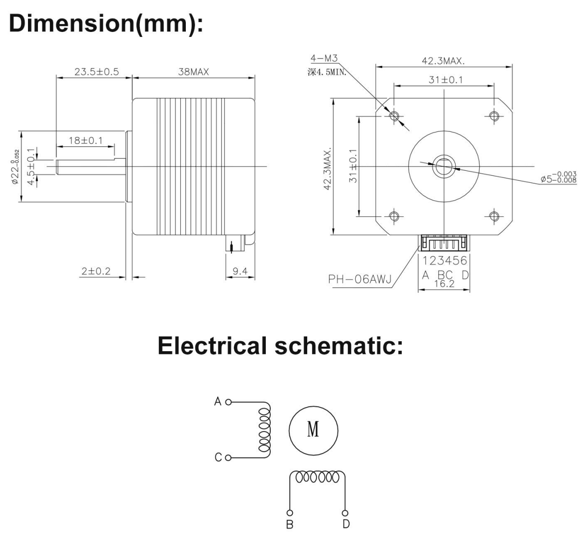


Figure 5.14: Bipolar stepper motor

A stepper motor is a brushless DC motor that divides a full rotation into several equal steps. The motor position can then be directed to move and hold at one of these steps without any position sensor for feedback. Due to these properties a stepper motor would be viable to be used as an actuator for the implementation of a z axis lift.

Nominal Stepper Motor Specifications

- Rated Voltage: 12V.
- Step Angle: 1.8 deg.
- Current per phase: 1.5A.
- Model 17HS4401.

Figure 5.15: Mecanum Wheels

To ensure that the robot can meet the design requirements its total speed must be calculated to be within parameters of human walking speed in a domestic environment. The average human walking speed is between 3-4 miles per hour. Therefore the robot must match or be very close to this speed.

$$V = r \times d \times \pi \times m \div c \tag{5.1}$$

Where:

- V is the robots speed in miles per hour.

- r denotes the wheel RPM.
- d denotes the wheel diameter.
- π is the circle constant, representing the ratio between the wheels' perimeter and its diameter.
- m denotes a constant 60, the number of minutes in an hour.
- c is the number of inches from d in a mile.

From equation 5.1 it can be found that the theoretical linear speed in the forward direction is 2.343 MPH. Whilst this is slightly lower than the defined average human walking speed it will be sufficient enough to provide a usable experience.

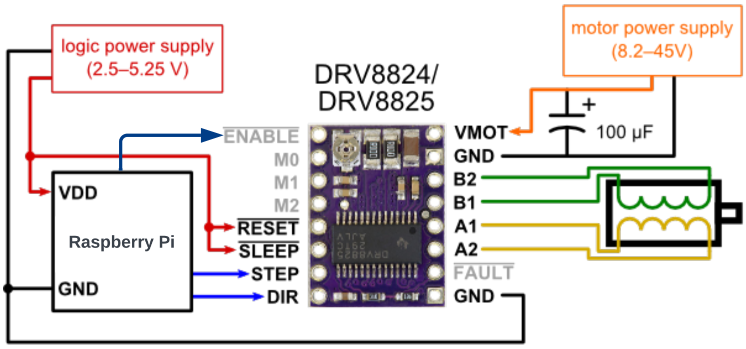


Figure 5.16: DRV8825 steeper motor driver schematic

The DRV8825 is connected to an I/O expansion board, M0, M1, M2 are resolution selection inputs connected to hardware switches on the expansion board.

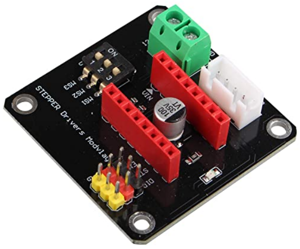


Figure 5.17: DRV8825 I/O expansion board

5.9 Task Feasibility

In order to design a complete system that meets the given requirements an evaluation of the feasibility of tasks that can be completed in VR is considered against tasks derived from the requirements survey. Task feasibility is evaluated from the understanding of the limitations of the hardware being used.

Task	Task ID	Direct	VR
Feed pet	T0	yes	yes
Light switch control	T1	no	no
Home Security	T2	no	yes
Pick and place	T3	yes	yes
Collecting parcels	T4	no	yes
Hang washing	T5	no	no
cook food	T6	no	no
Turning on appliances	T7	no	yes

Table 5.1: Task feasibility

In order to evaluate the feasibility of tasks each task needs to be explored more specifically, due to the consideration of the wide scope of this project there wont be enough time to explore each task in detail. Therefore the feasibility in this instance is decided based on the researchers knowledge of the capability's of the hardware to be implemented.

Chapter 6

Implementation

6.1 System Dependencies

Due to the wide scope of the project it wouldn't be viable to develop a novel and more appropriate solution for each component within the system due to the time constraints on the project. To mitigate the risk of running past the deadline focus and contribution are scoped accordingly. The utilisation of preexisting software solutions as dependencies for the complete system substantially reduces the workload to focus on other aspects of the solution.

6.1.1 ROS Sharp

ROS Sharp is a set of open source software libraries and tools developed by Siemens in the compiled language C# for communicating with ROS from .NET applications, in particular Unity (Bischoff, 2020).

6.1.2 Kinect Software

The OpenKinect project aimed to enable the use of Kinect hardware on windows Linux and mac OS. Libfreenect is an open source user-space driver for the Microsoft Kinect it includes all code necessary to activate, initialize, and communicate data with the Kinect hardware (Martin, 2021). This software is used to interface with Kinect and to publish its RGB images to the ROS message system that can be sent to the robot.

6.1.3 Kinect Auxiliary

Kinect aux is a ROS software package (Ivan Dryanovski, 2015) a standalone driver for the Kinect accelerometers and tilt motor. The package was tested to ensure the software meets the requirements for moving the motor however tilting the motor down was not possible after investigation the use of an unsigned datatype within the c++ language was the culprit. After this modification the software works as intended for this project.

```
void setTiltAngle(const std_msgs::Float64 angleMsg)
{
    uint8_t empty[0x1];
    double angle(angleMsg.data);

    angle = (angle<MIN_TILT_ANGLE) ? MIN_TILT_ANGLE : ((angle>MAX_TILT_ANGLE) ? MAX_TILT_ANGLE : angle);
    angle = angle * 2;
    const int ret = libusb_control_transfer(dev, 0x40, 0x31, (int16_t)angle, 0x0, empty, 0x0, 0);
    if (ret != 0)
    {
        ROS_ERROR_STREAM("Error in setting tilt angle, libusb_control_transfer returned " << ret);
        ros::shutdown();
    }
}
```

Figure 6.1: Kinect aux tilt function subscriber callback

6.1.4 Pigpio library

The pigpio library ("Pigpio Daemon", 2021) allows control of the raspberry pi GPIO written in the C programming language. Within this implementation the pigpio daemon is used, a daemon is a program that executes in the background ready to perform an operation when required without requiring any user interaction.

6.1.5 Inverse Kinematic Library for Arduino

To actuate the arm to match a users arm position the inverse kinematic approach was identified as the most appropriate since we know the users arm position form VR this library calculates the inverse kinematics for a 3 link arm with a rotating base Onchi (2020). This library uses the rule of cosines, the geometrical approach for kinematics calculation to calculate all motor angles to reach a given position in space.

6.1.6 Oculus Software

The Oculus software is also required to run the VR hardware. The Oculus unity SDK is then used together with the Oculus API to implement functionality within Unity, this paper focuses on the implementation of the touch controllers to interface with the mobile robot via button presses and motion data.

6.2 ROS Architecture

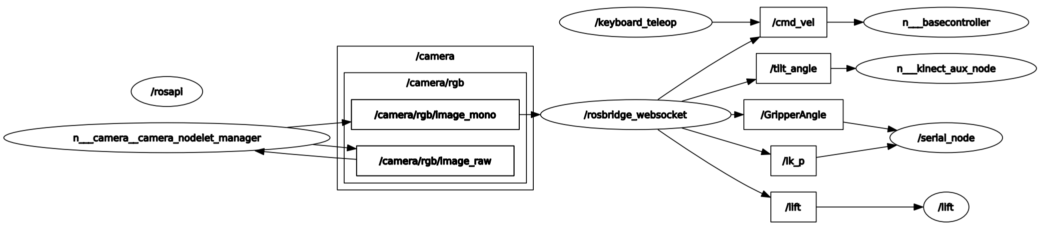


Figure 6.2: ROS Architecture

Ellipses denote nodes within the system each node is an executable program running inside the application. Squares denote a topic, topics are a part of ROS and are named buses over which nodes exchange messages.

Each node can either publish a message to a topic to send data or subscribe to a topic to receive data. To achieve this data is converted to the abstract data types used within ROS.

6.3 Lift

The lift feature of Mai has been implemented using a stepper motor controlled using a DRV8825 motor driver and the interpreted language python. The lift feature requires the implementation of a lift limiter function to ensure the lift cannot move past specified boundaries to prevent catastrophic hardware failure without this function the user will be able to cause damage to hardware.

Algorithm 1 Pseudo code, lift boundary function (limiter)

```
Require: D ← 0||1||2
  P
  L
  UB ← 10.0                                   ▷ Units: cm
  LB ← −10.0                                  ▷ Units: cm
  if D is equal to 1 then
      P ← P + 0.7
  else if D is equal to 2 then
      P ← P − 0.7
  end if
  if P < LB then
      print P
      print False
      L ← 0
      return True
  else if P > UB then
      print P
      print False
      L ← 0
      return True
  else
      print P
      return False
  end if
```

Where D denotes direction, P is position, L is lift data, UB is the upper bound and LB is lower bound. Print statements used for debugging.

6.3.1 Stepper Motor Control

Algorithm 2 Pseudo code, lift stepper control function

```
Require: D ← 1||2
  SD
  if D is equal to 1 then
      GPIO.output(direction,GPIO.LOW)          ▷ low = up, high = down
      limiter(1)
      for x in range(200) do
          GPIO.output(step,GPIO.HIGH)
          time.sleep(0.0005)                   ▷ delay 500 microseconds
          GPIO.output(step,GPIO.LOW)
          time.sleep(0.0005)                   ▷ delay 500 microseconds
          if SD is equal to 0 then
              break
          end if
      end for
  end if
  if D is equal to 2 then
      GPIO.output(direction,GPIO.LOW)          ▷ low = up, high = down
      limiter(2)
      for x in range(200) do
          GPIO.output(step,GPIO.HIGH)
          time.sleep(0.0005)                   ▷ delay 500 microseconds
          GPIO.output(step,GPIO.LOW)
          time.sleep(0.0005)                   ▷ delay 500 microseconds
          if SD is equal to 0 then
              break
          end if
      end for
  end if
  SD ← 1
```

This control method uses the full step mode for maximum holding torque required to hold the robots' lift. Micro stepping offers the programmer more resolution in applications that require more accuracy, considering the application of this lift function resolution is not required to be high since full step accuracy will suffice.

This implementation allows the motor to be locked when both phases are energised this helps to mitigate the lead screw being back driven in the case that the robot picks up too much mass that is enough to back drive the lead screw.

MS1	**MS2**	**MS3**	**Step increment(°)**	**Steps per revolution**
0	0	0	Full Step (1.8°)	200
1	0	0	Half Step (0.9°)	400
0	1	0	1/4 Step (0.45°)	800
1	1	0	1/8 Step (0.225°)	1600
0	0	1	1/16 Step (0.1125°)	3200
1	0	1	1/32 Step (0.05625°)	6400
1	1	0	1/32 Step (0.05625°)	6400
1	1	1	1/32 Step (0.05625°)	6400

Table 6.1: Micro stepping resolutions

6.4 High-Level Code

6.4.1 Python

```
def limiter(direction):
    global position
    global globaldata0
    upperBound = 10.0 #cm
    lowerBound = -10.0 #cm
    if direction == 1:
        position = position + 0.7 #cm
    elif direction == 2:
        position = position - 0.7 #cm
    if position < lowerBound:
        print(position)
        print("false")
        globaldata0 = 0
        return True
    elif position > upperBound:
        print(position)
        print("false")
        globaldata0 = 0
        return True
    else:
        print(position)
        return False
```

Figure 6.3: High level python implementation for lift limiter

The lift limiter function is an estimation of the robots lift position each 360 degree rotation translates to approximately 0.7 cm of lift movement. This function sets a limitation on the amount of translation that is allowed before an internal hardware collision. An alternative to this implementation would be to implement this feature in hardware using 2 limiting switches affixed to the maximum and minimum lift positions to interrupt actuation.

```python
# each callback fuction operates in its own thread
def callback2(data):
    data0 = data.data
    global globaldata0
    globaldata0 = data0

def callback(data):
    data0 = data.data
    global globaldata0
    rospy.loginfo(rospy.get_caller_id() + 'I heard %s', data.data)

    GPIO.setmode(GPIO.BCM)
    GPIO.setup(EN_pin,GPIO.OUT) # set enable pin as output
    GPIO.setup(step, GPIO.OUT)
    GPIO.setup(direction, GPIO.OUT)
    GPIO.output(EN_pin,GPIO.LOW)

    if data0 == 1:
        GPIO.output(direction,GPIO.LOW)
        limiter(1)
        for x in range(200):
            GPIO.output(step,GPIO.HIGH)
            time.sleep(0.0005) # delay 500 microseconds
            GPIO.output(step,GPIO.LOW)
            time.sleep(0.0005) # delay 500 microseconds
            if globaldata0 == 0:
                break
    if data0 == 2:
        GPIO.output(direction,GPIO.HIGH)
        limiter(2)
        for x in range(200):
            GPIO.output(step,GPIO.HIGH)
            time.sleep(0.0005) # delay 500 microseconds
            GPIO.output(step,GPIO.LOW)
            time.sleep(0.0005) # delay 500 microseconds
            if globaldata0 == 0:
                break

    globaldata0 = 1
```

Figure 6.4: High level python implementation for stepper motor control

Figure 6.3 shows the final python implementation for stepper motor control data of the type Int8 is published to the lift topic by the lift publisher script running within the unity environment. The callback function is passed the integer value as an argument, if the data is one the lift goes up, if the data is two the lift goes down. A second callback function named callback2 operates in a separate thread to assign data to a global variable, zero means the lift needs to stop and the for loops are interrupted buy the update to the global variable changed by callback2 to stop actuation. This approach is necessary since the thread for callback one is in a loop so a second thread is required to update the global variable.

6.4.2 C++ Omnidirectional Base Kinematics

```cpp
void chatterCallback(const geometry_msgs::Twist::ConstPtr& msg)
{
  //pin 19&26 have been burnt out cause: shorted motor driver
  // original mapping incorrect setPin(5, 6, 17, 18, 9, 25, 13, 12);
  setPin(9,25,13,12,5,6,17,18);

  //wheel n = (1/WHEEL_RADIUS) * (linear.x ± linear.y ± (WHEEL_SEPARATION_WIDTH + WHEEL_SEPARATION_LENGTH)*angular.z);
  float wheel_front_left = (1/0.50) * (msg->linear.x - msg->linear.y - (0.400 + 0.400)*msg->angular.z);
  float wheel_front_right = (1/0.50) * (msg->linear.x + msg->linear.y + (0.400 + 0.400)*msg->angular.z);
  float wheel_rear_left = (1/0.50) * (msg->linear.x + msg->linear.y - (0.400 + 0.400)*msg->angular.z);
  float wheel_rear_right = (1/0.50) * (msg->linear.x - msg->linear.y + (0.400 + 0.400)*msg->angular.z);

  //no need to invert right side (inverted in hardware)

  float wfr = rpm2pwm(wheel_front_right);
  float wfl = rpm2pwm(wheel_front_left);
  float wrr = rpm2pwm(wheel_rear_right);
  float wrl = rpm2pwm(wheel_rear_left);

  right_.drive(wfr);
  left_.drive(wfl);
  right2_.drive(wrr);
  left2_.drive(wrl);

  ROS_INFO("I heard x: [%f]", msg->linear.x);
  ROS_INFO("I heard y: [%f]", msg->linear.y);
  ROS_INFO("I heard y: [%f]", msg->angular.z);
  ROS_INFO("I heard wheel fl: [%f]", wfl);
  ROS_INFO("I heard wheel fr: [%f]", wfr);
  ROS_INFO("I heard wheel rl: [%f]", wrl);
  ROS_INFO("I heard wheel rr: [%f]", wrr);

}
```

Figure 6.5: High level C++ implementation mecanum wheel kinematics

Wheel actuation requires the ability to be interrupted in order to stop the robot in an emergency. To achieve this two asynchronous callbacks were implemented to achieve non-blocking execution between functions. This implementation is single threaded.

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \frac{1}{r}\begin{bmatrix}1 & -1 & -(l_x+l_y)\\ 1 & 1 & (l_x+l_y)\\ 1 & 1 & -(l_x+l_y)\\ 1 & -1 & (l_x+l_y)\end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ \omega_z\end{bmatrix}.$$

$$\begin{cases}\omega_1 = \frac{1}{r}\left(v_x - v_y - (l_x+l_y)\omega\right),\\ \omega_2 = \frac{1}{r}\left(v_x + v_y + (l_x+l_y)\omega\right),\\ \omega_3 = \frac{1}{r}\left(v_x + v_y - (l_x+l_y)\omega\right),\\ \omega_4 = \frac{1}{r}\left(v_x - v_y + (l_x+l_y)\omega\right).\end{cases}$$

Figure 6.6: Forward kinematics (Taheri et al., 2015)

The forward kinematics calculations return each wheel speed in rad/s, this is then converted to RPM.

$$RPM = \omega \times 9.549297 \tag{6.1}$$

Where: ω [rad/s], is the wheels angular velocity.

Per the motor specifications defined in the design each motor has a theoretical speed of 200RPM at a full PWM duty cycle. Therefore forward kinematic calculations that return values more than 200RPM are not viable as the limitation from the motors means that movement at the defined velocity in meters per second cannot be achieved.

```
float rpm2pwm(float radiansPS)
{
  //rad/s to rpm
  float RPM = radiansPS * 9.549297;
  float ViableRPM = RPM * 10;
  float RPMdutyCycle = ViableRPM * 0.78431372549; //200/255
  if (ViableRPM > 200)
    {
      RPMdutyCycle = 255;
    }
  else if (ViableRPM < -200)
    {
      RPMdutyCycle = -255;
    }
  return RPMdutyCycle;
}
```

Figure 6.7: RPM to PWM conversion

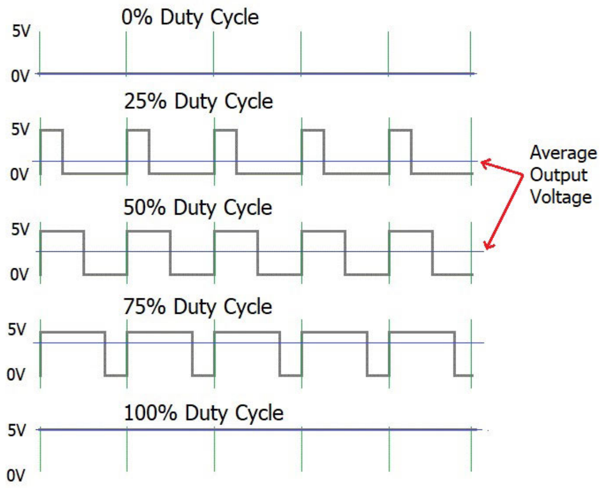


Figure 6.8: PWM Duty Cycle

PWM has been implemented to control the speed of the DC motors. A duty cycle is the fraction of one period when a system or signal is active figure 6.6 shows a graphical representation of PWM. PWM duty cycle allows the average voltage level to be adjusted this allows the motor to rotate at variable speeds by varying the current.

```
void drive(int power) const
{
  // L298N
  if (power > 0)
  {
    set_PWM_dutycycle(pi_, pin_in1_, power);
    gpio_write(pi_, pin_in2_, 0);
  }
  else
  {
    power = -power;
    gpio_write(pi_, pin_in1_, 0);
    set_PWM_dutycycle(pi_, pin_in2_, power);
  }
}
```

Figure 6.9: Function for direction inversion

Each motor driver is connected to the raspberry pi via four input pins, two pins are used to drive each motor. Directional control is implemented here by by setting one pin to a low voltage and the second to an adjustable voltage PWM signal. These mappings are then inverted to drive the motor in the opposite direction. Figure 6.7 shows the implementation of the inversion.

6.4.3 Simplified C (Arduino IDE)

```
void loop()
{
  Braccio.ServoMovement(10,_baseAngle,_shoulderAngle,_elbowAngle,_wrist_verAngle,_wrist_rotAngle,_gripperAngle);

  float a0, a1, a2, a3;

  if(InverseK.solve(x, y, z, a0, a1, a2, a3)) {
  _baseAngle = a2b(a0);
  _shoulderAngle = a2b(a1);
  _elbowAngle = a2b(a2);
  _wrist_verAngle = a2b(a3);

  }

  nh.spinOnce();
}
```

Figure 6.10: High level C implementation arm control loop

ROS serial is used to serialise data to be sent to the Arduino. The point in the VR space is received via websocket by ROS on the pi and is sent via the ikp topic to the serial node that interfaces with the serial port on the Arduino.

6.5 Robotic arm

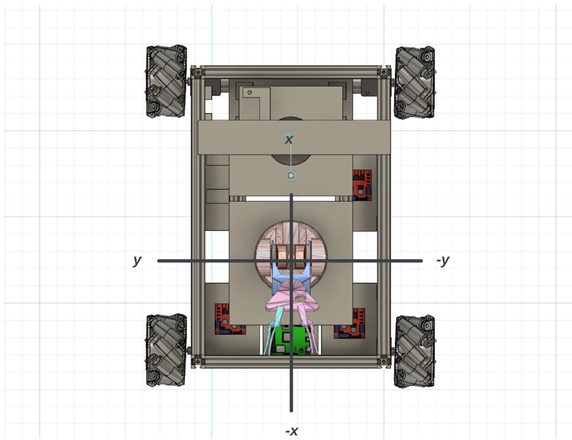


Figure 6.11: Arm coordinate system mapping

Unity uses a left hand cartesian coordinate system, coordinates from the Unity environment are translated to coordinates within the arms' right hand cartesian coordinate system.

After this conversion is complete a point within Unitys coordinate system is translated to the arms' coordinate system and passed as an argument to the inverse kinematics solver. If the solver returns true the calculated joint angles a0 - a3 are stored in the baseAngle, shoulderAngle, elbowAngle and wristverAngle variables respectively.

6.6 Unity C#

Scripting within unity was used to send data from the desktop PC to the robots' server. Oculus provides an SDK for the Unity development environment. The SDK has been used to implement key VR features such as user input, controllers, and rendering to build and immersive VR experience for Oculus devices that can be used within the unity scripting API.

```
private void UpdateMessage()
{
    OVRInput.Update();

    Vector2 coord = OVRInput.Get(OVRInput.Axis2D.PrimaryThumbstick);
    Vector2 coord2 = OVRInput.Get(OVRInput.Axis2D.SecondaryThumbstick);

    Vector3 linearVelocity = (PublishedTransform.localPosition - previousPosition) / Time.fixedDeltaTime;
    Vector3 angularVelocity = (PublishedTransform.localRotation.eulerAngles - previousRotation.eulerAngles) / Time.fixedDeltaTime;

    message.linear = GetGeometryVector3(linearVelocity.Unity2Ros());
    message.angular = GetGeometryVector3(-angularVelocity.Unity2Ros());
    message.linear.x = coord.y;
    message.linear.y = coord.x;
    message.angular.z = coord2.x;

    if (coord2.x > 0)
    {
        message.linear.x = message.linear.x + 0.01;
    }
    else if (coord2.x < 0)
    {
        message.linear.x = message.linear.x + (-0.01);
    }

    previousPosition = PublishedTransform.localPosition;
    previousRotation = PublishedTransform.localRotation;

    Publish(message);
    Debug.Log(coord.y);
    //Debug.Log(coord.x);
}
```

Figure 6.12: Twist Publisher

In order to get input from the controllers the OVRInput.Update() and OVRInput.FixedUpdate() functions are called once per frame at the beginning of any component's Update and FixedUpdate methods the button states from the controllers is therefore updated each frame.

The twist publisher script converts coordinates returned by the controller thumb sticks to velocity commands to control the robot. ROS uses meters per second for velocity calculation, due to lack of specialised hardware only estimated wheel RPM is considered. A float value of one returned by the analogue thumb stick is considered the maximum speed. A float returned by the Oculus API as shown in figure 6.9 resulted in the best outcome for progressive speed control allowing for the maximum 200RPM whilst allowing smooth ramp-up and ramp-down controlled directly by the user in order to minimise jerk.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

namespace RosSharp.RosBridgeClient
{
    public class GripperPublisher : UnityPublisher<MessageTypes.Std.UInt32>
    {
        private MessageTypes.Std.UInt32 message;

        // Start is called before the first frame update
        protected override void Start()
        {
            base.Start();
            InitializeMessage();
        }

        private void FixedUpdate()
        {
            OVRInput.FixedUpdate();
            UpdateMessage();
        }

        private void InitializeMessage()
        {
            message = new MessageTypes.Std.UInt32();
        }

        private void UpdateMessage()
        {
            OVRInput.Update();
            message.data = System.Convert.ToUInt32(OVRInput.Get(OVRInput.Axis1D.SecondaryHandTrigger, OVRInput.Controller.Touch) * 73);
            //Debug.Log(message.data);
            Publish(message);
        }
    }
}
```

Figure 6.13: Gripper Publisher

The gripper publisher script gets updates from controller button states to publish a control message to the GripperAngle topic. The data is transmitted by the computer running the VR application via WebSockets this data is then received by ROS on the raspberry pi and is serialised and sent to the Arduino.

```
        private void FixedUpdate()
        {
            OVRInput.FixedUpdate();
            UpdateMessage();
        }

        private void InitializeMessage()
        {
            message = new MessageTypes.Geometry.Point32();
        }

        private void UpdateMessage()
        {
            OVRInput.Update();

            if (OVRInput.Get(OVRInput.RawButton.RIndexTrigger) && (!active))
            {
                rightHandOrigin = PublishedTransform.position;
                active = true;
                //Debug.Log(rightHandOrigin.x);
            }

            if (OVRInput.Get(OVRInput.RawButton.RIndexTrigger) && (active))
            {
                rightHandUpdate = PublishedTransform.position;
                rightHandDifferance.x = rightHandOrigin.x - rightHandUpdate.x;
                rightHandDifferance.y = rightHandOrigin.y - rightHandUpdate.y;
                rightHandDifferance.z = rightHandOrigin.z - rightHandUpdate.z;

                rightHandDifferance.x = rightHandDifferance.x * 800;
                rightHandDifferance.y = rightHandDifferance.y * 800;
                rightHandDifferance.z = rightHandDifferance.z * 800;
                message.x = -600 + rightHandDifferance.z;
                message.y = -rightHandDifferance.x;
                message.z = -rightHandDifferance.y;
                Debug.Log(rightHandDifferance.x);
            }

            if (!OVRInput.Get(OVRInput.RawButton.RIndexTrigger))
            {
                active = false;
            }

            //Debug.Log(rightHand.y);
            Publish(message);


        }
    }
}
```

Figure 6.14: Inverse kinematics point publisher

The inverse kinematics point publisher converts the coordinates of the right hand controller to a point within the arms' local cartesian coordinate system. The inverse kinematics calculation is computed on the Arduino using this specified point in space.

6.7 VR interface

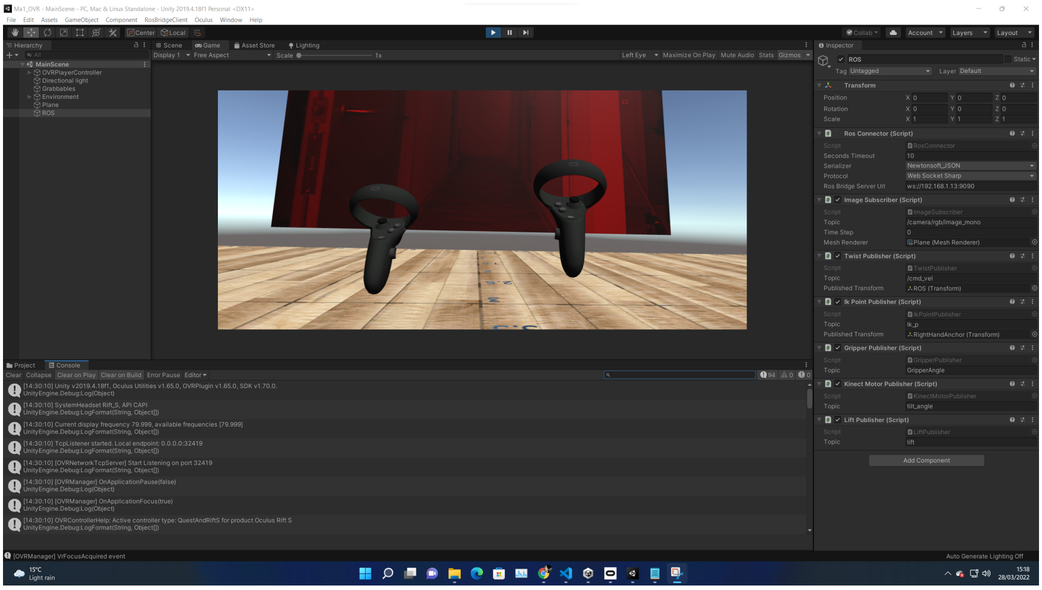


Figure 6.15: Unity scene - VR interface

Within the VR environment the user is presented with a large screen that displays what the robot is viewing. The touch controllers are rendered within the scene they are tracked by the headset using computer vision, this allows the user to learn more intuitively if they forget where a button is located they can view the physical controllers virtually without having to remove the HMD. The ImageSubscriber script receives the image returned by the Kinect sensor that is then projected onto a plane (screen) within Unity. The KinectMotorPublisher script sends commands to the ROS system via the tilt angle topic.

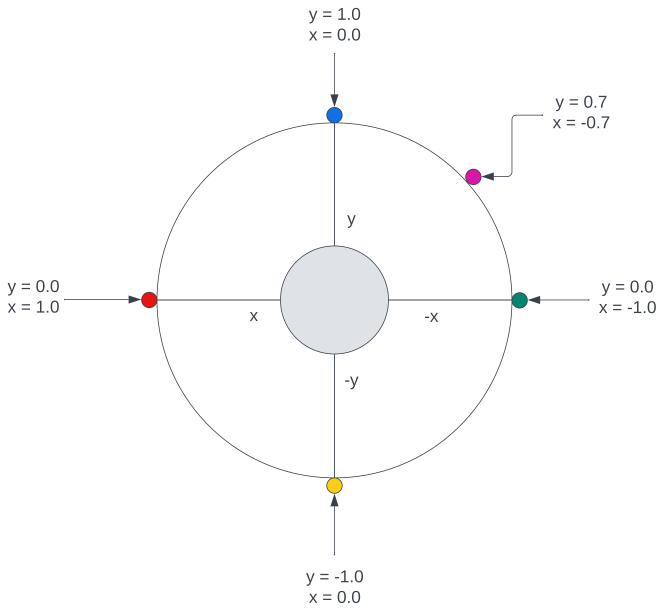


Figure 6.16: Thumb-stick mapping

The analog thumb stick mapping is shown in figure 6.16 when the user pushes the thumb stick forward indicated in blue the robots linear velocity matches the thumb position and the robot moves forward at its maximum speed at y values less than 1 the robots speed is decreased. Negative values of y move the robot backwards. The same applies to the x axis. Left moves the robot left and right moves the robot right. The robots' linear velocity mirrors the direction the user is pointing the thumb stick and speed is adjusted by moving closer to the thumb stick perimeter. The y axis is mapped to the robots' x axis and x is mapped to the robots' y as ROS and Unity use different coordinate systems. The right hand thumb stick is used for orientation control for the angular component of the robot, left orientates the robot anti clockwise, right orientates the robot clockwise at variable speed.

6.8 Boot Sequence

Each node within the ROS system can be launched using the command line, there is also a ROS launch system that runs defined nodes. This system uses a launch file written in XML ros launch evaluates the XML file in a single pass. Includes are processed in depth-first traversal order.

```
<launch>
  <param name="robot_description" textfile="$(find ma1)/urdf/ma1.urdf" />
  <node name="basecontroller" pkg="ma1" type="basecontroller" />
  <node name="lift" pkg="ma1" type="lift.py" />
  <node name="armcontroller" pkg="ma1" type="armcontroller" />
  <node name="kinect_aux_node" pkg="kinect_aux" type="kinect_aux_node" />
  <node pkg="ma1" type="tf_broadcaster" name="tf_broadcaster" />

  <node pkg="rosserial_python" type="serial_node.py" name="serial_node">
    <param name="port" value="/dev/ttyACM0"/>
    <param name="baud" value="57600"/>
  </node>

  <node name="keyboard_teleop" pkg="ma1" type="keyboard_teleop.py" output="screen" />

</launch>
```

Figure 6.17: ROS launch XML

6.9 Development Environment

6.9.1 SSH Protocol

Development was completed using a SSH connection to the Raspberry pi running Ubuntu server. Ubuntu server is running in headless operation so to program the board a client needs to be connected. SSH also provides access to the Linux terminal allowing command-line execution of various system and ROS commands. A static IP address is assigned on the router being used to ensure the host is always reachable, a dynamic IP can still be used however finding the new IP every time the router assigns a new one would cost time.

6.9.2 Visual Studio Code

Visual studio code was the main development environment used for programming of the mobile robot. Using an SSH plugin the client application is connected to the server on the robot the users file system is then displayed. First a Catkin work space is created, Catkin provides a Low-level build system, CMake macros and infrastructure for ROS, next is to create a ROS software package within the Catkin workspace. This software package is where the bulk of development on the mobile robot has been completed. CMake is cross-platform free and open-source software for build automation, testing, packaging and installation of software by using a compiler-independent method, CMakeLists.txt is responsible for preparing and executing the build process.

6.10 Source and Supporting Deliverables

Please find all source code including binaries and supporting files at the google drive link below.

https://drive.google.com/drive/folders/1CYMTOfj81MwvDVvgXs04XfTVHXEK3DVu?usp=sharing

Chapter 7

Testing

7.1 Unit Testing

Each component of the system has been tested prior to installation within the robots' chassis. The methodology used is independent for each component, testing does follow the agile approach but is conducted accordingly to test the specifics for each component appropriately.

7.1.1 Motor & Driver

The motors and drivers were tested using the raspberry pi GPIO before all hardware was affixed to the chassis this was done to ensure both the standard DC motor and stepper motor and selected drivers worked as expected. The enable pins of the L298n driver are connected to its own internal logic supply two pin jumpers (also called shunts) that will create an electrical connection between two pin headers are used to set the driver to always be enabled.

7.1.2 Limiter Function

The lift limiter function was tested before user testing to ensure it worked correctly in protecting hardware from damage from misuse by the user.

Increment	Value	Result
0	0.0	True
-1	-0.7	True
-2	-1.4	True
-3	-2.1	True
-4	-2.8	True
-5	-3.5	True
-6	-4.2	True
-7	-4.9	True
-8	-6.3	True
-9	-7.0	True
-10	-8.4	True
-11	-9.1	True
-12	-9.8	True
-13	-10.5	False

Table 7.1: Limiter Function Experimental and Analytical Results

The lift limiter function worked as intended, boundaries are set to 10.5cm of z-axis translation. This is an under estimation of the maximum potential movement of the lift in the z-axis to account for variations in starting position.

7.2 Omnidirectional Base Testing

Before moving ahead to user testing the system was first tested by the researcher to ensure that all components work as expected. A key component of the system is the omni-directional movement achieved using a 4 mecanum wheel configuration.

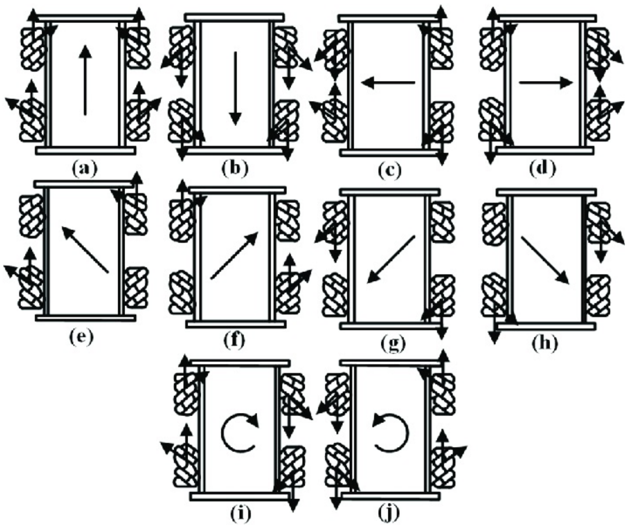


Figure 7.1: Mecanum wheel turning principle

Direction	v_x	v_y	ω_z	**Wheel FL**	**Wheel FR**	**Wheel RL**	**Wheel RR**
Forward (a)	0.8	0.0	0.0	1.6	1.6	1.6	1.6
Backward (b)	-0.8	0.0	0.0	-1.6	-1.6	-1.6	-1.6
Strafe R (d)	0.0	0.8	0.0	-1.6	1.6	1.6	-1.6
Strafe L (c)	0.0	-0.8	0.0	1.6	-1.6	-1.6	1.6
Diagonal (f)	0.8	0.8	0.8	0.0	3.2	3.2	0.0
Diagonal (e)	-0.8	0.8	0.0	-3.2	0.0	0.0	-3.2
Orientation(i)	0.0	0.0	0.8	-6.4	6.4	-6.4	6.4
Orientation(j)	0.0	0.0	-0.8	6.4	-6.4	6.4	-6.4

Table 7.2: Experimental and Analytical Results

- FL = Front Left
- FR = Front right
- RL = Rear Left
- RR = Rear Right

- Units = $[rad/s]$
- Wheel RPM at 1.6 is $15.2788752 \times 10 = 152.788752$. wheel RPM is multiplied by 10 to account for the low torque of the DC motors at low speeds.
- RPMs above 200 are evaluated to 200 using a PWM duty cycle of 100% as this is the maximum motor speed.

Each direction was tested, all worked well and as expected. However due to the lack of encoding from the motors some drift was experienced. The theoretical speed of each motor is not exact due to variations in manufacturing this results in each motor having a slight variation in its actual real world speed. The inclusion of odometry and inertia measurement would solve this problem.

7.3 User Testing

User testing has been conducted to evaluate the performance within a use case for the system. The scene for testing is detailed in figure 7.1 the setup of this scene remains constant throughout the testing phase. Within user testing users are asked to complete a set of tasks.

- Task 1: Pickup object 0 and place in bin.
- Task 2: Pickup object 1 and place in bin.
- Task 3: Activate switch on object 2.

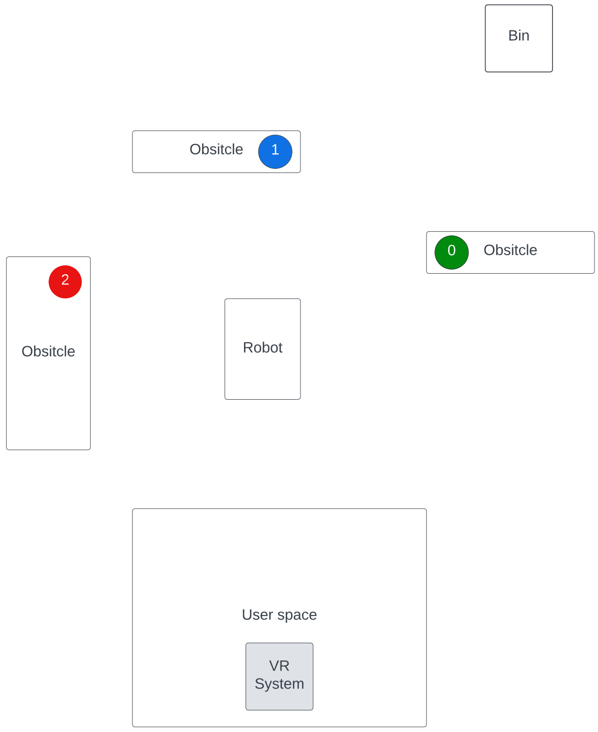


Figure 7.2: User testing scene (not to scale)

7.3.1 Observations

- Inverse kinematics approach angle changes unexpectedly causing the user to falter as the robots arm doesn't follow the users arm motion.
- Collisions were observed, each collision was the intersection of the robots base and an obstacle. No collisions were damaging to hardware or the external environment.
- Latency was observed intermittently in the actuation of arm movement, latency in the camera stream caused by a slow wireless connection when further from the access point.
- Difficulty picking up objects, due to the lack of depth data from the camera stream.

- Home position causing confusion in users, many users assumed to resume control of the arm from its stopping position.

7.3.2 Post User Testing Survey

After completing the test each user was asked to fill out a short survey that gathered data on their thoughts about their experience with Mai and the overall system.

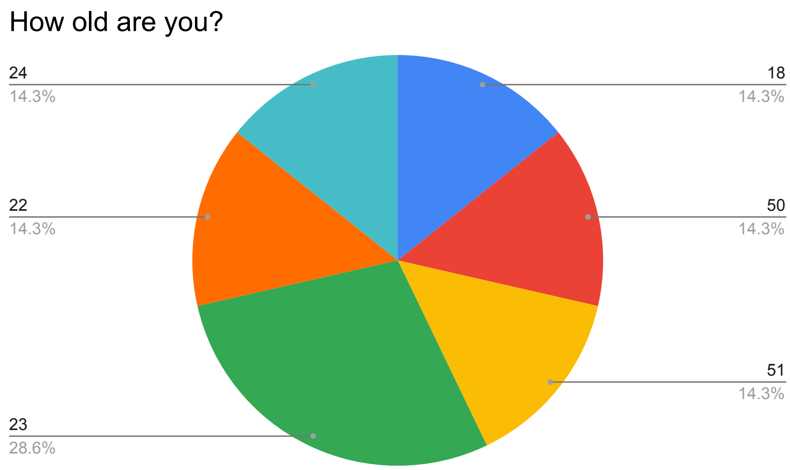


Figure 7.3: User testing survey Q1

Following the same procedure as the requirements survey the age of the participant is considered again to discern whether user testing and the associated post testing survey was successful in engaging participants from a variety of ages.

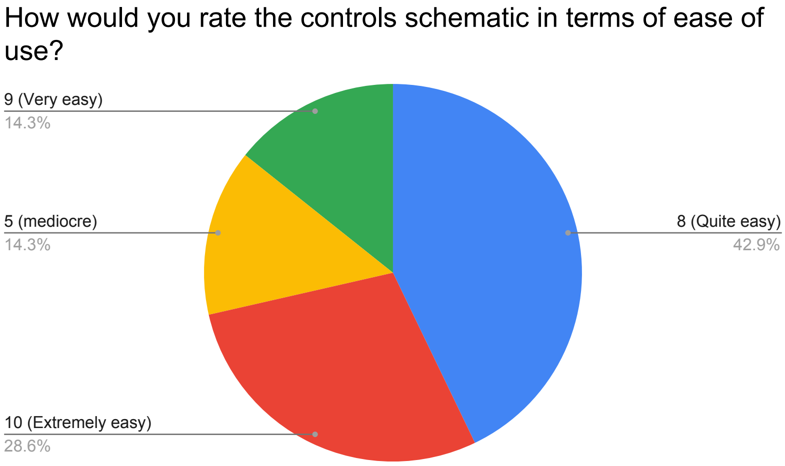


Figure 7.4: User testing survey Q2

Users were asked to rate their experience with the control schematic to evaluate the ease of use within the VR application.

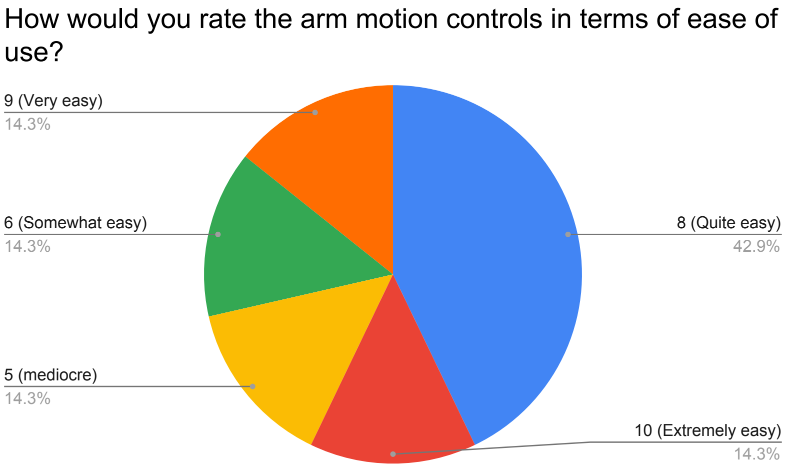


Figure 7.5: User testing survey Q3

Users were asked to rate their experience with the motion control of the robots' arm to help evaluate how successful this method of control was in terms of ease of use.

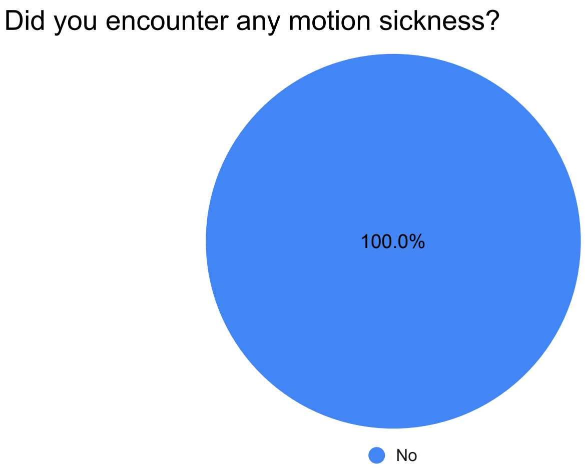


Figure 7.6: User testing survey Q4

All users answered no to this question, this was a positive yet unexpected result as one of my biggest concerns was with motion sickness within the VR application. Vibration from the robots actuators translates to excessive movement of the kinect sensor due to sub-par mounting and excessive movement from the kinect suspension shaft connected to the kinects' internal motor.

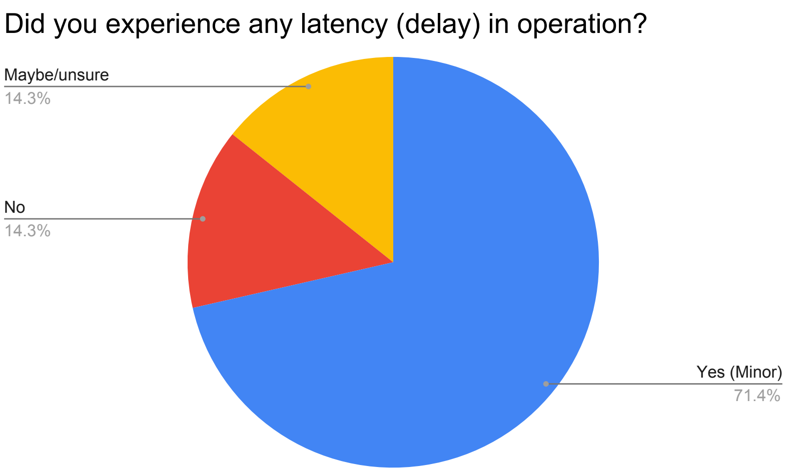


Figure 7.7: User testing survey Q5

Users were asked to identify whether they experienced any delay in operation this was asked to evaluate whether the complete system was successful in delivering a usable experience since latency is mission critical and has a direct affect on the users ability to complete set tasks.

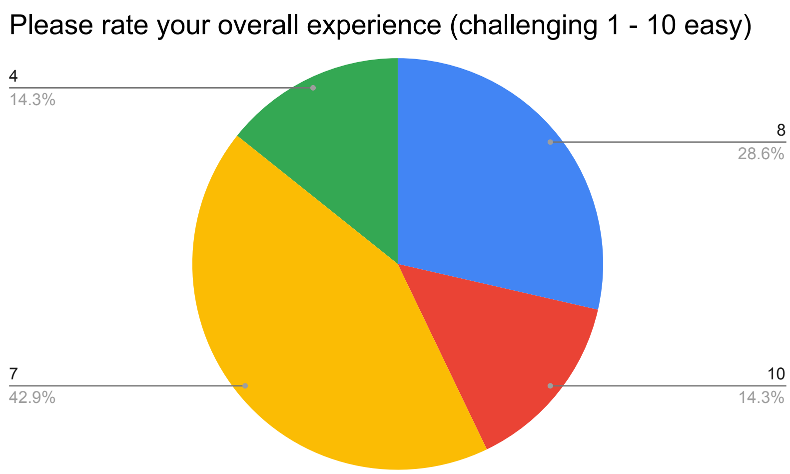


Figure 7.8: User testing survey Q6

Users were asked to rate their overall experience to discern the general consensus on whether they found the system to challenging or easy to use on a scale of 1-10. The results show that most participants found the system to be above 5 more easy than challenging except from 14.3% who found the experence to be more challenging.

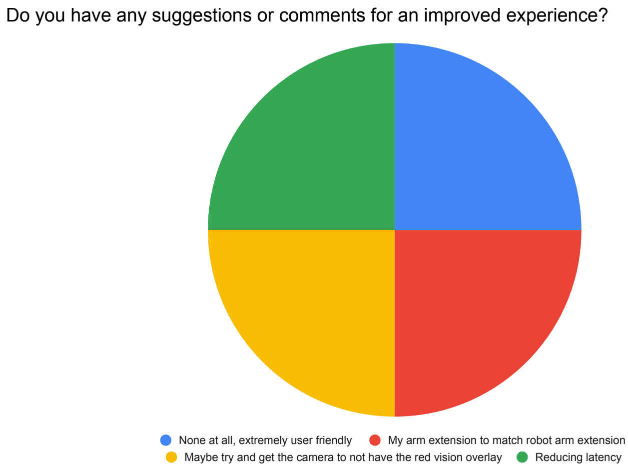


Figure 7.9: User testing survey Q7

Users were asked to provide suggestions for an improved experience this was to help influence future work and identify areas for further development within the system. Reducing latency was identified it may have been more appropriate to ask questions on each specific element to gather more accurate data since latency is an important considerable in each component of the system. A judgement can be made that the participant was referring to the latency observed in arm control. The latency for control of the mobile base was very minimal and did not effect the user experience from my observations.

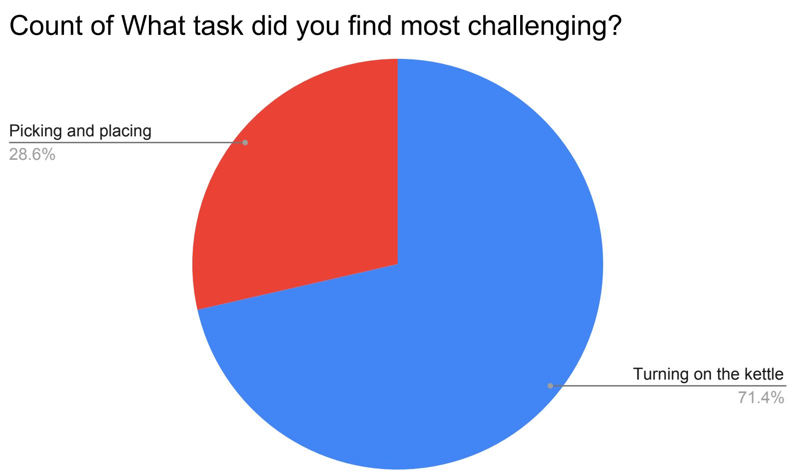


Figure 7.10: User testing survey Q8

The system was developed to perform pick and place tasks, these tasks were identified to be the least challenging. This is a good result. Turning on the kettle was identified from the user requirements survey as a task that users would be interested in completing it was then integrated as part of user testing. Since the system wasn't designed wit this use case in mind the results from the survey show how this task was more challenging.

Chapter 8

Evaluation

8.1 Evaluation Plan

An evaluation is conducted to asses the project, this assessment is conducted through the use of retrospectives to consider methodology used and what requirements were met. How each iteration can be improved over the former, including whether the

8.1.1 Retrospectives

The chosen methodology used was the agile methodology, throughout the project iterations of each component were developed, an Arduino program for arm control (Two iterations), a omnidirectional kinematics based Base controller (Two iterations) and lift controller (Three iterations). The agile approach was successful in accepting changes to requirements. Lots of time was wasted within the project, developing iterations that were not effective was a misuse of time, better planning would have been beneficial to avoid these pitfalls.

8.1.2 Iteration 0 (Omitted)

Within this iteration ROS was being tested with the hardware that needed to be integrated to ensure compatibility and to asses viability within my proposed design of the system. ROS2 was identified as the most recent release of ROS. Testing started using ROS2 due to a redesigned and improved architecture. However it was found in research that there is currently no preexisting solution for data serialisation compatible with the Arduino Uno MCU there is already a solution being developed called micro ROS that allows ROS2 to communicate with low cost embedded systems however it does not support the ATmega328P based 8-bit MCU in the Arduino Uno so it would not be feasible considering the time constraints on the project to develop a solution. For this reason the most recent ROS1 distribution noetic is used instead.

8.1.3 Iteration 1

Iteration 1 was successful in actuating the wheels of the mobile robot within ROS noetic, the approach was implemented using first order decision logic as well as ramping functions to slowly increase and decrease motor speed to reduce jerk and mechanical damage to the motors. This approach was successful in implementing forward, reverse and strafing movement using asynchronous callbacks to interrupt loops to stop actuation. This approach did not achieve full omnidirectional movement of the mobile base. An arm controller program was implemented in C++ to communicate with the Arduino to publish joint states to the arm. This approach was successful however this approach would be difficult to use in practice as inverse kinematics calculations for the control of the arm would need to be implemented within unity to send joint states to the Arduino MCU.

8.1.4 Iteration Two (Final)

Iteration two implemented forward kinematics for the robots' omni-directional base. This was successful in achieving omni-directional movement. Ramping functions ware not used here, this is due to the consideration that the user input is mapped directly. This design choice was appropriate as the robot should mimic the exact input provided by the user. Users can cause excessive jerk by moving the controller thumb sticks to fast however when applying progressive controlled movement jerk is minimised. Inverse kinematics were implemented on the Arduino Uno to control the movement of the arm to mimic a point of space in the VR environment. This allows the user to move the controller to actuate the arm for intuitive control. The final iteration does not consider speed within ROS.

8.2 Evaluation Against Requirements

The final iteration detailed in the implementation was successful in meeting the project requirements. A successful user interface has been presented with intuitive motion control although depth perception would be a valid improvement to improve the operators experience. Security has been implemented through the use of SSH whereby the user must provide a password in order to boot the system and communication between the robot server and a client machine is encrypted. The final iteration and prototype are reliable this is supported by the user testing results. Latency is minimal although intermittent spikes of latency are observed yet no diagnosis has been for this problem. The complete artefact is mobile in hardware and portable in software as a ROS package. Battery life has been substantial enough to provide over one hour of usage. The artefact fits within a home environment this was shown again by the user testing. Software architecture is maintainable and executes without significant defects over an extensive period of use. The artefact has been operable by laymen. The final artefact does have its limitations, these limitations come mostly from hardware, no encoding on DC motors that causes drifting and limited load capacity of the arm at a maximum weight of 150g at a 32cm operating distance relative to the arms' base.

8.3 Evaluation of expenditure

8.3.1 Budget

The working budget allocated to the project is £300. This does not include any tools, equipment or other resources. There is to be a set reserve of £100 this is a safety net for replacement hardware in anticipation for a failure, short, electrostatic discharge or any other unforeseen unfortunate event.

ID	Component	Quantity	Price
E0	Aluminum extrusion (anodised) 800mm	2	£13.82
E1	Aluminum extrusion (anodised) 250mm	7	£15.12
E2	Aluminum extrusion (anodised) 400mm	5	£17.28
E3	M5 Tapping service	2	£0.30
E4	V-Slot gantry plate	2	£25.98
E5	L298n motor driver	2	£6.60
E6	DRV8825 motor driver	1	£2.95
E7	Arduino	1	£6.59
E7	Arduino Braccio (robot arm)	1	£218.95
E8	Raspberry Pi 4B	1	£74.39
E9	DC/DC Converter 12V Step Down 5V USB	1	£7.77
E10	DC/DC Converter 12V Step Down 9V	1	£8.99
E11	DC/DC Converter 12V Step Down 5V	1	£12.75
E12	12V Regulator/ 4A current limiter	1	£23.99
E13	Stepper Motor Driver Expansion Board	1	£5.39
E14	12V LiFePo4 10Ah battery	1	£50
		Total:	£490.87

Table 8.1: Evaluation of expenditure

Overall the expenditure came to £490.87 this is £90.87 over budget. The over expenditure was taken from the £100 reserve, an excess of £90.87 was needed to complete the prototype the use of a reserve was intended for hardware replacement, however this reserve was used instead to complete the final prototype. There were two instances of having to replace hardware, the first instance was an electrical short on one L298n that was caused when constructing the robot. The second instance of hardware damage was caused in the testing phase where the robot collided heavily with household furniture causing one DC motor to fail.

8.4 Project management

The management structure used throughout the project was successful in keeping the project on track and contributed to the project achieving its goal. Using a gantt chart for visual representation of progress within excel (see appendix B) to show progress bars for each objective was invaluable in organising the project. Each objective was completed on schedule. However to keep the project on schedule certain aspects were slightly rushed, this led to some important details being overlooked, estimation calculations rather than more concrete mathematical functions for example. The project goal was achieved as users could implement Mai to improve domestic life.

8.5 Contribution

This paper contributes to academic literature in the field to show what can be achieved even with very simple hardware. How open source hardware and software products show great potential in allowing education of such complex devices. This contribution shows that modern robotics can indeed be integrated into the domestic environment to improve the quality of life for every day people. Despite IoT integration, smart devices and home automation, robotics still has a place and a role to play, laymen with unfamiliarity with technology and automation could implement Mai as a one size fits all approach to achieve multiple tasks within a dynamic environment.

Chapter 9

Conclusion

The overall system artefact created used pre-existing software solutions for the integration of system components as dependencies. This was beneficial to the project as it allowed the creation of a very complex system with less development time needed. However some of the problems identified within the system from testing originate from these pre-existing solutions therefore it has been difficult to diagnose the problem. One of these problems identified was the synchronisation between the Arduino MCU and ROS serial software package that bridges the two systems. Overall the system meets the requirements outlined at the start of the project and is successful in being operable by laymen.

The design was successful as the final prototype works as intended however more time should have been spent examining more specifics within the design process to help prevent pitfalls found during the development of the system. Designing a second prototype would be more feasible than continuing development on the existing prototype as better hardware solutions are needed to achieve a more robust artefact. That being said the final software iteration can still be used within the development of a second prototype specifically focusing on the inclusion of odometry.

The implementation worked well in achieving the project development objectives each requirement was met and the final implementation was successful in creating a usable device and acceptable user experience this is supported by the user testing survey.

Project planning worked well in achieving set objectives to meet the project goal however there was some oversight especially in regards to troubleshooting. When building the prototype circuit schematics were not already mapped this lead to problems when implementing each component it would have been more effective to produce circuit schematics before the implementation phase to ensure mapping and cable management are managed effectively.

Within a wider context the project draws parallels within IoT, the data collected from the requirements survey shows the general public are interested in using such a system as an alternative to IoT devices. Mai was developed as a one size fits all approach to remote tasking. Due to the interest in common IoT abilities for home automation. Mais' server could also run a MQTT broker to subscribe to IoT data this can be used for IoT control. Essentially Mai can be adapted as smart home hub to be a robotics application within an IoT context.

The project has been integrated with all of the researchers current studies and specific focus, components are relevant to previously studied modules. Appropriate data structures have been implemented, precise algorithms have been incorporated, a 3D graphics and physics engine (Unity) has been used for the UI and system architecture has been designed with a modular intent to enhance future work and adaptability.

The final artefact presented is free of significant defects however designing such a system became convoluted very quickly having so much to think about in such a short period of time lead to details being overlooked and development being slightly rushed. The approach to future projects and development of another prototype would focus much more on the fine details of each component within the system.

Chapter 10

Future work

Mai has been the first prototype developed, considering observations and results form user testing future work and development needs to be focused on adding depth perception within the VR environment. Further reducing latency within the system is another considerable factor for future work, if latency is handled better the user experience will be greatly improved.

It was identified in user testing that the arm controls need to be more responsive, users often exceeded boundaries for the capture and translation of controller position when extended beyond this boundary the arm inverse kinematics solver returns false for there is no solution and the arm remains static despite the user still moving the controller. Since the user is a layman that is not familiar with the specifics of the application they cannot discern that this is the problem, when users encountered this mitigation was to have the user re-position their arm within parameters.

Odometry and inertia measurement haven't been incorporated within the project due to the various constraints on the project. The use of an IMU to adjust individual wheel RPM would help to decrease drifting from a given velocity. Odometry and encoded DC motors would also be beneficial to return the absolute real-time speed of each motor, this will help to increase accuracy of movement and to identify and mitigate wheel slipping and drifting.

10.1 Future Improvements

- Hardware limiter switches for the lift (currently software limited). This would be more appropriate as if a bug in the software were to be present damage to hardware could still occur.
- A problem was encountered during development where the serial bus could not handle a point cloud stream from the Kinect, this limited the VR interface to a 2D plane on witch the image stream from the Kinect is projected. As of yet the exact problem has not been identified as the USB 3.0 bandwidth should be able to support the Kinect.
- Implementation of a collision avoidance system using an ultrasonic distance sensor to interrupt actuation to mitigate potential damage to hardware.
- A hardware switch for power would be beneficial rather than connecting and disconnecting the battery to the circuit manually.
- Investigation and mitigation of intermittent latency.
- Inclusion of odometry for more precise movement.

References

Angelov, V., Petkov, E., Shipkovenski, G., & Kalushkov, T. (2020). Modern virtual reality headsets. *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 1–5. https://doi.org/10.1109/HORA49412.2020.9152604

Arduino uno r3. (2019). http://cactus.io/platform/arduino/arduino-uno

Bate, A. (2019). *Thermal testing raspberry pi 4*. https://www.raspberrypi.com/news/thermal-testing-raspberry-pi-4/

Bischoff, M. (2020). *Ros*. https://github.com/siemens/ros-sharp/wiki

Components101. (2021). *L298n motor driver module*. https://components101.com/modules/l293n-motor-driver-module

Doelling, K., Shin, J., & Popa, D. O. (2014). Service robotics for the home: A state of the art review. *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*. https://doi.org/10.1145/2674396.2674459

Garcia, E., Jimenez, M. A., De Santos, P. G., & Armada, M. (2007). The evolution of robotics research. *IEEE Robotics Automation Magazine*, *14*(1), 90–103. https://doi.org/10.1109/MRA.2007.339608

Hong, H., Wen, Z., Bi, S., Zhang, Y., & Yang, W. (2019). Roveros: Linking ros with websocket for moblie robot. *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, 626–630. https://doi.org/10.1109/CYBER46603.2019.9066498

Iqbal, M. A., Shah, A., Ammar, F. A., & Rashed Aldaihani, A. (2020). Pre-elicitation processes hierarchy for gsd based requirements elicitation frameworks. *2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, 1–6. https://doi.org/10.1109/ICETAS51660.2020.9484180

Ivan Dryanovski, A. o. l., William Morris. (2015). *Kinect aux package summary*. https://wiki.ros.org/kinect_aux

Lipton, J. I., Fay, A. J., & Rus, D. (2018). Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *IEEE Robotics and Automation Letters*, *3*(1), 179–186. https://doi.org/10.1109/LRA.2017.2737046

Martin, H. (2021). *Libfreenect driver*. https://github.com/OpenKinect/libfreenect

Monica R, A. J. (2022). Evaluation of the oculus rift s tracking system in room scale virtual reality. https://doi.org/https://doi.org/10.1007/s10055-022-00637-3

Onchi, E. (2020). *Cgx-inversek*. https://github.com/cgxeiji/CGx-InverseK

Pigpio daemon. (2021). https://abyz.me.uk/rpi/pigpio/pigpiod.html

pollen-robotics. (2020). *Overall presentation*. https://github.com/pollen-robotics/reachy-2021-docs/blob/master/content/advanced/software/presentation/index.md

Pololu-Corporation. (2022). Drv8825 stepper motor driver. https://www.pololu.com/product/2133

Robotics, O. (2018). *Ros introduction*. http://wiki.ros.org/ROS/Introduction

Sinha, A., & Das, P. (2021). Agile methodology vs. traditional waterfall sdlc: A case study on quality assurance process in software industry. *2021 5th International Conference on Electronics, Materials Engineering Nano-Technology (IEMENTech)*, 1–4. https://doi.org/10.1109/IEMENTech53263.2021.9614779

Smoot, J. (2022). Comparing proximity sensor technologies. https://www.cuidevices.com/blog/comparing-proximity-sensor-technologies

StarVR-Corp. (2020). *Star vr one*. https://www.starvr.com/product/

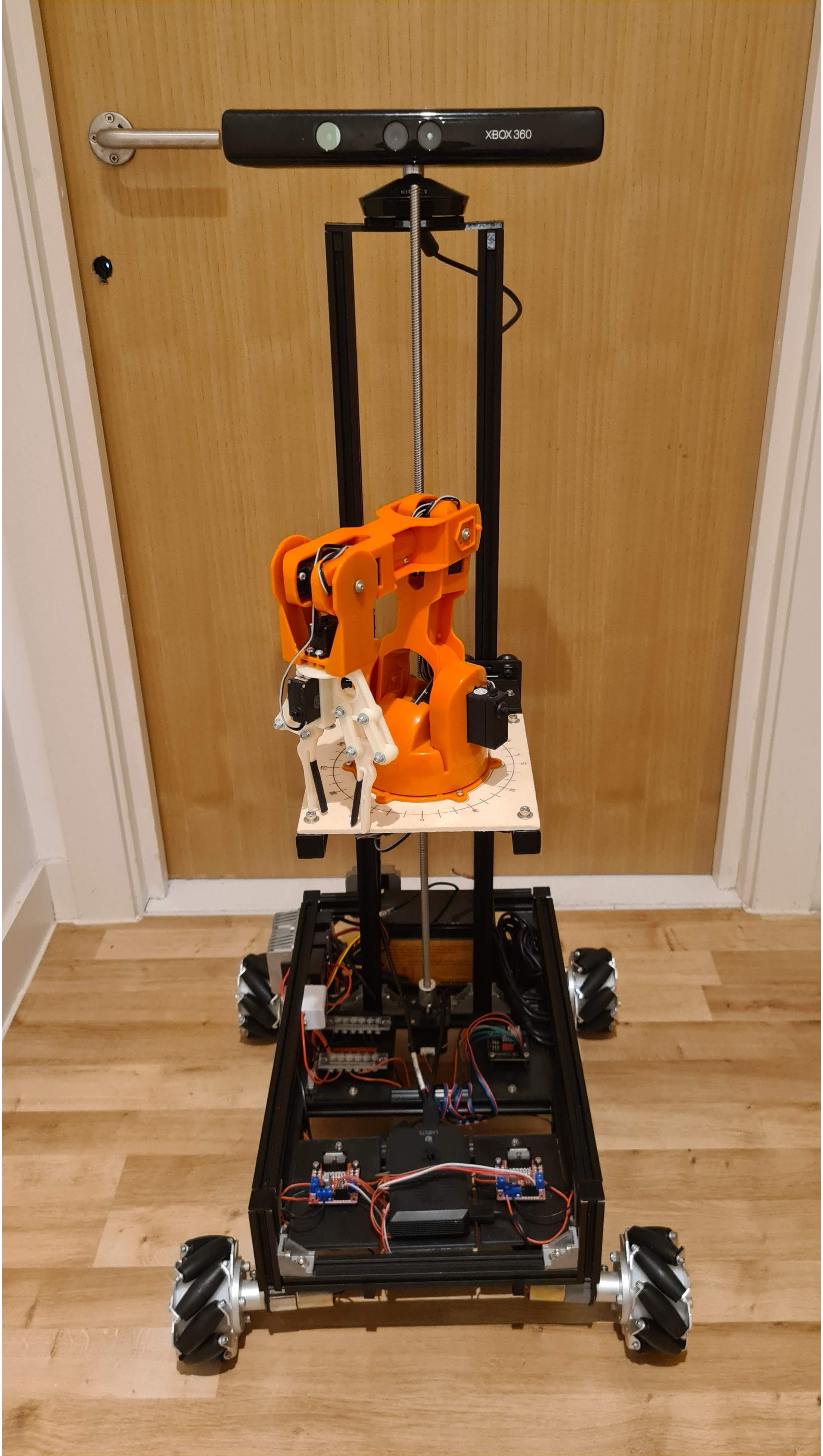
Taheri, H., Qiao, B., & Ghaeminezhad, N. (2015). Kinematic model of a four mecanum wheeled mobile robot. *International journal of computer applications*, *113*, 6–9. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.2034&rep=rep1&type=pdf

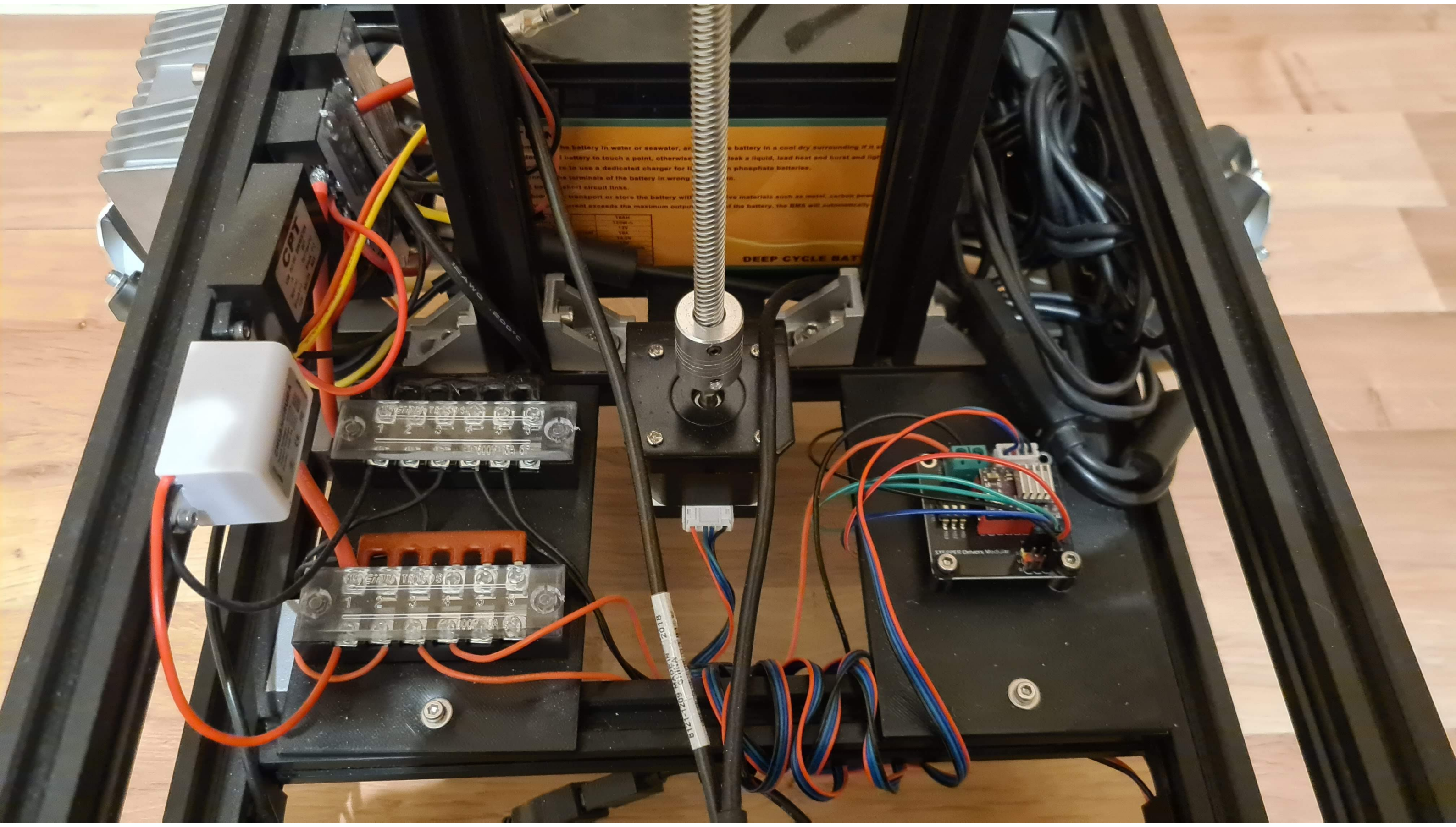
Vijayasarathy, L. R., & Butler, C. W. (2016). Choice of software development methodologies: Do organizational, project, and team characteristics matter? *IEEE Software*, *33*(5), 86–94. https://doi.org/10.1109/MS.2015.26

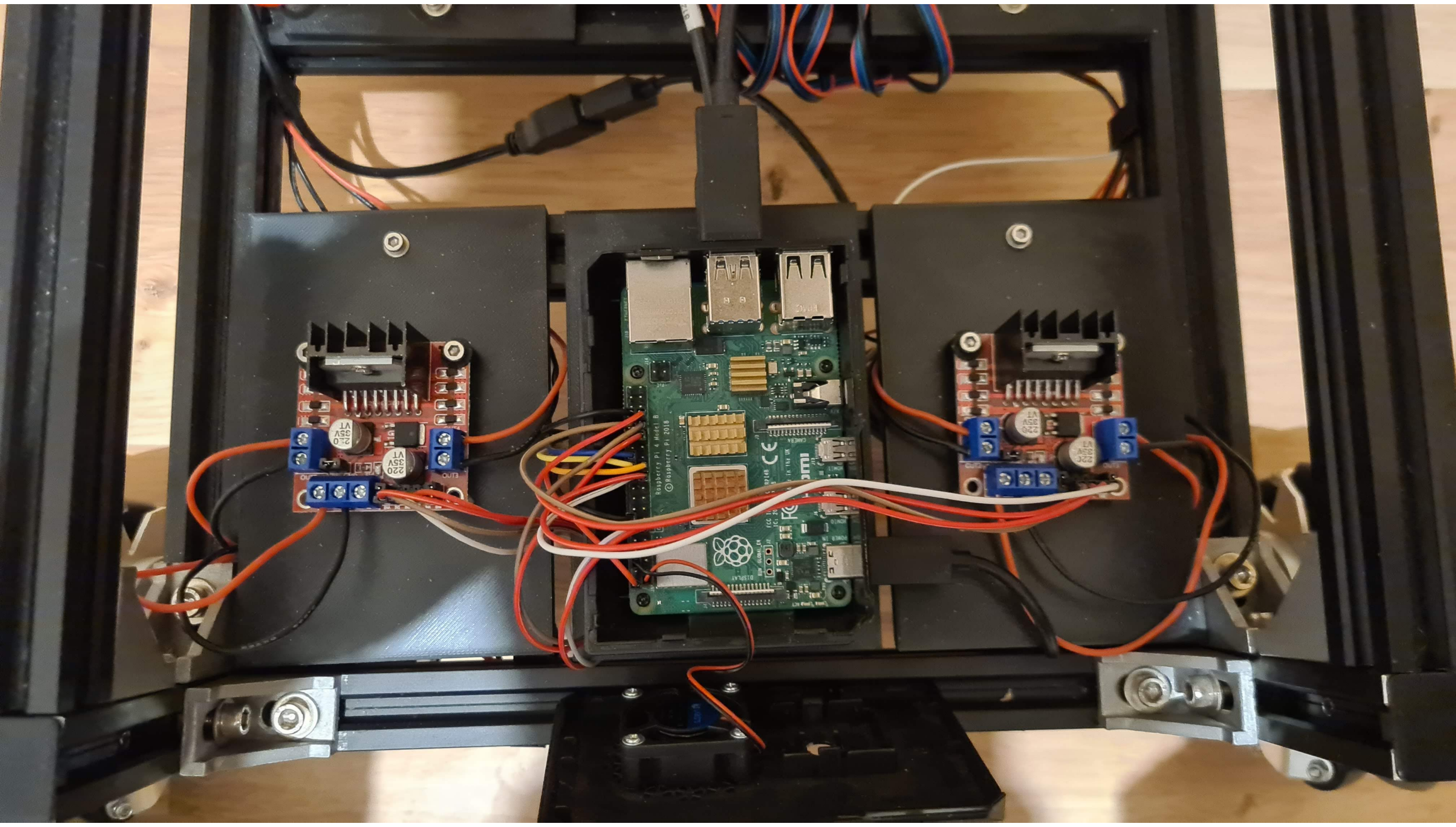
Whitney, D., Rosen, E., Ullman, D., Phillips, E., & Tellex, S. (2018). Ros reality: A virtual reality framework using consumer-grade hardware for ros-enabled robots. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–9. https://doi.org/10.1109/IROS.2018.8593513

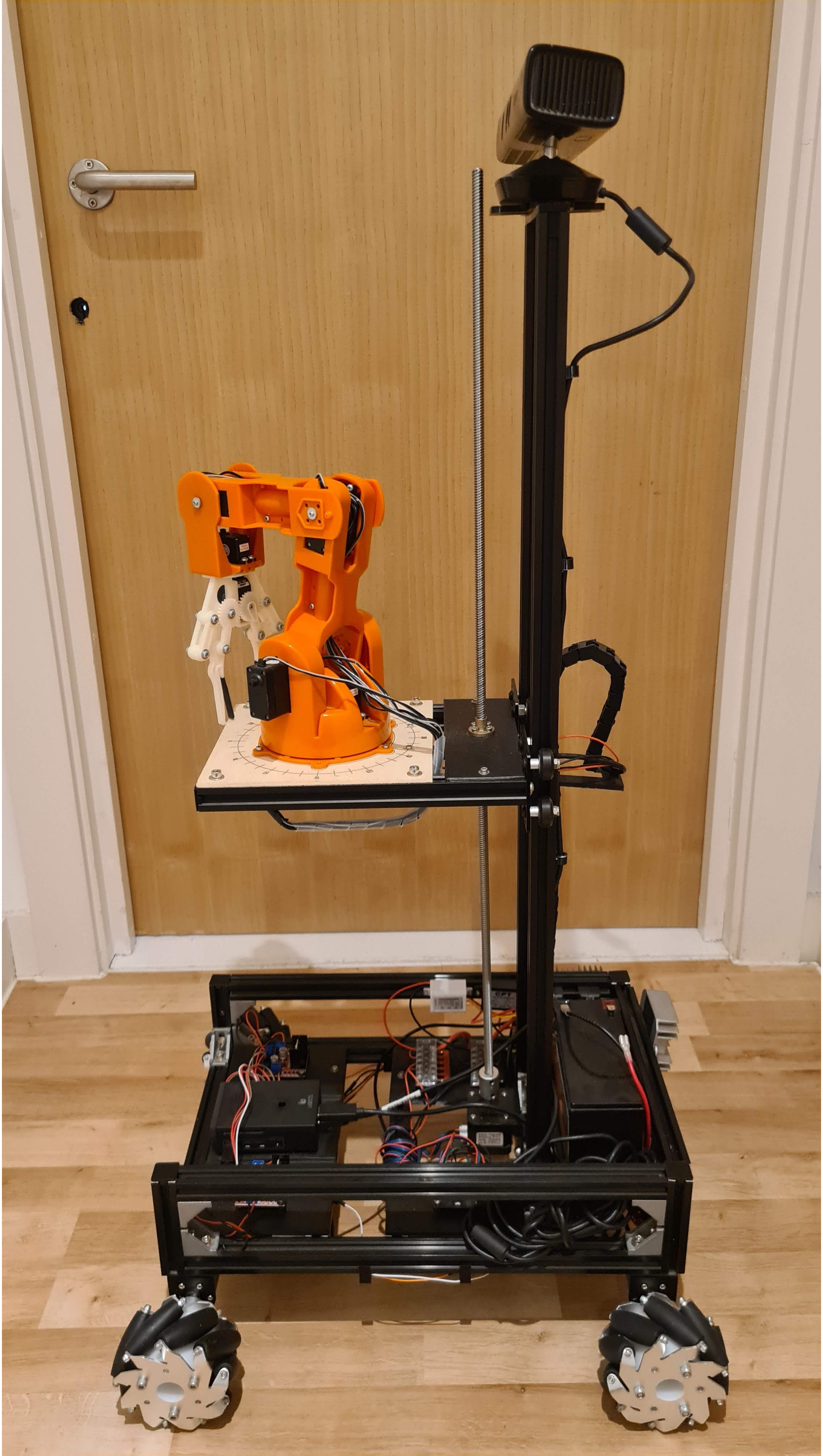
Appendix A

Images of the complete artefact


XBOX 360







Appendix B

Project initiation document

The project initiation document is designed by the university to give students a starting point for final year projects. The first document iteration was created at the initial stage of the projects creation and a final iteration was produced after considerations form research before the design phase.

UNIVERSITY OF PORTSMOUTH

School of Computing
Project Initiation Document

Zak Graham Rackham

Robotic teleoperation using VR

Engineering Project

1. Basic details

Student name:	Zak Graham Rackham
Draft project title:	Robot teleoperation using VR
Course:	Computer Science
Project supervisor:	Zhaojie Ju
Client organisation:	N/a
Client contact name:	N/a

2. Degree suitability

Computer science is the study of algorithmic processes and computational machines. As a discipline computer science is broad and covers many topics. My Project satisfies the criteria as a project for computer science as it implements both hardware and software to solve my problem statement. I believe that my project has significance and is important due to its various potential applications.

3. Outline of the project environment and problem to be solved

The problem I will investigate and attempt to solve is human-robot interaction and teleoperation, how current commercial solutions work, and how a layperson may struggle to interact with a mobile robot. By using VR to control a mobile robot, tasks that require human intuition can be completed easier and in a more natural way (using tracking to reach out and grab an object remotely).

The application for my robot will be simple domestic pick and place operations, however, the principle of VR control can be applied to any robot. For example, for operation in dangerous environments.

4. Project aim and objectives

This project aims to achieve teleoperation of a mobile robot over a local area network (Possibly WAN) using ROS, an Oculus VR system and Unity's XR platform (for compatibility). This includes achieving some domestic task; picking up and moving an object, feeding a pet, anything else the user would like to achieve (Hardware constrained)

My first objective will be to design and prototype a mobile robot using the Arduino and Raspberry pi platforms.

My second objective is to implement basic controls using a command line interface that will register keystrokes to form the basis for controlling my robot. To apply linear and angular Z velocities.

My final objective is to create a virtual interface using VR to control my robot remotely. Controls include but are not limited to, joystick control for linear and angular movement, tracked 5DOF arm control.

5. Project deliverables

- Mobile robot design documentation (using CAD)
- VR interface design documentation
- System design documentation
- Complete hardware & software artifacts
- Mobile robot (omnidirectional base, 5DOF arm, Z-axis lift)
- Project report

6. Project constraints

- Hardware and financial constraints
- Time constraints
- Resources and equipment
- Environmental context

7. Project approach

Establish requirements for both hardware and software, build hardware system, build software system, test (unit, system and user testing) and make final adjustments based on the testing. Finally to produce a working prototype and write a report concurrently.

8. Literature review plan

Using IEEE Xplore & Google scholar
Literature review relevance chart - Excel doc

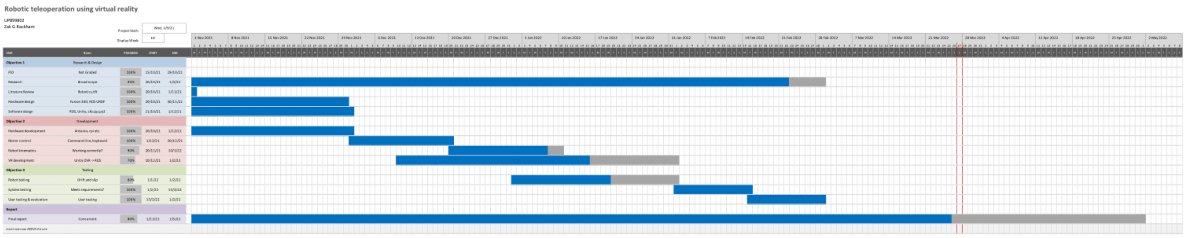
9. Facilities and resources

I require a VR capable desktop computer - personally own
VR headset - personally own
Access to an oscilloscope may be beneficial
Level ground for setup of omni-directional mecanum wheel configuration
Hardware tools

10. Log of risks

Description	Impact	Likelihood	Mitigation	First indicator
COVID-19 outbreak means I cannot get into a lab for usability testing	*Severe*	*Likely*	*Get in while I can, prioritise lab tasks in time.* *Make an alternate test plan that does not need the lab.*	*University informs that lab closure is likely*
Hardware malfunction & damage	Severe	Likely	Careful consideration before applying transformations to robot hardware, Software backups, redundant hardware.	Not behaving as expected, DC motor damaged.
Time management	Severe	Possible	Careful use and planning of time allocated to the project .	Behind Gantt chart schedule
Availability	Severe	Likely	Supply chain analysis. The availability of labs, Home office and laptop for Hybrid working.	News sources indicating shortages. Updates from retailers and vendors.
Experience	Severe	Likely	Careful planning and allocation of time for research and learning.	Beyond a reasonable amount of time to achieve an objective.

11. Project plan



12. Legal, ethical, professional, social issues (mandatory)

Legalities

When approaching the legalities of my project the most appropriate aspects in law to consider would be; the usage of licenced open source software and the accountability of the robot operator (the acknowledgement and assumption of responsibility of the user). For example if the system is to be held accountable (system error or malfunction) for the robots effect in a physical environment. Or whether the user is to be held accountable (misuse of the system).

Ethical considerations

Analysing the ethical aspects of the system is subjective to an individuals specific views and opinions, ethics are best analysed by more than one body or individual. Therefore I will present my interpretation on the ethics of my project that will be reviewed by the ethics committee of the university of portsmouth.

My stance and considerations of my projects ethics are based on my own moral principles that will govern the project these are as follows:

- My project must not harm or allow any human to be harmed during its creation and operation.
- Due to the remote operation of complex hardware my project must not allow the creation of an artifact that can cause detriment to its surrounding environment.
- My project must not cause harm to our planet's environment (Impact of hardware must be carefully considered) or such damage must be mitigated where possible.

Social considerations

When considering the implications of my project on society the points detailed below are the most important.

- Will the outputs of my project cause detriment or benefit to society?
- How may the outputs of my project affect the function of both our local society and societies and cultures around the world.

Improving our world through the development of new technologies is an objective of my project, the outputs of the project will be designed to benefit both society and individuals across the globe.

Appendix C

Ethics Review

Certificate of Ethics Review

Project title: Robotic teleoperation using virtual reality

Name:	Zak Rackham	**User ID**:	899802	**Application date**:	25/01/2022 16:59:43	**ER Number**:	TETHIC-2022-102334

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative(s) for the **School of Computing** is/are **Haythem Nakkas, David Williams**

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

Which school/department do you belong to?: **School of Computing**
What is your primary role at the University?: **Undergraduate Student**
What is the name of the member of staff who is responsible for supervising your project?: **Zhaojie Ju**
Is the study likely to involve human subjects (observation) or participants?: Yes
Will you gather data about people (e.g. socio-economic, clinical, psychological, biological)?: Yes
Will you gather data from people about some artefact or research question (e.g. opinions, feedback)?: Yes
Will the study involve National Health Service patients or staff?: No
Do human participants/subjects take part in studies without their knowledge/consent at the time, or will deception of any sort be involved? (e.g. covert observation of people, especially if in a non-public place): No
Will you collect or analyse personally identifiable information about anyone or monitor their communications or on-line activities without their explicit consent?: No
Does the study involve participants who are unable to give informed consent or are in a dependent position (e.g. children, people with learning disabilities, unconscious patients, Portsmouth University students)?: Yes
Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants?: No
Will blood or tissue samples be obtained from participants?: No
Is pain or more than mild discomfort likely to result from the study?: No
Could the study induce psychological stress or anxiety in participants or third parties?: No
Will the study involve prolonged or repetitive testing?: No
Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No
Are there risks of significant damage to physical and/or ecological environmental features?: No
Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: No
Does the project involve animals in any way?: No
Could the research outputs potentially be harmful to third parties?: No
Could your research/artefact be adapted and be misused?: No
Does your project or project deliverable have any security implications?: No

I confirm that I have considered the implications for data collection and use, taking into consideration legal requirements (UK GDPR, Data Protection Act 2018 etc)
I confirm that I have considered the impact of this work and and taken any reasonable action to mitigate potential misuse of the project outputs

I confirm that I will act ethically and honestly throughout this project

Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy.

Supervisor's signature: Zhaojie Ju Date:

Faculty Ethics Committee Review

Faculty Ethics Committee Member's signature: Date:

Appendix D

Requirement survey results

Timestamp	I consent that the data collected through this survey will be used solely for the purpose of the project and will remain anonymous.	How old are you?	Do you own a virtual reality headset?
11/02/2022 17:46:20	I consent	24	No
11/02/2022 17:47:31	I consent	50	No
11/02/2022 17:54:49	I consent	23	No
11/02/2022 17:56:08	I consent	22	No
11/02/2022 17:58:06	I consent	21	No
11/02/2022 18:05:35	I consent	22	No
11/02/2022 18:10:25	I consent	21	No
11/02/2022 18:44:01	I consent	51	No
11/02/2022 19:08:12	I consent	31	No
11/02/2022 19:09:55	I consent	20	No
11/02/2022 19:12:15	I consent	58	No
11/02/2022 19:12:43	I consent	48	No
11/02/2022 19:24:42	I consent	60	No
11/02/2022 19:34:26	I consent	46	No
11/02/2022 19:37:49	I consent	53	No
11/02/2022 19:48:23	I consent	57	No
11/02/2022 20:08:56	I consent	56	No
11/02/2022 20:43:22	I consent	21	No
11/02/2022 20:56:55	I consent	15	No
11/02/2022 21:18:13	I consent	26	Yes
11/02/2022 21:20:25	I consent	21	No
11/02/2022 23:29:34	I consent	23	No
11/02/2022 23:48:01	I consent	28	No
12/02/2022 08:20:51	I consent	22	No
12/02/2022 10:42:31	I consent	24	No
12/02/2022 22:31:51	I consent	28	No
14/02/2022 07:26:05	I consent	52	No
14/02/2022 10:31:17	I consent	26	No
14/02/2022 19:30:08	I consent	22	No
16/02/2022 10:00:50	I consent	23	No
16/02/2022 10:01:21	I consent	Twenty one	No
16/02/2022 10:01:39	I consent	41	No
16/02/2022 10:13:39	I consent		Yes
16/02/2022 10:22:00	I consent	18	No
16/02/2022 10:22:58	I consent	22	No
16/02/2022 10:29:52	I consent	53	No
16/02/2022 10:33:22	I consent	26 years	Yes
16/02/2022 10:35:48	I consent	32	No
16/02/2022 10:38:12	I consent	36	No
16/02/2022 10:45:19	I consent	42	No
16/02/2022 11:08:16	I consent	27	No
16/02/2022 11:27:42	I consent	21	No
16/02/2022 11:37:29	I consent	24	No
16/02/2022 11:37:42	I consent	36	No
16/02/2022 11:42:51	I consent	22	Yes
16/02/2022 11:44:01	I consent	20	Yes
16/02/2022 11:46:28	I consent	18	No
16/02/2022 11:49:27	I consent	19	No
16/02/2022 11:50:36	I consent	50	
16/02/2022 11:53:24	I consent	54	Yes
16/02/2022 11:55:53	I consent	32	No
16/02/2022 11:59:03	I consent	20	No
16/02/2022 12:24:03	I consent	34	No
16/02/2022 13:07:41	I consent	20	Yes
16/02/2022 13:11:13	I consent	18	No
16/02/2022 13:27:14	I consent	19	No
16/02/2022 13:35:00	I consent	22	Yes
16/02/2022 13:55:04	I consent	18	No
16/02/2022 14:02:00	I consent	19	Yes
16/02/2022 14:13:33	I consent	22	
16/02/2022 15:38:07	I consent	18	Yes
16/02/2022 19:34:27	I consent	21	Yes
16/02/2022 19:35:42	I consent	21	Yes
16/02/2022 22:59:05	I consent	34	No
17/02/2022 05:59:35	I consent	37	No
17/02/2022 08:27:07	I consent	51	No
17/02/2022 10:16:05	I consent	21	No
17/02/2022 11:03:55	I consent	25	Yes
17/02/2022 11:10:10	I consent	20	Yes
17/02/2022 18:52:55	I consent	47	No
20/02/2022 05:28:06	I consent	19	No
20/02/2022 10:21:33	I consent	37	No
21/02/2022 07:49:48	I consent	44	Yes
21/02/2022 20:34:12	I consent	53	No
23/02/2022 14:10:49	I consent	23	Yes
04/03/2022 11:37:53	I consent	20	No

Have you ever needed to complete a task at home but you're not there?	If yes, what task or set of tasks would you identify first?
Yes	Washing clothes
Yes	Putting the oven on, the list is endless.
No	
Yes	Furnishing and building furniture
Yes	Washing up, washing clothes
Yes	Collecting parcels, feeding pets
Yes	Clean, laundry
Yes	Feeding the dog
No	
Yes	cleaning, cooking, organising
No	
No	
No	
No	
No	
Yes	Washing
Yes	Washing, receiving a delivery
Yes	Turning on the heater, wash the plates, wash clothes
Yes	Checking if I turned off the straighteners
Yes	anything from seeing if i have something to checking if i left something on and if so turning it off
Yes	Take out laundry out of the washing machine
Yes	Feed pet
No	
Yes	Hang the washing out/get it in
Yes	Needing a document i.e. Passport Number for certain forms. Also doing typical day-to-day tasks such as turning lights off when forgot or forgetting to take out the rubbish on bin day.
Yes	Turn the heating on, change the bed sheets, put a load of washing on
Yes	Hang the washing out
Yes	Turn the kettle on
Yes	The clothes washing, vacuuming
No	
Yes	Put the laundry on
No	
No	
No	
Yes	Start the washing machine
No	
Yes	
No	
Yes	
Yes	Heat refrigerated food, switch lights on
No	
No	
Yes	
Yes	Turn of heating or lights
No	
Yes	Computer oporation, turning on/setting up remote access
No	
Yes	Mostly pushing to GIT, transferring files, putting laundry on, etc.
No	
Yes	Turn on lights and heating
No	
No	
Yes	House cleaning
Yes	Control over a computer, Move stuff around and general maintinence
No	
No	
Yes	Turn the heating on, or other appliances such as an oven.
Yes	Turn on heating, turn off light in rooms
No	
Yes	
Yes	cook food
No	
Yes	Cooking, laundry , cleaning, do my uni tasks
No	
Yes	
Yes	Turning off heating
Yes	Ensuring the flats' front door is shut; Turning off the oven / hob.
No	
Yes	Physically rewiring several tower computers
No	
No	
No	
No	
No	
Yes	wiriting
Yes	Project/Homework

Have you ever operated a robot before?	Do you have any concerns about home robotics?
No	No
No	No
Yes	
No	No
No	Not particularly, a fault leading to a fire or something maybe?
No	Listening to my every move, privacy, getting hacked
Yes	Security
No	Security- can someone unauthorised access control?
No	Yes, people becoming lazier
No	hacking and security liabilities
No	No
Yes	No
No	No
No	Yes. If it went wrong
No	
No	No
No	No
No	Is it easy to become prone to hacking? What is the security like ?
No	No
No	spying
No	High energy consumption
No	Not particularly, maybe ease of use
No	Nope
No	What if it malfunctions and explodes or starts trashing everything
Yes	In general, a big concern amongst the general public is how the collected data will be used, if the robot has built in microphones etc.
No	Yes, the robot doing something it should not
No	No
Yes	No
No	If I asked it to make me dinner for example, I'd be concerned that my house may be burnt to a crisp by the time I get home
No	No
Yes	Yes
No	No
Yes	Obvious security concerns. (Other people being able to hack in and have control. Other people being able to see what the robot's camera(s) see)
Yes	I guess there are potential concerns about security and privacy and aspects like that which will need addressing, although I believe home robotics can work well, and can help people with specific tasks. I guess on that note another concern I have is that home robotics might stop people from developing some skills around their home, and potentially rely too much on the robots. Hopefully that's a good enough answer in the context of the question.
No	Yes, that someone could hack into the home robot and maybe access sensitive information?
No	Yes. my concern is that robotics can only do a partial functionality
No	No
No	
No	
No	There are always unperceived risks, like in case of a natural disaster how will the robot behave
No	Is it safe, like what if someone else could access it
No	Pricing, Usability and functionality
Yes	It may malfunction and break something
Yes	Yes. Trust and confidence
No	
Yes	Security and unuthorised access
Yes	
No	No more than I do IOT and Virtual Assistants.
Yes	No (though don't have any robots at home!).
No	
Yes	No
No	Primary concern would be remote use of home robots - if its wirelessly connected over the internet it can and will be broken into from outside
No	No
Yes	some but not much depending on whats controlling them
No	The average consumer has awful security on their home Wifi that can be easily exploited. If a robot was on this home wifi it would be a target for attack.
No	
No	Price?
No	is my location being tracked? , Am I bring heard through microphones?
Yes	Sometimes paranoid that they could be listening to my conversations
No	
Yes	no
No	No
No	No
No	I am interested in home automation
No	
No	I haven't read much on them
Yes	Yes, crashing them into walls
No	Yes
Yes	Have you seen *any* distopian futures where the robotic blinds don't evolve into highly effective killing machines?
Yes	no
No	Security and Safety
No	I probably wouldn't be able to work them.
No	
No	Not really
Yes	no
No	Security

Have you used a joystick controller before?
Yes
Yes
Yes
Yes
Yes
No
Yes
Yes
Yes
Yes
Yes
Yes
No
No
No
No
Yes
Yes
No
Yes
Yes
Yes
Yes
Yes
Yes
No
Yes
Yes
Yes
No
Yes
Yes
Yes
Yes
Yes
Yes
No
Yes
Yes
Yes
No
Yes
Yes
Yes
No
Yes
Yes
Yes
Yes
Yes
Yes
Yes
No
Yes
Yes
Yes
Yes
Yes
Yes
No
Yes
Yes
Yes
Yes
Yes
No
Yes
No
No
Yes
Yes
No
Yes
No
Yes
Yes

Appendix E

User testing survey results

Timestamp	How old are you?	How would you rate the controls schematic in terms of ease of use?	How would you rate the arm motion controls in terms of ease of use?	Did you encounter any motion sickness?	Did you experience any latency (delay) in operation?	Please rate your overall experience	Do you have any suggestions or comments for an improved experience?	What task did you find most challenging?
26/02/2022 14:07:08	18	8 (Quite easy)	8 (Quite easy)	No	Yes (Minor)	8		Turning on the kettle
26/02/2022 14:07:35	50	10 (Extremely easy)	10 (Extremely easy)	No	No	10	None at all, extremely user friendly	Turning on the kettle
26/02/2022 14:08:22	51	5 (mediocre)	5 (mediocre)	No	Yes (Minor)	7	My arm extension to match robot arm extension	Turning on the kettle
05/03/2022 15:45:56	23	8 (Quite easy)	8 (Quite easy)	No	Maybe/unsure	4		Picking and placing items
12/03/2022 16:26:51	23	10 (Extremely easy)	6 (Somewhat easy)	No	Yes (Minor)	8		Turning on the kettle
12/03/2022 16:29:03	22	9 (Very easy)	8 (Quite easy)	No	Yes (Minor)	7	Maybe try and get the camera to not have the red vision overlay	Turning on the kettle
13/03/2022 18:56:52	24	8 (Quite easy)	9 (Very easy)	No	Yes (Minor)	7	Reducing latency	Picking and placing items

Appendix F

Participant information sheets

School of computing
Email: up899802@myport.ac.uk

PARTICIPANT INFORMATION SHEET

Title of Project: Robotic teleoperation using virtual reality

Name and Contact Details of Researcher(s): Mr. Zak G Rackham

Email: up899802@myport.ac.uk

Name and Contact Details of Supervisor: Professor Zhaojie Ju

Email: zhaojie.ju@port.ac.uk

Ethics Committee Reference Number: TETHIC-2022-102334

I would like to invite you to take part in end user testing for my research study. Joining the study is entirely up to you, before you decide I would like you to understand why the research is being done and what it would involve for you. I'll go through this information sheet with you, to help you decide whether you would like to and are able to participate and to answer any questions you may have. This should take approximately 10 minutes. Please feel free to talk to others about the study if you wish to. Do ask if anything is unclear.

Taking part in this research is entirely voluntary. It is up to you to decide if you want to volunteer for the study. I will describe the study in this information sheet. If you agree to take part, I will then ask you to sign the attached consent form, dated Febuary 2022, version number 2.

This study is concerned with human-robot interaction which has become increasingly important due to the advancements in applied robotics. I'm seeking healthy participants both male and female between the ages of 18-60 years. Participation in the research would require you to attend in person and will take approximately 30 minutes of your time.

The purpose of this study is to better evaluate control methods in service robotics, as part of the study I have designed a mobile service robot called Mai that can be operated using virtual reality. This study is designed to evaluate the effectiveness of such a control system by the method of user testing.

The researcher will provide you with a pre-configured scene, within this scene you will be asked to perform a series of short tasks. During each task you will be timed by the researcher. Each task is to be completed twice by the participant, once in the real world and again virtually using Mai. The only data being collected in this study will be the time to complete each task along with an anonymous questionnaire at the end.

There are very few risks to taking part, mild discomfort (motion sickness) may be experienced by some. If you experience discomfort of any kind, please let the researcher know and the appropriate action will be taken.

You will not receive any direct personal benefits from participating in this study, but society (or a sub-group of society) may benefit from the results of this work.

Anonymous data, that which does not identify you, may be publicly shared at the end of the project and made open access. A CC-BY licence will be applied to this publicly shared data. This will allow anyone else (including researchers, businesses, governments, charities, and the general public) to use the anonymised data for any purpose that they wish, providing they credit the University and research team as the original creators. No restrictions will be placed on this shared anonymised data limiting its reuse to non-commercial ventures only.

School of computing
Email: up899802@myport.ac.uk

As a volunteer you can stop your participation at any time, or withdraw from the study at any time, without giving a reason if you do not wish to. If you do withdraw from a study after some data has been collected this data will still be included, data is being anonymised so therefore it cannot be identified and withdrawn. Once the research has been completed, and the data analysed, it will not be possible for you to withdraw your data from the study.

If you have a query, concern or complaint about any aspect of this study, in the first instance you should contact the researcher(s) if appropriate. If the researcher is a student, there will also be an academic member of staff listed as the supervisor whom you can contact. If there is a complaint and there is a supervisor listed, please contact the Supervisor with details of the complaint. The contact details for both the researcher and supervisor are detailed on page 1.

If your concern or complaint is not resolved by the researcher or their supervisor, you should contact the Head of Department:

School of computing

The Head of Department: Mrs Petronella Beukman

Email: petronella.beukman@port.ac.uk

Portsmouth

PO1

If the complaint remains unresolved, please contact:

The University Complaints Officer

023 9284 3642   complaintsadvice@port.ac.uk

This research is being funded by the primary researcher. None of the researchers or study staff will receive any financial reward by conducting this study, other than their normal salary / student loan.

Research involving human participants is reviewed by an ethics committee to ensure that the dignity and well-being of participants is respected. This study has been reviewed by the Faculty of Technology Ethics Committee and been given favourable ethical opinion.

Thank you

Appendix G

Participant Consent Form

CONSENT FORM

Title of Project: Robotic teleoperation using virtual reality

Name and Contact Details of Researcher(s): Mr. Zak G Rackham

Email: up899802@myport.ac.uk

Name and Contact Details of Supervisor (if relevant): Professor Zhaojie Ju

Email: zhaojie.ju@port.ac.uk

University Data Protection Officer: Samantha Hill, 023 9284 3642 or information-matters@port.ac.uk

Ethics Committee Reference Number: TETHIC-2022-102334

	Please initial box
1. I confirm that I have read and understood the information sheet dated January 2022 (version 2) for the above study. I have had the opportunity to consider the information, ask questions and have had these answered satisfactorily.	☐
2. I understand that my participation is voluntary and that I am free to withdraw at any time without giving any reason.	☐
3. I understand that data collected during this study is anonymised and cannot be associated with personally identifiable information as explained in the participant information sheet version 2.0 March 2022.	☐
4. I agree to take part in the above study.	☐

Name of Participant:	**Date:**	**Signature:**
Name of Researcher:	**Date:**	**Signature:**

Note*: When completed, one copy to be given to the participant, one copy to be retained in the study file*