

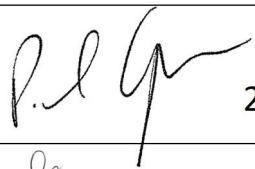



Final Report

Haptic Robot Teleoperation 5G Network Feasibility Study

Mar 2022

Issue Details

	Name(s)	Signature
Principal author	Emiel Bos	 24 th March 2022
Internal reviewer	Jonathan Butt	 24 th March 2022
External reviewer	Paul Cooper	 29 Mar 2022
Internal authorisation	Sachin Sree Raman	 28 th March 2022
Document number	DETI-PoC4-013	
Document issue	1.0	
Issue date	29/03/2022	

Document Issue Log

1.0	Initial version

ABOUT 5G-ENCODE

The 5G-ENCODE Project is a £9Million collaborative project aiming to develop clear business cases and value propositions for 5G applications in the manufacturing industry. The project is partially funded by the Department for Digital, Culture, Media, and Sport (DCMS), of the UK government as part of their 5G Testbeds and Trials programme. The project is one of the UK's biggest investments in using 5G to modernise manufacturing.

The key objective of the 5G-ENCODE project is to demonstrate the value of 5G as part of industrial use case delivery within the composites manufacturing industry. It is designed to validate the idea that using private 5G networks in conjunction with new business models can deliver better efficiency, productivity, and a range of new services and opportunities that would help the UK lead the development of advanced manufacturing applications.

The project will play a key role in ensuring that the UK industry exploits the 5G technology and remains a global leader in the development of robust digital engineering capabilities when implementing complex composites manufacturing processes.

The project will highlight how 5G features such as network slicing and network virtualisation can be applied to transform a private 5G network into a dynamically reconfigurable network able to support a wide range of applications (URLLC/eMBB/MMTC) including industrial applications of Augmented Reality/Virtual Reality (AR/VR), asset tracking of time sensitive materials and automated industrial control through IoT monitoring and big data analytics. Such a dynamic network would enable new business models and creation of bespoke virtual networks tailored to specific applications or use cases.

A state-of-the-art test bed was deployed across three sites centred around the National Composites Centre in the Southwest of England. In support of the West of England Combined Authority (WECA) industrial strategy, the NCC plans to keep the test bed as an open access facility for the experimentation and development of new products and services for the composites industry after the completion of the 5G-Encode project. The location and nature of NCC's business would ensure the creation of an industrial 5G ecosystem involving multiple industry sectors and SMEs.

The project consortium, led by Zeetta Networks, brings together leading industrial players (e.g., Siemens, Toshiba, Solvay), a Tier 1 operator (Telefonica), disruptive technology SMEs covering all aspects of network design, deployment, and applications (Zeetta Networks, MatiVision, Plataine), a world-leading 5G network research group (High Performance Networks Group in the University of Bristol) and the NCC representing the high value manufacturing industry.

EXECUTIVE SUMMARY

In this work, haptic robot control protocols were integrated into a teleoperation solution where an industrial robot located at the Bristol Robotics Laboratory is operated from the National Composites Centre. The need for robot teleoperation within the industry remains for use-cases of which the environment is too hazardous for direct human operation and too complex for automation.

For further steps to be made in robot teleoperation there need to be intuitive systems designed that allows an operator to control the robot with confidence. A system is proposed that uses hand tracking, haptic feedback, and an immersive experience to actuate a robotic arm from a distance for which optical fibre and network cables are used to facilitate the network needs.

The robotic actuation using the combination of a haptic hand tracking device and a haptic protocol that allows the operator to move the robot by moving his/her hand. The immersive experience using a stereo camera to capture the robot's environment and a Virtual Reality (VR)-headset to make this content visible for the operator. The teleoperation is cabled to function as a benchmark in this feasibility study.

The network consists of a combination of optical fibre, CAT 6 Ethernet and universal serial bus cables. The main connections within the network are limited to a 1 Gbps maximum throughput. During testing of the solution it was measured that the maximum data throughput was 27 Mbps. The current 5G networking capability allows for an uplink of 57 Mbps and a downlink of 410 Mbps.

With this it can be concluded that 5G could be integrated within the current solution. To limit the potential project risk, an initial study was undertaken where the operation of a robot arm was completely virtual proving that haptic control of an industrial robot arm is indeed feasible. This time around project risk was mitigated by first testing the network and the solution using a cable, proving the feasibility of 5G in a short time frame.

TABLE OF CONTENTS

About 5G-Encode	ii
Executive Summary	iv
Table of Contents.....	v
List of Figures.....	vii
List of Tables	ix
Abbreviations.....	x
1. Introduction.....	1
1.1 Motivation.....	3
1.2 Aims & Objectives	5
2. Solution selection.....	6
2.1 Haptic robot control.....	6
2.1.1 Kinect	7
2.1.2 Data gloves.....	8
2.1.3 Ultraleap motion controller	9
2.1.4 Robot teleoperation method	10
2.2 Streaming camera.....	12
2.3 VR Headset.....	13
3. Technical Description.....	15
3.1 Haptic Protocols	16
3.2 System interfacing.....	18
3.3 Haptics System	18
3.3.1 Robot system	23
3.3.2 Immersive system.....	25
3.4 Application.....	28
3.4.1 Main screen.....	28
3.4.2 Menu Screen.....	30
3.4.3 Controls	31
3.4.4 No-Camera teleoperation	31
3.4.5 Camera teleoperation	32
3.4.6 Virtual reality teleoperation	33

3.5	Network.....	34
3.5.1	Current network architecture	34
3.5.2	Consideration for using 5G	36
4.	Results.....	39
5.	Conclusion	42
	Bibliography	44
	Appendix	46

LIST OF FIGURES

Figure 1: Project enablers.....	2
Figure 2: Project collaborators.....	2
Figure 3: Bristol Robotics Laboratory teleoperation	3
Figure 4: KUKA Nuclear decommissioning solution	4
Figure 5: From left to right: Kinect v1, Kinect v2, Kinect v3 [9]	7
Figure 6: Data gloves – HaptX [8] & YoBu [7] (left to right)	9
Figure 7: Ultraleap motion controller [10].....	9
Figure 8: Ultraleap motion controller	9
Figure 9: UltraHaptics Stratos Explore	11
Figure 10: Camera options, Suometry system & ZED camera (left to right)	12
Figure 11: VR-Headset options, Oculus Quest 2 & HTC Vive (left to right)	13
Figure 12: Haptic Control Set-up.....	15
Figure 13: Ultraleap Devices, Motion controller & Stratos Explorer (left to right).....	16
Figure 14: PTV Control [1]	17
Figure 15: Pitch Roll Yaw diagram.....	17
Figure 16: Leap Motion cartesian system.....	19
Figure 17: UltraHaptics mid-air tactile sense.....	20
Figure 18: Hand portal in-session annotation.....	21
Figure 19: Hand portal architecture diagram.....	21
Figure 20: Hand controller finite state machine architecture diagram.....	22
Figure 21: Move target object	23
Figure 22: Franka Emika Panda robot	23
Figure 23: Franka Emika Panda robot inverse kinematics set-up.....	24
Figure 24: Bristol Robotics Laboratory set-up	25
Figure 25: Oculus VR-Headset	27
Figure 26: OpenXR	27
Figure 27: Scene overview diagram.....	28
Figure 28: Main screen	28
Figure 29: Connection button flow chart.....	29
Figure 30: Camera screen & Debug screen (left to right)	29
Figure 31: Menu screen	30
Figure 32: Keyboard controls	31

Figure 33: No-Camera teleoperation	32
Figure 34: Camera teleoperation	32
Figure 35: VR teleoperation menu screen	33
Figure 36: VR teleoperation camera feed.....	34
Figure 37: Network diagram.....	34
Figure 38: Network diagram max bandwidth.....	36
Figure 39: 5G-Network diagram max bandwidth.....	37
Figure 40: Network diagram required bandwidth.....	41
Figure 41: NetPerSec baseline.....	46
Figure 42: NetPerSec application idle	46
Figure 43: NetPerSec application camera	47
Figure 44: NetPerSec robot moving no camera	47
Figure 45: NetPerSec robot moving camera.....	48
Figure 46: NetPerSec robot not moving camera	48
Figure 47: NetPerSec robot not moving no camera.....	49

LIST OF TABLES

Table 1: Haptic protocols per device [1].....	6
Table 2 : Network table theoretical maximum bandwidth.....	35
Table 3: 5G-Network Throughputs.....	36
Table 4: Location and activity description	38
Table 5: NetPerSec network bandwidth	40
Table 6: Ping latency.....	41

ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
3G	Third Generation Mobile Network
4G LTE	Fourth Generation Long-Term Evolution Mobile Network
5G	Fifth Generation Mobile Network
AP	Access Point
APC	Automated Pre-Forming Cell
AR	Augmented Reality
CPE	Customer Premises Equipment
DCMS	Department for Digital, Culture, Media, and Sport
DL	Downlink
eMBB	Enhanced Mobile Broadband
IoT	Internet of Things
IP	Internet Protocol
Kbps	Kilobits Per Second
KPIs	Key Performance Indicators
LRI	Liquid Resin Infusion
Mbps	Megabits Per Second
MMTC	Massive Machine-Type Communications
NCC	National Composites Centre
NCC HQ	National Composites Centre Headquarters
NCCi	National Composites Centre – Filton Site
NR	New Radio
NTP	Network Timing Protocol

P2P	Peer-to-Peer
PTP	Precision Timing Protocol
RAN	Radio Access Network
SA	Standalone
SMEs	Small and Medium-Sized Enterprises
TCP	Tool Centre Point
UEs	User Equipment
UL	Uplink
UoB	University of Bristol
UPF	User Plane Function
URLL	Ultra-Reliable Low Latency
VMs	Virtual Machines
VR	Virtual Reality
WECA	West of England Combined Authority

1. INTRODUCTION

Robot teleoperation is used in numerous industries, this is currently often conducted through control pendants that require extensive training before it can be safely operated. In this project an alternative solution for the teleoperation of an industrial robotic arms has been trialled.

The need for robot teleoperation within the industry ranges far and wide. The use-cases for this research project are focussed on the use of an industrial 6-DoF (Degree of Freedom) robotic arm. These industrial robotic arms come with extensive safety protocols. An example of teleoperation of robotic arm use is when the environment proves to be too hazardous for a human being and too challenging and/or too perilous for an automated system. Industries where this may occur include nuclear decommissioning and industrial welding.

The current needs of robot teleoperation are met using a number of different solutions. Typical solutions use a 3D mouse attached to a control pendant which is directly controlled, often through cable, to its respective robotic system. Such a pendant allows the operator to either steer the Tool Centre Point (TCP) in Cartesian coordinates (in world or local axis) or the individual joints of the robot. This form of operation is conventionally done with the operator in the vicinity of the robotic system keep a good overview of the movement. The conventional pendant solution, therefore, is not usable when operation is required in a hazardous environment that requires active control.

Two challenges can immediately be identified that need to be surpassed to take the next step in industrial robot teleoperation. The first challenge is to develop an intuitive method for active actuation of the robot arm during operation such that the operator can perform all required operations with confidence. The second challenge relates to the situational awareness of the operator during robot teleoperation. The operator needs to be aware of what is always happening and the options they have available to update each operation. The first challenge was solved in a previous project [1] where research has been done on how intuitive different methods of haptic teleoperation are, including hard and soft measures. This project builds upon the haptic research conducted before and adds direct

active robot control over an extended distance, whereas this time, the second challenge is tackled through an immersive solution and a direct cabled teleoperation network.

Multiple collaborators are needed to complete this research project. The teleoperation will be conducted between two geographically separate facilities. The robot side is in Bristol Robotics Laboratory (BRL) which has conducted extensive research on teleoperation and has a state-of-art leader-follower teleoperation system. The application side is at the National Composites Centre (NCC) and its digital programme DETI (Digital Engineering Technology & Innovation). A high-speed high bandwidth cabled connection has been created by the South Gloucestershire Umbrella project between both facilities. This connection is referred to as “dark-fibre.” 5G-ENCODE is the main funding body of this project and has enabled this research to be undertaken.



Figure 1: Project enablers



Figure 2: Project collaborators

1.1 Motivation

Teleoperation of robotic systems is a great method for actuating robotic systems that operate in complex environments. Autonomous robots have made major steps in both industrial and commercial use-cases. However, when robots are required to perform intricate non-repetitive tasks in complex environments the actuation still relies on human control, as the human ability to interpret complex environments remains ahead of known autonomous technology.

To accommodate the need of robotics teleoperation solutions BRL designed a leader-follower set-up that consists of two Franka Emika robot arms (see figure 1.3 a). An operator moves the leader robot of which the motion is subsequently followed by the identical second robot resulting in active teleoperation control. The follower robot returns information to the leader robot when it has contact, to give a form of feedback to the operator. This is done by the leader follower not allowing itself to be pushed further in the direction of the contact. In a different project, a Franka Emika robot arm is equipped with a stereo camera of which the feed is visualised in a VR-headset which is worn by the operator. In this scenario, the operator can look around and be more aware of the robots working environment (see Figure 3).

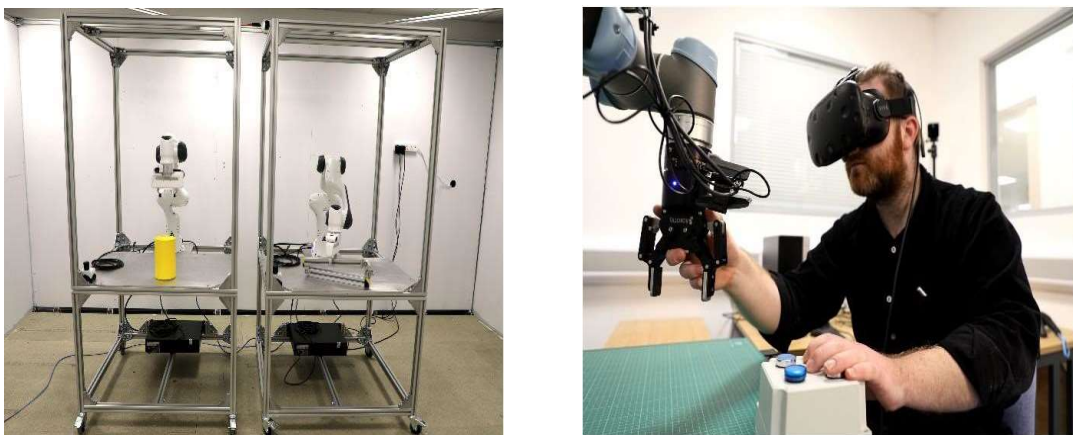


Figure 3: Bristol Robotics Laboratory teleoperation

In future projects, alternatives will be needed when larger industrial robotic arms are used as the operator will not be able to manually guide a large robotic arm and this solution can encounter issues with transforming the output of the smaller leader to a different larger follower robot. Hence, there is the need to further study robotic teleoperation control system so that the final solution is unaffected by the size of the robot being actuated.



Figure 4: KUKA Nuclear decommissioning solution

The need for teleoperation control unaffected by robot size becomes clear with one of the largest use cases for this research. Nuclear decommissioning is forecasted to have a market value of 121 billion pounds as part of the UK future clean-up. This market value is estimated to realise across 120 years creating a sizable and long-lasting market. This use case involves work with dangerous substances that render the environment hostile and not suitable for human presence without specialised protective gear. The current solution for the need of teleoperation in nuclear decommissioning involves the use of a nearby (at same location) controller (see Figure 4).

To increase operational efficiency, it is necessary to find more intuitive methods of teleoperating 6-DOF industrial robot arms. These systems will perform complex tasks in equally complex and hazardous environments. An additional use case is that of welding operators and inspectors which can also benefit from the research done in this project. Besides the safer work environment, the added benefit of decoupling the operator is an easier start-up and even the potential to teleoperate robots from all over the world.

Research needs to be done regarding the network requirements to realise teleoperation of robotic systems. During this project, a robotic teleoperation network is set-up to determine the network requirements. This initial network uses cables throughout for the trials to be conducted. The steps that have been taken to realise this are discussed in this report.

1.2 Aims & Objectives

This project focusses on the implementation of intuitive haptic control protocols and the creation of an active robotic arm teleoperation network to determine the feasibility of 5G implementation.

The aim of this project is to understand and generate knowledge relating to remote operation of robotic arms in an industrial context summarized in the following project statement: “To determine the feasibility of 5G connectivity elements in a robotic arm teleoperation scenario where haptic control is used”

The general objective statement describes the elements that require to be completed for the aim to be achieved: “To create an active robotic arm teleoperation network that allows the operator to actuate the robot arm using existing haptic control protocols whilst being made aware of the robot’s surroundings”

From both general statements, a list of aims, and objectives is created. This list will be referred to throughout the report to gauge the research success:

Aims

- Review the network statistics in the wired teleoperation trial
- Review current 5G options for robot teleoperation
- Evaluate the pros and cons of 5G implementation

Objectives

- Enable interfacing with existing haptic devices
- Implement control of robotic arm
- Implement stereo camera feed
- Create a virtual reality solution
- Create a teleoperation network
- Record network statistics during wired teleoperation trial

The research aims & objectives are chronically ordered, however, before these objectives can be tackled it is of importance to discuss the choices made and the steps that have been taken in the duration of this project. A technical description follows which evaluates the solutions found to address each objective. With this given, it is possible to achieve the research aims through discussing the results.

2. SOLUTION SELECTION

In this chapter relevant studies are researched and described with the aim to determine which solutions need to be used for different parts of the project. An important part of the project has been to get the right equipment for the right use-case. In this chapter the decision forming is focused on narrowing the scope of the project as the core of this research remains the network requirements and with that the feasibility of 5G implementation.

The haptic device is the main input of the solution and allows the operator to direct the robot arm with their hand. Once the operator's input is gathered it is necessary to translate this to an output on the robot. This translation is done using haptic protocols to which prior research has been done [1]. From this research a single protocol is chosen to limit the scope of this project. A VR-headset is necessary to create the immersive experience needed to give the operator a perspective of depth during teleoperation. Multiple VR-headset devices are currently available on the market and the decision process is described. As part of the immersive experience there is also a camera is required to view the robot.

2.1 Haptic robot control

New terminology was introduced during the literature review of the "Haptic Control Protocols for Simulated Industrial Robot Control" research [1] when it comes to haptic control. The terminology was designed to clearly describe the haptic protocol from a standardised point of view allowing for research undertaken on different hand recognition devices to be easily compared.

Hand recognition device	Protocols	Feedback
Kinect	PtPp [2], PtJv [3], PtPv [3], JtJp [4]	Not natively
Leap motion controller	PtPp [5][6][1], PtPv [1]	Lower hand, position only
Data glove	PtPp [7][8], JtJp [8]	Full hand, movement only

Table 1: Haptic protocols per device [1]

The meaning of the haptic protocol abbreviations can be determined in sections. The first capital letter describes the hand input and the second capital letter describes the robot output. Where P stands for "Pose" and J for "Joint". The smaller letter at the end describes the form of output whether this is done through "position" (p) or "velocity" (v) commands.

Hence Pose to Pose position (PtP) based haptic protocol uses the Cartesian coordinates of the operator's palm and translates these to position commands for the robot's TCP pose. Pose to Pose velocity (PtPv) based protocol uses the same Cartesian coordinates of the operator's palm but now translates this to a velocity vector placed on the robot's TCP pose. Pose to Joint velocity (PtJv) based control protocol changes the joint angle of a selected robot joint by a rate that depends on the position of the operator's hand. Joint to Joint position (JtJp) based protocol maps the different robot joints to different joint angles of the operator.

Three types of devices (Kinect, Leap and data glove) have been identified that have gained the most haptic robot control research. These devices are described and compared in order to explain which device is best for the project. It should be noted that this is undertaken to give a good overview of devices available on the market, however, individual prices have not been considered as these can fluctuate.

2.1.1 Kinect

There have been several versions of the Kinect device. The latest solution from Microsoft is called Azure Kinect (2019) which builds on its predecessors Kinect V2 (2014) and V1 (2010). The first two Kinect versions were aimed at the consumer market for use as an accessory to the XBOX game console. The device mapped the user's body which allowed the user to interact with the game by moving. Hardware with an ability to recognise body positions proved to be a very useful tool in research projects and consequently was used in a range of research projects including computer vision and robot teleoperation. The latest edition of the Kinect has seen a change in aim as this product is now focussed on the research in industrial applications.



Figure 5: From left to right: Kinect v1, Kinect v2, Kinect v3 [9]

The Kinect has been implemented in a solution for real-time remote robot teleoperation [2]. Here the depth images of the Kinect are used to segment the active arm from the rest of the picture. This allows for the index finger and the thumb to be efficiently tracked in 3D space. A PtPp (Pose to Pose position) based controller is implemented where the hand pose of the operator is used to direct the robot's TCP position and direction. Hand tremors are a natural phenomenon that can result in unwanted robot moves. During the same research it was found that the use of different modes (e.g. coarse and precise) is useful for robot operation. Another research project [3] used a velocity-based protocol and inadvertently found that this solved some of the operational issues. Firstly, operator's joint limitations only prevent faster motion as opposed to limiting the all-out controllable robot workspace which would be the case in position-based control. Secondly, using a velocity-based protocol limits the effect hand tremor has as any involuntary movements are not directly mimicked.

2.1.2 Data gloves

A device that is worn on the hand which measures hand motion and translation by use of tracking sensors. The hand movement is translated to a digital instruction. These devices do not require any computer-vision based systems which can reduce the computational power requirements. However, these devices may require calibration and can be expensive and bulky when set-up.

A data glove solution was introduced by HaptX which uses an additional Vive tracker. This device combination gives the operator an intuitive way of operating a robot arm that is equipped with an anthropomorphic hand [8]. The Vive tracker allows for a PtPp based control protocol where small increments in motion are translated to the robot's TCP position.

The anthropomorphic robot hand has tactile sensors which are translated within the data glove giving the operator haptic feedback in the form of touch. The anthropomorphic hand end-effector itself is controlled by the data glove using a JtJp protocol where the operator's hand joint Cartesian coordinates are mimicked.

Data gloves have proven to be good for tracking movement of individual fingers. However, to track the hand position in space requires additional sensors. Custom made and calibrated equipment is needed to accommodate different hands. It is only possible to apply resistance on finger movement and not the movement of the hand itself.

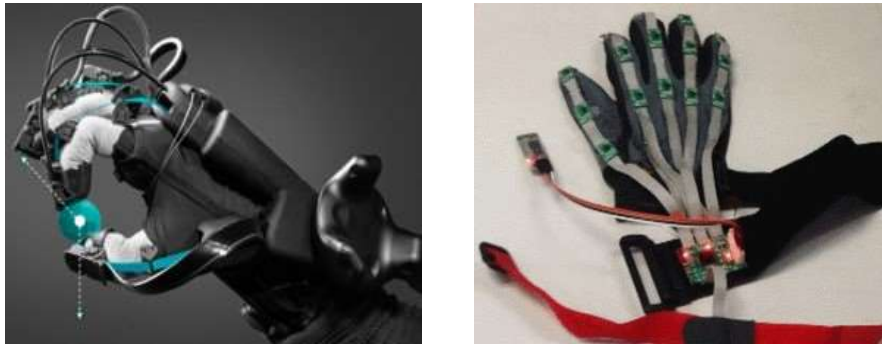


Figure 6: Data gloves – HaptX [8] & YoBu [7] (left to right)

2.1.3 Ultraleap motion controller

A compact optical hand tracking device designed to be used on a desk. The operator places his/her hand above the sensor and the dual infrared cameras are able to track the position of the hand in real-time.

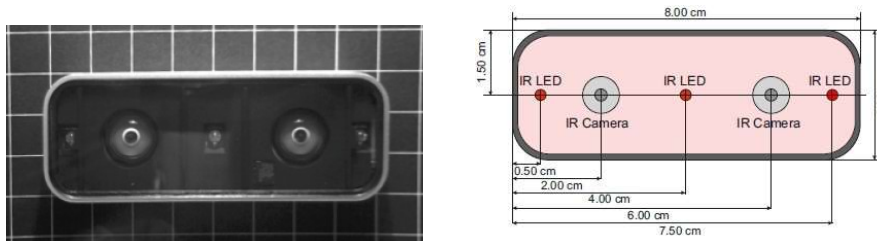


Figure 7: Ultraleap motion controller [10]

The device is designed by the company Ultraleap with the main focus to recognise hands and their relative position. It has an operational frequency of 120Hz; however, no quantitative accuracy measures have been given. One of the first studies on the Leap motion sensor was conducted to quantify the performance of the device, for both static and dynamic accuracy [10]. An industrial robotic arm was used to determine an average static position error lower than 0.2mm and an average dynamic error of 1.2mm.

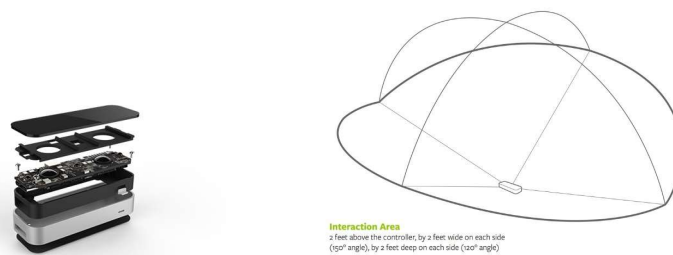


Figure 8: Ultraleap motion controller

Due to the standard desk placement of the device the field of view of the device can become a limiting factor. Both infrared cameras have an optimum distance of 60 cm above the sensor and have a field of view of 140° by 120°. The resulting workspace is a cone that morphs into a sphere that has an effective radius of approximately one metre. The device

tracks the movement from one point of view and therefore can struggle with specific hand positions, gestures and fingers that are hidden behind the palm as was found in multiple studies [11][12]. Direct sunlight and other infrared light are found to detrimental on the hand recognition ability [13].

The Ultraleap motion controller and its use-case for robot teleoperation has been less researched than the Kinect. This is, in part, related to the hype that surrounded the release of the first Kinect, potentially overshadowing the Ultraleap motion controller. This does not make it a less viable option. The Ultraleap motion controller outperforms the Kinect when it comes to hand recognition and tracking.

One prior research, dating back to 2015, has shown that an Ultraleap motion controller was used to control a robotic arm [6]. In that report, a UR10 robotic arm was simulated using control based on PtPp haptic protocol. During this research it was found that a low-pass filter helped reduce the effect of hand tremor or other noise related issues in the control process.

2.1.4 Robot teleoperation method

First, it is of importance to conclude which haptic device is most suitable to be used for haptic control in a teleoperation robot arm use case. A relatively high amount of research has been conducted using one of the different Kinect versions. However, it should be noted that these devices have been designed to track not just the hand and is, therefore, not necessarily the best option for a hand tracking only use-case. Data gloves enable both precise hand and finger positioning which allows for grasping control. However, these features can only be capitalised on when specific anthropomorphic hand end-effectors are used which, on top of the already expensive data glove, come with an expensive price tag. In addition, the data glove has the drawback of requiring an extensive setup and does not allow for hand positioning feedback. The Ultraleap motion controller is the device of choice as it outperforms the Kinect and the Vive tracker when it comes to hand tracking.

The Ultraleap motion controller has further advantages as it is small in size and easy to set-up for operation. When the Ultrahaptics STRATOS Explore, device is added to the solution it can also give haptic feedback. This is a high-end mid-air haptic feedback device that uses a set of 256 transducers that can project a tactile sensation on the hand through ultrasonic waves.



Figure 9: UltraHaptics Stratos Explore

Some common challenges have been found with different haptic protocols. These are discussed below to give a clear overview before explaining how the proposed solution solves these issues. Firstly, that of human joint limitations, both JtJ and to a lesser degree PtP have their workspace limited due to this natural occurrence. Operational difficulties arise when the operator is not able to position their hand in the right manner for the robot to perform a particular action. Secondly, the problem of hand tremor which can result in involuntarily movements of the robot that has a detrimental effect on the operational performance. Our hands are never perfectly still, and this phenomenon mainly impacts protocols that try to mimic the hand's movement like the ones that are position based.

Numerous types of robot control protocols have been found. All can be applied for haptic robot control, however, not all are necessarily sensical when operated just by hand movements. A gesture type PtPp based control protocol has been designed before for the Ultraleap motion device [6]. During the prior research [1] two types of protocols were designed for the Leap motion hand controller. The first is also a PtPp based controller, however, this time the palm position is directly used from the Leap SDK. The second protocol is a PtPv based controller which, during the research, was found to outperform its competition on both objective and subjective measures.

In short, the PtPv based haptic control uses the operator's relative palm position and translated to a velocity vector on the robot TCP position. This protocol allows for unlimited movement of the robot TCP and is naturally unaffected by the operator's joint limitations. This protocol is also impervious to hand tremors as it will move the robot in a specific direction with small changes due to hand tremors removed from the actual robot movement.

2.2 Streaming camera

Given that the operator is not in the same room as the robot itself, it is important that the robot is visualised to the operator. There are multiple options to do this, however, there are some requirements and useful notes that help limit the number of options which will be discussed here.

Firstly, it is of importance to determine what type of camera set-up is useful. The main types to choose between are a standard digital camera, 360° camera and a stereo camera. The standard digital camera comes with many options and allows for high quality capturing and many options as to type of cable and video signal. 360° cameras allow for a full capture of the environment with the drawback being that the sense of distances can be distorted. A stereo camera gives the opportunity to give a sense of depth if combined with a VR-headset. Especially when operating a robotic arm for a pick & place use-case it is important to be able to see such depth. Henceforth, there is only the need to look into stereo cameras for this project.

There are a number of stereo cameras on the market, however, many cameras are focussed on the 3D mapping potential that a stereo camera can realise. The two main products that focus on high quality imagery on top of having the distance between cameras in such a way that it can be used by VR-headset are produced by Stereolabs and Suometry.



Figure 10: Camera options, Suometry system & ZED camera (left to right)

Both options were suitable for this project as both can create an immersive experience and give the operator a sense of depth. It was important to also consider which option could better integrate into the already existing virtual teleoperation solution. The existing solution

uses Unity which is a 3D game engine to create the virtual environment and simulate the robot movement.

The Suometry “360° stereo video camera developer kit” comes with the main benefit that it allows stereo vision for the complete 360-degree field. It also allows for higher framerates at similar resolutions. However, the zed camera comes with a Unity plug-in and has been proven to work with VR by the project collaborator BRL [16]. Hence, the ZED camera 2 has been chosen as the device to be used for this project.

2.3 VR Headset

A head-mounted device that can provide a virtual reality experience for the operator. VR headsets are widely used for commercial use-cases like video games and have more recently also been used for simulators, research and industrial design use-cases.

The headset needs to be able to communicate with a desktop wirelessly which reduces the number of VR-headsets available. The main competition is between the Oculus Quest 2 and the Vive Pro 2. Both devices can give an excellent immersive experience. The HTC Vive Pro 2 has a higher resolution (4896 x 2448 px) compared to the Oculus Quest 2 (3664 x 1920 px), however, both resolutions are higher than the highest camera resolution at 30 fps (3840 x 1080 px). On top of that the HTC Vive Pro 2 requires additional equipment to make the experience wireless which is natively supported by the Oculus Quest 2 through the newly added feature “Air-Link”.



Figure 11: VR-Headset options, Oculus Quest 2 & HTC Vive (left to right)

With all the pros and cons weighed it was decided that the Oculus Quest 2 was the best option. This device is a VR-headset created by “Meta Platforms” and is a standalone headset with inside out tracking which results in minimal set-up time. The device runs an Android-based operation system and is compatible with a desktop computer, through the use of proprietary Oculus VR software, when connected over either USB or Wi-Fi.

During the project it was found that the Oculus Quest 2 headset is not natively compatible¹ with the ZED unity plug-in. This results in the plug-in not automatically creating the needed gameobject (ZedRigDisplayer) and assigning the correct assets (ZEDMixedRealityPlugin.cs). However, this does not mean that the camera and the headset are not compatible with one another. This additional gameobject in the headset can simply be removed which only results in limited in-game options but will give the same end-result of an immersive experience with a sense of depth.

¹ <https://support.stereolabs.com/hc/en-us/articles/360011975694>

3. TECHNICAL DESCRIPTION

This chapter documents the development process and research undertaken in this project.

It is necessary to further describe the workings of the control protocols as used within the teleoperation solution. The devices needed to interface with the created teleoperation solution are described. The created network is described in detail which is the foundation for the research that is done.

To understand the simulation facets, it is of importance to have a clear understanding of how the control set-up is created. This set-up was proven to work in a previous project [1] where the solution was simulated. This limited the number of variables in the project thus decreasing project risk.

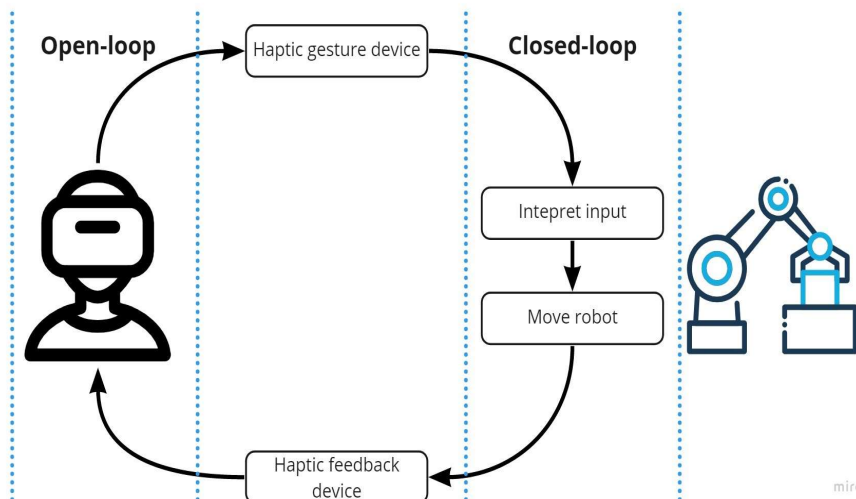


Figure 12: Haptic Control Set-up

The control setup illustrated in

Figure 12 shows how the control related devices are involved in the process of actuating the robot arm. The operator's side is essentially an open loop as the system has no direct influence on the actions of the operator. The robot side is, in contrast, a closed loop as the system handles the movement of the robot from the input data given. The operator interacts with the haptic tracking device which is interpreted by the algorithms embedded in the solution which in turn moves the robot. The application itself interacts back to the operator through the haptic feedback device by means of tactile sensation. It should be noted that the haptic feedback is on an application level and, therefore, does not use information coming back from the robot itself. Plans to integrate a touch sensor to the end-effector

which can be relayed back to the operator through the haptic feedback device have been discussed, however, this is considered out-of-scope for this project.

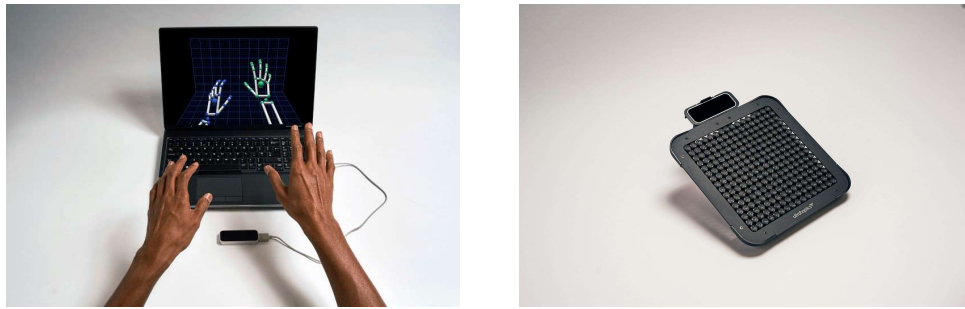


Figure 13: Ultraleap Devices, Motion controller & Stratos Explorer (left to right)

The haptic gesture device chosen is the Leap Motion Controller. An optical hand tracking device that creates a virtual model of the hand through what is called “skeletal tracking”. This allows for finger positions to be estimated when not directly visible. The haptic feedback device that is paired with the leap motion controller is the Stratos Explore. A haptics development kit that is able to create mid-air tactile feedback through the use of an array of transducers.

3.1 Haptic Protocols

The haptic control protocols are key to the success of this solution. As part of the solution selection multiple haptic protocol options were reviewed and a protocol to use decided. Additionally, a decision has been made as to which haptic protocol will be used. In this section the haptic protocol is described in more detail.

The figure below shows both the input given by the operator and the output that will be given to the robot's TCP. The device has a virtual centre point that is effectively 20cm above the centre of either the Ultrahaptics device or the Leap motion sensor. A velocity vector is drawn between the virtual centre point and the operator's hand position (e.g. middle of palm). This velocity vector is then transferred to the robot's TCP which means that the further the operator's hand is from the centre point the faster the robot will move in that direction.

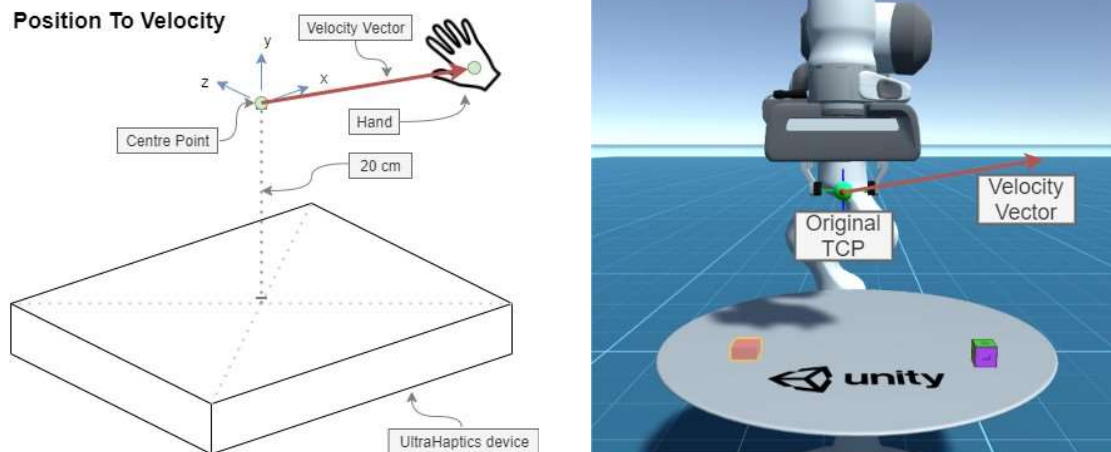


Figure 14: PTV Control [1]

The rotation is directly transferred from the operator's hand to the TCP on the world axis again in a velocity fashion. This is considered to be a pitch-roll-yaw rotation system where the rotation of the hand affects the TCP's rotational velocity. For instance, when the hand is pitched upwards the TCP will pitch in the same fashion. The higher the pitch of the operator's hand the higher the pitch-like rotational velocity of the TCP.

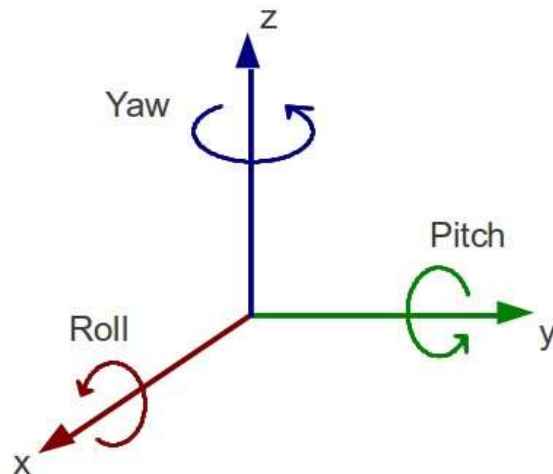


Figure 15: Pitch Roll Yaw diagram

The PtPv based controllers struggle less with the problem of the operator's joint limitations as well as sensory workspace issues. Additional scaling is used where the input of the operator is scaled and with that directly increasing the output on the robot arm. This is in direct correlation so when the scale-factor is set to two the velocity vector on the robot arm would be twice as high for any given input of the operator. This scale-factor can be actively changed by the operator in operation allowing.

3.2 System interfacing

There are a total of five separate systems that require interfacing with the application. In this section the devices in the systems are discussed as well as their interfacing solution and accompanying limitations.

It should be noted that not all system connections have an influence on the network which is at the core of this feasibility project. Only the robot system and the camera part of the immersive system influences the network. In the network section of this chapter this is discussed in more detail.

3.3 Haptics System

There are two devices that make up the haptics system. One is the Leap motion controller which tracks hand movement, and the other is the Ultrahaptics device that uses the tracked hand information and a set of transducers to create haptic feedback. Both devices come with their proprietary software development kits (SDK). This is used to interface with the relevant device and is described below.

Leap Motion

The Leap Motion Controller comes with an SDK that features a C-based API (Application Programming Interface) called LeapC. This API allows for higher-level scripting languages to access the tracking data of the Leap motion controller. There are different versions of the tracking software available depending on the development operating system (OS) used. The most recent version available is Version 4.1.0 for Windows. This tracking software is installed in order to access the SDK and its internal API. The Leap Motion C API is used as an intermediary between the application created and the Leap Motion service. The most important functionality required for this project is the hand tracking information. The Leap Motion system uses a right-handed Cartesian coordinate system of which the origin is found at the centre of the sensor's topside.

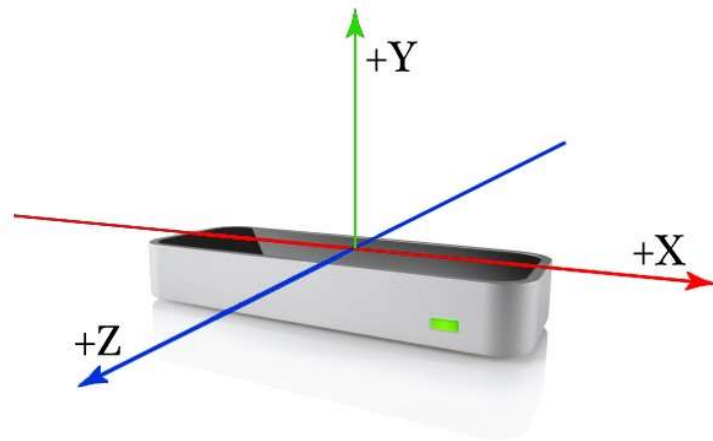


Figure 16: Leap Motion cartesian system

Before the tracking information is accessed a connection is created using the SDK. These steps can be skipped if the Unity plugin, created by Ultraleap is used. The Core unity module is all that is needed as this holds the crucial Leap service provider. This script is added as a component to a gameobject in Unity and is the class that communicates with the Leap service. This gameobject is subsequently addressed by all other scripts to gain valuable tracking data.

UltraHaptics

The UltraHaptics device SDK comes with drivers for the transducer array as well as C# and unity libraries. A unity component is included that is added to a gameobject within Unity, similar to the leap service provider. When this gameobject is active, the UltraHaptics device is given instructions to create a certain tactile sense. In this instance, this would result in a focused point sensation that represents the device centre point as discussed in chapter 3.1. Further research into the API and the Unity integration needs to be done to determine potential avenues for haptic feedback.

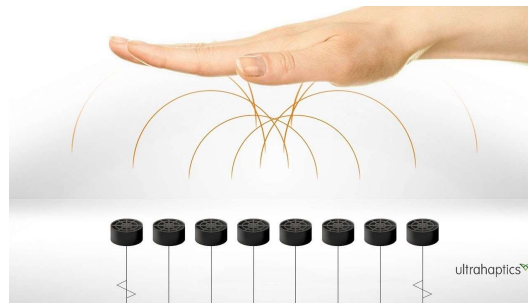


Figure 17: UltraHaptics mid-air tactile sense

Unity Implementation

A section called *hand portal* is created within Unity. Here the haptics system is both integrated as well as visualised, this involves a hand representation as well as overall control.

The hand portal shows important information regarding the inputs from the operator and how they are transferred to the target object (e.g. Robot TCP). The hand portal is a number of facets that can all be clustered into four groups (e.g. *Hand Controller*, *Input Drawer*, *Hand Object*, *Target Object*) that make up the logic. The figure below shows the two groups *Input Drawer* and *Hand object* in-session representation. The hand object is represented by a sphere that has a coarse form when disconnected and becomes circular over the connecting period. The input drawer representation holds the connect status as well as the dead man's switch, connection timer, control protocol text and input details.

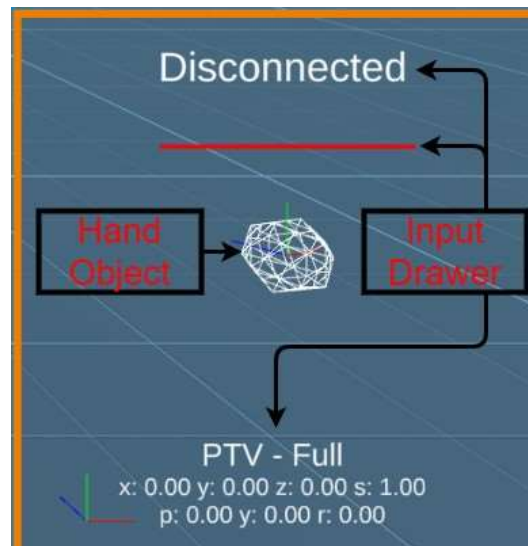


Figure 18: Hand portal in-session annotation

The *Hand Controller* outputs several information streams to the other groups as can be seen in the figure below. As variable inputs, it has the leap plug-in and keyboard which allows for interaction with the leap motion controller and added operator input, respectively. It also has a set of notable internal variables which are either used by the other groups or tells the hand controller which game objects to interact with.

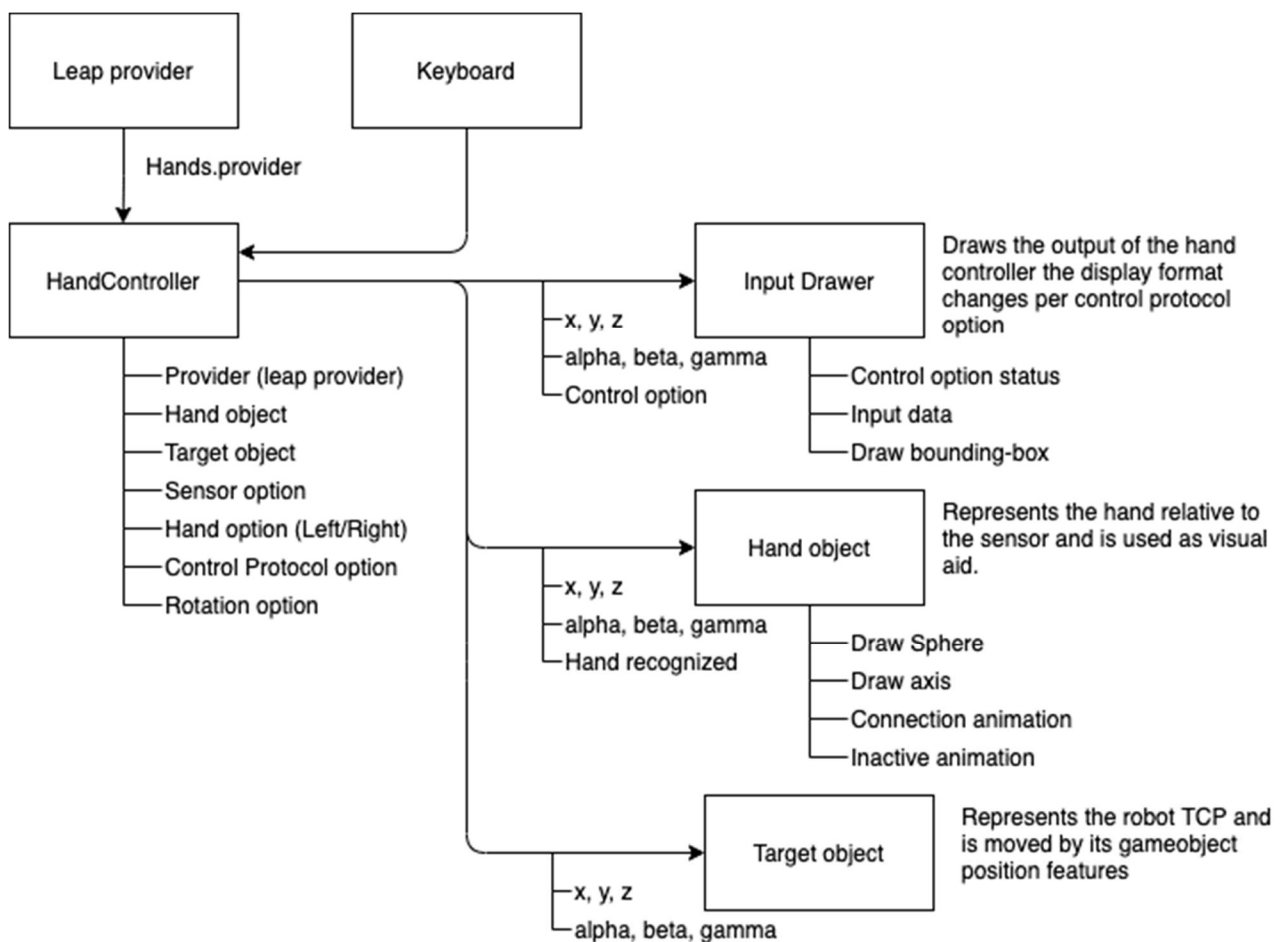


Figure 19: Hand portal architecture diagram

It should be noted that the keyboard inputs are crucial during the pick & place operation as they can change settings on the go and most importantly function the dead man's switch. Figure 3.21 shows the control layout of the keyboard.

The finite state machine of the HandController is managed by the ControlStatus for which there are three options: Connected, Disconnected and Connecting.

This segment lays the foundation of how the operator seizes control. In short, the intuition behind this system is that the operator does not want to be in control as soon as a hand is recognised or when they are not aware, as this can result in dangerous situations. The figure below shows the finite state machine set-up. First, a check is needed to determine whether the dead man's switch is pressed (e.g. space bar). This is standard procedure for human robot control and brings two benefits. Firstly, the operator is aware that control can be given to them. Secondly, control is quickly terminated on releasing the dead man's switch.

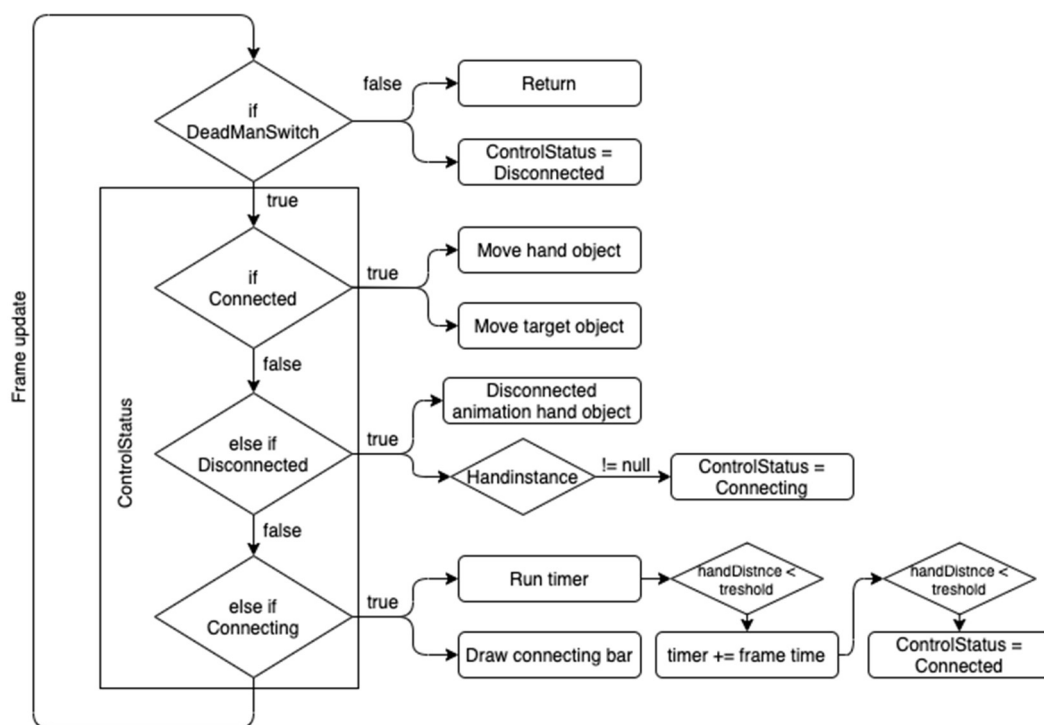


Figure 20: Hand controller finite state machine architecture diagram

The figure below shows how the position and rotation control is separated for both PTP and PTV and, therefore, addressed separately. It should be noted, that in this project, only PTV (a PtPv based haptic protocol) is used. However, it is still important to determine what control protocol mode is being used (e.g. Both, Position only or Rotation only) as this determines whether both position and rotation control is used or just one of the two.

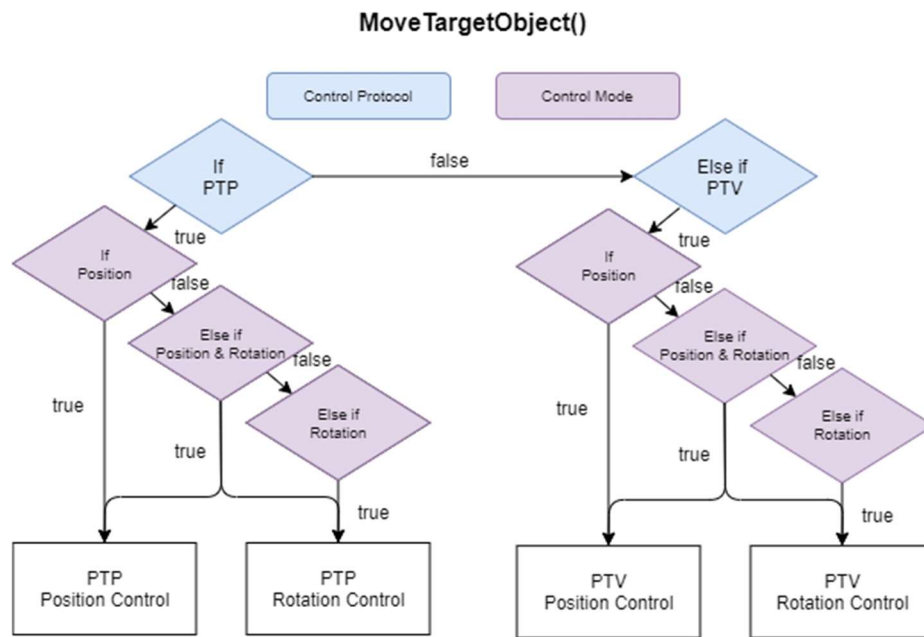


Figure 21: Move target object

The importance of being able to only translate or rotate should not be underrated. During the prior research [1] it was found that participants often struggled to recognise the small rotational changes they were asserting whilst being focused on translating the TCP. This often meant that the small changes overtime became considerable and once the position was reached the end-effector had to be corrected on rotation.

3.3.1 Robot system

The robot system consists of two parts. The first part is the virtual part where the robot arm is simulated. This virtual part is realised in Unity and both visualises the robot arm as well as performing the inverse kinematics of the robot arm. The inverse kinematics calculate the robot joint angles from the given TCP which is moved around by the operator. The joint angles are then sent through a TCP (Transmission Control Protocol) connection to the actual robot arm which is the second part of the robot system.



Figure 22: Franka Emika Panda robot

Virtual robot arm

Unity is originally a cross-platform game engine developed by Unity Technologies designed for game development. However, the strong ability to create 3D environments and the inert connection with powerful and efficient C# code it is also possible to cater for robotics use cases.

To realise inverse kinematics within the Unity simulation a plug-in is used. Bio IK is a generic geometric kinematic solver that allows for the inverse kinematics to be set up for any kinematic chain. The robot arm is set up in such a way that each link is its own gameobject. The base is the main parent, and each subsequent link is the child of the link prior, up to the gripper (see figure 3.12). The centre-point between the two gripper-fingers is set to be the TCP. Bio Ik works by adding a behaviour component to each link and telling the joint rotation limitations. It is of importance that the origin point of the gameobject is concentric with the joint's centre of rotation. With the robot links set up the TCP, which is placed last in the chain of gameobject children, is set to follow a target object. The inverse kinematics solver will then determine the joint angles and rotate the gameobjects accordingly.

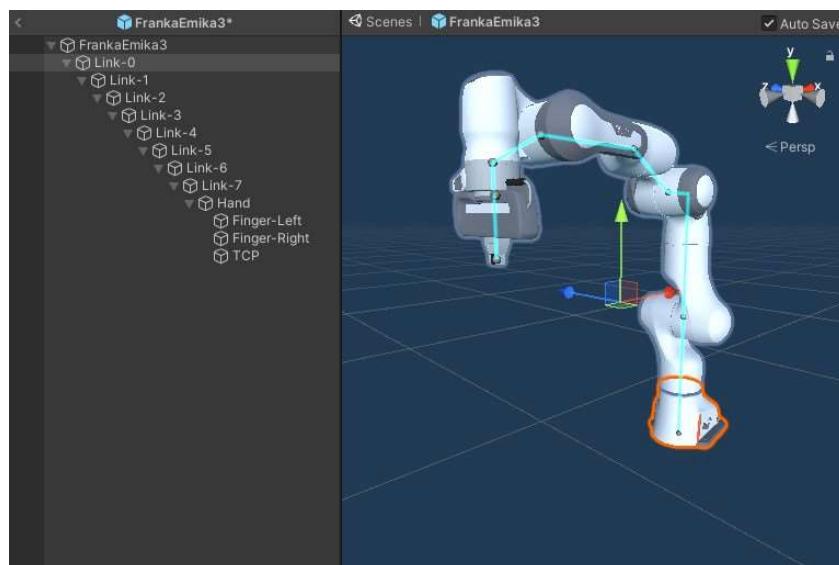


Figure 23: Franka Emika Panda robot inverse kinematics set-up

Physical robot arm

This robot arm is the same as its virtual counterpart and located at Bristol Robotics Laboratory. The Franka emika robot arm has a control module which is connected to the Linux based computer that has been dubbed the “robot computer.” The function of the robot computer is to connect with the TCP server hosted by the application, interpret the joint angles and subsequently give the robot movement commands.



Figure 24: Bristol Robotics Laboratory set-up

3.3.2 Immersive system

This section discusses all the elements that relate to the immersive side of this project. The immersive solution contains two separate devices one that is used to capture the environment and one to display the captured feed. The project selection of immersive devices is discussed above. It was decided to use a stereo camera referred to as ZED 2 and the Oculus Quest 2 VR-headset. The ZED camera implementation is described first in the following sections as this set-up is used by the VR-headset.

ZED Camera

The integration of the ZED camera and its feed comes in two parts. There is a sender side which makes the camera feed available to the local network and a receiver side which connects to the sender. It is important to get the correct software prior to configuration of the solution. For this project version 3.6.5 of the ZED SDK was used. To use this SDK another piece of software, NVIDIA CUDA, was needed that allows for accelerated GPU processes.

Sender

Connecting the camera and opening the feed for the local network is done by using one of the examples given by Stereolabs. The example uses the ZED SDK and allows a live video stream over local IP network. The feed is encoded to limit network bandwidth requirements.

The camera feed needs a minimum FPS of 30 to be usable within the immersive experience. From Stereolabs documentation, it can be seen that this comes with a maximum resolution of 1080p and an advised bitrate of 12500 [kbits/s]. These values are manually set within the example code.

Receiver

By using the ZED SDK for both the sending as well as the receiving side of the camera feed there is no inherent problem with implementing the feed differently compared to when an USB connection is used. The ZED API must be told to use a stream as input and the associated IP-address needs to be given but apart from this the solution will behave as if the camera has a direct connection.

For the integration within Unity the ZED Unity plugin is used. This plugin comes with multiple examples for use as well as prefab configurations. The main Unity prefabs that are being used in this project are the so-called “ZED Rig Mono” and “ZED Rig Stereo”. The mono variant is used in both the main screen to perform a connection check and give a quick view of the camera feed as well as in the camera teleoperation scene where it is used to show the robot during operation. The stereo variant is used for the VR headset solution where the feed of each camera is overlayed on the corresponding eye plane.

An additional behaviour script was added on top of the existing prefab to allow for the integration of the ZED camera within the general application. This was done by having the ZED Rig Mono on inactive until connection is needed for which it is first ensured that the settings of the ZED manager are correct. When it comes to the VR solution the ZED camera is always running, however, the Oculus environment is overlayed giving the illusion of the camera being turned off. It should be noted that turning of the receiver side within Unity can come with issues potentially not allowing the camera feed to be shown again even-though the camera is reconnected. Turning off the camera prefab which includes leaving a scene whilst the ZED camera prefab is not connected can also result in the application stalling and taking an extended time to perform the requested action.

VR-headset

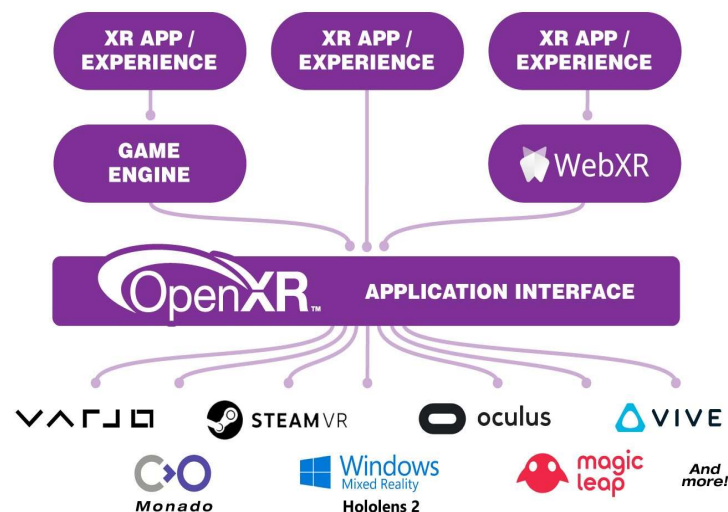
With the Oculus Quest 2 VR-headset selected for use it is possible to use specific functionality related to this device. In this case an SDK is provided to expedite the Unity development. Using this SDK it was possible to add device embedded features as controller input and visualisation as well as adding a general pipeline implementation for building the VR experience.

The Oculus headset is designed to run android applications; however, it can also be used as a standard VR-headset where the application is run from a desktop to which the headset is connected. For this project, the application was run on the desktop to ensure connectivity with the robot and the camera as well as assuring the high computation needs were met when decoding footage. Air-link was used to wirelessly connect the headset to the desktop. This feature was added as an experimental streaming solution as part of software version 28 in April 2021. More information on connecting the headset with the desktop over air-link is available on the oculus website. Another option is to use a direct cable which is referred to as “Oculus-link”.



Figure 25: Oculus VR-Headset

The SDK already uses an open standard called OpenXR. This standard allows one application to operate with different VR-headsets. This can be especially interesting when the teleoperation solution is used more broadly as it would allow operators to use any VR-headset available to them.



OpenXR provides a single cross-platform, high-performance API between applications and all conformant devices.

Figure 26: OpenXR

3.4 Application

A short overview is given of the application in order to make clear where the more in-depth technical descriptions apply to. The application was created in Unity and consists of six different scenes. Each scene was a created environment where the operator can perform certain actions. The figure below shows how each scene was connected.

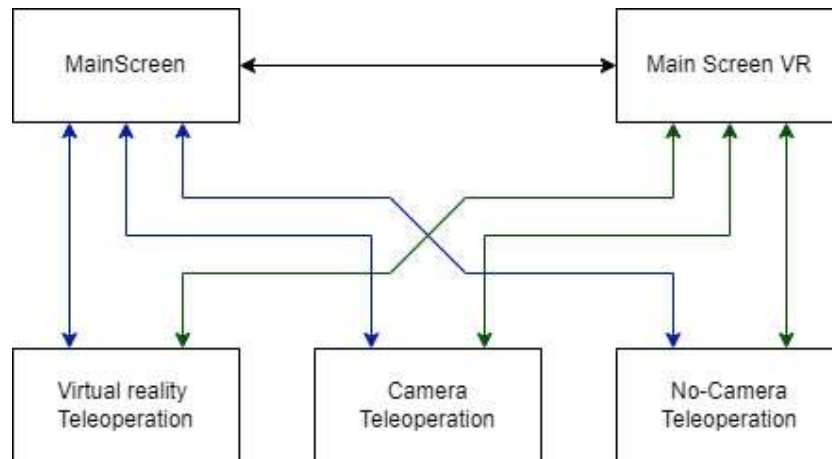


Figure 27: Scene overview diagram

3.4.1 Main screen

This is the first screen the operator will see when starting up the application. Effectively the heart of the application. From here it is possible to acquire more information on the project, go to the training environment, check device connection status, and go to one of the three teleoperation environments.

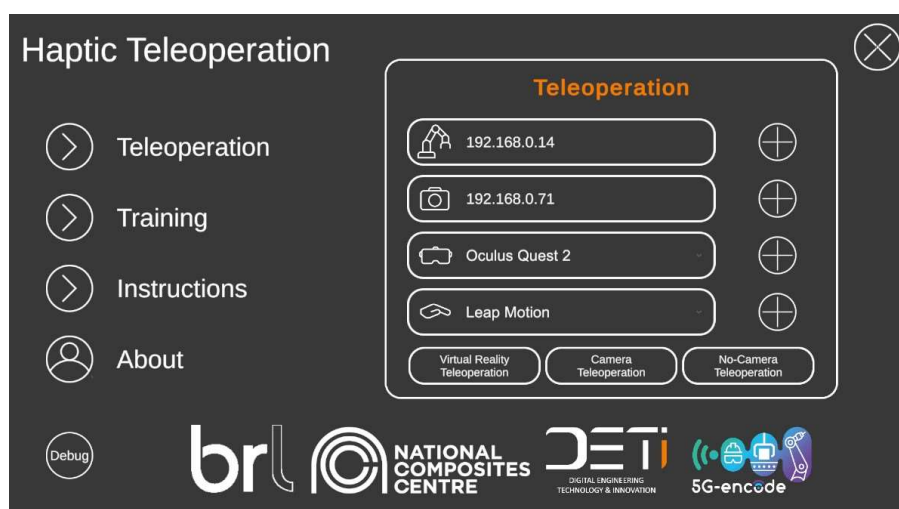


Figure 28: Main screen

Before any teleoperation is conducted it is of importance to fill out some device related information and automatically check the connection status of the device. The robot connection requires the TCP server host IP-address to be entered (currently IP-address of

device running application), the camera needs the IP-address of the device that makes the camera available to the network, the VR-headset needs to be specified as well as the haptic device.

Next to each input field a button runs a particular script when. The functionality of the button is to save the input given, check it and eventually use it to make sure that the device can be connected to. The figure below shows the logic that enables this.

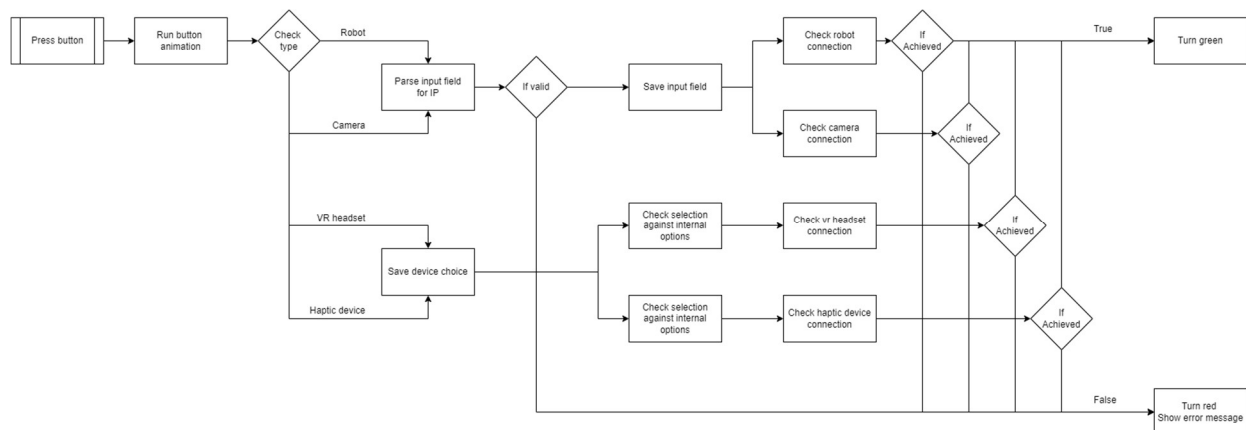


Figure 29: Connection button flow chart

The start event is one of the four buttons being pressed. The type is then checked to determine what kind of input is to be expected. There are four input fields, hence there are also four types. When it comes to the IP-addresses it is first checked whether it can be parsed and, therefore, is a valid IP-address. When it comes to the robot IP an additional check is done to determine whether the IP-address can be correlated with the DNS (Domain Name System) host info of the device. The input value is saved and automatically a check for connection is started. When this is conducted successfully both the button and the input field is turned green to indicate success. For the camera connection an additional screen is shown (see figure below) where the IP-address of the camera connected device is displayed as well as other variables.

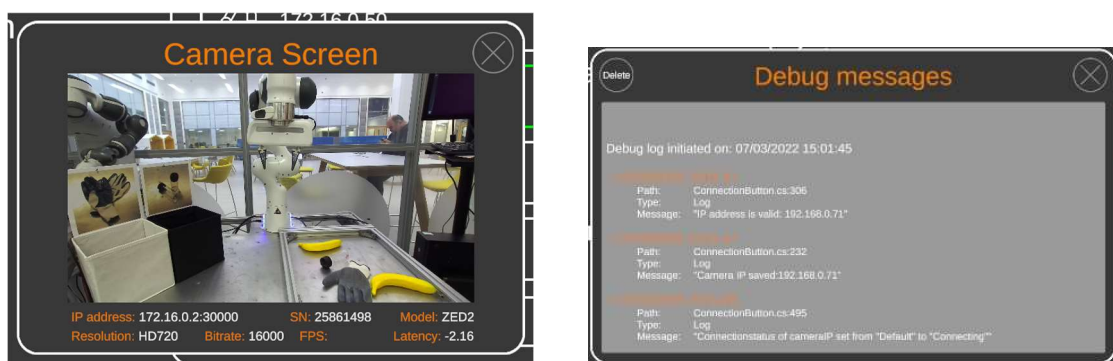


Figure 30: Camera screen & Debug screen (left to right)

Another screen that can be accessed from the main screen is the debug screen. Here all general debug log information can be read as well as any errors that might pop-up during operation. This is especially beneficial during test trials as this would mean that the errors are stored in one place and can be accessed from within the application. Normally the application is run from within the Unity Editor which gives a broad range of debug tools, however, from the Unity editor threaded TCP connections work very slowly and it was measured to take more than two minutes to achieve connection whereas that otherwise would take seconds.

3.4.2 Menu Screen

Each teleoperation scene has a menu screen that can be accessed through the “esc” button on the keyboard. This screen shows the saved input values as well as the current status of the four devices. It also gives the operator a way back to the main screen.

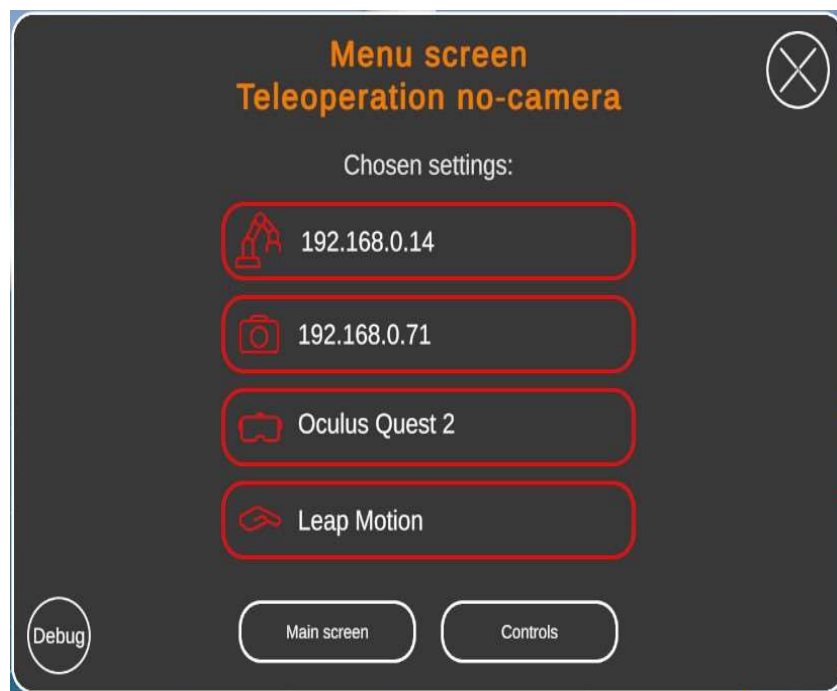


Figure 31: Menu screen

3.4.3 Controls

The figure below shows the control layout of the keyboard. This screen is reached by either going to the main screen and pressing the instructions button or accessing the menu screen from within a scenario (escape button) and press the controls button.



Figure 32: Keyboard controls

Main keys

- Key1 Escape: Opens menu screen
- Key2 Spacebar: Engages/Disengages dead man's switch

Control scale

- Key3 Z: Decrease control scale by 0.1
- Key4 X: Increase control scale by 0.1

Rotational

- Key5 R: Changes haptic control mode to rotation only
- Key6 F: Changes haptic control mode to rotation & position
- Key7 V: Changes haptic control mode to position only
- Key8 C: Toggles through the haptic control mode options

3.4.4 No-Camera teleoperation

The most basic scene where the digital robot representation is visualised show the desired robot configuration. This scene is an advancement of the free-roam environment development during the prior research [1]. Only two devices are necessary to allow this scene to work. A robot connection to perform teleoperation and a leap motion device to give instructions to the virtual robot.

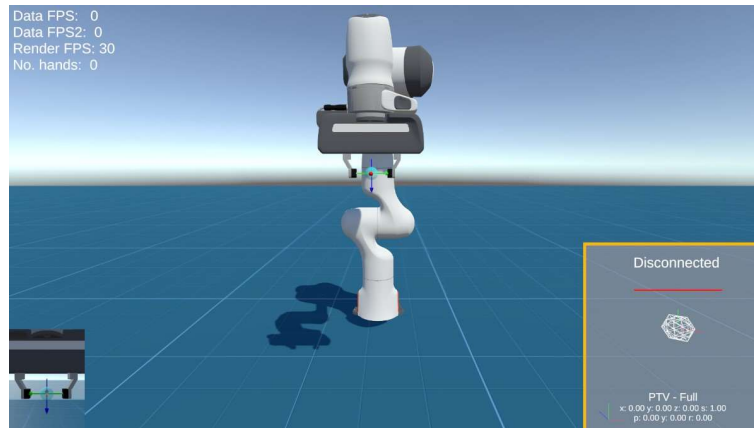


Figure 33: No-Camera teleoperation

It should be noted that it is not recommended to start robot teleoperation when it is not possible to see the robot arm. However, this scenario was created as part of the testing trials to determine the different bandwidths requirements for different configurations of the application. This set-up allows for the bandwidth of just the robot arm commands to be captured. It should also be noted that before acquiring teleoperation access multiple preparation steps had to be taken at the robot side. If the robot happened to be live and the applicable code was also running a key press was still needed for every teleoperation request before access is granted.

3.4.5 Camera teleoperation

This scene adds a direct camera connection and visualises this on screen. From here it is possible to conduct standard teleoperation using the leap hand tracking device like before, however, now it is also possible to see the robot through the camera feed displayed on the screen. This is the first step towards the goal of creating a fully immersive experience.

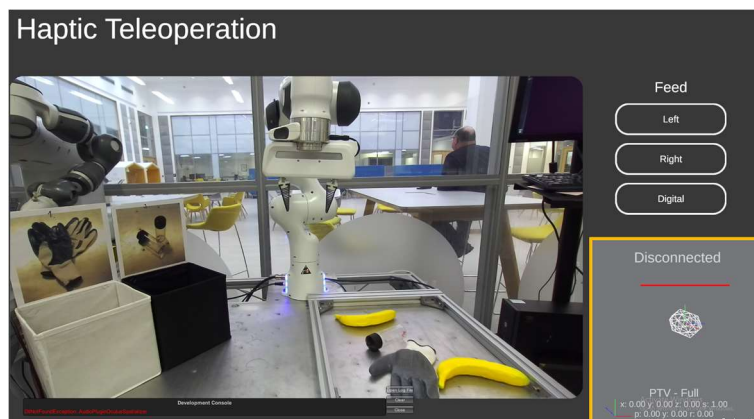


Figure 34: Camera teleoperation

When using this scene, it is possible to operate the robot arm adequately, however, it is relatively difficult to determine depth. In the trial this scene makes it possible to determine the bandwidth used when both utilising the camera as well as the robot.

3.4.6 Virtual reality teleoperation

This scene uses all devices that are integrated in this application. The robot arm connection allows for teleoperation, the leap motion device gives input to the virtual robot arm, the ZED camera shows a live feed of the robot and the VR-headset displays this feed. The camera feed is directly projected onto the VR eyes allowing the operator to gain a sense of depth. It should be noted that the camera is stationary, therefore, the camera is not able to turn when the operator turns his/her head. This can make the overall operator experience uncomfortable.



Figure 35: VR teleoperation menu screen

There are two modes when it comes to this scene. One is the dedicated camera feed where the operator will see just the feed itself. During this any head movement will not change what the operator is seeing. The other mode is where menu screen is visualised. The operator is placed in a minimalist room with a desk and a sizable menu screen behind it. On top of the desk the virtual robot is placed giving the operator even whilst viewing the menu screen a sense of what the robot's state is.

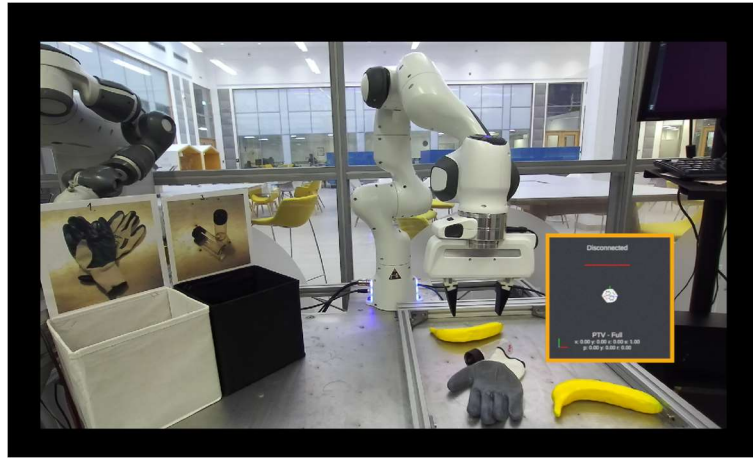


Figure 36: VR teleoperation camera feed

3.5 Network

An important part of robot teleoperation is the network capability. There are multiple devices incorporated in this solution and all of them have different network bandwidth requirements. All individual network links have a common need to be low latency.

In this section the network architecture is described along with its theoretical limitations. The network is used to determine the network requirements and options for future 5G deployment.

3.5.1 Current network architecture

The figure below shows the architecture of the network. Four major individual links are identified from this diagram. To lower potential risk for the project these links were tested on their own before being integrated in the whole project.

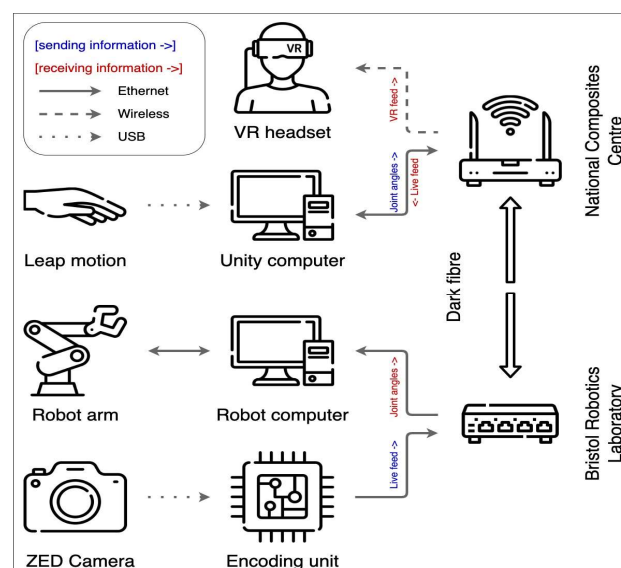


Figure 37: Network diagram

The four major links define what can directly affect the operator's experience and are considered inside the research scope of this feasibility project. It should be noted that a link can consist of multiple connections of which one or more can be the bottleneck effectively enforcing a maximum bandwidth.

- C1: Robot - Unity Computer
- C2: Camera - Unity Computer
- C3: VR-headset - Unity Computer
- C4: Dark Fibre

This meant that before the robot connection with the Unity computer was integrated, it was tested on its own locally before being tested in combination with the dark fibre connection and eventually being integrated in the complete network set-up. These steps were taken for all identified major connection links.

These connections are also a major part of the feasibility study as each comes with different network bandwidth requirements that could have potentially influenced network latency. Before these requirements were established it was important to have the limitations identified from the current trial network set-up. The table below shows all the theoretical maximum bandwidths associated with the connection between the devices.

Connection	Facilitator	Max bandwidth
VR headset - Wi-Fi access point	2.4 Ghz Wi-Fi	300 Mbps
Leap motion - Unity computer	USB 2.0	480 Mbps
Unity computer - Wi-Fi access point	Cat 6 Ethernet	1 Gbps
Robot arm - Robot computer	Cat 6 Ethernet	1 Gbps
Robot computer - Network switch	Cat 6 Ethernet	1 Gbps
ZED camera - Encoding unit	USB 3.1 Gen1	5 Gbps
Encoding unit - Network switch	Cat 6 Ethernet	1 Gbps
BRL - NCC (Dark Fibre)	OS2 9/125 (1310nm) Single Mode	10 Gbps

Table 2 : Network table theoretical maximum bandwidth

The figure below places the theoretical max bandwidth connections on top of the network diagram. The individual links are limited by different bandwidths. All connection links that were part of this feasibility study (C1 & C2) have a maximum network bandwidth of 1 Gbps.

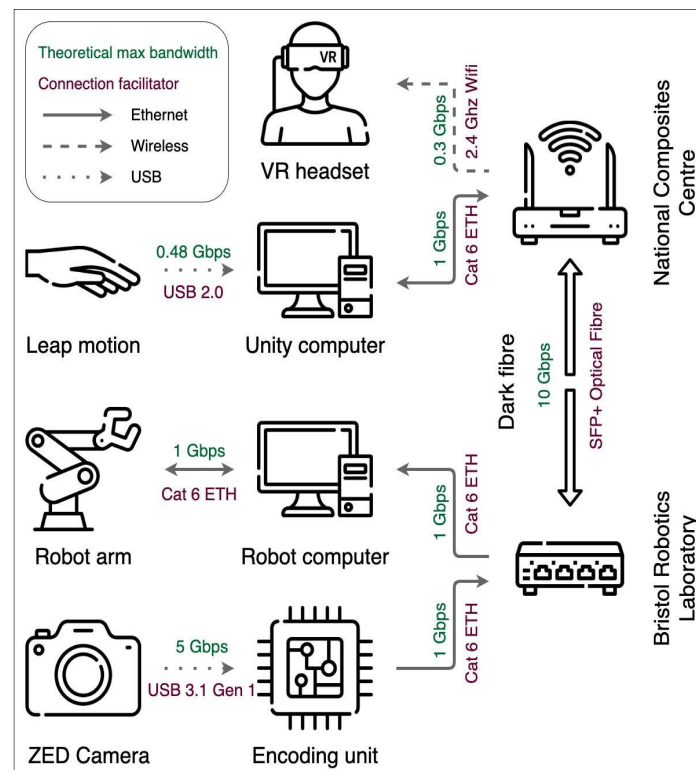


Figure 38: Network diagram max bandwidth

3.5.2 Consideration for using 5G

It is important to determine how 5G capability can be added to this project before assessing feasibility of adding 5G to the network. The current 5G network bandwidth capability (taken from network report issued to DCMS Jan-2022) is stated in the table below.

Throughputs	Values [Mbps]
Uplink	57
Downlink	410

Table 3: 5G-Network Throughputs

Currently there are two switch type devices used which allow for the cabled dark fibre network to be extended to the specific devices. These switch type devices could be replaced with 5G nodes to make all connections wireless. The implementation of the two nodes into the network architecture is visualised in the diagram below. On top of that the revised speeds are overlaid on the connections to get a clear understanding of the capability.

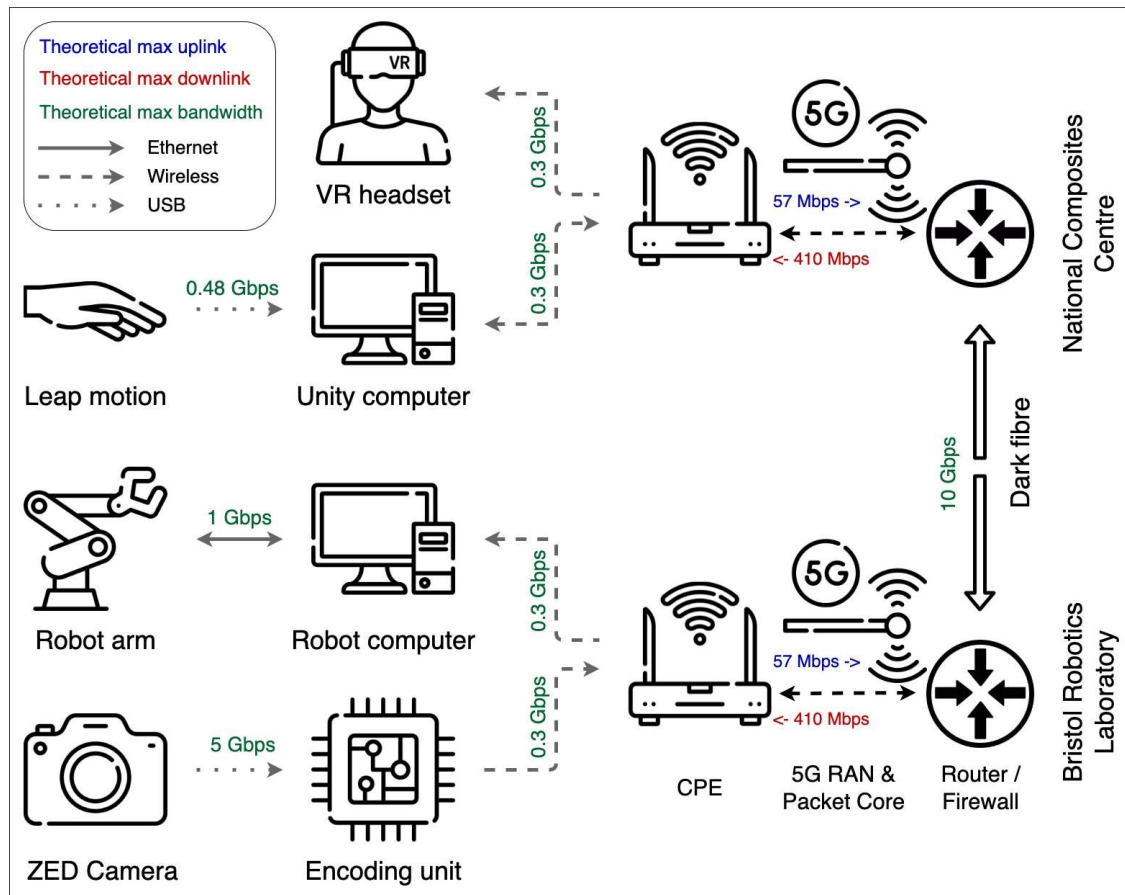


Figure 39: 5G-Network diagram max bandwidth

It should be noted that the throughputs described in Table 3 is per device. This means that every device connected to the 5G node can use up to the specified bandwidth. These bandwidths are considerably lower than that of the cabled trial network. This shows the necessity of doing a trial to test network bandwidth and confirm requirements. The ZED camera footage live stream will have the need for high uplink throughput, meaning there is a major concern that the 5G, as deployed in this project, cannot meet the uplink requirements of the camera. The ability to test the network prior to 5G integration has identified limitations and greatly reduced the risk of this project and created requirements for any future projects using this type of camera.

With the implementation of 5G determined and described it is now possible to start looking at the pros and cons of adding such a capability. The benefits are very dependent on the location of the implementation hence the assessment is split between the National Composites Centre and Bristol Robotics Laboratory.

Node location	Pros	Cons
National Composites Centre	<ul style="list-style-type: none"> • Full wireless operation • Enhanced freedom of movement • Flexible working area • Not location restricted 	<ul style="list-style-type: none"> • Lower uplink • Lower downlink • Potential wireless interference
Bristol Robotics Laboratory	<ul style="list-style-type: none"> • Wireless camera placement • Mobile robot teleoperation plug & play 	<ul style="list-style-type: none"> • Lower uplink • Lower downlink • Potential wireless interference

Table 4: Location and activity description

At the National Composites Centre, the main benefits relate to the increased freedom of operation. This includes being able to start conducting the operation from anywhere within the 5G network range instead of having to be connected to a specific Ethernet cable. Wireless operation also allows the operator to be untethered which gives more freedom of movement and can impact the operational performance positively. The overall time it would take to start operation of which setting-up connectivity could also be reduced.

At Bristol Robotics Laboratory, benefits are found when it comes to placement of the devices. Currently the camera location is limited by the length of the cable and can only be set-up in a limited space. Having the ability to place the camera anywhere due to the wireless connection is not only beneficial for camera placement but also opens the opportunity to investigate actively moving the camera for better overview of the environment. Further benefits could be found when a mobile robot arm system is being using wireless connectivity as this could give extended freedom and enable a plug & play form of teleoperation.

The 5G implementation to the existing teleoperation network aligns well with the use-cases envisioned in the main 5G-Encode work packages. From the AR/VR use case 5G enabled advanced human machine interfacing using a wireless high-bandwidth low-latency network connection. These results for video relay are likely to be seen in a future haptics use case trial.

4. RESULTS

During the trial tests from multiple data points were captured to identify the network performance. This makes it possible to start determining the minimal network requirements for this use case. The following network parameters were tested whilst using the current teleoperation network.

- Uplink bandwidth
- Downlink bandwidth
- Ping latency
- Ping lost percentage

Bandwidth parameters were determined using software called NetPerSec installed on the computer hosting the solution applications. This software gives an overview of the current uplink, downlink and average network traffic observed on the network port in the computer. These last two parameters were determined by running a continuous ping command whilst running the certain test scenario. The ping results were saved in a text file prior to processing in an excel file. All raw results data are in Appendix A. It should be noted that after going over the results it was found that the average numbers taken by NetPerSec were rolling averages and not the average value over time. This led to the need to interrogate the screenshot and take an estimate of the average value that would be closer to reality.

A total of seven test scenarios were determined for which each network parameter described was measured. Firstly, it was necessary to benchmark with all applications on the machine turned off. The same was then done with the application running in idle mode which in our case is just showing the main screen. Separate tests were then completed for the robot and the camera before a combined test of both operating together. It should be noted that different uplink speeds were being captured when the robot was moving opposed to not moving, hence, both were captured.

Entries	Robot move	Duration	Average	Max	Average	Max
Benchmark	N/A	10 min	3.80	12.67	3.65	10.34
Application idle	N/A	10 min	3.05	12.49	3.45	13.45
Camera only	N/A	10 min	20840	24790	4.34	14.82
Robot only	Not moving	5 min	20.27	23.70	38.00	44.00
	Moving	5 min	20.74	29.47	7.55	44.18
Robot & Camera	Not moving	5 min	22510	26950	57.78	70.40
	Moving	5 min	22140	25190	4.38	50.28

Table 5: NetPerSec network bandwidth

The table above shows the results of the network parameters of which a few findings can be discussed. It was seen that when the application is idle it is not using the network connection. It was also determined that the camera feed is the main consumer of network bandwidth. This was expected; however, it was beneficial to have determined the network bandwidth to understand network bandwidth consumption in detail.

There is a big difference in uplink bandwidth consumption dependent on whether the robot is moving or not. Investigation found that before messages can be sent through the TCP connection a check is being done as to whether the receiving end (robot computer) is waiting for a new command. The current code that is connecting to the robot arm and translating the commands does this in a sub-optimal way. For each command that would move the robot beyond a small threshold the robot arm is told to move to a certain joint angle configuration. The issue is that the robot is asked to take 0.75 seconds regardless of how big the requested movement is. For comparison, the normal loop time is 0.03 seconds but when the robot is moved this goes up to 0.078 seconds. This resulted in choppy robot movement and a connection that was not available most of the time when the robot was in operation. This means that less messages were being sent and consequently less bandwidth used.

From the values taken during trial tests the network diagram shown before was adjusted with now actual required bandwidth values.

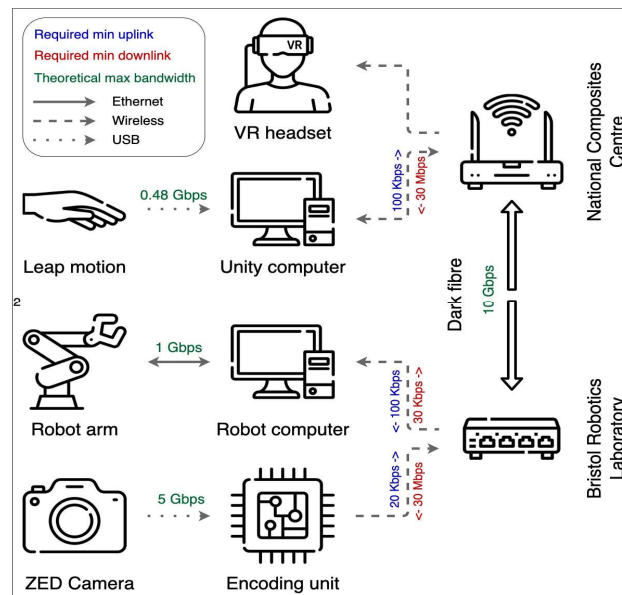


Figure 40: Network diagram required bandwidth

The ping latency results shown in table 5 shows that the current network is very stable as expected from a fully cabled network. No packets were lost, and the latency values were extremely low throughout the period of testing. These measurements create a benchmark to compare an integrated 5G network with.

Entries	Robot status	Send [#]	Lost [%]	Min [ms]	Max [ms]	Avg [ms]
Benchmark	N/A	617	0	0	5	0
Application idle	N/A	706	0	0	22	0
Camera only	N/A	615	0	0	8	0
Robot only	Not moving	313	0	0	27	0
	Moving	310	0	0	16	0
Robot & Camera	Not moving	307	0	0	15	0
	Moving	310	0	0	15	0

Table 6: Ping latency

5. CONCLUSIONS

In this work, haptic robot control protocols were integrated into a teleoperation solution where an industrial robot located at Bristol Robotics Laboratory was operated from the National Composites Centre. The need for robot teleoperation within the industry remains for use-cases where the operating environment is too hazardous for direct human operation and too complex for automation. To make further steps in robot teleoperation there is a need to develop intuitive systems that allow an operator to control the robot with confidence. A system was proposed that used hand tracking, haptic feedback and an immersive experience to actuate a robotic arm from a distance for which optical fibre and network cables were used to facilitate the network needs. The aim for this feasibility research project was stated as follows.

“To determine the feasibility of 5G connectivity elements in a robotic arm teleoperation scenario where haptic control is used”

In order meet the aim of this project a teleoperation solution was created. This required the creation of a teleoperation network, a haptic robot arm actuation solution and an immersive experience. These were all successfully created within the duration of the project.

To control the robotic arm a haptic hand tracking device was used. A haptic protocol developed in a previous research project was used to translate the input of the operator to the output of the robot. The so-called PTV (Position to Velocity) translates the position of the operator's hand to a velocity vector placed on the robot's tool centre point. The virtual robot arm follows the movement of the desired tool centre point (TCP) and actively calculates its joint angles using inverse kinematics. These joint angles are then sent to the physical robot using a direct “Transmission Control Protocol” connection.

The immersive experience used is enabled by a combination of stereo camera and VR-headset. The stereo camera had its feed encoded and directly streamed to the computer that ran the operational control application. This feed was then encoded by the computer and used within the application. The VR-headset allows for the operator to gain a sense of depth of field which directly impacts the operational performance.

The teleoperation network consisted of differing networking bandwidth capabilities. In an effort to reduce the project schedule risk the network was cabled, and a benchmark created

from which future 5G integration can be tested. In the centre of the network is an optical fibre connection that connects Bristol Robotics Laboratory to the National Composites Centre. This connection was able to have a bandwidth of up to 10 Gbps. The rest of the major connections were limited by CAT 6 Ethernet cables and standard issue network adaptors meaning that the overall max bandwidth of most parts on the network was 1 Gbps.

To determine the solution's network requirements a number of tests were conducted. During these tests all systems that make up the teleoperation solution were used. This means that a camera was sending footage from Bristol Robotics Laboratory to remote operator whilst that operator was actuating the robot arm from the National Composites Centre. During these trials a maximum bandwidth of 27 Mbps was measured. The current 5G networking capability allows for an uplink of 57 Mbps and a downlink of 410 Mbps. It is concluded that the current 5G capacity meets the requirements of this solution and potentially has capacity for some other users on the radio interface in use.

An additional demonstrator was created as an extra deliverable on top of the project deliverable. This demonstrator shows the solution working to pick and place items in a nuclear decommissioning operation.

BIBLIOGRAPHY

- [1] Emiel Evert Hendrik Bos. Haptic control protocols for simulated industrial robot control, 2021. 1, 4, 8, 10, 11, 15, 22
- [2] Guanglong Du, Ping Zhang, Jianhua Mai, and Zeling Li. Markerless kinect-based hand tracking for robot teleoperation. *International Journal of Advanced Robotic Systems*, 9, 2012. 4, 5
- [3] Goran Vasiljevic, Nikola Jagodin, and Zdenko Kovacic. Kinect-based robot teleoperation by velocities control in the joint/cartesian frames. volume 45, 2012. 4, 5
- [4] Jose Avalos and Oscar E. Ramos. Real-time teleoperation with the baxter robot and the kinect sensor. volume 2018-January, 2018. 4
- [5] D. Bassily, C. Georgoulas, J. Gu'ttler, T. Linner, and T. Bock. Intuitive and adaptive robotic arm manipulation using the leap motion controller. 2014. 4
- [6] Sunjie Chen, Hongbin Ma, Chenguang Yang, and Mengyin Fu. Hand gesture based robot control system using leap motion. volume 9244, 2015. 4, 7, 8
- [7] Bin Fang, Di Guo, Fuchun Sun, Huaping Liu, and Yupei Wu. A robotic hand-arm teleoperation system using human arm/hand with a novel data glove. 2015. 4, 6
- [8] Jeremy A. Fishel, Toni Oliver, Michael Eichermueller, Giuseppe Barbieri, Ethan Fowler, Toivo Hartikainen, Luke Moss, and Rich Walker. Tactile telerobots for dull, dirty, dangerous, and inaccessible tasks. 2020. 4, 5, 6
- [9] Michal T'olgyessy, Martin Dekan, Lubo's Chovanec, and Peter Hubinsky'. Evaluation of the` azure kinect and its comparison to kinect v1 and kinect v2. *Sensors (Switzerland)*, 21:1–25, 1 2021. 5
- [10] Frank Weichert, Daniel Bachmann, Bartholom'aus Rudak, and Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors (Switzerland)*, 13:6380–6393, 5 2013. 6
- [11] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. Hand gesture recognition with leap motion and kinect devices. 2014. 7
- [12] A. H. Butt, E. Rovini, C. Dolciotti, G. De Petris, P. Bongioanni, M. C. Carboncini, and F. Cavallo. Objective and automatic classification of parkinson disease with leap motion controller. *BioMedical Engineering Online*, 17, 2018. 7
- [13] Bin Hu and Jiacun Wang. Deep learning based hand gesture recognition and uav flight controls. *International Journal of Automation and Computing*, 17, 2020. 7

- [14] UltraLeap. Introducing leapuvc: A new api for education, robotics and more. <https://blog.leapmotion.com/leapuvc/>, 2018. [Online; accessed 14-Sep-2021]. 7
- [15] UltraLeap. How to build your own leap motion art installation. <https://medium.com/@LeapMotion/building-a-leap-motion-art-installation-38062e085955>, 2015. [Online; accessed 14-Sep-2021]. 7
- [16] Mehdi Sobhani, Manuel Giuliani, Alex Smith, and Anthony G Pipe. Robot teleoperation through virtual reality interfaces: Comparing effects of fixed and moving cameras. volume 2657, pages 1–9. CEUR-WS, 2020. 9

APPENDIX

NetPerSec Results

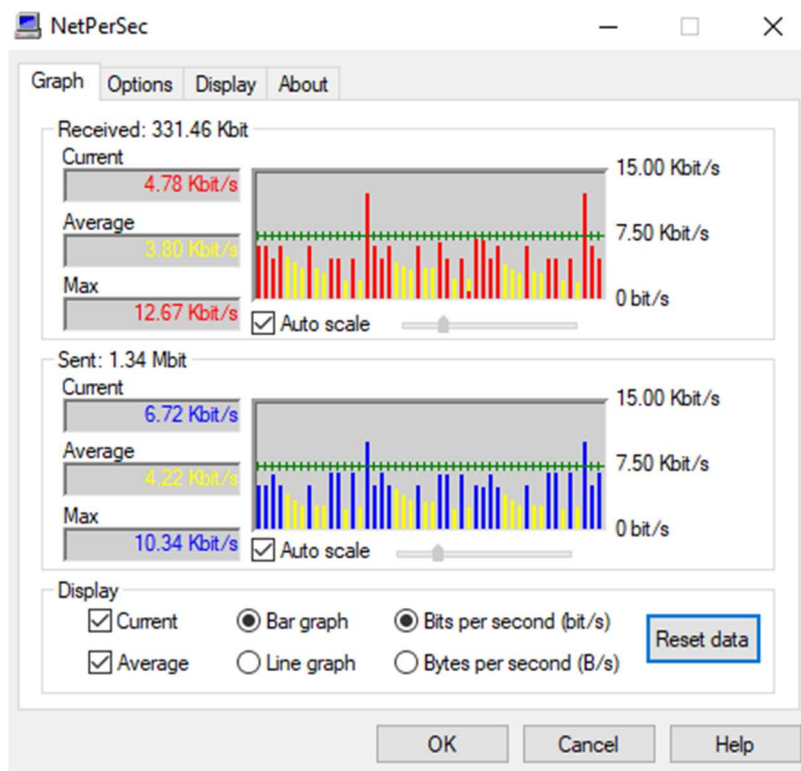


Figure 41: NetPerSec baseline

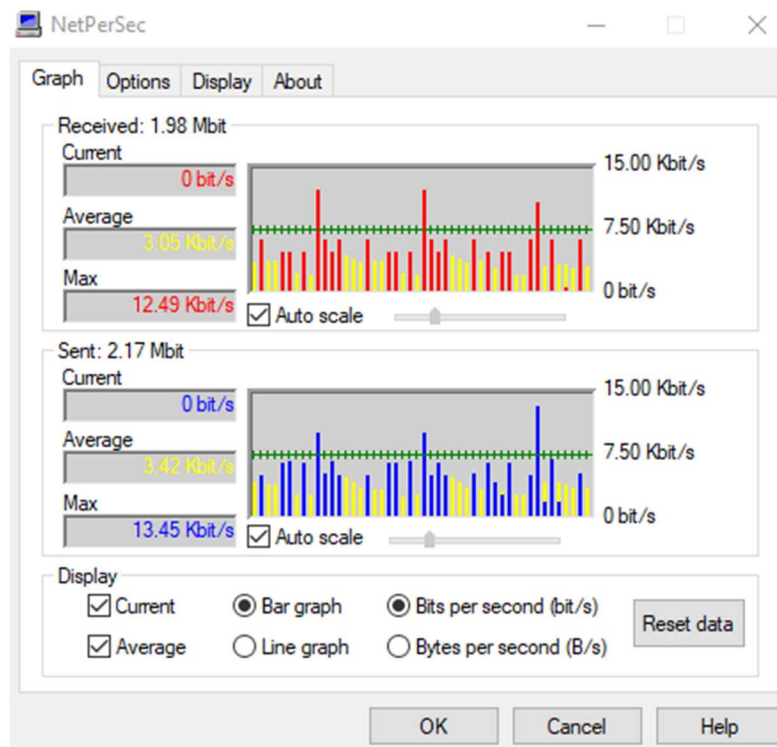


Figure 42: NetPerSec application idle

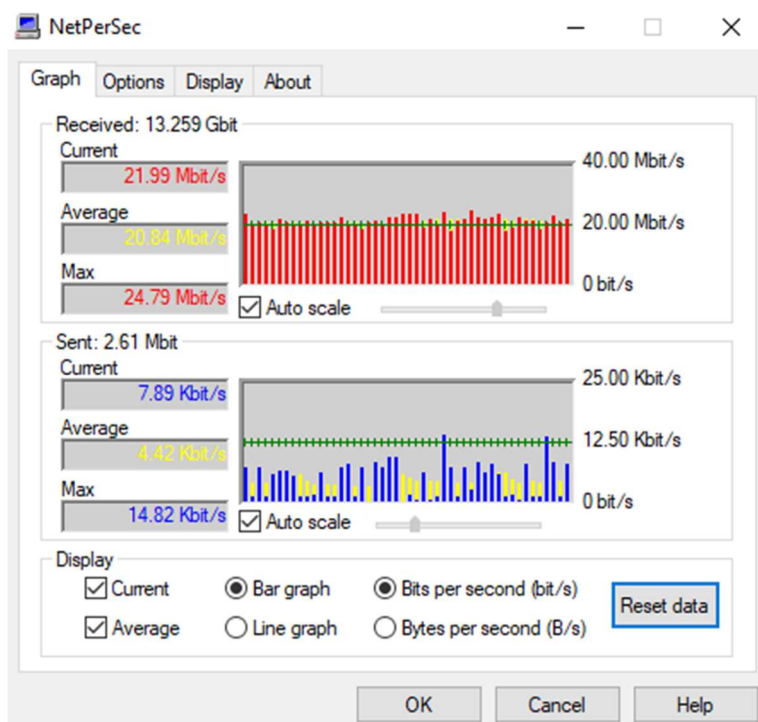


Figure 43: NetPerSec application camera

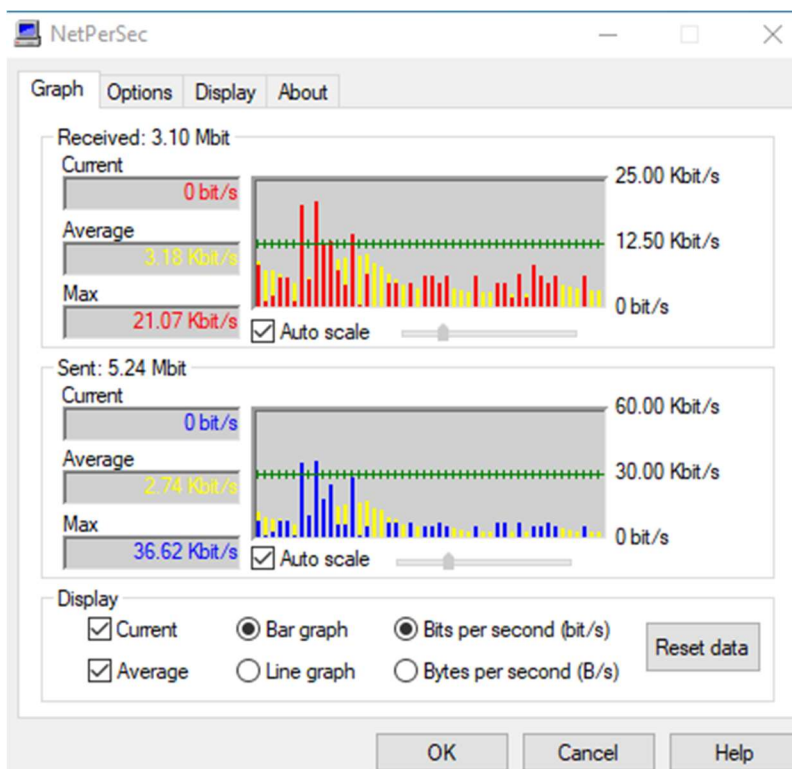


Figure 44: NetPerSec robot moving no camera

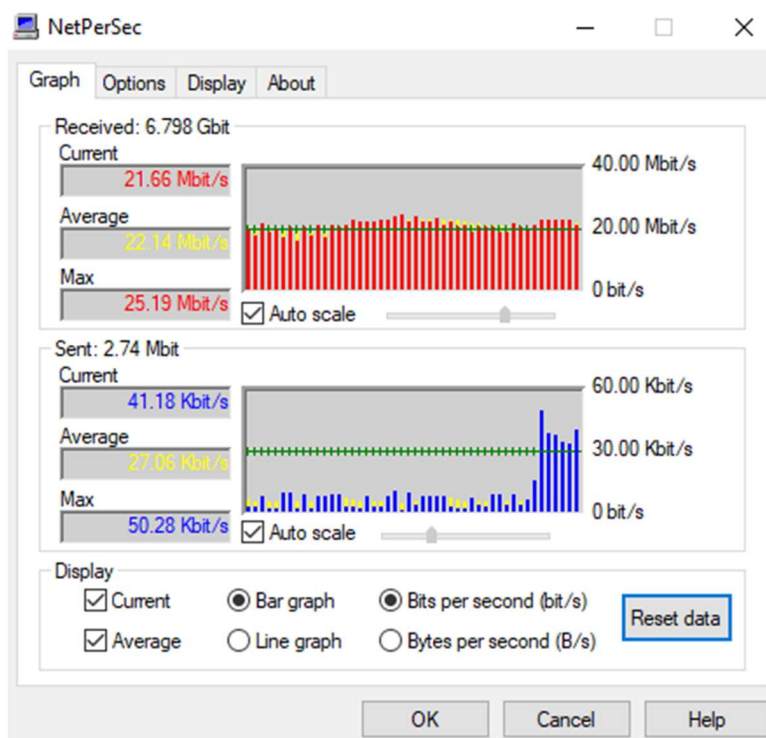


Figure 45: NetPerSec robot moving camera

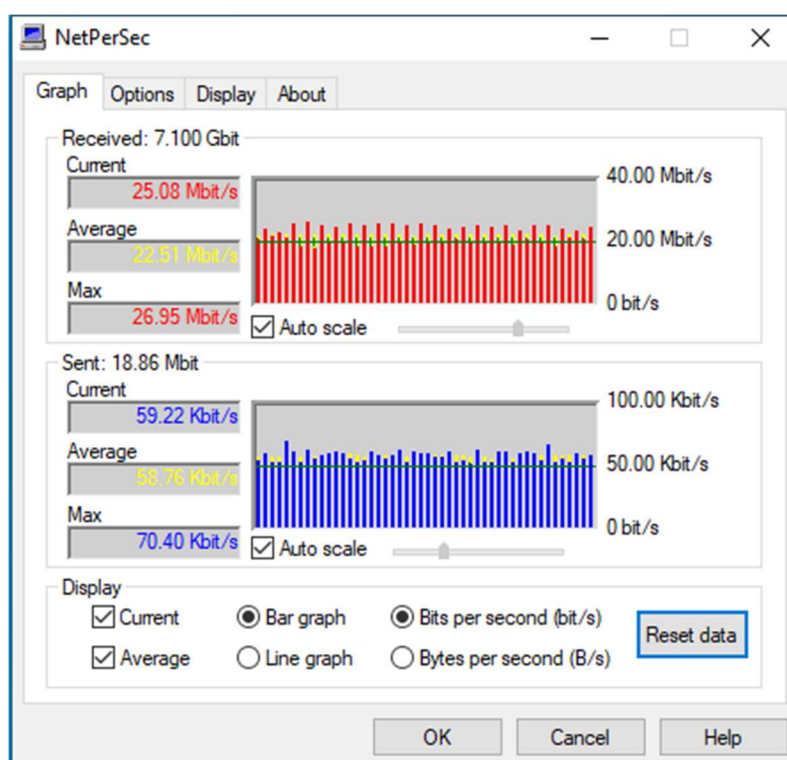


Figure 46: NetPerSec robot not moving camera

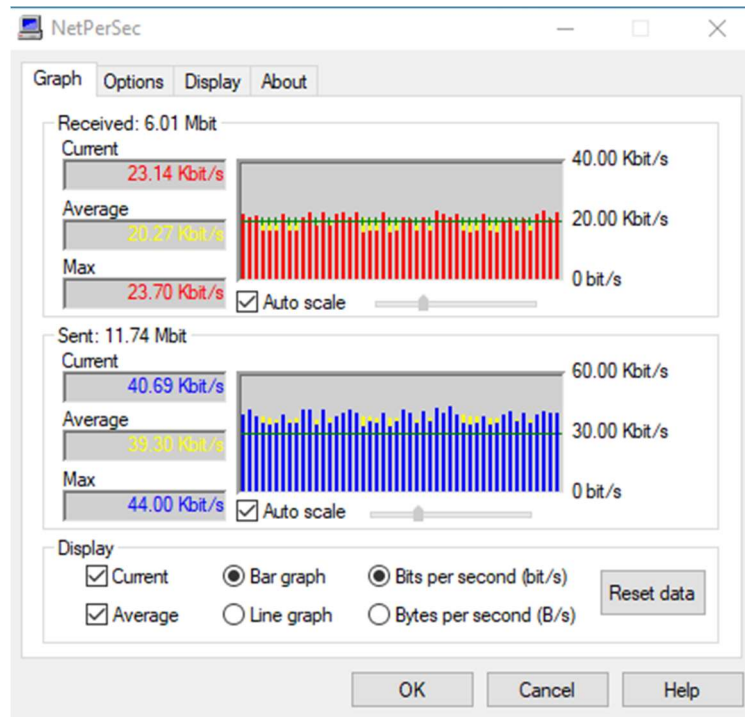


Figure 47: NetPerSec robot not moving no camera