

ATHABASCA UNIVERSITY

VIRTUAL TRAINING LAB (TLAB): TELEPRESENCE EMPOWERED REMOTE LAB

BY

NAQI U. KHAN

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION SYSTEMS

FACULTY OF SCIENCE AND TECHNOLOGY

ATHABASCA, ALBERTA

DECEMBER 2022

©NAQI U. KHAN

## **Approval Page**

The undersigned certify that they have read the thesis entitled

### **VIRTUAL TRAINING LAB (TLAB): TELEPRESENCE EMPOWERED REMOTE LAB**

Submitted by

**Naqi Khan**

In partial fulfillment of the requirements for the degree of

### **Master of Science in Information Systems**

The thesis examination committee certifies that the thesis  
and the oral examination is approved

**Supervisor:**

Dr. Qing Tan (Athabasca University)

**Committee Member:**

Dr. Ali Akber Dewan (Athabasca University)

Dr. Kadry Ali Ezzat Kadry (Higher Technological Institute)

January 25, 2023

## **Abstract**

Rapid growth of telepresence technology led to the creation of Telepresence Robot Empowered Smart Lab (TRESL) to overcome the challenges in distance education, lab and field work. This research created a similar setup (virtual and physical laboratory) over the broadband internet, tested numerous scenarios including dispersed and centralized system and discovered logistical challenges with performance, network connectivity and user training for controlling the equipment. Collected data showed the potential downsides of broadband connection, last mile problem occurred due the shared bandwidth among local consumers also, it was observed that user training is required to manipulate controls from a remote location. This research highlights the question of operational reliability of internet connection for telepresence technology specifically for laboratories which often requires machine-like point to point accuracy. It is recommended to explore technologies like Direct Internet Access (DIA) and suggested optimization of virtual device (digital twin) for increasing the performance and reliability.

*Keywords:* Telepresence Technology, Robotic Arm, Digital Twin, Smart remote Lab, Online Learning, Cyber-Physical System, Network connectivity

## Table of Contents

Approval Page.....	ii
Abstract.....	iii
Table of Contents .....	iv
List of Tables .....	vii
List of Figures.....	viii
Chapter 1 Introduction.....	1
Research Background.....	1
Research Issue.....	2
Research Objective.....	3
Research Methodology .....	4
Research Contributions.....	5
Research Scope .....	5
Chapter 2 Literature Review .....	7
Telepresence Technology .....	7
Teleoperation for Modern Lab .....	7
UTAUT and PAM for Telepresence Technology .....	9
Spatial Constraint for Telepresence.....	9
Skill Gap in Tele/Remote Operation .....	10
Telepresence Remote Lab User Training.....	10
Human-in-the-Loop Cyber Physical System and AR/VR .....	12
Digital Twin in Telepresence Empowered Lab.....	14
Digital Twin Platforms .....	15
Microsoft Azure.....	16
3D Unity.....	16
AWS IoT Twin Maker .....	17
Challenges of Digital Twin Technology.....	17
Design Gap and Potential Issues in Telepresence Technology .....	17
Network Reliability and Robustness .....	18
Last Mile Problem (Network Congestion).....	19
Video Handling .....	19
Audio Feedback.....	20
Haptics in Telepresence.....	20

Cyber Security..... 21

Legal and Ethical Concerns ..... 21

Chapter 3 Research Design..... 22

Design of the Ubiquitous Platform..... 22

Physical Robotic Arm..... 22

System..... 23

Raspberry Pi..... 24

Webserver ..... 24

Gunicorn..... 24

Flask..... 25

Camera..... 26

Networking..... 26

Digital Twin ..... 26

Chapter 4 Research Implementation and Results..... 28

Testing Strategy..... 28

Scenario 1..... 29

Scenario 2..... 29

Scenario 3..... 30

Scenario 4..... 30

Network Hops ..... 30

Test Results of Each Scenario..... 31

Server Speed Check..... 33

Training..... 33

Chapter 5 Opportunities and Challenges ..... 39

Opportunities and Challenges ..... 39

Research Implementation Challenges..... 41

Synopsis ..... 42

Chapter 6 Conclusion and Recommendations ..... 44

Conclusion ..... 44

Recommendations ..... 45

Improving Total Blocking Time (TBT) ..... 46

Remove unused JavaScript..... 46

Reduce bloated JavaScript Libraries..... 46

Defer unused JavaScript..... 46

Utilize Code splitting ..... 46

References..... 48

Appendix A - Program Code: Description and Attachments ..... 54

    Install Raspberry Pi..... 54

    Install Nginx ..... 54

    Optimize Nginx ..... 55

    Install Guicorn ..... 60

    Install Flask ..... 61

    Troubleshooting..... 64

    Networking..... 64

Appendix B - Certification of Ethical Approval ..... 67

### List of Tables

Table 1 Qualitative Questionnaire Survey Results.....	11
Table 2 Network Hops Statistics for 4 Scenarios .....	31
Table 3 Total Experiment Time .....	32
Table 4 Total Experiment Time Graph .....	32
Table 5 Server Metrics (Dispersed).....	33
Table 6 Training on Physical Arm (Before) .....	35
Table 7 Training on Digital Twin .....	36
Table 8 Training on Physical Arm (After) .....	37
Table 9 Training Statistics .....	38

**List of Figures**

Figure 1 Remote Lab .....9

Figure 2 Human-in-the-Loop Cyber-Physical System (HiLCPS) .....13

Figure 3 Sample of System Developed for This Research .....23

Figure 4 Electronic Parts Assembled for Physical Laboratory .....24

Figure 5 Webserver, Guicorn and Flask Webserver, Guicorn and Flask.....25

Figure 6 Digital and Physical Robotic Arm Screen .....27

Figure 7 Controlling Physical Arm Using its Digital Twin .....27

Figure 8 Centralized System Setup .....29

Figure 9 Dispersed Systems Setup A .....29

Figure 10 Dispersed System Setup B .....30

Figure 11 Dispersed System Setup C .....30

Figure 12 Training .....34

Figure 13 Evaluating on Actual (Before).....35

Figure 14 Training on Digital Twin .....36

Figure 15 Evaluating on Actual (After) .....37



## Chapter 1 Introduction

### Research Background

The concept of presence involving interactions via or created by technology is receiving an increasing amount of attention. (Lombard & Jones, 2015). Modern-day telepresence can be described as video conferencing on wheels created for ad-hoc conversation or for patients at medical facilities (M. Desai, 2011) but Telepresence Robot Empowered Smart Lab (TRESL) was developed to allow student ‘to be there and act there’ by the telepresence to perform experiments in the remote laboratory (Qing Tan, 2019).

Overall, Telepresence technology has developed over the period and expanded into various extensive fields, but its apparent diversity has made it difficult to correctly perceive how each piece is situated within the greater whole (Irene Rae, 2016). Also, actions requiring more than just an audio/video exchange could be shadowed greatly by the network performance. The purpose of this research is therefore to propose not only the idea of the training lab (TLAB) to allow students and scientists to understand what to realistically expect but to also address the importance of a perpetual and reliable network connectivity for the success of Telepresence Robot Empowered Smart Lab (TRESL) (Qing Tan, 2019).

Virtual labs are typically designed to allow simulated scientific experiment to be conducted by using nothing but a software. Although a remote laboratory makes the use of tools like virtual instruments (VI) and e-learning environments, they should not be mistaken for either simulation or virtual laboratories (Guzman, Lopez, Enriquez, Mathey, & Ramirez, 2017). Performing scientific experiment remotely not only demands for a reliable network connection but also requires users to be trained for computer peripherals like a mouse, keyboard to be calibrated/configured to understand the sensitivity level since the input dynamics of these tools

are somewhat different than their standard usage. In order to consider and demonstrate the need, a laboratory setup is created as a result of this research which can also be served by way of a training resource (sandbox) for students/scientists of any skill level to carefully build tapping, tracking, speed, intuitive perception, cognition skills and generate confidence in all areas before efficiently performing a successful experiment requiring utmost accuracy at a faraway location.

Similarly, internet connectivity is the backbone for any telepresence or remote lab experiment, generally a slow internet connection can cause considerable frustration let alone the issues with its stability which can introduce critical challenges to such operations. There is always a question of choosing between the practical convenience of Wi-Fi or the operational reliability of a wired connection. Although wired connections are tentatively linked to the faster and reliable internet connection setup, there are some potential downsides which include the last mile problem due the shared bandwidth among the local consumers in the area. This research subsequently highlights the essential question about the operational reliability of internet connection for running experiment with TRESL (Qing Tan, 2019)) which has a potential for performing meticulous scientific experiment on hazardous, non-reachable substances, environments such as nuclear objects or at a remote location requiring machine-like point to point accuracy. Some possible solutions are recommended in the end with suggestion to extend the research further on the topic.

### **Research Issue**

Virtual laboratory is entirely simulated by a software that can be instantly accessed by installing it on a computer or by a dedicated website (Budai, 2018) whereas, all the equipment in a remote laboratory are physically situated on their fixed positions somewhere in the institution and experiments are conducted from a remote location. But with the telepresence technology,

experiment can be conducted by moving from equipment to equipment using a telepresence robot and operating its actuator (robotic arm) from a remote location. Piloting of the telepresence robot can be done from a PC by “click and go” buttons and arrow keys (Kristoffersson, Loutfi, & Coradeschi, 2013) but controlling the robotic arm for a novice operator by a desktop mouse naturally requires training and customization, configuration/ calibration based on the specific user’s adaptive behaviours, etc. Training to operate telerobot (Labonté, 2010) was conducted and significant results were achieved within 2 hours. Also, Trilab a pedagogical model (Abdulwahed, 2010) was presented with triple access mode (virtual, hands-on and remote) laboratory idea emphasizing Kolb’s experiential learning cycle (Experiential Learning, n.d.) as a way of conducting laboratory education based on constructivist educational theories but at the time of this research, no resources were found that can enable a user conducting an experiment by telepresence to practice and develop skills, acquaint with the responsiveness of the components etc.

Similarly, it has never been a simple task to carry out optimal application experience from a network perspective, a web tool or locally installed application connecting to the remote lab could be all powerful and full of innovative features but having an unstable connection between the active user and remote lab could typically trigger critical issues at the lab. This research also put forward a question for an internet connection that is totally dedicated to a specific location like Direct Internet Access (DIA) a fully dedicated bandwidth that is not shared and the speed does not vary (n.d.) and if it can help resolve.

### **Research Objective**

- To review the core subject from literature including journal and conference papers, online materials such as surveys as well as industrial reports.

- To explore for a possible training solution available for telepresence remote lab users.
- To explore network connectivity issues while engaging telepresence technology for remote lab experience.
- To examine and modify soft configurations of the virtual device (digital twin) for better performance.
- To examine data collected while experimenting with remote lab.

### **Research Methodology**

As a likely result of this research, a proactive approach is developed to establish that telepresence technology for remote lab operation must be paired up with the uninterrupted network connectivity and a recommendation of training telepresence remote lab users to bridge the skill gap for safe operations.

This research focus on the user experience of a remote lab using telepresence technology, only responses meeting the following criteria will be included in this thesis:

- Research not older than the year 2010.
- Reviews on the core subject (remote labs) from literature including journal and conference papers, online materials as well as industrial reports.
- Results pertaining to the project experience related to TRESL, are not tied to the capabilities of a vendor.
- Design a prototype of a laboratory system to implement and evaluate the concept and develop features and specific functionalities.
- Factors to consider and constraints for remote experimentations.
- Review and discern Tele operation, virtual labs.

## **Research Contributions**

This research contributes to the area of telepresence robots in remote laboratories. Specifically, it explores techniques such as centralized vs dispersed setup to test laboratory equipped with physical robotic arm and its digital twin in the related context as well as experimental systems research in general. The primary objective of this research is to test the hypothesis that:

1. training practices can help users to develop confidence in using the telepresence robot remotely,
2. uninterrupted network connectivity is the backbone for any telepresence experiment,
3. solutions are available for better and safer operations as compared to broadband internet connections.
4. TRESL (Qing Tan, 2019) claim may not be practical unless paired with industry standard internet solutions such as Direct Internet Access (DIA) etc.

It should be noted that it is not possible to formally prove the correctness or falsehood of this hypothesis. Instead, this dissertation is limited to providing, hopefully strong, evidence for or against its validity. It does so by introducing a new test lab idea dedicated for telepresence robots in remote laboratory. I am thankful to Dr. Qing Tan and his publication “Toward a telepresence robot empowered smart lab” for providing guidance throughout this paper.

## **Research Scope**

The scope of this thesis is to provide information on the importance of the network stability and how it can introduce critical challenges due to the possible downsides of broadband connection and last mile problem which may occur due the shared bandwidth among the local

consumers in the area. Also it is to suggest a tool required for training for telepresence remote lab users allowing to practice and familiarize with the key components and issues that may arise during the process such as communication time delay due to network lag, reduced feedback, and sensory input due to restrictions of communication capacity, limitations of computation power at hand, reduced dexterity, reduced workspace etc.

A successful remote handling of any laboratory typically depends on operator's adaptation to the system; it will also be observed how: -

- Slow and steady movements are necessary to control the robotic arm any quick and abrupt movement could jeopardize the experiment.
- Configuration of mouse, modern keyboard settings and improving appropriate skills are necessary for operating sophisticated equipment remotely.
- Training with the tool will reveal the equipment limitation, surrounding environment, understanding of video refresh rate etc.
- Network lags, slow performance and how it can impact the ongoing experiment.

Subsequently, this research will investigate on how a dispersed vs centralized setup facilitate telepresence technology user and address a likely solution that can help telepresence remote lab concept make progress.

## **Chapter 2 Literature Review**

### **Telepresence Technology**

Objective forms of presence or telepresence is to easily observe and confirm via modes and technologies of communication (Lombard & Jones, 2015). Modern advancement in technologies has provided an apparent ease for computing, controlling, sensing, dealing with mechanical elements in a remote location, specially for the academic institutions. But the logistics of tailoring a real laboratory, for many students, often present a crucial problem for universities; the requirements for considerable space, modern instrumentation, and human support are high (Xuemin Chen, 2010).

Interest in telepresence robots is at an all time high, there has been a surge in companies designing telepresence robots (M. Desai, 2011) for various domains and applications. Although the concept of telepresence robots is very broad, the main objective of these systems is to represent a human operator (remotely located), giving the person the ability to interact with the environment (Macharet & Florencio, 2012) in this context one of the solutions was presented by Telepresence Robot Empowered Smart Lab (TRESL) system which allows student to ‘to be there and act there’ with the help of a tele present robot taking students avatar in the remote laboratory (Qing Tan, 2019).

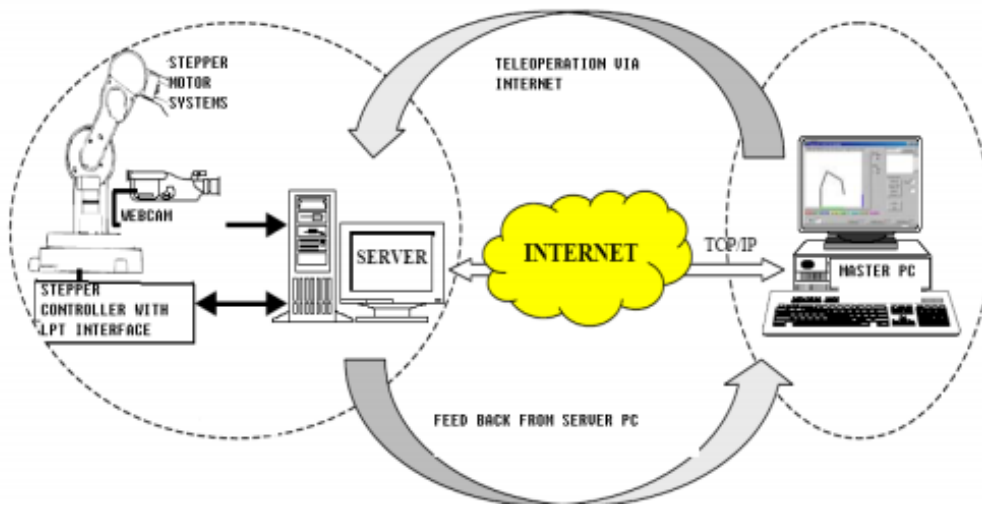
### **Teleoperation for Modern Lab**

Robotic telepresence systems that allow a remote user to drive around in a remote location provide an alternative to video-mediated communications as a way of interacting over distances (Rae, Mutlu, & Takayama, 2014). This idea gained momentum during the pandemic of Covid-19 when students did field and lab work without leaving home (Pennisi, 2020). There are

two traditional rather innovative approaches available for efficiently conducting labs online, remote laboratory where the real equipment is controlled by means of a desktop computer, mouse, keyboard via a network (Harms, 2000) as illustrated in Figure 1 (Vershinin, 2014) or by the Augmented Reality such as ARRL (augmented reality for remote laboratories (Andujar, 2010) and a virtual laboratory is entirely simulated by a software that can be instantly accessed by installing on a personal computer or by a dedicated website (Budai, 2018) but now, there is another type that is enough to be called a viable hybrid of both, a smart lab empowered by a controlled robot ‘to be there and act there’ by the telepresence for performing experiments in the remote laboratory (Qing Tan, 2019)

In recent years, research has pointed to many technological and psychological factors considered important in drawing the experience of telepresence but only few theories exist that can combine these findings in a coherent framework (Haans & IJsselsteijn, 2012). Though it may seem strange with the advancement in modern technology, human intermediation, however, remains imminent and required to be adequately trained for teleoperation technologies. Cognitive types of cyber-physical systems learn from the environment, human and from itself but human intermediation remains vital for the system’s decision-making process (King, n.d.). Similarly, telepresence technologies specially telepresence in a remote ream should be trained for better results but it seems there is no research addressing its training and sandboxed testing, in this context, in order to train a novice lab user to develop skills needed to carefully perform the experiments or to practice delivering accurate instructions to the precision instruments needed in the untried procedure.



**Figure 1***Remote Lab***UTAUT and PAM for Telepresence Technology**

Unified Theory of Acceptance and Use of Technology (UTAUT) (Venkatesh, Thong, & Xu, 2012) is one of the several models emerged for user acceptance and the use of a technology. HANCON (Han & Conti, 2020) is a derivative of UTAUT and Post Acceptance Model (PAM) claiming to be a new acceptance model to study robotics applications in social context. This model was used to predict and explain the acceptance of social telepresence robot and suggested it could make learning more effective.

**Spatial Constraint for Telepresence**

The concept of spatial presence emerged from the observation that users of virtual reality systems feel physically present in the mediated space (Hartmann, et al., 2016). While reasonable progress has been made several prominent challenges are still present in the development and use of telepresence robots, specifically when it comes to the spatial presence (Lombard & Jones, 2015), exocentric viewing constraints were identified for spatial presence, objects were removed physically to avoid interference (Igor M. Verner, 2018).

### **Skill Gap in Tele/Remote Operation**

In most current remote laboratories for mobile robots, the control which users have over the robot is limited creating a gap in the learning experience of the students (Franco-Vera, Chen, & Li, 2020). Augmented Reality (AR) based Remote maintenance shown to be an effective solution. (Malecaj, 2021) but even with this advanced technology skill gap is one of the critical issues (Malecaj, 2020). One of the effective ways to improve this gap is to improve the quality of technical training (Semenets, 2019). TLAB's controlled environment will reliably deliver the virtualized resources similar to what is typically found in the laboratory. It will help to increase the spatial awareness which represents the most significant factor for remote laboratory operations

### **Telepresence Remote Lab User Training**

The development of telepresence technology and its expansion into various domains originated Telepresence Robot Empowered Smart Lab (TRESL) for conducting experiments in remote laboratories. In an ideal robotic telepresence system, user would observe as being fully present at a remote location which can typically happen if the user had a first-person experience of the location including full sensory stimulation (Cohen, Lanir, Stone, & Gurevich, 2011). To measure the effectiveness of the remote laboratory system, a preliminary assessment study was conducted (Dr. Mert Bal, 2014) among the total of 65 student (on campus and distant learners) over the course of 3 semester by using developed client-side software application to operate the robotic arm while watching its movement via web camera. On campus students were able to perform better because they were aware of the lab surrounding.

Most robotic systems are evaluated in laboratory setting for a limited period. The limitation of lab evaluation does not consider the different challenges imposed by the fielding of

robotic solutions into real contexts (Amedeo, Gabriella, Andrea, & Lorenza, 2013).

Correspondingly, performing an experiment by operating a robotic arm mounted on a movable vehicle represent a challenge and typically requires potential users to be trained appropriately also, computer peripherals like a mouse, keyboard to be calibrated/configured properly to understand the sensitivity level as well performing safe operations on the scientific equipment in remote laboratories.

**Table 1**

*Qualitative Questionnaire Survey Results*

RUBRICS	GROUP 1: Local Students		GROUP 2: Distance Students	
	<i>(1=Strongly disagree, 2=Disagree, 3=Agree, 4=Strongly agree)</i>			
	MEAN	SD	MEAN	SD
The lab sessions were motivating for me to learn more about Industrial Robotics systems.	3.8	0.54	3.5	0.58
I got familiar with hands on practice of industrial robots with the lessons I learnt in this lab experience.	3.2	0.987	3.2	0.96
This method helped me learn the basics of industrial robot control and programming.	3.6	0.675	3.7	0.71
I would like to have lab sessions like this to help me learn.	3.8	0.55	3.8	0.59
Overall, I have had fun with this lab.	3.6	0.88	3.75	0.82

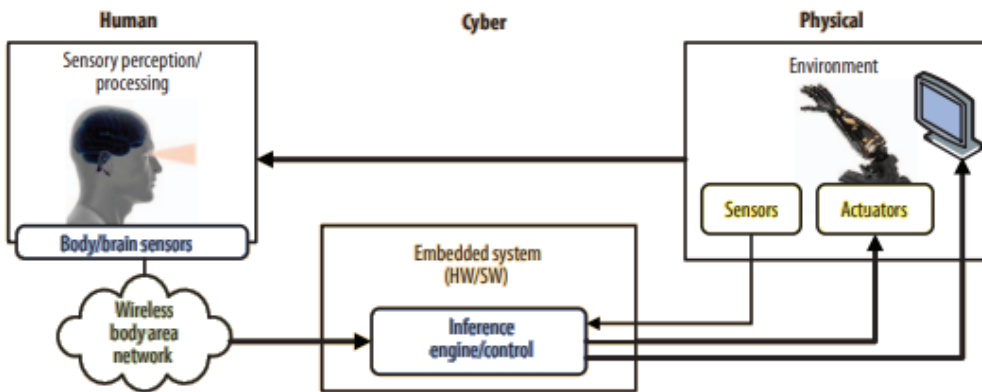
Question number 4 from the above chart (Dr. Mert Bal, 2014) attains the highest score overall for both on campus and remote learners which proves a key point that if a training resource is amply provided, students will not only just learn better, but they will equally perform better while carefully conducting proper experiments.

Telepresence remote lab user training will help to familiarize and train for dealing with the precision instrument in the lab (such as a robotic arm in TRESL), performing scientific

experiment for on hazardous, non-reachable substances or environments such as nuclear objects, conducting experiment on laboratories in space, distant planet like Mars or for a medical student who may be performing a surgery at a remote location requiring machine-like point to point crosshair placement.

### **Human-in-the-Loop Cyber Physical System and AR/VR**

Similar to a telepresence system, HiLCPS (Human-in-the-Loop Cyber Physical System) is usually embedded with the systems interfacing both the human and the physical environment by means of actuators and sensors. It is a class of applications that is widely used in the assistive technology (AT) to augment and enhance human interaction with the physical environment (Feng, Quivira, & Schirner, 2016). Through a human cognitive activity, HiLCPS translate and control signal to interact with physical system via robotic actuators with human closing the loop by observing the interactions (Gunar, Deniz, Kaushik, & Taskin, 2013) in a physical laboratory with remote access through telepresence system.

**Figure 2***Human-in-the-Loop Cyber-Physical System (HiLCPS)*

In contrast video communication in professional settings has not gained the considerable popularity however notwithstanding the limited resources and low-quality solutions, services such as Skype have thrived in consumer space (Brubaker, Venolia, & Tan, 2012). On the other hand, Augmented Reality (AR), Virtual Reality (VR) as emerging technologies gained momentum across the different disciplines including research and development. AR is a powerful medium for several fields. It is where digital and physical information are co-located in a seamless space (Roo & M.Hachet, 2017) letting user to see physical environment and interact with virtual objects. VR is certainly the most advanced way to induce telepresence. It is designed to immerse a user into a fictional existence through sensory stimulation (Moura, 2017). They can create an immersive experience for telepresence technology benefitting the remote laboratory function greatly. Also, they are a perfect tool for human–robot interaction (HRI) scenarios where a robot is controlled remotely but there are various cost and operational factors associated with their deployment such as the transfer of video data as well as the power consumption (Botev & Lera, 2021).

Similarly, Cyber-physical systems present a set of advantages for telepresence technology, since it allows individual entities to work together to form complex systems with new capacities (Sanislav & Miclea, 2012) by providing the best of both worlds (virtual and physical) such as cyber physical lab to carry out automation, robotics and control experiments and practices, equipped with real control devices and VR machines for process emulation (Cyber-Physical, 2020).

### **Digital Twin in Telepresence Empowered Lab**

A virtual representation of a physical device, digital twin is defined as simulation model that is both getting data from the field and triggering actions on the physical equipment (Elisa, Stefano, Luca, & Marco, 2020). It is a comprehensive physical and functional description of a component, product, or system that could be useful in current and subsequent lifecycle phases (S. & R, 2016). Use of digital world modelling techniques in modern factories are exploited for maximum benefits from the hybrid production system (Kousi, et al., 2019) adoption of virtualization and connectivity features of IoT (Internet of Things) is on the rise to achieve meaningful outcome with data and connected technology. There is a need for remote learning and virtual reality based digital twin system for remote laboratories and online practical learning (Palmer, et al., 2021) therefore, this research will be suggesting a training tool focused on providing the training resource for TRESL (Qing Tan, 2019) by proposing a synchronous representation of the real model (robotic arm) typically mounted on telepresence empowered robots for the specific purpose of training and set expectations for real experiments.

Depending on the product magnification there are numerous digital twins distinguished by their application (IBM, n.d.), it can be a simulation model, but a simulation model may not necessarily be a digital twin (Kannappan, 2020). Simulation models for digital twin collects data

to describes and make available system behaviour, performance evaluations and quality considerations (Boschert & Rosen, 2016). Idea is to design a synchronous representation of the real model for this research that can grow into other aspects of a digital twin later.

Furthermore, IoT is considered a backbone for digital twin, IoT provides the connection and access to intelligence in the physical world and interlinked with digital twin (Immerman, 2020). AWS IoT, Thingworx, or Thingspeak are some of the renowned IoT platform for collaborating with many digital twin applications also, IoT device shadowing is a great service for conducting experiments using digital twin, service such as the AWS IoT Device shadow can make a device's state available to apps and other services whether the device is connected to IoT or not (AWS, AWS IoT Device Shadow service, 2021).

### **Digital Twin Platforms**

Digital twin is seen as the essential technology to embrace the increasing complexity of cyber-physical systems to better understand, monitor and optimize the behaviour. Digital shadows are custom-made data structures that represent a specific system for a specific purpose. Since they are shadows in following the changes of the represented system under a specific illumination, they neither need to be complete nor can they change the represented system directly (Pfeiffer, Lehner, Wortmann, & Wimmer, 2022). Digital twin provides a virtual representation or simulation of its physical counterpart containing sensors to produce the data necessary to inform its digital counterpart. By transforming the assets into IoT enabled devices, user can track and monitor the physical piece of machinery in short, it is an advanced form of simulation software using sensor data from embedded machines to create accurate, real-time simulations of assets (What is Digital Twin Software, 2022).

Data flow latency was a challenge in optimizing the digital twin developed for this research. In search of an alternative, following digital twin technology vendors were reviewed but found to be out of scope for this research.

### ***Microsoft Azure***

Azure Digital Twins is an Internet of Things (IoT) platform that creates a digital representation of real-world things, places, business processes and people. Models for Azure Digital Twins are defined using the Digital Twins Definition Language (DTDL) which is based on JSON-LD and is programming-language independent. DTDL isn't exclusive to Azure Digital Twins but is also used to represent device data in other IoT services such as IoT Plug and Play (Microsoft, 2022).

### ***3D Unity***

3D Unity platform allows developers to create a dynamic virtual copy of a physical assets, process, system or environment. It ingests data from virtually any format (e.g., BIM, CAD) and system (e.g., PLM, ERP, IoT) into Unity for interactive visualization. This platform allows users to interact with digital twins on mobile phones, tablets, PCs, and AR and VR devices and provides support for over 20 platforms, including HoloLens, Quest, Windows, Mac, iOS, Android and more. al-world (Digital-twin, n.d.). This tool could suffice the real-world digital twin requirements, but the scope of this research did not require to build such a dynamic example of a digital twin. Also, it is widely used in game development as well as the designing of training simulation systems. If the training idea of this research were to be developed as an entity, 3D Unity will be the best platform to be leveraged.



### ***AWS IoT Twin Maker***

The IoT TwinMaker is provided by Amazon's cloud platform called Amazon Web Services. With the ability to use existing data from multiple sources, this tool allows to create virtual representations of any physical environment and combine any existing 3D models with real-world data (AWS, AWS IoT TwinMaker, 2022). This tool was reviewed with the idea to install active sensors on each joint (degree) of the robotic arm to be able to capture the speed of the motors, and time elapsed and accuracy in reference to the next and preceding joint. Focus of this tool to build digital twins to help you optimize building operations, increase production output, and improve equipment performance.

### **Challenges of Digital Twin Technology**

The concept of the digital twin originated with NASA who sought effective ways to handle the difficulties associated with the Apollo 13 mission. Using data collected from sensors, a "virtual model of a process, product or service" is created (IMI, n.d.). One of the biggest challenges of a digital twin technology is the optimization of its performance, heavy data integration requirements and inherent data flow latency makes it problematic. Evaluating data requirements and sources is a fundamental for successful digital twin implementation also, since the massive amount of data is collected from numerous endpoints a potential risk for compromise increases thus requiring intense security protocols.

### **Design Gap and Potential Issues in Telepresence Technology**

Technology and design problems aren't limited to computers and software. In general, people are increasingly challenged to understand and utilize technology of all types effectively (Deutsch). Similarly, remote operation via telepresence can be misapprehended by teleoperation

if not defined correctly. Apparently, if there is no clear definition or singular context of a concept, how effectively can the concept be shared and built upon in the field or domain. For telepresence, it is assumed that the user views the remote world from the vantage point of the remote device whereas, in teleoperation operator uses a remote device to interact with the environment, while viewing that device from another perspective such as external camera (Sherman & Craig, 2019).

Telepresence terminology is widely used in social space such as online meetings, video conferencing etc., but there are qualitatively some known issues and design gaps when integrating into remote operations after all, video cameras, microphones and computer peripheral which have their own limitation are typically used in telepresence to substitute for the corresponding senses of the participant (Sherman & Craig, 2019) as well as the network which is prone to lag and distortion in connection due to the nearby interferences. Specific details of some of the key issues are as follows:

### **Network Reliability and Robustness**

Internet connectivity introduces critical challenges as telepresence robotic technology is prone to the communication issues due to the probable loss of connection on either side i.e., remote laboratory and the location from where the experiment is being conducted a slow internet connection can typically cause considerable frustration. There is always a question of choosing between the practical convenience of Wi-Fi or the operational reliability of a wired connection. Although wired connections are tentatively linked to the faster and reliable internet connection setup, there are some potential downsides which include the last mile problem due the shared bandwidth among the local consumers in the area.

Also, a remote laboratory or a remote user could be present at a location where metal objects such as elevators may possibly cause invisible network shadows (Kristoffersson, Loutfi, & Coradeschi, 2013) triggering the lag and or an apparent disconnect to the network. Social aspect of telepresence technology usage may not be affected as bad but for remote laboratory experimentation network lags, slow performance could result in serious accidents.

Subsequently, some institutions rely on dispersed vs centralized setup for various reasons to provide remote laboratory experience, but internet latency is crucial in providing reliable and efficient networked services when servers are placed in geographically diverse locations (Kwon, 2015). It becomes an issue when a user face slow file downloads, poor application performance and in our case of remote lab experiment, Quality of Service (QoS) issues with real-time services such as voice and video for conducting experiment in a remote lab.

### **Last Mile Problem (Network Congestion)**

Remote lab users will repeatedly experience last mile problem when accessing the lab from a consumer grade internet access. Last mile connection frequently creates performance problems that could influence user experience while performing the experiment. Network performance takes a hit when the data near the remote user as bandwidth is not guaranteed and generally there are no Service Level Agreements in terms of latency and availability with the internet service provider (The 'Last Mile' Connectivity Problem and How to Solve It, 2019).

### **Video Handling**

6 degrees of freedom are minimally required to operate a robotic arm in 3D space (Robotics, 2020) requiring multidirectional approach to the position and direction of the objects in the operational capacity. Telepresence robots are mounted with a single, narrow-view camera

to provide vision for the remote user videoconferencing systems which typically rely on affordances designed for stationary systems (Johnson, Rae, Mutlu, & Takayama, 2015). A main camera that can pan, tilt, and zoom independently of the robot's base possessing a field of view with a wide enough angle to emulate peripheral vision when fully zoomed out (Desai, Tsui, & Yanco, 2011) could greatly benefit the telepresence system for laboratory function. Needless to stress the importance of bandwidth required to deliver the quality video output to the end user. Any delay or buffering in the live stream during the experiment could result in the failure of the experiment. A camera for a livestream of the experiment was installed using Raspbian 'Buster' on Raspberry Pi. Delays in video feedback were noticed for end user with WIFI vs plugged into the internet modem.

### **Audio Feedback**

Telepresence allows for real time collaboration between two parties over the Internet providing live video and audio communication (Hartsell-Gundy, Johnson, & Kromer, 2015) but bidirectional communication may not be required for a robotic telepresence system performing experiment in the remote laboratory which seems true but its importance cannot be ignored for spatial awareness as well as for a laboratory experiment where experimenter ordinarily expect to listen to audio feedback as a confirmation for job being done. Audio feedback is as critical as the video for spatial awareness after all if you are not fully linked via audio, video to the remote environment, you are not tele present.

### **Haptics in Telepresence**

Perceptual aspects or haptic feedback can help increase presence in virtual environments, by adding a sensory channel such as sense of touch (Maiero, 2021), (for example: replacing a traditional mouse with a haptic mouse providing sensory feeling by vibration) to the instruments

interacting with telepresence technology could greatly enhance the performance and help to increase spatial awareness for remote experiments.

### **Cyber Security**

With the constant growth of Internet of Things, cyber threat has been increasing exponentially as data, systems and people are being connected digitally. It has been estimated that the Robotics and automation industry will grow from \$62 billion to \$1.2 trillion in the next ten years and the risk of security will grow with it (Ishaani, 2018). It is prevalent for inadequately secured systems to be hacked but if an unauthorized individual hacks into the telepresence robot's stream, it could cause more damage than a usual hacking of a system since a hacked service robot could be intentionally used to harm people or perform apparent breakdowns.

This paper will not be demonstrating a solution to secure telepresence robot but provide a suggestion in conclusion.

### **Legal and Ethical Concerns**

Generally, telepresence technology is widely used to open access to social spaces for people who can't otherwise be present thus creating legal and ethical concerns about privacy. As a result of this technology's ability to sense remotely it necessarily raise concerns about surveillance which may be particularly acute in the workplace given commercial sensitivities around materials discussed or shared in office settings. (Barabas, Bavitz, Matias, Xie, & Xu, 2015).

## Chapter 3 Research Design

The primary goal of this research is to investigate and explore the solution to a potential network as well as training problem for remote laboratories using telepresence technology. To demonstrate the proof of concept, a physical as well as virtual (digital twin) laboratory consisting of a robotic arm is created for an active experimentation. A set of tasks were performed on virtual (digital twin) and physical robotic arm with a camera/live video feed provided for observations.

### Design of the Ubiquitous Platform

With this research activity, an intended outcome is to convey the understanding of network lags, slow performance, and their effects as well as what challenges can be typically encountered while performing the experiment in remote lab setting. To achieve these goals, this research:

- Created a physical lab using a 6 DoF robotic arm equipped with a magnetic gripper and 6 servos controlled via PWM (Pulse Width Modulator) interface on the Raspberry Pi. Since Pi only has 1 PWM interface, SunFounder PCA9685 16-Channel 12-Bit PWM Servo Driver was installed.
- Raspberry Pi Zero W to drive the system.
- Installed a digital camera for remote viewing and controlling.
- Linked the system to Web (Internet) for remote operation.

### *Physical Robotic Arm*

A Mechanical Arm, with 6 MG996 Analog Steering Gear, 6DOF Robot Mechanical Arm Clamp Claw Kit DOF Manipulator replaced with magnet. Working voltage: 4.8V-7.2V, machine torsion of 10kg.cm, limit angle: 180°, metal gears with dead time 5uS. Servos controlled via the pwm (pulse width modulator) interface on the Raspberry Pi. However, since the Pi only has 1

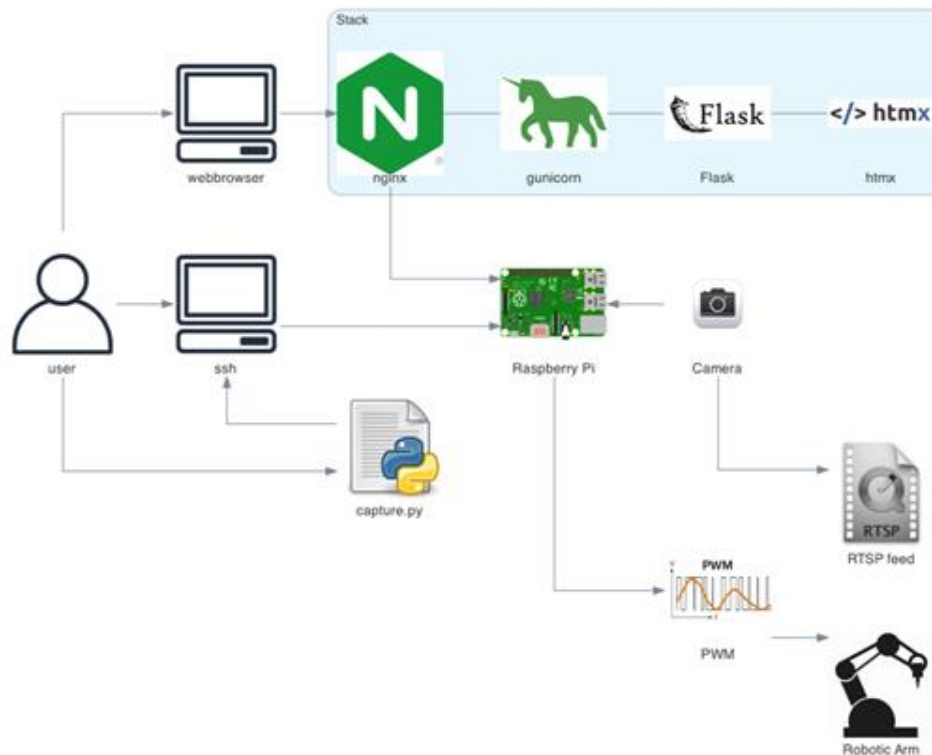
pwm interface, it was connected to a pwm driver. In our case, we used SunFounder’s PCA9685 16-Channel 12-Bit PWM Servo Driver.

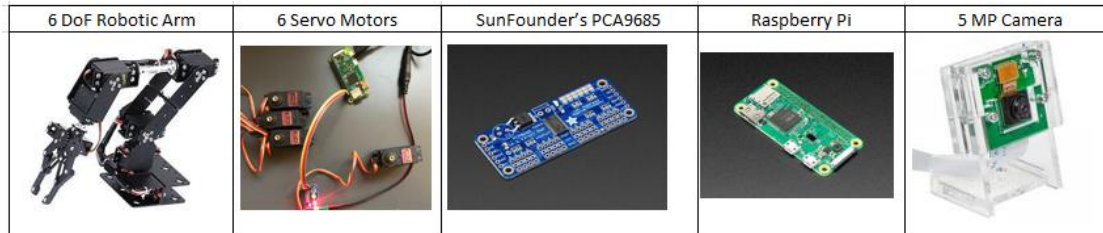
**System**

A complex system was created consisting of a consumer-grade Robotic Arm, a Raspberry Pi which controls the arm via a PWM module and hosts a webpage that allows a user to interact with the system. The Raspberry Pi is used to host a Flask application that exposes the Robotic Arm’s controls, a Digital Twin implemented via WebGL and serves as the infrastructure to conduct ‘training’ and ‘evaluation’ sessions of the system.

**Figure 3**

*Sample of System Developed for This Research*



**Figure 4***Electronic Parts Assembled for Physical Laboratory****Raspberry Pi***

Raspberry Pi OS was used in this research to drive the system. It is a free operating system based on Debian, optimised for the Raspberry Pi hardware, and is the recommended operating system for normal use on a Raspberry Pi. The most current documentation can be found at the official Raspberry Pi homepage at [www.raspberrypi.com/documentation](http://www.raspberrypi.com/documentation).

Please see Appendix section [Install Raspberry Pi](#) for Full Code base.

***Webserver***

Nginx is an open-source HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. It was used in this research to handle all the http communication and serving the static content as well as a reverse proxy server that passes the dynamic content requests to the backend server (i.e. Gunicorn). More about Nginx can be found here <https://nginx.org/en/>.

Please see Appendix section [Install Nginx](#) for Full Code base.

***Gunicorn***

Since this research required to run multiple instances of the web application at the same time (many users could be logging into the system at the same time), Gunicorns was installed for strict process management, resources and permissions control as well as making sure they are healthy and restart as needed, distributing incoming requests across those instances and



communicate with the web server. Gunicorn ‘Green Unicorn’ is a Python WSGI HTTP Server for UNIX. It’s a pre-fork worker model. The Gunicorn server is broadly compatible with various web frameworks, simple implementation, fast and lightweight on server resources. More about the Guicorn can be found here <https://gunicorn.org/>.

Please see Appendix section [Install Guicorn](#) for Full Code base.

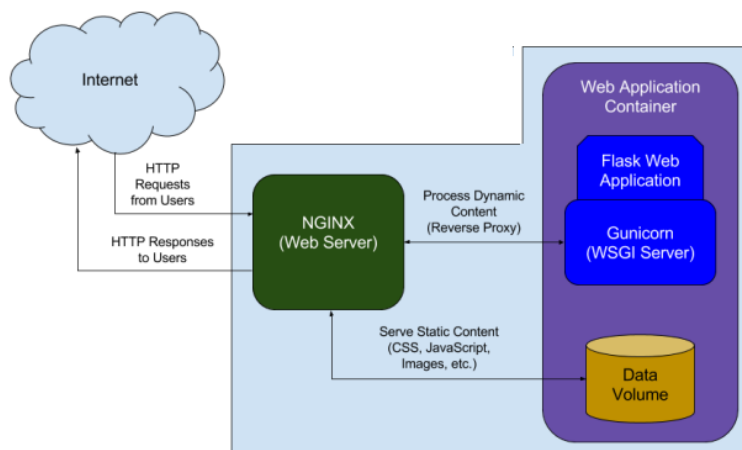
### ***Flask***

To process HTTP requests that are received by NGINX and passed along to Gunicorn a web application framework Flask was used, it’s a Python module that allows to develop web applications with small and easy-to-extend core: it’s a microframework that doesn’t include an ORM (Object Relational Manager) or such features. Figure 5 illustrates Webserver, Guicorn and Flask configuration (Patrick, n.d.) More can be found at <https://pythonbasics.org/what-is-flask-python/>.

Please see Appendix section [Install Flask](#) for Full Code base.

### **Figure 5**

#### *Webserver, Guicorn and Flask*



## ***Camera***

Installed a 5 Megapixels 1080p Sensor OV5647 Camera Module for Raspberry Pi with Raspbian 'Buster'. Documentation for Raspberry Pi camera can be found here:

<https://www.raspberrypi.com/documentation/accessories/camera.html>

## ***Networking***

For this experiment VLAN (Virtual Local Area Network) was created. It is used to separate local networks on a single physical network which is done by separating the communication of each network by adding a VLAN tag into a frame header.

An interface tagging and receiving (i.e. accepting) tagged communication is referred here as a VLAN tagging interface. Each of these interfaces has to be configured to a specific VLAN ID and ports.

VLAN tagging is a method through which more than one VLAN is handled on a port. VLAN tagging is used to tell which packet belongs to which VLAN on the other side. To make recognition easier, a packet is tagged with a VLAN tag in the Ethernet frame.

The purpose of a 'tagged' port is to pass traffic for multiple VLAN's, whereas an 'untagged' port accepts traffic for only a single VLAN.

Please see Appendix section [Networking](#) for more details and Full Code base.

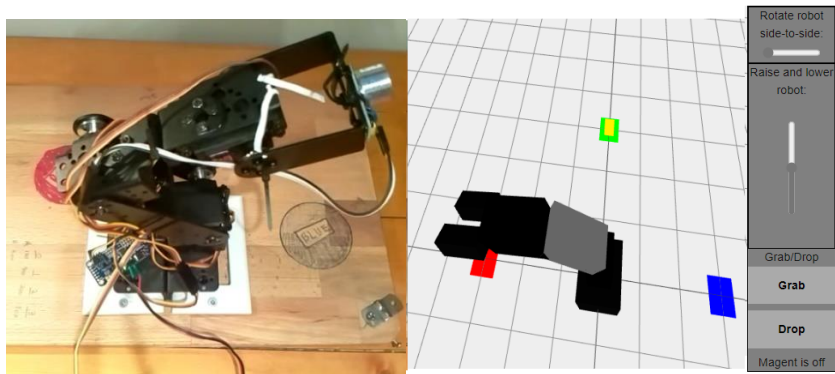
## ***Digital Twin***

Digital Twin was implemented using WebGL, a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins developed

by the Khronos Group WebGL Overview - The Khronos Group Inc as opposed to some other systems such as Matlab, LabVIEW requires user not only requires software license but also install support packages, Add-ons etc. (citation required).

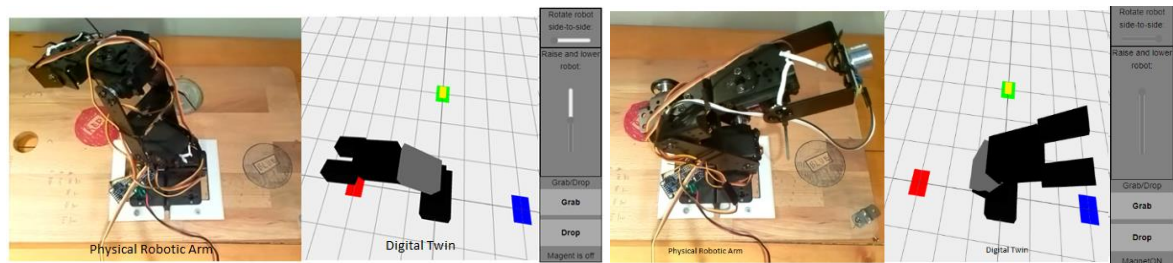
**Figure 6**

*Digital and Physical Robotic Arm Screen*



**Figure 7**

*Controlling Physical Arm Using its Digital Twin*



## Chapter 4 Research Implementation and Results

This research contributes to the area of telepresence robots in remote laboratories. Specifically, it explores techniques such as centralised vs dispersed setup to test performance and reliability, main objective of this research is to test the hypothesis that practices can be developed for users to gain confidence in using telepresence robot remotely, offline testing (digital twin) could work as a tool to understand the telepresence equipment, and that the uninterrupted network connectivity is the backbone for successful telepresence operations. A scenario-based approach is adopted to evaluate the pain points as well as user approachability in using the lab setup for *before and after* method to understand the prerequisite for training.

### Testing Strategy

A system is created to allow remote users access to a remote laboratory by remotely interacting with a physical robotic arm, after testing for a centralized setup by allocating all components at Location 1, locations were swapped to adopt various setting to carry on with testing, following example shows how locations for dispersed setup were determined for the implementation to test network latency as well as the Quality of Service (QoS):-

- Location 1 represents Remote lab user who will typically interact with the remote lab.
- Location 2 represents DT Server and Robotic Arm (a likely location where both systems will coexist under same IP address / same network).
- Location 3 represents DT server (a convenient location where DT will be set up physically away from Robotic Arm under different IP address / different network).

Entities in this experiment i.e., user, digital twin server and robotic arm will be strategically positioned conforming to the four possible scenarios presented below with the following instruction relayed by the user to interact with the system: -

- Move the robotic arm.
- Pick up object.
- Drop object at a specific location.
- Return to the original location.

### ***Scenario 1***

“Status-quo”. All component of the lab are configured at one geographical location (Centralized Location 1).

**Figure 8**

*Centralized System Setup*

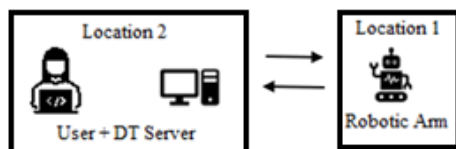


### ***Scenario 2***

Later, location was decentralized and configured with various components to be utilized for more experimentation. For scenario 2, User and Digital twin at Location 2 connecting to the robotic arm at Location 1 and issue the commands.

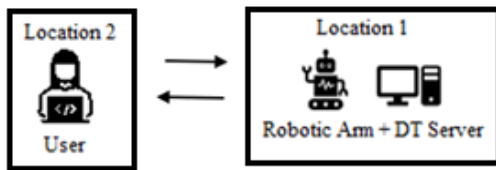
**Figure 9**

*Dispersed Systems Setup A*

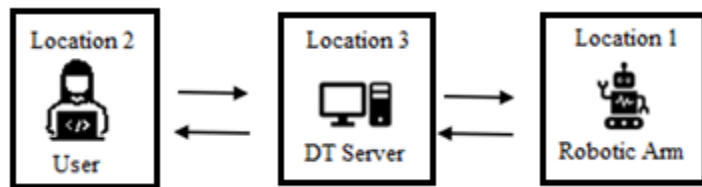


**Scenario 3**

User from Location 2 connects to the digital twin server and Robotic Arm at Location 1 and issue the commands. This scenario is close to be very practical, traditionally resources such as the lab and the servers are housed at the same establishment.

**Figure 10***Dispersed System Setup B***Scenario 4**

User from location 2 issue command for DT Server at location 3 intended for robotic arm at Location 1. This scenario is best suited for the evolving technologies, digital twin server could be located on a cloud as most institutions are moving their information technology resources to cloud while keeping the physical lab on premise.

**Figure 11***Dispersed System Setup C***Network Hops**

Hop is the logical distance between networks based on the number of routers that must be traversed by packets sent between them. A lower hop count also doesn't necessarily mean the

connection between two devices will be faster. A higher hop count via one path might perform better than a lower hop count via a different path due to the faster routers along the longer path (Fisher, 2022). Using Microsoft Windows platform TRACERT command at a Command Prompt is used here to find hops which is a computer networking term that refers to the number of routers that a packet (a portion of data) passes through from its source to its destination (Fisher, 2022). Although TRACERT will not help us in finding the latency, but it will help us to understand why it is important to have a dedicated internet access.

### **Test Results of Each Scenario**

Scenario 1 to 3 supposedly have a smaller number of hops since most of the component for experiment were located at the same location, but the count of hops was the highest for scenario 4 since all parts of this experiment are located at different location.




**Table 2**

*Network Hops Statistics for 4 Scenarios*

	Scenario			
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
User to DT Server	2	2	2	10
DT Server to Arm	5	3	3	9
Arm to User	3	6	4	11

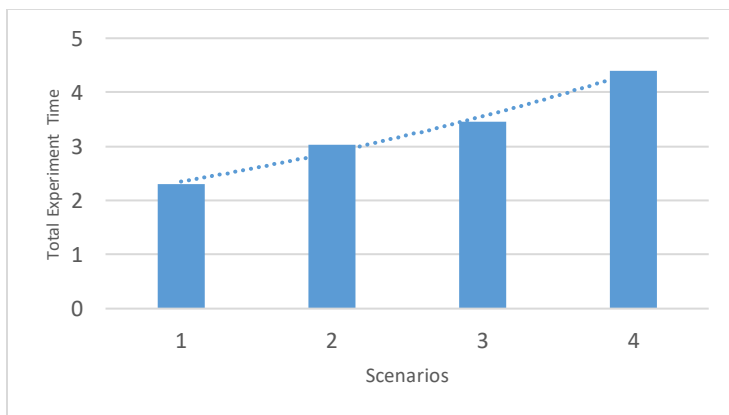
**Table 3**

*Total Experiment Time*

Scenario	User 	Digital Twin Server 	Robotic Arm Server 	Total Experiment Time
1	Location 1	Location 1	Location 1	2 min 30 sec
2	Location 2	Location 2	Location 1	3 min 03 sec
3	Location 2	Location 1	Location 1	3 min 45 sec
4	Location 2	Location 3	Location 1	4 min 39 sec

**Table 4**

*Total Experiment Time Graph*



Since all components of the experiment were present at the same location total experiment time for testing scenario 1 was optimal but as they started to be dispersed, total time for experiment started to increase and it was at its peak when all component of the experiment were in a complete dispersed setting.

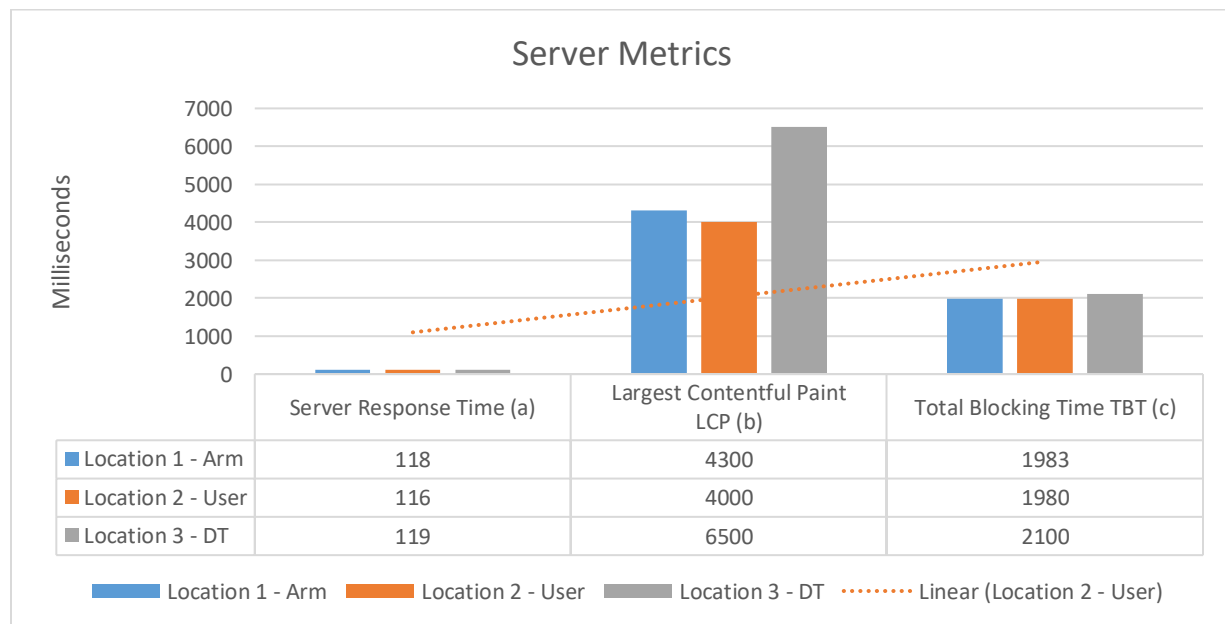


## Server Speed Check

Server speed was tested using the [www.Bitcatcha.com](http://www.Bitcatcha.com) Server Speed Checker. The test conducted for following three metrics by measuring the response time for the server from eight locations around the world, testing each location three times and using the median value as the result they display.

**Table 5**

*Server Metrics (Dispersed)*



*Note.* (a) **Server Response Time** Server speed, it shows how fast a website actually does anything is dependent on how fast the web hosting service is. It is measured by time it takes (in milliseconds) for the server to respond to requests.

(b) **Largest Contentful Paint (LCP)** To measure LCP, it is checked to see how long the largest visible element on a site takes to load (in seconds converted to milliseconds).

(c) **Total Blocking Time TBT** works by measuring the estimated total amount of time a browser is blocked when a site is being loaded.

## Training

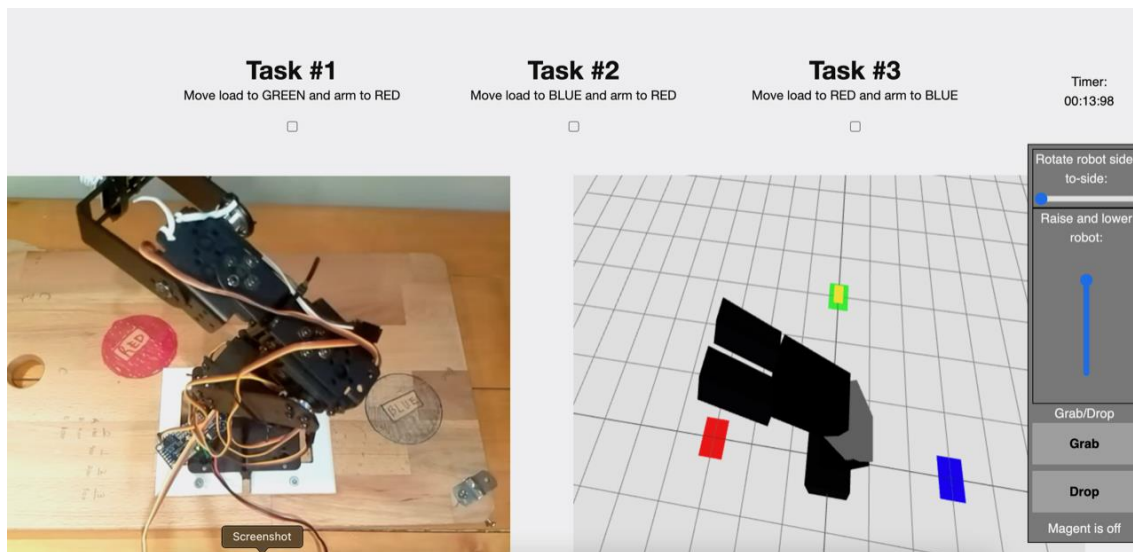
In order to address user acceptance, option for training was explored in this research that is effective for multiple learning styles (i.e., visual, auditory and kinesthetic). In order to find ways to effectively communicate training and better engage learners regardless of location, an

experiment was conducted with having more than one task for the user to perform. Users were asked to perform the sequence as follows:-

- 1- Complete the task on the actual arm (Evaluating on the actual arm before).
- 2- Complete the tasks on the digital arm.
- 3- Lastly, complete the task again on the actual arm (Evaluating on the actual arm after).

**Figure 12**

*Training*



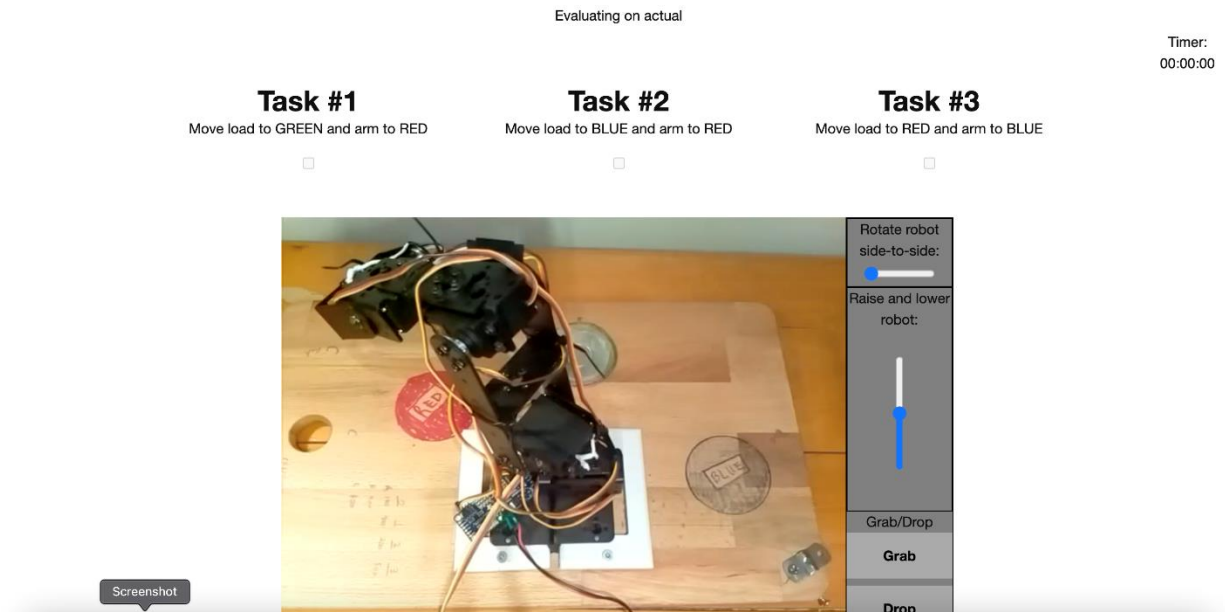
Time for all movements / interactions with the system were recorded by task, a check box was auto checked once the task is complete with the time being recorded.

```
while startbutton is True
starttime = time.clock()
while stopbutton is True
stoptime = time.clock()
elapsedtime= stoptime - starttime
```

1- User completing the task on the actual arm (Evaluating on the actual arm before).

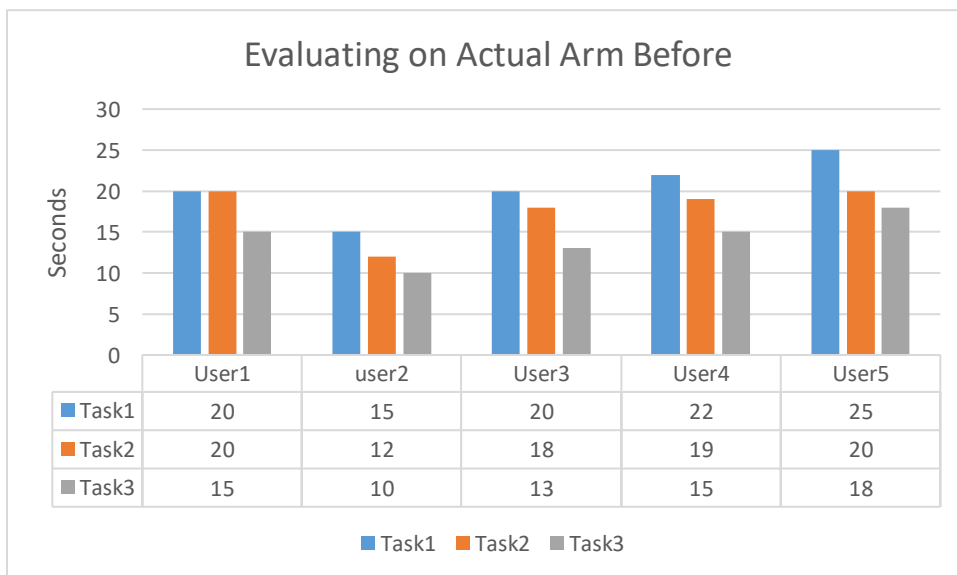
**Figure 13**

*Evaluating on Actual (Before)*



**Table 6**

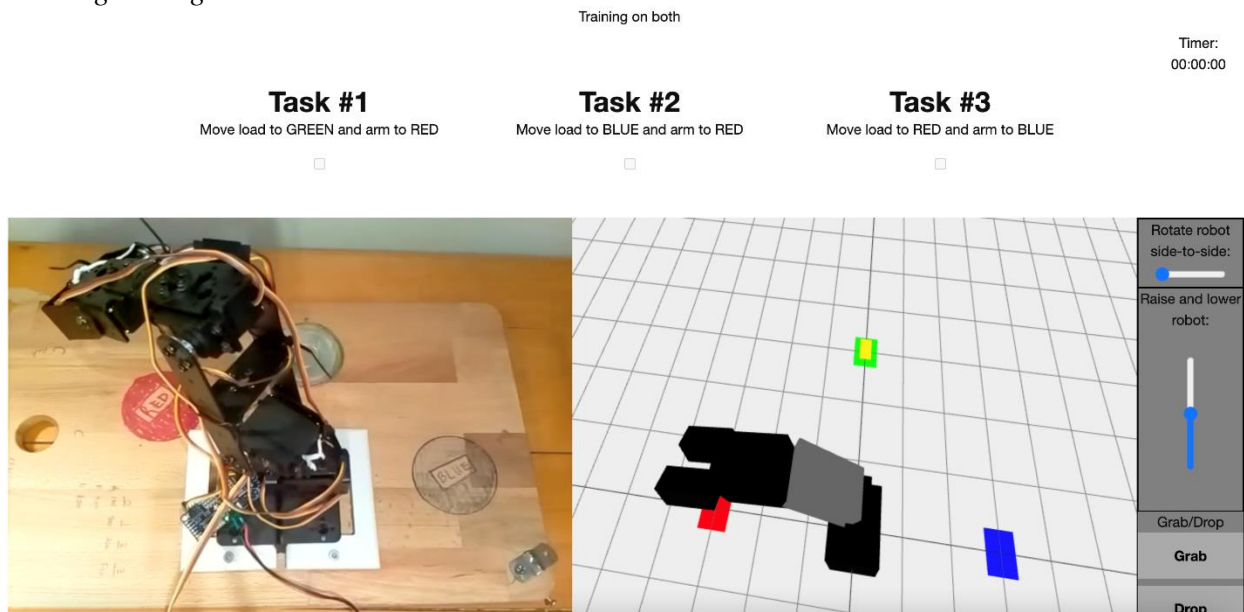
*Training on Physical Arm (Before)*



2- User completing the tasks on the digital arm.

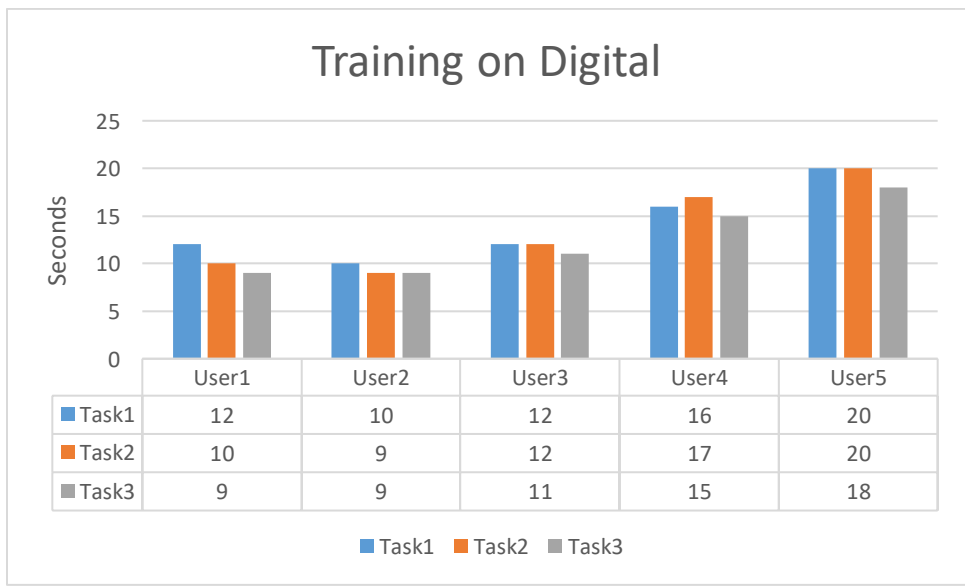
**Figure 14**

*Training on Digital Twin*



**Table 7**

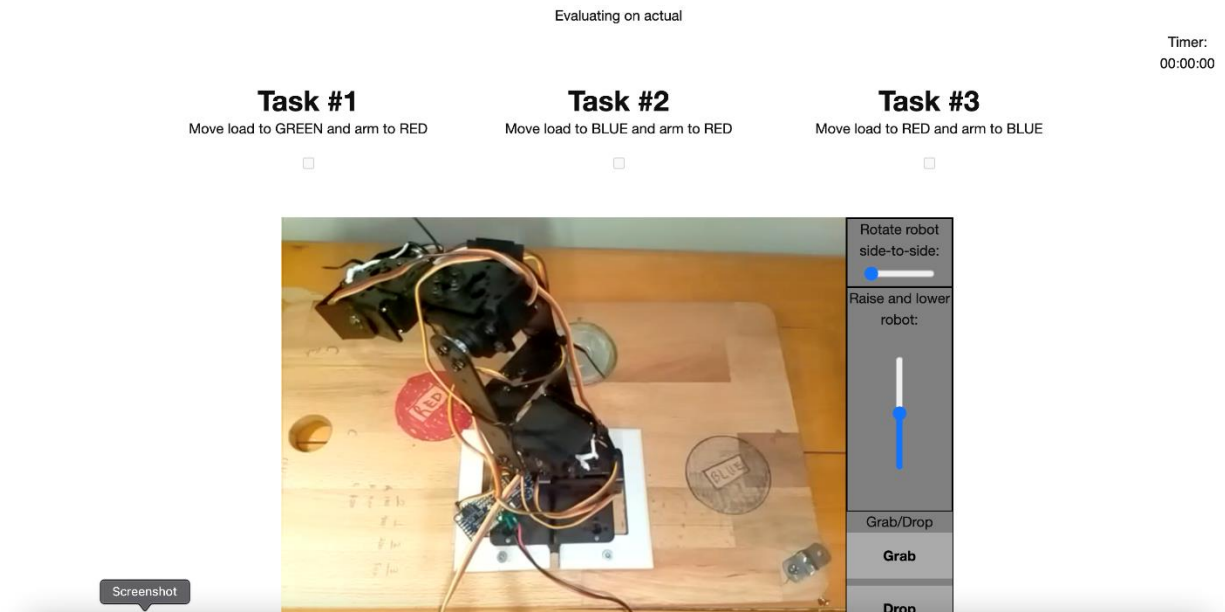
*Training on Digital Twin*



3- User completing the task again on the actual arm (Evaluating on the actual arm After)

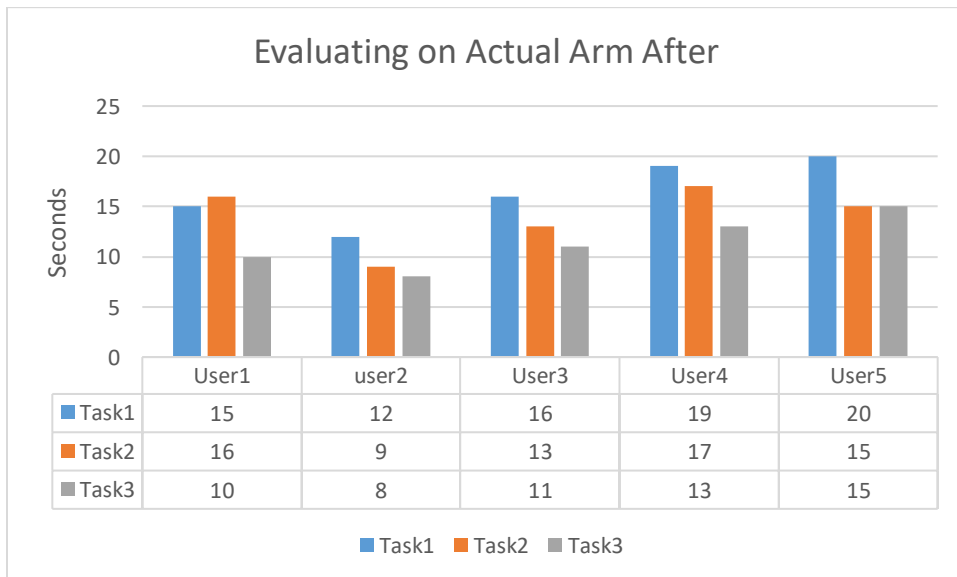
**Figure 15**

*Evaluating on Actual (After)*



**Table 8**

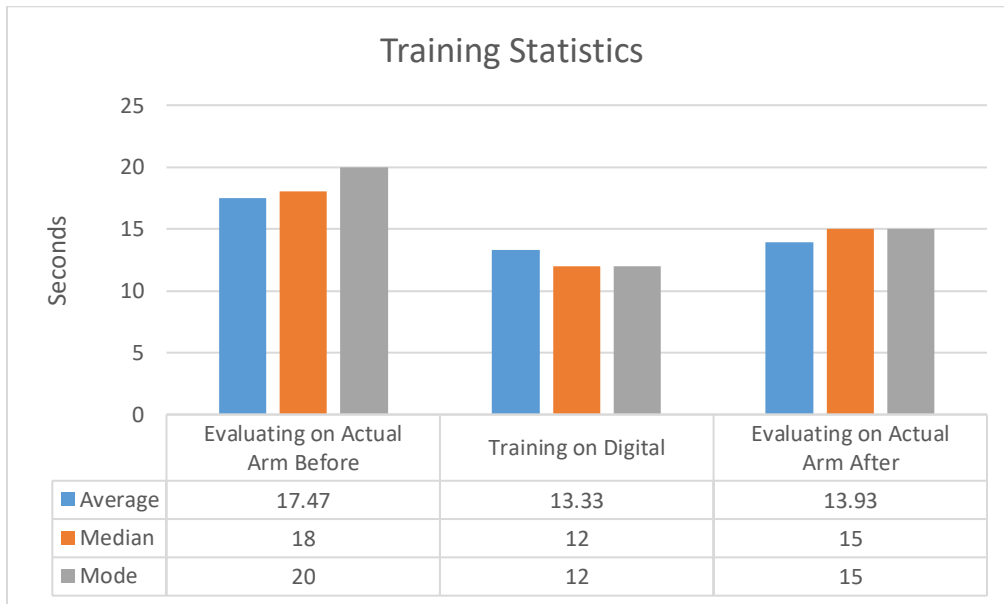
*Training on Physical Arm (After)*



It was noted that time to complete the task for each user started to decrease as they progressed toward the next, each progression made user more familiar with controls and response to their input as well as instilling confidence in their performance.

**Table 9**

*Training Statistics*



Data shows that user manipulating with the Actual arm took less time after getting familiarized by trying their hands on a digital arm. User spent average time of 17.47 seconds on the actual arm before, 13.33 seconds on digital arm whereas 13.93 on the actual arm after training on digital arm hence demonstrating the benefits of the training.

## Chapter 5 Opportunities and Challenges

### Opportunities and Challenges

While the benefits of telepresence technology are evident in the latest research, there are several challenges that need to be addressed if the technology will be used outside its conventional usage i.e. social interaction. Remote lab experiments necessitate a precise use of laboratory equipment which requires a certain (spatial) knowledge and the ability to be aware of the objects in space and a relative telepresent robot's position. In addition to this:

- Robotic arm turned out to be more difficult to control than expected. It was controlled by six servo motors that are not all that precise and tended to 'jump' randomly especially at start up. Original idea of controlling all six servo motors independently with sliders to control the 4 DoF proved to be an inadequate control system. This, in addition to the 'choppiness' of the servos themselves resulted in a user experience that was disappointing and confusing.
- The decision was then taken to come up with three defined 'targets' ('red', 'green', 'blue') which, once implemented, would allow the user to click a button and have the Robotic Arm go to that location. The tension between concept and ultimate execution was the decision to restrict the robot to three 'targets'. This was a necessary direction to limit the complexity of the overall application given time limitations but seems to have fallen short of the expectation that the users would have more control over the robot's movements. Implementing a fuller control system for the robotic arm and digital twin will require more time and likely better knowledge of inverse kinematics.

- Connectivity was the major issue transmission of the data was sluggish due to a slower connection which also dragged down the performance of a livestream video.
- Nginx was installed for Http server and reverse proxy server which were required for this experiment and then deployed a production WSGI server for properly running Flask in production. Although the Nginx worked out of box with the default pre-configured settings however, it could have been optimized for better performance by altering the settings of `/etc/nginx/nginx.conf`.
- Installing a camera for the Raspberry Pi turned out to be more complicated than expected due to the current migration of driver support to a new library called 'libcamera'. A camera bought off Amazon was not supported by the new libcamera library hence the migration to libcamera was made for Raspbian 'Bullseye'. Also detecting failure was a challenging task in dispersed setting, thanks to the livestream camera which provided the slightly delayed but useful input to the commands issued from a remote location.
- This presents obvious concerns with regards to the spatial awareness of the environment for example location, poor recognition of your space, which may result in either standing too close or too far away from the objects also, there could be a difficulty in pinpointing the location of something while seeing through the eyes of a camera. For example, knowing that an apparatus is on a table as opposed to under it. An illustrated map with a live position of the robot showing how far or how close things are from your current location may also be helpful.
- In terms of movement, robots and people can co-exist in the same laboratory, navigating through the environment could be challenging, you need to be informed of how other telepresent robots and objects move through the environment. Either a walkthrough the



space before experiment (which may not be practical in most cases), practice on a virtual lab (digital twin of the lab) could make things easier or a virtual map of the location which must be provided to the user prior to the experiment.

- One of the challenges that was recognized during the experiment was reading markings (Red, Blue, Green), labelling to the products/components must be visible to be utilized for the experiment must be visible to the camera.

### **Research Implementation Challenges**

Implementation of this research was destined to assess network connectivity and how its performance is crucial for successful telepresence operations, due to the limitation of resources only broadband connection was used, affluent technologies such as Direct Internet Access (DIA) would have been the ideal solution for telepresence operation as it provides uninterrupted internet access with no down time or last mile issue.

Main challenge in this experiment was to strategically place the entities such as user, digital twin server and robotic arm to test four possible scenarios to evaluate the pain points, connections were lost numerous times, technical challenges with servo motor calibration were difficult as well as platform stability. Because uniformed instruction was supposed to be relayed by the user for each scenario it was taken as the opportunity to tune the system in homogeneity which turned out to be the step towards success.

From user's perspective, the biggest challenge is to become skilled before carrying out the experiment which is necessary for the wellbeing and safety of apparatus at the lab, an experiment was conducted using before and after analysis to understand the prerequisite for

training although limited number of users participated but small set of data collected allowed to build the narrative for recommendation.

### *Synopsis*

For scenario 1, entire lab setup such as digital twin, robotic arm and corresponding servers were placed at the same location, ideally this scenario should have been divided into local area network (LAN) testing vs external network, assuming LAN performance is obvious to surpass, latter was tested in this research. Total experiment time noted was the shortest possible for this scenario (2:30 minutes) but the network hop count was average as compared to the scenario 1 & 2. On the other hand, the highest time (4:39 minutes) was recorded for scenario 4 with the highest hop count since all components of the lab were placed at a dispersed geographical location. Take into consideration that a lower hop count doesn't necessarily mean a faster connection between two devices, a higher hop count via one path might perform better due to the better hardware (faster routers etc.).

It should be noted that once each scenario testing was completed locations were swapped for research components (devices), they were placed at various geographical locations (e.g. Location 1 for Scenario 1 which held all devices at the beginning but later changed to hold robotic arm only for scenario 2). In order to achieve better results, components of this research were transported in and out of town due to which, monitoring strategies for measuring infrastructure performance and user experience turned out to be a daunting task. As an added measure, server speed check was conducted once all components were placed at different location such as robotic arm server installed at location 1, user engagement from location 2 and digital twin server on location 3.

Before and after method was tested on dispersed setting, managing user inputs into digital twin were relatively easy but connecting user to the physical robotic arm via digital twin was challenging. Also, it was very difficult to live stream videos of the events taking place and physical robotic arm location to user's location.

## Chapter 6 Conclusion and Recommendations

### Conclusion

This research contributes to the area of telepresence robots in remote laboratories, it explored techniques such as centralised vs dispersed setup to test performance and reliability, main objective of this research was to test the hypothesis that practices can be developed, and digital twin can help users to gain confidence in using telepresence robot remotely. With the help of numerous scenarios, it was established that uninterrupted network connectivity is the backbone for successful telepresence operations. An experiment was conducted using *before and after* analysis to understand the prerequisite for training although limited number of users participated but small set of data collected allowed to build the narrative for recommendation.

Through the experimental data collected, it was noticed that broadband internet connectivity introduced critical challenges in performance and connection stability, last mile problem occurred due the shared bandwidth among the local consumers in the area. It is apparent that Telepresence Robot Empowered Smart Lab (TRESL) (Qing Tan, 2019) is long way from overcoming the challenges in distance education, lab and field work unless using affluent technologies such as Direct Internet Access (DIA) etc.

System designed for this research turned out to be very complicated, the idea of allowing external parties without proper network security to interact with actual electrical equipment hosted behind a residential modem exposed the location to a risk of malicious activity (e.g., fire etc.). In addition, the low bandwidth of a residential connection impaired the performance of a system from a latency perspective hence negated the idea of providing the shared access to the physical arm.

Considering the challenges identified during the experiments, remote lab users can greatly benefit from Dedicated Internet access (DIA) which always provides a subscriber with guaranteed bandwidth. Unlike broadband Internet connection which typically comprise of a network shared amongst many other users in the area resulting unpredictable performance and reliability, the recommendation of Dedicated Internet Access DIA is due to the fact that the bandwidth for DIA is not shared with anyone else other than the dedicated users of that physical circuit. Since it is a dedicated connection, the performance is always consistent with a high reliability. Performance metrics like network latency, packet loss, uptime, and repair timeframes are also guaranteed (Dedicated Internet vs Broadband – What’s the Difference?, n.d.). Installing such a technology is a costly undertaking, it would be required to install a DIA circuit to each city from where users will connect to the remote lab but as a solution, a partial DIA circuit can be leveraged which could provide optimal performance. DIA can be installed from a laboratory to a cloud service provider data center which will host the digital twin and or remote access to the actual lab, performance will be supposedly better than broadband once user is connected to the cloud. Also, further research is required on Full vs Split Tunnelling and how its variants such as inverse, dynamic tunnelling etc. could potentially be used as a workaround.

## **Recommendations**

Recommendations in terms of hardware could not be made due to the fact that they vary by lab and experiment and equipment are largely sourced from manufacturers expert in the trade but setting up a 6 DoF robotic arm on a smaller scale turned out to be very challenging, required tuning, trial and error etc. On the other hand, there is a lot can be done for performance gains for digital twin (virtual robotic arm) for a broadband connectivity. Here are some recommendations for creating the Digital Twin:-

### ***Improving Total Blocking Time (TBT)***

Total Blocking Time measures the total time that the browser's main thread is blocked by tasks longer than 50 milliseconds. Input responsiveness or the interactivity between the web page and user during the loading are related to the "Long Tasks" which are the blockers for the main thread of the browser. If TBT is optimized, load-responsiveness will be better and First Input Delay, First CPU Idle also will create a better user experience (GÜBÜR, 2020). Here are a few things that can be done to improve TBT scores:

#### ***Remove unused JavaScript***

JavaScript take resources to run. If some are running but doing nothing, it is increasing browser's blocking time. Identify which ones are not needed and remove for better experience.

#### ***Reduce bloated JavaScript Libraries***

Some plugins and their dependencies use a lot of jQueryes, which can take up quite a fair number of resources. Reducing them could be more resource efficient.

#### ***Defer unused JavaScript***

By deferring uncritical JavaScript to be loaded in later, it will free up browser's resources, which would help it render the site faster, improving the TBT scores.

#### ***Utilize Code splitting***

Breaking your JavaScript code into smaller chunks that can be loaded on certain conditions allows for module fetching on demand. Basically, the module won't be loaded in unless needed. This can be done if client-side framework like React, Angular, and Vue are used.

Also, use highly available network connectivity for your workload public endpoints: These endpoints and the routing to them must be highly available. To achieve this, use highly available DNS, content delivery networks (CDNs), API Gateway, load balancing, or reverse proxies (AWS., 2022). Institutions offering telepresence services such as TRESL should resort to arrange for CDN and the best economical way forward in providing the service via cloud subscription that has datacenters strategically placed all across the region such as Azure or AWS.

If the environment is coded in different programming languages a diverse representation for characters and data structures such as arrays and records should be taken into account so that programs written in different languages are to be able to communicate with one another.

## References

- Abdulwahed, M. (2010). Towards enhancing laboratory education by the development and evaluation of the “TriLab”: a triple access mode (virtual, hands-on and remote) laboratory. Loughborough, U.K. Retrieved January 15, 2022
- Amedeo, C., Gabriella, C., Andrea, O., & Lorenza, T. (2013). Evaluating telepresence robots in the field. In *Agents and Artificial Intelligence*. (pp. 433-448). Springer, Berlin Heidelberg, DEU. Retrieved February 15, 2022
- Andujar, J. M. (2010). Augmented reality for the improvement of remote laboratories: an augmented remote laboratory. *IEEE transactions on education*, 54(3), 492-500. Retrieved January 16, 2022
- AWS. (2021). AWS IoT Device Shadow service. Retrieved December 20, 2021, from <https://docs.aws.amazon.com/iot/latest/developerguide/iot-device-shadows.html>
- AWS. (2022). AWS IoT TwinMaker. Retrieved December 20, 2022, from <https://aws.amazon.com/iot-twinmaker/>
- AWS. (2022). Use highly available network connectivity for your workload public endpoints. Retrieved December 20, 2022, from [https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/rel\\_planning\\_network\\_topology\\_ha\\_conn\\_users.html](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/rel_planning_network_topology_ha_conn_users.html)
- Barabas, C., Bavitz, C., Matias, J. N., Xie, C., & Xu, J. (2015). LEGAL AND ETHICAL ISSUES IN THE USE OF TELEPRESENCE ROBOTS: BEST PRACTICES AND TOOLKIT. We Robot 2015 Fourth Annual Conference on Robotics, Law & Policy. Washington. Retrieved July 06, 2022
- Boschert, S., & Rosen, R. (2016). Digital Twin - The Simulation Aspect. In P. D. Bradely, *Mechatronics Futures Challenges and Solutions for Mechatronic Systems and their Designers* (pp. 59-74). Springer-Verlag. Retrieved February 09, 2022
- Botev, J., & Lera, F. J. (2021). Immersive Robotic Telepresence for Remote Educational Scenarios. Retrieved March 02, 2022, from <https://www.mdpi.com/>
- Brubaker, J. R., Venolia, G., & Tan, J. C. (2012). Focusing on Shared Experiences: Moving beyond the camera in video communication. *DIS 2012* (pp. 96-105). ACM Press. Retrieved October 12, 2022
- Budai, T. a. (2018). Towards a modern, integrated virtual laboratory system. *Acta Polytechnica Hungarica*, 15(3), 191-204. Retrieved March 09, 2022
- Cohen, B., Lanir, J., Stone, R., & Gurevich, P. (2011). Requirements and Design Considerations for a Fully Immersive Robotic Telepresence System. *HRI 2011 Workshop on Social Robotic Telepresence* (pp. 16-22). ACM Press. Retrieved February 02, 2022



- Cyber-Physical. (2020). Retrieved April 10, 2022, from <http://laiboyan.mx/ciber.html>
- Dedicated Internet vs Broadband – What’s the Difference? (n.d.). Retrieved November 12, 2022, from <https://www.bsimplify.com/dedicated-internet-vs-broadband-differences/>
- Desai, M., Tsui, K. M., & Yanco, H. A. (2011). Essential features of telepresence robots. IEEE Conference on Technologies for Practical Robot Applications (pp. 15-20). IEEE. Retrieved April 10, 2022
- Deutsch, J. (n.d.). THE DESIGN GAP Why Design Matters, and Why It Deserves a Greater Focus. Retrieved October 15, 2022, from <https://idoc.pub/documents/the-design-gap-dv1r9g0e1wlz>
- Digital-twin. (n.d.). Retrieved October 15, 2022, from <https://unity.com/solutions/digital-twin-definition>
- Dr. Mert Bal, M. U. (2014). ASEE International Forum, (pp. 20.5.1 - 20.5.13). Indianapolis, Indiana. Retrieved April 13, 2022
- Elisa, N., Stefano, B., Luca, F., & Marco, M. (2020). MES-integrated digital twin frameworks. Journal of Manufacturing Systems, 58-71. Retrieved March 10, 2022
- Experiential Learning. (n.d.). Retrieved November 10, 2022, from <https://www.torontomu.ca/experiential-learning/faculty-staff/kolbs-el-cycle/>
- Feng, S., Quivira, F., & Schirner, G. (2016). Framework for rapid development of embedded human-in-the-loop cyber-physical systems. IEEE 16th International Conference on Bioinformatics and Bioengineering (BIBE), (pp. 208-215). Retrieved November 09, 2021
- Fisher, T. (2022). lifewire.com. Retrieved August 06, 2022, from What are hops: <https://www.lifewire.com/what-are-hops-hop-counts-2625905>
- Franco-Vera, R., Chen, X., & Li, W. (2020). Programmable Remote Laboratory for Mobile Robots. 17th International Conference on Remote Engineering and Virtual Instrumentation (pp. 75-81). Springer, Cham. Retrieved July 09, 2022
- GÜBÜR, K. T. (2020). What is Total Blocking Time and How to Optimize It? Retrieved October 09, 2022, from <https://www.holisticseo.digital/pagespeed/tbt/>
- Gunar, S., Deniz, E., Kaushik, C., & Taskin, P. (2013). The Future of Human in the Loop Cyber Physical System. IEEE Computer Society, 36-45. Retrieved January 10, 2022
- Guzman, J. G., Lopez, F. H., Enriquez, J. A., Mathey, L. A., & Ramirez, A. R. (2017). Remote Laboratories for Teaching and Training in Engineering. In Design, Control and Applications of Mechatronic Systems in Engineering. Intech. Retrieved May 09, 2022
- Haans, A., & IJsselsteijn, W. A. (2012). Embodiment and telepresence: Toward a comprehensive theoretical framework. Interacting with Computers, (pp. 211-218). Retrieved February 02, 2022

- Han, J., & Conti, D. (2020). The Use of UTAUT and Post Acceptance Models to Investigate the Attitude towards a Telepresence Robot in an Educational Setting. MDPI, 1-19. Retrieved May 02, 2022
- Harms, U. (2000). Virtual and remote labs in physics education. Second. Retrieved February 12, 2022
- Hartmann, T., Wirth, W., Schramm, H., Klimmt, C., Vorderer, P., Gysbers, A., . . . Sacau, A. M. (2016). The Spatial Presence Experience Scale (SPES). *Journal of Media Psychology*, Vol 28(1): 1-15. Retrieved May 10, 2022
- Hartsell-Gundy, J., Johnson, E. O., & Kromer, J. (2015). Testing telepresence: remote reference service via robotics. *Reference and User Services Quarterly* 55.2, 118-122. Retrieved August 10, 2022
- IBM. (n.d.). What is a digital twin. Retrieved May 03, 2022, from <https://www.ibm.com/topics/what-is-a-digital-twin>
- Igor M. Verner, S. G. (2018). Development of spatial awareness and operation skills in a remote robot laboratory. 2018 IEEE Global Engineering Education Conference (EDUCON) (pp. 389-303). Santa Cruz de Tenerife, Spain: DOI: <https://doi.org/10.1109/EDUCON.2018.8363256>. Retrieved August 10, 2022
- IMI. (n.d.). digital-twin-technology-benefits-and-challenges. Retrieved November 12, 2022, from <https://identitymanagementinstitute.org/digital-twin-technology-benefits-and-challenges/>
- Immerman, D. (2020). Why IoT is the backbone for digital twin. Retrieved April 03, 2022, from <https://www.ptc.com/en/blogs/corporate/iot-digital-twin>
- Irene Rae, G. V. (2016, 02). A Framework for Understanding and Designing Telepresence. 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (pp. 1552-1566). Vancouver: ACM Digital Library. Retrieved August 15, 2022, from <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/cscwf145-raeA.pdf>
- Ishaani, P. (2018). Cybersecurity risks in Robotics. In *Cyber security and threats: concepts, methodologies, tools, and applications* (pp. 1235-1250). IGI Global. Retrieved November 19, 2022
- Johnson, S., Rae, I., Mutlu, B., & Takayama, L. (2015). Can You See Me Now? How Field of View Affects Collaboration in Robotic Telepresence. 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). (pp. 2397–2406). Association for Computing Machinery, New York, NY. DOI:<https://doi.org/10.1145/2702123.2702526>. Retrieved April 16, 2022

- Kannappan, A. (2020). All You Need to Know About the Life-Changing Digital Twin Technology. Retrieved January 16, 2022, from <https://hackernoon.com/all-you-need-to-know-about-the-life-changing-digital-twin-technology-ix4e3y3e>
- King, A. (n.d.). What are cyber-physical systems? (RMIT University Australia) Retrieved March 08, 2021, from <https://www.rmit.edu.au/news/c4de/what-are-cyber-physical-systems>
- Kousi, N., Gkournelos, C., Aivaliotis, S., Giannoulis, C., Michalos, G., & Makris, S. (2019). Digital twin for adaptation of robot's behaviour in flexible robotic assembly lines. International Conference on Changeable, Agile, Reconfigurable and Virtual Production. 28, pp. 121-126. Patras, Greece: Elsevier B.V. Retrieved August 05, 2022
- Kristoffersson, A., Loutfi, A., & Coradeschi, S. (2013). A Review of Mobile Robotic Telepresence. *Advances in Human-Computer Interaction*, 1-17. Retrieved January 18, 2022
- Kwon, M. (2015). A Tutorial on Network Latency and Its Measurements. In T. Soyata, *Enabling Real-Time Mobile Cloud Computing through Emerging Technologies* (pp. 272-293). IGI Global. Retrieved August 19, 2022
- Labonté, D. B. (2010). Comparative analysis of 3D robot teleoperation interfaces with novice users. *IEEE Transactions on Systems, Man, and Cybernetics Part B*, vol. 99, 1-12. Retrieved March 16, 2022
- Lombard, M., & Jones, M. T. (2015). Part I - Telepresence, concepts and theories. In M. Lombard, F. Biocca, J. Freeman, W. IJsselsteijn, & R. J. Schaevitz, *Immersed in media - Telepresence theory measurement & technology* (pp. 25-47). Springer. Retrieved November 11, 2022
- M. Desai, K. M. (2011). Essential features of telepresence robots. *IEEE Conference on Technologies for Practical Robot Applications* (pp. 15-20). Woburn, MA: IEEE. Retrieved July 06, 2022
- Macharet, D. G., & Florencio, D. A. (2012). A Collaborative Control System for Telepresence Robots. *INTELLIGENT ROBOTS AND SYSTEMS - IROS 12, OCT., 2012* (pp. 5105-5111). IEEE Press. Retrieved October 12, 2022
- Maiero, J. (2021). Novel haptic interfaces and their impact on perception and performance. Retrieved November 12, 2022, from <https://bura.brunel.ac.uk/handle/2438/22615>
- Malecaj, L. (2020). BRIDGING THE SKILLS GAP IN FIELD SERVICE WITH AR REMOTE ASSISTANCE. Retrieved July 06, 2022, from <https://www.vsigth.io/augmented-reality/bridging-the-skills-gap-with-ar-remote-assistance/>
- Malecaj, L. (2021). REMOTE MAINTENANCE WITH AR TECHNOLOGY. Retrieved October 07, 2022, from <https://www.vsigth.io/remote-maintenance-with-ar-technology/>

- Microsoft. (2022). digital-twins/concepts-models. Retrieved October 01, 2022, from <https://learn.microsoft.com/en-ca/azure/digital-twins/concepts-models>
- Moura, D. T. (2017). Telepresence: The extraordinary power of virtual reality. Retrieved November 10, 2022, from <https://liveinnovation.org/telepresence-extraordinary-power-virtual-reality/>
- Palmer, C., Roullier, B., Aamir, M., Stella, L., Diala, U., Anjum, A., . . . Calvert, A. (2021, June). Virtual Reality based Digital Twin System for remote laboratories and online practical learning. Retrieved May 10, 2022, from <https://arxiv.org/ftp/arxiv/papers/2106/2106.09344.pdf>
- Patrick. (n.d.). how-to-configure-nginx-for-a-flask-web-application. Retrieved December 10, 2021, from <https://www.patricksoftwareblog.com/how-to-configure-nginx-for-a-flask-web-application/>
- Pennisi, E. (2020). During pandemic students do field and lab work without leaving home. Retrieved March 10, 2022, from <https://www.sciencemag.org/news/2020/07/during-pandemic-students-do-field-and-lab-work-without-leaving-home>
- Pfeiffer, J., Lehner, D., Wortmann, A., & Wimmer, M. (2022). Modeling Capabilities of Digital Twin Platforms Old Wine in New Bottles? Journal of Object Technology. Retrieved November 20, 2022
- Qing Tan, M. D.-M. (2019). Toward a telepresence robot empowered smart lab. Springer Link. Retrieved December 15, 2021, from <https://link.springer.com/article/10.1186/s40561-019-0084-3>
- Rae, I., Mutlu, B., & Takayama, L. (2014). Bodies inMotion: Mobility, Presence and Task Awareness in Telepresence. Human Factors in Computing Systems (pp. 2152-2162). Toronto: ACM Press. Retrieved August 03, 2022
- Robotics, M. C. (2020). Unraveling Degrees of Freedom and Robot Axis. Retrieved November 06, 2022, from <https://motioncontrolsrobotics.com/>
- Roo, J., & M.Hachet. (2017). One reality: Augmenting how the physical world is experienced by combining multiple mixed reality modalities. Annual ACM Symposium on User Interface Software and Technology (pp. 787-795). ACM. Retrieved March 17, 2022
- S., B., & R, R. (2016). Digital Twin—The Simulation Aspect. In H. P., & B. D., Mechatronic Futures (pp. 59-74). Springer, Cham. Retrieved January 20, 2022
- Sanislav, T., & Miclea, L. (2012). Cyber-Physical Systems - Concept, Challenges and Research Areas↔. CONTROL ENGINEERING AND APPLIED INFORMATICS, Vol.14, No.2, 28-33. Retrieved November 10, 2021

Semenets, V. S. (2019). Trends in Training Modern Technicians. International Scientific and Practical Conference (pp. 35-36). Kharkiv, Ukraine: Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs. Retrieved November 10, 2021

Sherman, W. R., & Craig, A. B. (2019). Understanding Virtual Reality. <https://doi.org/10.1016/B978-0-12-800965-9.03001-4>. Retrieved August 05, 2022

The 'Last Mile' Connectivity Problem and How to Solve It. (2019). Retrieved August 15, 2022, from <https://www.getgds.com/resources/blog/connectivity/the-last-mile-connectivity-problem-and-how-to-solve-it>

Venkatesh, V., Thong, J. Y., & Xu, X. (2012). Consumer Acceptance And Use of Information Technology: Extending the UTAUT. MIS Quarterly Vol 36, 157-178. Retrieved January 05, 2022

Vershinin, Y. (2014). Internet-based Teleoperation Laboratory. Coventry University Teaching and Learning Conference 2014. Coventry, UK. Retrieved May 10, 2022

WHAT IS DIA AND HOW CAN YOUR BUSINESS BENEFIT FROM IT? (n.d.). Retrieved September 10, 2022, from <https://www.wiline.com/briefs/dedicated-internet-access>

What is Digital Twin Software. (2022). Retrieved December 16, 2022, from <https://www.g2.com/categories/digital-twin#learn-more>

Xuemin Chen, G. S. (2010). Virtual and Remote Laboratory Development: A Review. Earth and Space 2010: Engineering, Science, Construction, and Operations in Challenging Environments © 2010 ASCE. Honolulu, Hawaii. Retrieved March 10, 2022

## Appendix A - Program Code: Description and Attachments

### *Install Raspberry Pi*

Mount your SSD and enable a serial connection to your device

```
nano /boot/config.txt
```

```
enable_uart=1
```

You need to create a file called 'ssh' on the /boot volume

```
touch ssh
```

Configure your wireless settings

```
nano /boot/wpa_supplicant.conf
```

```
country=CA # Your 2-digit country code
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
    ssid="YOUR_NETWORK_NAME"
    psk="YOUR_PASSWORD"
    key_mgmt=WPA-PSK
}
```

### *Install Nginx*

Install Nginx using apt-get

```
sudo apt-get update
sudo apt-get install nginx
```

```
# The Nginx server should start automatically.
# If for some reason it does not, start the server with:
```

```
sudo service nginx start
```

By default, Nginx creates a web server root directory at `/var/www/html/` with a default html web page in the directory. When nginx is up and running, the webpage will be visible by pointing the browser url to your Linux machine.

### ***Optimize Nginx***

We need to setup the Nginx as http server and reverse proxy server, and then deploy a production WSGI server for properly running Flask in production.

Although the Nginx works out of box with the default pre-configured settings, however, it can be optimized for better performance by altering the settings of `/etc/nginx/nginx.conf`:

#### **Nginx settings**

```
sudo nano /etc/nginx/nginx.conf
```

Make the following changes for improving potential performance:

- 1) Uncommented the `multi_accept` directive and set to `on`, which informs each `worker_process` to accept all new connections at a time, opposed to accepting one new connection at a time.
- 2) Change the `keepalive_timeout` from default 65 to 30. The `keepalive_timeout` determines how many seconds a connection to the client should be kept open before it's closed by Nginx. This directive should be lowered so that idle connections can be closed earlier at 30 seconds instead of 65 seconds.
- 3) Uncomment the `server_tokens` directive and ensure it is set to `off`. This will disable emitting the Nginx version number in error messages and response headers.
- 4) Uncomment the `gzip_vary on`, this tell proxies to cache both the gzipped and regular version of a resource where a non-gzip capable client would not display gibberish due to the gzipped files.
- 5) Uncomment the `gzip_proxied` directive and set it to `any`, which will ensure all proxied request responses are gzipped.

6) Uncomment the `gzip_comp_level` and change the value to 5. Level provide approximate 75% reduction in any ASCII type of files to achieve almost same result as level 9 but not have significant impact on CPU usage as level 9.

7) Uncomment `gzip_http_version 1.1;`, this will enable compression both for HTTP/1.0 and HTTP/1.1.

8) Add a line `gzip_min_length 256;` right before `gzip_types` directive, this will ensure that the file smaller than 256 bytes would not be gzipped, the default value was set at 20 bytes which is too small and could cause the gzipped file even bigger due to the overhead.

9) Replace the `gzip_types` directive with the following more comprehensive list of MIME types. This will ensure that JavaScript, CSS and even SVG file types are gzipped in addition to the HTML file type.

```
gzip_types
```

9) Replace the `gzip_types` directive with the following more comprehensive list of MIME types. This will ensure that JavaScript, CSS and even SVG file types are gzipped in addition to the HTML file type.

```
gzip_types
```

```
application/atom+xml
application/javascript
application/json
application/rss+xml
application/vnd.ms-fontobject
application/x-font-ttf
application/x-web-app-manifest+json
application/xhtml+xml
application/xml
font/opentype
image/svg+xml
image/x-icon
text/css
text/plain
text/x-component
text/javascript
text/xml
```



So far, we have configured the *event and http directive blocks* through `/etc/nginx/nginx.conf` file, the *server directive* is usually stored in a separate configuration file and gets loaded into the overall configuration structure through an

```
include /etc/nginx/sites-enabled/*;
```

directive at the end of `/etc/nginx/nginx.conf` file. The actual server configuration file for each server is usually stored in `/etc/nginx/sites-available` directory and linked by a symbolic link in `/etc/nginx/sites-enabled` directory, this allows you to keep multiple server configuration files in `sites-available` directory and only activate a particular server by adding a symbolic link at `sites-enabled` directory.

The principle is that most files will first be passed to the Flask app for pre-processing prior to being served by the Nginx http server. This will include all files ending in `.py` and most of the directories in a basic Flask application.

Any files that do not need to be passed to the Flask app first will be handled directly by the Nginx http server. Importantly, this will include most files located in the 'static' directory as well as files such as `sitemap.xml`, `robots.txt` and `favicon.ico` or `site-logo.png`. Those files are typically used by search engines and browser and are usually located at document root directory of a web site.

First, create a file at `/etc/nginx/sites-available/`, give it a name reflect your server application purpose such as **robotapp**:

server directives

```
cd /etc/nginx/sites-available/  
sudo nano robotapp  
  
# Enter the following nginx directives and save the file:  
  
server {  
    listen 80 default_server;  
    listen [::]:80;  
  
    root /var/www/html;  
  
    server_name example.com;  
  
    location /static {  
        alias /var/www/html/static;  
    }  
  
    location / {  
        try_files $uri @wsgi;  
    }  
  
    location @wsgi {  
        proxy_pass http://unix:/tmp/gunicorn.sock;  
        include proxy_params;  
    }  
  
    location ~* \.(ogg|ogv|svg|svgz|eot|otf|woff|mp4|ttf|css|rss|atom|js|jpg|jpeg|gif|  
png|ico|zip|tgz|gz|rar|bz2|doc|xls|exe|ppt|tar|mid|midi|wav|bmp|rtf)$ {  
        access_log off;  
        log_not_found off;  
        expires max;  
    }  
}
```

With this set of Nginx configurations, we told Nginx to listen on port 80 for HTTP requests directed at the domain example.com. For every such request, we tell Nginx to reference the directory /var/www/html as the document root.

The location `/static` map the request to the file inside the `/var/www/html/static` directory and serve the static content directly without passing it to the Flask app.

With this set of Nginx configurations, we told Nginx to listen on port 80 for HTTP requests directed at the domain `example.com`. For every such request, we tell Nginx to reference the directory `/var/www/html` as the document root.

The location `/static` map the request to the file inside the `/var/www/html/static` directory and serve the static content directly without passing it to the Flask app.

The location `/` block map request to a file inside the document root directory and return it as a HTTP response if there is one. This ensures the content such as `robots.txt` get served by the Nginx server. If the request is not a file that can be found at this root directory, Nginx should issue an internal redirect to the upstream server defined in `@wsgi` block.

The location `@wsgi` block proxies HTTP requests to the `/tmp/gunicorn.sock` unix socket where Flask app will listen to, it also setup some common HTTP headers defined inside the `/etc/nginx/proxy_params` file.

The last location block tells the Nginx to turn off the access log and set the cache expiry date to maximum for those specific static content types.

By default, Nginx enable the server configuration located in `/etc/nginx/sites-available/default` by creating a symbolic link at `/etc/nginx/sites-enabled/`, we need to disable the symbolic link linked to the default configuration and create a link to enable our new server configuration.

## Final steps

```

cd /etc/nginx/sites-enabled/
sudo rm default
sudo ln -s /etc/nginx/sites-available/robotapp .

# Test the Nginx configuration with command:

sudo nginx -t

# If everything looks ok, go ahead and reload the Nginx configuration:

sudo service nginx reload

```

We have completed the setup of Nginx server, let's move on to setup the upstream

### ***Install Guicorn***

Below are configurations files and instructions for using systemd to create a unix socket for incoming Gunicorn requests. Systemd will listen on this socket and start gunicorn automatically in response to traffic. We will then provide instructions for configuring Nginx to forward web traffic to the newly created unix socket:

/etc/systemd/system/gunicorn.service:

```

[Unit]
Description=gunicorn daemon
Requires=gunicorn.socket
After=network.target

[Service]
Type=notify
# the specific user that our service will run as
User=someuser
Group=someuser
# another option for an even more restricted service is
# DynamicUser=yes
# see http://0pointer.net/blog/dynamic-users-with-systemd.html
RuntimeDirectory=gunicorn
WorkingDirectory=/home/someuser/applicationroot
ExecStart=/usr/bin/gunicorn applicationname.wsgi
ExecReload=/bin/kill -s HUP $MAINPID

```

```

KillMode=mixed
TimeoutStopSec=5
PrivateTmp=true

[Install]
WantedBy=multi-user.target
/etc/systemd/system/gunicorn.socket:

```

```

[Unit]
Description=gunicorn socket

[Socket]
ListenStream=/run/gunicorn.sock
# Our service won't need permissions for the socket, since it
# inherits the file descriptor by socket activation
# only the nginx daemon will need access to the socket
SocketUser=www-data
# Optionally restrict the socket permissions even more.
# SocketMode=600

[Install]
WantedBy=sockets.target

```

Next enable and start the socket (it will autostart at boot too):

**systemctl**

### ***Install Flask***

```

sudo apt-get install python3-pip
pip3 install pipenv
nano ~/.bashrc
# Add this to the bottom of file
export PATH=$PATH:/home/pi/.local/bin
# Then you save and exit, and you need to "source" the file since it only
# runs automatically when you start a new terminal session
source ~/.bashrc
sudo chown -R $USER:www-data /var/www/html
mkdir /var/www/html/robotapp
cd /var/www/html/robotapp
touch .env
nano .env
FLASK_APP=wsgi.py

```

```

FLASK_ENV=production

pipenv shell
pipenv install flask gunicorn
touch application.py wsgi.py

```

Create minimal flask application.py

```
nano application.py
```

```

from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    '''Index page route'''

    return '<h1>Application Deployed!</h1>'

```

Create minimal wsgi.py

```
nano wsgi.py
```

```

from application import app

if __name__ == '__main__':

    app.run(debug=False)

```

Create flask application

```

sudo apt install git<br>

# missing ssh keys? https://gist.github.com/adamjohnson/5682757<br>

cd /var/www/
sudo chown pi:www-data html
cd html
git clone https://writ@bitbucket.org/writ/robot.git

```

```
# For the camera
pipenv install picamera
### pipenv install opencv-python==4.5.3.56
### pipenv install imutils
```

Video-flask\_stream seems like a better path, looks like you just need a generator function

Test the flask application with :

```
gunicorn --workers 4 --bind 0.0.0.0:5000 wsgi:app
export FLASK_APP=app
export FLASK_ENV=development
flask run --host=0.0.0.0
```

<https://blog.miguelgrinberg.com/post/running-your-flask-application-over-https>

Install Adafruit library, necessary for interacting with the PWM driver board

```
pipenv install Adafruit-PCA9685
```

Video solution seems best using python...

<https://github.com/EbenKouao/pi-camera-stream-flask>  
main program

```
from __future__ import division

# Importing the Flask modules required for this project
from flask import Flask, render_template, request

# Import sleep module from time library to add delays
from time import sleep

# Import the PCA9685 module.
import Adafruit_PCA9685

# Initialise the PCA9685 using the default address (0x40).
pwm = Adafruit_PCA9685.PCA9685()

# Set frequency to 60hz, good for servos.
```

```

pwm.set_pwm_freq(60)

# Flask constructor takes the name of current module (__name__) as argument.
app = Flask(__name__)
# Enable debug mode
app.config['DEBUG'] = True

```

## Troubleshooting

These are a few errors that were encountered while installing the setup:

### Seg fault opencv

```
pip3 install --upgrade opencv-python==4.5.3.56
```

```
FileNotFoundError: [Errno 2] No such file or directory: '/dev/i2c-1'
```

Make sure to enable I2C interface options using `'sudo raspi-config'`

## Networking

Instruction applied to setup the network:-

In LuCI, go to Network -> Switch. First change the VLAN ID of your WAN network (identifiable by its untagged connection to the WAN port) from 1 or 2 to 11. Also change the VLAN ID of your LAN network from 1 or 2 to 21. Leaving space between VLAN IDs is often a good practice to prepare for any further network segmentation in the future. In the example figure at the beginning of this guide an IP range of 192.168.21.0/24 is used for the LAN network. This might be different for your network, and it is unnecessary to change this.

Now create a new VLAN with VLAN ID 41, which will be the guest network. Configure for this VLAN the embedded Ethernet adapters of the router with the same tagged configuration as VLAN ID 21 (the LAN interface).



## VLANs on "switch0"

VLAN ID	Description	CPU (eth0)	CPU (eth1)	LAN 1	LAN 2	LAN 3	LAN 4	WAN	
Port status:		1000baseT full-duplex	1000baseT full-duplex	no link	no link	no link	no link	1000baseT full-duplex	
<input type="text" value="11"/>	<input type="text" value="WAN"/>	<input type="text" value="tagged"/>	<input type="text" value="off"/>	<input type="text" value="off"/>	<input type="text" value="off"/>	<input type="text" value="off"/>	<input type="text" value="off"/>	<input type="text" value="untagged"/>	<input type="button" value="Delete"/>
<input type="text" value="21"/>	<input type="text" value="LAN"/>	<input type="text" value="off"/>	<input type="text" value="tagged"/>	<input type="text" value="untagged"/>	<input type="text" value="untagged"/>	<input type="text" value="untagged"/>	<input type="text" value="off"/>	<input type="text" value="off"/>	<input type="button" value="Delete"/>
<input type="text" value="41"/>	<input type="text" value="Guest"/>	<input type="text" value="off"/>	<input type="text" value="tagged"/>	<input type="text" value="off"/>	<input type="text" value="off"/>	<input type="text" value="off"/>	<input type="text" value="untagged"/>	<input type="text" value="off"/>	<input type="button" value="Delete"/>
<input type="button" value="Add VLAN"/>									
<input type="button" value="Save &amp; Apply"/> <input type="button" value="Save"/> <input type="button" value="Reset"/>									

In the default configuration all the LAN ports of the switch are assigned as untagged to the LAN network (VLAN 21). From VLAN 21, pick one LAN port and change the configuration from untagged to off. For the same LAN port, change for VLAN ID 41 the configuration from off to untagged.

Code:-

```
globals 'globals'
    option ula_prefix 'fd5a:d917:5819::/48'
config device
    option name 'br-lan'
    option type 'bridge'
    list ports 'eth1.21'
config device
    option type 'bridge'
    option name 'bridge-guest'
    list ports 'eth1.41'
config interface 'loopback'
    option device 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'
config interface 'lan'
    option device 'br-lan'
    option proto 'static'
```

```
option netmask '255.255.255.0'
option ip6assign '60'
option ipaddr '192.168.21.1'
config interface 'guest'
option proto 'static'
option device 'bridge-guest'
option ipaddr '192.168.41.1'
option netmask '255.255.255.0'
option delegate '0'
option type 'bridge'

config interface 'wan'
option proto 'dhcp'
option device 'eth0.11'
config interface 'wan6'
option proto 'dhcpv6'
option device 'eth0.11'
option reqaddress 'try'
option reqprefix 'auto'
config switch
option name 'switch0'
option reset '1'
option enable_vlan '1'
config switch_vlan
option device 'switch0'
option vlan '1'
option ports '6t 4'
option vid '41'
option description 'Guest'
config switch_vlan
option device 'switch0'
option vlan '2'
option ports '6t 1 2 3'
option vid '21'
option description 'LAN'
config switch_vlan
option device 'switch0'
option vlan '3'
option ports '0t 5'
option vid '11'
option description 'WAN'
```

## Appendix B - Certification of Ethical Approval



### Certification of ethical approval

The Athabasca University Research Ethics Board (REB) has reviewed and approved the research project noted below. The REB is constituted and operates in accordance with the current version of the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS2) and Athabasca University Policy and Procedures.

**Ethics File No.:** 24634

**Principal Investigator:**

Mr. Naqi Khan, Graduate Student  
Faculty of Science & Technology\Master of Science in Information Systems (MScIS)

**Supervisor:**

Dr. Qing Tan (Supervisor)

**Project Title:**

Virtual Training Lab (TLAB): Telepresence Empowered Remote Lab

**Effective Date:** April 21, 2022

**Expiry Date:** April 20, 2023

**Restrictions:**

Any modification or amendment to the approved research must be submitted to the AUREB for approval.

Ethical approval is valid *for a period of one year*. An annual request for renewal must be submitted and approved by the above expiry date if a project is ongoing beyond one year.

A Project Completion (Final) Report must be submitted when the research is complete (*i.e. all participants contact and data collection is concluded, no follow-up with participants is anticipated and findings have been made available/provided to participants (if applicable)*) or the research is terminated.

**Approved by:**

**Date:** April 22, 2022

Carolyn Greene, Chair  
Athabasca University Research Ethics Board