T.C.
BAHCESEHIR UNIVERSITY
GRADUATE SCHOOL
DEPARTMENT OF MECHATRONICS ENGINEERING

VIRTUAL REALITY ENHANCED UNDERWATER MANIPULATION WITH SOFT ROBOTIC LIMBS

MASTER’S THESIS

SUMEY EL MÜFTÜ

ISTANBUL 2024

T.C.
BAHCESEHIR UNIVERSITY
GRADUATE SCHOOL
DEPARTMENT OF MECHATRONICS ENGINEERING

VIRTUAL REALITY ENHANCED UNDERWATER MANIPULATION WITH SOFT ROBOTIC LIMBS

MASTER'S THESIS

THESIS ADVISOR

ASSOC.PROF. DR. BERKE GÜR

ISTANBUL 2024

T.C.

BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL

MASTER THESIS APPROVAL FORM

Program Name:	MECHATRONICS ENGINEERING (ENGLISH, THESIS)
Student's Name and Surname:	SUMEY EL MÜFTÜ
Name Of The Thesis:	VIRTUAL REALITY ENHANCED UNDERWATER MANIPULATION WITH SOFT ROBOTIC LIMBS
Thesis Defense Date:	06.06.2024

This thesis has been approved by the Graduate School which has fulfilled the necessary conditions as Master thesis.

Assoc. Prof. Yücel Batu SALMAN
Director of Graduate School

This thesis was read by us, quality and content as a Master's thesis has been seen and accepted as sufficient.

	Title/Name	**Institution**	**Signature**
Thesis Advisor's	Assoc. Prof. Dr. Berke Gür	Marmara Üniversitesi	
2nd Member	Prof. Dr. Barbaros Bostan	Bahçeşehir Üniversitesi	
3rd Member (Outside institution)	Prof. Dr. M. İ. Can Dede	İzmir Institute of Technology	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Sumey El Müftü

Signature:

ABSTRACT

VIRTUAL REALITY ENHANCED UNDERWATER MANIPULATION WITH SOFT ROBOTIC LIMBS

Sumey El Müftü
Master's Program in Mechatronics Engineering (English, Thesis)
Supervisor: Assoc. Prof. Mehmet Berke Gür

May 2024, 60 pages

This thesis explores the advantages of a virtual reality interface compared to a traditional two-dimensional screen for controlling an underwater squid robot with soft robotic limbs. Utilizing a Unity simulation, we tested both virtual reality and normal mode approaches, employing a wireless controller for input and a virtual reality headset for three-dimensional environmental feedback. The evaluation criteria included operator engagement, ease of use and control, task completion and average speed of manipulation and dizziness levels.

Experimental studies revealed that the virtual reality interface significantly improved task completion time by 29.82% and increased user engagement by 51.72% compared to the two-dimensional screen. Additionally, virtual reality mode enhanced ease of use and control by 76%, although it also increased dizziness levels by 57.14%. The results indicate that virtual reality technology can enhance remote robotic operations by providing more intuitive and realistic control options. However, challenges such as dizziness associated with virtual reality use were identified as potential drawbacks. Further studies are recommended to address these issues and fully exploit the advantages of virtual reality in underwater robotic teleoperation

Key Words: Virtual Reality, Underwater Robotics, Manipulation, Soft Robots, User Interfaces, Unity Simulation, Teleoperation

ÖZ

SANAL GERÇEKLIK ILE DESTEKLENMIŞ YUMUŞAK UZUVLAR ILE SUALTINDA MARIFETLI MANIPÜLASYON

Sumey El Müftü
Mekatronik Mühendisliği Yüksek Lisans Programı (İngilizce, Tez)
Tez Danışmanı: Doç. Dr. BERKE GÜR

Mayıs 2024, 60 sayfa

Bu tez, yumuşak robotik uzuvlara sahip bir sualtı mürekkepbalığı robotunu kontrol etmek için geleneksel iki boyutlu ekrana kıyasla sanal gerçeklik arayüzünün avantajlarını araştırmaktadır. Unity simülasyonu aracılığıyla hem sanal gerçeklik hem de normal mod yaklaşımları, giriş için kablosuz bir kontrol cihazı ve üç boyutlu çevresel geri bildirim için bir sanal gerçeklik başlığı kullanılarak test edilmiştir. Değerlendirme kriterleri arasında operatör katılımı, kullanım ve kontrol kolaylığı, görev tamamlama süresi ve ortalama manipülasyon hızı ile baş dönmesi seviyeleri yer almıştır. Deneysel çalışmalar, Sanal Gerçeklik arayüzünün görev tamamlama süresini %29,82 oranında önemli ölçüde iyileştirdiğini ve iki boyutlu ekrana kıyasla kullanıcı katılımını %51,72 oranında artırdığını ortaya koymuştur. Ek olarak, sanal gerçeklik modu kullanım ve kontrol kolaylığını %76 oranında artırmış, ancak aynı zamanda baş dönmesi seviyelerini %57,14 oranında artırmıştır. Sonuçlar, sanal gerçeklik teknolojisinin, daha sezgisel ve gerçekçi kontrol seçenekleri sunarak uzaktan robotik operasyonları iyileştirebileceğini göstermektedir. Bununla birlikte, sanal gerçeklik kullanımıyla ilişkili baş dönmesi gibi zorluklar, potansiyel dezavantajlar olarak belirlenmiştir. Bu sorunları ele almak ve sanal gerçeklik'in sualtı robotik teleoperasyonundaki avantajlarını tam olarak kullanmak için daha fazla çalışma önerilmektedir.

Anahtar Kelimeler: VR, Sualtı Robotları, manipülasyon, Yumuşak Robotlar, Kullanıcı Arayüzleri, Unity Benzetim, Teleoperasyon

To My Family

ACKNOWLEDGEMENTS

After praising and thanking ALLAH almighty, I wish to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Berke Gür for his guidance, advice, criticism, encouragements and insight throughout the research.

I am profoundly grateful to my mother, Iptisam El-Müftü, and my brother, Sami El-Müftü, for their immense support and understanding. Their constant encouragement has been crucial to my pursuit of higher education and the completion of this study.

Also, I want to give special gratitude to Oytun Kal (Executive Procedure at bug LAB TEKMER) and to Kutay Perk. (Unity XR Developer) and to Süleyman Faruk Koçyiğit (Level Designer) for their support in design of the Unity environment VR integration.

This research is supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) through the project entitled "Robotic Squid for Underwater Manipulation and Intervention" (grant no. 216M201).

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AR	Augmented Reality
DL-ATR	Deep Learning-based Automatic Target Recognition
DOB	Disturbance Observer
MR	Mixed Reality
NMPC	Nonlinear Model Predictive Control
PM-Net	Predictor Model Network
PS5	PlayStation 5
RBF	Radial Basis Function
ROS	Robot Operating System
TCP	Transmission Control Protocol
TPIK	Task Priority Inverse Kinematics
UNITY	Unity 3D Engine
UVMS	Underwater Vehicle-Manipulator System
VR	Virtual Reality
XR	Extended Reality

Chapter 1

Introduction

The ultimate objective of robotics is to develop completely autonomous entities that can function alongside with humans. Despite significant progress in robotics technology in recent years, the potential applications for autonomous robots remain rather restricted. In such instances, the use of teleoperated human-in-the-loop robots technologies may be feasible. Teleoperated robots have the ability to carry out jobs that are physically impossible or inconvenient for humans. Teleoperated robots have the ability to be swiftly deployed and operated in hazardous environments such as underwater or space. They provide a variety of practical applications that enhance workplace safety and efficiency. Recently, DARPA-organized tournaments have prominently shown teleoperated robots (Hudson et al., 2021), Although all teams used a certain degree of autonomy, there was a lack of agreement over the optimal amount of autonomy (Atkeson et al., 2021). The screen, keyboard, and mouse interface were standard in these events. Although a display interface has its benefits, virtual reality (VR) provides a potential alternative as immersive interfaces that enable operation in three-dimensional space.

Researchers have adopted the usage of virtual VR to enhance robot navigation and improve the user experience of controlling robots remotely. The enhanced performance and reduced cost of VR devices have made them more viable for teleoperating robots. Prior research has shown that VR offers a significant benefit in terms of enhancing a user's spatial awareness and immersion, in comparison to conventional methods (Whitney, Rosen, Phillips, Konidaris, & Tellex,). VR also enables unique experiences that were not achievable with earlier technology (LeMasurier, Allspaw, & Yanco, 2021).

1.1 Statement of the Problem

Although teleoperated robots have several benefits for working in places that are difficult to reach or dangerous, the interfaces used to control these robots may often restrict the performance and effectiveness of the operations they carry out.

Conventional 2D screen interfaces, while often used, may lack the necessary sense of environment and depth perception required for complex underwater tasks. This constraint could affect the operator's capacity to execute accurate operations and efficiently explore complex surroundings. Moreover, the comprehensive investigation of incorporating sophisticated control interfaces like VR to possibly address these constraints has not been extensively studied in the field of underwater soft robotics.

The existence of this gap highlights how important it is to evaluate and compare the functionalities of virtual reality interfaces with conventional two-dimensional displays in order to determine their effectiveness in improving the teleoperation of underwater robotic systems.

1.2 Purpose of The Study

The primary purpose of this research is to investigate and evaluate the effectiveness of a VR interface compared to a standard two-dimensional (2D) screen interface in remotely controlling underwater robot and limbs. This study employs a master device “PS5 controller” for input and a 3D feedback system linked to a Quest 2 VR headset to replicate the underwater environment. The research intends to determine if a VR interface can improve the teleoperation of underwater robots in comparison to traditional approaches, by focusing on performance, ease of use and control, and the operator's level of engagement.

1.3 Hypotheses / Research Questions

The hypothesis of this research is that operators will achieve better control and faster task completion using the VR interface compared to the traditional 2D screen interface in underwater manipulation.

While the Research questions are, how does the use of a 3D simulated environment through VR affect the operator's level of engagement with the environment, performance, and control accuracy compared to a 2D screen? And also, to see what are the disadvantages that come with using VR?

1.4 Significance of the Study

The results of this research are anticipated to provide valuable knowledge in the area of robotic teleoperation, especially in challenging environments such as underwater. The research aims to evaluate the possibility of VR technology in improving operator experience and efficiency by comparing VR and 2D interfaces. This validation might be crucial for applications like underwater exploration and maintenance. Additionally, this study has the potential to provide the foundation for upcoming developments in teleoperation interfaces and approaches, hence impacting the development of robotic control systems and associated technologies.

1.5 Definitions

Robot Operating System (ROS): Is an open-source middleware suite commonly used in robotics research and industry. It provides tools and libraries to help software developers create robot applications. ROS includes functionalities for hardware abstraction, device control, message-passing between processes, and package management. It is designed to support complex and heterogeneous robot systems across a wide variety of robotic platforms.

Unity 3D Engine: Unity is a cross-platform game engine developed by Unity Technologies, used to develop video games for web plugins, desktop platforms, consoles, and mobile devices. It is notably powerful for creating interactive 3D applications such as simulations, training programs, and visualizations, and is increasingly used in industries beyond gaming, including film, automotive, architecture, and virtual reality.

Virtual Reality (VR): VR is a simulated experience that can be similar or different from the real world. VR typically requires the use of electronic devices such as a headset with a screen or gloves fitted with sensors. It immerses the user in a digitally created environment where they can interact with 3D worlds. VR is used in a variety of applications, including entertainment, education, and training, allowing users to experience a situation in a controlled yet immersive manner.

Augmented Reality (AR): AR is a technology that overlays digital information (such as images, videos, or sounds) on the real-world environment in real-time. AR enhances the user's perception of reality by adding virtual elements to the physical world.

Extended Reality (XR): XR is an umbrella term encompassing all forms of computer-altered reality, including VR, AR, and MR. XR includes any technology that blurs the line between the real and digital worlds.

Mixed Reality (MR): MR merges real and virtual worlds to create new environments and visualizations where physical and digital objects coexist and interact in real-time. Unlike AR, MR allows for more interaction between real and virtual elements.

Teleoperation: Teleoperation refers to the remote control of robots or machinery. In teleoperation, a human operator controls the system from a distance, often using real-time feedback to perform tasks that might be dangerous or difficult to do in person.

WebSocket: WebSocket is a communication protocol that provides full-duplex communication channels over a single TCP connection. It is designed to be used in scenarios where real-time, bidirectional communication is required, such as in live chats, online gaming, and real-time updates in web applications.

Chapter 2

Literature Review

The exploration of VR-enhanced underwater manipulation with soft robotic limbs involves the confluence of underwater robotics, virtual reality, soft robotics, and simulation platforms. This literature survey provides a comprehensive review of these intersecting fields, enhancing the reader's understanding of the current state of research and its historical evolution.

2.1 Current State of Underwater Robotic Manipulation

The journey of underwater robotic manipulation began in the mid-20th century with the development of rudimentary remotely operated vehicles (ROVs) designed for tasks like submarine rescue and deep-sea exploration. These early systems laid the groundwork for the sophisticated autonomous underwater vehicles (AUVs) we see today. The drive to handle hazardous and inaccessible environments, especially following the advent of nuclear energy, spurred significant advancements in underwater robotics.

One of the recent milestones in this field is the SUONO project, as detailed by (Topini et al., 2021). This project represents a significant leap forward by integrating advanced technologies to develop autonomous underwater robotic systems capable of complex interventions. The Task Priority Inverse Kinematics (TPIK) framework, a key innovation of SUONO, manages multiple control objectives, ensuring the underwater vehicle manipulator system (UVMS) performs precise tasks like valve-turning with remarkable stability. The incorporation of Deep Learning-based Automatic Target Recognition (DL-ATR) enhances the robot's interaction with its environment, pushing the boundaries of underwater robotic capabilities.

Parallel to this (Cai et al., 2024), addressed the persistent challenges of autonomous manipulation in environments with unknown disturbances. They developed a composite control scheme combining a disturbance observer (DOB), a predictor model network (PM-Net), and nonlinear model predictive control (NMPC). This scheme leverages a radial basis function (RBF) neural network to estimate and compensate for disturbances in real-time, significantly improving control performance under varying conditions. This robust framework suggests a promising direction for enhancing underwater robots' operational capabilities.

As the field advanced (Morgan, Carlucho, Ard, & Barbalata, 2022), emphasized the necessity of achieving full autonomy in underwater manipulation. Transitioning from simulation-based methods to experimental validations, their work advocates for the creation of autonomous systems capable of dynamic intervention operations without human oversight. This highlights the need for advancements in perception systems and integrated system components to navigate the complexities of underwater environments effectively.

Contrasting with these autonomous systems, (Rafael, et al., 2021) presented TXRob, a low-cost teleoperated robot designed for post-disaster exploration. TXRob relies on traditional teleoperation control systems, highlighting potential enhancements achievable through more intuitive control mechanisms, and paving the way for future innovations in teleoperated robotic systems.

Despite these advancements, (Aldhaheri, De Masi, Pairet, & Ardon, 2022), review the ongoing challenges in underwater robot manipulation. Environmental complexities such as dynamic disturbances and poor visibility significantly hinder manipulation tasks. They argue that underwater manipulation still lags behind terrestrial robotics due to these challenges and the limitations of current sensing and actuation technologies. Their review calls for advancements in sensor integration and actuation methods to elevate the robustness and capabilities of underwater robotic systems.

2.2 Current State of Soft Robotic Control

The field of soft robotics, although relatively young, has seen rapid development since the early 2000s. Inspired by biological organisms, early soft robots were designed to navigate through complex environments with a level of flexibility and adaptability that rigid robots could not achieve. Over the years, advancements in materials science and computational modeling have led to the creation of sophisticated soft robotic systems capable of performing complex tasks. A seminal work in this domain is (Youssef et al., 2022), provide an insightful overview of underwater soft robotics, emphasizing bioinspired designs in actuation, modeling, and control. They highlight the adaptability and compliance of soft robots, which make them particularly suited for navigating challenging underwater environments. However, the review also identifies key challenges in developing functional soft robotic systems, particularly in the areas of modeling and control, which must be addressed to fully exploit their potential in complex tasks.

A significant step forward in this domain is the simulation study conducted by (Azizkhani, Godage, & Chen, 2022), They explored dynamic control strategies for a pneumatically actuated soft robotic limb, revealing the inherent challenges due to their nonlinear dynamics and external disturbances. Various control approaches, including kinematic control, PD + feedback linearization, and adaptive passivity control, were compared, ultimately highlighting the superior performance of adaptive passivity control in handling uncertainties and disturbances.

Furthering this field, (Ahmed et al., 2022), advanced further with the development of a multi-material bio-inspired soft octopus robot. Utilizing shape memory alloy (SMA) muscle wires for actuation, their innovative approach achieved synchronous swimming by precisely controlling the SMA muscles. This work illustrates how closely mimicking natural marine organisms can significantly enhance underwater robotic capabilities.

Continuing the narrative, (Fang et al., 2022) reviewed the evolution from efficiency to adaptability in biomimetic interactive soft robotics within wet environments. They delved into various bio-inspired designs, materials, and integration strategies that enable soft robots to perform effectively underwater. This focus on biomimicry underscores the importance of adapting soft robotic functionalities to emulate the versatility and resilience of marine life, addressing the challenges posed by dynamic aquatic conditions.

(Della Santina, Duriez, & Rus, 2023), provided a comprehensive survey of model-based control strategies for soft robots. They charted the progress from early machine learning techniques to sophisticated model-based approaches, revealing a significant shift towards finite-dimensional modeling to enhance soft robot functionality. This shift reflects a growing acceptance of model-based methods as viable solutions to the complexities inherent in soft robotic systems.

2.3 Current State of Virtual Reality Integration in Robotics

The integration of VR into robotics has a rich history dating back to the late 20th century. Early VR systems aimed to enhance human-robot interaction through immersive environments, although they were limited by the technology of the time. Over the years, advancements in computing power, graphics processing, and sensor technology have significantly improved the realism and responsiveness of VR systems, enabling their application in more complex robotic systems. A pioneering study by (Alemzadeh et al., 2021) investigated the use of VR for remote robotic surgery. Their system uses a VR headset to provide surgeons with an immersive view of the surgical site, enhancing precision and control. The integration of haptic feedback further improves the surgeon's ability to perform delicate operations, highlighting the potential of VR in medical robotics.

A seminal work in this domain is (Su et al., 2023), which provided a thorough review of extended reality (XR) applications in remote robotic operations, encompassing VR, augmented reality (AR), and mixed reality (MR). They highlight XR's role in enhancing human-robot interaction, offering depth perception and facilitating decision-making in dynamic settings, thus improving the effectiveness and performance of remote operations. Extending these concepts to underwater robotics, (Ma et al., 2022) developed a VR-based teleoperation system for AUVs. Their system provides operators with a 360-degree view of the underwater environment, enabling more intuitive and precise control of the AUV. The use of VR enhances situational awareness and reduces the cognitive load on operators, making it easier to perform complex tasks in challenging conditions.

Another significant contribution (Naceri et al., 2021), introduced the Vicarios interface, a VR-based system aimed at enhancing remote robotic teleoperation through immersive visualization and intuitive motion mapping. This innovative approach allows operators to "teleport" within the VR environment, significantly improving situational awareness and control during telemanipulation tasks.

(Xu, Moore, & Cosgun, 2022), demonstrated a VR-based teleoperation interface for robotic manipulation, where operators use VR to set task-space waypoints for a robot's end effector. This shared-control approach integrates 3D point clouds and video streams for enhanced environmental awareness, showcasing VR's potential in facilitating more intuitive and immersive human-robot interactions. The interface's success in pick-and-place scenarios underscores VR's utility in robotic teleoperation, though it also highlights areas for increased fluency and autonomy. Despite these advancements, challenges remain in integrating VR with underwater robotics. As noted by (Lin et al., 2021), issues such as latency, bandwidth limitations, and the need for high-fidelity sensors pose significant hurdles. Their review emphasizes the importance of developing new technologies and algorithms to address these challenges, ensuring that VR can be effectively utilized in underwater environments.

2.4 Current State of Simulation Platforms for Underwater Robotics

The development of simulation platforms for underwater robotics began with early simulations used for training submarine operators and testing underwater equipment. These early simulators were limited by the computational power and graphical capabilities of the time. However, the field has progressed significantly with the advent of advanced computational models, realistic physics engines, and high-fidelity graphics, enabling the creation of sophisticated simulation environments that closely mimic real-world underwater conditions. (Chaudhary, Mishra, Kalyan, & Chitre, 2021) developed an open-source underwater simulator using Unity3D and ROS, designed for simulating ROV operations such as pipeline inspection and tether-less control. This simulator enhances the realism of underwater missions by allowing complex environmental interactions and sensor integration within a virtual space, demonstrating Unity3D's potential in creating detailed, realistic simulations for underwater robotics research and development.

A noteworthy contribution in this field is by (Zhang, Fan, Liu, Zhang, & Sha, 2023), who developed an Autonomous Underwater Vehicle (AUV) simulation system using MOOS and Unreal Engine 4 (UE4). Their integrated framework emphasizes modularity and comprehensive functionality, providing a substantial foundation for algorithm verification, operational training, and software testing. This advancement leverages UE4's graphical and physical modeling capabilities, presenting a promising approach for realistic underwater simulations. Further advancing the field (Potokar, Ashford, Kaess, & Mangelson, 2022), introduced HoloOcean, an innovative underwater robotics simulator built on Unreal Engine 4, designed to facilitate the testing and development of algorithms in simulated marine environments. This open-source tool, featuring multi-agent support, various sensor simulations, and novel sonar imaging, exemplifies the importance of high-fidelity simulators for advancing underwater robotics research. By providing a platform for detailed environmental interaction and realistic sensor feedback, HoloOcean underscores the role of simulation in overcoming the challenges of underwater exploration and operation.

Despite these advancements, challenges remain in creating truly realistic and interactive simulation environments. As highlighted by (Li et al., 2021), issues such as computational complexity, the need for real-time interaction, and the accurate modeling of underwater physics pose significant hurdles. Their review emphasizes the importance of continued research and development in simulation technologies to overcome these challenges and support the advancement of underwater robotics.

2.5 Conclusion on Literature Survey

The field of underwater robotic manipulation has seen significant advancements over the past decade, driven by the need for reliable robotic systems capable of performing complex tasks in challenging environments. This literature review explored four key themes central to the development and application of underwater robots which are underwater robotic manipulation, soft robotic control, VR integration in robotics, and simulation platforms for underwater robotics. Virtual reality technology has a rich history, dating back to the 1960s when the first head-mounted display system was developed by Ivan Sutherland. Initially, VR was primarily used for military training and flight simulation. However, advancements in computing power and display technology have expanded VR's applications across various fields, including entertainment, education, and medicine. In robotics, VR began to be integrated in the late 1990s and early 2000s, offering immersive teleoperation capabilities that enhanced human-robot interaction and control. This integration enabled operators to perform complex tasks remotely with greater precision and awareness.

Despite the progress in these areas, a significant gap remains in the literature regarding the empirical study and real-world application of integrating VR with soft robotic manipulation for underwater environments. Most existing research has focused on either rigid robotic systems or general VR teleoperation without addressing the specific challenges of underwater applications. The unique dynamics of underwater environments, combined with the flexibility and compliance of soft robotics, require a distinct approach to control and feedback mechanisms. Moreover, there is a scarcity of studies that provide comprehensive empirical data on the effectiveness of VR feedback in enhancing operator performance and engagement in underwater tasks. This gap highlights the need for in-depth research to develop and validate systems that can seamlessly integrate VR feedback with soft robotic manipulators in underwater conditions. This thesis aims to address these gaps by developing a system that leverages VR technology to provide feedback from soft robotic limbs. This thesis aims not only to enhance the ergonomic and operational efficiency of underwater robotic systems but also to provide a comparative analysis with traditional 2D interfaces, thus offering a robust evaluation of VR's potential over conventional methods. The integration of these technologies could significantly influence future developments in robotic teleoperation, making advanced control systems more intuitive and accessible for a broader range of applications.

Chapter 3

Methodology

The teleoperation system being designed for the operational mode of the robotic squid consists of four manipulators on the slave system. Hence, the operation of these four limbs would be carried out remotely through a single master device. The detailed specifications of both the hardware and software components of the planned system, as well as the specific hardware and software that have been chosen and intended for usage are given in this chapter. The system is defined as a multi-master multi-slave configuration. A virtual reality simulation framework has been developed in the preliminary analysis.

3.1 Research Design

This research examines the creative use of VR to control underwater soft robotic limbs that mimic the movement of squid. A simulation framework was created allowing users to operate a simulation of the robot in a virtual environment.

The study used a case study design as its methodological technique. This architecture is well-suited for this research because it enables a thorough investigation of the interaction system, which involves an advanced mix of VR, robotics, and machine learning. The case study method is crucial for understanding the complexity of this integration and how users can adeptly manage the robotic limbs in a virtual reality environment.

The case study is tailored to the research questions, time and trajectory measurements, which seek to uncover how VR can enhance the manipulation of soft robotic limbs in an underwater scenario. The cross-sectional nature of this study, analyzing the system at a single point in time, is chosen to provide a clear picture of the current state of the system's functionality and user interaction, which is critical for identifying immediate areas for improvement and for testing the system's viability without the need for long-term observation, which would be necessitated by a longitudinal design. The research is developmental in design, as it seeks to establish a foundation for future real-world applications of VR in underwater soft robotics. The cross-sectional study provides a basis for potential longitudinal or sequential studies in the future, which could track the evolution of the system over time and across different development stages.

The significance of this research lies in its potential to improve the way humans interact with robots, particularly in environments that are challenging to access like underwater. The research will contribute detailed insights that could inform a wide range of applications, from underwater exploration to remote surgical procedures.

This part of the thesis will establish the foundation for the following parts on, data collecting and analysis, ensuring an integrated and rational flow of the study topic.

3.1.1 Virtual reality simulation framework. In the development of the control system for the soft robotic limbs, a virtual reality simulation framework was adopted. This configuration utilizes actual control hardware, including master device, VR headset and the software that will be implemented in the final physical system. The robotic limbs are represented within Unity simulation environment. This setup enables the testing of control strategies, software integration, and user interaction in a way that closely mirrors the intended real-world operation, but with the flexibility and safety of a virtualized context.

It was determined that the use of simulation framework was suitable for the design of the system. This is due to the fact that it enables the creation of the soft robot's limbs in a manner that is suitable to the budget.

For the purpose of this simulation, hardware components included both the software system and the physical equipment that make up the real system were used. In addition, the communication protocols that were designed to be used with the actual system were put into action.

3.1.1.1 Hardware. Hardware-wise, it has been determined that in addition to the necessary power adapters and connection cables, two computers, one master device, and an VR Headset are required. One of the computers is the master system and is selected as DELL G3 15 with Linux Ubuntu 20.4 operating system. The other computer is the slave system computer and is selected as DELL G16 7630 with Windows 11 operating system. The primary instrument for control in this study is the master device joystick controller, known for its precise joystick and button layout, offers a high degree of control and responsiveness, making it an excellent tool for teleoperation projects. Its ergonomic design ensures comfort during extended use, and its wireless functionality allows for flexibility in operator movement. The master device (Sony Interactive Entertainment, 2024) adaptability and user-friendly interface enhance the operator's ability to carry out complex actions in a simulated environment, underscoring its utility in complex robotic control tasks.

Also, used VR headset (Meta, 2024). designed to offer an immersive experience. It represents the latest advancement in VR technology, providing users with unparalleled visuals. This device offering a versatile platform for exploring virtual worlds, learning environments, and creative projects.

3.1.1.2 Software. This project used the features of the Unity 3D engine, together with a set of Python scripts, to achieve the simulation and control of the soft robotic system. Table 1 provides a clear breakdown of the different software components and scripts used, along with a detailed explanation of the particular functionality of each script. Each script and program will be explained in next sections.

Table 1

Overview of Software Components and Script Functionalities

Software/Script Name	Description of Functionality
Script 1 (Tip_position_input_Handler.py)	Interprets user input from the physical controller and translates it into simulation commands.
Script 2 (Kineformer.py)	Implements the machine learning model to determine the kinematics of the robotic limbs.
Script 3 (Limb_Dynamic_Translator.py)	Simulates the physical dynamics of the robotic limbs based on the input data.
Script 4 (Matrix_Subscriber.cs)	Used to create and render the virtual environment for the soft robotic robot inside Unity.

The whole system consists of several subsystems, and in order to guarantee seamless communication without any synchronization problems. Among these subsystems, the software library known as ROS (ROS Noetic, 2020) is used. ROS is a freely available software that is especially built to work well with both C++ and Python programming languages. ROS has optimal performance on operating systems that are based on Linux. Given that the computer intended for use on the robot already runs on a Linux-based operating system. Through its publisher/subscriber model, ROS efficiently manages the flow of information between the controller, the processing scripts, and the simulation environment, ensuring a cohesive and synchronized teleoperation experience.

3.1.2 Deployment of VR-enhanced control system. The teleoperation system is divided into two main components: the master system and the slave system. The master system consists of a master connected to a computer (Laptop1) that has the master programming to operate the robot. Simultaneously, a second computer (Laptop2) runs the Unity simulation to virtually simulate the slave robot. Furthermore, the operator will be equipped with a VR headset that is directly connected to the slave system. This VR headset serves as an interface to present a comprehensive and immersive simulation of the slave environment. It offers a multi-dimensional and interactive experience that goes beyond traditional video streams, allowing the operator to perceive and manipulate the soft robotic limbs with a high degree of precision and depth. Through this system, the operator receives intricate feedback, enhancing the perception of the underwater scenario and enabling more accurate and nuanced control over the robotic limbs in the simulated environment. Figure 1 shows an overview of the teleoperation setup. Detailed examinations of the sub-systems are provided in their respective sections. The master system serves as the control hub for operator interactions, utilizing a single master device designed to command the virtual representation of an underwater squid manipulators. The robot is equipped with four soft robotic limbs. Each limb is designated a specific role: two of them is known as task limbs (Right and Left Arms), third as a light limb (Left Tentacle), and the last as a camera limb (Right Tentacle). The movement of each manipulator is achieved by pulling four internal cables with specific force value to achieve required limb shape.

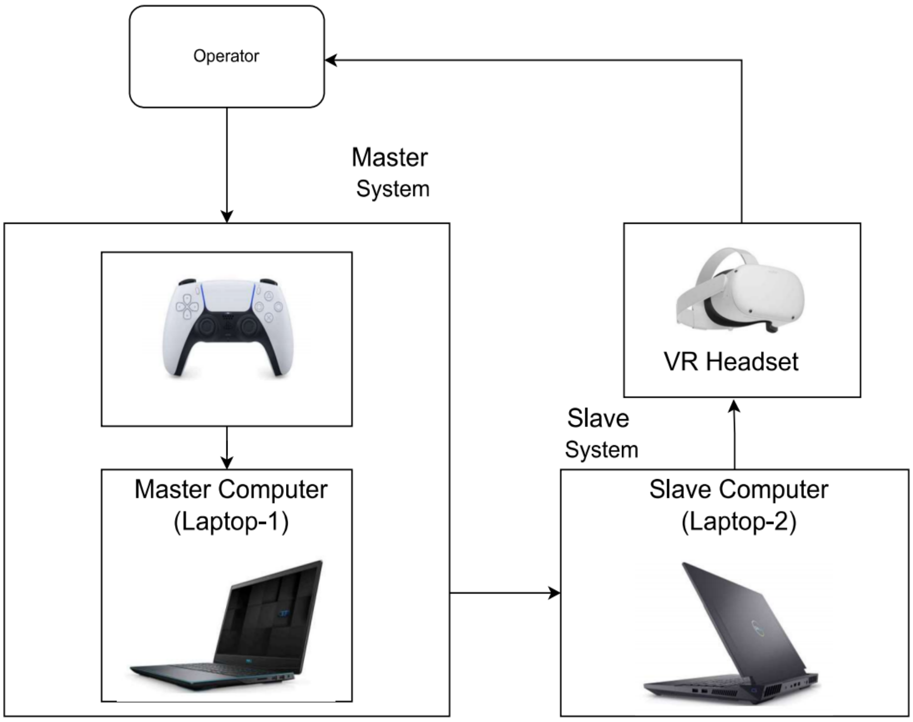

Figure 1. Schematic of teleoperation system

Tip position input from the operator for each limb through the master device is processed by the “Kineformer” script (see Table 1), which computes the necessary pulling forces on the cables to achieve the desired movement. These calculated forces are then translated into simulation data of the manipulator's shape by the “Limb_Dynamic_Translator” script.

The master computer is tasked with gathering input data, executing the control scripts, and managing communication through to the simulation. ROS is used for data transmission, enabling seamless integration and effective communication with the Unity-based simulation. This process ensures that the simulation receives accurate movement data, which is visually represented within the virtual environment. The architectural schematic detailing the master system's operation is depicted in Figure 2. The slave system consists of the Unity engine, which is responsible for the detailed simulation of the robotic limbs. The major function of the slave system is to understand and display the data that is delivered from the master system. The system converts the position information it receives into precise motions inside the simulated environment, simulating the behavior of each limb. In addition, the operator uses a virtual reality

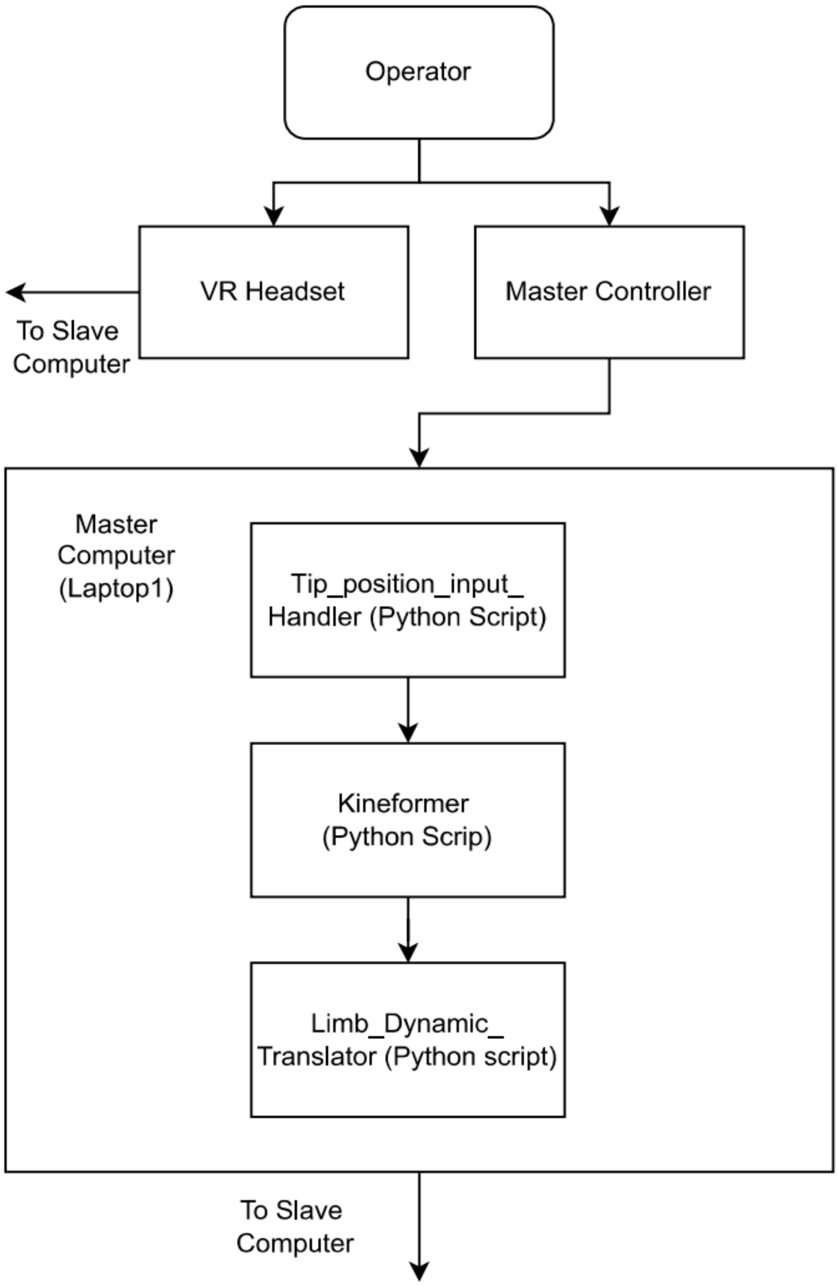

Figure 2. Schematic of master system

headset connected to the slave system, allowing the operator to visualize and control the limbs within the simulated underwater environment. The slave system consists of the Unity engine, which is responsible for the detailed simulation of the robotic limbs. The major function of the slave system is to understand and display the data that is delivered from the master system. The system converts the position information it receives into precise motions inside the simulated environment, simulating the behavior of each limb. In addition, the operator uses a virtual reality (VR) headset connected to the slave system, allowing the operator to visualize and control the limbs within the simulated underwater environment.

3.1.2.1 Controller mapping and button functions. This subsection will detail the mapping of the master device used in "Tip_position_input_Handler" script, providing a clear guide to how each button and joystick influences the teleoperation of the underwater soft robot. It encompasses an exhaustive list of button assignments and joystick axes as defined in our programming, linking these inputs directly to the robot's limb movement and adjustment mechanisms. Also detailed figures will be provided which contain descriptions of the actions triggered by the joysticks, buttons, and other controller features, linking these controls to the specific movements and adjustments of the robot's limbs. Table 2 describes master device internal configuration, correlating each button and joystick with designated control functionalities within the code. This comprehensive mapping ensures precise teleoperation of the underwater soft robot, translating user inputs into specific robotic movements.

- Left joystick gear: The vertical movement of the left joystick gear up and down directions directly adjusts the Y-axis of the left limb tip position. Pushing the joystick forward increases the Y-axis value, whereas pulling it backward decreases the Y-axis value, enabling precise vertical positioning.While the horizontal movement of the left joystick gear right and left directions directly adjusts the X-axis of the left limb tip position. Pushing the joystick right increases the X-axis value, and moving it to the left decreases it, facilitating lateral adjustments.
- L1 button: Pressing the L1 button increases the Z-axis value of the left limbs, effectively changing tip position z value.
- L2 button: Pressing the L2 button decreases the Z-axis value for the left limbs, effectively changing tip position z value.
- Right joystick gear: The vertical movement of the right joystick directly adjusts the Y-axis of the right limb s tip position. Pushing the joystick forward increases the Y-axis value, whereas pulling it backward decreases the Y-axis value, enabling precise vertical positioning. While the horizontal movement of the right joystick directly adjusts the X-axis of the right limbs tip position. Pushing the joystick right increases the X-axis value, and moving it to the left decreases it, facilitating lateral adjustments.

Table 2

Mapping of Master Controller Buttons and Axes to ROS Joy Message Inputs

Joystick Input Buttons and Axes Names	**Joy Message Input**
Left Joystick Gear (Up and Down)	Up: - axes [1] / Down: + axes [1]
Left Joystick Gear (Right and Left)	Left: - axes [0] / Right: + axes [0]
L1 Button	Button [4]
L2 Button	Up: - axes [2] / Down: + axes [2]
Directional Pad (Up and Down)	Up: - axes [7] / Down: + axes [7]
Directional Pad (Left and Right)	Left: - axes [6] / Right: + axes [6]
Right Joystick Gear (Up and Down)	Up: - axes [4] / Down: + axes [4]
Right Joystick Gear (Right and Left)	Left: - axes [3] / Right: + axes [3]
R1 Button	Button [5]
R2 Button	Up: - axes [5] / Down: + axes [5]
Triangle Button	Button [2]
Circle Button	Button [1]
Cross (X) Button	Button [0]
Square Button	Button [3]
Select Button	Button [8]
Start Button	Button [9]
PS Button	Button [10]
L3 Button	Button [11]

- R1 button**:** Pressing the R1 button increases the Z-axis value of the right limbs, effectively changing tip position z value.
- R2 button: Pressing the R2 button decreases the Z-axis value for the right limbs, effectively changing tip position z value.
- Directional pad up and down: Controls the robot's surge, managing forward and backward body movements to navigate the underwater environment effectively.
- Directional pad left and right: Manages the robot's sway by controlling left and right body movements, aiding in lateral navigation.
- Triangle button: Serves as a heave control, increasing the robot's elevation for upward movement.

- Cross (X) button: Another heave control, this time decreasing the robot's elevation for downward movement.
- Circle button: Acts as a yaw control, rotating the robot's body to the right, facilitating turning and directional changes.
- Square button: Another yaw control, rotating the robot's body to the left, facilitating turning and directional changes.
- PS button: Toggles the control scheme between manipulating the robot's limbs, a crucial feature for switching focus during operations. Only two limbs can be controlled at once.
- Select button**:** Resets the tip positions of the limbs to their initial predefined states, ensuring a standardized starting point for operation.
- L3 button: During test 1 and test 2, the limbs will be tasked with holding and moving certain objects. These objects will be attached securely to the limbs. By pressing the L3 button, these objects will be detached / released.

This detailed breakdown will be supported by visual aids in Figure 3 and Figure 4, showing each button's location and function on the master device to offer a clear, intuitive understanding of the control setup for teleoperation. Additionally, Figure 5 illustrates the various movements of the robot including Heave, Sway, Surge, Pitch, Roll, and Yaw, providing a view of how the robot operates in its environment. Also includes the limb tip position axes to demonstrate the specific movements of the robotic limbs as explained earlier.

3.1.2.2 Input handling script. The "Tip_position_input_Handler" script is a Python script created to connect human input with robotic action. The system's initial interaction points capture and interpret the operator's intents using master input device. This section explains how the script converts manual inputs into commands for the underwater soft robot.

The script's functionality relies on multiple important features, each playing a role in accurately translating joystick movements and button clicks to operate the robot's tip position of the limb. Utilizing the JOY Library, which is a ROS driver compatible with input master device, it reads the movements and button presses of the controller. This is facilitated through joy_node, which emits "Joy" topic transferring the current controller commands (ROS Joy library,2022).

Figure 3. Top view of PS5 controller with defined control functions

Figure 4. Upper buttons view of PS5 controller with defined control functions

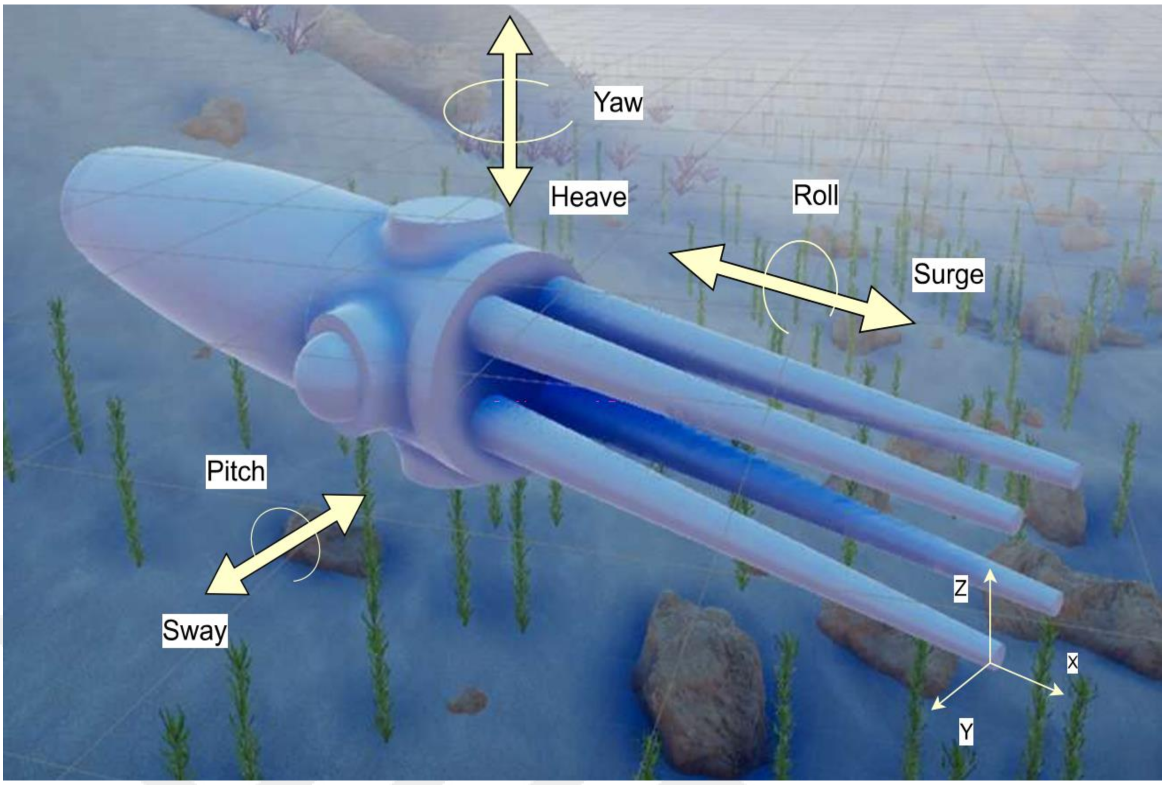

Figure 5. Movements and axes of robot

The process progresses through a sequence of steps, starting with input capture and ending with the integration of these inputs into the robot's control system. The ROS framework significantly contributes to this process by providing a robust communication infrastructure that allows seamless data transfer between different components of the system. ROS topics enable the exchange of data and control commands, ensuring that the operator's inputs are accurately captured and transmitted to the robot.

The script dynamically adjusts the positions of up to three coordinate (x,y,z), representing the limbs tip positions, within a 3-D environment. These positions are published as Vector3 messages through ROS topics. These data are fed to "Kineformer" script which is a machine learning algorithm that can interpret and act upon these data. The script processes Vector3 messages to understand the current positions of the limbs tip positions and use it to determine the optimal forces and torques needed to pull each cable inside the limb to archive required position. Figure 6 provides a detailed description of the "Tip_position_input_Handler" code's functionalities. The flowchart visually depicts the operational logic and procedural

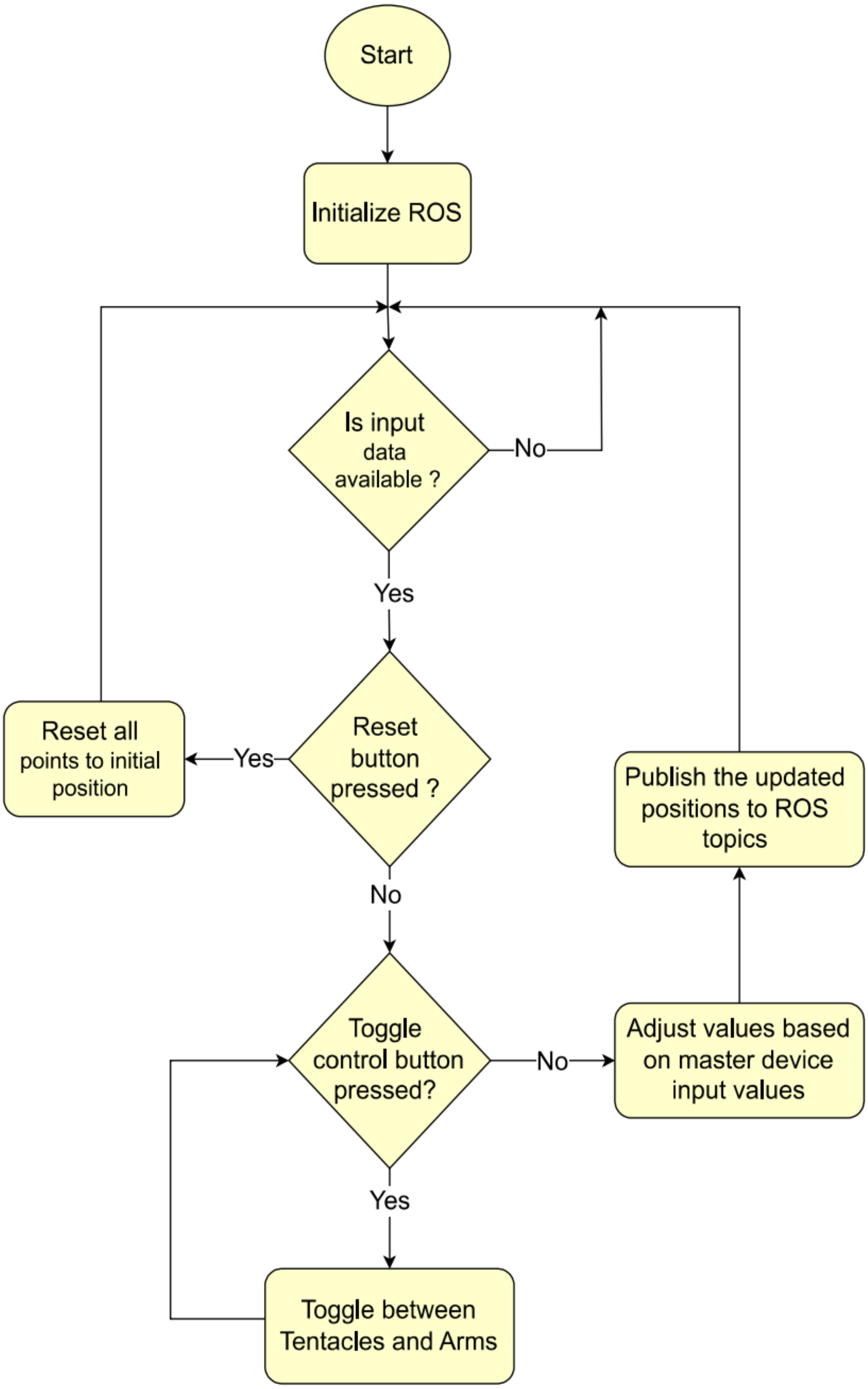

Figure 6. Tip_position_input_handler script flowchart

flow of the script, offering an illustrative guide that maps out the step-by-step process from the initial input capture to the final integration of these inputs into the robotic control system. The flowchart begins with the initialization of the ROS framework, which sets up the necessary nodes, subscribers, and publishers for communication. First it checks if input data from the master device is available or not, if no data is available then it keeps listening for any input data. When input data is detected, the system determines if the reset button is pressed. If the reset button is pressed, all the limbs will be back to their initial position. If the reset button is not pressed, the system checks if the control toggle button is pressed. If the toggle button is pressed, the system switches between controlling the tentacles and the arms. If the toggle button is not pressed, the system adjusts the values based on the input from the master device. Finally, the updated positions are published to the ROS topics, ensuring that the control commands are sent to the slave system. It offers a clear, concise overview of how each component of the script interacts, highlighting the sequential steps involved in translating user inputs from the master device into actionable commands for the underwater soft robot.

3.1.2.3 Machine learning script. The “KineFormer” machine learning script (Alkhodary & Gur, 2023), leveraging transformer architecture, is introduced as a solution for controlling soft manipulators. Unlike traditional methods, “KineFormer” utilizes an autoregressive model that incorporates previous actions to ensure consistency and uniformity in the manipulator's movement.

This approach allows for precise control over the soft robotic limbs tip positions, it surpasses traditional Feedforward neural network (FFNN) methods by providing a more reliable prediction of actuation forces needed for specific positions in 3-D space. For more detailed information full article (Alkhodary & Gur, 2023) can be reviewed. Tip position values from the controller will be sent to this script, and these values are inputted into the "Kineformer" machine learning model. The model predicts the required displacement forces as pull forces in the four internal cables inside each limb to position them at the exact location specified by the user. This process ensures the execution of the operator's commands accurately. Tip position values from the controller will be sent to this script, and these values are inputted into the "Kineformer" machine learning model. The model predicts the required displacement forces as pull forces in the four internal cables inside each limb to position them at the exact location specified by the user. This process ensures the execution of the operator's commands accurately.

3.1.2.4 Conversion to visual representation. In this section, we explore the transformative process of converting tendon force calculations into matrix values representations for simulations. Central to this methodology is the "Limb_Dynamic_Translator" Python script, which is translating tendon actuation forces into a (3x601) matrix representation. This matrix symbolizes 601 3-D points that accurately portray the configurations of limb-like structures, closely mimicking the movement capabilities of a limb.

The underpinning of this transformation is a geometrically precise, steady-state model for tendon-driven limbs, directly inspired by the complex anatomy and movement of an octopus limb. This model is predicated on a continuum-based approach (Renda, Cianchetti, Giorelli, Arienti, & Laschi, 2012), optimizing for rapid computations. It encompasses an elaborate actuation mechanism model, drawing from the continuum mechanics to simulate the limb s behaviour accurately.

The implementation of an octopus-inspired robotic limb involves a single conical silicone segment, modeled as a Cosserat beam, and actuated by four embedded cables. These cables are anchored at different lengths from the limb's base via rigid plastic discs, allowing the limb to perform complex bending and twisting motions in three dimensions. Using Cosserat theory, the limb's shape is described through longitudinal strain and curvatures, with forces and moments derived from Newtonian mechanics. The model incorporates material properties like Young’s modulus and mass density. A two-stage numerical scheme is employed for forward kinematics, first solving for strain and curvatures, then computing position and orientation. Random tendon forces generate a dataset to train machine learning models for predicting actuation values.

The model covers a crucial gap in soft robotics by clarifying the intricate connection between actuation forces and the limbs shape. The average speed of computation allows it to be used on microcontrollers for control and monitoring, representing an important breakthrough in this field.

3.1.2.5 Virtual modelling of soft robotics limbs. The matrix values representing the limb shape obtained from last section are incorporated into a Unity simulation through the ROS# Library (ROS Noetic, 2020), utilizing web-socket communication.

This stage visualizes the soft robot's movements in a virtual underwater setting, mirroring real-world dynamics.

Unity is a powerful and versatile development platform widely recognized for its robust capabilities in creating interactive 3D and 2D content. In the context of robotics and VR, Unity offers an array of features that make it an ideal choice for simulating complex systems and environments (Unity Technologies, 2024). Unity's significance in robotics and VR is derived from certain important features:

- Visual fidelity: Unity's rendering engine allows for the creation of highly detailed and realistic visual simulations. This visual accuracy is crucial when replicating the physical world within a virtual space, ensuring that the robot's interactions with its environment can be closely studied and understood.
- Flexibility and customization: Unity's customizable C# scripting capabilities enable developers to create complex behaviours and control systems. This flexibility is particularly valuable for simulating the complex movement of a soft robot, which requires precise control over numerous variables.
- Integration with other systems: Unity can connect with a variety of software and hardware systems. Its compatibility with ROS is a prime example, allowing for seamless data exchange between the robot's control system and the simulation environment.
- Virtual reality support: Unity has built-in support for VR, making it straightforward to implement immersive VR experiences. For teleoperated robots, this means that operators can control the robot as if they were physically present in the simulated environment, leading to more intuitive control and better understanding of the robot's capabilities and limitations.

The most important reason we chose Unity in this work is because of its strong support for virtual reality. Unity's capabilities to produce high-fidelity visuals and support VR make it a top choice for our project, facilitating the creation of a responsive and lifelike representation of the squid soft robot, its environment, and the VR testing framework. In summary, Unity is not just a game development engine but also a comprehensive platform for simulating and testing robotic systems within an interactive and realistic 3D environment.

Within the Unity environment, the simulation of the squid soft robot's limbs is handled by the “Matrix_Subscriber” script. This script is a crucial component of the system, as it establishes the connection between the ROS messages containing matrix data from "Limb_Dynamic_Translator" script and the virtual representation of the robot's limbs in Unity.

In our underwater robot simulation, the robotic squid is designed with four primary manipulators, mirroring the versatility and adaptability of a real squid's limbs. This configuration includes two upper limbs, named Right-Tentacle and Left-Tentacle, and two lower limbs, named Right-Arm and Left-Arm.

RT and LT: These upper limbs are specialized for observation and illumination within the simulated environment. The Right-Tentacle is equipped with a camera to provide visual feedback and facilitate navigation, while the Left-Tentacle houses a light source to illuminate the robot's surroundings. These limbs are designed not to interact physically with objects, focusing instead on enhancing the robot's sensory capabilities.

RA and LA: Contrary to the upper limbs, the lower limbs are designed for direct interaction with the environment. The limbs function as grasping and palpation tools, allowing the robot to manipulate objects. These limbs are integral to the robot's ability to perform tasks requiring precision and tactile feedback, such as collecting samples or interacting with underwater structures.

To visualize this configuration, a detailed diagram is included Figure 7. This illustration demonstrates the positioning and functionality of each limb, providing a clear understanding of their roles within the robotic system.

Our simulation approach begins by establishing communication with the ROS This system acts as the central processing unit of our experimental setup, main throughly receiving and processing input data. The communication is focused on monitoring messages transmitted via the /matrix_data topic from “Limb_Dynamic_Translator” code.

The messages contain numerical data arranged systematically in a format known as Float32MultiArray. This format is a structured array of coordinates that specifies the exact spatial location needed for each segment of our robotic limb in a virtual environment. The robotic limb is viewed as a series of interconnected segments/objects. Within Unity, the segments are created as GameObjects in our selected simulation platform. Total 601 GameObjects are used for each limb, each symbolically called a 'bone'.

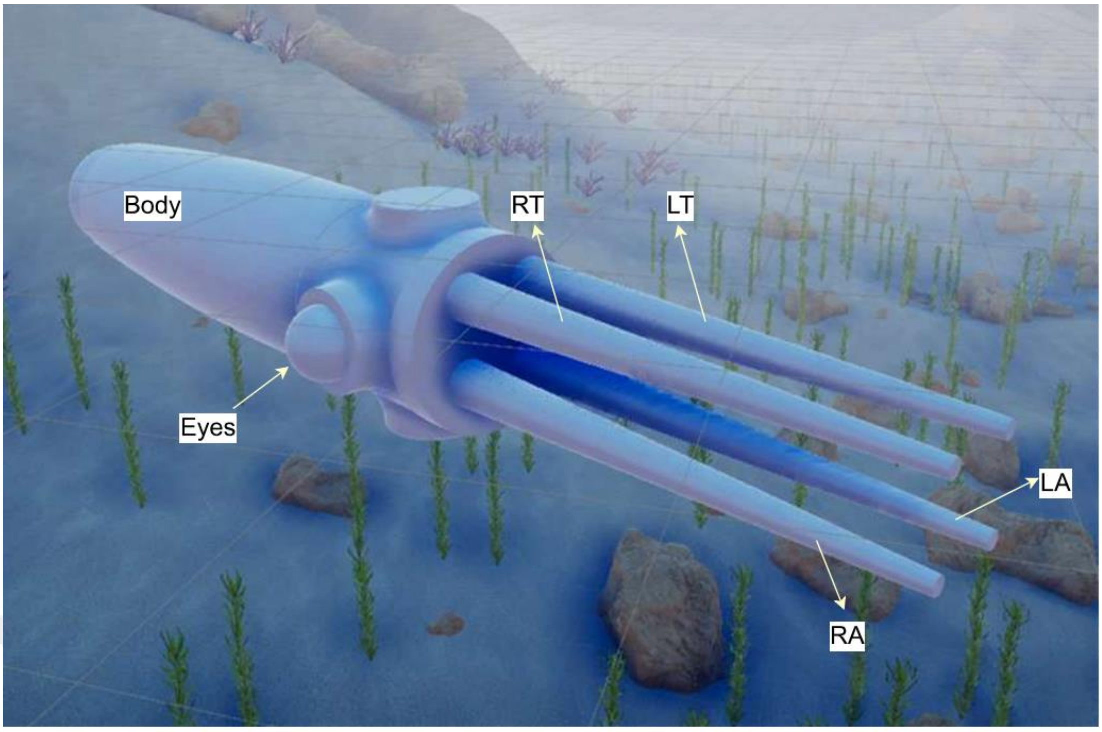

Figure 7. Layout of squid soft robots with limbs

This naming convention compares to the skeletal structure of a limb, helping with the movement and flexibility of the robotic limb. Each 'bone' is associated with a certain coordinate, which is determined from the array of 601 points received from “Limb_Dynamic_Translator” code. It aims to coordinate the robotic limbs movements to align with the predetermined objectives of the simulation.

The Receive Message function is triggered upon the receipt of new matrix data from the ROS topic. It decodes the message into a series of 3-D vectors (Vector3), Each vector represents a point in Unity's 3D space, and each point is assigned to its corresponding game object from the limb, as demonstrated in Figure 8.

An update loop is triggered inside “MatrixSubscriber” script. Within this loop, the script iterates through the points, updating the position and orientation of each bone in the limb. The orientation of each segment is adjusted to point towards the next segment in the list, creating a natural bending motion that simulates the smooth movements of a squid's limb. The script also contains logic to handle the end of the limb, where the last segment is treated separately since it does not point towards a subsequent segment. The position is updated while the rotation is matched to the second-to-last segment.

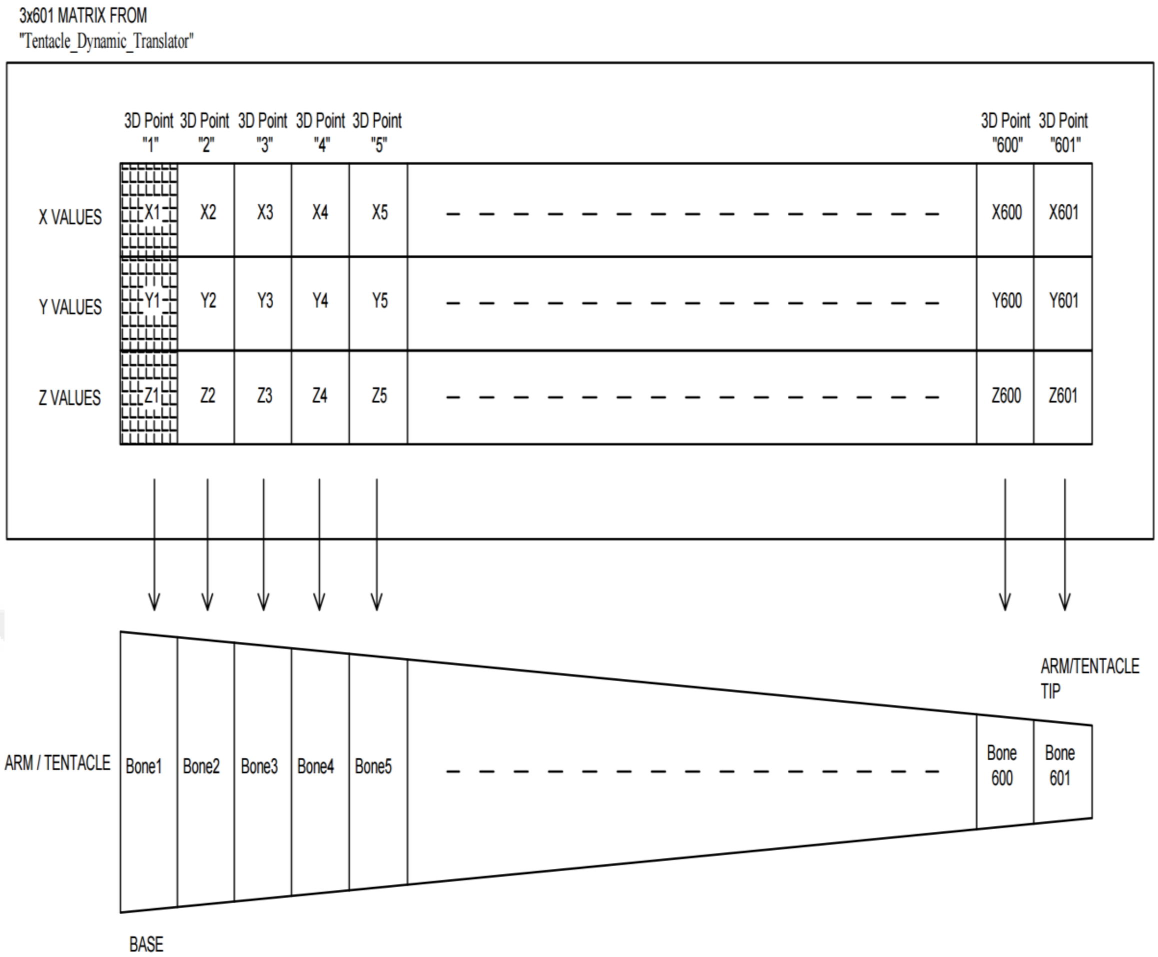

Figure 8. Mapping of 3x601 matrix data to limb segments in Unity

In summary, the Matrix_Subscriber script is a sophisticated bridge that not only accurately translates the ROS messages into Unity's coordinate system but also dynamically updates the virtual model's geometry. This results in a responsive and life-like simulation of the squid soft robot's limbs, effectively visualizing the complex movements that are calculated by the machine learning algorithms in the ROS framework.

3.1.3 VR headset feedback. In this setup, the operator uses the VR headset to receive immersive visual feedback from the simulation running on the slave system allowing the operator to feel like they are physically present in the underwater environment. The VR environment provides the operator with a realistic and immersive view of the underwater surroundings, allowing them to visually assess their surroundings and the positioning of the robotic limbs with depth and perspective.

The integration of VR feedback enriches the operator's experience by providing a more intuitive and immersive interface for manipulating the soft robotic limbs. It enhances surroundings understandings, perception, and responsiveness, ultimately improving the efficiency and effectiveness of underwater manipulation task.

Due to software compatibility constraints, the simulation of the VR environment for the underwater squid soft robot was implemented on a Windows operating system. This shift was necessitated by the VR headset software requirements, which is not supported on Linux platforms. The Windows operating system on a secondary laptop (Dell G16 7630) facilitated the use of the Oculus app, thereby enabling a seamless VR experience.

The VR headset was selected for its superior performance and versatility in VR applications. Its integration into the system was achieved through a USB Type-C connection, ensuring high-bandwidth data transfer and low-latency feedback essential for real-time simulation. The VR headset is advantageous for our simulation requirements due to its High-Resolution display that offers a life-like visual representation of the underwater environment and the robot's interactions. Additionally, the availability of developer tools and a supportive community helps in troubleshooting and improving the VR experience.

In the VR test setup, Unity software serves as the platform for the VR environment, rendering the movements of the soft robot as dictated by the matrix data received through the ROS-Bridge WebSocket connection. This rendering is critical for evaluating the robot's performance and user interaction in a controlled virtual setting that mimics real-world conditions. The VR simulation setup exhibits the project's interdisciplinary approach by connecting advanced robotics with user-friendly interfaces. This showcases how VR technology can be used to control and test robots in difficult-to-replicate situations, leading to advancements in telepresence and remote operation of underwater robotics.

3.1.4 Comprehensive system architecture. For a comprehensive understanding of the entire system's workflow, please refer to Figure 9, which presents the System block diagram. This diagram illustrates the interconnections of each component within the data collection process, highlighting the flow from initial input through to the simulated robotic movement.

3.2 Universe and Participants/Working Group

The study aims to assess the efficacy of using VR technology to promote interaction with a robotic system among various users and to determine any performance differences between normal traditional screen and VR-operated control modes. This section describes the participant demographics and selection criteria for the experimental study examining standard screen-based and VR interfaces for underwater robotic manipulation tests.

Participant characteristics: This study includes persons with different levels of experience in engineering and gaming. This varied group of participants seeks to investigate the usability and learning curve of the virtual reality simulation framework among users with differing backgrounds.

Selection criteria: Participants are chosen to guarantee a balanced representation in gender, racial/ethnic diversity. The objective is to select a sample size in order to gain a thorough insight into the system's usability among various demographic groups. The selection was made to ensure gender-balanced participant as possible. Total 12 Participants subject were selected (6 Males, 6 Females).

Background diversity: The group is divided equally between those with engineering expertise and those with gaming background. This aims to determine if previous experience in similar domains influences the ease and effectiveness of operating the robotic system.

To ensure confidentiality and privacy of participants, all data will be anonymized, and personal identifiers will be deleted from the study records. Participants will get information on confidentiality measures.

3.3 Data Collection

In this section, we detail the procedures employed for data collection, encompassing the execution of tests, and the subsequent analysis of gathered data. The utilization of Unity-based simulations, participant interaction, questionnaire responses, time and positional data recorded for limb tip parts constitutes the core of our data collection methodology.

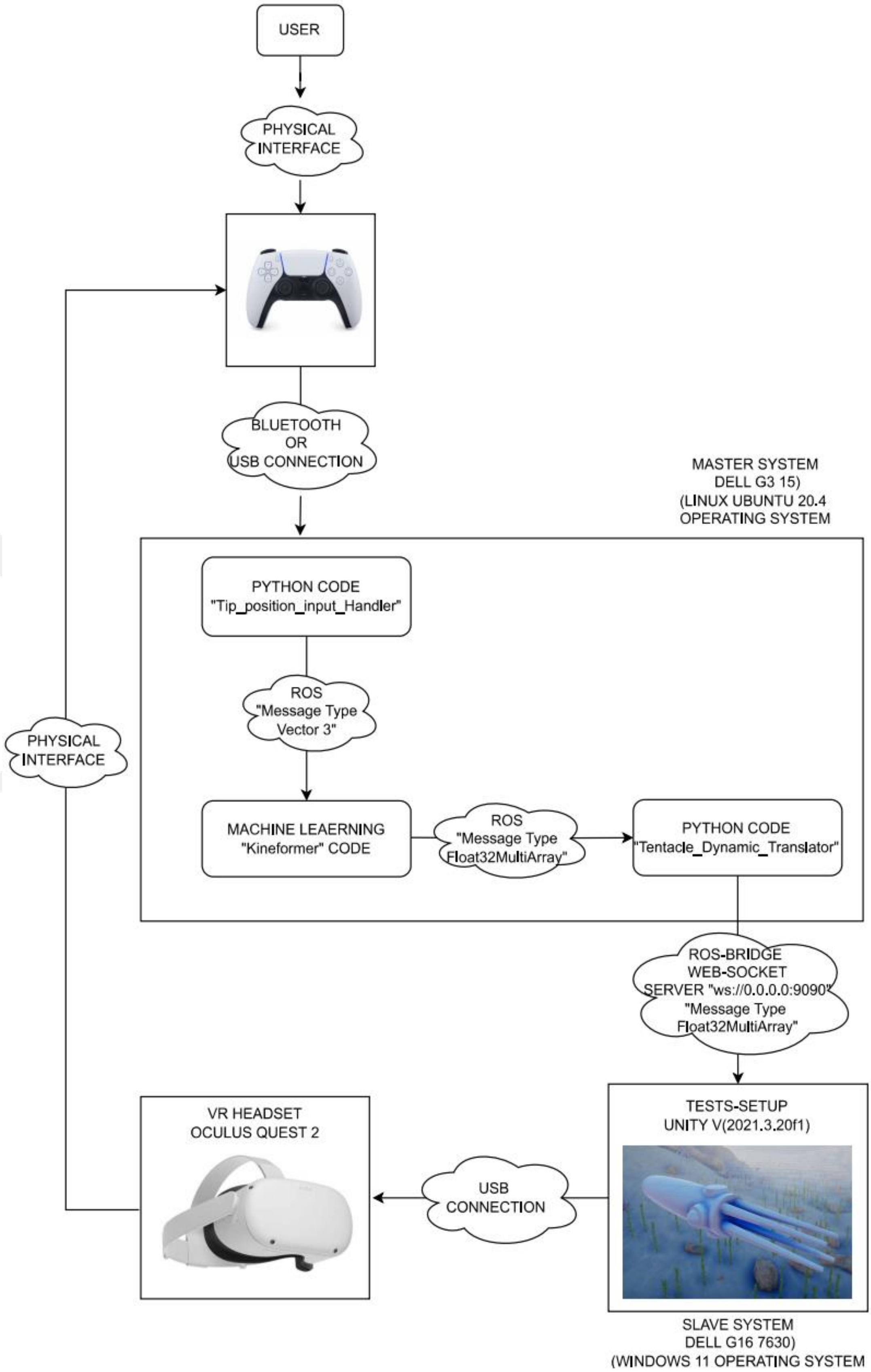

Figure 9. Virtual reality simulation framework schematic

3.3.1 Data collection instruments. Data collection tools are instrumental in capturing and recording various aspects of participants' interactions and experiences within the study environment. These tools encompass a range of instruments and methodologies aimed at systematically collecting data to address the research objectives. In this section, we outline the key tools employed in our study and discuss their significance in enabling comprehensive data collection.

3.3.1.1 Unity tests. The Unity environment was utilized to develop three distinct tests to assess participants' performance in manipulating soft robotic limbs under different conditions. Each test was designed to measure specific aspects of participants' abilities and interactions with the virtual environment. Additionally, time recording and limbs tip position recording mechanisms were implemented to accurately capture the duration of each test session. Each test had been conducted two times one through standard laptop screen (normal mode) and one through VR headset (VR mode).

Before performing each of the three tests, a trial time was given to each participant to allow them to get used to the test scenario. This verified that participants know the activities they were obligated to do with precision. The trial period was included to reduce variations in measured time between the two modes of operation by ensuring that participants were familiar with the test procedures. This method enabled more precise comparisons of the lengths of tests between the normal and VR modes.

Test 1: In this test, participants manipulated robotic task/manipulation limbs to touch an object represented by a sphere. The sphere will move to different position after it is being touched, with the task being making contact with the sphere five times. The test begins with the robot limbs in their initial position. Time recording was automated within Unity through a script that closed the game once the task was completed. This automated feature ensured precise time measurement and accuracy of data collection. Additionally, the script recorded the x, y, z coordinates of the limb tip, providing trajectory detailed positional data for analysis., with trials conducted both through a normal mode and VR mode. Challenges of Test 1:

- Precision and control: Manipulating the robotic limbs accurately to touch a target requires precise control skills, especially as the object changes position upon contact.

- Adaptation to viewing modes: The transition between normal and VR modes introduces a challenge in adapting to different depth perceptions and control sensitivities. The realistic view of VR can affect the user's understanding regarding the environment and control precision differently than a flat screen.
- Time comparison: Time taken to complete the task in both normal and VR modes will be measured and compared to assess efficiency and adaptation in each mode.

Below are photos illustrating Test 1. The first photo Figure 10 provides a general view of the test setup, while the second photos Figure 11 depict the robotic limbs in action during the test.

Test 2: Participants were tasked with using robotic limbs to pick up tours objects and place them inside a box, The task manipulation limbs are assumed to have limb suckers that attach objects to the limb upon collision or touch, the test begins with the robot limbs in their initial position. Similar to Test 1, time recording was automated using a Unity script that stopped the game upon successful completion of the task. This automated process ensured consistent and accurate time measurements, eliminating potential discrepancies that could arise from manual intervention. Challenges of Test 2 is as below:

- Precision in manipulation: Accurately picking up and placing circular objects inside the box requires fine control of the robotic limbs. This task tests the user's ability to execute precise movements and alignments.
- Adaptation to viewing modes: Transitioning between normal and VR modes tests the user's ability to adapt their manipulation strategy and environment awareness. The realistic VR environment might affect depth perception and interaction differently than viewing on a laptop screen.

  Time comparison: Time taken to complete the task in both normal and VR modes will be measured and compared to assess efficiency and adaptation in each mode.

Photos illustrating Test 2. The first photo Figure 12 provides a general view of the test setup, while the second photos Figure 13 depict the robotic limbs in action during the test.

Test 3: In this test participants navigate a virtual underwater environment to locate and repair a damaged pipe. The task involves identifying a damaged pipe from which bubbles arise, retrieving a repair cover plate from a nearby small submarine and

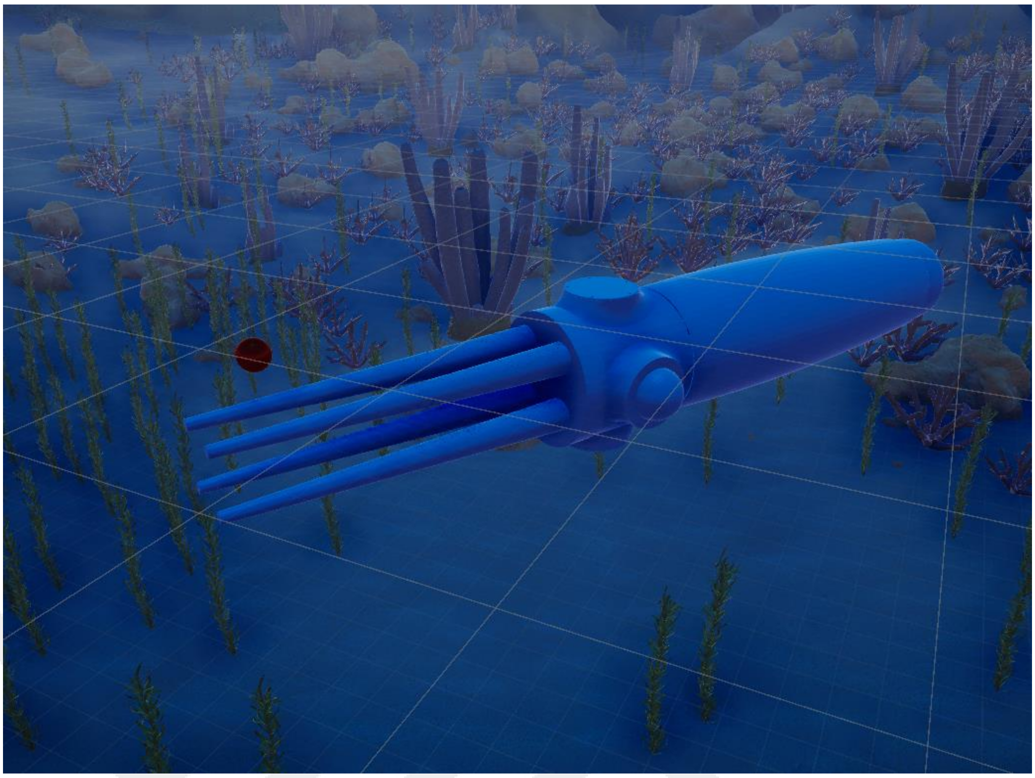

Figure 10. Unity test1 manipulating sphere general view

Figure 11. Unity test1 robot action

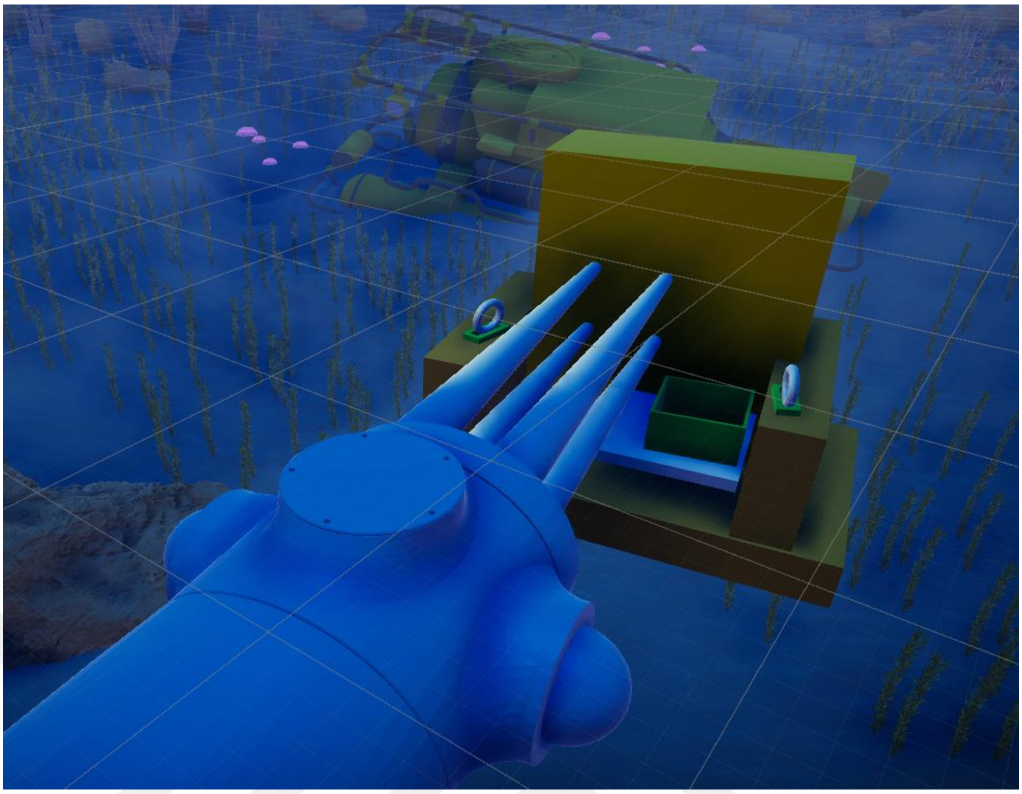

Figure 12. Unity test2 picking tours objects general view

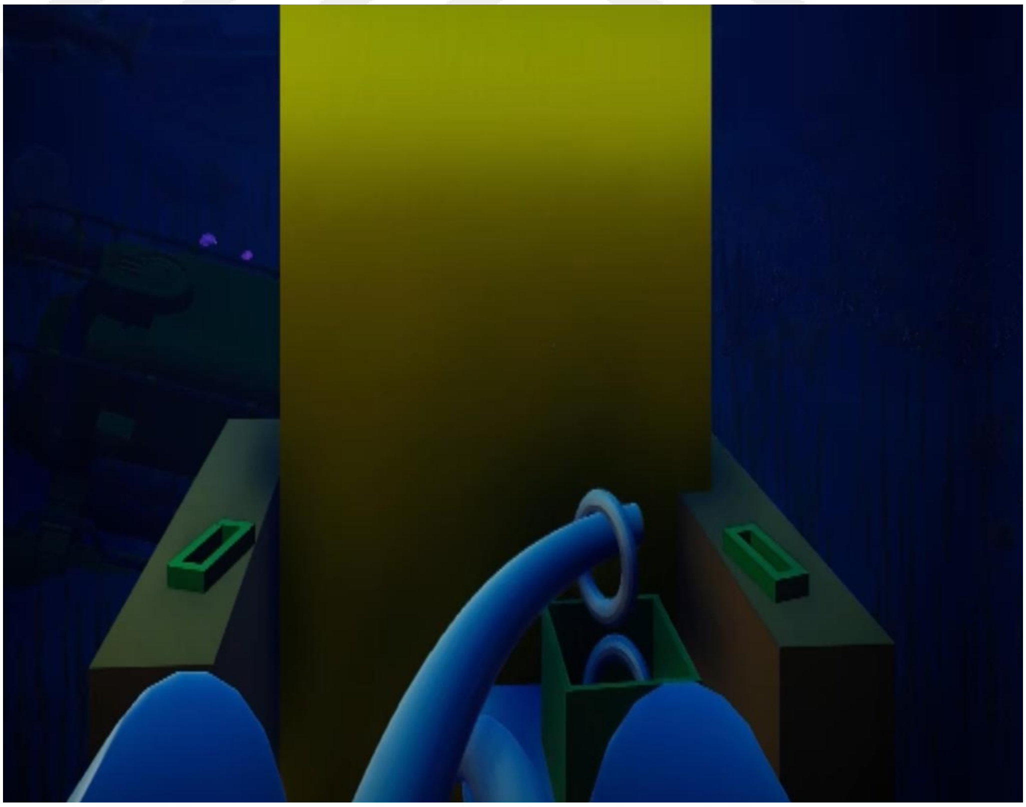

Figure 13. Unity test2 robot action

fixing the damaged spots on the pipe by placing the cover upon the leak position. The task manipulation limbs are assumed to have limb suckers that attach objects to the limb upon collision or touch, both task/manipulation and the light limbs, are utilized in this test, the test begin with the robot limbs in their initial position.

Time recording is integrated into the Unity environment through a script that terminates the game once the repair cover is placed over the leakage point on the pipe. This automated stop mechanism facilitates precise time tracking and ensures that data collection remains synchronized with participants' interactions in the virtual environment. Challenges of Test 3 is as below:

- Navigation in complete darkness: Utilizing the light-equipped limb to navigate and explore a completely dark underwater environment to locate the damaged pipe requires spatial orientation skills and effective use of the lighting to identify Pipe place and the source of bubbles indicating damage.
- Identification and precision in repair: Identifying the exact locations of damage based on bubble streams and then precisely controlling the robotic limbs to cover these with repair plate’s tests both observation skills and the precision of limb control.
- Coordination of Multiple Limbs: Employing the task limbs to grasp and manipulate repair plates while navigating and illuminating the environment with the light-equipped limb challenges the user's ability to coordinate multiple limbs with distinct functions.
- Adaptation to viewing modes: Performing the task in both normal and VR modes examine the user's ability to adapt their control strategies and spatial understanding to different sensory feedback mechanisms, with VR likely providing a more immersive experience.

Below are photos illustrating key stages of Test 3, where participants navigate a virtual underwater environment to locate and repair a damaged pipe. In Figure 14, The first photo “Action-1” depicts the robotic limbs navigating the environment to identify potential leakage positions on the pipe. Figure 15, In the second photo “Action-2”, the robot is seen attempting to retrieve a cover from a nearby submarine to use for repairing the identified leakage spots. Figure 16, The third photo “Action-3” showcases the successful placement of the cover by the robot over the leaking area, completing the repair process.

Figure 14. Unity test3 underwater robot Action-1

3.3.1.2 Questionnaire. The questionnaire utilized in this study served to gather qualitative feedback from participants regarding their experience and perceptions while interacting with the environment in both modes and controlling the soft robotic limbs. This section outlines the development process of the questionnaire, including considerations for validity and reliability.

The questionnaire was thoughtfully crafted to evaluate various facets of participants' experiences, considering factors such as mental/sensory effort, task difficulty, ease of use and control, level of engagement, performance, dizziness level, stress level and overall satisfaction with the system for both modes (normal mode and VR mode). Inspired by NASA Task Load Index guidelines (NASA Ames Research Center, 1986), the questionnaire featured structured questions designed to measure participants' perceptions across different dimensions. However, to ensure simplicity and ease of response, the answer scale was adjusted from 1 to 5, rather than the traditional 1 to 21 scale used in the NASA Task Load Index. Notably, physical exertion

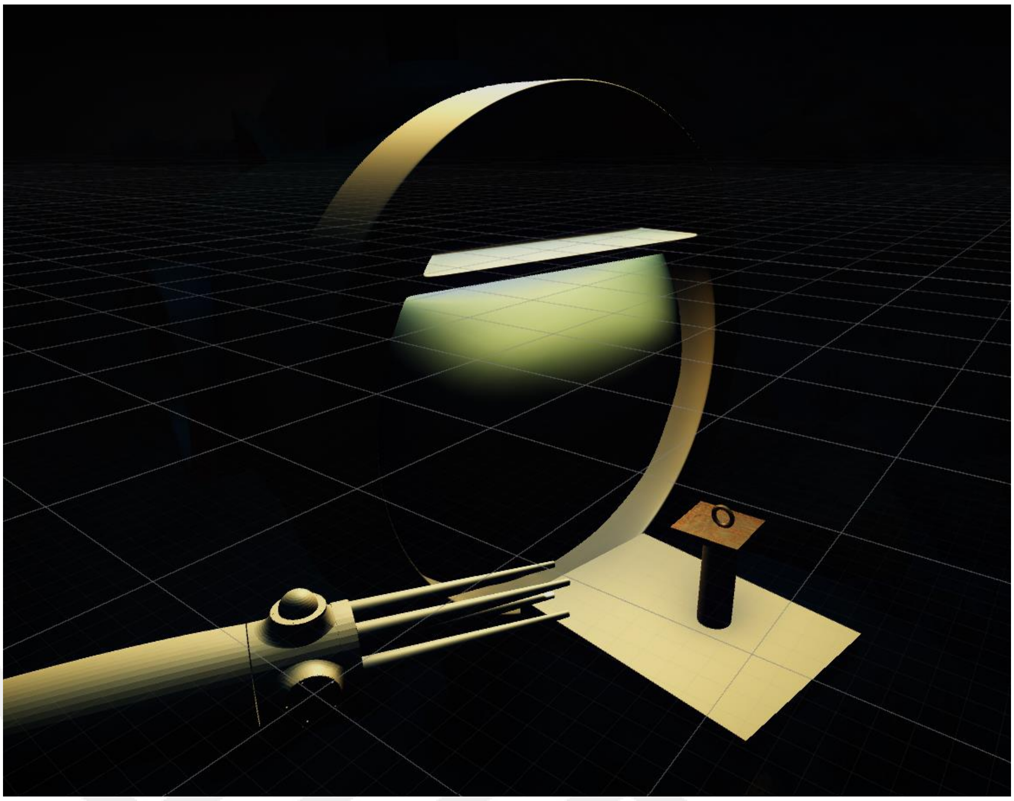

Figure 15. Unity test3 underwater robot action-2

Figure 16. Unity test3 underwater robot action-3

was not assessed in the questionnaire, as both modes of operation involved minimal physical effort; participants controlled the robotic limbs using a joystick while seated, eliminating the need for significant physical exertion.

Validity and reliability of the questionnaire were crucial to ensure the accuracy and consistency of the gathered data. Prior to implementation, the questionnaire underwent a validation process to assess its validity in measuring the intended constructs. This involved expert review by professionals in the fields of and virtual reality, who provided feedback on the clarity, relevance, and comprehensiveness of the questions.

In Appendix A, the provided questions on the questionnaire presented. The questionnaire is divided into two sections. The first is general information regarding age, gender and level of experience of using VR headset before. Second section focuses on answering questions related to user experience and performance evaluation for both normal mode, which corresponds to the control of the robot through standard laptop monitoring and to VR mode, which corresponds to the control of the robot through VR headset. The question numbers for this section range from 4 to 29, Except of questions 24 and 29 of the questionnaire which participants answered considering both modes. Every question was carefully designed with specific goals in order to gather comparison data between the normal Mode and VR Mode of operation. The questions were grouped in pairs, with one question focusing on the normal Mode and the next question targeting the VR Mode, except for question 24 and question 29. Question 24 was a straightforward inquiry that asked participants to state their choice between the two modes, providing valuable information on their preferred style of operation. Question 29 was a free-response question, enabling participants to provide whatever thoughts they had about their experience. Scores were assigned based on specific tasks that participants were required to complete. Some scores were interpreted such that higher values indicated better performance (e.g., ease of use and engagement), while other scores were interpreted in reverse (e.g., levels of dizziness or difficulty). Refer to Appendix A for full questions and score explanations.

3.3.2 Data collection procedures. The 12 participants for the study were selected based on specific criteria outlined in the Participant characteristics and selection criteria section (3.2 Universe and Participants/Working Group). This ensured a diverse representation across gender, racial/ethnic backgrounds, and expertise levels in engineering and gaming.

Every participant was given a comprehensive follow chart that explained the functioning of the whole system and visual demonstration that explains how to use the controller. Subsequently, each participant is provided with a familiarization time on the deployed system using a standard monitoring screen. The familiarization duration ranged from a minimum of 4 minutes to a maximum of 10 minutes for each test scenario. Subsequently, the participants were presented with three scenarios to examine. The experiments were done using two distinct modes: one included the use of a regular monitoring screen, while the other used the VR headset. Every participant successfully completed a total of six experiments and a period of familiarization.

The total of 12 participants were divided into two groups to compare the effectiveness of Standard Mode and VR Mode. The first group, consisting of seven participants, initiated the tests with the Standard Mode (normal monitoring screen) and then continued with VR Mode. The remaining participants started the tests with the VR Mode (Quest 2 VR headset) before continuing with the Standard Mode. This counterbalanced design was implemented to minimize the learning effect and ensure that the order of experiencing each mode did not bias the results.

The task scenario simulates underwater manipulation scenarios where operators control robotic limbs to perform various tasks as explained in the previous section, such as touching objects, manipulating objects, and repairing damaged pipes. Operators switch between different limbs, each with specific functions, including task execution and illumination. These scenarios test the system's capability to navigate dynamic underwater environments, demonstrating the versatility and effectiveness of the teleoperation system in real-life underwater applications. Participants were given a questionnaire through Microsoft forums QR code to complete after completing the test scenarios. The questionnaire aimed to gather feedback on participants' experiences with both the Standard and VR modes of operation, assessing usability and overall satisfaction with the system. Also following the completion of each test scenario, Unity automatically generated trajectory data containing the duration of each test and recorded the tip position data throughout the entire test. This automated data collection process ensured precise measurement of the time taken for each test and provided detailed positional data for analysis.

3.3.3 Data analysis procedures. The analysis of results in this study was methodically structured to evaluate the effectiveness of teleoperation in normal Mode versus VR Mode. The following step-by-step procedures were adhered to for a comprehensive analysis of the gathered data:

- Trajectory data analysis: A MATLAB was utilized to process the raw data on the trajectory of the robotic limb's movements. This included calculating the path length and the average speed (distance/time) for each manipulation limb (right and left limbs) in both modes for each test. For each participant, path lengths and average speed values were compared between normal and VR modes to identify any statistically significant differences. MATLAB's robust visualization tools were employed to create graphs that illustrated the comparative data, providing a clear visual representation of the findings.
- Questionnaire data analysis: All questionnaire responses were compiled in an Excel spreadsheet. This included Likert scale ratings (Likert, 1932) for each question provided by each participant. The average (mean) ratings for each question were calculated to determine the central tendency, while the standard deviation was computed to assess the variability in participants' responses.

The criteria for a successful outcome were predefined as a general preference for VR mode over normal mode. This was assessed by higher average ratings for questions related to ease of use, engagement, and performance satisfaction in VR mode.

Evaluation criteria:

1- User experience: Responses were evaluated based on the expectation that VR Mode would be favored, as indicated by higher average scores for questions related to telepresence, ease of use, control, and engagement

2- Task efficiency: The completion times in the VR Mode were anticipated to be less than those in the normal mode, indicating higher efficiency in the VR mode. This criterion assessed how quickly participants could complete tasks using VR compared to the traditional screen-based approach

3- Speed performance: The average speed of movement during tasks in VR mode was expected to be higher, reflecting the improved responsiveness and control purported by VR technology. These criteria were predefined to determine the success of VR mode over normal mode.

3.3.4 Reliability and validity. In the context of the VR simulation for the underwater squid soft robot, reliability refers to the consistency and stability of the simulation results over time. To ensure reliability:

- System Stability: The hardware and software systems, including the laptops and VR headset, were tested for performance stability over extended periods to ensure that there were no crashes or glitches that could affect the simulation outcome.
- Message Integrity Verification: The accuracy and efficiency of message transmission across ROS subjects were confirmed through systematic verification. This approach involved using the “rostopic” echo command to monitor and verify the messages sent through each topic.

The accuracy and reliability of the released data were verified by carefully comparing the received messages with the expected values, guaranteeing no loss or corruption of data during communication. Validity involves the accuracy and reliability of the simulation in representing real-world occurrences. For validity, the VR simulation of the squid robot was designed to accurately replicate the real-world and biological movements of actual squids. This was achieved by cross-verifying the matrix data received by Unity from ROS against the expected outcomes, ensuring that the simulation precisely followed the inputs from the "Limb_Dynamic_Translator" script. This rigorous data verification process guaranteed the simulation's accuracy and reliability in mimicking the desired movements. Through systematic testing and cross-verification, the reliability and validity of the VR simulation system have been established. This ensures that the study's findings are robust and can be confidently attributed to the designed interactions and behaviours within the simulated environment.

3.4 Limitations

Simulation Dependency: Before implementing the system on the actual robot, we wanted to test its effectiveness, so we conducted tests in a simulation environment. This is a validation test for the technology, and to ensure practicality within a finite amount of time, we involved 10 participants. Conducting these tests in a simulation is a typical and standard procedure in robotics.

Chapter 4

Results

This section presents a detailed analysis of the data collected through three experimental tasks utilizing robotic limbs in both normal and virtual reality modes. The findings are organized into three sections: 1) participant feedback from questionnaires, 2) time durations for each task and 3) trajectory data analysis of the robotic limbs tip position.

4.1 Participant Feedback from Questionnaires

All participants were successful at finishing the given tasks. The questionnaire answers provided by the subjects reflect their individual experiences and preferences. Therefore, the results are investigated and analysed separately. In Appendix A, the questionnaire questions are presented. As explained previously the questionnaire is divided into two sections. The first section is general information regarding age, gender, and level of experience regarding using VR headset before, while the second section focuses on answering questions related to user experience and performance evaluation for both modes.

The questionnaire was designed to evaluate participant feedback for two separate methods of operating a robotic system: normal mode, which includes operating the robot using a regular laptop screen, and VR mode, which includes operating the robot using a virtual reality headset. The total number of participants was 12, and the questionnaire had 29 questions specifically designed to evaluate different elements of the teleoperation experience. In Table 3, the questionnaire answers and analysis are presented.

The answers to (Q1) are the age numerical values, gender is noted as Male (M) or Female (F) for Q2. Experience on VR environments is rated from 1 for 'None' to 5 for 'Experienced' for Q3. For questions Q4 to Q27, where even-numbered questions pertain to normal Mode and odd-numbered questions to VR Mode. Q4 and Q5 Telepresence is rated from 1 for 'Not realistic at all' to 5 for 'Very realistic', Q6 and Q7 Mental/Sensory Effort range from 1 for 'Very low' to 5 for 'Very high', Q8 to Q13 Task difficulty is scaled from 1 for 'Very easy' to 5 for 'Very hard', Q14 and Q15 Ease of use and control is rated from 1 for 'Very imprecise' to 5 for 'Very precise', Q16 and Q17 Level of Engagement from 1 for 'Very Low' to 5 for 'Very high', Q18 to Q23 Performance is scaled from 1 for 'Very dissatisfied' to 5 for 'Very satisfied', Q24 Preferences are indicated from 1 for 'Strongly prefer normal mode' to 5 for 'Strongly prefer VR mode', Q25 and Q26 Dizziness level is rated from 1 for 'None' to 5 for 'Severe', Q27 and Q28: Stress levels range from 1 for 'Very stressed' to 5 for 'Very calm', Q29 Comments are open-ended for additional qualitative feedback.

The questionnaire used a scale based on Likert factors (Likert, 1932), that maintained a uniform measurement across all questions. The scale's minimum and maximum values were contextually linked to certain aspects of the user experience. A score of 1 represents a negative characteristic, while a score of 5 represents a good attribute, such as 'Very realistic' or 'Very calm', in terms of realism, accuracy, engagement, satisfaction, and stress. A lower score on the task difficulty scale corresponds to a more user-friendly experience. Q24 individually identifies preferences for the mode of operation, differentiating between a higher tendency towards either normal or VR mode. The questionnaire design incorporates alternating between normal and VR modes for the questions, enabling a comparative study between the two modes. After examining the questionnaire answers, it seems that the data provided by participants 8 and 12 may not adequately represent the specific experiences that were meant to be recorded in the normal and VR modes. The argument is supported by two observations: firstly, there is a significant consistency in their replies in both modes, indicating a possible lack of understanding of the unique characteristics that each mode provides. Furthermore, while the tasks were completed, the questionnaire scores remained similar across both modes, which is opposed to the expected difference caused by the different displays. Therefore, the study is continued using data from the remaining ten participants, which will guarantee a more reliable evaluation of the system's usability and user experience. Focused analysis was conducted on the accepted responses. The average and standard deviation for each question in this section were calculated to provide a statistical comparison of the remaining participants' responses, thereby ensuring a more accurate assessment of the system's usability and the user experience.

Table 3

Questionnaire Answers and Analysis

Questions	Participant Numbers											Excluded Data	
	P1	P2	P3	P4	P5	P6	P7	P9	P10	P11	Avg.	Std. Dev.	P8	P12
Q1 Age	20	25	29	20	20	21	21	20	20	22	21,8	2,97	19	25
Q2 Gender	M	M	F	F	M	F	M	F	F	M	-	-	M	F
Q3 Experience	4	2	1	1	4	2	3	2	4	3	2,60	1,17	3	2
Q4 Telepresence NM	3	3	2	2	2	2	3	3	3	3	2,60	1,17	3	3
Q5 Telepresence VR	5	5	4	3	4	5	5	5	4	4	4,40	0,52	3	3
Q6 Mental Effort NM	4	3	4	3	4	5	4	4	3	3	3,70	0,70	4	3
Q7 Mental Effort VR	3	3	3	2	3	3	2	3	3	2	2,70	0,67	4	3
Q8 Task1 Difficulty NM	3	3	4	4	3	3	3	4	4	4	3,50	0,48	3	2
Q9 Task1 Difficulty VR	2	1	1	2	2	2	1	2	3	2	1,80	0,53	3	2
Q10 Task2 Difficulty NM	4	5	4	3	4	5	5	4	3	5	4,20	0,63	2	3
Q11 Task2 Difficulty VR	3	2	1	2	2	2	3	2	2	3	2,20	0,79	2	3
Q12 Task3 Difficulty NM	3	4	3	2	3	3	3	3	4	3	3,10	0,63	1	1
Q13 Task3 Difficulty VR	2	1	1	1	2	1	1	2	3	2	1,60	0,57	1	1
Q14 Ease of use NM	4	2	2	4	2	2	2	3	2	2	2,50	0,70	3	2
Q15 Ease of use VR	5	4	4	5	3	5	5	5	4	4	4,40	0,85	3	2
Q16 Level of Engagement NM	3	2	3	4	3	3	3	3	2	3	2,90	0,70	5	4
Q17 Level of Engagement VR	4	4	5	4	4	5	5	5	3	5	4,40	0,57	5	4
Q18 Task1 Performance NM	3	2	4	3	4	3	2	4	3	2	3,00	0,70	1	2
Q19 Task1 Performance VR	4	4	5	5	5	4	5	5	4	5	4,60	0,82	1	2
Q20 Task2 Performance NM	2	1	2	3	3	2	3	3	4	3	2,60	0,52	2	1
Q21 Task2 Performance VR	3	4	5	5	4	5	5	5	4	4	4,40	0,84	2	1
Q22 Task3 Performance NM	4	2	3	4	3	3	3	4	3	2	3,10	0,70	1	1
Q23 Task3 Performance VR	5	5	5	4	4	5	5	5	4	4	4,60	0,74	1	1
Q24 Preference	3	5	4	5	5	5	5	4	5	4	4,50	0,52	3	2
Q25 Dizziness Level NM	2	1	2	3	2	2	2	2	2	3	2,10	0,71	1	1
Q26 Dizziness Level VR	4	3	3	3	2	3	4	4	3	4	3,30	0,57	1	1
Q27 Stress levels NM	3	2	3	3	3	2	3	4	2	3	2,80	0,67	5	4
Q28 Stress levels VR	4	2	3	4	4	4	4	3	3	4	3,50	0,63	5	4
Q29 Comments	No Comments Provided from Participant's													

4.2 Time Duration Records

The time taken to complete each task was measured in both normal and VR modes to evaluate efficiency and the effect of viewing modes on performance. In Table 4, times across normal and VR modes for each participant are presented.

Table 4

Task Completion Times Across Normal and VR Modes for Each Participant

Participant Numbers	Test 1 Duration Time (s)		Test 2 Duration Time (s)		Test 3 Duration Time (s)	
	Normal Mode	VR Mode	Normal Mode	VR Mode	Normal Mode	VR Mode
P1	95,03	89,03	107,43	66,27	169,33	117,93
P3	145,03	80,03	102,73	73,57	199,73	134,8
P5	161,03	121,03	129,46	94,33	159,67	106,7
P7	103,03	84,03	130,67	113,3	193,23	131,13
P10	142,03	95,03	93,43	56,45	269,17	137,17
Sub-1 Average	129,23	93,83	112,744	80,784	198,226	125,546
Sub-1 S.Deviation	28,64	16,21	16,60	22,89	42,96	12,89
P2	138,03	119,03	109,57	86,1	165,4	123,33
P4	107,03	99,03	120,27	109,47	251,93	135,6
P6	170,03	113,03	104,38	97,1	190,03	117,67
P9	114,03	81,03	107,6	58,1	225,93	149,27
P11	129,03	74,03	109,03	99,2	163,17	101,27
Sub-2 Average	131,63	97,23	110,17	89,994	199,292	125,428
Sub-2 S.Deviation	24,68	19,55	6,00	19,67	38,79	18,17
Total Average	130,43	95,53	111,46	85,39	198,76	125,49
Total S.Deviation	25,24	17,02	11,84	20,70	38,59	14,85

Participants 1, 3, 5, 7, and 10 started the series of testing starting with normal mode first, and then switched to VR mode. On the contrary, participants 2, 4, 6, 9, and 11 began their testing sequence by using VR mode, and then proceeded to have sessions in normal mode. The alternating sequence of mode initiation is crucial as it helps minimize any possible order effects that might impact performance as a result of familiarity or learning advancement. The table documents the time durations recorded for each participant in seconds throughout three separate tests conducted in both operating systems. Calculations are done as shown in (1) to calculate the percentage difference between Mode 1 and Mode 2 in each set of the test for each subject. The results are presented in Table 5.

$$\% = \frac{time_{VRM}}{time_{NM}} 100 \quad (1)$$

In order to provide an easy-to-understand summary of the relative efficacy of the two modes, we calculated the mean and standard deviation of the percentage differences. These statistical measurements provide a summary of the average and range of performance changes noticed while switching between modes.

Table 5

Task Completion Time % Between Normal and VR Modes

Participant Numbers	Test 1 Duration Time (s)	Test 2 Duration Time (s)	Test 3 Duration Time (s)
P1	93,7%	61,7%	69,6%
P3	55,2%	71,6%	67,5%
P5	75,2%	72,9%	66,8%
P7	81,6%	86,7%	67,9%
P10	66,9%	60,4%	51,0%
Sub-1 Average	74,5%	70,7%	64,6%
Sub-1 S. Deviation	14,6%	10,6%	7,7%
P2	86,2%	78,6%	74,6%
P4	92,5%	91,0%	53,8%
P6	66,5%	93,0%	61,9%
P9	71,1%	54,0%	66,1%
P11	57,4%	91,0%	62,1%
Sub-2 Average	74,7%	81,5%	63,7%
Sub-2 S. Deviation	14,4%	16,4%	7,5%
Total Average	74,6%	76,1%	64,1%
Total S. Deviation	13,7%	14,2%	7,2%

4.3 Position Data Records

This section examines the recorded trajectory data for the tip positions of the robotic manipulation limbs during the first two tests. The reason for restricting trajectory analysis to these tests is because the robot's body remains stationary throughout these activities. This assures that just the data on limb movements are recorded, without the interference of the robot's overall movement.

Test 1 and Test 2, consisted of maintaining the robot's body stationary in order to accurately measure the motions of the robotic limb. The gathered trajectory data encompasses the distance covered by the tip of the limb, providing valuable insights into the effectiveness and accuracy of participant control in both normal and VR modes.

Test 3 was omitted from trajectory data analysis due to the robot's extensive mobility in a simulated underwater environment. The robot's base mobility may introduce extra factors that might impact distance measures, making it difficult to identify the data specifically related to limb manipulation. Thus, by concentrating on fixed-base activities, one may get a more precise evaluation of the participants' ability to control the limb motions, without the interruption caused by moving the robot's location.

In Table 6, path length trajectory data for both manipulation limbs in both normal and VR modes for each participant are presented. Furthermore, we made use of the time values obtained from Section 4.2 (Table 5) to compute the velocity of the robotic limb motions, in addition to the length of the trajectory data provided. Average speed, measured in centimeters per second (cm/s), adds an additional aspect of analysis, providing a deeper understanding of how quickly and smoothly participants are able to use the robotic limbs in both modes.

The average speed numbers are calculated by dividing the route length values by the respective time required to finish each activity. The average speed calculation provides a thorough comprehension of how the mode affects not only the accuracy and precision of motions, but also their velocity. Table 7 displays the computed average speed values for each participant in both the normal and VR modes during Test 1 and Test 2.

Table 6

Path Length Trajectories for Manipulation Limbs in Normal and VR Modes

Participant Numbers	Test 1 Path Length (cm)				Test 2 Path Length (cm)			
	Normal Mode RA	VR Mode RA	Normal Mode LA	VR Mode LA	Normal Mode RA	VR Mode RA	Normal Mode LA	VR Mode LA
P1	38,91	51,55	69,93	73,17	49,72	80,60	64,04	66,0
P3	97,39	51,41	86,62	76,73	72,64	83,81	62,67	76,12
P5	103,80	93,49	75,15	77,25	58,49	92,43	74,61	64,76
P7	69,85	63,39	59,64	42,13	74,88	122,85	71,67	70,14
P10	82,94	72,80	74,16	46,09	56,80	35,19	71,19	76,45
Sub-1 Average	78,58	66,53	73,10	63,07	62,51	82,98	68,84	70,69
Sub-1S. Deviation	25,78	17,53	9,74	17,44	10,82	31,50	5,19	5,49
P2	83,39	91,33	55,36	135,97	73,77	100,65	66,32	70,99
P4	58,78	78,47	50,07	42,34	83,54	89,83	74,73	76,53
P6	179,20	97,06	57,19	90,86	68,47	96,85	67,77	88,30
P9	76,79	59,52	44,29	46,13	75,65	71,93	71,37	84,31
P11	63,11	60,04	72,53	66,57	60,70	75,03	70,05	79,62
Sub-2 Average	92,25	77,29	55,89	76,37	72,43	86,86	70,05	79,95
Sub-2 S.Deviation	49,62	17,34	10,57	38,51	8,49	12,86	3,27	6,72
Average	85,42	71,91	64,49	69,72	67,47	84,92	69,44	75,32
S.Deviation	37,97	17,39	13,20	29,04	10,55	22,77	4,14	7,57

In Table 6, the path length trajectories are presented for manipulation limbs in both normal and VR modes for each participant. For Test 1, where participants could use either the left or right manipulation limb to interact with the sphere, the individual path lengths for each limb are shown. This method recognizes the flexibility in using either or both limbs for the task and provides a detailed view of the user's performance in both normal and VR modes. For Test 2, where each limb has distinct duties interacting with different objects, the path lengths for each limb is presented separately. This approach allows for a more detailed analysis of each limb’s performance, reflecting each limb's ability to handle specific tasks with precision and control in both normal and VR modes. This differentiation enables a more comprehensive examination of the robotic limbs' ability to manipulate with high accuracy and control in both normal and VR modes, even when faced with different situations. Moreover, accurately determining the velocity of the robotic limb motions is essential for a reliable comparison between normal and VR modes. Due to variation in path lengths across participants, a simple comparison of distances travelled may not provide an accurate representation of movement efficiency. For example, one person may choose a little longer route but yet finish the assignment in less time than another, indicating a faster velocity and perhaps higher effectiveness. By assessing velocity, measured in centimetres per second, we get a more uniform measurement that takes into consideration the differences in distance and time. Average Speed analysis allows us to establish whether mode allows for quicker movement, offering a more accurate evaluation of performance in various viewing situations. The average speed figures calculated and shown in Table 7 provide a precise comparison to determine the advantage of one mode over the other in terms of task execution speed.

Table 7

Average *Speed Trajectories for Manipulation Limbs in Normal and VR Modes*

Participant Numbers	Test 1 Average Speed (cm/s)				Test 2 Average Speed (cm/s)			
	Normal Mode RA	VR Mode RA	Normal Mode LA	VR Mode LA	Normal Mode RA	VR Mode RA	Normal Mode LA	VR Mode LA
P1	0,41	0,579	0,74	0,82	0,46	1,22	0,60	1,00
P3	0,67	0,642	0,60	0,96	0,71	1,14	0,61	1,03
P5	0,64	0,772	0,47	0,64	0,45	0,98	0,58	0,69
P7	0,68	0,754	0,58	0,50	0,57	1,08	0,55	0,62
P10	0,58	0,766	0,52	0,48	0,61	0,62	0,76	1,35
Sub-1 Average	0,60	0,70	0,58	0,68	0,56	1,01	0,62	0,94
Sub-1 S.Deviation	0,11	0,09	0,10	0,21	0,11	0,23	0,08	0,30
P2	0,604	0,77	0,40	1,14	0,67	1,17	0,61	0,82
P4	0,549	0,79	0,47	0,43	0,69	0,82	0,62	0,70
P6	1,054	0,86	0,34	0,80	0,66	1,00	0,65	0,91
P9	0,673	0,73	0,39	0,57	0,70	1,24	0,66	1,45
P11	0,489	0,81	0,56	0,90	0,56	0,76	0,64	0,80
Sub-2 Average	0,67	0,79	0,43	0,77	0,66	1,00	0,64	0,94
Sub-2 S.Deviation	0,22	0,05	0,09	0,28	0,06	0,21	0,02	0,30
Average	0,64	0,75	0,51	0,72	0,61	1,00	0,63	0,94
S.Deviation	0,17	0,08	0,12	0,24	0,10	0,21	0,06	0,28

Chapter 5

Discussions and Conclusions

This section of the thesis presents a thorough examination of the research findings, establishing clear connections between the study's goals, applied methodology, and obtained outcomes. Furthermore, it provides clear results that correspond to the study questions and hypotheses stated previously. Moreover, this section defines future possibilities for study, presenting strategies for overcoming current challenges and proposing areas for additional investigation to improve the efficacy and usability of VR-enhanced teleoperation systems. This procedure guarantees the right analysis and integration of the data, allowing continuous contributions to this constantly developing area of research.

5.1 Discussion of Findings for Research Questions

The analysis of the results is offered in connection with the research questions that are raised at the beginning of the study. This systematic assessment takes into account the subtleties of participant feedback, the importance of time duration records, and the impact of location data records, thereby providing a thorough understanding of the study's results.

5.1.1 Discussion of findings regarding questionnaires. As previously discussed, the results of questionnaire answers given in Table 3 are analysed as below.

- Age (Q1): The average age of participants was 21.80 with a standard deviation of 2.97, indicating a sample group with a range of adult ages but centered around the early twenties. This age demographic is important as it may reflect individuals who are more likely to be familiar with VR technologies, potentially influencing their adaptability and comfort level with the system.
- Gender (Q2): The participant group consisted of five males and five females, which is a balanced gender distribution.

- Previous VR experience (Q3): The average score for previous VR experience is relatively low, at 2,60 with a standard deviation of 1,17, suggesting that most participants had limited exposure to VR environments prior to the study.
- Telepresence (Q4 and Q5): Participants found the telepresence in VR mode to be more realistic than in normal mode, with average scores of 4,4 (SD = 0,52) in VR mode compared to 2,6 (SD = 1,17) in normal mode. This confirms the hypothesis that VR's realistic environment enhances the user's sense of being present within a remote setting.
- Mental/Sensory effort (Q6 and Q7): The mental and sensory effort reported by participants was higher in normal mode, with averages of 3,7 (SD = 0,7) compared to 2,7 (SD = 0,67) in VR mode. This suggests that VR technology may be more intuitive and less mentally taxing.
- Task difficulty (Q8-Q13): Participants reported more difficulty in completing the activities in normal mode compared to VR mode. This is evident from the higher average scores in normal mode for Test 1 (Q8: average = 3,5 and SD = 0,48), Test 2 (Q10: average = 4,2 and SD = 0,63), and Test 3 (Q12: average = 3,1 and SD = 0,63). In contrast, the average scores in VR mode were lower for Test 1 (Q9: average = 1,8 and SD = 0,53), Test 2 (Q11: average = 2,2 and SD = 0,79), and Test 3 (Q13: average = 1,6 an SD = 0,57). The reduced difficulty in VR mode may be attributed to the accurate feedback and natural engagement that VR offers, potentially reducing the mental effort required for visual reasoning and task completion.
- Ease of Use and Control (Q14 and Q15): The mean values in VR mode are higher, indicating an enhancement in accuracy enabled by VR technology. In normal mode, the average score was 2,5 with a standard deviation of 0,7 compared to an average score of 4,4 and a standard deviation of 0,85 in VR mode. This consistency with the deep and interactive characteristics of VR allows for more precise control in teleoperation activities.
- Level of engagement (Q16 and Q17): The VR mode resulted in a higher level of engagement, with an average score of 2,9 and a standard deviation of 0.7 in normal mode, compared to an average score of 4.4 and a standard deviation of 0.57 in VR mode. Most likely because of its realistic characteristics. This is essential for tasks

that need continuous attention and might have favourable consequences for user satisfaction and task execution.

- Performance satisfaction (Q18-Q23): The averages for performance satisfaction are consistently higher in VR mode, indicating a preference for VR's interactive feedback mechanisms. In normal mode, the average scores are lower for Test 1 (Q18: average = 3.0, SD = 0.7), Test 2 (Q20: average = 2.6, SD = 0.52), and Test 3 (Q22: average = 3.1, SD = 0.7), compared to VR mode. In VR mode, the average scores are higher for Test 1 (Q19: average = 4.6, SD = 0.82), Test 2 (Q21: average = 4.4, SD = 0.84), and Test 3 (Q23: average = 4.6, SD = 0.74). The reduced difficulty in VR mode may be attributed to the feedback and natural engagement that VR offers, potentially reducing the mental effort required for visual reasoning and task completion.
- Mode preference (Q24): There is a clear preference for VR Mode, with average scores of 4,5 (SD = 0,52) underscoring its potential for enhancing teleoperation experiences despite the challenges it may present.
- Dizziness levels (Q25 and Q26): According to the responses, dizziness levels are indeed higher in VR mode, which aligns with common concerns regarding VR usage. The immersion and motion within a 3D environment can sometimes lead to a dissonance in the user's sensory perception, known to cause discomfort. In normal mode, the average score was 2.1 (SD = 0.71), compared to an average score of 3.3 (SD = 0.57) in VR mode.
- Stress levels (Q27 and Q28): The stress levels recorded in VR mode are somewhat lower than those in normal mode. This might be attributed to the fully immersive feeling that is more engaging, helping to relax feelings of tension while performing tasks. In normal mode, the average score was 2.8 (SD = 0.67), compared to an average score of 3.5 (SD = 0.63) in VR mode.
- Commentary (Q29): There were no comments provided by participants in response to Question 29, which was designed as an open-ended question to collect qualitative feedback.

Standard deviations in the responses serve as a measure of agreement or variability among participants. For instance, a higher standard deviation in responses to questions regarding telepresence (Q4 and Q5) would suggest a varied interpretation of what constitutes a 'realistic' experience among users, which could stem from personal expectations of telepresence. On the other hand, lower standard deviations indicate more consensus among the participants, which can be indicative of a shared user experience that may transcend individual differences.

Looking across all the questions, varying levels of standard deviation reveal that certain aspects of the teleoperation experience are more uniformly experienced than others. For example, a lower standard deviation in questions about engagement (Q16 and Q17) suggests that regardless of mode, there is a general agreement on how engaging the task was. Conversely, a higher standard deviation in questions about dizziness (Q25 and Q26) in VR mode suggests that some participants may be more susceptible to VR-induced motion sickness than others.

The analysis of the questionnaires reveals a preference for the VR mode in, suggesting that it enhances realism and engagement, and reduce task difficulty. Responses indicated VR demands less mental effort. Notably, VR mode induced more dizziness, aligning with common VR challenges, yet stress levels were lower, pointing to the engaging quality of VR.

5.1.2 Discussion of findings regarding time duration records. Table 5 provides a comparative analysis of task completion times between normal mode and VR mode, expressed as percentages, to offer a quantitative perspective on performance efficiency across different tasks. The average and standard deviation for each subgroup were also calculated and analysed. Subgroup 1 includes participants 1, 3, 5, 7, and 10, who commenced testing with normal mode before switching to VR mode. Conversely, Subgroup 2 comprises participants 2, 4, 6, 9, and 11, who began their testing sequence with VR mode before transitioning to normal mode.

Task 1: The average completion time for Task 1 shows that participants generally performed the task more efficiently in VR mode, with an average of 74,6% completion time compared to normal mode. For Subgroup 1, the average completion time was 74,5% with a standard deviation of 14,6%, while for Subgroup 2, the average was 74,7% with a standard deviation of 14,4%. This suggests that VR may facilitate faster task completion.

Task 2: The results for Task 2 reveal a similar trend, with participants completing the task in VR mode at 76,1% of the time it took in normal mode on average. Subgroup

1 had an average completion time of 70,7% with a standard deviation of 10,6%, and Subgroup 2 had an average of 81,5% with a standard deviation of 16,4%. This consistency across different tasks reinforces the notion that VR technology might offer advantages in task efficiency.

Task 3: Interestingly, the average completion time percentage for Task 3 demonstrates a more pronounced difference, with tasks being completed in VR mode in only 64.1% of the time it took in normal mode. Subgroup 1 had an average of 64,6% with a standard deviation of 7,7%, while Subgroup 2 had an average of 63,7% with a standard deviation of 7,5%. This significant difference may reflect the added complexity of Task 3.

Participants 1, 3, 5, 7, and 10 (Subgroup 1) started the series of testing with normal mode first and then switched to VR mode. Conversely, participants 2, 4, 6, 9, and 11 (Subgroup 2) began their testing sequence by using VR mode first. The analysis of the two subgroups indicates that starting with VR mode does not significantly alter the overall efficiency gains provided by VR. Both subgroups showed a consistent trend of improved performance in VR mode across all tasks. The standard deviation across all three tests is somewhat variable, with the lowest in Test 3 (7,2%), indicating that participants had more consistent times in VR mode for this test. For Tests 1 and 2, the standard deviations are higher, suggesting a greater variability in how much VR Mode affected participant performance.

These findings suggest that VR has the potential to enhance performance for underwater robotics limb manipulation. The consistency across both subgroups further supports the robustness of VR's benefits in improving task efficiency and accuracy.

5.1.3 Discussion of findings regarding position data records. Table 6 and Table 7 offer a quantitative perspective on the path length and average speed of robotic limb movements in normal and VR Modes across two tests.

Test 1: Observations from Table 6 indicate that, on average, the path lengths are longer in VR Mode compared to Normal Mode. For Subgroup 1, the average path length in VR Mode is 66,53 cm (SD = 17,53) for the right arm (RA) and 63,07 cm (SD = 17,44) for the left arm (LA). Subgroup 2 shows an average path length of 77,29 cm (SD = 17,34) for RA and 76,37 cm (SD = 38,51) for LA in VR Mode. This could suggest that participants in VR Mode are exploring the space more or require additional corrections in movement. Notably, some individuals (P1, P3, P5, P10) travelled a shorter path in VR, which may indicate more efficient movements for certain users.

While, observations from Table 7 indicate that the average speed of movements generally increased in VR Mode for both limbs. For Subgroup 1, the average speed in VR Mode is 0.70 cm/s (SD = 0,09) for RA and 0,68 cm/s (SD = 0,21) for LA. Subgroup 2 shows an average speed of 0,79 cm/s (SD = 0,05) for RA and 0,77 cm/s (SD = 0,28) for LA in VR Mode. Despite the longer path lengths noted in Table 6, the increased average speed suggests that participants might be moving quicker in VR, reflecting higher confidence or responsiveness.

Test 2: For Subgroup 1, the average path length in VR Mode is 82,98 cm (SD = 31,50) for RA and 70,69 cm (SD = 5,49) for LA. Subgroup 2 shows an average path length of 86,86 cm (SD = 12,86) for RA and 79,95 cm (SD = 6,72) for LA in VR Mode. Some participants (P7, P10) had shorter path lengths in VR Mode, while others (P2, P4, P6) showed longer path lengths. These inconsistencies may be attributed to the different object manipulation tasks assigned to each limb and individual participant interaction with the VR environment.

From table 7 For Subgroup 1, the average speed in VR Mode is 1,01 cm/s (SD = 0,23) for RA and 0,94 cm/s (SD = 0,30) for LA. Subgroup 2 shows an average speed of 1,00 cm/s (SD = 0,21) for RA and 0,94 cm/s (SD = 0,30) for LA in VR Mode. Notably, P10 showed a significant increase in average speed for the left limb in VR Mode, possibly reflecting a high degree of adaptability or proficiency with the VR system for this specific task.

When comparing Subgroup 1 (participants 1, 3, 5, 7, and 10) and Subgroup 2 (participants 2, 4, 6, 9, and 11), it is observed that both groups exhibit improvements in VR Mode, but there are notable differences. For Test 1, Subgroup 1's average path length in VR Mode is shorter for the right arm (66,53 cm) compared to Subgroup 2 (77,29 cm), suggesting that Subgroup 1 may have achieved more efficient movements. In contrast, for Test 2, Subgroup 2 has a slightly higher average path length in VR Mode for both the right arm (86,86 cm) and left arm (79,95 cm) compared to Subgroup 1 (82,98 cm and 70,69 cm, respectively), indicating more exploration or corrections in movement. The average speeds in VR Mode are relatively consistent between the two subgroups, with both groups showing increased speed in VR Mode across tests, though Subgroup 2 exhibits slightly higher variability in speeds for certain tasks. These comparisons highlight how initial mode exposure may influence subsequent performance and adaptation to VR environments.

The findings from the position data analysis reveal nuanced insights into the teleoperation behaviour of participants in different modes. While VR appears to allow for faster movement, the increased path lengths suggest that efficiency in movement is not necessarily improved and could even be compromised. These observations highlight the complexity of the interaction between human operators and VR interfaces, suggesting a trade-off between average speed and precision.

The variability in participant performance, as evidenced by the differing path lengths and average speeds, underscores the importance of individual differences in teleoperation tasks.

This suggests that while VR can enhance certain aspects of the teleoperation experience, it does not do so uniformly for all users or tasks. Personal proficiency, adaptability to VR, and the specific nature of the task all appear to influence performance outcomes. Overall, the position data records suggest that VR technology has the potential to influence how tasks are performed, with implications for both average speed and efficiency. The data supports the notion that VR can enhance certain aspects of teleoperation but also highlights the need for a deeper understanding of how VR affects movement strategies. Future research should focus on identifying the factors that contribute to the observed variability in order to optimize VR teleoperation interfaces for a broad range of users and tasks.

5.2 Conclusions

This study corroborates the initial hypothesis that operators achieve better control and quicker task completion using the VR interface compared to the traditional two-dimensional (2D) screen. The integration of a VR-enhanced teleoperation system with advanced control systems within a virtual simulation headset has demonstrated a marked improvement in operational efficiency and accuracy. Virtual reality maintained superior performance by significantly enhancing the operator's engagement with the environment, increasing control accuracy, and promoting overall performance. The improvement is credited to the realistic quality of VR, which offers a more natural understanding of surrounding environment.

Our findings also highlight the system's ability to simulate real-world teleoperation effectively, thanks to the multi-master multi-slave configuration that leverages both actual control hardware and the virtual reality framework. The realistic characteristics of the VR headset, combined with the ergonomic design of the master device, have substantially improved the teleoperation experience. This is supported by structured questionnaires and analyses of recorded time and trajectory data, which show a consistent preference for the VR mode among participants.

However, while VR enhances the sense of presence and environmental awareness, critical for precision tasks in remote settings, it also introduces challenges such as increased dizziness among users. This indicates a need for further refinement in VR interface design to mitigate such drawbacks and enhance user comfort, ensuring that VR's potential in robotic teleoperation can be fully realized without compromising user safety and comfort.

5.3 Recommendations for Future Research

Based on the discussions and findings of this thesis, several recommendations for future research can be proposed:

1- Future research could explore the integration of VR headsets with haptic device controllers. This combination could provide a more natural and intuitive control scheme, offering tactile feedback that might further enhance operator performance.

2- While this study utilized a PS5 controller master device, future research could explore the integration of VR controllers, like those available with the VR headset, to provide a more natural and intuitive control scheme that might further enhance operator performance.

3- As increased levels of discomfort such as dizziness have been identified among users of VR interfaces, research should focus on designing VR experiences that minimize such adverse effects. This could involve ergonomic studies to adjust the VR interface and experimenting with different levels of immersion to find a balance that maximizes control efficiency while minimizing user discomfort

4- Another crucial area for future research is the enhancement of the machine learning models used to simulate the limb dynamics of the robotic limbs. Improvements could include expanding the datasets used for training these models to encompass a wider range of environmental conditions and manipulation scenarios.
5- While this study utilized a PS5 controller master device, future research could explore the integration of VR controllers, like those available with the VR headset, to provide a more natural and intuitive control scheme that might further enhance operator performance.

REFERENCES

Ahmed, F., Waqas, M., Shaikh, B., Khan, U., Soomro, A. M., Kumar, S., ... & Choi, K. H. (2022). Multi-material bio-inspired soft octopus robot for underwater synchronous swimming. Journal of Bionic Engineering, 19(5), 1229-1241.

Aldhaheri, S., De Masi, G., Pairet, È., & Ardón, P. (2022, February). Underwater robot manipulation: advances, challenges and prospective ventures. In OCEANS 2022-Chennai (pp. 1-7). IEEE.

Alemzadeh, H., Pradhan, S. K., Sandberg, W. S., & Goldberg, M. E. (2021). Virtual reality-enhanced teleoperation for robotic surgery. Journal of Robotic Surgery, 15(2), 157-169. doi:10.1007/s11701-020-01128-9

Alkhodary, A & Gur, B. (2023). Generative Model-based Control Strategies for Soft Robotic Arms.

Atkeson, C. G., Benzun, P. B., Banerjee, N., Berenson, D., Bove, C. P., Cui, X., ... & Xinjilefu, X. (2018). What happened at the DARPA robotics challenge finals. The DARPA robotics challenge finals: Humanoid robots to the rescue, 667-684.

Azizkhani, M., Godage, I. S., & Chen, Y. (2022). Dynamic control of soft robotic arm: A simulation study. IEEE Robotics and Automation Letters, 7(2), 3584-3591.

Cai, M., Wang, Y., Wang, S., Wang, R., & Tan, M. (2023). Autonomous manipulation of an underwater vehicle-manipulator system by a composite control scheme with disturbance estimation. IEEE Transactions on Automation Science and Engineering.

Chaudhary, A., Mishra, R., Kalyan, B., & Chitre, M. (2021, September). Development of an underwater simulator using unity3d and robot operating system. In OCEANS 2021: San Diego–Porto (pp. 1-7). IEEE.

Della Santina, C., Duriez, C., & Rus, D. (2023). Model-based control of soft robots: A survey of the state of the art and open challenges. IEEE Control Systems Magazine, 43(3), 30-65.

Fang, J., Zhuang, Y., Liu, K., Chen, Z., Liu, Z., Kong, T., ... & Qi, C. (2022). A shift from efficiency to adaptability: recent progress in biomimetic interactive soft robotics in wet environments. Advanced science, 9(8), 2104347.

Hudson, N., Talbot, F., Cox, M., Williams, J., Hines, T., Pitt, A., ... & Arkin, R. C. (2021). Heterogeneous ground and air platforms, homogeneous sensing: Team CSIRO Data61's approach to the DARPA subterranean challenge. arXiv preprint arXiv:2104.09053.

LeMasurier, G., Allspaw, J., & Yanco, H. A. (2021). Semi-autonomous planning and visualization in virtual reality. arXiv preprint arXiv:2104.11827.

Likert, R. (1932). A technique for the measurement of attitudes. Archives of Psychology, 140, 1-55.

Li, Y., Liu, Y., & Wu, Z. (2021). Advances in simulation technologies for underwater robotics. Simulation Modelling Practice and Theory, 109, 102263. doi:10.1016/j.simpat.2021.102263

Lin, H., Jiang, Z., & Li, X. (2021). Challenges in integrating virtual reality with underwater robotics. Journal of Marine Science and Engineering, 9(3), 318. doi:10.3390/jmse9030318

Ma, Y., Fan, Z., & Chen, J. (2022). Virtual reality-based teleoperation system for autonomous underwater vehicles. Ocean Technology, 11(2), 98-110. doi:10.1016/j.oceaneng.2022.103062

Meta. (2024). https://www.meta.com/quest/products/quest-2/.

Morgan, E., Carlucho, I., Ard, W., & Barbalata, C. (2022). Autonomous underwater manipulation: Current trends in dynamics, control, planning, perception, and future directions. Current Robotics Reports, 3(4), 187-198.

Naceri, A., Mazzanti, D., Bimbo, J., Tefera, Y. T., Prattichizzo, D., Caldwell, D. G., ... & Deshpande, N. (2021). The vicarios virtual reality interface for remote robotic teleoperation: Teleporting for intuitive tele-manipulation. Journal of Intelligent & Robotic Systems, 101, 1-16.

NASA Ames Research Center. (1986). NASA Task Load Index (TLX) v. 1.0 manual. Moffett Field, CA: NASA Ames Research Center. Retrieved from https://humansystems.arc.nasa.gov/groups/tlx/downloads/TLXScale.pdf.

Potokar, E., Ashford, S., Kaess, M., & Mangelson, J. G. (2022, May). HoloOcean: An underwater robotics simulator. In 2022 International Conference on Robotics and Automation (ICRA) (pp. 3040-3046). IEEE.

Rafael, V. M., Caceres, J., Abel, A. H., Montoya, A., Guevara, J., Jarelh, G. B., & Talavera, J. (2022). Development of a Low-Cost Teleoperated Explorer Robot (TXRob). International Journal of Advanced Computer Science and Applications, 13(7).

Renda, F., Cianchetti, M., Giorelli, M., Arienti, A., & Laschi, C. (2012). A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm. Bioinspiration & biomimetics, 7(2), 025006.

ROS Noetic. (2020). https://wiki.ros.org/noetic

ROS Joy library. (2022). https://wiki.ros.org/joy

Sony Interactive Entertainment. (2024). https://www.playstation.com/tr-tr/accessories/dualsense-wireless-controller/.

Su, Y. P., Chen, X. Q., Zhou, C., Pearson, L. H., Pretty, C. G., & Chase, J. G. (2023). Integrating Virtual, Mixed, and Augmented Reality into Remote Robotic

Applications: A Brief Review of Extended Reality-Enhanced Robotic Systems for Intuitive Telemanipulation and Telemanufacturing Tasks in Hazardous Conditions. Applied Sciences, 13(22), 12129.

Topini, E., Bucci, A., Gelli, J., Topini, A., Ridolfi, A., & Allotta, B. (2021). Advanced underwater manipulation systems: an overview of the suono project. OCEANS 2021: San Diego–Porto, 1-7.

Unity Technologies. (2024). https://unity.com/

Whitney, D., Rosen, E., Phillips, E., Konidaris, G., & Tellex, S. (2019, November). Comparing robot grasping teleoperation across desktop and virtual reality with ROS reality. In Robotics Research: The 18th International Symposium ISRR (pp. 335-350). Cham: Springer International Publishing.

Xu, S., Moore, S., & Cosgun, A. (2022, June). Shared-control robotic manipulation in virtual reality. In 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-6). IEEE.

Youssef, S. M., Soliman, M., Saleh, M. A., Mousa, M. A., Elsamanty, M., & Radwan, A. G. (2022). Underwater soft robotics: A review of bioinspiration in design, actuation, modeling, and control. Micromachines, 13(1), 110.

Zhang, X., Fan, Y., Liu, H., Zhang, Y., & Sha, Q. (2023). Design and implementation of autonomous underwater vehicle simulation system based on moos and unreal engine. Electronics, 12(14), 3107.