IMPLEMENTATION OF A REAL-TIME TELEOPERATION SYSTEM FOR THE CONTROL OF A ROBOTIC SQUID

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Mechanical Engineering

by
Hasan CEZAYİRLİ

July 2023
İZMİR

We approve the thesis of **Hasan CEZAYİRLİ**

Examining Committee Members:

Prof. Dr. Mehmet İsmet Can DEDE
Department of Mechanical Engineering, Izmir Institute of Technology

Assoc. Dr. Tolga AYAV
Department of Computer Engineering, Izmir Institute of Technology

Assoc. Dr. Mehmet Berke GÜR
Department of Mechatronics Engineering, Bahcesehir University

20 / July / 2023

Prof. Dr. Mehmet İsmet Can DEDE
Supervisor, Department of Mechanical Engineering, Izmir Institute of Technology

Prof. Dr. Mehmet İsmet Can DEDE
Head of the Department of Mechanical Engineering, Izmir Institute of Technology

Prof. Dr. Mehtap EANES
Dean of the Graduate School

ACKNOWLEDGMENTS

First and foremost, I wish to thank my supervisor Dr. Mehmet İsmet Can Dede for his guidance, patience, encouragement, and support. I am glad that I had the chance of him being my supervisor.

I would like to thank the rest of my thesis committee, Dr. Tolga Ayav and Dr. Mehmet Berke Gür for their insightful comments and suggestions.

I wish to thank Dr. Emre Uzunoğlu, without his help and guidance the study would not have finished on time.

I would like to express my sincere gratitude to my mother, Sema Cezayirli, and my sister Zeynep Cezayirli for their patience and unconditional support.

I would like to thank the members of the IRL, MPL, HURL, and RAML, for their help and contribution.

I would like to thank in no particular order, Tuğrul Yılmaz, Muhammed Rıza Bozelli, Mehmet Alp Balkan, Oğuz Güler, Esra Özdemir, İbrahimcan Görgülü, and Merve Özkahya for participating the experiments.

This study is funded by *The Scientific and Technological Research Council of Turkey (TÜBİTAK)* with the project name *Robotic Squid for Underwater Manipulation and Intervention* and grant number *216M219*.

ABSTRACT

IMPLEMENTATION OF A REAL-TIME TELEOPERATION SYSTEM FOR THE CONTROL OF A ROBOTIC SQUID

Teleoperation is defined as the remote control of a robotic system from an operational environment. Teleoperation of soft robots has been a growing research topic in recent years and there are still areas awaiting further studies.

In this study, a real-time teleoperation system has been implemented for a robotic squid with four soft arms, to be used in underwater operations. The teleoperation system consists of dissimilar master-slave system kinematics, with multiple master systems and multiple slave systems. An operator utilizes two haptic devices for the manipulation of the four soft robot arms. Haptic feedback is incorporated into the system for ease of use. The slave system within the implemented teleoperation system is simulated using hardware-in-the-loop simulation. For this purpose, communication protocols from the real system are employed. In other words, the applied teleoperation system is integrated within the hardware-in-the-loop simulation of the real system.

Experiments were conducted to validate that the implemented system is a real-time system and to evaluate the ease of use of the system from the operator's perspective. Additionally, experiments were expanded to measure the impact of haptic feedback on the performance of the operator. The experimental results indicate that the system is a real-time system and haptic feedback improves the system's ease of use.

ÖZET

BİR ROBOTİK MÜREKKEPBALIĞININ KONTROLÜ İÇİN GERÇEK-ZAMANLI BİR TELEOPERASYON SİSTEMİNİN UYGULANMASI

Teleoperasyon, bir robotik sistemin, operasyon ortamından uzaktan kontrol edilmesi olarak tanımlanır. Yumuşak robotların teleoperasyonu son yıllarda artarak devam eden araştırma konusu olmuştur ve halen araştırılmayı bekleyen konular vardır.

Bu çalışmada gerçek-zamanlı bir teleoperasyon sistemi bir robotik mürekkep balığının su altı operasyonlarda kullanılmak üzere sahip olduğu dört yumuşak kol modeli için uygulanmıştır. Teleoperasyon sistemi, benzemez ana-bağımlı sistem kinematiği ve çoklu ana sistem ve çoklu bağımlı sistemden oluşan bir teleoperasyon sistemidir. Operatör, dört yumuşak robot kolunun manipülasyonu için iki haptik cihaz kullanmaktadır. Kullanım kolaylığı amacıyla sisteme haptik geribesleme tanımlanmıştır. Uygulanan teleoperasyon sistemindeki bağımlı sistem donanım içeren simülasyon vasıtasıyla simüle edilmiştir. Bunun için, gerçek sistemdeki iletişim protokolleri kullanılmıştır. Başka bir deyişle uygulanan teleoperasyon sistemi, gerçek sisteme donanım içeren simülasyon uygulanarak entegre edilmiştir.

Uygulanan sistemin gerçek zamanlı olup olmadığı ve sistemin operatör açısından kullanılabilir olduğunu test etmek amacıyla deneyler yapılmıştır. Ayrıca, haptik geribeslemenin operatörün performansı üzerindeki etkisini ölçmek amacıyla deneyler genişletilmiştir. Deney sonuçları, sistemin gerçek zamanlı olduğunu ve haptik geribeslemenin sistemin kullanım kolaylığını geliştirdiğini göstermiştir.

to my father, Osman Cezayirli

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1

INTRODUCTION

Real-time refers to systems that can respond to inputs or events with no or minimal time delay. Depending on the time delay of the system's response real-time systems are mainly divided into two categories, hard real-time and soft real-time systems. Hard real-time systems are expected to respond to inputs or events within a defined time delay. The use of hard real-time systems is necessary where precise timing is essential such as flight control systems, and medical applications. Soft real-time systems on the other hand can tolerate occasional minor failures of response time. The use of soft real-time systems considering the need for response precision extends to autonomous navigation, teleoperation, and human robot interactions. Real-time teleoperation refers to the operation of a device or a system remotely in real time.

1.1. Teleoperation & Haptic

Teleoperation, in essence, refers to the expansion of an individual's skills to operate in a remote environment. The word "tele" comes from Greek and means "at a distance." Teleoperation serves as a bridge between human capabilities such as control, perception, and planning, and the precision, durability, and mobility of robots (Sheridan, 1989). The use of teleoperation is necessary for specific operations that may be costly, dangerous, or impossible for humans. For instance, it is employed in applications conducted in environments like space or underwater, where human involvement could be hazardous or unfeasible. Additionally, it is utilized in scenarios involving materials such as nuclear substances that pose significant health risks or in minimally invasive surgical procedures where human error could lead to lasting consequences.

Teleoperation systems have two sub-systems which are called master and slave. The master system is where the user/operator establishes physical contact, enabling them to perform the desired tasks in the operation environment. The slave system is where the teleoperated manipulator is located and it performs the tasks given by the master system.

Figure 1.1. A Polygraph

(Source: Wikipedia)

Teleoperation can be either a single mechanical system or can consist of two or more separate mechanical systems with a designated communication channel and method. Figure 1.1 represents a teleoperation system, a polygraph, patented by John Isaac Hawkins in 1803 for copying writing, which is a completely mechanical single system and does not have a separate communication channel. Figure 1.2 depicts a representation of a teleoperation system consisting of two separate mechanical systems. The teleoperation system mentioned in the thesis discussed further, is the teleoperation system shown in Figure 1.2, which includes a communication line.

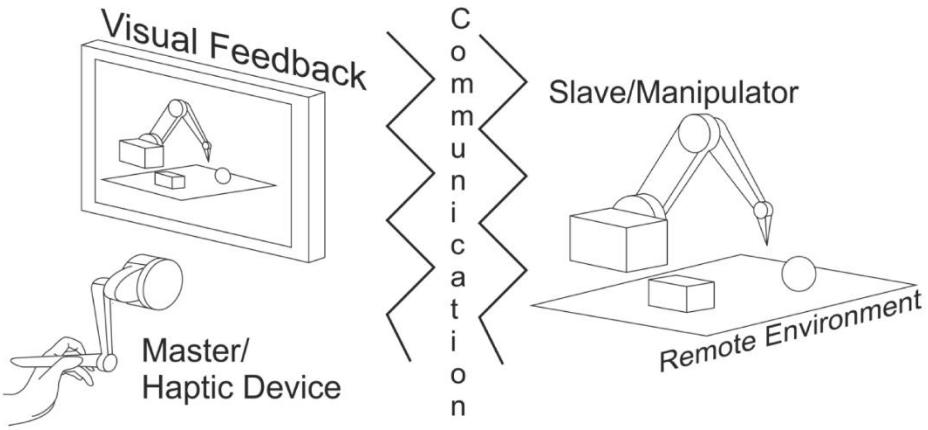

Figure 1.2. Representation of a teleoperation system

(Source: Uzunoglu, 2012)

The word "haptic" is derived from the Greek word "haptomai" which means "to touch" or "to feel". The purpose of haptics is to allow the user to perceive a specific situation, feature, or stimulus. Haptic technology has a broad range of applications, including teleoperation. Examples of haptic feedback include the vibration of mobile phones when receiving calls or messages, notched structures found on the edges of certain roadways, the ridged structure beneath the F and J keys on computer keyboards, and the vibration of computer and console game controllers.

In teleoperation systems, haptic feedback is a method of conveying the interaction between the slave system and the operational environment to the user via the master system. An example of this is a surgical robot, the da Vinci surgical robot. When operating, da Vinci provides haptic feedback to the surgeon, enabling them to perceive the interactions of the slave system, whose working environment is the patient, during the surgery through the main system used by the surgeon during the operation. In Figure 1.3, da Vinci surgical robot can be seen.

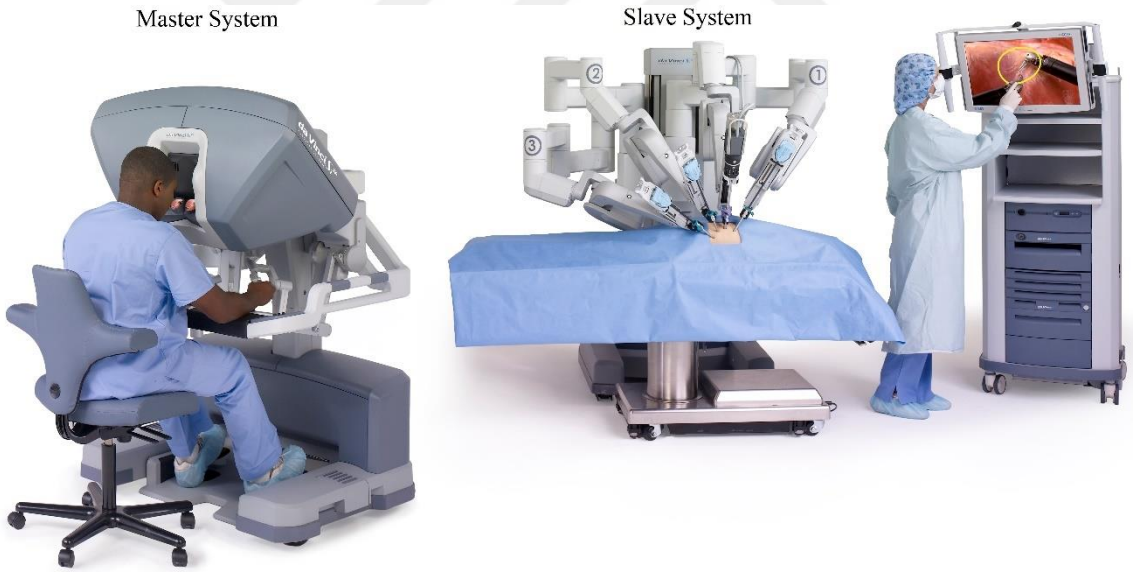

Figure 1.3. da Vinci surgical teleoperation system

(Source: da Vinci)

1.2. Continuum and Soft Robots

Most commonly, robot mechanisms are rigid structured mechanisms that consist of rigid links and joints that connect these links. Depending on the joint types and connection methods, rigid robot mechanisms have a finite number of degrees of freedom

(DoF). And when the DoF of a robot mechanism exceed the necessary degree of freedom for the task defined, they are called redundant mechanisms. When a mechanism has a high degree of redundancy, they are called hyper redundant mechanisms. Continuum robots are hyper-redundant mechanisms which means, if not restricted, they always have more DoF than needed. When continuum robots are built with soft materials, such as silicone or rubber-based materials, they are called soft robots. Thuruthel et al. (2018) also stated soft robots have a virtually infinite number of freedoms, including bending, extension, torsion, and buckling. In Figure 1.4, a classification is shown for robot types. Also, in Figure 1.5 a representation of the definition of soft and continuum robots made by Robinson and Davies (1999) and Santina et al., (2020) is shown.

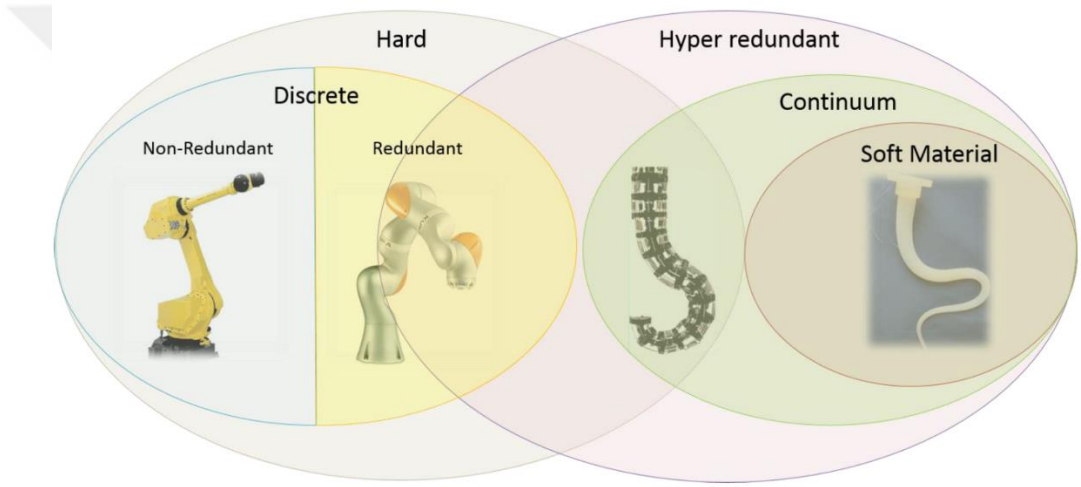

Figure 1.4. Robot manipulator classification
(Source: Triverdi et al., 2008)

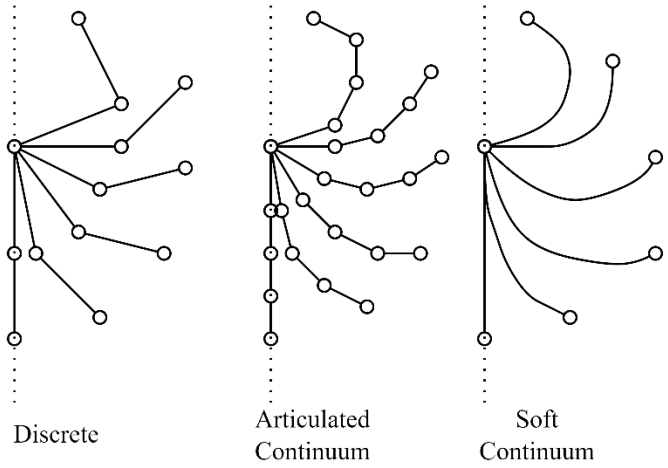

Figure 1.5. Continuum and soft robot classification

1.3. Robotic Squid for Underwater Manipulation and Intervention

The proposed research is to design a biomimetic unmanned underwater vehicle (UUV) that takes inspiration from squids. The primary objective of designing this robot is to develop an underwater system that can effectively interact with and manipulate the underwater environment, surpassing the capabilities of existing systems. In this model, the robot incorporates a streamlined and agile body, along with highly flexible and versatile multitask tentacles. These tentacles resemble those of squids and enable the robot to perform various actions such as reaching, grabbing, pulling, and exploring. Just like the squid tentacles, the robot arms are constructed with hyper-redundant links using soft and deformable materials.

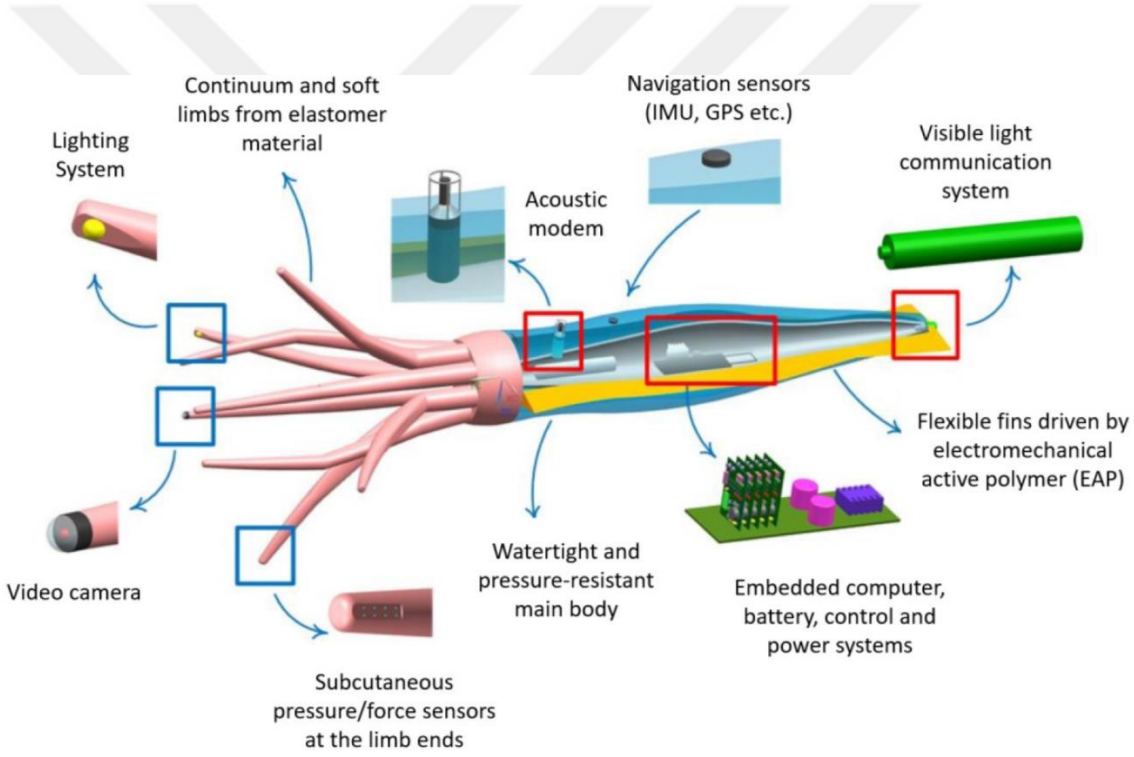

Figure 1.6. Robotic squid

(Source: Emet, 2022)

This work is conducted under a project funded by *1003 Priority Areas R&D projects and Support Program* (Project ID: 216M219) via TUBITAK in collaboration between Bahcesehir University and Izmir Institute of Technology. The proposed robot model is represented in Figure 1.6. The robot has two main modes, one is the cruising mode and the other is the manipulation mode. During the cruising mode, the aim is to reach the destination. Upon reaching its destination, the manipulation mode can be

activated. During the manipulation mode, the robot uses its tentacles to interact with the environment. Manipulation mode is the work package of Izmir Institute of Technology. The thesis proposes to design a teleoperation system with a haptic interface for manipulation mode for the robotic squid.

1.4. Aim of the Thesis

This thesis aims to develop a real-time teleoperation system for four robotic squid arms that are simulated in a hardware-in-the-loop simulation and teleoperated with two haptic devices. The study investigates teleoperation of multi-master multi-slave systems, teleoperation with dissimilar master-slave kinematics, teleoperation of soft/continuum robots, and haptic interfaces for the control of soft/continuum robots. The real-time operation of the teleperation system is validated via experiments. User experiments are carried out to evaluate the impact of the haptic interface on the overall system's usability.

1.5. Contributions

This thesis contributes to the teleoperation of soft robots, specifically the haptic enabled teleoperation approach for tendon driven soft robots with multi-master multi-slave teleoperation system and dissimilar slave-master kinematics.

1.6. Outline

The study is presented in six chapters. The chapters are Introduction, Literature Survey, Methodology, Implementation, Experiments & Results, and Conclusion. In Chapter 2, teleoperation systems, and haptic interfaces in the literature are surveyed and presented. In Chapter 3, the chosen methods along with hardware and software selection and information conveying are presented. In Chapter 4, the implementation of the chosen methods is presented. In Chapter 5, real-time validation experiments and user experiment procedure with implemented system, and the results of the experiments are provided. Lastly, in Chapter 6, the thesis concluded, and future work is addressed.

CHAPTER 2

LITERATURE SURVEY

In the literature, various types of teleoperation systems have been studied and classified based on specific aspects of teleoperation. These aspects include the communication channel used between the master and slave devices, the number of master and slave devices, as well as the kinematic type and structure of both the master and slave devices. These categorizations aim to address specific problems and provide insights into the design and implementation of teleoperation systems.

2.1. Teleoperation System Architectures

Teleoperation systems are divided into two categories, unilateral and bilateral by their communication structure. In unilateral teleoperation, the information flow is only in one direction. The information flow is from the operator through the master system to the operation environment through the slave system. On the other hand, in bilateral teleoperation, the flow of information is two-way. The information received from the operator through the master system is transmitted to the slave system through a communication channel, while the information gathered from the operation environment by the slave system is conveyed to the master system and thus the operator. The specific type of information to be collected from the slave system depends on the nature of the defined task. This information is collected and compiled through sensors. For example, force/torque sensors can be used, as well as image, sound, position/proximity, and velocity sensors. The collected information can be fed back to the user through various means, such as force/torque, visual display, sound, or vibration. Figure 2.1 illustrates a simplified schematic diagram of unilateral and bilateral teleoperation systems.

Bilateral teleoperation systems are classified into two different architectures based on the communication method and the quantity of command signals (Dede and Tosunoglu, 2006). These architectures are known as two-channel and four-channel architectures. The mentioned channel configuration is still an active research area (Kubo et al., 2007).

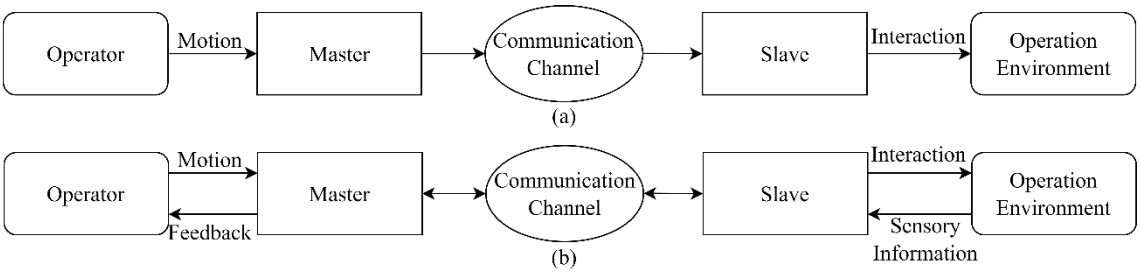

Figure 2.1. Teleoperation types (a) Unilateral and (b) Bilateral teleoperation

In a two-channel bilateral teleoperation system, as can be seen in Figure 2.1 (b), the operator typically feeds motion (position and/or velocity) information to the master system. The master system then transmits this information to the slave system. The force or torque resulting from the interaction between the slave system and the operational environment is conveyed to the user as force feedback through the main system. This system is commonly referred to as the "position-force" architecture.

In a four-channel bilateral teleoperation system, the master system transmits both motion data (e.g. position/velocity), as well as force or torque information, to the slave system. The slave system generates feedback containing the same type of information based on its interactions with the operational environment. This feedback is then provided to the master system. The four-channel teleoperation is preferred in situations where there may be undesired delays in the communication channel. It is chosen when the information/data flow between the master and slave systems is required to be at a desired speed and/or level of adequacy. Figure 2.2 illustrates a simplified schematic diagram of a four-channel bilateral teleoperation system.

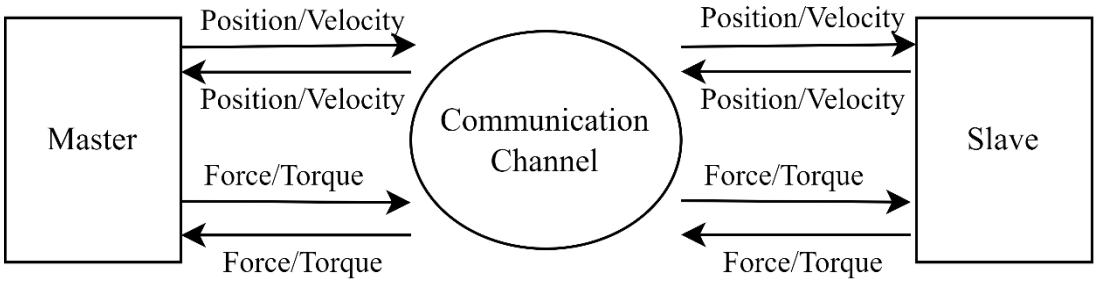

Figure 2.2. Four-channel bilateral teleoperation system

2.2. Teleoperation Control Strategies

In the implementation of a teleoperation system, there are variables that need to be considered. Examples of these variables include the effectiveness of the communication channel, the conditions of the operating environment, and the resolution and capacities of sensors and actuators. For instance, if there is no significant delay or insufficiency in the communication channel, a bidirectional teleoperation system can be utilized. However, if such delays or limitations exist, relying solely on this method may impede the stable operation of the system. This is because delays in transmitting information from the master system to the slave system can result in phase differences. In the early literature, three distinct control methods have been proposed (Stefano et al., 1999). These methods include the move-and-wait strategy, direct control, and supervised control. Figure 2.3 illustrates a schematic representation of these approaches.

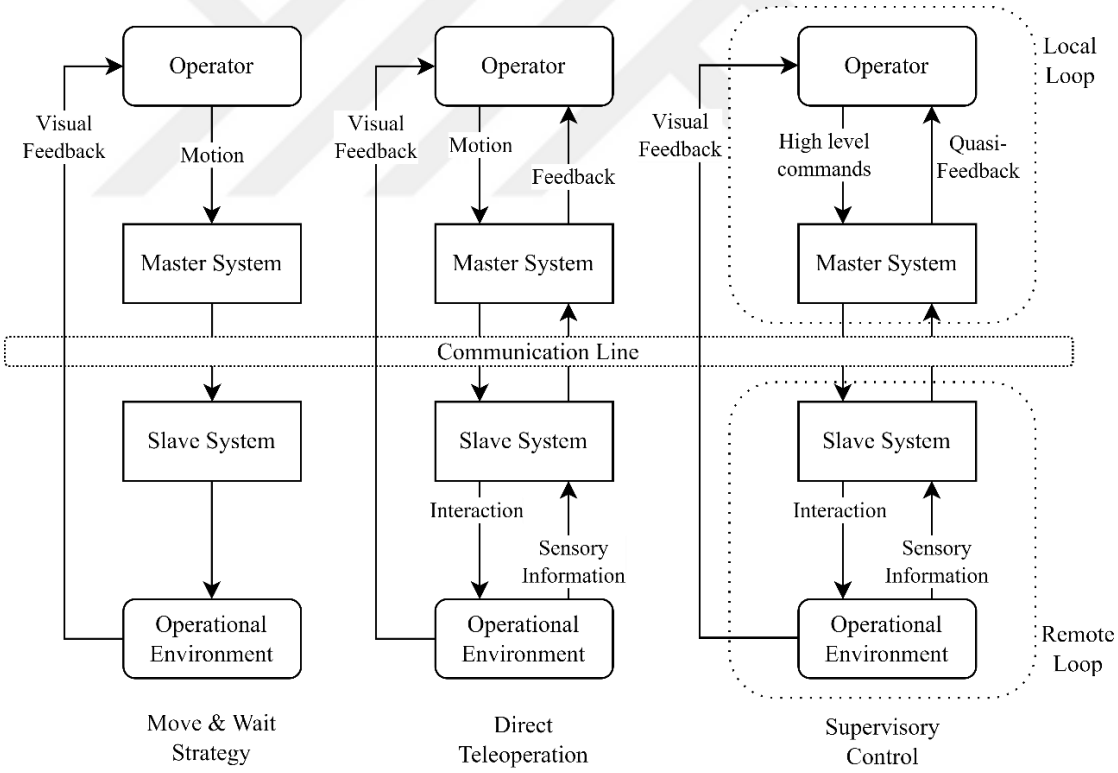

Figure 2.3. Teleoperation control strategies

Ferrell (1965) stated that users developed a move-and-wait strategy in a one-way teleoperation system with visual feedback and a time-delayed communication channel. This method, as the name implies, involves the user visually tracking the slave system in

the operational environment during the time delay and waiting for the completion of the slave system's motion.

The direct control method can be used when time delays are minimal or negligible. In this method, the slave system operates based on real-time information received from the main system and sends real-time feedback to the main system. This system can be compared to a closed-loop control system.

For teleoperation systems with time-delayed communication channels, Ferrell and Sheridan (1967) proposed the supervised control method. In the supervised control method, instead of controlling every movement of the slave system, certain predefined actions are automated. These predefined actions are referred to as high-level commands. These commands are sent from the operator to the slave system through the master system, and since the slave system performs these actions autonomously, the resulting time delays are compensated for. Additional control loops, namely local loop and remote loop are defined to implement this method. The remote loop is responsible for receiving and interpreting information from the main system and coordinating the autonomous motion of the slave system. The local loop involves a model of the slave system and can generate partial feedback using this method.

In addition to these three methods, various approaches have been proposed for teleoperation systems with direct control and time-delayed communication channels. These include the passivity-based control method suggested by Anderson and Spong (1989), the wave-variable method proposed by Niemeyer (1996), and the model-mediated teleoperation methods proposed by Mitra and Niemeyer (2008).

The passivity-based control method proposed by Anderson and Spong (1989) is based on scattering theory. A system is considered passive when the energy generated within the system is less than the energy consumed by the system. This method suggests that if a system is passive, it will be stable. In other words, the aim is to distribute or store the input power of the system in such a way that it cannot provide more energy than the initial energy of the system. The velocity and force variables are transformed into scattering variables. Stability is achieved by transmitting these variables over the communication channel.

The wave-variable method proposed by Niemeyer (1996), is designed for bilateral haptic feedback teleoperation systems. This method emerged from the re-formulation of scattering approaches. In this method, both the control of the master system and the control of the slave system are exposed to a virtual wave generator. This virtual wave

generator acts as a transformed representation of the master system manipulator in the wave domain. Hence, it serves as a desired trajectory for the generated wave controllers. In general, wave-variable-based controllers are considered protective, and robust, and do not require any information about the remote environment or time delay (Alise et al., 2005). As a result, most bilateral teleoperation systems are designed within the framework of passivity, using scattering or wave-variable methods to ensure stability against time delays in the communication channel.

The model-mediated teleoperation method proposed by Mitra and Niemeyer (2008) introduces a different teleoperation structure. In this teleoperation structure, the slave system is modeled in a virtual world, and this virtual model is displayed to the user in a graphical user interface. The operator interacts with this modeled virtual world through the master system. The haptic feedback resulting from the virtual interaction is transmitted to the user. Subsequently, the user's motion is transmitted to the slave system through the time-delayed communication channel via the master system.

Figure 2.4 illustrates the model-mediated teleoperation system schematic. The section referred to as the proxy in the figure is the virtual model of the slave system within the master system. The section labeled as the virtual model represents the virtual operation environment within the master system. As previously defined, when the user interacts with the master system, the master system interprets this interaction through the proxy and exposes it to the virtual operation environment within the master system. The resulting information is then conveyed to the user through the haptic device via the slave system proxy. The part referred to as virtual model update is where the interactions of the slave system's sensors with the operation environment are measured and interpreted, and then this information is fed into the virtual world within the master system. Possible changes in the operation environment are represented in the virtual world created within the master system using this method.

2.3. Multi-Master Multi-Slave Teleoperation

The concept of teleoperation extends to collaborative integration with multiple master and multiple slave systems (MMMS). MMMS concept is proposed to comply with the problem of single-master single-slave (SMSS) systems' lack of dexterity and flexibility with complex tasks (Minelli et al., 2019). Multiple master systems can be

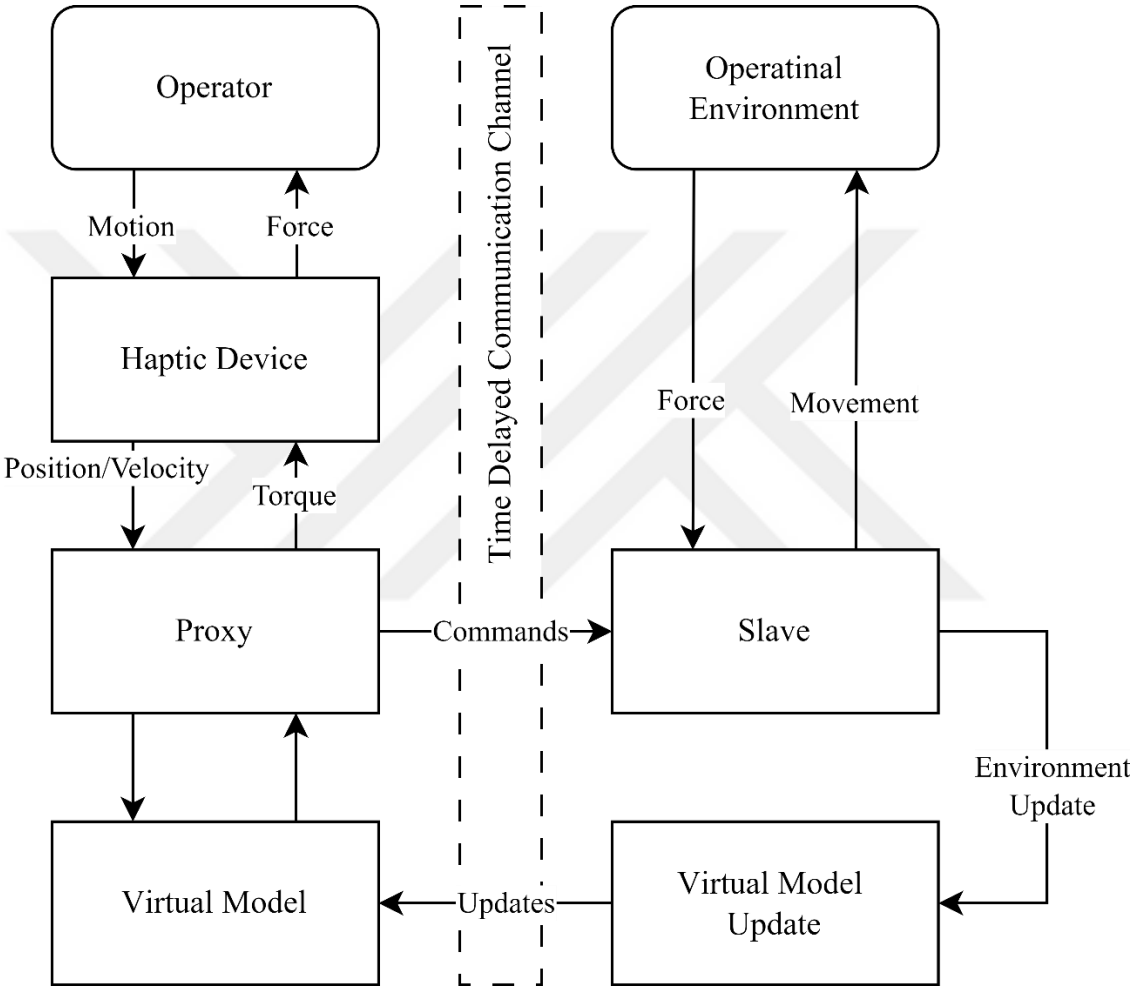

Figure 2.4. Model mediated teleoperation system schematic

multiple operators with multiple masters or a single operator with multiple masters. In Figure 2.5 single operator with multi-master multi-slave teleoperation schematic is presented with *n* many slaves and *N* many masters. The previously mentioned da Vinci surgical robot is an example of this type of teleoperation which is shown in Figure 1.3. Da Vinci surgical robot has two hand-held master manipulators to teleoperate four slave arms. In Figure 2.6 multi operator with multi-master multi-slave teleoperation schematic is presented with *n* many slaves, *N* and *M* many masters, and *K* many operators.

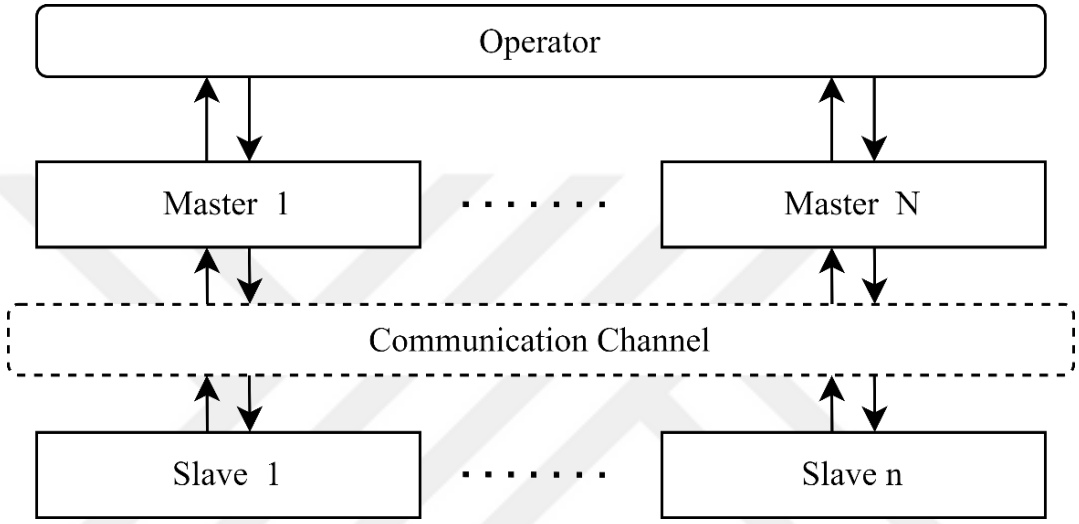

Figure 2.5. Single operator multi-master multi-slave teleoperation schematic

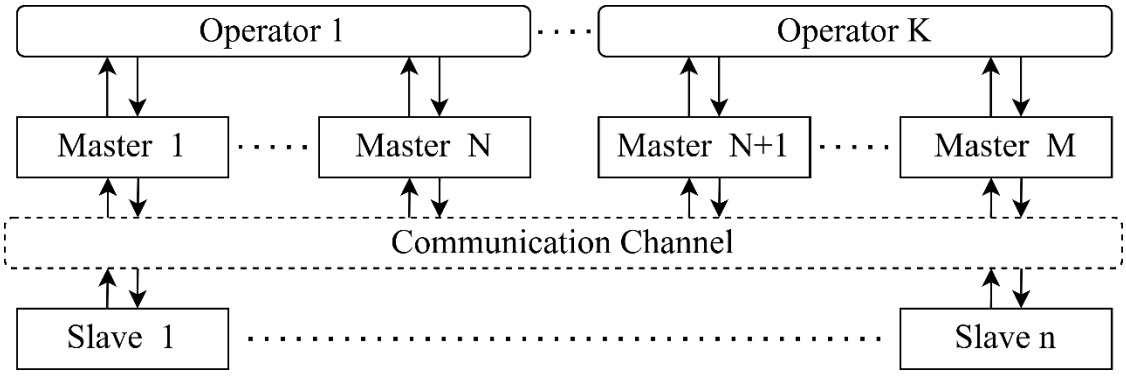

Figure 2.6. Multi operator multi-master multi-slave teleoperation schematic

Sirospour and Setoodeh (2005) studied coordinated control for MMMS teleoperation with multi operators using an adaptive nonlinear control method and with two 1-DoF slaves and two 3-Dof planar twin pantograph haptic interfaces as masters. Yang et al. (2023) designed a new control strategy for MMMS teleoperation with multi operators and concluded with the developed method, operators can accomplish complex

tasks with higher precision. Khademian et al. (2011) examined environmental factors that affect operator performance in a dual-user system involved in collaborative tasks in a virtual environment. Shahbazi et al. (2011) presented an adaptive impedance controller against fixed time delays and conducted stability analysis to stabilize a dual-user system. Passenberg et al. (2010) extended the model mediation teleoperation control method to MMMS teleoperation with multi operators using 1-DoF masters and 1-DoF slaves.

Balch and Arkin (1998) have developed a behavior-based control theory for collusion based on formation distortions in multiple slave systems. Lalish et al. (2006) proposed a virtual structure approach where the formation of the slave system is treated as a single virtual rigid structure. This method ensures the maintenance of the formation and predicts the behavior of all slave system robots, but it leads to an increased communication delay. Desai et al. (2001) suggested a leader-follower strategy in which selected leaders are responsible for guiding the followers. The followers are directed with predefined gaps by the leaders. This leader-follower approach improves the system's simplicity, modularity, and reliability. However, it also presents potential challenges such as leader failures that could disrupt the functionality of the system.

Minelli et al. (2019) have developed a new two-layer architecture for an MMMS bilateral teleoperation system with one operator incorporating the concept of a shared energy tank for tele-assisted laparoscopic surgery. On the master side, two devices were used, 4-DoF Simball Joystick laparoscopic haptic device and 6-DoF Geomagic Touch haptic device. On the slave side, 7-DoF KUKA LWR 4+ and 6-DoF Universal Robots UR5 manipulator arms are used.

2.4. Teleoperation with Dissimilar Master-Slave Kinematics

As previously mentioned, teleoperation in general is composed of two separate mechanical systems, manipulators; the master and the slave. When these two manipulators share a similar kinematic structure; the position and/or velocity of the master manipulator can be mapped to actuation of the slave manipulator with ease. However, when these two manipulators have dissimilar kinematics, there must be a kinematic interface connecting master and slave manipulators. Matsuhira et al. (1993) defined this as master-slave manipulation with different configurations.

Peer et al. (2005) discussed a compliant control for 10 DoF haptic device (ViSHaRD10) as master manipulator and a 7-DoF slave manipulator. The study utilized only 6 DoF for each manipulator and concluded the redundancies of both manipulators were utilized for certain tasks such as avoiding singularities or joint limit collisions.

2.4.1. Teleoperation of Soft/Continuum Robots

The primary consideration in teleoperating soft robots is the difference in kinematics between the master system and the slave system. While the slave system, being a soft robot, is considered to have virtually infinite degrees of freedom (Thuruthel et al., 2018), the master system has a limited number of degrees of freedom. This gives rise to dissimilar master-slave teleoperation. Also, there are studies that use commercially available devices that are not manipulators such as HTC Vive augmented reality kit.

Stroppa et al. (2020) designed a wearable master that converts the operator's arm to a manipulator to control a continuous robot, instead of using a traditional rigidly connected master manipulator. El-Hussieny et al. (2018) developed a flexible master for a soft-growing robot.

Amaya et al. (2021) compared direct and indirect mapping methods for two different master devices, a Geomagic Touch and an HTC Vive, to manipulate a soft robot.

Csencsits et al. (2005) introduced a new series of mappings from a joystick to a continuum robot to assess the intuitiveness and effectiveness of control at both position and velocity levels. They conducted a set of tests with human subjects to evaluate the proposed mappings.

Fellmann et al. (2015) conducted a comparative study where they evaluated the performance of controlling a concentric tube continuum robot for surgical tasks, using different types of master devices, including a 3D mouse, Novint Falcon haptic device, and gamepad. The study assessed the performance in terms of task completion times and accuracy.

Frazelle et al. (2016) suggested a mapping and studied the effectiveness and intuitiveness of a teleoperation system with a slave manipulator which is a three-segment, 9-DoF continuum robot, using a 6-DoF rigid-link master device. They concluded after testing the system on fifteen volunteers that the teleoperation of continuum robot manipulators with rigid-link master manipulators with mapping suggested is viable.

Ouyang et al. (2018) presented a teleoperation system and a shape control method for a three-section continuum robot as slave and Geomagic Touch haptic device as master; and concluded that the shape control method enhances the behavior of the system. Bhattacherjee et al. (2018) presented a teleoperation system using Geomagic Touch haptic device as the master and a tendon driven continuum robot arm as the slave. The position of the wrist point of Geomagic Touch is mapped to the kinematic model of the slave which is modelled in a virtual environment.

2.4.2. Haptic Interfaces for Soft/Continuum Robots

The literature on the haptic interface with teleoperation can be divided into two sub-sections depending on where soft/continuum robots are used. Some of the studies use the soft/continuum robots for master system while others use them for slave systems.

Frazelle et al. (2020) designed a new continuum haptic master manipulator named HaptOct, an impedance type of haptic device to teleoperate continuum robots. The haptic feedback is generated through controlling the cable tensions which are the actuators of the manipulator. They reported that although there are limitations, the resultant haptic enabled manipulator works as intended and is capable of providing haptic feedback to the operator. This study develops a soft/continuum master for teleoperation of soft/continuum robots, and is not in the subject of dissimilar master-slave kinematics.

Xie et al. (2022) developed a new 3-D haptic trackball master device for the teleoperation of continuum robots. A tendon driven two-sectioned continuum robot as the slave manipulator is used which is manipulated with cable displacements. They reported that the master device yielded the best results with and without haptic feedback compared to a joystick.

Xie et al. (2023) designed a new impedance type haptic a 4-DoF joystick master manipulator for the teleoperation of continuum manipulators for medical applications. They reported that the designed haptic manipulator makes the operator manipulate intuitively and with ease.

Naghibi et al. (2019) developed a new soft sensing module for pneumatically actuated soft slave for endoscopic teleoperation applications. The haptic feedback is generated via the deformation of the slave module and fed to the master system which is

a Geomagic Touch haptic device. They reported that the sensing module is capable of producing realistic haptic feedback.

2.5. Conclusion on Literature Survey

The background on the teleoperation systems is given in this chapter. The teleoperation system architecture and control methods are reviewed. For the teleoperation of the robotic squid, the communication channel is a visible light system. A foreseen problem is with the bandwidth of the communication channel. The data transfer is considered not sufficient for conveying all the information gathered by the slave system to the master system in real time. As a result, some information had to be sacrificed, such as real-time camera footage. Because of this very reason, model mediated teleoperation method is used since the slave system's environment can be represented in the master side in an abstract way.

Based on the investigations presented in this Chapter, the novelty of this thesis is validated to be including the subjects of (1) multi-master and multi-slave teleoperation, (2) haptic enabled (3) rigid master – soft slave teleoperation. A list of studies that targeted these 3 topics are listed in Table 2.1.

Table 2.1. Literature survey summary relevant to the thesis' novelty

Subjects / Studies	Multi-Master & Multi Slave	Rigid Master & Soft Slave	Haptic Enabled
Sirospour and Setoodeh (2005)	✓	×	×
Csencsits et al. (2005)	×	✓	×
Passenger et al. (2010)	✓	×	×
Fellman et al. (2015)	×	✓	×
Frazelle et al. (2016)	×	✓	×
Ouyang et al. (2018)	×	✓	×
Naghibi et al. (2019)	×	✓	✓
Minelli et al. (2020)	✓	×	×
Stroppa et al. (2020)	×	✓	×
Frazelle et al. (2020)	×	×	✓
Amaya et al. (2021)	×	✓	×
Xie et al. (2022)	×	✓	✓
Xie et al. (2023)	×	✓	✓

CHAPTER 3

METHODOLOGY

In the teleoperation system is developed to operate the robotic squid, which has four tentacles (e.g., soft arms). Therefore, these four soft arms are to be teleoperated in pairs using two haptic devices. The hardware and software specifications of the planned system, as well as the selected and intended hardware and software to be used, are provided in detail in the relevant sections. In addition to this teleoperation system, a user interface that will work in conjunction with this system has been developed.

The developed system is a multi-master-multi-slave system and it involves a kinematic dissimilarity between the master and slave systems. Additionally, the presence of possible time delays and abstract information exchange capability in the communication channel has been considered. Therefore, the model-mediated teleoperation method has been employed to compensate for this delay and re-generate the abstracted slave environment in the master side. Furthermore, without having the physical models of the soft robotic arms that constitute the slave system, a hardware-in-the-loop simulation system has been created.

3.1. Model Mediated Teleoperation

The working principle of the model-mediated teleoperation method is previously explained under the section "2.2. Teleoperation Control Strategies". In Figure 2.4, this method is illustrated. As this method is deemed suitable, this section will discuss the working and implementation method of the model mediated teleoperation method.

The motion of the master system is transmitted to a virtually created replica of the slave system, named as the "proxy". The proxy then interacts with the virtual replica of the operation environment. While the resulting interactions are conveyed to the user, the proxy also relays these commands to the actual slave system. This process, considering time change and motion for a single DoF, is shown in Figure 3.1. When there is no contact between the proxy and the virtual replica of the operation environment, the proxy and the

master move together. This motion is observed with a delay on the slave. The delay is caused by the communication system used. The schematic labelled "Intermediate Position" on the right-hand side of Figure 3.1 provides an example of this motion.

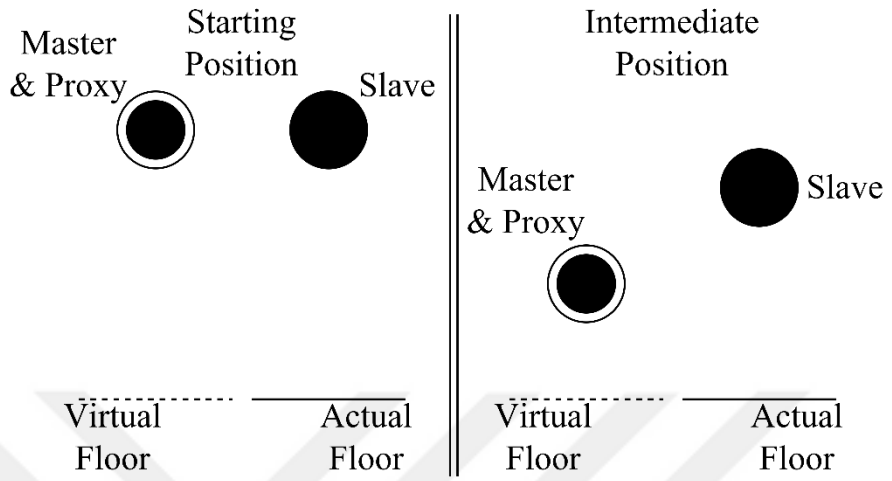

Figure 3.1. Master and slave representation when there is no interaction

The interaction of the proxy with the created virtual world is shown in Figure 3.2. When interaction occurs, haptic feedback is to be provided to the user. The purpose of this haptic feedback is to alert the user before any interaction takes place in the slave system. The haptic feedback to be transmitted is calculated based on the interactions in the virtual world. This calculation is performed through haptic rendering, which can be mathematically modelled as a simple spring-damper system.

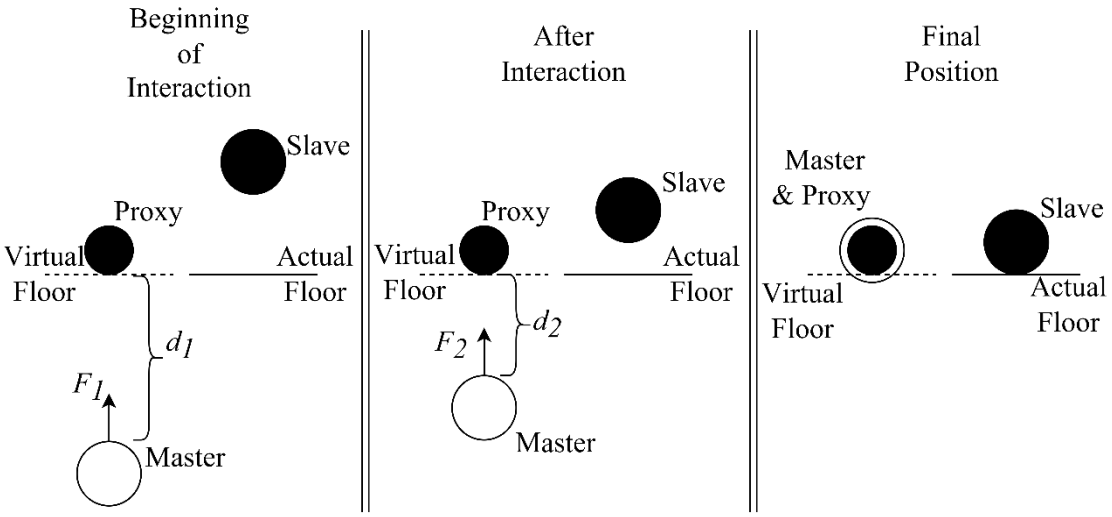

Figure 3.2. Master and slave representation during and after interaction

The virtual floor and the proxy can interact in the virtual world. However, to perceive this interaction on the master system side, a mathematical model is required. As mentioned before, when there is no contact, there is no positional difference between the proxy and the master system. However, when contact occurs, the proxy interacts with the virtual world while the master system continues its motion without such interaction, resulting in a positional difference between them. The mathematical spring-damper model accounts for this positional difference between the master system and the proxy. In its simplest form, the model is represented by the spring equation given in (3.1), where F is the calculated force to be transmitted to the user, and d is the positional difference between the master and the proxy. The spring constant k is a property of the virtual floor in the virtual world and can have different values for different types of virtual floors. Therefore, the transmitted force to the user increases proportionally with the positional difference between the proxy and the master. This allows the user to feel the force until the positional difference is minimized. Figure 3.2 represents this process symbolically, and the notation in the figure indicates that F_1 is greater than F_2.

$$F = k\,d \tag{3.1}$$

The described haptic rendering and feedback production apply not only when the master, the proxy, and the slave are changing but also when the master, the proxy, and the slave are stationary and the operational environment changes. This occurs on the slave system side as previously mentioned in the model-mediated teleoperation. The virtual world update information collected by the sensors of the slave system in the operational environment is transmitted to the master system. The master system updates the virtual world and is to provide interaction feedback to the operator. This is depicted in Figure 3.3.

If the situations that mentioned in Figure 3.2 and Figure 3.3 happen simultaneously, the described system behavior would not change.

The haptic rendering process defined so far is not used for the implemented system in this Thesis study. Instead, the information for the haptic feedback is acquired via the cable tensions that are placed on the soft arms (tentacles) of the squid. The haptic feedback generation is discussed in detail in Section 4.1.2.

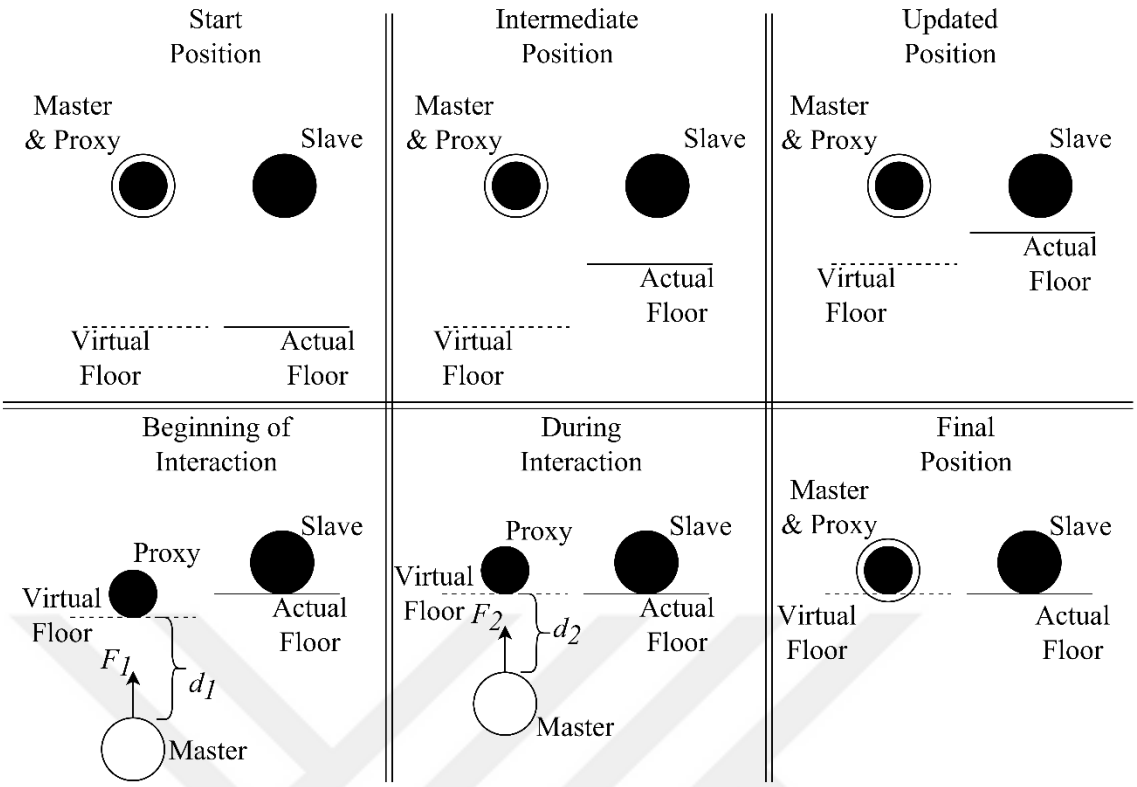

Figure 3.3. Changing environment with stationary master and slave representation

3.2. Hardware-in-the-Loop Simulation for Slave System

Hardware-in-the-loop (HIL) is a testing technique used before implementing the generated codes on the actual systems hardware and when conducting actual experiments on a prototype of the actual system is difficult, or when the hardware to be used in the experiments is prohibitively expensive. In this technique, a virtual replica of the actual system is simulated. The simulated environment interacts with the actual hardware as it would, allowing for testing and evaluation.

In the design of the teleoperation system, it was deemed appropriate to use hardware-in-the-loop simulation because it allows for budged friendly development of the soft robot's arms. In Figure 3.4, the real system components and communication protocols are given. In this figure and the following figures presented, double lines represent the connections and interactions, and black arrows represent the conveyed information flow. In this HIL simulation, the embedded system and computers for the real system were used as the hardware components. Additionally, the communication protocols intended for the real system were implemented. The system diagram of the

hardware-in-the-loop simulation, along with the planned communication protocols, methods, and interaction aspects, are shown in Figure 3.5.

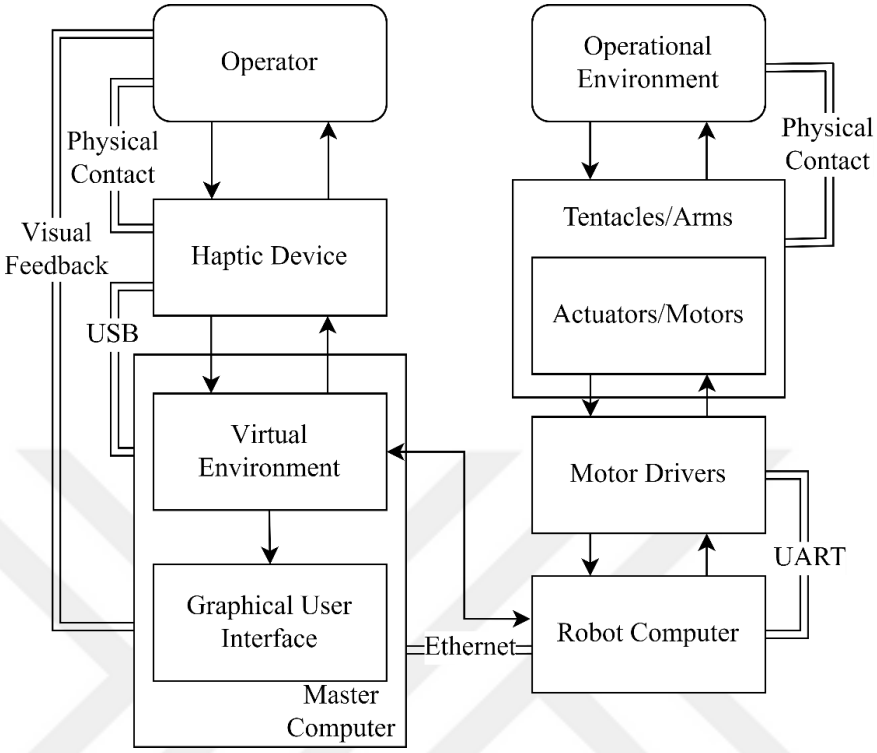

Figure 3.4. Actual system schematic

3.2.1. Hardware

Hardware-wise, it has been determined that in addition to the necessary power adapters and connection cables, three computers, two haptic devices, and an embedded system board are required. Two of the three computers are master system computer and HIL simulation computer. Both computers are selected to be the same. They have Intel i7-13700H processors, Nvidia GeForce RTX3060 graphic cards and 16GB of RAMs. The other computer is the slave system computer, the Nvidia Jetson Nano, which is referred to as a developer kit. It is called a developer kit because it can directly establish connections with low-level hardware without the need for any additional interfaces or cables. It is classified as a computer because it operates on a Linux-based operating system. The two haptic devices selected are the Geomagic Touch and Geomagic Touch X haptic devices from 3D Systems. For the technical specifications, please see Appendix. Finally, for the HIL simulation, the STM Electronics STM32F407 Discovery Board was

chosen as the microcontroller embedded system board. For the technical specifications, please see Appendix.

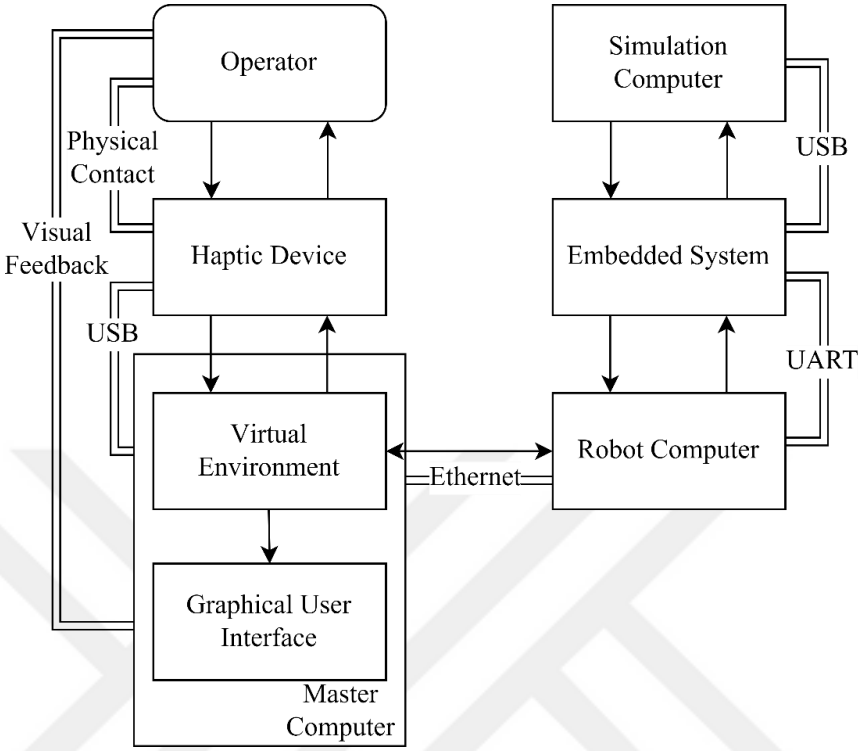

Figure 3.5. Hardware-in-the-Loop schematic

3.2.2. Software

Firstly, a software called Simulation Open Framework Architecture (SOFA) was chosen for the simulation environment. SOFA is an open-source software that started as a project in 2006. It is a C++ based system with integrated support for Python. SOFA is specifically designed for simulating soft tissue models and soft robots. Additionally, plugins have been developed for SOFA, including methods for soft robot control. SOFA is used in both the hardware-in-the-loop simulation and the model-mediated teleoperation in the main system.

The working principle of SOFA can be summarized as follows:

1. The first step in the simulation is to create a root node.
2. Each object to be simulated is added as a child node to the root node. Additionally, variables that affect the entire simulation environment are defined at this stage, such as gravitational acceleration direction and magnitude, and the time step used for the simulation.

3. The added child nodes are further expanded their own child nodes containing information as solver types used by the program, visual information of the model, physical dimensions of the model, material properties, and other inputs that affect the simulation.
4. Finally, one of the animation loop options provided by SOFA is selected.

For a more detailed explanation of how SOFA works, its website (https://www.sofa-framework.org/) can be visited. It should be noted that the models that can be imported to SOFA must be mesh models. The process of model creation and how the model is used during animation are discussed in detail in later sections.

The overall system is composed of several subsystems, and to ensure smooth communication without synchronization issues between these subsystems, a software library called Robot Operating System (ROS) has been chosen. ROS is an open-source software development kit specifically designed to operate seamlessly with both C++ and Python. For more detailed information on how ROS functions, their website (https://www.ros.org/) can be visited.

ROS performs most efficiently on Linux-based operating systems. Considering the compatibility of SOFA with Linux-based systems and the fact that the computer to be utilized on the robot already operates on a Linux-based operating system, it is decided that all three computers will run on Ubuntu, which is a Linux-based operating system. Finally, for modelling purposes, an open-source program named gmsh was used, and further details regarding its usage will be provided in subsequent section.

3.2.3. Modelling of Soft Arms for Simulation

For the simulation model, two of the four arms have been modelled as they are identical to each other. All four arms have a conical shape. The two arms that are defined are used for gripping, reaching, and similar tasks and named task arms. The other two arms are used for the light source and the camera. The task arms have a base radius of 30mm, a tip radius of 10mm, and a length of 600mm. The camera and light arms have a base radius of 30mm, a tip radius of 20mm, and a length of 300mm.

All arms are made of Smooth-On's Ecoflex 00-30, which is a rubber-silicone based material. The Young's modulus is 10 psi or 69.95 kPa. The density is 1.07 g/cm3. Lastly, the Poisson's ratio for rubber is considered as 0.499. When this value is 0.5, it

means that the material is incompressible. Therefore, the arms used are considered almost incompressible.

SOFA simulation has specific mesh formats that can be imported, which include:

- Files with *.obj extension: These mesh files contain surface mesh information.
- Files with *.vtk extension: These mesh files can contain either surface mesh or volume mesh information.
- Files with *.stl extension: These mesh files contain surface mesh information.
- Files with *.off extension: These mesh files can contain either surface mesh or volume mesh information.
- Files with *.msh extension: These mesh files can contain either surface mesh or volume mesh information.

For use in SOFA, the *.stl and *.vtk file extensions have been selected. These extensions were generated using an open-source software called gmsh. Figure 3.6 shows the generated mesh for these two file extensions using the gmsh program, as well as their appearances in the SOFA simulation program as an example of the camera and light arms. Models with *.stl extension define a surface using triangles of various sizes. Models with *.vtk extension, on the other hand, define a volume using tetrahedra. In SOFA, these two models are combined to create the final composite model.

In SOFA simulation, the purpose of these models is not only to visually represent the desired object. The *.stl surface mesh model is used by SOFA to simulate collisions during animation. This modeling can be calculated using different solvers, but the objective is the same. Along with the surface mesh, variables such as alarm distance and collision distance are defined in SOFA. The alarm distance is the proximity between any two triangles on different surface meshes. It is named alarm distance because if there is no other object model within this defined distance, SOFA does not trigger the solvers for collision. In other words, the alarm distance is the starting distance for collision calculations. The collision distance, on the other hand, is the distance at which collision occurs. Therefore, it should be smaller than the alarm distance. Collision calculations are performed considering the behavior of the model and the defined operational environment.

The *.vtk volume mesh models in SOFA are used for the behaviors of the model. It calculates the relationships between each tetrahedron within the mesh, considering the

defined material and physical properties. These properties include Young's modulus of the material, Poisson's ratio, gravitational acceleration, the mass of the material, information about whether the object is fixed or movable, and if it is fixed, the location and which surface/part is fixed.

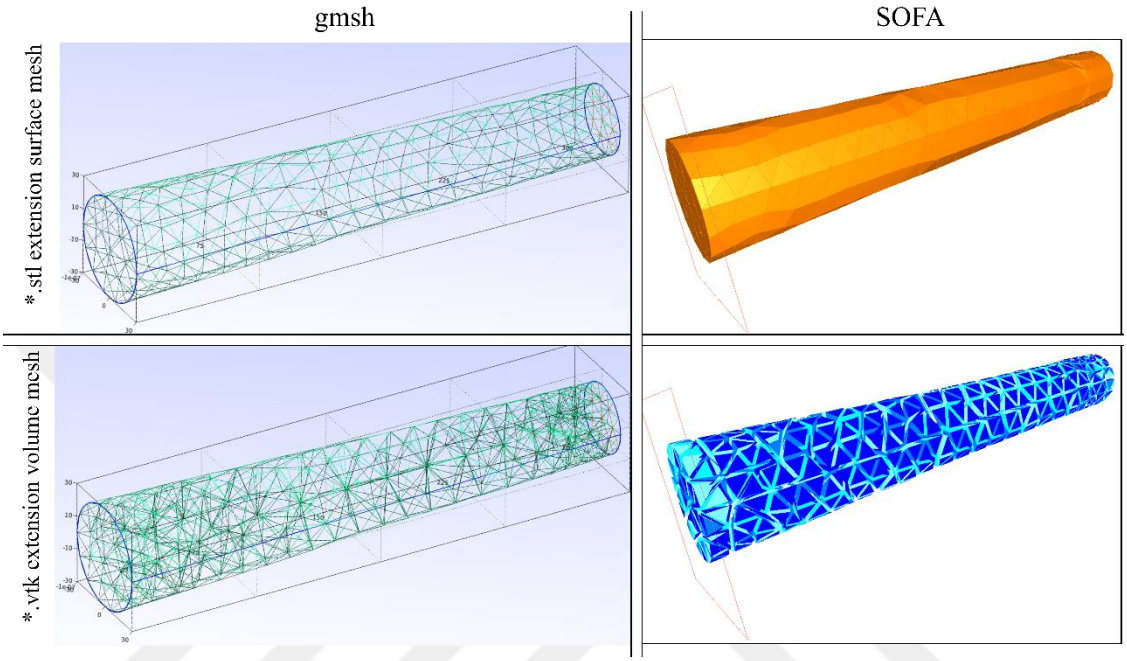

Figure 3.6. Software models

3.2.4. Actuation of Soft Arms for Simulation

Manti et al. (2016) discussed various actuation methods for soft robots. Some of these methods include flexible fluidic actuators, shape-memory materials, electroactive polymers, tendon-driven actuators, and low-melting-point materials. For this project, the tendon-driven actuation method has been selected. In this method, as the name implies, the robot is intended to move by attaching cables to specific points of the soft robot and pulling these cables. This method has lower bending stiffness compared to other methods and is commonly used in continuum soft robots. The tendon-driven actuation method is directly supported by the SOFA simulation program through a dedicated library.

In the generated model, four symmetric cables are placed and fixed to the end points of each arm. It is envisioned to use a motor with a pulley attached to the end of each cable. By rotating the motors, it is possible to pull and release the cables wound

around these pulleys. The control of cable pulling can be done in two different ways using the library available in SOFA: displacement control and tension control.

Measurement of tension on the cables can be achieved in two ways in the actual system. These methods involve adding a force/torque sensor to the system or monitoring the current drawn by the motor. The force/torque sensor should be attached to the cable and continuously acquiring the measurements. Monitoring the current drawn by the motor implies measuring the torque produced by the motor. Knowing the torque produced by the motor and considering that the cable is connected to the motor through a pulley, the force on the cable can be calculated. When the control of motor rotation angle and thus the amount of pulley rotation is measured, the displacement of the cable can also be determined.

Measurement of cable tension and displacement is possible in the SOFA simulation. These two values are interrelated in the simulation. In other words, when one of these values is provided as input, the program can output the other value.

3.3. Information Conveying Between Operator-Master-Slave

As mentioned before, the main task of the master system is to enable the operator to manipulate the arms of the slave system. Additionally, it should provide the operator with a sense of interaction with the operational environment of the slave system, in this case, the simulation environment. In the system at hand, the operator manipulates the slave system through haptic devices and perceives haptic feedback. The information conveyed between the operator and the slave system is presented in Figure 3.7.

Due to the tendon-driven actuation method, the necessary manipulation information is obtained from the operator through joint positions (angles) of the haptic device. The information received by the haptic device is transmitted to the slave system as cable displacements. The information from the slave system to the haptic device is in the form of torque which is calculated from the force after/during the interaction. This mapping of information between the master and the slave is explained in Section 4.1.1.

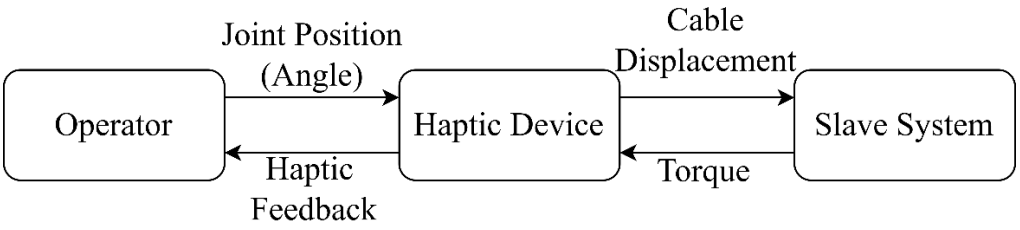

Figure 3.7. Information conveyed between the operator and the slave system

The limitation in the information flow between the haptic device and the slave system arises from the kinematic dissimilarity between the master system and the slave system. Generally, in such cases, the mapping between two systems (master and slave) is achieved in the task space. This mapping would require the kinematic models of both systems. However, since the slave system is a soft robot arm, instead of developing a computationally expensive (finite-element based) inverse kinematics model, the motion of the haptic device axes is mapped to the change of the cable lengths. The detailed mapping information is given in Section 4.1.1.

The information received by the haptic device from the slave system can be in the form of force or torque. Force information can be measured, interpreted, and transmitted to the haptic device by using a force sensor located at the end of the arm to capture the interaction between the slave system and the operating environment. However, in this thesis, the tension in the cables is used to formulate the force feedback information. This method is discussed in detail in Section 4.1.2.

CHAPTER 4

IMPLEMENTATION OF THE TELEOPERATION OF HARDWARE-IN-THE-LOOP SYSTEM

The teleoperation system is composed of two main parts as master system and slave system which is simulated in a hardware-in-the-loop simulation. The master system consists of two haptic devices and a computer. The haptic devices are connected to the computer which also runs the SOFA simulation for generating the slave system in the master side. The slave system consists of soft robot arms modeled in SOFA simulation environment which is running on a separate slave system computer. In Figure 4.1, a simplified schematic of the teleoperation system is shown. The sub-systems are further investigated in their regarding sections.

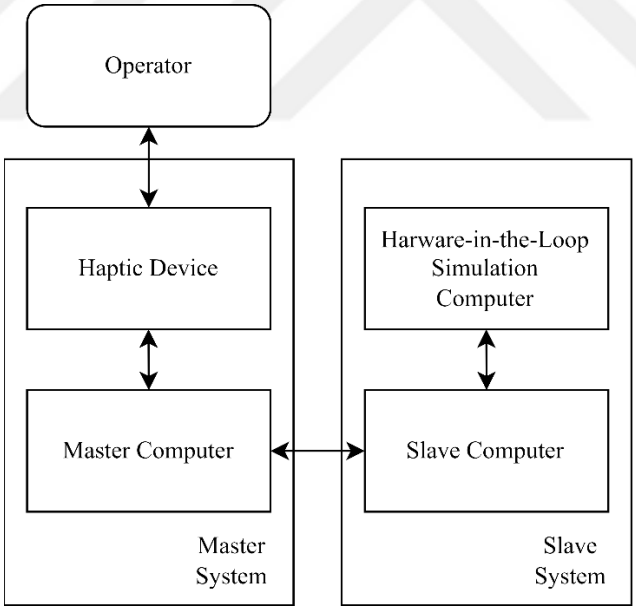

Figure 4.1. Schematic of the teleoperation system running with HIL simulation

4.1. Master System

The master system is where the operator interacts with the HIL simulation of the slave system through two haptic devices and a graphical user interface. A proxy of the

slave system, i.e., four soft manipulator arms are simulated in a virtual environment. The soft manipulator arms are named by their duty. Two of these arms are named task arms, one of them is named light arm and the other one is camera arm. Manipulation of each robot arm is achieved through the displacement of four cables attached to each soft arm. The cable displacements are obtained through the position information received from the operator via the haptic device. The master computer is responsible for acquiring the joint angles from haptic devices, interpreting them as cable lengths (displacements) and use them for abstracted simulation the slave soft arms. Also the transmission of the necessary information for the graphical user interface is done through ROS (Robot Operating System). ROS seamlessly integrates with SOFA. The master computer feeds the SOFA simulation with haptic device position readings. Haptic feedback generated in the SOFA simulation, and all the required information for the graphical user interface are sent through ROS communication which effectively addresses any potential synchronization issues. Schematic of the master system is shown in Figure 4.2.

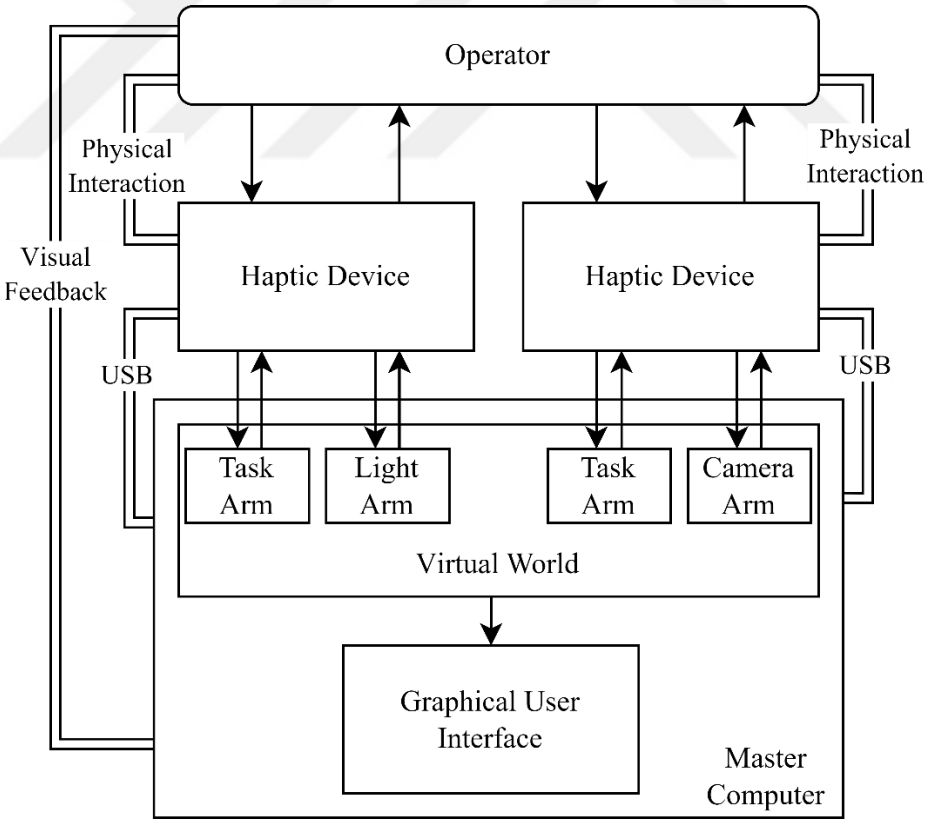

Figure 4.2. Schematic of the master system

Figure 4.3 shows a screenshot of the SOFA simulation running on the master computer. Note that this screenshot does not depict the image captured from the camera

attached to the camera arm. It is a manually selected camera position within the simulation to showcase the arm configurations and the scene created. The scene illumination and camera positioning are explained in detail in the subsequent section.

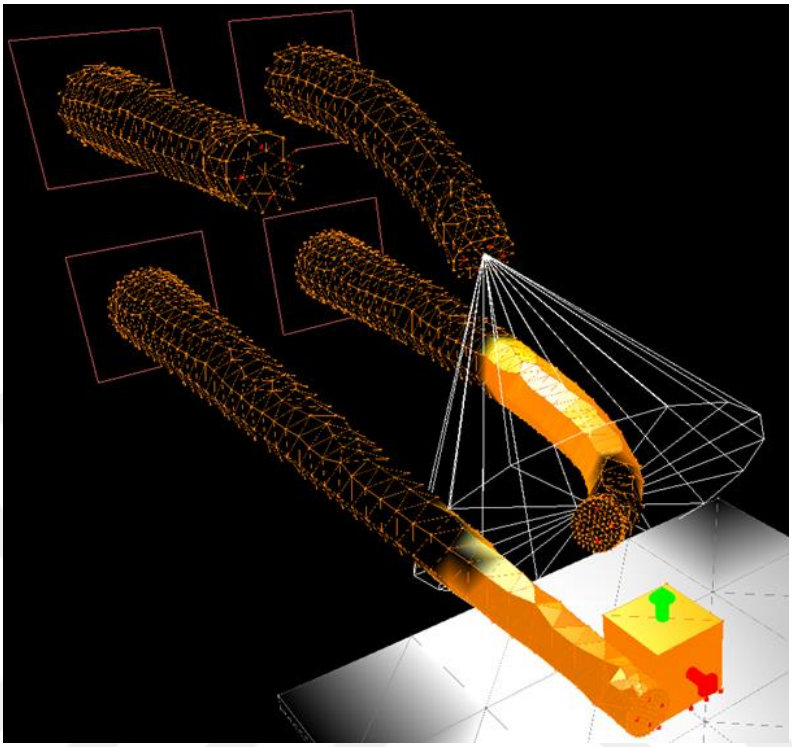

Figure 4.3. SOFA simulation scene

4.1.1. Cable Displacement Control with Haptic Device

The main challenge arises from the kinematic dissimilarity between the master manipulator and the slave arms. The master system consists of haptic devices with six degrees of freedom, while the slave system employs the previously mentioned soft robot arms.

Although it was mentioned earlier that a soft robot has infinite degrees of freedom, the modeling and actuation methods have resulted in soft robot arms having finite degrees of freedom. Firstly, the choice of silicone-based rubber material for the arms, which has a Poisson's ratio close to 0.5, makes the material incompressible, thereby restricting the soft robot arm's degrees of freedom. Furthermore, the arrangement of cables on the soft robot arm further constrains its movement. As a result, each soft robot arm in the slave system has essentially two degrees of freedom.

The mapping between the cable displacements in the soft robot arms to the positions of the haptic devices is achieved using the first and second joint angles (Joint-1 and Joint-2) of the haptic device, as shown in Figure 4.4. Only two degrees of freedom from the haptic devices' six degrees of freedom are utilized in this mapping process.

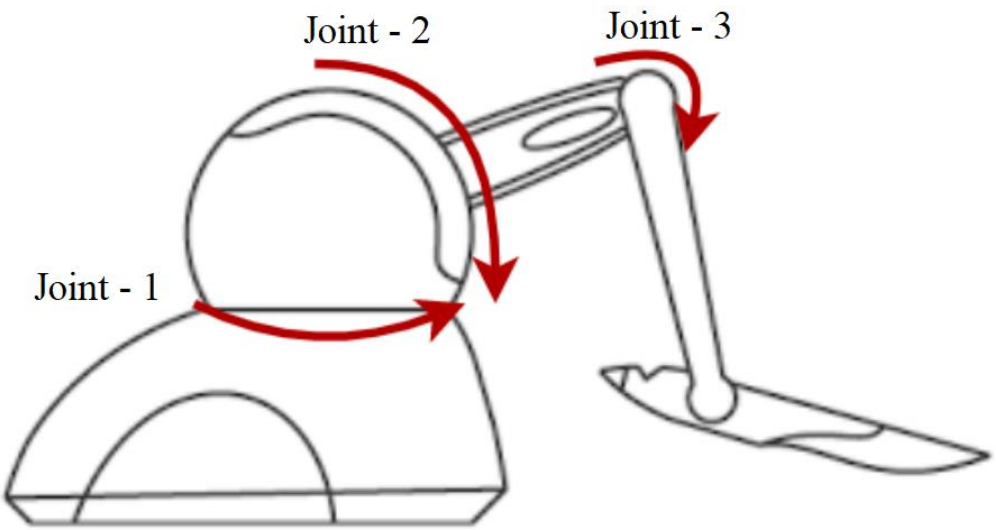

Figure 4.4. Haptic device joint definitions
(Source: Emet, 2022)

The cables on the soft robot arms have maximum and minimum values, which represent the limit values of cable length. These values are the same for cable pairs on arms of the same length. The symbols d_1, d_2, d_3, and d_4 represent the average lengths of the first, second, third, and fourth cables on an arm, respectively. The average length of a cable is calculated as the average of its maximum and minimum lengths. The average cable lengths matrix, denoted as $\boldsymbol{d}$ matrix, shown in (4.1), is composed of these four elements. The symbols with subscripts max and min are used to represent the cable limit values.

For the 300mm light/camera arms, the average cable length is 300.07mm, and for the 600mm task arms, the average cable length is 600.14mm. Since the difference between arm length and cable length is very small, the average cable lengths are taken at the length of the corresponding arm.

Cable displacement represents the change in average cable length. The two joint angles obtained from the haptic device, represented as $\boldsymbol{q}$ matrix, are multiplied by a gain matrix, represented as $\boldsymbol{k}$ to obtain the cable displacement $\Delta\boldsymbol{d}$, as shown in (4.2) and (4.3).

The values Δd_1, Δd_2, Δd_3, and Δd_4 represent the displacements of the first, second, third, and fourth cables on the arm, respectively. The values q_1 and q_2 represent the angles of the first and second joint angles of the haptic device in radians.

$$\boldsymbol{d} = \frac{\boldsymbol{d}_{min}+\boldsymbol{d}_{max}}{2} \tag{4.1}$$

$$\Delta\boldsymbol{d} = \boldsymbol{k}\,\boldsymbol{q} \tag{4.2}$$

$$\begin{bmatrix}\Delta d_1 & \Delta d_2\\ \Delta d_3 & \Delta d_4\end{bmatrix} = \begin{bmatrix}k_1 & 0\\ 0 & k_2\end{bmatrix}\begin{bmatrix}q_1 & -q_1\\ q_2 & -q_2\end{bmatrix} \tag{4.3}$$

In this context, cable displacement directly represents the amount of cable being pulled or released. A negative cable displacement value indicates that the cable is pulled, i.e., it reaches a value lower than the average, while a positive cable displacement value indicates that the cable is released, i.e., it reaches a value higher than the average. The first and second cable displacements are in opposite directions to the third and fourth cable displacements, respectively. In other words, while one of the pairs is being pulled, the other pair is released by the same amount. This is achieved by multiplying the second column of the $\boldsymbol{q}$ matrix, representing the joint angles, by a negative sign, as shown in (4.3).

The joint angles q_1 and q_2 (measured in radians) of the haptic device vary between positive and negative values. In other words, the angle value of the two joints used in the haptic device passes through zero and changes sign. Figure 4.5 illustrates the initial position of the haptic device, where the stylus on the device is placed in its holder. The movement of the first joint is symmetric with respect to the initial position, and the initial position corresponds to a zero position for the cable displacements. In other words, in the initial position, the cable average lengths are equal to the arm length. However, the movement of the second joint is not symmetric with respect to the initial position. Therefore, a fixed angle value, θ, is added to create a zero position for the cable displacements in the middle of the movement limits of the second joint. As a result, Equations (4.3) have been rearranged as (4.4). The numerical values for $\bar{k}$ and θ are shown in (4.5).

Figure 4.5. Haptic device initial position

(Source: 3D Systems)

$$\begin{bmatrix} \Delta d_1 & \Delta d_2 \\ \Delta d_3 & \Delta d_4 \end{bmatrix} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} q_1 & -q_1 \\ q_2 + \theta & -q_2 + \theta \end{bmatrix} \tag{4.4}$$

$$\begin{bmatrix} \Delta d_1 & \Delta d_2 \\ \Delta d_3 & \Delta d_4 \end{bmatrix} = \begin{bmatrix} 60 & 0 \\ 0 & 60 \end{bmatrix} \begin{bmatrix} q_1 & -q_1 \\ q_2 - 0.8176 & -q_2 + 0.8176 \end{bmatrix} \tag{4.5}$$

4.1.2. Haptic Feedback

The force perceived by the operator through the haptic device in the system is generated in the virtual environment that represents the slave environment. Haptic feedback can be generated in two ways. The first method is using haptic rendering to create a force when the virtual counterparts, i.e., the arms representing the slave system, interact with the virtual world. The haptic rendering is used to model this force. The second method is to provide the operator with the sensation of cable tension. Since the cable displacement is the actuation method of the robot and the tension is already calculated in SOFA, the preferred method, in this case, is to provide the operator with the cable tension as haptic feedback.

Like cable displacement and haptic device joint position, cable tension is also transmitted to the haptic device through the corresponding joint variables. As the actuation method of the robot arm is displacement and the tension is already calculated in SOFA at each time step, the calculated tension is used for haptic feedback. Unlike haptic rendering, which relies on a spring-damper system, the calculated tension is derived from the properties of the model, its interaction with the environment, and environmental variables. This method is simpler compared to haptic rendering. However, even when the arm does not interact with the virtual world, there will be cable tension due to the material properties of the arm, resulting in haptic feedback. Especially as the arm approaches its limit values, this feedback increases. This means that the operator feels haptic feedback at every position except the zero position.

When the arm encounters a stationary object, it stays on top of the object. However, since the actuation method is cable displacement, the arm tries to move in the direction of the motion, but it cannot move inside the object. This increases the tension on the cables and allows the operator to feel the surface. However, this method provides less realistic haptic feedback compared to haptic rendering. In haptic rendering, the force acting on the arm is calculated directly, while here, this force is indirectly perceived through the tension resulting from the cable displacement.

4.1.3. Camera and Light Placement in SOFA Simulation

The two shorter arms in the slave system carry a light and a camera. The camera and light modules available in SOFA are used for this purpose. Figure 4.6 shows a simulation image taken from the camera mounted on the camera arm, and Figure 4.7 shows the illumination images from the light arm. In the simulation images, in addition to the task arms, a box, and a ground are placed to test the interaction capabilities of the arms. The white lines resembling a cone in Figure 4.7 indicate the illumination area of the light. The same lines can also be seen in the top left corner of Figure 4.6.

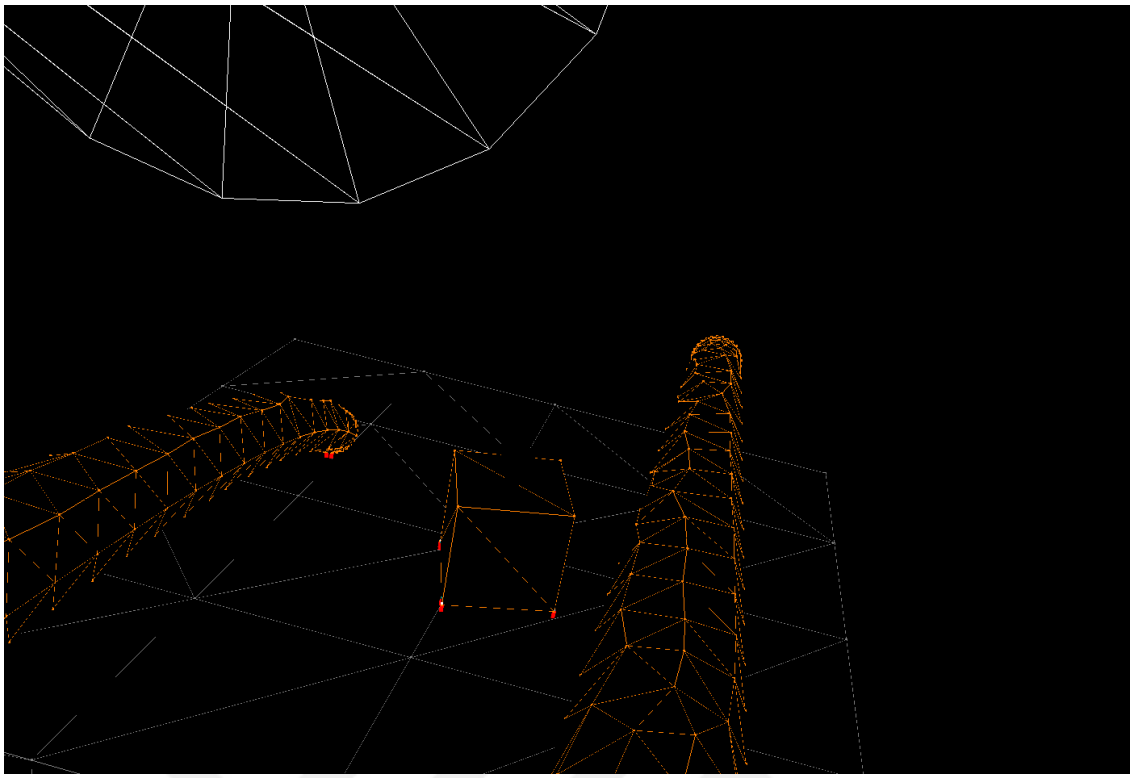

Figure 4.6. Camera display in SOFA simulation not illuminated

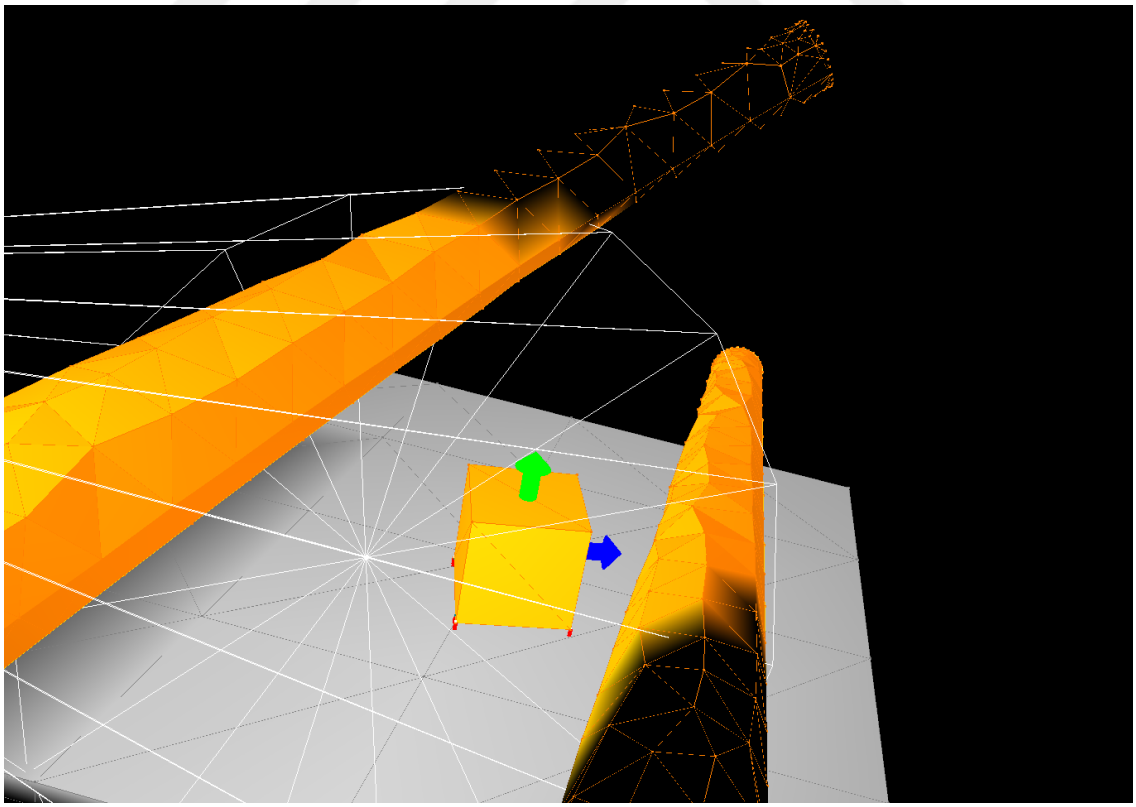

Figure 4.7. Camera display in SOFA simulation illuminated

4.1.4. Selecting Manipulation of Slave Arms Using Haptic Devices

In the system, since there are two haptic devices and four arms, the operators with their left hands will use one haptic device to manipulate the light arm and a task arm, and with their right hands will use the other haptic device to manipulate the camera arm and a task arm. Therefore, the operator needs to select the arm to manipulate with the haptic device. This selection is made using one of the buttons located on the haptic device, as outlined in the flowchart of the operation presented in Figure 4.8. Four stages have been defined for the transition.

The simulation starts with the manipulation of the task arms by both haptic devices. This is the first stage. When the button on the haptic device is pressed, the position of the task arm at the instant when the button is pressed, i.e., the cable displacements, is stored, and the system proceeds to the second stage. In the second stage, there is no actively manipulated arm. When the button on the device is pressed again, the manipulation of the light or camera arm begins, and the system enters the third stage. Pressing the button again stores the position of the light or camera arm at the instant when the button is pressed, i.e., the cable displacements, and systems enter to the fourth stage. In the fourth stage, as in the second stage, there is no actively manipulated arm. Pressing the button again returns the system to the first stage, and the manipulation of the task arm begins.

The passive transition stages are included to ensure smooth transitions between soft arm manipulations. The problem here is that the cable displacement, i.e., position information of the task arm stored at the end of the first stage, may differ from the position of the device at the end of the fourth stage. This difference, if disregarded, can result in sudden changes in the position by the master system during the transition, which can damage the arm or cable, destabilize the haptic feedback, and introduce inconsistencies in the simulation. For example, consider a scenario without passive transition stages: in the initial, i.e., the first stage, suppose the first and third cables of the left task arm are fully extended, indicating that the arm reaches its limits as shown in Figure 4.9. When the button is pressed, the position of the haptic device will be at its corresponding position. However, since none of the cables on the light arm have moved initially, the arm will instantly try to go to the position, causing the aforementioned problems. Consequently, passive stages have been defined between arm transitions so that the position mismatch

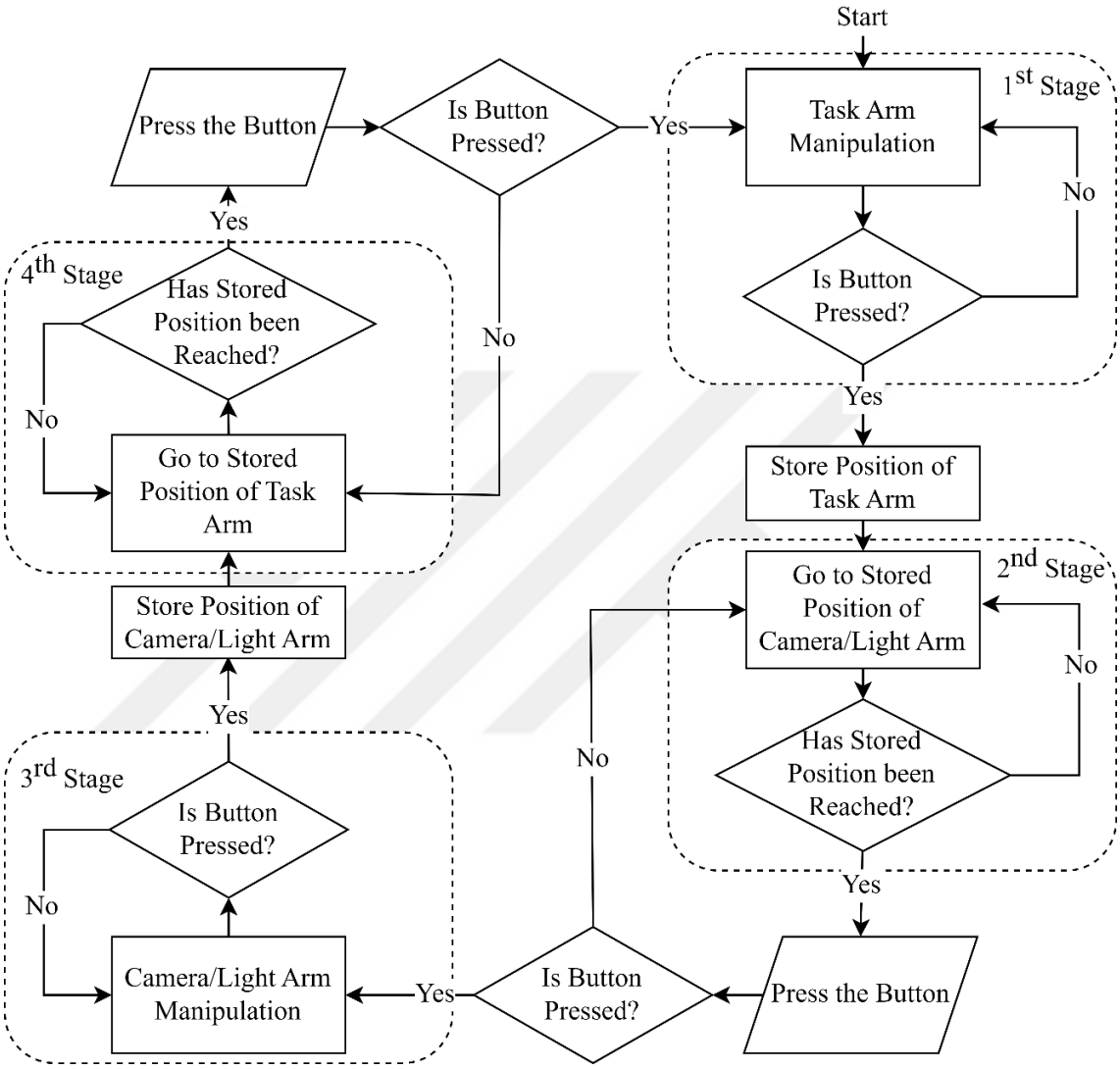

Figure 4.8. Flowchart of transition stages

problem can be handled. The stages and transition method for the other haptic device and the other two arms are the same.

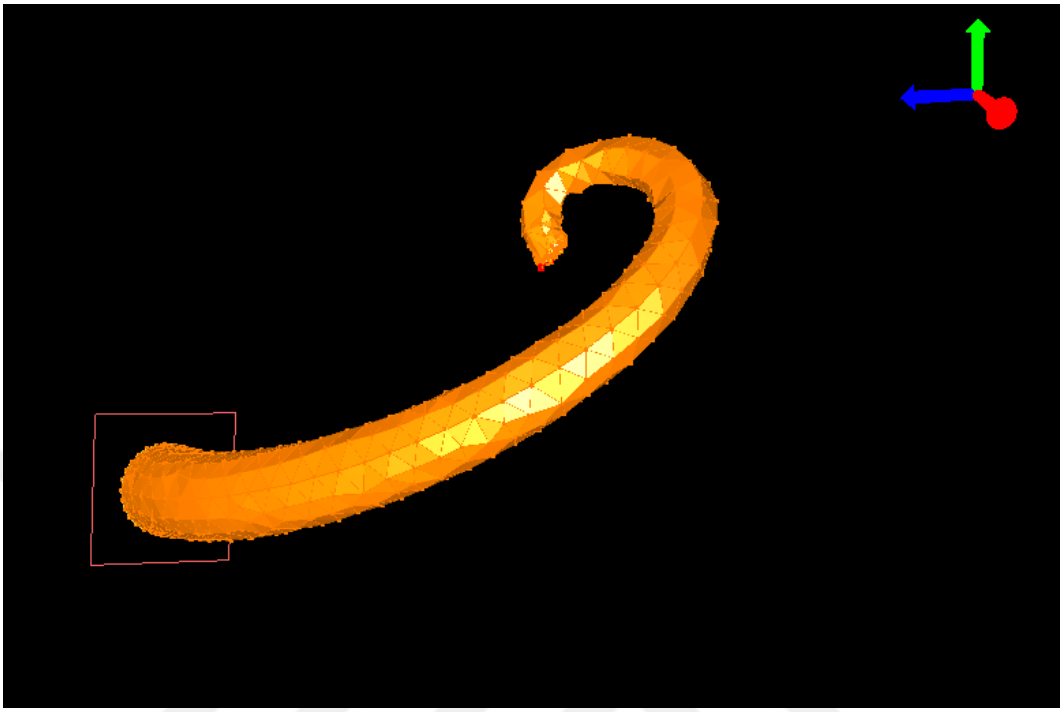

Figure 4.9. Task arm at its limit

During arm transitions, the operator can see the target arm on the simulation screen through a graphical user interface. In addition to this visual feedback, during passive transition stages, the operator is guided to the target arm through the haptic devices. This guidance is provided through haptic feedback. In other words, the operator feels that the haptic device arm is pulled/pushed towards the stored position. The magnitude of the force is inversely proportional to the distance from the stored position. In other words, as the operator approaches the stored position, they feel less force. Additionally, there is an indicator in graphical user interface to inform the operator that haptic device is close enough to the stored position and it is safe to press the button to control the designated soft arm. This interface is shown in Section 4.1.5.

4.1.5. Graphical User Interface

Various information needs have been identified to be shown to the operator on the master computer. These include cable displacements, cable tension forces, and the

operation stages of the arms. For example, if the cable displacements exceed a certain range, it could indicate that the cable has been disconnected. Similarly, if the displayed cable tension is too high, it could indicate that the arm has encountered an obstacle. Especially considering the limited bandwidth of communication, visual feedback such as underwater camera footage may have delays. However, the tension or displacement values in the cable can be transmitted much faster compared to visual data. Therefore, it is important to convey the necessary information to the operator through visually understandable indicators. This visualization is achieved using a library called "tktools" run through the Ubuntu terminal using Python.

Figure 4.10 shows the interface for cable displacements. Although there are sixteen cables in the overall slave system, only eight cable displacements are shown. This is because the operator can manipulate a maximum of two arms at a time. Therefore, the eight cable displacements on the manipulated arms are displayed. The top row represents the cables controlled by the left hand, while the bottom row represents the cables controlled by the right hand. In both rows, the first, second, third, and fourth cables are shown from left to right.

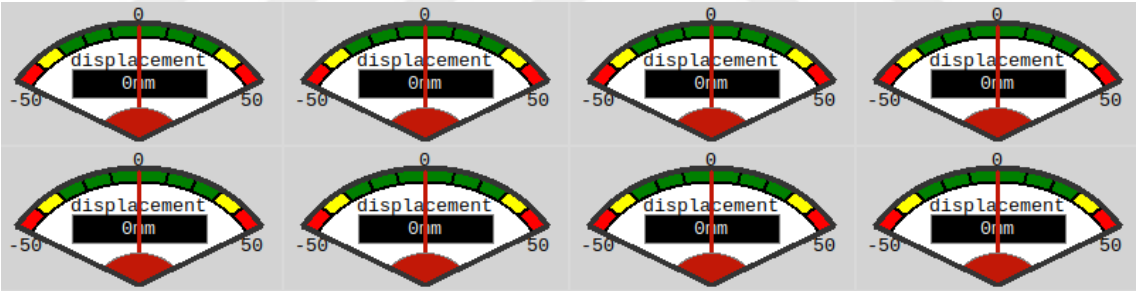

Figure 4.10. Cable displacement interface

Figure 4.11 shows the interface that conveys the cable tensions and the current operation stage to the operator. Here, the four cables on each arm are shown separately. Their arrangement is identical to the positioning of the soft arms in the simulation. Therefore, the eight cable tension indicators on the left side of the figure are for the device controlled by the left hand, and the eight indicators on the right side are for the device controlled by the right hand. The top indicators are for the light and camera arms, while the bottom indicators are for the two task arms. By observing these indicators, the operator can understand that the forces on the cables should not reach certain values or that caution should be exercised when they reach specific levels.

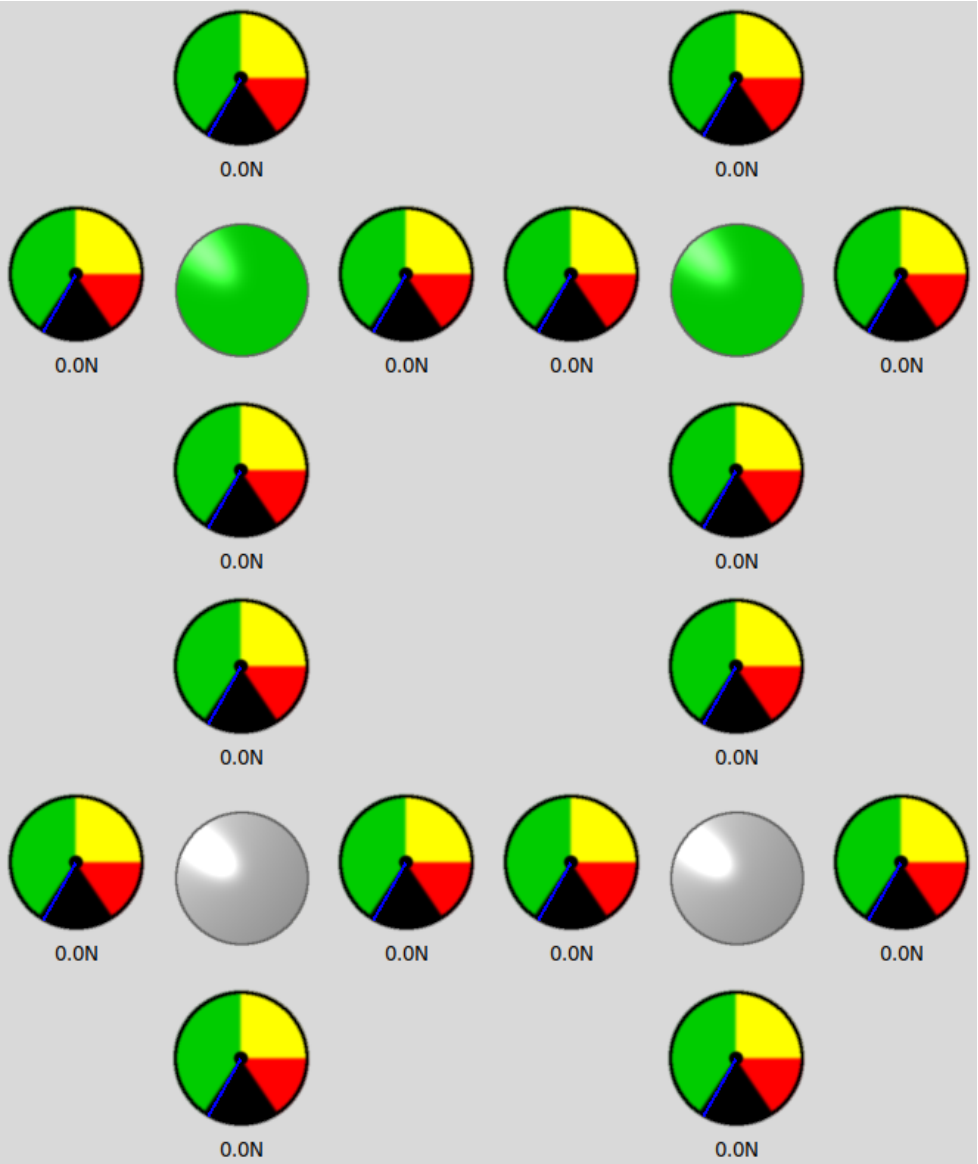

Figure 4.11. Cable tensions and indicator lights interface

In Figure 4.11, there are indicators in the middle of the cable tension indicators that change color from green to yellow to red to gray. These indicators and colors are designed to facilitate the transition between the device and the arms. To give an example based on the stages described in section 4.1.4, for the device and arms controlled by the left hand, in the first stage, the indicator located at the bottom left of Figure 4.11 is colored yellow. This indicates the actively manipulated arm. In the second stage, the indicator for the light arm, which is the upper left indicator, is colored red, indicating that the operator should not press the button yet and that there is a difference between the position of the haptic device and the position of the light arm. When this indicator turns green, it means that this difference has been eliminated and the operator can press the button for transition, thus the manipulation of the light arm can begin. The gray color represents the arm that is not being manipulated. The color representations are given in Figure 4.12. Also, a flowchart with these indicators is illustrated in Figure 4.13. The flowchart in Figure 4.13 is created based on the flowchart in Figure 4.9.

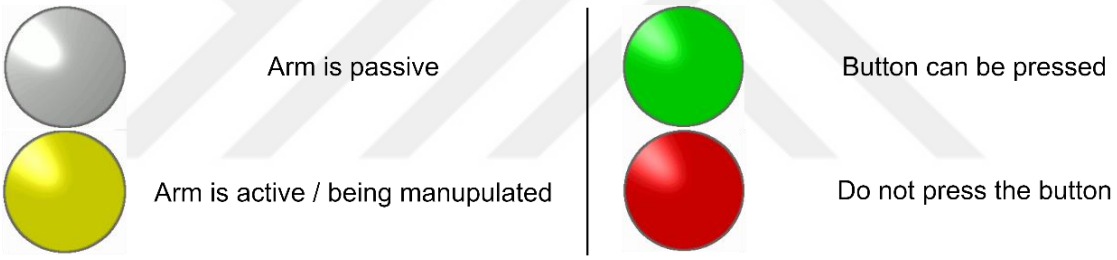

Figure 4.12. Graphical user interface light indicators

4.2. Slave System

The slave system consists of a robot computer located on the robot, a simulation computer for HIL simulation, and a data acquisition embedded system card. Figure 4.14 depicts the schematic diagram of the HIL simulation of the slave system.

4.2.1. Robot Computer

The robot computer is connected to the master system computer and the slave simulation computer running the simulation of the slave model. It is responsible for compiling and interpreting the information received from the master system as if it were

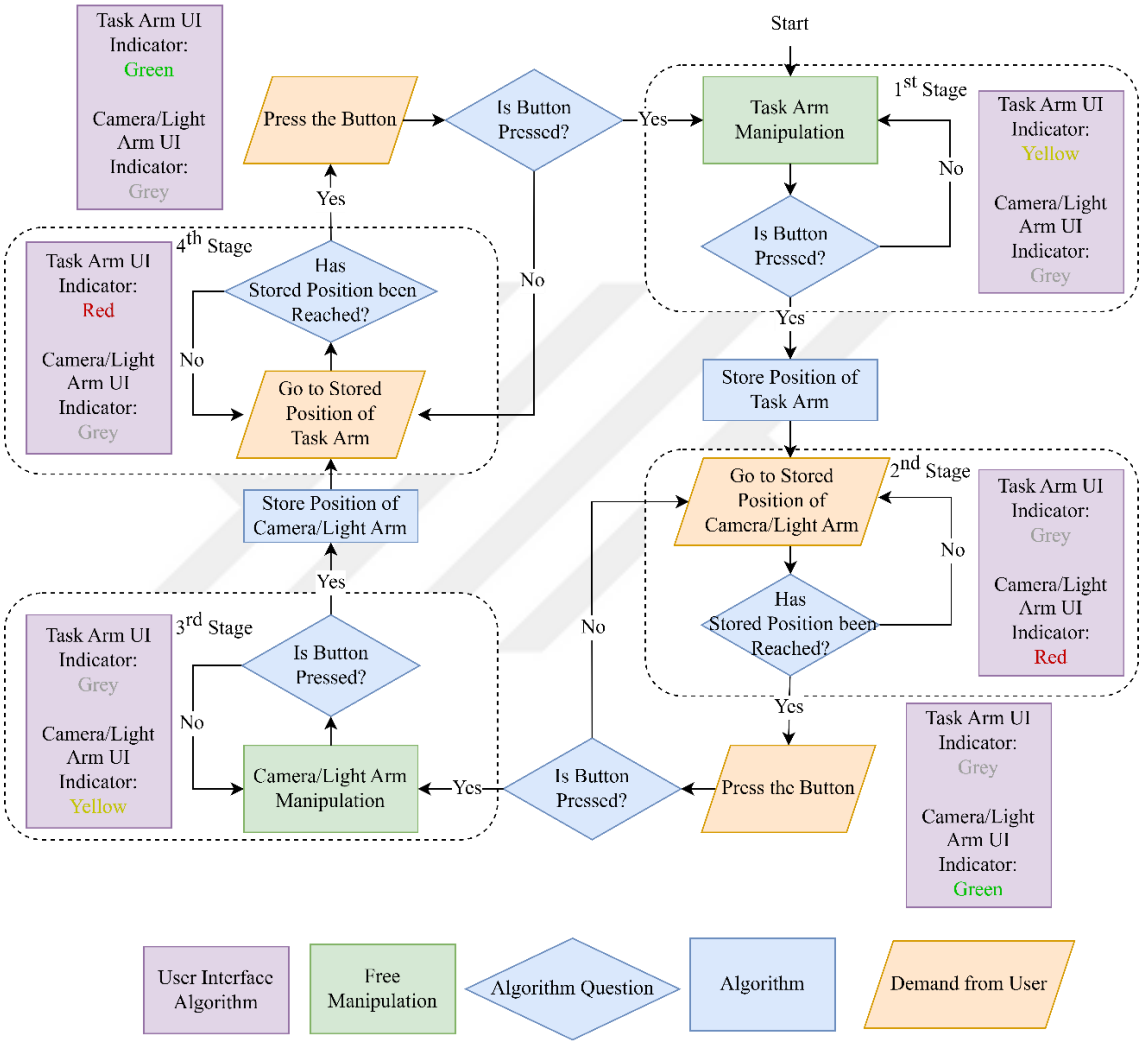

Figure 4.13. Flowchart of transition stages with user interface information

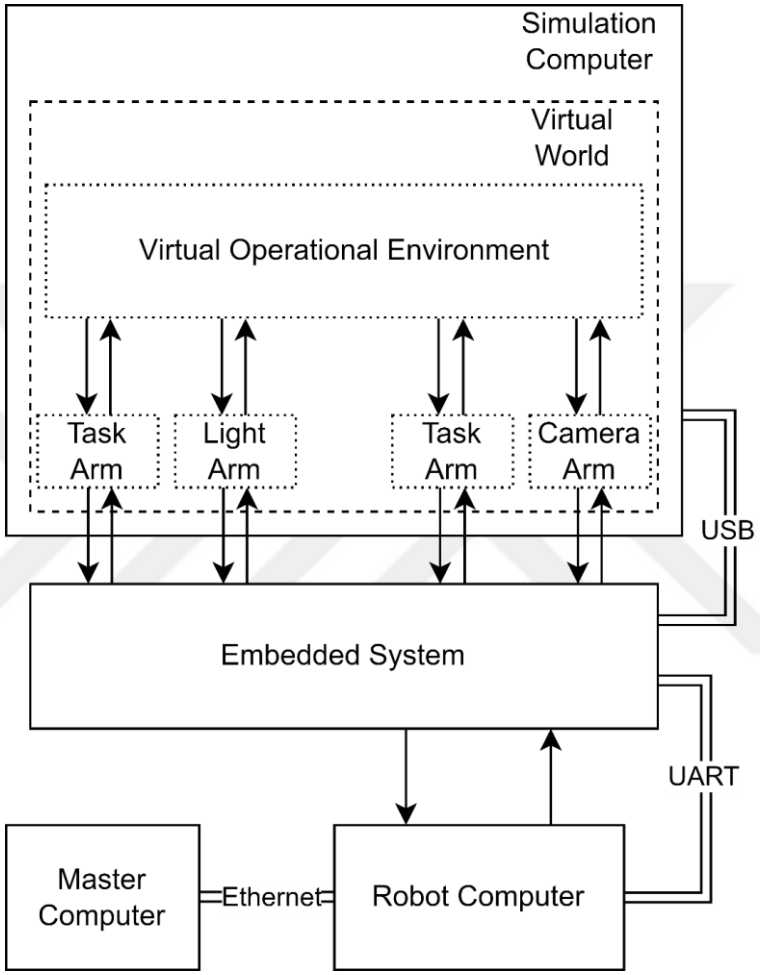

Figure 4.14. Slave system in HIL simulation

to be sent to real slave hardware. Additionally, it collects and interprets sensory data planned to be obtained from the actual slave system to be sent to the master system computer.

The Nvidia Jetson Nano development board is used as the robot computer. Figure 4.15 depicts this computer board. It is called a computer board or development board because, as can be seen from the figure, it can communicate with various hardware components. For example, it can generate and read communication protocols like UART, SPI, and I2C without the need for additional interfaces. Moreover, since it runs a Linux-based operating system, it can also run ROS. The ability to run ROS ensures communication between its internal modules and synchronization with the master system computer, avoiding any synchronization issues.

Figure 4.15. Nvidia Jetson Nano
(Source: Nvidia)

In the implemented system, the robot computer is connected to the master system computer via an Ethernet cable. It is not directly connected to the simulation computer. It communicates with the embedded system card via UART. This connection between the robot computer and the embedded system card is the only communication in the entire system where ROS is not used.

4.2.2. Simulation Computer

The simulation computer is the computer where the HIL simulation of the virtual world is running. The SOFA simulation software is used for simulating the soft arm models. Interactions of the soft arms with their environment in the virtual world take place here and are sent to the master system computer through the robot computer. It communicates with the embedded system card via the USB protocol. It interprets the low-level commands received from the robot computer. All communications, including this communication, are done through ROS.

In Figure 4.16, the SOFA scene running on the simulation computer is shown. No lights or cameras are used in this scene. The camera of the scene can be freely controlled by the operator using a computer mouse. Also, the arrangement of the arms in the scene is the same as the simulation running on the master system. The reason for selecting this scene is to visualize the movement of the robot arm. Additionally, the four red dots visible at the ends of the arms indicate the attachment points of the cables.

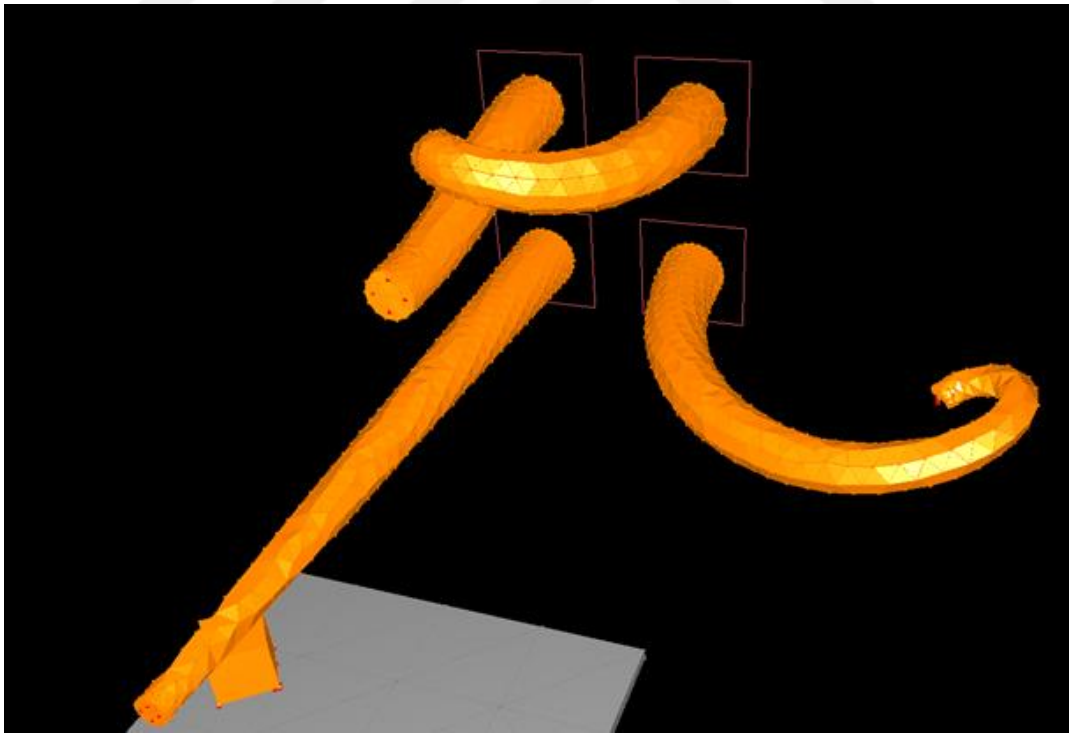

Figure 4.16. SOFA scene of simulation computer

4.3. Overall System Architecture

The hardware representations and communication protocols of the implemented system, which include the teleoperation system and HIL simulation systems, are shown in Figure 4.17. In Figure 4.18, a picture of the implemented system is presented.

The screen view for the master computer is composed of indicators of the graphical user interface and an abstract view built with the information given by the simulation computer. The abstract view is part of the virtual world scene that is re-generated in the master computer. In the virtual world re-generated in the master computer, objects are represented by a geometrical shape like a cube or a cylinder. The actual slave system, in this case represented by the HIL simulation, is foreseen to have time delayed communication channel. Although there are no communication problems in the implemented system, since the aim is to simulate the real system conditions, abstraction of the view is used in re-generating the slave system environment.

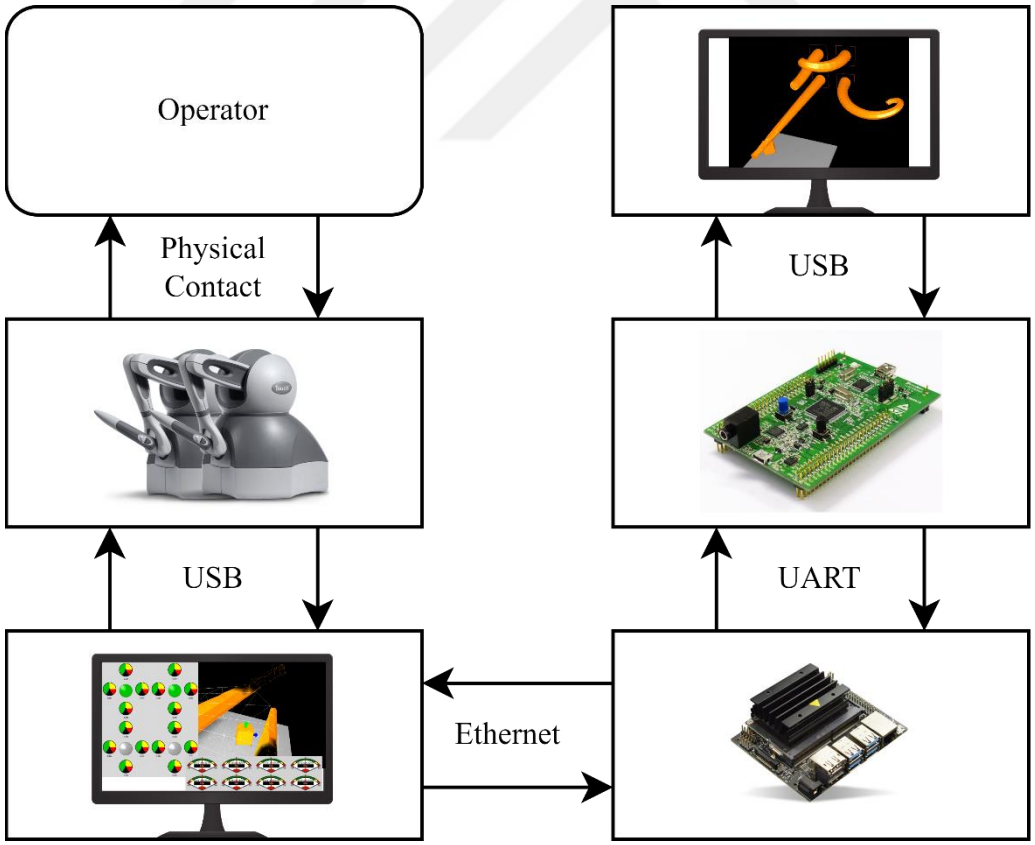

Figure 4.17. Overall system schematic representation

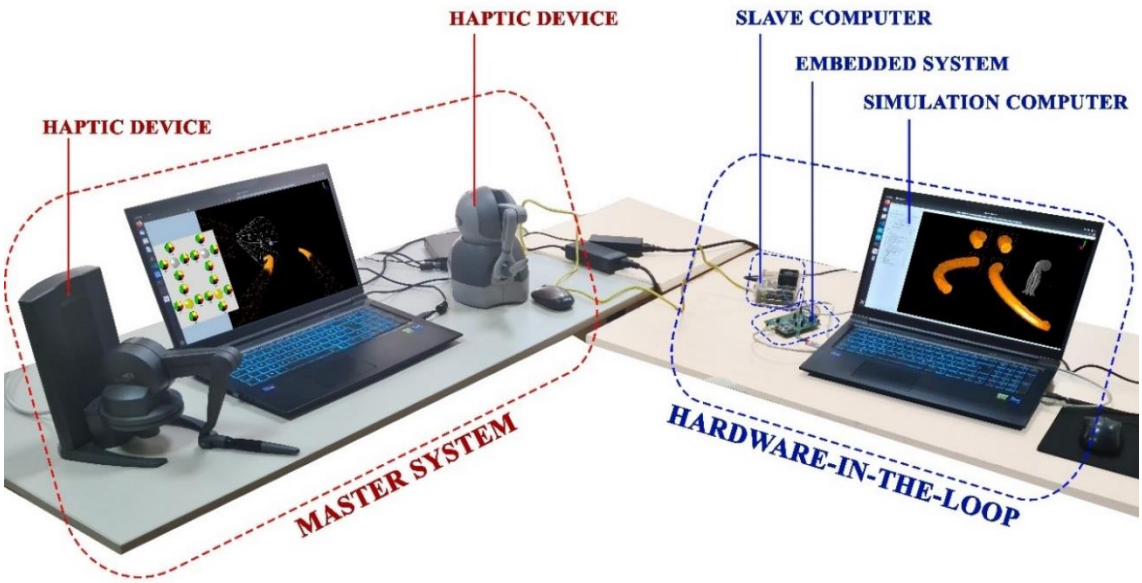

Figure 4.18. Overall system picture

In Figure 4.19, the abstract view from the master computer and from the camera is presented. The grey cube represents an object to be manipulated. In Figure 4.20, the same scene from the simulation computer is presented. The cube in Figure 4.19 is the abstraction of the octopus in Figure 4.20.

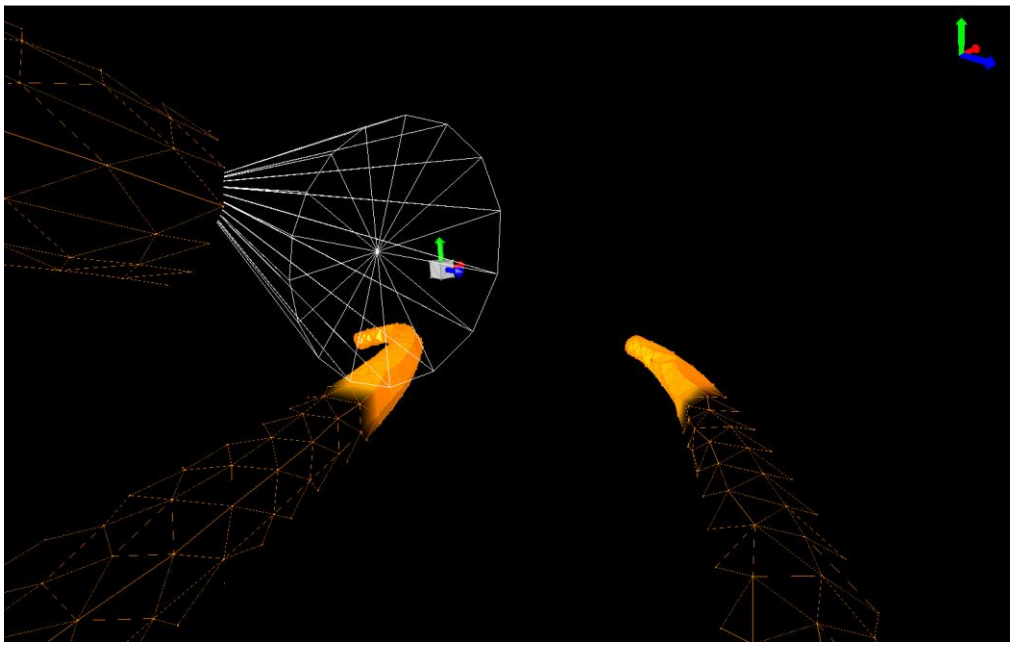

Figure 4.19. Abstract view from master computer

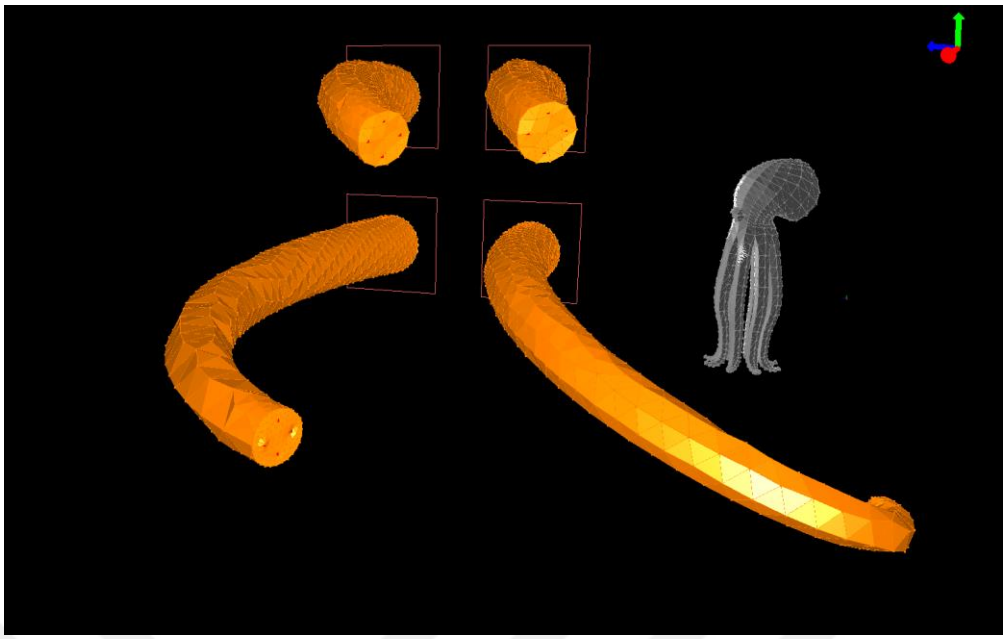

Figure 4.20. View from simulation computer

Schematic representations of the implemented systems, communication methods, and interactions are presented in Figure 4.21.

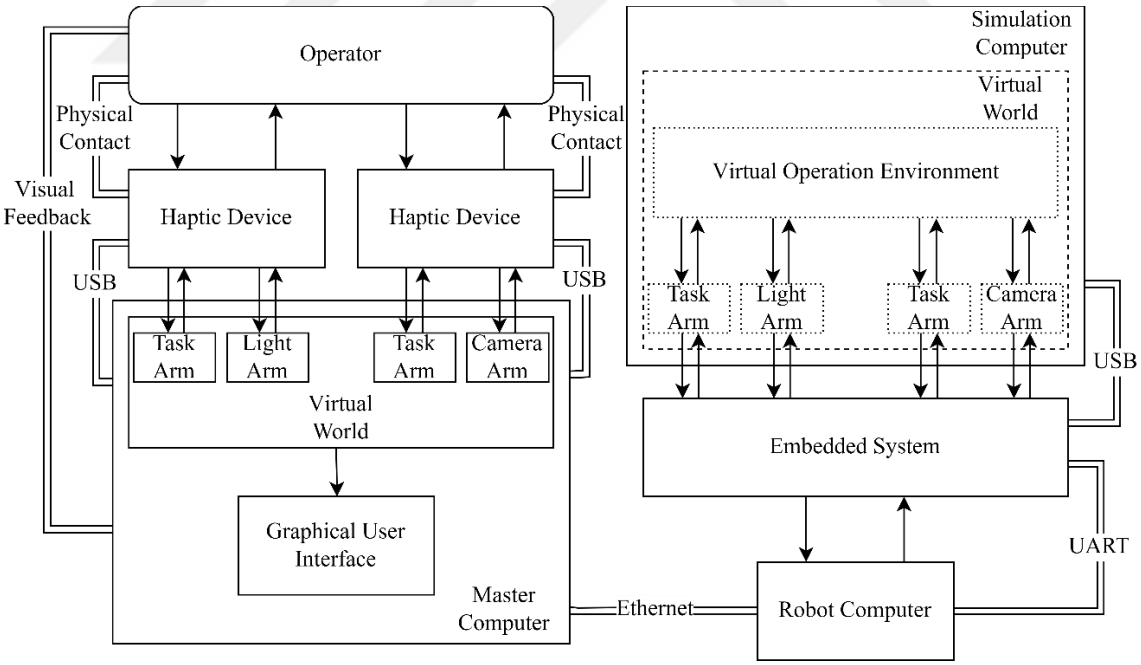

Figure 4.21. Overall system schematic

The sub-systems and information packages between the sub-systems that are running inside the computers in the master system and in the slave system are provided in Appendix.

CHAPTER 5

EXPERIMENTS & RESULTS

The implemented system is a soft real-time system. Considering the human reaction time, the frequency of real-time system is selected to be 10 Hz (Park et.al., 2022). The experiments were conducted to validate the requirement. Also, user experiments were conducted to evaluate the ease of use of the implemented teleoperation system in two modes, haptic enabled and haptic disabled. The efficiency of implemented teleoperation and the impact of haptic feedback was analyzed via experiments conducted by volunteer human test subjects. The analyses are carried out using the feedbacks from the test subjects taken by a questionnaire and recorded task completion times.

5.1. Real-Time Validation Experiments

Two different experiments are conducted to validate real-time operation of the implemented teleopration system. The first one is to determine if additional software is needed. The experiment results are used for comparing the performance of the interrupts generated. The second experiment is to validate the implemented real-time system.

5.1.1. RTAI versus ROS Interrupt Experiment

This experiment compared the performances of "Real Time Application Interface" (RTAI), which was installed on a Linux-based Ubuntu, and the "Robot Operating System" (ROS). The experiment was conducted on a simulation scene running with ROS.

For this comparison, the experimental setup was prepared as follows: RTAI and ROS were programmed to generate interrupts at specific intervals. These generated interrupts were then interpreted using code written in ROS, and data collection was performed with the same code. In this code, when an interrupt notification was received, a random command was generated. The generated command could be used as a driver

command for a simulated car running on ROS, and the transmission of this data to the simulation was ensured. Here, the meaningfulness of the sent data and how well it drove the car was not measured. Instead, apart from time measurement, no data was recorded regarding what happened in the simulation. After this command production and transmission, the elapsed time was recorded independently of what the simulation did with the sent data. This recording process was carried out using libraries in ROS.

The experiment consisted of two stages. In the first stage, interrupt generation was tested to run once every 1 ms, i.e., at a rate of 1000Hz. Except for the two programs generating and interpreting interrupts, no other programs were run in the background, and this stage was named "Control." In the second stage, in addition to the first stage, various programs were opened in the background, including the simulation where the produced and transmitted data would operate, to stress and destabilize the computer. After this addition, the computer couldn't keep up with the interrupts generated every 1 ms. So, the results of the interrupt generation interval of 1 ms are discarded. The data collected and processed after these two experiments are shown in comparison in Table 5.1.

Tablo 5.1. Comparison of RTAI and ROS interrupts

	RTAI Interrupt		ROS Interrupt		Difference (RTAI - ROS)	
	Control	**3ms**	**Control**	**3ms**	**Control**	**3ms**
Mean	1.5714	3.5172	1.1128	3.3684	0.45859	0.14885
Standard Deviation (σ)	0.5883	2.1987	0.3185	2.2565	0.269773	-0.05775
% of data outside 1 σ	8.4006	23.041	4.0301	27.2614	4.370484	-4.22008
% of data outside 2 σ	2.4104	5.6508	2.3605	6.1474	0.0499	-0.49663
% of data outside 3 σ	0.5511	1.2346	1.7196	1.4099	-1.16849	-0.17533
% of data outside 4 σ	0.5089	0.3673	0.6458	0.4772	-0.13694	-0.10996
% of data outside 5 σ	0.1092	0.0996	0.3797	0.1220	-0.27046	-0.02247
% of data outside 6 σ	0.1018	0.0310	0.2997	0.0501	-0.19794	-0.01915
% of data outside 7 σ	0.0943	0.0133	0.1447	0.0174	-0.05034	-0.00416
% of data outside 8 σ	0.0720	0.0089	0.0958	0.0065	-0.02385	0.002313

As seen in Table 5.1, the interrupt generated on RTAI is relatively more consistent within itself and works better with longer time steps. However, when compared to the ROS interrupt, the difference is negligibly small, especially for soft real-time applications. That is why no additional software is used for the implemented system.

5.1.2. Real-Time Validation of Implemented System

This experiment is conducted to validate the implemented system. SOFA solves the models in time steps. After every time step, the time is logged via ROS. Then the elapsed times between each step are calculated. In Figure 5.1, the results are presented. In Table 5.2 the results are given. Also in the Appendix part, the Gaussian distribution of the results can be found.

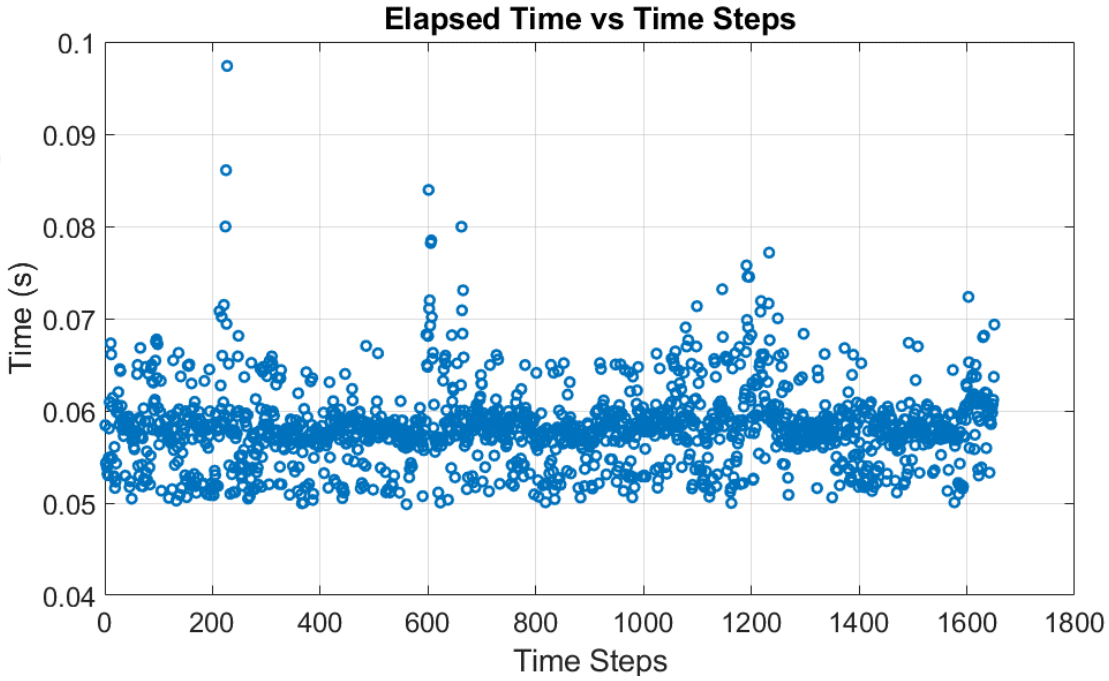

Figure 5.1. Elapsed time vs time steps

Table 5.2. Results of the real-time validation experiment

Mean (seconds)	0.0582
Standard Deviation (σ)	0.0043
% of data outside 1 σ	27.62
% of data outside 2 σ	3.45
% of data outside 3 σ	1.12
% of data outside 4 σ	0.55
% of data outside 5 σ	0.30
% of data outside 6 σ	0.18
% of data outside 7 σ	0.06
% of data outside 8 σ	0.06
% of data outside 12 σ	0

With the results presented, it can be said that the system works within the time response defined i.e., 10 Hz sampling frequency. In other words, the system is validated to be a soft real-time system.

5.2. User Experiments

User experiments are conducted to measure the relative performance of the overall system by the operator with voluntary subjects. 7 subjects participated in the experiments.

5.2.1. Experiment Procedure

Each participant was provided with a user manual explaining how the overall system works. Also, each participant is given a familiarization period on the implemented system, without the graphical user interface and haptic disabled. Based on their preference, the familiarization period was, not exceeding 10 minutes and not less than 5 minutes. Afterward, participants were given a task scenario. The scenario was conducted in two separate modes, one with haptic feedback enabled, and the other with haptic feedback disabled. Three sets of experiments were conducted with the same task. Each participant completed a total of six experiments and a familiarization period.

The task scenario is defined to utilize the full capacity of the implemented teleoperation system, including switching between all four slave arms, operating with task arms, illuminating with the light arm, and looking at different positions with the camera arm. The defined task is inspired by a real-life interaction that might occur between robotic squid and an underwater creature during the operation of the robotic squid. The creature flees to a spot when touched. After the fourth time of touching the creature, it runs away, and the operator may continue to the duty at hand.

To represent this task, a creature is abstracted as a box to the operator. The defined task is to touch this box four times; after each time, the operator must find the box in the simulation screen and touch it again. After the fourth time touching the box, it goes away, and the last task is defined as grasping two cylindrical objects placed. In Figure 5.2 the beginning of the experiment scene presented, it should be noted that this screenshot is not taken from the camera arm which the experiment view was conducted. In Figure 5.3 a

screenshot of the final part of the task, which is grasping the cylindrical object placed, is given from the view of the camera arm which subjects have experienced.

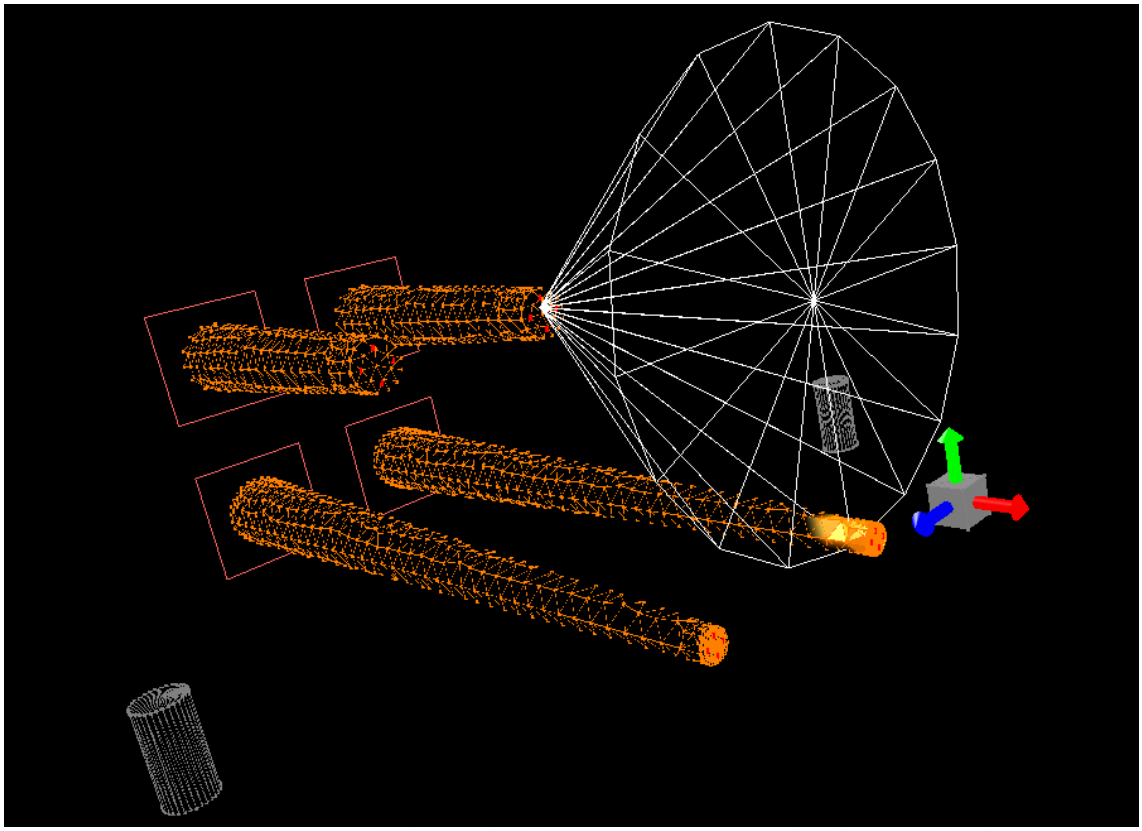

Figure 5.2. Experiment scene from the master computer

The task scenario is repeated for two modes which are haptic enabled and haptic disabled modes to complete a set. Haptic disabled mode is called the first mode, and haptic enabled mode is called the second mode.

There are three sets for the experiment. After each set is completed, the subjects waited 10 minutes before proceeding to the next set. For the first set, the subjects started and finished the task with the first mode then proceeded to the second mode. For the second set, the subjects started and finished with the second mode then repeat the task with the first mode. For the last set, subjects' preferences are asked on which mode they wanted to begin the experiment. For the subjects who had no preference, a beginning mode is randomly selected.

5.2.2. Results & Discussion

The results of the experiment were obtained based on the task completion time and questionnaire responses. All the subjects were successful at finishing the given task. The questionnaire answers provided by the subjects reflect their individual experiences and preferences. Therefore, the results are investigated and analyzed separately.

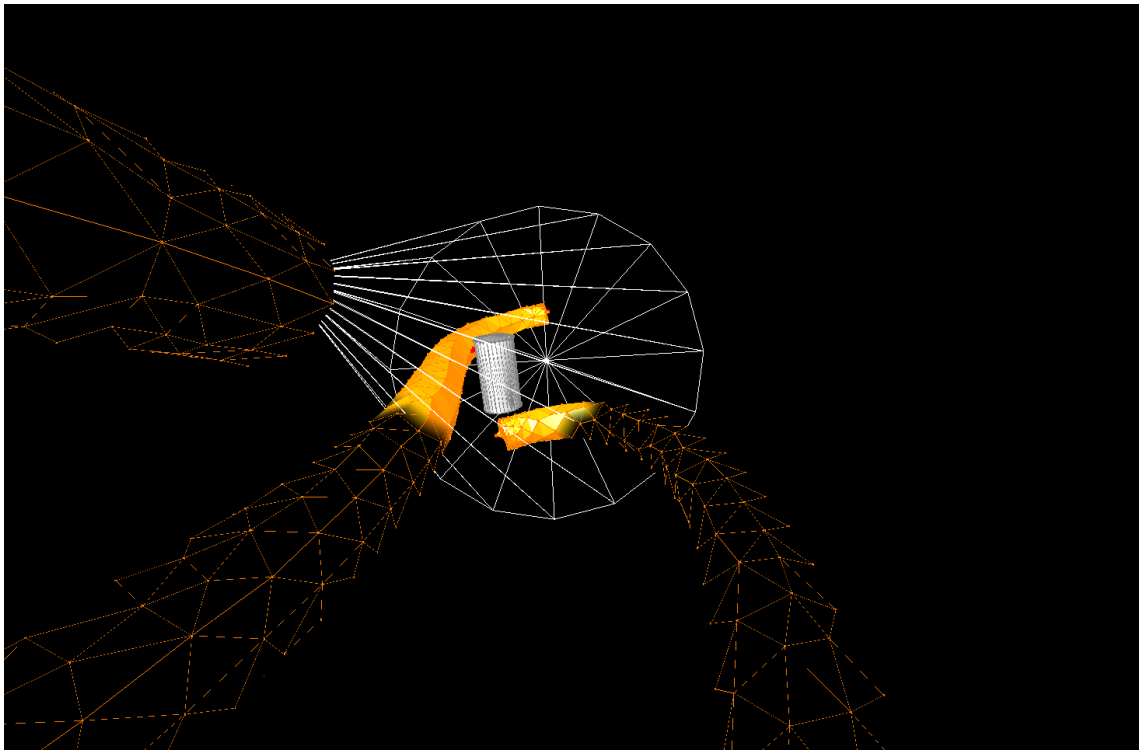

Figure 5.3. Experiment scene from master computer camera arm view

In Table 5.3, the provided questions on the questionnaire are presented. The questionnaire is divided into three sections. The first section focuses on answering questions related to Mode 1, which corresponds to the haptic disabled mode of the teleoperation system. The question numbers in this section range from 1 to 6. The same set of questions is repeated in the second section, which pertains to Mode 2 of the teleoperation system, where haptic feedback is enabled. The question numbers for this section range from 7 to 12. In the third section of the questionnaire, participants answered questions considering both Mode 1 and Mode 2, making comparisons and expressing preferences. In Table 5.4, the questionnaire answers are presented.

Table 5.3. Questions of the questionnaire

Question #	Question
1 & 7	The task required extreme mental and/or perceptual activity. (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.) (Mental/Sensory Effort)
2 & 8	The task required extreme physical activity. (e.g., pushing, pulling, turning, controlling, activating, etc.) (Physical Effort)
3 & 9	I felt extreme time pressure due to the rate at which the task elements occurred. (Considering the task as slow and leisurely or rapid and frantic) (Time Pressure)
4 & 10	I think I was not successful in doing what was asked to do and not satisfied with what I have accomplished. (Performance)
5 & 11	The task was demanding, complex and exacting. (Versus easy, simple, and forgiving) (Task difficulty)
6 & 12	I felt anxious, worried, uptight, and harassed. (Versus calm, tranquil, placid, and relaxed) (Stress level)
13	Initially, it was easier to control the robot arms with Mode 1 rather than Mode 2. (Ease of use)
14	After a certain number of repetitions, it was easier to control the robot arms with Mode 1 rather than Mode 2. (Ease of use)
15	When only manipulated robot arm change process (manipulated arm transition) is considered, I would prefer Mode 2 over Mode 1 (Preference)
16	When only manipulation of robot arm is considered, I would prefer Mode 2 over Mode 1. (Preference)
17	Generally, I would prefer to use the system with Mode 2 rather than Mode 1. (Preference)

Table 5.4. Questionnaire answers

Subject # / Question #	1	2	3	4	5	6	7
1	4	4	4	5	5	2	4
2	4	3	5	4	3	4	4
3	2	3	4	3	4	1	2
4	2	3	2	1	3	5	1
5	3	5	2	3	4	1	3
6	3	5	3	2	4	1	2
7	2	4	2	5	2	2	3
8	2	3	3	2	3	3	3
9	1	3	1	2	2	1	2
10	1	1	2	1	2	5	1
11	2	4	1	3	2	1	3
12	2	3	1	2	2	1	1
13	1	1	1	1	1	3	2
14	1	1	1	3	3	4	1
15	5	5	5	3	5	4	4
16	4	5	4	4	3	4	4
17	5	4	5	4	5	4	4

The answers to the questions vary between 1 and 5. The corresponding meaning of each number is given in Table 5.5.

Table 5.5. Answers and meanings

Answer	Meaning
1	I strongly disagree
2	I disagree
3	I am neutral
4	I agree
5	I strongly agree

For the questions from 1 to 12, the lower values indicate better results in terms of overall systems' and given task's usability and ease of use. For questions 13 and 14, the lower values indicate the system was easy to use in Mode 2, haptic enabled mode. For questions 15, 16, and 17, the higher values indicate user prefers to use haptic enabled mode, Mode 2 over haptic disabled mode, Mode 1.

When the given answers are considered, Subject 6 can be said to be an outlier. Because of two reasons. Firstly, during the experiment's first set, the subject had not considered the user manual presented and the user interface; secondly, even though the subject was successful at finishing the task in the second and third sets without any failures, the subject did not consider themselves successful.

The results can be investigated also in two sections as implemented teleoperation systems' performance in terms of user experience and the effect of haptic feedback on the system.

5.2.2.1. Results on Overall System

In Table 5.6. the questionnaire answers measure the mental/sensory effort of the subject, the physical effort of the subject, time pressure felt by the subject, subject performance, task difficulty, and stress level of the subject for the given task and overall system presented.

Mental/sensory effort: When the answers to questions 1 and 7, which measure the mental/sensory effort of the subject, are investigated, enabling haptic feedback

reduces the mental and perceptual difficulty of the task. However, the overall system, with or without haptic feedback, requires mental and perceptual effort from the user.

Table 5.6. Questionnaire answers for Mode 1 and Mode 2

Question # / Subject #	**1**	**7**	**2**	**8**	**3**	**9**	**4**	**10**	**5**	**11**	**6**	**12**
1	4	2	4	2	2	1	2	1	3	2	3	2
2	4	4	3	3	3	3	3	1	5	4	5	3
3	4	2	5	3	4	1	2	2	2	1	3	1
4	5	5	4	2	3	2	1	1	3	3	2	2
5	5	2	3	3	4	2	3	2	4	2	4	2
7	4	3	4	3	2	2	1	1	3	3	2	1
Average	4.3	3.0	3.8	2.7	3.0	1.8	2.0	1.3	3.3	2.5	3.2	1.8
Standard Deviation	0.5	1.3	0.8	0.5	0.9	0.8	0.9	0.5	1.0	1.0	1.2	0.8

Physical effort: When the answers to questions 2 and 8, which measure the physical effort of the subject, are investigated, similar to mental/sensory effort, haptic feedback reduces the physical activity of the subjects. Also, compared to the mental and perceptual effort needed, the implemented system requires less physical effort from the user.

Time pressure: When the answers to questions 3 and 9, which measure the time pressure felt by the subject during the execution of the given task, are investigated, it can be said that the haptic feedback reduces the time pressure felt by the user. Also with haptic enabled mode, Mode 2, none of the subjects felt significant time pressure while with haptic disabled mode, Mode 1, the time pressure is present for some subjects.

Performance: When the answers to questions 4 and 10 which want the subject to rate their performance are investigated, none of the subjects rated themselves below 3, for both Mode 1 and Mode 2. This indicates all the subjects think of themselves as successful.

Task difficulty: When the answers to questions 5 and 11, which measures the task difficulty, are investigated, and when compared to questions 1, 7, 2, and 8 it can be deduced that the given task was easy, but the overall system was not easy to use especially in haptic disabled mode, Mode 1.

Stress level: When the answers to questions 6 and 12, which measure the stress level during the task execution, are investigated, there is a significant difference between Mode 1 and Mode 2. Moreover, none of the subjects felt stressed, especially when haptic feedback is present, in Mode 2.

In Table 5.7. The task completion times are given for each set with Mode 1 and Mode 2. Comparing the subject with each other in terms of task completion time would not yield meaningful results on learning curve and/or ease of use of the overall system and/or the effect of haptic feedback. Because each user is an individual with different background for instance, some of the subjects are gamers while other have never played video games, some of the subjects have experience with teleoperation while some other has never used a teleoperation system. This is why comparing the percentages of each user with their results in different sets with the same modes is calculated. The calculation is done as shown in (5.1). And the results of the calculation are presented in Table 5.8. $t_{Set(i)Mode j}$ represents the task completion time of set *i* and mode *j*. The calculations are done to compare sets 1-2, 1-3, and 2-3 for both Modes.

Table 5.7. Task completion time of each subject

Time in Seconds						
Test # / Subject #	**Set 1**		**Set 2**		**Set 3**	
	Mode 1	**Mode 2**	**Mode 1**	**Mode 2**	**Mode 1**	**Mode 2**
1	206	185	165	148	168	125
2	243	142	129	90	138	64
3	415	198	277	137	246	173
4	436	179	246	236	296	194
5	626	369	324	190	252	189
7	485	407	292	261	383	281

$$\% = \frac{t_{Set(i+1)Mode j}}{t_{Set(i)Mode j}} 100 \tag{5.1}$$

For Table 5.8, the percentages are calculated and presented under set numbers. In other words, percentages compare the completion time of the later set to the previous set. The values bigger than 100% indicate that the completion time is longer than the previous.

For both Mode 1 and Mode 2 there is a significant time percentage difference between sets 1 and 2. Which indicates there is a learning curve. However, the same does not apply to the time percentage difference between 2 and 3. This might be because, as explained in the previous chapter, all the subjects had their familiarization time with the system and the subjects might have reached a plateau in their learning curve.

Table 5.8. Task completion time % between sets

Test # / Subjects	**Mode 1**			**Mode 2**		
	Set 1 - 2	**Set 2 - 3**	**Set 1 - 3**	**Set 1 - 2**	**Set 2 - 3**	**Set 1 - 3**
1	80.1 %	101.8 %	81.6 %	80.0 %	84.5 %	67.6 %
2	53.1 %	105.4 %	56.0 %	63.4 %	71.1 %	45.1 %
3	66.7 %	88.8 %	59.3 %	69.2 %	126.3 %	87.4 %
4	56.4 %	120.3 %	67.9 %	131.3 %	82.6 %	108.4 %
5	51.8 %	77.8 %	40.3 %	51.5 %	99.5 %	51.2 %
7	60.2 %	131.2 %	79.0 %	64.1 %	107.7 %	69.0 %
Average	61.4 %	104.2 %	64.0 %	76.6 %	95.3 %	71.4 %
Standard Deviation	10.6 %	19.6 %	15.5 %	28.3 %	20.0 %	23.4 %

It should be noted that repeating the same task only three times would not give a meaningful result on learning curve. In fact, giving user manual and providing the users with a familiarization time was aimed to negate the learning curve.

5.2.2.2. Effect of Haptic Feedback

As previously discussed, the results of questionnaire answers given to questions 1 to 12, show that there is a significant difference between haptic disabled and haptic enabled modes, Mode 1 and Mode 2 respectively.

In Table 5.9. the questionnaire answers for questions from 13 to 17 are presented. These questions are formulated to determine user preferences. For questions 13 and 14, lower values mean haptic enabled mode is better. On the other hand, for questions 15, 16, and 17, higher values mean haptic enabled mode is better in terms of user experience.

The answers to question 13, which is an ease-of-use question that compares initial user experience between Mode 1 and Mode 2, show haptic enabled mode is initially far superior to the haptic disabled mode. The answers to question 14, which is an ease-of-use question that compares user experience between Mode 1 and Mode 2 after repetitions, show that there is a significant difference between Mode 1 and Mode 2. However, when compared to answers given to question 13, some users after repetitions do not have a preference. This result might indicate, for some subjects, haptic feedback is a way to make learning simpler, and after the learning period, the effect of haptic feedback is negated by experience.

Table 5.9. Questionnaire answers on comparison and preference

Question # / Subject #	**13**	**14**	**15**	**16**	**17**
1	1	1	5	4	5
2	1	1	5	5	4
3	1	1	5	4	5
4	1	3	3	4	4
5	1	3	5	3	5
7	2	1	4	4	4
Average	1.2	1.7	4.5	4.0	4.5
Standard Deviation	0.4	1.0	0.8	0.6	0.5

The answers to questions 15, 16, and 17 measure user preference between haptic enabled and haptic disabled modes. Question 15 only compares Mode 1 and Mode 2 during the arm selection process. The answers given to question 15 show that subjects prefer to use haptic feedback during the arm selection process. Question 16 only compares Mode 1 and Mode 2 during the arm manipulation process. The answers given to question 16 show that subjects prefer to use haptic feedback during the arm manipulation process. When answers of 15 and 16 are compared, it is apparent subjects prefer to use haptic feedback for the arm selection more than they prefer to use haptic feedback for arm manipulation. The answers to question 17 show all the subjects prefer to use the haptic feedback enabled system over the haptic disabled system in general.

To measure the effect of haptic feedback on the completion time of the tasks, a calculation is done. Because comparing the subject within themselves would not yield meaningful results. From Table 5.7, calculations are done as shown in (5.2) to calculate the percentage difference between Mode 1 and Mode 2 in each set of tests for each subject. The results are presented in Table 5.10.

$$\% = \frac{t_{Mode2}}{t_{Mode1}} 100 \tag{5.2}$$

Table 5.10. Task completion time % between Mode 1 and Mode 2

Test # / Subject #	**Set 1**		**Set 2**		**Set 3**	
	Mode 1	**Mode 2**	**Mode 1**	**Mode 2**	**Mode 1**	**Mode 2**
1	89.8 %		89.7 %		74.4 %	
2	58.4 %		69.8 %		47.1 %	
3	47.7 %		49.5 %		70.3 %	
4	41.1 %		95.5 %		65.5 %	
5	58.9 %		58.6%		75.0 %	
7	83.9 %		89.4 %		73.4 %	
Average	63.3 %		75.4 %		67.6 %	
Standard Deviation	19.5 %		18.9 %		10.7 %	

In Table 5.7 the lower percentage values mean haptic enabled mode resulted in faster task completion durations. The results of every set indicate that enabling haptic feedback results in a shorter task completion time. In conclusion, it can be said that the haptic feedback has a positive effect on this task.

CHAPTER 6

CONCLUSIONS

A soft real-time teleoperation system is implemented for the teleoperation of soft manipulators of an underwater robotic squid in hardware-in-the-loop simulation. This study presents a haptic enabled interface of multi-master multi-slave teleoperation. Two virtual worlds are created for the system. One is for model mediated teleoperation which runs on the master system computer and the other is for hardware-in-the-loop simulation which runs on the simulation computer. Hardware-in-the-loop simulation simulates a virtual world to copy the real-world interactions that are to be conveyed to the master computer via the communication channel of the model mediated teleoperation. The real-time operation is validated with experiments. Additionally, to investigate the overall system's usability and the impact of the haptic feedback on the system user experiments are conducted.

All the subjects successfully completed the given tasks for user experiments. The results were obtained based on the task completion time and questionnaire responses. On one hand, answers to the questionnaire reflect the user experiences and preferences, and on the other hand, task completion time gives a quantitative comparison between haptic enabled and haptic disabled modes.

The answers given to the questionnaire by the subjects suggest task difficulty was harder than it should be, and the system was not easy to use. i.e., the resultant teleoperation system is hard to use, and/or the defined task is too demanding for inexperienced users. However, the answers provided for the questionnaire and the task completion time show that the haptic feedback increases the system's usability.

The subject number is only 7 for this thesis. Although the sample size is small, the results indicate a future investigation on this subject may be conducted. As future work, experiments can be extended (1) by formulating different task scenarios that may be easier to work and (2) to include at least 30 subjects to have meaningful results and a meaningful comparison between haptic enabled and disabled modes.

REFERENCES

Alise, Marc, Rodney G. Roberts, and Daniel W. Repperger. 2005. “Time delayed teleoperation using wave variables on multiple degree-of-freedom systems.” *In Proceedings of the Thirty- Seventh Southeastern Symposium on System Theory, 2005. SSST ’05.* 253–257. https://doi.org/10.1109/SSST.2005.1460916.

Amaya, Arturo, Dimuthu D. K. Arachchige, Jonathan Grey, and Isuru S. Godage. 2021. “Evaluation of Human-Robot Teleoperation Interfaces for Soft Robotic Manipulators.” In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 412–417. https://doi.org/10.1109/RO-MAN50785.2021.9515508.

Anderson, Robert J., and Mark W. Spong. 1992. “Asymptotic Stability for Force Reflecting Teleoperators with Time Delay.”*The International Journal of Robotics Research* 11, no.2: 135–149. https://doi.org/10.1177/027836499201100204.

Balch, T., and R.C. Arkin. 1998. “Behavior-based formation control for multirobot teams.” *IEEE Transactions on Robotics and Automation* 14, no. 6: 926–939. https://doi.org/10.1109/70.736776.

Bhattacherjee, Saptak, Sudipta Chattopadhayay, Vikas Rao, Sayan Seth, Sourajit Mukherjee, Aparajita Sengupta, and Subhasis Bhaumik. 2018. “Kinematics and Teleoperation of Tendon Driven Continuum Robot.” *International Conference on Robotics and Smart Manufacturing (RoSMa2018), Procedia Computer Science* 133: 879–886. issn: 1877-0509. https://doi.org/10.1016/j.procs.2018.07.106.

Csencsits, M., B.A. Jones, W. McMahan, V. Iyengar, and I.D. Walker. 2005. “User interfaces for continuum robot arms.” In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3123–3130. https://doi.org/10.1109/IROS.2005.1545434.

Dede, Mehmet I.C., and Sabri Tosunoglu. 2006. “Fault-tolerant teleoperation systems design.” *Industrial Robot: An International Journal* 33, no. 5: 365–372. https://doi.org/10.1108/01439910610685034.

Della Santina, Cosimo, Manuel G. Catalano, and Antonio Bicchi. 2020. “Soft Robots.” In *Encyclopedia of Robotics*. 489. https://doi.org/10.1007/978-3-642-41610-1_146-2.

Desai, Jadhev P., James P. Ostrowski, and Vijay Kumar. 2001. “Modeling and control of formations of non-holonomic mobile robots.” *IEEE Transactions on Robotics and Automation* 17, no.: 905–908. https://doi.org/10.1109/70.976023

Emet, Hazal. 2022. “Teleoperation of a Biomimetic Squid Robot’s Arms Via Multiple Haptic Interfaces.” MSc diss., Izmir Institute of Technology (Turkey).

Fellmann, Carolin, Daryoush Kashi, and Jessica Burgner-Kahrs. 2015 “Evaluation of input devices for teleoperation of concentric tube continuum robots for surgical tasks.” In *Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling*, 9415:411–419. https://doi.org/10.1117/12.2076741

Ferrell, William R. “Remote manipulation with transmission delay.” 1965. *IEEE Transactions on Human Factors in Electronics* HFE-6, no. 1: 24–32. https://doi.org/10.1109/THFE.1965.6591253

Ferrell, William R., and Thomas B. Sheridan. 1967. “Supervisory control of remote manipulation.” *IEEE Spectrum* 4, no. 10: 81–88. https://doi.org/10.1109/MSPEC.1967.5217126

Frazelle, Chase G., Apoorva D. Kapadia, Katelyn E. Fry, and Ian. D. Walker. 2016. “Teleoperation mappings from rigid link robots to their extensible continuum counterparts.” In *2016 IEEE International Conference on Robotics and Automation* (ICRA), 4093–4100. https://doi.org/10.1109/ICRA.2016.7487600

Frazelle, Chase G., Apoorva D. Kapadia, and Ian D. Walker. 2020. “A Haptic Continuum Interface for the Teleoperation of Extensible Continuum Manipulators.” *IEEE Robotics and Automation Letters* 5, no. 2: 1875–1882.

George Thuruthel, Thomas, Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. 2018. “Control Strategies for Soft Robotic Manipulators: A Survey.” PMID: 29297756, *Soft Robotics* 5, no. 2: 149–163. https://doi.org/10.1089/soro.2017.0007.

El-Hussieny, Haitham, Usman Mehmood, Zain Mehdi, Sang-Goo Jeong, Muhammad Usman, Elliot W. Hawkes, Allison M. Okarnura, and Jee-Hwan Ryu. 2018. “Development and Evaluation of an Intuitive Flexible Interface for Teleoperating Soft Growing Robots.” In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), 4995–5002.

Khademian, Behzad, Jacob Apkarian, and Keyvan Hashtrudi-Zaad. 2011. “Assessment of Environmental Effects on Collaborative Haptic Guidance.” *Presence* 20, no. 3:191–206. https://doi.org/10.1162/PRES_a_00044

Kubo, Ryogo, Noriko Iiyama, Kenji Natori, Kouhei Ohnishi, and Hirotaka Furukawa. 2007. “Performance analysis of a three-channel control architecture for bilateral teleoperation with time delay.” *IEEJ Transactions on Industry Applications* 127, no. 12:1224–1230. https://doi.org/10.1541/ieejias.127.1224

Lalish, Emmett, Kristi A. Morgansen, and Takashi Tsukamaki. 2006. “Formation Tracking Control using Virtual Structures and Deconfliction.” In *Proceedings of the 45th IEEE Conference on Decision and Control*, 5699–5705. https://doi.org/10.1109/CDC.2006.377187

Matsuhira, Nobuto, Makoto Asakura, and Hiroyuki Bamba. 1993. “Manoeuvrability of a master-slave manipulator with different configurations and its evaluation tests.” *Advanced Robotics* 8, no. 2: 185–202.

Minelli, Marco, Federica Ferraguti, Nicola Piccinelli, Riccardo Muradore, and Cristian Secchi. 2019. “An energy-shared two-layer approach for multi-master-multi-slave bilateral teleoperation systems.” In *2019 International Conference on Robotics and Automation* (ICRA), 423–429. https://doi.org/10.1109/ICRA.2019.8794335

Mitra, Probal, and Günter Niemeyer. 2008. “Model-mediated Telemanipulation.” *The International Journal of Robotics Research* 27, no. 2: 253–262.

Naghibi, H., M. W. Gifari, W. Hoitzing, J. W. Lageveen, D.M.M. van As, S. Stramigioli, and M. Abayazid. 2019. “Development of a Multi-level Stiffness Soft Robotic Module with Force Haptic Feedback for Endoscopic Applications.” In *2019 International Conference on Robotics and Automation* (ICRA), 1527–1533. https://doi.org/10.1109/ICRA.2019.8793584

Niemeyer, Günter Dieter. 1996. “Using wave variables in time delayed force reflecting teleoperation.” PhD diss., Massachusetts Institute of Technology.

Ouyang, Bo, Yunhui Liu, Hon-Yuen Tam, and Dong Sun. 2018. “Design of an Interactive Control System for a Multisection Continuum Robot.” *IEEE/ASME Transactions on Mechatronics* 23, no. 5: 2379–2389.

Park, Jonghwa, Dong-Hee Kang, Heeyoung Chae, Sujoy Kumar Ghosh, Changyoon Jeong, Yoojeong Park, Seungse Cho et.al. 2022. “Frequency-selective acoustic and haptic smart skin for dual-mode dynamic/static human-machine interface.” *Science Advances*, Vol 8, issue 12. https://doi.org/ 10.1126/sciadv.abj9220

Passenberg, Carolina, Angelika Peer, and Martin Buss. 2010. “Model-Mediated Teleoperation for multi-operator multi-robot systems.” In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4263–4268. https://doi.org/10.1109/IROS.2010.5653012

Peer, A., B. Stanczyk, and M. Buss. 2005. “Haptic telemanipulation with dissimilar kinematics.” In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3493–3498. https://doi.org/10.1109/IROS.2005.1545349

Robinson, G., and J.B.C. Davies. 1999. “Continuum robots - a state of the art.” In *Proceedings 1999 IEEE International Conference on Robotics and Automation*, vol. 4, 2849–2854 vol.4. https://doi.org/10.1109/ROBOT.1999.774029

Shahbazi, M., H.A. Talebi, S. F. Atashzar, F. Towhidkhah, R. V. Patel, and S. Shojaei. 2011. “A novel shared structure for dual user systems with unknown time-delay utilizing adaptive impedance control.” In *2011 IEEE International Conference on Robotics and Automation*, 2124–2129.

Sheridan, Thomas B. 1989. “Telerobotics.” *Automatica* 25, no. 4: 487–507. issn: 0005-1098. https://doi.org/10.1016/0005-1098(89)90093-9.

Sirouspour, Shahin, and Peyman Setoodeh. “Adaptive nonlinear teleoperation control in multi-master/multi-slave environments.” (2005) In *Proceedings of 2005 IEEE Conference on Control Applications*. CCA 2005. 1263–1268.

Stroppa, Fabio, Ming Luo, Kyle Yoshida, Margaret M. Coad, Laura H. Blumenschein, and Allison M. Okamura. 2020. “Human Interface for Teleoperated Object Manipulation with a Soft Growing Robot.” In *2020 IEEE International Conference on Robotics and Automation* (ICRA), 726–732.

Thuruthel, Thomas G., Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. (2018). “Control Strategies for Soft Robotic Manipulators: A Survey.” *Soft Robotics* 5, no. 2: 149–163. https://doi.org/10.1089/soro.2017.0007.

Uzunoğlu, Emre. 2012. “Position/Force Control of Systems Subjected to Communication Delays and Interruptions in Bilateral Teleoperation.” MSc diss., Izmir Institute of Technology (Turkey).

Xie, Mufeng, Cedric Girerd, and Tania K. Morimoto. 2022. “A 3-D Haptic Trackball Interface for Teleoperating Continuum Robots.” In *2022 9th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics* (BioRob), 1–8. https://doi.org/10.1109/BioRob52689.2022.9925384

Xie, Yiping, Xilong Hou, and Shuangyi Wang. 2023. “Design of a Novel Haptic Joystick for the Teleoperation of Continuum-Mechanism-Based Medical Robots.” *Robotics* 12, no. 2. issn: 2218-6581. https://doi.org/10.3390/robotics12020052.

Yang, Yana, Xinru Feng, Junpeng Li, and Changchun Hua. 2023. “Robust fixed-time cooperative control strategy design for nonlinear multiple-master/multiple-slave teleoperation system.” *Journal of the Franklin Institute* 360, no. 3: 2193–2214. issn: 0016-0032. https://doi.org/10.1016/j.jfranklin.2022.10.006

APPENDIX A

PROBABLE PROBLEMS

The only problem without a complete solution is the processing speed of SOFA. Unless otherwise specified, SOFA operates using only a single core of the computer's processor. This issue can be addressed using the "MultiThreading" plugin of SOFA. This plugin enables SOFA to utilize all cores of the computer's processor in parallel. In contrast to using a single core, this provides noticeable speed gains. However, it is not the definitive solution to the speed issue. Additionally, SOFA does not utilize the graphics card on the computer unless specified otherwise. There is a development interface called CUDA, especially used by Nvidia graphics cards, which facilitates the use of the graphics card on the computer. Despite having a SOFA plugin named "SOFACuda," there are integration issues. These integration problems stem from the fact that the simulation is written in the Python language and the plugin that provides cable actuation methods - mentioned earlier - is the source of the issue. Communication has been established with the developers of the plugin related to this problem, but as of the writing date of this report, the issue has not been resolved. In conclusion, SOFA, particularly used for visualization, is unable to utilize the graphics card on the computer.

Another encountered issue is related to the drivers of haptic devices on Ubuntu. Initially, this issue arises from the haptic device manufacturer not keeping up with the latest Ubuntu version. As of the information accessed during the writing of the report, the most recent update date for haptic device drivers is April 22, 2022, and this update is for Ubuntu 20.04 released on April 23, 2020. The latest version of Ubuntu, Ubuntu 22.04, was released on April 21, 2022. Programs used for software development, such as SOFA and ROS, follow the latest version of Ubuntu. This implies that the entire system needs to operate on older versions due to outdated haptic device drivers. Furthermore, after the latest Ubuntu 20.04 update, updated haptic device drivers are unable to run two identical haptic devices simultaneously. Among the two USB-connected and one ethernet-connected haptic devices we have, two USB-connected devices cannot be used simultaneously. Therefore, this issue has been resolved by using one USB-connected device and one ethernet-connected device.

APPENDIX B

APPROVAL FORM & QUESTIONNAIRE

İZMİR YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
FEN VE MÜHENDİSLİK BİLİMLERİ
BİLİMSEL ARAŞTIRMA VE YAYIN ETİK KURULU

BİLGİLENDİRİLMİŞ ONAY FORMU

Sizi İzmir Yüksek Teknoloji Enstitüsü, Mühendislik Fakültesi, Makine Mühendisliği öğretim üyesi Prof. Dr. Mehmet İsmet Can Dede tarafından yürütülen, "Kablo ile Eylenen Yumuşak Robot Kollarının Çoklu Haptik Arayüzlerle Teleoperasyonunda Haptik Geribeslemenin Etkisi" başlıklı araştırmaya katılmaya davet ediyoruz. Aşağıda ayrıntılı bilgileri verilen çalışmaya katılmadan önce bu formun okunması önem taşınmaktadır. Bu araştırmaya katılmak tamamen kendi iradenizle olması koşulu esasına dayanmaktadır. Araştırmaya katılmama ya da istediğiniz zaman, hiçbir sebep göstermeden ayrılma hakkına sahipsiniz. Araştırma hakkında anlamadığınız herhangi bir konuyu çekinmeden sorun. Elde edilecek kişisel bilgiler tamamen gizli tutulacak olup, anket sonuçlarından derlenecek istatistiki bilgiler proje sonuçlarının değerlendirileceği raporlarda ve bilimsel yayınlarda kullanılacaktır.

1. Çalışmanın Amacı
4 adet yumuşak robot kolunun kablo ile sürülerek çift yönlü (haptik geribeslemeli) teleoperasyon yöntemi ile kontrol sisteminin oluşturulması, oluşturulan teleoperasyon sisteminin bilgisayar ortamında simülasyonu ve oluşturulan kontrol sistemindeki haptik geribeslemenin etkisinin belirlenmesi amaçlanmaktadır. Sistem, 2 ana kontrolcü ve 1 ana bilgisayardan oluşmaktadır. Simülasyon; ana bilgisayar içerisinde modellenmiş kolların ve görev uzayının bulunduğu bir sanal dünyada yapılmaktadır. Modellenmiş 4 kolun birisinin ucuna kamera takılmış, birisinin ucuna ışık takılmış ve kalan 2 kol da tanımlı görevler için oluşturulmuştur. Operatör, ana kontrolcüleri kullanarak sanal dünyada modellenen dört yumuşak robot kolunu manipüle edecektir. Operatörden verilen görev senaryolarını tamamlanması istenecektir. Verilecek görev senaryoları, sistemin tanımlanan bütün kapasitesinin kullanılması için oluşturulmuş simülasyon senaryolarıdır. Senaryolarda operatör 4 yumuşak robot kolunu aynı anda iki tanesini aktif kullanacak olup; iki ana kontrolcüyle atıl olan diğer iki robot kolu arasında geçiş yapacaktır. Verilen görev senaryoları; haptik geribesleme etkisinin ölçülmesi için iki ayrı set şeklinde yapılacak olup, her iki sette de aynı görev senaryosu simüle edilecektir. Testler sonucunda, haptik geribeslemenin kablo sürücülü yumuşak robot kolları üzerinde etkisini ölçmek üzere 17 sorulu bir anket yapılacaktır.
2. Çalışmanın Süresi: 5 gün
3. Planlanan Katılımcı Sayısı: 7 kişi
4. Araştırmada Yapılacak Genel İşler (Sorular hakkında genel bilgi, soru sayısı, ortalama cevaplama süresi)
Katılımcıların her birine öğrenme eğrisinin göz ardı edilebilmesi için, bir kullanım kılavuzu verilecektir. Ayrıca, kullanıcının isteği doğrultusunda 5 dakikadan az, 10 dakikadan fazla olmamak kaydıyla oluşturulan simülasyon üzerinde görev senaryoları, haptik geribesleme ve grafik kullanıcı arayüzü olmaksızın alışma süresi tanınacaktır. Daha sonra kullanıcılara tanımlı görev senaryoları verilecektir. Bu senaryolar haptik geribesleme açık ve kapalı olarak 2 ayrı set olarak yapılacaktır. Bu iki setten ilkinin hangisi olacağı, yine öğrenme eğrisini göz ardı edebilmek için, rastgele belirlenecektir. Her kullanıcı için bütün testler 3 defa tekrarlanacak ve öğrenme eğrisinin gerçekten göz ardı edilip edilememeği belirlenecektir.

Figure B.1 Informed approval form for user experiments – page 1 of 2

Her kullanıcı toplamda 6 testi tamamlamış olacaktır. Her kullanıcı 5-10 dakika alışma süresi, 2-5 dakika bir test seti ve test setleri arasında 10 dakika bekleme süresi olmak üzere toplamda 31-40 dakika test için ayırması gerekmektedir. Bunun yanında 17 sorulu anketin ortalama cevap süresi 5 dakikadır. Yani kullanıcının neticede 36-45 dakikasını ayırması beklenmektedir.

Katılım Onayı:
Yukarıda yapılan açıklamaları okudum ve anladım. Araştırma hakkında yazılı ve sözlü açıklama tarafıma yapıldı, sorularımı sordum ve tatmin edici yanıtlar aldım. İstediğim zaman araştırmadan ayrılma hakkına sahip olduğum bilinci ile çalışmaya gönüllü olarak katılmayı onaylıyorum. Bu formun bir kopyası tarafıma verildi.

Katılımcının adı soyadı: _____ Tarih: __ /__ /2023

Katılımcının imzası:

Yürütücünün adı soyadı: Prof. Dr. Mehmet İsmet Can Dede

Yürütücünün imzası:

Figure B.2 Informed approval form for user experiments – page 2 of 2

Table B.1 Questionnaire for the user experiments

	1. Bölüm Lütfen her soruyu 1. Moda göre cevaplayın.					
	Puan / **Soru**	**1**	**2**	**3**	**4**	**5**
1	Görev, aşırı zihinsel ve/veya algısal aktivite gerektiriyordu. (Örneğin düşünme, karar verme, hesaplama, hatırlama, bakma, arama vb.) (Zihinsel/Duyusal Çaba)					
2	Görev, aşırı fiziksel aktivite gerektiriyordu. (Örneğin, itme, çekme, döndürme, kontrol etme, harekete geçirme vb.) (Fiziksel Efor)					
3	Görev öğelerinin oluşma hızı nedeniyle aşırı zaman baskısı hissettim. (Görev yavaş ve yavaş ya da hızlı ve çılgın olarak düşünüldüğünde) (Zaman Baskısı)					

(cont. on next page)

Table B.1 (cont.)

4	Yapmam gerekenleri yapmakta başarılı olmadığımı düşünüyorum ve/veya başardıklarımdan memnun değilim. (Performans)					
5	Görev zorlu, karmaşık ve titizlik gerektiriyordu. (Kolay, basit ve toleranslıya karşı) (Görev Zorluğu)					
6	Endişeli, kaygılı, gergin ve rahatsız hissettim. (Sakin, rahat, uysal ve gevşemişe karşı) (Stres Seviyesi)					
	2. Bölüm Lütfen her soruyu 2. Moda göre cevaplayın.					
	Puan / **Soru**	**1**	**2**	**3**	**4**	**5**
7	Görev, aşırı zihinsel ve/veya algısal aktivite gerektiriyordu. (Örneğin düşünme, karar verme, hesaplama, hatırlama, bakma, arama vb.) (Zihinsel/Duyusal Çaba)					
8	Görev, aşırı fiziksel aktivite gerektiriyordu. (Örneğin, itme, çekme, döndürme, kontrol etme, harekete geçirme vb.) (Fiziksel Efor)					
9	Görev öğelerinin oluşma hızı nedeniyle aşırı zaman baskısı hissettim. (Görev yavaş ve yavaş ya da hızlı ve çılgın olarak düşünüldüğünde) (Zaman Baskısı)					
10	Yapmam gerekenleri yapmakta başarılı olmadığımı düşünüyorum ve/veya başardıklarımdan memnun değilim. (Performans)					
11	Görev zorlu, karmaşık ve titizlik gerektiriyordu. (Kolay, basit ve toleranslıya karşı) (Görev Zorluğu)					
12	Endişeli, kaygılı, gergin ve rahatsız hissettim. (Sakin, rahat, uysal ve gevşemişe karşı) (Stres Seviyesi)					

(cont. on next page)

Table B.1 (cont.)

	3. Bölüm Lütfen her soruyu iki modu da dikkate alarak cevaplayın.					
	Puan **Soru**	**1**	**2**	**3**	**4**	**5**
13	Başlangıçta, görevi 1. Mod ile kontrol etmek, 2. Moda göre daha kolaydı. (Kullanım kolaylığı)					
14	Belirli bir tekrardan sonra, robot kollarını 1. Mod ile kontrol etmek, 2. Moda göre daha kolaydı. (Kullanım kolaylığı)					
15	Sadece robot kolları arası geçişler (robot kolu değiştirme) düşünüldüğünde 1.Mod yerine 2.Mod ile kullanmayı tercih ederim. (Tercih)					
16	Sadece robot kolu manipülasyonu düşünüldüğünde 1.Mod yerine 2.Mod ile kullanmayı tercih ederim. (Tercih)					
17	Genel olarak robot kollarını 1. Mod yerine 2. Mod ile kullanmayı tercih ederim. (Tercih)					

APPENDIX B

USER MANUAL

KULLANICI KILAVUZU

-Kullanmak üzere olduğunuz simülasyonda yumuşak robot kollarının eylenmesi her bir kolun içine yerleştirilen dört adet kablo ile sağlanmaktadır.
-Kablo yerleşimleri Şekil.1'de gösterilmiştir.
-Kolların manipülasyonu haptik cihazlar tarafından yapılacaktır.
-Haptik cihazın ilk iki mafsal açısı; kolun üzerindeki kabloların deplasmanlarını kontrol etmektedir.
-Haptik cihazın diğer mafsalları pasiftir. Haptik cihazın ve kolların konumlarının bir örneği, Şekil.2'de gösterilmiştir.

Şekil.1 Kol üzerindeki kablo yerleşimleri

Şekil.2 Haptik cihaz ve kolların durum örnekleri

-Sistemde iki haptik cihazın manipüle ettiği dört kol bulunmaktadır. Şekil.3'te simülasyon programı üzerindeki kol yerleşimleri gösterilmiştir.
-Dört robot kolu; iki görev kolu, bir ışık kolu ve bir kamera kolu olarak atanmıştır.
-Kullanıcı simülasyonda kamera kolunun ucuna yerleştirilen kamera üzerinden görsel geribildirim alacaktır. Kulanıcının simülasyonda kol yerleştimlerini göreceği sıralama Şekil 6'da verilen kullanıcı arayüzündeki gibidir.
-İki haptik cihazdan birisi ışık kolunu ve bir görev kolunu (Şekil.3'e göre A ve C), diğeri bir görev kolu ve kamera kolunu (Şekil.3'e göre B ve D) manipüle etmektedir.
-Şekil.3'e göre A ve C kollarını bilgisayarın solunda bulunan *Touch X* haptik cihazı, Şekil.3'e göre B ve D kollarını bilgisayarın sağında bulunan *Touch* haptik cihazı kontrol etmektedir.
-Haptik cihaza atanan kollar arasındaki geçiş haptik cihazların üzerindeki düğmeler vasıtasıyla sağlanır. Düğmeler Şekil.5'te gösterilmiştir.
-Bilgisayarın sağında bulunan kamerayı da kontrol eden *Touch* haptik cihazının son mafsalı kamera yakınlaştırmasını kontrol etmektedir. Ayrıca bu cihazın diğer düğmesi de kamera yakınlaşmasını kilitlemektedir.
-Kollar arasındaki geçiş dört aşamadan meydana gelmektedir. Haptik cihazın düğmesine her basıldığında aşama değişmektedir.
-Aşama değişiklikleri ve düğme basma indikatörleri Şekil 4'te gösterilen bir kulanıcı arayüzü ile kulanıcıya bildirilmektedir.
-Şekil.6'da verilen şemalarda kol değiştirme aşamaları ve algoritmaları gösterilmiştir.

Figure B.3 User Manual (A3 paper sized) (cont. on next page)

Figure B.3 (cont.)

APPENDIX C

RESULTS

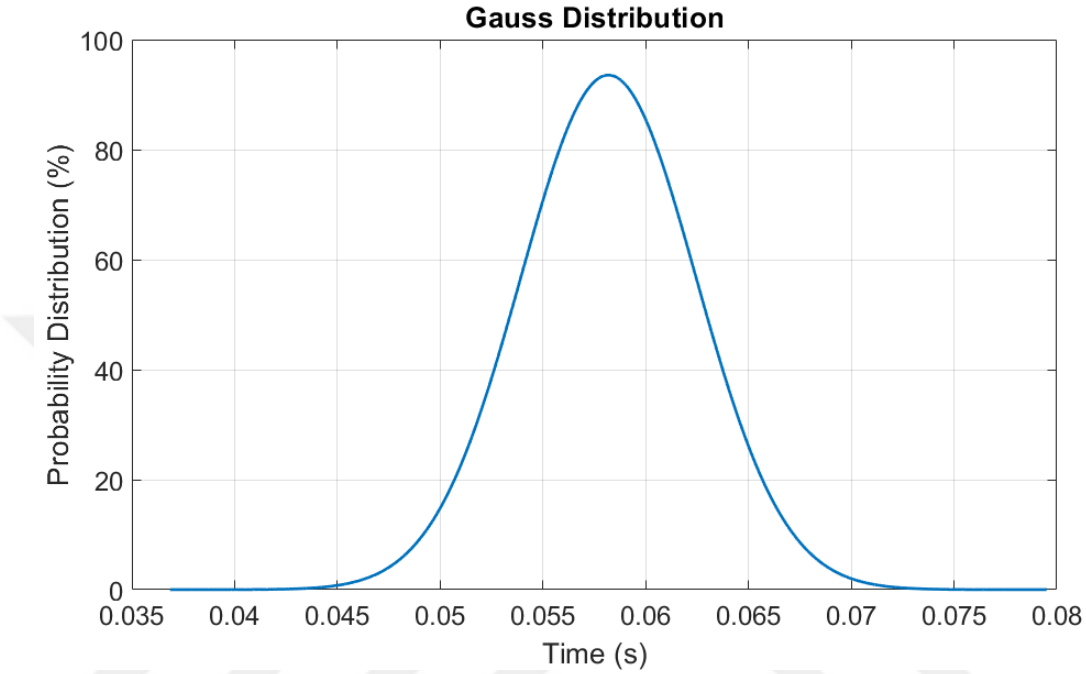

Figure C.1 Real-time experiment Gauss distribution

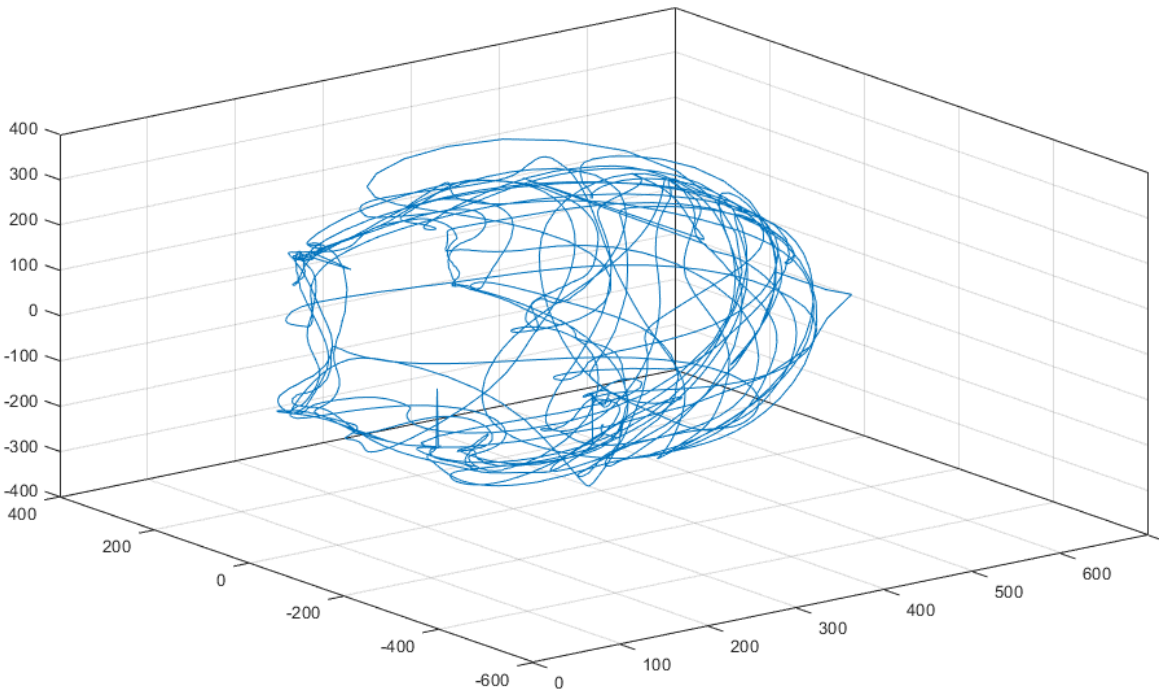

Figure C.2 The reach of the tip of an arm (line)

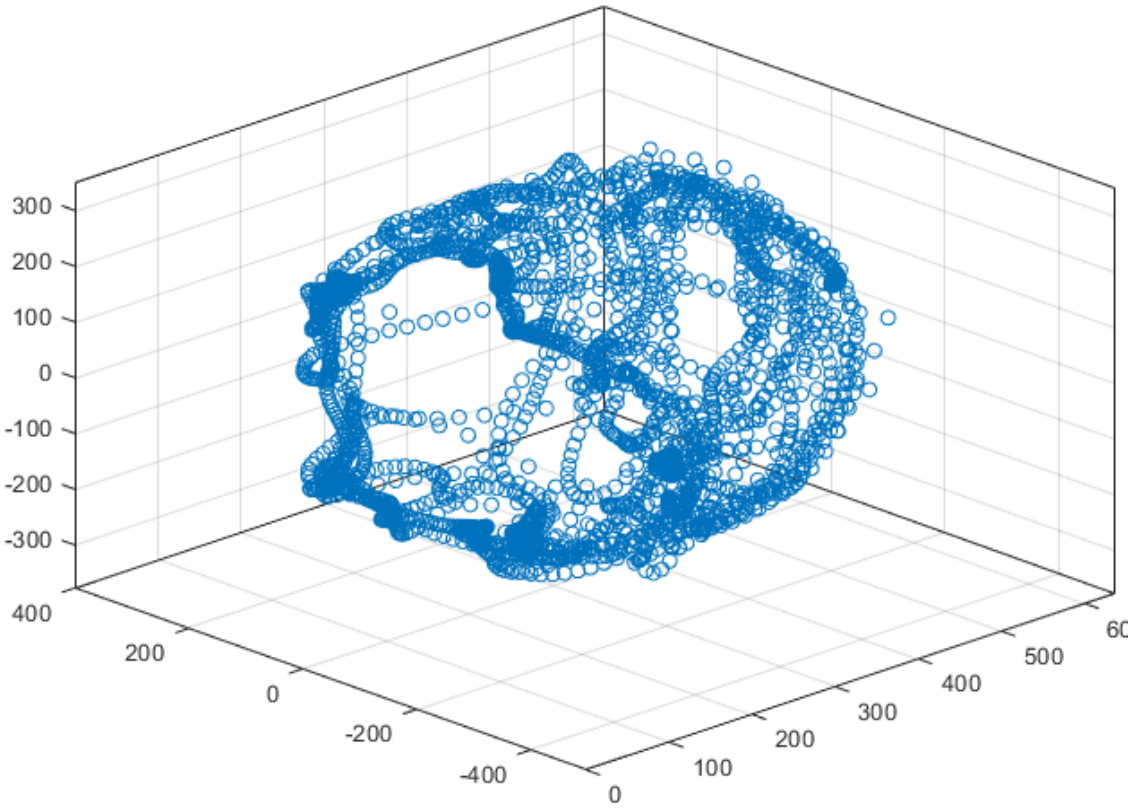

Figure C.3 The reach of the tip of an arm (scatter)

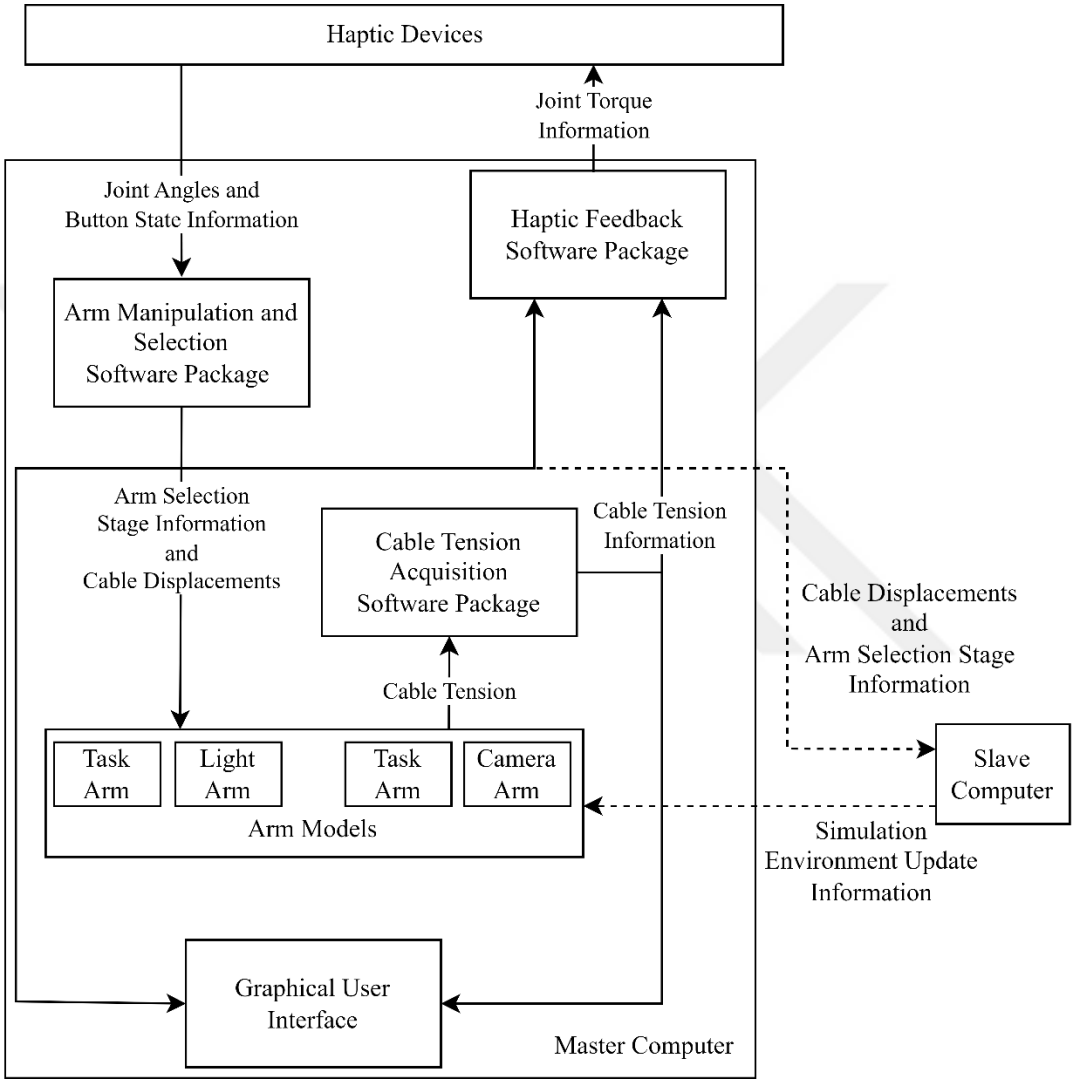

Figure C.4. Master system software schematic

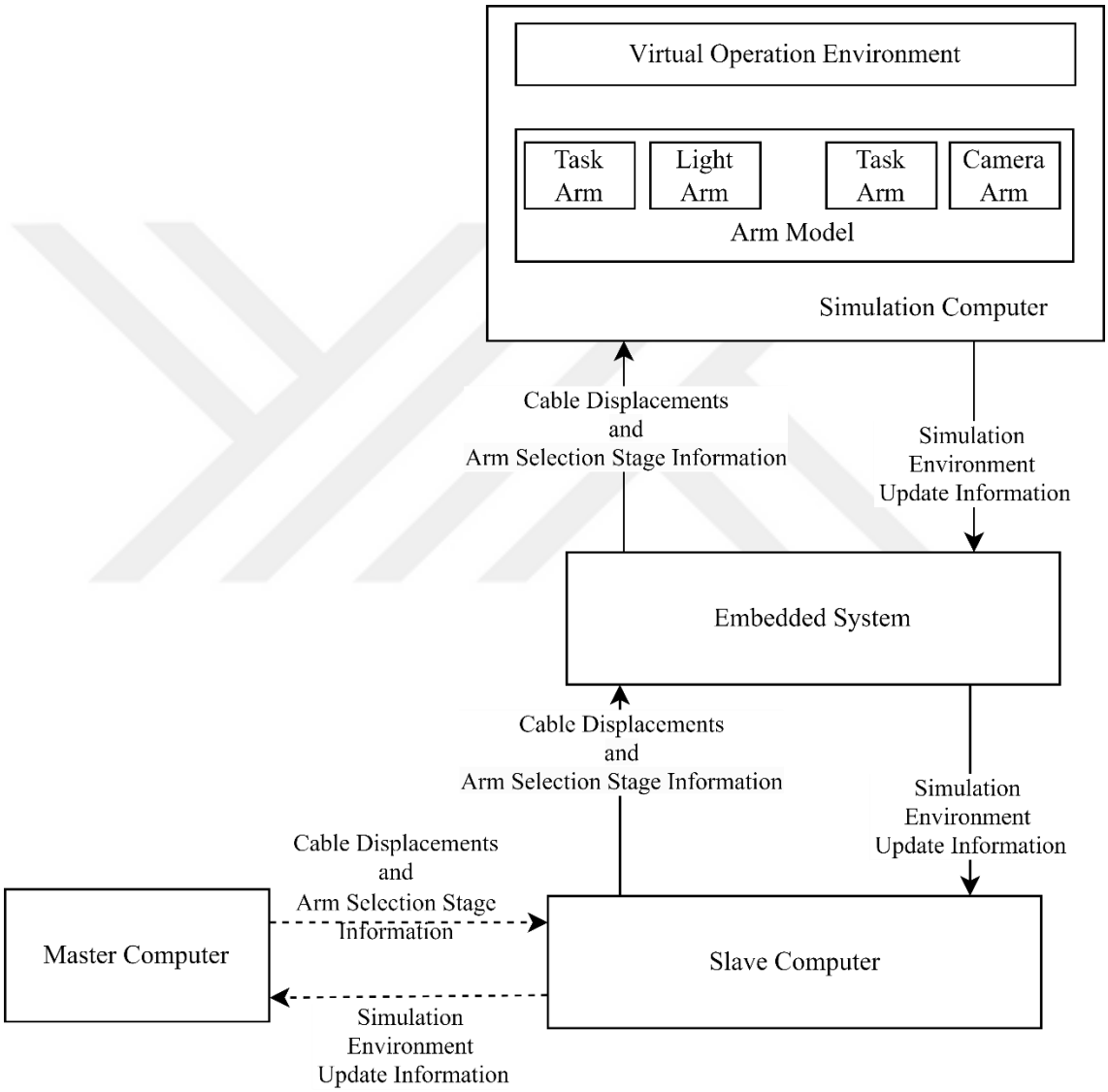

Figure C.5. Slave system software schematic

APPENDIX D

SPECIFICATIONS

Table D.1 Technical Specifications of Haptic Devices

Specification	Geomagic Touch	Geomagic Touch X
Workspace	> 431 W x 348 H x 165 D mm	> 355 W x 228 H x 180 D mm
Footprint (physical area the base of the device occupies on a surface)	~ 168 W x 203 D mm	~ 168 W x 184 D mm
Weight	~1.42 kg	~3.257 kg
Range of Motion	Hand movement pivoting at wrist	
Nominal Position Resolution	~0.055 mm	~0.023 mm
Backdrive Friction	< 0.26 N	< 0.06 N
Maximum Exertable Force (at nominal orthogonal arms position)	3.3 N	7.9 N
Continuous Exertable Force (24 hrs)	> .88 N	> 1.75 N
Stiffness	X axis > 1.26 N/ mm Y axis > 2.31 N/mm Z axis > 1.02 N/mm	X axis > 1.86 N/ mm Y axis > 2.35 N/mm Z axis > 1.48 N/mm
Inertia (apparent mass at tip)	~ 45 g	~ 35 g
Force Feedback	X, Y, Z	
Position Sensing	X, Y, Z (digital encoders)	
Stylus gimbal	Pitch, roll, yaw (± 5% linearity potentiometers)	Pitch, roll, yaw (Magnetic absolute position sensor, 14-bit precision.)
Interface	USB 2.0	Ethernet
OpenHaptics® SDK, Unreal®, Unity® compatibility	Yes	Yes