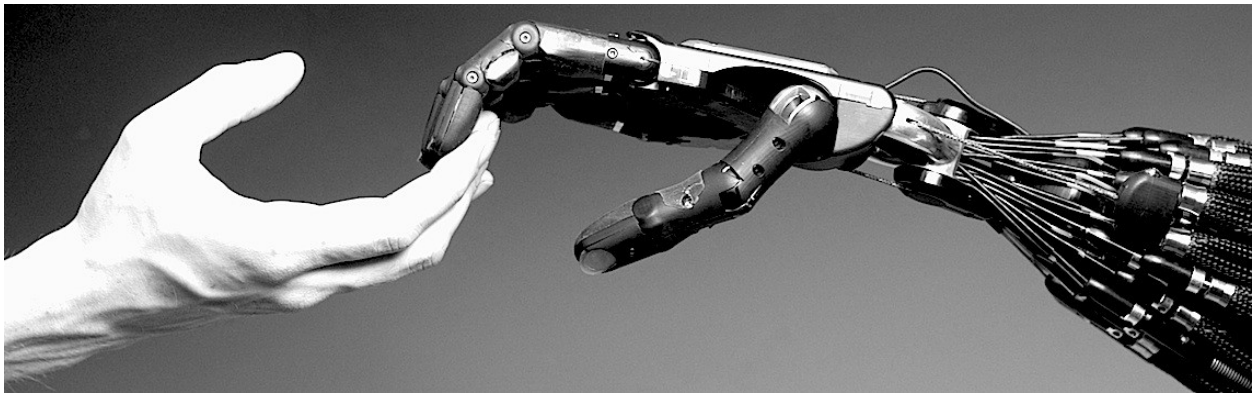
CHALMERS
UNIVERSITY OF TECHNOLOGY



Stability Analysis of Haptic Teleoperation Systems Undergoing Delays in Robot-Assisted Telesurgery

Master's thesis in Biomedical Engineering

AMAL ELAWAD

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis EX063/2017

Stability Analysis of Haptic Teleoperation Systems Undergoing Delays in Robot-Assisted Telesurgery

AMAL ELAWAD

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Biomedical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2017

Stability Analysis of Haptic Teleoperatio Systems
Undergoing Delays in Robot-Assisted Telesurgery
AMAL ELAWAD

Supervisor: Yiannis Karayiannidis, Electrical Engineering Department
Examiner: Balázs Kulcsár, Electrical Engineering Department

Master's Thesis EX063/2017
Electrical Engineering Department
Biomedical Engineering Division
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Courtesy of Shadow Robot Company: *Shadow Dexterous Hand*™ pressing against human hand as a from of haptic feedback control.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2017

Stability Analysis of Haptic Teleoperation Systems Undergoing Delays
in Robot-Assisted Telesurgery
AMAL ELAWAD
Electrical Engineering Department
Chalmers University of Technology

Abstract

During the last decades, attention of robot manufacturers has been directed towards utilizing robots in teleoperation, in order to perform tasks in hazardous or inaccessible sites. In medical applications, Robot-assisted Minimally Invasive Surgery (RMIS) has been employed to increase surgeons' accuracy and dexterity; thus minimizing patients' trauma and reducing recovery time. However, the success of such techniques has been restricted due to their dependency on vision and their lack of haptic feedback to the surgeon (e.g. force feedback). This is due to the fact that vision fidelity can be compromised by low frame rate or subtended angle of vision, in addition to its limited ability to explore surface features.
In contrast, the success of haptic feedback is affected by the delays in the communication channel connecting a surgeon-robot interface to a surgical end-effector. Delays can lead to unstable and inaccurate feedback signals, which can affect the transparency and passivity of the system.

The main aim of this thesis work was to analyze the previously conducted researches on the control design and compensation schemes used to guarantee passivity in case of delays. An adaptive controller was analyzed and implemented for a 3-DOF (Degree of Freedom) manipulator. The slave manipulator was capable of synchronizing with the desired trajectory of the master, while feeding back positions, velocities and contact forces; during both free and constrained motion of the end-effector. In addition, a comparative experiment was performed on a 1-DOF manipulator.

Methods were analyzed and experimented using the 3D Systems Geomagic® Touch™ Haptic Device, and the single-link SensoDrive Master-Slave-System. The software used for programming and simulations for the first manipulator was MATLAB/ SIMULINK R2014a, on a *Windows 7* computer system. The second manipulator used MATLAB/ SIMULINK R2007a and a LINUX computer system. *Wolfram Mathematica 10.0* was used for mathematical manipulations,

Keywords: surgery, bilateral teleoperation, haptic, communication delay, adaptive, passivity, stability.

Acknowledgements

All thanks are due to *Allah*, the most gracious and most merciful.

This thesis report is the result of the Master's work carried out during the final year of my studies at Chalmers University of Technology. This opportunity would not be possible without the Swedish Institute Study Scholarship (SISS), who believed in my potentials and invested in my future, for which I am thankful.

This is to my supervisor, Yiannis Karayiannidis. I will always cherish our fruitful discussions, his gentle guidance, and his precious friendship through this research work. His patience, dedication, kindness and immense trust have paved my way. He was always present to guide me through doubts and insecurities, to celebrate my success, and continued to inspire me.

To my examiner, Balázs Kulcsár, who gave me the great opportunity to fulfill my research goals, and who has always been encouraging, supportive and kind.

Special and genuine thanks to Giuseppe Giordano, the PhD student at Chalmers. I have been fortunate, twice, to have him as a teaching assistant. His support and enlightening discussions continued along the way. I will forever be grateful for having him as a teacher and as a friend.

I would also like to thank Dr. Georg Stillfried and Dr. Jordi Artigas-Esclusa, from the German Aerospace Center (DLR), for giving me the chance to perform comparative experiments in the Institute of Robotics and Mechatronics, and for their endless support and motivation.

This journey could not be possible without unconditional love and support from my family. I dedicate this to my parents, Abdelgadir Elawad and Najat Elfaki, the genesis of all blessings; and to my siblings, Ahmed, Ammar and Banan.

As Aristotle said: "a friend is a single soul dwelling in two bodies". Mine is dwelling in yours: Arwa, Fatehia, Twasul, Ola, Omar, Ahmed, Alaaddin, Heba, Aljaali and Mazin. May we all fulfill our wildest dreams, side by side.

Heartfelt thanks to my colleague and friends, Mareike, Mamadou and Björn. I enjoyed every activity: group assignments, bike rides, swimming classes, shared meals and even shared study rooms. Your companionship and kindness helped me navigating through, and cheered me up whenever I was down.

To you all, I dedicate this thesis.

Amal Elawad, Gothenburg, June 2017

Contents

List of Figures

List of Tables

1
Introduction

This first chapter is an introduction to the thesis. It starts with background information on haptics, general overview and motivation behind using haptic teleoperation, followed by literature review and thesis objectives, culminating with thesis layout.

1.1 Background

Recently, research in the realm of Robot-Assisted Minimally Invasive Surgery (RMIS) have been growing interest. In order to increase surgeons' accuracy and dexterity and minimize patients' trauma and recovery time, robots have been utilized in minimal invasive surgery. However, the success of such techniques has been limited due to their dependency on vision and their lack of a haptic feedback (e.g. force feedback) [5].

1.1.1 Haptics and haptic devices

The origin of the word **Haptic** goes back to ancient Greek (haptikós) meaning "being able to touch or grasp". Haptic technology recreates the sense of touch by applying forces, vibrations or motions to the user. In the human body, there are two types of haptic senses, which play a major role in haptic perception; Cutaneous and Kinesthetic senses. Cutaneous sensation is associated to feeling of temperature, texture, slip, vibration, etc.; which are stimulated at the sensory receptors found in the dermis or epidermis layers of the skin. On the other hand, Kinesthesia refers to the feeling of location, configuration, motion, force, etc. Kinesthetic receptors are located in muscles, tendons, joints, and the internal ear - referred to as proprioceptors.

Therefore, there are two main types of haptic devices:

1. **Tactile haptic devices:** Generate tactile feedback (concentrated or distributed) using a tactile display, in order to stimulate the skin. An example for that are surface display techniques (figure 1.1a).

2. **Kinesthetic haptic devices:** Generate force feedback through a tool, in order to guide or inhibit body movements. These devices can be classified into two categories, the Impedance type and the Admittance type (figure 1.1b). In the first, user's motion (e.g. position) is fed into the system, and forces

(a) Tactile haptic device **(b)** Kinesthetic haptic devices

Figure 1.1: Types of haptic devices[1]

are subsequently generated as a function of motion, using impedance control law. In contrast, the Admittance type incorporates using forces as input to the system, which is then utilized to compute motion signals (e.g. position) using admittance control law. Due to the difficulty to accurately sense or transform force information, the latter type is less common. Examples for such are exoskeletons, manipulandums or grasp techniques, such as the da Vinci surgical robot from Intuitive Surgical, Inc.

The 3-DOF manipulator used in this thesis work is of Kinesthetic-Admittance type, having force/torque as input and motion (i.e. joints positions and velocities) as output.

1.1.2 Teleoperation controllers

The prefix "tele-" has Greek origins meaning "at a distance". Thus, teleoperation indicates operating at a distance. A remote device which is responsive to the control signal of another device is referred to as a *slave*, while the controlling device is called a *master*. Therefore, the master is on the user side while the slave is on the environment side.

The main purpose of teleoperated robots is to operate in hazardous or inaccessible sites. Examples of such include space exploration using robots operated from Earth, operating Unmanned Underwater Vehicles (UUV), handling hazardous materials such as radioactive or explosive substances, military applications in combat areas, and telesurgery to exchange distant medical expertise or to operate on sensitive sites (such as sewing small sutures).

With regards to teleoperators control, it can be divided into two main categories:

1. **Unilateral Teleoperation:** Position information flows in only one direction, from the operator to the remote environment, figure 1.2.

2. **Bilateral Teleoperation:** In order to acquire information about the remote environment, signals have to be fed back to the operator at the master side. Thus, in bilateral teleoperation control, information (e.g. position, velocity and/or force) flows in both directions between the operator and the environment. The reflected signal partly composes the input torque of the receiving-

end manipulator. Figures 1.3a and 1.3b compare two different approaches for bilateral control, position based and force based, respectively.

As shown in figures 1.3a and 1.3b, the position-force controller, in comparison to the position-position controller, explicitly measures the contact forces between the slave and the environment, which is then fed back to the master, while the position of the master is fed forward to the slave via proportional-derivative (and maybe integral) controller. One drawback of this approach is the contact instability, as a result of the constant force-induced excitation between the master and the slave. This constant excitation means that the reflected forces from the environment excites master's motion which further excites the slave's contact forces, and so forth. Therefore, forces reflected to the master should be attenuated while keeping in mind the viability of performance [6].

In this thesis, the implemented teleoperation setting is of bilateral type, where joints positions and velocities are transmitted through the communication channel. Since the used device does not have force sensors, reflected forces could not be explicitly measured.

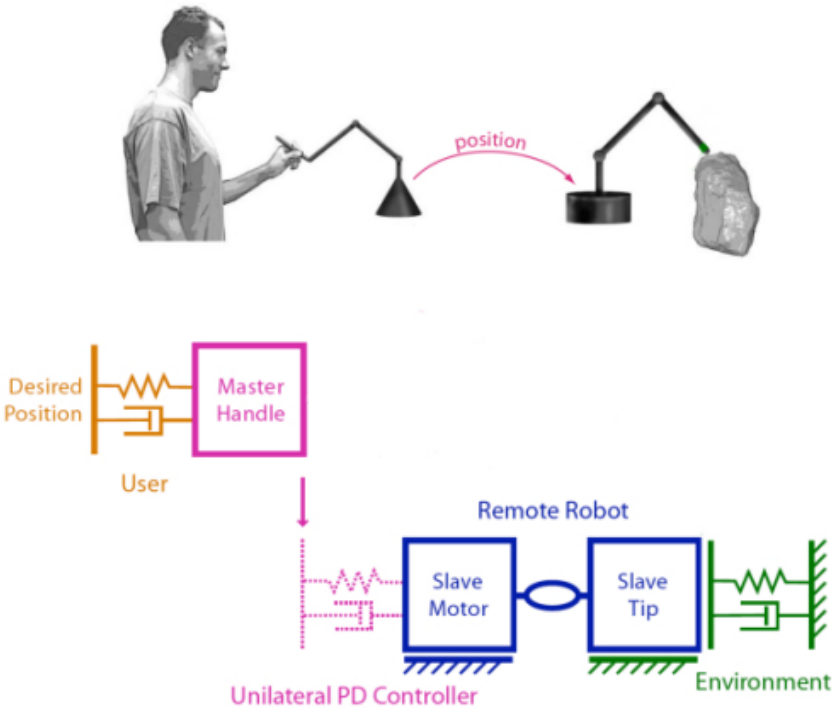
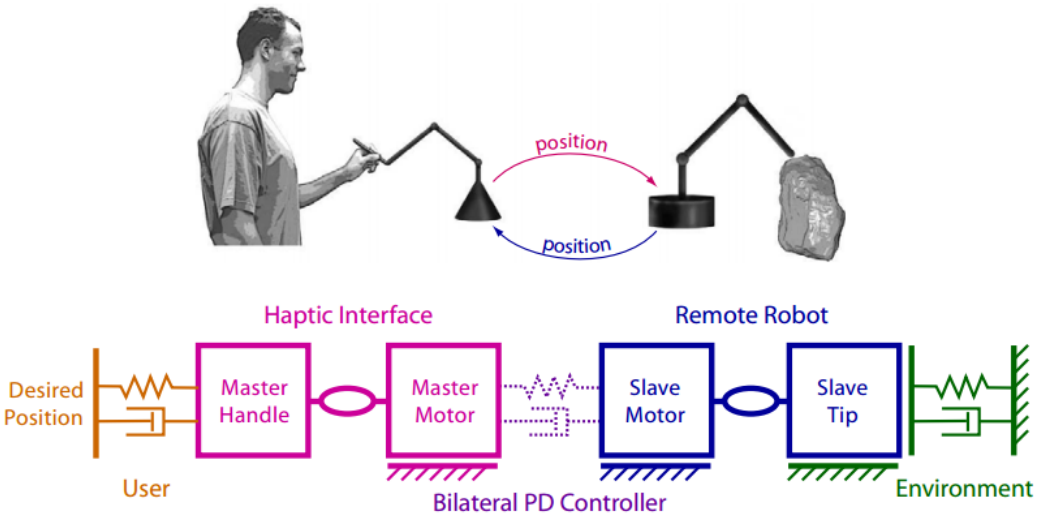
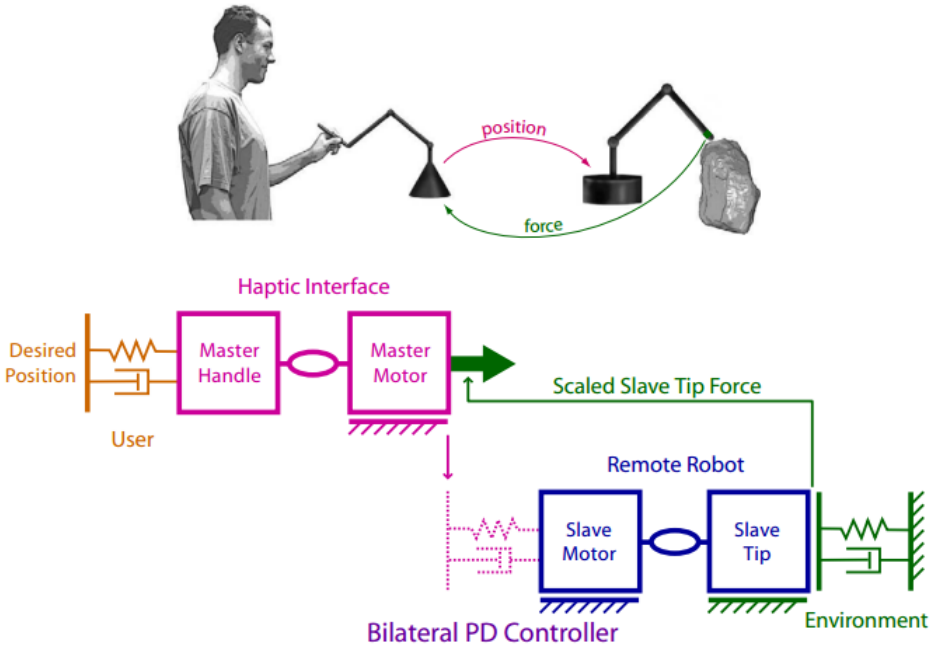


Figure 1.2: Unilateral teleoperation (modified from [2])



(a) Position based



(b) Force based

Figure 1.3: Bilateral teleoperation approaches [2]

1.1.3 Surgical robots: the existing robots and some considerations

One of the most popular surgical robotic system is the da Vinci Surgical System, made by Intuitive Surgical Inc. and approved by the Food and Drug Administration (FDA) in 2001.

Most of the existing surgical robots, such as the da Vinci, can be used in many medical procedures to teleoperate a patient from an adjacent room. Such procedures include prostate surgery, fibroids removal, joint replacement, cardiothoracic and urology surgeries, cutting, suturing and biopsies extraction - aided by Magnetic Resonance Imaging (MRI) or Computed Tomography (CT). Up to May 2012, over 1,450 hospitals around the world have installed more than 1,840 da Vinci Systems.

As discussed in the previous sections, using robots for surgery has many benefits, one of which is reduction of health care costs, through reducing the number of personnel needed in the operation room and reducing patient recovery time. However, some considerations must be thoroughly evaluated, like the cost of the robotic system versus the cost of traditional care. Such systems may cost up to $1million to purchase and average of $\$100,000$ as maintenance expenses [7]. There is also the issue of cost against surgeons' practical training time, in order for them to be able to operate such systems.

1.1.4 Why haptics?

The necessity of providing haptic feedback in teleoperation arises from the drawbacks of the traditional visual teleoperation. Vision fidelity can be compromised by low frame rate or subtended angle of vision. Moreover, utilizing force feedback has been proven to reduce task completion time and errors, and prevent applying excessive force; since the user can always feel the reflected forces or motion from the environment. [8]. Another factor is that the exploration of surface features or deformations is quite limited when only visual cues are used, making the sense of touch more valuable [9].

1.2 Literature review

1.2.1 Performance requirements

The main requirements of haptic teleoperation performance are stability, motion tracking and transparency. In this subsection, the definition and necessity of each requirement will be explained.

1.2.1.1 Stability and passivity

In bilateral teleoperation, and due to time delays, the direct exchange of forces and positions/velocities (i.e. power variables) between the master and the slave can result in creating a "virtual" energy in the communication channel. This turns it into an active element, which can cause system instability [10]. An elegant way to solve the stability problem of the telemanipulation closed loop is by guaranteeing passivity, which can be defined in terms of energy balance - through dissipation and transformation, between input and output of the system. Dissipated energy is the difference between stored energy and supplied energy. Therefore, if passivity could be guaranteed for a system, this also implies that the system has bounded output energy if its input energy is bounded, i.e. a Bounded Input Bounded Output system (BIBO). Therefore, passivity can be seen from an input/output stability perspective. It should be noted that in contrast, Lyapunov stability handles system's internal stability and how far the states of the system are from their desired value, i.e. system's behaviour in comparison to the desired performance.

Passivity-based teleoperation control aims at passifying the master and the slave, and implicitly passifying the communication channel. The latter is a fundamental property of passive systems[11], while a human operator is assumed to be passive.

1.2.1.2 Motion tracking and transparency

Motion tracking is the ability of the slave manipulator to track the position and the velocity of the master (also referred to as performance). Whereas, transparency - which only applies to bilateral teleoperation - can be defined as impedance matching, where the mechanical impedance generated by the environment (e.g. stiffness, represented by force and displacement, and/or damping, represented by force and velocity) should be the same felt by the user. In other words, a faithful transmission of signals must be guaranteed.

An ideal teleoperation system is a completely transparent one if it completely couples the user to a remote environment. Assuming the case of transmitting only forces/torques and velocities, the two quantities are related by the impedance (Z):

$$\tau = Z\dot{q}$$

where $\dot{q}$ is the joint velocity and τ represents the interaction torques. Therefore, in order for the system to be completely transparent, the operator's force and velocity should be the same as the ones reflected by the environment, such that:

$$\begin{aligned} \dot{q}_s(t) &= \dot{q}_m(t) \\ \tau_m(t) &= \tau_s(t) \end{aligned} \tag{1.1}$$

where the subscripts "s" and "m" represent the slave and master sides, respectively. This can be satisfied by impedance matching, i.e. the transmitted impedance (to

the master) and the reflected impedance (from the slave) are equal ($Z_t = Z_e$) [6]. In the presence of arbitrary communication delay (T), equation 1.1 would become [10]:

$$\begin{aligned} \dot{q}_s(t) &= \dot{q}_m(t-T) \\ \tau_m(t-T) &= \tau_s(t) \end{aligned} \tag{1.2}$$

This type of delay is even more hazardous in sensitive applications, like surgical operations. In this thesis, a synchronization scheme is adopted to minimize the error between the transmitted and received positions and velocities to zero.

However, before insuring transparency, the whole teleoperation system should be stable. In general, a trade-off should be made between transparency and stability, according to the intended task and the teleoperator and controller architectures. For example, excess accuracy in position matching in presence of time-delays can cause the system to be heavy and experience large reaction forces (due to the lag), which jeopardizes system stability. In this thesis, this trade-off was made between tuning of controller parameter to improve performance versus stability.

1.2.2 Previous work

Several complications arise when studying teleoperated systems, since the communication medium contributes substantially to the complexity of the overall system and introduces distortion, delay, packet loss, etc. This has an impact on stability and performance. Such issues have motivated the theoretic research on teleoperation control over the past decades [12].

Throughout time, many schemes have been utilized in order to satisfy stability conditions in the presence of communication delay. Here, the interest is only directed towards passivity-based schemes, briefly described below.

The first scheme, developed by Anderson and Spong in [13], and Niemeyer and Slotine in [14], was a passivity-based architecture using scattering concept, which is considered a milestone in teleoperation control. It assured robust stability against constant time delay. However, due to feeding back only the velocity (but not the position information), if initial condition mismatch is present, position drift between the master and the slave may occur. Thus, it only assured robust velocity tracking, but position tracking was not guaranteed.

In order to improve the aforementioned approach, schemes were developed by Chopra et al. [15], Lee and Spong [16] and Niemeyer and Slotine [17]; where both position and velocity information were fed back through the system.

In the work of Chopra et al.[15] and Lee and Spong[16], coupling gains were delay dependent. Moreover, Chopra et al. used a single DOF manipulator and a constant-time delay network, and designed a scattering-wave-based controller with additional

proportional controllers, which used the delayed position data as a reference signal. A Lyapunov-based analysis was used to determine the allowable range of the delay [15].

In contrast, Lee and Spong used a passive framework undergoing constant communication delay for a multi-DOF system, without using the scattering transform and with explicit position feedback. Instead, a proportional-derivative (PD) control framework was used to enforce energetic passivity of the closed-loop system. This was done by utilizing the controller passivity concept, Lyapunov-Krasovskii theory for delayed systems, and Parseval's identity. Position tracking was achieved, but force tracking was influenced by the delay, resulting in a delay-dependent force peak in the transient period, and the design was only suitable for low-frequency region.

Returning briefly to Niemeyer and Slotine's paper [17], where they considered the case of time-varying delays, and showed that stability can be satisfied through designing wave-variable filters (energy conserving filters). Both wave integral and wave energy were transmitted across the channel, where position information was encoded, and then reconstructed using the filter. However, using scattering transformation can degrade the performance of the system by leading to wave reflections [18].

Therefore, synchronization-based approaches have been adopted. Agents are said to be synchronizing if their outputs (position error and velocity error) converge asymptotically to the origin. Those approaches were developed by Chopra and Spong [18, 19], and Rodriguez-Angeles and Nijmeijer [20].

The focus of Chopra and Spong [18] was to introduce a novel passivity dependent approach, which guaranteed position and force tracking, and was robust against initial conditions offset and packet loss. Without using the scattering transform, delay independent exponential stability was guaranteed (for both position and velocity tracking errors). Only simulations were performed using a manipulator with only single DOF. Chopra and Spong reported that environmental forces were accurately transmitted to the master. However, it should be noted that only constant-time delays were considered, and convergence of velocities was not illustrated.

The cornerstone of this thesis is the extended passivity architecture developed by Chopra et al. [21], in order to guarantee state synchronization. Their work handled the synchronization of both position and velocity, without using the scattering transform. The type of controller used was an adaptive architecture with state feedback, defining new passive outputs to state synchronize master and the slave robots in free motion. Furthermore, in contact with a passive rigid environment, the proposed architecture also guaranteed state synchronization and ultimate boundedness of master-slave trajectories.

A medical application, done by Sherman et. al. [22], should be mentioned here. The authors used a haptic devices through comparing three teleoperator controller architectures with respect to their capability of detecting the changes a compli-

ant environment (such as soft tissue). The three methods differ in their feedback mechanism, whether it is a position-error-based force feedback, a kinesthetic force feedback, or a hybrid controller of position and force feedback. The task was to mimic palpation of soft tissue. It was found that the latter method had the best performance in comparison to the other two [22].

Since the comparative experiments were conducted in the laboratories of the Institute of Robotics and Mechatronics, at the German Aerospace Center (DLR); some of DLR teleoperation experiments should be mentioned here. One of the implemented approaches is the symmetric position-position teleoperation architecture [23]. The method used was the Time Domain Passivity Control Approach (TDPA) in order to passify the communication channel in time-domain. In this control method, passivity was established as a system property and not as a design constraint (in order to avoid forcing conservative rules). The energy of the delayed communication channel could be modulated by using passivity observers (PO), which monitor energy flow in the time domain, and passivity controllers (PC) which dissipate the active observed energy by PO through by acting like a damper; such that the network is always passive.

It has been found that an offset between the current and desired position may occur, due to using a position-position architecture. The forward and backward PC modify the velocity to produce the dissipation effect. This velocity is then integrated to obtain the desired position. The position offset outputs as force/torque, which may result in force accumulation that can be felt by the user if the delay is large [23].

1.3 Objectives

The main objective of this Master's thesis work is to study the stability of bilateral teleoperation in surgical robotics, when delays are present in the communication channel. A passivity-based adaptive controller is implemented, simulated and experimented onto a 3-DOF manipulator, in order to fulfill the following objectives:

- Synchronize and transmit position and velocity signals between a master and slave manipulators through a communication channel, while providing a force feedback at the master's side.
- Ensure and analyze stability, transparency and accuracy conditions throughout the teleoperation process.

1.4 Thesis layout

- Chapter 1: Is devoted to the introduction to haptics and teleoperation types, with literature review and objectives layout.

- Chapter 2: The dynamic model and equation of motion of the manipulator are derived (in joint space). Furthermore, modeling of human forces and environmental forces is discussed.

- Chapter 3: The controller to solve the problem stated in chapter 2 is analyzed and discussed.

- Chapter 4: The simulations and experiments are performed and reported, with a discussion on the outcomes.

- Chapter 5: The results and discussion of the comparative experiment, which is conducted using the same controller and implemented on a single-DOF manipulator, are illustrated in this chapter.

- Chapter 6: General discussions, concluding remarks and recommendations for future work.

2

Modelling and Problem Formulation

In this chapter, the mathematical model of the manipulator used to carry out the results of this thesis will be laid out. It starts with a short description of the mechanical structure of the manipulator. Then, a discussion about modelling human and environmental forces will be demonstrated.

2.1 Manipulator modelling

In this thesis, the 3D Systems Geomagic® Touch™ Haptic Device (also called Omni™ Bundle, and formerly referred to as Sensable Phantom Omni) was used. This device can be used for many applications, including simulation, rehabilitation and robotic control. It has the ability to provide a 6-DOF position sensing (x,y and z axes via digital encoders, with pitch, roll and yaw) and a 3-DOF force feedback (x,y and z axes) to the user. It also has a J45 compliant on-board Ethernet Port or USB Port interface [24].

Figure 2.1 shows the 3D Systems Geomagic® Touch™ Haptic Device. It can be seen that the system has 3 independently moving rotational joints: J1, J2 and J3. It was assumed that joints 1 and 2 are connected to a point, such that the whole system is connected via 2 links: L1 and L2. Therefore, the system has 3-DOF, since each joint can move independently from the other two. There are neither redundancy nor motion constraints, only physical constraints due to mechanical construction, which limits the task space of the manipulator (e.g. joint locking). Figure 2.2 shows the coordinate frames of the manipulator, defined using the Denavit–Hartenberg (DH) convention.
Hence, the generalized coordinates in the joint space for the system are the three angles, q_1, q_2 and q_3.

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2.1}$$

Figure 2.1: the 3D Systems Geomagic® Touch™ Haptic Device [3]

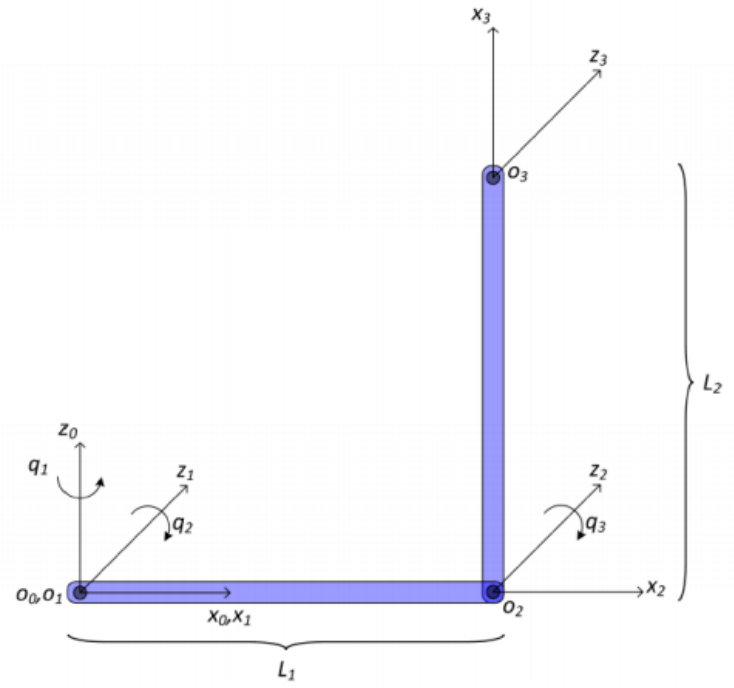


Figure 2.2: Manipulator coordinate frames[3]

In order to simulate the motion and design control algorithms, deriving the dynamic model is a crucial step. The dynamic model relates the torques acting on the joints to the resulting motion of the whole structure. One of the methods to derive the equation of motion of a manipulator -in joint space- is the Lagrangian formulation, which is considered to be simple, elegant and methodical.

The Lagrangian of the mechanical system is defined as a function of the generalized coordinates, and using the energy equations of the system, as follows:

$$\mathcal{L}(q, \dot{q}) = \mathcal{T}(q, \dot{q}) - \mathcal{U}(q) \tag{2.2}$$

where $\mathcal{L}$ denotes the Lagrangian, $\mathcal{T}$ denotes the total kinetic energy of the system, and $\mathcal{U}$ is the total potential energy.

The Euler-Lagrangian equation is given by:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \qquad ; \quad i = 1, 2, 3 \tag{2.3}$$

where ξ_i is the generalized force associated to the generalized coordinate q_i, while i refers to the corresponding joint of the manipulator system under consideration.

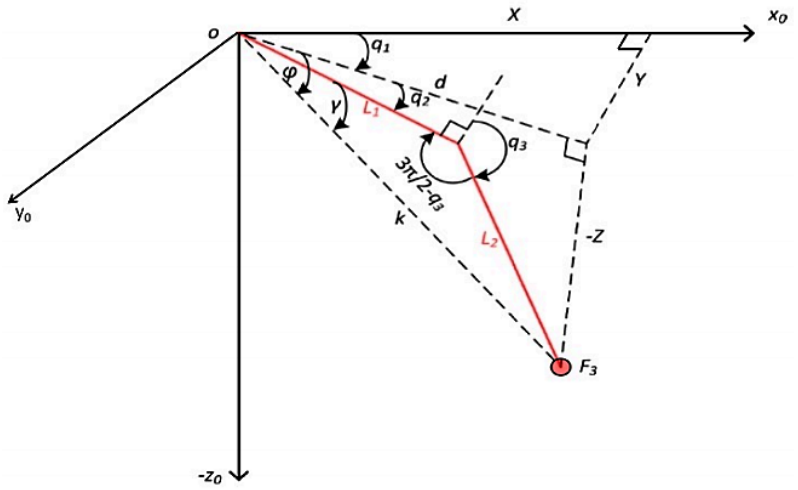


Figure 2.3: System's geometric structure[3]

Figure 2.3 illustrates the geometry of the system, used to derive the model and the inverse kinematics; where triangle -Zdk is in the vertical (x-z) plane, and triangle YXd is in the horizontal (x-y) plane.

The final form of the equation of motion for the system under study is as follows:

$$\begin{aligned} B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) &= \xi \\ B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + F_v\,\dot{q} + F_s\,\mathrm{sgn}\,(\dot{q}) &= \tau - \tau_e \\ &= \tau - \mathrm{J}^{\mathrm{T}}(q)\,h_e \end{aligned} \tag{2.4}$$

where $B(q)$ is the inertia forces matrix, $C(q, \dot{q})$ is the Coriolis and centrifugal forces matrix, $g(q)$ is the gravitational forces vector. Also, the generalized forces, referred to as ξ, include input forces, dissipative forces (such as friction) and constraint-induced forces. These include non-conservative forces acting on the manipulator such as the motors actuating torques (τ), the viscous friction torques (i.e. $F_v\,\dot{q}$), and Coulomb friction (i.e. $F_s\,\mathrm{sgn}(\dot{q})$). Frictions F_v and F_s are both $(n \times n)$ diagonal matrices of friction coefficients, and $\mathrm{sgn}(\dot{q})$ is a $(n \times 1)$ vector of sign function of joints velocities. Furthermore, if the end-effector is in contact with an environment, the balancing torques (due to the external contact forces) are referred to as τ_e, which is given by the product of the geometric Jacobian (i.e. J) and the vector of forces and moments exerted on the environment by the end-effector (i.e. h_e).

There are three important properties of equation 2.4 (due to its Lagrangian dynamics) which should be demonstrated:

1. **Positive definiteness:** The inertia matrix $B(q)$ is symmetric positive definite.

2. **Linearity in the Dynamic Parameters:** The dynamics equation (2.4) is linearly parameterizable with respect to the uncertain dynamics (e.g. mass and inertia). Assuming the simplest case where there are neither contact forces (i.e. $h_e = 0$) nor frictions (i.e. $F_v = F_s = 0$), one gets:

$$B(q)\ddot{q} + n(q, \dot{q}) + g(q) = Y(q, \dot{q}, \ddot{q})\,\pi = \tau \tag{2.5}$$

   where $Y(q, \dot{q}, \ddot{q})$ is a $\in R^{n \times p}$ matrix, as a function of the generalized coordinates and their derivatives, containing all the known terms in the dynamic equation, while $\pi \in R^{p \times 1}$ is a constant vector of the uncertain dynamic parameters (e.g. moments of inertia and link masses).

3. **Skew-Symmetry:** If, for the matrix $n(q, \dot{q}) = C(q, \dot{q})\dot{q}$, the matrix $C(q, \dot{q})$ was properly defined, the matrix:

$$N(q, \dot{q}) = \dot{B} - 2C(q, \dot{q})$$

   is skew-symmetric. Thus: $\dot{q}^{\mathrm{T}} N(q, \dot{q})\,\dot{q} = 0$.

2.2 Modelling human and environmental forces

A teleoperator consists of five subsystems: the human operator, the master, the communication block, the slave and the environment. In this thesis, as it will be discussed later in chapter 3, the master and slave manipulators are connected via an adaptive controller, which incorporates synchronizing position and velocity. Figure 2.4 shows the block diagram of the system.

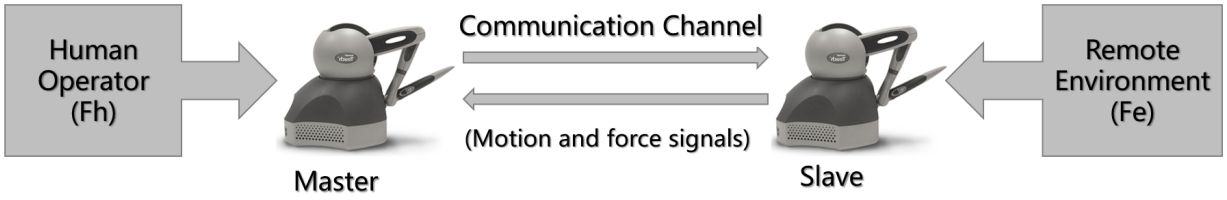


Figure 2.4: Bilateral teleoperation System

Generally, a human operator can be modelled as a mass-spring-damper system, representing the arm mechanics (including muscle and skin); while the environment can be modelled in a similar manner, using a spring stiffness to represent the type of the constraint (e.g. rigid vs. compliant wall). The communication channel can also be modelled as a PD system [2].

Two scenarios have been simulated and implemented in the study at hand, which are classified according to the way of interaction between the manipulator and the human operator, and between the manipulator and the environment. Therefore, there are many ways to simulate the human and environmental forces. Human forces can be differentiated according to whether the operator is considering or ignoring the manipulator's planned trajectory. While the environment modelling depends on whether the motion of the slave manipulator is free or constrained, keeping in mind the different nature of constraints (compliant vs. rigid).
It should be noted that throughout this thesis report, the subscripts m and s represent the master and slave sides, respectively.

2.2.1 Human forces

In this study, forces are simulated as a non-passive PD system, using the model:

$$\begin{aligned} F_h(t) &= -C_h\,\dot{x}_m(t) - K_h\,(\tilde{x}_h(t)) \\ &= -C_h\,\dot{x}_m(t) - K_h\,(x_m(t) - x_{hd}(t)) \end{aligned} \tag{2.6}$$

where K_h and C_h are constant positive definite matrices representing the stiffness and damping coefficients, respectively, while $\tilde{x}_h(t) = x_m(t) - x_{hd}(t)$ is the error between master's trajectory and human's desired trajectory ($x_{hd}(t)$), given in operational space. In some cases, where the desired human position (x_{hd}) is a constant position rather than a trajectory, a bounded constant force can be added to the relation (2.6).

The values of impedance parameters vary according to the intended task. For instance, a writing task or a surgical procedure requires high precision, low velocity and small movements. Therefore, high values of the stiffness and damping parameters are needed. In contrast, high velocity and big movement require low parameter values [25].

2.2.2 Environmental Forces

Turning attention to the environment, the slave is either allowed to move freely, and in this case there are no environmental forces, or the slave motion in constrained.

The only type of constraints implemented here is a rigid wall constraint (e.g. Aluminum wall). In this case, forces are represented in a similar way to equation (2.6):

$$\begin{aligned} F_e(t) &= -C_e\,\dot{x}_s(t) - K_e\,(\tilde{x}_e(t)) \\ &= -C_e\,\dot{x}_s(t) - K_e\,(x_s(t) - x_e(t)) \end{aligned} \tag{2.7}$$

Similarly, K_e and C_e are constant positive definite matrices representing the stiffness and damping coefficients of the environment, respectively. Also, the error between slave trajectory ($x_s(t)$) and the location of the constraint in the environment ($x_e(t)$), in operational space, is given by $\tilde{x}_e(t)$.

In both cases of the human and environmental end-effector forces, a Jacobian matrix must be used to transform external forces at the end-effector into joint torques. As mentioned previously in equation (2.4), this is performed using the relation:

$$\tau_e = \mathrm{J}^{\mathrm{T}}(q)\,h_e \tag{2.8}$$

The Jacobian was found using the forward kinematice, and by assuming that there is no alteration in the orientation of the end-effector, thus:

$$\dot{x}_e = \dot{p}_e = J_p\,\dot{q} \tag{2.9}$$

where J_p is the part of the Jacobian related to the linear velocity of the end-effector and $\dot{p}_e$ is the end-effector velocity. More details on the direct kinematics and how to acquire the vector p_e are in appendix A.

3
Controller Design

In this chapter, the controller used to perform the intended task is illustrated and explained, using the manipulator described in chapter 2. The focus of this controller design is to extend the one designed by Chopra et al. in [21] and to perform further testing on a 3-DOF manipulator, in order to ensure stability against network delay in a bilateral teleoperation setting. Here, an adaptive controller was used for state (position and velocity) synchronization through the communication channel.

Recalling from chapter 2 and from equation (2.4), and taking into consideration the viscous friction (i.e. $F_v\,\dot{q}$) while ignoring the Coulomb friction, the following equation of motion is obtained (for both master and slave sides):

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + F_v\,\dot{q} = \tau - \tau_e \tag{3.1}$$

The symbols were explained in chapter 2.

Since there are two identical manipulators in the system understudy (i.e. a master and a slave), and by putting the human operator and the environmental forces in the loop, we get the following dynamic equation:

$$\begin{aligned} B_m(q_m)\ddot{q}_m + C_m(q_m,\dot{q}_m)\dot{q}_m + g_m(q_m) + Fv_m\,\dot{q}_m &= F_h(t) + \tau_m(t) \\ B_s(q_s)\ddot{q}_s + C_s(q_s,\dot{q}_s)\dot{q}_s + g_s(q_s) + Fv_s\,\dot{q}_s &= \tau_s(t) - F_e(t) \end{aligned} \tag{3.2}$$

where F_h is the force exerted by the human operator and F_e is the force reflected from the environment. The terms Fv_m and Fv_s denote the viscous friction torques for the master and slave side, respectively. For simplicity, throughout the following equations, F_h and F_e are assumed to be zero (i.e. complete free motion).
Recalling the property of *linearity in the dynamic parameters*; equation (3.2) can be written as:

$$\begin{aligned} B_m(q_m)\ddot{q}_m + C_m(q_m,\dot{q}_m)\dot{q}_m + g_m(q_m) + Fv_m\,\dot{q}_m &= Y_m(q_m,\dot{q}_m,\ddot{q}_m)\,\pi_m \\ B_s(q_s)\ddot{q}_s + C_s(q_s,\dot{q}_s)\dot{q}_s + g_s(q_s) + Fv_s\,\dot{q}_s &= Y_s(q_s,\dot{q}_s,\ddot{q}_s)\,\pi_s \end{aligned} \tag{3.3}$$

where π_m and π_s are column vectors of the uncertain dynamic parameters (shown in appendix A), and Y_m and Y_s are the remaining terms in equation (3.2) after extracting the π vector. It is worth mentioning that the terms $(Y_i(q_i,\dot{q}_i,\ddot{q}_i)\,\pi_i)$ include the product of viscous friction and joint velocity $(Fv_i\,\dot{q}_i)$, as in equation (A.9). The numerical values and exact terms of the vectors/matrices in relation (3.3) are elaborated in appendix A.2.

In case of using an adaptive controller to synchronize the motion of a master and a slave manipulators, a "sliding surface" for each manipulator is defined, referred to as σ_m and σ_s. Those surfaces are defined as a combination of the velocity and scaled position of the corresponding manipulator. The control objective is to impose synchronization by matching the two sliding surfaces of both manipulators, which are given by:

$$\begin{aligned} \sigma_m(t) &= \dot{q}_m(t) + \Lambda q_m(t) \qquad \text{Master} \\ \sigma_s(t) &= \dot{q}_s(t) + \Lambda q_s(t) \qquad \text{Slave} \end{aligned} \tag{3.4}$$

where Λ is a constant positive definite matrix. Two bilateral teleoperators are said to be state synchronized if the following was satisfied:

$$\begin{aligned} \lim_{t\to\infty} ||q_m(t-T) - q_s(t)|| &= \lim_{t\to\infty} ||q_s(t-T) - q_m(t)|| = 0 \\ \lim_{t\to\infty} ||\dot{q}_m(t-T) - \dot{q}_s(t)|| &= \lim_{t\to\infty} ||\dot{q}_s(t-T) - \dot{q}_m(t)|| = 0 \end{aligned} \tag{3.5}$$

where T is the delay time, and $||.||$ is the Euclidean norm. The terms inside the norms are coordination errors of the master and slave along with their derivatives, defined as:

$$\begin{aligned} e_m(t) &= q_m(t-T) - q_s(t); \qquad \text{Master position error} \\ e_s(t) &= q_s(t-T) - q_m(t); \qquad \text{Slave position error} \end{aligned} \tag{3.6}$$

Hence, the master and slave manipulators are said to be state-synchronized if their position and velocity errors asymptotically converge to the origin [21].

To state-synchronize the two manipulators, coordinating torques are required:

$$\begin{aligned} \bar{\tau}_m(t) &= K(\sigma_s(t-T) - \sigma_m(t)) + K_p\, e_s(t); \qquad \text{Master side} \\ \bar{\tau}_s(t) &= K(\sigma_m(t-T) - \sigma_s(t)) + K_p\, e_m(t); \qquad \text{Slave side} \end{aligned} \tag{3.7}$$

It can be noticed that the master side uses the delayed slave signals, and vice versa. The coordinating torques incorporate a PD term (since σ_i includes position and velocity information), in addition to a position control term (i.e. $K_p\, e_i(t)$), for finer position control, where $K, K_p \in R^{n\times n}$ are constant positive-definite matrices.

Defining the dynamic equation of the estimated dynamic parameters as:

$$\begin{aligned} \hat{B}_m(q_m)\,\Lambda\dot{q}_m(t) + \hat{C}_m(q_m,\dot{q}_m)\,\Lambda q_m(t) - \hat{g}_m(q_m) - \hat{F}v_m\,\dot{q}_m &= Y_m(q_m,\dot{q}_m)\,\hat{\pi}_m \\ \hat{B}_s(q_s)\,\Lambda\dot{q}_s(t) + \hat{C}_s(q_s,\dot{q}_s)\,\Lambda q_s(t) - \hat{g}_s(q_s) - \hat{F}v_s\,\dot{q}_s &= Y_s(q_s,\dot{q}_s)\,\hat{\pi}_s \end{aligned} \tag{3.8}$$

such that $\hat{B}_i(q_i)$, $\hat{C}_i(q_i,\dot{q}_i)$, $\hat{g}_i(q_i)$, $\hat{F}v_i$ and $\hat{\pi}_i(t)$ are the estimations of the corresponding matrices at each run, where $i = m, s$.

In order to achieve the ultimate boundedness of master/slave trajectories, the input torques are chosen as [21]:

$$\begin{aligned}\tau_m(t) &= \bar{\tau}_m(t) - \hat{B}_m(q_m)\,\Lambda\dot{q}_m(t) - \hat{C}_m(q_m,\dot{q}_m)\,\Lambda q_m(t) + \hat{g}_m(q_m) + \hat{F}v_m\,\dot{q}_m \\ &= \bar{\tau}_m(t) - Y_m(q_m,\dot{q}_m)\,\hat{\pi}_m \\ \tau_s(t) &= \bar{\tau}_s(t) - \hat{B}_s(q_s)\,\Lambda\dot{q}_s(t) - \hat{C}_s(q_s,\dot{q}_s)\,\Lambda q_s(t) + \hat{g}_s(q_s) + \hat{F}v_s\,\dot{q}_s \\ &= \bar{\tau}_s(t) - Y_s(q_s,\dot{q}_s)\,\hat{\pi}_s\end{aligned} \tag{3.9}$$

By plugging (3.9) in (3.2), we get:

$$\begin{aligned}B_m(q_m)\ddot{q}_m + C_m(q_m,\dot{q}_m)\dot{q}_m + g_m(q_m) + Fv_m\,\dot{q}_m &= \bar{\tau}_m(t) - \hat{B}_m(q_m)\,\Lambda\dot{q}_m(t) - \hat{C}_m(q_m,\dot{q}_m)\,\Lambda q_m(t) \\ &\quad + \hat{g}_m(q_m) + \hat{F}v_m\,\dot{q}_m \\ B_s(q_s)\ddot{q}_s + C_s(q_s,\dot{q}_s)\dot{q}_s + g_s(q_s) + Fv_s\,\dot{q}_s &= \bar{\tau}_s(t) - \hat{B}_s(q_s)\,\Lambda\dot{q}_s(t) - \hat{C}_s(q_s,\dot{q}_s)\,\Lambda q_s(t) \\ &\quad + \hat{g}_s(q_s) + \hat{F}v_s\,\dot{q}_s\end{aligned} \tag{3.10}$$

Recalling, from equation (3.4), that $\dot{q}_i(t) = \sigma_i(t) - \Lambda q_i(t)$ and its derivative $\ddot{q}_i(t) = \dot{\sigma}_i(t) - \Lambda\dot{q}_i(t)$, and then substituting these into the equation above, one gets:

$$\begin{aligned}B_m(q_m)\dot{\sigma}_m(t) + C_m(q_m,\dot{q}_m)\sigma_m(t) &= \bar{\tau}_m(t) + (B_m(q_m) - \hat{B}_m(q_m))\,\Lambda\dot{q}_m(t) \\ &\quad + (C_m(q_m,\dot{q}_m) - \hat{C}_m(q_m,\dot{q}_m))\,\Lambda\,q_m(t) \\ &\quad - (g_m(q_m) - \hat{g}_m(q_m)) - (Fv_m - \hat{F}v_m)\dot{q}_m(t) \\ B_s(q_s)\dot{\sigma}_s(t) + C_s(q_s,\dot{q}_s)\sigma_s(t) &= \bar{\tau}_s(t) + (B_s(q_s) - \hat{B}_s(q_s))\,\Lambda\dot{q}_s(t) \\ &\quad + (C_s(q_s,\dot{q}_s) - \hat{C}_s(q_s,\dot{q}_s))\,\Lambda\,q_s(t) \\ &\quad - (g_s(q_s) - \hat{g}_s(q_s)) - (Fv_s - \hat{F}v_s)\dot{q}_s(t)\end{aligned} \tag{3.11}$$

The error between the real value and the estimated value, for each of the terms in relation (3.8), is given by:

$$\begin{aligned}\tilde{B} = B_i - \hat{B}_i; \qquad \tilde{C} = C_i - \hat{C}_i; \qquad \tilde{g} = g_i - \hat{g}_i; \\ \tilde{F}v_i = Fv_i - \hat{F}v_i; \qquad \tilde{\pi}_i(t) = \pi_i - \hat{\pi}_i(t)\end{aligned} \tag{3.12}$$

In a similar manner as before, the matrix $Y_i(q_i,\sigma_i)$ is obtained, yielding:

$$\begin{aligned}B_m(q_m)\dot{\sigma}_m(t) + C_m(q_m,\dot{q}_m)\sigma_m(t) &= \bar{\tau}_m(t) + Y_m(q_m,\sigma_m)\,\tilde{\pi}_m(t) \\ B_s(q_s)\dot{\sigma}_s(t) + C_s(q_s,\dot{q}_s)\sigma_s(t) &= \bar{\tau}_s(t) + Y_s(q_s,\sigma_s)\,\tilde{\pi}_s(t)\end{aligned} \tag{3.13}$$

Furthermore, the update law of the time-varying estimates is given by:

$$\begin{aligned}\dot{\hat{\pi}}_m(t) &= \Gamma\; Y_m^{\mathrm{T}}(q_m,\sigma_m)\;\sigma_m(t); \\ \dot{\hat{\pi}}_s(t) &= \Phi\; Y_s^{\mathrm{T}}(q_s,\sigma_s)\;\sigma_s(t)\end{aligned} \tag{3.14}$$

where $\Gamma, \Phi \in R^{p \times p}$ are constant positive definite matrices.

It worth mentioning that the initial constant values of the parameter vector π_i are given in appendix (A.8). These constant values were acquired through system identification, and were then used as initial values for the integrator in building the adaptive controller at hand, as it is described by the update law.

4

Simulations and Experiments

In this chapter, the simulation and experimental results are illustrated and discussed. In the simulation, tuning the control parameters was a relatively simple task. However, this task was challenging in the experiment. Therefore, after implementing the controller onto the 3-DOF manipulator and setting the controller gains, the same values were used again in the simulation in order to obtain more comparable results.

It is wroth mentioning that what determines which of the two manipulators is the master and which is the slave depends on which side is held by the operator. Therefore, in the case of total free motion, it might occasionally seem that the slave is leading the master. This is due to the fact that both sides are trying to synchronize without the master side being constrained by the operator's hand.

4.1 Simulation

Different scenarios were considered here. Mainly, simulations were divided into two categories, one is free motion (i.e. neither human influence nor constraints to the environment), while the other case was subject to human and environmental forces. In the following subsections, simulation results of both cases will be illustrated and discussed.

4.1.1 Free motion

For this category, the values of the human and environmental forces were $F_h = F_e = 0$. Two cases have been considered, with regards to time delay, symmetric and asymmetric constant communication delays. Following the remarks of Chopra et al. in [21], the synchronization conditions are satisfied even if the forward and backward delays are not equal. Delays in SIMULINK environment were represented by the *Transport Delay* block.

4.1.1.1 Symmetric constant time delay

First, the performance in the case of no communication delay is illustrated (i.e. $T = 0$ sec). Results are in figures 4.1, 4.2 and 4.3, while table 4.1 shows the mean errors of positions and velocities. It can be seen that in general, the manipulators

could perfectly synchronize.

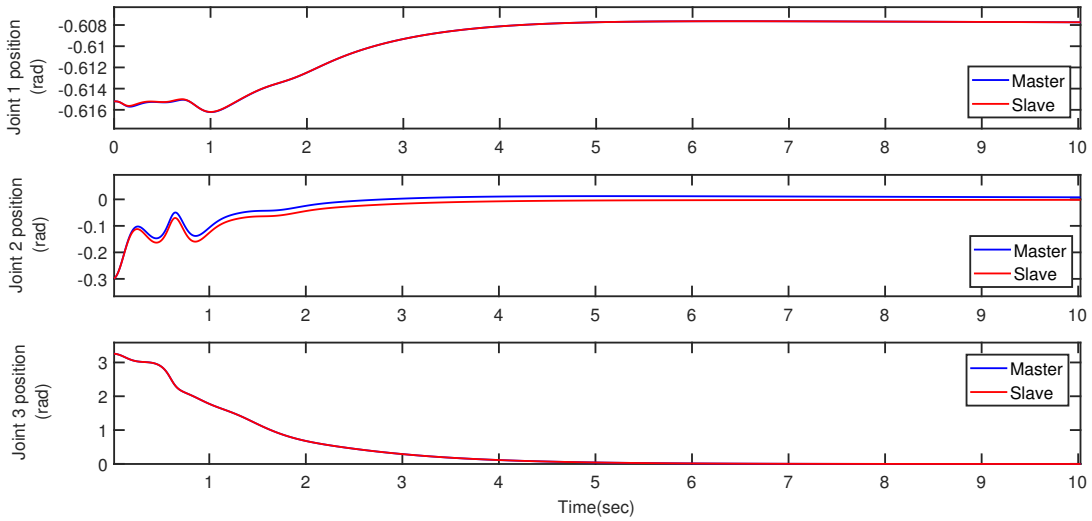


Figure 4.1: Joints positions during free motion with no communication delay

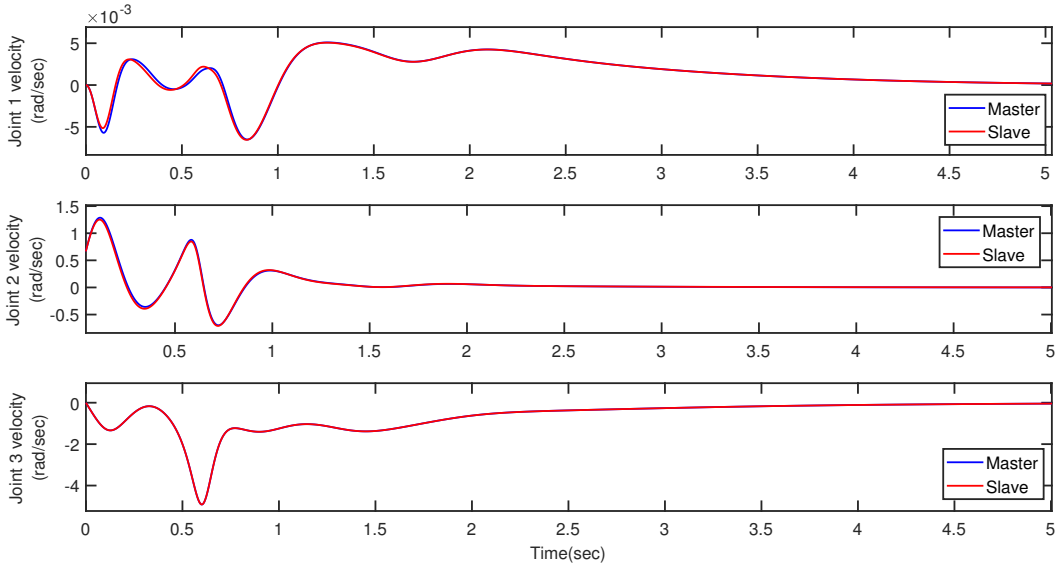


Figure 4.2: Joints velocities during free motion with no communication delay

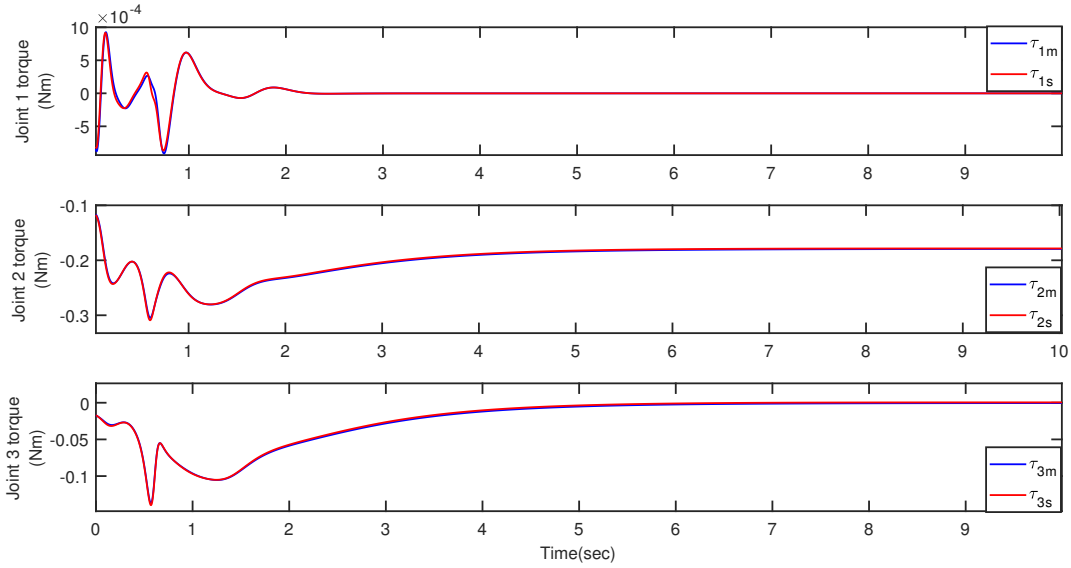


Figure 4.3: Input torque during free motion with no communication delay

Category	Mean error
Joint 1 position	0
Joint 2 position	0.0077
Joint 3 position	0
Joint 1 velocity	0
Joint 2 velocity	0
Joint 3 velocity	0

Table 4.1: Mean error of positions and velocities in the case of no communication delay

Then, a constant time delay of $T = 1$ sec was applied for both forward and backward communication delays. Table 4.2 shows the mean error of the position and velocity synchronization, between the master and the slave. Moreover, figures 4.4, 4.5, 4.6 and 4.7 show the simulation results of the symmetric constant communication delay. Both positions and velocities could be synchronized within approximately 15 sec.

Category	Mean error
Joint 1 position	0
Joint 2 position	0.0071
Joint 3 position	-0.0005
Joint 1 velocity	0
Joint 2 velocity	0
Joint 3 velocity	0

Table 4.2: Mean error of positions and velocities during symmetric time delay of 1 sec

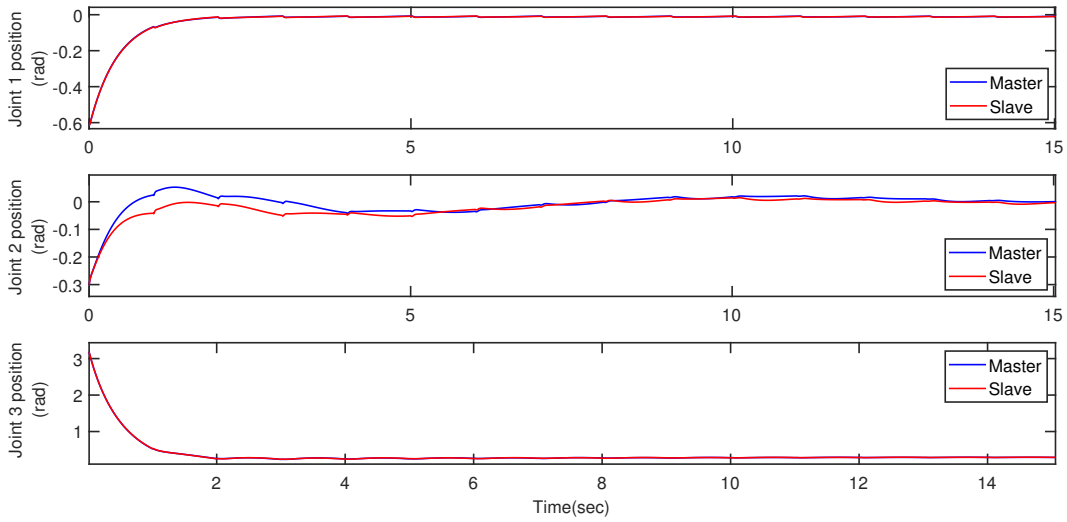


Figure 4.4: Joints positions during free motion with symmetric communication delay

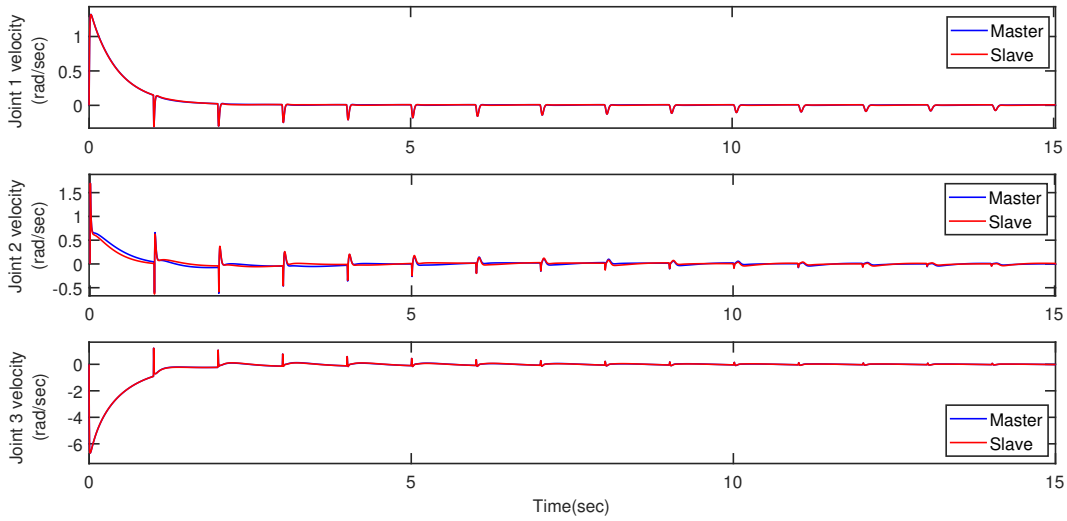


Figure 4.5: Joints velocities during free motion with symmetric communication delay

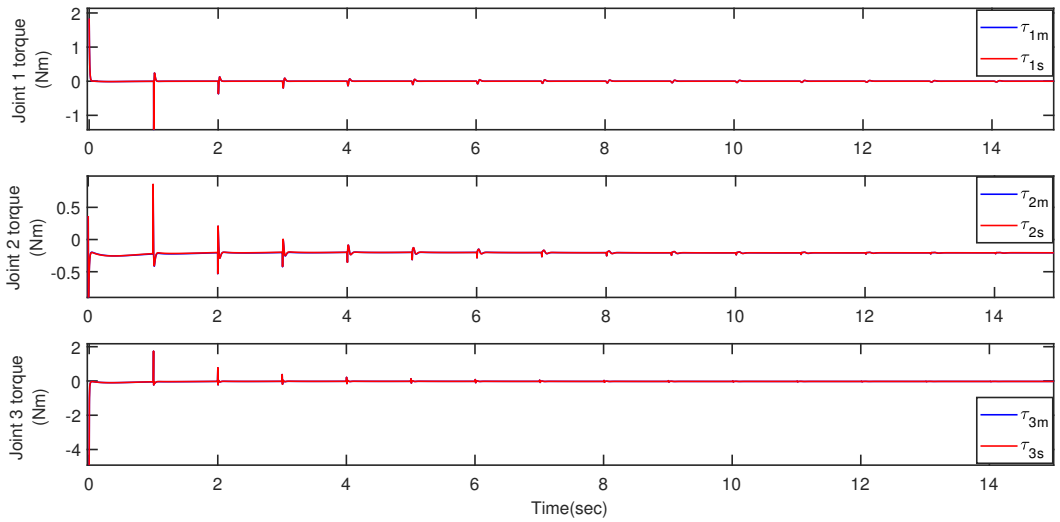


Figure 4.6: Input torque during free motion with symmetric communication delay

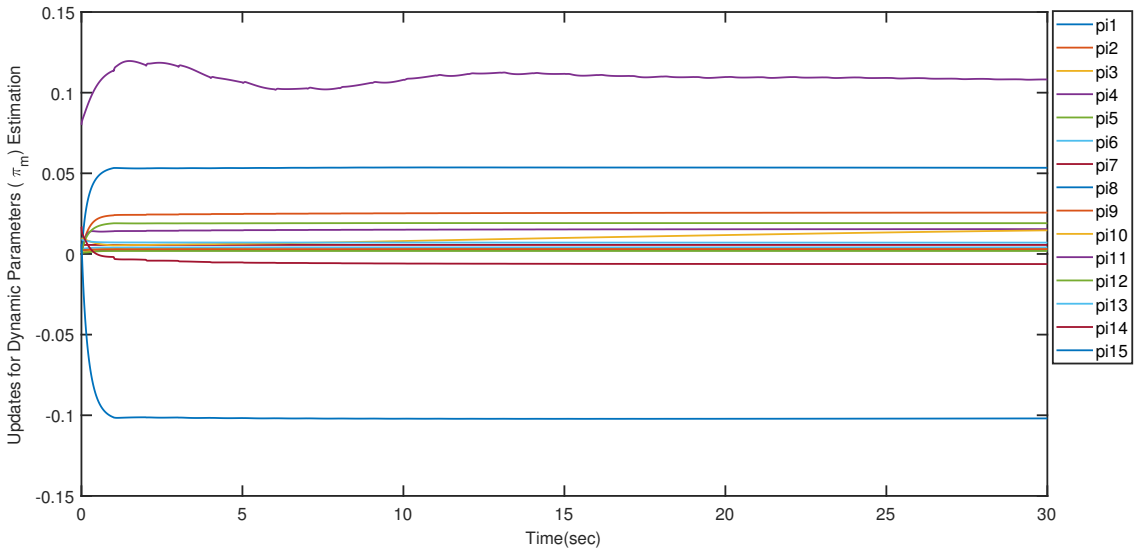


Figure 4.7: Evolution of dynamic parameters estimation during free motion with symmetric communication delay

4.1.1.2 Asymmetric constant time delay

In this case, a forward delay of 1 second and a backward delay of 1.2 seconds were applied. Mean errors are in table 4.3, while results are in figures 4.8, 4.9 and 4.10. One can notice that regardless of the asymmetric delay, synchronization could be guaranteed, within around 20 seconds.

For all delay cases, the tuned PD parameters λ and K, and the position error gain K_p had the values below. It should be noted that the highest weights were assigned to the heaviest joint (i.e. joint 1).

$$\lambda = \begin{bmatrix} 1.3 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}; \qquad K = \begin{bmatrix} 1.3 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.7 \end{bmatrix}; \qquad K_p = \begin{bmatrix} 1.3 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}$$

Category	Mean error
Joint 1 position	0.0001
Joint 2 position	0.0073
Joint 3 position	-0.0011
Joint 1 velocity	0.0001
Joint 2 velocity	0
Joint 3 velocity	0

Table 4.3: Mean error of positions and velocities during asymmetric constant time delay

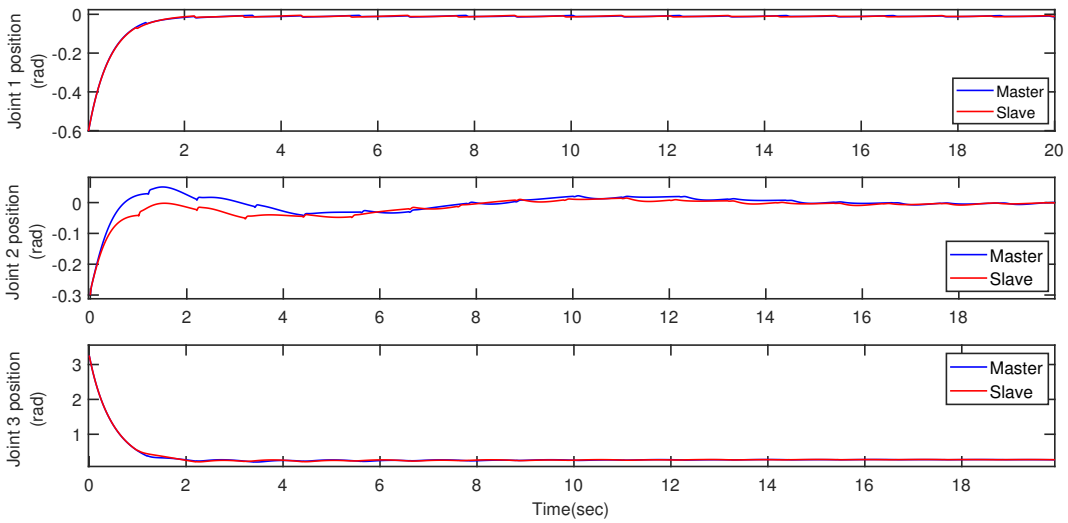


Figure 4.8: Joints positions during free motion with asymmetric communication delay

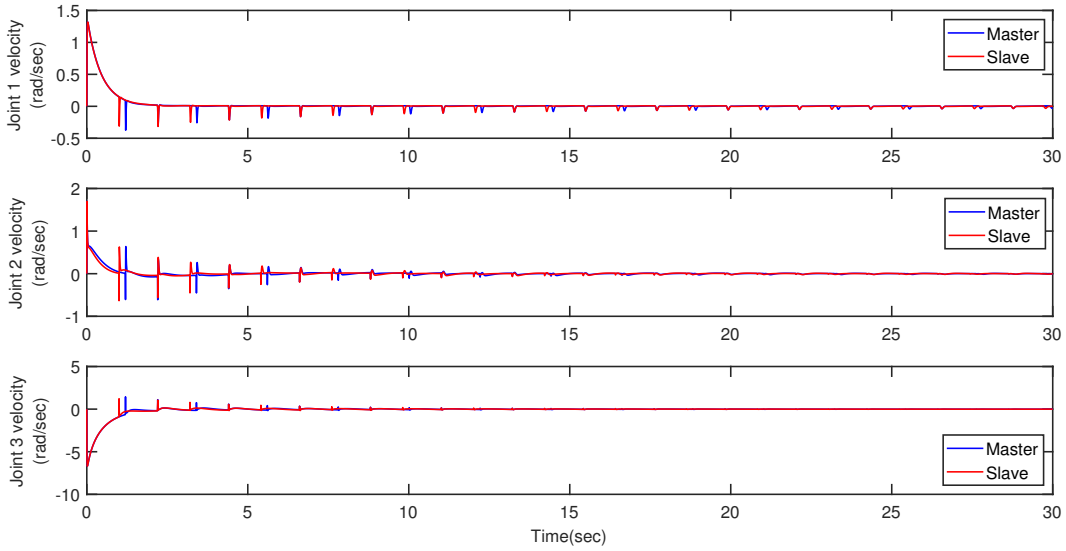


Figure 4.9: Joints velocities during free motion with asymmetric communication delay

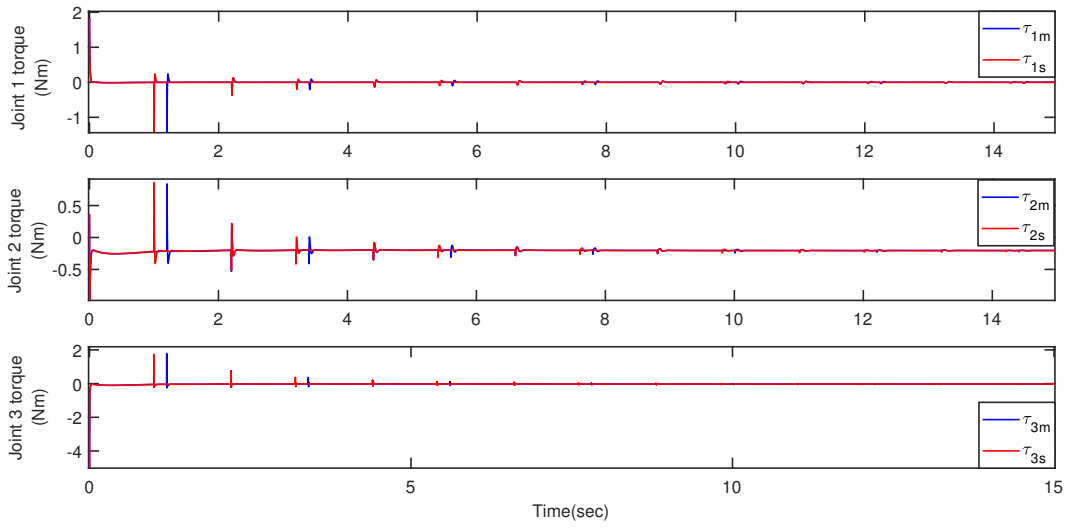


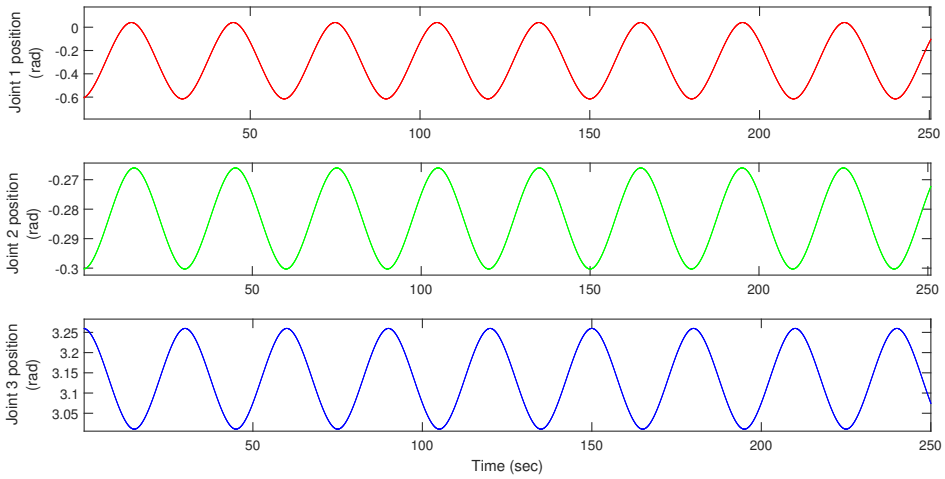
Figure 4.10: Input torque during free motion with asymmetric communication delay

4.1.2 Human operator and constrained motion

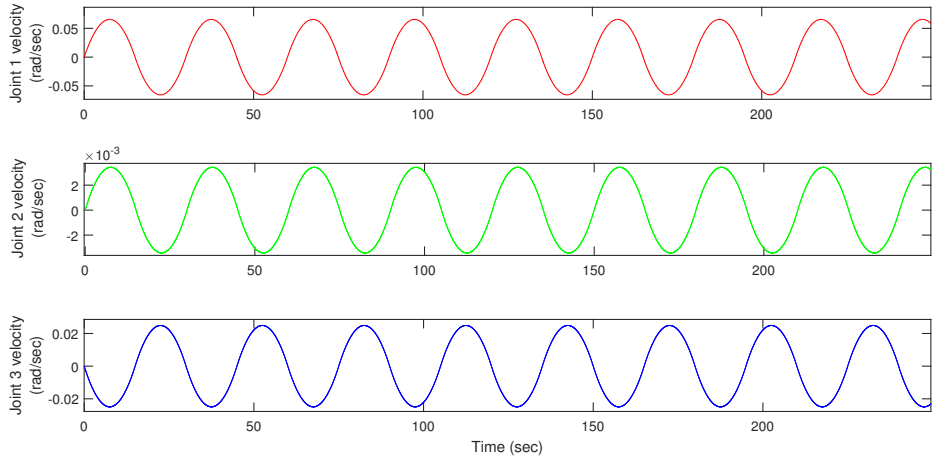
For simulating the controller behaviour with a human operator in the control loop, a desired trajectory has been generated to represent the human operator's intention, which was modelled as explained in chapter 2. To represent the task of cutting a soft tissue in a surgery, the end-effector was commanded to move back and forth between 2 points (in this case points 1 and 5 - figure 4.11). Figures 4.12a and 4.12b show the position and velocity of the human-generated trajectory. As discussed in chapter 2, the values of stiffness and damping parameters should be high since the intended task requires slower and finer motion.

Figure 4.11: Human trajectory for the simulation task (the blue line represents the motion of the end-effector)

Two scenarios were considered, human control while the slave experiences free motion, and while its motion is constrained by a rigid wall. Both simulations were run while there is symmetric delay of $T = 1$ sec in the communication channel.



(a) Human trajectory (position)



(b) Human trajectory (velocity)

Figure 4.12: Position and velocity of the human trajectory

4.1.2.1 Human operating on a free environment

As mentioned in chapter 2, the human operator was modelled as a PD system. Since the motion is only on the xy-plane (figure 4.11), the stiffness and damping generated forces were applied only to the xy-direction. Following the recommendations of Lee and Spong in [16], the values of the stiffness and damping parameters were chosen as follows:

$$C_h = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \qquad K_h = \begin{bmatrix} 75 & 0 & 0 \\ 0 & 75 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.1}$$

The results are in figures 4.13, 4.14, 4.15 and 4.16 below, while the synchronizations mean errors are in table 4.4. Approximately, synchronization was achieved by the first 15 seconds.

Category	Mean error
Joint 1 position	-0.0154
Joint 2 position	0.0060
Joint 3 position	-0.0001
Joint 1 velocity	-0.0012
Joint 2 velocity	-0.0002
Joint 3 velocity	-0.0003

Table 4.4: Mean error of positions and velocities when simulating a human operating on a free environment

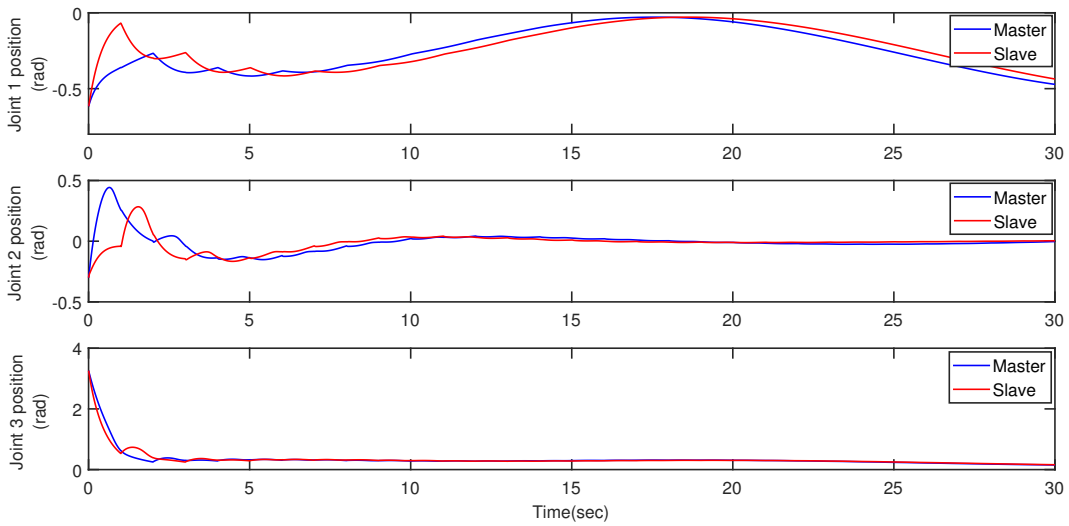


Figure 4.13: Joints positions with human operating on a free environment

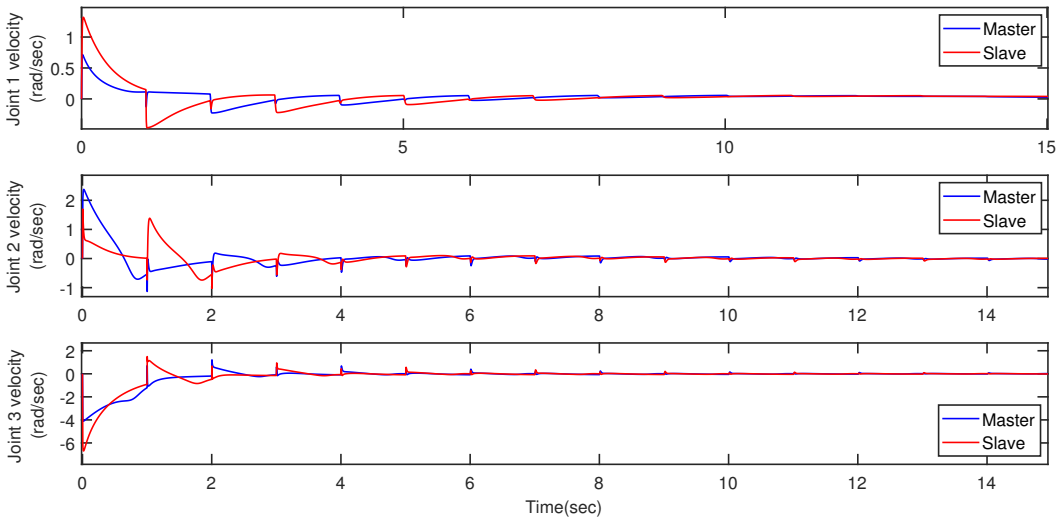


Figure 4.14: Joints velocities with human operating on a free environment

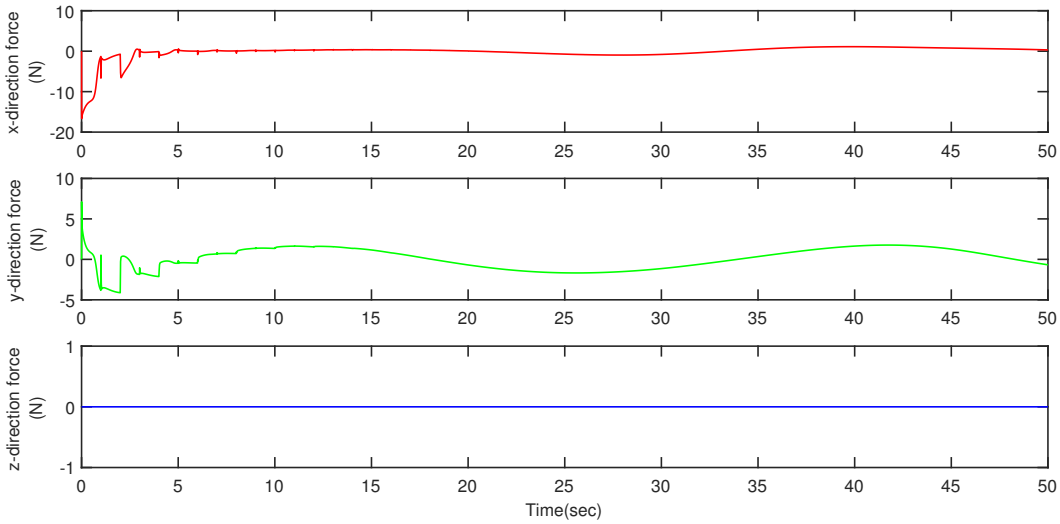


Figure 4.15: Human generated forces

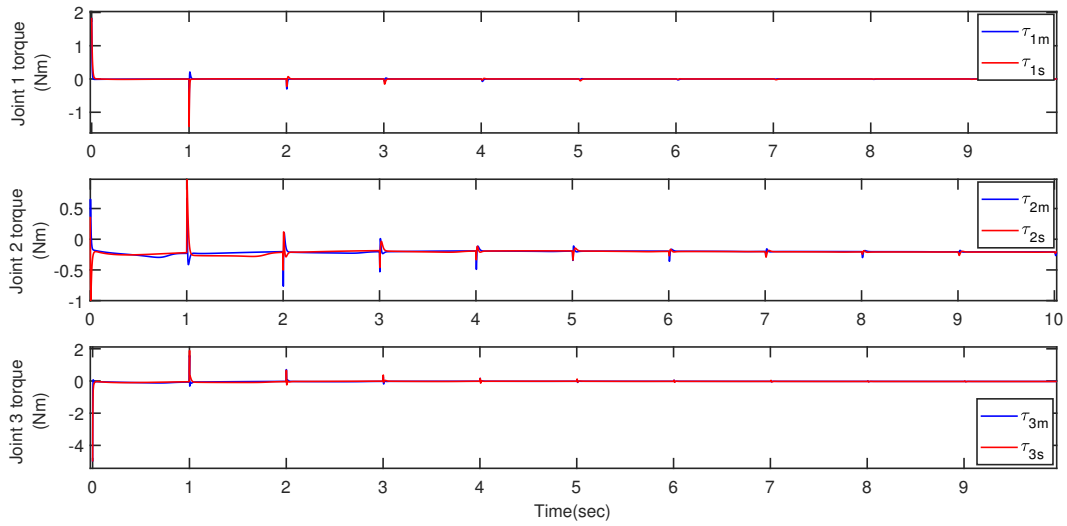


Figure 4.16: Input torques to the master and the slave when a human is operating on a free environment

4.1.2.2 Human operating on a constrained environment

In the case of constrained environment, only a rigid wall constraint was simulated and experimented. The wall was located on point 5 on the manipulator's board (figure 4.11), constraining the movement of the end-effector in the x-direction. Subsequently, the stiffness and damping parameters were set as follows [16]:

$$C_e = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \qquad K_e = \begin{bmatrix} 500 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.2}$$

The constraint was applied during the simulation time of (25-29) seconds. Figures 4.17, 4.18, 4.19, 4.20 and 4.21 below show the simulation results. The update of the dynamic parameters estimation is shown in figure 4.22, with the final values in relation (4.3), and the initial values in appendix A, relation (A.8). Also, the synchronizations mean errors are in table 4.5.
It can be seen that when hitting the constraint, a position and velocity mismatch occurred, with a slight increase in the human generated forces. However, the master and slave could synchronize again within approximately 10 seconds, although it took longer for joint 2 position. By observing the real experiment in the following section, one can notice the same performance for the position of joint 2. Hence, it can be attributed to the mechanical structure and motion of joint 2.

Category	Mean error
Joint 1 position	-0.0116
Joint 2 position	0.4914
Joint 3 position	-0.0401
Joint 1 velocity	0.0005
Joint 2 velocity	0.0009
Joint 3 velocity	0.0003

Table 4.5: Mean error of positions and velocities when simulating a human operating on a constrained environment

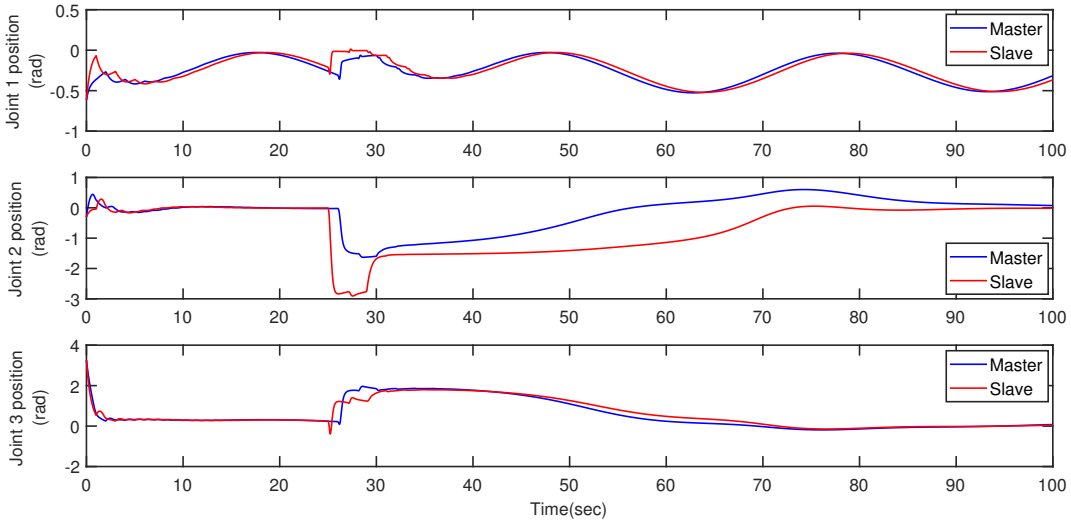


Figure 4.17: Joints positions with human operating on constrained environment

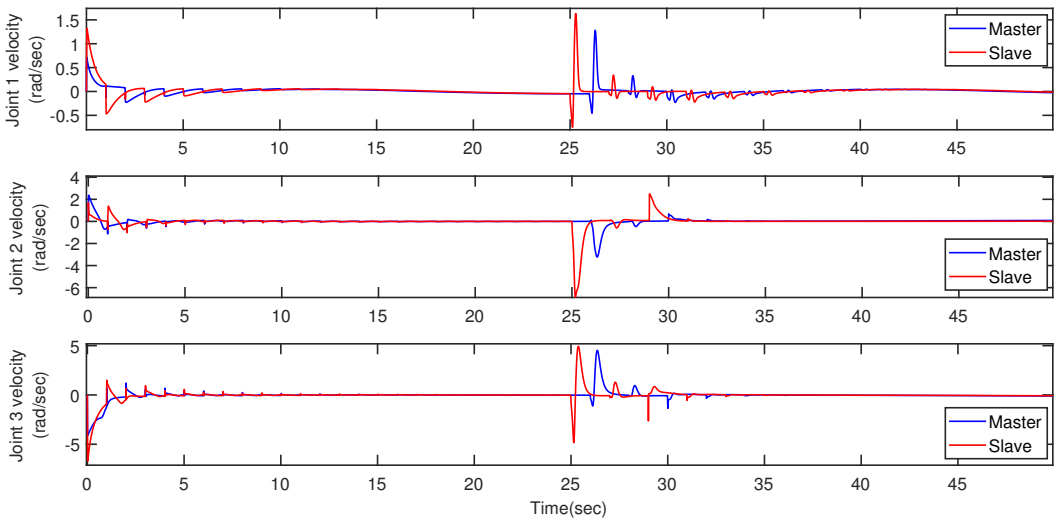


Figure 4.18: Joints velocities with human operating on constrained environment

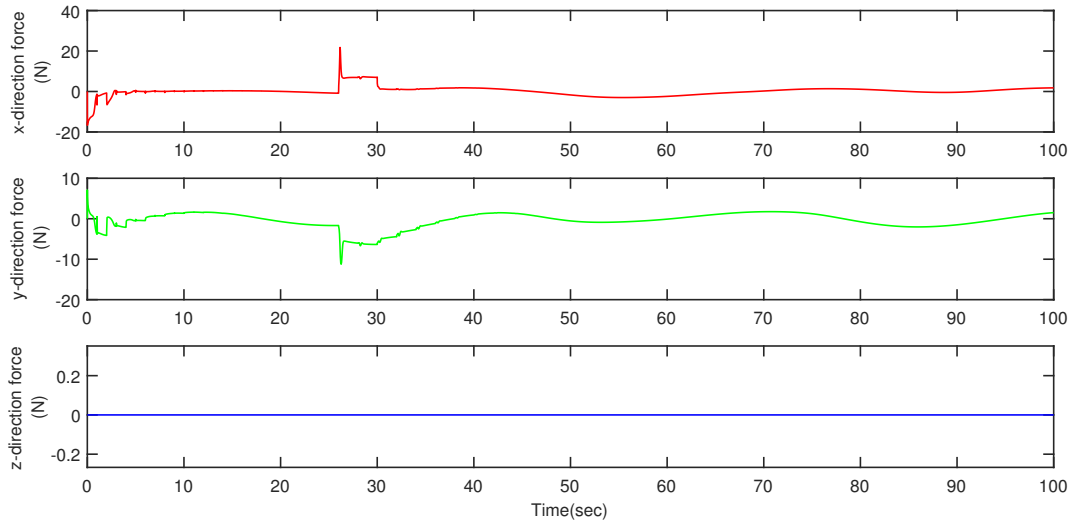


Figure 4.19: Human generated forces

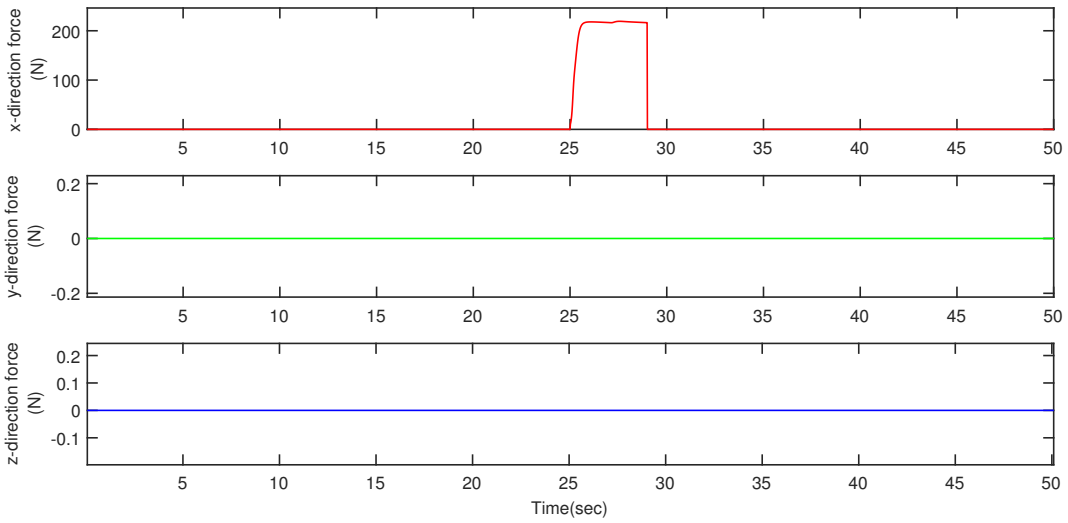


Figure 4.20: Environment reflected forces

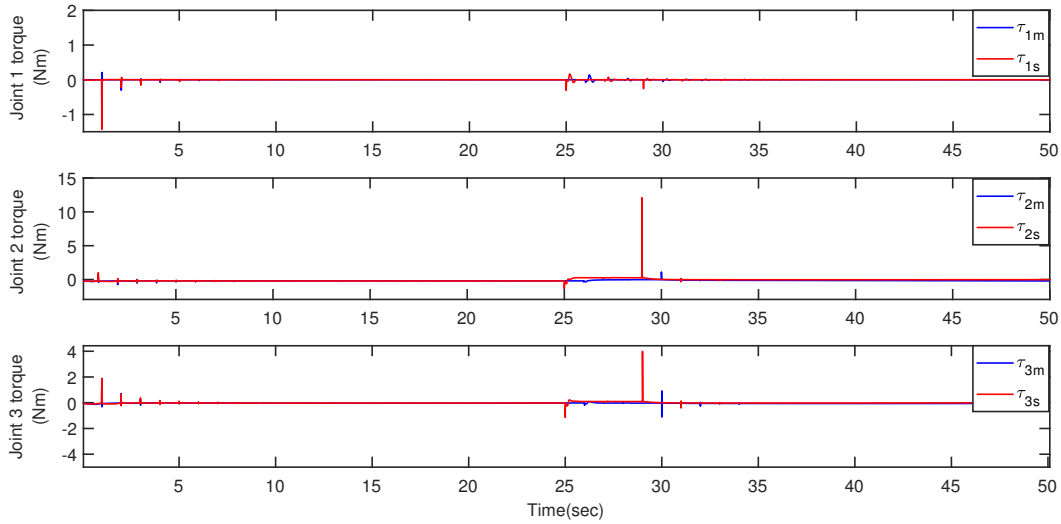


Figure 4.21: Input torques with human operating on constrained environment

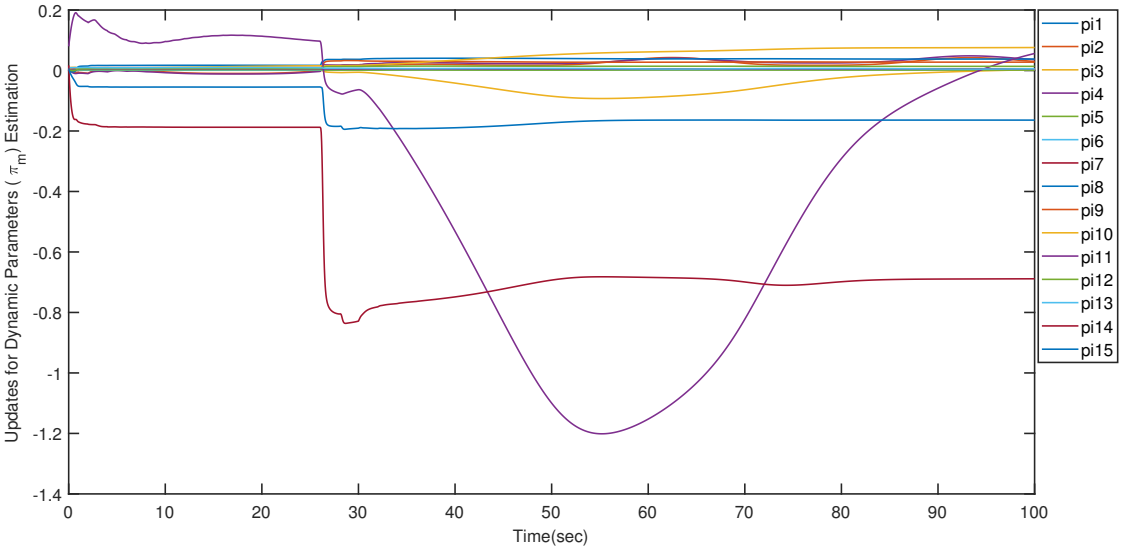


Figure 4.22: Evolution of dynamic parameters estimation during constrained motion

The final values of the estimated dynamic parameters (on the master side) were:

$$\hat{\pi}_m = \begin{bmatrix} IL1_x \\ IL1_y \\ l_1^2 m_2 \\ IL2_x \\ IL2_y \\ Isp_y \\ l_1 m_3 \\ IL2_z \\ IL1_z \\ l_2 m_3 \\ m_3 \\ l_2^2 m_3 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 0.0029 \\ 0.0299 \\ 0.0027 \\ 0.0360 \\ 0.0015 \\ 0.0047 \\ 0.0047 \\ 0.0056 \\ 0.0379 \\ 0.0759 \\ 0.0562 \\ 0.0137 \\ 0.0057 \\ -0.6887 \\ -0.1640 \end{bmatrix} \tag{4.3}$$

4.2 Experiments

For the experiments, two 3D Systems Geomagic® Touch™ manipulators were used to represent the master and slave teleoperation system. The master side was controlled by a human operator, while the slave side was constrained by a rigid aluminum wall. The teleoperators were connected using User Datagram Protocol (UDP). The detailed practical steps on how to establish such connection are documented in appendix B. The controller designed in chapter 3 was implemented and experimented. The round trip delay was around 0.32 seconds. This delay was measured by sending an impulse signal from the master manipulator to the slave manipulator, and measuring the delay time necessary for the master to receive the reflected impulse from the slave side.

4.2.1 Free motion

First task to be tested was the synchronization of position and velocity when the slave is experiencing free-motion. Figures 4.23 and 4.24 show the synchronization results. It can be seen that the master and slave could synchronize with reasonable mean errors, which are shown in table 4.6.

Category	Mean error
Joint 1 position	0.0006
Joint 2 position	-0.0156
Joint 3 position	-0.0271
Joint 1 velocity	-0.0007
Joint 2 velocity	0.0064
Joint 3 velocity	-0.0977

Table 4.6: Mean error of positions and velocities during free-motion experiment

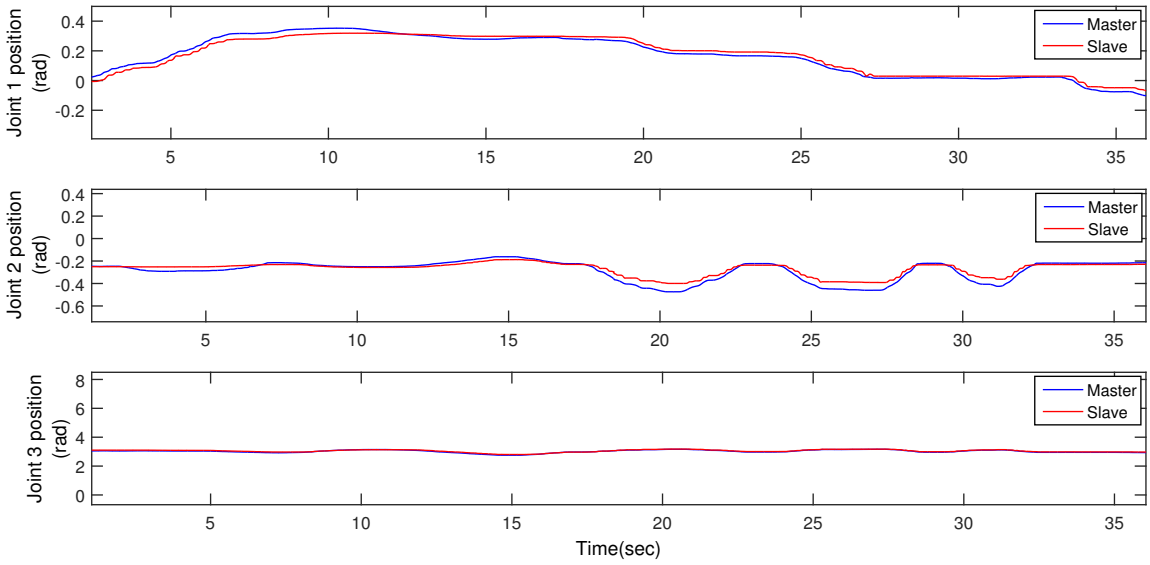


Figure 4.23: Synchronization of joints positions during free motion

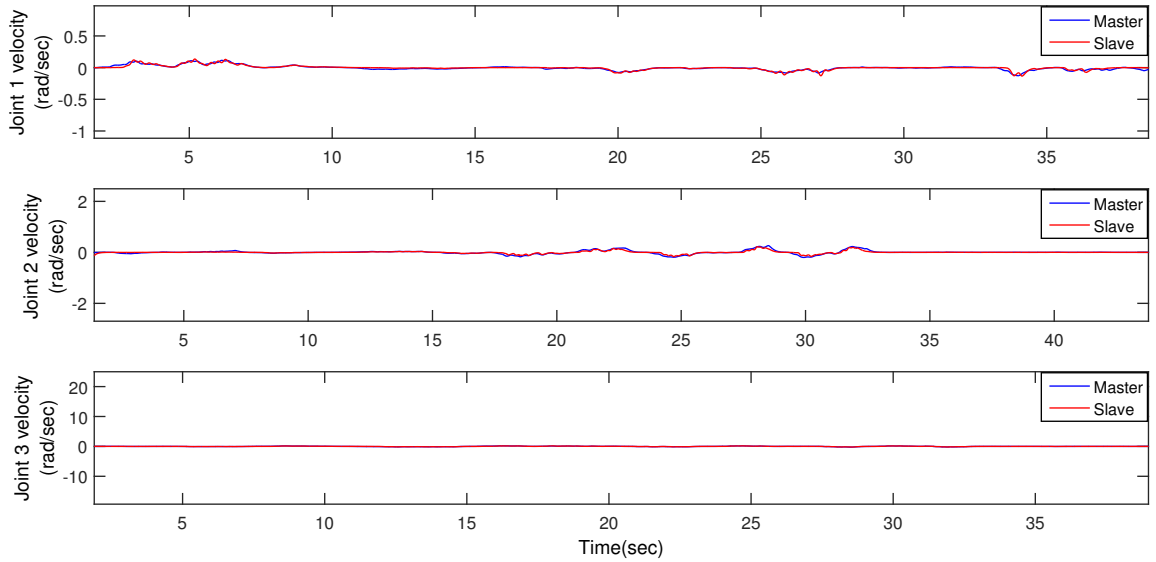


Figure 4.24: Synchronization of joints velocities during free motion

4.2.2 Constrained motion

In this subsection, synchronization of position and velocity and boundedness of the tracking error were investigated, whilst the slave is experiencing a rigid aluminum wall, located along points 2, 5 and 8 (figure 4.11).

During the experiment, it was noticed that the reflected forces of hitting a rigid wall were too small. In order to have forces large enough for the user to feel, K_p, in equation (3.7), was set to be scaled by a factor whenever the position error between the two manipulators exceeds a predefined threshold. Since the constraint is located at the x-direction, the position error of joint 1 is the main reference point for the scaled stiffness values, and subsequently the direction of the reflected torque experienced by the operator when the slave hits a wall is the x-direction.

Figures 4.25 and 4.26 illustrate the position and velocity synchronization during constrained motion. The contact with the wall was during the time between (35-45) and (47-50) seconds. It can be seen that after hitting the wall, the two manipulators were able to re-synchronize their position and velocity, with small fluctuations. The master and slave torques during constrained motion are in figures 4.27 and 4.28. One can notice the increase in input torques to the master when hitting the wall in the remote environment, which is physically translated to forces felt by the user.

The position and velocity errors throughout the trajectory, including the free and constrained motion parts, are illustrated in figures 4.29 and 4.30.

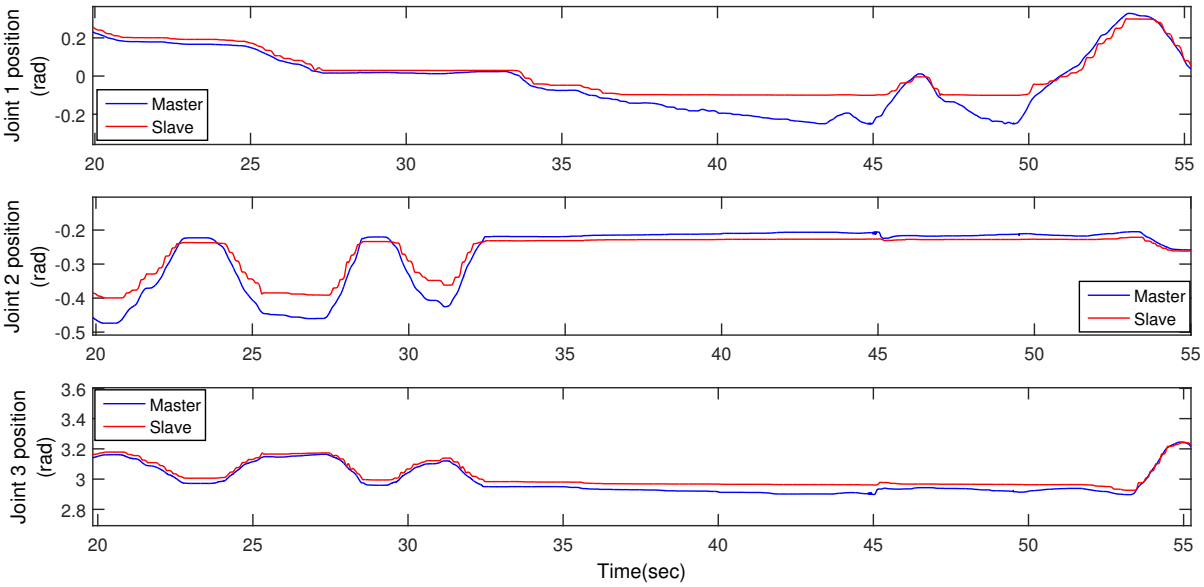


Figure 4.25: Synchronization of joints positions during constrained motion

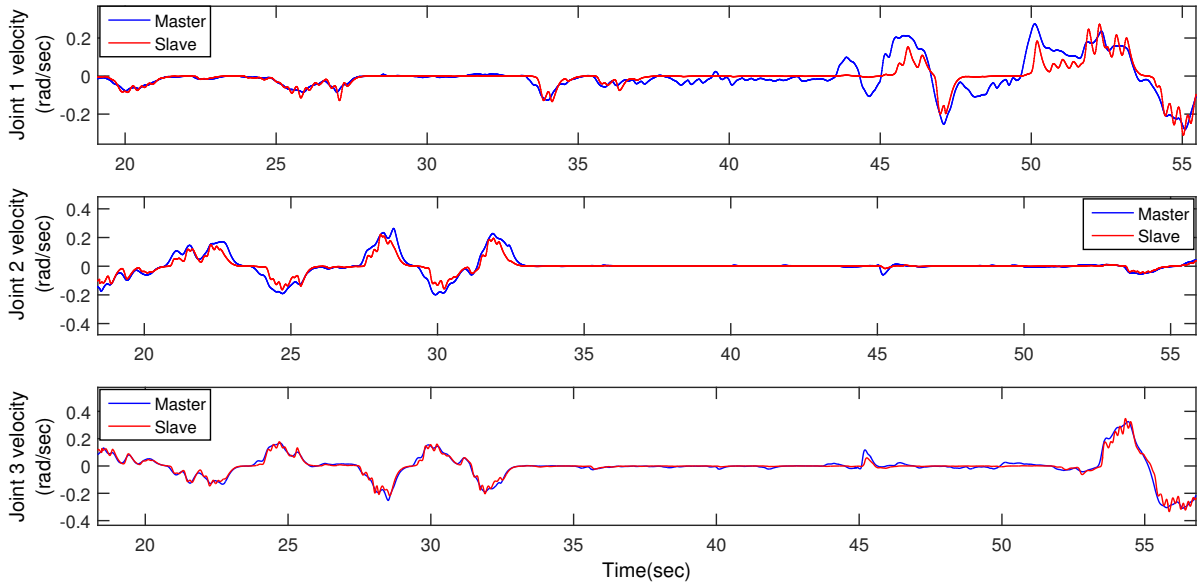


Figure 4.26: Synchronization of joints velocities during constrained motion

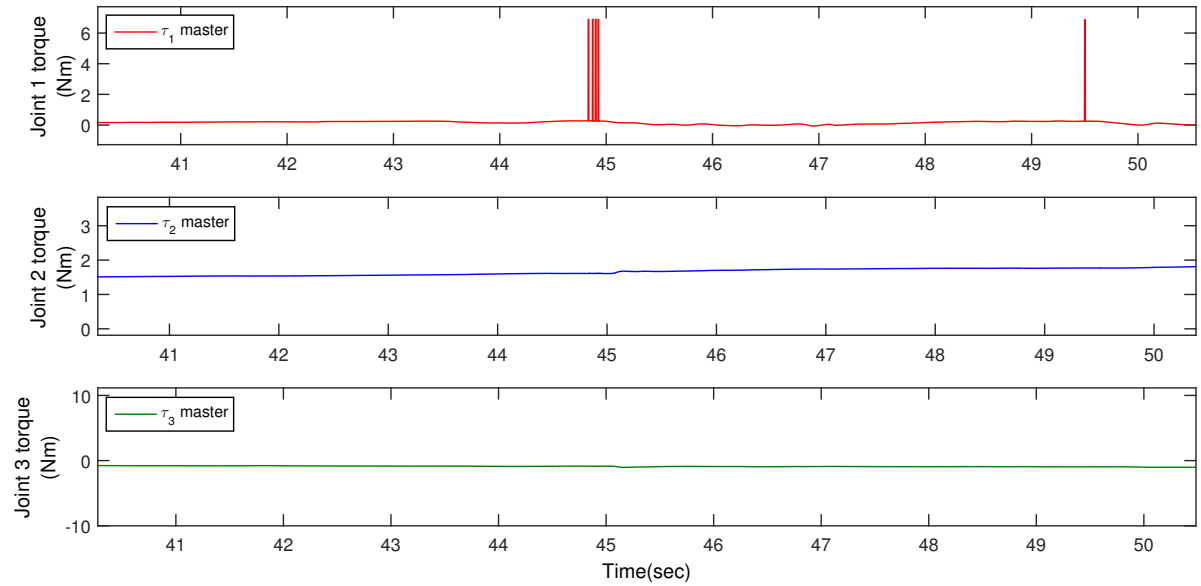


Figure 4.27: Master input torque during constrained motion

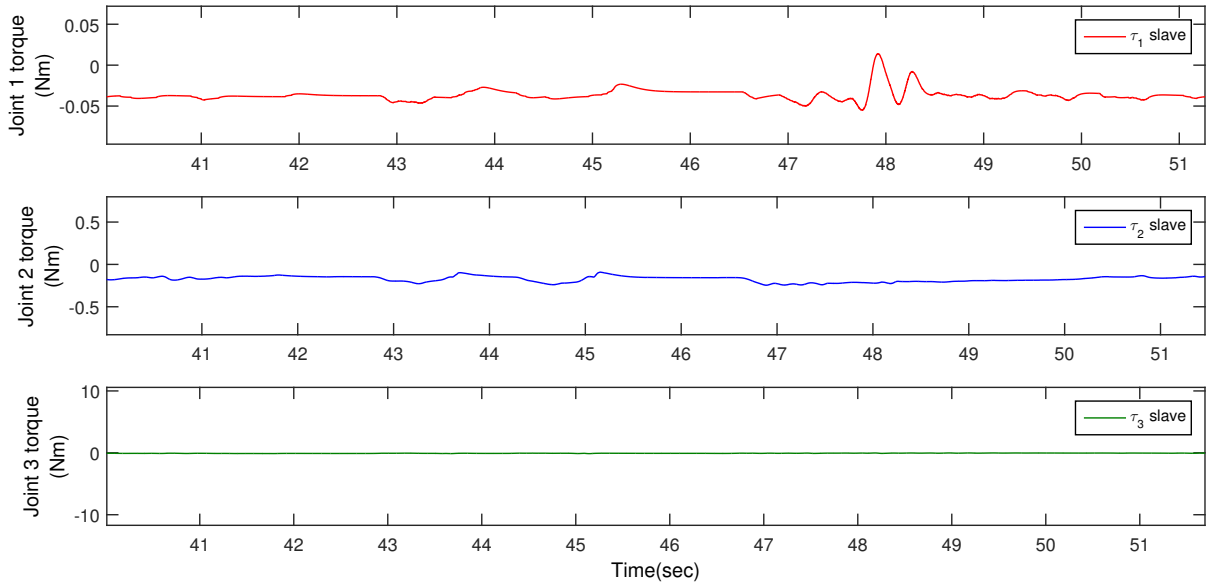


Figure 4.28: Slave input torque during constrained motion

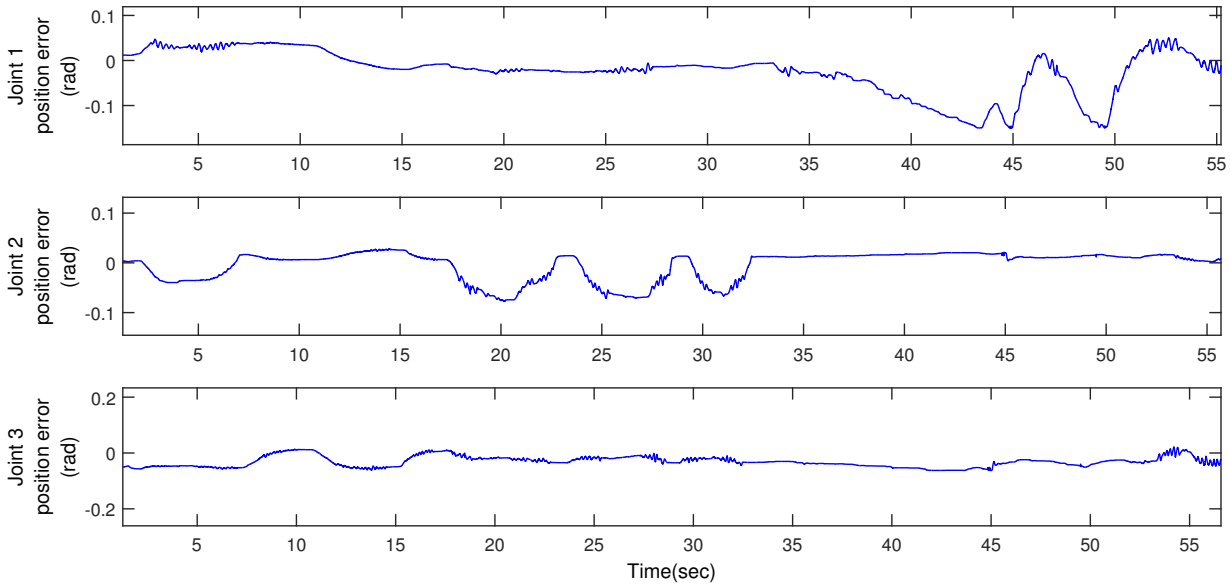


Figure 4.29: Position error during constrained motion

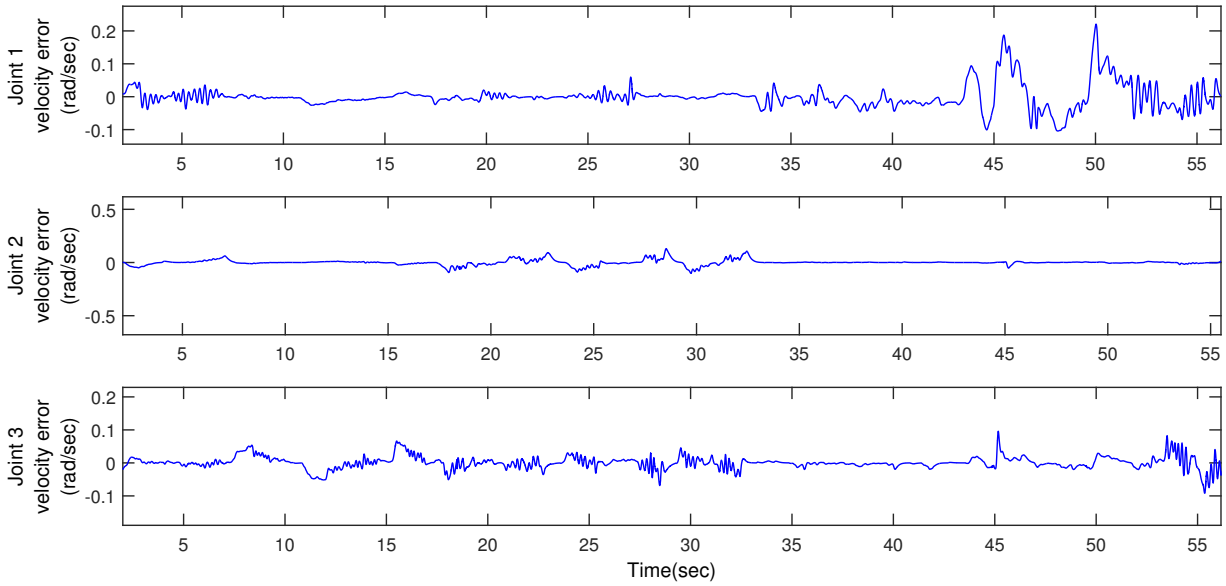


Figure 4.30: Velocity error during constrained motion

4.3 Understanding the delay

Teleoperation over the Internet is always subjected to random time-varying delays, which is sometimes associated with packet loss. Delays can jeopardize the stability of the teleoperated system and deteriorate its performance. When transmitting a discrete-time information, the data packets experience time-varying forward and backwards delays (i.e. T_{in} and T_{out}). Taking the OSI (Open System Interconnection) 7-layer model as a reference model, after passing data through the Network Layer, it gets to the Transport Layer, which is responsible for transmitting data between hosts or systems. Subsequently, a decision should be made, either to use TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) [26].

During transmission, packets can be duplicated, dropped, delayed or arrive out of order. The majority of Internet applications use TCP. It provides reliable communication since it performs error detection and correction, controlling the speed of transmission, detection of congestion in the transmission network, and flow control. However, these features mean that delays in TCP are inevitable, whenever an error occurs or a packet is needed to be retransmitted. Therefore, this attribute can be disadvantageous in real-time applications such as teleoperation. In contrast, UDP provides neither congestion control nor flow control. It only focuses on the fast transmission of data to the receiving end, which makes it more suitable for teleoperated systems. Therefore, UDP was used to connect the master and slave teleoperators in this experiment.

5

Comparative Experiments

In order to compare and evaluate the performance of this controller on other manipulator systems, it has been implemented to a single-link manipulator (i.e. 1-DOF). The system used was the SensoDrive Master-Slave-System (SD-MS1), shown in figure 5.1. It consists of two force-feedback devices, each has motors (optimized for haptic applications) and high-precision position sensor. Adding a torque sensor is optional [4]. For the bilateral communication, a LINUX system with QNX (real-time operating system), was used. The communication was running through a single computer.

In order to test the response of the system, three cases were considered. The first is constrained motion without introducing any time delays. The other two are free motion and constrained motion, both in the case of 0.32 seconds communication delay (as symmetrical round trip delay). It should be clarified that the term "delay-free" actually means a "quasi-delay-free". When there is no delay block introduced to the SIMULINK model, the system delay is one sample time (i.e. $1ms$ for this experiment).

Those experiments were carried out at the German Aerospace Center (DLR), Germany, in the Institute of Robotics and Mechatronics, using the lab facilities of the Telemanipulation Group, under Department of Analysis and Control of Advanced Robotic Systems.

Figure 5.1: SensoDrive Master-Slave System[4]

5.1 Synchronization without time delay

In this part, the synchronization of the master and slave was carried out independent of the communication delays, while the environment is constrained by a rigid wall. As previously mentioned, without introducing a delay block to the SIMULINK model, the system delay is about $1\,ms$. The purpose of this experiment is to test the synchronization and force feedback in a minimal delay setting.
In figure 5.2, a rigid aluminum wall was hit by the slave manipulator between sec (1-2.7), (3.6-5.5) and (6.6-7.4). The position and velocity synchronization errors are shown in figure 5.3, having mean error of -0.0044 rad for position, and 0.0001 rad/sec for velocity.

In order to evaluate system stability, i.e. neither oscillations nor divergence, the master link was pushed and then let free in order to test whether the system will reach stability and synchronize or not, and how long would it take for it to synchronize. In the case of no delays, this was done between (7.8-10.4) sec of figure 5.2. The system was found to be stable.

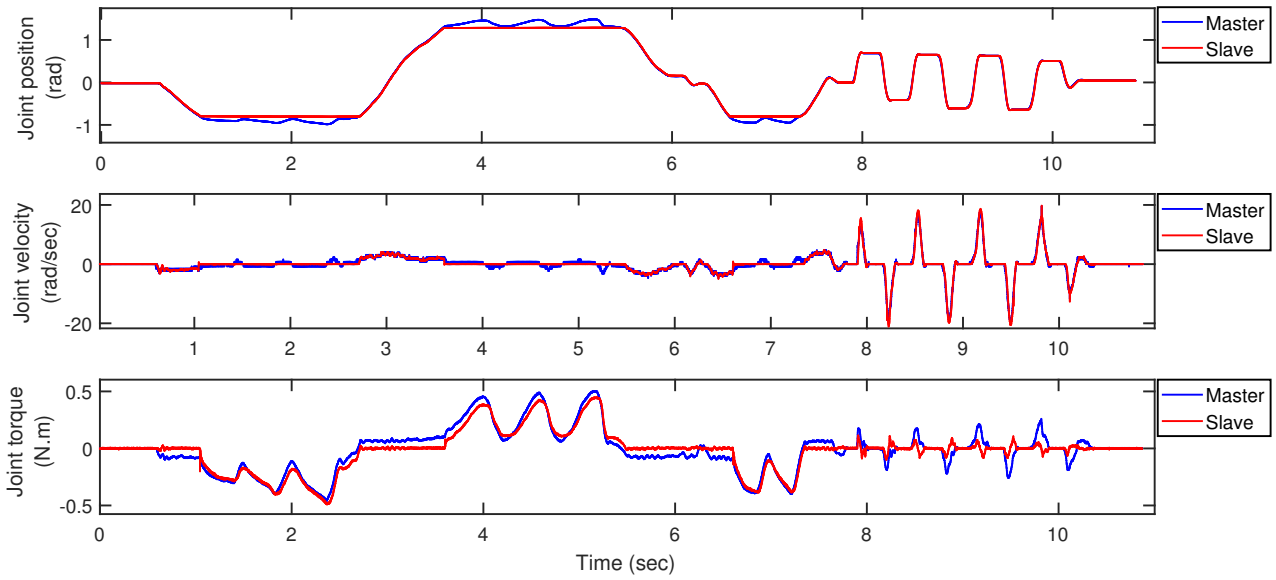


Figure 5.2: Synchronization results for position, velocity and torque without communication delay

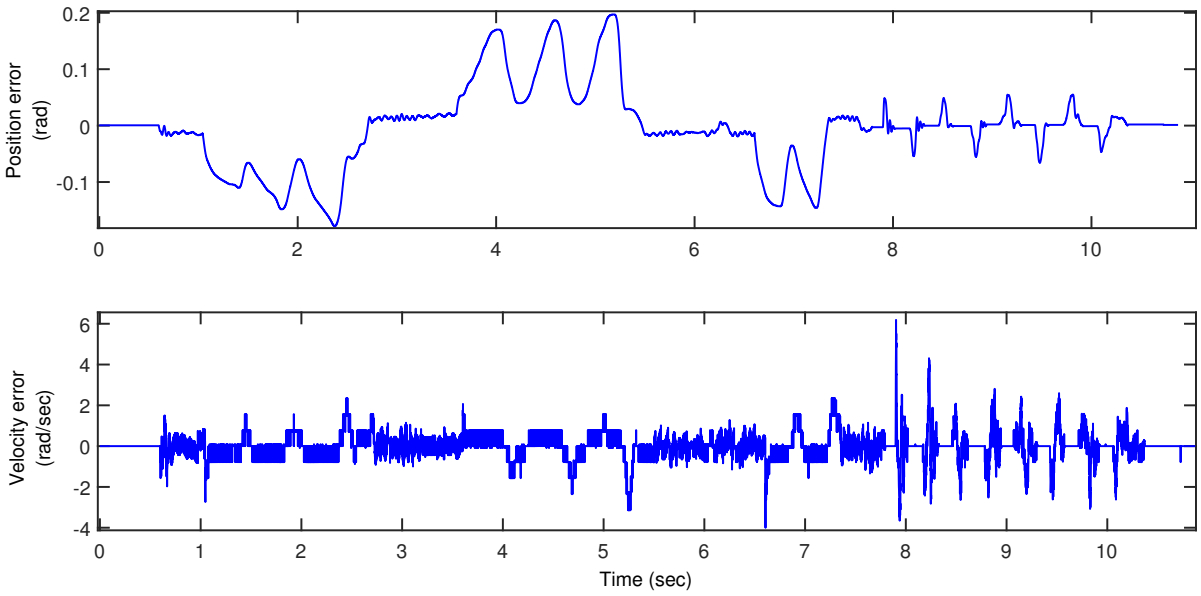


Figure 5.3: Synchronization error for position and velocity without communication delay

5.2 Synchronization with time delay

To evaluate the controller in this setting, a time delay of 0.16 sec was implemented in both directions (i.e. 0.32 sec round trip delay). The delay was introduced using the *Transport Delay* block in SIMULINK.

5.2.1 Free motion

To test system stability while time delay is implemented, the master arm was randomly pushed - and then waited for the slave to synchronize with it. As it is visible in figure 5.4, the master push was recorded at sec 6.4, and it was noticed that the slave was stable and converging to the position and velocity of the master, and continued to approach the synchronization point, until reaching it after about 7 sec. The mean error for the position was (-0.9410×10^{-3}) rad, and for the velocity (0.1267×10^{-3}) rad/sec. The synchronization error over time is illustrated in figure 5.5. In comparison to the performance without delays, it could be noticed that the system felt more damped with increasing delay time, while maintaining stability.

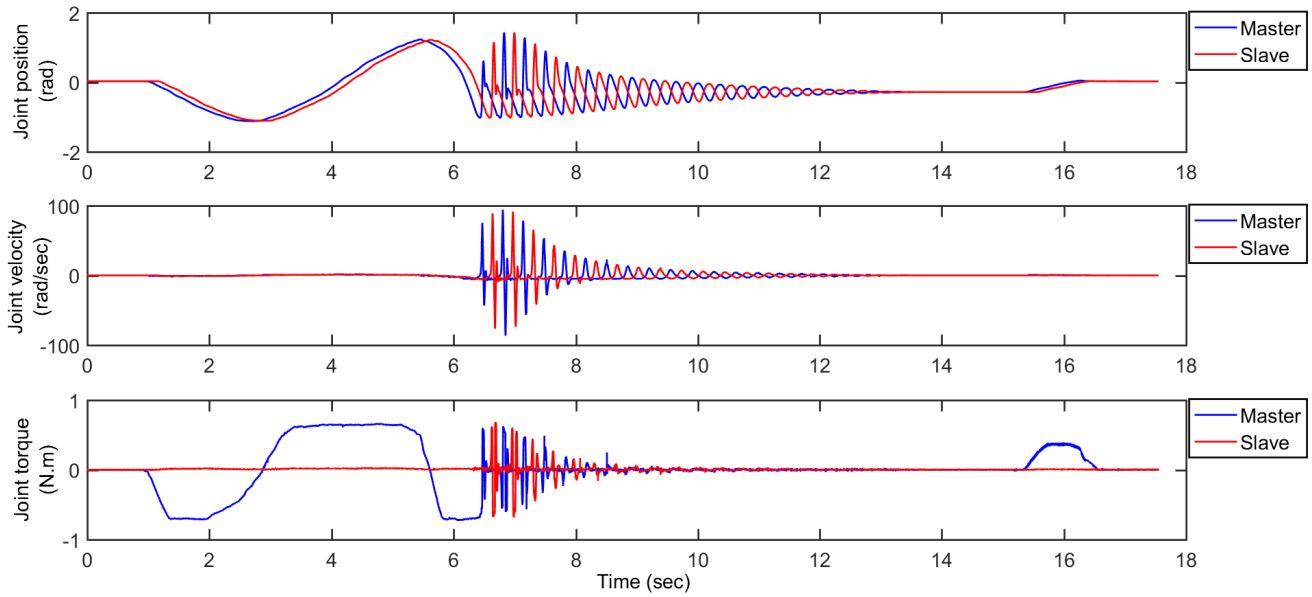


Figure 5.4: Synchronization results for position, velocity and torque with communication delay - free motion

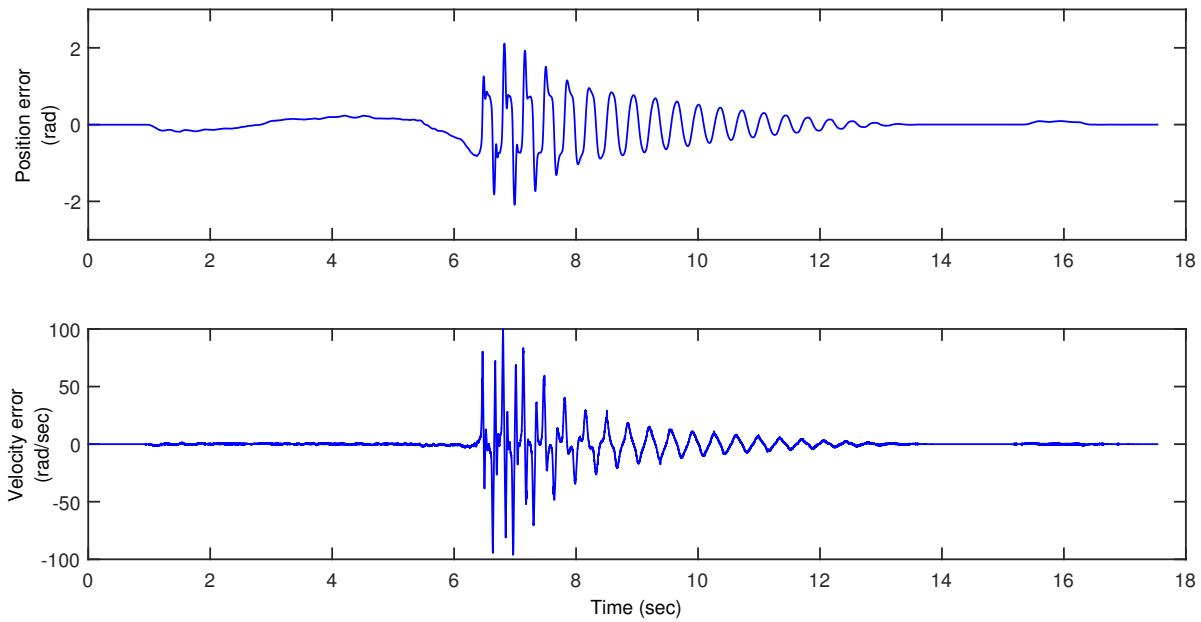


Figure 5.5: Synchronization error for position and velocity with communication delay - free motion

5.2.2 Constrained motion

As seen in figure 5.6, the slave manipulator was constrained by a rigid aluminum wall between seconds (3.4-6.8) and (14-17.7). The system was stable, but in order to improve transparency, the value of K_p (equation (3.7)) needed to be increased. The system had a mean error of 0.0103 rad for position and 0 rad/sec for velocity. The evolution of error through time is in figure 5.7.

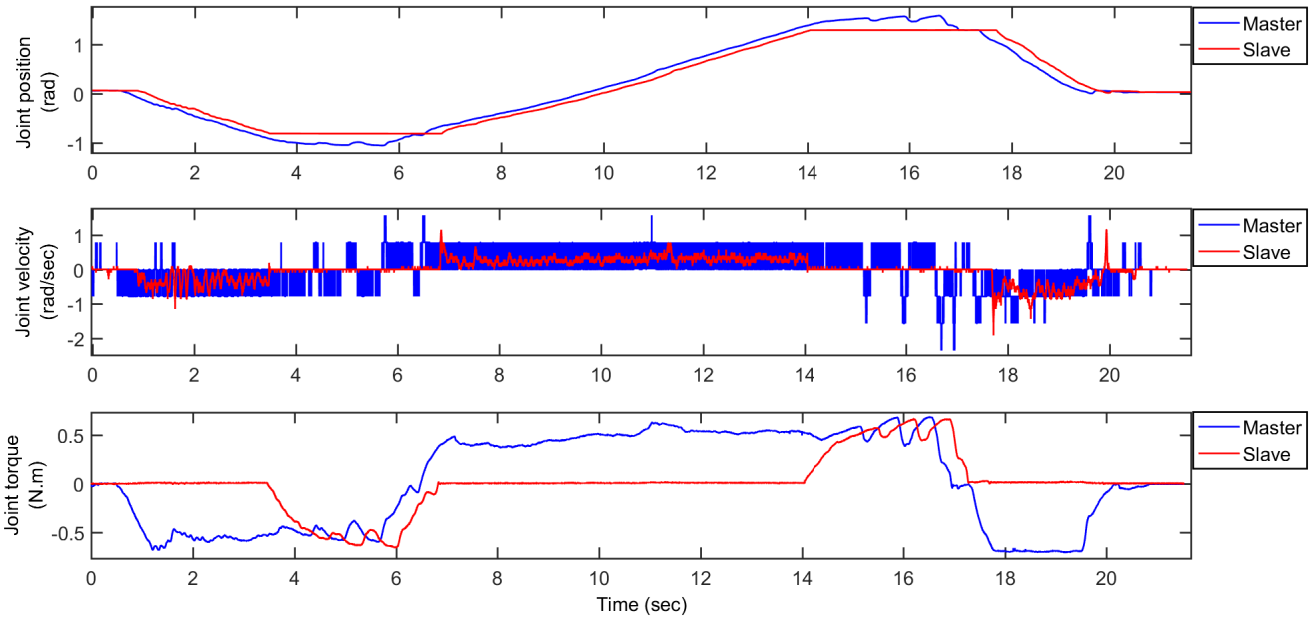


Figure 5.6: Synchronization results for position, velocity and torque with communication delay - constrained motion

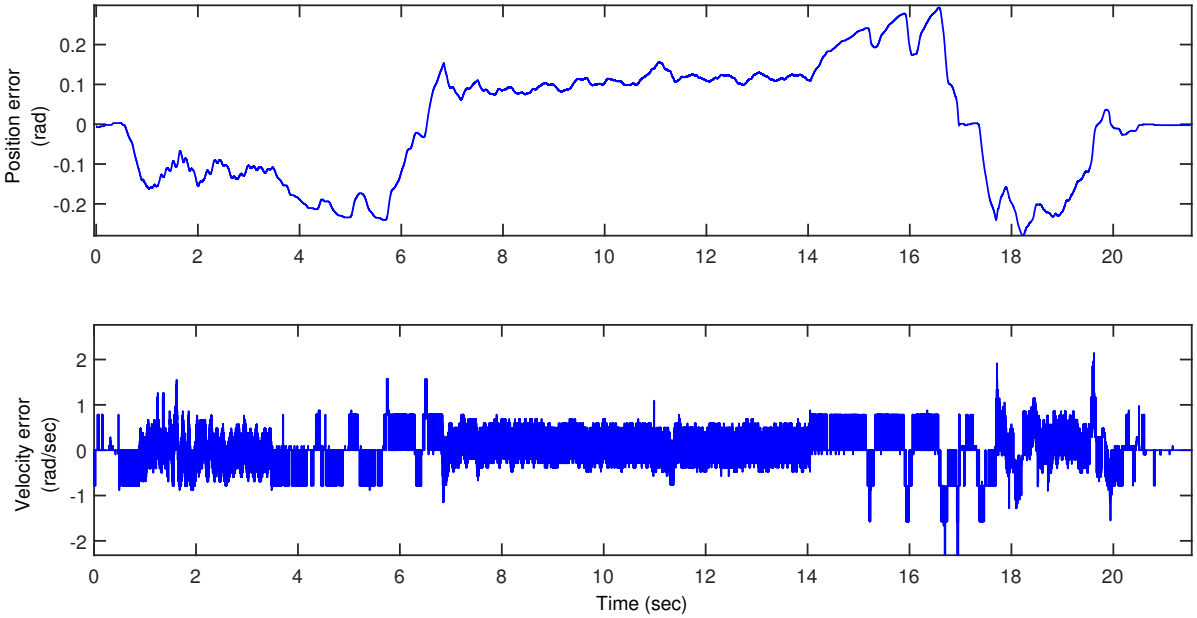


Figure 5.7: Synchronization error for position and velocity with communication delay - constrained motion

5.3 Performance evaluation

In practice, it has been noticed that the motion was smooth when the system was free from communication delays, in comparison to the performance in presence of delays. Then, the system was more damped, although the values of the controlling parameters were the same in both cases (i.e. with and without delays). The system was stable in all cases.

With regards to transparency, the higher the values of K_p the more it was clear to feel the reflected environmental forces. A similar approach of scaling the value of K_p (when the position error exceeds a certain threshold) can be applied here (section 4.2.2).

6

Discussion, Conclusion and Future Work

6.1 Discussion: performance and stability analysis

The main goal of the current study was to determine performance and stability conditions of the designed adaptive controller. In this section, the previously reported results will be interpreted and analyzed.

To start, referring to the coordinating torques in equation (3.7), the terms $K_p e_s(t)$ and $K_p e_m(t)$ are extra terms, in comparison to the terms represented by Chopra et. al. in [21]. If one only considered the master side, breaking down equation (3.7) yields:

$$\begin{aligned}\bar{\tau}_m(t) &= K((\dot{q}_s(t-T) + \Lambda q_s(t-T)) - (\dot{q}_m(t) + \Lambda q_m(t))) + K_p\,(q_s(t-T) - q_m(t)) \\ &= K(\dot{q}_s(t-T) - \dot{q}_m(t)) + K\,\Lambda(q_s(t-T) - q_m(t)) + K_p\,(q_s(t-T) - \, q_m(t)) \\ &= K\,\dot{e}_s(t) + K\,\Lambda\,e_s(t) + K_p\,e_s(t)\end{aligned} \tag{6.1}$$

Therefore, the value of K_p can be compensated by $K\Lambda$. The reason of using this additional K_p term is to facilitate the tuning task. With this controller architecture, tuning can be tedious, since the number of parameters to be tuned increases greatly with increasing degrees of freedom. This also makes it harder to distinguish the effect of each parameter on the performance of the system. In both experiments (the three and single DOF), increasing K_p to increase system transparency and position tracking was very beneficial and straight forward. During the experiments reported in chapter 4, it has been found that larger values if K_p decreases the position drift between the master and the slave.

An attempt has been made, using the single-arm manipulator, to remove K_p and use only $K\Lambda$. However, the performance was unstable, represented in uncontrolled oscillations and lack of motion tracking. Therefore, more tuning needs to be carried out.

Furthermore, due to the architecture of the approach (i.e. the coordinating torques terms), a transparency issue arises. As shown in figure 5.6, there is high velocity

damping on the master side, which affects user's experience of transparency, as the rigid constraint could be felt but with some compliance. It can be noticed in figure 5.6 that only part of the constrained motion plot has this damping effect. This is due to the researcher's hand motion, where the motion was stopped when hitting the wall, resulting in zero velocity during transition motion.

For a position-position controller, velocity damping is a common issue, and with increasing delays the system becomes more damped. This makes it hard to differentiate between free motion and constrained motion. The damping also increases with increasing values of K_p and so does transparency. Ideally, during free motion the user should have zero damping at the slave side. Therefore, a trade-off should be made between small values of K_p and maintaining good transparency.

For the aforementioned issues, some solutions and experiments can be recommended. One recommendation is to keep the slave controller tightly tuned (i.e. with higher coordinating gains) and the master controller loosely tuned.
Further recommended experiments include neglecting the computed master torque, and directly feeding the computed torque from the slave side to the master (i.e. position-force architecture). However, unless the reflected force to the master is significantly attenuated, this architecture could be hard to stabilize, since it is a static approach (i.e. not considering slave dynamics) [6].

6.2 Conclusion

In this thesis work, a passivity-based adaptive controller was implemented into a bilateral teleoperation setting, in order to synchronize the position and velocity of a master and slave manipulators, while maintaining system transparency and stability. The controller was first implemented to a 3-DOF system, and later on a 1-DOF system as a comparison study. In general, it has been found that the system was able to synchronize with tolerable mean tracking errors while being stable with acceptable transparency.

In comparison to other bilateral teleoperation approaches, a drawback of using the controller presented in this thesis is that it is a plant-dependent controller architecture. This means that every time one needs to implement this controller to a new manipulator, a dynamic model of the manipulator needs to be derived in order to build the controller. This can be time-consuming and troublesome especially when using a manipulator with higher degrees of freedom. The process of parameter tuning would also be challenging with more complicated manipulators designs.

6.3 Future Work

This controller can be improved by adding robustness. However, as it may improve the performance, it might induce chattering, which should be mitigated.

Furthermore, performance evaluation on a higher degrees-of-freedom manipulator should be carried out. In addition, the rigid wall on the slave side can be replaced by a compliant surface, in order to evaluate the stability and transparency of the system for different constraints.

Moreover, since there are no mounted force sensors, force estimation techniques can also be implemented, which can improve the transparency of the system.

The presence of delay is the main constraint in the face of bilateral teleoperation through the Internet, which affects performance. One of the proposed solutions is to implement, in the server close to the slave, a mechanism for decision-making (in acute situations), independent of the master side. This can be informed by incorporating the Internet-of-things (IoT) technology and cloud-based platforms to robotic teleoperation [27].

Bibliography

[1] Allison M Okamura. Design and control of haptic systems. Online Course Lecture Notes, 2015.

[2] Katherine Julianne Kuchenbecker. *Characterizing and controlling the high-frequency dynamics of haptic interfaces*. PhD thesis, Stanford University, 2006.

[3] Yiannis Karayiannidis. Modelling and control of mechatronic systems. Lecture Notes, 2016.

[4] Sensodrive master-slave-system. `https://www.sensodrive.de/EN/products/master-slave-system.php`. Accessed: 2017-06-07.

[5] Allison M Okamura. Haptic feedback in robot-assisted minimally invasive surgery. *Current opinion in urology*, 19(1):102, 2009.

[6] Dale A Lawrence. Stability and transparency in bilateral teleoperation. *IEEE transactions on robotics and automation*, 9(5):624–637, 1993.

[7] Surgical Robots - A Brief History. `http://allaboutroboticsurgery.com/surgicalrobots.html`. Accessed: 2016-11-24.

[8] Michael J Massimino and Thomas B Sheridan. Teleoperator performance with varying force and visual feedback. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 36(1):145–157, 1994.

[9] Roberta Klatzky and Susan Lederman. The haptic identification of everyday life objects. In Yvette Hatwell, Arlette Streri, and Edouard Gentaz, editors, *Touching for knowing: cognitive psychology of haptic manual perception*, chapter 7, page 105–121. John Benjamins Publishing Company, Philadelphia, 2003.

[10] MCJ Franken, Stefano Stramigioli, Rob Reilink, Cristian Secchi, and Alessandro Macchelli. Bridging the gap between passivity and transparency. Robotics Science and Systems, 2009.

[11] Antonio Loría and Henk Nijmeijer. Passivity based control. *Encyclopaedia of Life Support Systems (EOLSS), Vol. Perspectives and Overview of Life Support Systems and Sustainable Development, EOLSS Publishers Ltd,a parraıtre*, 2002.

[12] Peter F Hokayem and Mark W Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035–2057, 2006.

[13] Robert J Anderson and Mark W Spong. Bilateral control of teleoperators with time delay. *IEEE Transactions on Automatic control*, 34(5):494–501, 1989.

[14] Günter Niemeyer and J-JE Slotine. Stable adaptive teleoperation. *IEEE Journal of oceanic engineering*, 16(1):152–162, 1991.

[15] Nikhil Chopra, Mark W Spong, Romeo Ortega, and Nikita E Barabanov. On tracking performance in bilateral teleoperation. *Robotics, IEEE Transactions on*, 22(4):861–866, 2006.

[16] Dongjun Lee and Mark W Spong. Passive bilateral teleoperation with constant time delay. *Robotics, IEEE Transactions on*, 22(2):269–281, 2006.

[17] Günter Niemeyer and Jean-Jacques E Slotine. Towards force-reflecting teleoperation over the internet. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 1909–1915. IEEE, 1998.

[18] Nikhil Chopra and Mark W Spong. On synchronization of networked passive systems with time delays and application to bilateral teleoperation. In *proceedings of the SICE annual conference*, pages 3424–3429, 2005.

[19] Nikhil Chopra and Mark W Spong. Passivity-based control of multi-agent systems. In *Advances in robot control*, pages 107–134. Springer, 2006.

[20] Alejandro Rodriguez-Angeles and Henk Nijmeijer. Mutual synchronization of robots via estimated state feedback: a cooperative approach. *Control Systems Technology, IEEE Transactions on*, 12(4):542–554, 2004.

[21] Nikhil Chopra, Mark W Spong, and Rogelio Lozano. Synchronization of bilateral teleoperators with time delay. *Automatica*, 44(8):2142–2148, 2008.

[22] Alan Sherman, Murat Cenk Çavusoglu, and Frank Tendick. Comparison of teleoperator control architectures for palpation task. In *Proceedings of the ASME Dynamic Systems and Control Division, part of the ASME Internatioal Mechanical Engineering Congress and Exposition (IMECE 2000)*, pages 1261–1268, 2000.

[23] Jordi Artigas, Jee-Hwan Ryu, and Carsten Preusche. Time domain passivity control for position-position teleoperation architectures. *Presence*, 19(5):482–497, 2010.

[24] The Geomagic Touch Haptic Device Overview. `http://www.geomagic.com/en/products/phantom-omni/overview/`. Accessed: 2016-08-04.

[25] Fanny Ficuciello, Luigi Villani, and Bruno Siciliano. Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. *IEEE Transactions on Robotics*, 31(4):850–863, 2015.

[26] S. Lichiardopol. A survey on teleoperation. *Dept. Mech. Eng., Dynamics Control Group, Technische Universiteit Eindhoven, Eindhoven, Dept., Mech. Eng., Dyn. Control Group, The Netherlands, Tech. Rep. DCT2007*, 155, 2007.

[27] LES Oliveira and AJ Álvares. Axiomatic design applied to the development of a system for monitoring and teleoperation of a cnc machine through the internet. *Procedia CIRP*, 53:198–205, 2016.

[28] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

A

Derivation of Equation of Motion

As seen in chapter 2, formulating the Lagrange equation is the first step towards deriving the equation of motion.

A.1 Lagrange Formulation

As mentioned in chapter 2, the Lagrangian of the mechanical system is defined as a function of the generalized coordinates, and using the energy equations of the system, as in relation A.2. For the system under study, there are 3-DOF. Hence, the generalized coordinates in the joint space are the three angles, q_1, q_2 and q_3, as in A.1.

$$q = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^{\mathrm{T}} \tag{A.1}$$

$$\mathcal{L}(q, \dot{q}) = \mathcal{T}(q, \dot{q}) - \mathcal{U}(q) \tag{A.2}$$

where $\mathcal{L}$ denotes the Lagrangian, $\mathcal{T}$ denotes the total kinetic energy of the system, and $\mathcal{U}$ is the total potential energy. Next step is to compute system's energy.

A.1.1 Kinetic Energy

For a manipulator with n rigid links, the total kinetic energy is given by sum of contributions from moving joints and moving links, that is:

$$\begin{aligned} \mathcal{T} &= \sum_{i=1}^{n} (\mathcal{T}_{l_i} + \mathcal{T}_{m_i}) \\ &= (\mathcal{T}_{m_1}) + (\mathcal{T}_{l_1} + \mathcal{T}_{m_2}) + (\mathcal{T}_{l_2} + \mathcal{T}_{m_3}) \end{aligned} \tag{A.3}$$

where $\mathcal{T}_{l_i}$ is the kinetic energy of link i, and $\mathcal{T}_{m_i}$ is the kinetic energy of the motor actuating joint i. It should be noted that since link 0 is fixed, its kinetic energy contribution is 0.

Link's kinetic energy is a result of the contribution of both translational and rotational link movements. Therefore, $\mathcal{T}_{l_i}$ should be:

$$\begin{aligned}\mathcal{T}_{l_i} &= \frac{1}{2}\, m_{l_i} \dot{p}_{l_i}^{\mathrm{T}} \dot{p}_{l_i} + \frac{1}{2}\, \omega_i^{\mathrm{T}} R_i\, I_{l_i}^i\, R_i^{\mathrm{T}}\, \omega_i \\ &= \frac{1}{2}\, m_{l_i} \dot{q}^{\mathrm{T}}\, J_P^{(li)\mathrm{T}}\, J_P^{(li)}\, \dot{q} + \frac{1}{2}\, \dot{q}^{\mathrm{T}}\, J_O^{(li)\mathrm{T}}\, R_i\, I_{l_i}^i\, R_i^{\mathrm{T}}\, J_O^{(li)}\, \dot{q}\end{aligned} \tag{A.4}$$

where m_{l_i} is the link mass, p_{l_i} is its translational velocity, ω_i is its angular velocity, R_i is the rotation matrix from link i frame to the base frame and $I_{l_i}^i$ is the inertia tensor referred to link frame. The Geometric Jacobian $[J_P \quad J_O]^{\mathrm{T}}$ are related to the end-effector linear and angular velocities ($\dot{p}_e$ and ω_e) by:

$$\begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} J_P \\ J_O \end{bmatrix} \dot{q} \tag{A.5}$$

In a similar fashion:

$$\begin{aligned}\mathcal{T}_{m_i} &= \frac{1}{2}\, m_{m_i} \dot{p}_{m_i}^{\mathrm{T}} \dot{p}_{m_i} + \frac{1}{2}\, \omega_{mi}^{\mathrm{T}}\, I_{m_i}\, \omega_{mi} \\ &= \frac{1}{2}\, m_{m_i} \dot{q}^{\mathrm{T}}\, J_P^{(mi)\mathrm{T}}\, J_P^{(mi)}\, \dot{q} + \frac{1}{2}\, \dot{q}^{\mathrm{T}}\, J_O^{(mi)\mathrm{T}}\, R_{mi}\, I_{m_i}^{mi}\, R_{mi}^{\mathrm{T}}\, J_O^{(mi)}\, \dot{q}\end{aligned} \tag{A.6}$$

where m_{m_i} is the rotor mass (stator mass assumed to be included in the link), and p_{m_i} and ω_{mi} are the translational and angular velocities at rotor's center of mass, respectively. While $I_{m_i}^i$ is the inertia tensor of the rotor relative to its center of mass.

It should be noted that for joint 1, the contribution is only from rotational motion, therefore the linear motion part for joint 1 is zero. Furthermore, rotors at joints and links are assumed to have a symmetric mass distribution about their rotation axis, and hence they have diagonal inertia tensors.
The reader is referred to Chapter 7 in [28] for more details about computing Jacobains and rotational matrices.

A.1.2 Potential Energy

Manipulator's potential energy is given by the contributions from links and rotors. This is given by A.7:

$$\begin{aligned}\mathcal{U} &= \sum_{i=1}^{n} (\mathcal{U}_{l_i} + \mathcal{U}_{m_i}) \\ &= -\sum_{i=1}^{n} (m_{li} g_0^{\mathrm{T}}\, p_{li}) + (m_{mi} g_0^{\mathrm{T}}\, p_{mi})\end{aligned} \tag{A.7}$$

where the gravity acclerelation vector in the base fram is $g_0 = [0 \quad 0 \quad -g]^{\mathrm{T}}$. This is under the assumption of rigid links, therefore there are no effects due to elastic forces, only gravitational forces. Joint 1 is assumed to be the reference point (i.e. height for joint 1: $p_{m1} = 0$).

A.2 Resulting equations

Utilizing the previously mentioned equations, and figures 2.2 and 2.3, the equation of motion has been derived.

Some of the specifications of the manipulator's structure were given by the manufacturer, which are shown in table A.1 below.

Table A.1: Manipulator's structure specifications[3]

Structure	**Specifications**
L_1 (Length of link 1)	0.132 m
L_2 (Length of link 2)	0.132 m
r_1 (Radius of joint 1)	0.03 m

Using system identification, the values of the vector π (given that $F_v \neq 0$) have been identified as:

$$\pi = \begin{bmatrix} I_{l1_x} \\ I_{l1_y} \\ l_1^2 m_2 \\ I_{l2_x} \\ I_{l2_y} \\ I_{m1_y} \\ l_1 m_3 \\ I_{l2_z} \\ I_{l1_z} \\ l_2 m_3 \\ m_3 \\ l_2^2 m_3 \\ F_{v1} \\ F_{v2} \\ F_{v3} \end{bmatrix} = \begin{bmatrix} 0.0022 \\ 0.0023 \\ 0.0005 \\ 0.0010 \\ 0.0009 \\ 0.0031 \\ 0.0056 \\ 0.0011 \\ 0.0022 \\ 0.0096 \\ 0.0800 \\ 0.0012 \\ 0.0089 \\ 0.0170 \\ 0.0058 \end{bmatrix} \tag{A.8}$$

where I_{li_j} is the inertia of link i in direction j, l_i is the distance of center of mass of link i from its tip, I_{m1_j} is the inertia of the sphere (i.e. joint 1) in direction j, m_i is the mass of link i and F_{vi} is the viscous friction of joint i, given that $i = 1, 2, 3$ and $j = x, y, z$.

Thus, equation 2.5 will have the form:

$$\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\end{bmatrix} = \begin{bmatrix} & \dot{q}_1 & 0 & 0\\ Y_(q,\dot{q},\ddot{q}) & 0 & \dot{q}_2 & 0\\ & 0 & 0 & \dot{q}_3\end{bmatrix} \begin{bmatrix}\pi_1\\ \pi_2\\ \vdots\\ \pi_{15}\\ F_{v1}\\ F_{v2}\\ F_{v3}\end{bmatrix} \tag{A.9}$$

Eventually, the final form of the dynamic equation of the system under study is as follows:

$$B(q)\ddot{q} + n(q,\dot{q}) + g(q) + F_v\,\dot{q} = \tau \tag{A.10}$$

$$\begin{aligned}
B(1,1) &= \frac{1}{2}(I_{l1_x} + I_{l1_y} + I_{l2_x} + I_{l2_y} + 2I_{m1_y} + l_1^2 m_2 + (L_1^2 + l_2^2)m_3 + \\
&\quad (-I_{l1_x} + I_{l1_y} + l_1^2 m_2 + L_1^2 m_3)\cos(2q_2) + (I_{l2_x} - I_{l2_y} - l_2^2 m_3)\cos(2(q_2+q_3)) + \\
&\quad 4\,L_1\,l_2\,m_3\,\cos(q_2)\sin(q_2+q_3))\\
B(1,2) &= B(2,1) = B(1,3) = B(3,1) = 0\\
B(2,2) &= I_{l1_z} + I_{l2_z} + l_1^2 m_2 + L_1^2 m_3 + l_2^2 m_3 + 2L_1\,l_2\,m_3\,\sin(q_3)\\
B(2,3) &= B(3,2) = I_{l2_z} + l_2^2 m_3 + L_1\,l_2\,m_3\,\sin(q_3)\\
B(3,3) &= I_{l2_z} + l_2^2 m_3
\end{aligned} \tag{A.11}$$

$$\begin{aligned}
n(1,1) &= \dot{q}_1(2\dot{q}_3\,\cos(q_2+q_3)\,(L_1\,l_2\,m_3\,\cos(q_2) + (-I_{l2_x} + I_{l2_y} + l_2^2\,m_3)\sin(q_2+q_3)) + \\
&\quad -\dot{q}_2\,(-2\,L_1\,l_2\,m_3\,\cos(2q_2+q_3) + (-I_{l1_x} + I_{l1_y} + l_1^2 m_2 + L_1^2 m_3)\,\sin(2q_2) + \\
&\quad (I_{l2_x} - I_{l2_y} - l_2^2 m_3)\,\sin(2(q_2+q_3))))\\
n(2,2) &= \frac{1}{2}(2\,L_1\,l_2\,m_3\,\dot{q}_3\,(2\dot{q}_2 + \dot{q}_3)\,\cos(q_3) + \dot{q}_1^2\,(-2\,L_1\,l_2\,m_3\,\cos(2q_2+q_3) + \\
&\quad (-I_{l1_x} + I_{l1_y} + l_1^2 m_2 + L_1^2 m_3)\,\sin(2q_2) + (I_{l2_x} - I_{l2_y} - l_2^2 m_3)\,\sin(2(q_2+q_3))))\\
n(3,3) &= -\,L_1\,l_2\,m_3\,\dot{q}_2^2\,\cos(q_3) - \dot{q}_1^2\,\cos(q_2+q_3)(L_1 l_2 m_3\cos(q_2) + (-I_{l2_x} + I_{l2_y} + l_2^2 m_3) \\
&\quad \sin(q_2+q_3)))
\end{aligned} \tag{A.12}$$

$$\begin{aligned}
g(1,1) &= 0\\
g(2,2) &= g\,(-l_1 m_2\,\cos(q_2) - L_1\,m_3\,\cos(q_2) - l_2\,m_3\,\sin(q_2+q_3))\\
g(3,3) &= -g\,l_2\,m_3\,\sin(q_2+q_3)
\end{aligned} \tag{A.13}$$

$$F_v = \begin{bmatrix} F_{v1} & 0 & 0\\ 0 & F_{v2} & 0\\ 0 & 0 & F_{v3}\end{bmatrix} \tag{A.14}$$

A.3 Direct and Inverse Kinematics

A.3.1 Direct Kinematics

Used for converting from Configuration space to Task space, i.e.:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \longrightarrow x_e = \begin{bmatrix} p_e \\ \phi_e \end{bmatrix}$$

Where p_e is the end-effector position vector, computed of the origin of the end-effector frame with respect to the base frame, and ϕ_e is the orientation of the end-effector, represented by unit vectors of the end-effector frame.

Utilizing figure 2.2, the DH-parameters were extracted, as follows:

Table A.2: DH-parameters

Link	α_i	θ_i	a_i	d_i
1	$-\frac{\pi}{2}$	q_1	0	0
2	0	q_2	L_1	0
3	0	$q_3 - \frac{\pi}{2}$	L_2	0

Subsequently, the Homogeneous Transformation matrix ($T_e^b(q)$) is computed out of parameters in table A.2, and then used to form the Task space matrix.

$$\begin{aligned} T_e^b(q) &= \begin{bmatrix} n_e^b(q) & s_e^b(q) & a_e^b(q) & p_e^b(q) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_1c_3s_2 + c_1c_2s_3 & c_1c_2c_3 - c_1s_2s_3 & -s_1 & L_1c_1c_2 + L_2c_1c_3s_2 + L_2c_1c_2s_3 \\ c_3s_1s_2 + c_2s_1s_3 & c_2c_3s_1 - s_1s_2s_3 & c_1 & L_1c_2s_1 + L_2c_3s_1s_2 + L_2c_2s_1s_3 \\ c_2c_3 - s_2s_3 & -c_3s_2 - c_2s_3 & 0 & L_2c_2c_3 - L_1s_2 - L_2s_2s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

where $R_e^b = \begin{bmatrix} n_e^b(q) & s_e^b(q) & a_e^b(q) \end{bmatrix}$, $p_e = p_e^b(q)$, and c and s referes to the trigonometric functions of 'cos' and 'sin', respectively, where the subscript corresponds to the joint's angle. Note that the unit vectors: a_e is in the *approach* direction, s_e is the *sliding* and normal to a_e, and n_e is *normal* to the other two.

A.3.2 Inverse Kinematics

Used to convert from Task space to Configuration space:

$$x_e = \begin{bmatrix} p_e \\ \phi_e \end{bmatrix} \longrightarrow q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Utilizing figure 2.3, Inverse Kinematics were computed as follows:

$$\begin{aligned} q_1 &= \arctan \frac{Y}{X} \\ q_2 &= \arctan \frac{-Z}{\sqrt{X^2+Y^2}} - \arccos \frac{K^2 + L_1^2 - L_2^2}{2L_1K} \\ q_3 &= -\arccos \frac{-K^2 + L_1^2 + L_2^2}{2L_1L_2} + (\frac{3\pi}{2}) \\ \text{where} : k &= \sqrt{Z^2 + X^2 + Y^2} \end{aligned} \tag{A.15}$$

The reader is directed to go through Chapter 2 in [28] to know on how to formulate the Homogeneous Transformation matrix, and for more information on Direct and Inverse kinematics.

B

Establishing a teleoperation connection between two manipulators

This appendix is intended to be a detailed document on how to work with the 3D Systems Geomagic® Touch™ manipulator as a teleoperator haptic device (i.e. master and slave setting) in a MATLAB/SIMULINK R2014A environment.

B.1 Stream server/client: establishing a connection

First, in order to establish a connection between two manipulators, one have to turn off the Windows firewall in both of the PCs intended to be used.

QUARC stream blocks can be used to send/receive data through a Transmission Control Protocol/Internet Protocol (TCPIP) or User Datagram Protocol (UDP) connection. However, for this to happen, one need to configure the PCs (both PCs need to have QUARC) so they can communicate between each other. Some examples can be found, which uses stream block in the QUARC demo library. To access these demos, one can type qc_show_demos in MATLAB command window. These demos are located in the communication section.

In order to get to the documentation on how to use/interpret the **Stream Server** or **Stream Client** blocks and signals, the necessary documentation can be found by typing qc_show_demos in MATLAB command window, choosing QUARC Basic Communications Demo under Communications block will, then Basic Communications. One can choose either Stream Client or Stream Server to have access to the relevant block documentation.

To establish a connection in the server model, the Universal Resource Identifier (URI) parameter should be configured with the Client PC IP address. Whereas in the client model, the URI parameter should be configured with the Server PC IP address. Either UDP or TCPIP can be used.

In order to send multidimensional data, zeros (elements) should be added in the

Default output value parameter, in both the client and server stream blocks. For example, for sending/receiving 3 elements, one should enter $\begin{bmatrix}0 & 0 & 0\end{bmatrix}$, to get a 1×3 vector.

B.2 Implementation notes

For implementation, there are some notes to be aware of while using SIMULINK:

- It is possible to use more that one Stream Client block (i.e. multiple clients). However, only single server can be used for transmission. Therefore, in order to send both position and velocity, information form both can be combined as single input for Stream Server or Stream Client block, i.e. using only 1 server and 1 client.

- To combine server/client inputs, a multiplexer can be used. It should be noted that the multiplexer generally combines the signals in column vector. Thus, for the case within hand, each sampling time, the position is a (3×1) vector, and so is the velocity vector. The multiplexer will combine these as a (6×1) vector. In this case, the *Default output value* parameter will be set to $\begin{bmatrix}0; & 0; & 0; & 0; & 0; & 0;\end{bmatrix}$.

- To split the received position and velocity information at the output of the server/client, a submatrix block can be used.