NONLINEAR ADAPTIVE ROBUST CONTROL FOR BILATERAL TELEOPERATED ROBOTIC MANIPULATORS WITH ARBITRARY TIME DELAYS

by

Moyang Zou

Submitted in partial fulfillment of the
requirements for the degree of
Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
April 2017

Table of Contents

List of Figures

Abstract

Bilateral teleoperation systems have been extensively developed over decades. The communication delay can cause instability and is one of the most challenging control design problems. Additionally, nonlinearity, parameter variation, and uncertainty in the environment dynamics and robot model are also challenging issues that need to be considered in order to achieve excellent control performance.

This thesis aims to develop a globally stable nonlinear adaptive robust control structure, dealing with the following problems. Firstly, this new control structure can tolerate arbitrary, long, and time-varying delays. Secondly, in order to ensure excellent tracking performance on both sides, a nonlinear adaptive robust control algorithm is proposed. Lyapunov method is used with stability proof. Thirdly, an environmental torque estimator is designed to estimate unmeasurable torques by a least square adaptive law. Moreover, a novel structure of communication block is developed. From the master side to the slave side, the position signal of the master manipulator is being transmitted. However, from the slave side backwards, only the estimated parameters of the environmental torque are sent back. This structure is designed to enhance the control performance of the adaptive robust controller. To ensure the desired transparency performance, an impedance control structure is developed on the master side.

Simulations are carried out to verify the robust stability, excellent transparency and synchronization of the proposed design under arbitrary time-varying delays. Simulation studies on the control gain tuning and model mismatches are carried out in order to verify the effectiveness of the design under different circumstances. Two different control algorithms are also presented to compare with the proposed method. In the last chapter, conclusions and the possible future work are presented.

Acknowledgements

I would like to thank my supervisor Dr. Ya-Jun Pan of the Mechanical Engineering Department for her mentorship and support. She has spent plenty of hours in developing this work with me. I would also like to thank Dr. Mae Seto and Dr. Jason Gu for their role on my supervisory committee, Peter Jones for his technical assistance, and the Advanced Controls and Mechatronics research group for their suggestions and help over the years (Shane Forbrigger, Usman Ahmad and Ajinkya). Last, but not least, many thanks to my family for all their support and love to me.

Chapter 1

Introduction

In this chapter, the background information of this research work is outlined, including the introduction of teleoperation and bilateral teleoperation, the literature review for existing control methods that have been applied to bilateral teleoperation system.

1.1 Teleoperation and Bilateral Teleoperation

Teleoperation represents operation of a machine at a distance [1] or remotely. The meaning of "operation" can be almost anything. The "distance" may refer to a physical distance between the master hardware and the slave hardware like "remote control", but it can also represent a change in scale.

Fig.1.1 shows the block diagram of a typical teleoperation system. As it can be seen in the figure, typical teleoperation system consists of five basic subsystems: the human operator, the master hardware, the communication block, the slave hardware and the environment. The master hardware is the device with which the human operator interacts and the slave hardware is the device which operates on the environment. The operator commands a velocity forward. Then this command passes through the master, the communication block and the slave subsystems to the environment. Likewise, the forces sensed at the environment are transmitted back through those subsystems, to the human operator [2].

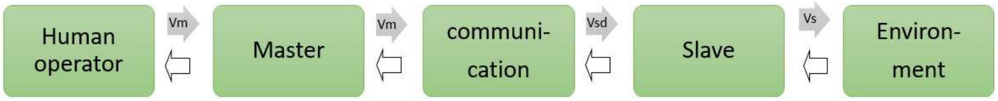

Figure 1.1: Block diagram of typical teleoperation system

Bilateral teleoperation systems have been widely developed over decades. They are designed to let the interaction occurred between the remote slave robot and the remote environment. A kind of haptic feedback that can be transferred back to the human operator through the slave robot [3].

1.2 Applications

Over the past decades, bilateral teleoperation has been applied in various areas such as telesurgery and space teleoperation.

Telesurgery

Teleoperation has been applied to telesurgery area for more than twenty years. It allows the surgeon to perform surgeries in any location of the world without requiring them to travel. There are two systems for telesurgery have been commercially available in the market, Zeus robotic surgical system and Da Vinci surgical system(Fig.1.2). Zeus surgical system was produced by the American robotics company Computer Motion. It was designed for minimally invasive microsurgery procedures, like heart surgery and endoscopic coronary artery bypass grafting (E-CABGTM)[1][4]. Da Vinci system, produced by the American company Intuitive Surgical, was commonly used for prostatectomies, and increasingly for cardiac valve repair and gynecologic surgical procedures[5][6]. As it can be seen in Fig.1.2, The surgeon sits at a console away from the operating table and controls the surgical instruments with sensitive thumb and finger grips. The system is calibrated to the surgeon so that his or her hand movements are robotically scaled down and translated into minute adjustments of the instruments working inside the patient.

Space teleoperation

Considering the risks and costs for the outer-space works for human, more and more space teleoperated robots are being used for the international space station works and the repair works on moon or mars [8].

Applications in space teleoperation area are being classified into two different fields based on different tasks.

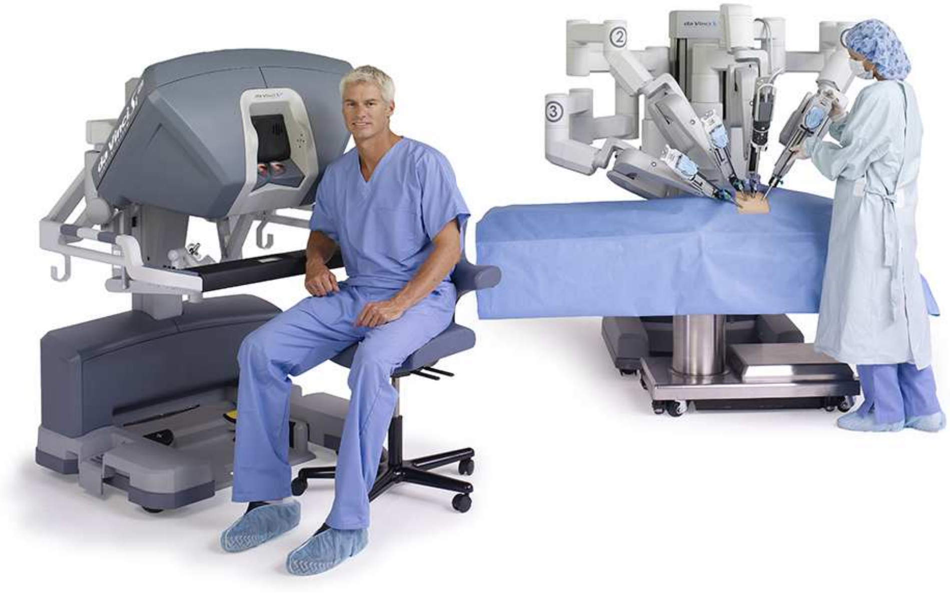

Figure 1.2: Da Vinci surgical system [7]

a) Robotic exploration rover

Exploration robots are being used in plenty of exploration projects in the past century. Mars rovers and lunar rovers are important members in exploration robots family. The curiosity is a Mars rover exploring Gale Crater on Mars as part of NASA's Mars Science Laboratory mission (MSL)[9]. The Yutu rover, launched at 17:30 UTC on 1 December 2013, is an lunar rover that was part of the Chinese Chang'e 3 mission to the Moon[10].

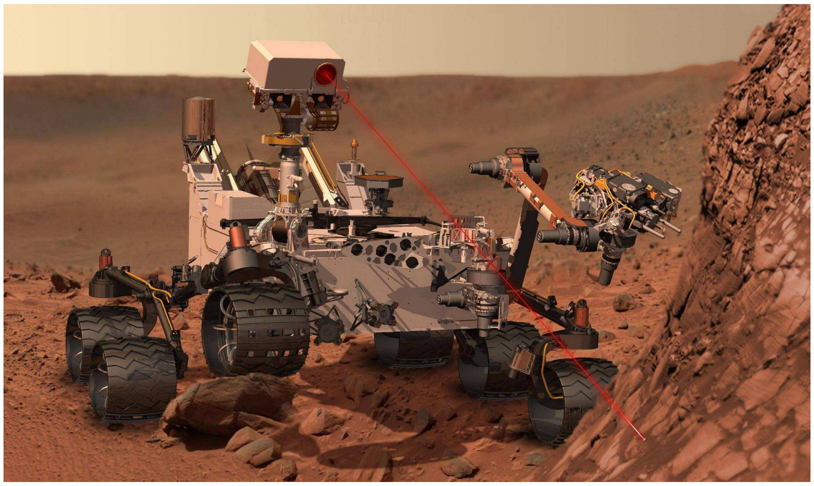

Figure 1.3: The Curiosity Mars rover [11]

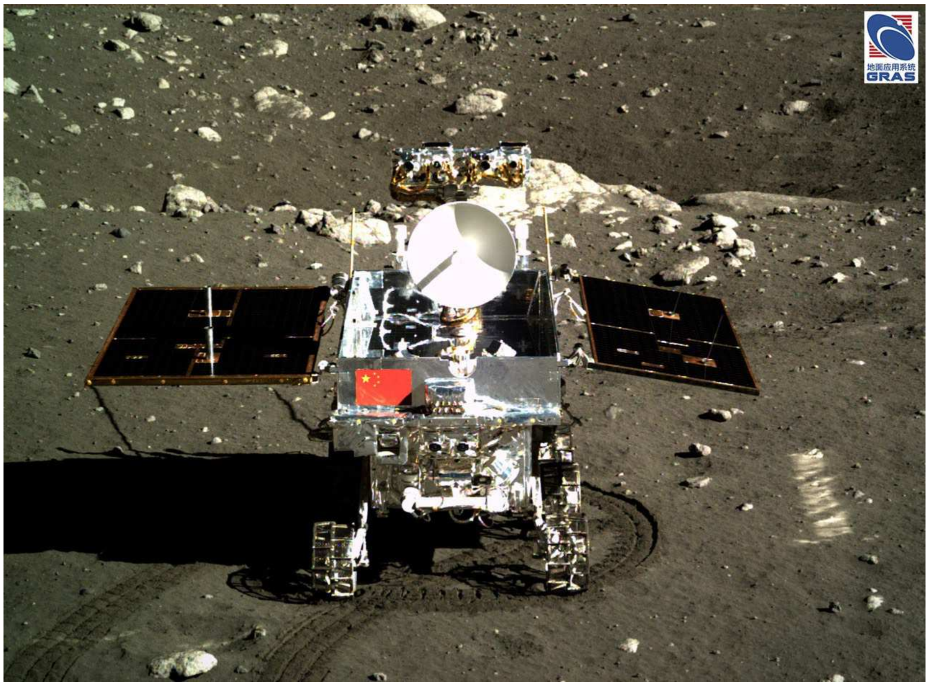

Figure 1.4: The Yutu Lunar rover [12]

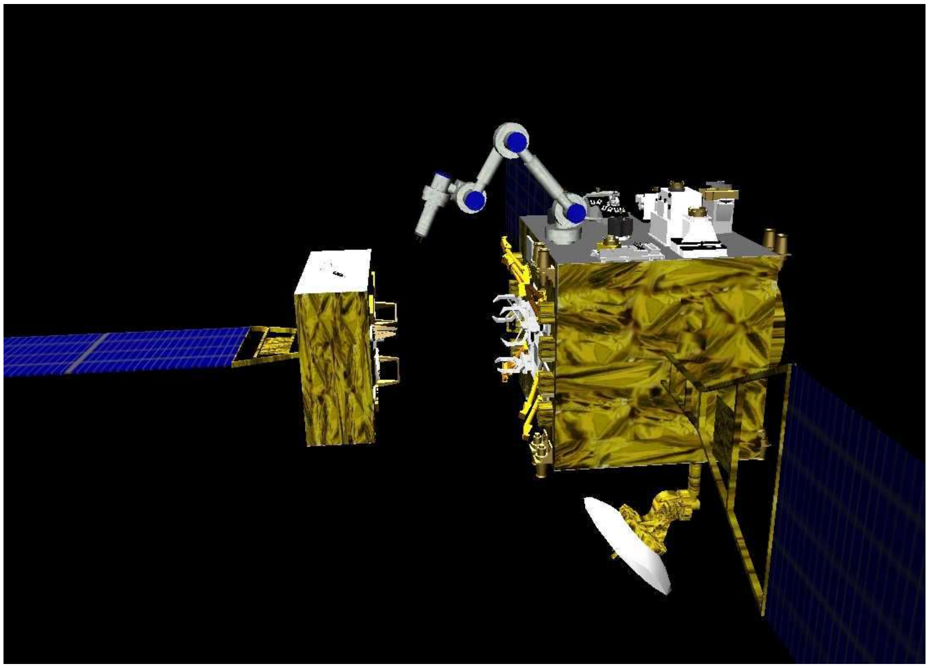

Figure 1.5: The target capture by the manipulator arm by integrating technologies verified by ETS-VII [13]

b) Outer-space robot manipulators

Teleoperated robot manipulators have been applied in the international space station to perform dangerous tasks. There are few examples that have been used in outer-space operations. The Engineering Test Satellite VII(ETS-VII) Japanese Flying Space Robot is one example that is being experimentally used in outer-space [13].

1.3 Literature Review

Control signals experience a time delay when transmitted over a communications network. This communication delay could destabilize the whole system if it is not dealt with properly. Additionally, nonlinearity, parameter variation, and uncertainty in the environment dynamics and robot model are also challenging issues that need to be considered in order to achieve excellent control performance.

Many different kinds of control methods have been developed to solve these issues. In order to choose appropriate controller for the bilateral teleoperation system, it is necessary to review the advantages and disadvantages of the available control strategies.

1.3.1 PID Control

The proportional–integral–derivative(PID) control method is a commonly used control method that has been applied for bilateral teleoperation. The tracking performance is based on the tracking error which is the difference between the desired trajectory and the actual trajectory.

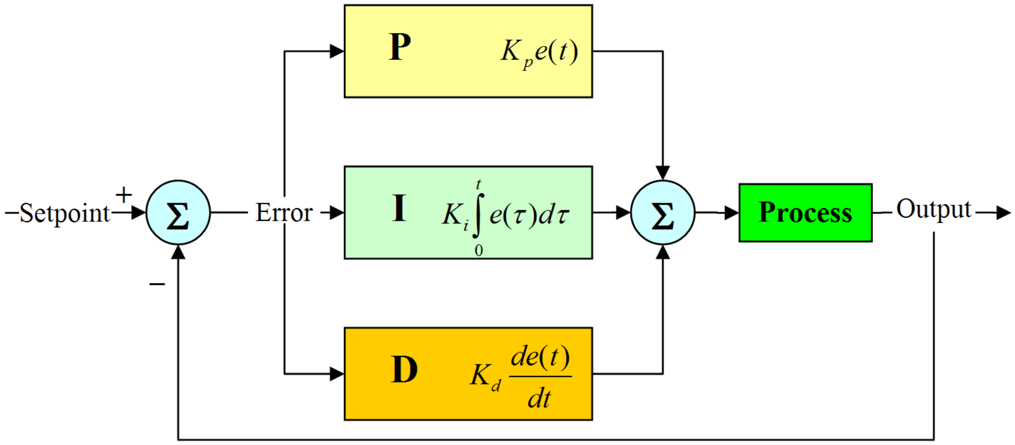

Figure 1.6: The block diagram of a PID controlle [14]

The following equation is a typical PID controller transfer function [15][19].

$$K(s) = K_p + \frac{K_i}{s} + K_d s$$
$$K(s) = K_p(1 + \frac{1}{T_i s} + T_d s)$$

where K_p is the proportional gain and a coefficient of tuning, K_i is the integral gain and a coefficient of tuning, K_d is the derivative gain and a coefficient of tuning.

The tuning strategies of PID controller are discussed in [15]. Each control gain has a different effect on the performance of the controller. A large proportional gain can lead to a more sensitive response and the controller will be more sensitive. On the contrary, a small control gain can result in a less responsive controller. K_i is the integral gain in the controller. The tuning of K_i can accelerates the process of reaching a steady-state and cancel out the steady-state error of purely proportional control. The derivative gain K_d is used to smooth the output signal and counterbalance the overshoots of the output.

In various works [16][17][18], the PID approaches are typically designed for bilateral system with no delay or constant delays. When it comes to the large time-varying delay circumstances, the strategy that presented in the thesis is obviously more effective.

1.3.2 Predictive Control

In a system with time delay, the value of control gain must be reduced in order to stabilize the system [21]. The Smith predictor control algorithm can be added to the system to deal with this situation [20]. However, the coefficients in the model need to match with the coefficients in the plant precisely. A Smith Predictor can be applied to a system when the time delay is known and the time delay should be constant case.

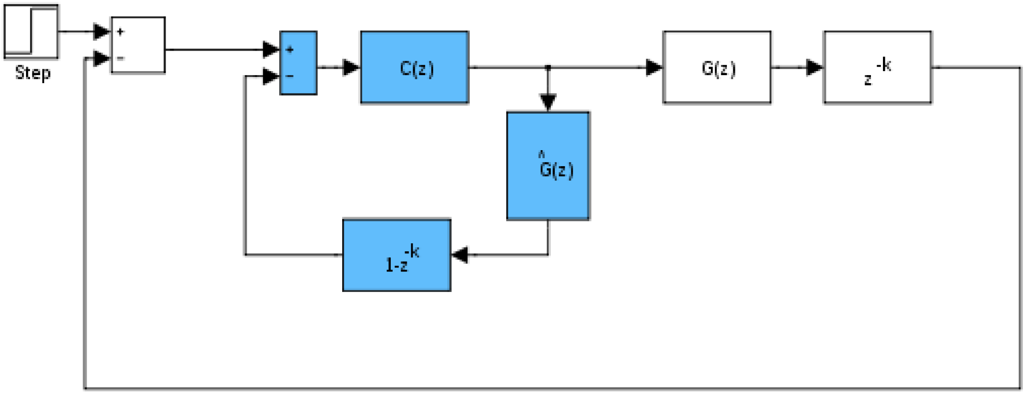

Figure 1.7: The smith predictor [22]

Model predictive control(MPC) has been applied to teleoperation system in different ways. Prediction could be applied to the system at either slave side [23] or master side [24] [25]. In [26], in order to avoid using the delayed transmitted information, four predictors have been applied to both side of the system respectively(each side have 2 predictors).

Predictive structure can be applied to the system to reduce the influence of time delay. However, the dynamic model of the master or the slave all need to be well-known in those schemes in the references. Comparing to the controller that is derived in this work, the dynamic model of the master and the slave hardware don't have to be well-known to make the controller works effectively.

1.3.3 Passivity Based Method

Passivity based control methods are the most widely used methods that can deal with time-delay problem. Consider a system with input u and output y, where y is depended on u. The system is passive if there exists a constant $\beta \geq 0$ that

$$\int_0^{\tau} y^T(t)u(t)dt \geq -\beta, \tag{1.1}$$

for any inputs u and time $\boldsymbol{\tau}$ [27].

Passivity based control method is a method that have two power variables transmitted from the master side to the slave side. Typically, for the mechanical system, the variables are velocity and force. Every element in the system needs to be passive in order to ensure the entire system to be passive. Wave variable is one of the passivity theory based control method that has been commonly used. Instead of sending power variables to the slave side, a transformation is made and the new variables are transmitted over the communication channel. Usually a pair of force and velocity is chosen as new variables but other variables can also be used. The whole concept of wave variable approach comes under the umbrella of passivity and wave variable transformation ensures that the passivity is preserved under this scheme. Power flow can be redefined as,

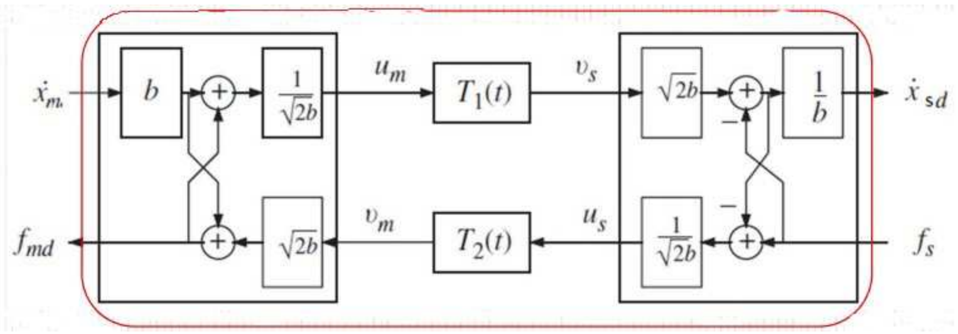

Figure 1.8: Wave variable control [30]

$$P = \dot{x}^T F = \frac{1}{2}u^T u + \frac{1}{2}v^T v, \tag{1.2}$$

where, $\frac{1}{2}u^T u$ is the power flowing along the main direction considering a positive sign, and $\frac{1}{2}v^T v$ is the power flowing against the main direction considering a negative sign. In this approach, the value of u, v can be derived as follows,

$$u = \frac{b\dot{x} + F}{\sqrt{2b}} \qquad v = \frac{b\dot{x} - F}{\sqrt{2b}}, \tag{1.3}$$

where b is a characteristic impedance associated with the wave variables and it can be selected arbitrarily [31]. This transformation is one to one and always remains unique. All the information is preserved under this transformation.

Passivity based approaches mostly focus on ensuring the robust stability. However, it cannot ensure control performance simultaneously [27] [31].

1.3.4 Adaptive Control

Adaptive control is a method that modifies the parameters to adapt to the environment. It can obtain a certain performance through the adaptation of parameters in the controller. One of the important applications is to adapt the performance of the slave side to match the performance of the master side [32]. Besides, adaptive control method is applied to situations to account for variations in parameters in other control schemes [33].

In [34], a Gain Scheduled Middleware is developed to adapt current ordinary controllers into networked controllers. The system is presented in Fig.1.9.

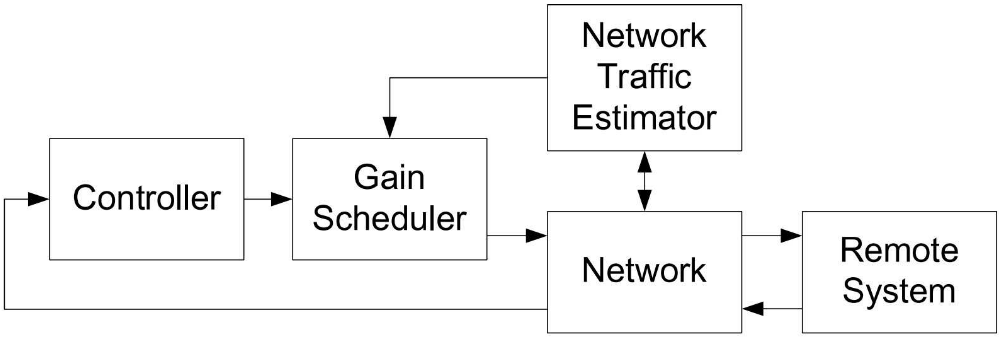

Figure 1.9: Gain Scheduled Middleware [34]

Adaptive methods all require a model to adapt to. In the case of teleoperation discussed here, these models are difficult to characterize under the same form.

1.4 Thesis Motivation

The existing control methods that were reviewed in the literature review section all have limitations. The PID approach is typically designed for bilateral system with no delay or constant delays [16-18]. Passivity based approaches mostly focus on ensuring the robust stability, which cannot simultaneously ensure control performance. For model predictive control, although Predictive structure can be applied to the system to reduce the influence of time delay, the dynamic model of the master or the slave all need to be well-known in order to make the entire system work properly.

In real-world applications, the force sensors may not be available since they can increase the cost and the weight of the manipulator. Thus, the environmental torque needs to be estimated when the real torque is not measurable.

In this work, a new adaptive robust control scheme will be presented. Firstly, this control structure can tolerate arbitrary long time-varying delays. Secondly, a new nonlinear adaptive robust control algorithm is proposed to deal with nonlinearities, unknown parameters, modelling errors and uncertainties in the system. Thirdly, an environmental torque estimator is designed to estimate unmeasurable torques which commonly occur in real applications. The unknown parameters of the environmental torques are estimated online by a least square adaptive law. Furthermore, a novel structure of communication block is developed. The trajectory of the master manipulator is transmitted to the slave side. However, from the slave side to the master side, the estimated environmental torque parameters are transmitted back. This structure is developed to improve the control performance of the adaptive robust controller. To ensure the desired transparency performance, an impedance control structure is developed on the master side. In conclusion, the nonlinear adaptive robust control in this work can achieve stability, the excellent transparency and synchronization performance under arbitrary time-varying delays. Simulation results on a pair of two degree of freedom robotic manipulators are considered to verify the effectiveness of the control design.

Chapter 2

System Modelling

In this chapter, each subsystem of the whole teleoperation model will be discussed separately in each section. Those subsystems are: the human operator subsystem, the master manipulator subsystem, the slave manipulator subsystem, the signal optimal processing subsystem and the environmental model.

As in Fig.2.1, a pair of two degree of freedom manipulators have been chosen as models for this teleoperation system. The sign convention that has been adopted to the model is also shown in the figure. The dynamic of the master and slave models will be demonstrated in detail in next section.

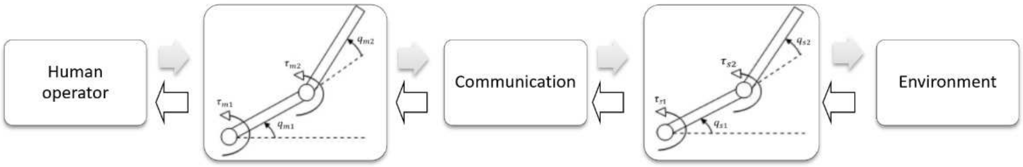

Figure 2.1: Diagram of torques and their directions for 2-DOF manipulators

2.1 Lagrange Formulation

The following formulation is a typical form of the equation of motion of a n-DOF robot mechanism [35].

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_g(\mathbf{q}) = \boldsymbol{\tau}. \tag{2.1}$$

where H is a function of $\mathbf{q}$, C is a function of $\mathbf{q}$ and $\dot{\boldsymbol{q}}$, $\boldsymbol{\tau}_g$ is a function of $\mathbf{q}$. $\mathbf{q}$ is the rotation angle and $\dot{\mathbf{q}}$ is the angular velocity of each joint.

Lagrange formulation is one of the most commonly used method for deriving the terms in (2.1).

The Lagrange formulation proceeds through the Lagrangian of the robot mechanism.

$$L = T - U, \tag{2.2}$$

where T is the total kinetic and U is the potential energy of the mechanism. T can be derived by,

$$T = \frac{1}{2}\dot{\mathbf{q}}^T H\dot{\mathbf{q}} \tag{2.3}$$

Then, the dynamic equation can be derived using Lagrange's equation as following,

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial \dot{q}_i} = \boldsymbol{\tau}_i. \tag{2.4}$$

The following is the result in scalar form [35]:

$$\sum_{j=1}^{n} H_{ij}\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} C_{ijk}\dot{q}_j\dot{q}_k + \boldsymbol{\tau}_{gi} = \boldsymbol{\tau}_i, \tag{2.5}$$

which shows the structure of the velocity-product terms, where C_{ijk} are Christoffel symbols of the first type and can be expressed in the following form,

$$C_{ijk} = \frac{1}{2}(\frac{\partial H_{ij}}{\partial q_k} + \frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i}) \tag{2.6}$$

The elements of C in (2.1) can be developed using,

$$C_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k. \tag{2.7}$$

With the derivation of C in (2.7), it is possible to show that the matrix N, given by,

$$N(q,\dot{q}) = \dot{H}(q) - 2C(q,\dot{q}), \tag{2.8}$$

is skew-symmetric. Therefore, for any $n \times 1$ vector α,

$$\alpha^T N(q,\dot{q})\alpha = 0. \tag{2.9}$$

This principle is pretty useful in control.

2.2 Dynamics

The dynamics of a robot manipulator relate the forces and torques acting on the robot to its motion.

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) = \boldsymbol{\tau}. \tag{2.10}$$

Where $q \in R^{n\times 1}$, $\dot{q} \in R^{n\times 1}$, are the angular position and velocity seperately. M, C and G can be derived from a combination of geometry and the Lagrange formulation. M denotes an $n \times n$ matrix representing the inertia torque that experienced by each joint. It can be derived geometrically. G represents an $n \times 1$ vector representing the external torques on each joint. In this work, it represents the gravity torques of each joint. It can also be derived from the geometry of the manipulator. Once M was derived, the elements of C matrix can also be derived accordingly using the Lagrange formulation: [36]

$$C_{ij} = \sum_{k=1}^{N} \frac{1}{2}\left(\frac{\partial H_{ij}}{\partial q_k} + \frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i}\right)\dot{q}_k.$$

Wrench (vector of torque and force) acting on the end-effector can be derived in the joint space using the geometric Jacobian.

$$\boldsymbol{\tau} = J^{-1} f.$$

The Jacobian is a multidimensional form of the derivative. For example, if we have six functions, each of which is a function of six independent variables [36]as

$$\begin{aligned}
y_1 &= f_1(x_1, x_2, x_3, x_4, x_5, x_6),\\
y_2 &= f_2(x_1, x_2, x_3, x_4, x_5, x_6),\\
&\vdots\\
y_6 &= f_6(x_1, x_2, x_3, x_4, x_5, x_6).
\end{aligned} \tag{2.11}$$

We can also write the above equations in vector notation:

$$Y = F(X). \tag{2.12}$$

where $X = x_1, x_2 \cdots x_6$; $Y = y_1, y_2 \cdots y_6$. Now, we can use the chain rule to calculate the differentials of y_i, and we get,

$$\begin{aligned}
\delta y_1 &= \frac{\partial f_1}{\partial x_1}\delta x_1 + \frac{\partial f_1}{\partial x_2}\delta x_2 + \cdots + \frac{\partial f_1}{\partial x_6}\delta x_6, \\
\delta y_2 &= \frac{\partial f_2}{\partial x_1}\delta x_1 + \frac{\partial f_2}{\partial x_2}\delta x_2 + \cdots + \frac{\partial f_2}{\partial x_6}\delta x_6, \\
&\vdots \\
\delta y_2 &= \frac{\partial f_2}{\partial x_1}\delta x_1 + \frac{\partial f_2}{\partial x_2}\delta x_2 + \cdots + \frac{\partial f_2}{\partial x_6}\delta x_6,
\end{aligned} \tag{2.13}$$

which can also be represented in vector notation:

$$\delta Y = \frac{\partial F}{\partial X}\delta X. \tag{2.14}$$

The 6×6 matrix of partial derivatives in (2.13) is the Jacobian. In robotic applications, we generally use Jacobians that relate joint velocities to Cartesian velocities of the end-effector.

2.3 Human Operator

The human operator in this work will not be modelled specifically. The human operator model is decided by the input device and the specific person that operates it. The human operator input is a given signal. It is an operation that will move at a gentle pace which is not too fast. In the simulation part, the input human operator torque will be modelled as a known signal to simulate two different situations. In the experiment part, there will be a force sensor that can be attached to the end effector of the manipulator which can sense the value of the input torque and transmit it to the master controller.

2.4 Master Dynamic Model

Consider an n-degree-of-freedom manipulator, the dynamic model of the master manipulator is proposed in the following expression:

$$M_m(\mathbf{q}_m)\ddot{\mathbf{q}}_m + C_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)\dot{\mathbf{q}}_m + G_m(\mathbf{q}_m) = \boldsymbol{\tau}_{hum} - \boldsymbol{\tau}_m + \Delta_m, \tag{2.15}$$

where $M_m(\mathbf{q}_m)$ is a $n \times n$ inertia matrix that is symmetric and positive definite, which stands for the inertial forces the robot manipulator experiences. $C_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)$

is a $n \times n$ matrix; $C_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)\dot{\mathbf{q}}_m$ represents an n-dimensional vector of Centripetal, Coriolis torques acting on the master side. $G_m(\mathbf{q}_m)$ denotes an n-dimensional vector of gravitational torques. $\mathbf{q}_m$ is a n-dimensional vector, representing the angular position of the n-links manipulator joint. $\dot{\mathbf{q}}_m$ represents the angular velocity of the manipulator joint. $\ddot{\mathbf{q}}_m$ is the joint acceleration. $\boldsymbol{\tau}_m$ is an n-vector denotes the control input torque for master side. τ_{hum} is the torque that the human operator applied to the master manipulator. Δ_m includes the approximation modelling error and external disturbance.

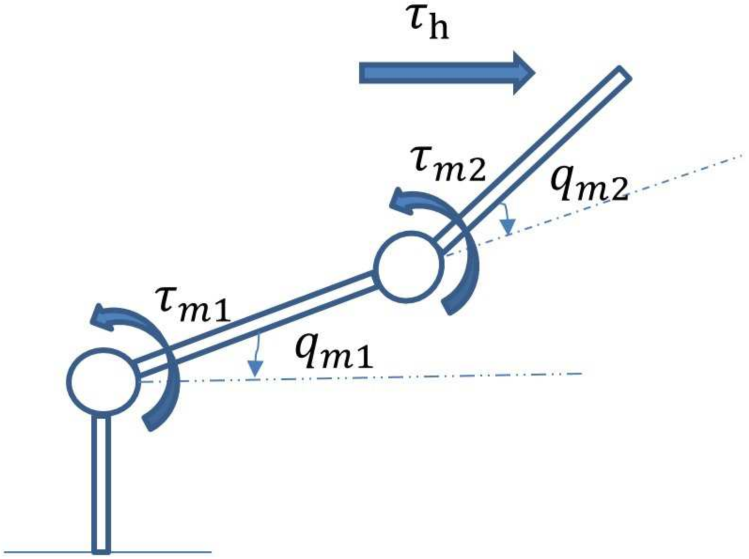

Figure 2.2: A Planar 2-Degree-of-Freedom master manipulator (as an example)

Fig.2.2 shows the sketch of the master manipulator and how the human operator torque $\boldsymbol{\tau}_h$ is applied. The movement of the master manipulator is planar.

2.5 Slave Dynamic Model

Similar to the master side, the slave manipulator was modeled as an n-degree-of-freedom manipulator with mass stiffness and damping properties. The dynamic expression is as follows,

$$M_s(\mathbf{q}_s)\ddot{\mathbf{q}}_s + C_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s + G_s(\mathbf{q}_s) = \boldsymbol{\tau}_s - \boldsymbol{\tau}_{ext} + \Delta_s, \tag{2.16}$$

where $M_s(\mathbf{q}_s)$ is a $n \times n$ inertia matrix that are symmetric and positive definite, dealing with the inertial forces the robot manipulator experiences. $C_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)$ is a $n \times n$ matrix; $C_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s$ denotes n-dimensional vector of Centripetal, Coriolis torques acting on the master and the slave side respectively. $G_s(\mathbf{q}_s)$ is an n-dimensional vector of gravitational torque. $\mathbf{q}_s$ is an n-dimensional vector, representing angular position of the n-links manipulator joints. $\dot{\mathbf{q}}_s$ represents the angular velocities of the manipulator joints. $\ddot{\mathbf{q}}_s$ is the joint acceleration for slave manipulator.

$\boldsymbol{\tau}_s$ is an n-vector representing the slave control input torque. $\boldsymbol{\tau}_{ext}$ stands for the measurable environmental torque applied to the slave manipulator. Δ_s is the approximation modelling error and external disturbance.

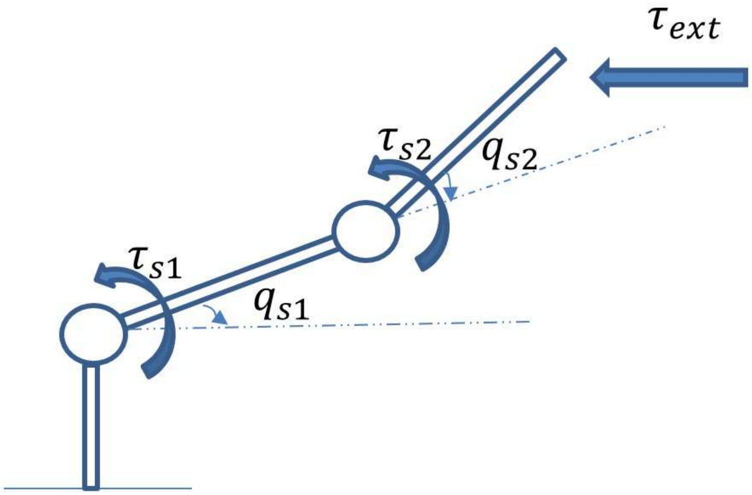

Figure 2.3: Planar 2-Degree-of-Freedom slave manipulator (as an example)

Fig.2.3 demonstrates the slave manipulator and how the environmental torque $\boldsymbol{\tau}_{ext}$ is formulated.

2.6 Important Properties

Here are some important properties of the nonlinear dynamic model in Eq.(2.16).

Property 1 For a manipulator with revolute joints, $M_m(\mathbf{q}_m)$ and $M_s(\mathbf{q}_s)$ are symmetric and positive definite. Moreover, their upper and lower bounds are also known:

$$0 < \lambda_{min}(M_i)I \leqslant M_i(\mathbf{q}_i) \leqslant \lambda_{max}(M_i)I \leqslant \infty$$

. Here M_i stands for M_m and M_s; $M_i(\mathbf{q}_i)$ stands for $M_m(\mathbf{q}_m)$ and $M_s(\mathbf{q}_s)$.

Property 2 The following equation shows the relationship between the Coriolis/ Centrifugal and the inertia matrices for a robot manipulator:

$$\dot{M}_i(\mathbf{q}_i) = C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + C_i^T(\mathbf{q}_i, \dot{\mathbf{q}}_i)$$

.

Property 3 For a manipulator with revolute joints, there exists a positive number η bounding the Coriolis and centrifugal term as follows:

$$|C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i| \leqslant \eta \dot{\mathbf{q}_i}^2$$

.

Due to the definition of the nonlinear dynamic model equation, we can know that all unknown or partially known parameters are contained within $M_i(\mathbf{q}_i)$, $C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ and $G_i(\mathbf{q}_i)$. Although $M_i(\mathbf{q}_i)$, $C_i(\dot{\mathbf{q}}_i, \mathbf{q}_i)$ and $G_i(\mathbf{q}_i)$ do contain highly nonlinear functions, it is still possible to state that the unknown or partially known parameters are coefficients of these functions. Then, we can define p as this m-dimensional vector, containing all the unknown parameters as well as the partially known parameters. And therefore, the above equation can be then written in the following form:

$$M_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i + G_i(\mathbf{q}_i) = Y_i(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i)p_i$$

Here $Y_i(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i)\mathbf{p}_i$ is an $n \ \times \ m$ matrix of known functions of the joint position, velocity and acceleration, which contains nonlinearities. And $\mathbf{p}_i$ is an m-dimensional vector containing all the unknown and partially known parameters.

This is a great property of robot manipulators dynamic equations that can be very useful. It gives the robot manipulators dynamic equation the ability to express as linear with respect to its parameters. So that the adaptive control method can be used here.

2.7 Environmental Model

The importance of the modelling of the environment can never be exaggerated to the whole teleoperation system. It is necessary to simulate the situation when the environment is taking effect on the slave manipulator. The more accurate the environment model is modelled, the more effective the controller works on the experimental hardware.

For my research work, the situation that being chosen is that when the slave manipulator is being positioned against a non-rigid surface, for example, a foam. This environmental situation is modelled as a spring and damping system based on the position, the velocity and the acceleration of the slave hardware with the parameters B_{ext}, K_{ext} and C_{ext}. When the slave manipulator interact with the foam, it would generate a reactionary torque. When the slave hardware is away from the foam, there will be no environmental interaction with the system.

The environmental torque is modelled in a general form as follows:

$$\boldsymbol{\tau}_{ext} = B_{ext}\dot{x}_s + K_{ext}x_s + C_{ext} = \boldsymbol{\theta}_{ext}^T \varphi_{ext}(\dot{\mathbf{q}}_s, \mathbf{q}_s, t) \tag{2.17}$$

$=$ Where $\theta_{ext} = [B_{ext}, K_{ext}, C_{ext}]^T$ represent all the unknown parameters, $\varphi_{ext} = [\dot{\mathbf{q}}_s, \mathbf{q}_s, t]$ are the known nonlinear functions with respect to regressors of the slave manipulator. This model can be used to describe different situations of the environment without modelling errors and disturbance. For example, when $\theta_{ext} = 0$, it stands for free motion.

2.8 Communication Channel

In this thesis, the new control structure of teleoperation system are designed to deal with arbitrary, long, and time-varying delays. Therefore, in order to test the effectiveness of the control strategy, it is important to set some appropriate assumptions for the communication channel. The upper limit of the time delay is set go be 2 seconds which is a very long delay for the normal teleoperation system. The lower bound of the time delay is set to be 0 second. The delays are modelled as a random number between the upper and lower bounds to represent the random nature of delays when working on network infrastructure.

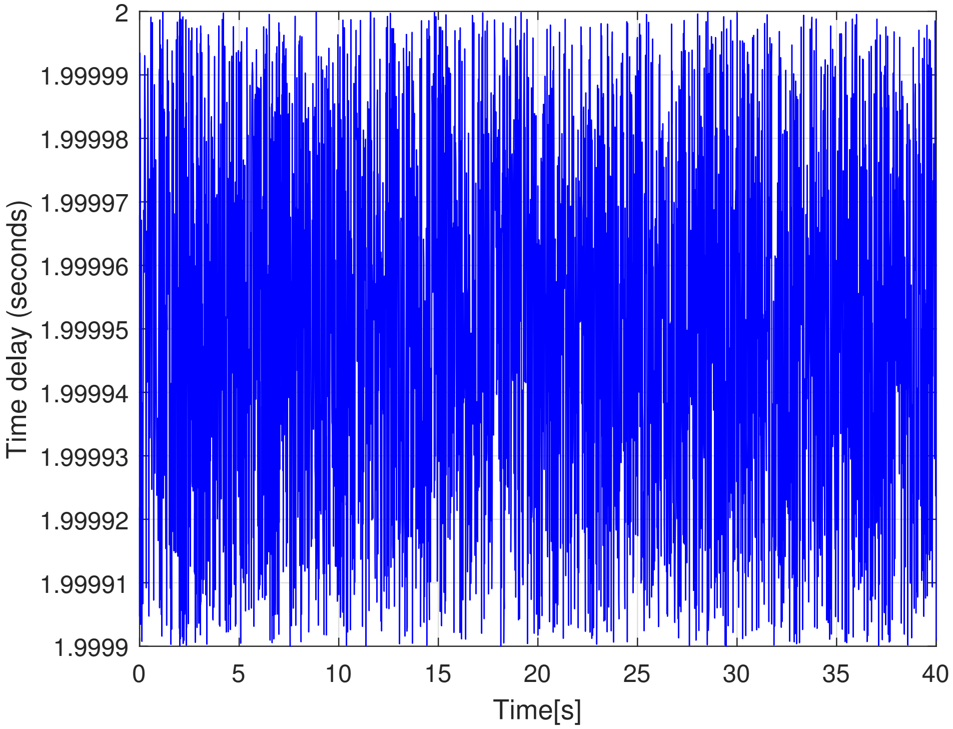

Figure 2.4: Communication channel time delay T_d

2.9 Summary

This chapter discussed the structure and models of the teleoperation system. The dynamic models of the master and the slave manipulator are presented. The human operator input are introduced. The modelling of the environment and the communication block are outlined.

Chapter 3

Control Design and Analysis

The master and slave controller design will be discussed in this chapter, as well as the stability proof of the teleoperation system with the designed controller.

It is assumed that there are some unknown parameters in the dynamic model of the robot manipulator. Hence part of the design of the control algorithm is based on the adaptive control algorithm.

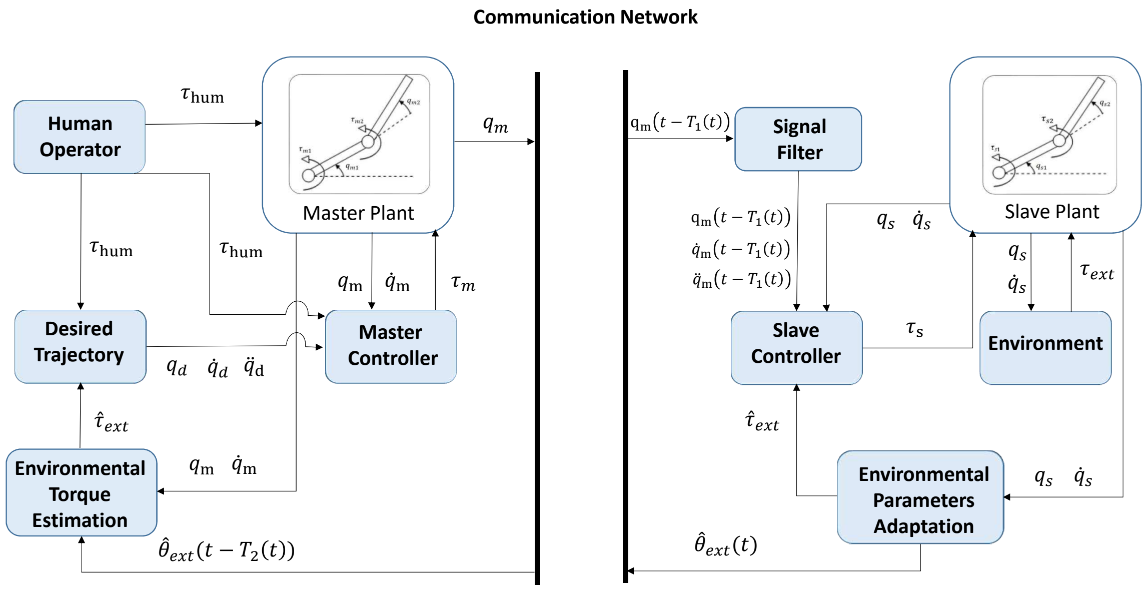

Figure 3.1: Control architecture of the closed-loop teleoperation system

The entire architecture of the proposed control design is as shown in Fig.3.1. A novel structure of communication block is developed where the master position signal q_m is transmitted to the slave side and from the slave side to the master side, estimated parameters of the environmental torque θ_e are estimated online and transmitted back.

3.1 Adaptive Control

Adaptive control method is usually developed to synchronize the state behavior of the local and the remote robots. The application of the adaptive controllers can be classified into three major areas [1].

1) Adaptive control for disturbance cancellation

This controller is designed to deal with the internal disturbance that comes from the model uncertainties from the master and slave dynamic model. It can also handle the external disturbances that resulting from the environment and the measurement noise.

There are external disturbances and internal disturbances in the control systems. The internal disturbances are usually caused by nonlinearities, unknown parameters, modelling errors and model mismatch between the real dynamic model and the simulated dynamic model. The external disturbance are usually resulting from unknown external forces, variation in friction, unknown external forces and characteristics of the system. Disturbances can usually destabilize the system and affect the performance of the overall system if it is not been properly dealt with. Therefore, the controller must be developed to guarantee the stability and the overall performance of the entire teleoperation system.

2) Adaptive control to estimate the operator and environment model

A major obstacle in the modeling and control of teleoperation systems is the largely unknown dynamics of the remote environment and the human operator, which are part of the global control loop. When a teleoperation system interacts with an arbitrary environment, the environment can exhibit zero stiffness (slave robot motion in free space), near infinite stiffness (slave robot motion in constrained space, such as pressing against a wall), and any stiffness in between. When analyzing the stability of the system, the environment stiffness directly affects the gain margin of the system. In most practical teleoperation systems, exact environmental stiffness values are not known in advance and vary during the manipulation. Hence, they should be estimated and updated in real time to ensure the stability and task performance of a system. Predicting the operator behavior and estimating the operator model facilitate task execution and improve the fidelity of the system. One of the major challenges of operator-oriented adaptive controllers is the modeling of the operator's behavior. In this section, some methods from the field of human-robot collaborative manipulation that assist in the design of operator-related adaptive control controllers are reviewed.

Environment-related adaptive controllers are adaptive schemes enhancing the robustness and performance of the teleoperation system through estimating and incorporating the environment model parameters into the controllers. The dynamic model of the environment is initially studied before the introduction of these controllers.

3) Adaptive control for communication delay compensation

Communication delay is one of the most challenging issue that need to be considered during control design. This issue can destabilize the system if it is not been well treated. The adaptive schemes are applied to the system to deal with the time delay problem.

Time delays have been divided into different groups based on their characteristics, such as constant time delays and time-varying delays.

3.2 Lyapunov-based Method Theory

Lyapunov proposed two different method for stability proof in his work in 1892 [37]. The first method developed the solution in a series which was then proved convergent within limits. The second method, which is now referred to as the Lyapunov stability criterion, makes use of a Lyapunov function $V(x)$ which has an analogy to the potential function of classical dynamics.

Consider the autonomous system

$$\dot{x} = f(x),$$

having a point of equilibrium at $x = 0$. If there exists a so-called Lyapunov function $V(x) : R^n \to R$, which satisfies the following conditions:

1. $V(x) = 0$ if and only if $x = 0$;

2. $V(x) > 0$ if $x \neq 0$(positive definite);

3. $\dot{V}(x) = \frac{d}{dt}V(x) \leq 0$ for all values of $x \neq 0$ (negative semi-definite). Note: for asymptotic stability, $\dot{V}(x) < 0$ for $x \neq 0$ is required (negative definite).

Then it is safe to say that the system is asymptotically stable at $x = 0$. In this case, the stability is independent of the initial condition $x(0)$. $V(x)$ is called a Lyapunov function candidate.

3.3 Master Controller Design

3.3.1 Controller Design for the Master Manipulator

For the master controller, a target impedance is designed to be followed by the master manipulator. The target impedance behaviour of the master manipulator is designed as:

$$M_d(\mathbf{q}_d)\ddot{\mathbf{q}}_d + C_d(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + G_d(\mathbf{q}_d) = \boldsymbol{\tau}_{hum} - \boldsymbol{\tau}_{ext}.$$

where M_d, C_d and G_d are the targeted parameters of the master manipulator.

It is assumed that there are some unknown parameters in the dynamic model of the robot manipulator. Hence part of the design of the control algorithm is based on the adaptive control algorithm.

The estimation of parameters in the dynamics are embedded in the expressions of the control law. Defining a switching-function-like quantity δ_m as follows.

$$\delta_m = \dot{\mathbf{e}}_{pm} + K_{m1}\mathbf{e}_{pm}, \tag{3.1}$$

where

$$\mathbf{e}_{pm} = \mathbf{q}_m(t) - \mathbf{q}_d(t), \tag{3.2}$$

and $K_{m1} > 0$ is a diagonal matrix. According to $\mathbf{e}_{pm} = \mathbf{q}_m(t) - \mathbf{q}_d(t)$, the following derivation can be drawn,

$$\dot{\mathbf{e}}_{pm} = \dot{\mathbf{q}}_m(t) - \dot{\mathbf{q}}_d(t). \tag{3.3}$$

Set a switching-function-like quantity

$$\dot{\mathbf{q}}_{mo} = \dot{\mathbf{q}}_d(t) - K_{m1}\mathbf{e}_{pm}. \tag{3.4}$$

Hence,

$$\ddot{\mathbf{q}}_{mo} = \ddot{\mathbf{q}}_d(t) - K_{m1}\dot{\mathbf{e}}_{pm}. \tag{3.5}$$

Therefore,

$$\dot{\boldsymbol{\delta}}_m = \ddot{\mathbf{q}}_m - \ddot{\mathbf{q}}_{mo}. \tag{3.6}$$

Seeing from above,

$$\begin{aligned} M_m\dot{\boldsymbol{\delta}}_m + C_m\boldsymbol{\delta}_m &= M_m\ddot{\mathbf{q}}_m - M_m\ddot{\mathbf{q}}_{mo} + C_m\dot{\mathbf{q}}_m - C_m\dot{\mathbf{q}}_{mo} \\ &= M_m\ddot{\mathbf{q}}_m + C_m\dot{\mathbf{q}}_m - [M_m\ddot{\mathbf{q}}_{mo} + C_m\dot{\mathbf{q}}_{mo}], \end{aligned} \tag{3.7}$$

where $\dot{\mathbf{q}}_{mo} = \dot{\mathbf{q}}_d(t) - K_{m1}\mathbf{e}_{pm}$, and $\ddot{\mathbf{q}}_{mo} = \ddot{\mathbf{q}}_d(t) - K_{m1}\dot{\mathbf{e}}_{pm}$. Rearranging (2.15),

$$M_m(\mathbf{q}_m)\ddot{\mathbf{q}}_m + C_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)\dot{\mathbf{q}}_m = \boldsymbol{\tau}_{hum} - \boldsymbol{\tau}_m + \boldsymbol{\Delta}_m - G_m(\mathbf{q}_m). \tag{3.8}$$

Substituting (3.8) into (3.7), we have

$$M_m\dot{\boldsymbol{\delta}}_m + C_m\boldsymbol{\delta}_m = \boldsymbol{\tau}_{hum} - \boldsymbol{\tau}_m + \boldsymbol{\Delta}_m - [G_m(\mathbf{q}_m) + M_m\ddot{\mathbf{q}}_{mo} + C_m\dot{\mathbf{q}}_{mo}] \tag{3.9}$$

Defining

$$M_m\ddot{\mathbf{q}}_{mo} + C_m\dot{\mathbf{q}}_{mo} + G_m(\mathbf{q}_m) = -Y_{mo}(\mathbf{q}_m, \dot{\mathbf{q}}_m, \ddot{\mathbf{q}}_{mo}, \dot{\mathbf{q}}_{mo})^T\theta_{mo}, \tag{3.10}$$

then

$$M_m\dot{\boldsymbol{\delta}}_m + C_m\boldsymbol{\delta}_m = \boldsymbol{\tau}_{hum} - \boldsymbol{\tau}_m + \boldsymbol{\Delta}_m + Y_{mo}(\mathbf{q}_m, \dot{\mathbf{q}}_m, \ddot{\mathbf{q}}_{mo}, \dot{\mathbf{q}}_{mo})^T\boldsymbol{\theta}_{mo}. \tag{3.11}$$

The following adaptive robust control law for the master side is designed,

$$\begin{aligned} &\boldsymbol{\tau}_m = \boldsymbol{\tau}_{m1} + \boldsymbol{\tau}_{m2}, \\ &\boldsymbol{\tau}_{m1} = -\hat{M}_m(\mathbf{q}_m)\ddot{\mathbf{q}}_{mo} - \hat{C}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)\dot{\mathbf{q}}_{mo} - \hat{G}_m(\mathbf{q}_m) + \boldsymbol{\tau}_{hum} \\ &\boldsymbol{\tau}_{m2} = K_{m2}\boldsymbol{\delta}_m + K_{\boldsymbol{\Delta m}}sgn(\boldsymbol{\delta}_m), \end{aligned} \tag{3.12}$$

where K_{m2} and $K_{\Delta m}$ are symmetric positive definite gain matrices. $\hat{M}_m(\mathbf{q}_m)$, $\hat{C}_s(\mathbf{q}_m, \dot{\mathbf{q}}_m)$, $\hat{G}_m(\mathbf{q}_m)$ are estimates of the actual $M_m(\mathbf{q}_m)$, $C_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)$ and $G_m(\mathbf{q}_m)$ vector respectively. And they have the same function form.

3.3.2 System Stability Proof

Substituting $\boldsymbol{\tau}_m$ into (3.11), we can get

$$\begin{aligned} M_m\dot{\boldsymbol{\delta}}_m + C_m\boldsymbol{\delta}_m = \hat{M}_m\ddot{\mathbf{q}}_{mo} + \hat{C}_m\dot{\mathbf{q}}_{mo} + \hat{G}_m(\mathbf{q}_m) - K_{m2}\boldsymbol{\delta}_m, \\ -K_{\Delta m}sgn(\boldsymbol{\delta}_m) + \boldsymbol{\Delta}_m - M_m\ddot{\mathbf{q}}_{mo} - C_m\dot{\mathbf{q}}_{mo} - G_m(\mathbf{q}_m), \end{aligned}$$

$$\begin{aligned} M_m\dot{\boldsymbol{\delta}}_m + C_m\boldsymbol{\delta}_m =& - \tilde{M}_m\ddot{\mathbf{q}}_{mo} - \tilde{C}_m\dot{\mathbf{q}}_{mo} - \tilde{G}_m(\mathbf{q}_m) - K_{m2}\boldsymbol{\delta}_m \\ & - K_{\Delta m}sgn(\boldsymbol{\delta}_m) + \boldsymbol{\Delta}_m \end{aligned} \tag{3.13}$$

where $\tilde{M}_m = M_m - \hat{M}_m$, $\tilde{C}_m = C_m - \hat{C}_m$, $\tilde{G}_m = G_m - \hat{G}_m$.

From (3.10), it follows that

$$Y_{mo}(\mathbf{q}_m, \dot{\mathbf{q}}_m, \ddot{\mathbf{q}}_{mo}, \dot{\mathbf{q}}_{mo})^T\tilde{\theta}_{mo} = -\tilde{M}_m\ddot{\mathbf{q}}_{mo} - \tilde{C}_m\dot{\mathbf{q}}_{mo} - \tilde{G}_m(\mathbf{q}_m).$$

The following Lyapunov function is used to prove the stability of the teleoperation system,

$$V_m = \frac{1}{2}\boldsymbol{\delta}_m^T M_m\boldsymbol{\delta}_m + \frac{1}{2}\tilde{\boldsymbol{\theta}}_m^T\Gamma_m^{-1}\tilde{\boldsymbol{\theta}}_m$$

where Γ_m represents a positive definite matrix.

The time derivative of V_m is

$$\dot{V}_m = \frac{1}{2}\boldsymbol{\delta}_m^T\dot{M}_m\boldsymbol{\delta}_m + \boldsymbol{\delta}_m^T M_m\dot{\boldsymbol{\delta}}_m + \tilde{\theta}_m^T\Gamma_m^{-1}\dot{\tilde{\boldsymbol{\theta}}}_m. \tag{3.14}$$

From (12), we can get,

$$M_m\dot{\boldsymbol{\delta}}_m = -C_m\boldsymbol{\delta}_m + Y_{mo}(\mathbf{q}_m, \dot{\mathbf{q}}_m, \ddot{\mathbf{q}}_{mo}, \dot{\mathbf{q}}_{mo})^T\tilde{\boldsymbol{\theta}}_{mo} - \tau_{m2} + \boldsymbol{\Delta}_m. \tag{3.15}$$

Substituting (3.15) into (3.14), we have

$$\begin{aligned} \dot{V}_m =& \boldsymbol{\delta}_m^T(\frac{1}{2}\dot{M}_m - C_m)\boldsymbol{\delta}_m + \boldsymbol{\delta}_m^T[Y_m\tilde{\boldsymbol{\theta}}_m - \boldsymbol{\tau}_{m2}] \\ & + \boldsymbol{\delta}_m^T\boldsymbol{\Delta}_m + \tilde{\boldsymbol{\theta}}_m^T\Gamma_m^{-1}\dot{\tilde{\boldsymbol{\theta}}}_m \\ =& [\boldsymbol{\delta}_m^T Y_m\tilde{\boldsymbol{\theta}}_m + \tilde{\boldsymbol{\theta}}_m^T\Gamma_m^{-1}\dot{\tilde{\boldsymbol{\theta}}}_m] - \boldsymbol{\delta}_m^T\boldsymbol{\tau}_{m2} + \boldsymbol{\delta}_m^T\boldsymbol{\Delta}_m. \end{aligned} \tag{3.16}$$

Property 2 can be applied to cancel $\boldsymbol{\delta}_m^T(\frac{1}{2}\dot{M}_m - C_m)\boldsymbol{\delta}_m$.

The following equation is the adaptive law used to estimate the manipulator unknown parameters $\boldsymbol{\theta}_m$.

$$\dot{\hat{\boldsymbol{\theta}}}_m = -\Gamma_m Y_m^T\boldsymbol{\delta}_m, \tag{3.17}$$

where Γ_m is the gain matrix. Here $\tilde{\boldsymbol{\theta}}_m = \boldsymbol{\theta}_m - \hat{\boldsymbol{\theta}}_m$, with the assumption that the variation of unknown parameter $\boldsymbol{\theta}_m$ is low, we can get $\dot{\tilde{\boldsymbol{\theta}}}_m = -\dot{\hat{\boldsymbol{\theta}}}_m$.

Combining (3.17) with (3.16), we have,

$$\begin{aligned}\dot{V}_m &= [\boldsymbol{\delta}_m^T Y_m \tilde{\boldsymbol{\theta}}_m + \tilde{\boldsymbol{\theta}}_m^T \Gamma_m^{-1}(-\Gamma_m Y_m^T \boldsymbol{\delta}_m)] - \boldsymbol{\delta}_m^T \boldsymbol{\tau}_{m2} + \boldsymbol{\delta}_m^T \boldsymbol{\Delta}_m \\ &= \tilde{\boldsymbol{\theta}}_m [Y_m^T \boldsymbol{\delta}_m - Y_m^T \boldsymbol{\delta}_m] - \boldsymbol{\delta}_m^T \boldsymbol{\tau}_{m2} + \boldsymbol{\delta}_m^T \boldsymbol{\Delta}_m \\ &= -\boldsymbol{\delta}_m^T \boldsymbol{\tau}_{m2} + \boldsymbol{\delta}_m^T \boldsymbol{\Delta}_m \\ &= -\boldsymbol{\delta}_m^T K_{m2} \boldsymbol{\delta}_m - \boldsymbol{\delta}_m^T K_{\boldsymbol{\Delta} m} sgn(\boldsymbol{\delta}_m) + \boldsymbol{\delta}_m^T \boldsymbol{\Delta}_m \\ &\leq -\lambda_{min}(K_{m2}) \parallel \boldsymbol{\delta}_m \parallel^2 \\ &\quad - \lambda_{min}(K_{\boldsymbol{\Delta} m}) \parallel \boldsymbol{\delta}_m \parallel + \parallel \boldsymbol{\Delta}_m \parallel \cdot \parallel \boldsymbol{\delta}_m \parallel, \end{aligned} \tag{3.18}$$

Note that $\boldsymbol{\Delta}_m$ is bounded as was introduced in the previous section. In the controller design, $\lambda_{min}(K_{\boldsymbol{\Delta} m})$ is designed to be larger than $\parallel \boldsymbol{\Delta}_m \parallel$. It can be derived that

$$- \parallel \boldsymbol{\delta}_m \parallel \cdot (\lambda_{min}(K_{\boldsymbol{\Delta}_{m,i}}) - \parallel \boldsymbol{\Delta}_m \parallel) < 0. \tag{3.19}$$

Thus (3.18) can be reduced to

$$\dot{V}_m \leq -\lambda_{min}(K_{m2}) \parallel \boldsymbol{\delta}_m \parallel^2 \leq 0. \tag{3.20}$$

Applying Barbalat's lemma, it can be concluded that,

$$\lim_{t \to \infty} \boldsymbol{\delta}_m(t) \to 0,$$

and

$$\lim_{t \to \infty} \mathbf{e}_m(t) \to 0.$$

3.4 Slave Controller Design

This section introduces the controller design process for the slave manipulator, along with the Lyapunov stability proof process for the designed slave controller.

3.4.1 Controller Design for the Slave Manipulator

Since in many real situation, the velocity and the acceleration of the master are not transmitted to the slave side directly, a signal filter is designed to produce $\dot{\mathbf{q}}_m(t - T_1(t))$

and $\ddot{\mathbf{q}}_m(t-T_1(t))$. A signal filter $H_r(s)\cdot s=\frac{1}{(Z_r s+1)^2}\cdot s$, in which Z_r is chosen as a positive real number, is used to produce $\dot{\mathbf{q}}_m(t-T_1(t))$, which can produce $\ddot{\mathbf{q}}_m(t-T_1(t))$ simultaneously.

Defining a switching-function-like quantity δ_s as follows.

$$\delta_s \triangleq \dot{\mathbf{e}}_{ps}+K_{s1}\mathbf{e}_{ps}, \tag{3.21}$$

where $\mathbf{e}_{ps}=\mathbf{q}_s(t)-\mathbf{q}_m(t-T_1(t))$, and $K_{s1}>0$ is a diagonal matrix. Thus,

$$\begin{aligned}\boldsymbol{\delta}_s &= \dot{\mathbf{e}}_{ps}+K_{s1}\mathbf{e}_{ps}\\ &= \dot{\mathbf{q}}_s-\dot{\mathbf{q}}_m(t-T_1(t))+K_{s1}(\mathbf{q}_s-\mathbf{q}_m(t-T_1(t)))\\ &= \dot{\mathbf{q}}_s-\dot{\mathbf{q}}_{so}.\end{aligned} \tag{3.22}$$

Hence, we can derive $\dot{\boldsymbol{\delta}}_s=\ddot{\mathbf{q}}_s-\ddot{\mathbf{q}}_{so}$, where $\dot{\mathbf{q}}_{so}=\dot{\mathbf{q}}_m(t-T_1(t))-K_{s1}\mathbf{e}_{ps}$, and $\ddot{\mathbf{q}}_{so}=(1-\dot{T}_1(t))\cdot\ddot{\mathbf{q}}_m(t-T_1(t))-K_{s1}\dot{\mathbf{e}}_{ps}$.

Deriving from (2.16) and (3.22), we can get the following expression.

$$\begin{aligned}M_s\dot{\boldsymbol{\delta}}_s+C_s\boldsymbol{\delta}_s &= M_s\ddot{\mathbf{q}}_s-M_s\ddot{\mathbf{q}}_{so}+C_s\dot{\mathbf{q}}_s-C_s\dot{\mathbf{q}}_{so}\\ &= M_s\ddot{\mathbf{q}}_s+C_s\dot{\mathbf{q}}_s-[M_s\ddot{\mathbf{q}}_{so}+C_s\dot{\mathbf{q}}_{so}],\end{aligned} \tag{3.23}$$

Rearranging the dynamic model of the slave manipulator,

$$M_s(\mathbf{q}_s)\ddot{\mathbf{q}}_s+C_s(\mathbf{q}_s,\dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s=\boldsymbol{\tau}_s-\boldsymbol{\tau}_{ext}+\boldsymbol{\Delta}_s-G_s(\mathbf{q}_s). \tag{3.24}$$

Substituting (3.24) into (3.23), we have

$$M_s\dot{\boldsymbol{\delta}}_s+C_s\boldsymbol{\delta}_s=\boldsymbol{\tau}_s-\boldsymbol{\tau}_{ext}+\boldsymbol{\Delta}_s-[G_s(\mathbf{q}_s)+M_s\ddot{\mathbf{q}}_{so}+C_s\dot{\mathbf{q}}_{so}]. \tag{3.25}$$

Define

$$M_s\ddot{\mathbf{q}}_{so}+C_s\dot{\mathbf{q}}_{so}+G_s(\mathbf{q}_s)=-Y_{so}(\mathbf{q}_s,\dot{\mathbf{q}}_s,\ddot{\mathbf{q}}_{so},\dot{\mathbf{q}}_{so})^T\theta_{so}, \tag{3.26}$$

then the error dynamics become:

$$M_s\dot{\boldsymbol{\delta}}_s+C_s\boldsymbol{\delta}_s=\boldsymbol{\tau}_s-\boldsymbol{\tau}_{ext}+\Delta_s+Y_{so}(\mathbf{q}_s,\dot{\mathbf{q}}_s,\ddot{\mathbf{q}}_{so},\dot{\mathbf{q}}_{so})^T\boldsymbol{\theta}_{so}. \tag{3.27}$$

The following adaptive robust control law for the slave side can be derived,

$$\begin{aligned}\boldsymbol{\tau}_s &= \boldsymbol{\tau}_{s1}+\boldsymbol{\tau}_{s2},\\ \boldsymbol{\tau}_{s1} &= \hat{M}_s(\mathbf{q}_s)\ddot{\mathbf{q}}_{so}+\hat{C}_s(\mathbf{q}_s,\dot{\mathbf{q}}_s)\dot{\mathbf{q}}_{so}+\hat{G}_s(\mathbf{q}_s)+\hat{\tau}_{ext}\\ \boldsymbol{\tau}_{s2} &= -K_{s2}\boldsymbol{\delta}_s-K_{\boldsymbol{\Delta s}}sgn(\boldsymbol{\delta}_s)\end{aligned} \tag{3.28}$$

where K_{s2} and $K_{\Delta s}$ are symmetric positive definite matrices. $\hat{M}_s(\mathbf{q}_s)$, $\hat{C}_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)$, $\hat{G}_s(\mathbf{q}_s)$ are estimates of the actual $M_s(\mathbf{q}_s)$, $C_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)$ and $G_s(\mathbf{q}_s)$ vector respectively. And they have the same function form.

3.4.2 System Stability Proof for the Slave Side

Substituting the expression for τ_s into (3.27), we can get

$$\begin{aligned} M_s\dot{\boldsymbol{\delta}}_s + C_s\boldsymbol{\delta}_s =& \hat{M}_s\ddot{\mathbf{q}}_{so} + \hat{C}_s\dot{\mathbf{q}}_{so} + \hat{G}_s(\mathbf{q}_s) - K_{s2}\boldsymbol{\delta}_s - K_{\boldsymbol{\Delta}s}sgn(\boldsymbol{\delta}_s) \\ &+ \boldsymbol{\Delta}_s - M_s\ddot{\mathbf{q}}_{so} - C_s\dot{\mathbf{q}}_{so} - G_s(\mathbf{q}_s), \end{aligned} \tag{3.29}$$

$$\begin{aligned} M_s\dot{\boldsymbol{\delta}}_s + C_s\boldsymbol{\delta}_s =& -\tilde{M}_s\ddot{\mathbf{q}}_{so} - \tilde{C}_s\dot{\mathbf{q}}_{so} - \tilde{G}_s(\mathbf{q}_s) \\ &- K_{s2}\boldsymbol{\delta}_s - K_{\boldsymbol{\Delta}s}sgn(\boldsymbol{\delta}_s) + \boldsymbol{\Delta}_s, \end{aligned} \tag{3.30}$$

where $\tilde{M}_s = M_s - \hat{M}_s$, $\tilde{C}_s = C_s - \hat{C}_s$, $\tilde{G}_s = G_s - \hat{G}_s$.

From (3.26) it follows that

$$Y_{so}(\mathbf{q}_s, \dot{\mathbf{q}}_s, \ddot{\mathbf{q}}_{so}, \dot{\mathbf{q}}_{so})^T\tilde{\theta}_{so} = -\tilde{M}_s\ddot{\mathbf{q}}_{so} - \tilde{C}_s\dot{\mathbf{q}}_{so} - \tilde{G}_s(\mathbf{q}_s).$$

To prove the stability of the slave side, the following Lyapunov function is constructed:

$$V_s = \frac{1}{2}\boldsymbol{\delta}_s^T M_s\boldsymbol{\delta}_s + \frac{1}{2}\tilde{\boldsymbol{\theta}}_s^T\Gamma_s^{-1}\tilde{\boldsymbol{\theta}}_s, \tag{3.31}$$

where Γ_s represents a positive definite matrix.

From (3.30),

$$M_s\dot{\boldsymbol{\delta}}_s = -C_s\boldsymbol{\delta}_s + Y_{so}(\mathbf{q}_s, \dot{\mathbf{q}}_s, \ddot{\mathbf{q}}_{so}, \dot{\mathbf{q}}_{so})^T\tilde{\theta}_{so} + \boldsymbol{\tau}_{s2} + \boldsymbol{\Delta}_s. \tag{3.32}$$

The time derivative of V_s is

$$\dot{V}_s = \frac{1}{2}\boldsymbol{\delta}_s^T\dot{M}_s\boldsymbol{\delta}_s + \boldsymbol{\delta}_s^T M_s\dot{\boldsymbol{\delta}}_s + \tilde{\boldsymbol{\theta}}_s^T\Gamma_s^{-1}\dot{\tilde{\boldsymbol{\theta}}}_s. \tag{3.33}$$

Using (3.32) and rearranging

$$\begin{aligned} \dot{V}_s &= \boldsymbol{\delta}_s^T(\frac{1}{2}\dot{M}_s - C_s)\boldsymbol{\delta}_s + \boldsymbol{\delta}_s^T[Y_s\tilde{\theta}_s + \boldsymbol{\tau}_{s2}] + \boldsymbol{\delta}_s^T\boldsymbol{\Delta}_s + \tilde{\theta}_s^T\Gamma_s^{-1}\dot{\tilde{\theta}}_s \\ &= [\boldsymbol{\delta}_s^T Y_s\tilde{\theta}_s + \tilde{\theta}_s^T\Gamma_s^{-1}\dot{\tilde{\theta}}_s] + \boldsymbol{\delta}_s^T\boldsymbol{\tau}_{s2} + \boldsymbol{\delta}_s^T\boldsymbol{\Delta}_s. \end{aligned} \tag{3.34}$$

Property 2 can be then applied to cancel the $\boldsymbol{\delta}_s^T(\frac{1}{2}\dot{M}_s - C_s)\boldsymbol{\delta}_s$ term.

The following adaptive law is designed to estimate the slave unknown parameters:

$$\dot{\hat{\theta}}_s = -\Gamma_s Y_s^T \boldsymbol{\delta}_s.$$

Define $\tilde{\boldsymbol{\theta}}_s = \boldsymbol{\theta}_s - \hat{\boldsymbol{\theta}}_s$. With the assumption that the variation of unknown parameter $\boldsymbol{\theta}_s$ is low, we can get $\dot{\tilde{\boldsymbol{\theta}}}_s = -\dot{\hat{\boldsymbol{\theta}}}_s$.

Combining (3.34) with the adaptation law, the equation becomes,

$$\begin{aligned}
\dot{V}_s &= [\boldsymbol{\delta}_s^T Y_s \tilde{\theta}_s + \tilde{\theta}_s^T \Gamma_s^{-1}(-\Gamma_s Y_s^T \boldsymbol{\delta}_s)] - \boldsymbol{\delta}_s^T \boldsymbol{\tau}_{s2} + \delta_s^T \boldsymbol{\Delta}_s \\
&= \tilde{\theta}_s [Y_m^T \boldsymbol{\delta}_s - Y_s^T \boldsymbol{\delta}_s] - \boldsymbol{\delta}_s^T \boldsymbol{\tau}_{s2} + \boldsymbol{\delta}_s^T \boldsymbol{\Delta}_s \\
&= -\boldsymbol{\delta}_s^T \boldsymbol{\tau}_{s2} + \boldsymbol{\delta}_s^T \boldsymbol{\Delta}_s \\
&= -\boldsymbol{\delta}_s^T K_{s2} \boldsymbol{\delta}_s - \boldsymbol{\delta}_s^T K_{\boldsymbol{\Delta} s} sgn(\boldsymbol{\delta}_s) + \boldsymbol{\delta}_s^T \boldsymbol{\Delta}_s \\
&\leq -\lambda_{min}(K_{s2}) \parallel \boldsymbol{\delta}_s \parallel^2 - \lambda_{min}(K_{\boldsymbol{\Delta} s}) \parallel \boldsymbol{\delta}_s \parallel + \parallel \boldsymbol{\Delta}_s \parallel \parallel \boldsymbol{\delta}_s \parallel . \qquad (3.35)
\end{aligned}$$

$\boldsymbol{\Delta}_s$ is bounded as introduced in the previous section. In the controller design, $\lambda_{min}(K_{\boldsymbol{\Delta} s})$ is designed to be larger than $\parallel \boldsymbol{\Delta}_s \parallel$. It can be derived that,

$$- \parallel \boldsymbol{\delta}_s \parallel (\lambda_{min}(K_{\Delta_s}) - \parallel \boldsymbol{\Delta}_s \parallel) < 0. \qquad (3.36)$$

Thus (3.35) can be reduced to

$$\dot{V}_s \leq -\lambda_{min}(K_{s2}) \parallel \boldsymbol{\delta}_s \parallel^2 \leq 0. \qquad (3.37)$$

Applying Barbalat's lemma, we can get the following conclusions,

$$\lim_{t\to\infty} \boldsymbol{\delta}_s(t) \to 0,$$

and

$$\lim_{t\to\infty} \mathbf{e}_s(t) \to 0.$$

3.5 Environmental Parameter Estimation

In real-world applications, the force sensors may not be available since they can increase the cost and the weight of the manipulator. Thus, the environmental torque needs to be estimated when the real torque is not measurable.

In this work, an environmental torque estimator is designed to estimate immeasurable torques which commonly occur in real applications. The unknown parameters of the environmental torques are estimated online by a least square adaptive law.

The following adaptive update law is used to estimate the immeasurable parameters in the environmental torque:

$$\dot{\hat{\boldsymbol{\theta}}}_{ext} = Proj_{\hat{\theta}_{ext}}(\Gamma_{ext} s_{ext}), \qquad \hat{\boldsymbol{\theta}}_{ext}(0) \in \Omega_{\theta_{ext}}. \tag{3.38}$$

Here s_{ext} is an estimation function, Γ_{ext} is a positive symmetric matrix. s_{ext}, Γ_{ext} will be determined later.

A projection mapping is used to keep the adaptation parameters within a known boundary,

$$Proj_{\hat{\theta}_{ext}}(\zeta) = \begin{cases} \zeta, & if \hat{\theta}_{ext} \in \overset{o}{\Omega}_{\theta_{ext}} \ \ or \ \ n^T_{\hat{\theta}_{ext}} \zeta \leq 0 \\ \left(I - \Gamma \frac{n_{\hat{\theta}_{ext}} n^T_{\hat{\theta}_{ext}}}{n^T_{\hat{\theta}_{ext}} \Gamma n_{\hat{\theta}_{ext}}} \right), & \hat{\theta}_{ext} \in \sigma\Omega_{\theta_{ext}} \ \ and \ \ n^T_{\hat{\theta}_{ext}} \zeta > 0 \end{cases}, \tag{3.39}$$

where $\zeta \in \Re^P$ is any function. $\overset{o}{\Omega}_{\theta_{ext}}$ and $\sigma\Omega_{\theta_{ext}}$ represent the interior and the boundary of $\Omega_{\theta_{ext}}$ respectively, and $n_{\hat{\theta}_{ext}}$ denotes the outward unit normal vector at $\hat{\theta}_{ext} \in \sigma\Omega_{\theta_{ext}}$.

With the adaptation law in (3.38)-(3.39), the following properties hold.

Property 4

The parameter estimates are always within the known bounded set $\overline{\Omega}_{\theta_{ext}}$, i.e., $\hat{\theta}_{ext}(t) \in \overline{\Omega}_{\theta_{ext}}$, $\forall t > 0$. Therefore, according to Assumption 1, $\theta_{min} \leqslant \hat{\theta}(t) \leqslant \theta_{max}$, $\forall t > 0$.

Property 5

$$\tilde{\theta}^T_{ext}(\Gamma^{-1} Proj_{\hat{\theta}_{ext}}(\Gamma s) - s) \leq 0, \qquad \forall t > 0. \tag{3.40}$$

The environmental torque is defined as $\tau_{ext,j} = Y^T_{ext,j}\theta_{ext}$, where $Y_{ext,j} = [\dot{q}_{s,j}, q_{s,j}, 1]^T$.

The estimated environmental torque can be defined as :

$$\hat{\tau}_{ext,j} = Y^T_{ext,j}\hat{\theta}_{ext}. \tag{3.41}$$

Thus, the estimation error is $\varepsilon_j = \hat{\tau}_{ext,j} - \tau_{ext,j}$. This error can be modeled as follows,

$$\varepsilon_j = Y^T_{ext,j}\tilde{\theta}_{ext}. \tag{3.42}$$

Γ_{ext} and s_{ext} can be designed as,

$$\dot{\Gamma}_{ext} = \begin{cases} \upsilon\Gamma_{ext} - \frac{1}{1+\alpha Y_{ext,j}^{T}\Gamma_{ext}Y_{ext,j}}\Gamma_{ext}Y_{ext,j}Y_{ext,j}^{T}\Gamma_{ext}, & \text{if } \lambda_{max}(\Gamma_{ext(t)}) \\ & \leq \beta_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.43}$$

and,

$$s_{ext} = \frac{1}{1+\alpha Y_{ext,j}^{T}\Gamma_{ext}Y_{ext,j}}Y_{ext,j}\varepsilon, \tag{3.44}$$

where υ is the forgetting factor, $\alpha \geq 0$, β_{max} is the upper bound that is set to avoid the estimator windup.

Property 6 When the projection least square type estimation algorithm is used, $\hat{\theta}_{ext}$ is always within the known boundary: $\theta_{min} \leqslant \hat{\theta}(t) \leqslant \theta_{max}$, $\forall t > 0$, if the following condition is met,

$$\int_{t}^{t+T} Y_{ext,j}Y_{ext,j}^{T}ds \geq \gamma I, \forall t > t_0 \ for\ some\ T > 0\ and\ \gamma < 0, \tag{3.45}$$

then $\hat{\theta}_{ext}$ converges to θ_{ext}.

3.6 Summary

This chapter discussed about the control design for the bilateral teleoperation system. The derivation of the master and the slave controller and the stability proof of the controller design are depicted in details. The development of the environmental estimator is also outlined in the chapter.

Chapter 4

Simulation and Results of the Proposed Approach

In this chapter, the simulation results on a pair of two DOF manipulators are shown.

4.1 Numerical Dynamic Model

A pair of two-degree-of-freedom, articulated robotic manipulators with revolute joints are being used as the master and the slave hardware.

Master and slave dynamic models have the moments of inertia, Coriolis/centrifugal and gravity matrices

$M_i(q_i) = \begin{bmatrix} M_{i_{11}} & M_{i_{12}} \\ M_{i_{21}} & M_{i_{22}} \end{bmatrix}$, $C_i(q_i, \dot{q}_i) = \begin{bmatrix} C_{i_{11}} & C_{i_{12}} \\ C_{i_{21}} & C_{i_{22}} \end{bmatrix}$ and

$G_i(q_i) = \begin{bmatrix} G_{i_1} \\ G_{i_2} \end{bmatrix}$, for $i \in \{m, s\}$,

where

$$\begin{aligned}
M_{i_{11}} &= l_{i_2}^2 m_{i_2} + l_{i_1}^2(m_{i_1} + m_{i_2}) + 2l_{i_1} l_{i_2} m_{i_2} cos(\mathbf{q}_{i_2}), \\
M_{i_{21}} &= M_{i_{12}} = l_{i_2}^2 m_{i_2} + l_{i_1} l_{i_2} m_{i_2} cos(\mathbf{q}_{i_2}), \\
M_{i_{22}} &= l_{i_2}^2 m_{i_2}, \\
C_{i_{/}11} &= -2l_{i_1} l_{i_2} m_{i_2} sin(\mathbf{q}_{i_2}) \dot{\mathbf{q}}_{i_2}, \\
C_{i_{12}} &= -l_{i_1} l_{i_2} m_{i_2} sin(\mathbf{q}_{i_2}) \dot{\mathbf{q}}_{i_2}, \\
C_{i_{21}} &= l_{i_1} l_{i_2} m_{i_2} sin(\mathbf{q}_{i_2}) \dot{\mathbf{q}}_{i_1}, \\
C_{i_{22}} &= 0, \\
G_{i_1} &= g l_{i_2} m_{i_2} cos(\mathbf{q}_{i_1} + \mathbf{q}_{i_2}) + l_{i_1}(m_{i_1} + m_{i_2}) cos(\mathbf{q}_{i_1}), \\
G_{i_2} &= g l_{i_2} m_{i_2} cos(\mathbf{q}_{i_1} + \mathbf{q}_{i_2}).
\end{aligned}$$

$\mathbf{q}_{i_1}$ and $\mathbf{q}_{i_2}$ represent the angular position of the first and second revolute joints respectively. l_{i_1} and l_{i_2} denote the length of the first and second link for the manipulator while m_{i_1} and m_{i_2} are the masses of the two links respectively.

4.2 Human Operator

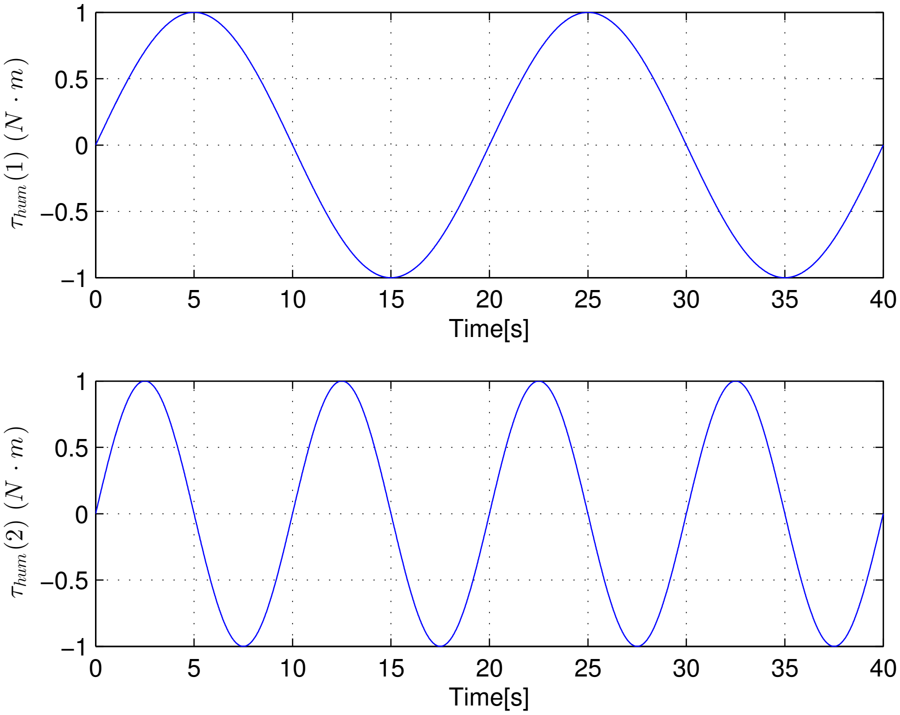

Figure 4.1: Simulated sinusoidal human operator torque $\boldsymbol{\tau}_{hum}$

Two cases are considered for the human operator input. In the first case, in order to test how the master and the slave manipulators will react to a constantly changing input, a sine wave input is considered. In the second case, a square input which can be seen in Fig.4.2 is chosen, considering how the manipulators will react to an abrupt changing input.

$$\begin{aligned} \tau_{hum}(1) &= sin(0.1\pi t), \\ \tau_{hum}(2) &= sin(\pi t/5). \end{aligned} \tag{4.1}$$

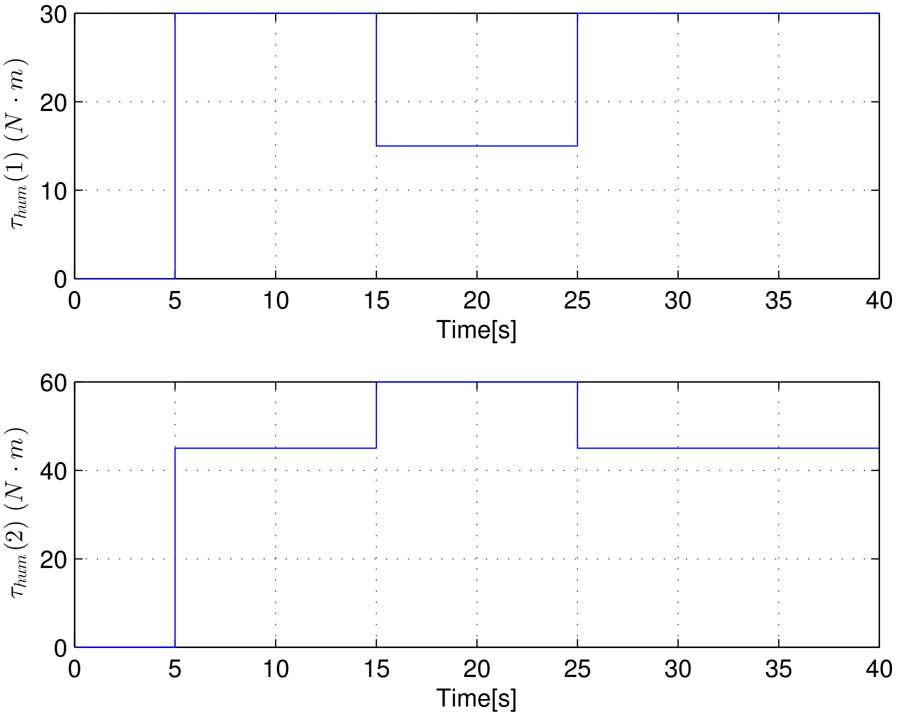

Figure 4.2: Simulated square wave human operator torque $\boldsymbol{\tau}_{hum}$

4.3 System Parameters

The following table contains the list of parameters being used when doing the simulation.

$m_{m_1} = 4.0$ kg	$m_{m_2} = 0.5$ kg	$l_{m_1} = 50$ cm	$l_{m_2} = 50$ cm
$m_{s_1} = 4.0$ kg	$m_{s_2} = 0.5$kg	$l_{s_1} = 50$ cm	$l_{s_2} = 50$ cm

The same linear parameterization used for both manipulators is shown as follows:

$$Y_i(\mathbf{q}_i, \dot{\mathbf{q}}_i, \dot{\mathbf{q}}_{io}, \ddot{\mathbf{q}}_{io}) = \begin{bmatrix} Y_{i_{11}} & Y_{i_{12}} & Y_{i_{13}} & Y_{i_{14}} & Y_{i_{15}} \\ Y_{i_{21}} & Y_{i_{22}} & Y_{i_{23}} & Y_{i_{24}} & Y_{i_{25}} \end{bmatrix},$$

$$\hat{\theta}_i = [\hat{\theta}_{i_1}, \hat{\theta}_{i_2}, \hat{\theta}_{i_3}, \hat{\theta}_{i_4}, \hat{\theta}_{i_5}]$$

where

$$
\begin{aligned}
Y_{i_{11}} &= -\ddot{\mathbf{q}}_{io_1},\\
Y_{i_{12}} &= -2\ddot{\mathbf{q}}_{io_1}cos(\mathbf{q}_{i_2}) - \ddot{\mathbf{q}}_{io_2}cos(\mathbf{q}_{i_2}) + \dot{\mathbf{q}}_{i_2}\dot{\mathbf{q}}_{io_2}sin(\mathbf{q}_{i_2}) + 2\dot{\mathbf{q}}_{io_1}\dot{\mathbf{q}}_{i_2}sin(\mathbf{q}_{i_2}),\\
Y_{i_{13}} &= -\ddot{\mathbf{q}}_{io_2},\\
Y_{i_{14}} &= -gcos(\mathbf{q}_{i_1} + \mathbf{q}_{i_2}),\\
Y_{i_{15}} &= -gcos(\mathbf{q}_{i_1}),\\
Y_{i_{21}} &= 0,\\
Y_{i_{22}} &= -\ddot{\mathbf{q}}_{io_1}cos(\mathbf{q}_{i_2}) - \dot{\mathbf{q}}_{i_1}\dot{\mathbf{q}}_{io_1}sin(\mathbf{q}_{i_2}),\\
Y_{i_{23}} &= -\ddot{\mathbf{q}}_{io_1} - \ddot{\mathbf{q}}_{io_2},\\
Y_{i_{24}} &= -gcos(\mathbf{q}_{i_1} + \mathbf{q}_{i_2}),\\
Y_{i_{25}} &= 0,
\end{aligned}
$$

and

$$
\begin{aligned}
\hat{\theta}_{i_1} &= \hat{l}_{i_2}^2\hat{m}_{i_2} + \hat{l}_{i_1}^2(\hat{m}_{i_1} + \hat{m}_{i_2}),\\
\hat{\theta}_{i_2} &= \hat{l}_{i_1}\hat{l}_{i_2}\hat{m}_{i_2},\\
\hat{\theta}_{i_3} &= \hat{l}_{i_2}^2\hat{m}_{i_2},\\
\hat{\theta}_{i_4} &= \hat{l}_{i_2}\hat{m}_{i_2},\\
\hat{\theta}_{i_5} &= \hat{l}_{i_1}(\hat{m}_{i_1} + \hat{m}_{i_2}),
\end{aligned}
$$

for $i \in \{m, s\}$. Using the preceding definition of the elements in the vector $\hat{\theta}_i$, we can estimate the elements in M_i, C_i and G_i.

4.4 Controller and Design Parameters

The environmental torque is modeled in a general form as

$$\boldsymbol{\tau}_{ext} = B_{ext}\dot{\mathbf{q}}_s + K_{ext}\mathbf{q}_s + C_{ext}, \tag{4.2}$$

where $B_{ext} = 4.5$, $K_{ext} = 100$, $C_{ext} = -0.2$.

$T_1(t)$ and $T_2(t)$ are two large time-varying delays that are set in the two channels between the master and the slave side. In the simulation, $T_1(t)$ and $T_2(t)$ are all set to a maximum of 2 seconds.

The approximated modeling errors and external disturbance in (2.2) and (2.3) are assumed to be

$$\Delta_m = [3sin(4\pi t/3), 3sin(4\pi t/3)]^T$$
$$\Delta_s = [3sin(4\pi t/3), 3sin(4\pi t/3)]^T.$$

Robust control gains are set to be

$$K_{\Delta m} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, K_{\Delta s} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \tag{4.3}$$

respectively. This value is decided to compensate the effect that caused by the disturbance. If the value is too high, the output will have some unexpected chattering.

In the controller design part, the parameters are set to be

$$\nu = 0.02,$$
$$\alpha = 0.1,$$
$$\beta_{max} = 5000.$$

The initial adaptation rates are $\Gamma_{ext}(0) = diag\{100, 100, 100\}$. The initial parameter estimates are $\hat{\theta}_{ext}(0) = [2, 50, 0]^T$. In the first case and $\hat{\theta}_{ext}(0) = [0, 0, 0]^T$, in the second case respectively.

The upper and the lower bounds of the estimated environmental parameters are set to

$$\theta_{ext,max} = [10, 10, 10]^T,$$

and

$$\theta_{ext,min} = [-10, -10, -10]^T.$$

$\Gamma_m = 40I$, $\Gamma_s = 60I$. K_{m1} and K_{s1} are set to be

$$K_{m1} = K_{s1} = diag\{1, 1\}.$$

$$K_{m2} = K_{s2} = diag\{1, 1\}.$$

The filter constant was set to $Z_r = 20$.

4.5 Case 1: Simulation with Sinusoidal Input

A sinusoid with a magnitude of $1N$ and a frequency of 0.1 and 0.2 is chosen for the first and second link respectively as shown in Fig.4.1. Fig.4.3 shows the desired trajectories $\mathbf{q}_d$ of the first and the second joint respectively. The final value of the desired trajectory converges to a bound (from 0.01205 rad to -0.008 rad for link 1 and from 0.01216 rad to -0.008195 rad for link 2.)

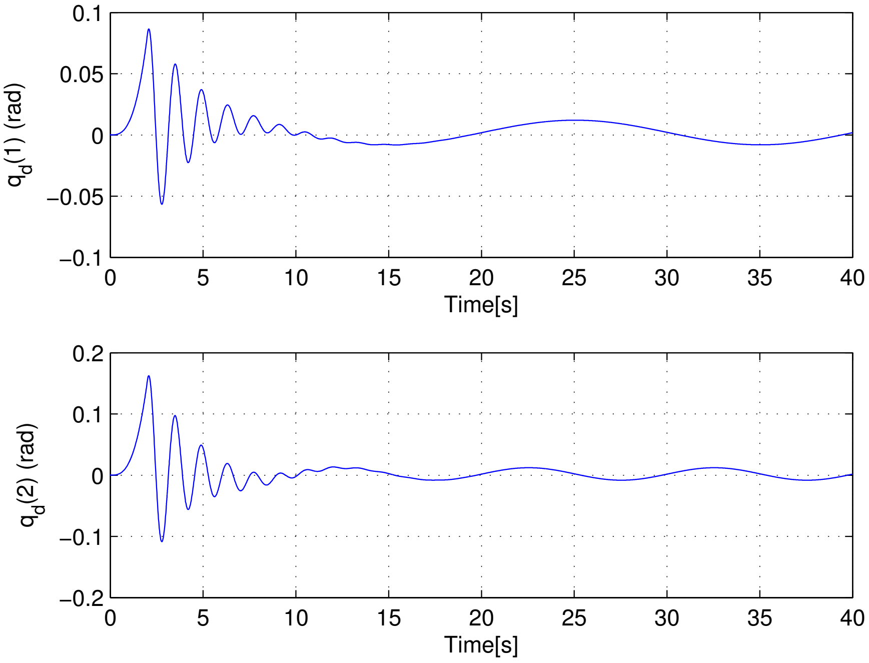

Figure 4.3: Desired trajectory $\mathbf{q}_d$

Fig.4.4 shows the trajectories of the master manipulator for two joints. And the final trajectories of the master hardware also reach to a steady state within a bound (from 0.01202 rad to -0.008 rad for link 1 and from 0.01268 rad to -0.008483 rad for link 2).

Fig.4.5 shows the tracking error $\mathbf{e}_{pm}$ between the master and the desired trajectories. As can be seen in the figures, the tracking errors converge to zero in the end within 4 seconds. For link 1, the tracking error converge to zero at the point of 5 seconds; for link 2, the tracking error converge to zero at the point of 4.878 seconds which can verify the effectiveness of the controller designed in the previous sections.

Fig.4.6 shows the trajectories of the slave manipulator for link 1 and link 2

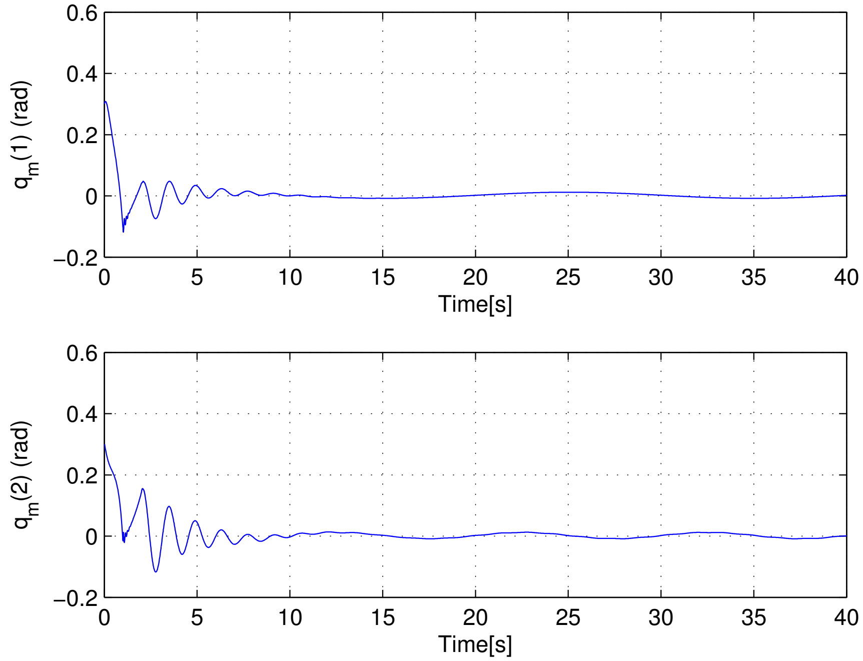

Figure 4.4: Master trajectory $\mathbf{q}_m$

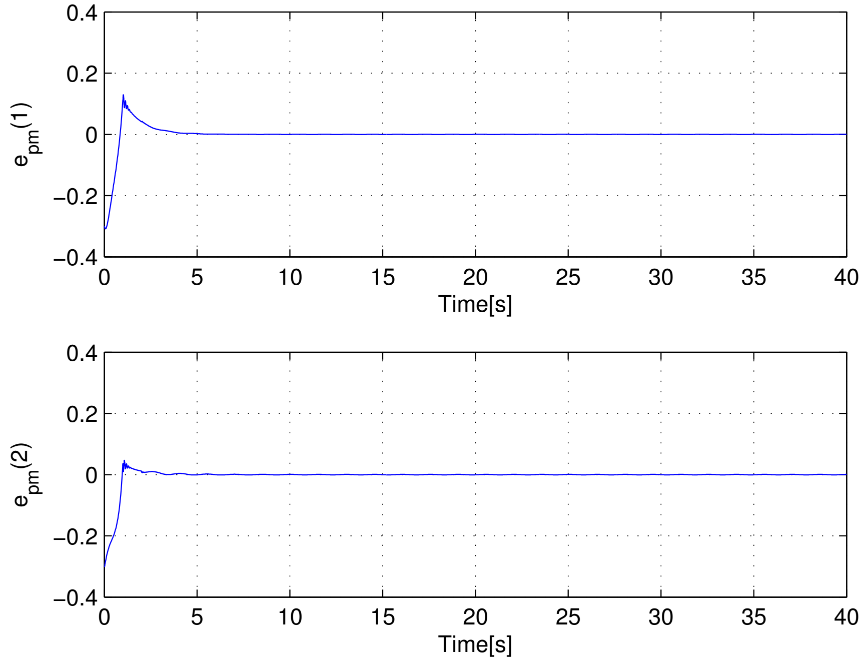

Figure 4.5: Tracking error $\mathbf{e}_{pm} = \mathbf{q}_m(t) - \mathbf{q}_d(t)$

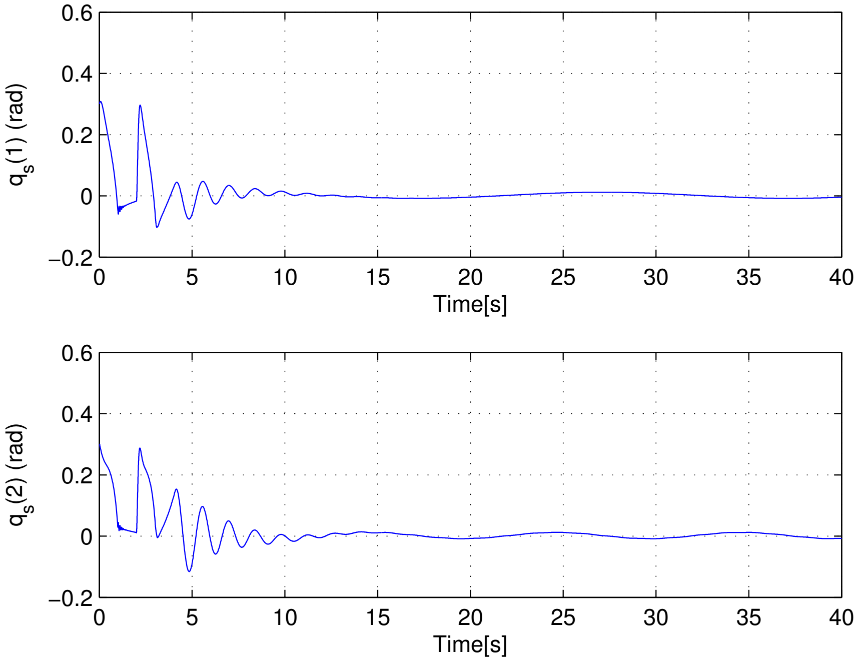

Figure 4.6: Slave trajectory $\mathbf{q}_s$

respectively. The final value of the slave positions converge to a bound (from 0.01205 rad to -0.008033 rad for link 1 and from 0.01257 rad to -0.008449 rad for link 2).

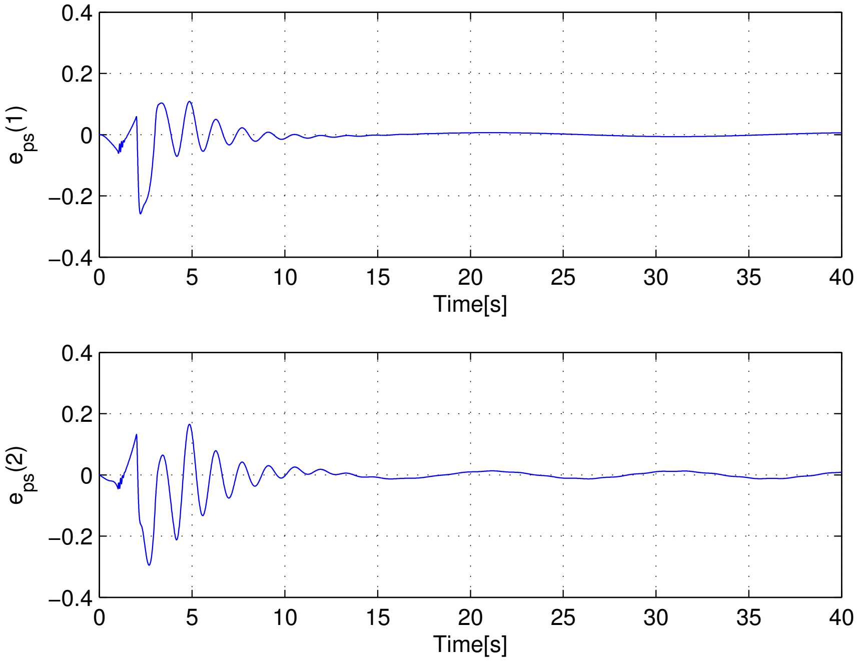

Figure 4.7: Tracking error $\mathbf{e}_{ps} = \mathbf{q}_s(t) - \mathbf{q}_m(t - T_1(t)))$

The tracking errors between the master and the slave manipulators for the two joints are shown in Fig.4.7. It converges to approximately zero after around 10 seconds, which verifies the robust stability and the effectiveness of the control method as well, even under the communication delay as large as up to 2 seconds. The master and the slave control input torques $\boldsymbol{\tau}_m$ and $\boldsymbol{\tau}_s$ are shown in Fig.4.8 and Fig.4.11 respectively. As it can be seen in the plots, the control input torque has a high peak around 1.5 seconds. One possible reason is that, the adaptive control gain was set to 40 on the master side and 60 on the slave side. The control gain value is set to be a high value, so that the response of the control torque is high as well. Another possible reason is that, as it can be seen in Fig.4.16, the value of K_{ext} is set to be 100, which is also a high value.

Fig.4.14 shows the real value of the environmental torque that fed back to the slave manipulator. For joint 1, the value converge to a bound from 1.006 $N \cdot m$ to -1 $N \cdot m$. For joint 2, the value converge to a bound from 1.005 $N \cdot m$ to -1.005 $N \cdot m$. Fig.4.15 shows the estimation of the environmental torque. It converges to a bound (from 1.007 $N \cdot m$ to -1.007 $N \cdot m$ for link 1 and from 1.063 $N \cdot m$ to -1.051 $N \cdot m$ for link 2).

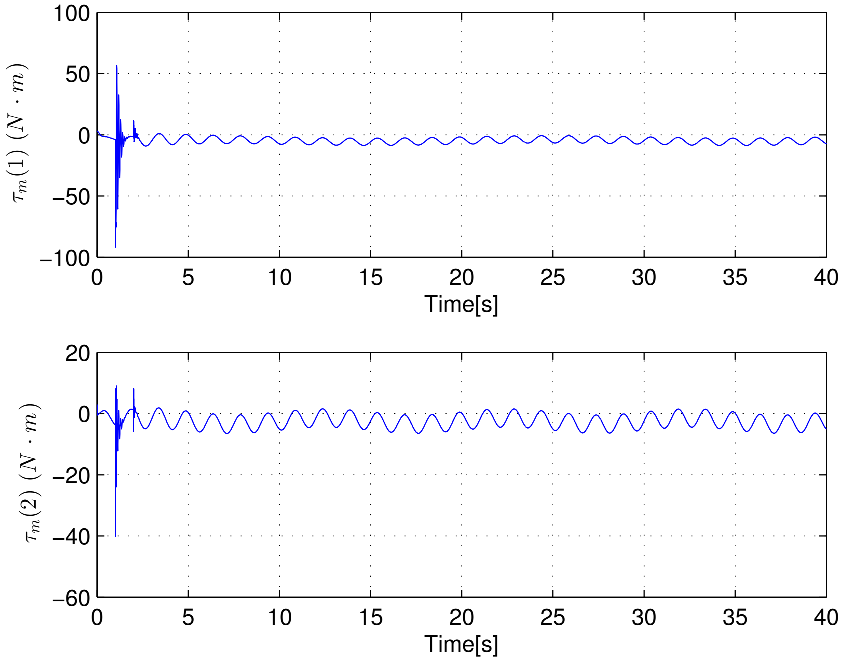

Figure 4.8: Master control input torque $\boldsymbol{\tau}_m$

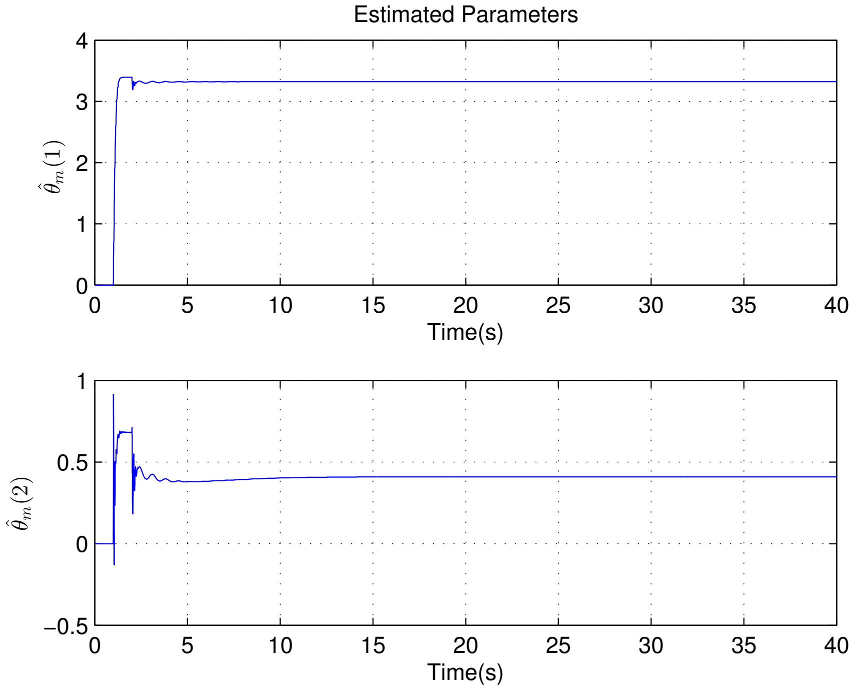

Figure 4.9: Estimated parameters $\hat{\theta}_m(1)$ and $\hat{\theta}_m(2)$

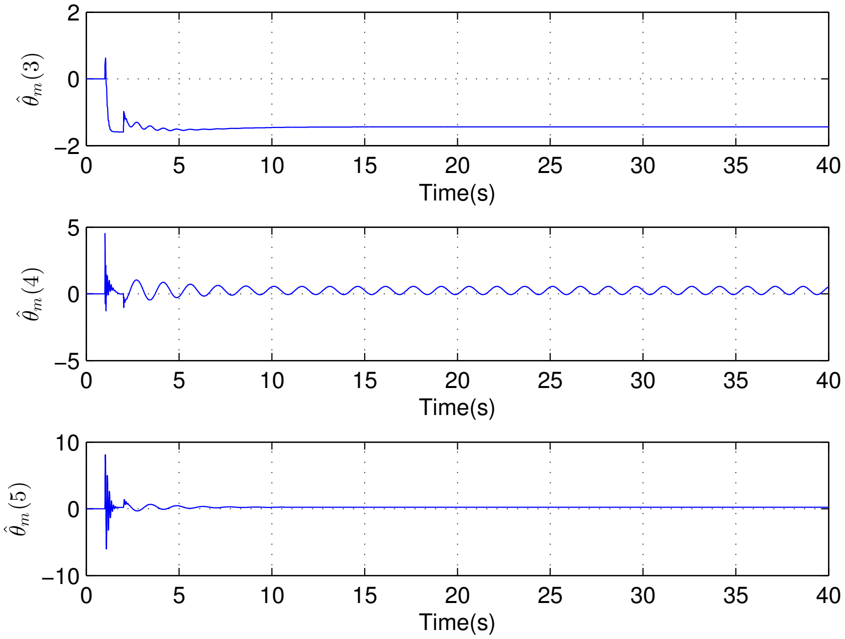

Figure 4.10: Estimated parameters $\hat{\theta}_m(3)$,$\hat{\theta}_m(4)$ and $\hat{\theta}_m(5)$

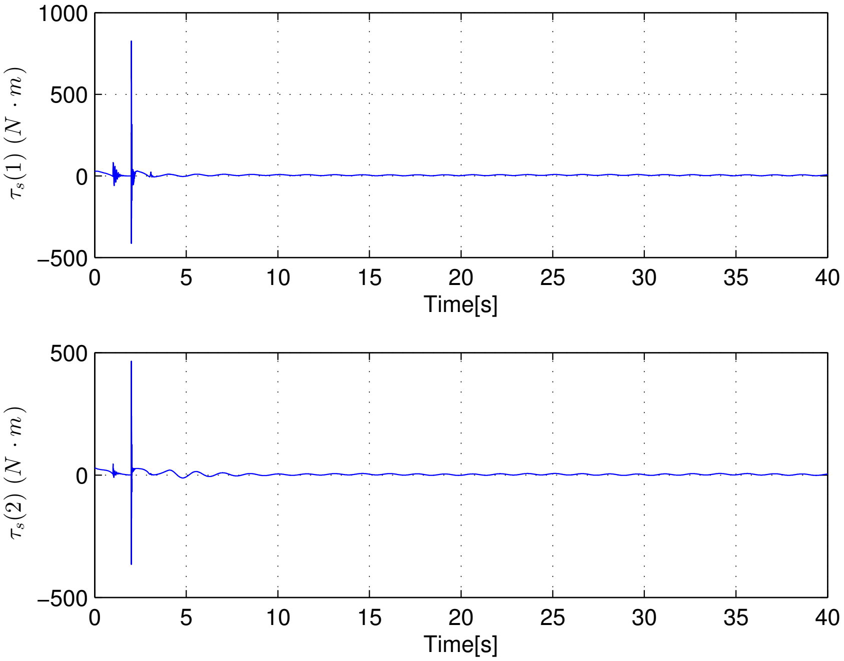

Figure 4.11: Slave control input torque $\boldsymbol{\tau}_s$

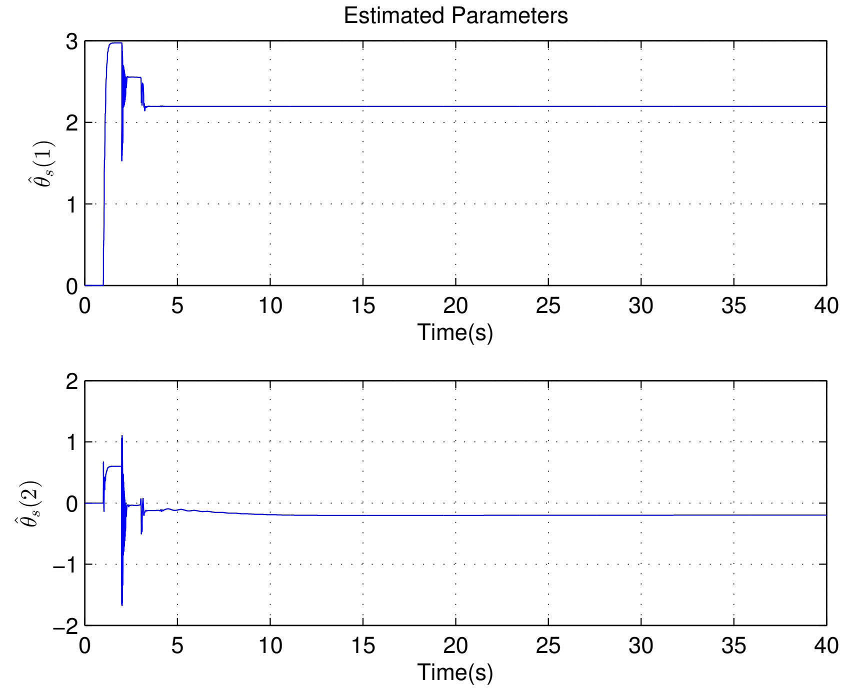

Figure 4.12: Estimated parameters $\hat{\theta}_s(1)$ and $\hat{\theta}_s(2)$

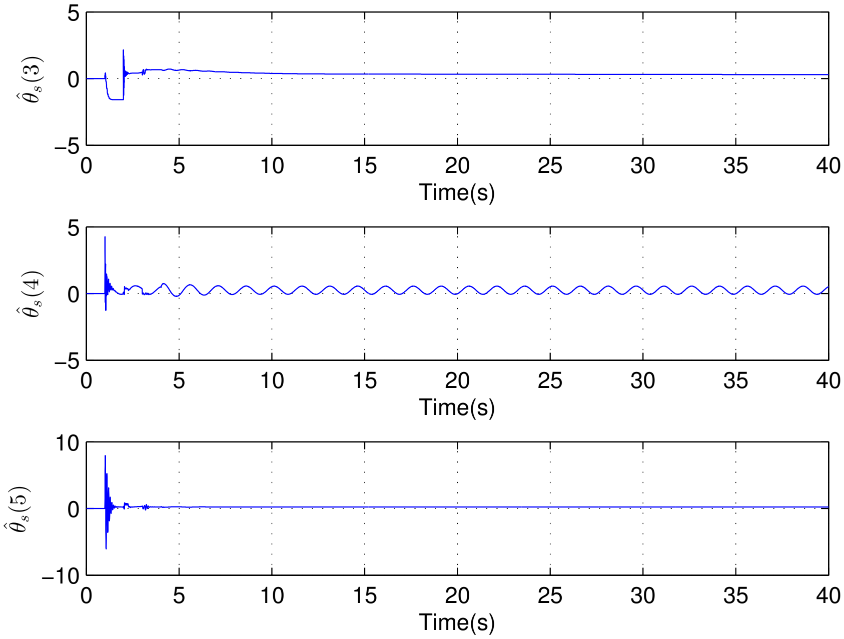

Figure 4.13: Estimated parameters $\hat{\theta}_s(3)$,$\hat{\theta}_s(4)$ and $\hat{\theta}_s(5)$

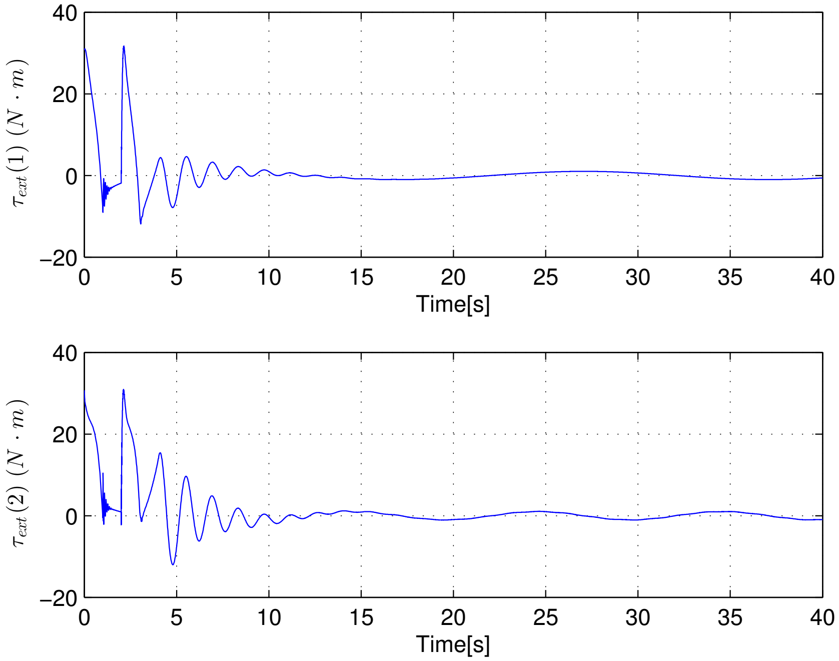

Figure 4.14: Environmental torque $\boldsymbol{\tau}_{ext}$

The values of the estimated environmental torque and the real environmental torque are almost same. Therefore, the environmental torques are effectively estimated.

The estimated environmental parameters $\hat{\theta}_{ext}$ are shown in Fig.4.17. The parameters converge to their true values B_{ext} for 4.5, K_{ext} for 100 and C_{ext} for -0.2 respectively.

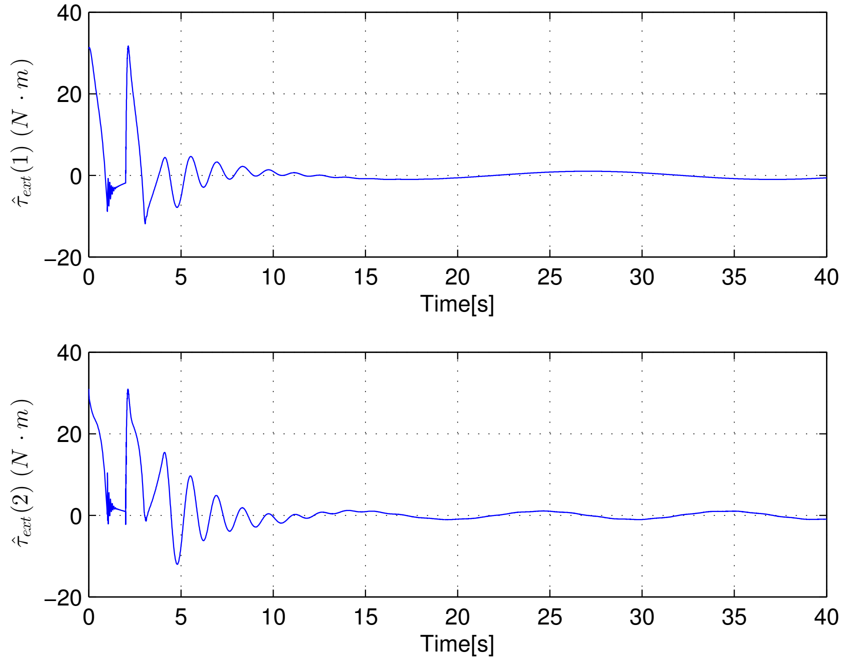

Figure 4.15: Environmental torque estimation $\hat{\tau}_{ext}$

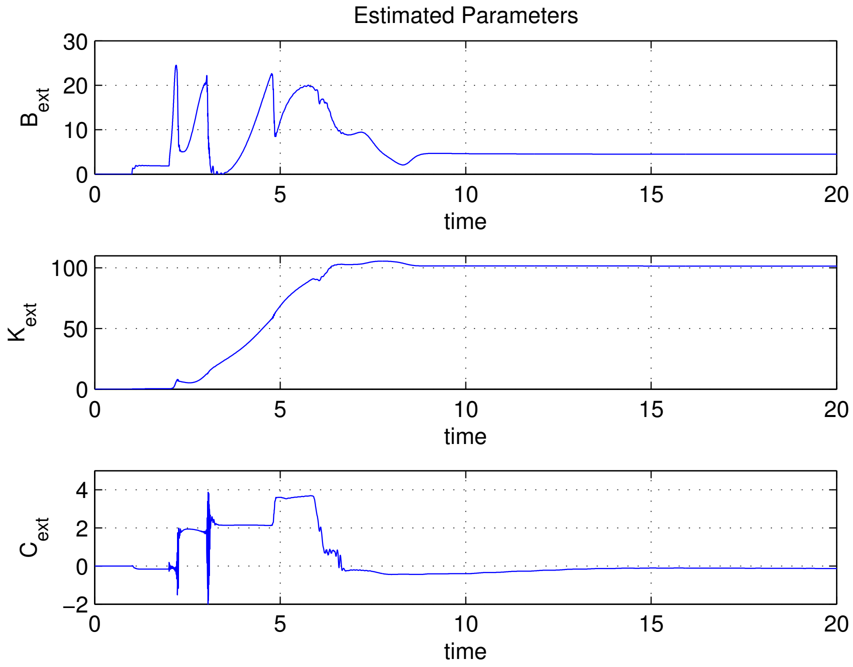

Figure 4.16: Environmental estimated parameters $\boldsymbol{\theta}_{ext} = [B_{ext}, K_{ext}, C_{ext}]$

4.6 Case 2: Simulation with Square Input

Fig.4.17 shows the desired trajectories $\mathbf{q}_d$ of the first and the second joint respectively. The final value of the desired trajectory converge to steady state of 0.3202 rad for link 1 and 0.4698 rad for link 2.

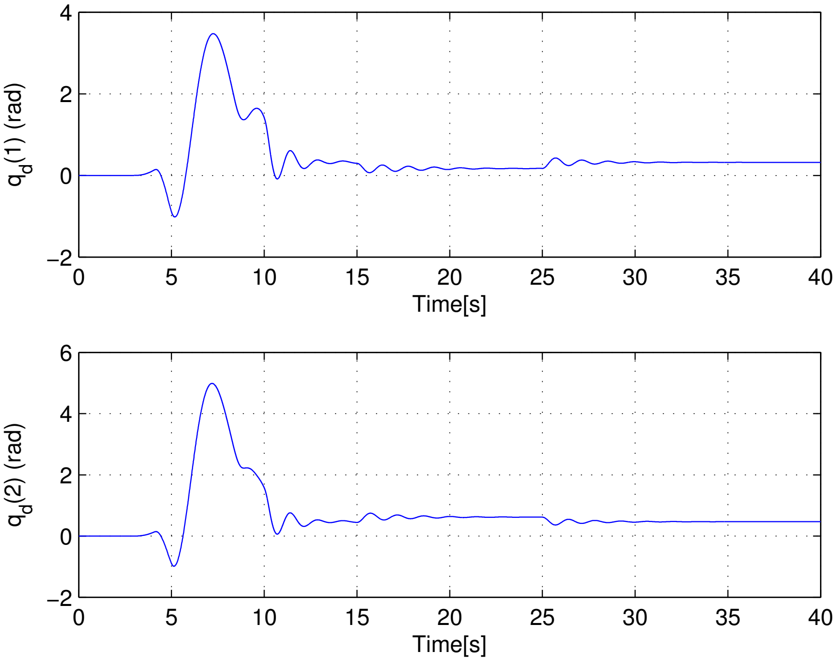

Figure 4.17: Desired trajectory $\mathbf{q}_d$

Fig.4.18 shows the trajectories of the master manipulator for two joints. And the final trajectories of the master hardware come to a real value of 0.3203 rad for link 1 and 0.4717 rad for link 2.

Fig.4.19 shows the tracking error $\mathbf{e}_{pm}$ between the master and the desired trajectories. As in the figures, the tracking errors converge to zero in the end within a few seconds. For link 1, the tracking error converge to zero at the point of 3.088 seconds; for link 2, the tracking error converge to zero at the point of 4.475 seconds which can verify the effectiveness of the controller designed in the previous sections.

Fig.4.20 shows the trajectories of the slave manipulator for link 1 and link 2 respectively. The final value of the slave positions converge to 0.3204 rad for link 1 and 0.4689 rad for link 2).

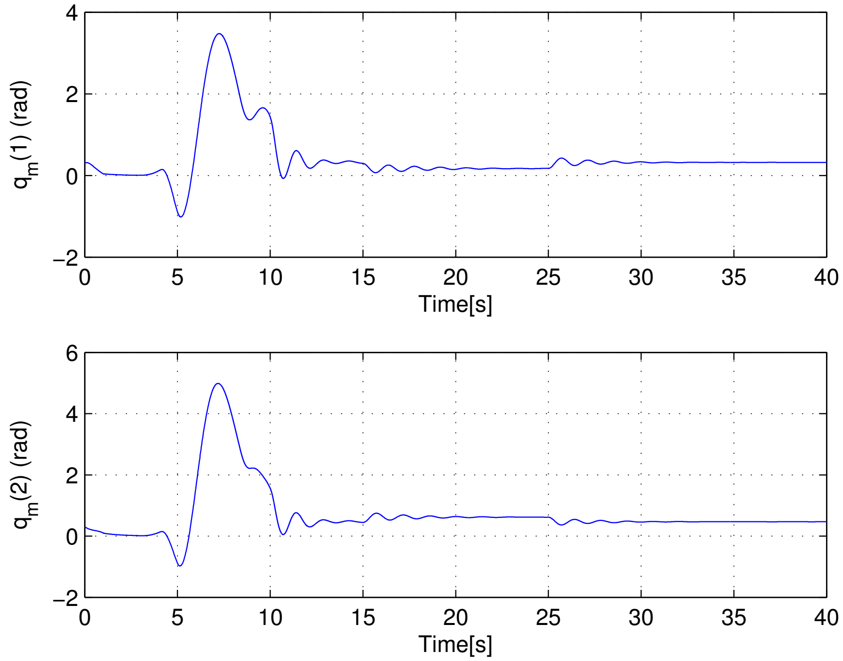

Figure 4.18: Master trajectory $\mathbf{q}_m$

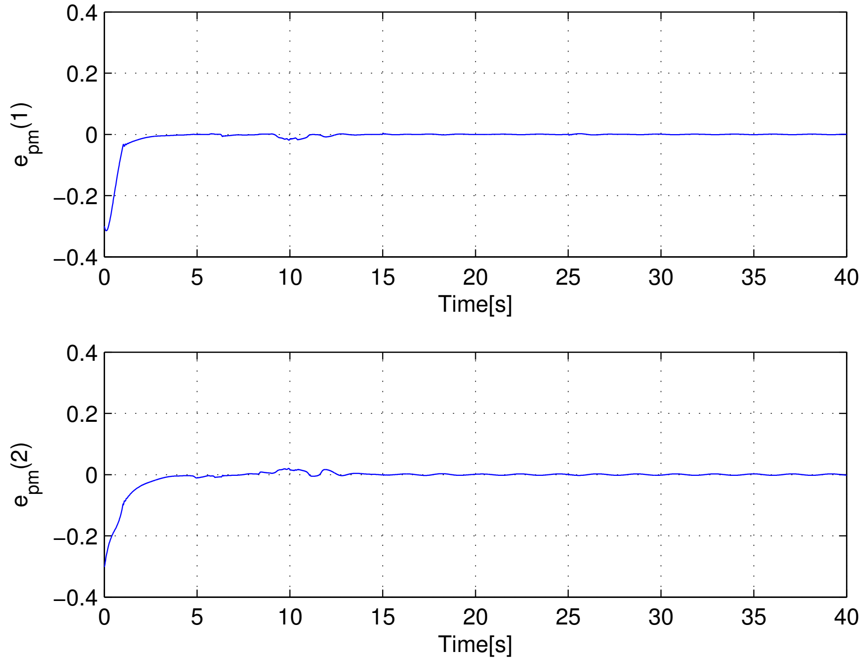

Figure 4.19: Tracking error $\mathbf{e}_{pm} = \mathbf{q}_m(t) - \mathbf{q}_d(t)$

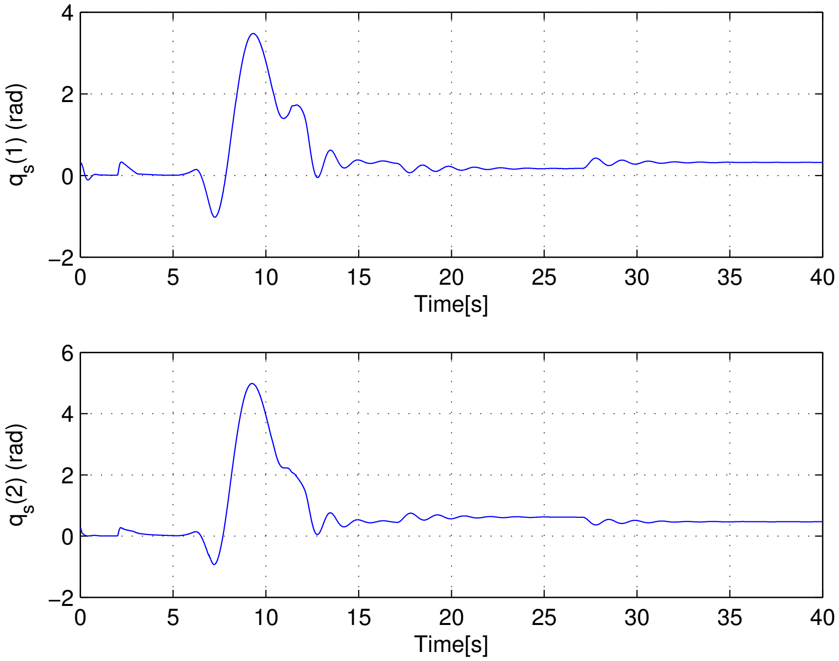

Figure 4.20: Slave trajectory $\mathbf{q}_s$

The tracking errors between the master and the slave manipulators for the two joints are shown in Fig.4.21. It converges to zero within a few seconds which verifies the robust stability and the effectiveness of the control method as well, even under the communication delay as large as up to 2 seconds. The master and the slave control input torques $\boldsymbol{\tau}_m$ and $\boldsymbol{\tau}_s$ are shown in Fig.4.22 and Fig.4.25 respectively.

Fig.4.28 shows the real value of the environmental torque that feed back to the slave manipulator. For joint 1, the value converge to a steady state of 30.17 $N \cdot m$. For joint 2, the value converge to 45.27 $N \cdot m$. Fig.4.15 shows the estimation of the environmental torque. It converges to 30.09 $N \cdot m$ for link 1 and 45.2 $N \cdot m$ for link 2. The values of the estimated environmental torque and the real environmental torque are almost same. Therefore, the environmental torques are effectively estimated.

The estimated environmental parameters $\hat{\theta}_{ext}$ are shown in Fig.4.31. The parameters converge to their true values B_{ext} for 4.5, K_{ext} for 100 and C_{ext} for -0.2 respectively.

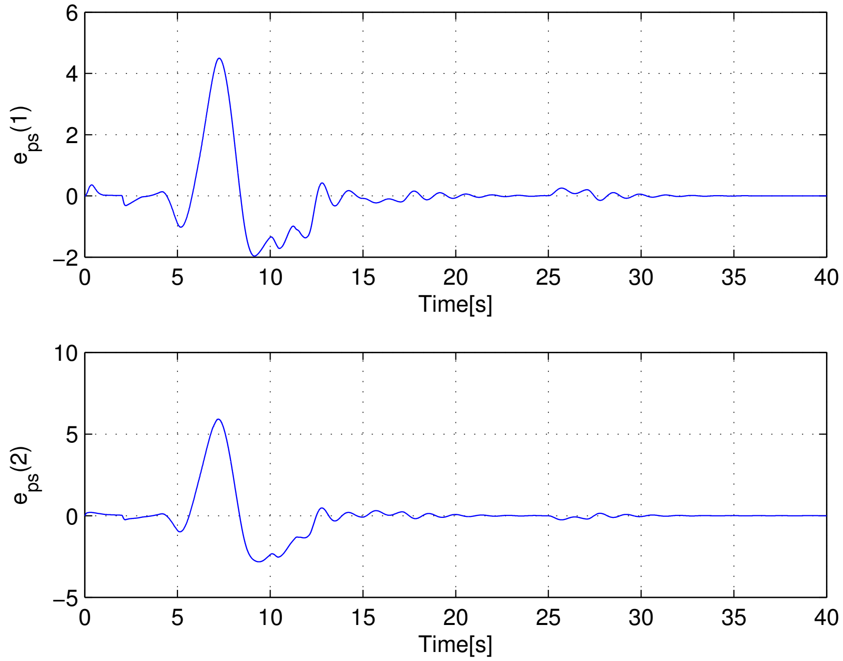

Figure 4.21: Tracking error $\mathbf{e}_{pm} = \mathbf{q}_s(t) - \mathbf{q}_m(t - T_1(t)))$

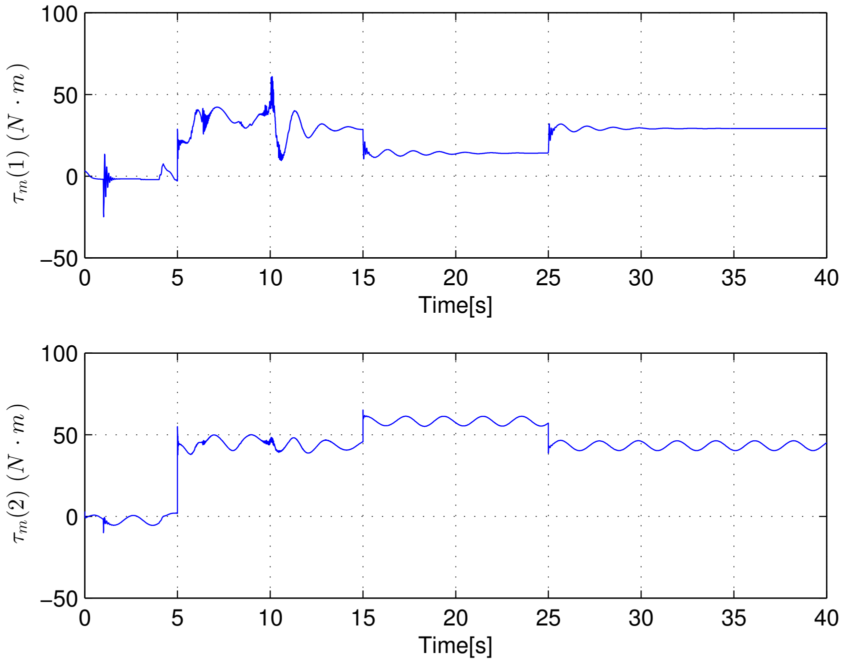

Figure 4.22: Master control input torque $\boldsymbol{\tau}_m$

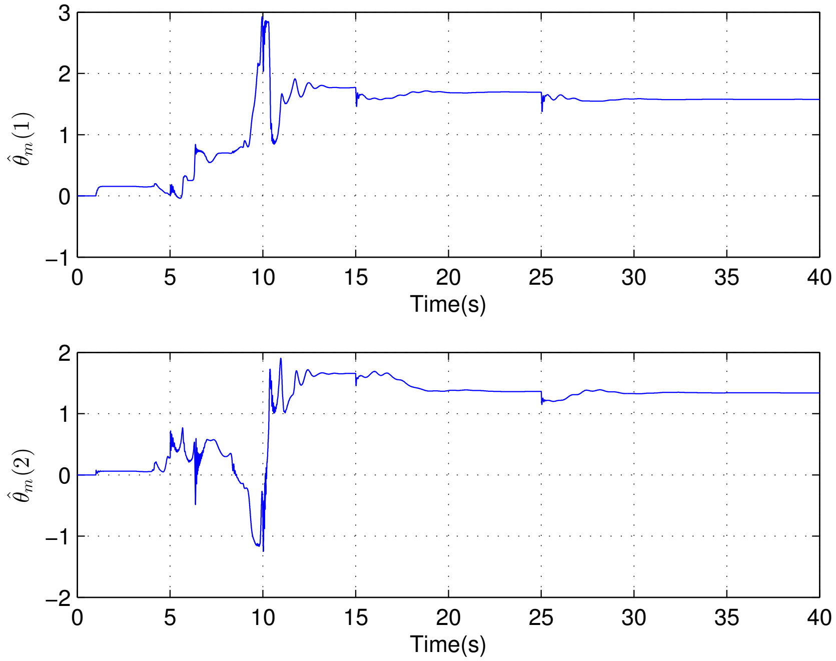

Figure 4.23: Estimated parameters $\hat{\theta}_m(1)$ and $\hat{\theta}_m(2)$

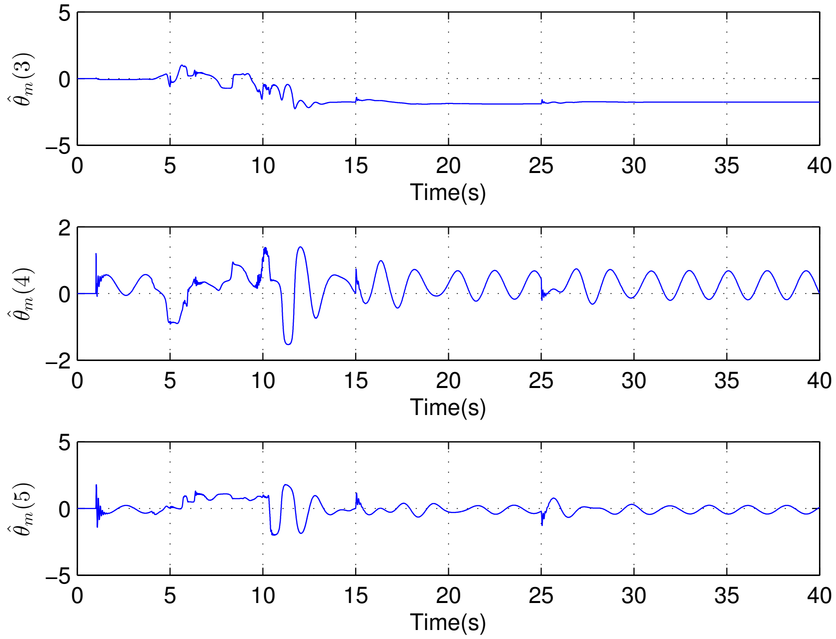

Figure 4.24: Estimated parameters $\hat{\theta}_m(3)$,$\hat{\theta}_m(4)$ and $\hat{\theta}_m(5)$

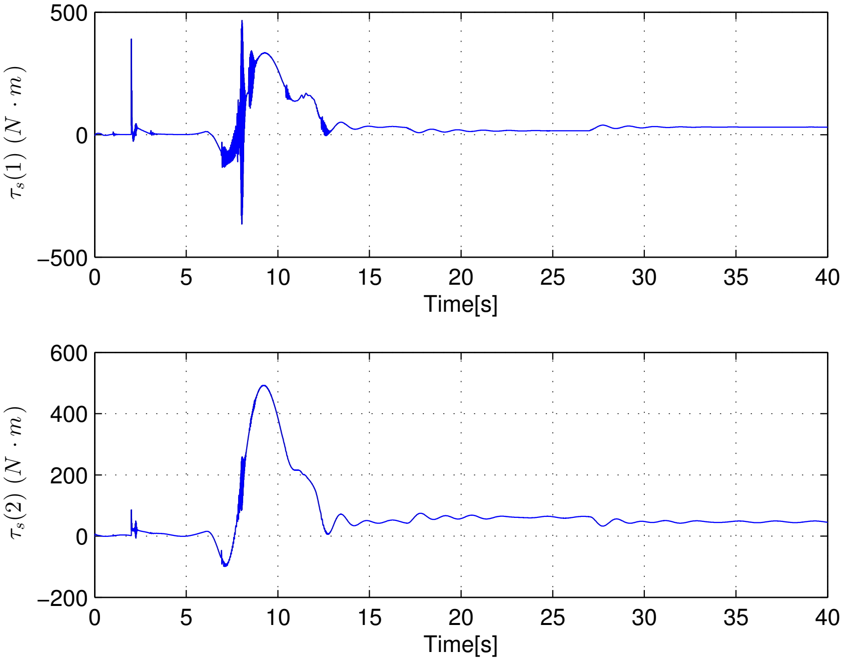

Figure 4.25: Slave control input torque $\boldsymbol{\tau}_s$

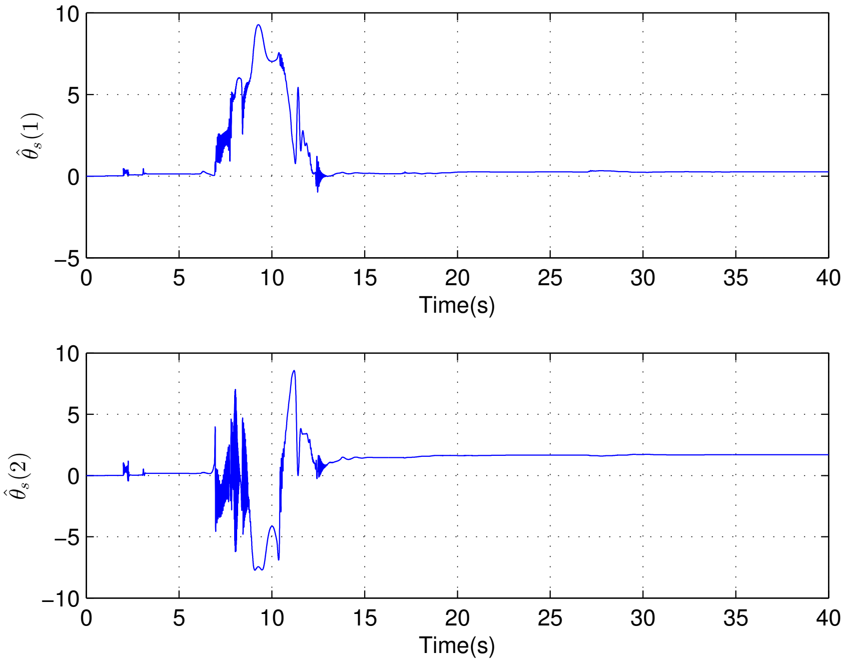

Figure 4.26: Estimated parameters $\hat{\theta}_s(1)$ and $\hat{\theta}_s(2)$

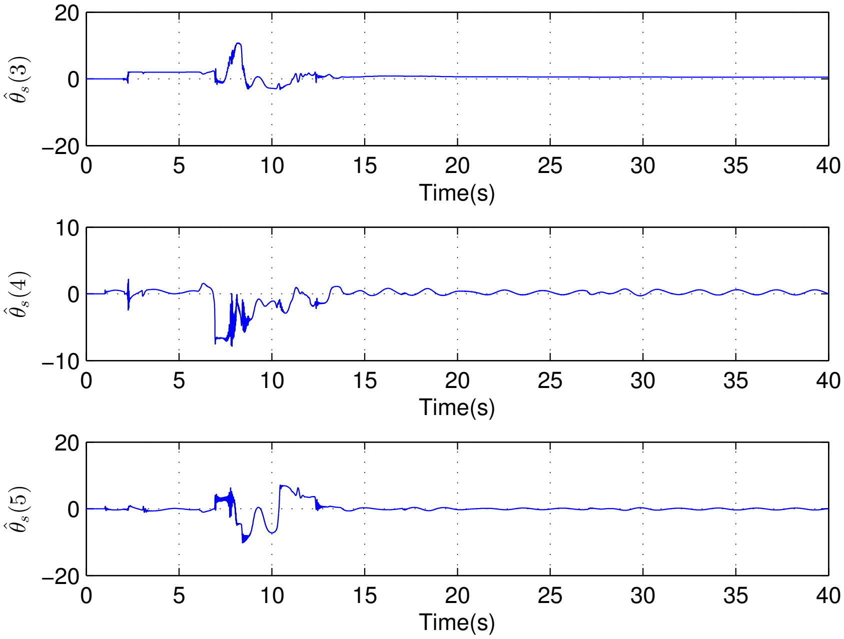

Figure 4.27: Estimated parameters $\hat{\theta}_s(3)$, $\hat{\theta}_s(4)$ and $\hat{\theta}_s(5)$

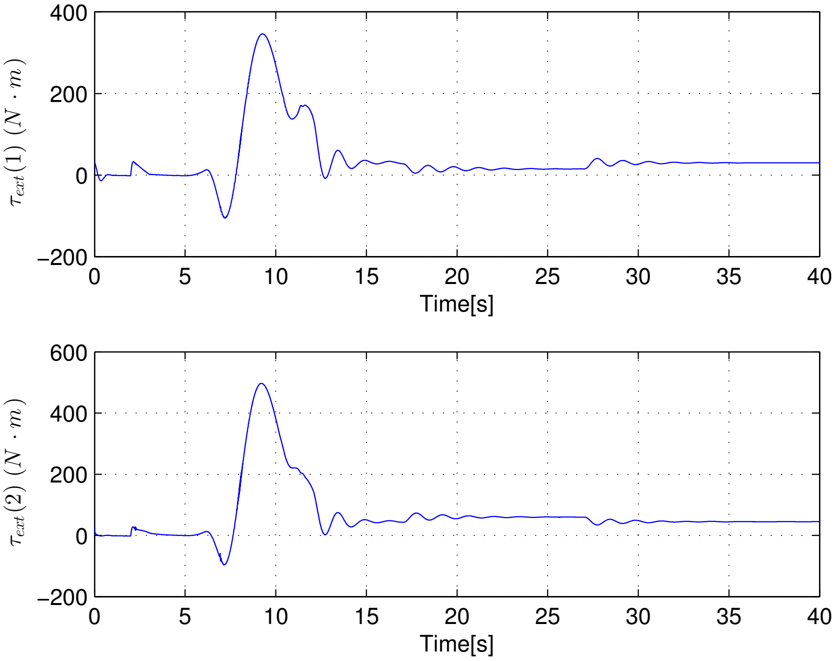

Figure 4.28: Actual environmental torque $\boldsymbol{\tau}_{ext}$

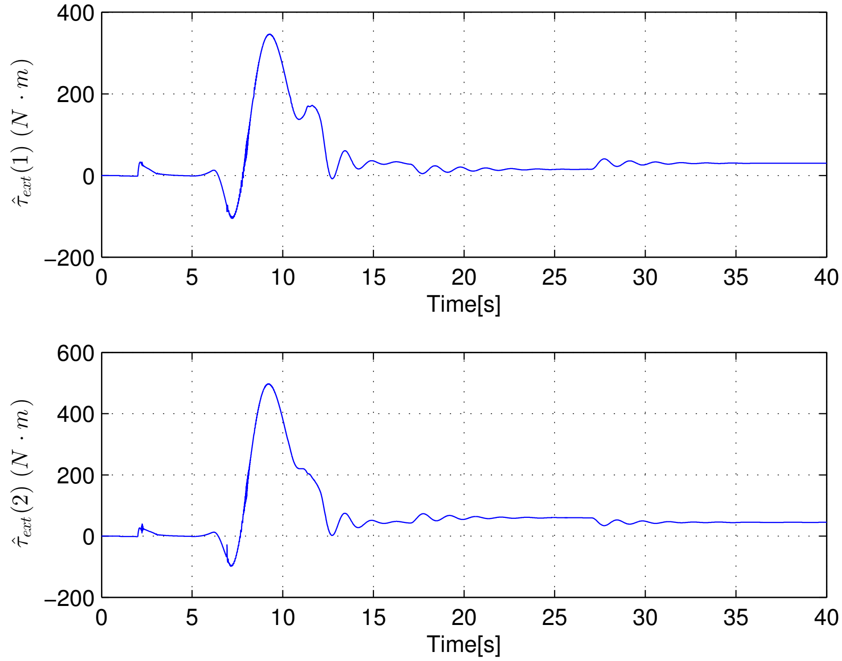

Figure 4.29: Environmental torque estimation $\hat{\tau}_{ext}$

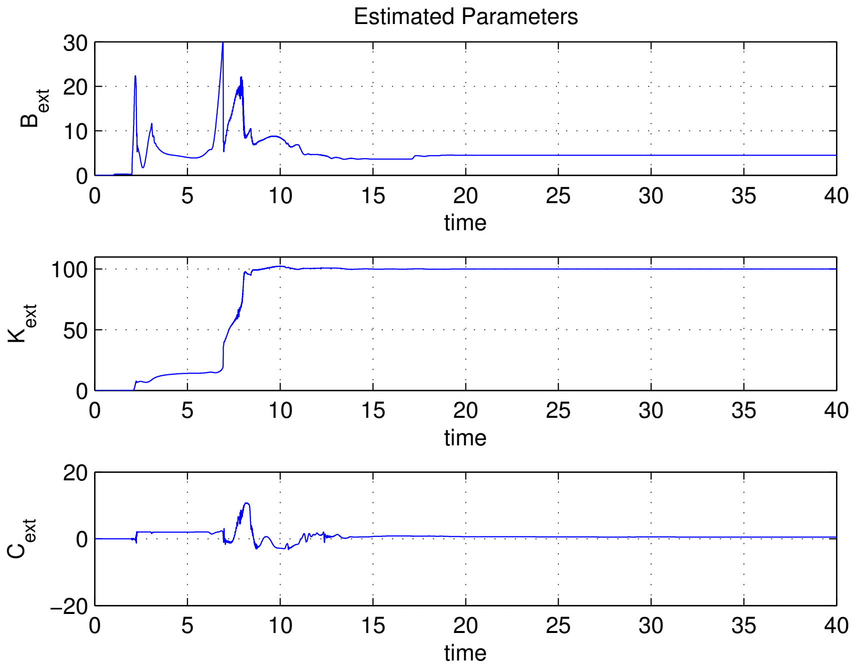

Figure 4.30: Environmental estimated parameters $\boldsymbol{\theta}_{ext}$

Chapter 5

Simulation Studies

In this chapter, there are two further studies for the designed system in simulation. The first section is the control gain tuning part. In this section, the study will be presented to see how the adaptive gain Γ_i, proportional gain K_{i1}, K_{i2} and the robust gain $K_{\Delta i}$, $i \in \{m, s\}$ affect the tracking error $\mathbf{e}_{pi}$, $i \in \{m, s\}$ respectively.

The second section is the model mismatch section. This section will further test the performance of the controller designed in the previous section. The parameters in controller will have unmatched values compared with the real parameter values in the dynamic model.The value of tracking error is recorded to see how the controller performs under this circumstances.

5.1 Control Gain Tuning

There are three sets of control gains in the simulation: adaptive gain Γ_i in equation (5.1); proportional gain K_{i1} and K_{i2} in equation (5.2) and robust gain $K_{\Delta i}$ in equation (5.3), $i \in \{m, s\}$.

$$\dot{\hat{\boldsymbol{\theta}}}_i = -\Gamma_i Y_i^T \boldsymbol{\delta}_i \tag{5.1}$$

where Γ_i, $i \in \{m, s\}$ is the adaptive gain.

$$\boldsymbol{\delta}_m = \dot{\mathbf{e}}_{pm} + K_{m1}\mathbf{e}_{pm}, \tag{5.2}$$

where K_{i1}, $i \in \{m, s\}$ is the proportional gain.

$$\begin{aligned}
\boldsymbol{\tau}_i &= \boldsymbol{\tau}_{i1} + \boldsymbol{\tau}_{i2}, \\
\boldsymbol{\tau}_{i1} &= -\hat{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_{io} - \hat{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_{io} - \hat{G}_i(\mathbf{q}_i) + \boldsymbol{\tau}_{hum} \\
\boldsymbol{\tau}_{i2} &= K_{i2}\boldsymbol{\delta}_i + K_{\Delta i} sgn(\boldsymbol{\delta}_i),
\end{aligned} \tag{5.3}$$

where $K_{\Delta i}$ is the robust gain and K_{i2} is the proportional gain, $i \in \{m, s\}$.

It is very important to choose appropriate values for different gains. The selection of the gain values can significantly affect the results of the tracking performance.

5.1.1 Selection of the adaptive gain Γ_i

In this section, the simulation plan for the effect of the adaptive gain on the tracking performance is presented.

The goal is to see how the adaptive gain Γ_i affects the value of the tracking error e_{pi}, $i \in \{m, s\}$. Taking sample points of adaptive gain Γ_m and Γ_s between the values of 0 to 200. For each value of Γ, there will be a different curve of tracking error. The following three sets of Γ_i values are chosen.

$$\begin{aligned} &Set\ 1 : \Gamma_i = 0.5; \\ &Set\ 2 : \Gamma_i = 2; \\ &Set\ 3 : \Gamma_i = 200, \qquad i \in \{m, s\}. \end{aligned} \tag{5.4}$$

Proportional gains K_{i1}, K_{i2} and the robust gain $K_{\Delta i}$, $i \in \{m, s\}$ are set to be the following values respectively.

$$K_{m1} = K_{s1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, K_{m2} = K_{s2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{5.5}$$

$$K_{\Delta m} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, K_{\Delta s} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}. \tag{5.6}$$

The following plots are the output results for different values of the adaptive gains Γ_m and Γ_s.

As shown in Fig.5.1, when the value of the adaptive gain Γ is equal to 0.5, the tracking error converges to a bound (from 1.8×10^{-7} to 0.001238). In Fig.5.3, when the adaptive gain Γ is equal to 2, the tracking error converges to a bound (from 1.044×10^{-8} to 1.985×10^{-5}). In Fig.5.5, when the adaptive gain Γ is set to the value of 200, the tracking error converge to a bound (from 1.927×10^{-14} to 1.298×10^{-11}) . It is can be concluded that when the value of adaptive gain increases, the tracking error decreases and converges to zero at the end, so that the performance of the controller becomes more effective.

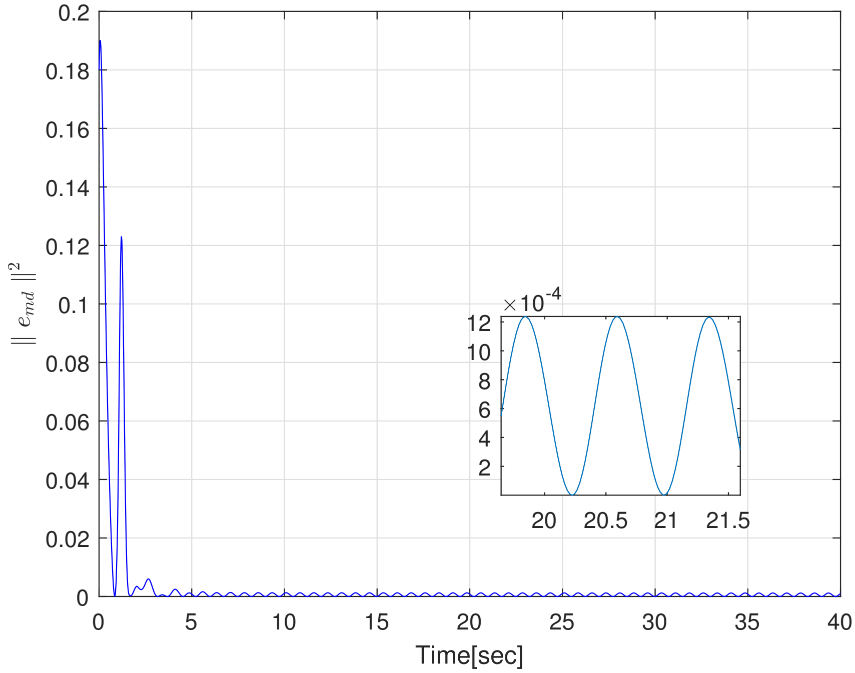

Figure 5.1: Tracking error between master and desire trajectory with $\Gamma_i = 0.5I_{2\times 2}$, $i \in \{m, s\}$

The y label of the figure is $\| \mathbf{e} \|^2$, where the expression of $\| \mathbf{e} \|^2$ is as follows.

$$\| \mathbf{e} \|^2 = \mathbf{e}^2_{joint1} + \mathbf{e}^2_{joint2} \ , \tag{5.7}$$

where e_{joint1} stands for the tracking error of the joint 1, e_{joint2} stands for the tracking error of joint 2.

Fig.5.2 shows the plot of tracking error between master and slave hardware when the value of adaptive gain equals to 0.5. The tracking error converges to a bound (from 5.215×10^{-5} to 0.0042). Fig 5.5 shows that when the adaptive gain Γ is set to be 2, the tracking error converges to a bound (from 1.26×10^{-5} to 7.45×10^{-4}). Fig. 5.6 shows that when adaptive gain Γ is equal to 200, the tracking error converge to a bound (from 8.895×10^{-5} to 1.938×10^{-4}).

As shown in the plots, the range of tracking error decreases when adaptive gain increases, verifying the effectiveness of the adaptive controller design.

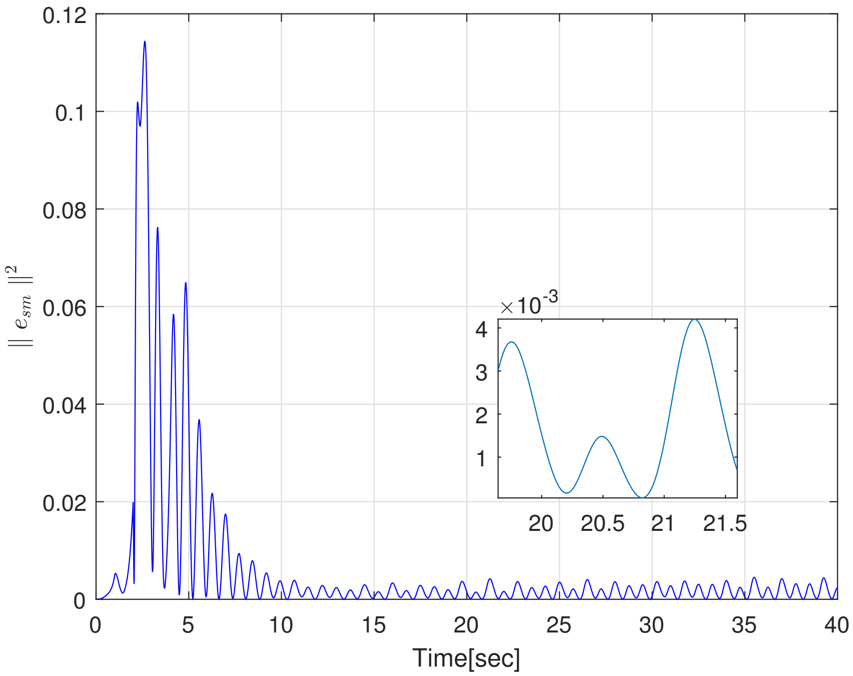

Figure 5.2: Tracking error between the master and slave trajectory with $\Gamma_i = 0.5I_{2\times 2}$, $i \in \{m, s\}$

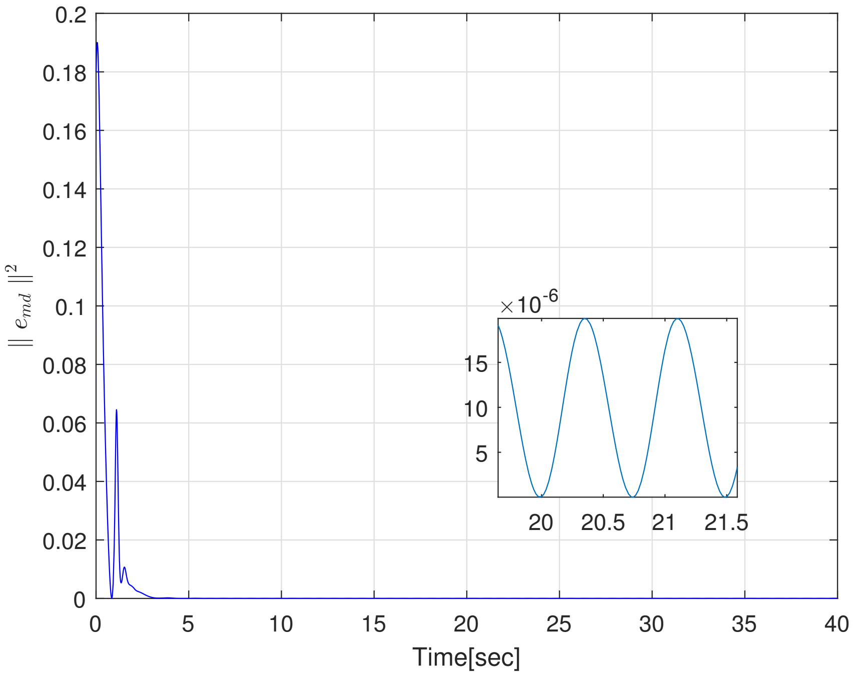

Figure 5.3: Tracking error between the master and desire trajectory with $\Gamma_i = 2I_{2\times 2}$, $i \in \{m, s\}$

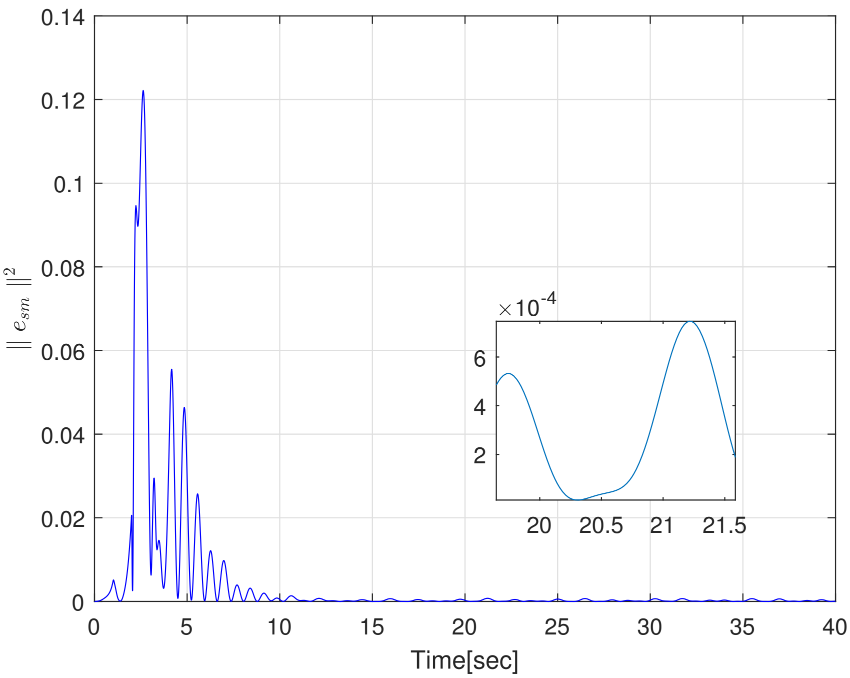

Figure 5.4: Tracking error between the master and slave trajectory with $\Gamma_i = 2I_{2\times2}$, $i \in \{m, s\}$

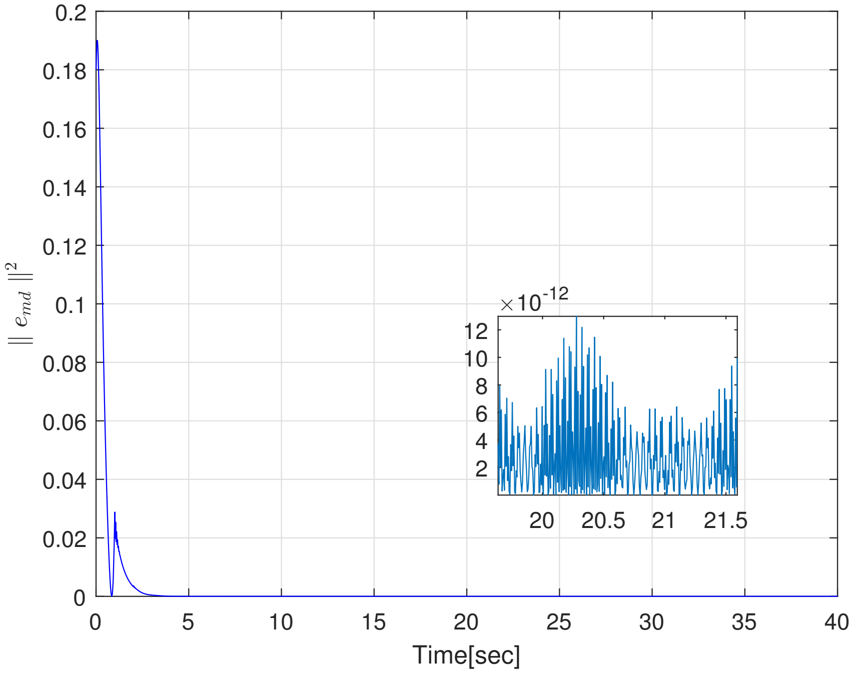

Figure 5.5: Tracking error between the master and desire trajectory with $\Gamma_i = 200I_{2\times2}$, $i \in \{m, s\}$

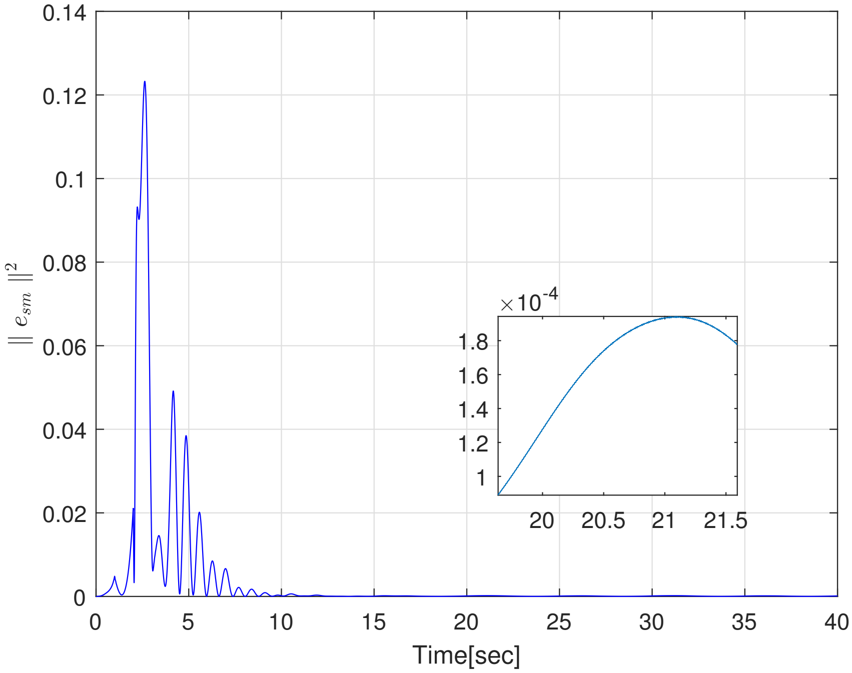

Figure 5.6: Tracking error between the master and desire trajectory with $\Gamma_i = 200I_{2\times2}$, $i \in \{m, s\}$

The results are concluded in the following table,

Example	Value of Γ_i $i \in \{m, s\}$	Value of $\parallel e \parallel^2$	Converges Time(seconds)
1	0.5	Converge to a bound from 5.215×10^{-5} to 0.0042	4.45
2	2	Converge to a bound from $1.26 \times 10^{-5}$to 7.45×10^{-4}	4.16
3	200	Converge to a bound from 8.895×10^{-5} to 1.938×10^{-4}	3.95

5.1.2 Selection of proportional gains K_{i1} and K_{i2}

In this section, the analysis of the effect of the proportional gain on the tracking performance is presented.

Using the master side as an example, the values of K_{m1} and K_{m2}, can be seen in the following equations,

$$\boldsymbol{\delta}_m = \dot{\mathbf{e}}_{pm} + K_{m1}\mathbf{e}_{pm} \ , \tag{5.8}$$

where K_{m1} is the proportional gain.

$$\begin{aligned}
\boldsymbol{\tau}_m &= \boldsymbol{\tau}_{m1} + \boldsymbol{\tau}_{m2}, \\
\boldsymbol{\tau}_{m1} &= -\hat{M}_m(\mathbf{q}_m)\ddot{\mathbf{q}}_{mo} - \hat{C}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)\dot{\mathbf{q}}_{mo} - \hat{G}_m(\mathbf{q}_m) + \boldsymbol{\tau}_{hum} \\
\boldsymbol{\tau}_{m2} &= K_{m2}\boldsymbol{\delta}_m + K_{\Delta m} sgn(\boldsymbol{\delta}_m),
\end{aligned} \tag{5.9}$$

where $K_{\Delta m}$ is the robust gain and K_{m2} is the proportional gain.

where $\mathbf{e}_{pm}$ is the tracking error between the master and the desire trajectory. The value of K_{m1} affects the proportional ratio between the position error and the velocity error. The value of K_{m2} denotes the proportional ratio of the switching-function-like quantity $\boldsymbol{\delta}_m$.

In the simulation plan, samples of the proportional gain K_{m1} and K_{m2} are selected between 0 and 2. For each value of K_{m1} and K_{m2}, there will be a different curves of the tracking error.

The following five sets of values are being chosen for K_{i1} and K_{i2}, $i \in \{m, s\}$.

$$\begin{aligned}
Set\ 1 : & K_{m1} = K_{s1} = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}, K_{m2} = K_{s2} = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}; \\
Set\ 2 : & K_{m1} = K_{s1} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, K_{m2} = K_{s2} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}; \\
Set\ 3 : & K_{m1} = K_{s1} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}, K_{m2} = K_{s2} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}; \\
Set\ 4 : & K_{m1} = K_{s1} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}, K_{m2} = K_{s2} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}; \\
Set\ 5 : & K_{m1} = K_{s1} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, K_{m2} = K_{s2} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}; \qquad (5.10)
\end{aligned}$$

The value of the adaptive gain Γ_i and the value of robust gain K_{Δ_i}, $i \in \{m, s\}$ are set to be the following values.

$$\begin{aligned}
\Gamma_m = \Gamma_s = 40 \ , \\
K_{\Delta m} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, K_{\Delta s} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \ . \qquad (5.11)
\end{aligned}$$

Fig.5.7 shows the plot of tracking error between the master and desire hardware when the value of proportional gains K_{i1}, K_{i2} equal to 0.02, the tracking error converges to 0.0006829 in the end. Fig.5.9 shows the tracking error converges to 4.907×10^{-7} when proportional gains K_{i1}, K_{i2} are set to be 0.1. Fig.5.11 shows the tracking error converges to 7.337×10^{-11} when proportional gains K_{i1}, K_{i2} are equal to 0.2. Fig.5.13 shows the figure of tracking error when proportional gains K_{i1}, K_{i2} are set to be 0.5, the tracking error converges to 0. Fig.5.15 shows the figure of tracking error when proportional gains K_{i1}, K_{i2} are set to be 2, the tracking error converges to 1.131×10^{-11}.

As shown in the figures, when proportional gain becomes bigger, the tracking error is smaller and the tracking performance improves.

In Fig.5.10, when the value of proportional gains K_{i1}, K_{i2} are equal to 0.5, the tracking error between the master and slave trajectory is obviously very large and converges to 0.01128 in the end. In Fig.5.12, Fig.5.14, Fig.5.16 and Fig.5.18, the tracking errors converge to a constant value around 0.0001114. Therefore, It can be concluded that when the value of proportional gain increases, the value of the tracking error decreases, so that the performance of the controller become more effective.

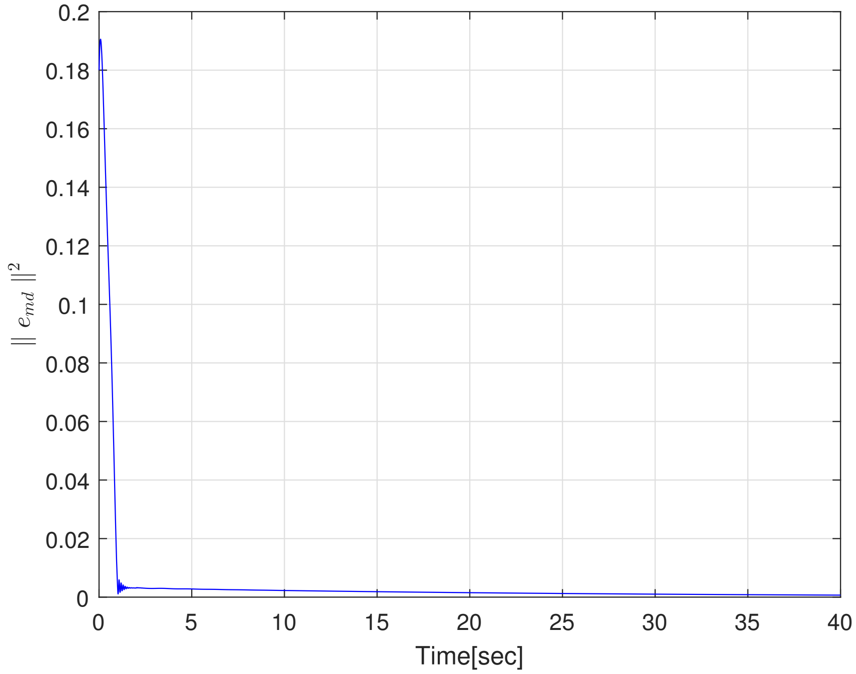

Figure 5.7: Tracking error between the master and desire trajectory with $K_{i1} = K_{i2} = [0.02, 0; 0, 0.02]$, $i \in \{m, s\}$

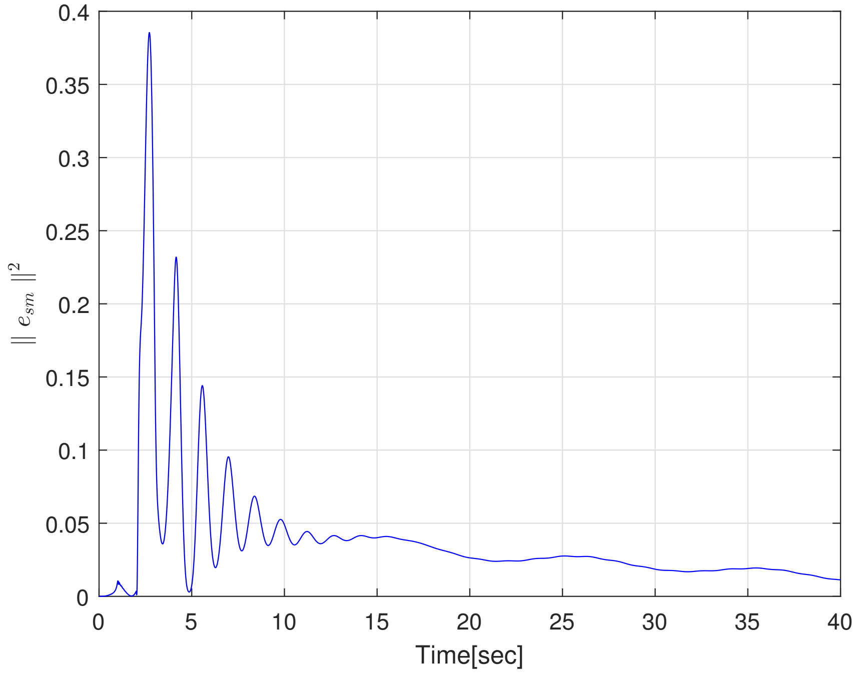

Figure 5.8: Tracking error between the master and slave trajectory with $K_{i1} = K_{i2} = [0.02, 0; 0, 0.02]$, $i \in \{m, s\}$

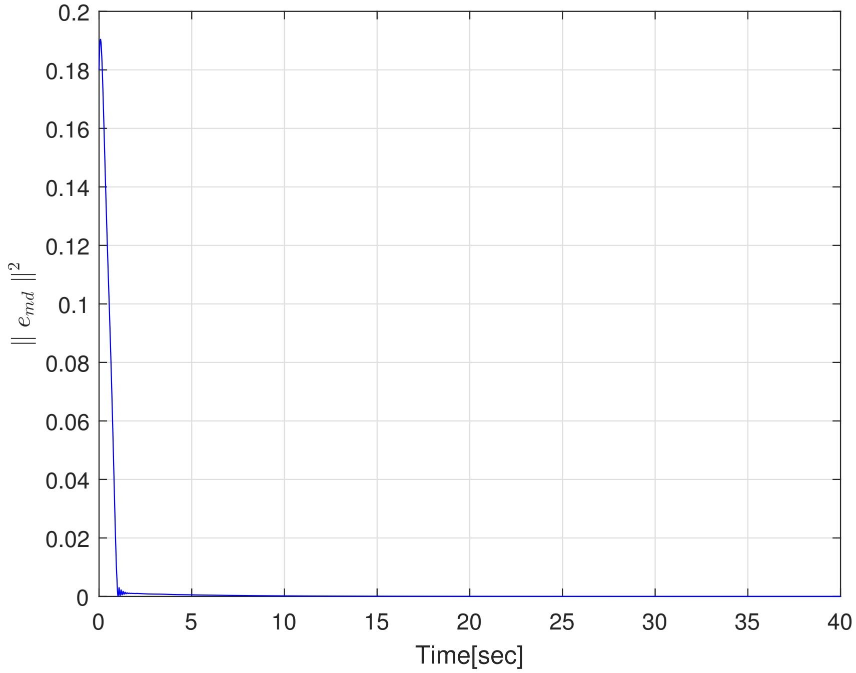

Figure 5.9: Tracking error between the master and desire trajectory with $K_{i1} = K_{i2} = [0.1, 0; 0, 0.1]$, $i \in \{m, s\}$

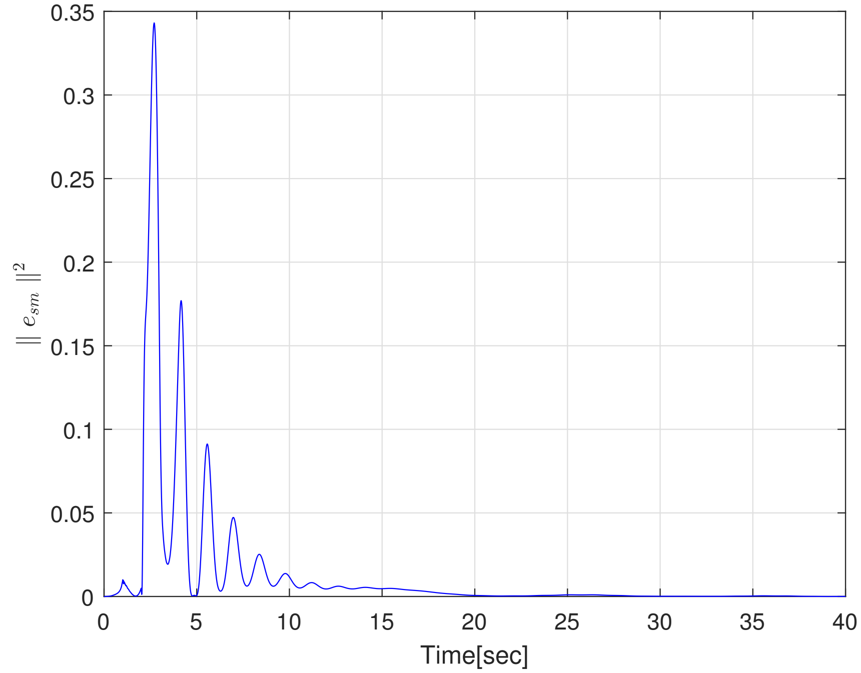

Figure 5.10: Tracking error between the master and slave trajectory with $K_{i1} = K_{i2} = [0.1, 0; 0, 0.1]$, $i \in \{m, s\}$

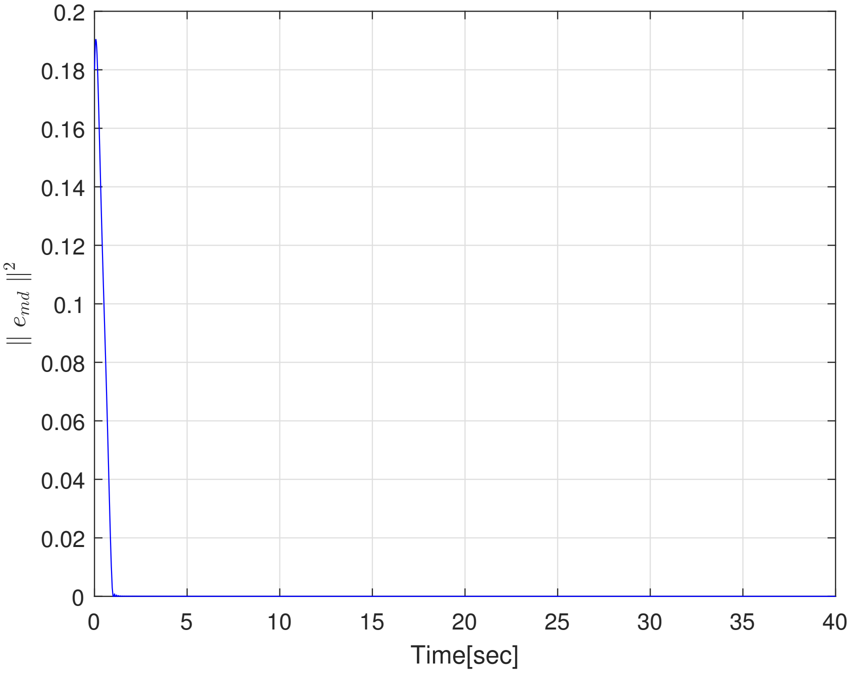

Figure 5.11: Tracking error between the master and desire trajectory with $K_{i1} = K_{i2} = [0.2, 0; 0, 0.2]$, $i \in \{m, s\}$

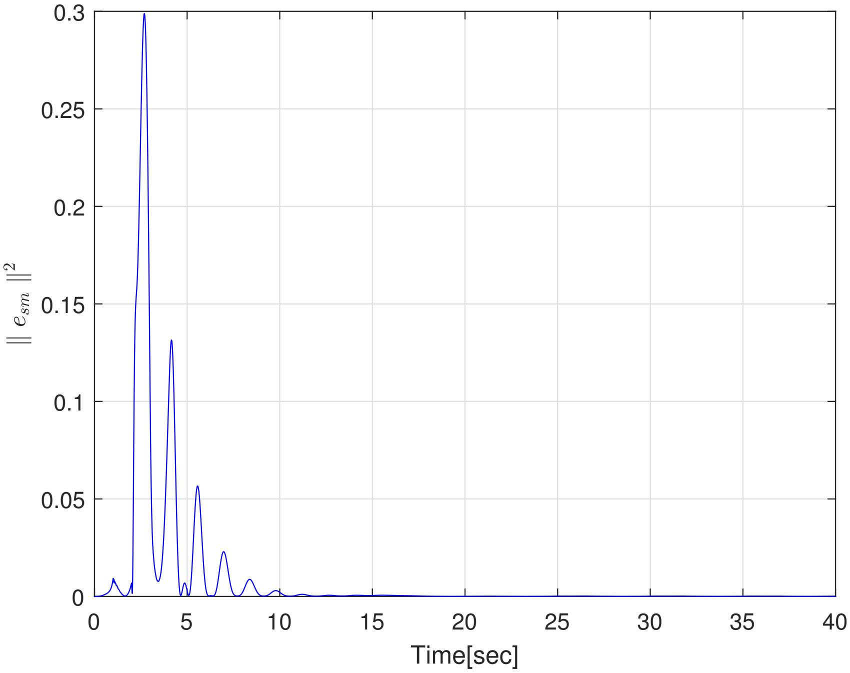

Figure 5.12: Tracking error between the master and slave trajectory with $K_{i1} = K_{i2} = [0.2, 0; 0, 0.2]$, $i \in \{m, s\}$

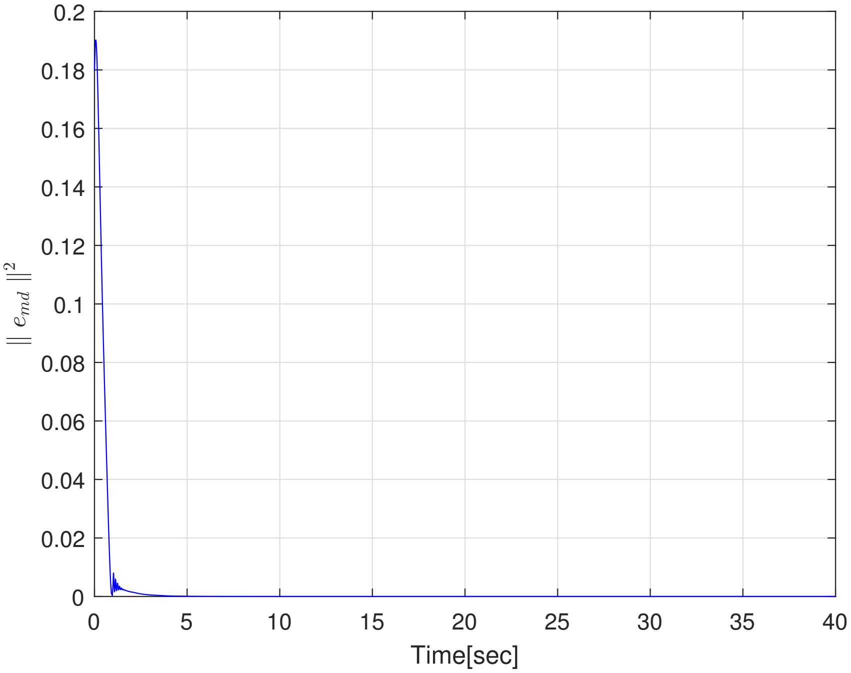

Figure 5.13: Tracking error between the master and desire trajectory with $K_{i1} = K_{i2} = [0.5, 0; 0, 0.5]$, $i \in \{m, s\}$

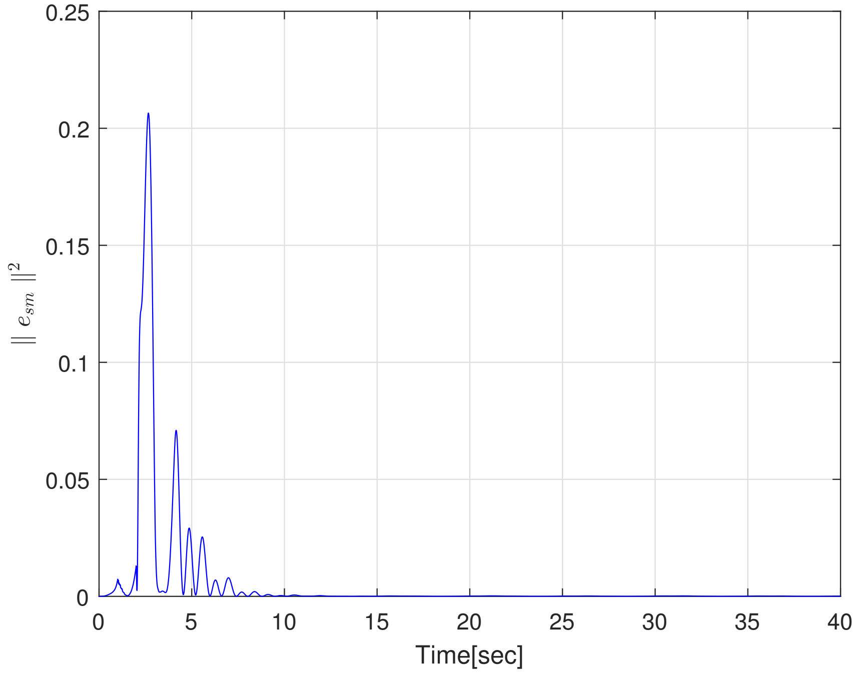

Figure 5.14: Tracking error between the master and slave trajectory with $K_{i1} = K_{i2} = [0.5, 0; 0, 0.5]$, $i \in \{m, s\}$

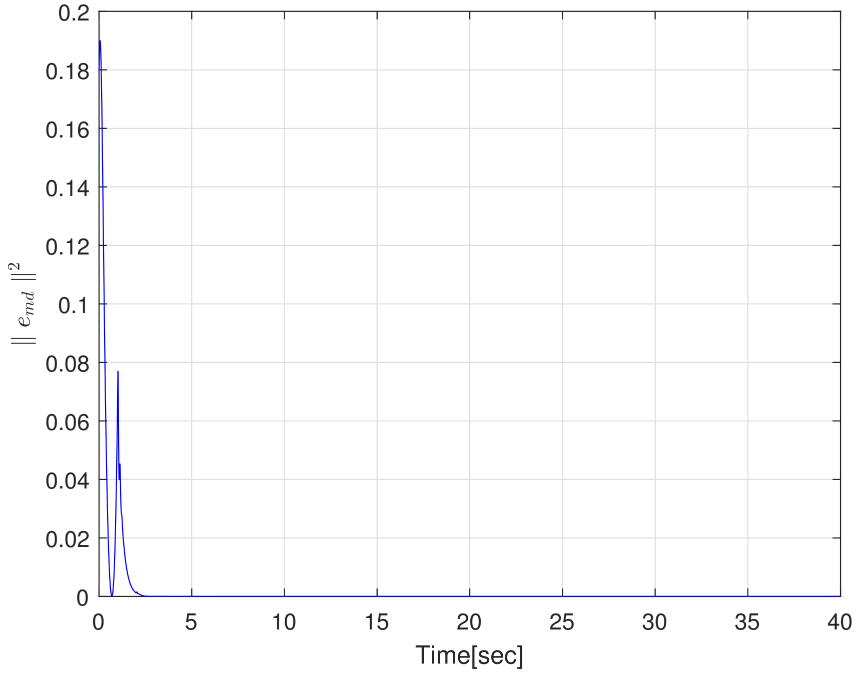

Figure 5.15: Tracking error between the master and desire trajectory with $K_{i1} = K_{i2} = [2, 0; 0, 2]$, $i \in \{m, s\}$

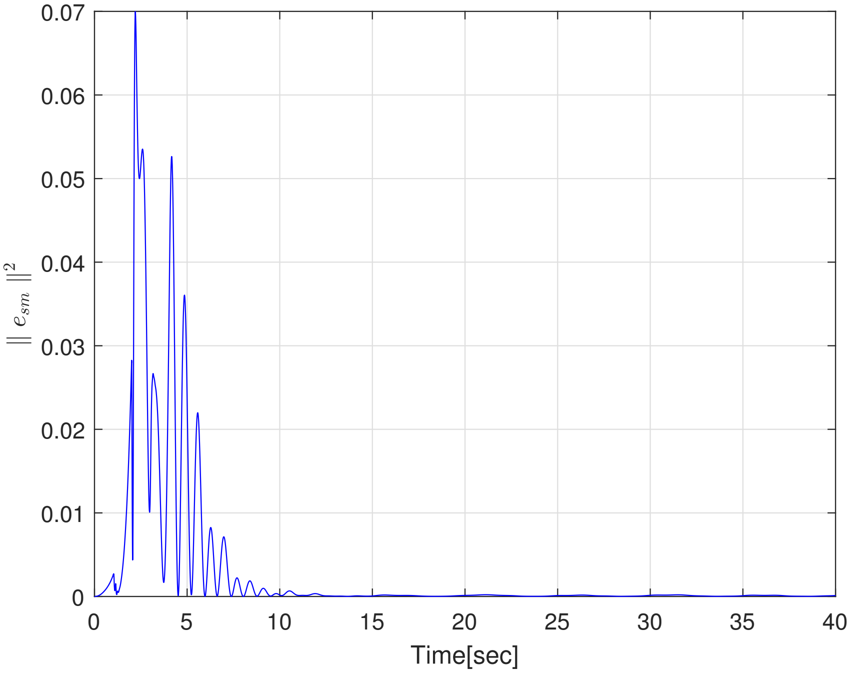

Figure 5.16: Tracking error between the master and slave trajectory with $K_{i1} = K_{i2} = [2, 0; 0, 2]$, $i \in \{m, s\}$

The results are concluded in the following table,

Example	Value of K_{i1} and K_{i2} $i \in \{m, s\}$	Value of $\| e \|^2$	Converges Time(seconds)
1	$\begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}$	Converges to 0.0006829 in the end	2.46
2	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$	Converges to 4.907×10^{-7}	2.16
3	$\begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$	Converges to 7.337×10^{-11}	2.10
4	$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$	Converges to 3.751×10^{-11}	2.08
5	$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$	Converges to 1.131×10^{-11}	2.05

5.1.3 Selection of the robust gain $K_{\Delta i}$

In this section, the effect of the robust gain $K_{\Delta i}$ to the tracking performance of the controller is analyzed.

The robust term $K_{\Delta i}$ in the controller is as following.

$$\boldsymbol{\tau}_{i2} = K_{i2}\boldsymbol{\delta}_i + K_{\boldsymbol{\Delta i}} sgn(\boldsymbol{\delta}_i) \ , \tag{5.12}$$

where $K_{\Delta i}$, $i \in \{m, s\}$ is the robust term in the controller.

The gain tuning design plan is to take samples of the robust gain between 0.2 to 50. For each different values of $K_{\Delta i}$, $i \in \{m, s\}$, there is a different curve of $\parallel \mathbf{e} \parallel^2$, for

$$\parallel \mathbf{e} \parallel^2 = \mathbf{e}^2_{joint1} + \mathbf{e}^2_{joint2} \ .$$

The following four sets of values are being selected from the results since the results plotted from those values

$$\begin{aligned}
Set\ 1 : K_{\Delta m} &= \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} K_{\Delta s} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} \ , \\
Set\ 2 : K_{\Delta m} &= \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} K_{\Delta s} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \ , \\
Set\ 3 : K_{\Delta m} &= \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} K_{\Delta s} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \ , \\
Set\ 4 : K_{\Delta m} &= \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix} K_{\Delta s} = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix} \ .
\end{aligned} \tag{5.13}$$

The adaptive gain Γ_i and proportional gains K_{i1} and K_{i2} are set to be the following values respectively.

$$K_{m1} = K_{s1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} . K_{m2} = K_{s2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.14}$$

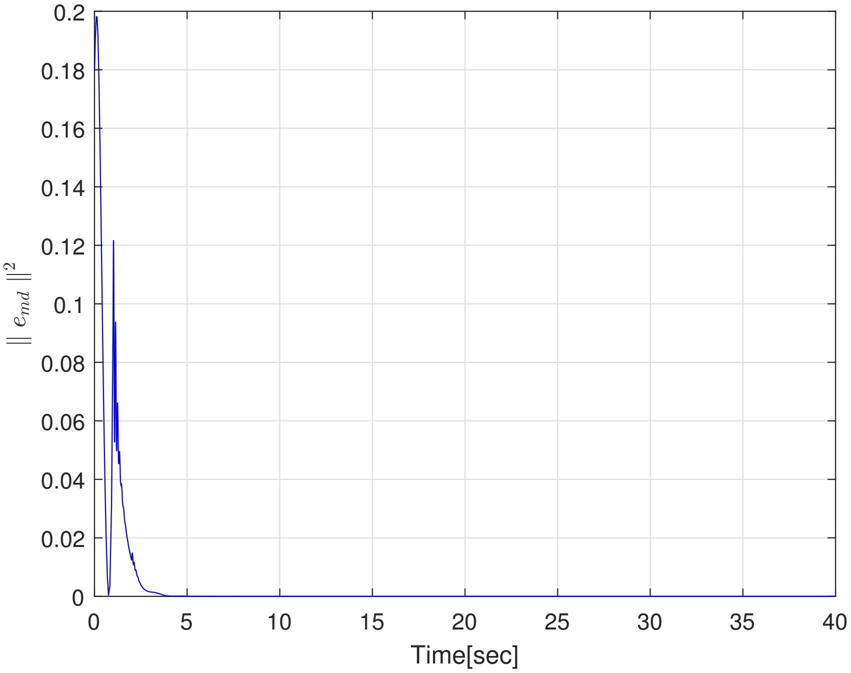

Figure 5.17: Tracking error between the master and desire trajectory with $K_{\Delta i} = 0.2$, $i \in \{m, s\}$

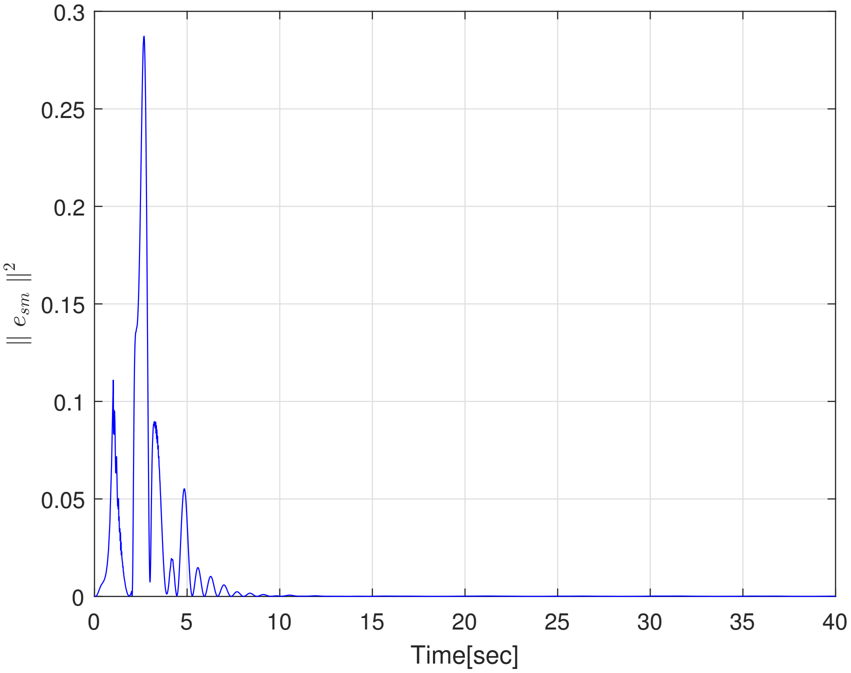

Figure 5.18: Tracking error between the master and slave trajectory with $K_{\Delta i} = 0.2$, $i \in \{m, s\}$

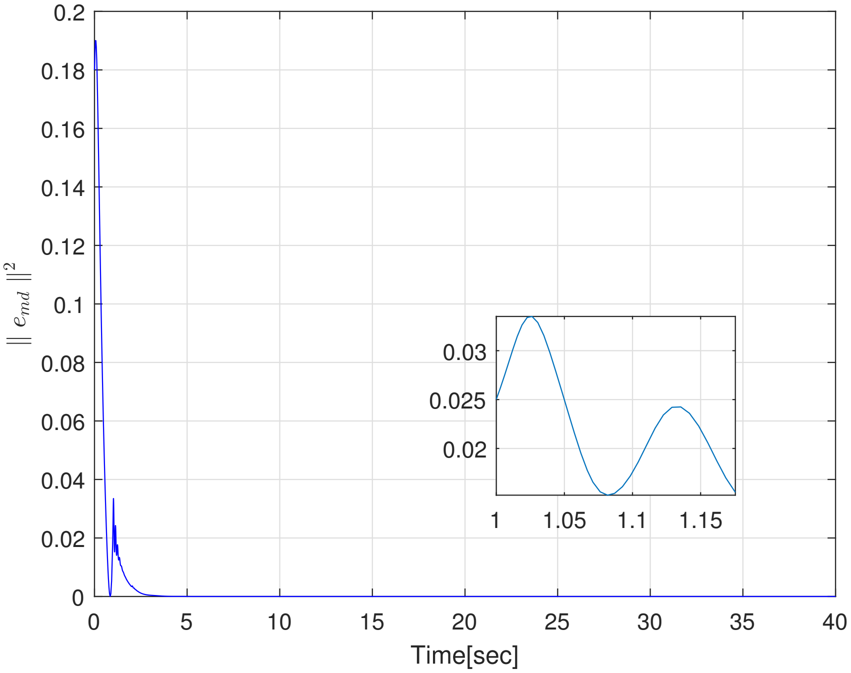

Figure 5.19: Tracking error between the master and desire trajectory with $K_{\Delta i} = 5$, $i \in \{m, s\}$

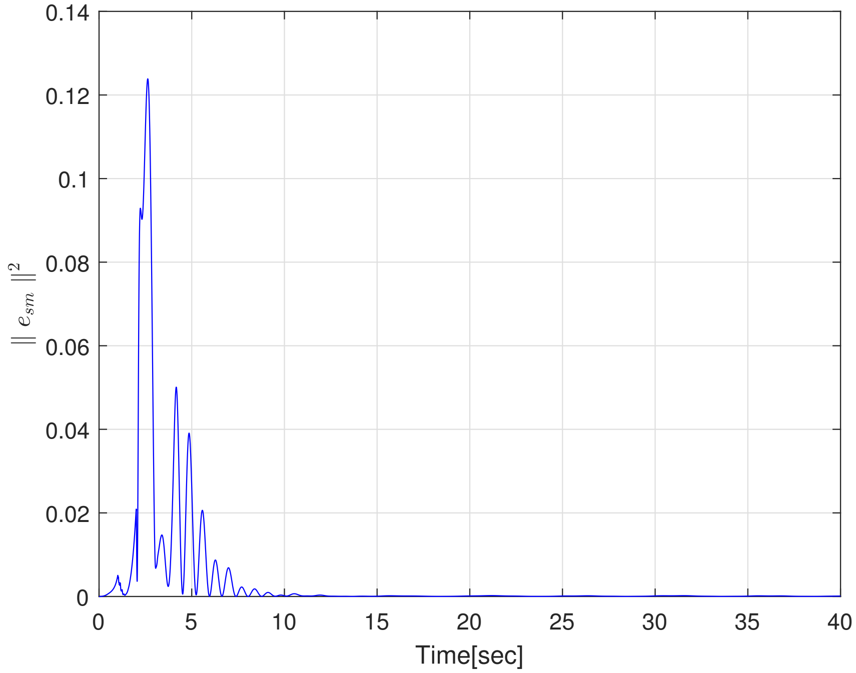

Figure 5.20: Tracking error between the master and slave trajectory with $K_{\Delta i} = 5$, $i \in \{m, s\}$

Fig. 5.17 to Fig. 5.25 show the effect of the robust gain to the tracking performance of the controller.

As it can be seen in Fig.5.17, Fig.5.19, Fig.5.21 and Fig.5.23, although the tracking error both converge to zero in the end, the oscillation at the beginning between 0 to 5 seconds is getting smaller as the robust gain become bigger. In Fig.5.25, when the robust gain $K_{\Delta s} = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}$, the effect of the perturbation is been canceled.

In Fig.5.18, Fig.5.20, Fig.5.22 and Fig.5.24, the tracking error both converges to zero in the end. The oscillation between zero to three seconds is getting smaller as the robust gain value is getting larger. Therefore, it is safe to say that the tracking performance become better when the value of the robust gain become larger and the effect of the robust gain in the system is being verified as well.

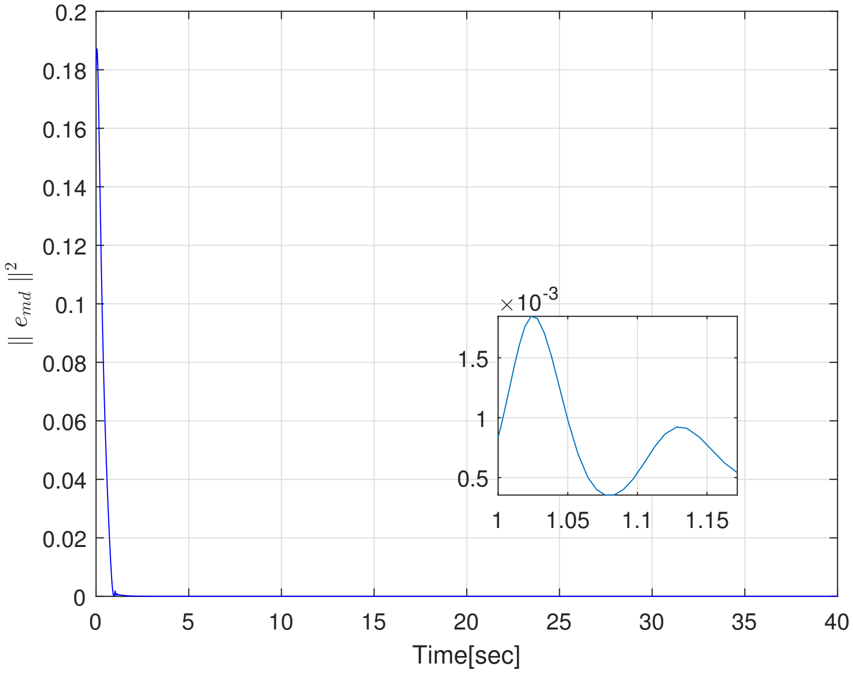

Figure 5.21: Tracking error between the master and desire trajectory with $K_{\Delta i} = 10$, $i \in \{m, s\}$

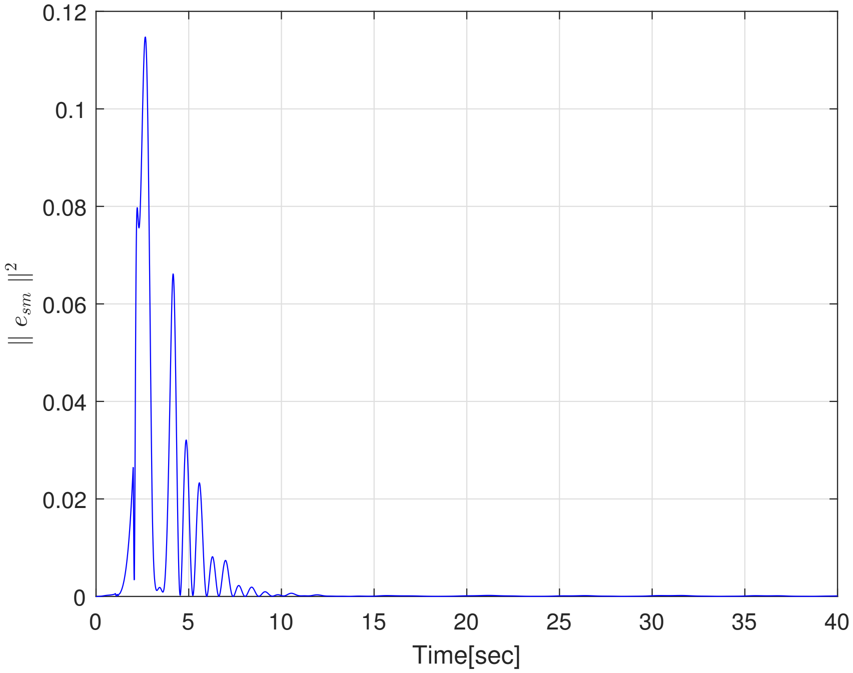

Figure 5.22: Tracking error between the master and slave trajectory with $K_{\Delta i} = 10$, $i \in \{m, s\}$

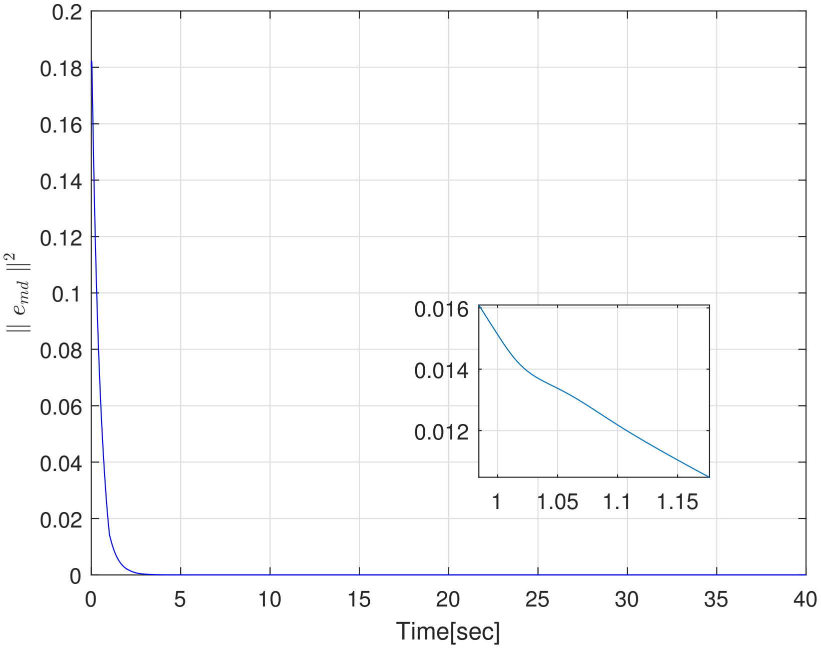

Figure 5.23: Tracking error between the master and desire trajectory with $K_{\Delta i} = 50$, $i \in \{m, s\}$

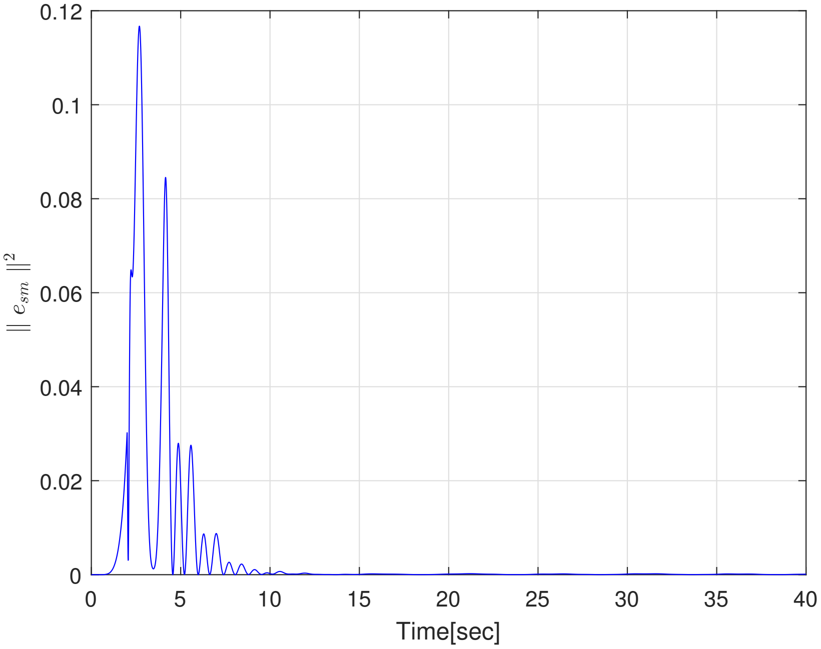

Figure 5.24: Tracking error between the master and slave trajectory with $K_{\Delta i} = 50$, $i \in \{m, s\}$

5.2 Studies on the Case with Mismatched Model

In this section, the dynamic model mismatch study will be carried out to verify the effectiveness of the controller design. The mismatch rate is 5%.

The original parameters of the dynamic model are as following,

$$\begin{aligned}
\theta_{i_1} &= l_{i_2}^2 m_{i_2} + l_{i_1}^2 (m_{i_1} + m_{i_2}) = 1.25, \\
\theta_{i_2} &= l_{i_1} l_{i_2} m_{i_2} = 0.125, \\
\theta_{i_3} &= l_{i_2}^2 m_{i_2} = 0.125, \\
\theta_{i_4} &= l_{i_2} m_{i_2} = 0.25, \\
\theta_{i_5} &= l_{i_1} (m_{i_1} + m_{i_2}) = 2.25,
\end{aligned}$$

where the values of m_{i_1}, m_{i_2}, l_{i_1} and l_{i_2}, for $i \in \{m, s\}$, are defined as follows,

$m_{m_1} = 4.0$ kg	$m_{m_2} = 0.5$ kg	$l_{m_1} = 50$ cm	$l_{m_2} = 50$ cm
$m_{s_1} = 4.0$ kg	$m_{s_2} = 0.5$kg	$l_{s_1} = 50$ cm	$l_{s_2} = 50$ cm

The values of the parameters used for the controller design is,

$\theta'_{i_1} = 1.1875$	$\theta'_{i_2} = 0.1188$	$\theta'_{i_3} = 0.1188$	$\theta'_{i_4} = 0.2375$	$\theta'_{i_5} = 2.1375$

The human input that is applying to the bilateral system is the same as in Fig.4.1. The simulation results are shown in Fig.5.25 to Fig.5.34. As shown in Fig.5.27 and Fig.5.29, tracking errors converge to zero at the end, it verifies the effectiveness of the controller design. As a result, the tracking performance of the derived controller are not affected by the unmatched dynamic model.

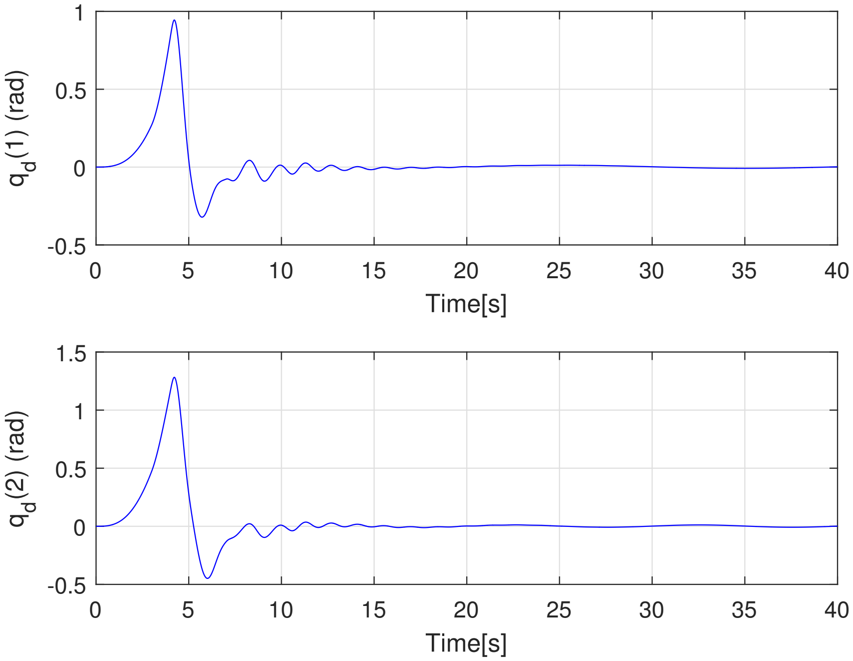

Figure 5.25: Desired trajectory $\mathbf{q}_d$

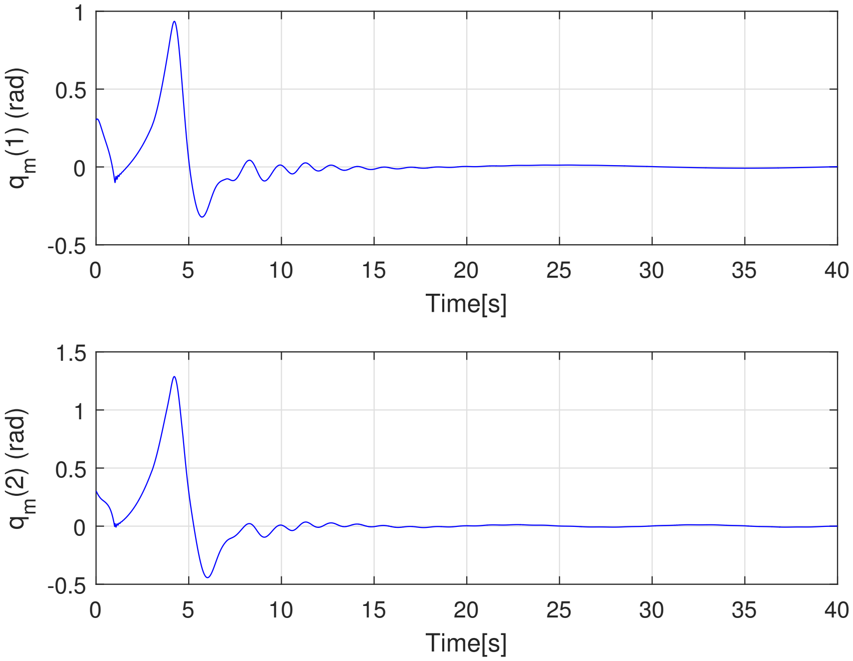

Figure 5.26: Master trajectory $\mathbf{q}_m$

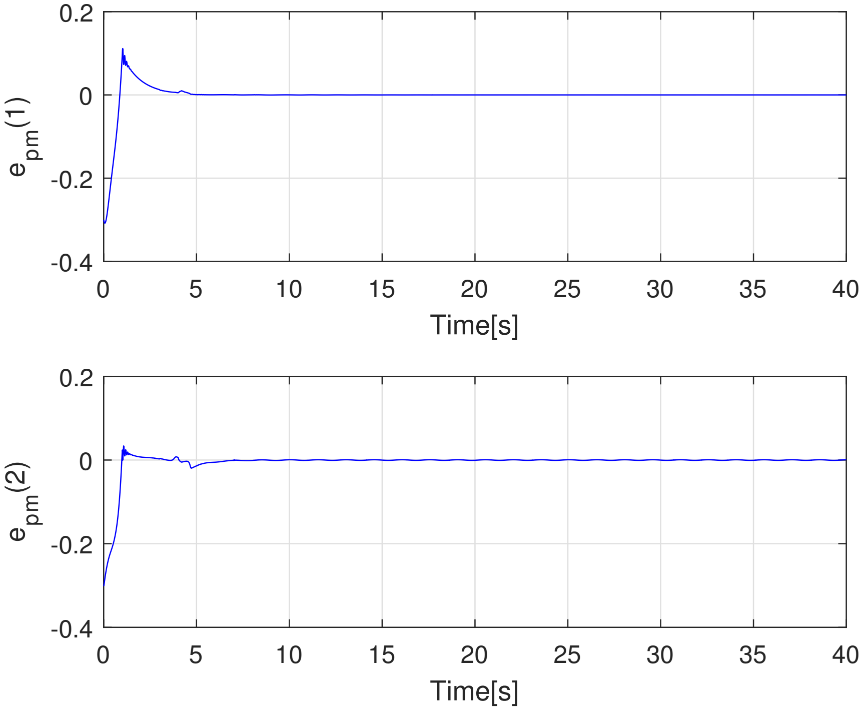

Figure 5.27: Tracking error $\mathbf{e}_{pm} = \mathbf{q}_m(t) - \mathbf{q}_d(t)$

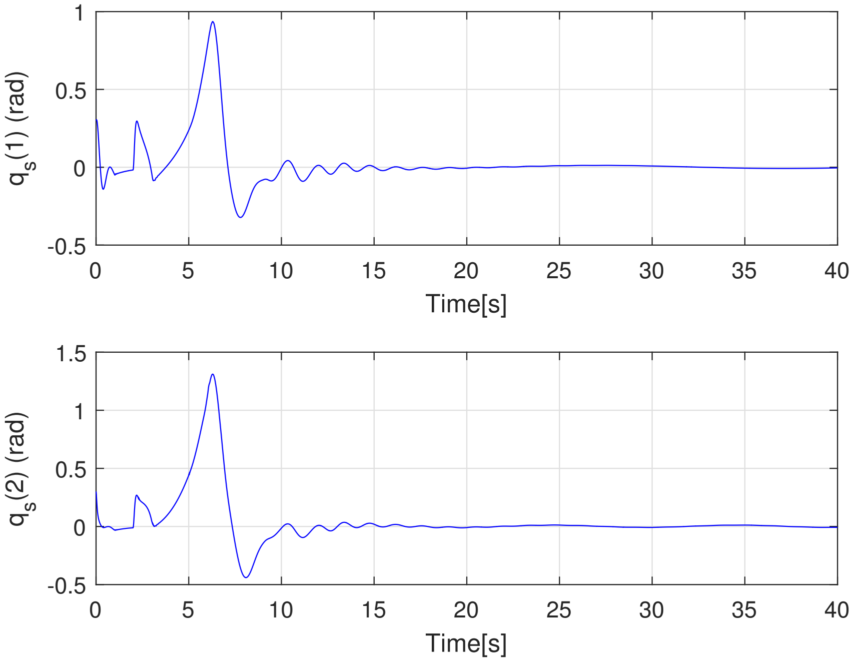

Figure 5.28: Slave trajectory $\mathbf{q}_s$

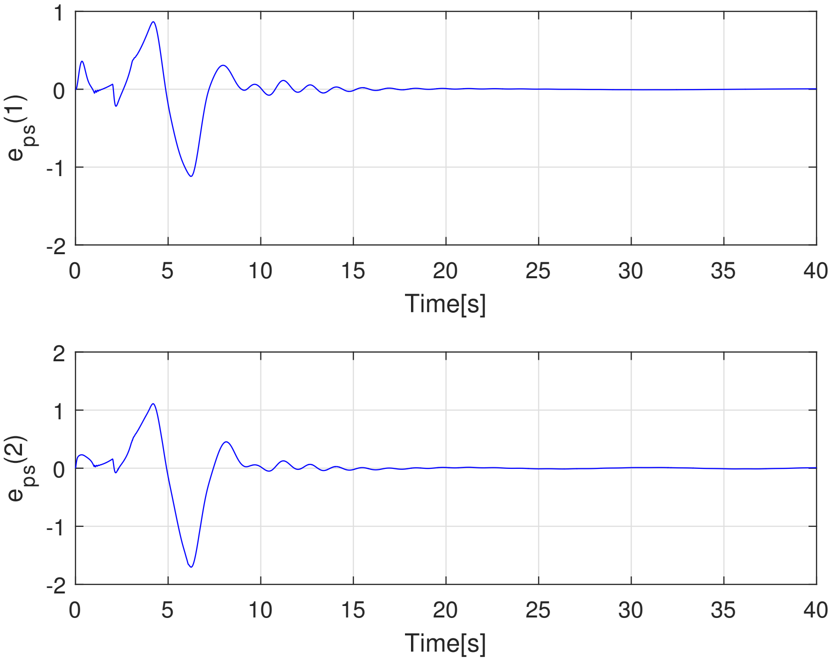

Figure 5.29: Tracking error $\mathbf{e}_{ps} = \mathbf{q}_s(t) - \mathbf{q}_m(t - T_1(t))$)

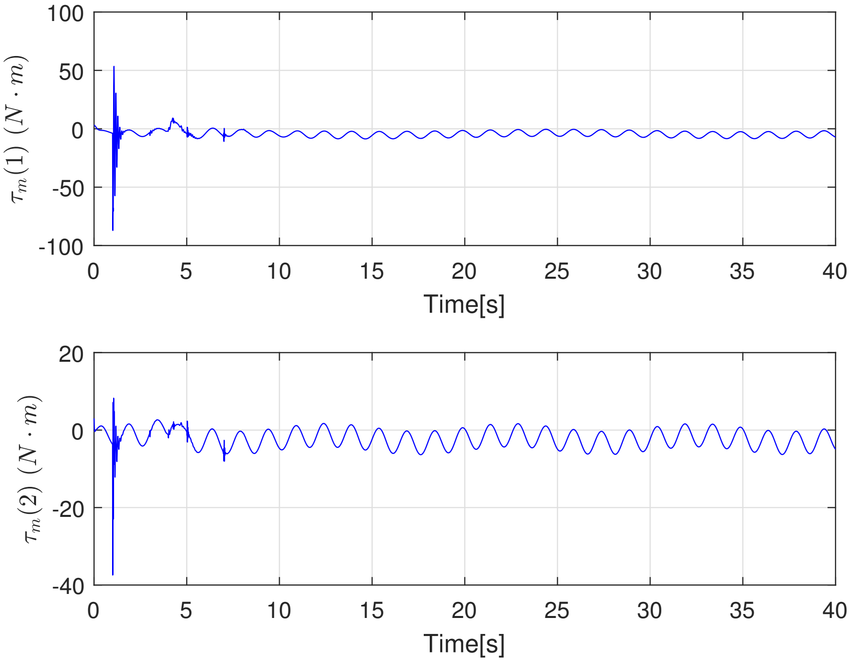

Figure 5.30: Master control input torque $\boldsymbol{\tau}_m$

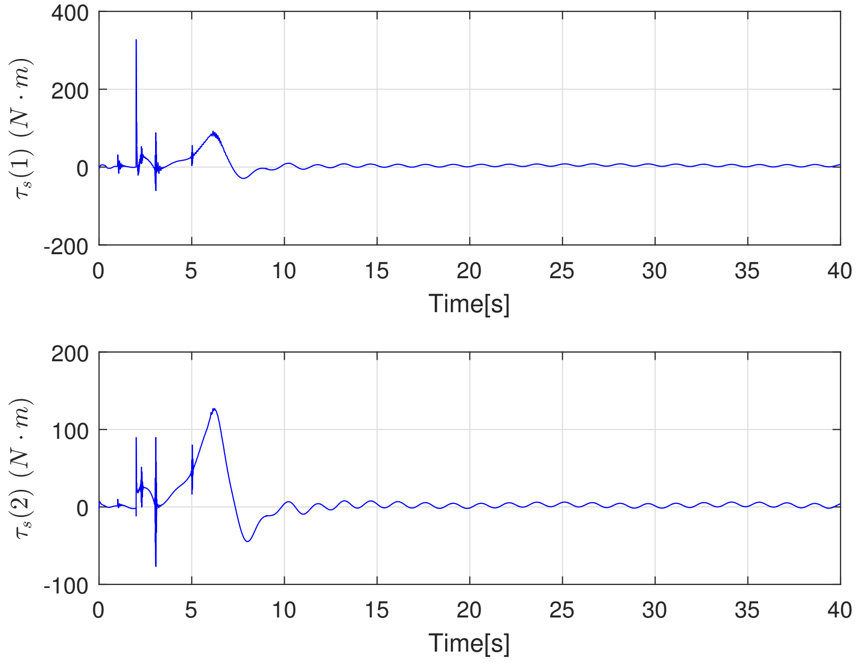

Figure 5.31: Slave control input torque $\boldsymbol{\tau}_s$

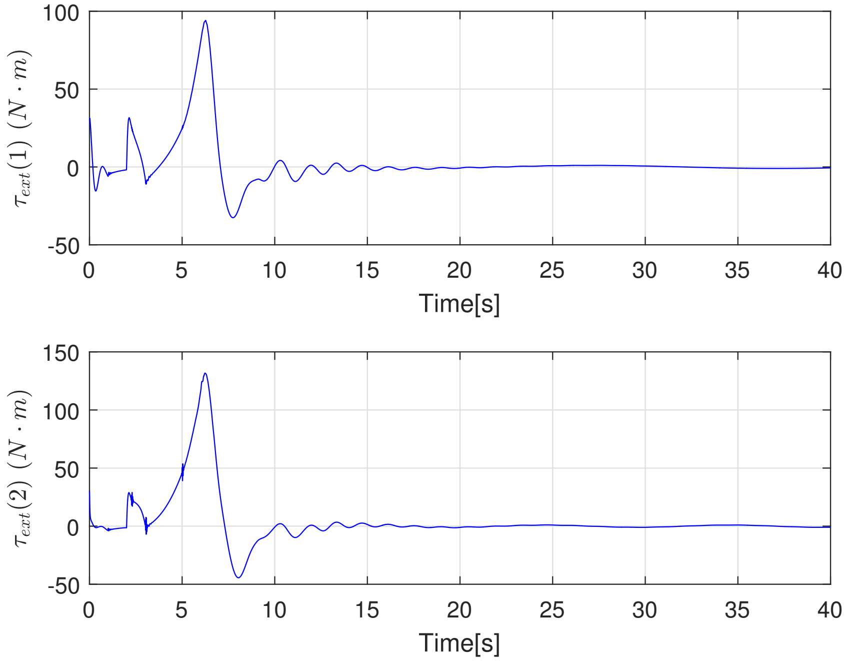

Figure 5.32: Environmental torque $\boldsymbol{\tau}_{ext}$

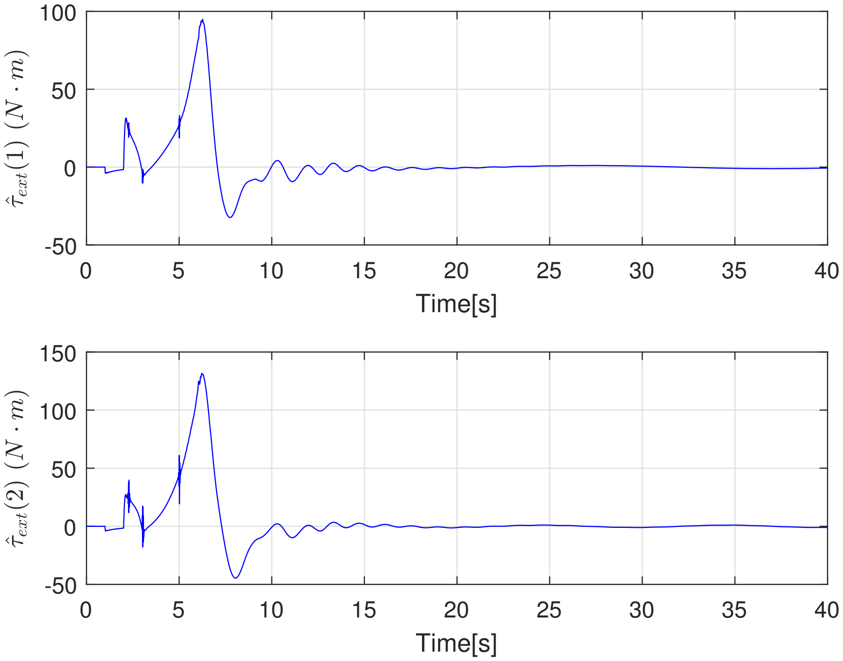

Figure 5.33: Environmental torque estimation $\hat{\tau}_{ext}$

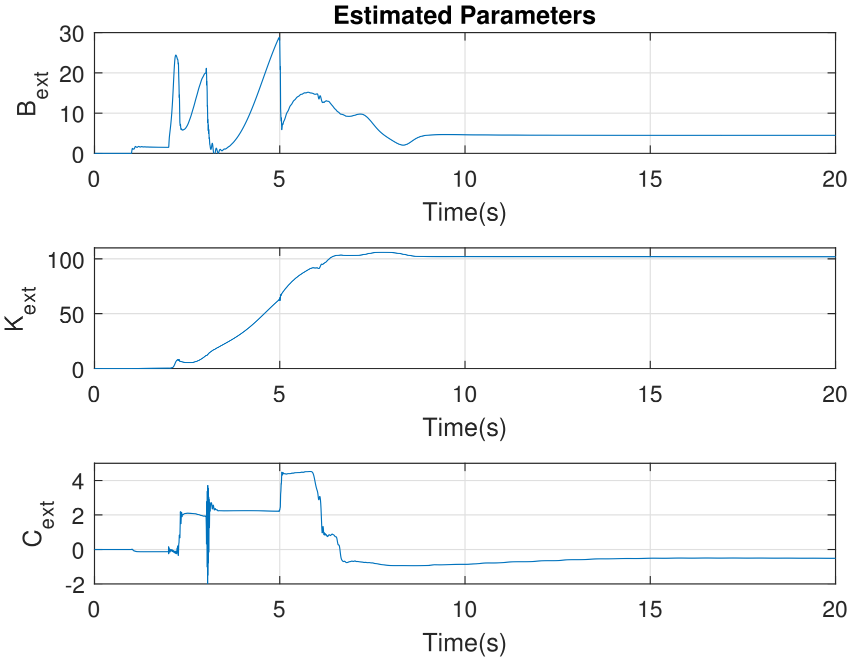

Figure 5.34: Estimated parameters $\boldsymbol{\theta}_{ext} = [B_{ext}, K_{ext}, C_{ext}]$

5.3 Communication Delay Studies

In this section, a study on communication delay will be carried out. The design in this research can tolerate arbitrary, long and time-varying delays. Therefore, a time delay for which upper limit is set to be 200 seconds is applied to the system to test the effectiveness of the controller under long and time-varying delays.

The results are as shown from Fig.5.35 to Fig.5.44. As it can be seen in Fig.5.37 and Fig.5.39, the tracking performance didn't affected by the length of the time delay. Seeing from the results, it can be concluded that, the effectiveness of the controller won't be affected by the length of the time delay.

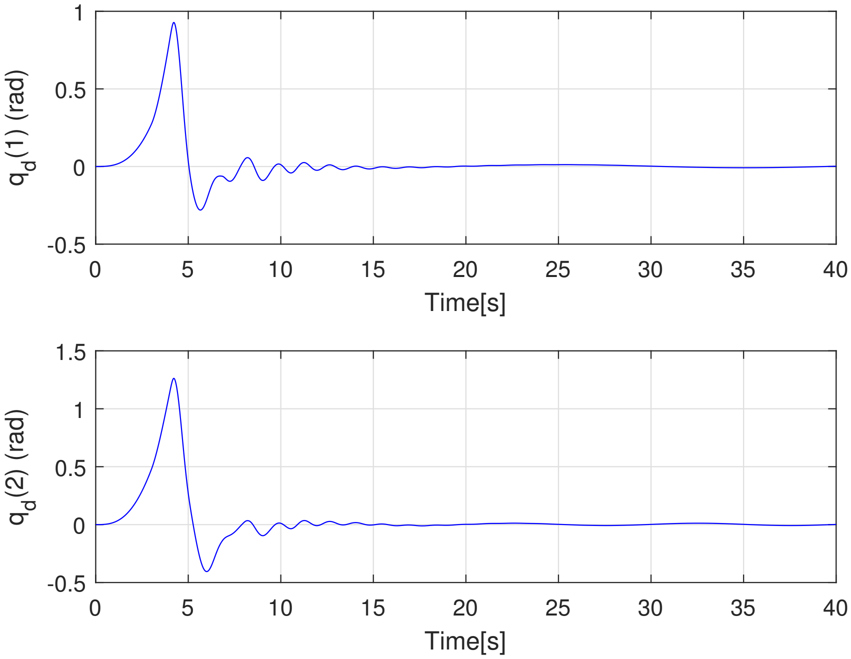

Figure 5.35: Desired trajectory $\mathbf{q}_d$

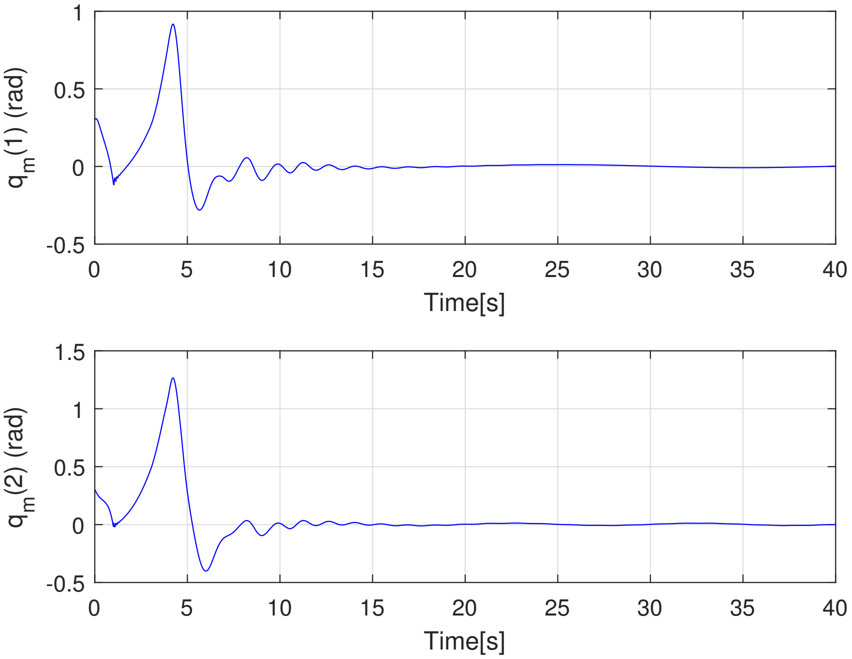

Figure 5.36: Master trajectory $\mathbf{q}_m$

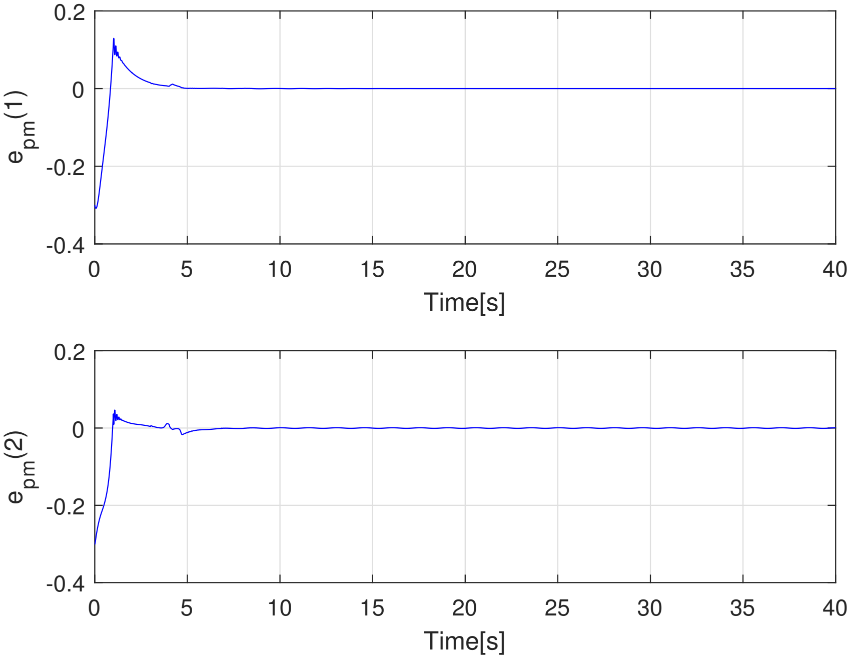

Figure 5.37: Tracking error $\mathbf{e}_{pm} = \mathbf{q}_m(t) - \mathbf{q}_d(t)$

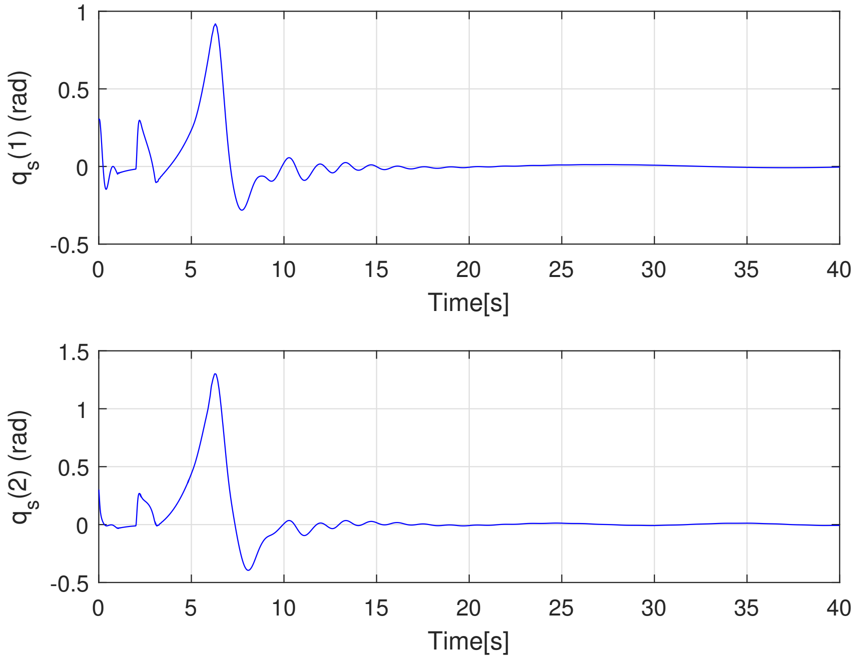

Figure 5.38: Slave trajectory $\mathbf{q}_s$

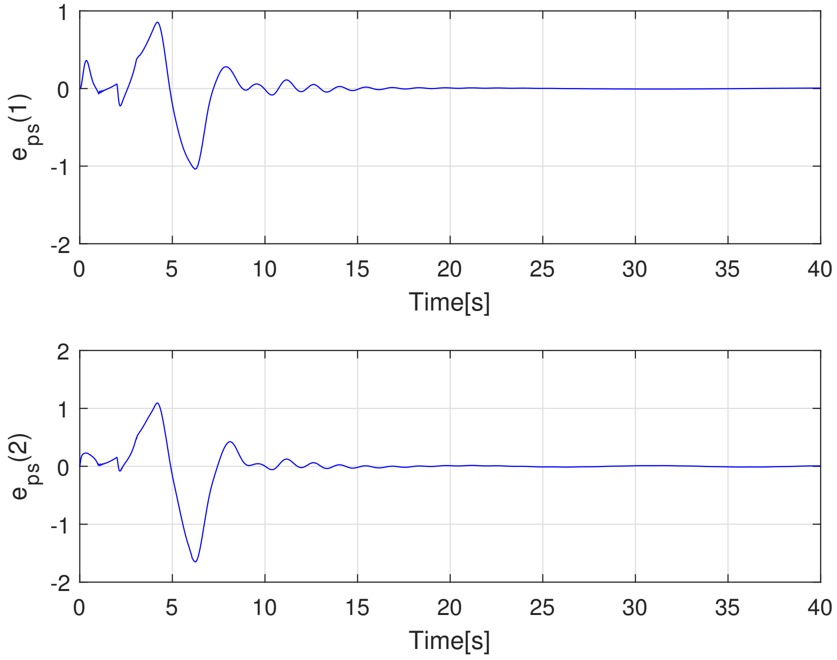

Figure 5.39: Tracking error $\mathbf{e}_{ps} = \mathbf{q}_s(t) - \mathbf{q}_m(t - T_1(t)))$

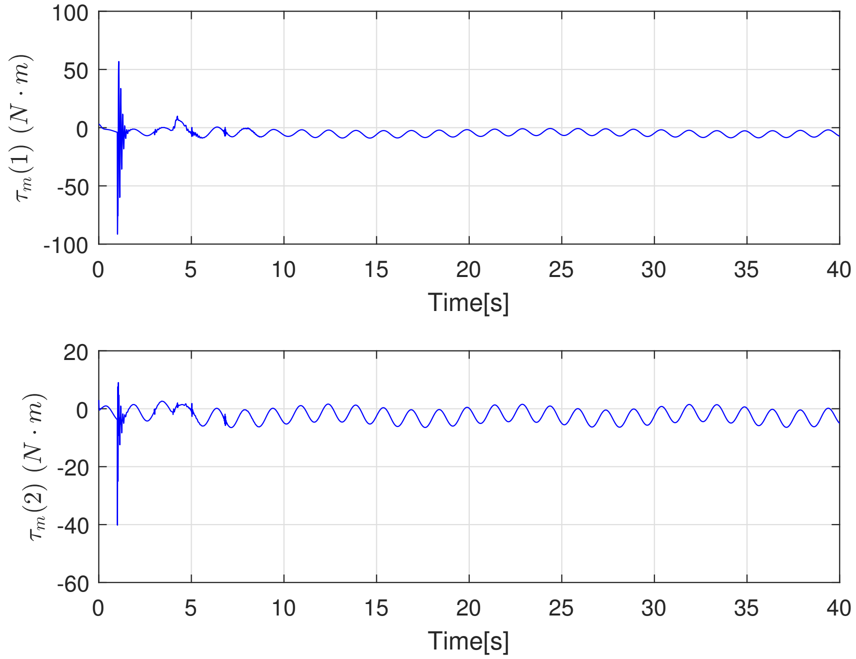

Figure 5.40: Master control input torque $\boldsymbol{\tau}_m$

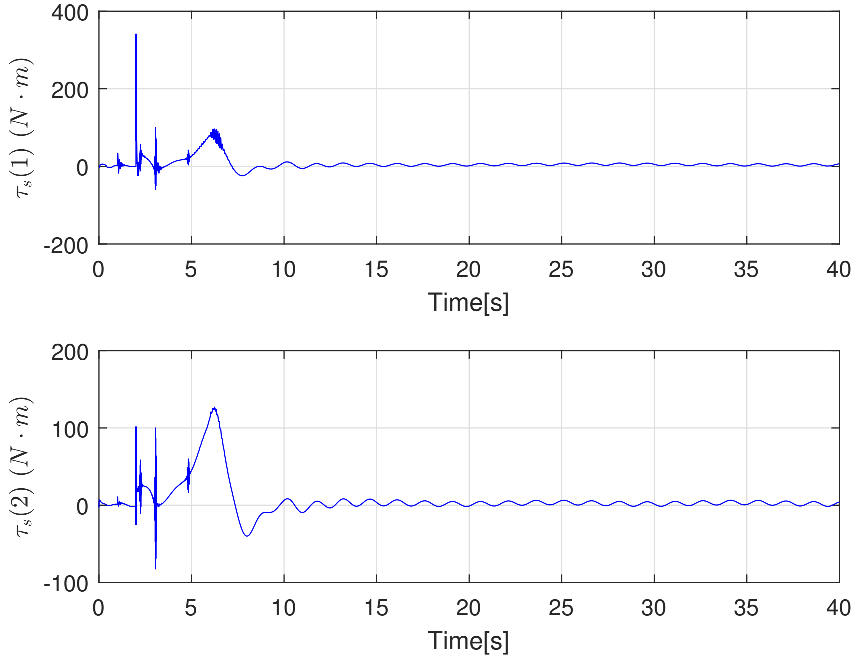

Figure 5.41: Slave control input torque $\boldsymbol{\tau}_s$

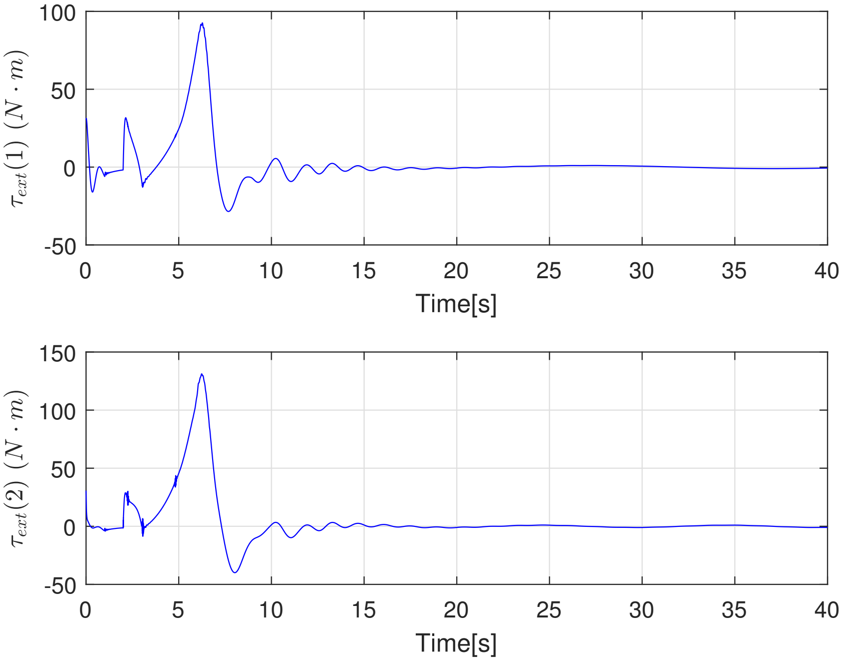

Figure 5.42: Environmental torque $\boldsymbol{\tau}_{ext}$

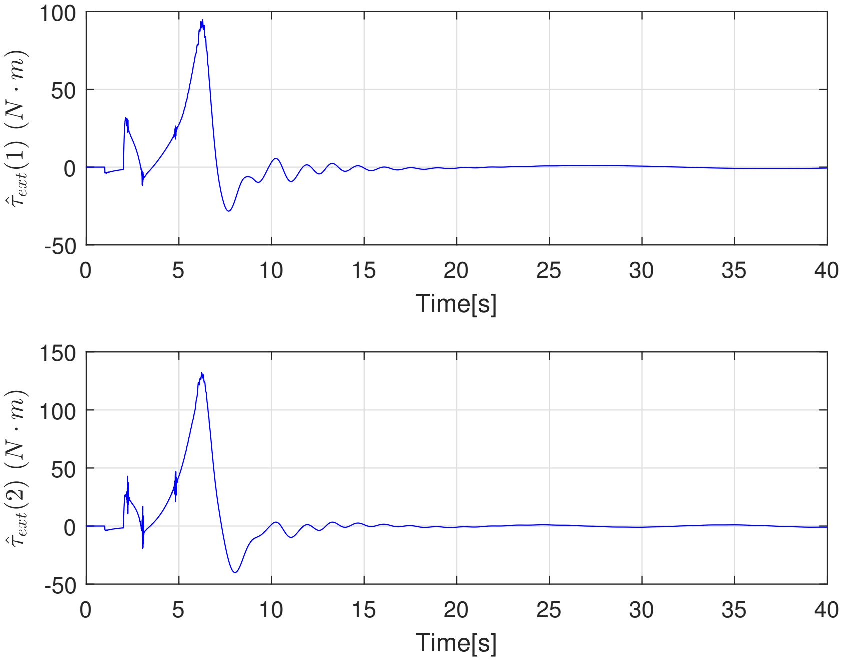

Figure 5.43: Environmental torque estimation $\hat{\tau}_{ext}$

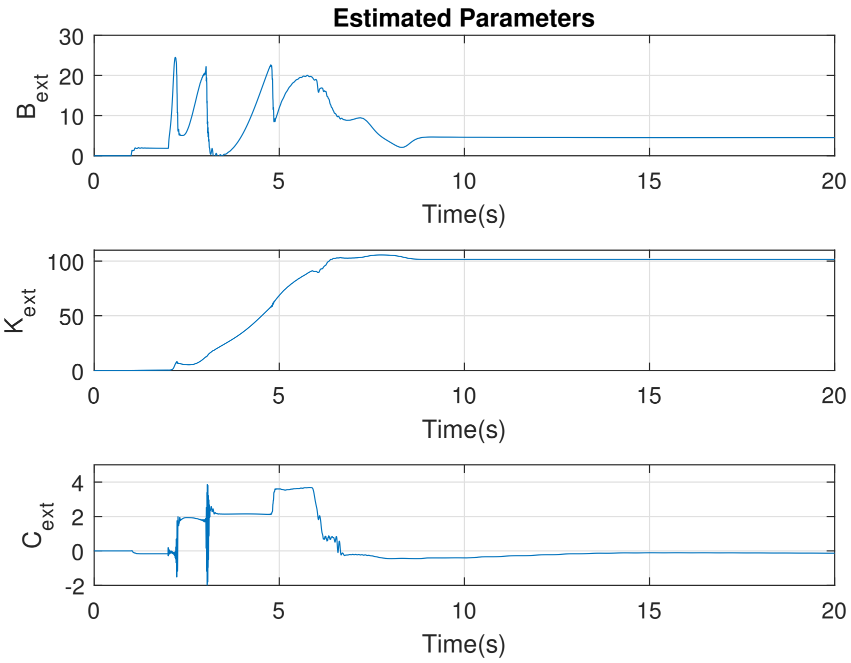

Figure 5.44: Estimated parameters $\boldsymbol{\theta}_{ext} = [B_{ext}, K_{ext}, C_{ext}]$

5.4 Summary

In this chapter, three further studies for the designed control method are presented. The first section is the control gain tuning part. The results of the adaptive gain Γ_i, proportional gain K_{i1}, K_{i2} and the robust gain $K_{\Delta i}$, $i \in \{m, s\}$ tuning verify the effectiveness of the control design respectively.

The second section is the model mismatch section. This section further tested the performance of the controller designed in the previous section. As shown in the simulation results, the mismatch of the parameter values did not affect the performance of the controller.

The third section is the communication delay study. As a result, the teleoperation system can tolerate large, time-varying delay.

Chapter 6

Studies on Different Controllers

In this chapter, two different control methods are applied to the two-degree-of freedom model in order to compare with the proposed approach.

The first method is the wave variable method while the second method is another adaptive control method.

6.1 Wave Variable Method

In this section, the wave variable method on a two degree of freedom manipulator will be introduced. The results of the wave variable method is compared to the results of the method presented in Chapter 2 to 4.

The main objective of this section is to implement the wave variable approach to the teleoperation system in which instead of sending power variables to slave, a transformation is made first and the new variables are transferred over the communication channel. Usually, a pair of torque and velocity is chosen as new variables but other variables can also be chosen. The whole concept of the wave variable approach comes under the umbrella of passivity; wave variable transformation ensures that the passivity is preserved under this scheme. The power flow can be redefined as:

$$P = \dot{x}^T F = \frac{1}{2}u^T u + \frac{1}{2}v^T v, \tag{6.1}$$

where, $\frac{1}{2}u^T u$ is the power flowing along the main direction (considering a positive sign), $\frac{1}{2}v^T v$ is the power flowing against the main direction (considering a negative sign). In this approach, the value of u, and v can be derived as follows,

$$u = \frac{b\dot{x} + F}{\sqrt{2b}}, \qquad v = \frac{b\dot{x} - F}{\sqrt{2b}}, \tag{6.2}$$

where b is a characteristic impedance associated with the wave variables and it can be selected arbitrarily.[38] This transformation is one to one and always remains unique. All the information is preserved under this transformation.

Power variables can be derived as follows,

$$b\dot{x} = \sqrt{\frac{b}{2}}(u+v) \qquad F = \sqrt{\frac{b}{2}}(u-v) \tag{6.3}$$

Then, the value of F and u can be calculated using,

$$F = b\dot{x} - \sqrt{2b}v \qquad u = -v + \sqrt{2b}\dot{x} \tag{6.4}$$

The characteristic impedance b is a key parameter that can affect the behaviour of the system directly. It is related to motion and torque levels and can be used for tuning the system. If the value of b is increased, it will reduce the motion and lead to an increase in torque levels and the system is more damped. On the contrary, if the value of b decreases, motion is increased and torque levels will decrease and so, the system appears less damped.

Wave based communication in terms of input and output variables is depicted in Fig.6.1.

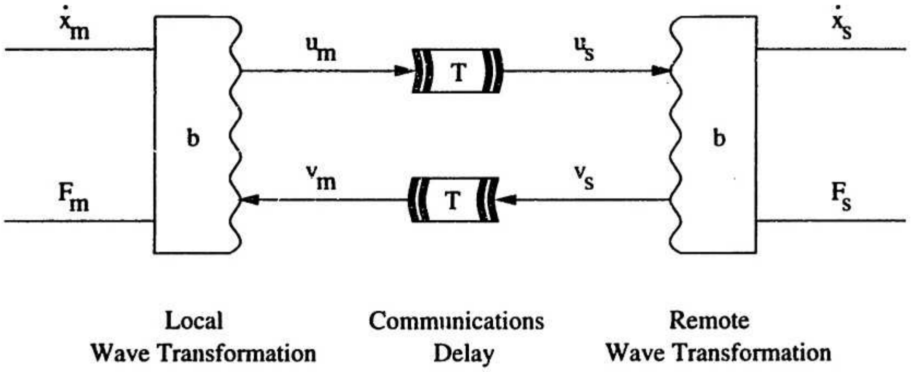

Figure 6.1: Wave based communication

The governing equations for transmission are,

$$\begin{aligned} u_s(t) &= u_m(t-T) \\ v_m(t) &= v_s(t-T) \end{aligned} \tag{6.5}$$

The input is given as,

$$\begin{aligned} u_m(t) &= \frac{b\dot{x}_m(t) + F_m(t)}{\sqrt{2b}}, \\ v_s(t) &= \frac{b\dot{x}_s(t) - F_s(t)}{\sqrt{2b}}. \end{aligned} \tag{6.6}$$

The output equations are,

$$\begin{aligned} \dot{x}_m(t) &= \sqrt{\frac{b}{2}} v_m(t) + \frac{1}{b} F_m(t), \\ \dot{x}_s(t) &= \sqrt{\frac{b}{2}} v_s(t) - \frac{1}{b} F_s(t). \end{aligned} \tag{6.7}$$

The dynamic models for master and slave manipulators are,

$$\begin{aligned} \boldsymbol{\tau}_m &= J_m \ddot{\theta} + B_m \dot{\theta}, \\ \boldsymbol{\tau}_c &= J_s \ddot{\theta} + B_s \dot{\theta}, \end{aligned} \tag{6.8}$$

where B_m, B_s are the damping terms, $\boldsymbol{\tau}_m$, $\boldsymbol{\tau}_s$ are the input torques and J_m, J_s are the inertias of the system being simulated.

The system parameters are selected as,

$$\begin{aligned} B_m = B_s &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \\ J_m = J_s &= \begin{bmatrix} 4 & 0 \\ 0 & 3 \end{bmatrix} \end{aligned} \tag{6.9}$$

A large time-varying delay is introduced to the system. The maximum value of the delay is 2 seconds. The human input torque and the remote environment are considered in two different cases. The first case is when the input human operator torque and the remote environment are both step torques as shown in Fig.6.2 and Fig.6.8. The second case is when the human operator torque and the environment torque are sinusoid torques as shown in Fig.6.9 and Fig.6.15.

As it can be seen in Fig.6.3, in the first case, the tracking errors between the master and the slave manipulator for both joints are not converging to zero in the end. In Fig.6.10, when the human and the environmental torque are both sinusoid signals, the plot of the tracking error for joint 1 is oscillating within a bound from

-0.43 to 0.45. And for joint 2, the tracking error is also oscillating within a bound from -0.59 to 0.5.

Compared to the results in Chapter 4, from Fig.4.3 to Fig.4.16, it can be concluded that the control method, was introduced in Chapter 3, is better than the wave variable method.

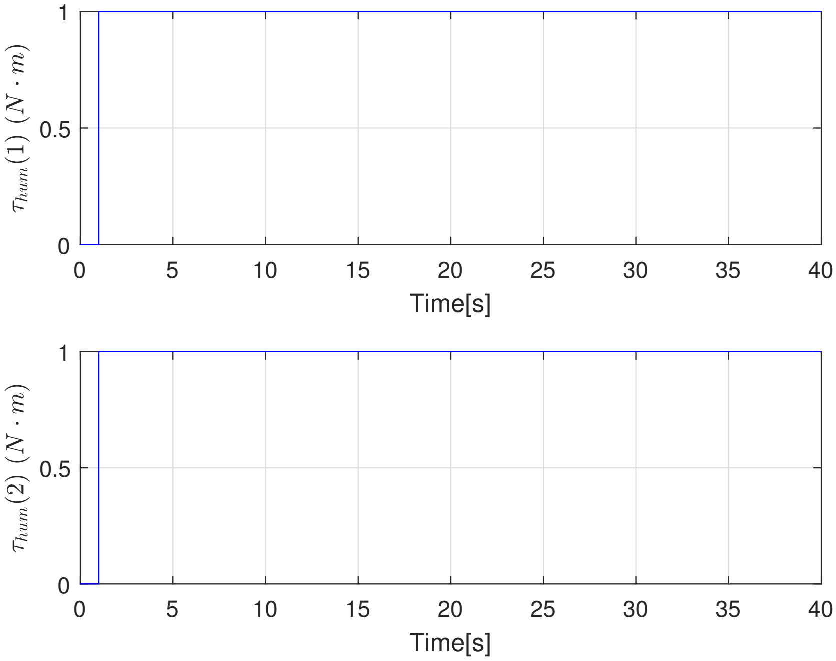

Figure 6.2: Simulated step human operator torque $\boldsymbol{\tau}_{hum}$ for Case 1

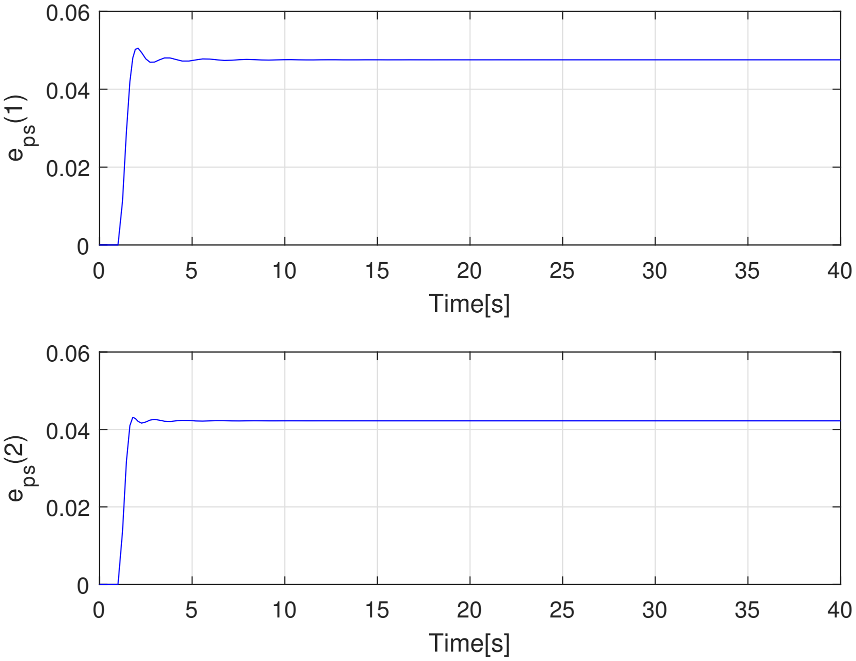

Figure 6.3: Tracking error $\mathbf{e}_{ps}$ for Case 1

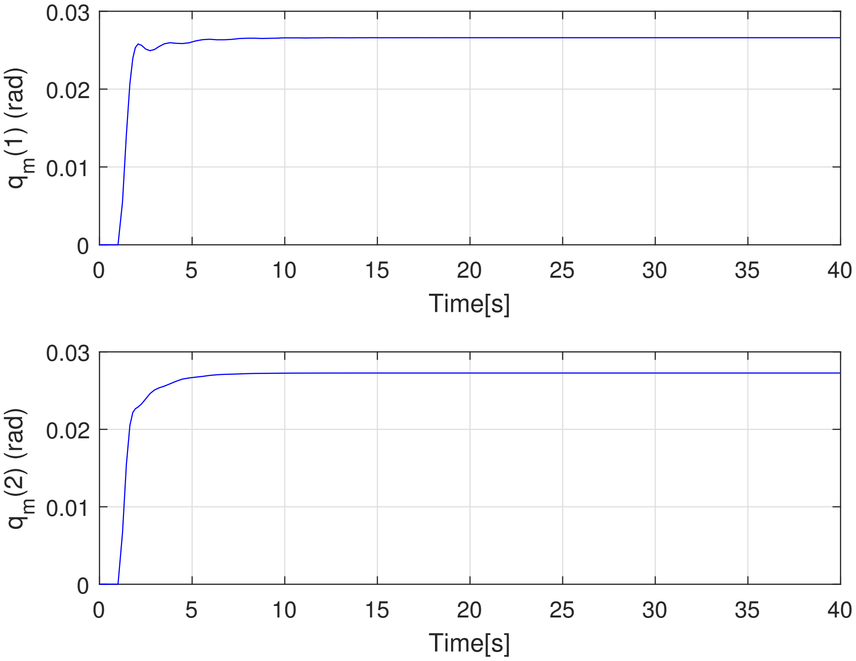

Figure 6.4: Master trajectory $\mathbf{q}_m$ for Case 1

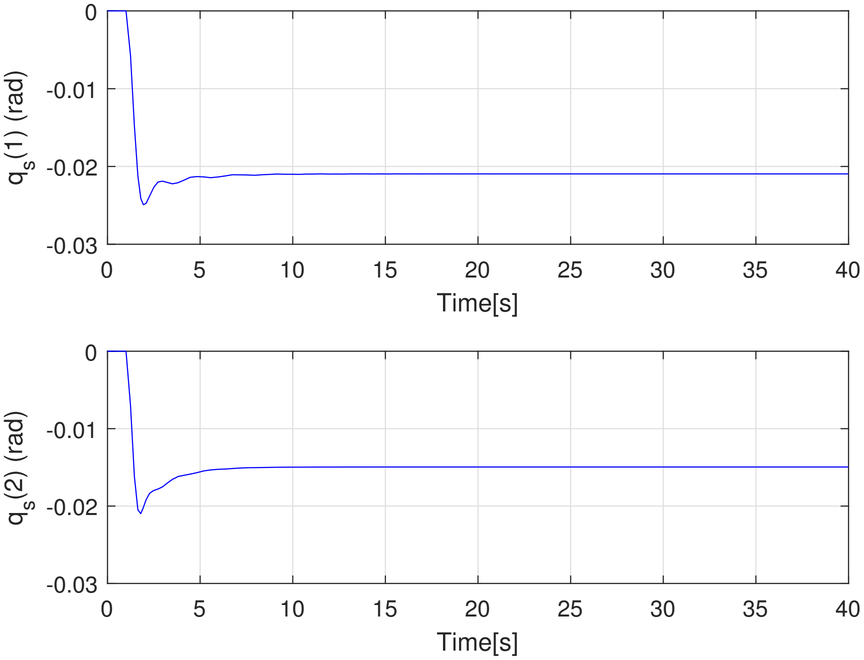

Figure 6.5: Slave trajectory $\mathbf{q}_s$ for Case 1

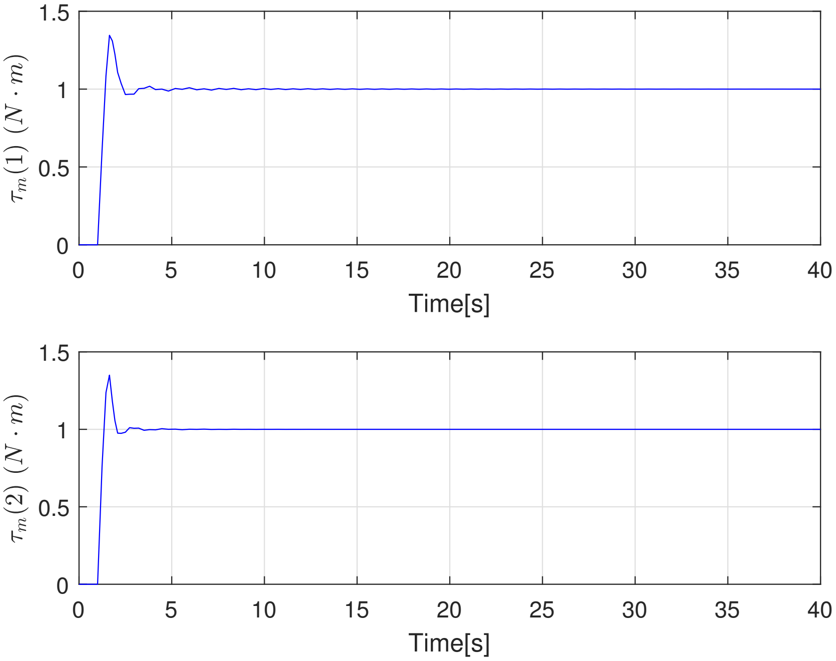

Figure 6.6: Master control input torque $\boldsymbol{\tau}_m$ for Case 1

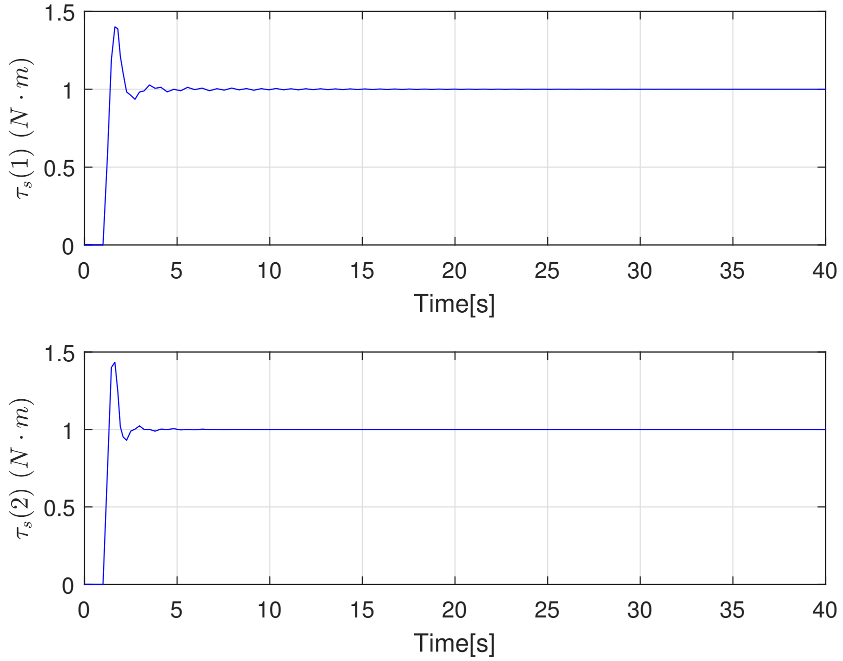

Figure 6.7: Slave control input torque $\boldsymbol{\tau}_s$ for Case 1

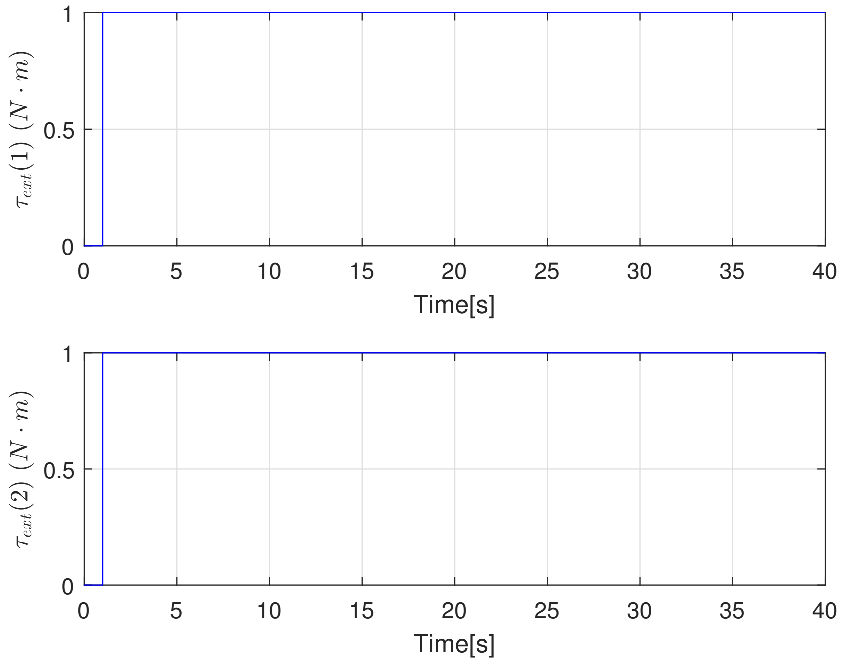

Figure 6.8: Environmental torque $\boldsymbol{\tau}_{ext}$ for Case 1

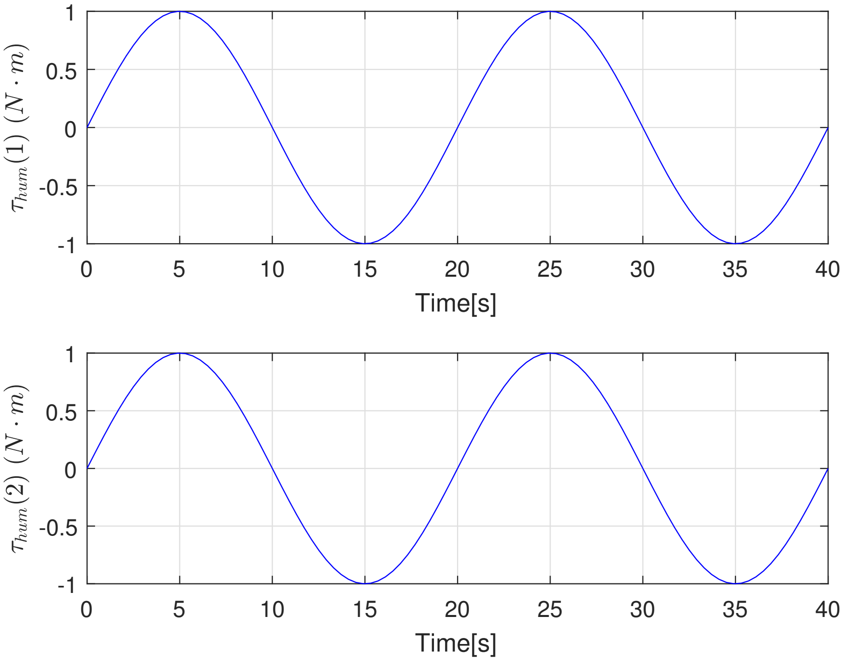

Figure 6.9: Simulated sinusoidal human operator torque $\boldsymbol{\tau}_{hum}$ for Case 2

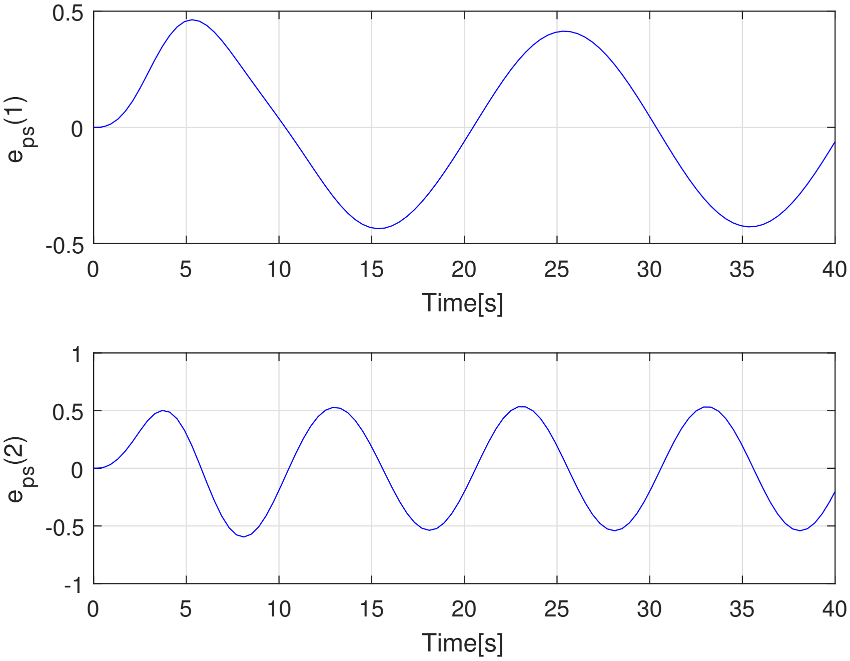

Figure 6.10: Tracking error $\mathbf{e}_{ps}$ for Case 2

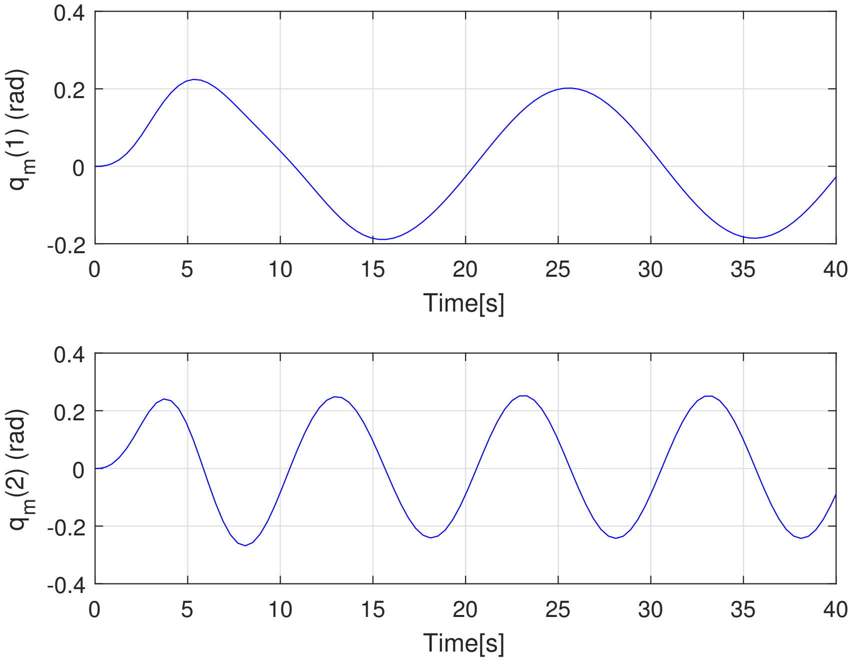

Figure 6.11: Master trajectory $\mathbf{q}_m$ for Case 2

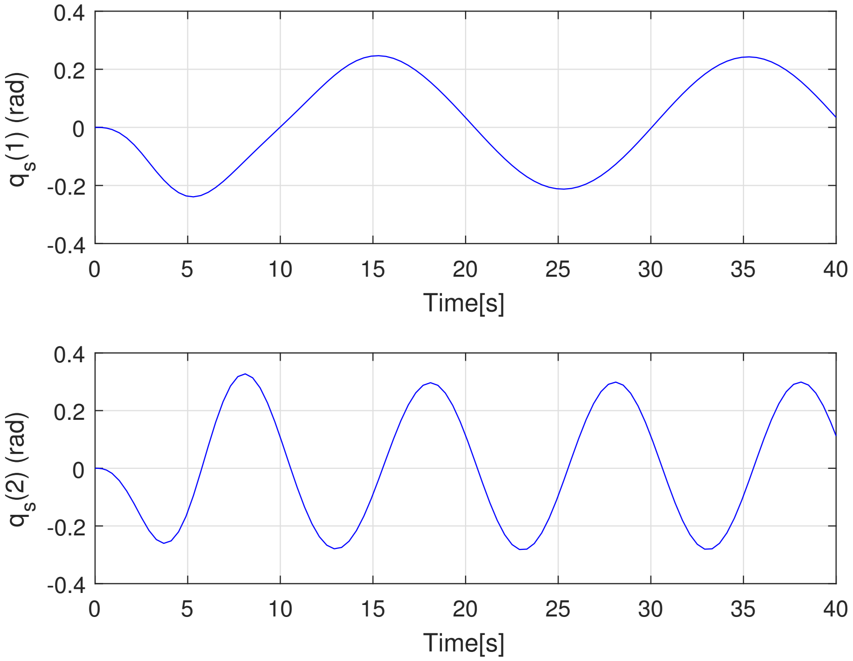

Figure 6.12: Slave trajectory $\mathbf{q}_s$ for Case 2

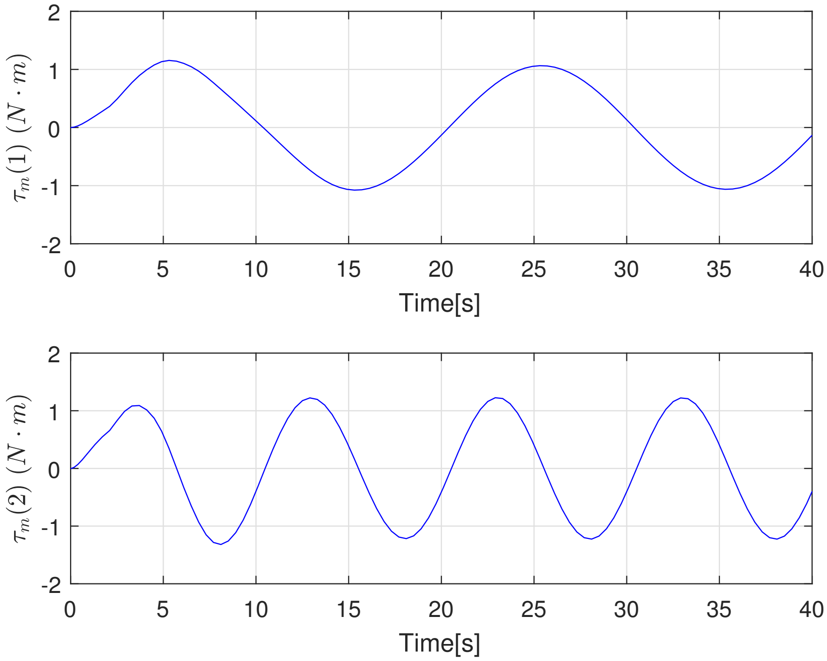

Figure 6.13: Master control input torque $\boldsymbol{\tau}_m$ for Case 2

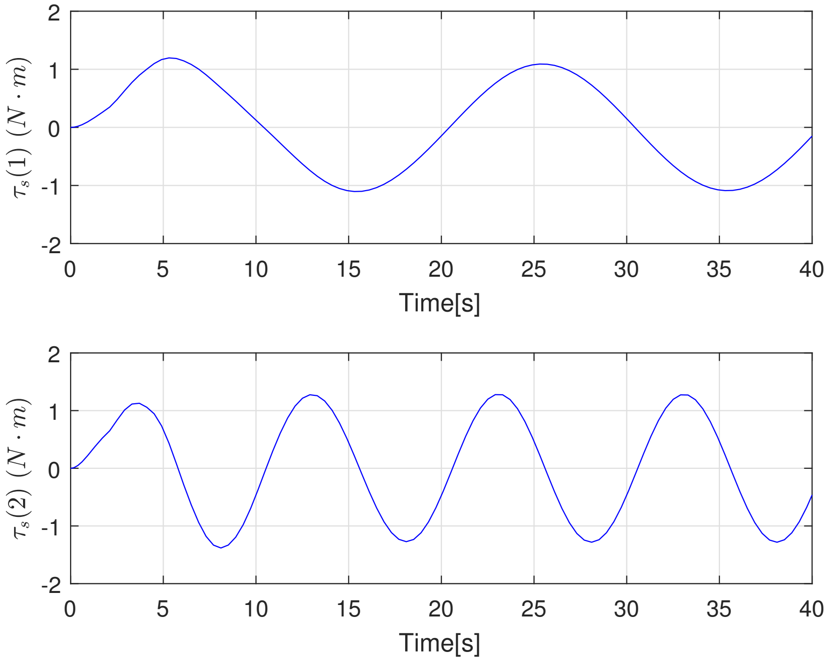

Figure 6.14: Slave control input torque $\boldsymbol{\tau}_s$ for Case 2

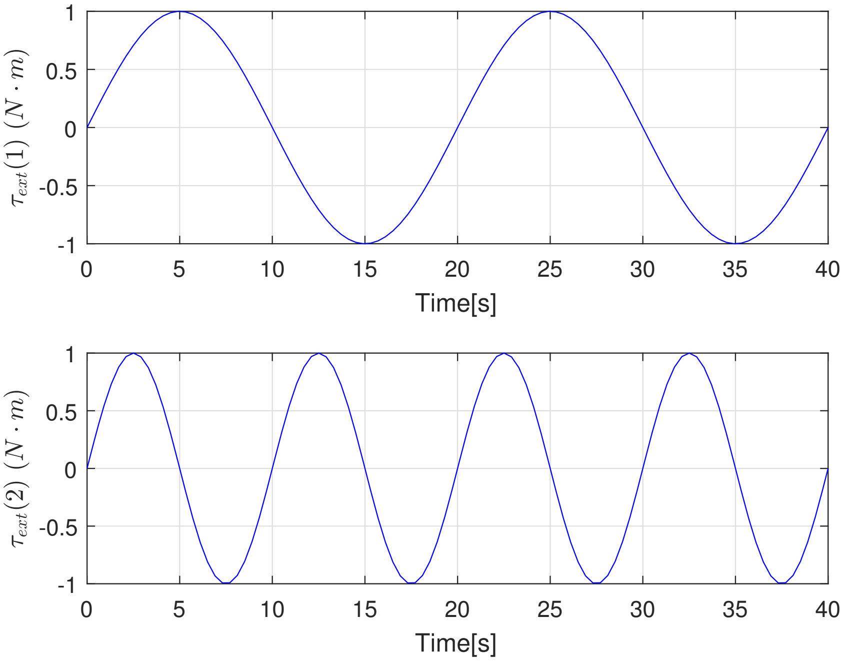

Figure 6.15: Environmental torque $\boldsymbol{\tau}_{ext}$ for Case 2

6.2 Adaptive Control Method

In this section, an adaptive control scheme for nonlinear teleoperation systems with varying asymmetric time delays is presented in order to compare with the method that is being used in the thesis [39].

6.2.1 Dynamic Model

The dynamics model of the bilateral teleoperation system with two-link planar manipulators is proposed as follows,

$$\begin{aligned} M_m(\mathbf{q}_m)\ddot{\mathbf{q}}_m + C_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)\dot{\mathbf{q}}_m + G_m(\mathbf{q}_m) &= \boldsymbol{\tau}_{hum} - \boldsymbol{\tau}_m + \Delta_m, \\ M_s(\mathbf{q}_s)\ddot{\mathbf{q}}_s + C_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s + G_s(\mathbf{q}_s) &= \boldsymbol{\tau}_s - \boldsymbol{\tau}_{ext} + \Delta_s. \end{aligned} \tag{6.10}$$

Variables with the subscripts m and s are associated with the master and slave devices, respectively. The notation $(\dot{\bullet})$ and $(\ddot{\bullet})$ represents the first and second time derivatives, respectively. $M_i(\mathbf{q}_i) \in \Re^{2\times 2}$, where $i \in \{m, s\}$ is the symmetric, positive definite inertia matrix, dealing with the inertial forces the robot manipulator experiences. $C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i$ denotes the centripetal and Coriolis torque where $C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) \in \Re^{2\times2}$. $G_i(\mathbf{q}_i) \in \Re^{2\times1}$ is the gravitational torque. $\mathbf{q}_i \in \Re^{2\times1}$ represents the angular position of the manipulator joints. $\boldsymbol{\tau}_i \in \Re^{2\times1}$ denotes the control input torque. $\boldsymbol{\tau}_{hum} \in \Re^{2\times1}$ is the torque applied by the human operator to the master manipulator, and $\boldsymbol{\tau}_{ext} \in \Re^{2\times1}$ is the environmental torque applied to the slave manipulator. $\Delta_i \in \Re^{2\times1}$ is the approximated modelling errors and external disturbance.

6.2.2 Control Design

In this part, the control design of this method is presented. The control architecture of the entire teleoperation system is shown in Fig.6.16.

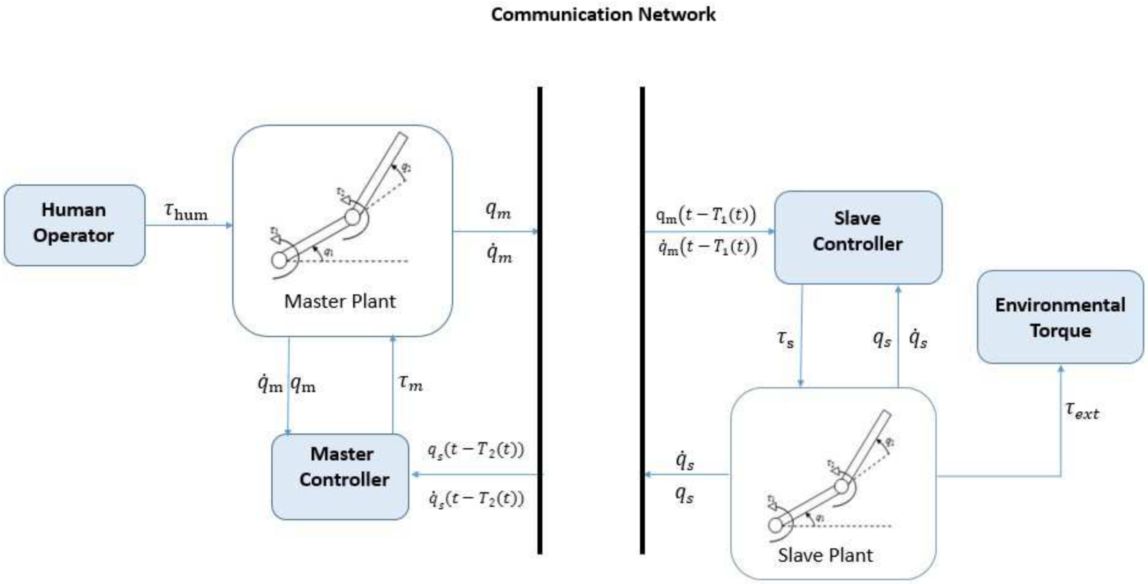

Figure 6.16: Control architecture of the closed-loop teleoperation system

The control input torque $\boldsymbol{\tau}_m$ and $\boldsymbol{\tau}_s$ are defined as follows,

$$\begin{aligned}\boldsymbol{\tau}_m &= -\hat{M}_m(q_m)\dot{e}_p m - \hat{C}_m(q_m, \dot{q}_m)e_{pm} - \hat{g}_m(q_m) + \bar{\boldsymbol{\tau}}_m \\ \boldsymbol{\tau}_s &= \hat{M}_s(q_s)\dot{e}_p s + \hat{C}_s(q_s, \dot{q}_s)e_{ps} + \hat{g}_s(q_s) - \bar{\boldsymbol{\tau}}_s,\end{aligned} \tag{6.11}$$

where $\bar{\boldsymbol{\tau}}_i$ for $i \in m, s$ are the new control signals, $\hat{}$ denotes the estimates of the master and slave parameters. $\mathbf{e}_{pm}$ and $\mathbf{e}_{ps}$ stands for the position error of the master and slave manipulators respectively as follows,

$$\begin{aligned}\mathbf{e}_{pm} &= \mathbf{q}_s(t-T(t)) - \mathbf{q}_m(t)\\ \mathbf{e}_{ps} &= \mathbf{q}_m(t-T(t)) - \mathbf{q}_s(t)\end{aligned} \tag{6.12}$$

where $T(t)$ is a large time-varying delay which is set to a maximum of 2 seconds.

The new control signals $\bar{\boldsymbol{\tau}}_m$ and $\bar{\boldsymbol{\tau}}_s$ are defined as follows,

$$\begin{gathered}\bar{\boldsymbol{\tau}}_s = \begin{cases} 0, & \| \varepsilon_s \|_2 = 0 \\ K_s\varepsilon_s - \frac{1}{2}\dot{\mathbf{e}}_{ps} - \frac{1}{2}\mathbf{e}_{vs} - \frac{e_{vs}^T(\mathbf{e}_{ps} + \dot{\mathbf{e}}_{ps} - \mathbf{e}_{vs})}{2 \| \varepsilon_s \|_2^2}\varepsilon_s + K_{\Delta s} sgn(\varepsilon_s), & \| \varepsilon_s \|_2 \neq 0 \end{cases}\\ \bar{\boldsymbol{\tau}}_m = \begin{cases} 0, & \| \varepsilon_m \|_2 = 0 \\ K_m\varepsilon_m - \frac{1}{2}\dot{\mathbf{e}}_{pm} - \frac{1}{2}\mathbf{e}_{vm} - \frac{e_{vm}^T(\mathbf{e}_{pm} + \dot{\mathbf{e}}_{pm} - \mathbf{e}_{vm})}{2 \| \varepsilon_m \|_2^2}\varepsilon_m + K_{\Delta m} sgn(\varepsilon_m), & \| \varepsilon_m \|_2 \neq 0 \end{cases}\end{gathered} \tag{6.13}$$

where K_i for $i \in \{m, s\}$ is a positive definite matrix which is set to be $3I$ and $\| \, . \, \|_2$ represents Euclideaqn norm. The velocity error $\mathbf{e}_{vi}$ and ε_i for $i \in \{m, s\}$ are defined as follows,

$$\begin{aligned}\varepsilon_i &= \dot{\mathbf{q}}_i - \mathbf{e}_{pi}\\ \boldsymbol{e}_{vm} &= \dot{\mathbf{q}}_s(t-T(t)) - \dot{\mathbf{q}}_m\\ \mathbf{e}_{vs} &= \dot{\mathbf{q}}_m(t-T(t)) - \dot{\mathbf{q}}_s\end{aligned} \tag{6.14}$$

$K_{\Delta m} sgn(\varepsilon_m)$ and $K_{\Delta s} sgn(\varepsilon_s)$ are the robust terms in the controller, where the robust control gains are set to be time $K_{\Delta m} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$, $K_{\Delta s} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$ respectively.

6.2.3 Simulation Results

As can be seen in Fig.6.17 and Fig.6.23, a human operator torque has been applied to the master manipulator. A sinusoid case and a step input case are considered in the simulation.

As it can be seen in the simulation plots, the performance of this control method is good, it has a limitation in the theory of the Lyapunov stability proof. Although the human input torque to the system don't has to be zero when running the simulation, Lyapunov stability theory can only be carried out if $\boldsymbol{\tau}_h = \boldsymbol{\tau}_e = 0$(free motion)[39]. Therefore, it can be concluded that the controller designed in my work is better.

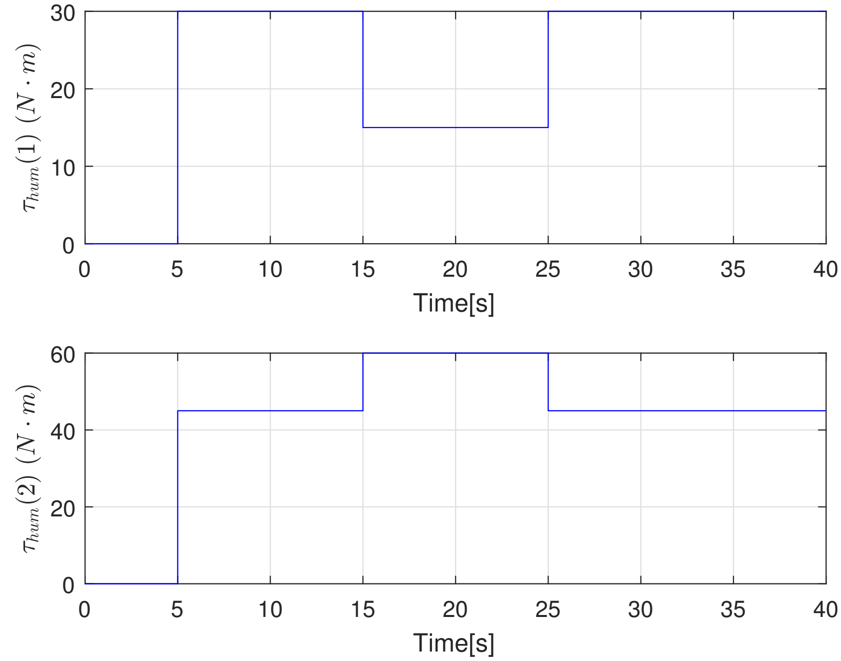

Figure 6.17: Simulated step human operator torque $\boldsymbol{\tau}_{hum}$ for Case 1

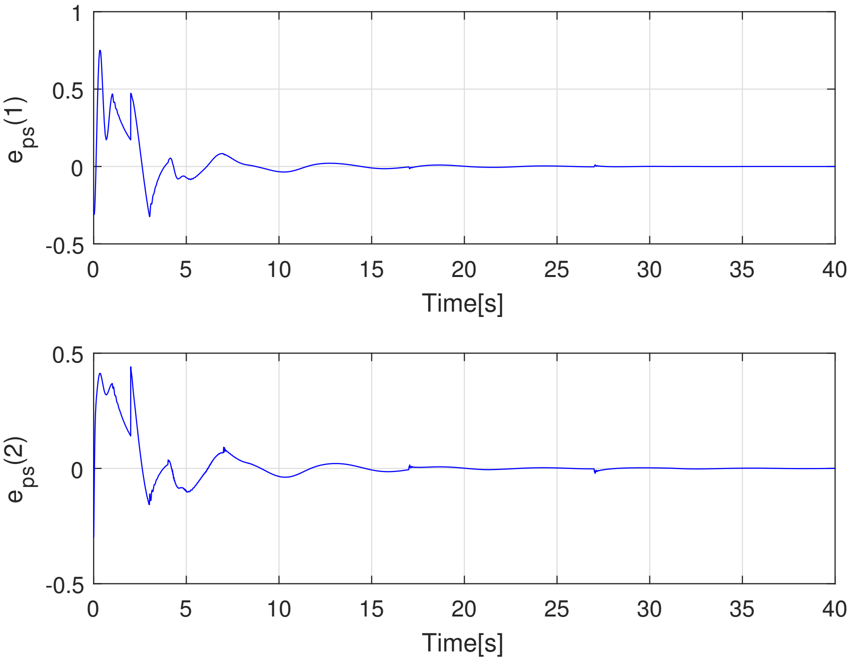

Figure 6.18: Tracking error $\mathbf{e}_{ps}$ for Case 1

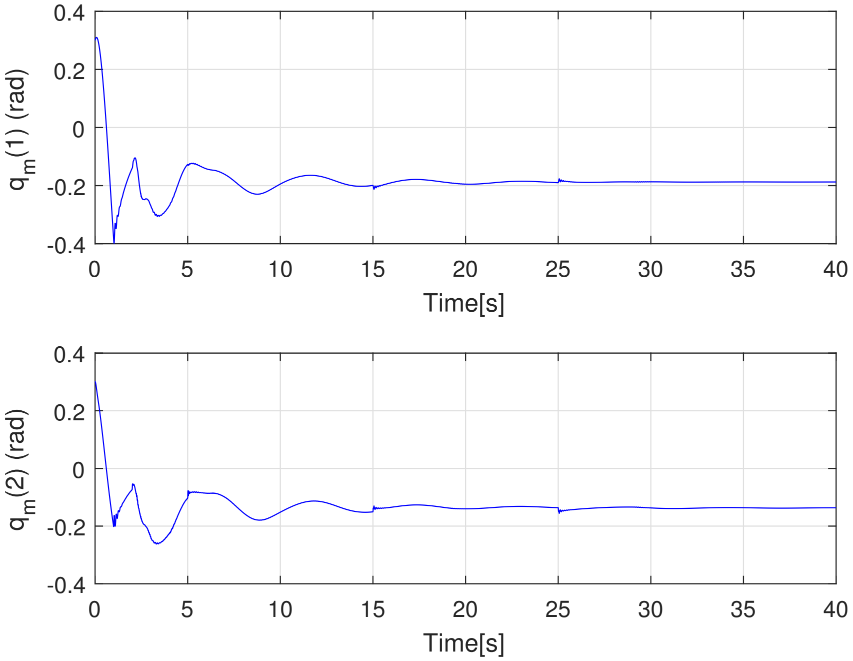

Figure 6.19: Master trajectory $\mathbf{q}_m$ for Case 1

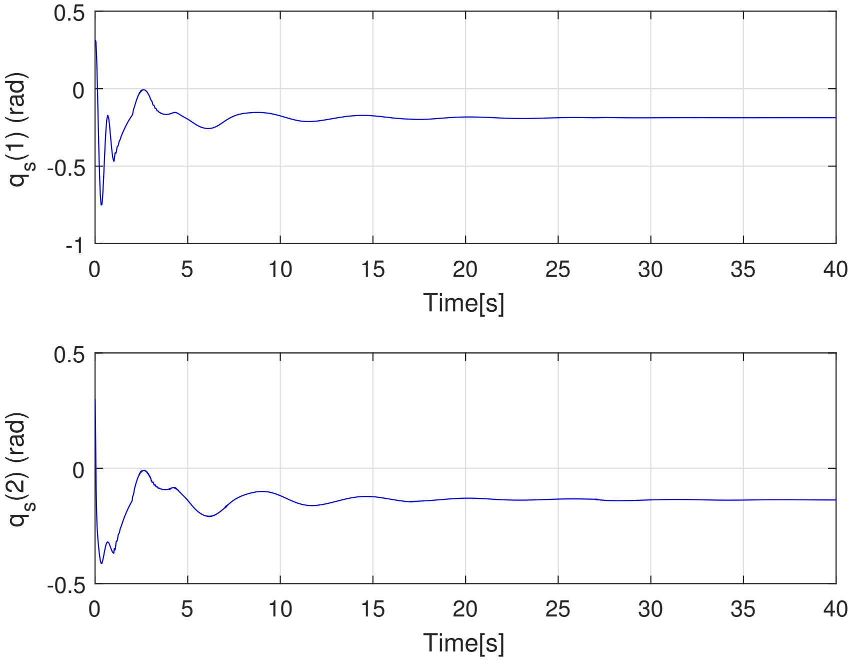

Figure 6.20: Slave trajectory $\mathbf{q}_s$ for case 1

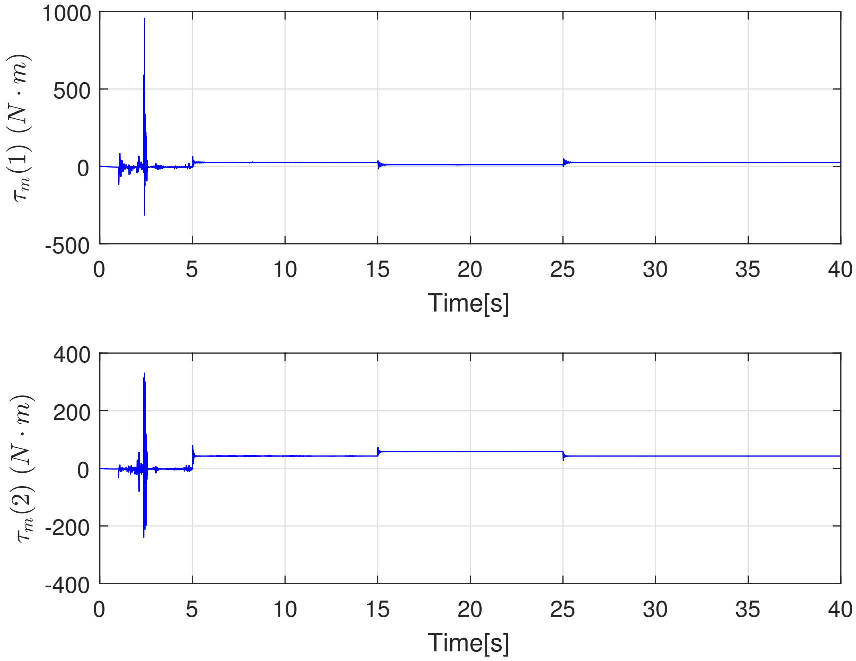

Figure 6.21: Master control input torque $\boldsymbol{\tau}_m$ for Case 1

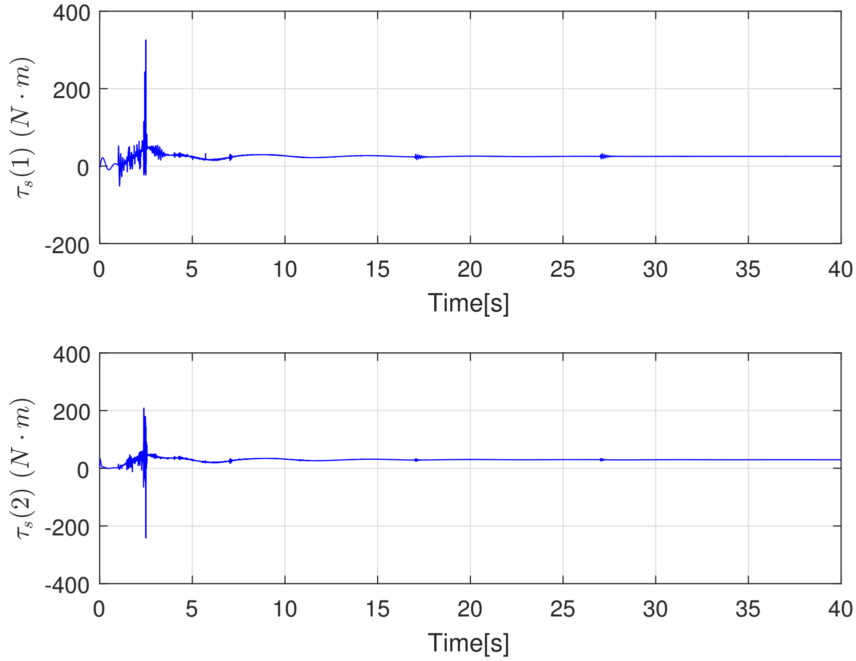

Figure 6.22: Slave control input torque $\boldsymbol{\tau}_s$ for Case 1

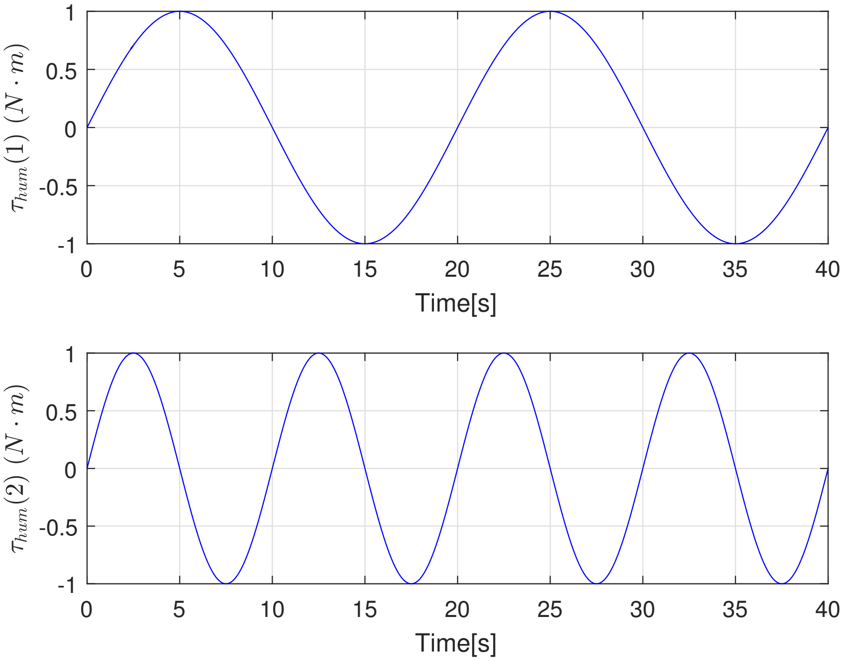

Figure 6.23: Simulated sinusoidal human operator torque $\boldsymbol{\tau}_{hum}$ for Case 2

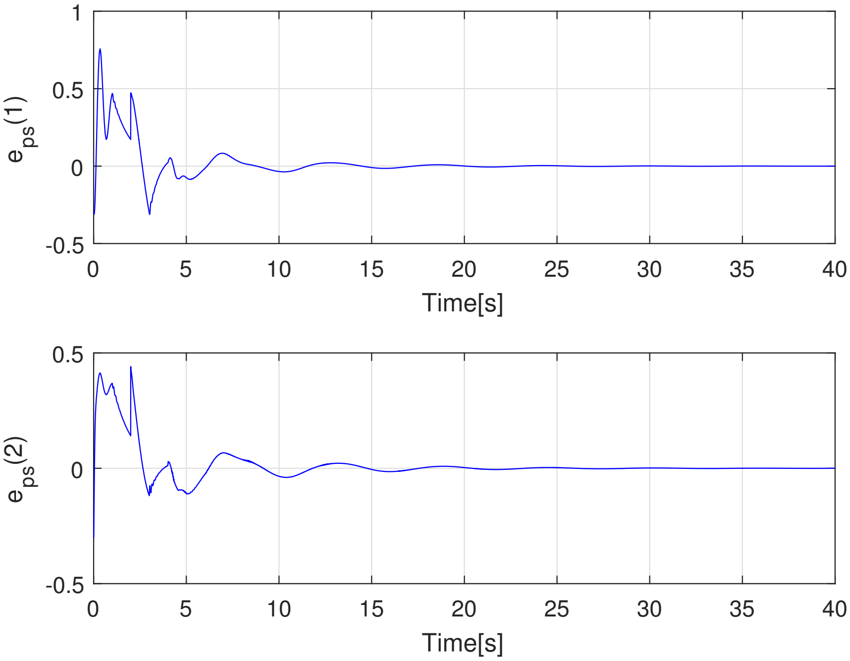

Figure 6.24: Tracking error $\mathbf{e}_{ps}$ for Case 2

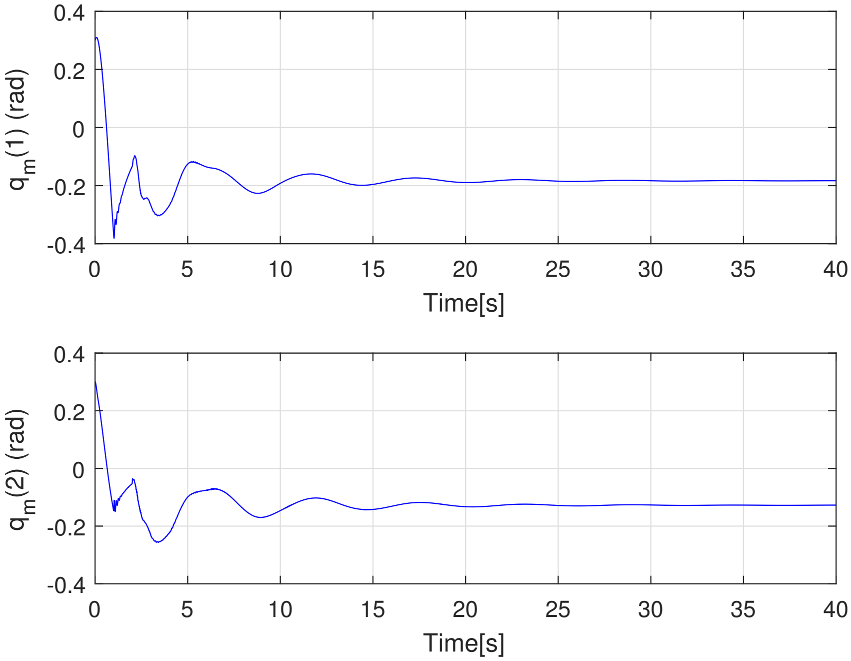

Figure 6.25: Master trajectory $\mathbf{q}_m$ for Case 2

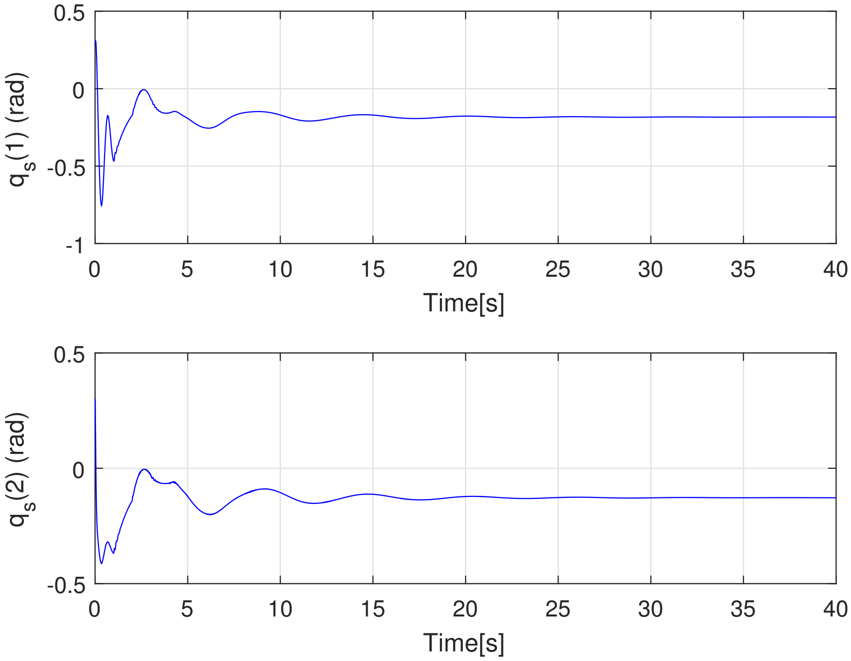

Figure 6.26: Slave trajectory $\mathbf{q}_s$ for Case 2

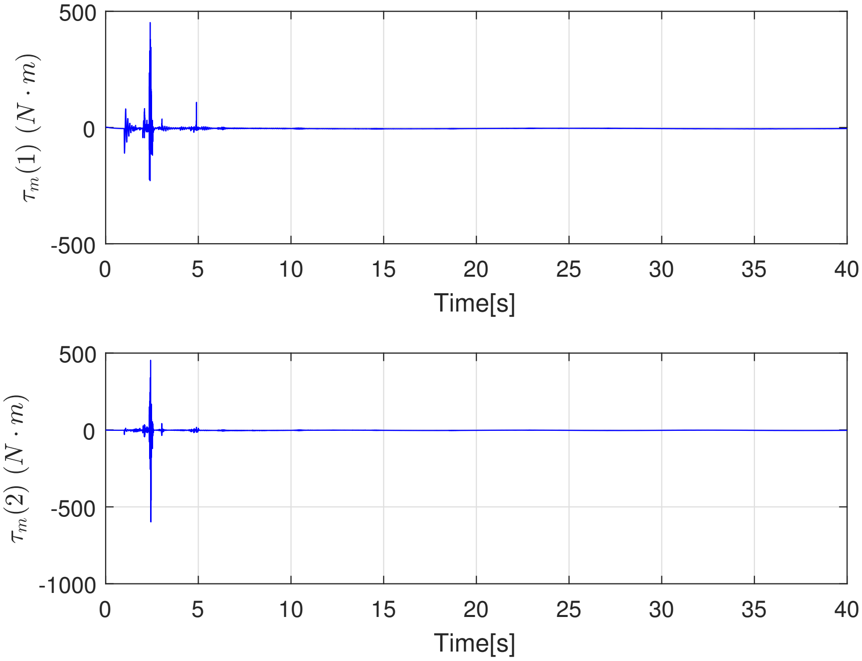

Figure 6.27: Master control input torque $\boldsymbol{\tau}_m$ for Case 2

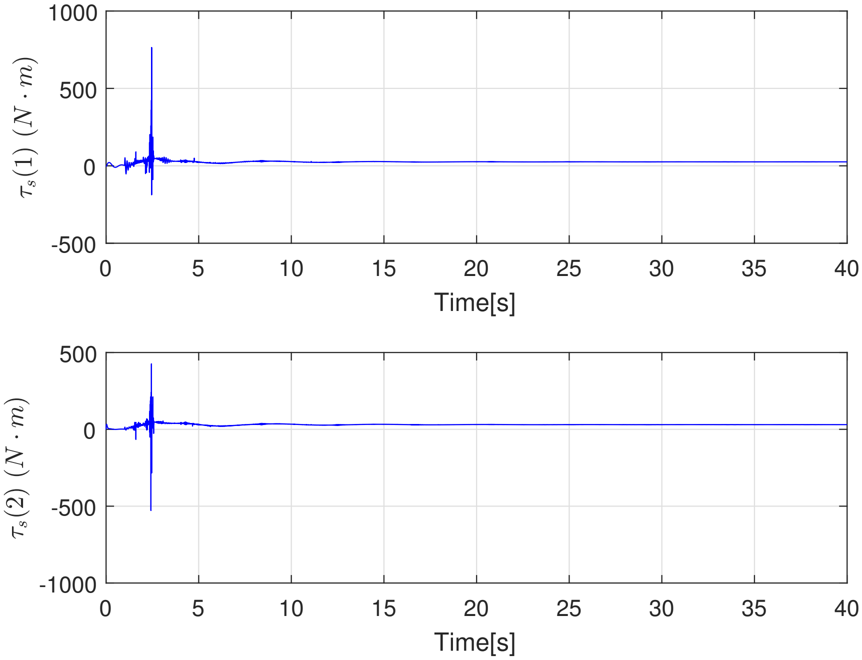

Figure 6.28: Slave control input torque $\boldsymbol{\tau}_s$ for Case 2

6.3 Summary

In this chapter, two different control methods have been applied to the two-degree-of freedom model.

The first method is the wave variable method. The results showed that the performance of the adaptive robust controller that developed in this work is better than the controller derived using the wave variable method. The second method is another adaptive control method. Although the performance of this control method is good, it has a limitation in the theory of the Lyapunov stability proof. Therefore, it can be concluded that the proposed controller designed is better.

Chapter 7

Conclusions and Future Works

In this chapter, the results of this work summarized. Furthermore, the future work that can be done to this design are concluded in the second section as well.

7.1 Conclusions

In conclusion, A new adaptive robust control scheme is developed in this thesis. A novel nonlinear adaptive robust control method is presented to deal with nonlinearities, unknown parameters, modeling errors and uncertainties in the bilateral system. An environmental torque estimator is designed to estimate immeasurable torques which commonly exists in real applications. The unknown parameters of the environmental torques are estimated online by a least square adaptive law. A novel structure of communication block is developed in this work. This structure is designed to improve the control performance of the adaptive robust controller. An impedance control structure is developed on the master side to ensure the desired transparency performance.

Simulation results on a pair of two degree of freedom robotic manipulators verified the robust stability, excellent transparency and synchronization of the design. Simulation studies of control gain tuning and model mismatch are carried out to further verify the effectiveness of the control design. Two other control methods are applied to two-degree-of freedom model for the possible comparison with the proposed design. In conclusion, the nonlinear adaptive robust control in this paper can achieve stability, the excellent transparency and synchronization performance under arbitrary time-varying delays.

7.2 Future Work

Future work can be further conducted based on the design in this work. Firstly, the controller can be developed to apply to a multilateral system. A rebuild of the communication block should be considered with the new design. The slave devices are not necessarily to be two degree-of-freedom manipulators, manipulators with more degree-of-freedom can also be considered in the simulation with the proposed design in multilateral system. Control gain tuning are needed in order to achieve stability with the design. Experiment work can be further conducted to the designed control strategy. Frictions of the hardware need to be considered during the tuning process.

Bibliography

[1] Li, Zhijun, Yuanqing Xia, and Chun-Yi Su. Intelligent Networked Teleoperation Control. Springer, 2015.

[2] R.J. Anderson and M.W. Spong, Bilateral control of teleoperators with time delay. Automatic Control, IEEE Transactions on, vol. 23,no. 5, pp.494-501, 1989

[3] P.F. Hokayem, and M.W. Spong, Bilateral teleoperation: An historical survey, Automatica, vol. 42, no. 12, pp. 2035-2057, 2006.

[4] "Digital Surgery". Biomed.brown.edu. October 1, 2000. Retrieved March 5, 2017.

[5] "Robots as surgical enablers". MarketWatch. 3 February 2005. Retrieved March 5, 2017.

[6] "Prepping Robots to Perform Surgery". New York Times. 4 May 2008. Retrieved 5 March 2017.

[7] "Surgery Enabled by da Vinci". Women's Medicine of Niagara. Retrieved 6 March 2017.

[8] Skaar SB, Ruoff CF (1994) Teleoperation and robotics in space. Progress In astronautics and aeronautics, American Institute of Aeronautics and Astronautics

[9] Nelson, Jon. "Mars Science Laboratory Curiosity Rover". NASA. Retrieved March 6, 2017.

[10] "China lands Jade Rabbit robot rover on Moon". BBC. 14 December 2013.

[11] "http://wallpaperus.org/american-science-3000x1776-wallpaper-2227306/". Retrieved March 6, 2017.

[12] "China's Yutu rover dies on the moon". Spaceflight Now. Retrieved March 6, 2017.

[13] Yoshida K. (2001) ETS-VII Flight Experiments For Space Robot Dynamics and Control. In: Rus D., Singh S. (eds) Experimental Robotics VII. Lecture Notes in Control and Information Sciences, vol 271. Springer, Berlin, Heidelberg

[14] "https://radhesh.wordpress.com/2008/05/11/pid-controller-simplified/". Retrieved March 6, 2017.

[15] K. Ogata, Modern Control Engineering, 4th ed. Upper Saddle River, N.J.: Prentice Hall, 2002.

[16] E. Nuno, R. Ortega, N. Barabanov, L. Basanez, “A Globally Stable PD Controller for Bilateral Teleoperators,” Robotics, IEEE Transactions on , vol. 24, no. 3, pp.753-758, June 2008.

[17] A. Forouzantabar, H.A. Talebi, A.K. Sedigh, “Bilateral control of master-slave manipulators with constant time delay,” American Control Conference (ACC), 2011, pp. 1133-1138.

[18] L.G. Garcia-Valdovinos, V. Parra-Vega, J.-A. Mendez-Iglesias, M.A. Arteaga, “Cartesian sliding PID force/position control for transparent bilateral teleoperation,” 31st Annual Conference of IEEE Industrial Electronics Society, 2005, pp. 1979-1985.

[19] K. H. Ang, G. Chong, L. Yun, PID control system analysis, design, and technology, Control Systems Technology, IEEE Transactions on , vol. 13, no. 4, pp.559-576, July 2005.

[20] Smith, J. M. Closer control of loops with dead time. Chemical Engineering Progress, 53(5):217-219, May 1957.

[21] P. B. Deshpande and R. H. Ash, Elements of Computer Process Control With Advanced Control Applications. Research Triangle Park NC: Instrument Society of America, 1981. pp. 227-250.

[22] "https://en.wikipedia.org/wiki/Smithpredictor". Retrieved March 7, 2017.

[23] Prokopiou, P. and Tzfestas, S. A novel scheme for human-friendly and timedelays robust neuropredictive teleoperation. Journal of Intelligent and Robotic Systems, 25:311-340, 1999.

[24] Bemporad, A. Predictive control of teleoperated constrained systems with unbounded communication delays. In Proceedings of the 37th IEEE Conference on Decision and Control, pages 305-310, Tampa, Florida, December 1998.

[25] Sheng, J. and Spong, M. Model predictive control for bilateral teleoperation systems with time delays. In Canadian Conference on Electrical and Computer Engineering, pages 1877-1880, May 2004.

[26] Y. J. Pan, C. Canudas-de-Wit and O. Sename, "A New Predictive Approach for Bilateral Teleoperation With Applications to Drive-by-Wire Systems," in IEEE Transactions on Robotics, vol. 22, no. 6, pp. 1146-1162, Dec. 2006.

[27] Hatanaka, Takeshi, et al. Passivity-based control and estimation in networked robotics. Springer, 2015.

[28] J. Artigas; J.H. Ryu; C. Preusche, “Position drift compensation in time domain passivity based teleoperation,” Intelligent Robots and Systems (IROS),IEEE/RSJ International Conference on, 2010, pp.4250-4256.

[29] N. Chopra, M. W. Spong, R. Lozano, “Synchronization of bilateral teleoperators with time delay,” Automatica, vol. 44, no. 8, pp.2142-2148, August 2008.

[30] Zheng Chen, Adaptive Robust Control Teleoperation, Retrieved on March 15, 2017.

[31] G. Niemeyer and J.-J. E. Slotine. Stable adaptive teleoperation. IEEE Journal of Oceanic Engineering, 16(1):152–162, January 1991.

[32] Hashtrudi-Zaad, K. and Salcudean, S. E. Adaptive transparent impedance reflecting teleoperation. In Proceedings of the IEEE International Conference on Robotic and Automation, pages 1369-1374, April 1996.

[33] Ryu, J.-H. and Kwon, D.-S. A novel adaptive bilateral control scheme using similar closed-loop dynamic characteristics of master/slave manipulators. Journal of Robotic Systems, 18(9):533-543, September 2001.

[34] Tipsuwan, Y. and Chow, M.-Y. Gain scheduler middleware: A methodology to enable existing controllers for networked control and teleoperation-part i: Networked control. IEEE Transactions on Industrial Electronics, 51(6):1218- 1227, December 2004.

[35] Yao B. Integrated direct/indirect adaptive robust control of SISO nonlinear systems in semi-strict feedback form. Proceedings of the American Control Conference 2003; 4:3020–3025. (O. Hugo Schuck Best Paper (Theory) Award).

[36] John J. Craig. 1989. Introduction to Robotics: Mechanics and Control (2nd ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[37] Lyapunov A. M. The General Problem of the Stability of Motion (In Russian), Doctoral dissertation, Univ. Kharkov 1892 English translations: (1) Stability of Motion, Academic Press, New-York,London, 1966 (2) The General Problem of the Stability of Motion, (A. T. Fuller trans.) Taylor,Francis, London 1992. Included is a biography by Smirnov and an extensive bibliography of Lyapunov’s work.

[38] G. Niemeyer and J.-J. E. Slotine. Stable adaptive teleoperation. IEEE Journal of Oceanic Engineering, 16(1):152–162, January 1991. 1.3, 1.4, 1.4, 3.1, 3.1

[39] Hashemzadeh, Farzad, et al. "Adaptive control of nonlinear teleoperation systems with varying asymmetric time delays." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

[40] Leung GMH, Francis BA, Apkarian J. Bilateral controller for teleoperators with time delay via -synthesis. IEEE Transactions on Robotics and Automation 1995; 11(1):105-116.

[41] Suzuki A, Ohnishi K. Novel four-channel bilateral control design for haptic communication under time delay based on modal space analysis. IEEE Transactions on Control Systems Technology 2013; 21(3):882-890.