Adaptive Control of Robot Manipulators in Varying Environments

---

A Thesis
Presented to
the Faculty of the School of Engineering and Applied Science
University of Virginia

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Electrical Engineering

---

by

Jiacheng Chen

May 2022

APPROVAL SHEET

This thesis is submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

Jiacheng Chen, Author

This thesis has been read and approved by the examining committee:

Professor Gang Tao, Thesis Advisor

Professor Zongli Lin

Professor Homa Alemzadeh

Accepted for the School of Engineering and Applied Science:

Dean, School of Engineering and Applied Science

May 2022

Acknowledgments

First, I want to express my sincere gratitude to my supervisor Gang Tao. His expertise in control theory helped me solve many problems, and his patience supported me through the process of writing this thesis. Even under the difficult pandemic situation, he always provided me with guidance. My thesis would never have been finished without his inspiration.

I would also like to thank my family for supporting me financially and emotionally. Their unconditional support gave me the courage to chase what I wanted, especially in this challenging pandemic period for everyone.

I want to thank my friend Guangzong Chen for giving me advice on writing the thesis and my fellow Qianhong Zhao, who is also in Professor Tao's group, for sharing resources with me and discussing control topics.

Abstract

The application of autonomous robots is drawing increasing attention in many fields. An autonomous robot can accomplish dangerous or tedious tasks that are difficult for humans, including aerial and underwater tasks. To achieve the task goals precisely and steadily, using a suitable control scheme is vital to an autonomous robot. Adaptive control, with its advantage in overcoming parametric uncertainties, is widely accepted as an advanced control method for robots.

Most research in this field has been focused on the adaptive control of underwater robots, but robot control in a varying environment remains an open problem. This thesis proposes a multiple-model-based adaptive control scheme to deal with the effect of the varying environment. The equations of motion of a robot manipulator are specifically derived. The complete model is based on the original model of a robot manipulator, with the effects of added mass, buoyancy, damping, drag, and lift considered as the varying environmental factors. Then a multiple-model-based control scheme is adopted to deal with the varying environment. Multiple controllers are being compared while controlling the robot. The controller with the slightest error is adopted to compensate for the varying environment effect, and the control performance can be better than a single-model controller. The control scheme is applicable for the robot moving in the air or any other fluid environments. Simulation results of the developed multiple-model adaptive controller on a planar elbow robot moving in the air and underwater are given to illustrate the improved control performance.

Contents

List of Figures

Chapter 1

Introduction

Robots are a kind of mechanical device that can achieve various tasks. A robot is generally integrated with knowledge from different fields such as mechanics, electronics, and information processing. Robots can have many forms, including robot manipulators, wheeled robots, aerial robots, etc. Among these forms, the robot manipulator is widely applied in welding, grabbing, surgery, and assembly line. To meet the requirements of the tasks, using a good control scheme is an essential part of designing a robot manipulator. As people's demand for autonomous robots is increasing, better control methods need to be carried out to guarantee the performance of the robot.

The control of underwater robots is a challenging task. The dynamics of such a system are nonlinear, coupled, and can be time-varying. In application, some system parameters may not be precisely known. In this case, a parameter estimator that can identify the parameters' uncertainty is necessary to optimize the control performance. Adaptive controllers are suitable for this problem. An adaptive controller consists of a controller and a parameter update law, which can guarantee the system output is as good as if the parameters were known.

This thesis will consider the case where the robot is moving in the wind or in/out of the water or some liquid. The effect of changing environment can be compensated by switching the robot estimation model for adequate control.

1.1 Literature Review

In this section, we introduce some contributions in robot manipulator control in a fluid environment, including the modeling of the robot, the adaptive control schemes, the control of underwater robots, and multiple-model based control schemes. By going through the literature, we can glance at the current research stage.

Modeling of robot manipulators under varying environments. To design a robot and analyze its stability, we need the mathematical model of the robot at first. For the robot manipulator, a detailed description of the dynamic models can be found in [25]. Here we mainly discuss the literature on aerial and underwater manipulators.

The dynamic model is different from the original model for the robot acting in the air or underwater. The aerial manipulator has prospered in the past decade. The work of carrying a manipulator on an unmanned aerial vehicle (UAV) was first proposed in 2012. In [27], Korpela et al. first brought the idea of combining a UAV and a 4-DOF manipulator. Following this, the modeling and control [31], the dynamic stability analysis [30] and model reference adaptive control [32] of the UAV-manipulator system were carried out. Disturbances are also an essential fact that needs to be considered. In [28], Lippielo and Ruggiero considered the external disturbances in the modeling and control of their research. However, the disturbances were not explicitly expressed. They were only regarded as an additional term. Besides the theoretical study, implementation was also done by researchers. In [36], Ore et al. showed an

experimental result of an aerial manipulator which is capable of doing water sampling tasks. The performance stability can be guaranteed in a wind environment up to 10m/s.

As for the underwater case, the modeling of a robot manipulator is much different from that in the atmosphere. In 1991, Janocha and Papadimitriou simulated and analyzed the difference between an underwater manipulator and an original one [6]. The simulation result showed that a more substantial torque is needed underwater, and the coupling is also higher among axes. Later in 1994, Levesque and Richard gave a detailed modeling method of robot manipulator link in the water, including the geometric description and forces exerted on the manipulator [10]. In [13], detailed modeling of underwater manipulator considering the added mass, buoyancy, current load, drag, and lift force was introduced by Schjolberg. The drag and lift force has attracted the most attention among the forces exerted on the robot manipulator. In 1998, Leabourne and Rock used strip theory to model the drag force exerted on a robot manipulator link [16]. They separated a robot link into small segments, computed the drag force on each segment, and added them together to get the total drag force. In the same year, McLain and Rock validated this modeling method through experiment [17]. Some researchers also obtain the drag force through an integral computation on the manipulator link [5] and [38].

Adaptive control of robot manipulators. Since the robot manipulator may have many unknown parameters in the application, the adaptive control scheme is commonly adopted. There are two different kinds of adaptive control schemes applied in the robot manipulator control field, namely direct adaptive control and indirect adaptive control. The difference lies in the error signal used to generate the parameter adaptation law. As its name suggests, direct adaptive control takes the joint position and velocity error to update parameters. In 1987, Craig et al. proposed an adaptive

control scheme for robot manipulators [1]. Their work combined adaptive control law with a computed-torque controller, and global convergence was achieved. Still, it required the knowledge of the measurement of the system acceleration. To avoid this, Slotine and Li added the first-order filter into the controller, and the global convergence can also be reached [2]. A case study was then carried out with the theoretical research [4]. Indirect adaptive control uses the error between the true and estimated parameters to generate the parameter update law. In 1988, Li and Slotine introduced an indirect robot manipulator control scheme, and exponential convergence of tracking errors and estimated parameters are proved [3].

With the support of theoretical results, many researchers have applied adaptive control schemes for aerial manipulators and underwater manipulators. In 2014, Antonelli and Cataldi did research on the adaptive control of arm-equipped quadrotors, where the wind disturbance was not considered [33]. In the same year, Caccavale et al. implemented an adaptive controller on a manipulator-UAV system [34]. However, their method requires measurement of the external disturbances. Research with a more detailed model was carried out in the underwater robot adaptive control field. In 1991, Fossen and Sagatun implemented an adaptive controller and a hybrid controller with an adaptive and sliding mode controller to the underwater robot system. The uncertainties of the vehicle thruster were considered, and a simulation study was done on an underwater vehicle. In the same year, Broome and Wang applied an adaptive parameter update law with a PID controller on an underwater manipulator. The PID gains are also updated during the control process. In 1998, Antonelli and Chiaverini researched the adaptive control of a vehicle-manipulator system where the hydrodynamic forces were considered [15]. The tracking error's asymptotic convergence and the parameter estimations' boundedness were proven. In 1999, Lee and Yuh applied a non-regressor based adaptive control to an underwater manipulator

[19], and later in 2000, the experimental validation was carried out [20]. Indirect adaptive control was also applied in the field of underwater robot control. In 2012, Mohan and Kim proposed an indirect adaptive control scheme on an underwater vehicle-manipulator system [29]. In their simulation, the pick and place operation and the drilling operation can be successfully achieved.

In addition, there is also research on the control of time-varying parameters. In [23], the adaptive control of a linear system with time-varying parameters is discussed. In 2014, impedance control of robots interacting with the environment was introduced [35]. Later in 2015, Wang et al. proposed critic learning for robot control in varying environments.

Multiple-model control of robot manipulators. In adaptive control, when the structure of the system remains identical, but the parameters of the dynamic model of the system change, it takes a period for the adaptation. Moreover, a huge error could be caused at the beginning of the adaptation. To solve this problem, a multiple-model and switching method was proposed. In 1994, Narendra and Balakrishnan developed an adaptive control scheme using multiple-model and switching, where the controller output was determined online to minimize the transient error [11]. The proof of global stability was given despite the chosen switching scheme. Later in 1997, a specific switching rule was proposed [14], suggesting that a cost function corresponding to the error of each model should be computed at every moment, and the controller is derived using the model with the least cost function. The simulation showed that the multiple-model based controller performs significantly better than the single-model based one.

The multiple-model and switching is suitable for the robot manipulator control, especially when the robot's operating environment varies. The multiple-model based adaptive control was first brought into the field of robot control in 1994 by Ciliz and

Narendra [9]. In their work, the robot manipulator's modeling and the algorithm's formulation were introduced. In 2006, Ciliz proposed a combined direct and indirect multiple-model adaptive control [24]. The simulation results showed that the combination improved the controller performance.

1.2 Research Problems

In the robot manipulator control literature, the aerial and underwater manipulator controls are mature. However, previous work has not addressed the situation when the robot manipulator is in a varying environment, for example, when the manipulator is moving in and out of the water. To achieve some sampling and manipulating tasks, the knowledge of canceling the effect is necessary. To achieve this, several problems need to be solved.

The modeling of the robot moving in varying environments. To achieve an optimal control objective, knowledge of the system dynamic model is necessary. However, for an object moving in a varying environment, various effects would influence the equations of motion, e.g., the added mass added inertia, drag, and buoyancy. These are unknown environmental factors. When we derive the equations of motion of a robot, these effects will be exerted on each link and will also impact other links. Thus, the modeling of the robot is complicated. In addition, to estimate the system and environment parameters, we need to write the equations of motion in a parameterized form.

The asymptotic tracking of the desired trajectory. In the past several decades, many researchers have focused on the control of robots, and the controller stability has been proven for different control methods. However, little attention has

been drawn to the situation where the environment varies. The asymptotic tracking of the robot under this situation remains a problem.

1.3 Research Motivation

Robotics has gained increasing attention in the past few decades. Robots are designed to help a human do different tasks automatically. For some tasks, we only want the robot to detect the environment. In such cases, an automatic mobile vehicle can reach the requirement. These vehicles include land vehicles, aerial platforms, underwater vehicles, etc. When the manipulator is working in strong wind, it has to reject the effect of the air. However, automatic mobile vehicles are only able to sense the environment but are not capable of interacting with it.

One way to interact with the environment is to carry a manipulator. In order to achieve different tasks, the manipulator may need to work in various environments. When it is mounted on an underwater vehicle, the hydrostatic and hydrodynamic forces should be well considered. There are also situations that the manipulator needs to work both in the air and underwater. When the task requires the manipulator to pick an object underwater and place it on the land or sample the water and analyze it on the land station, both the effect of the wind and the water should be taken into account. Robot manipulators operating in varying environments such as in and out of water and gust winds have additional dynamic uncertainties, which are characterized by additional dynamic functions and parameters.

Although existing literature has concerned about the adaptive control scheme of robot manipulators, there has still been little discussion on the multiple-model control schemes applied in this field. Our research goals are to study the characteristics of the additional dynamic uncertainties to establish the complete dynamic models of

robot manipulators operating in uncertain environments and to discuss the multiple-model adaptive control schemes for control of robot manipulators under the presence of additional uncertainties.

1.4 Thesis Outline

After analyzing the research problems, we show that the thesis research objectives are modeling the robot moving in a varying environment, controlling the robot, and dealing with the situation when the environment changes.

In Chapter 2, we introduce some related backgrounds in the robot modeling and adaptive control field, including the dynamic modeling of a robot manipulator, the adaptive control of robot manipulators, and the multiple-model control method. With this knowledge, we can better understand the control of the robot manipulator.

In Chapter 3, we discuss the effects on the robot manipulator in varying environments. These effects include the added mass, the added moment of inertia, friction and drag forces, and buoyancy. We first analyze these effects on an object. We use the Newton-Euler method to derive the dynamic equations of the whole robot by going through each link, and we give an example of the modeling of a planar elbow robot.

In Chapter 4, we apply single-model based adaptive control methods on the motion of a robot considering the environmental factors, including the direct adaptive control method and the indirect adaptive control method, as well as the stability analysis. The simulation results show that these adaptive control methods can achieve the asymptotic tracking of the robot trajectory.

In Chapter 5, with the knowledge from Chapter 4, we develop a multiple-model adaptive control method on the varying environment situation and compare the

multiple-model control method and the single-model control method.

In Chapter 6, we give the conclusions of the thesis and discuss the future research topics.

Chapter 2

Research Background

Before discussing the adaptive control, some background about robot manipulator control will be presented. In this chapter, the dynamic model, the adaptive control scheme, and the multiple-model based control scheme will be introduced first. Then, the research motivation will be addressed. The research problems will be presented in the last section of this chapter.

2.1 Dynamic Models of Robot Manipulators

In this section, we introduce the Denavit-Hartenberg convention for robot kinematics and the Newton-Euler formulation to derive the dynamic equations of the robot manipulator.

2.1.1 Kinematics

The kinematics of the robot is the description of the relationship between the joint values of the robot and the position/orientation of the robot. Given a robot manipulator with n joints, we first assume that each joint has only one degree of

freedom since we can decompose joints with two or more degrees of freedom into one degree of freedom joints with zero distance in between. Then the action of each joint can be represented by a single real number: the angle or distance of each joint. We denote this real number as joint variable q_i.

For the convenience of further computation, we introduce the numbering rules of the robot as shown in Figure 2.1. A robot with n joints will have $n + 1$ links, including the base, and each joint connects two links. For the joints, we name the joint which connects the base as the first joint, then the second to the n-th joint are named sequentially. For the links, we name the base as link 0. Then each joint i connects the link $i - 1$ and i. Thus, we consider the link i moves when the joint i is actuated.

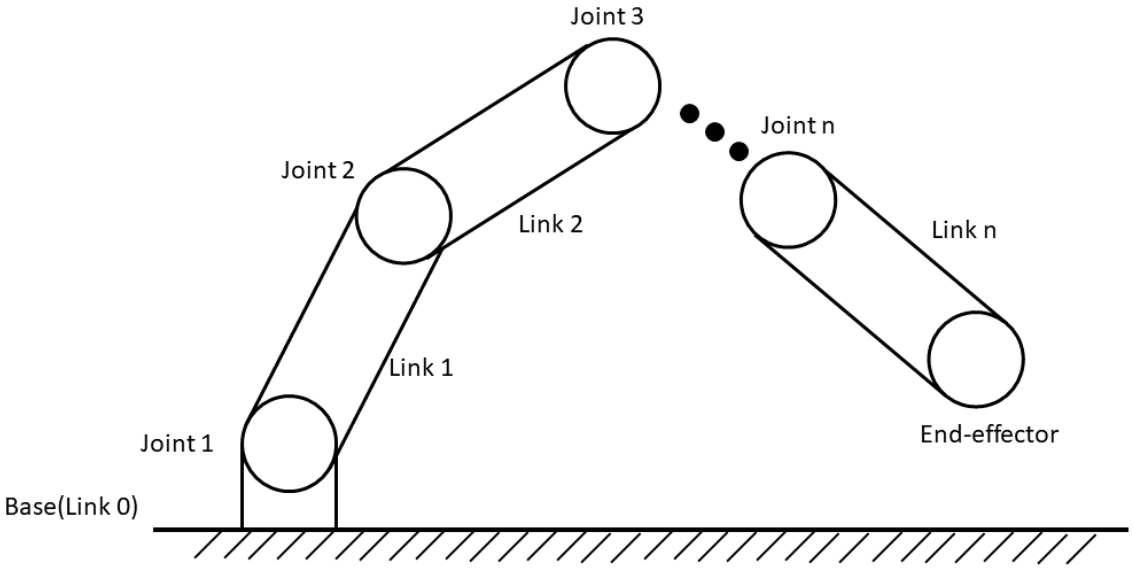

Figure 2.1: Joint and link numbering rules.

To simplify the analysis of the kinematics, we assign coordinate frames according to the Denavit-Hartenberg (DH) convention [39], where the i-th frame is fixed with the i-th link. This convention describes a robot link i with four parameters l_i, α_i, d_i and θ_i, namely the link length, link twist, link offset, and joint angle, respectively.

Suppose we consider the link's position between the i-th frame and the $(i+1)$-th frame. The link length is the distance between the two origins of each frame along the x_{i+1} direction, the link twist is the angle between the axis z_i and z_{i+1} measured in the plane normal to x_{i+1}, the link offset is the distance between two origins along the z_i axis and the joint angle is the angle between x_i and x_{i+1} measured in the plane normal to z_i.

The transformation between any two joints can be expressed by a matrix $\boldsymbol{T}_i = \begin{bmatrix} \boldsymbol{R}_i & \boldsymbol{o}_i \\ \boldsymbol{0} & 1 \end{bmatrix}$, namely the homogeneous transformation matrix, where $\boldsymbol{R}_i$ is a 3×3 matrix describing the orientation between the frame i and $i+1$, and $\boldsymbol{o}_i$ is a 3×1 vector describing the position between the frame i and $i+1$. Using the DH convention,

the homogeneous transformation matrix can be written as

$$
\begin{aligned}
\boldsymbol{T}_i &= \mathrm{Rot}_{z,\theta_i}\,\mathrm{Trans}_{z,d_i}\,\mathrm{Trans}_{x,a_i}\,\mathrm{Rot}_{x,\alpha_i} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix},
\end{aligned}
\tag{2.1}
$$

with

$$
\boldsymbol{R}_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} \end{bmatrix}, \boldsymbol{o}_i = \begin{bmatrix} a_i c_{\theta_i} \\ a_i s_{\theta_i} \\ d_i \end{bmatrix}. \tag{2.2}
$$

2.1.2 Dynamics

Typically, there are two ways of deriving the dynamic equations of the robot, namely the Euler-Lagrange method and the Newton-Euler method. The Euler-Lagrange method computes the dynamic equations through the robot's energy, while

the Newton-Euler method recursively computes the torques of the robot through the force and torque analysis. In this thesis, the Newton-Euler formulation is adopted to analyze the dynamics of the manipulator since the energy of the environment will have a significant influence on the robot energy, making it hard to obtain the energy of the robot. The Newton-Euler method consists of two parts, the forward recursion and the backward recursion, where the forward recursion computes the velocity, angular velocity, acceleration, and angular acceleration from the first link to the n-th link, and the backward recursion computes the force and torque of each joint from the n-th link to the first one.

Before presenting the recursion equations, several notations need to be introduced at first. We use superscript ${}^{k}(\cdot)$ and subscript $(\cdot)_k$ to represent a vector expressed in the k-th frame and the vector of link k, respectively. Then the forward recursion computation steps are [25]:

$$ {}^{k+1}\boldsymbol{\omega}_{k+1} = \boldsymbol{R}_k^{k+1}({}^{k}\boldsymbol{\omega}_k + \boldsymbol{z}_k\dot{q}_{k+1}) \tag{2.3} $$

$$ {}^{k+1}\boldsymbol{\alpha}_{k+1} = \boldsymbol{R}_k^{k+1}({}^{k}\boldsymbol{\alpha}_k + {}^{k}\,\boldsymbol{\omega}_k \times \boldsymbol{z}_k\dot{q}_{k+1} + \boldsymbol{z}_k\ddot{q}_{k+1}) \tag{2.4} $$

$$ {}^{k+1}\boldsymbol{v}_{k+1} = \boldsymbol{R}_k^{k+1}{}^{k}\boldsymbol{v}_k + {}^{k+1}\,\boldsymbol{\omega}_{k+1} \times {}^{k+1}\,\boldsymbol{d}_{k/k+1} \tag{2.5} $$

$$ {}^{k+1}\boldsymbol{v}_{k+1,c} = \boldsymbol{R}_k^{k+1}{}^{k}\boldsymbol{v}_k + {}^{k+1}\,\boldsymbol{\omega}_{k+1} \times {}^{k+1}\,\boldsymbol{d}_{k/k+1,c} \tag{2.6} $$

$$ {}^{k+1}\boldsymbol{a}_{k+1} = \boldsymbol{R}_k^{k+1}{}^{k}\boldsymbol{a}_k + {}^{k+1}\,\boldsymbol{\alpha}_{k+1} \times {}^{k+1}\,\boldsymbol{d}_{k/k+1} + {}^{k+1}\,\boldsymbol{\omega}_{k+1} \times ({}^{k+1}\boldsymbol{\omega}_{k+1} \times {}^{k+1}\,\boldsymbol{d}_{k/k+1}) \tag{2.7} $$

$$ {}^{k+1}\boldsymbol{a}_{k+1,c} = \boldsymbol{R}_k^{k+1}{}^{k}\boldsymbol{a}_k + {}^{k+1}\,\boldsymbol{\alpha}_{k+1} \times {}^{k+1}\,\boldsymbol{d}_{k/k+1,c} + {}^{k+1}\,\boldsymbol{\omega}_{k+1} \times ({}^{k+1}\boldsymbol{\omega}_{k+1} \times {}^{k+1}\,\boldsymbol{d}_{k/k+1,c}) \tag{2.8} $$

where $\boldsymbol{\omega}_k$ is the angular velocity of link k, $\boldsymbol{\alpha}_k$ is the angular acceleration of link k, $\boldsymbol{v}_k$ is the linear velocity of link k at the position of joint $k+1$, $\boldsymbol{v}_{k,c}$ is the linear velocity of the center of mass of link k, $\boldsymbol{a}_k$ is the linear acceleration of link k at the position of

joint $k+1$, $\boldsymbol{a}_{k,c}$ is the linear acceleration of the center of mass of link k, $\boldsymbol{d}_{k/k+1}$ is the vector from frame k to frame $k+1$, $\boldsymbol{d}_{k/k+1,c}$ is the vector from frame k to the center of mass of link $k+1$, $\boldsymbol{R}_k^{k+1}$ is the transformation matrix from frame k to frame $k+1$ and $\boldsymbol{z}_k$ is the unit vector along the z-axis of frame k.

The backward recursion computation steps are:

$$
\begin{aligned}
{}^k\boldsymbol{f}_k &= \boldsymbol{R}_k^{k+1}{}^{k+1}\boldsymbol{f}_{k+1} + \boldsymbol{F}_k - m_k{}^k\boldsymbol{g}_k && (2.9)\\
{}^k\boldsymbol{\tau}_k &= \boldsymbol{R}_k^{k+1}{}^{k+1}\boldsymbol{\tau}_{k+1} + \boldsymbol{d}_{k/k+1} \times \left(\boldsymbol{R}_k^{k+1}{}^{k+1}\boldsymbol{f}_{k+1}\right) + \boldsymbol{d}_{k/k+1,c} \times \boldsymbol{F}_k + \boldsymbol{T}_k && (2.10)\\
&\quad + \boldsymbol{d}_{k/k+1,c} \times (-m_k\boldsymbol{g}_k)
\end{aligned}
$$

where ${}^k\boldsymbol{f}_k$ and ${}^k\boldsymbol{\tau}_k$ represent the force and torque acting on the k-th link expressing in the frame fixed on the k-th link respectively, m_k is the mass of link k, $\boldsymbol{g}_k$ is the gravity vector of link k. The vector $\boldsymbol{F}_k$ are the forces acting at the center of mass of link k

$$\boldsymbol{F}_k = m_k^k\boldsymbol{a}_k \tag{2.11}$$

and the vector $\boldsymbol{T_k}$ are the moments acting at the center of mass of link k

$$\boldsymbol{T}_k = \boldsymbol{I}_k{}^k\boldsymbol{\alpha}_k + {}^k\boldsymbol{\omega}_k \times \boldsymbol{I}_k{}^k\boldsymbol{\omega}_k. \tag{2.12}$$

The process of recursively deriving the dynamic equations of the robot is described as follows. First, let us assume that the configuration of the robot is known, i.e., the link twist between two coordinate frames does not need to be estimated. Then the rotation matrix between any two link-attached frames is merely a function of the joint angle, denoted by $\boldsymbol{R}_i^j(q)$, which indicates the rotation matrix transforming a coordinate from frame i to frame j. For simplicity, we write it as $\boldsymbol{R}_i^j$ instead. Using (2.3), we can recursively get the equations of the angular velocities for each robot link.

The angular velocities consist of several terms, and each term is a multiplication of a rotation matrix, which is a function of $\boldsymbol{q}$, and a joint velocity vector containing $\dot{q}$. Using the knowledge of the angular velocities $\omega_k, k = 1, 2, \cdots, n$ and (2.4), we can obtain the angular accelerations α_i of each link i. The angular acceleration consists of two kinds of terms. One is the multiplication of a rotation matrix and a joint acceleration vector containing $\ddot{q}$. The other is the product of a rotation matrix and two joint angular velocity vectors' cross product. Substituting the angular velocities into (2.5) and (2.6), we can get the linear velocities of each link on the center of mass, which is the sum of terms in ω multiplying the link length parameters—in the same way, using (2.7) and (2.8), the linear acceleration of each link's center of mass can be obtained as a sum consisted with terms in the form of the product of link length parameters, rotation matrix, and joint angular acceleration vector or a cross product of two joint angular velocity vectors.

For the backward steps, the forces and the torques of each link are computed using the kinetic variables above. The force contains two parts. One is the product of the mass of the link and the acceleration, and the other is the gravity force which is a function of the joint angles. The torque of each link is computed by the cross products of the link length and forces and the product of the moment of inertia and the angular acceleration. Finally, the torque applied on each joint is the z-axis value of the torque vector. The detailed process of deriving the dynamic equations can be found in Chapter 3.

After going through the forward and the backward steps, each torque τ_i can be written as a sum of different terms, containing:

- Mass, link length, rotation matrix elements and $\ddot{q}_j$ term.
- Moment of inertia, rotation matrix elements and $\ddot{q}_j$ term.
- Mass, link length, rotation matrix elements and two of $\dot{q}_j$ term.

- Moment of Inertia, rotation matrix elements and $\dot{q}$ term.
- Mass, gravitational acceleration constant and rotation matrix elements.

Then we regroup the equations of the robot, i.e. write torques in a column vector. Extract the $\ddot{q}_j$ term from each torque i, we can get a mass matrix $\boldsymbol{M}(\boldsymbol{q})$ which is a function of mass, link length and joint positions, and a joint acceleration vector $\ddot{\boldsymbol{q}}$. In the same manner, we can get a coriolis and centripetal matrix $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ with a joint velocity vector $\dot{\boldsymbol{q}}$, and a gravity vector $\boldsymbol{g}(\boldsymbol{q})$.

Thus, the dynamic model of an n-link manipulator can be written as [25]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.13}$$

where $\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ is the vector of inertial forces and moments of the manipulator, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^n$ is the vector of Coriolis and centripetal effects of the manipulator, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the restoring vector of the manipulator and $\boldsymbol{\tau}$ is the control input vector. The detailed structure of these matrices can be found in [25].

2.1.3 Model of a Two-link Manipulator

Let us go through the modeling process of a two-link planar elbow robot manipulator. The geometry of the manipulator is shown in Figure 2.2. q_1 and q_2 are joint variables, l_1 and l_2 are the length of each link, l_{1c} and l_{2c} are the length from each joint to the center of the corresponding link, and the mass of the two links are m_1 and m_2, respectively.

To simplify the notation of the variables in the equations, we omit the superscript of each variable, and all variables are represented in the frame attached to their own

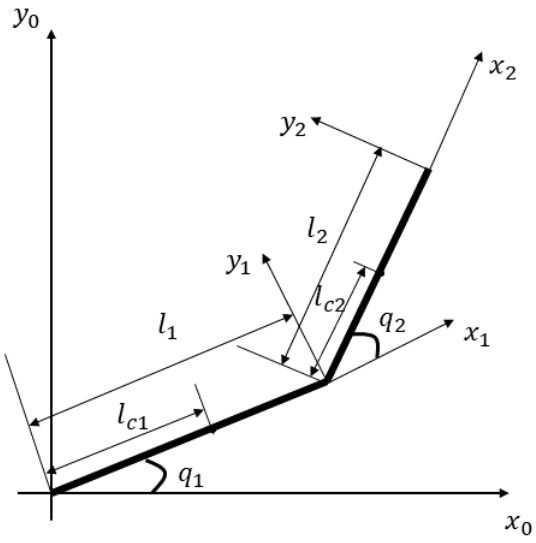

Figure 2.2: Planar elbow robot manipulator.

link. In the beginning, the initial conditions of the recursion should be settled as

$$\boldsymbol{\omega}_0 = \mathbf{0}, \boldsymbol{\alpha}_0 = \mathbf{0}, \boldsymbol{v}_0 = \mathbf{0}, \boldsymbol{a}_0 = \mathbf{0} \tag{2.14}$$

and

$$\boldsymbol{f}_3 = \mathbf{0}, \boldsymbol{\tau}_3 = \mathbf{0}. \tag{2.15}$$

The forward recursion is computed first to find the value of kinematic variables. Since the robot arm is planar, the angular velocity and angular acceleration can be easily obtained as

$$\boldsymbol{\omega}_1 = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_1 \end{pmatrix}, \boldsymbol{\omega}_2 = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_1 + \dot{q}_2 \end{pmatrix}, \boldsymbol{\alpha}_1 = \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_1 \end{pmatrix}, \boldsymbol{\alpha}_2 = \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_1 + \ddot{q}_2 \end{pmatrix}. \tag{2.16}$$

Then according to (2.5) and (2.6), the velocities can be computed as

$$\boldsymbol{v}_{1,c} = \boldsymbol{R}_1^0 \boldsymbol{v}_0 + \boldsymbol{\omega}_1 \times \boldsymbol{d}_{0/1,c} = \begin{pmatrix} 0 \\ l_{1c}\dot{q}_1 \\ 0 \end{pmatrix}, \tag{2.17}$$

$$\boldsymbol{v}_1 = \boldsymbol{R}_1^0 \boldsymbol{v}_0 + \boldsymbol{\omega}_1 \times \boldsymbol{d}_{0/1} = \begin{pmatrix} 0 \\ l_1\dot{q}_1 \\ 0 \end{pmatrix}, \tag{2.18}$$

$$\boldsymbol{v}_{2,c} = \boldsymbol{R}_2^1 \boldsymbol{v}_1 + \boldsymbol{\omega}_2 \times \boldsymbol{d}_{1/2,c} = \begin{pmatrix} l_1 \sin(q_2)\dot{q}_1 \\ l_{2c}(\dot{q}_1 + \dot{q}_2) + l_1 \cos(q_2)\dot{q}_1 \\ 0 \end{pmatrix}. \tag{2.19}$$

Finally, the linear acceleration of each link and joint can be obtained according to (2.8) as

$$\boldsymbol{a}_{1,c} = \boldsymbol{R}_1^0 \boldsymbol{a}_0 + \boldsymbol{\alpha}_1 \times \boldsymbol{d}_{0/1,c} + \boldsymbol{\omega}_1 \times (\boldsymbol{\omega}_1 \times \boldsymbol{d}_{0/1,c}) = \begin{pmatrix} -l_{1c}\dot{q}_1^2 \\ l_{1c}\ddot{q}_1 \\ 0 \end{pmatrix}, \tag{2.20}$$

$$\boldsymbol{a}_1 = \boldsymbol{R}_1^0 \boldsymbol{a}_0 + \boldsymbol{\alpha}_1 \times \boldsymbol{d}_{0/1} + \boldsymbol{\omega}_1 \times (\boldsymbol{\omega}_1 \times \boldsymbol{d}_{0/1}) = \begin{pmatrix} -l_1\dot{q}_1^2 \\ l_1\ddot{q}_1 \\ 0 \end{pmatrix}, \tag{2.21}$$

$$\begin{aligned}\boldsymbol{a}_{2,c} &= \boldsymbol{R}_2^1\boldsymbol{a}_1 + \boldsymbol{\alpha}_1 \times \boldsymbol{d}_{1/2,c} + \boldsymbol{\omega}_1 \times (\boldsymbol{\omega}_1 \times \boldsymbol{d}_{1/2,c}) \\ &= \begin{pmatrix} l_1 \sin(q_2)\ddot{q}_1 - l_{2c}(\dot{q}_1 + \dot{q}_2)^2 - l_1 \sin(q_2)\dot{q}_1^2 \\ l_1 \sin(q_2)\dot{q}_1^2 + l_{2c}(\ddot{q}_1 + \ddot{q}_2) + l_1 \cos(q_2)\ddot{q}_1 \\ 0 \end{pmatrix}. \end{aligned} \tag{2.22}$$

The accelerations are what we need to compute the force and torque of each link. Substituting the results into (2.11) and (2.12) leads to

$$\boldsymbol{F}_2 = m_2\boldsymbol{a}_{2,c} = m_2 \begin{pmatrix} l_1 \sin(q_2)\ddot{q}_1 - l_{2c}(\dot{q}_1 + \dot{q}_2)^2 - l_1 \sin(q_2)\dot{q}_1^2 \\ l_1 \sin(q_2)\dot{q}_1^2 + l_{2c}(\ddot{q}_1 + \ddot{q}_2) + l_1 \cos(q_2)\ddot{q}_1 \\ 0 \end{pmatrix} \tag{2.23}$$

$$\boldsymbol{T}_2 = \boldsymbol{I}_2\boldsymbol{\alpha}_2 + \boldsymbol{\omega}_2 \times \boldsymbol{I}_2\boldsymbol{\omega}_2 = \begin{pmatrix} 0 \\ 0 \\ I_2(\ddot{q}_1 + \ddot{q}_2) \end{pmatrix} \tag{2.24}$$

$$\boldsymbol{F}_1 = m_1\boldsymbol{a}_{1,c} = m_1 \begin{pmatrix} -l_{1c}\dot{q}_1^2 \\ l_{1c}\ddot{q}_1 \\ 0 \end{pmatrix} \tag{2.25}$$

$$\boldsymbol{T}_1 = \boldsymbol{I}_1\boldsymbol{\alpha}_1 + \boldsymbol{\omega}_1 \times \boldsymbol{I}_1\boldsymbol{\omega}_1 = \begin{pmatrix} 0 \\ 0 \\ I_1\ddot{q}_1 \end{pmatrix}. \tag{2.26}$$

Substitute the initial conditions (2.15) into the equation (2.9), we can get

$$\begin{aligned}\boldsymbol{f}_2 &= \boldsymbol{R}_2^3\boldsymbol{f}_3 + \boldsymbol{F}_2 - m_2\boldsymbol{g}_2 \\ &= \begin{pmatrix} m_2 g\sin(q_1+q_2) - m_2(l_{2c}(\dot{q}_1+\dot{q}_2)^2 - l_1\sin(q_2)\ddot{q}_1 + l_1\sin(q_2)\dot{q}_1^2) \\ m_2 g\cos(q_1+q_2) + m_2(l_1\sin(q_2)\dot{q}_1^2 + l_{2c}(\ddot{q}_1+\ddot{q}_2) + l_1\cos(q_2)\ddot{q}_1) \\ 0 \end{pmatrix}\end{aligned} \tag{2.27}$$

and

$$\boldsymbol{f}_1 = \boldsymbol{R}_1^2\boldsymbol{f}_2 + \boldsymbol{F}_1 - m_1\boldsymbol{g}_1 \tag{2.28}$$

which leads to

$$\begin{aligned} f_{1,x} &= (m_1+m_2)gsin(q_1) - l_1 m_2\dot{q}_1^2 - l_{1c}m_1\dot{q}_1^2 \\ &\quad - m_2 l_{2c}\sin(q_2)(\ddot{q}_1+\ddot{q}_2) - m_2 l_{2c}\cos(q_2)(\dot{q}_1+\dot{q}_2)^2 \end{aligned} \tag{2.29}$$

$$\begin{aligned} f_{1,y} &= (m_1+m_2)g\cos(q_1) + m_2 g\cos(q_1) + m_2 l_1\ddot{q}_1 + m_1 l_{1c}\ddot{q}_1 \\ &\quad + l_{2c}m_2\cos(q_2)(\ddot{q}_1+\ddot{q}_2) - m_2 l_{2c}\sin(q_2)(\dot{q}_1+\dot{q}_2)^2. \end{aligned} \tag{2.30}$$

Substituting $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ into (2.10), the torque of each joint can be finally obtained as

$$\begin{aligned} \boldsymbol{\tau}_2 &= (\ddot{q}_1(m_2 l_{2c}^2 + m_2 l_1 l_{2c}\cos(q_2) + I_2) \\ &\quad + \ddot{q}_2(m_2 l_1 l_{2c}\cos(q_2) + I_2) \\ &\quad + \dot{q}_1(m_2 l_1 l_{2c}\sin(q_2)\dot{q}_1) \\ &\quad + m_2 l_{2c} g\cos(q_1+q_2))k \end{aligned} \tag{2.31}$$

and

$$\begin{aligned}\boldsymbol{\tau}_1 = &(\ddot{q}_1(m_1 l_{1c}^2 + m_2 l_{2c}^2 + m_2 l_1^2 + 2m_2 l_1 l_{2c}\cos(q_2) + I_1 + I_2) \\ &+ \ddot{q}_2(m_2 l_{2c}^2 + m_2 l_1 l_{2c}\cos(q_2) + I_2) \\ &+ \dot{q}_1(-m_2 l_1 l_{2c}\sin(q_2)\dot{q}_2) \\ &+ \dot{q}_2(-m_2 l_1 l_{2c}\sin(q_2)(\dot{q}_1 + \dot{q}_2)) \\ &+ (m_1 l_{1c} + m_2 l_1) g\cos(q_1) + m_2 l_{2c} g\cos(q_1 + q_2))k\end{aligned} \tag{2.32}$$

where k is the unit vector on the z-axis direction of the corresponding frame of each torque.

2.1.4 Properties

There are several important properties of the robot manipulator dynamics equation. Proof of these properties are shown in [25].

Property 2.1. *There exists an $n \times l$ function $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ and an l dimensional vector $\boldsymbol{\theta}$ such that the rigid robot dynamic equation can be written as*

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}. \tag{2.33}$$

The function $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ is called the regressor and $\boldsymbol{\theta} \in \mathbb{R}^l$ is the parameter vector. The number of parameters needed to write the dynamics is not unique.

From the above description in Section 2.1.2, we can see that the each torque is a sum of terms which are products of joint variables $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ and system parameters (mass, link length, moment of inertia, gravity acceleration). Suppose there are a total of l different combinations of system parameters among n torques, then for each torque, we can write it in the form of a dot product of two vectors, where the joint

variables and the system parameters are in the two vectors correspondingly.

Property 2.2. *The mass matrix for an n-link rigid robot is symmetric and positive definite. Specifically, for a fixed value of the generalized coordinate q, let* $0 < \lambda_1(q) \leq \lambda_2(q) \leq \cdots \leq \lambda_n(q)$ *denote the n eigenvalues of* $\boldsymbol{M(q)}$*, then the inertia matrix* $\boldsymbol{M}(\boldsymbol{q})$ *satisfies*

$$\lambda_1(q)\boldsymbol{I}_{n\times n} \leq \boldsymbol{M}(\boldsymbol{q}) \leq \lambda_n(q)\boldsymbol{I}_{n\times n}.$$

In addition, from basic linear algebra knowledge, we know that a positive definite matrix is invertible. Thus, the matrix $\boldsymbol{M}(\boldsymbol{q})$ is invertible.

2.1.5 Parameterization of the Two-link Manipulator Model

In this section, we show the parameterized form of the two-link manipulator. From the two-link manipulator dynamic equations (2.32) and (2.31), we can write them in the form of

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \\ \tau_2 \end{bmatrix} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}^* \tag{2.34}$$

where

$$\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \begin{pmatrix} \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 & \ddot{q}_1 & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 & 2\cos(q_2)\ddot{q}_1 + \cos(q_2)\ddot{q}_2 + \sin(q_2)\dot{q}_1^2 \\ \ddot{q}_1 + \ddot{q}_2 & 0 & 0 & \ddot{q}_2 & 0 & \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 \end{pmatrix}$$

$$\begin{pmatrix} \cos^2(q_2)\ddot{q}_1 + \sin(q_2)\cos(q_2)\dot{q}_1^2 & \sin^2(q_2)\ddot{q}_1 - \sin(q_2)\cos(q_2)\dot{q}_1^2 \\ 0 & 0 \\ -\sin(q_2)\dot{q}_1^2 - 2\sin(q_2)\dot{q}_1\dot{q}_2 - \sin(q_2)\dot{q}_2^2 & \cos(q_1) \quad \cos(q_1+q_2) \\ 0 & 0 \quad \cos(q_1+q_2) \end{pmatrix} \tag{2.35}$$

$$\boldsymbol{\theta}^* = \Big(\; m_2 l_{2c}^2 \quad m_1 l_{1c}^2 \quad I_{m1} \quad I_{m2} \quad 0 \quad m_2 l_1 l_{2c} \quad m_2 l_1^2 \quad m_2 l_1^2$$
$$m_2 l_1 l_{2c} \quad (m_2 l_1 + m_1 l_{1c} - l_1 l_2 \rho \pi r_2^2 - l_1 l_{1c} \rho \pi r_1^2) g \tag{2.36}$$
$$l_{2c}(m_2 - l_2 \rho \pi r_2^2) g \; \Big)^T .$$

2.2 Adaptive Control of Robot Manipulators

Adaptive control is a control algorithm that can estimate the system parameters, which may vary or be unknown, while reaching the control goal simultaneously so that the system output can track the desired output asymptotically, canceling the effect of the unknown system parameters. In the adaptive control of robot manipulators, the control objective is that given a bounded desired robot joint trajectory $\boldsymbol{q}_d(t)$ and its bounded derivatives $\dot{\boldsymbol{q}}_d(t)$ and $\ddot{\boldsymbol{q}}_d(t)$, design a feedback control input signal $\boldsymbol{u}$ for the robot manipulator system without the knowledge of the system parameters in $\boldsymbol{M}(q)$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{g}(\boldsymbol{q})$, so that all signals in the closed-loop system are bounded and the joint position $\boldsymbol{q}(t)$ tracks $\boldsymbol{q}_d(t)$ asymptotically.

Preliminaries. In order to analyse the stability of the control system, we need to introduce several definitions and properties at first [22].

Definition 2.1. *A vector* $x(t) \in \mathbb{R}^n$ *belongs to the signal space* L^2 *if* $\int_0^\infty (x_1^2(t) + x_2^2(t) + ... + x_n^2(t))\, dt < \infty$.

Definition 2.2. *A vector* $x(t) \in \mathbb{R}^n$ *belongs to the signal space* L^∞ *if* $\sup_{t \geq 0} \max_{1 \leq i \leq n} |x_i(t)| < \infty$.

Lemma 2.1. *If* $\boldsymbol{x}(t) \in L^2$ *and* $\dot{\boldsymbol{x}}(t) \in L^\infty$, *then* $\lim_{t \to \infty} \boldsymbol{x}(t) = 0$.

Then we introduce two kinds of adaptive control methods on the robot, namely the direct adaptive control and the indirect adaptive control. The direct adaptive control

method uses the tracking error $\boldsymbol{e}(t)$ to generate the adaptive parameter update law. For the indirect adaptive control method, the adaptive parameter update law is driven by the prediction error, which is the difference between the predicted torque and the real torque applied on joints.

2.2.1 Direct Adaptive Control

Using the fact that the inertia matrix is invertible, an adaptive control algorithm based on the concept of feedback linearization is proposed. The control law $\boldsymbol{u}$ is

$$\boldsymbol{u} = \hat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{a}_q + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}(\boldsymbol{q}) \tag{2.37}$$

where the notation $\hat{(\cdot)}$ represents the estimated value of $(\cdot)$, the error of the mismatch is denoted as $\tilde{(\cdot)} = \hat{(\cdot)} - (\cdot)$, and

$$\boldsymbol{a}_q = \ddot{\boldsymbol{q}}_d(t) - \boldsymbol{K}_1(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d) - \boldsymbol{K}_0(\boldsymbol{q} - \boldsymbol{q}_d) \tag{2.38}$$

where $\boldsymbol{K}_0$, $\boldsymbol{K}_1$ are diagonal matrices with diagonal elements consisting of position and velocity gains, respectively.

Using the linear parameterization property, we can get

$$\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_1\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_0\tilde{\boldsymbol{q}} = \hat{\boldsymbol{M}}^{-1}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{\theta}} \tag{2.39}$$

where $\boldsymbol{Y}$ is the regressor function and $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$, where $\hat{\boldsymbol{\theta}}$ is the estimation of the parameter vector $\boldsymbol{\theta}$. Writing (2.39) in the state space form, we can get

$$\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}\boldsymbol{\phi}\tilde{\boldsymbol{\theta}} \tag{2.40}$$

where

$$\boldsymbol{e} = \begin{bmatrix} \tilde{\boldsymbol{q}} \\ \dot{\tilde{\boldsymbol{q}}} \end{bmatrix} \tag{2.41}$$

is the joint space error, and

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{n\times n} & \boldsymbol{I}_{n\times n} \\ -\boldsymbol{K}_0 & -\boldsymbol{K}_1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0}_{n\times n} \\ \boldsymbol{I}_{n\times n} \end{bmatrix}, \boldsymbol{\phi} = \hat{\boldsymbol{M}}^{-1}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}). \tag{2.42}$$

Let $\boldsymbol{Q}$ be a matrix satisfying $\boldsymbol{Q} = \boldsymbol{Q}^T > 0$. Since $\boldsymbol{K}_0$ and $\boldsymbol{K}_1$ are diagonal matrices of positive elements, the matrix $\boldsymbol{A}$ is stable. Let $\boldsymbol{P}$ be the solution of the Lyapunov equation

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q} \tag{2.43}$$

then set the adaptive update law as

$$\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}^{-1}\boldsymbol{\phi}^T\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{e} \tag{2.44}$$

where $\boldsymbol{\Gamma}$ is a positive definite constant symmetric matrix.

Theorem 2.1. *For the robot system described as (2.13) with the system parameters unknown, using the control law (2.37) and the adaptive parameter update law (2.44), it can be guaranteed that the parameter estimates are bounded and* $\lim_{t\to\infty} \boldsymbol{e}(t) = 0$.

Proof. Using the Lyapunov function

$$V = \boldsymbol{e}^T\boldsymbol{P}\boldsymbol{e} + \tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}\tilde{\boldsymbol{\theta}}, \tag{2.45}$$

we can show that the tracking error will converge globally to zero while all internal

signals are bounded. The derivative of the Lyapunov function V is

$$\dot{V} = -\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} + 2\tilde{\boldsymbol{\theta}}^T\{\boldsymbol{\phi}^T\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{e} + \boldsymbol{\Gamma}\dot{\hat{\boldsymbol{\theta}}}\} \tag{2.46}$$

then substitute the adaptive control law(2.44) into(2.46) we can get

$$\dot{V} = -\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} \tag{2.47}$$

thus we have

$$\boldsymbol{e}(t) \in L^\infty, \boldsymbol{e}(t) \in L^2, \dot{\boldsymbol{e}}(t) \in L^\infty, \tilde{\boldsymbol{\theta}} \in L^\infty \tag{2.48}$$

which indicates that the parameter estimation error remains bounded and the position tracking error can converge to zero asymptotically. [37] □

2.2.2 Indirect Adaptive Control

For the indirect adaptive control on the robot, the parameter estimation and the control input are divided into two parts. The estimated parameters are used in the dynamic equations to predict the torques on the joints, and the actual torques on the joints are obtained by sensors. The difference between the predicted torques and the sensed torques is computed, namely the prediction error. Compared to the direct adaptive control method, this prediction error is used to generate the parameter update law instead of the joint tracking error.

The predicted torques are computed as

$$\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}(\boldsymbol{q}) = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\hat{\boldsymbol{\theta}}. \tag{2.49}$$

Then the torque prediction errors are

$$\begin{aligned}\tilde{\boldsymbol{\tau}} &= \hat{\boldsymbol{\tau}} - \boldsymbol{\tau} \\ &= \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta} - \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\hat{\boldsymbol{\theta}} \\ &= \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{\theta}}.\end{aligned} \tag{2.50}$$

The adaptive update law is chosen as

$$\dot{\tilde{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\boldsymbol{Y}^T(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{\tau}} \tag{2.51}$$

where $\boldsymbol{\Gamma}$ is a symmetric positive definite coefficient matrix.

The control input has the same structure as in (2.37). Then the stability of this method can be proved as follows [12].

Theorem 2.2. *The adaptive control scheme (2.49)-(2.51) guarantees boundedness for all closed-loop signals and* $\lim_{t\to\infty} \boldsymbol{e}(t) = 0$.

Proof. Considering the Lyapunov function

$$V = \frac{1}{2}\tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}}, \tag{2.52}$$

taking its derivative

$$\dot{V} = \frac{1}{2}\dot{\tilde{\boldsymbol{\theta}}}^T\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}} + \frac{1}{2}\tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\theta}}}, \tag{2.53}$$

and substituting the adaptive parameter update law (2.51) in (2.53), we get

$$\begin{aligned}\dot{V} &= \frac{1}{2}\left(\left(-\boldsymbol{\Gamma}\boldsymbol{Y}^T(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{\tau}}\right)^T\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}} + \tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\left(-\boldsymbol{\Gamma}\boldsymbol{Y}^T(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{\tau}}\right)\right) \\ &= -\tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\left(\boldsymbol{\Gamma}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{\tau}}\right) \\ &= -\tilde{\boldsymbol{\tau}}^T\tilde{\boldsymbol{\tau}} \leq 0.\end{aligned} \tag{2.54}$$

Thus, we can conclude that

$$\tilde{\boldsymbol{\theta}} \in L^{\infty}, \tilde{\boldsymbol{\tau}} \in L^{2}. \tag{2.55}$$

Since $\tilde{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}} - \boldsymbol{\tau}$, and we have the equations (2.49) and (2.37) for $\hat{\boldsymbol{\tau}}$ and $\boldsymbol{\tau}$, the prediction error can be derived as

$$\tilde{\boldsymbol{\tau}} = \hat{\boldsymbol{M}}(\boldsymbol{q})(\ddot{\boldsymbol{e}} + \boldsymbol{K}_D\dot{\boldsymbol{e}} + \boldsymbol{K}_P\boldsymbol{e}), \tag{2.56}$$

which indicates that

$$\boldsymbol{e} = (s^2\boldsymbol{I} + s\boldsymbol{K}_D + \boldsymbol{K}_P)\hat{\boldsymbol{M}}^{-1}(\boldsymbol{q})\tilde{\boldsymbol{\tau}}. \tag{2.57}$$

Since $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q}_d$ and $\boldsymbol{q}_d$, $\dot{\boldsymbol{q}}_d$ are bounded, we have

$$\boldsymbol{e}, \dot{\boldsymbol{e}} \in L^{2}, \boldsymbol{e} \in L^{\infty}. \tag{2.58}$$

□

2.3 Multiple-model Control of Robot Manipulators

In this section, we introduce the Multiple-model based adaptive control for robot manipulators. The main idea of this control scheme is to have multiple models with different dynamic structures, and the initial estimates of the system parameters are different. Suppose there are total N models, then the i-th model uses the i-th parameter estimate to compute the corresponding torque prediction. We choose the closest approximation to be the model of computing the actual controller.

In practice, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\boldsymbol{\tau}$ can be detected by sensors, and the joint acceleration $\ddot{\boldsymbol{q}}$ can be derived from $\dot{\boldsymbol{q}}$. Thus, for each model, we can individually obtain the torque prediction computed from joint variables and parameter estimates, and then compute the torque prediction errors for each model. The predictions are computed as

$$\hat{\boldsymbol{\tau}}_i = \boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\hat{\boldsymbol{\theta}}_i \tag{2.59}$$

where $\boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ is the regressor for the i-th model and $\hat{\boldsymbol{\theta}}_i$ is the parameter estimate vector for the i-th model. Since the actual torques are obtained by sensors, we can compute the predictions as

$$\tilde{\boldsymbol{\tau}}_i = \hat{\boldsymbol{\tau}} - \boldsymbol{\tau}. \tag{2.60}$$

We then choose the adaptive parameter update law for the i-th model as

$$\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}_i \boldsymbol{Y}_i^T(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{\tau}}_i, \tag{2.61}$$

where $\boldsymbol{\Gamma}_i$ is the coefficient matrix for the i-th model satisfying $\boldsymbol{\Gamma}_i = \boldsymbol{\Gamma}_i > 0$, and let the initial estimate for the i-th parameter vector in the parameter space $\mathbb{R}^l$ be $\boldsymbol{\theta}_{i0}$. The control signal for each model is computed as

$$\boldsymbol{u}_i = \hat{\boldsymbol{M}}_i(\boldsymbol{q})\boldsymbol{a}_{qi} + \hat{\boldsymbol{C}}_i(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}_i(\boldsymbol{q}) \tag{2.62}$$

where

$$\boldsymbol{a}_{qi} = \ddot{\boldsymbol{q}}_d - \boldsymbol{K}_{Di}\dot{\boldsymbol{e}} - \boldsymbol{K}_{Pi}\boldsymbol{e} \tag{2.63}$$

and $\boldsymbol{K}_{Di}$ and $\boldsymbol{K}_{Pi}$ are diagonal positive definite coefficient matrices for the model i. From Theorem 2.2, we know that for each model i, the computed controller i can achieve asymptotic tracking of the desired trajectory.

The reason for using a multiple-model adaptive controller is to reduce the transient error. For the single model case, since we have no information for the true values of system parameters, the prediction error may be huge and take a long time for the estimation. However, for the multiple-model case, we can have multiple initial estimates for the parameters, and we can choose the model with the least prediction error to generate the control signal. From the prediction errors, we have a function to decide which model estimate we should use to generate the controller. Generally, the function is named as a performance index, and is defined as [9]

$$J\left(\tilde{\boldsymbol{\tau}}_i(t)\right) = \gamma \tilde{\boldsymbol{\tau}}_i^T(t)\tilde{\boldsymbol{\tau}}_i(t) + \beta \int_0^t \tilde{\boldsymbol{\tau}}_i^T(t)\tilde{\boldsymbol{\tau}}_i(\zeta)d\zeta \quad \text{with } \gamma, \beta > 0 \tag{2.64}$$

where $\tilde{\boldsymbol{\tau}}_i(t) = \hat{\boldsymbol{\tau}}_i(t) - \boldsymbol{\tau}_i(t)$ is the torque prediction error vector, $\hat{\boldsymbol{\tau}}_i(t)$ is the torque prediction, $\gamma > 0$ and $\beta > 0$ are weights to be tuned in practice. Then, at the start of the control process, the performance index can be obtained, and the model with the least performance index under the current situation will be chosen.

Chapter 3

Modeling of Robot Manipulators in Varying Environments

The modeling of a robot manipulator moving in wind or water is shown in this chapter. We first introduce the effects of fluid on a robot link. Then we use the Newton-Euler formulation to derive the dynamic equations for the whole robot manipulator. Most robot manipulators consist of rectangular or cylindrical links. In this thesis, we discuss the cylindrical case and assume that the robot is operating in an ideal fluid.

3.1 Fluid Effects on a Robot Manipulator Link

The motion of the rigid body driven by forces in fluid causes the fluid to give forces and moments proportional to the rigid body acceleration. These effects are called the added mass and the added moment of inertia. The fluid friction forces are denoted as drag forces. They are acting along the direction of the relative velocity between the link and the fluid velocity v_f. The motion will also cause vortex shedding around the rigid body, and this results in the lift forces, which are in the direction

orthogonal with the drag forces. The displacement of water exerts buoyant forces on the manipulator, attacking in the gravity center of displaced water and in the opposite direction of the gravity force. These forces are shown in Figure 3.1.

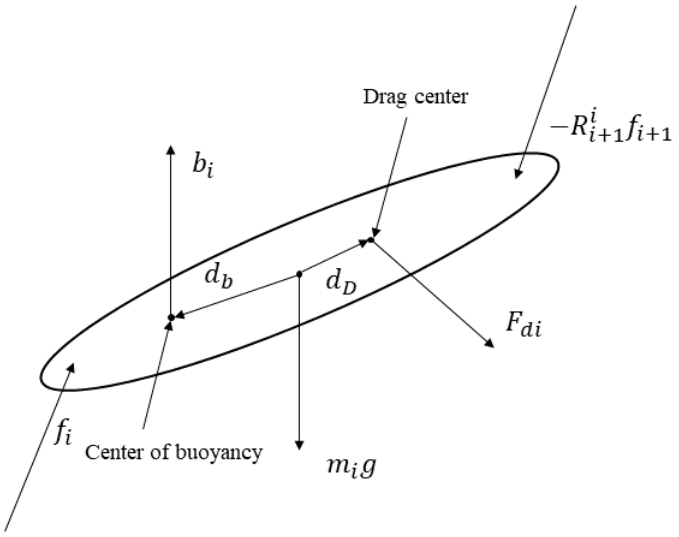

Figure 3.1: Fluid forces on a rigid body.

3.1.1 Added Mass and Added Moment of Inertia

The added mass and added moment of inertia of a rigid body are denoted as $\boldsymbol{Aa}$, where $\boldsymbol{A}$ is the coefficient matrix, and $\boldsymbol{a}$ is the acceleration of the rigid body. For a cylinder, the cross elements are small due to the symmetry of the cylinder, so they can be ignored. The coefficient values are

$$\boldsymbol{A}_{11} = 0.1m, \qquad \boldsymbol{A}_{22} = \pi\rho r^2 l, \qquad \boldsymbol{A}_{33} = \pi\rho r^2 l$$
$$\boldsymbol{A}_{44} = 0, \qquad \boldsymbol{A}_{55} = \frac{1}{12}\pi\rho r^2 l^3, \qquad \boldsymbol{A}_{66} = \frac{1}{12}\pi\rho r^2 l^3$$

where ρ is the fluid density, r is the radius of a link, l is the length of the link and m is the mass of the link. The directions of the added mass on a cylinder is shown in Figure 3.2.

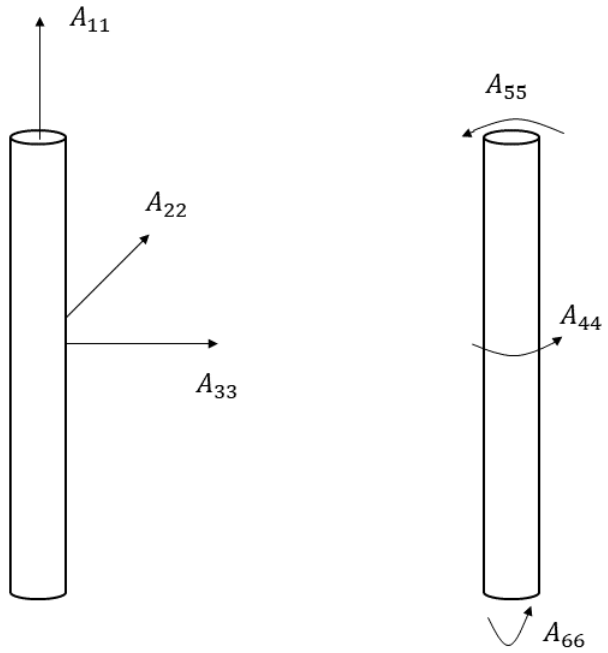

Figure 3.2: Added mass on a cylinder.

Then the total mass $\boldsymbol{M}_k$ of the link including the added mass is

$$\boldsymbol{M}_k = diag(\boldsymbol{A}_{11}, \boldsymbol{A}_{22}, \boldsymbol{A}_{33}) + m\boldsymbol{I} \tag{3.1}$$

where $\boldsymbol{I}$ is the identity matrix, and the total moment of inertia $\boldsymbol{I}_k$ including the added moment of inertia is

$$\boldsymbol{I}_k = diag(\boldsymbol{A}_{44}, \boldsymbol{A}_{55}, \boldsymbol{A}_{66}) + \boldsymbol{I}_m \tag{3.2}$$

where $\boldsymbol{I}_m$ is the inertia of the link.

3.1.2 Drag and Lift

The total hydrodynamic drag force can be approximated as a nonlinear expansion

$$\boldsymbol{F}_D = \boldsymbol{D}_s\boldsymbol{v}_r + \boldsymbol{D}_r|\boldsymbol{v}_r|\boldsymbol{v}_r + \boldsymbol{D}_q\boldsymbol{v}_r^2 + o(\boldsymbol{v}_r^3) \tag{3.3}$$

where $\boldsymbol{D}_s$, $\boldsymbol{D}_r$ and $\boldsymbol{D}_q$ are friction coefficients and $\boldsymbol{v}_r$ is the relative velocity between the rigid body and the fluid. In the cylinder case, due to the symmetry, the third term can be canceled, and higher order terms are small compared to the second term so that it can be ignored.

In order to express the direction of the fluid velocity, we define an axis system($\boldsymbol{x}_f$, $\boldsymbol{y}_f$, $\boldsymbol{z}_f$) such that the fluid velocity is along the x-axis of the coordinate frame, which means that the drag force is along the same direction, and the lift force is along the y-axis. The z-axis is placed to satisfy the right-hand rule. The transformation between the flow frame and the link fixed frame($\boldsymbol{x}_k$, $\boldsymbol{y}_k$, $\boldsymbol{z}_k$) can be achieved with two rotations: rotation α about the y_k axis and then a rotation β about the new z_k axis. The transformation matrix can be written as

$$\boldsymbol{R}_f^k = \boldsymbol{R}_z(-\beta)\boldsymbol{R}_y(\alpha) \tag{3.4}$$

$$= \begin{bmatrix} \cos(\beta)\cos(\alpha) & \sin(\beta) & \cos(\beta)\sin(\alpha) \\ -\sin(\beta)\cos(\alpha) & \cos(\beta) & -\sin(\beta)\sin(\alpha) \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \tag{3.5}$$

then the relative velocity $\boldsymbol{v}_r$ can be expressed as

$$\boldsymbol{v}_r = \boldsymbol{v} - v_f \boldsymbol{R}_k^f \boldsymbol{x}_f = [\boldsymbol{v}_{rx}, \boldsymbol{v}_{ry}, \boldsymbol{v}_{rz}]^T \tag{3.6}$$

where $\boldsymbol{v}$ is the translational velocity expressed in the link fixed frame, v_f is the velocity value of the fluid, and $\boldsymbol{x}_f$ is the unit vector along the $\boldsymbol{x}_f$ axis.

After getting the knowledge of the expression of the relative velocity, we are able to write the total drag force. The linear term in(3.3) is

$$\boldsymbol{F}_s = \boldsymbol{D}_s \boldsymbol{v}_r \tag{3.7}$$

where $\boldsymbol{D}_s$ is a 3×3 diagonal matrix containing linear friction coefficients, and the quadratic term can be approximated as [26]

$$\boldsymbol{F}_d = (\boldsymbol{D}_D + \boldsymbol{D}_L)|\boldsymbol{v}_r|^T \boldsymbol{v}_r \tag{3.8}$$

where $\boldsymbol{D}_D$ and $\boldsymbol{D}_L$ are 3×3 diagonal matrix containing drag and lift coefficients, respectively.

3.1.3 Buoyancy

The buoyant force is given as $b = \rho g \Delta$ where g is the acceleration of gravity, and Δ is the total volume of fluid displaced by the rigid body. The center of buoyancy depends on the geometry of the manipulator link. For a cylindrical link, it coincides with the center of mass and along the opposite direction of gravity.

3.1.4 Other Fluid Effects

Rotational damping. Not only the translational motion can cause damping force in the fluid. When the manipulator link is rotating, rotational damping force is exerted on the link. The damping is denoted as

$$\boldsymbol{\tau}_d = [\tau_{d_x}, \tau_{d_y}, \tau_{d_z}]^T, \tag{3.9}$$

where

$$\tau_{d_x} = C_{d_x} ||\boldsymbol{v}_r|| \omega_x \tag{3.10}$$

$$\tau_{d_y} = C_{d_y} ||\boldsymbol{v}_r|| \omega_y \tag{3.11}$$

$$\tau_{d_z} = C_{d_z} ||\boldsymbol{v}_r|| \omega_z \tag{3.12}$$

Current loads. The current loads τ_{cl} are torques that are only effective to rigid bodies with non-circular cross sections. The equation is shown as [8]

$$\boldsymbol{\tau}_{cl} = \boldsymbol{S}(\boldsymbol{v}_r)[diag(A_{11}, A_{22}, A_{33})\boldsymbol{v}_r] \tag{3.13}$$

where $\boldsymbol{S}$ is the skew-symmetric matrix operator. It is defined as $\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{S}(\boldsymbol{a})\boldsymbol{b}$.

The total friction torques exerted on a link is denoted as [13]

$$\boldsymbol{\tau}_D = \boldsymbol{\tau}_d + \boldsymbol{\tau}_{cl}. \tag{3.14}$$

3.2 Dynamic Equations for the Manipulator in Varying Environments

3.2.1 Modeling of the Manipulator

We use the Newton-Euler formulation to compute the force and moment of each link recursively and write them together to obtain the dynamic equations of the whole robot manipulator. The forward steps are introduced in Chapter 2. To recall, the

forward steps are [25]

$$^{k+1}\boldsymbol{\omega}_{k+1} = \boldsymbol{R}_k^{k+1}(^k\boldsymbol{\omega}_k + \boldsymbol{z}_k\dot{q}_{k+1}) \tag{3.15}$$

$$^{k+1}\boldsymbol{\alpha}_{k+1} = \boldsymbol{R}_k^{k+1}(^k\boldsymbol{\alpha}_k + ^k\boldsymbol{\omega}_k \times \boldsymbol{z}_k\dot{q}_{k+1} + \boldsymbol{z}_k\ddot{q}_{k+1}) \tag{3.16}$$

$$^{k+1}\boldsymbol{v}_{k+1} = \boldsymbol{R}_k^{k+1}{}^k\boldsymbol{v}_k + ^{k+1}\boldsymbol{\omega}_{k+1} \times ^{k+1}\boldsymbol{d}_{k/k+1} \tag{3.17}$$

$$^{k+1}\boldsymbol{v}_{k+1,c} = \boldsymbol{R}_k^{k+1}{}^k\boldsymbol{v}_k + ^{k+1}\boldsymbol{\omega}_{k+1} \times ^{k+1}\boldsymbol{d}_{k/k+1,c} \tag{3.18}$$

$$^{k+1}\boldsymbol{a}_{k+1} = \boldsymbol{R}_k^{k+1}{}^k\boldsymbol{a}_k + ^{k+1}\boldsymbol{\alpha}_{k+1} \times ^{k+1}\boldsymbol{d}_{k/k+1} + ^{k+1}\boldsymbol{\omega}_{k+1} \times (^{k+1}\boldsymbol{\omega}_{k+1} \times ^{k+1}\boldsymbol{d}_{k/k+1}) \tag{3.19}$$

$$^{k+1}\boldsymbol{a}_{k+1,c} = \boldsymbol{R}_k^{k+1}{}^k\boldsymbol{a}_k + ^{k+1}\boldsymbol{\alpha}_{k+1} \times ^{k+1}\boldsymbol{d}_{k/k+1,c} + ^{k+1}\boldsymbol{\omega}_{k+1} \times (^{k+1}\boldsymbol{\omega}_{k+1} \times ^{k+1}\boldsymbol{d}_{k/k+1,c}). \tag{3.20}$$

In order to get the explicit dynamic equations of the robot and the parameterized form, we apply the forward steps and obtain the equations for velocities and accelerations of robot links. By using (3.15) recursively, we can get

$$\boldsymbol{\omega}_k = \sum_{i=0}^{k-1} \boldsymbol{R}_i^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{i+1} \end{pmatrix}. \tag{3.21}$$

Using (3.16) recursively, and substitute (3.21) into it, we have

$$\boldsymbol{\alpha}_k = \sum_{i=0}^{k-1} \boldsymbol{R}_i^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{i+1} \end{pmatrix} + \sum_{i=0}^{k-2} \left(\boldsymbol{R}_i^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{i+1} \end{pmatrix} \times \left(\sum_{j=i+1}^{k-1} \boldsymbol{R}_j^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \right) \right). \tag{3.22}$$

By (3.17) and (3.18), substituting (3.21), we have

$$\boldsymbol{v}_k = \sum_{i=0}^{k-1} \left(\boldsymbol{R}_i^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{i+1} \end{pmatrix} \times \left(\sum_{j=i+1}^{k} \boldsymbol{R}_j^k \boldsymbol{d}_j \right) \right) \tag{3.23}$$

and

$$\boldsymbol{v}_{k,c} = \boldsymbol{R}_k^{k+1} \boldsymbol{v}_{k-1} + \left(\sum_{j=0}^{i-1} \boldsymbol{R}_j^i \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \right) \times \boldsymbol{d}_{i,c} \Bigg) . \tag{3.24}$$

For the linear acceleration, from (3.20) and using (3.21) and (3.22), we can get

$$
\begin{aligned}
\boldsymbol{a}_{k,c} = & \sum_{i=0}^{k-2} \left(\boldsymbol{R}_i^k \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_{i+1} \end{pmatrix} \times \sum_{j=i+1}^{k-1} \boldsymbol{R}_j^k \boldsymbol{d}_j \right) + \sum_{i=0}^{k-1} \boldsymbol{R}_i^k \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_{i+1} \end{pmatrix} \times \boldsymbol{d}_{k,c} \\
& + \sum_{i=0}^{k-3} \sum_{l=i+2}^{k-1} \left(\boldsymbol{R}_i^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{i+1} \end{pmatrix} \times \sum_{j=i+1}^{k-2} \boldsymbol{R}_j^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \times \boldsymbol{R}_l^k \boldsymbol{d}_l \right) \\
& + \sum_{i=0}^{k-2} \boldsymbol{R}_i^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{i+1} \end{pmatrix} \times \sum_{j=i+1}^{k-1} \boldsymbol{R}_j^k \left(\begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \right) \times \boldsymbol{d}_{k,c} \Bigg) \\
& + \sum_{i=0}^{k-2} \left(\sum_{j=0}^{i} \boldsymbol{R}_j^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \times \left(\sum_{l=0}^{i} \boldsymbol{R}_l^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{l+1} \end{pmatrix} \times \boldsymbol{R}_{i+1}^k \boldsymbol{d}_{i+1} \right) \right) \\
& + \sum_{j=0}^{k-1} \boldsymbol{R}_j^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \times \left(\sum_{l=0}^{k-1} \boldsymbol{R}_l^k \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{i+1} \end{pmatrix} \times \boldsymbol{d}_{k,c} \right).
\end{aligned}
\tag{3.25}
$$

For the backward steps to compute the force and moment considering the hydrodynamic forces, we use [25] [13]

$$
{}^k\boldsymbol{f}_k = \boldsymbol{R}_{k+1}^k \, {}^{k+1}\boldsymbol{f}_{k+1} + \boldsymbol{F}_k - m_k \boldsymbol{g}_k + \boldsymbol{b}_k + \boldsymbol{F}_{Dk} \tag{3.26}
$$

$$
{}^k\boldsymbol{\tau}_k = \boldsymbol{R}_{k+1}^k \, {}^{k+1}\boldsymbol{\tau}_{k+1} + \boldsymbol{d}_{k/k+1} \times \left(\boldsymbol{R}_{k+1}^k \, {}^{k+1}\boldsymbol{f}_{k+1} \right) + \boldsymbol{d}_{k/k+1,c} \times {}^k\boldsymbol{f}_k + \boldsymbol{\tau}_{Dk} + \boldsymbol{T}_k \tag{3.27}
$$

where $\boldsymbol{d}_{k/k,p}$ is the vector from frame k to the center of pressure of link k, $\boldsymbol{d}_{k/k,b}$ is the

vector from frame k to the center of buoyancy of link k, $\boldsymbol{F}_{Dk}$ is the total hydrodynamic drag force exerted on the link k and $\boldsymbol{\tau}_{Dk}$ is the total friction moment acting on the link k. Another difference from the normal case is that when we compute the force and moment acting on the center of mass, we need to use the total mass of the link, including the added mass and added moment of inertia. The vector $\boldsymbol{F}_k$ is written as

$$\boldsymbol{F}_k = \boldsymbol{M}_k({}^k\boldsymbol{a}_k +^k \boldsymbol{\alpha}_k \times^k \boldsymbol{d}_{k/k+1,c} +^k \boldsymbol{\omega}_k \times ({}^k\boldsymbol{\omega}_k \times^k \boldsymbol{d}_{k/k+1,c})) \tag{3.28}$$

and the vector $\boldsymbol{T}_k$ is written as

$$\boldsymbol{T}_k = \boldsymbol{I}_{mk}{}^k\boldsymbol{\alpha}_k +^k \boldsymbol{\omega}_k \times \boldsymbol{I}_{mk}{}^k\boldsymbol{\omega}_k \tag{3.29}$$

where $\boldsymbol{M}_k$ is the total mass of link k and $\boldsymbol{I}_{mk}$ is the total inertia of link k, and the added mass and added inertia are included, respectively.

Using (3.26) and substituting (3.28) into it, we can get

$$\boldsymbol{f}_k = \sum_{i=0}^{n-k} \boldsymbol{R}_n^{k+i} \left(\boldsymbol{M}_{n-i}\boldsymbol{a}_{n-i,c} + p_{n-i}\boldsymbol{g}_{n-i} - \boldsymbol{F}_{Dn-i}\right) \tag{3.30}$$

where n is the total number of links, $p_{n-i} = m_{n-i} - \rho_{fluid} \cdot V_{n-i}$ is the parameter for the gravity vector $\boldsymbol{g}_{n-i}$.

Finally, by (3.27), we can obtain the torque of each link as

$$\begin{aligned}\boldsymbol{\tau}_k = &\sum_{i=k}^{n} \boldsymbol{R}_i^k \left(\boldsymbol{I}_i\boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times \boldsymbol{I}_i\boldsymbol{\omega}_i\right) \\ &+ \sum_{i=k}^{n} \boldsymbol{R}_i^k \left(\boldsymbol{d}_{i,c} \times \left(\boldsymbol{M}_i\boldsymbol{a}_{i,c} + p_i\boldsymbol{g}_i - \boldsymbol{F}_{D_i}\right)\right) \\ &+ \sum_{i=k+1}^{n} \boldsymbol{R}_i^k \left(\left(\sum_{j=k}^{i-1} \boldsymbol{R}_j^i \boldsymbol{d}_j\right) \times \left(\boldsymbol{M}_i\boldsymbol{a}_{i,c} + p_i\boldsymbol{g}_i - \boldsymbol{F}_{D_i}\right)\right).\end{aligned} \tag{3.31}$$

Note that the angular velocity (3.21), angular acceleration (3.22) and the linear acceleration (3.25) of each link are sums of one term, two types of terms, and six types of terms, respectively. In addition, for the cross product, we have

$$\boldsymbol{a} \times (\boldsymbol{b} + \boldsymbol{c}) = \boldsymbol{a} \times \boldsymbol{b} + \boldsymbol{a} \times \boldsymbol{c}, \tag{3.32}$$

and for an orthogonal matrix $\boldsymbol{R}$, we have

$$\boldsymbol{R}(\boldsymbol{a} + \boldsymbol{b}) = \boldsymbol{R}\boldsymbol{a} \times \boldsymbol{R}\boldsymbol{b}, \tag{3.33}$$

where $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ are arbitrary vectors with the corresponding dimensions with the matrix $\boldsymbol{R}$.

Substituting the angular velocity (3.21), angular acceleration (3.22) and linear velocity (3.25) into (3.31), we can get the terms which contain mass terms and moment of inertia terms.

The terms coming from $\boldsymbol{R}_i^k \boldsymbol{I}_i \boldsymbol{\alpha}_i$ are

$$\boldsymbol{\tau}_{k_{z1}} = \sum_{i=k}^{n} \left(R_{i31}^k I_{i1} \sum_{i=0}^{i-1} R_{j13}^i \ddot{q}_{j+1} + R_{i32}^k I_{i2} \sum_{i=0}^{i-1} R_{j33}^i \ddot{q}_{j+1} + R_{i,3}^k I_{i3} \sum_{j=1}^{i=1} R_{j33}^i \ddot{q}_{j+1} \right) \tag{3.34}$$

$$\begin{aligned} \boldsymbol{\tau}_{k_{z2}} = & \sum_{i=k}^{n} \left(R_{i31}^k \sum_{j=0}^{i=2} \left(R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l33}^i \dot{q}_{i+1} - R_{j33}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l23}^i \dot{q}_{l+1} \right) I_{i1} \right. \\ & + R_{i32}^k \sum_{j=0}^{i-2} \left(R_{j33}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l13}^i \dot{q}_{i+1} - R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l33}^i \dot{q}_{l+1} \right) I_{i2} \\ & \left. + R_{i33}^k \sum_{j=0}^{i-2} \left(R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l23}^i \dot{q}_{i+1} - R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l13}^i \dot{q}_{l+1} \right) I_{i3} \right), \end{aligned} \tag{3.35}$$

the terms coming from $\boldsymbol{R}_i^k\left(\boldsymbol{\omega}_i \times \boldsymbol{I}_i\boldsymbol{\omega}_i\right)$ is

$$\begin{aligned}\boldsymbol{\tau}_{k_{z3}} = \sum_{i=k}^{n} \Bigg(& R_{i31}^k \sum_{j=0}^{i-1} R_{j23}^i \dot{q}_{j+1} \sum_{l=0}^{i=1} R_{l33}^i \dot{q}_{j+1} \left(I_{i3} - I_{i2}\right) \\ & + R_{i32}^k \sum_{j=0}^{i-1} R_{j33}^i \dot{q}_{j+1} \sum_{l=0}^{i=1} R_{l13}^i \dot{q}_{j+1} \left(I_{i1} - I_{i3}\right) \\ & + R_{i33}^k \sum_{j=0}^{i-1} R_{j13}^i \dot{q}_{j+1} \sum_{l=0}^{i=1} R_{l23}^i \dot{q}_{j+1} \left(I_{i2} - I_{i1}\right) \Bigg),\end{aligned} \tag{3.36}$$

the terms containing $\ddot{\boldsymbol{q}}$ and mass $\boldsymbol{M}$ coming from the linear accelerations are

$$\begin{aligned}\tau_{k_{z4}} = & \sum_{i=k}^{n} \boldsymbol{R}_i^k \Bigg(\boldsymbol{d}_{i,c} \times \boldsymbol{M}_i \Bigg(\sum_{j=0}^{i-2} \Bigg(\boldsymbol{R}_j^i \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_{j-1} \end{pmatrix} \times \sum_{l=j+1}^{i-1} \boldsymbol{R}_l^i \boldsymbol{d}_l \Bigg) \\ & + \sum_{j=0}^{i-1} \boldsymbol{R}_j^i \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_{j+1} \end{pmatrix} \times \boldsymbol{d}_{i,c} \Bigg) \Bigg)_z \\ = & \sum_{i=k}^{n} \Bigg(-R_{i32}^k \sum_{j=0}^{i-2} \sum_{l=j+1}^{i-1} \left(R_{j13}^i \ddot{q}_{j+1} R_{l21}^i - R_{j23}^i \ddot{q}_{j+1} R_l^i \right) a_{i,c} M_{i3} a_l \\ & + R_{i33}^k \sum_{j=0}^{i-2} \sum_{l=j+1}^{i-1} \left(R_{j33}^i \ddot{q}_{j+1} R_{l11}^i - R_{j13}^i \ddot{q}_{j+1} R_{l31}^i \right) a_{i,c} M_{i2} a_l \\ & - R_{i32}^k \sum_{j=0}^{i=1} R_{j23}^i \ddot{q}_{j+1} a_{i,c}^2 M_{i3} \\ & + R_{i33}^k \sum_{j=0}^{j-1} R_{j13}^i \ddot{q}_{j+1} a_{i,c}^2 M_{i2} \Bigg)\end{aligned} \tag{3.37}$$

$$\begin{aligned}
\tau_{k_{z5}} &= \sum_{i=k+1}^{n} \boldsymbol{R}_i^k \left(\sum_{j=k}^{i-1} \boldsymbol{R}_j^k \boldsymbol{d}_j \times \boldsymbol{M}_i \sum_{l=0}^{i-1} \boldsymbol{R}_l^i \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_{l+1} \end{pmatrix} \times \boldsymbol{d}_{i,c} \right)_z \\
&= \sum_{i=k+1}^{n} \left(R_{i31}^k \sum_{j=k}^{i-1} \left(\sum_{l=0}^{i-1} R_{j21}^k R_{l23}^i \ddot{q}_{l+1} - R_{i32}^k \sum_{j=k}^{i-1} \sum_{l=0}^{i-1} R_{j11}^k R_{l23}^i \ddot{q}_{l+1} \right) a_j M_{i3} a_{i,c} \right. \\
&\left. + R_{k33}^i \sum_{j=k}^{i-1} \left(\sum_{l=0}^{i-1} R_{j11}^k R_{l33}^i \ddot{q}_{l+1} - R_{i31}^k \sum_{j=k}^{i-1} \sum_{l=0}^{i-1} R_{j31}^k R_{l33}^i \ddot{q}_{l+1} \right) a_j M_{i2} a_{i,c} \right)
\end{aligned} \tag{3.38}$$

$$\begin{aligned}
\tau_{k_{z6}} &= \sum_{i=k+1}^{n} \boldsymbol{R}_i^k \left(\left(\sum_{j=k}^{i-1} \boldsymbol{R}_j^i \boldsymbol{d}_j \right) \times \boldsymbol{M}_i \sum_{l=0}^{i-2} \left(\boldsymbol{R}_l^i \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_{l+1} \end{pmatrix} \times \sum_{m=j+1}^{i-1} \boldsymbol{R}_m^i \boldsymbol{d}_m \right) \right)_z \\
&= \sum_{i=k+1}^{n} \left(R_{i31}^k \sum_{j=k}^{i-1} \sum_{l=0}^{i-2} \sum_{m=j+1}^{i-1} \left(R_{j21}^i \left(R_{l_{13}}^i \ddot{q}_{j+1} R_{m21}^i - R_{l23}^i \ddot{q}_{j-1} R_{m11}^i \right) a_j M_{i3} a_m \right. \right. \\
&\quad \left. - R_{j31}^i \left(R_{33}^i \ddot{q}_{j+1} R_{m11}^i - R_{l13}^i \ddot{q}_{j+1} R_{m31}^i \right) a_j M_{i2} a_m \right) \\
&\quad + R_{i32}^k \sum_{j=k}^{i-1} \sum_{l=0}^{i-2} \sum_{m=j+1}^{i-1} \left(R_{j31}^i \left(R_{l_{23}}^i \ddot{q}_{j+1} R_{m31}^i - R_{l33}^i \ddot{q}_{j-1} R_{m21}^i \right) a_j M_{i1} a_m \right. \\
&\quad \left. - R_{j11}^i \left(R_{13}^i \ddot{q}_{j+1} R_{m21}^i - R_{l23}^i \ddot{q}_{j+1} R_{m11}^i \right) a_j M_{i3} a_m \right) \\
&\quad + R_{i33}^k \sum_{j=k}^{i-1} \sum_{l=0}^{i-2} \sum_{m=j+1}^{i-1} \left(R_{j11}^i \left(R_{l_{33}}^i \ddot{q}_{j+1} R_{m11}^i - R_{l13}^i \ddot{q}_{j-1} R_{m31}^i \right) a_j M_{i2} a_m \right. \\
&\quad \left. \left. - R_{j21}^i \left(R_{23}^i \ddot{q}_{j+1} R_{m31}^i - R_{l33}^i \ddot{q}_{j+1} R_{m21}^i \right) a_j M_{i1} a_m \right) \right),
\end{aligned} \tag{3.39}$$

and terms containing $\dot{\boldsymbol{q}}$ coming from the linear accelerations are

$$\begin{aligned}
\tau_{k_{z7}} = &\sum_{i=k}^{n} \boldsymbol{R}_i^k \left(\boldsymbol{d}_{i,c} \times \boldsymbol{M}_i \sum_{j=0}^{i-3} \sum_{m=j+2}^{i-1} \left(\boldsymbol{R}_j^i \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \right. \right. \\
&\left. \left. \times \sum_{l=j+1}^{i-2} \boldsymbol{R}_l^i \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{l+1} \end{pmatrix} \times \boldsymbol{R}_m^i \boldsymbol{d}_m \right) \right) \\
= &\, R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l33}^i \dot{q}_{1+1} - R_{j33}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l13}^i \dot{q}_{l+1} \\
= &\sum_{i=k}^{n} R_{i32}^k \sum_{j=0}^{i-3} \sum_{m=j+2}^{i-1} \left(R_{m21}^i \left(R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l33}^i \dot{q}_{1+1} - R_{j33}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l23}^i \dot{q}_{l+1} \right) \right. \\
&\left. - R_{m11}^i \left(R_{j33}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l13}^i \dot{q}_{1+1} - R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l33}^i \dot{q}_{l+1} \right) \right) a_{i,c} M_{i3} a_m \\
+ &\, R_{i32}^k \sum_{j=0}^{i-3} \sum_{m=j+2}^{i-1} \left(R_{m11}^i \left(R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l23}^i \dot{q}_{1+1} - R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l13}^i \dot{q}_{l+1} \right) \right. \\
&\left. - R_{m31}^i \left(R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l33}^i \dot{q}_{1+1} - R_{j33}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R_{l23}^i \dot{q}_{l+1} \right) \right) a_{i,c} M_{i2} a_m
\end{aligned} \tag{3.40}$$

$$\begin{aligned}
\tau_{k_{z8}} = &\sum_{i=k}^{n} \boldsymbol{R}_i^k \left(\boldsymbol{d}_{i,c} \times \boldsymbol{M}_i \left(\sum_{j=0}^{i-2} \boldsymbol{R}_j^i \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \times \sum_{l=j+1}^{i-1} \boldsymbol{R}_l^i \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \times \boldsymbol{d}_{i,c} \right) \right)_z \\
= &\sum_{i=k}^{n} \left(R_{i32}^k \sum_{j=0}^{i-2} \left(R_{j33}^i \dot{q}_{j+1} \sum_{i=j+1}^{i-1} R_{l13}^i \dot{q}_{l+1} - R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l33}^i \dot{q}_{i+1} \right) M_{i3} a_{i,c}^2 \right. \\
&\left. + R_{i33}^k \sum_{j=0}^{i-2} \left(R_{j13}^i \dot{q}_{j+1} \sum_{i=j+1}^{i-1} R_{l23}^i \dot{q}_{l+1} - R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l13}^i \dot{q}_{i+1} \right) M_{i2} a_{i,c}^2 \right)
\end{aligned} \tag{3.41}$$

$$
\begin{aligned}
\tau_{k_{z9}} = \sum_{i=k+1}^{n} \Bigg(& R^k_{i31} \sum_{h=k}^{i-1} \sum_{j=0}^{i-3} \sum_{m=j+2}^{i-1} R^i_{h21} \left(t_1 R^i_{m21} - t_2 R^i_{m11} \right) a_h M_{i3} a_m \\
& - R^i_{h_{31}} \left(t_3 R^i_{m11} - t_1 R^i_{m31} \right) a_n M_{i2} a_m \\
& + R^k_{i32} \sum_{h=k}^{i-1} \sum_{j=0}^{i-3} \sum_{m=j+2}^{i-1} R^i_{h31} \left(t_2 R^i_{m31} - t_3 R^i_{m21} \right) a_h M_{i1} a_m \\
& - R^i_{h11} \left(t_1 R^i_{m21} - t_2 R^i_{m11} \right) a_n M_{i3} a_m \\
& + R^k_{i33} \sum_{h=k}^{i-1} \sum_{j=0}^{i-3} \sum_{m=i+2}^{i-1} R^i_{h11} \left(t_3 R^i_{m11} - t_1 R^i_{m31} \right) a_h M_{i2} a_m \\
& - R^i_{h_{21}} \left(t_2 R^i_{m31} - t_3 R^i_{m21} \right) a_n M_i, a_m \Bigg)
\end{aligned}
\tag{3.42}
$$

$$
\begin{aligned}
\tau_{k_{z10}} = & \sum_{i=k+1}^{n} \boldsymbol{R}_i^k \left(\sum_{h=k}^{i-1} \boldsymbol{R}_h^i \boldsymbol{d}_h \right. \\
& \left. \times \boldsymbol{M}_i \begin{pmatrix} 0 \\ \sum_{j=0}^{i-2} \left(R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l23}^i \dot{q}_{l+1} - R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l13}^i \dot{q}_{l+1} \right) a_{i,c} \\ -\sum_{j=0}^{i-2} \left(R_{j33}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l13}^i \dot{q}_{l+1} - R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l33}^i \dot{q}_{l+1} \right) a_{i,c} \end{pmatrix} \right) \\
= & \sum_{i=k+1}^{n} \left(R_{i31}^k \left(-\sum_{h=k}^{i-1} \sum_{j=0}^{i=2} R_{h21}^i \left(R_{j33}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l13}^i \dot{q}_{l+1} \right. \right. \right. \\
& \left. \left. \left. - R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l33}^i \dot{q}_{l+1} \right) a_h M_{i3} a_{i,c} \right. \right. \\
& - \sum_{h=k}^{i-1} \sum_{j=0}^{i-2} R_{h31}^i \left(R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{i23}^i \dot{q}_{l+1} \right. \\
& \left. \left. - R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l13}^i \dot{q}_{l+1} \right) a_h M_{i2} a_{i,c} \right) \\
& + R_{i32} \sum_{h=k}^{i-1} \sum_{j=0}^{i-2} R_{h11}^i \left(R_{j33}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{i13}^i \dot{q}_{l+1} \right. \\
& \left. - R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l33}^i \dot{q}_{l+1} \right) a_h M_{i3} a_{i,c} \\
& + R_{i33} \sum_{h=k}^{i-1} \sum_{j=0}^{i-2} R_{h11}^i \left(R_{j13}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{i23}^i \dot{q}_{l+1} \right. \\
& \left. \left. - R_{j23}^i \dot{q}_{j+1} \sum_{l=j+1}^{i-1} R_{l13}^i \dot{q}_{l+1} \right) a_h M_{i2} a_{i,c} \right)
\end{aligned}
\tag{3.43}
$$

$$\tau_{k_{z11}} = \sum_{i=k}^{n} \boldsymbol{R}_i^k \left(\boldsymbol{d}_{i,c} \times \boldsymbol{M}_i \left(\sum_{j=0}^{i=2} \left(\sum_{l=0}^{j} \boldsymbol{R}_l^i \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{l+1} \end{pmatrix} \right. \right. \right.$$
$$\left. \left. \left. \times \left(\sum_{m=0}^{j} \boldsymbol{R}_m^i \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{m+1} \end{pmatrix} \times \boldsymbol{R}_{j+1}^i \boldsymbol{d}_{i,c} \right) \right) \right) \right)$$
$$= \sum_{i=k}^{n} \left(-R_{i32}^k \sum_{j=0}^{i-2} \left(\sum_{l=0}^{j} R_{l13}^i \dot{q}_{l+1} \sum_{m=0}^{j} t_{5m} - \sum_{l=0}^{j} R_{l_{23}}^i \dot{q}_{l+1} \sum_{m=0}^{j} t_{4m} \right) a_{i,c} M_{i3} a_{j+1} \right.$$
$$\left. + R_{i33}^k \sum_{j=0}^{i-2} \left(\sum_{l=0}^{j} R_{i33}^i q_{l+1} \sum_{m=0}^{j} t_{4m} - \sum_{l=0}^{j} R_{l13}^i \dot{q}_{1+1} \sum_{m=0}^{j} t_{6m} \right) a_{i,c} M_{i2} a_{j+1} \right) \tag{3.44}$$

$$\tau_{k_{z12}} = \sum_{i=k}^{n} \boldsymbol{R}_i^k \left(\boldsymbol{d}_{i,c} \times \boldsymbol{M}_i \left(\sum_{j=0}^{i-1} \boldsymbol{R}_j^i \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{j+1} \end{pmatrix} \times \left(\sum_{l=0}^{i-1} \boldsymbol{R}_l^i \begin{pmatrix} 0 \\ 0 \\ \dot{q}_{l+1} \end{pmatrix} \times \boldsymbol{d}_{i,c} \right) \right) \right)$$
$$= -R_{i32}^k \left(\sum_{j=0}^{i-1} R_{j13}^i \dot{q}_{j+1} \right) \left(\sum_{l=0}^{i-1} R_{l33}^i \dot{q}_{l+1} \right) M_{i3} a_{i,c}^2$$
$$- R_{i33}^k \left(\sum_{j=0}^{i-1} R_{j13}^i \dot{q}_{j+1} \right) \left(\sum_{l=0}^{i-1} R_{l23}^i \dot{q}_{l+1} \right) M_{i2} a_{i,c}^2 \tag{3.45}$$

$$\tau_{k_{z13}} = \sum_{i=k+1}^{n} \left(R_{i31}^k \sum_{h=k}^{i-1} \sum_{j=0}^{i-2} \left(R_{h21}^i t_{9j} M_{i3} a_h a_{j+1} - R_{h31}^i t_{8j} M_{i2} a_h a_{j+1} \right) \right.$$
$$+ R_{i32}^k \sum_{h=k}^{i-1} \sum_{j=0}^{i-2} \left(R_{h31}^i t_{7j} M_{i1} a_h a_{j+1} - R_{h11}^i t_{9j} M_{i3} a_h a_{j+1} \right) \tag{3.46}$$
$$\left. + R_{i33}^k \sum_{h=k}^{i-1} \sum_{j=0}^{i-2} \left(R_{h11}^i t_{8j} M_{i2} a_h a_{j+1} - R_{h21}^i t_{7j} M_{i1} a_h a_{j+1} \right) \right)$$

$$\begin{aligned}\tau_{k_{z14}} = \sum_{i=k+1}^{n} \Bigg(& R^k_{i31} \sum_{h=k}^{i-1} \left(R^i_{h21} t_{12} M_{i3} a_{i,c} a_h - R^i_{h31} t_{11} M_{i2} a_{i,c} a_h \right) \\ & + R^k_{i32} \sum_{h=k}^{i=1} \left(R^i_{h31} t_{10} M_{i1} a_{i,c} a_h - R^i_{h11} t_{12} M_{i3} a_{i,c} a_h \right) \\ & + R^k_{i33} \sum_{h=k}^{i=1} \left(R^i_{h11} t_{11} M_{i2} a_{i,c} a_h - R^i_{h21} t_{10} M_{i1} a_{i,c} a_h \right) \Bigg)\end{aligned} \tag{3.47}$$

where in (3.42),

$$\begin{aligned} t_1 =& R^i_{j23} \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R^i_{l33} \dot{q}_{l+1} - R^i_{j33} \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R^i_{l23} \dot{q}_{l+1} \\ t_2 =& R^i_{j33} \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R^i_{l13} \dot{q}_{l+1} - R^i_{j13} \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R^i_{l33} \dot{q}_{l+1} \\ t_3 =& R^i_{j13} \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R^i_{l23} \dot{q}_{l+1} - R^i_{j23} \dot{q}_{j+1} \sum_{l=j+1}^{i-2} R^i_{l13} \dot{q}_{l+1}, \end{aligned} \tag{3.48}$$

in (3.44),

$$\begin{aligned} t_{4m} &= R^i_{m23} \dot{q}_{m+1} R^i_{(j+1)31} - R^i_{m33} \dot{q}_{m+1} R^i_{(j+1)21} \\ t_{5m} &= R^i_{m33} \dot{q}_{m+1} R^i_{(j+1)11} - R^i_{m13} \dot{q}_{m+1} R^i_{(j+1)31} \\ t_{6m} &= R^i_{m13} \dot{q}_{m+1} R^i_{(j+1)21} - R^i_{m23} \dot{q}_{m+1} R^i_{(j+1)11}, \end{aligned} \tag{3.49}$$

in (3.46),

$$\begin{aligned} t_{7j} &= \sum_{j=0}^{i-2} \left(\sum_{l=0}^{j} R^i_{l23} \dot{q}_{l+1} \sum_{m=0}^{j} t_{6m} - \sum_{l=0}^{j} R^i_{l33} \dot{q}_{l+1} \sum_{n=0}^{j} t_{5m} \right) \\ t_{8j} &= \sum_{j=0}^{i-2} \left(\sum_{l=0}^{j} R^i_{l33} \dot{q}_{l+1} \sum_{m=0}^{j} t_{4m} - \sum_{l=0}^{j} R^i_{l13} \dot{q}_{l+1} \sum_{n=0}^{j} t_{6m} \right) \\ t_{9j} &= \sum_{j=0}^{i-2} \left(\sum_{l=0}^{j} R^i_{l13} \dot{q}_{l+1} \sum_{m=0}^{j} t_{5m} - \sum_{l=0}^{j} R^i_{l23} \dot{q}_{l+1} \sum_{n=0}^{j} t_{4m} \right), \end{aligned} \tag{3.50}$$

and in (3.47),

$$
\begin{aligned}
t_{10} &= \left(\sum_{j=0}^{i-1} R^i_{j23}\dot{q}_{j+1}\right)\left(\sum_{l=0}^{i=1} R^i_{l23}\dot{q}_{l+1}\right) - \left(\sum_{j=0}^{i-1} R^i_{j33}\dot{q}_{j+1}\right)\left(\sum_{l=0}^{i=1} R^i_{l33}\dot{q}_{l+1}\right) \\
t_{11} &= \left(\sum_{j=0}^{i-1} R^i_{j13}\dot{q}_{j+1}\right)\left(\sum_{l=0}^{i=1} R^i_{l23}\dot{q}_{l+1}\right) \\
t_{12} &= \left(\sum_{j=0}^{i-1} R^i_{j13}\dot{q}_{j+1}\right)\left(\sum_{l=0}^{i=1} R^i_{l33}\dot{q}_{l+1}\right).
\end{aligned}
\tag{3.51}
$$

The gravity and the buoyancy terms are

$$
\begin{aligned}
\tau_{k_{z15}} &= \sum_{i=k}^{n} \boldsymbol{R}^k_i \left(\boldsymbol{d}_{i,c} \times p_i \boldsymbol{g}_i\right) \\
&= \sum_{i=k}^{n} \left(R^k_{i32} R^i_{w32} a_{i,c} p_i g - R^k_{i33} R^i_{w22} a_{i,c} p_i g\right)
\end{aligned}
\tag{3.52}
$$

$$
\begin{aligned}
\tau_{k_{z16}} &= \sum_{i=k}^{n} \boldsymbol{R}^k_i \left(\left(\sum_{j=k}^{i-1} \boldsymbol{R}^i_j \boldsymbol{d}_j\right) \times p_i \boldsymbol{g}_i\right) \\
&= \sum_{i=k+1}^{n} \Bigg(-R^k_{i31} \sum_{j=k}^{i-1} \left(R^i_{j21} R^i_{w32} + R^i_{j31} R^i_{w22}\right) p_i g \\
&\quad - R^k_{i32} \sum_{j=k}^{i-1} \left(R^i_{j31} R^i_{w12} + R^i_{j11} R^i_{w32}\right) p_i g \\
&\quad - R^k_{i33} \sum_{j=k}^{i-1} \left(R^i_{j11} R^i_{w22} + R^i_{j21} R^i_{w12}\right) p_i g \Bigg)
\end{aligned}
\tag{3.53}
$$

where $\boldsymbol{g}_i$ is the gravity vector described in the i-th frame, $\boldsymbol{R}^i_w$ denotes the rotation matrix transforming from the world attached frame to the i-th frame and $p_i = m_i - \rho_{\text{fluid}} V_i$ is the coefficient of the gravity acceleration vector which contains the gravity part and the buoyancy part.

Substituting the velocity (3.17) into the friction and drag forces, we can get terms

in the torques as

$$\tau_{k_{z17}} = \sum_{i=k}^{n} \left(-R_{i32}^{k} \sum_{j=0}^{i=2} \sum_{l=j+1}^{i=1} \left(R_{j13}^{i} \dot{q}_{j+1} R_{l21}^{i} - R_{j23}^{i} \dot{q}_{j+1} R_{l11}^{i-1} \right) D_s a_l a_{i,c} \right. \\ \left. + R_{i33}^{k} \sum_{j=0}^{i-2} \sum_{l=j+1}^{i-1} \left(R_{j33}^{i} \dot{q}_{j+1} R_{l11}^{i} - R_{j13}^{i} \dot{q}_{j+1} R_{l31}^{i-1} \right) D_s a_l a_{i,c} \right) \tag{3.54}$$

$$\tau_{k_{z18}} = \sum_{i=k}^{n} \left(R_{i32}^{k} \sum_{j=0}^{i-1} R_{j23}^{i} \dot{q}_{j+1} D_s a_{i,c}^2 + R_{i33}^{k} \sum_{j=0}^{i-1} R_{j33}^{i} \dot{q}_{j+1} D_s a_{i,c}^2 \right) \tag{3.55}$$

$$\tau_{k_{z19}} = \sum_{i=k}^{n} R_{i32}^{k} R_{f31}^{i} D_s v_f a_{i,c} - R_{i,33}^{k} R_{f21}^{i} D_s v_f a_{i,c} \tag{3.56}$$

$$\tau_{k_{z20}} = \sum_{i=k+1}^{n} \left(R_{i31}^{k} \sum_{h=k}^{i-1} \sum_{j=0}^{i=2} \sum_{l=j+1}^{i-1} \left(R_{h21}^{i} t_{15jl} - R_{h31}^{i} t_{14jl} \right) D_s a_h a_l \right. \\ + R_{i32}^{k} \sum_{h=k}^{i-1} \sum_{j=0}^{i=2} \sum_{l=j+1}^{i-1} \left(R_{h31}^{i} t_{13jl} - R_{h11}^{i} t_{15jl} \right) D_s a_h a_l \\ \left. + R_{i33}^{k} \sum_{h=k}^{i-1} \sum_{j=0}^{i=2} \sum_{l=j+1}^{i-1} \left(R_{h11}^{i} t_{14jl} - R_{h21}^{i} t_{13jl} \right) D_s a_h a_l \right) \tag{3.57}$$

$$\tau_{k_{z21}} = \sum_{i=k+1}^{n} \left(R_{i31}^{k} \sum_{h=k}^{i-1} \sum_{j=0}^{i-1} \left(R_{h21}^{i} R_{j23}^{i} \dot{q}_{j+1} - R_{h31}^{i} R_{j33}^{i} \dot{q}_{j+1} \right) D_s a_h a_{i,c} \right. \\ - R_{i32}^{k} \sum_{h=k}^{i-1} \sum_{j=0}^{i-1} R_{h11}^{i} R_{j23}^{i} \dot{q}_{j+1} D_s a_h a_{i,c} \\ \left. + R_{i33}^{k} \sum_{h=k}^{i-1} \sum_{j=0}^{i-1} R_{h11}^{i} R_{j33}^{i} \dot{q}_{j+1} D_s a_h a_{i,c} \right) \tag{3.58}$$

$$\tau_{k_{z22}} = \sum_{i=k+1}^{n} \left(R_{i31}^{k} \sum_{n=k}^{i-1} \left(R_{h21}^{i} R_{f31}^{i} - R_{h31}^{i} R_{f21}^{i} \right) D_s v_f a_h \right. \\ + R_{i32}^{k} \sum_{n=k}^{i-1} \left(R_{h31}^{i} R_{f11}^{i} - R_{h11}^{i} R_{f31}^{i} \right) D_s v_f a_h \\ \left. + R_{i33}^{k} \sum_{n=k}^{i-1} \left(R_{h11}^{i} R_{f21}^{i} - R_{h21}^{i} R_{f11}^{i} \right) D_s v_f a_h \right) \tag{3.59}$$

where in (3.57),

$$\begin{aligned} t_{13jl} &= R^i_{j23}\dot{q}_{j+1}R^i_{l31} - R^i_{l33}\dot{q}_{j+1}R^{i-1}_{l21} \\ t_{14jl} &= R^i_{j33}\dot{q}_{j+1}R^i_{l11} - R^i_{l13}\dot{q}_{j+1}R^{i-1}_{l31} \\ t_{15jl} &= R^i_{j13}\dot{q}_{j+1}R^i_{l21} - R^i_{l23}\dot{q}_{j+1}R^{i-1}_{l11}. \end{aligned} \tag{3.60}$$

Then the torque of the k-th joint is the sum of all τ_{k_z} terms shown in (3.34) to (3.59).

Next, we write the equations into the matrix form. First, we write the n joint torque values into a column vector. In this vector, each element equals the torque described in the above equations. Then we separate the equations into four parts.

The first part is the mass terms which contains the added mass and $\ddot{q}$ terms, as described in equations (3.34), (3.37), (3.38) and (3.39). Each of these terms can be regarded as a product of $\ddot{q}_i, i = 1, 2, ..., n$, system parameters added mass, the moment of inertia, and link length, and a function of $\boldsymbol{q}$. Thus we can write it in the form of a dot product of two vectors where the row vector contains the system parameters and the $\boldsymbol{q}$ elements, and the column vector is the $\ddot{\boldsymbol{q}}$ vector. In this way, we can write all the torque values in the same form with the same vector $\ddot{\boldsymbol{q}}$ and an exclusive row vector. Write these row vectors in the same order as the torque vector vertically. We can get the mass matrix $\boldsymbol{M}(\boldsymbol{q})$.

The second part is the Coriolis and centripetal terms which contains the added mass and $\dot{q}$ terms, as shown in equations (3.35), (3.36) and (3.40)-(3.47). From these equations, we can see that each term contains two $q_i, i = 1, 2, ..., n$, system parameters added mass, added moment of inertia and link length, and a function of $\boldsymbol{q}$. Using the same method above, we can write terms in each torque as an exclusive row vector and a $\dot{\boldsymbol{q}}$ vector, and write them in a matrix to get $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ matrix.

The third and fourth parts are the friction and drag part and the gravity and buoyancy part, described in equations (3.52)-(3.53) and (3.54)-(3.59), respectively. We simply write these terms in the same order as the torque values in column vectors

to get the drag vector $\boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and the gravity vector $\boldsymbol{g}(\boldsymbol{q})$.

Finally, we can get the dynamic equations of a robot manipulator moving in fluid as [13]

$$\boldsymbol{M}_m(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}_m(\boldsymbol{q}) = \boldsymbol{\tau}_m \tag{3.61}$$

which is linear in the parameters and linear in the generalized joint coordinates. $\boldsymbol{q} \in \mathbb{R}^n$ is the vector of generalized coordinates, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of control input of each joint, and n is equal to the number of joints of the manipulator. The matrix $\boldsymbol{M}_m(\boldsymbol{q})$ is the matrix of total mass and inertia, $\boldsymbol{C}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})$ contains the Coriolis and centripetal terms, $\boldsymbol{D}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the hydrodynamic damping terms and $\boldsymbol{g}_m(\boldsymbol{q})$ is the vector containing gravity and buoyancy terms. The difference between the matrices in (3.61) and those in (2.13) is the environmental factors. The matrices $\boldsymbol{M}_m(\boldsymbol{q})$ and $\boldsymbol{C}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})$ contain the added mass and added moment of inertia effect comparing to $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and $\boldsymbol{g}_m(\boldsymbol{q})$ contains the buoyancy effect comparing to $\boldsymbol{g}(\boldsymbol{q})$.

In addition, we can write the equations in a parameterized form. In the torque equations (3.34)-(3.59), we can see that each term is a product of one or more system parameters and a function of $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$. Thus each torque value can be written in a form of a dot product of two vectors: A row vector of joint variables and a column vector containing all combinations of system parameters appearing in equations (3.34)-(3.59). Writing the row vectors in the same order of the torque vector, we can get the matrix form of parameterizing the torque equations, namely the regressor $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ and the parameter vector $\boldsymbol{\theta}_m^*$. Thus, it can be denoted as

$$\boldsymbol{\tau} = \boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}_m^*. \tag{3.62}$$

The robot system has the following property:

Property 3.1. *The mass matrix $\boldsymbol{M}_m(\boldsymbol{q})$ is positive definite and symmetric, i.e.*

$\boldsymbol{M}_m(\boldsymbol{q}) = \boldsymbol{M}_m(\boldsymbol{q})^T > 0.$

The mass matrix is positive definite due to the positive kinetic energy, and Newton's third law indicates that the matrix is symmetric. The proof can be found in [13] and [18].

3.2.2 Effects of Varying Environments

When the environment of the robot manipulator changes, the difference reflects on the varying system parameters, which indicates that the parameter vector $\boldsymbol{\theta}_m^*$ is a function of time, namely $\boldsymbol{\theta}_m^*(t)$. In the same manner, the matrices containing the environment factors in the dynamic model are also functions of time. Thus, the dynamic model of the system will become

$$\boldsymbol{M}_m(\boldsymbol{q},t)\ddot{\boldsymbol{q}} + \boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}},t)\dot{\boldsymbol{q}} + \boldsymbol{D}_m(\boldsymbol{q},\dot{\boldsymbol{q}},t) + \boldsymbol{g}_m(\boldsymbol{q},t) = \boldsymbol{\tau}, \tag{3.63}$$

and the linearity in the parameters property becomes

$$\boldsymbol{\tau} = \boldsymbol{Y}_m(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\theta}_m^*(t). \tag{3.64}$$

3.3 The Two-link Manipulator Model

Recall that using the Newton-Euler formulation, we need to go through the forward recursion to obtain the kinematic variables and the backward recursion to compute the forces and torques. For the underwater case, since the geometric properties are identical to the normal case, the kinematic variables remain the same. Thus these variables are as shown from (2.16) to (2.22). Next, we consider the backward recursion. Compared to the normal case, we need to take the hydrostatic and hydro-

dynamic forces into account. Here we consider the added mass, drag, and buoyancy. The force and torque acting on the second link can then be computed as

$$\boldsymbol{f}_2 = M_2 a_{2,c} - m_2 g_2 - b_2 - \boldsymbol{F}_{D2} \tag{3.65}$$

$$\boldsymbol{\tau}_2 = \boldsymbol{T}_2 - d_{2,c/2} \times \boldsymbol{f}_2 \tag{3.66}$$

where

$$\boldsymbol{T}_2 = \boldsymbol{I}_{m2}\boldsymbol{\alpha}_2 + \boldsymbol{\omega}_2 \times (\boldsymbol{I}_{m2}\boldsymbol{\omega}_2) \tag{3.67}$$

$$\boldsymbol{b}_2 + m_2\boldsymbol{g}_2 = (m_2 g - \rho g \pi r_2^2 l_2) \begin{bmatrix} -\sin(q_1+q_2) \\ -\cos(q_1+q_2) \end{bmatrix} \tag{3.68}$$

$$\boldsymbol{F}_{D2} = \boldsymbol{D}_s \boldsymbol{v}_{2,c} + \boldsymbol{D}_d |\boldsymbol{v}_{2,c}|^T \boldsymbol{v}_{2,c} \tag{3.69}$$

and

$$\boldsymbol{M}_2 = diag(A_{11}, A_{22}) + m\boldsymbol{I} \tag{3.70}$$

where $\boldsymbol{I}$ is a 2×2 identity matrix. Finally we get

$$\begin{aligned} \tau_2 = {} & \ddot{q}_1(M_{22}l_{2c}^2 + M_{22}l_1 l_{2c}\cos q_2 + I_{m2}) \\ & + \ddot{q}_2(M_{22}l_{2c}^2 + I_{m2}) \\ & + \dot{q}_1(M_{22}l_1 l_{2c}\sin q_2 \dot{q}_1) \\ & + \dot{q}_1(D_{s_2}l_{2c}^2 + D_{s_2}l_1 l_{2c}\cos q_2 + D_{d_2}l_{2c}\,\mathrm{sgn}(v_{r_2 y})S_{21}) \\ & + \dot{q}_2(D_{s_2}l_{2c}^2 + D_{d_2}l_{2c}^3\,\mathrm{sgn}(v_{r_2 y})\dot{q}_2) \\ & + l_{2c}(m_2 - l_2\rho\pi r_2^2)g\cos(q_1+q_2) \end{aligned} \tag{3.71}$$

where

$$S_{21} = l_{2c}^2\dot{q}_1 + l_1^2\cos^2 q_2\dot{q}_1 + 2l_{2c}^2\dot{q}_2 + 2l_1 l_{2c}\cos q_2(\dot{q}_1 + \dot{q}_2) \tag{3.72}$$

and $M_{21} = m_2 + A_{11_2}$, $M_{22} = m_2 + A_{22_2}$ are the total mass of the link 2 on the x and y direction, respectively. $I_{m2} = I_2 + A_{66_2}$ is the total moment of inertia of link 2 and $\mathrm{sgn}(\cdot)$ denotes the sign function.

In the same manner, we can get

$$\begin{aligned}
\tau_1 = {} & \ddot{q}_1(M_{22}l_{2c}^2 + 2M_{22}l_1 l_{2c}\cos(q_2) + M_{22}l_1^2\cos^2(q_2) + M_{21}l_1^2\sin^2(q_2) \\
& \quad + M_{12}l_{1c}^2 + I_{m1} + I_{m2}) \\
& + \ddot{q}_2(M_{22}l_{2c}^2 + M_{22}l_1 l_{2c}\cos(q_2) + I_{m2}) \\
& + \dot{q}_1(M_{22}l_1 l_{2c}\sin(q_2)\dot{q}_1 + M_{22}l_1^2\sin(q_2)\cos(q_2)\dot{q}_1 \\
& \quad - M_{21}l_1 l_{2c}\sin(q_2)(\dot{q}_1 + \dot{q}_2) - M_{21}l_1^2\sin(q_2)\cos(q_2)\dot{q}_1) \\
& + \dot{q}_2(-M_{21}l_1 l_{2c}\sin(q_2)(\dot{q}_1 + \dot{q}_2)) \\
& + \dot{q}_1(D_{s_2}l_{2c}^2 + D_{s_2}l_1 l_{2c}\cos(q_2) + D_{d_2}l_{2c}\,\mathrm{sgn}(v_{r_2y})S_{21} + D_{s_2}l_1 l_{2c}\cos(q_2) \\
& \quad + D_{s_2}l_1^2\cos^2(q_2) + D_{d_2}l_1\cos(q_2)\,\mathrm{sgn}(v_{r_2y})S_{21} + D_{s_1}l_1^2\sin^2(q_2) \\
& \quad + D_{d_1}l_1^3\sin^3(q_2)\,\mathrm{sgn}(v_{r_2x})\dot{q}_1 + D_{s_2}l_{1c}^2 + D_{d_2}l_{1c}^3\,\mathrm{sgn}(v_{r_1y})\dot{q}_1) \\
& + \dot{q}_2(D_{s_2}l_{2c}^2 + D_{d_2}l_{2c}^3\,\mathrm{sgn}(v_{r_2y})\dot{q}_2 + D_{s_2}l_1 l_{2c}\cos(q_2) \\
& \quad + D_{d_2}l_1 l_{2c}^2\cos(q_2)\,\mathrm{sgn}(v_{r_2y})\dot{q}_2) \\
& + (m_2 l_1 + m_1 l_{1c} - l_1 l_2\rho\pi r_2^2 - l_1 l_{1c}\rho\pi r_1^2)g\cos(q_1) \\
& \quad + (m_2 l_{2c} - l_2 l_{2c}\rho\pi r_2^2)g\cos(q_1 + q_2)
\end{aligned} \tag{3.73}$$

where $M_{11} = m_1 + A_{22_1}$ is the total mass of the link 1, $I_{m2} = I_2 + A_{66_1}$ is the total moment of inertia of link 2. In these equations, the environment fact is modeled in the total mass and total moment of inertia which contain the added mass and the

added moment of inertia, the friction and drag coefficients, and the buoyancy terms.

The system can also be written in the form of

$$\boldsymbol{M}_m(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_m(\boldsymbol{q}) = \boldsymbol{\tau} \tag{3.74}$$

where

$$\begin{aligned} \boldsymbol{M}_m(\boldsymbol{q}) &= \begin{bmatrix} M_{m11} & M_{m12} \\ M_{m21} & M_{m22} \end{bmatrix}, \boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} C_{m11} & C_{m12} \\ C_{m21} & C_{m22} \end{bmatrix} \\ \boldsymbol{D}_m(\boldsymbol{q},\dot{\boldsymbol{q}}) &= \begin{bmatrix} D_{m11} & D_{m12} \\ D_{m21} & D_{m22} \end{bmatrix}, \boldsymbol{g}_m(\boldsymbol{q}) = \begin{bmatrix} g_{m1} \\ g_{m2} \end{bmatrix} \end{aligned} \tag{3.75}$$

and

$$\begin{aligned} M_{m11} &= M_{22}l_{2c}^2 + 2M_{22}l_1l_{2c}\cos(q_2) + M_{22}l_1^2\cos^2(q_2) + M_{21}l_1^2\sin^2(q_2) \\ &\quad + M_{12}l_{1c}^2 + I_{m1} + I_{m2} \\ M_{m12} &= M_{22}l_{2c}^2 + M_{22}l_1l_{2c}\cos(q_2) + I_{m2} \\ M_{m21} &= M_{22}l_{2c}^2 + M_{22}l_1l_{2c}\cos q_2 + I_{m2} \\ M_{m22} &= M_{22}l_{2c}^2 + I_{m2} \end{aligned} \tag{3.76}$$

$$\begin{aligned} C_{m11} &= M_{22}l_1l_{2c}\sin(q_2)\dot{q}_1 + M_{22}l_1^2\sin(q_2)\cos(q_2)\dot{q}_1 \\ &\quad - M_{21}l_1l_{2c}\sin(q_2)(\dot{q}_1+\dot{q}_2) - M_{21}l_1^2\sin(q_2)\cos(q_2)\dot{q}_1 \\ C_{m12} &= -M_{21}l_1l_{2c}\sin(q_2)(\dot{q}_1+\dot{q}_2) \\ C_{m21} &= M_{22}l_1l_{2c}\sin q_2\dot{q}_1 \\ C_{m22} &= 0 \end{aligned} \tag{3.77}$$

$$
\begin{aligned}
D_{m11} &= D_{s_2} l_{2c}^2 + D_{s_2} l_1 l_{2c} \cos(q_2) + D_{d_2} l_{2c} \operatorname{sgn}(v_{r_2 y}) S_{21} + D_{s_2} l_1 l_{2c} \cos(q_2) \\
&\quad + D_{s_2} l_1^2 \cos^2(q_2) + D_{d_2} l_1 \cos(q_2) \operatorname{sgn}(v_{r_2 y}) S_{21} + D_{s_1} l_1^2 \sin^2(q_2) \\
&\quad + D_{d_1} l_1^3 \sin^3(q_2) \operatorname{sgn}(v_{r_2 x}) \dot{q}_1 + D_{s_2} l_{1c}^2 + D_{d_2} l_{1c}^3 \operatorname{sgn}(v_{r_1 y}) \dot{q}_1 \\
D_{m12} &= D_{s_2} l_{2c}^2 + D_{d_2} l_{2c}^3 \operatorname{sgn}(v_{r_2 y}) \dot{q}_2 + D_{s_2} l_1 l_{2c} \cos(q_2) \\
&\quad + D_{d_2} l_1 l_{2c}^2 \cos(q_2) \operatorname{sgn}(v_{r_2 y}) \dot{q}_2 \\
D_{m21} &= D_{s_2} l_{2c}^2 + D_{s_2} l_1 l_{2c} \cos q_2 + D_{d_2} l_{2c} \operatorname{sgn}(v_{r_2 y}) S_{21} \\
D_{m22} &= D_{s_2} l_{2c}^2 + D_{d_2} l_{2c}^3 \operatorname{sgn}(v_{r_2 y}) \dot{q}_2
\end{aligned}
\tag{3.78}
$$

$$
\begin{aligned}
g_{m1} &= (m_2 l_1 + m_1 l_{1c} - l_1 l_2 \rho \pi r_2^2 - l_1 l_{1c} \rho \pi r_1^2) g \cos(q_1) \\
&\quad + (m_2 l_{2c} - l_2 l_{2c} \rho \pi r_2^2) g \cos(q_1 + q_2) \\
g_{m2} &= l_{2c} (m_2 - l_2 \rho \pi r_2^2) g \cos(q_1 + q_2).
\end{aligned}
\tag{3.79}
$$

From equations (3.76) to (3.79), we can see that the martix form terms are the corresponding terms in (3.73) and (3.71).

The dynamic equations can also be written as

$$
\boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \boldsymbol{\theta}_m^* = \boldsymbol{\tau} \tag{3.80}
$$

where $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \in \mathbb{R}^{24\times 2}$ is a matrix and $\boldsymbol{\theta}_m^* \in \mathbb{R}^{24}$ is a vector.

$$
\boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \begin{pmatrix}
\ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 & \ddot{q}_1 & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 & 2\cos(q_2)\ddot{q}_1 + \cos(q_2)\ddot{q}_2 + \sin(q_2)\dot{q}_1^2 \\
\ddot{q}_1 + \ddot{q}_2 & 0 & 0 & \ddot{q}_2 & 0 & \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 \\
\end{pmatrix}
$$

$$
\begin{pmatrix}
\cos^2(q_2)\ddot{q}_1 + \sin(q_2)\cos(q_2)\dot{q}_1^2 & \sin^2(q_2)\ddot{q}_1 - \sin(q_2)\cos(q_2)\dot{q}_1^2 \\
0 & 0 \\
-\sin(q_2)\dot{q}_1^2 - 2\sin(q_2)\dot{q}_1\dot{q}_2 - \sin(q_2)\dot{q}_2^2 & \cos(q_1) \quad \cos(q_1+q_2) \quad \dot{q}_1+\dot{q}_2 \quad \dot{q}_1 \\
0 & 0 \quad \cos(q_1+q_2) \quad \dot{q}_1+\dot{q}_2 \quad 0 \\
2\cos(q_2)\dot{q}_1 & \operatorname{sgn}(v_{r_2y})\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\dot{q}_1\dot{q}_2 + \operatorname{sgn}(v_{r_2y})\dot{q}_2^2 \quad 2\operatorname{sgn}(v_{r_2y})\cos^2 q_2\dot{q}_1^2 \\
\cos(q_2)\dot{q}_1 & \operatorname{sgn}(v_{r_2y})\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\dot{q}_1\dot{q}_2 + \operatorname{sgn}(v_{r_2y})\dot{q}_2^2 \quad \operatorname{sgn}(v_{r_2y})\cos^2 q_2\dot{q}_1^2 \\
\operatorname{sgn}(v_{r_2y})\cos^2 q_2\dot{q}_1^2 + 2\cos^2(q_2)\operatorname{sgn}(v_{r_2y})\dot{q}_1\dot{q}_2 + \cos(q_2)\dot{q}_2 & \cos^2(q_2)\dot{q}_1 \\
0 & 0 \\
2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1\dot{q}_2 & \cos^3 q_2\operatorname{sgn}(v_{r_2y})\dot{q}_1^2 \\
0 & 0 \\
\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1\dot{q}_2 + \cos(q_2)\operatorname{sgn}(v_{r_2y})\dot{q}_2^2 & \sin^2(q_2)\dot{q}_1 \\
2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1\dot{q}_2 & 0 \\
\sin^3(q_2)\operatorname{sgn}(v_{r_2x})\dot{q}_1^2 & \operatorname{sgn}(v_{r_1y})\dot{q}_1^2 \\
0 & 0
\end{pmatrix}
\tag{3.81}
$$

$$\boldsymbol{\theta}_m^* = \Big(\; M_{22}l_{2c}^2 \quad M_{12}l_{1c}^2 \quad I_{m1} \quad I_{m2} \quad 0 \quad M_{22}l_1l_{2c} \quad M_{22}l_1^2 \quad M_{21}l_1^2$$
$$M_{21}l_1l_{2c} \quad (m_2l_1 + m_1l_{1c} - l_1l_2\rho\pi r_2^2 - l_1l_{1c}\rho\pi r_1^2)g$$
$$l_{2c}(m_2 - l_2\rho\pi r_2^2)g \quad D_{s_2}l_{2c}^2 \quad D_{s_2}l_{1c}^2 \quad D_{s_2}l_1l_{2c} \quad D_{d_2}l_{2c}^3 \tag{3.82}$$
$$D_{d_2}l_1^2l_{2c} \quad D_{d_2}l_{2c}l_1^2 \quad D_{s_2}l_1^2 \quad D_{d_2}l_1l_{2c}^2 \quad D_{d_2}l_1^3 \quad D_{d_2}l_1l_{2c}^2$$
$$D_{s_1}l_1^2 \quad D_{d_1}l_1^3 \quad D_{d_2}l_{1c}^3 \; \Big)^T .$$

From (3.82), we can see that the added mass, the added moment of inertia, drag coefficients, and fluid density are all modeled in the system parameters.

Chapter 4

Adaptive Single-model Based Control Designs

In this chapter, we build the robot dynamic models for different environmental situations, present single-model adaptive controllers to satisfy asymptotic tracking of manipulator joints and stable estimation of unknown system parameters for each situation, then use a two-link planar elbow manipulator to illustrate the controller performance for each fixed situation. Then we use the obtained knowledge to develop the multiple-model based adaptive control design to deal with the situations when environments change in the next chapter.

4.1 Nominal Controller

A nominal controller is designed as if the estimated parameters were exactly known. An adaptive controller can be built based on the knowledge obtained from the nominal controller. Recall that the system model of a robot moving in a fluid

environment is

$$\boldsymbol{M}_m(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_m(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}_m(\boldsymbol{q}) = \boldsymbol{\tau} \tag{4.1}$$

where the system parameters contain the environmental effects. In this case, these system parameters are known, and joint variables can be obtained by sensors, which means that matrices $\boldsymbol{M}_m(\boldsymbol{q})$, $\boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}})$, $\boldsymbol{D}_m(\boldsymbol{q},\dot{\boldsymbol{q}})$, $\boldsymbol{g}_m(\boldsymbol{q})$ and vectors $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ can be obtained. We can adopt a controller structure [21]

$$\boldsymbol{u} = \boldsymbol{M}_m(\boldsymbol{q})\boldsymbol{a}_q + \boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_m(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}_m(\boldsymbol{q}) \tag{4.2}$$

where

$$\boldsymbol{a}_q = \ddot{\boldsymbol{q}}_d - \boldsymbol{K}_1\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{K}_0\tilde{\boldsymbol{q}}. \tag{4.3}$$

Theorem 4.1. *For the robot moving in a specific environment described as (4.1), using the controller (4.2), the robot can track the desired trajectory asymptotically.*

Proof. Since the inertia matrix $\boldsymbol{M}_m(\boldsymbol{q})$ is invertible, the system reduces to

$$\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_1\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_0\tilde{\boldsymbol{q}} = 0. \tag{4.4}$$

Let $\boldsymbol{e}$ be the system state error vector

$$\boldsymbol{e} = \begin{bmatrix} \tilde{\boldsymbol{q}} \\ \dot{\tilde{\boldsymbol{q}}} \end{bmatrix}. \tag{4.5}$$

Then we have

$$\begin{aligned}\dot{\boldsymbol{e}} &= \begin{bmatrix}\dot{\tilde{\boldsymbol{q}}}\\ \ddot{\tilde{\boldsymbol{q}}}\end{bmatrix}\\ &= \begin{bmatrix}\boldsymbol{0}_{n\times n} & \boldsymbol{I}_{n\times n}\\ -\boldsymbol{K}_0 & -\boldsymbol{K}_1\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{q}}\\ \dot{\tilde{\boldsymbol{q}}}\end{bmatrix}\\ &= \boldsymbol{A}\boldsymbol{e}. \end{aligned} \tag{4.6}$$

Since matrix $\boldsymbol{A}$ is stable, we have the whole system stable, which also indicates that the system tracking error $\boldsymbol{e}$ will converge to 0, i.e., $\lim_{t\to\infty}\boldsymbol{e} = 0$. Thus the system output can track the desired trajectory asymptotically. □

4.2 Direct Adaptive Control

In an application, the system parameters may not be exactly known, and the environmental factors will also have a huge influence on the dynamic equations. We need to use an adaptive controller and parameter estimation to achieve asymptotic tracking. First, from (3.62), we parameterize the dynamic equations of the robot as

$$\boldsymbol{M}_m(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_m(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}_m(\boldsymbol{q}) = \boldsymbol{Y}_m(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\theta}_m, \tag{4.7}$$

where $\boldsymbol{Y}_m(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ is the regressor matrix and $\boldsymbol{\theta}_m$ contains the parameters of the environment and robot.

In the unknown parameters case, we adopt the controller structure

$$\boldsymbol{u} = \hat{\boldsymbol{M}}_m(\boldsymbol{q})\boldsymbol{a}_q + \hat{\boldsymbol{C}}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}_m(\boldsymbol{q},\dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}_m(\boldsymbol{q}) \tag{4.8}$$

where $\boldsymbol{a}_q$ has the same structure with (4.3), the matrices $\hat{\boldsymbol{M}}_m$, $\hat{\boldsymbol{C}}_m$, $\hat{\boldsymbol{D}}_m$ and $\hat{\boldsymbol{g}}_m$ represents the estimated value of $\boldsymbol{M}_m$, $\boldsymbol{C}_m$, $\boldsymbol{D}_m$ and $\boldsymbol{g}_m$, respectively.

The closed-loop system becomes

$$\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_1\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_0\tilde{\boldsymbol{q}} = \hat{\boldsymbol{M}}^{-1}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}. \tag{4.9}$$

We then choose the adaptive update law

$$\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}^{-1}(\hat{\boldsymbol{M}}^{-1}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}))^T\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{e} \tag{4.10}$$

where $\boldsymbol{\Gamma}$ is a positive definite coefficient matrix, and $\boldsymbol{P}$ is the solution to the Lyapunov equation

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q}. \tag{4.11}$$

Theorem 4.2. *For the robot moving in a specific fluid environment (4.1) where the environment system parameters are unknown, using the controller structure shown in (4.8) and the adaptive update law (4.10), the global asymptotic tracking of the joint angles can be achieved, and the parameter estimates are bounded.*

Proof. Let the Lyapnov function V be

$$V = \boldsymbol{e}^T\boldsymbol{P}\boldsymbol{e} + \tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}\tilde{\boldsymbol{\theta}}. \tag{4.12}$$

Then the derivative of V is

$$\dot{V} = -\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} < 0 \tag{4.13}$$

where $\boldsymbol{Q}$ is a random positive matrix satisfying $\boldsymbol{Q} = \boldsymbol{Q}^T > 0$. By Lyapunov direct method [7], we can see that the the position tracking errors $\boldsymbol{e}$ converge to zero asymp-

totically, and the parameter estimation errors $\tilde{\boldsymbol{\theta}}$ remain bounded, which indicates that the system parameter estimates $\hat{\boldsymbol{\theta}}$ are bounded. [25] □

4.3 Indirect Adaptive control

With the parameterize method shown in (4.7) and the controller structure shown in (4.8), the indirect adaptive control scheme uses the error between the actual torque and the estimated torque to generate the parameter update law. Let the torque prediction and its parameterized form be

$$\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{M}}_m(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}_m(\boldsymbol{q},\dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}_m(\boldsymbol{q}) = \boldsymbol{Y}_m(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\hat{\boldsymbol{\theta}}_m. \tag{4.14}$$

Then the prediction error can be written as

$$\tilde{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}} - \boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\tilde{\boldsymbol{\theta}} \tag{4.15}$$

where $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ is the parameter estimation error. The adaptive update law can be chosen as

$$\dot{\tilde{\boldsymbol{\theta}}} = \dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})^T\tilde{\boldsymbol{\tau}}. \tag{4.16}$$

Theorem 4.3. *For the robot moving in a specific environment described by (4.1), applying the controller (4.8) and the adaptive update law (4.16) guarantees boundedness for parameter estimates $\hat{\boldsymbol{\theta}}$ and globally asymptotic tracking of the desired trajectory.*

Proof. To analyse the stability and parameter estimate convergence of the closed-loop system, let us consider the Lyapunov function

$$V = \frac{1}{2}\tilde{\boldsymbol{\theta}}\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}} \tag{4.17}$$

and substitute (4.16) into its derivative to get

$$\begin{aligned} \dot{V} &= \tilde{\boldsymbol{\theta}}^T \boldsymbol{\Gamma}^{-1} \dot{\tilde{\boldsymbol{\theta}}} \\ &= -\tilde{\boldsymbol{\theta}}^T \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})^T \tilde{\boldsymbol{\tau}} \\ &= -\tilde{\boldsymbol{\tau}}^T \tilde{\boldsymbol{\tau}} \\ &\leq 0 \end{aligned} \tag{4.18}$$

thus

$$\tilde{\boldsymbol{\theta}} \in L^\infty, \tilde{\boldsymbol{\tau}} \in L^2 \tag{4.19}$$

which follows that parameter estimates $\hat{\boldsymbol{\theta}}$ are bounded.

Substituting the torque prediction (4.14) and the actual torque (4.8) into the prediction error (4.15), we can get

$$\begin{aligned} \tilde{\boldsymbol{\tau}} &= \hat{\boldsymbol{\tau}} - \boldsymbol{\tau} \\ &= (\hat{\boldsymbol{M}}\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q})) - (\hat{\boldsymbol{M}}\boldsymbol{a}_q + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q})) \\ &= \hat{\boldsymbol{M}}(\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_1 \dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_0 \tilde{\boldsymbol{q}}) \end{aligned} \tag{4.20}$$

which leads to

$$\tilde{\boldsymbol{q}} = \left(s^2 \boldsymbol{I} + s\boldsymbol{K}_D + \boldsymbol{K}_P\right) \hat{\boldsymbol{M}}^{-1}(\boldsymbol{q}) \tilde{\tau}. \tag{4.21}$$

From (4.19), we know that $\hat{\boldsymbol{M}}^{-1}\tilde{\boldsymbol{\tau}}$ is bounded. In addition, $\tilde{\boldsymbol{q}} = \boldsymbol{q} - \boldsymbol{q}_d$ and $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d$ are bounded. Thus, we can conclude that

$$\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}} \in L^2, \tilde{\boldsymbol{q}} \in L^\infty \tag{4.22}$$

and according to Lemma 2.1, the theorem holds. □

4.4 Simulation Study

Consider the planar elbow robot manipulator as shown in Figure 2.2, which is moving in the water. We have already analyzed the dynamics of this manipulator in Chapter 2. Now we consider the forces that are exerted on the manipulator to get the dynamic equations. Then we carry out adaptive controllers and parameter update laws for both the normal case and the underwater case.

Dynamic model. The dynamic equations for the underwater manipulator can be found in Chapter 3.3, as shown in (3.74)

$$\boldsymbol{M}_m(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_m(\boldsymbol{q}) = \boldsymbol{\tau}, \tag{4.23}$$

which can also be written as

$$\boldsymbol{\tau} = \boldsymbol{Y}_m(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\theta}_m^*, \tag{4.24}$$

where

$$\boldsymbol{Y}_m(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}) = \left(\begin{array}{cccccc}
\ddot{q}_1+\ddot{q}_2 & \ddot{q}_1 & \ddot{q}_1 & \ddot{q}_1+\ddot{q}_2 & \ddot{q}_1 & 2\cos(q_2)\ddot{q}_1+\cos(q_2)\ddot{q}_2+\sin(q_2)\dot{q}_1^2 \\
\ddot{q}_1+\ddot{q}_2 & 0 & 0 & \ddot{q}_2 & 0 & \cos(q_2)\ddot{q}_1+\sin(q_2)\dot{q}_1^2 \\
\end{array}\right.$$

$$\begin{array}{cc}
\cos^2(q_2)\ddot{q}_1+\sin(q_2)\cos(q_2)\dot{q}_1^2 & \sin^2(q_2)\ddot{q}_1-\sin(q_2)\cos(q_2)\dot{q}_1^2 \\
0 & 0 \\
-\sin(q_2)\dot{q}_1^2-2\sin(q_2)\dot{q}_1\dot{q}_2-\sin(q_2)\dot{q}_2^2 & \cos(q_1)\quad \cos(q_1+q_2)\quad \dot{q}_1+\dot{q}_2\quad \dot{q}_1 \\
0 & 0\quad \cos(q_1+q_2)\quad \dot{q}_1+\dot{q}_2\quad 0 \\
2\cos(q_2)\dot{q}_1 & \operatorname{sgn}(v_{r_2y})\dot{q}_1^2+2\operatorname{sgn}(v_{r_2y})\dot{q}_1\dot{q}_2+\operatorname{sgn}(v_{r_2y})\dot{q}_2^2\quad 2\operatorname{sgn}(v_{r_2y})\cos^2 q_2\dot{q}_1^2 \\
\cos(q_2)\dot{q}_1 & \operatorname{sgn}(v_{r_2y})\dot{q}_1^2+2\operatorname{sgn}(v_{r_2y})\dot{q}_1\dot{q}_2+\operatorname{sgn}(v_{r_2y})\dot{q}_2^2\quad \operatorname{sgn}(v_{r_2y})\cos^2 q_2\dot{q}_1^2 \\
\operatorname{sgn}(v_{r_2y})\cos^2 q_2\dot{q}_1^2+2\cos^2(q_2)\operatorname{sgn}(v_{r_2y})\dot{q}_1\dot{q}_2+\cos(q_2)\dot{q}_2 & \cos^2(q_2)\dot{q}_1 \\
0 & 0 \\
2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1^2+2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1\dot{q}_2 & \cos^3 q_2\operatorname{sgn}(v_{r_2y})\dot{q}_1^2 \\
0 & 0 \\
\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1^2+2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1\dot{q}_2+\cos(q_2)\operatorname{sgn}(v_{r_2y})\dot{q}_2^2 & \sin^2(q_2)\dot{q}_1 \\
2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1^2+2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1\dot{q}_2 & 0 \\
\sin^3(q_2)\operatorname{sgn}(v_{r_2x})\dot{q}_1^2 & \operatorname{sgn}(v_{r_1y})\dot{q}_1^2 \\
0 & 0
\end{array}\right) \tag{4.25}$$

$$\begin{aligned}\boldsymbol{\theta}_m^* = \Big(& M_{22}l_{2c}^2 \quad M_{12}l_{1c}^2 \quad I_{m1} \quad I_{m2} \quad 0 \quad M_{22}l_1l_{2c} \quad M_{22}l_1^2 \quad M_{21}l_1^2 \\ & M_{21}l_1l_{2c} \quad (m_2l_1 + m_1l_{1c} - l_1l_2\rho\pi r_2^2 - l_1l_{1c}\rho\pi r_1^2)g \\ & l_{2c}(m_2 - l_2\rho\pi r_2^2)g \quad D_{s_2}l_{2c}^2 \quad D_{s_2}l_{1c}^2 \quad D_{s_2}l_1l_{2c} \quad D_{d_2}l_{2c}^3 \\ & D_{d_2}l_1^2l_{2c} \quad D_{d_2}l_{2c}l_1^2 \quad D_{s_2}l_1^2 \quad D_{d_2}l_1l_{2c}^2 \quad D_{d_2}l_1^3 \quad D_{d_2}l_1l_{2c}^2 \\ & D_{s_1}l_1^2 \quad D_{d_1}l_1^3 \quad D_{d_2}l_{1c}^3 \Big),\end{aligned} \tag{4.26}$$

as shown in (3.81) and (3.82).

The dynamic equations of the robot moving without the effects of the fluid can also be written in the form of

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}^* \tag{4.27}$$

where the matrix $\boldsymbol{Y}$ and the parameter $\boldsymbol{\theta}^*$ are

$$\begin{aligned}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \Bigg(& \ddot{q}_1 + \ddot{q}_2 \quad \ddot{q}_1 \quad \ddot{q}_1 \quad \ddot{q}_1 + \ddot{q}_2 \quad \ddot{q}_1 \quad 2\cos(q_2)\ddot{q}_1 + \cos(q_2)\ddot{q}_2 + \sin(q_2)\dot{q}_1^2 \\ & \ddot{q}_1 + \ddot{q}_2 \quad 0 \quad 0 \quad \ddot{q}_2 \quad 0 \quad \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 \\ & \cos^2(q_2)\ddot{q}_1 + \sin(q_2)\cos(q_2)\dot{q}_1^2 \quad \sin^2(q_2)\ddot{q}_1 - \sin(q_2)\cos(q_2)\dot{q}_1^2 \\ & 0 \quad 0 \\ & -\sin(q_2)\dot{q}_1^2 - 2\sin(q_2)\dot{q}_1\dot{q}_2 - \sin(q_2)\dot{q}_2^2 \quad \cos(q_1) \quad \cos(q_1 + q_2) \\ & 0 \quad 0 \quad \cos(q_1 + q_2) \Bigg)\end{aligned} \tag{4.28}$$

$$\begin{aligned}\boldsymbol{\theta}^* = \Big(& m_2l_{2c}^2 \quad m_1l_{1c}^2 \quad I_{m1} \quad I_{m2} \quad 0 \quad m_2l_1l_{2c} \quad m_2l_1^2 \quad m_2l_1^2 \\ & m_2l_1l_{2c} \quad (m_2l_1 + m_1l_{1c} - l_1l_2\rho\pi r_2^2 - l_1l_{1c}\rho\pi r_1^2)g \\ & l_{2c}(m_2 - l_2\rho\pi r_2^2)g \Big)^T.\end{aligned} \tag{4.29}$$

.

Here we suppose Let the true values of the system parameters be

$$\begin{aligned} m_1 &= 25.494, \quad m_2 = 21.245, \quad l_1 = 1.2, \quad l_{1,c} = 0.6, \\ l_2 &= 1, \quad l_{2,c} = 0.5, \quad r_1 = 0.05, \quad r_2 = 0.05, \\ g &= 9.8, \quad \rho_{fluid} = 997, \\ D_s &= diag\{0.04, 0.04\} \\ D_d &= diag\{0.8, 1.2\}. \end{aligned} \tag{4.30}$$

The parameters simulate a real robot manipulator, and the mass of the manipulator is computed from the geometry with the density of iron. In addition, the fluid density and friction/drag coefficients simulate the water.

The initial conditions of joint variables be $(q_1, q_2, \dot{q}_1, \dot{q}_2)|_{t=0} = (1, 1, -0.5, -0.2)$ and the desired joint variables be $q_{d1}(t) = q_{d2}(t) = \sin t$. Then the initial tracking errors are $(\tilde{q}_1, \tilde{q}_2, \dot{\tilde{q}}_1, \dot{\tilde{q}}_2)|_{t=0} = (1, 1, -1.5, -1.2)$

Nominal controller. First we show the control result of the nominal controller. Take (4.2) as the controller structure, and set the controller parameters as

$$\boldsymbol{K}_1 = diag\{50, 50\}, \boldsymbol{K}_0 = diag\{75, 75\}. \tag{4.31}$$

The system tracking errors are shown in Figure 4.1. From the figure, we can see that the joint positions and velocities track the desired trajectory asymptotically.

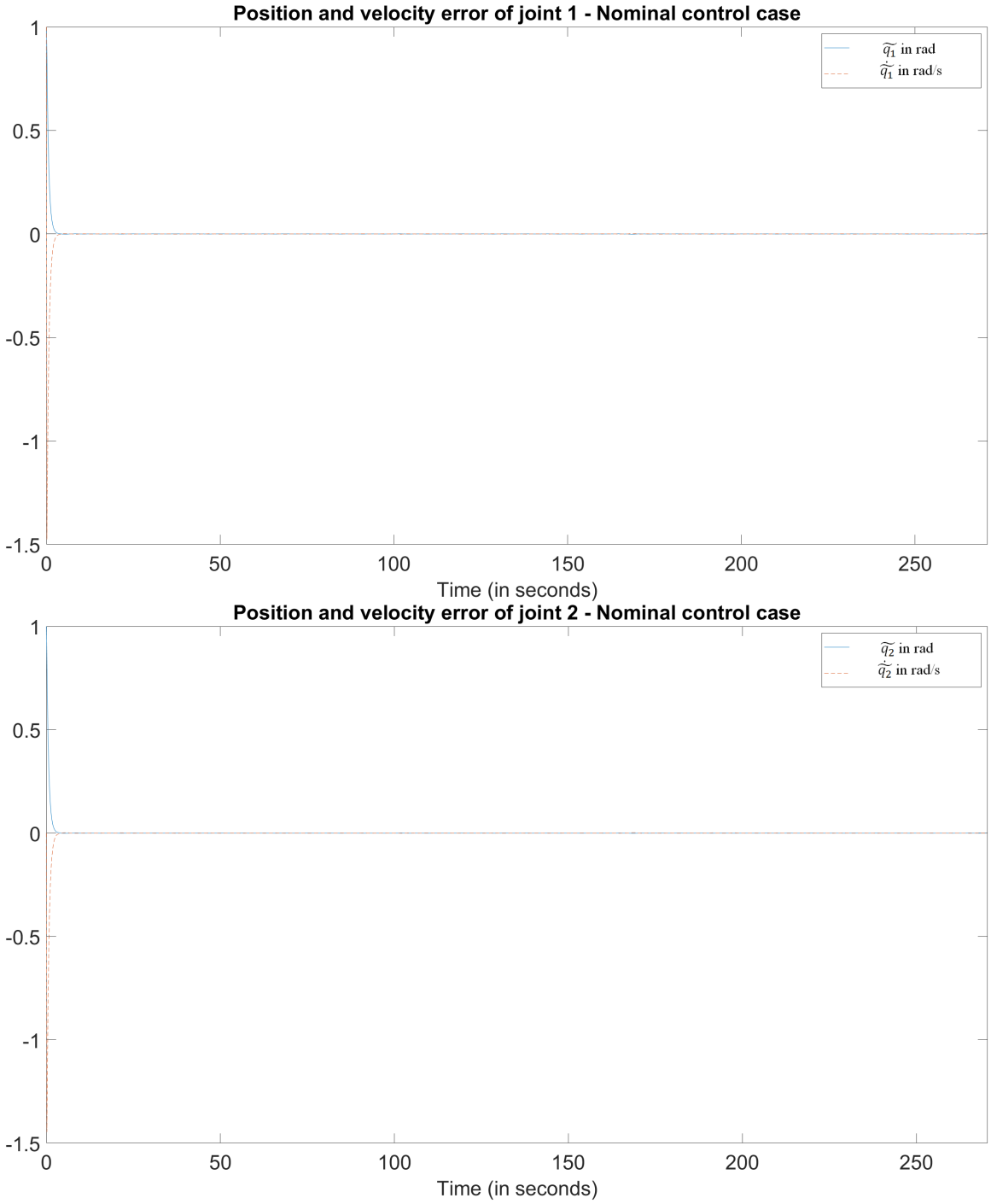

Figure 4.1: System tracking error for nominal control.

Direct adaptive controller. Use the adaptive controller as shown in (4.8) and (4.3). Set the initial conditions of joint variables as $(q_1, q_2, \dot{q}_1, \dot{q}_2)|_{t=0} = (1, 1, -0.5, -0.2)$

and the desired joint variables as $q_{d1}(t) = q_{d2}(t) = \sin t$. Choose the controller parameters as

$$\boldsymbol{K}_1 = diag\{50, 50\}, \boldsymbol{K}_0 = diag\{75, 75\}. \tag{4.32}$$

Let the estimate of the system parameters be

$$\hat{\boldsymbol{\theta}}(t) = \begin{bmatrix} \hat{\theta}_1(t) & \hat{\theta}_2(t) & \hat{\theta}_3(t) & \cdots & \hat{\theta}_{24}(t) \end{bmatrix} \tag{4.33}$$

then the parameter estimation error is denoted by $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_m^*$. Set the initial values of the parameter estimates $\hat{\boldsymbol{\theta}}_0$ as 90% of the true values. The adaptive update law is given as (4.10) with the parameters as

$$\boldsymbol{\Gamma} = diag\{0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.45, 0.9, 0.03, 0.03, \cdots, 0.03\}. \tag{4.34}$$

Figure 4.2 shows that the close-loop tracking errors converge to zero asymptotically. The parameter estimation errors are shown in Figures 4.3 to 4.9. From the figures, we can see that the parameter estimates are bounded.

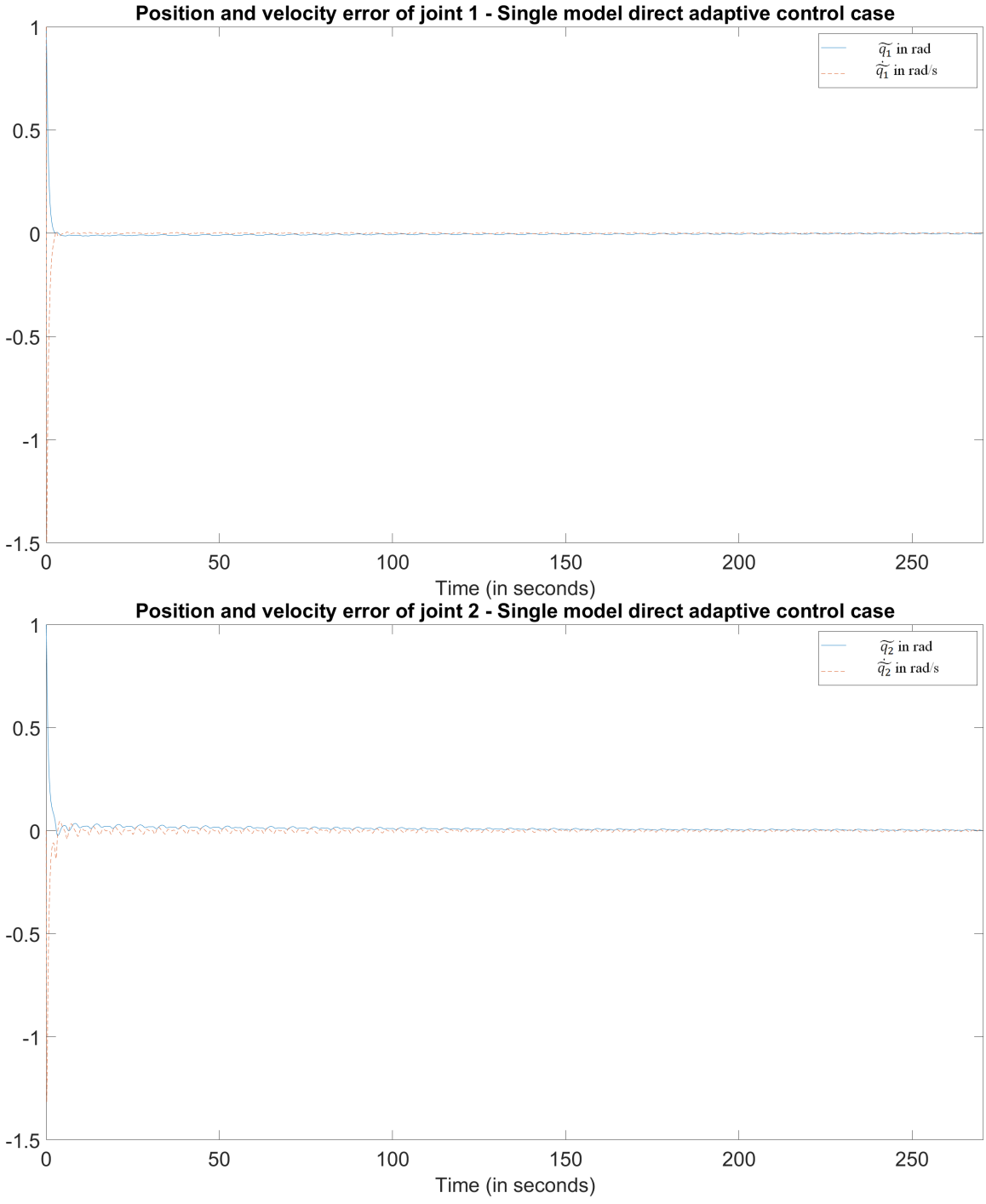

Figure 4.2: System tracking error for single model direct adaptive control.

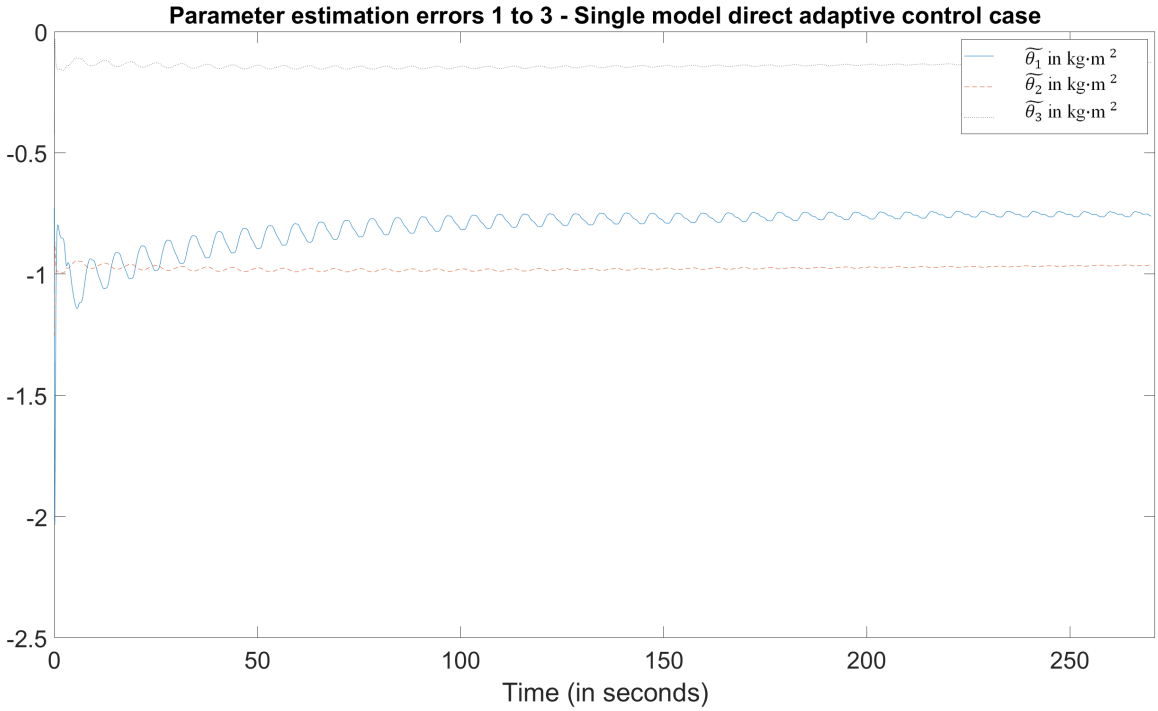

Figure 4.3: Parameter estimation errors 1 to 3 for single model direct adaptive control.

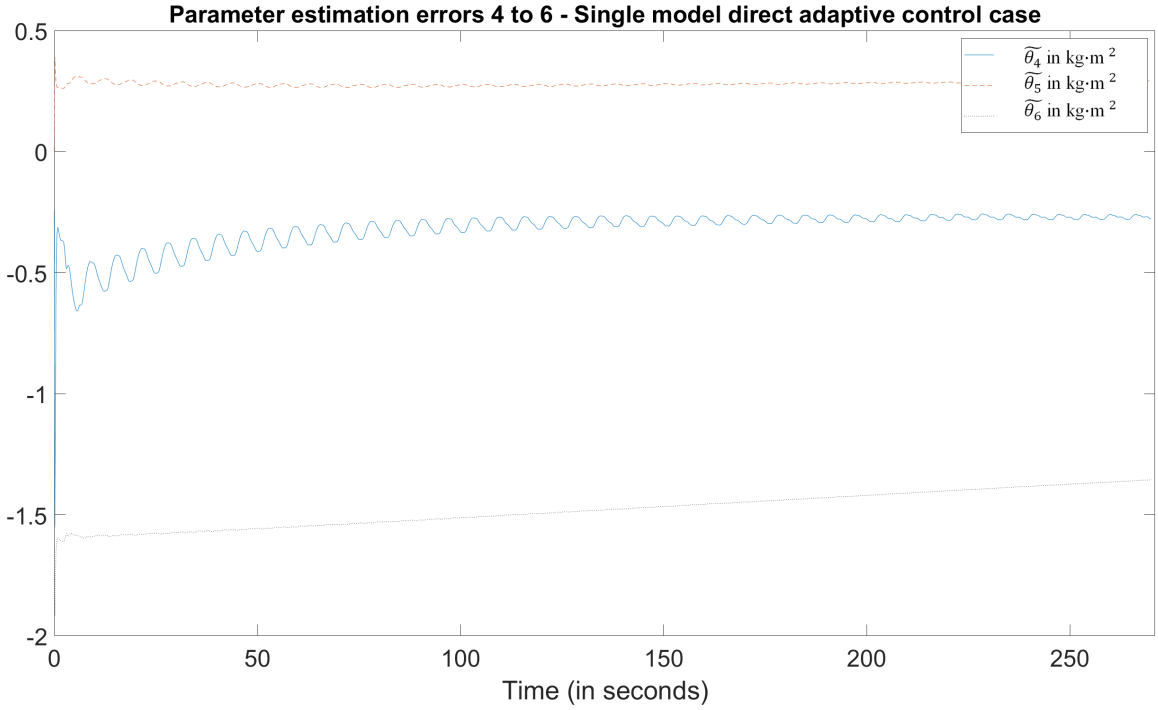

Figure 4.4: Parameter estimation errors 4 to 6 for single model direct adaptive control.

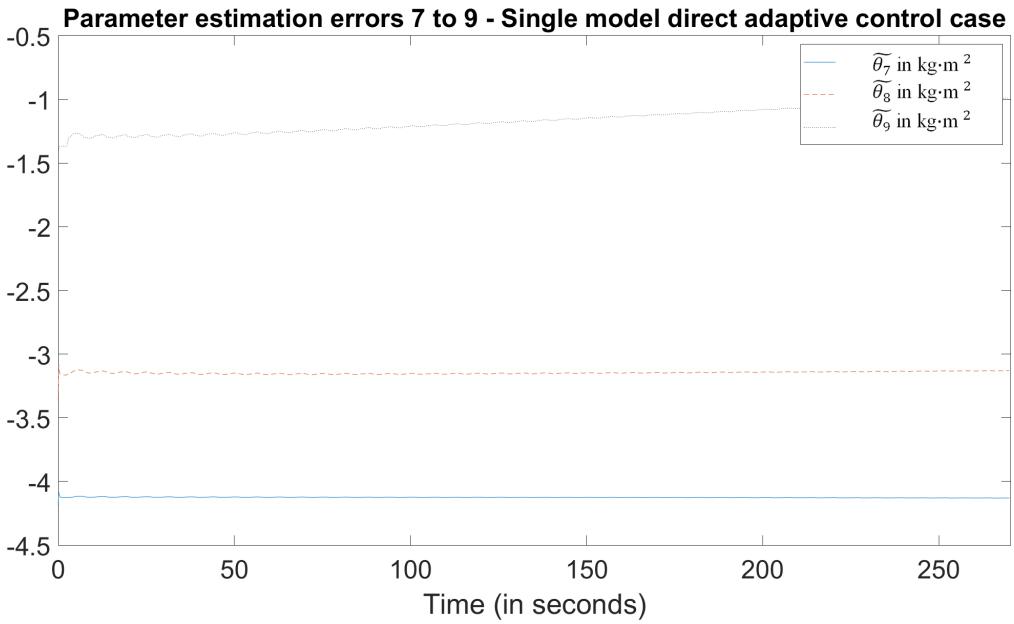

Figure 4.5: Parameter estimation errors 7 to 9 for single model direct adaptive control.

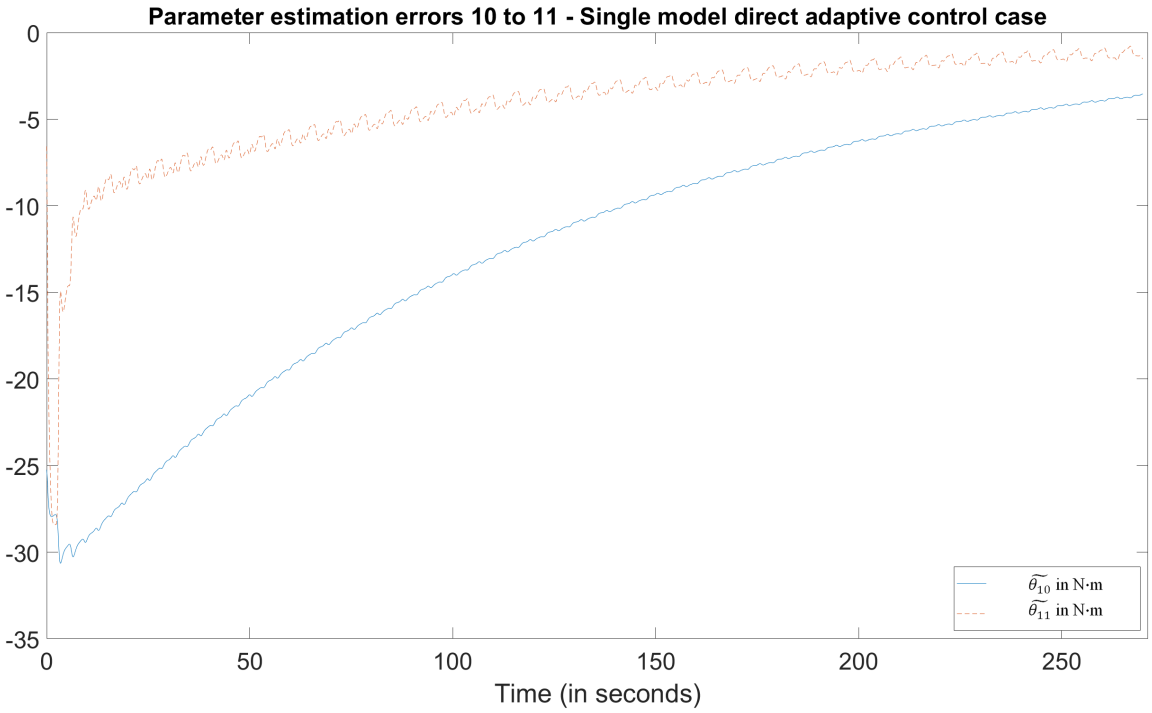

Figure 4.6: Parameter estimation errors 10 and 11 for single model direct adaptive control.

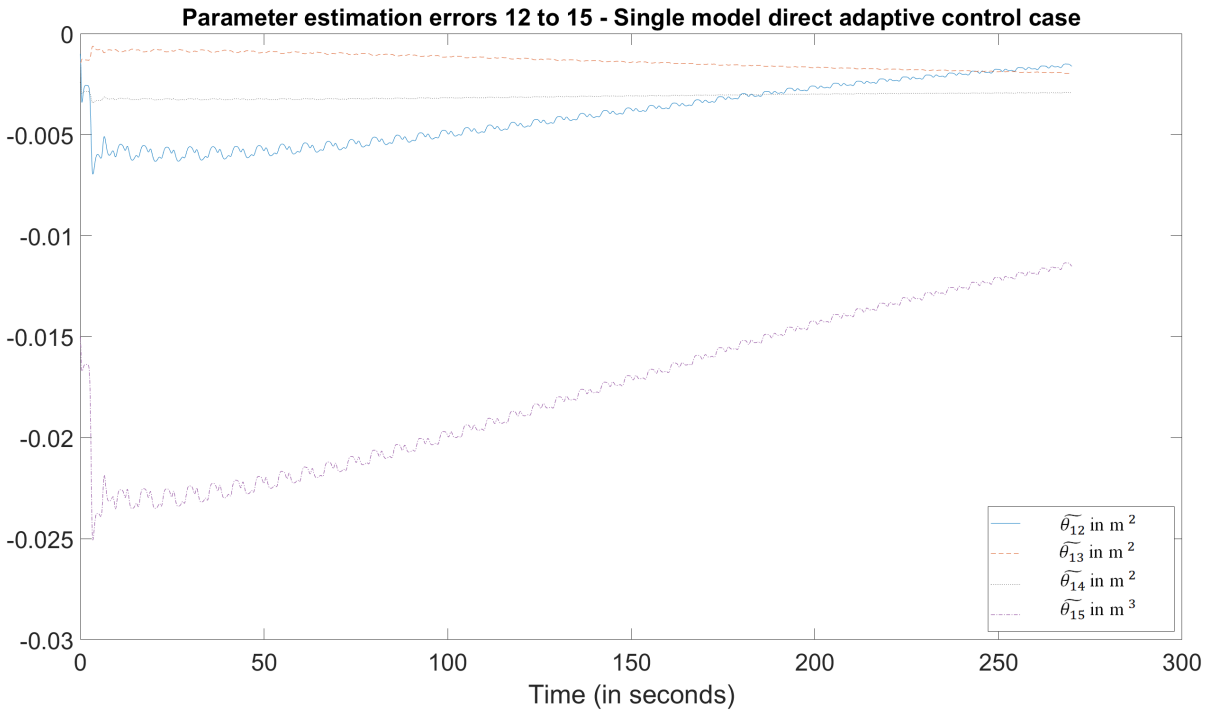

Figure 4.7: Parameter estimation errors 12 to 15 for single model direct adaptive control.

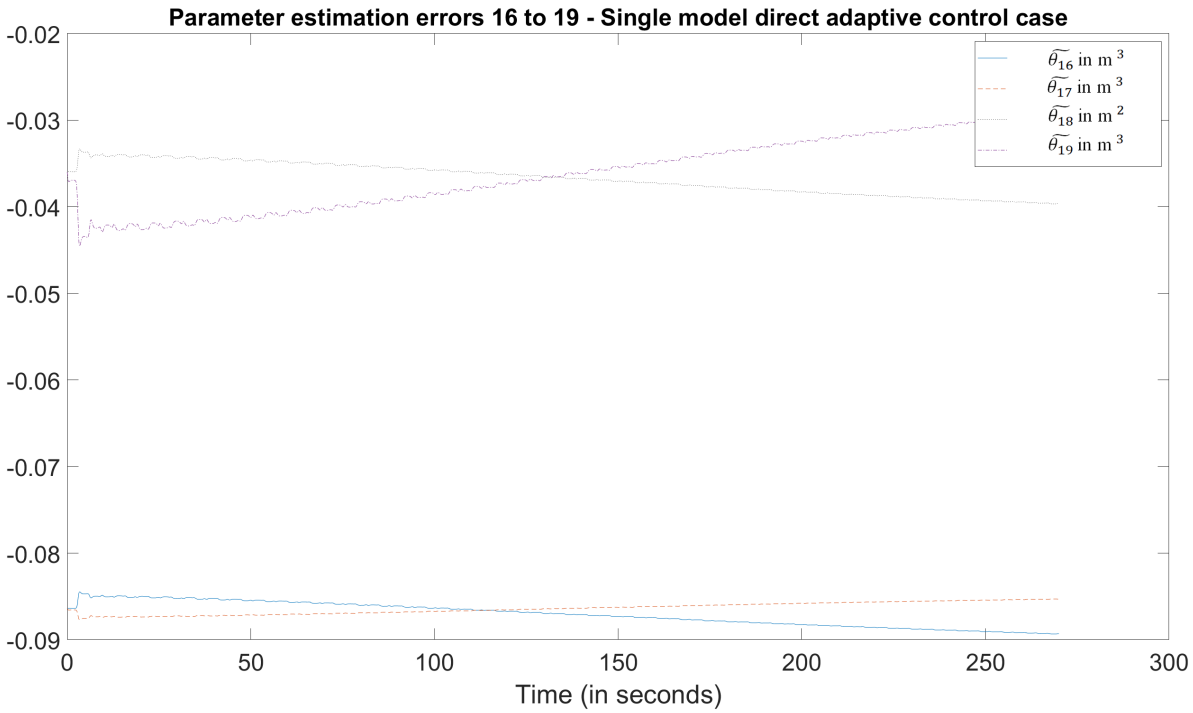

Figure 4.8: Parameter estimation errors 16 to 19 for single model direct adaptive control.

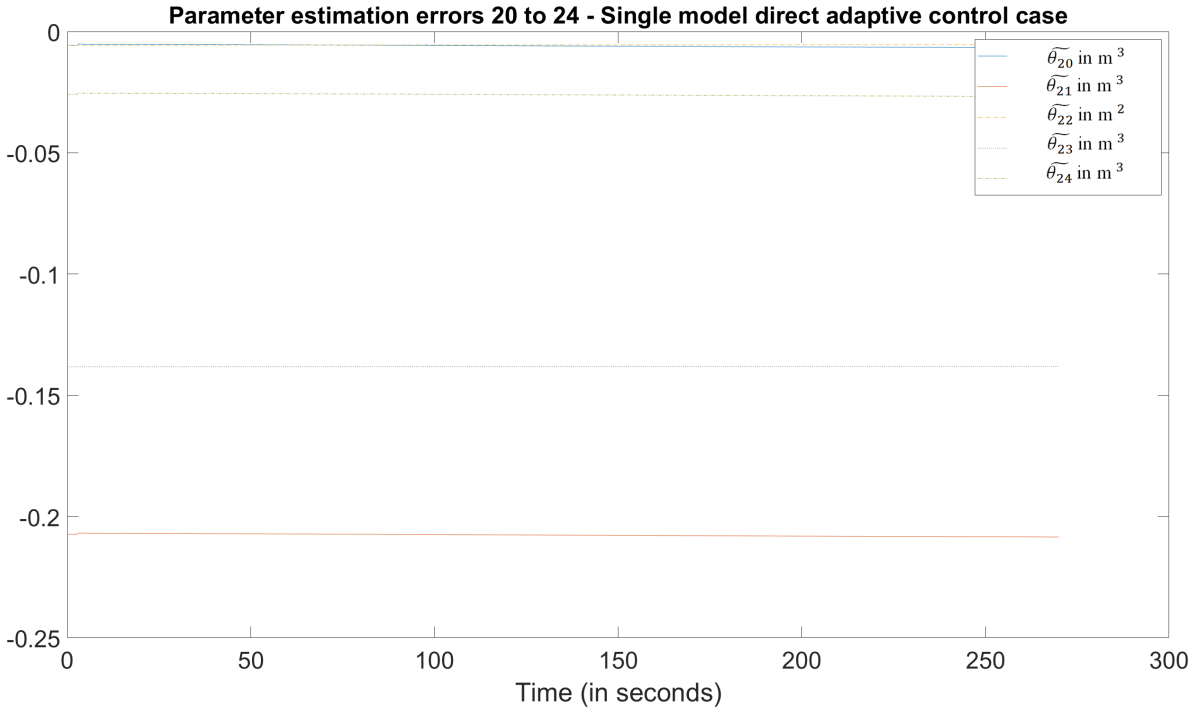

Figure 4.9: Parameter estimation errors 20 to 24 for single model direct adaptive control.

Indirect adaptive controller. Apply the controller (4.8) and the parameter update law (4.16) to the system (3.74) with the estimate of the system parameters being

$$\hat{\boldsymbol{\theta}}(t) = \begin{bmatrix} \hat{\theta}_1(t) & \hat{\theta}_2(t) & \hat{\theta}_3(t) & \cdots & \hat{\theta}_{24}(t) \end{bmatrix} \tag{4.35}$$

and set the initial values of the parameter estimates $\hat{\boldsymbol{\theta}}_0$ as 90% of the true values. Set the initial conditions of joint variables as $(q_1, q_2, \dot{q}_1, \dot{q}_2)|_{t=0} = (1, 1, -0.5, -0.2)$ and the desired joint variables as $q_{d1}(t) = q_{d2}(t) = \sin t$. Choose the controller parameters as (4.31) and the parameter update coefficients as

$$\boldsymbol{\Gamma} = diag\{0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.45, 0.9, 0.03, 0.03, \cdots, 0.03\}. \tag{4.36}$$

The tracking errors are shown in Figure 4.10, and the parameter estimation errors

are shown in Figures 4.11 to 4.17. The figures indicate that the tracking errors converge to zero asymptotically, and the parameter estimates are bounded.

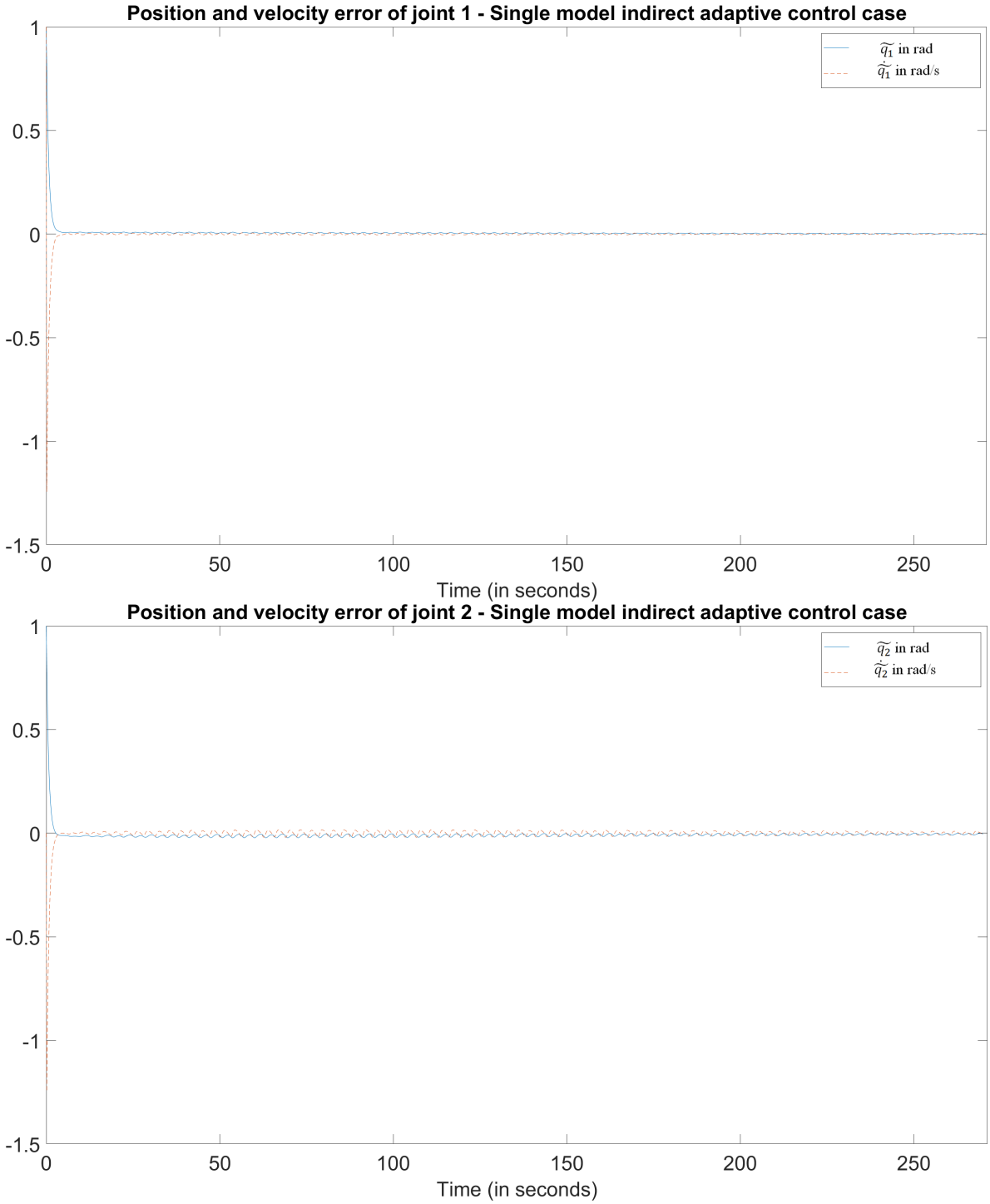

Figure 4.10: System tracking error for single model indirect adaptive control.

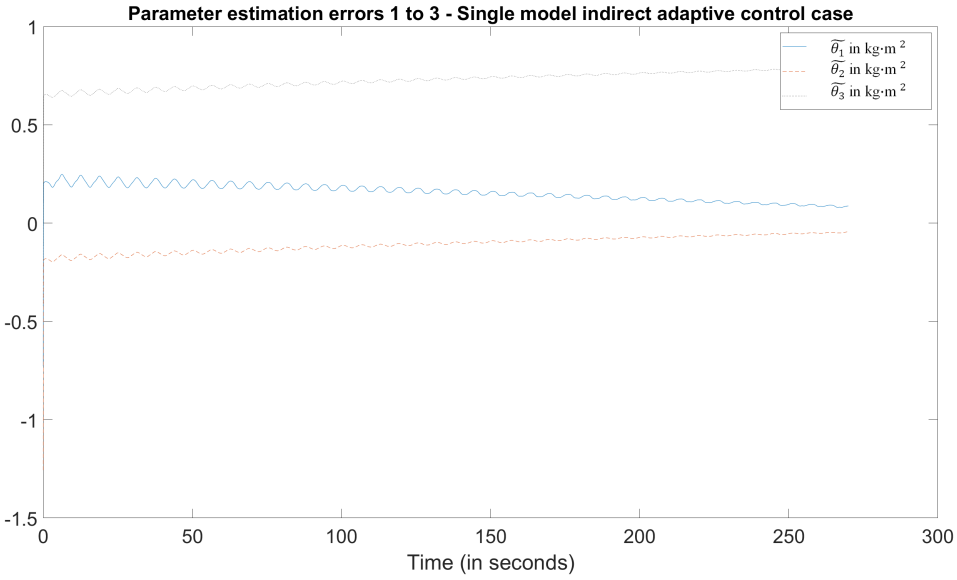

Figure 4.11: Parameter estimation errors 1 to 3 for single model indirect adaptive control.

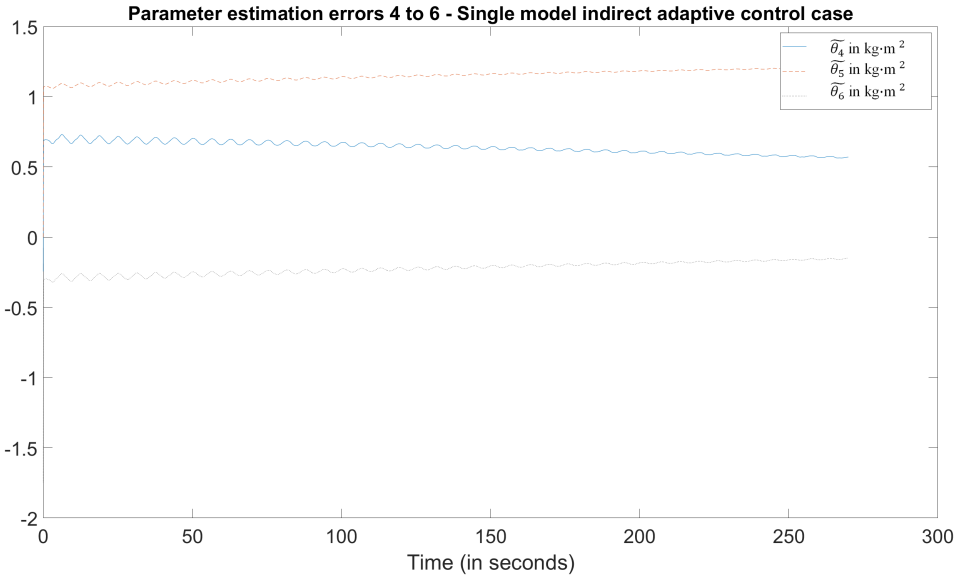

Figure 4.12: Parameter estimation errors 4 to 6 for single model indirect adaptive control.

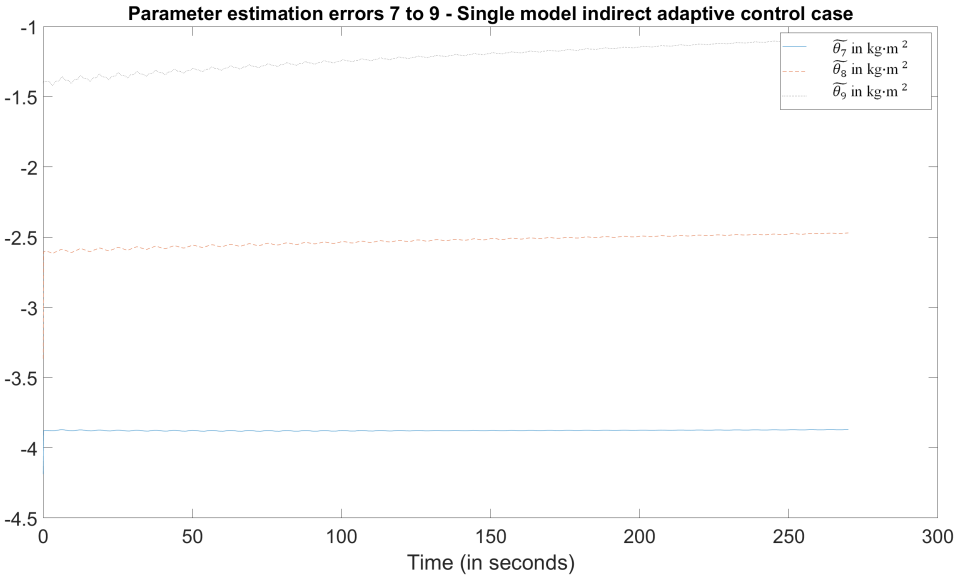

Figure 4.13: Parameter estimation errors 7 to 9 for single model indirect adaptive control.

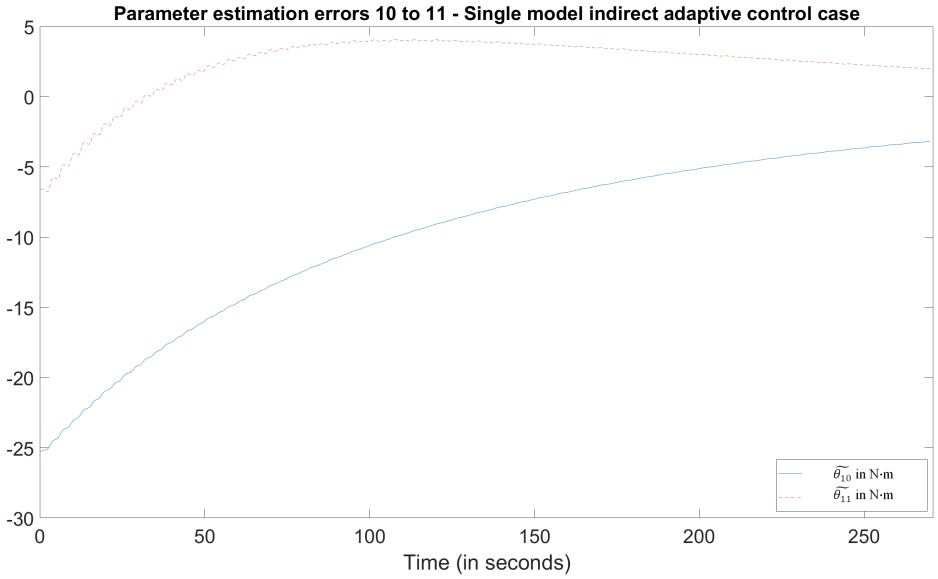

Figure 4.14: Parameter estimation errors 10 and 11 for single model indirect adaptive control.

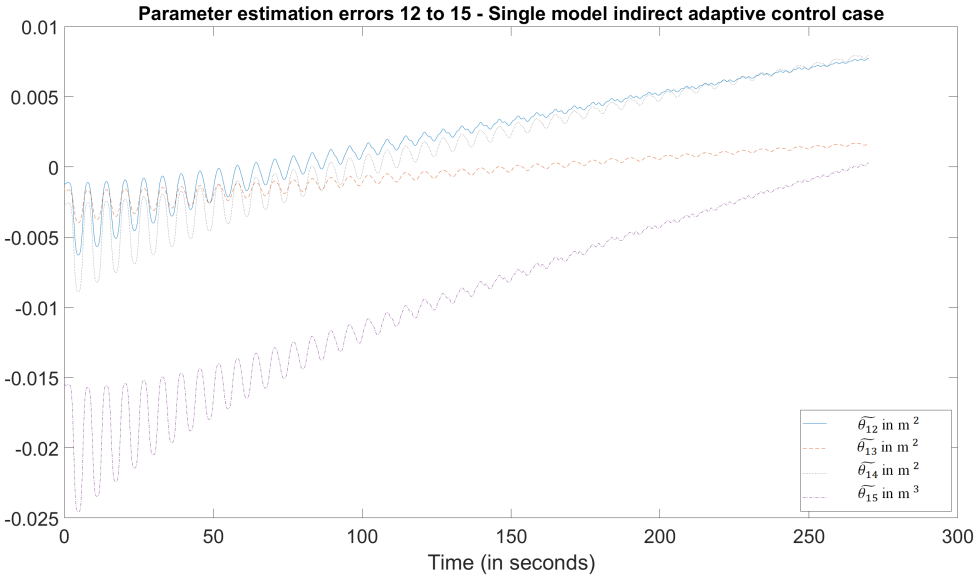

Figure 4.15: Parameter estimation errors 12 to 15 for single model indirect adaptive control.

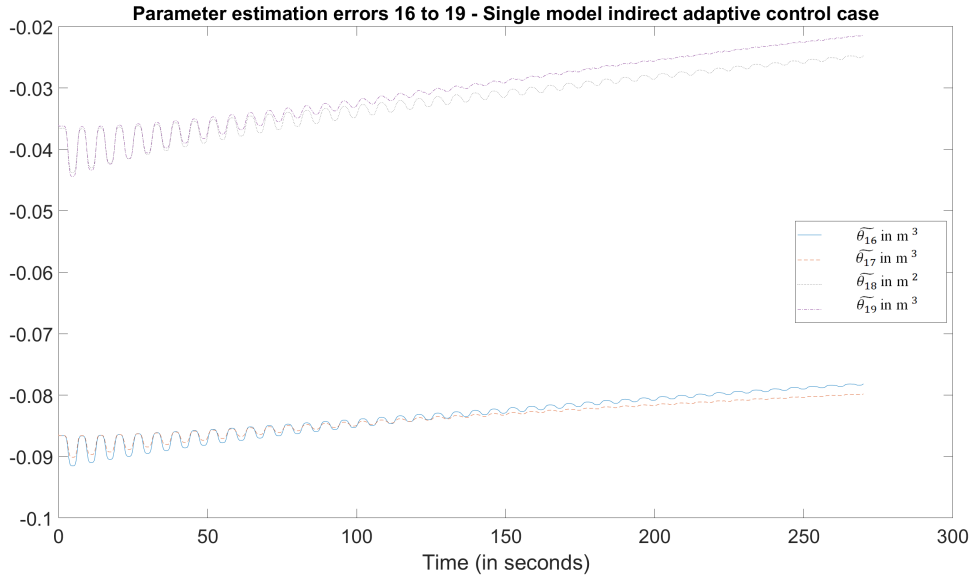

Figure 4.16: Parameter estimation errors 16 to 19 for single model indirect adaptive control.

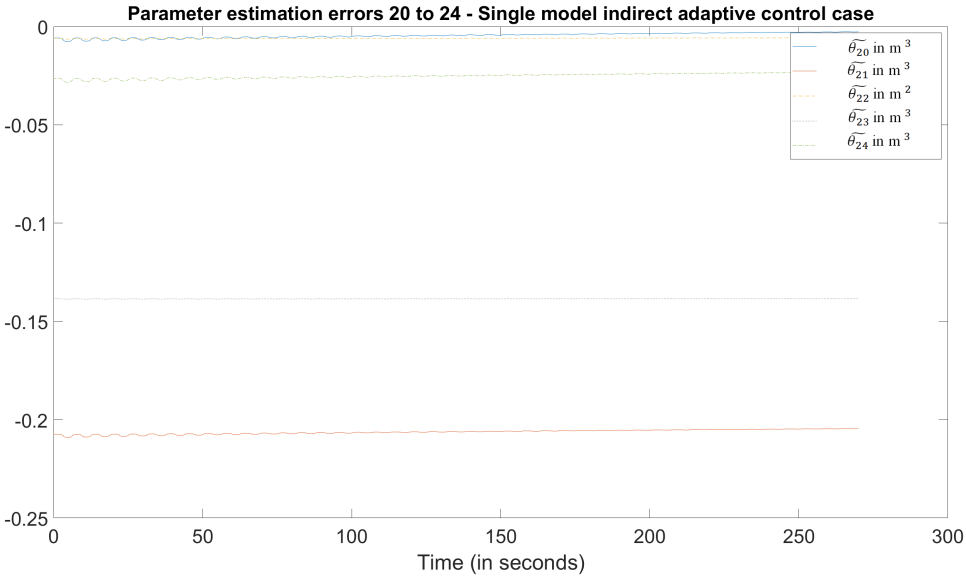

Figure 4.17: Parameter estimation errors 20 to 24 for single model indirect adaptive control.

Chapter 5

Adaptive Multiple-model Based Control Design

When the robot manipulator is moving in a fluid environment or carrying a load, the parameters of the robot dynamic equations are different from the original ones. The single-model based adaptive control is able to learn the parameters when they change, but this may cause a huge transient error during the learning process. To avoid this error, we adopt a multiple-model based adaptive control scheme so that when the model of the robot changes, the parameter estimates can rapidly convert to a relatively closer one with respect to the new true value. Thus after learning all the possible parameters, the transient error can be largely reduced after the controller with the least estimation error is chosen.

5.1 Controller Design

In this section, we develop an adaptive multiple-model control scheme to deal with the changing environment. First, we analyze the system models and show the reason why we adopt a multiple-model based control scheme, then we introduce the

adaptive controller we use to solve this problem, and finally, we analyze the stability of the closed-loop system.

5.1.1 System Models

Recall that the original dynamic equations of a robot manipulator is shown as in (2.13):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{5.1}$$

and the dynamic equations of a robot in a fluid environment is shown as in (3.61):

$$\boldsymbol{M}_m(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_m(\boldsymbol{q}) = \boldsymbol{\tau} \tag{5.2}$$

where $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{M}_m(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{C}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})$ have the same structure, but the mass terms are changed because of the added mass, and $\boldsymbol{g}(\boldsymbol{q})$ and $\boldsymbol{g}_m(\boldsymbol{q})$ are changed because of the buoyancy. In addition, there is an extra term $\boldsymbol{D}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})$ which models the effects of fluid friction and drag forces. The detailed description of the matrices $\boldsymbol{M}_m(\boldsymbol{q})$, $\boldsymbol{C}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})$, $\boldsymbol{g}_m(\boldsymbol{q})$ and $\boldsymbol{D}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})$ can be found in (3.34)-(3.47), (3.52)-(3.53) and (3.54)-(3.59), respectively.

Recall that for single-model indirect adaptive control, the controller structure is

$$\boldsymbol{u} = \hat{\boldsymbol{M}}_m(\boldsymbol{q})\boldsymbol{a}_q + \hat{\boldsymbol{C}}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}_m(\boldsymbol{q}), \tag{5.3}$$

where $\boldsymbol{u}$ is the input torque. The adaptive update law using the prediction error is

$$\dot{\hat{\boldsymbol{\theta}}}(t) = -\boldsymbol{\Gamma}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})^T\tilde{\boldsymbol{\tau}}. \tag{5.4}$$

Consider the following condition: a robot manipulator is rapidly submerged by

water or moving out of the water. Suppose the manipulator is submerged at time t_c, then the dynamic model of the robot manipulator becomes

$$\boldsymbol{M}(\boldsymbol{q},t)\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},t)\dot{\boldsymbol{q}} + \boldsymbol{D}(\boldsymbol{q},\dot{\boldsymbol{q}},t)\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q},t) = \boldsymbol{\tau}, \tag{5.5}$$

where

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q},t) &= \begin{cases} \boldsymbol{M}(\boldsymbol{q}), & \text{if } t \le t_c \\ \boldsymbol{M}_m(\boldsymbol{q}), & \text{if } t > t_c \end{cases}, \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},t) = \begin{cases} \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}), & \text{if } t \le t_c \\ \boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}}), & \text{if } t > t_c \end{cases} \\
\boldsymbol{D}(\boldsymbol{q},\dot{\boldsymbol{q}},t) &= \begin{cases} \boldsymbol{0}, & \text{if } t \le t_c \\ \boldsymbol{D}_m(\boldsymbol{q},\dot{\boldsymbol{q}}), & \text{if } t > t_c \end{cases}, \boldsymbol{g}(\boldsymbol{q},t) = \begin{cases} \boldsymbol{g}(\boldsymbol{q}), & \text{if } t \le t_c \\ \boldsymbol{g}_m(\boldsymbol{q}), & \text{if } t > t_c \end{cases},
\end{aligned} \tag{5.6}$$

which also has the form of

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}},t)\boldsymbol{\theta}^*(t) \tag{5.7}$$

where

$$\boldsymbol{\theta}^*(t) = \begin{cases} \boldsymbol{\theta}^*, & \text{if } t < t_c \\ \boldsymbol{\theta}^*_m, & \text{if } t \ge t_c. \end{cases} \tag{5.8}$$

and

$$\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}},t) = \begin{cases} \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}), & \text{if } t < t_c \\ \boldsymbol{Y}_m(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}), & \text{if } t \ge t_c. \end{cases} \tag{5.9}$$

For a system described in (5.5)-(5.9), a single-model based adaptive controller may need to go through a learning process, which increases the transient error. Thus, a multiple-model adaptive control scheme is considered.

5.1.2 Multiple Model Adaptive Controller

Here we propose a multiple-model based adaptive controller to avoid having huge transient errors when the robot dynamic model changes. Before the environment changes, we have no prior knowledge on what environment it will be, so multiple models and the performance index can make sure that the most suitable model can be adopted. The controller consists of multiple identification models with corresponding parameter estimate initial values. Based on (4.14), the multiple models can be chosen as

$$\hat{\boldsymbol{\tau}}_i = \boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\hat{\boldsymbol{\theta}}_i(t), i = 1, 2, \cdots, N \tag{5.10}$$

where each model i is an estimate of one corresponding environment, $\hat{\boldsymbol{\theta}}_i$ is the parameter estimate of the environmental system parameters $\boldsymbol{\theta}_i^*$, and each model has a corresponding regressor $\boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$. Note that each $\boldsymbol{\theta}_i^*$ and $\boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ can have much different structures, respectively. The input torque of these models are computed as

$$\boldsymbol{u}_i = \hat{\boldsymbol{M}}_i \boldsymbol{a}_q + \hat{\boldsymbol{C}}_i(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}_i(\boldsymbol{q}). \tag{5.11}$$

Note that $\hat{\boldsymbol{\tau}}_i$ is the torque prediction vector of each model, and $\boldsymbol{u}_i$ is the actual torque input of the manipulator computed from each model.

Then the prediction error of each model is given as

$$\tilde{\boldsymbol{\tau}}_i = \hat{\boldsymbol{\tau}}_i - \boldsymbol{\tau}, i = 1, 2, \cdots, N, \tag{5.12}$$

where $\boldsymbol{\tau}$ is the selected actual torque input.

The switching strategy is described as follows. Suppose we know the time when the environment condition will change, and we have N models in total. At the time t, only one among the N models is adopted to obtain the controller and be updated

by the adaptation law. Other models remain constant until adopted. The adoption follows the rule that the model with the minimal performance index is adopted. Then the torque input is selected as

$$\boldsymbol{\tau} = \boldsymbol{u}_j \tag{5.13}$$

where j is the id of the selected model, and the adaptation law is obtained as

$$\dot{\tilde{\boldsymbol{\theta}}}_i(t) = \begin{cases} -\boldsymbol{\Gamma}_i \boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})^T \tilde{\boldsymbol{\tau}}_i, & \text{if } i = j \\ \\ \boldsymbol{0}, & \text{otherwise} \end{cases} \tag{5.14}$$

Parameter Projection. In the case when we have a prior knowledge on the range of the parameter vector $\boldsymbol{\theta}_i$, we can limit the parameters in the range by using a parameter projection design. The parameter region of the j-th parameter in the i-th model is defined as $[\theta^a_{i,j}, \theta^b_{i,j}], j = 1, 2, \ldots, n_\theta$, such that $\theta^*_{i,j} \in [\theta^a_{i,j}, \theta^b_{i,j}], j = 1, 2, \ldots, n_\theta$, for $\boldsymbol{\theta}^*_i = [\theta^*_{i,1}, \theta^*_{i,2}, \ldots, \theta^*_{i,n_\theta}]^T, i = 1, 2, \ldots, N$. In our parameter update law design, let

$$g_i(t) = -\boldsymbol{\Gamma}_i \boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})^T \tilde{\boldsymbol{\tau}}_i \tag{5.15}$$

and

$$\dot{\tilde{\boldsymbol{\theta}}}_i(t) = g_i(t) - h_i(t) \tag{5.16}$$

when the model i is chosen, where

$$h_{i,j}(t) = \begin{cases} & \text{if } \theta_{i,j}(t) \in (\theta^a_{i,j}, \theta^b_{i,j}), \text{ or} \\ 0, & \text{if } \theta_{i,j}(t) = \theta^a_{i,j}, \ g_{i,j}(t) \geq 0, \text{ or} \\ & \text{if } \theta_{i,j}(t) = \theta^b_{i,j}, \ g_{i,j}(t) \leq 0, \\ \\ -g_{i,j}(t), & \text{otherwise} \end{cases} \tag{5.17}$$

and choose the initial estimates as

$$\theta_{i,j}(0) \in [\theta_{i,j}^a, \theta_{i,j}^b]. \tag{5.18}$$

With the prior knowledge $[\theta_{i,j}^a, \theta_{i,j}^b], j = 1, 2, \ldots, n_\theta$ and the projection function $h_i(t)$, we can ensure that the parameter estimates are in the region and so that $\boldsymbol{\theta}_i(t) \in L^\infty, i = 1, 2, \ldots, N$.

5.1.3 Stability Analysis

For each identification model, we have the Lyapunov function

$$V_i = \frac{1}{2}\tilde{\boldsymbol{\theta}}_i^T(t)\boldsymbol{\Gamma}_i\tilde{\boldsymbol{\theta}}_i(t) \tag{5.19}$$

and its derivative

$$\begin{aligned} \dot{V}_i &= \tilde{\boldsymbol{\theta}}_i^T\boldsymbol{\Gamma}_i^{-1}\dot{\tilde{\boldsymbol{\theta}}}_i \\ &= -\tilde{\boldsymbol{\theta}}_i^T\boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})^T\tilde{\boldsymbol{\tau}}_i \\ &= -\tilde{\boldsymbol{\tau}}_i^T\tilde{\boldsymbol{\tau}}_i \\ &\leq 0 \end{aligned} \tag{5.20}$$

which indicates that $\tilde{\boldsymbol{\tau}}_i \in L^2$ and $\tilde{\boldsymbol{\theta}}_i \in L^\infty$. Then from the equation (4.20)

$$\begin{aligned} \tilde{\boldsymbol{\tau}} &= \hat{\boldsymbol{\tau}} - \boldsymbol{\tau} \\ &= (\hat{\boldsymbol{M}}\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q})) - (\hat{\boldsymbol{M}}\boldsymbol{a}_q + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q})) \\ &= \hat{\boldsymbol{M}}(\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_1\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_0\tilde{\boldsymbol{q}}), \end{aligned} \tag{5.21}$$

we have

$$\tilde{\boldsymbol{q}} = \left(s^2\boldsymbol{I} + s\boldsymbol{K}_D + \boldsymbol{K}_P\right)\hat{\boldsymbol{M}}_i^{-1}(\boldsymbol{q})\tilde{\tau}_i \tag{5.22}$$

which shows that

$$\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}} \in L^2, \tilde{\boldsymbol{q}} \in L^\infty. \tag{5.23}$$

Thus for each selected model, all signal is bounded, and the trajectory tracking error $\tilde{\boldsymbol{q}} \to 0$ asymptotically.

5.2 Simulation Study

In this section we show the performance of multiple model switching and the advantage of using switching strategy.

Dynamic models. Suppose the robot manipulator is moving in the atmosphere in $t \in [0, 90)$, then submerged by water in $t \in [90, 180)$, and finally move out in $t \in [180, 270]$. The dynamic model of the robot is as shown in (2.31) and (2.32)

$$\begin{aligned}\boldsymbol{\tau}_2 = &(\ddot{q}_1(m_2 l_{2c}^2 + m_2 l_1 l_{2c}\cos(q_2) + I_2) \\ &+ \ddot{q}_2(m_2 l_1 l_{2c}\cos(q_2) + I_2) \\ &+ \dot{q}_1(m_2 l_1 l_{2c}\sin(q_2)\dot{q}_1) \\ &+ m_2 l_{2c} g\cos(q_1 + q_2))k\end{aligned} \tag{5.24}$$

and

$$\begin{aligned}\boldsymbol{\tau}_1 = &(\ddot{q}_1(m_1 l_{1c}^2 + m_2 l_{2c}^2 + m_2 l_1^2 + 2m_2 l_1 l_{2c}\cos(q_2) + I_1 + I_2) \\ &+ \ddot{q}_2(m_2 l_{2c}^2 + m_2 l_1 l_{2c}\cos(q_2) + I_2) \\ &+ \dot{q}_1(-m_2 l_1 l_{2c}\sin(q_2)\dot{q}_2) \\ &+ \dot{q}_2(-m_2 l_1 l_{2c}\sin(q_2)(\dot{q}_1 + \dot{q}_2)) \\ &+ (m_1 l_{1c} + m_2 l_1)g\cos(q_1) + m_2 l_{2c} g\cos(q_1 + q_2))k,\end{aligned} \tag{5.25}$$

and for the underwater case, the dynamic model is shown in (3.71) and (3.73)

$$\begin{aligned}\tau_1 = &\ddot{q}_1(M_{22}l_{2c}^2 + 2M_{22}l_1 l_{2c}\cos(q_2) + M_{22}l_1^2\cos^2(q_2) + M_{21}l_1^2\sin^2(q_2) \\ &\quad + M_{12}l_{1c}^2 + I_{m1} + I_{m2}) \\ &+ \ddot{q}_2(M_{22}l_{2c}^2 + M_{22}l_1 l_{2c}\cos(q_2) + I_{m2}) \\ &+ \dot{q}_1(M_{22}l_1 l_{2c}\sin(q_2)\dot{q}_1 + M_{22}l_1^2\sin(q_2)\cos(q_2)\dot{q}_1 \\ &\quad - M_{21}l_1 l_{2c}\sin(q_2)(\dot{q}_1 + \dot{q}_2) - M_{21}l_1^2\sin(q_2)\cos(q_2)\dot{q}_1) \\ &+ \dot{q}_2(-M_{21}l_1 l_{2c}\sin(q_2)(\dot{q}_1 + \dot{q}_2)) \\ &+ \dot{q}_1(D_{s_2}l_{2c}^2 + D_{s_2}l_1 l_{2c}\cos(q_2) + D_{d_2}l_{2c}\,\mathrm{sgn}(v_{r_2y})S_{21} + D_{s_2}l_1 l_{2c}\cos(q_2) \\ &\quad + D_{s_2}l_1^2\cos^2(q_2) + D_{d_2}l_1\cos(q_2)\,\mathrm{sgn}(v_{r_2y})S_{21} + D_{s_1}l_1^2\sin^2(q_2) \\ &\quad + D_{d_1}l_1^3\sin^3(q_2)\,\mathrm{sgn}(v_{r_2x})\dot{q}_1 + D_{s_2}l_{1c}^2 + D_{d_2}l_{1c}^3\,\mathrm{sgn}(v_{r_1y})\dot{q}_1) \\ &+ \dot{q}_2(D_{s_2}l_{2c}^2 + D_{d_2}l_{2c}^3\,\mathrm{sgn}(v_{r_2y})\dot{q}_2 + D_{s_2}l_1 l_{2c}\cos(q_2) \\ &\quad + D_{d_2}l_1 l_{2c}^2\cos(q_2)\,\mathrm{sgn}(v_{r_2y})\dot{q}_2) \\ &+ (m_2 l_1 + m_1 l_{1c} - l_1 l_2 \rho\pi r_2^2 - l_1 l_{1c}\rho\pi r_1^2)g\cos(q_1) \\ &\quad + (m_2 l_{2c} - l_2 l_{2c}\rho\pi r_2^2)g\cos(q_1 + q_2)\end{aligned} \tag{5.26}$$

$$
\begin{aligned}
\tau_2 = {} & \ddot{q}_1(M_{22}l_{2c}^2 + M_{22}l_1 l_{2c}\cos q_2 + I_{m2}) \\
& + \ddot{q}_2(M_{22}l_{2c}^2 + I_{m2}) \\
& + \dot{q}_1(M_{22}l_1 l_{2c}\sin q_2 \dot{q}_1) \\
& + \dot{q}_1(D_{s_2}l_{2c}^2 + D_{s_2}l_1 l_{2c}\cos q_2 + D_{d_2}l_{2c}\,\mathrm{sgn}(v_{r_2 y})S_{21}) \\
& + \dot{q}_2(D_{s_2}l_{2c}^2 + D_{d_2}l_{2c}^3\,\mathrm{sgn}(v_{r_2 y})\dot{q}_2) \\
& + l_{2c}(m_2 - l_2\rho\pi r_2^2)g\cos(q_1 + q_2).
\end{aligned} \tag{5.27}
$$

Let the true values of system parameters be as in (4.30)

$$
\begin{aligned}
& m_1 = 25.494, \quad m_2 = 21.245, \quad l_1 = 1.2, \quad l_{1,c} = 0.6, \\
& l_2 = 1, \quad l_{2,c} = 0.5, \quad r_1 = 0.05, \quad r_2 = 0.05, \\
& g = 9.8, \quad \rho_{fluid} = 997, \\
& D_s = diag\{0.04, 0.04\} \\
& D_d = diag\{0.8, 1.2\}.
\end{aligned} \tag{5.28}
$$

For the underwater case, the dynamic model can be expressed as

$$\boldsymbol{\tau} = \boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}_m^* \tag{5.29}$$

and for the atmosphere case, the dynamic model can be written in the same form as

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}^* \tag{5.30}$$

which can also be expressed as

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}^*(t) \tag{5.31}$$

where

$$\boldsymbol{\theta}^*(t) = \begin{cases} \boldsymbol{\theta}^*, & \text{if } t \in [0, 90), \text{or } t \in [180, 270], \\ \boldsymbol{\theta}^*_m, & \text{if } t \in [90, 180), \end{cases} \tag{5.32}$$

and $\boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, $\boldsymbol{\theta}^*_m$ are shown in (3.81) and (3.82) as

$$\begin{aligned} \boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \Bigg(& \begin{matrix} \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 & \ddot{q}_1 & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 & 2\cos(q_2)\ddot{q}_1 + \cos(q_2)\ddot{q}_2 + \sin(q_2)\dot{q}_1^2 \\ \ddot{q}_1 + \ddot{q}_2 & 0 & 0 & \ddot{q}_2 & 0 & \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 \end{matrix} \\ & \begin{matrix} \cos^2(q_2)\ddot{q}_1 + \sin(q_2)\cos(q_2)\dot{q}_1^2 & \sin^2(q_2)\ddot{q}_1 - \sin(q_2)\cos(q_2)\dot{q}_1^2 \\ 0 & 0 \end{matrix} \\ & \begin{matrix} -\sin(q_2)\dot{q}_1^2 - 2\sin(q_2)\dot{q}_1\dot{q}_2 - \sin(q_2)\dot{q}_2^2 & \cos(q_1) & \cos(q_1+q_2) & \dot{q}_1 + \dot{q}_2 & \dot{q}_1 \\ 0 & & 0 & \cos(q_1+q_2) & \dot{q}_1 + \dot{q}_2 & 0 \end{matrix} \\ & \begin{matrix} 2\cos(q_2)\dot{q}_1 & \operatorname{sgn}(v_{r_2y})\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\dot{q}_1\dot{q}_2 + \operatorname{sgn}(v_{r_2y})\dot{q}_2^2 & 2\operatorname{sgn}(v_{r_2y})\cos^2 q_2\dot{q}_1^2 \\ \cos(q_2)\dot{q}_1 & \operatorname{sgn}(v_{r_2y})\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\dot{q}_1\dot{q}_2 + \operatorname{sgn}(v_{r_2y})\dot{q}_2^2 & \operatorname{sgn}(v_{r_2y})\cos^2 q_2\dot{q}_1^2 \\ \operatorname{sgn}(v_{r_2y})\cos^2 q_2\dot{q}_1^2 + 2\cos^2(q_2)\operatorname{sgn}(v_{r_2y})\dot{q}_1\dot{q}_2 + \cos(q_2)\dot{q}_2 & \cos^2(q_2)\dot{q}_1 & \\ 0 & 0 & \end{matrix} \\ & \begin{matrix} 2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1\dot{q}_2 & \cos^3 q_2\operatorname{sgn}(v_{r_2y})\dot{q}_1^2 \\ 0 & 0 \end{matrix} \\ & \begin{matrix} \operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1\dot{q}_2 + \cos(q_2)\operatorname{sgn}(v_{r_2y})\dot{q}_2^2 & \sin^2(q_2)\dot{q}_1 \\ 2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1^2 + 2\operatorname{sgn}(v_{r_2y})\cos(q_2)\dot{q}_1\dot{q}_2 & 0 \\ \sin^3(q_2)\operatorname{sgn}(v_{r_2x})\dot{q}_1^2 & \operatorname{sgn}(v_{r_1y})\dot{q}_1^2 \\ 0 & 0 \end{matrix} \Bigg) \end{aligned} \tag{5.33}$$

$$\begin{aligned}\boldsymbol{\theta}_m^* = \Big(& M_{22}l_{2c}^2 \quad M_{12}l_{1c}^2 \quad I_{m1} \quad I_{m2} \quad 0 \quad M_{22}l_1l_{2c} \quad M_{22}l_1^2 \quad M_{21}l_1^2 \\ & M_{21}l_1l_{2c} \quad (m_2l_1 + m_1l_{1c} - l_1l_2\rho\pi r_2^2 - l_1l_{1c}\rho\pi r_1^2)g \\ & l_{2c}(m_2 - l_2\rho\pi r_2^2)g \quad D_{s_2}l_{2c}^2 \quad D_{s_2}l_{1c}^2 \quad D_{s_2}l_1l_{2c} \quad D_{d_2}l_{2c}^3 \\ & D_{d_2}l_1^2l_{2c} \quad D_{d_2}l_{2c}l_1^2 \quad D_{s_2}l_1^2 \quad D_{d_2}l_1l_{2c}^2 \quad D_{d_2}l_1^3 \quad D_{d_2}l_1l_{2c}^2 \\ & D_{s_1}l_1^2 \quad D_{d_1}l_1^3 \quad D_{d_2}l_{1c}^3 \Big)^T,\end{aligned} \tag{5.34}$$

and $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, $\boldsymbol{\theta}^*$ are

$$\begin{aligned}\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \Bigg(& \ddot{q}_1 + \ddot{q}_2 \quad \ddot{q}_1 \quad \ddot{q}_1 \quad \ddot{q}_1 + \ddot{q}_2 \quad \ddot{q}_1 \quad 2\cos(q_2)\ddot{q}_1 + \cos(q_2)\ddot{q}_2 + \sin(q_2)\dot{q}_1^2 \\ & \ddot{q}_1 + \ddot{q}_2 \quad 0 \quad 0 \quad \ddot{q}_2 \quad 0 \quad \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 \\ & \cos^2(q_2)\ddot{q}_1 + \sin(q_2)\cos(q_2)\dot{q}_1^2 \quad \sin^2(q_2)\ddot{q}_1 - \sin(q_2)\cos(q_2)\dot{q}_1^2 \\ & 0 \quad 0 \\ & -\sin(q_2)\dot{q}_1^2 - 2\sin(q_2)\dot{q}_1\dot{q}_2 - \sin(q_2)\dot{q}_2^2 \quad \cos(q_1) \quad \cos(q_1 + q_2) \\ & 0 \quad 0 \quad \cos(q_1 + q_2) \Bigg)\end{aligned} \tag{5.35}$$

$$\begin{aligned}\boldsymbol{\theta}^* = \Big(& m_2l_{2c}^2 \quad m_1l_{1c}^2 \quad I_1 \quad I_2 \quad m_2l_1^2 \quad m_2l_1l_{2c} \quad 0 \quad 0 \\ & m_2l_1l_{2c} \quad (m_1l_{1c} + m_2l_1)g \quad m_2l_{2c}g \Big)^T.\end{aligned} \tag{5.36}$$

We can see from the models that the regressors of the two models $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ and $\boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ are significantly different.

Adaptive controller. First we apply a single-model adaptive controller for the varying environment. For the estimation model, we use $\boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ and $\hat{\boldsymbol{\theta}}$, where

$$\hat{\boldsymbol{\theta}}(t) = \Big[\hat{\theta}_1(t) \quad \hat{\theta}_2(t) \quad \hat{\theta}_3(t) \quad \cdots \quad \hat{\theta}_{24}(t)\Big], \tag{5.37}$$

and set the initial values of the parameter estimates as 90% of the parameter true values of $\boldsymbol{\theta}_m^*$. Set the initial conditions of joint variables as $(q_1, q_2, \dot{q}_1, \dot{q}_2)|_{t=0} = (1, 1, -0.5, -0.2)$ and the desired joint variables as $q_{d1}(t) = q_{d2}(t) = \sin t$. Use the adaptive controller

$$\boldsymbol{u} = \hat{\boldsymbol{M}}_m(\boldsymbol{q})\boldsymbol{a}_q + \hat{\boldsymbol{C}}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}_m(\boldsymbol{q}), \tag{5.38}$$

and choose the controller parameters as

$$\boldsymbol{K}_1 = diag\{50, 50\}, \boldsymbol{K}_0 = diag\{75, 75\}, \tag{5.39}$$

and the parameter update law

$$\dot{\tilde{\boldsymbol{\theta}}} = \dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})^T\tilde{\boldsymbol{\tau}}. \tag{5.40}$$

with the coefficients

$$\boldsymbol{\Gamma} = diag\{0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.45, 0.9, 0.03, 0.03, \cdots, 0.03\}. \tag{5.41}$$

The tracking errors $\tilde{\boldsymbol{q}}$ and $\tilde{\dot{\boldsymbol{q}}}$ are shown in Figure 5.1, and parameter estimation errors are shown in Figures 5.2 to 5.8. From the figures, we can see that in $t \in [90, 180)$, the environmental factors are close to the estimated model, and the parameter estimate can converge and the system tracking error goes to zero. However, in $t \in [0, 90)$ and $t \in [180, 270]$, the environment parameters are far from the model, the parameter estimates start to oscillate and the tracking error does not converge to zero.

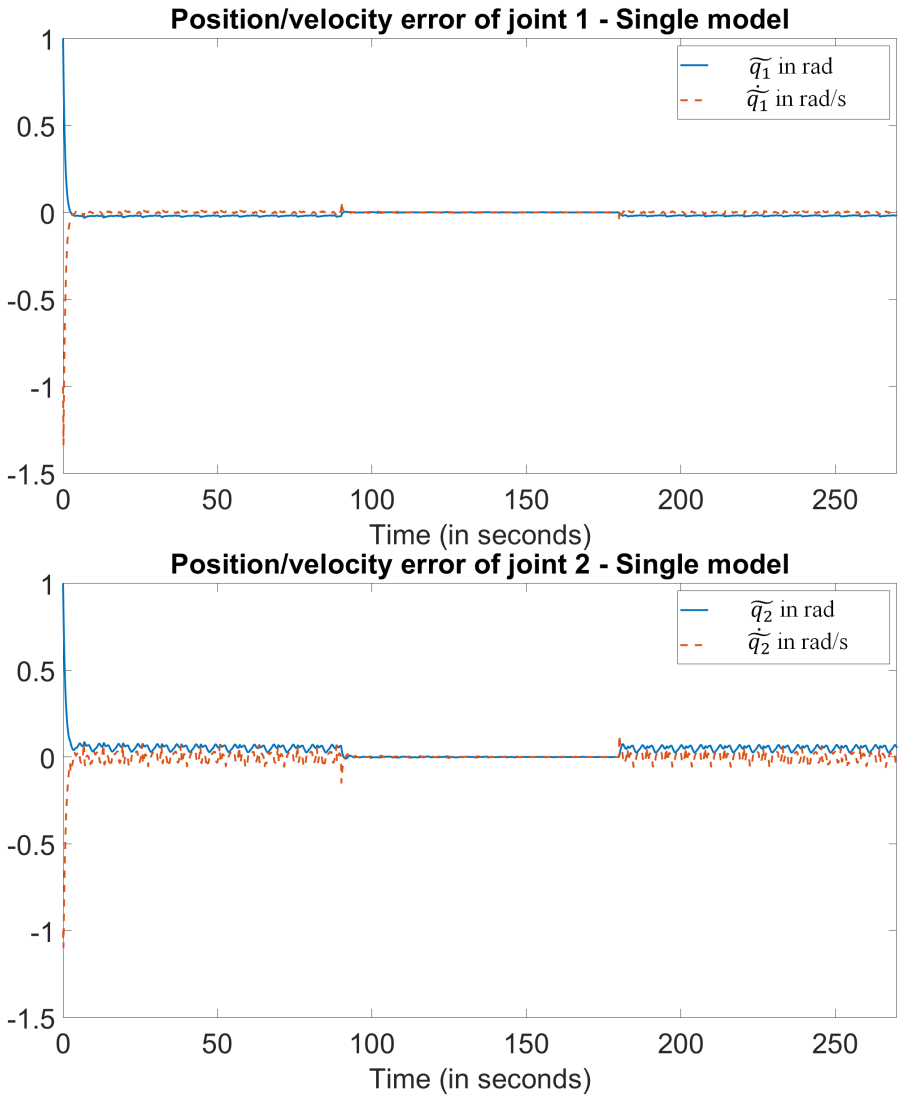

Figure 5.1: System tracking error for single model adaptive control when system parameters change.

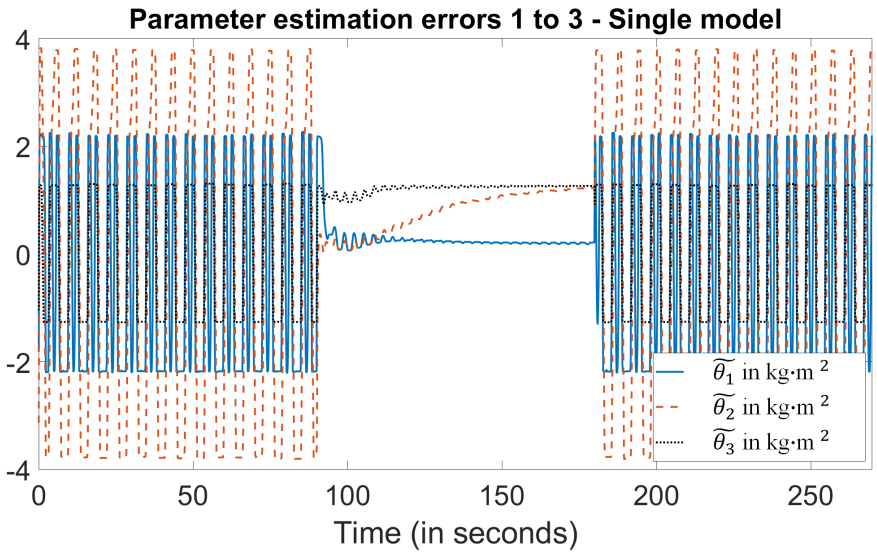

Figure 5.2: Parameter estimation errors 1 to 3 for single model adaptive control when system parameters change.

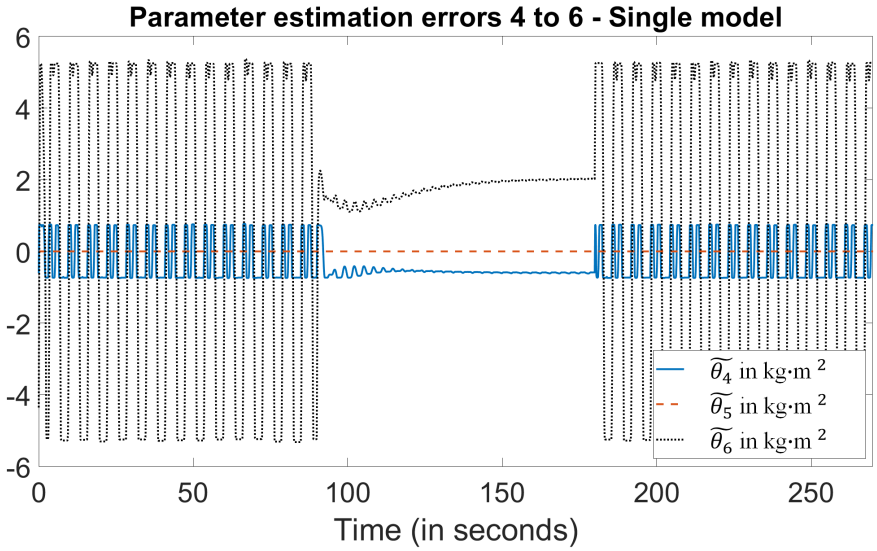

Figure 5.3: Parameter estimation errors 4 to 6 for single model adaptive control when system parameters change.

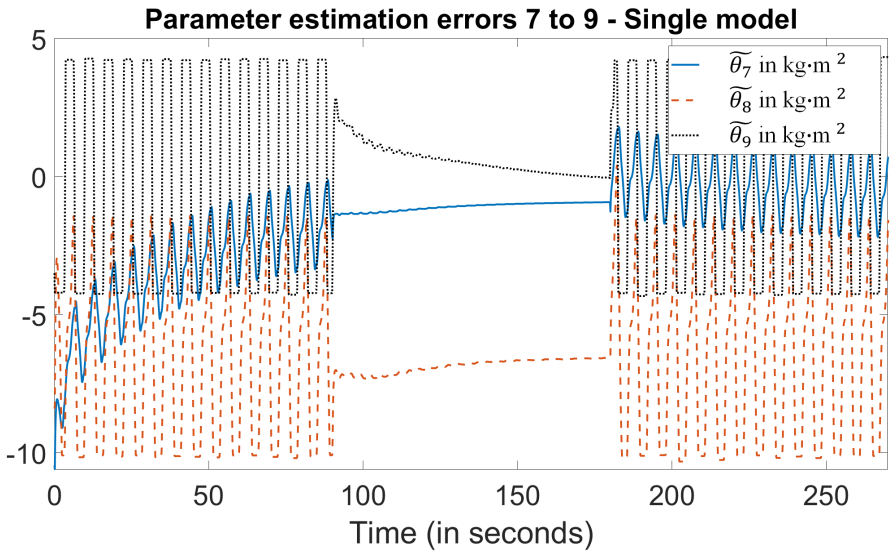

Figure 5.4: Parameter estimation errors 7 to 9 for single model adaptive control when system parameters change.

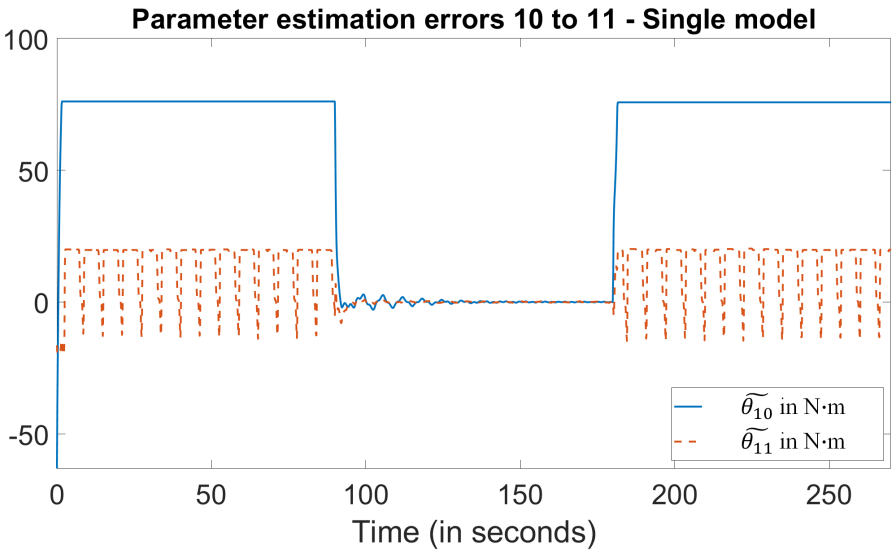

Figure 5.5: Parameter estimation errors 10 and 11 for single model adaptive control when system parameters change.

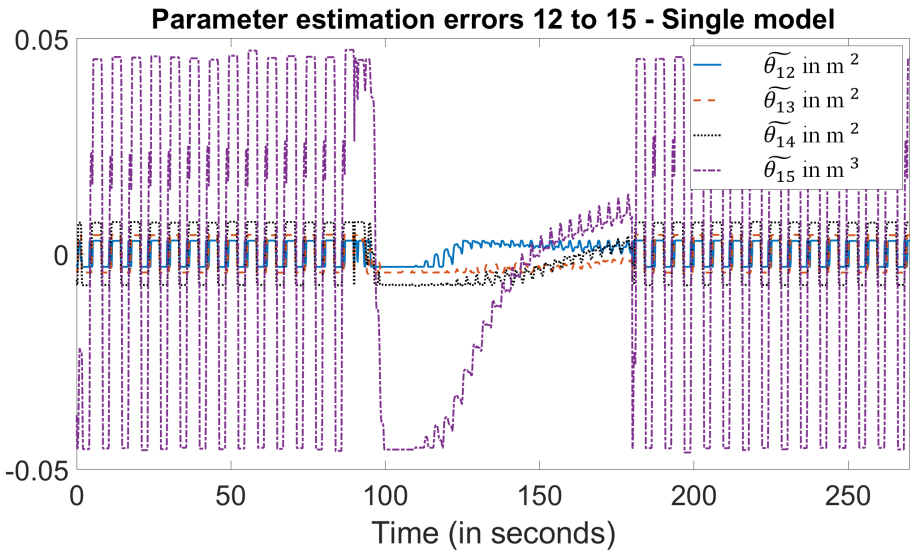

Figure 5.6: Parameter estimation errors 12 to 15 for single model adaptive control when system parameters change.

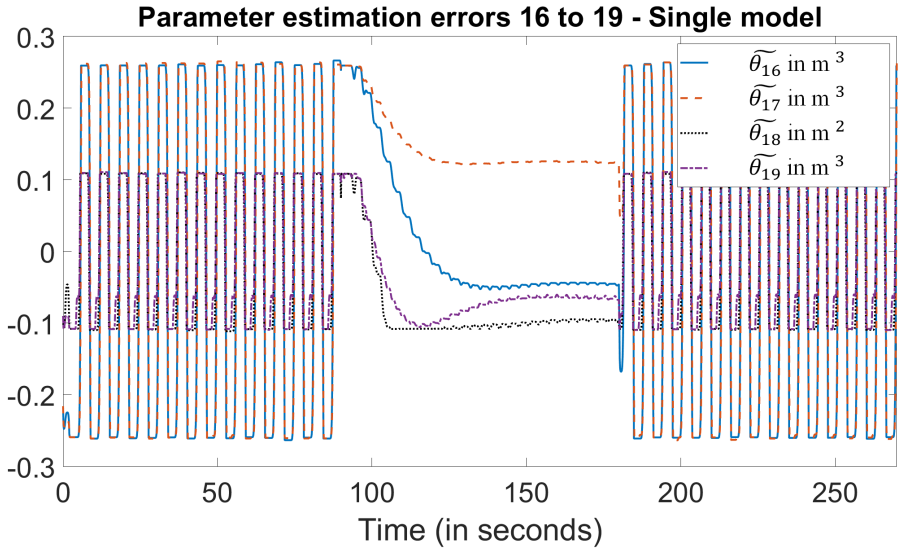

Figure 5.7: Parameter estimation errors 16 to 19 for single model adaptive control when system parameters change.

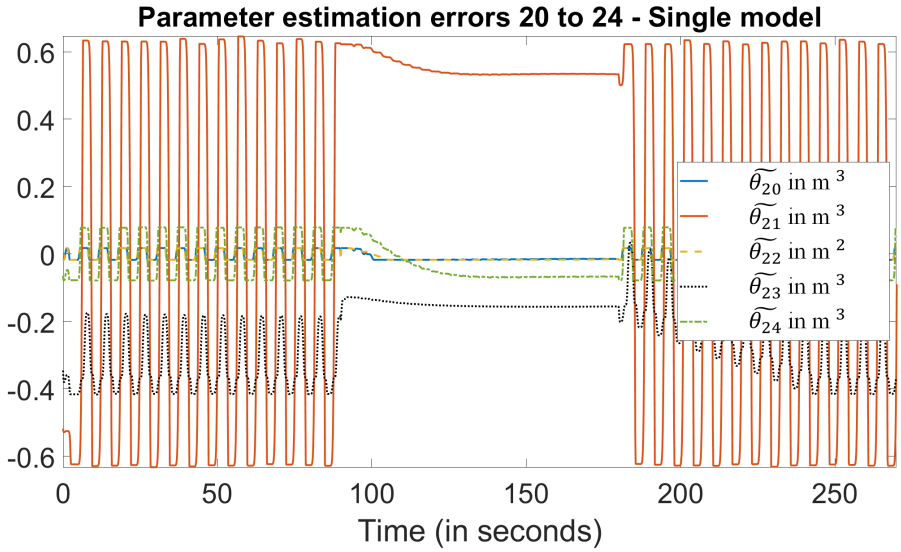

Figure 5.8: Parameter estimation errors 20 to 24 for single model adaptive control when system parameters change.

Multiple-model based adaptive controller. The dynamic model of the system is shown through (5.24) to (5.36). The robot moves into water at time $t_1 = 30$s, then move out of water at time $t_2 = 60$s. Let the initial conditions of joint variables be $(q_1, q_2, \dot{q}_1, \dot{q}_2)|_{t=0} = (1, 1, -0.5, -0.2)$ and the desired joint variables be $q_{d1}(t) = q_{d2}(t) = \sin t$. In this case, we use a controller with two identification models and switch between them to reduce the transient error. These models are

$$\hat{\boldsymbol{\tau}}_i = \boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\hat{\boldsymbol{\theta}}_i, i = 1, 2 \tag{5.42}$$

where the regressor $\boldsymbol{Y}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{Y}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ and $\boldsymbol{Y}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, the parameter estimates are $\hat{\boldsymbol{\theta}}_1$, $\hat{\boldsymbol{\theta}}_2$, and the initial values of the parameter estimates $\hat{\boldsymbol{\theta}}_1|_{t=0} = 0.9 * \boldsymbol{\theta}_m^*$, $\hat{\boldsymbol{\theta}}_2|_{t=0} = 0.85 * \boldsymbol{\theta}^*$, respectively. The controller for the first model is

$$\boldsymbol{u} = \hat{\boldsymbol{M}}_m(\boldsymbol{q})\boldsymbol{a}_q + \hat{\boldsymbol{C}}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{D}}_m(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}_m(\boldsymbol{q}) \tag{5.43}$$

and for the second model is

$$\boldsymbol{u} = \hat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{a}_q + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}(\boldsymbol{q}) \tag{5.44}$$

with the controller coefficients as

$$\boldsymbol{K}_1 = diag\{50, 50\}, \boldsymbol{K}_0 = diag\{75, 75\}. \tag{5.45}$$

Choose the parameter adaptation law as

$$\dot{\hat{\boldsymbol{\theta}}}_i(t) = \begin{cases} -\boldsymbol{\Gamma}_i \boldsymbol{Y}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})^T \tilde{\boldsymbol{\tau}}_i, & \text{if } i = j \\ \boldsymbol{0}, & \text{otherwise} \end{cases} \tag{5.46}$$

with the coefficients

$$\boldsymbol{\Gamma} = diag\{0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.45, 0.9, 0.03, 0.03, \cdots, 0.03\}. \tag{5.47}$$

and

$$\boldsymbol{\Gamma} = diag\{0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.45, 0.9\}. \tag{5.48}$$

In addition, we have the performance index

$$J_i(t) = \int_0^t e^{-\lambda(t-\sigma)} \tilde{\boldsymbol{\tau}}_i^T(\sigma) \tilde{\boldsymbol{\tau}}_i(\sigma) d\sigma, i = 1, 2 \tag{5.49}$$

to decide which model should be updated and used to generate the controller.

Discussion. The tracking errors of the system are shown in Figure 5.9. The parameter estimation errors of model 1 and 2 are shown in Figure 5.10 to 5.13 and 5.14 to 5.17, respectively. The simulate time periods are chosen so that the conver-

gence of the parameter estimates can be shown, although the tracking errors converge faster than the parameter estimates. The figures illustrate that the transient error is significantly reduced compared to the single-model case, and after switching the model, the system tracking error converge to zero and parameter estimates does not oscillate.

In practical use, There is a trade-off between the model number and the computation complexity. For each model added, the regressor, the torque prediction and the performance index should be computed separately, and more computation is needed than in the single model case. In addition, to obtain the regressor of each model, we need sensors on each link to measure the relative velocity to the fluid environment.

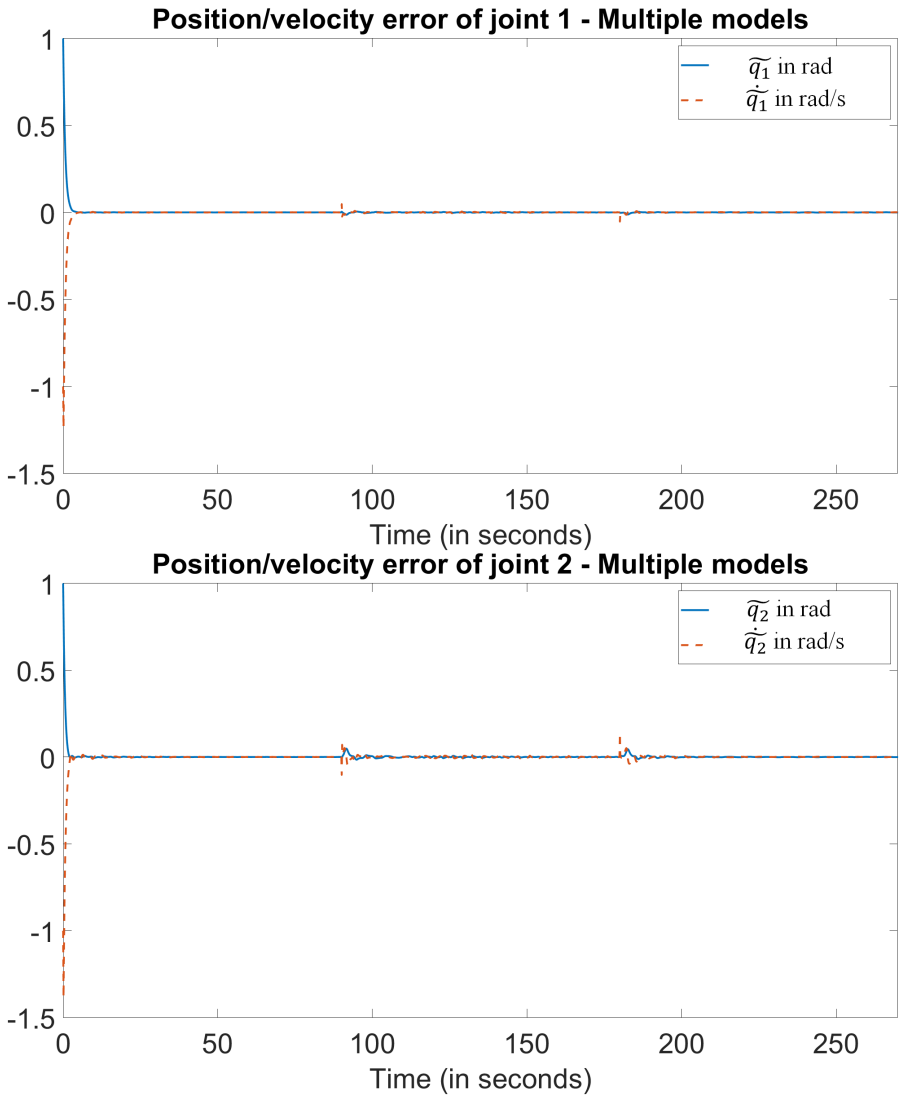

Figure 5.9: System tracking error for multiple model indirect adaptive control.

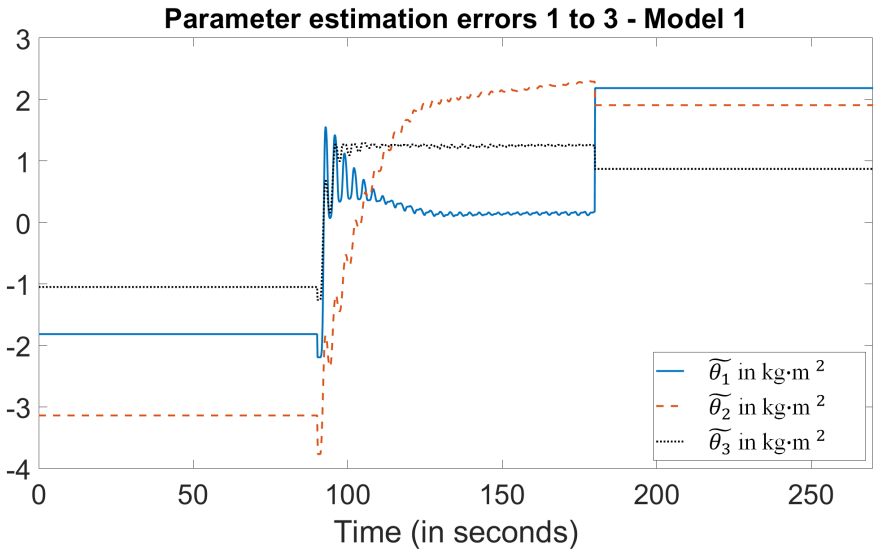

Figure 5.10: Parameter estimation errors 1 to 3 of model 1 for multiple model indirect adaptive control.

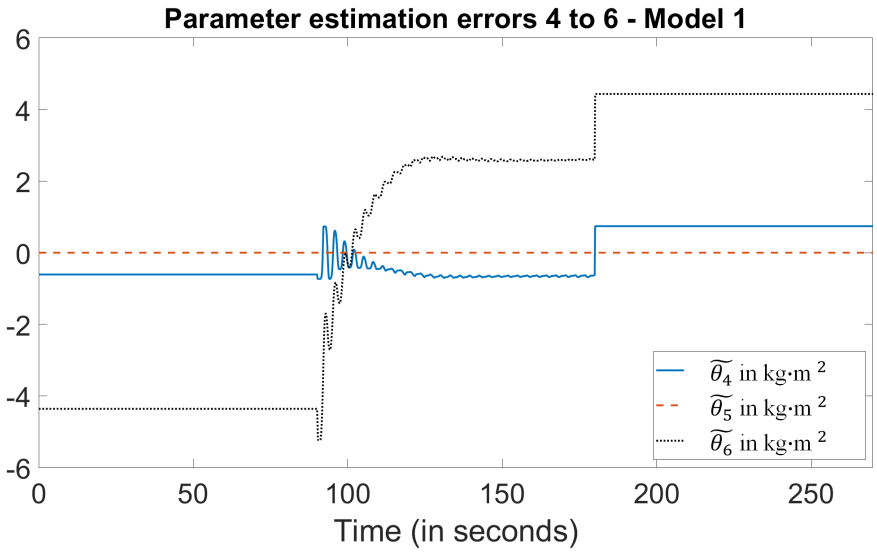

Figure 5.11: Parameter estimation errors 4 to 6 of model 1 for multiple model indirect adaptive control.

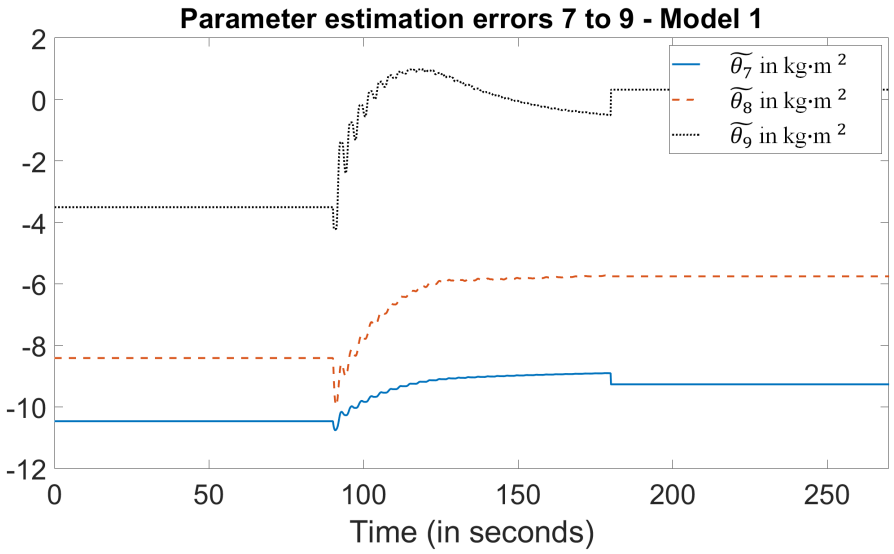

Figure 5.12: Parameter estimation errors 7 to 9 of model 1 for multiple model indirect adaptive control.

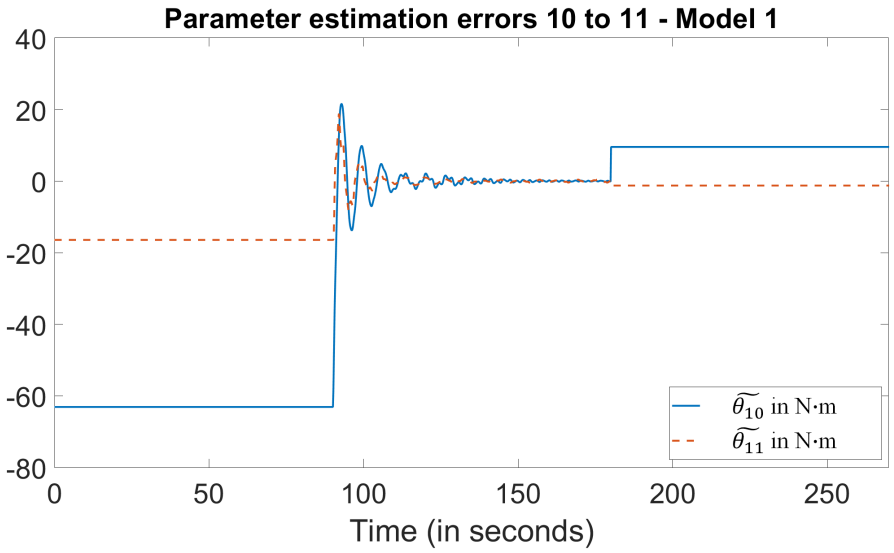

Figure 5.13: Parameter estimation errors 10 and 11 of model 1 for multiple model indirect adaptive control.

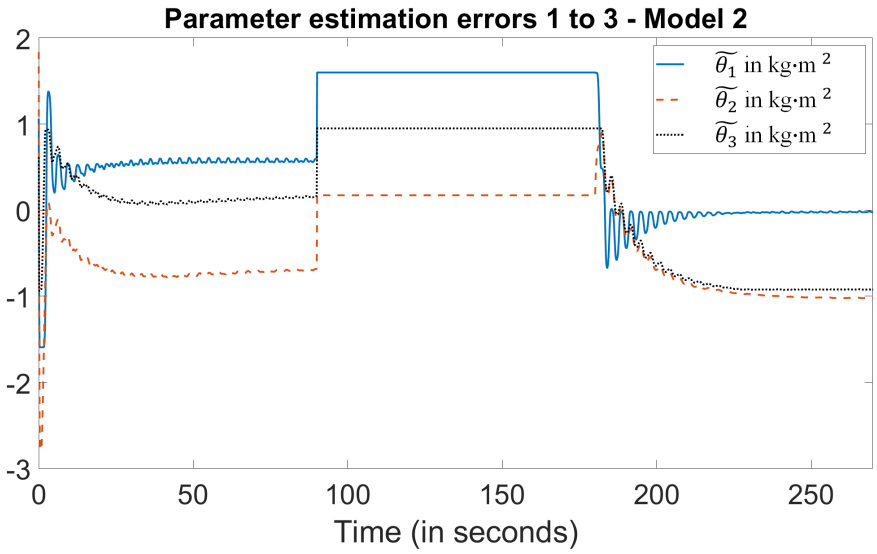

Figure 5.14: Parameter estimation errors 1 to 3 of model 2 for multiple model indirect adaptive control.

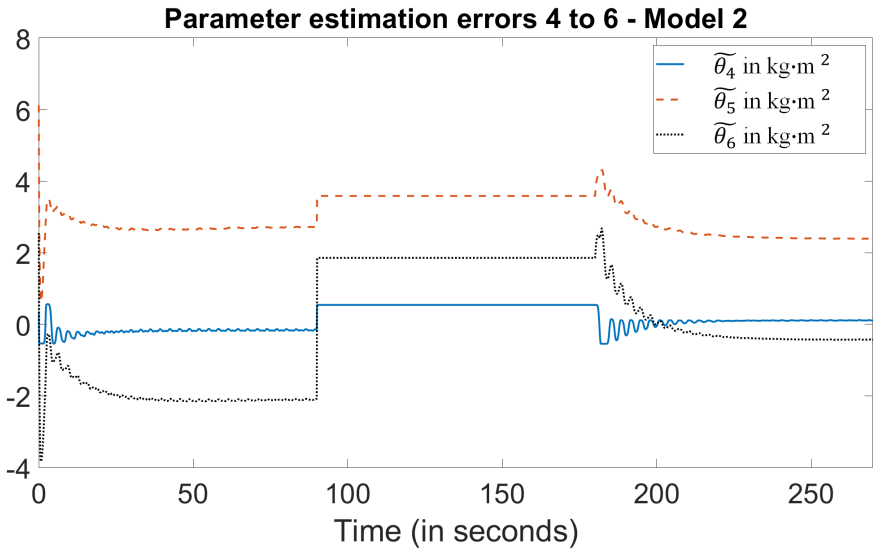

Figure 5.15: Parameter estimation errors 4 to 6 of model 2 for multiple model indirect adaptive control.

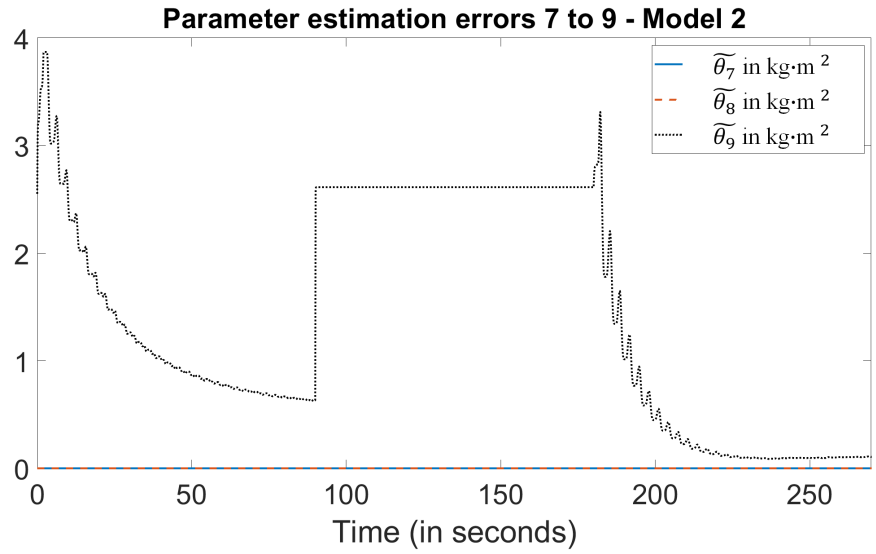

Figure 5.16: Parameter estimation errors 7 to 9 of model 2 for multiple model indirect adaptive control.

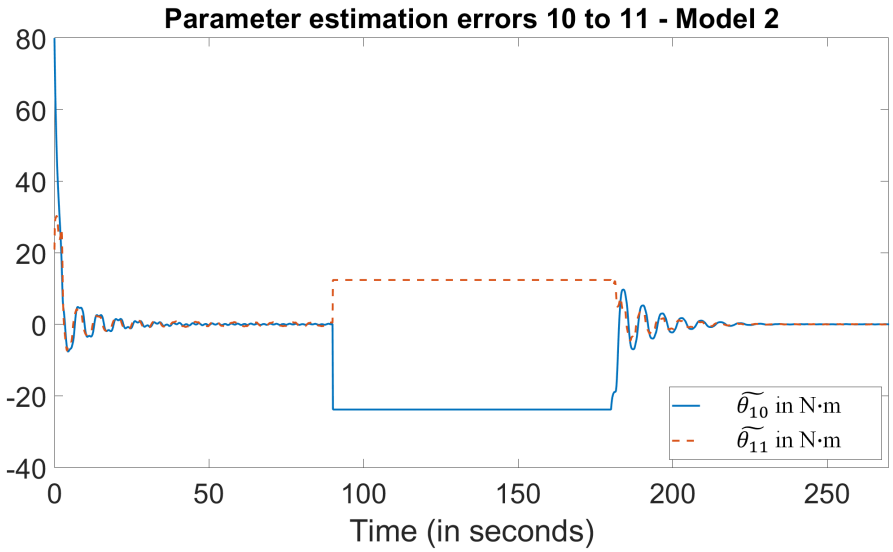

Figure 5.17: Parameter estimation errors 10 and 11 of model 2 for multiple model indirect adaptive control.

Chapter 6

Conclusions and Future Work

6.1 Summary and Conclusions

In this thesis, we have built the dynamic models of the robot manipulator moving in a varying environment, i.e., in gust wind or in/out of the water, and developed a multiple-model based adaptive control scheme for the manipulator. The background of the research topic and the research motivation are introduced at first. Following this, the effects of the fluid exerting on the robot are analyzed, and the dynamic model of the robot moving in the fluid is proposed. After gaining the knowledge of the model, we developed the model reference adaptive control accordingly. The direct and indirect adaptive control method for controlling the robot was introduced, then the multiple model adaptive control was presented to handle the case when the environment varies. The movement of a two-link planar robot manipulator going in and out of the water was considered in the simulation. The comparison work between a single model adaptive controller and a multiple model adaptive controller is proposed. The simulation result implied that the tracking of the robot is not disturbed by the variance of the environment parameters in the multiple model case,

which has better performance than the single model case.

6.2 Future Research Topics

Applying the multiple model adaptive control on the robot manipulator in the varying environment is a new attempt. In this work, we only discuss the general situations, but there are still many topics in this field to be addressed.

In Chapter 3, we discussed the effects that the fluid environment would have on the robot. However, when the robot is moving at high speed, the effects will be more complicated, and the structure of the system is different. Thus, the modeling of the robot in this situation is worth investigating. In addition, we only consider the robot link to be a cylinder, but in real life, according to the usage of the robot, links can be in different shapes. When putting this work into application, the specific shape of the robot should be considered to obtain the system model.

In Chapters 4 and 5, the adaptive control schemes were carried out for the robot moving in a varying environment. In these control schemes, we assumed that the joint variables could be exactly obtained. In real-life situations, the joint variable values are detected by sensors, and errors may exist between sensed values and real values. Thus, how to cancel the effect brought by observation error is a topic to be studied.

Bibliography

[1] John J Craig, Ping Hsu, and S Shankar Sastry. "Adaptive control of mechanical manipulators". In: *The International Journal of Robotics Research* 6.2 (1987), pp. 16–28.

[2] Jean-Jacques E Slotine and Weiping Li. "On the adaptive control of robot manipulators". In: *The international journal of robotics research* 6.3 (1987), pp. 49–59.

[3] Weiping Li and J-JE Slotine. "Indirect adaptive robot control". In: *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE. 1988, pp. 704–709.

[4] J-JE Slotine and Li Weiping. "Adaptive manipulator control: A case study". In: *IEEE transactions on automatic control* 33.11 (1988), pp. 995–1003.

[5] Kiyoshi Ioi and Kenji Itoh. "Modelling and simulation of an underwater manipulator". In: *Advanced Robotics* 4.4 (1989), pp. 303–317.

[6] H. Janocha and I. Papadimitriou. "Simulation of the dynamic behavior of robots in an extreme environment". In: *Robotics and Computer-Integrated Manufacturing* 8.3 (1991). Special Issue: Second International Conference on Robotics, pp. 163–169. ISSN: 0736-5845. DOI: https://doi.org/10.1016/0736-

5845(91)90016-L. URL: https://www.sciencedirect.com/science/article/pii/073658459190016L.

[7] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*. Vol. 199. 1. Prentice hall Englewood Cliffs, NJ, 1991.

[8] Odd Faltinsen. *Sea loads on ships and offshore structures*. Vol. 1. Cambridge university press, 1993.

[9] MK Ciliz and Kumpati S Narendra. "Multiple model based adaptive control of robotic manipulators". In: *Proceedings of 1994 33rd IEEE Conference on Decision and Control*. Vol. 2. IEEE. 1994, pp. 1305–1310.

[10] Benoît Lévesque and Marc J. Richard. "Dynamic Analysis of a Manipulator in a Fluid Environment". en. In: *The International Journal of Robotics Research* 13.3 (June 1994), pp. 221–231. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/027836499401300304. URL: http://journals.sagepub.com/doi/10.1177/027836499401300304 (visited on 10/12/2021).

[11] Kumpati S Narendra and Jeyendran Balakrishnan. "Improving transient response of adaptive control systems using multiple models and switching". In: *IEEE Transactions on automatic control* 39.9 (1994), pp. 1861–1866.

[12] M Kemal Ciliz and Kumpati S Narendra. "Adaptive control of robotic manipulators using multiple models and switching". In: *The International journal of robotics research* 15.6 (1996), pp. 592–610.

[13] I Schjølberg. "Modeling and control of underwater robotic systems". In: (1996).

[14] Kumpati S Narendra and Jeyendran Balakrishnan. "Adaptive control using multiple models". In: *IEEE transactions on automatic control* 42.2 (1997), pp. 171–187.

[15] Gianluca Antonelli and Stefano Chiaverini. "Adaptive tracking control of underwater vehicle-manipulator systems". In: *Proceedings of the 1998 IEEE International Conference on Control Applications (Cat. No. 98CH36104)*. Vol. 2. IEEE. 1998, pp. 1089–1093.

[16] Kortney N Leabourne and Stephen M Rock. "Model development of an underwater manipulator for coordinated arm-vehicle control". In: *IEEE Oceanic Engineering Society. OCEANS'98. Conference Proceedings (Cat. No. 98CH36259)*. Vol. 2. IEEE. 1998, pp. 941–946.

[17] Timothy W McLain and Stephen M Rock. "Development and experimental validation of an underwater manipulator hydrodynamic model". In: *The International Journal of Robotics Research* 17.7 (1998), pp. 748–759.

[18] Thor I Fossen. "Guidance and control of ocean vehicles". In: *University of Trondheim, Norway, Printed by John Wiley & Sons, Chichester, England, ISBN: 0 471 94113 1, Doctors Thesis* (1999).

[19] Pan-Mook Lee and J Yuh. "Application of non-regressor based adaptive control to an underwater mobile platform-mounted manipulator". In: *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)*. Vol. 2. IEEE. 1999, pp. 1135–1140.

[20] Junky Yuh and Jing Nie. "Application of non-regressor-based adaptive control to underwater robots: experiment". In: *Computers & Electrical Engineering* 26.2 (2000), pp. 169–179.

[21] Frank L Lewis, Darren M Dawson, and Chaouki T Abdallah. *Robot manipulator control: theory and practice*. CRC Press, 2003.

[22] Gang Tao. *Adaptive control design and analysis*. Vol. 37. John Wiley & Sons, 2003.

[23] Yongliang Zhu and Prabhakar R Pagilla. "Adaptive estimation of time-varying parameters in linear systems". In: *Proceedings of the 2003 American Control Conference, 2003.* Vol. 5. IEEE. 2003, pp. 4167–4172.

[24] M Kemal Ciliz. "Combined direct and indirect adaptive control of robot manipulators using multiple models". In: *Advanced Robotics* 20.4 (2006), pp. 483–497.

[25] Mark W Spong, Seth Hutchinson, Mathukumalli Vidyasagar, et al. *Robot modeling and control*. Vol. 3. Wiley New York, 2006.

[26] Mohan Santhakumar and Jinwhan Kim. "Modelling, simulation and model reference adaptive control of autonomous underwater vehicle-manipulator systems". In: *2011 11th International Conference on Control, Automation and Systems*. IEEE. 2011, pp. 643–648.

[27] Christopher Korpela, Todd Danko, and Paul Oh. "MM-UAV: Mobile manipulating unmanned aerial vehicle". In: *Journal of Intelligent and Robotic Systems* 65 (Nov. 2012), pp. 93–101. DOI: 10.1007/s10846-011-9591-3.

[28] Vincenzo Lippiello and Fabio Ruggiero. "Cartesian impedance control of a UAV with a robotic arm". In: *IFAC Proceedings Volumes* 45.22 (2012), pp. 704–709.

[29] Santhakumar Mohan and Jinwhan Kim. "Indirect adaptive control of an autonomous underwater vehicle-manipulator system for underwater manipulation tasks". In: *Ocean Engineering* 54 (2012), pp. 233–243.

[30] Christopher Korpela et al. "Dynamic stability of a mobile manipulating unmanned aerial vehicle". In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 4922–4927. DOI: 10.1109/ICRA.2013.6631280.

[31] Matko Orsag, Christopher Korpela, and Paul Oh. "Modeling and Control of MM-UAV: Mobile Manipulating Unmanned Aerial Vehicle". en. In: *Journal of Intelligent & Robotic Systems* 69.1-4 (Jan. 2013), pp. 227–240. ISSN: 0921-0296, 1573-0409. DOI: 10.1007/s10846-012-9723-4. URL: http://link.springer.com/10.1007/s10846-012-9723-4 (visited on 10/12/2021).

[32] Matko Orsag et al. "Lyapunov based model reference adaptive control for aerial manipulation". In: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2013, pp. 966–973. DOI: 10.1109/ICUAS.2013.6564783.

[33] Gianluca Antonelli and Elisabetta Cataldi. "Adaptive control of arm-equipped quadrotors. theory and simulations". In: *22nd Mediterranean Conference on Control and Automation*. IEEE. 2014, pp. 1446–1451.

[34] Fabrizio Caccavale et al. "Adaptive control for UAVs equipped with a robotic arm". In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 11049–11054.

[35] Shuzhi Sam Ge, Yanan Li, and Chen Wang. "Impedance adaptation for optimal robot–environment interaction". In: *International Journal of Control* 87.2 (2014), pp. 249–263.

[36] John-Paul Ore et al. "Autonomous aerial water sampling". In: *Journal of Field Robotics* 32.8 (2015), pp. 1095–1113.

[37] Jingjing Hao. "Adaptive Multiple-Model Switching Control of Robotic Manipulators". PhD thesis. University of Virginia, 2016.

[38] Anil Kumar Sharma and Subir Kumar Saha. "Simplified Drag Modeling for the Dynamics of an Underwater Manipulator". In: *IEEE Journal of Oceanic Engineering* (2019).

[39] J. Denavit and R. S. Hartenberg. “A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices”. In: *Journal of Applied Mechanics* 22.2 (June 2021), pp. 215–221. ISSN: 0021-8936. DOI: 10.1115/1.4011045. eprint: https://asmedigitalcollection.asme.org/appliedmechanics/article-pdf/22/2/215/6748803/215_1.pdf. URL: https://doi.org/10.1115/1.4011045.