End-effector Motion and Force Control for Mobile Redundant Manipulators

by

Jiliang Wang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 7, 2021

Keywords: Underwater Vehicle Manipulator System, Trajectory Tracking, Joint Space
Control, Task Space Control, Neural Network

Abstract

Manipulators extend the application of mobile platforms, such as unmanned aerial vehicles, underwater vehicles, and satellites. The dynamic coupling between the manipulators and mobile base brings great challenges to the motion control of this complex model. The existence of external disturbances and systematic uncertainties requires high robust control strategies. This dissertation focuses on the trajectory tracking control and force control in joint space and in task space. Theoretic analysis and simulation work are given to show the effectiveness of the proposed controllers.

Three types of control strategy are proposed to follow desired joint trajectories:

1. adaptive backstepping with fuzzy logic
2. neural-adaptive control
3. adaptive dual integral sliding mode control

These controllers are explored by computer simulations.

A controller designed in task space is proposed to follow desired end-effector trajectory. The mapping relationship between joint space and task space is modified to guarantee system stabilities and a neural network is used to approximate system uncertainties. Simulation results show the stability when applied to trajectory and force control tasks.

Since neural networks are used in several of the proposed controllers, a simplified robot arm is built to verify the effectiveness of neural network. Two types of controllers, torque control and position control, are tested in this platform and reasons are given to explain the performance of the two controllers. In the experiments, the neural network compensator is able to reduce the Integral of Squared Error (ISE) by more than 15x that achieved by the uncompensated ("open-loop") commercial controller, and 8x-20x better than the PID compensated system.

Acknowledgments

Firstly, I would like to thank my research adviser, Prof. John Y. Hung, for giving me this opportunity to pursue my Ph.D. degree under his supervision. He helped me a lot in my research, including giving me suggestions and inspirations, revising papers, and solving my confusions. I also appreciate his help in my daily life. Moreover, I would like to thank my committee members, Prof. George Flowers, Prof. Thaddeus Roppel, and Prof. Christopher Harris, for giving me suggestions and assistance on my research and experiments. I also would like to thank Prof. Dan Marghitu as my University Reader.

Many thanks to my office-mate Markus Kreitzer and his wife Heike Kreitzer. Their friendliness makes my life much easier. I also would like to thank my good friend Ethan Swize. We enjoyed happy time together in many activities.

I would like to thank my grandfather, Keyuan Wang, my parents, Guangzhong Wang, and Yuxia Wang, for their selfless love and support for my career choice. I also would like to thank my uncle, Kai Wang, for his help in my life.

Last, I would like to thank China Scholarship Council for the financial support.

Table of Contents

Abstract ii

Acknowledgments iii

1 Introduction 1

1.1 Mobile end-effector applications 1

1.2 Trajectory planning methods 2

1.3 Trajectory tracking control methods 3

1.3.1 Coupled and decoupled approaches 3

1.3.2 Joint space and task space control 5

1.4 Research problems 6

1.5 Overview of the contributions 7

2 Kinematics and dynamics modeling of a UVMS 8

2.1 Kinematic analysis 8

2.2 Dynamics modeling 11

2.2.1 Dynamic modeling in free space 11

2.2.2 Hydrodynamics 12

2.2.3 Current effects 15

2.2.4 Restoring force 16

2.2.5 Complete dynamics model 16

2.3 Dynamic model characteristics and transformation 17

2.3.1 Characteristics of the dynamic model 17

2.3.2 Dynamic model transformation . . . 17

2.4 Summary . . . 18

3 Joint space trajectory tracking control . . . 19

3.1 Adaptive backstepping control with fuzzy logic . . . 19

3.1.1 Controller design . . . 20

3.1.2 Fuzzy logic scheme . . . 23

3.1.3 Stability analysis of the complete system . . . 24

3.1.4 Summary of backstepping control . . . 25

3.2 Trajectory tracking control using a neural-adaptive network . . . 26

3.2.1 Controller design based on parametric uncertainties . . . 26

3.2.2 RBF neural network approximation . . . 28

3.2.3 Adaptive law and stability analysis . . . 30

3.2.4 Summary of neural network control . . . 32

3.3 Adaptive dual integral sliding mode control . . . 32

3.3.1 Proposed controller . . . 32

3.3.2 Inner loop design . . . 33

3.3.3 Adaptive law and stability analysis . . . 35

3.3.4 Outer loop design . . . 36

3.3.5 Summary of dual integral sliding mode control . . . 38

3.4 Simulation results and discussion . . . 38

3.4.1 Description of the UVMS . . . 38

3.4.2 Description of the task . . . 39

3.4.3 Parameters of the controllers . . . 39

3.4.4 Simulation results and discussion . . . 40

3.5 Summary . . . 41

4 Task space trajectory tracking and force control 47
4.1 Problem specification 47
4.2 Task space trajectory tracking control 49
4.2.1 Task space trajectory tracking controller design 51
4.2.2 Adaptive law design and stability analysis in task space 52
4.2.3 Stability Analysis in Joint Space 55
4.2.4 Simulation results and discussion 55
4.2.5 Summary of task space trajectory tracking control 60
4.3 Task space force control 60
4.3.1 Contact with the environment 61
4.3.2 Force controller design in task space 62
4.3.3 Force control task description 63
4.3.4 Simulation results and discussion for force control 64
4.3.5 Summary of force control 64
4.4 Chapter summary 64

5 Trajectory tracking control experiment using neural network 70
5.1 Servo characteristics 70
5.2 Robot design 71
5.3 Dynamics modeling 73
5.4 Control strategies 75
5.4.1 Torque control 75
5.4.2 Position control 76
5.5 Experiment results and discussion 78
5.5.1 Desired trajectory 79
5.5.2 Robot parameters 79

5.5.3 Experiment based on torque control 79
5.5.4 Experiment based on position control 87
5.6 Summary 93
6 Conclusion and future work 94
6.1 Conclusion 94
6.2 Future work 95
References 96
Appendices 106
A Experiment details 107

List of Figures

1.1 Mobile manipulator application examples 2

2.1 Frame assignment of a UVMS 9

3.1 The general joint space trajectory tracking controller scheme 19

3.2 The proposed adaptive backstepping controller scheme 20

3.3 Trajectory tracking control scheme using neural network 28

3.4 RBF neural network structure 29

3.5 Complete control scheme 33

3.6 Inner loop scheme 33

3.7 Outer loop scheme 36

3.8 Vehicle position in x direction 41

3.9 Vehicle position in y direction 42

3.10 Vehicle position in z direction 42

3.11 Vehicle attitude along x-axis 43

3.12 Vehicle attitude along y-axis 43

3.13 Vehicle attitude along z-axis 44

3.14 Desired joint trajectory 44

3.15 Joint 1 error 45

3.16 Joint 2 error 45

3.17 Joint 3 error 46

4.1 The relationship between joint and task space force 49

4.2 Unstable case 50

4.3 Neural network functions . . . 56
4.4 Vehicle displacement . . . 57
4.5 Vehicle velocity . . . 58
4.6 Joint response . . . 59
4.7 End-effector trajectory . . . 60
4.8 Tracking error . . . 61
4.9 Planar view of the chosen contact model . . . 62
4.10 Initial configuration for force control . . . 63
4.11 Vehicle position and linear velocity for force control . . . 65
4.12 Vehicle attitude and angular velocity for force control . . . 66
4.13 Joint response for force control . . . 67
4.14 End-effector position in x idrection . . . 68
4.15 End-effector position in y idrection . . . 68
4.16 End-effector position in z idrection . . . 69
4.17 End-effector force . . . 69
5.1 The front view of AX-12A . . . 71
5.2 Compliance slope of AX-12A . . . 71
5.3 Assembled robot manipulator . . . 72
5.4 Simplified robot with assigned frames . . . 72
5.5 U2D2 . . . 73
5.6 SMPS2DYNAMIXEL . . . 74
5.7 Torque control scheme . . . 75
5.8 Direct servo motion . . . 76
5.9 Control scheme . . . 77
5.10 Joint 1 response using torque control . . . 81
5.11 Joint 2 response using torque control . . . 81

5.12 Joint 3 response using torque control 82
5.13 Joint 4 response using torque control 82
5.14 Joint 1 error using torque control 83
5.15 Joint 2 error using torque control 83
5.16 Joint 3 error using torque control 84
5.17 Joint 4 error using torque control 84
5.18 Joint 1 torque using torque control 85
5.19 Joint 2 torque using torque control 85
5.20 Joint 3 torque using torque control 86
5.21 Joint 4 torque using torque control 86
5.22 Nonlinearity test 88
5.23 Joint 1 response using position control 89
5.24 Joint 2 response using position control 89
5.25 Joint 3 response using position control 90
5.26 Joint 4 response using position control 90
5.27 Joint 1 error using position control 91
5.28 Joint 2 error using position control 91
5.29 Joint 3 error using position control 92
5.30 Joint 4 error using position control 92
A.1 Read and write process 108

List of Tables

3.1 The mass and geometric parameters of the UVMS 38

3.2 Hydrodynamic parameters 38

5.1 Physical parameters of the robot 79

5.2 The values of the ISE of the joint errors 93

Chapter 1

Introduction

Robots have been widely used in many areas in recent decades. One typical application is that they can work in dangerous environments. In industrial fields, robots are adopted to reduce labor costs for sorting, fabrication, assembly, and painting. In the field of search and rescue, robots are deployed to transimit real-time information. To explore the application of robot arms, one novel strategy is to install one or more arms on a mobile base, such as drones, land vehicles, and underwater vehicles. Compared to industrial robots with fixed base, they are not restricted and move freely in space. Furthermore, with manipulators mounted on a floating base, the combined system can carry out aerial and underwater manipulation tasks, such as inspection of infrastructure, detection of fault, and submarine sampling.

1.1 Mobile end-effector applications

The utilization of manipulators extends the application of mobile platforms. Fig. 1.1 shows four application examples [1][2][3]. Installed on a ground mobile base, the combined robot can be applied in the areas of defense, rescue and security, and counter-terrorism [4]. In space, robot arms are used to finish dangerous tasks, such as repair, capture debris and manoeuvre it to lower orbit, which requires that the robotic spacecraft controls its orbit and attitude to reduce the relative motion between the chaser and the target [5]. In the field of aerial manipulation, mobile robots are used to deliver packages, inspect and repair high-voltage electric lines, and wind rotor blade [6]. In underwater environment, the manipulator can be used to remove the oil spill, grasp and transport objects, and dock to a subsea structure [7][8][9]. In addition, mobile manipulators are widely used in other interaction environment, such as medical and education system.

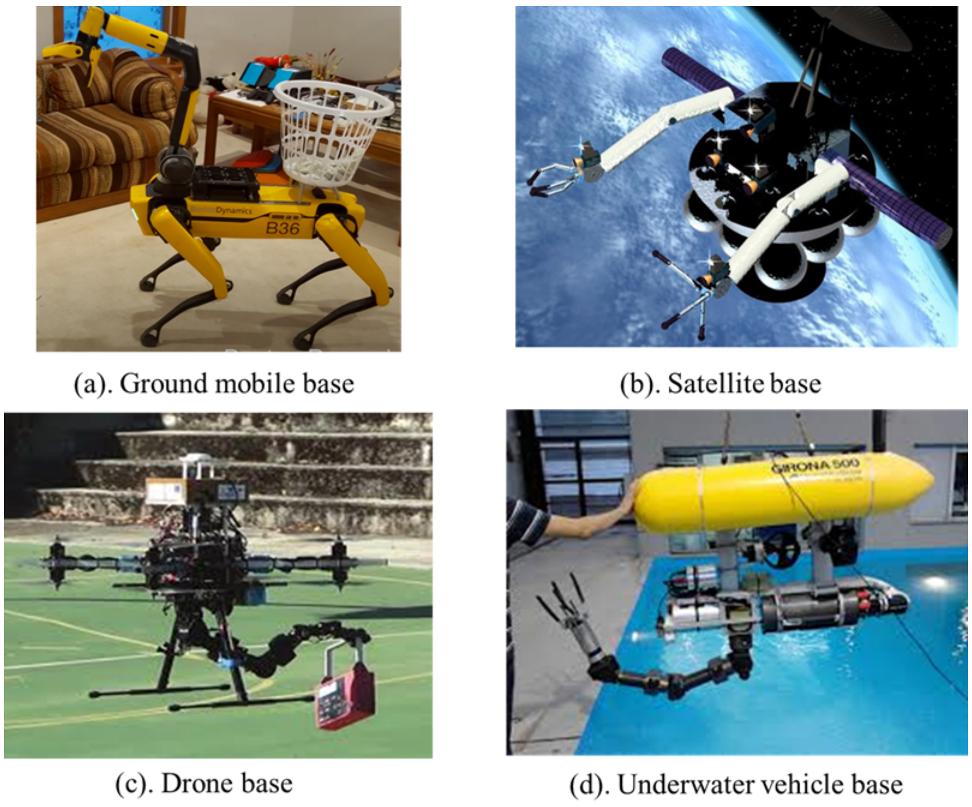


Figure 1.1: Mobile manipulator application examples

1.2 Trajectory planning methods

To move an end-effector to a desired position in task space, inverse kinematics is necessary to calculate the desired position for each joint. However, as the number of degree of freedom (DOF) increases, infinite solutions exist due to system redundancy, which means it is difficult to find one specified trajectory for every joint. Therefore, multiple trajectory planning methods have been proposed based on different considerations.

Task-priority redundancy resolution has been suggested for end-effector path planning. [10]. One primary task is assigned and multiple secondary tasks are selected. Lower level tasks yield to higher level task when they conflict with each other. Antonelli and Chiaverini proposed a fuzzy technique merged with task-priority inverse kinematics approach to distribute the motion between vehicle and manipulator for an underwater vehicle-manipulator system (UVMS)

[11]. A multi-tasking, co-operative control framework was presented in [12] to handle multi-tasking conflicts both in task and joint space for dual-arm robots. However, inequality control objectives are not integrated efficiently, which means the joint trajectories are not smooth. To address the non-smoothness problem, Tang et al. [13] proposed a redundancy resolution with restoring moments optimized on acceleration level to avoid sudden acceleration change. Simetti et al. [14][15] used task-oriented regularization and the singular value-oriented method to eliminate discontinuities.

Joint-space redundancy makes it challenging to plan a path for a mobile end-effector, but its advantages are utilized to avoid obstacles. Slotine et al. [16] implemented a constraint task that allowed the manipulator to avoid an obstacle while the end-effector tracked a trajectory satisfactorily. Lillo et al. [17] proposed an algorithm to generate a repulsive velocity for collision avoidance by computing the minimal distance between the control points and the objects in the environment. A novel approach taking into account multiple control points and multiple obstacles was presented in [18][19] to avoid obstacles. If the obstacle avoidance has the highest priority in the hierarchy, all of the constraints will be respected. Considering multiple different obstacles, Mu et al. [20] used the self-motion of the redundant manipulator to optimize the normalized pseudo-distance by adaptive redundancy resolution.

1.3 Trajectory tracking control methods

A mobile base provides more flexibility while the manipulators remain kinematically redundant, which increases the difficulty of trajectory planning. The dynamic coupling between the mobile vehicle and manipulators makes it a highly nonlinear system. External disturbances, such as airflow, water current, obstacles, and collision, increases the difficulties of stability analysis and control. Considering the above factors, it is necessary to design robust controllers to reduce the effect of disturbances and guarantee system stabilities for specific tasks.

1.3.1 Coupled and decoupled approaches

There are mainly two types of controllers depending on the structural features, coupled and decoupled strategies. The coupled approach takes a model as a complete system while decoupled approach takes it as a combination of two independent devices, a mobile platform and multiple manipulators. The effect of dynamic coupling is seen as an external disturbance in the decoupled approach.

Hovering capability of aerial platforms is an important factor for manipulators attached on unmanned aerial vehicles, also called unmanned aerial manipulator (UAM) [21]. As a coupled system, the dynamic modeling is developed and a full-state feedback linear quadratic regulator controller is designed through obtaining linearized model near steady state in [22]. In order to keep position holding when picking and releasing an object, Kim et al. [23] proposed and tested an adaptive sliding mode control strategy. A stable backstepping-based controller for the multirotor that uses the coupled full dynamic model is proposed in [24], and an admittance controller for the manipulator arm is outlined. In addition, A hierarchical control structure was proposed in [25] to tackle the problem of motion control of the end-effector of an UAM. For mobile platforms working in underwater, such as underwater vehicles, the effect of external environment should be considered when designing control strategies. To mitigate the disturbances of system noise and measurement noise, an indirect adaptive controller fused with an extended Kalman filter (EKF) has shown high robustness even under severe disturbance conditions [26]. Considering uncertainties in system parameters and thruster nonlinearities, a robust controller with integral action was developed to improve the performance of a UVMS in uncertain conditions [27]. For highly nonlinear systems, fuzzy logic systems have been employed to model nonlinear functions and unknown components, which means fuzzy logic theory is also able to be applied to trajectory tracking control of a mobile end-effector. A sliding mode controller using fuzzy logic was designed in [28] to track the trajectory by adjusting the gains.

One advantage of the decoupled approach is researchers are able to design controllers separately for the mobile base and manipulators. Since the two components are controlled independantly and dynamic coupling exists between them, the motion of manipulators can be seen as a perturbation for the platform and vice versa. The literature review reveals a multilayer architecture to control multirotor UAVs equipped with a servo robot arm [29], where the momentum-based observer presented in [30], [31] is employed to compensate neglected aerodynamic effects and the arm dynamics. A neural network was utilized to approximate the dynamics of the UVMS in decentralized form in [32]. However, to guarantee stability, it is necessary design a disturbance compensator if the coupling effect is large [33].

Coupled and decoupled approach can be both adopted if they meet our trajectory tracking requirements. In [34], decoupled and coupled control strategies were tested on a same model using a parallel position/force control structure with sliding mode controllers and incorporating the mathematical model of the system, in which both of the two different control strategies showed reliable performance.

1.3.2 Joint space and task space control

Depending on the space where controllers are designed, control methods can be divided into joint space control and task space control. The control input is derived directly for each joint for joint space control while the controller is designed in task space and the input is mapped into joint space for task space control. Joint space control focuses on the trajectory of each joint and task space control is usually used to follow a trajectory for the end-effector.

The mobile base and manipulators suffer from external disturbances, such as wind, current, and collision. Some disturbance compensators are designed in joint space to increase trajectory tracking accuracy. A sliding mode control fused with an uncertainty and disturbance estimator is proposed in [35] to follow a desired trajectory for the end-effector. A nonlinear disturbance observer using delayed estimates is utilized in [36] to perform motion control for the mobile base and each joint. A joint space coordinate controller using nonlinear disturbance observer is designed in [37] to reduce the effect of unknown disturbances. One crucial aspect of control strategies is to consider the parametric uncertainties since the dynamic modeling is developed in joint space firstly. Neural networks can be used to approximate unknown continuous functions and control complicated multi-input multi-output (MIMO) systems [38][39]. A recurrent wavelet neural network structure was designed for the approximation of uncertain dynamics in [40]. In UVMS areas, the neural network is also used to determine task-oriented dexterity indices, which avoids lengthy calculation and reduces computational time [41].

In order to follow a desired trajectory for the end-effector, inverse kinematics is necessary for joint space control. However, infinite solutions exist due to system redundancy. As a result, task space control, which avoids the calculation of inverse kinematics, is adopted in many areas. The first step for task space control is to transform the dynamic model expressed in joint space to task space [42]. A feedback linearizing control in task coordinates and an extended Kalman filter (EKF) as a state observer are used in [43][44] to maintain its position in the presence of unknown disturbances. A task space passivity-based controller with self-motion of the mobile base was proposed to perform power efficient trajectory in task space because of its kinematically redundant nature [45]. A robust single input fuzzy logic control scheme was proposed in [46] and applied to UVMS task-space trajectory control. To reduce the end-effector interaction forces with the environment, an impedance control scheme was proposed in [47] in end-effector space.

Transformation of dynamic model is necessary and the mapping between two spaces is involved in task space control. However, tracking errors of end-effector can be used in the feedback loop without calculating the inverse kinematics. It provides a convenient way for the force control as well in task space. There is no need to transform the dynamic model and disturbances can be integrated to controllers directly in joint space control. However, inverse kinematics may be a tedious process for joint space control.

1.4 Research problems

Although the control methods described above demonstrate good performance in some cases, the main drawback of these works is implementation complexity that is due to the high computation or complicated adaptation laws. The backstepping control methodology [48][49][50] is one decentralized control scheme with a systematic recursive design procedure that has widely attracted researchers attention in recent years. There is also no need of linearizing the model and the controller can be directly applied to nonlinear system satisfying strict feedback form [51]. A fuzzy logic can be used to eliminate nonlinear terms. Inspired by these observations, an adaptive backstepping control method fused with fuzzy logic is proposed for joint trajectory tracking problems.

One crucial aspect of control strategies is to consider the parametric uncertainties. As neural network can approximate unknown continuous functions, it is used to compensate for system parametric uncertainties. A nonlinear control method based on nominal dynamic model is designed and an adaptive neural network is adopted to reduce the effect of systematic uncertainties. Integral sliding mode control has the advantage of remaining high robustness under uncertain conditions. Therefore, a dual integral sliding mode control strategy is proposed to guarantee stability. For the control methods designed in task space [52][53][54][55], the dynamic model in task space is acquired using pseudo-inverse Jacobian matrix, which increases calculations. The corresponding stability is verified in task space and the control input in task space is mapped into joint space through a system Jacobian matrix. However, considering system redundancy, a stable analysis in task space is not guaranteed to correspond to a stable state in joint space. To address these issues, a control scheme without using pseudo-inverse Jacobian matrix is proposed and the stabilities are verified both in joint space and task space.

In practice, there are unknown controllers inside the servos. Position and torque limit are commonly used as inputs. A joint trajectory tracking controller using a neural network is tested on a four-link robot arm. Besides, a neural network trajectory compensator is designed to improve the performance of unknown controllers in the servo.

1.5 Overview of the contributions

In the following chapters, the research problems mentioned above are discussed in detail.A UVMS model is used to study the performance of the proposed controllers in this work. In Chapter 2, the kinematic and dynamic model of a UVMS with a three-link manipulator are developed. In Chapter 3, three joint space controllers, backstepping control, adaptive neural network control, and dual integral sliding mode control, are proposed and simulations explore their effectiveness. In Chapter 4, a trajectory and force control method designed in task space is proposed and system stability is analyzed. In Chapter 5, a torque-based controller and a position-based neural network compensator are tested on a robot arm. Conclusions and future work are given in Chapter 6.

Chapter 2

Kinematics and dynamics modeling of a UVMS

The controllers with a feedback form in this work are designed based on the kinematics and dynamics model. Therefore, kinematic analysis is given and dynamic model of a UVMS is developed in this chapter.

2.1 Kinematic analysis

The simplified model of a UVMS with assigned frames is shown in Fig. 2.1. It consists of one ellipsoid vehicle and a three-joint manipulator. The manipulator is mounted on a base in the vehicle. The generalized coordinates are expressed as:

$$\boldsymbol{\zeta} = [\boldsymbol{\eta}_1^{\mathrm{T}}\ \boldsymbol{\eta}_2^{\mathrm{T}}\ \boldsymbol{q}^{\mathrm{T}}]^{\mathrm{T}}. \tag{2.1}$$

where vector $\boldsymbol{\eta}_1 = [x\ y\ z]^{\mathrm{T}}$ is the position of the vehicle expressed in the inertial frame. Scalars x, y, and z are called surge, sway, and heave position, respectively. Vector $\boldsymbol{\eta}_2 = [\phi\ \theta\ \psi]^{\mathrm{T}}$ is the set of vehicle Euler angles in the inertial frame (also called roll, pitch, yaw angles, respectively). Vector $\boldsymbol{q} = [q_1\ q_2\ q_3]^{\mathrm{T}}$ is the set of manipulator joint angles.

The inertial frame Σ_I and vehicle coordinate frame Σ_b can be written as $\boldsymbol{O}_I - \boldsymbol{x}_I\boldsymbol{y}_I\boldsymbol{z}_I$, and $\boldsymbol{O}_b - \boldsymbol{x}_b\boldsymbol{y}_b\boldsymbol{z}_b$, respectively. Define $\boldsymbol{v}_1 = [u\ v\ w]^{\mathrm{T}}$ as the linear velocity of the vehicle expressed in vehicle frame Σ_b. The velocity is related to the inertial frame velocity as:

$$\boldsymbol{v}_1 = \boldsymbol{R}_I^B\dot{\boldsymbol{\eta}}_1. \tag{2.2}$$

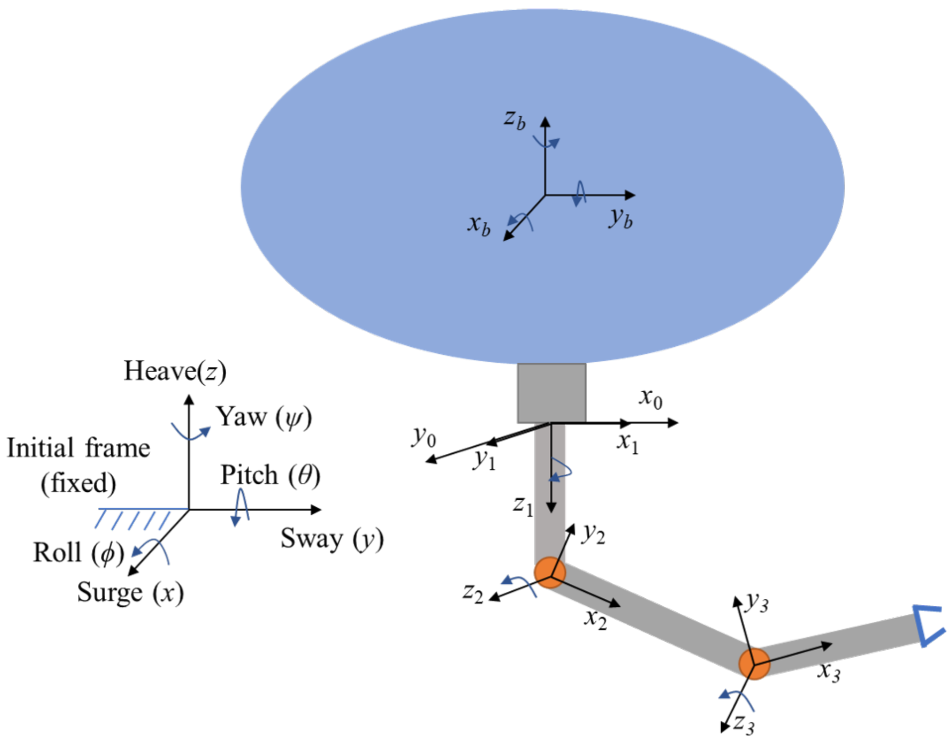


Figure 2.1: Frame assignment of a UVMS

where $\boldsymbol{R}_I^B$ is is the rotation matrix expressing the transformation from Σ_I to Σ_b. Expressed in terms of Euler angles, $\boldsymbol{R}_I^B$ can be written as:

$$\boldsymbol{R}_I^B = \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ -s_\psi c_\phi + c_\psi s_\theta s_\phi & c_\psi c_\phi + s_\psi s_\theta s_\phi & s_\phi c_\theta \\ s_\psi s_\phi + c_\psi s_\theta c_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi & c_\phi c_\theta \end{bmatrix}. \tag{2.3}$$

where c_α and s_α are short notations for cos(α) and sin(α), respectively.

Define $\boldsymbol{v}_2 = [p\ q\ r]^{\mathrm{T}}$ as the angular velocity of the body-fixed frame for the vehicle with respect to the earth-fixed frame expressed in the body-fixed frame. We have the following relationship between $\boldsymbol{v}_2$ and $\dot{\boldsymbol{\eta}}_2$ as:

$$\boldsymbol{v}_2 = \boldsymbol{J}_{ab}\dot{\boldsymbol{\eta}}_2. \tag{2.4}$$

where the matrix $\boldsymbol{J}_{ab}$ can be expressed in terms of Euler angles as:

$$\boldsymbol{J}_{ab} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\psi & c_\theta s_\psi \\ 0 & -s_\psi & c_\theta c_\psi \end{bmatrix}. \tag{2.5}$$

The manipulator base coordinate frame Σ_0 is written as: $\boldsymbol{O}_0 - \boldsymbol{x}_0\boldsymbol{y}_0\boldsymbol{z}_0$. The vector from the origin of Σ_b to the origin of Σ_0 expressed in vehicle body-fixed frame is written as $\boldsymbol{r}_{b0}^b$. The rotation matrix expressing the transformation from Σ_0 to Σ_b is:

$$\boldsymbol{R}_0^B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \tag{2.6}$$

The linear and angular velocities for the center of the i^{th} link expressed in local frame are expressed as:

$$\boldsymbol{v}_{c,i}^i = \boldsymbol{J}_{Lc,i}^i \boldsymbol{\xi}. \tag{2.7}$$

$$\boldsymbol{\omega}_i^i = \boldsymbol{J}_{A,i}^i \boldsymbol{\xi}. \tag{2.8}$$

where $\boldsymbol{\xi} = [\boldsymbol{v}_1^{\mathrm{T}}\ \boldsymbol{v}_2^{\mathrm{T}}\ \dot{\boldsymbol{q}}^{\mathrm{T}}]^{\mathrm{T}}$ represents the velocity of the UVMS expressed in body frames. Matrices $\boldsymbol{J}_{Lc,i}^i$ and $\boldsymbol{J}_{A,i}^i$ are the Jacobians corresponding to the point of force/moment application [56]. Similarly, the linear and angular accelerations for the center of the i^{th} link expressed in local frame are expressed as:

$$\dot{\boldsymbol{v}}_{c,i}^i = \boldsymbol{J}_{Lc,i}^i \dot{\boldsymbol{\xi}} + \dot{\boldsymbol{J}}_{Lc,i}^i \boldsymbol{\xi}. \tag{2.9}$$

$$\dot{\boldsymbol{\omega}}_i^i = \boldsymbol{J}_{A,i}^i \dot{\boldsymbol{\xi}} + \dot{\boldsymbol{J}}_{A,i}^i \boldsymbol{\xi}. \tag{2.10}$$

Based on Eq. (2.2) and (2.4), we have the relationship between body frame and inertial frame velocities:

$$\boldsymbol{\xi} = \boldsymbol{J}_I^b \dot{\boldsymbol{\zeta}}. \tag{2.11}$$

where

$$\boldsymbol{J}_I^b = \begin{bmatrix} \boldsymbol{R}_I^B & & \\ & \boldsymbol{J}_{ab} & \\ & & \boldsymbol{I}_3 \end{bmatrix}. \tag{2.12}$$

2.2 Dynamics modeling

There are mainly two methods used to develop the dynamic model for a UVMS, Lagrange's method and Newton-Euler method. Considering the structure of the UVMS used in this work, the recursive Newton-Euler method is adopted to build its dynamic model. The process includes three steps. Firstly, a dynamic model in free space is developed, which means the hydrodynamics and restoring force are ignored. Secondly, the effect of hydrodynamics is added to the model. Finally, a complete dynamic model is built by adding the restoring force and moment.

2.2.1 Dynamic modeling in free space

The inertial force and moment for the center of the i^{th} component (including the vehicle and manipulator) expressed in local frame can be written as:

$$\boldsymbol{F}_i^i = m_i \dot{\boldsymbol{v}}_{c,i}^i = m_i \boldsymbol{J}_{Lc,i}^i \dot{\boldsymbol{\xi}} + m_i \dot{\boldsymbol{J}}_{Lc,i}^i \boldsymbol{\xi}. \tag{2.13}$$

$$\boldsymbol{N}_i^i = \boldsymbol{I}_{c,i}^i \dot{\boldsymbol{\omega}}_i^i + \boldsymbol{\omega}_i^i \times \boldsymbol{I}_{c,i}^i \boldsymbol{\omega}_i^i = \boldsymbol{I}_{c,i}^i (\boldsymbol{J}_{A,i}^i \dot{\boldsymbol{\xi}} + \dot{\boldsymbol{J}}_{A,i}^i \boldsymbol{\xi}) + (\boldsymbol{J}_{A,i}^i \boldsymbol{\xi}) \times \boldsymbol{I}_{c,i}^i (\boldsymbol{J}_{A,i}^i \boldsymbol{\xi}). \tag{2.14}$$

where m_i is the mass of the i^{th} component and $\boldsymbol{I}_{c,i}^i$ is the moment of inertia at the center. The total force and moment exerted on the i^{th} component expressed in local frame are expressed as:

$$\boldsymbol{f}_i^i = \boldsymbol{R}_{i+1}^i \boldsymbol{f}_{i+1}^{i+1} + \boldsymbol{F}_i^i. \tag{2.15}$$

$$\boldsymbol{n}_i^i = \boldsymbol{N}_i^i + \boldsymbol{R}_{i+1}^i \boldsymbol{n}_{i+1}^{i+1} + \boldsymbol{p}_{c_i}^i \times \boldsymbol{F}_i^i + \boldsymbol{p}_{i+1}^i \times \boldsymbol{R}_{i+1}^i \boldsymbol{f}_{i+1}^{i+1}. \tag{2.16}$$

where $\boldsymbol{R}_{i+1}^{i}$ is the transformation matrix from the $(i+1)^{th}$ frame to the i^{th} frame. Vector $\boldsymbol{p}_{c_i}^{i}$ is the set of mass center position for the i^{th} component expressed in body frame. Vector $\boldsymbol{p}_{i+1}^{i}$ is the origin of the $(i+1)^{th}$ coordinate frame expressed in the i^{th} coordinate frame. Then, the moment of the i^{th} component is written as:

$$\tau_i = \boldsymbol{n}_i^{i\mathrm{T}} \hat{\boldsymbol{z}}. \tag{2.17}$$

where $\hat{\boldsymbol{z}} = [0\ 0\ 1]^{\mathrm{T}}$. Incorporating all the forces and moments, the dynamic equation of a UVMS in a microgravity environment can be expressed as:

$$\boldsymbol{M}_f \dot{\boldsymbol{\xi}} + \boldsymbol{C}_f \boldsymbol{\xi} = \boldsymbol{\tau}_f. \tag{2.18}$$

where $\boldsymbol{M}_f$ and $\boldsymbol{C}_f$ are the mass matrix and Coriolis and centripetal matrix, respectively. Vector $\boldsymbol{\tau}_f$ represents the resultant input in free space.

2.2.2 Hydrodynamics

The theory of fluidodynamics is rather complex and it is difficult to develop a reliable model for most of the hydrodynamic effects. A rigorous analysis for incompressible fluids would need to resort to the Navier-Stokes equations. However, the hydrodynamic effects are considered in a context of automatic control, which means the calculation of hydrodynamic effects is simplified. In underwater environment, the hydrodynamics mainly includes added mass and inertia, hydrodynamic damping, current effects and buoyancy [57]. In this part, the effect of hydrodynamics is discussed in detail.

(1). Added mass and inertia

The movement of a rigid body is able to cause the acceleration of the surrounding fluid, which results in a reaction force that is equal in magnitude and opposite to the direction of motion. That reaction force is called added mass and inertia term. Different properties hold with respect to the (6 × 6) inertia matrix of a rigid body due to the fact that the added mass is a function of the body's surface geometry. Considering the geometry shape of the UVMS, the added mass and inertia for an ellipsoid and cylinder are introduced in this part.

For a 6-DOF rigid body, the added mass and inertia term can be expressed as [58]:

$$\boldsymbol{M}_{rA}\dot{\boldsymbol{v}} + \boldsymbol{C}_{rA}\boldsymbol{v} = \boldsymbol{\tau}_{rA}. \tag{2.19}$$

in which

$$\boldsymbol{M}_{rA} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}. \tag{2.20}$$

$$\boldsymbol{C}_{rA} = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & -S(\boldsymbol{A}_{11}\boldsymbol{v}_L + \boldsymbol{A}_{12}\boldsymbol{v}_A) \\ -S(\boldsymbol{A}_{11}\boldsymbol{v}_L + \boldsymbol{A}_{12}\boldsymbol{v}_A) & -S(\boldsymbol{A}_{21}\boldsymbol{v}_L + \boldsymbol{A}_{22}\boldsymbol{v}_A) \end{bmatrix}. \tag{2.21}$$

$$\boldsymbol{v} = [\boldsymbol{v}_L^{\mathrm{T}}\ \boldsymbol{v}_A^{\mathrm{T}}]^{\mathrm{T}}. \tag{2.22}$$

where $\boldsymbol{v}_L$ and $\boldsymbol{v}_A$ are the linear and angular velocity of the rigid body. $\dot{\boldsymbol{v}}$ is the acceleration of the rigid body. S is the skew-symmetric cross product matrix.

If the body is completely submerged in the water, the velocity is low and it has three planes of symmetry as common for underwater vehicles, the following structure of matrices $\boldsymbol{M}_{rA}$ and $\boldsymbol{C}_{rA}$ can therefore be considered:

$$\boldsymbol{M}_{rA} = -\mathrm{diag}\{X_{\dot{u}},\ Y_{\dot{v}},\ Z_{\dot{w}},\ K_{\dot{p}},\ M_{\dot{q}},\ N_{\dot{r}}\} \tag{2.23}$$

$$\boldsymbol{C}_{rA} = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix}. \tag{2.24}$$

The added mass coefficients, $[X_{\dot{u}},\ Y_{\dot{v}},\ Z_{\dot{w}},\ K_{\dot{p}},\ M_{\dot{q}},\ N_{\dot{r}}]$, can be theoretically derived by exploiting the geometry of the rigid body and applying the strip theory. For a cylindrical rigid body of mass $\bar{m}$, length $\bar{L}$, and circular section of radius $\bar{r}$, the following added mass coefficients can be derived:

$$\begin{aligned}
X_{\dot{u}} &= -\ 0.1\bar{m} \\
Y_{\dot{v}} &= -\ \pi\rho\bar{r}^2\bar{L} \\
Z_{\dot{w}} &= -\ \pi\rho\bar{r}^2\bar{L} \\
K_{\dot{p}} &= 0 \\
M_{\dot{q}} &= -\ \frac{1}{12}\pi\rho\bar{r}^2\bar{L}^3 \\
N_{\dot{r}} &= -\ \frac{1}{12}\pi\rho\bar{r}^2\bar{L}^3
\end{aligned} \quad . \tag{2.25}$$

Now, for the i^{th} component, the added mass effects can be expressed as external forces/moments in the task space as:

$$\boldsymbol{F}_{rA,i} = \boldsymbol{M}_{rA,i}\begin{bmatrix}\boldsymbol{J}^i_{Lc,i}\dot{\boldsymbol{\xi}} + \dot{\boldsymbol{J}}^i_{Lc,i}\boldsymbol{\xi} \\ \boldsymbol{J}^i_{A,i}\dot{\boldsymbol{\xi}} + \dot{\boldsymbol{J}}^i_{A,i}\boldsymbol{\xi}\end{bmatrix} + \boldsymbol{C}_{rA,i}\begin{bmatrix}\boldsymbol{J}^i_{Lc,i}\boldsymbol{\xi} \\ \boldsymbol{J}^i_{A,i}\boldsymbol{\xi}\end{bmatrix} \tag{2.26}$$

By multiplying the Jacobian matrix of the application pint, the added mass effects can be written in joint space as:

$$\boldsymbol{\tau}_{rA,i} = \begin{bmatrix}\boldsymbol{J}^b_{Lc,i} \\ \boldsymbol{J}^b_{A,i}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{R}^b_i & 0_{3\times3} \\ 0_{3\times3} & \boldsymbol{R}^b_i\end{bmatrix}\boldsymbol{F}_{rA,i} \tag{2.27}$$

As the Jacobian matrices are expressed in the base-fixed coordinate, it is necessary to change the expression of external forces/moments from the local coordinate of i^{th} component to that of the base before multiplying by rotation matrices.

(2). Hydrodynamic damping

The viscosity of the fluid also causes the presence of dissipative drag and lift forces on the body. The former are parallel to the relative velocity between the body and the fluid, while the latter are normal to it. Both drag and lift forces are supposed to act on the center of mass of the body. A common simplification is to consider only linear and quadratic damping terms and group these terms in a matrix $\boldsymbol{D}(\boldsymbol{v})\boldsymbol{v}$, where

$$\begin{aligned}\boldsymbol{D}(\boldsymbol{v}) =& \boldsymbol{D}_1(\boldsymbol{v}) + \boldsymbol{D}_2(\boldsymbol{v}) = \text{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\} \\ & + \text{diag}\{X_{u|u|}|u|, Y_{v|v|}|v|, Z_{w|w|}|w|, K_{p|p|}|p|, M_{q|q|}|q|, N_{r|r|}|r|\}.\end{aligned} \tag{2.28}$$

In Eq. (2.28), the coefficients of this matrix are considered to be constant, and the detailed analysis can be found in [59].

For cylindrical manipulators, if the x-axis of the local coordinate coincides with the cylinder axis, the damping effects along y and z axis are written as [60]:

$$f_{Dy} = -0.5\rho C_{D,i} r_i \int_0^{l_i} (v_y(x))^2 dx. \tag{2.29}$$

$$f_{Dz} = -0.5\rho C_{D,i} r_i \int_0^{l_i} (v_z(x))^2 dx. \tag{2.30}$$

in which ρ is the water density, $C_{D,i}$ is the drag coefficient, r_i is the radius of link, v_y and v_z are the link velocities along y and z axis. To simplify the calculation process, the velocity of mass center is used instead of integration. Therefore, the total damping force of the i^{th} link in body frame is written as

$$\boldsymbol{f}^i_{D,i} = \begin{bmatrix} 0 & f_{Dy,i} & f_{Dz,i} \end{bmatrix}^{\mathrm{T}}. \tag{2.31}$$

Expressed in joint space, the damping effect of the i^{th} link is written as:

$$\boldsymbol{D}_i = {\boldsymbol{J}^b_{Lc,i}}^{\mathrm{T}} \boldsymbol{R}^b_i \boldsymbol{f}^I_{D,i} \tag{2.32}$$

2.2.3 Current effects

Control of marine vehicles cannot neglect the effects of specific disturbances such as waves, wind and ocean current. Currents can be very different due to local climatic and/or geographic characteristics. In this work, we assume that ocean current, expressed in the inertial frame, $\boldsymbol{v}_c^I$ is constant and irrotational, such that

$$\boldsymbol{v}_c^I = \begin{bmatrix} v_{c,x}, \ v_{c,y}, \ v_{c,z}, \ 0, \ 0, \ 0 \end{bmatrix}^{\mathrm{T}}. \tag{2.33}$$

$$\dot{\boldsymbol{v}}_c^I = \mathbf{0}. \tag{2.34}$$

Current effects can be added to the dynamic of the i^{th} component moving in a fluid simply considering the relative velocity in body-fixed frame

$$\boldsymbol{v}_{r,i} = \boldsymbol{v}_i - \boldsymbol{R}_I^i \boldsymbol{v}_c^I. \tag{2.35}$$

in the derivation of the Coriolis and centripetal terms and the damping terms.

2.2.4 Restoring force

The gravity and buoyancy of any component of a UVMS in inertial frame are obtained as:

$$\boldsymbol{G}_i + \boldsymbol{B}_i = (m_i - \rho \triangle_i)^I \boldsymbol{g}. \tag{2.36}$$

Here, $\boldsymbol{G}_i$ and $\boldsymbol{B}_i$ are the gravity and buoyancy of any component of a UVMS expressed in inertial frame. $\triangle_i$ is the volume. ${}^I\boldsymbol{g}$ is the gravitational acceleration. Denoted by $\boldsymbol{\beta}$, the cumulative effect of the gravity and buoyancy in joint space is obtained as:

$$\boldsymbol{\beta}_i = {\boldsymbol{J}_{Lc,i}^b}^{\mathrm{T}} \boldsymbol{R}_I^b (\boldsymbol{G}_i + \boldsymbol{B}_i). \tag{2.37}$$

2.2.5 Complete dynamics model

By writing all the terms described above together, the complete dynamics equation of the UVMS is:

$$\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\xi})\dot{\boldsymbol{\xi}} + \boldsymbol{C}(\boldsymbol{\zeta}, \boldsymbol{\xi})\boldsymbol{\xi} + \boldsymbol{D}(\boldsymbol{\zeta}, \boldsymbol{\xi}) + \boldsymbol{G}(\boldsymbol{\zeta}) = \boldsymbol{\tau} + \boldsymbol{f}_e. \tag{2.38}$$

where $\boldsymbol{M}(\boldsymbol{\zeta},\boldsymbol{\xi})$ is the inertia (mass) matrix and $\boldsymbol{C}(\boldsymbol{\zeta},\boldsymbol{\xi})$ is the Coriolis and centripetal matrix. Vectors $\boldsymbol{D}(\boldsymbol{\zeta},\boldsymbol{\xi})$ and $\boldsymbol{G}(\boldsymbol{\zeta})$ describe the drag and restroing forces, respectively. The process input is written as $\boldsymbol{\tau}$ and $\boldsymbol{f}_e$ is the external force/moment mapped into joint space.

2.3 Dynamic model characteristics and transformation

2.3.1 Characteristics of the dynamic model

Controllers in the following chapters are designed based on the model (2.38). The dynamic model posseses two mathematical properties:

1. Matrix $\boldsymbol{M}(\boldsymbol{\zeta},\boldsymbol{\xi})$ is positive definite, which means $\sigma_0 > 0$ ($\sigma_0 \in R$) exists to satisfy $0 < \boldsymbol{M}(\boldsymbol{\zeta},\boldsymbol{\xi}) < \sigma_0 \boldsymbol{I}$.
2. The relationship between $\boldsymbol{M}(\boldsymbol{\zeta},\boldsymbol{\xi})$ and $\boldsymbol{C}(\boldsymbol{\zeta},\boldsymbol{\xi})$ is

$$\boldsymbol{\xi}^{\mathrm{T}}(\dot{\boldsymbol{M}}(\boldsymbol{\zeta},\boldsymbol{\xi}) - 2\boldsymbol{C}(\boldsymbol{\zeta},\boldsymbol{\xi})\boldsymbol{\xi} = 0. \tag{2.39}$$

To make the design process in a compact manner, in the following chapters, $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{D}$ and $\boldsymbol{G}$ are used to represent $\boldsymbol{M}(\boldsymbol{\zeta},\boldsymbol{\xi})$, $\boldsymbol{C}(\boldsymbol{\zeta},\boldsymbol{\xi})$, $\boldsymbol{D}(\boldsymbol{\zeta},\boldsymbol{\xi})$ and $\boldsymbol{G}(\boldsymbol{\zeta})$, respectively.

2.3.2 Dynamic model transformation

From Eq. (2.2) and (2.4), we have

$$\begin{aligned} \boldsymbol{\xi} &= \boldsymbol{J}_I^b \dot{\boldsymbol{\zeta}} \\ \dot{\boldsymbol{\xi}} &= \boldsymbol{J}_I^b \ddot{\boldsymbol{\zeta}} + \dot{\boldsymbol{J}}_I^b \dot{\boldsymbol{\zeta}}. \end{aligned} \tag{2.40}$$

where

$$\boldsymbol{J}_I^b = \begin{bmatrix} \boldsymbol{R}_I^B & & \\ & \boldsymbol{J}_{ab} & \\ & & \boldsymbol{I}_3 \end{bmatrix}. \tag{2.41}$$

Substituting Eq. (2.40) into Eq. (2.38), we have

$$\boldsymbol{M}_I\ddot{\boldsymbol{\zeta}} + \boldsymbol{C}_I\dot{\boldsymbol{\zeta}} + \boldsymbol{D} + \boldsymbol{G} = \boldsymbol{\tau} + \boldsymbol{f}_e. \tag{2.42}$$

where

$$\begin{aligned} \boldsymbol{M}_I &= \boldsymbol{M}\boldsymbol{J}_I^b \\ \boldsymbol{C}_I &= \boldsymbol{M}\dot{\boldsymbol{J}}_I^b + \boldsymbol{C}\dot{\boldsymbol{J}}_I^b \end{aligned} \tag{2.43}$$

Since Eq. (2.42) is a transformation of (2.38), it does not satisfy the characteristics described above. In the following chapter, both dynamic models will be used.

2.4 Summary

In this chapter, the kinematics of a UVMS with a three-joint manipulator is derived and Jacobian matrix is used to describe the velocities and accelerations. To develop the dynamic model, the recursive Newton-Euler method is adopted and hydrodynamics, such as added mass and inertia, hydrodynamic damping, currents, and restoring forces, is considered into the dynamic model. The corresponding characteristics of the dynamic model are introduced and a transformation is derived.

Chapter 3

Joint space trajectory tracking control

To complete a trajectory tracking task for an end-effector, the controllers are commonly designed in joint space or task space. The general scheme for joint space controller is shown in Fig. 3.1. It can be seen that inverse kinematics is necessary and the controller is designed to follow desired joint trajectory. Considering systematic uncertainties, three joint space trajectory tracking controllers are proposed in this chapter. The control theories are demonstrated in detail and simulation results show their effectiveness when applied to follow desired joint trajectories.

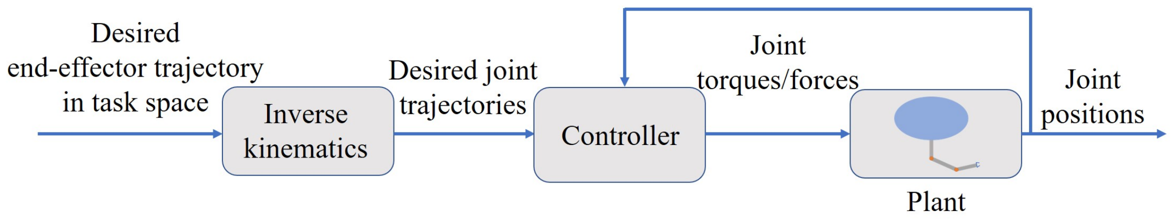

Desired end-effector trajectory in task space
Inverse kinematics
Desired joint trajectories
Controller
Joint torques/forces
Plant
Joint positions

Figure 3.1: The general joint space trajectory tracking controller scheme

3.1 Adaptive backstepping control with fuzzy logic

As a decentralized control scheme, the backstepping control methodology can be applied to a nonlinear system without linearizing the model. A fuzzy logic estimator is able to approximate unknown system terms. Therefore, we can design an adaptive backstepping controller with fuzzy logic for a UVMS by using their advantages. The detailed design process is demonstrated as below [61].

1. System states should be defined and trajectory tracking methods are kinematically designed.
2. Dynamic controller is designed in this step to follow desired trajectory.
3. A fuzzy logic estimator is designed to eliminate system uncertainties and disturbances.
4. Stability is verified.

The proposed control scheme is shown in Fig. 3.2.

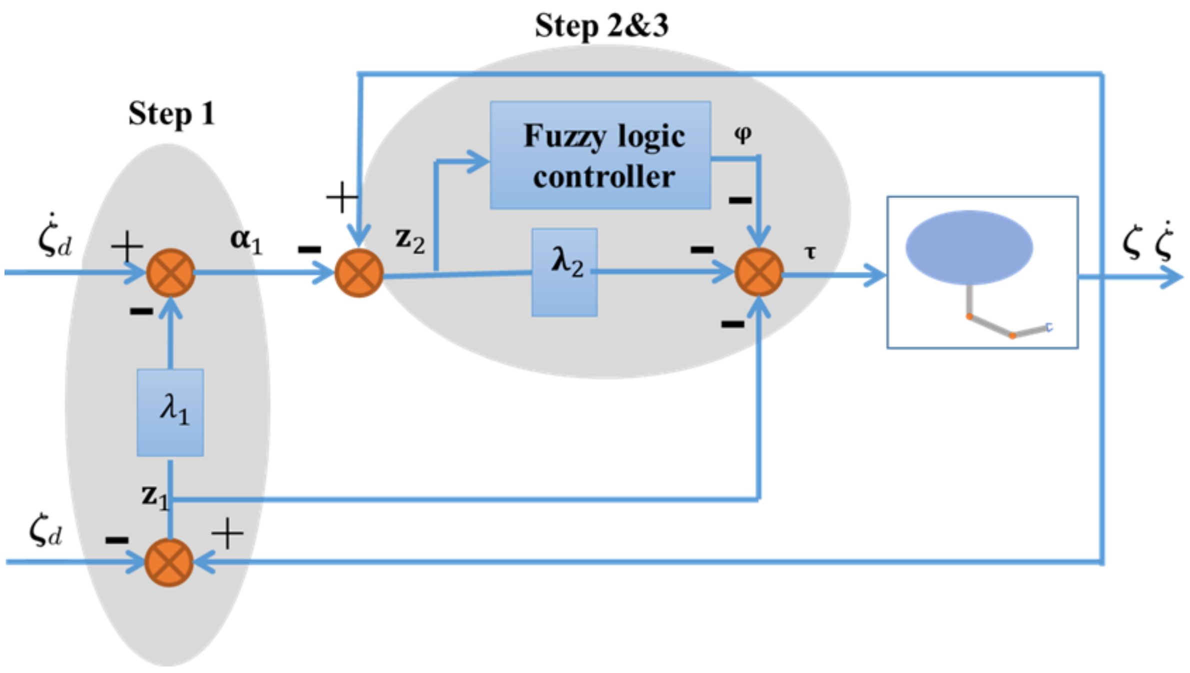


Figure 3.2: The proposed adaptive backstepping controller scheme

3.1.1 Controller design

To apply backstepping control strategy, two states are defined as:

$$\begin{aligned} x_1 &= \zeta \\ x_2 &= \dot{\zeta}. \end{aligned} \tag{3.1}$$

Here, $\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}$ are the generalized coordinates defined in the previous chapter. According to Eq. (2.42), we have the state equations:

$$
\begin{aligned}
\dot{\boldsymbol{x}}_1 &= \boldsymbol{x}_2 \\
\dot{\boldsymbol{x}}_2 &= \boldsymbol{M}_I^{-1}[\boldsymbol{\tau} - \boldsymbol{C}_I \boldsymbol{x}_2 - \boldsymbol{D} - \boldsymbol{G}].
\end{aligned}
\tag{3.2}
$$

Step 1:

The main objective of this controller is to follow desired joint trajectories. Given $\boldsymbol{p}_d$ as the desired position of a UVMS. Define error

$$
\begin{aligned}
\boldsymbol{z}_1 &= \boldsymbol{x}_1 - \boldsymbol{p}_d \\
\boldsymbol{z}_2 &= \boldsymbol{x}_2 - \boldsymbol{\alpha}_1
\end{aligned}
\tag{3.3}
$$

where $\boldsymbol{\alpha}_1$ is the estimated value of $\boldsymbol{x}_2$. By selecting $\boldsymbol{\alpha}_1$ appropriately, $\boldsymbol{z}_2$ approaches to zero. Differentiating $\boldsymbol{z}_1$, we have:

$$
\dot{\boldsymbol{z}}_1 = \dot{\boldsymbol{x}}_1 - \dot{\boldsymbol{p}}_d = \boldsymbol{z}_2 + \boldsymbol{\alpha}_1 - \dot{\boldsymbol{p}}_d \tag{3.4}
$$

Select $\boldsymbol{\alpha}_1$ as:

$$
\boldsymbol{\alpha}_1 = -\lambda_1 \boldsymbol{z}_1 + \dot{\boldsymbol{p}}_d. \tag{3.5}
$$

in which λ_1 is a constant.

For step 1, the Lyapunov function is defined as:

$$
V_1 = \frac{1}{2} \boldsymbol{z}_1^{\mathrm{T}} \boldsymbol{z}_1. \tag{3.6}
$$

Then,

$$
\dot{V}_1 = \boldsymbol{z}_1^{\mathrm{T}} \dot{\boldsymbol{z}}_1 = -\lambda_1 \boldsymbol{z}_1^{\mathrm{T}} \boldsymbol{z}_1 + \boldsymbol{z}_1^{\mathrm{T}} \boldsymbol{z}_2. \tag{3.7}
$$

If $\boldsymbol{z}_2 = 0$, then $\dot{V}_1 \leq 0$. which means the system is stable.

Step 2:

The state variable of $\boldsymbol{z}_2$ can be written as:

$$\dot{\boldsymbol{z}}_2 = \dot{\boldsymbol{x}}_2 - \dot{\boldsymbol{\alpha}}_1 = \boldsymbol{M}_I^{-1}[\boldsymbol{\tau} - \boldsymbol{C}_I\boldsymbol{x}_2 - \boldsymbol{D} - \boldsymbol{G}] - \dot{\boldsymbol{\alpha}}_1. \tag{3.8}$$

Define the Lyapunov function as:

$$V_2 = V_1 + \frac{1}{2}(\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{J}_I^b\boldsymbol{z}_2). \tag{3.9}$$

Then,

$$\dot{V}_2 = -\lambda_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 + \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_2 + (\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}\boldsymbol{M}(\dot{\overline{\boldsymbol{J}_I^b\boldsymbol{z}_2}}) + \frac{1}{2}(\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}\dot{\boldsymbol{M}}(\boldsymbol{J}_I^b\boldsymbol{z}_2). \tag{3.10}$$

Substitute Eq. (3.8) into the above equation,

$$\dot{V}_2 = -\lambda_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 + \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_2 + (\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}(\boldsymbol{f} + \boldsymbol{\tau}). \tag{3.11}$$

where

$$\boldsymbol{f} = -\boldsymbol{C}_I\dot{\boldsymbol{\zeta}} - \boldsymbol{D} - \boldsymbol{G} - \boldsymbol{M}\boldsymbol{J}_I^b\dot{\boldsymbol{\alpha}}_1 + \frac{1}{2}\boldsymbol{M}\boldsymbol{J}_I^b\boldsymbol{z}_2 + \boldsymbol{M}\dot{\boldsymbol{J}}_I^b\boldsymbol{z}_2. \tag{3.12}$$

It can be seen that $\boldsymbol{f}$ contains modeling parameters of the UVMS.

Define the control input as:

$$\boldsymbol{\tau} = -\boldsymbol{\lambda}_2\boldsymbol{J}_I^b\boldsymbol{z}_2 - \frac{\boldsymbol{J}_I^b\boldsymbol{z}_2}{||\boldsymbol{J}_I^b\boldsymbol{z}_2||^2}\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_2 - \boldsymbol{\varphi} \tag{3.13}$$

in which $\boldsymbol{\lambda}_2$ is a positive diagonal matrix. Vector $\boldsymbol{\varphi}$ is the output of a fuzzy logic system used to approximate the UVMS modeling parameters. Substitute Eq. (3.13) into Eq. (3.11),

$$\dot{V}_2 = -\lambda_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 - (\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}\boldsymbol{\lambda}_2\boldsymbol{J}_I^b\boldsymbol{z}_2 + (\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{\varphi}). \tag{3.14}$$

If $\boldsymbol{\varphi}$ is selected approximately to satisfy $\boldsymbol{f} - \boldsymbol{\varphi} = 0$, then $\dot{V}_2 \leq 0$, so the trajectory tracking error is stable.

3.1.2 Fuzzy logic scheme

In this part, a fuzzy logic system is used to approximate the modeling parameters in Eq. (3.12). A fuzzy logic system consists of four parts: the knowledge base, the fuzzifier, the fuzzy inference engine and the defuzzifier [62]. The knowledge base is composed of a collection of fuzzy IF-THEN rules in the following form:

R^k: IF h_1 is μ_1^k and ... and h_n is $\mu_{n_f}^k$, then β is B^k $(k = 1, 2, ..., N)$

where N is the number of rules. The input vector is expressed as $\boldsymbol{H} = [h_1, ..., h_{n_f}]^{\mathrm{T}} \in \mathbb{R}^{n_f}$ and B^k is the corresponding output. In this case, $\boldsymbol{H} = \boldsymbol{x}_1$. The membership function of $h_j (j = 1, ..., n_f)$ is written as μ_j^k.

In this fuzzy logic system, the singleton fuzzifizer, product inference engine and center average defuzzification are used to calculate the output, which is written as:

$$\beta = \frac{\sum\limits_{i=1}^{N} \theta_i \prod\limits_{j=1}^{n_f} (\mu_j^i(h_j))}{\sum\limits_{i=1}^{N} \prod\limits_{j=1}^{n_f} \mu_j^i(h_j)} = \boldsymbol{\Gamma}^{\mathrm{T}}(\boldsymbol{H})\boldsymbol{\Theta}. \tag{3.15}$$

where

$$\boldsymbol{\Gamma} = [\xi_1(\boldsymbol{H}), \xi_2(\boldsymbol{H}), ..., \xi_N(\boldsymbol{H})]^{\mathrm{T}}. \tag{3.16}$$

$$\xi_i(\boldsymbol{H}) = \frac{\prod\limits_{j=1}^{n_f} \mu_j^i(h_j)}{\sum\limits_{i=1}^{N} \prod\limits_{j=1}^{n_f} \mu_j^i(h_j)}. \tag{3.17}$$

$$\boldsymbol{\Theta} = [\theta_1 \; \theta_2 \; ... \; \theta_N]^{\mathrm{T}}. \tag{3.18}$$

The m^{th} output can be written as:

$$\varphi_m(\boldsymbol{H}) = \frac{\sum\limits_{i=1}^{N} \theta_{mi} \prod\limits_{j=1}^{n_f} (\mu_j^i(h_j))}{\sum\limits_{i=1}^{N} \prod\limits_{j=1}^{n_f} \mu_j^i(h_j)} = \boldsymbol{\Gamma}_m^{\mathrm{T}}(\boldsymbol{H})\boldsymbol{\Theta}_m \tag{3.19}$$

The control input consists of six input force/moment of the vehicle and three joint torques. Therefore, the total fuzzy logic output is

$$\begin{aligned}\boldsymbol{\varphi} = \boldsymbol{\Lambda\Phi} &= [\varphi_1(\boldsymbol{H})\ \varphi_2(\boldsymbol{H})\ ...\ \varphi_9(\boldsymbol{H})]^{\mathrm{T}} \\ &= \begin{bmatrix} \boldsymbol{\Gamma}_1^{\mathrm{T}}(\boldsymbol{H}) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \boldsymbol{\Gamma}_9^{\mathrm{T}}(\boldsymbol{H}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Theta_1} \\ \vdots \\ \boldsymbol{\Theta}_9 \end{bmatrix}.\end{aligned} \tag{3.20}$$

3.1.3 Stability analysis of the complete system

For a given arbitrary $\epsilon(\epsilon > 0)$, an optimal parameter vector $\boldsymbol{\Phi}^*$ exists to satisfy [63]

$$\|\boldsymbol{f} - \boldsymbol{\varphi}^*\| \leq \epsilon. \tag{3.21}$$

$$\widetilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi}^* - \boldsymbol{\Phi}. \tag{3.22}$$

Select the adaptive control theory as:

$$\dot{\boldsymbol{\Phi}} = \gamma[(\boldsymbol{J}_I^b \boldsymbol{z}_2)^{\mathrm{T}} \boldsymbol{\varphi}^{\mathrm{T}}]^{\mathrm{T}} - 2\kappa\boldsymbol{\Phi}. \tag{3.23}$$

where γ and κ are constants, and $\gamma > 0$, $\kappa > 0$. For the complete system, the Lyapunov function is chosen as:

$$V = \frac{1}{2}\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 + \frac{1}{2}(\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{J}_I^b\boldsymbol{z}_2) + \frac{1}{2\gamma}\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\widetilde{\boldsymbol{\Phi}}. \tag{3.24}$$

$$\begin{aligned}\dot{V} = &- \lambda_1\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 - (\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}\boldsymbol{\lambda}_2(\boldsymbol{J}_I^b\boldsymbol{z}_2) + (\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{\Lambda\Phi}^*) \\ &+ (\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}(\boldsymbol{\Lambda\Phi}^* - \boldsymbol{\Lambda\Phi}) - \frac{1}{\gamma}\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\dot{\boldsymbol{\Phi}}.\end{aligned} \tag{3.25}$$

$$\begin{aligned}\dot{V} \leq & -\frac{2}{2}\lambda_1 \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 - (\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}(\boldsymbol{\lambda}_2 - \frac{\boldsymbol{I}}{2})\boldsymbol{M}^{-1}\boldsymbol{M}\boldsymbol{J}_I^b\boldsymbol{z}_2 - \frac{\kappa}{2\gamma}\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\widetilde{\boldsymbol{\Phi}} \\ & + \frac{2\kappa}{\gamma}\boldsymbol{\Phi}^{*\mathrm{T}}\boldsymbol{\Phi}^{*} + \frac{1}{2}\epsilon^2\end{aligned}. \tag{3.26}$$

in which $\boldsymbol{\lambda}_2$ is selected as $\boldsymbol{\lambda}_2 > \boldsymbol{I}$. Since $\boldsymbol{M} \leq \sigma_0\boldsymbol{I}$, then $-\boldsymbol{M}^{-1} \leq -\frac{1}{\sigma_0}\boldsymbol{I}$.

$$\begin{aligned}\dot{V} \leq & -\frac{2}{2}\lambda_1 \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 - \frac{2}{2\sigma_0}(\boldsymbol{J}_I^b\boldsymbol{z}_2)^{\mathrm{T}}(\boldsymbol{\lambda}_2 - \frac{\boldsymbol{I}}{2})\boldsymbol{M}(\boldsymbol{J}_I^b\boldsymbol{z}_2) - \frac{\kappa}{2\gamma}\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\widetilde{\boldsymbol{\Phi}} \\ & + \frac{2\kappa}{\gamma}\boldsymbol{\Phi}^{*\mathrm{T}}\boldsymbol{\Phi}^{*} + \frac{1}{2}\epsilon^2\end{aligned}. \tag{3.27}$$

Define $c_0 = \min\{2\lambda_1, \frac{2}{\sigma_0}\min\{\boldsymbol{\lambda}_2 - \frac{\boldsymbol{I}}{2}\}, \kappa\}$, where $\min\{\boldsymbol{\lambda}_2 - \frac{\boldsymbol{I}}{2}\}$ represents the minimum nonzero element in the matrix. Then

$$\dot{V} \leq c_0 V + c_{vmax}. \tag{3.28}$$

where

$$c_{vmax} = \frac{2\kappa}{\gamma}\boldsymbol{\Phi}^{*\mathrm{T}}\boldsymbol{\Phi}^{*} + \frac{1}{2}\epsilon^2 \tag{3.29}$$

Solve Eq. (3.28),

$$\begin{aligned}V(t) &\leq V_0(t)e^{-c_0 t} + \frac{c_{vmax}}{c_0}(1 - e^{-c_0 t}) \\ &\leq V(0) + \frac{c_{vmax}}{c_0},\ \forall t \geq 0\end{aligned} \tag{3.30}$$

Therefore, V is bounded and all the signals in closed-loop system are bounded. Moreover, the tracking errors can be made as small as desired by adjusting the control parameters.

3.1.4 Summary of backstepping control

In this section, the adaptive backstepping control using a fuzzy logic scheme is proposed for a UVMS. The Lyapunov function is used to prove its stability in each step. The simulation results will be given in the last section of this chapter.

3.2 Trajectory tracking control using a neural-adaptive network

In section 3.1, the Lyapunov function is bounded, which means the steady-state tracking error problem exists. To address this problem, there are commonly two strategies, adding a feedforward term and an integration term. In this section, the integration method is adopted. A nonlinear controller is designed to eliminate the nonlinear term in the dynamic equation. Furthermore, an adaptive neural network is adopted to compensate for the uncertainty term [64].

3.2.1 Controller design based on parametric uncertainties

Under parametric uncertainties, model (2.38) can be written as:

$$(\hat{\boldsymbol{M}}_I + \delta \boldsymbol{M}_I)\ddot{\boldsymbol{\zeta}} + (\hat{\boldsymbol{C}}_I + \delta \boldsymbol{C}_I)\dot{\boldsymbol{\zeta}} + \hat{\boldsymbol{D}} + \delta \boldsymbol{D} + \hat{\boldsymbol{G}} + \delta \boldsymbol{G} = \boldsymbol{\tau} + \boldsymbol{f}_e. \tag{3.31}$$

where

$$\begin{aligned} \delta \boldsymbol{M}_I &= \boldsymbol{M}_I - \hat{\boldsymbol{M}}_I \\ \delta \boldsymbol{C}_I &= \boldsymbol{C}_I - \hat{\boldsymbol{C}}_I \\ \delta \boldsymbol{D} &= \boldsymbol{D} - \hat{\boldsymbol{D}} \\ \delta \boldsymbol{G} &= \boldsymbol{G} - \hat{\boldsymbol{G}} \end{aligned} \tag{3.32}$$

In Eq. (3.32), $\hat{\boldsymbol{M}}_I$, $\hat{\boldsymbol{C}}_I$, $\hat{\boldsymbol{D}}$, and $\hat{\boldsymbol{G}}$ are the nominal model parameter values of the inertial matrix, Coriolis and centripetal matrix, damping matrix, and restoring force matrix, respectively.

Consider the control law:

$$\boldsymbol{\tau}_1 = \hat{\boldsymbol{M}}_I(\ddot{\boldsymbol{\zeta}}_d + \boldsymbol{K}_D\dot{\boldsymbol{e}} + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_I\int \boldsymbol{e}) + \hat{\boldsymbol{C}}_I\dot{\boldsymbol{\zeta}} + \hat{\boldsymbol{D}} + \hat{\boldsymbol{G}} \tag{3.33}$$

where

$$\begin{aligned} \boldsymbol{e} &= \boldsymbol{\zeta}_d - \boldsymbol{\zeta} \\ \dot{\boldsymbol{e}} &= \dot{\boldsymbol{\zeta}}_d - \dot{\boldsymbol{\zeta}} \end{aligned} \tag{3.34}$$

and $\boldsymbol{\zeta}_d$ and $\dot{\boldsymbol{\zeta}}_d$ are the desired position and velocity vectors of the UVMS. Parameters $\boldsymbol{K}_P$, $\boldsymbol{K}_I$, and $\boldsymbol{K}_D$ are the proportional, integral, and derivative gains, respectively, which are chosen as diagonal matrices.

Substituting Eq. (3.33) into Eq. (3.31) yields:

$$\ddot{e} + \boldsymbol{K}_D\dot{\boldsymbol{e}} + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_I\int \boldsymbol{e} = \boldsymbol{f}_u. \tag{3.35}$$

where $\boldsymbol{f}_u = \hat{\boldsymbol{M}_I}^{-1}(\delta\boldsymbol{M}_I\ddot{\boldsymbol{\zeta}} + \delta\boldsymbol{C}_I\dot{\boldsymbol{\zeta}} + \delta\boldsymbol{D} + \delta\boldsymbol{G} - \boldsymbol{f}_e)$. In the absence of uncertainties, the closed-loop dynamics are decoupled.

Define the error vector:

$$\boldsymbol{x} = \begin{bmatrix} \int \boldsymbol{e} \\ \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix}. \tag{3.36}$$

Then, Eq. (3.35) can be expressed in term of state vector as:

$$\dot{\boldsymbol{x}} = \boldsymbol{E}\boldsymbol{x} + \boldsymbol{F}\boldsymbol{f}_u \tag{3.37}$$

where

$$\boldsymbol{E} = \begin{bmatrix} 0 & \boldsymbol{I} & 0 \\ 0 & 0 & \boldsymbol{I} \\ -\boldsymbol{K}_I & -\boldsymbol{K}_P & -\boldsymbol{K}_D \end{bmatrix}, \ \boldsymbol{F} = \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{I} \end{bmatrix}. \tag{3.38}$$

It can be seen that the existence of uncertainties makes it more difficult for tracking error to converge to zero. Since the uncertainty term is unknown, it is necessary to design a function to approximate $\boldsymbol{f}_u$ and add a compensation term to the controller. Therefore, the modified control law is expressed as:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2 \tag{3.39}$$

where $\boldsymbol{\tau}_2 = \hat{\boldsymbol{M}}_I\hat{\boldsymbol{f}}_u$, and $\hat{\boldsymbol{f}}_u$ is the approximation of $\boldsymbol{f}_u$. The diagram of the proposed controller is shown in Fig. 3.3. Details of the approximation function $\hat{\boldsymbol{f}}_u$ are presented next.

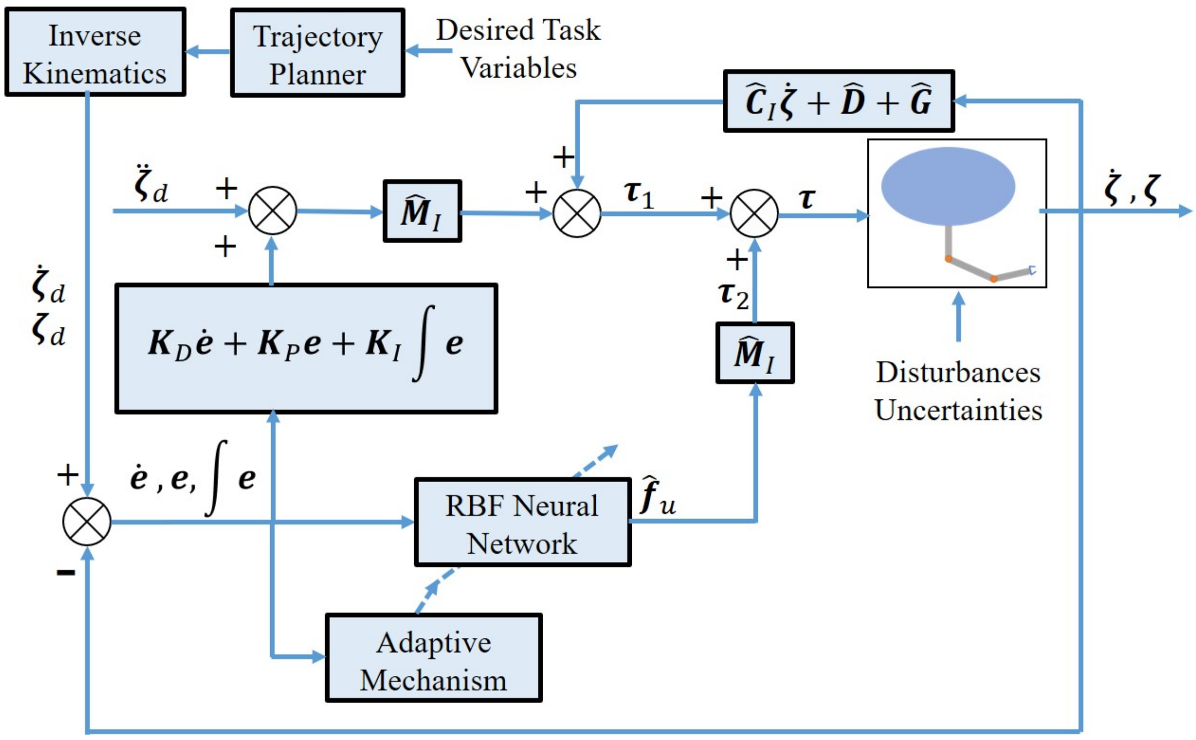


Figure 3.3: Trajectory tracking control scheme using neural network

3.2.2 RBF neural network approximation

The radial basis function (RBF) neural network is a three-layer feed forward network, which uses a Gaussian function as a transfer function. The mapping from hidden layer to output layer is linear and the advantage of using RBF neural network is to avoid local minimum problems.

The structure of a multiple-input and multiple-output (MIMO) RBF neural network is shown in Fig. 3.4. The input vector is denoted as $\boldsymbol{X} = [x_1\ x_2\ ...\ x_N]^{\mathrm{T}}$, and $\boldsymbol{H} = [h_1\ h_2\ ...\ h_m]^{\mathrm{T}}$ is the radial basis vector, where h_j has the Gaussian function form:

$$h_j = \exp(-\frac{||\boldsymbol{X} - \boldsymbol{c}_j||^2}{2b_j^2}),\ j = 1, 2, ..., m. \tag{3.40}$$

Vector $\boldsymbol{c}_j = [c_{j1}\ c_{j2}\ ...\ c_{jN}]^{\mathrm{T}}$ is the set of center points of the j^{th} basis function, and $\boldsymbol{B} = [b_1\ b_2\ ...\ b_m]^{\mathrm{T}}$ is the stretch constant vector. The weight matrix is $\boldsymbol{W}^{\mathrm{T}} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ ...\ \boldsymbol{w}_m]$, where $\boldsymbol{w}_s = [w_{s1}\ w_{s2}\ ...\ w_{sn}]^{\mathrm{T}}$, $s = 1, 2, ..., m$. Therefore, the output is

$$\boldsymbol{Y} = [y_1\ y_2\ ...\ y_n]^{\mathrm{T}} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{H}. \tag{3.41}$$

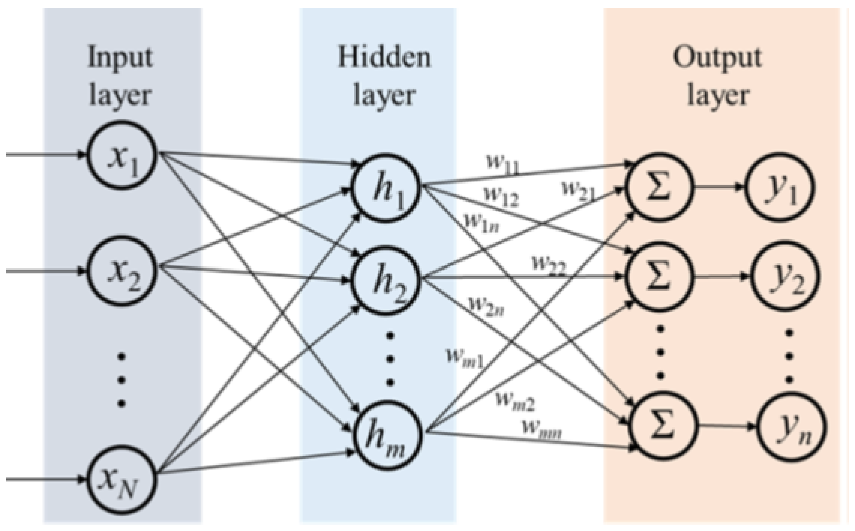


Figure 3.4: RBF neural network structure

If the unknown uncertainty term $\boldsymbol{f}_u$ is continuous and bounded, there exist a RBF neural network $\boldsymbol{f}_u(\boldsymbol{W}^\star)$ to approximate $\boldsymbol{f}_u$.

$$\max ||\boldsymbol{f}_u(\boldsymbol{W}^\star) - \boldsymbol{f}_u|| \leq \epsilon_0. \tag{3.42}$$

$$\boldsymbol{W}^\star = \arg \min_{\boldsymbol{W}} \{\sup ||\boldsymbol{f}_u - \boldsymbol{f}_u(\boldsymbol{W})||\}. \tag{3.43}$$

in which $\boldsymbol{f}_u(\boldsymbol{W}^\star) = \boldsymbol{W}^{\star\mathrm{T}}\boldsymbol{H}$, and ϵ_0 is a very small positive number. Here, $\boldsymbol{W}^\star$ is the best neural network output of the approximation function. Therefore, the tracking error state vector equation (3.37) can be expressed as:

$$\dot{\boldsymbol{x}} = \boldsymbol{E}\boldsymbol{x} + \boldsymbol{F}\{\boldsymbol{f}_u(\boldsymbol{W}^\star) + \boldsymbol{\mu}\} \tag{3.44}$$

where $\boldsymbol{\mu}$ is the RBF neural network error, with the form $\boldsymbol{\mu} = \boldsymbol{f}_u - \boldsymbol{f}_u(\boldsymbol{W}^\star)$. In addition, $\boldsymbol{\mu}$ is bounded, with bound $\boldsymbol{\mu}_0 = \sup \; ||\boldsymbol{f}_u - \boldsymbol{f}_u(\boldsymbol{W}^\star)||$. So Eq. (3.44) can be written as:

$$\dot{\boldsymbol{x}} = \boldsymbol{E}\boldsymbol{x} + \boldsymbol{F}\{\boldsymbol{W}^{\star\mathrm{T}}\boldsymbol{H} + \boldsymbol{\mu}\} \tag{3.45}$$

3.2.3 Adaptive law and stability analysis

In this part, we choose to approximate $\boldsymbol{W}^{\star}$ in Eq. (3.45). Chosse $\hat{\boldsymbol{f}}_u$ as:

$$\hat{\boldsymbol{f}}_u = \hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H}. \tag{3.46}$$

Then substituting (3.46) into (3.39) and (3.31) yields:

$$\dot{\boldsymbol{x}} = \boldsymbol{E}\boldsymbol{x} + \boldsymbol{F}\{-\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H} + \boldsymbol{\mu}\}. \tag{3.47}$$

$$\widetilde{\boldsymbol{W}}^{\mathrm{T}} = \hat{\boldsymbol{W}}^{\mathrm{T}} - \boldsymbol{W}^{\star\mathrm{T}}. \tag{3.48}$$

Define the Lyapunov function:

$$V = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} + \frac{1}{2\gamma}||\widetilde{\boldsymbol{W}}||^2. \tag{3.49}$$

where $\gamma > 0$. Matrix $\boldsymbol{P}$ is symmetric positive definite, and satisfies the Lyapunov equation:

$$\boldsymbol{P}\boldsymbol{E} + \boldsymbol{E}^{\mathrm{T}}\boldsymbol{P} = -\boldsymbol{Q}. \tag{3.50}$$

in which $\boldsymbol{Q} \geq \boldsymbol{0}$.

The time derivative of Eq. (3.49) is:

$$\begin{aligned}
\dot{V} &= \frac{1}{2}[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}] + \frac{1}{\gamma}tr(\dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}}) \\
&= \frac{1}{2}\{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}[\boldsymbol{E}\boldsymbol{x} + \boldsymbol{F}(-\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H} + \boldsymbol{\mu})] \\
&\qquad + [\boldsymbol{x}^{\mathrm{T}}\boldsymbol{E}^{\mathrm{T}} + (-\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H} + \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}]\boldsymbol{P}\boldsymbol{x}\} + \frac{1}{\gamma}tr(\dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}}) \\
&= \frac{1}{2}[\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{P}\boldsymbol{E} + \boldsymbol{E}^{\mathrm{T}}\boldsymbol{P})\boldsymbol{x}] + [-\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H} + \boldsymbol{\mu}]^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} + \frac{1}{\gamma}tr(\dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}}) \\
&= -\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{H}^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} + \frac{1}{\gamma}tr(\dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}}).
\end{aligned} \tag{3.51}$$

Because

$$\boldsymbol{H}^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} = tr(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}\boldsymbol{H}^{\mathrm{T}}\widetilde{\boldsymbol{W}}). \tag{3.52}$$

Then Eq. (3.51) can be written as;

$$\dot{V} = -\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} + \frac{1}{\gamma}tr(-\gamma\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}\boldsymbol{H}^{\mathrm{T}}\widetilde{\boldsymbol{W}} + \dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}}). \tag{3.53}$$

The adaptive law is selected as:

$$\dot{\hat{\boldsymbol{W}}} = \gamma\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{F}. \tag{3.54}$$

Therefore,

$$\dot{V} = -\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}. \tag{3.55}$$

As noted earlier, $\boldsymbol{\mu}$ is bounded, and $\boldsymbol{F}$ is known in Eq. (3.38). Therefore,

$$||\boldsymbol{\mu}^{\mathrm{T}}|| \leq ||\boldsymbol{\mu}_0||,\ ||\boldsymbol{F}|| = 1. \tag{3.56}$$

If $\lambda_{min}(\boldsymbol{Q})$ is the minimum eigenvalue of $\boldsymbol{Q}$ and $\lambda_{max}(\boldsymbol{P})$ is the maximum eigenvalue of $\boldsymbol{P}$, then

$$\begin{aligned}\dot{V} &\leq -\frac{1}{2}\lambda_{min}(\boldsymbol{Q})||\boldsymbol{x}||^2 + ||\boldsymbol{\mu}_0||\lambda_{max}(\boldsymbol{P})||\boldsymbol{x}|| \\ &= -\frac{1}{2}||\boldsymbol{x}||[\lambda_{min}(\boldsymbol{Q})||\boldsymbol{x}|| - 2||\boldsymbol{\mu}_0||\lambda_{max}(\boldsymbol{P})]\end{aligned}. \tag{3.57}$$

To make $\dot{V} \leq 0$, $\lambda_{min}(\boldsymbol{Q})$ should satisfy the condition:

$$\lambda_{min}(\boldsymbol{Q}) \geq \frac{2\lambda_{max}(\boldsymbol{P})}{||\boldsymbol{x}||}||\boldsymbol{\mu}_0||. \tag{3.58}$$

Therefore, V is bounded and the error states are bounded in the closed loop system. Moreover, the tracking errors can approximate to zero with the integral term in $\boldsymbol{\tau}_1$.

3.2.4 Summary of neural network control

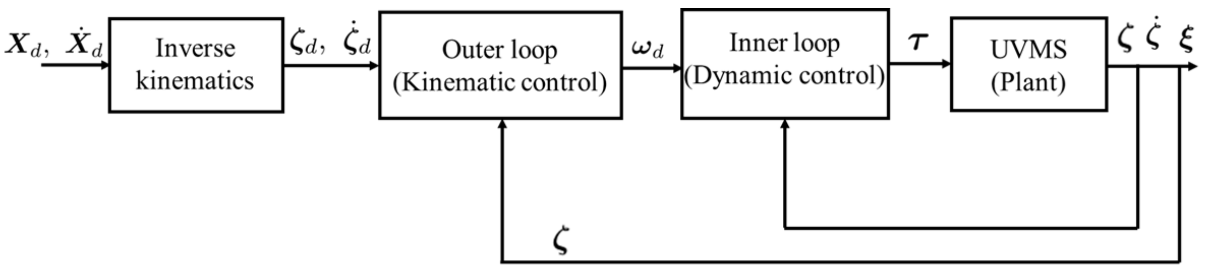
In this section, a nonlinear controller is used to eliminate the nonlinear dynamic term and a neural network is used to approximate system uncertainties. An adaptive law is given and the stability of the system is proved using Lyapunov function. The simulation results will be given in the last section of this chapter.

3.3 Adaptive dual integral sliding mode control

Sliding mode control is assumed to be a robust technique capable of stabilizing nonlinear systems in uncertain conditions [65][66]. However, chattering phenomenon exists when applied to a nonlinear system [67]. In a worse situation, the dynamic system can not remain robust against uncertainties. Therefore, integral sliding mode control was proposed in [68][69][70] to guarantee a strong robustness. This strategy handles problems based on the prerequisite that the nonlinear dynamics and all the states of the system are available [71]. However, not all states measurements are available in real applications and nonlinear dynamics is not exactly known. One approach is to use neural networks to enhance robustness by eliminating the uncertainties and disturbances [72][73]. Therefore, the combination of integral sliding mode control and neural networks can guarantee strong robustness for an uncertain and nonlinear system. In this part, an adaptive dual integral sliding mode controller is proposed based on the nominal model, which contains an inner dynamics control loop and an outer kinematics control loop. A neural network is designed to eliminate the uncertain disturbance term.

3.3.1 Proposed controller

The proposed controller contains three parts: an inner integral sliding mode dynamic control loop (also called velocity loop) is proposed to control the position of the UVMS; An outer kinematic control loop (also called position loop) is designed to give the desired velocity for the inner loop; A neural network is used to overcome the effect of uncertainties and disturbances. The combined three parts work together to provide strong robustness for the system. Before the explicit explanation of the three parts, the complete control scheme is shown in Fig. 3.5.



$X_d,\ \dot{X}_d$
Inverse kinematics
$\zeta_d,\ \dot{\zeta}_d$
Outer loop (Kinematic control)
ω_d
Inner loop (Dynamic control)
τ
UVMS (Plant)
$\zeta\ \dot{\zeta}\ \xi$
ζ

Figure 3.5: Complete control scheme

3.3.2 Inner loop design

Based on Eq. (2.38), we have

$$\dot{\xi} = \hat{M}^{-1}(\tau - \hat{C}\xi - \hat{D} - \hat{G} - d). \tag{3.59}$$

The scheme of inner loop controller is shown in Fig. 3.6. Vector ω_d is the reference velocity of the inner loop, which is the output of outer loop as well. The tracking error is written as:

$$\omega_e = \omega_d - \xi. \tag{3.60}$$

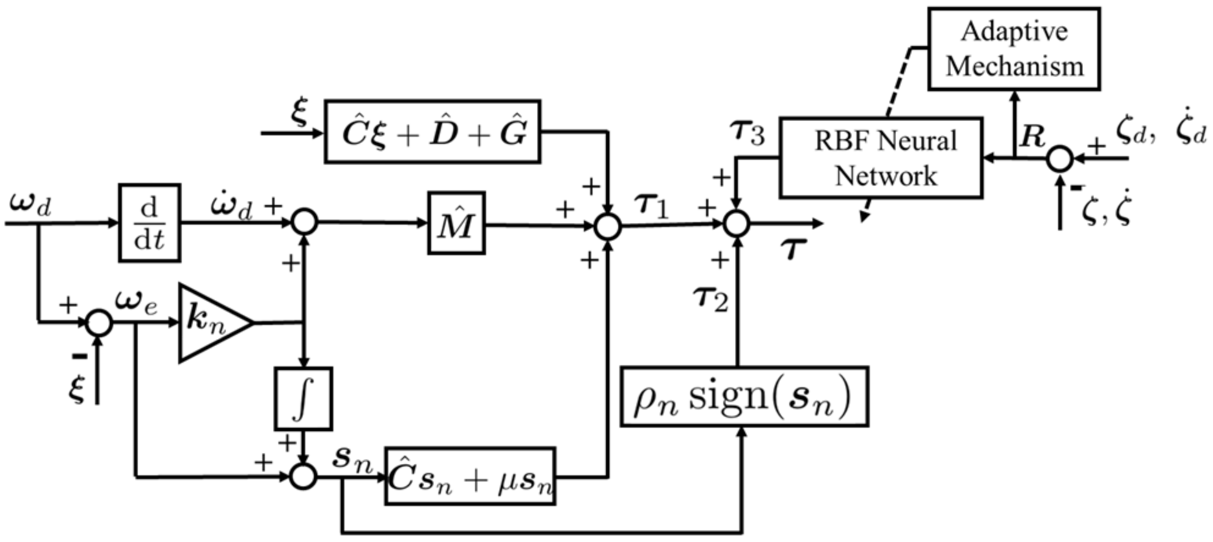


Figure 3.6: Inner loop scheme

Then, an integral sliding mode surface is defined as below,

$$\boldsymbol{s}_n = \boldsymbol{\omega}_e + \boldsymbol{k}_n \int_0^t \boldsymbol{\omega}_e \mathrm{d}t. \tag{3.61}$$

where $\boldsymbol{k}_n = \mathrm{diag}\{k_{n1},\ k_{n2},\ ...,\ k_{n8}\}$.

The derivative of the integral sliding mode surface is written as:

$$\dot{\boldsymbol{s}}_n = \dot{\boldsymbol{\omega}}_e + \boldsymbol{k}_n\boldsymbol{\omega}_e = \dot{\boldsymbol{\omega}}_d - \dot{\boldsymbol{\xi}} + \boldsymbol{k}_n\boldsymbol{\omega}_e. \tag{3.62}$$

Substituting Eq. (3.59) into Eq. (3.62), we have

$$\dot{\boldsymbol{s}}_n = \dot{\boldsymbol{\omega}}_d - \hat{\boldsymbol{M}}^{-1}(\boldsymbol{\tau} - \hat{\boldsymbol{C}}\boldsymbol{\xi} - \hat{\boldsymbol{D}} - \hat{\boldsymbol{G}} - \boldsymbol{d}) + \boldsymbol{k}_n\boldsymbol{\omega}_e. \tag{3.63}$$

Based on Eq. (3.63), the control input is designed as:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2 + \boldsymbol{\tau}_3. \tag{3.64}$$

where

$$\boldsymbol{\tau}_1 = \hat{\boldsymbol{M}}(\dot{\boldsymbol{\omega}}_d + \boldsymbol{k}_n\boldsymbol{\omega}_e) + \hat{\boldsymbol{C}}\boldsymbol{\xi} + \hat{\boldsymbol{D}} + \hat{\boldsymbol{G}} + \hat{\boldsymbol{C}}\boldsymbol{s}_n + \mu\boldsymbol{s}_n. \tag{3.65}$$

$$\boldsymbol{\tau}_2 = \rho_n \,\mathrm{sign}(\boldsymbol{s}_n). \tag{3.66}$$

$$\boldsymbol{\tau}_3 = \hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H}. \tag{3.67}$$

For the control input in Eq. (3.64), $\boldsymbol{\tau}_1$ is designed based on the nominal model and $\boldsymbol{\tau}_2$ is a robust term to overcome the effect of control errors. Parameters μ and ρ_n are constant numbers. To avoid trajectory tracking chattering, a neural network, $\boldsymbol{\tau}_3$, is utilized to eliminate the effect of uncertainties and disturbances.

Substituting Eq. (3.64), (3.65), (3.66), and (3.67) into Eq. (3.63),

$$\dot{\boldsymbol{s}}_n = -\hat{\boldsymbol{M}}^{-1}[\hat{\boldsymbol{C}}\boldsymbol{s}_n + \mu\boldsymbol{s}_n + \rho_n \operatorname{sign}(\boldsymbol{s}_n) - \widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H} - \boldsymbol{\mu}_0]. \tag{3.68}$$

where $\boldsymbol{\mu}_0$ is bounded and satisfies $||\boldsymbol{\mu}_0|| \leq \epsilon_0$.

3.3.3 Adaptive law and stability analysis

Define the following Lyapunov function for the UVMS,

$$V = \frac{1}{2}\boldsymbol{s}_n^{\mathrm{T}}\hat{\boldsymbol{M}}\boldsymbol{s}_n + \frac{1}{2\gamma}||\widetilde{\boldsymbol{W}}||^2. \tag{3.69}$$

The time derivative of Eq. (3.69) is

$$\dot{V} = \frac{1}{2}\boldsymbol{s}_n^{\mathrm{T}}\dot{\hat{\boldsymbol{M}}}\boldsymbol{s}_n + \boldsymbol{s}_n^{\mathrm{T}}\hat{\boldsymbol{M}}\dot{\boldsymbol{s}}_n + \frac{1}{\gamma}\operatorname{tr}(\widetilde{\boldsymbol{W}}^{\mathrm{T}}\dot{\widetilde{\boldsymbol{W}}}). \tag{3.70}$$

Because $\dot{\hat{\boldsymbol{M}}} - 2\hat{\boldsymbol{C}}$ is a skew-symmetric matrix, then substituting Eq. (3.68) into Eq. (3.70) yields:

$$\begin{aligned}\dot{V} &= -\boldsymbol{s}_n^{\mathrm{T}}[\mu\boldsymbol{s}_n + \rho_n \operatorname{sign}(\boldsymbol{s}_n) - \boldsymbol{\mu}_0 - \widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H}] \\ &\quad + \frac{1}{\gamma}\operatorname{tr}(\widetilde{\boldsymbol{W}}^{\mathrm{T}}\dot{\widetilde{\boldsymbol{W}}}) \\ &= -\mu\boldsymbol{s}_n^{\mathrm{T}}\boldsymbol{s}_n - (|\boldsymbol{s}_n|\rho_n - \boldsymbol{s}_n^{\mathrm{T}}\boldsymbol{\mu}_0) + \boldsymbol{s}_n^{\mathrm{T}}\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H} \\ &\quad + \frac{1}{\gamma}\operatorname{tr}(\widetilde{\boldsymbol{W}}^{\mathrm{T}}\dot{\widetilde{\boldsymbol{W}}}).\end{aligned} \tag{3.71}$$

Because

$$\boldsymbol{s}_n^{\mathrm{T}}\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H} = \frac{1}{\gamma}\operatorname{tr}(\gamma\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{s}_n^{\mathrm{T}}). \tag{3.72}$$

Select the adaptive law as:

$$\dot{\widetilde{\boldsymbol{W}}} = -\dot{\hat{\boldsymbol{W}}} = -\gamma\boldsymbol{H}\boldsymbol{s}_n^{\mathrm{T}}. \tag{3.73}$$

Substituting Eq. (3.72) and (3.73) into Eq. (3.71), we have

$$\dot{V} = -\mu \boldsymbol{s}_n^{\mathrm{T}} \boldsymbol{s}_n - (|\boldsymbol{s}_n|\rho_n - \boldsymbol{s}_n^{\mathrm{T}} \boldsymbol{\mu}_0). \tag{3.74}$$

If ρ_n is chosen as $\rho_n > ||\boldsymbol{\mu}_0||$, then $\dot{V} < 0$, which means the inner loop is asymptotically stable.

3.3.4 Outer loop design

The outer loop is a position loop, which is used to generate desired velocity inputs for the inner loop. The outer loop scheme is shown in Fig. 3.7.

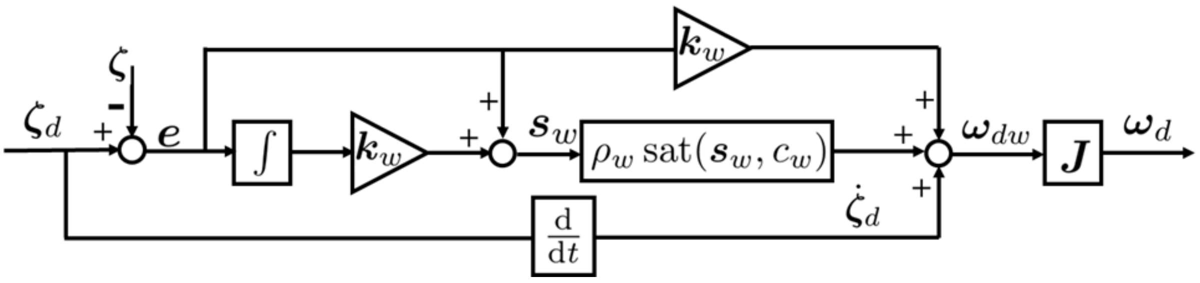


Figure 3.7: Outer loop scheme

Give the desired position input $\boldsymbol{\zeta}_d$, the tracking error is:

$$\boldsymbol{e} = \boldsymbol{\zeta}_d - \boldsymbol{\zeta}. \tag{3.75}$$

The outer loop integral sliding mode function is defined as:

$$\boldsymbol{s}_w = \boldsymbol{e} + \boldsymbol{k}_w \int_0^t \boldsymbol{e} \mathrm{d}t. \tag{3.76}$$

where $\boldsymbol{k}_w = \mathrm{diag}\{k_{w1}, k_{w2}, ..., k_{w8}\}$. Then,

$$\dot{\boldsymbol{s}}_w = \dot{\boldsymbol{e}} + \boldsymbol{k}_w \boldsymbol{e} = \dot{\boldsymbol{\zeta}}_d - \dot{\boldsymbol{\zeta}} + \boldsymbol{k}_w \boldsymbol{e}. \tag{3.77}$$

Based on the inner velocity loop analysis, there exists a small error vector $\boldsymbol{\epsilon}$ to satisfy:

$$\dot{\boldsymbol{\zeta}} = \boldsymbol{\omega}_{dw} + \boldsymbol{\epsilon}. \tag{3.78}$$

where $\boldsymbol{\omega}_{dw}$ is the output velocity of the outer loop, in which the velocity of vehicle is expressed in inertial frame. The relationship between $\boldsymbol{\omega}_{dw}$ and $\boldsymbol{\omega}_d$ is written as:

$$\boldsymbol{\omega}_d = \boldsymbol{J}_I^b \boldsymbol{\omega}_{dw}. \tag{3.79}$$

Therefore,

$$\dot{\boldsymbol{s}}_w = \dot{\boldsymbol{\zeta}}_d - \boldsymbol{\omega}_{dw} - \boldsymbol{\epsilon} + \boldsymbol{k}_w \boldsymbol{e}. \tag{3.80}$$

Here, $\boldsymbol{\omega}_{dw}$ is selected as:

$$\boldsymbol{\omega}_{dw} = \dot{\boldsymbol{\zeta}}_d + \boldsymbol{k}_w \boldsymbol{e} + \rho_w \operatorname{sat}(\boldsymbol{s}_w, c_w). \tag{3.81}$$

where ρ_w is a constant number. $\operatorname{sat}(\boldsymbol{s}_w, c_w)$ is a saturation function, which has the following form:

$$\operatorname{sat}(\Delta, c) = \begin{cases} \frac{\Delta}{c}, & |\Delta| \le c \\ \operatorname{sign}(\Delta), & |\Delta| > c \end{cases} \tag{3.82}$$

where c is a constant number.

Substituting Eq. (3.81) into Eq. (3.80) yields:

$$\dot{\boldsymbol{s}}_w = -\rho_w \operatorname{sat}(\boldsymbol{s}_w, c_w) - \boldsymbol{\epsilon}. \tag{3.83}$$

Then,

$$\boldsymbol{s}_w^{\mathrm{T}} \dot{\boldsymbol{s}}_w = -\rho_w \boldsymbol{s}_w^{\mathrm{T}} \operatorname{sat}(\boldsymbol{s}_w, c_w) - \boldsymbol{s}_w^{\mathrm{T}} \boldsymbol{\epsilon}. \tag{3.84}$$

If ρ_w is selected large enough to satisfy $\boldsymbol{s}_w^{\mathrm{T}} \dot{\boldsymbol{s}}_w \le 0$, then the outer position loop is stable.

3.3.5 Summary of dual integral sliding mode control

In this part, a dual integral sliding mode controller is proposed and the stability is analyzed. A neural network is adopted to eliminate system uncertainties as well. Lyapunov function shows its robustness and the performance is shown in next section.

3.4 Simulation results and discussion

In this section, the application of the proposed three joint space trajectory tracking control strategies are simulated and the performance is compared.

3.4.1 Description of the UVMS

The UVMS model consists of one ellipsoid vehicle and a three-joint manipulator with cylindrical links. The geometric and mass properties and the hydrodynamic parameters of the UVMS are shown in Table. 3.1 and 3.2. Vector $\boldsymbol{r}_{b0}^{b}$ is set to be $\boldsymbol{r}_{b0}^{b} = [0\ 0\ -0.4]^{\mathrm{T}}\mathrm{m}$.

	Mass (kg)	Dimensions (m)	I_{xx} $(kg \cdot m^2)$	I_{yy} $(kg \cdot m^2)$	I_{zz} $(kg \cdot m^2)$	$\triangle$ (m^3)
Vehicle	200	$b = 0.5$ $a, c = 0.3$	13.6	7.2	13.6	0.1885
Link 1	5	$L = 0.3$ $r = 0.025$	0.0407	0.0407	0.0063	5.89×10^{-4}
Link 2	10	$L = 0.4$ $r = 0.025$	0.0093	0.1047	0.1047	7.854×10^{-4}
Link 3	10	$L = 0.4$ $r = 0.025$	0.0093	0.1047	0.1047	7.854×10^{-4}

Table 3.1: The mass and geometric parameters of the UVMS

Vehicle	$M_{rA} = diag\{38.633,\ 72.54,\ 38.633,\ 0.579,\ 0,\ 0.579\}$ $\boldsymbol{D}_1(\boldsymbol{v}) = diag\{-20,\ -7,\ -20,\ -10,\ -3,\ -10\}$ $\boldsymbol{D}_2(\boldsymbol{v}) = diag\{-150,\ -100,\ -150,\ -50,\ -7,\ -50\}$
Link 1	$M_{rA} = diag\{0.589,\ 0.589,\ 0.5,\ 0.0044,\ 0.0044,\ 0\}$ $C_D = 1.1$
Link 2	$M_{rA} = diag\{1,\ 0.7854,\ 0.7854,\ 0,\ 0.0105,\ 0.0105\}$ $C_D = 1.1$
Link 3	$M_{rA} = diag\{1,\ 0.7854,\ 0.7854,\ 0,\ 0.0105,\ 0.0105\}$ $C_D = 1.1$

Table 3.2: Hydrodynamic parameters

3.4.2 Description of the task

A desired joint trajectory is given to test the three controllers. The desired motion of the vehicle is to remain stationary, which means $\boldsymbol{\eta}_d = 0$. The desired joint motion is:

$$\begin{aligned} q_{1d} &= \sin(-\frac{\pi}{10}t) \\ q_{2d} &= \sin(\frac{\pi}{10}t) + \frac{\pi}{6} \\ q_{3d} &= \sin(\frac{\pi}{10}t) - \frac{\pi}{6} \end{aligned} \tag{3.85}$$

3.4.3 Parameters of the controllers

A. Parameters of backstepping control

For each input variable $\boldsymbol{H}$ of the fuzzy logic, the Gaussian membership functions are defined as

$$\mu_j^1 = e^{-\frac{1}{2}(\frac{h_i+1}{0.5})^2} \tag{3.86}$$

$$\mu_j^2 = e^{-\frac{1}{2}(\frac{h_i}{0.5})^2} \tag{3.87}$$

$$\mu_j^3 = e^{-\frac{1}{2}(\frac{h_i-1}{0.5})^2} \tag{3.88}$$

The initial $\boldsymbol{\Phi}$ and other parameters are set as follows:

$$\begin{aligned} &\boldsymbol{\Phi} = 0.1 \times \mathbf{1}_{3\times 9} \\ &k = 100,\ \gamma = 30,\ \lambda_1 = 50 \\ &\boldsymbol{\lambda}_2 = diag\{500\ 500\ 1000\ 100\ 100\ 100\ 100\ 100\ 100\} \end{aligned} \tag{3.89}$$

B. Parameters of the neural network control

The center point vector matrix of the neural network is :

$$\begin{aligned} \boldsymbol{C}_{neural} &= [\boldsymbol{c}_1\ \boldsymbol{c}_2\ \boldsymbol{c}_3\ \boldsymbol{c}_4\ \boldsymbol{c}_5] \\ &= [-2\mathbf{1}_{27\times 1}\ -\mathbf{1}_{27\times 1}\ \mathbf{0}_{27\times 1}\ \mathbf{1}_{27\times 1}\ 2\mathbf{1}_{27\times 1}] \end{aligned} \tag{3.90}$$

The stretch constant vector is selected as $\boldsymbol{B} = [1\ 1\ 1\ 1\ 1]^{\mathrm{T}}$. The initial weight matrix is set as $\boldsymbol{W} = 0.1 \times \mathbf{1}_{5\times 9}$. The matrix $\boldsymbol{Q}$ is selected as $\boldsymbol{Q} = 200 \times \boldsymbol{I}_{27}$. The adaptation gain is $\gamma = 50$. Gain matrices $\boldsymbol{K}_P$, $\boldsymbol{K}_D$, and $\boldsymbol{K}_I$ are selected as:

$$\begin{aligned} \boldsymbol{K}_P &= diag\{[250\ 250\ 250\ 100\ 100\ 100\ 250\ 250\ 250]\} \\ \boldsymbol{K}_D &= diag\{[50\ 50\ 50\ 50\ 50\ 50\ 50\ 100\ 100]\} \\ \boldsymbol{K}_I &= diag\{[50\ 50\ 50\ 50\ 50\ 50\ 50\ 100\ 100]\} \end{aligned} \tag{3.91}$$

C. Parameters of the dual integral sliding mode control

The center point vector matrix of the neural network is :

$$\begin{aligned} \boldsymbol{C}_{neural} &= [\boldsymbol{c}_1\ \boldsymbol{c}_2\ \boldsymbol{c}_3\ \boldsymbol{c}_4\ \boldsymbol{c}_5] \\ &= [-2\mathbf{1}_{18\times 1}\ -\mathbf{1}_{18\times 1}\ \mathbf{0}_{18\times 1}\ \mathbf{1}_{18\times 1}\ 2\mathbf{1}_{18\times 1}] \end{aligned} \tag{3.92}$$

The stretch constant vector is selected as $\boldsymbol{B} = [3\ 3\ 3\ 3\ 3]^{\mathrm{T}}$. The adaptation gain is $\gamma = 2$. For the inner integral sliding mode control, $\boldsymbol{k}_n = \boldsymbol{I}_{9\times 9}$, $\rho_n = 1.5$, and $\mu = 150$. For the outer integral sliding mode loop, $\boldsymbol{k}_w = \boldsymbol{I}_{8\times 8}$, $\rho_w = 1$, and $c_w = 0.003$.

3.4.4 Simulation results and discussion

In this part, simulation results are given to show the performance of the three controllers. To demonstrate the results briefly, backstepping control, neural network control, and dual integral sliding mode control are written as BS, NN, and DL, respectively. The vehicle position and attitude response are shown in Fig. 3.8 - 3.13. It can be seen that there are steady state tracking errors for the backstepping control. As explained in Section. 3.1, the adaptive law and parameters are selected to guarantee system stability and tracking errors can be made as small as desired by adjusting the control parameters, which means tracking errors can not be eliminated completely. For the neural network control strategy, tracking errors converge to zero due to the integral term in the control model. For the dual integral sliding mode control, it can be seen that the UVMS remains highly robustness and the tracking errors converge to zero quickly.

The periodic desired joint trajectories are shown in Fig. 3.14. The joint tracking errors are shown in Fig. 3.15 - 3.17. For the backstepping control, there are fluctuations when the joint starts to move because the fuzzy logic adjusts its parameters adaptively. For the neural network control, similar tracking errors occur in every period. The UVMS keeps high robustness and the tracking errors remains very small for the dual integral sliding mode control.

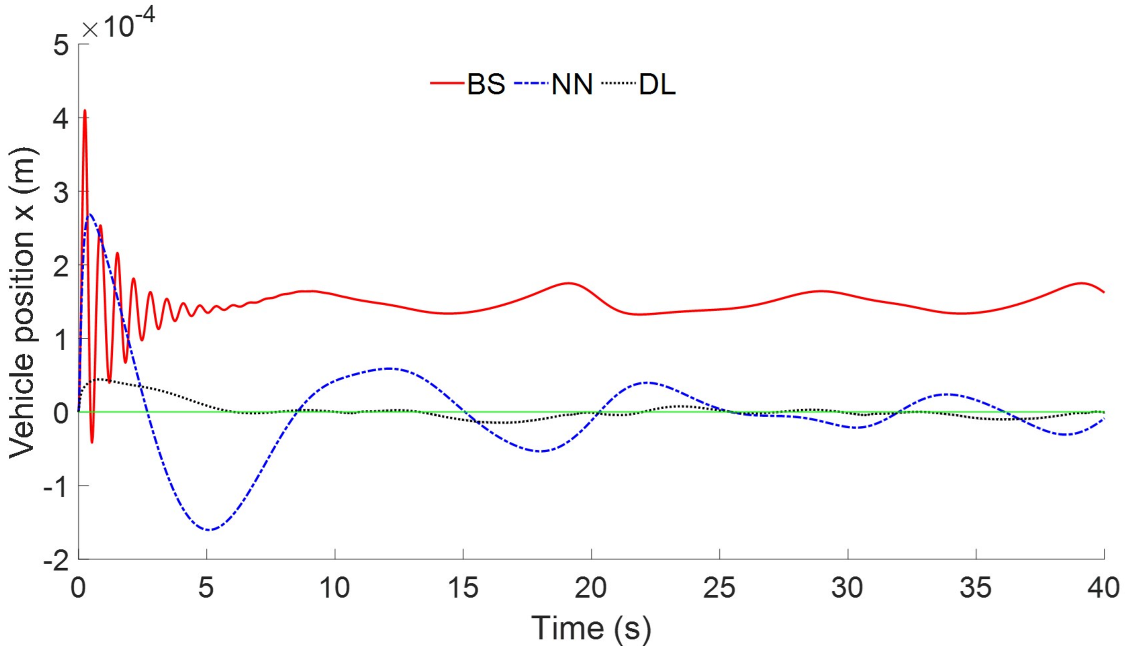


Figure 3.8: Vehicle position in x direction

3.5 Summary

In this chapter, three joint space trajectory tracking controllers, backstepping control, neural network control, and dual integral sliding mode control, are proposed and tested. Backstepping control is a decentralized strategy which can be applied to nonlinear system directly. Dynamic model of the UVMS is not used in this method, which means there is no need to develop the dynamic model for complex system. A fuzzy logic is designed to approximate uncertain dynamic terms. In the neural network control part, a nonlinear controller based on nominal dynamic model is given and an adaptive neural network is proposed to eliminate system uncertainties. Considering the advantages of integral sliding mode control and neural network, the

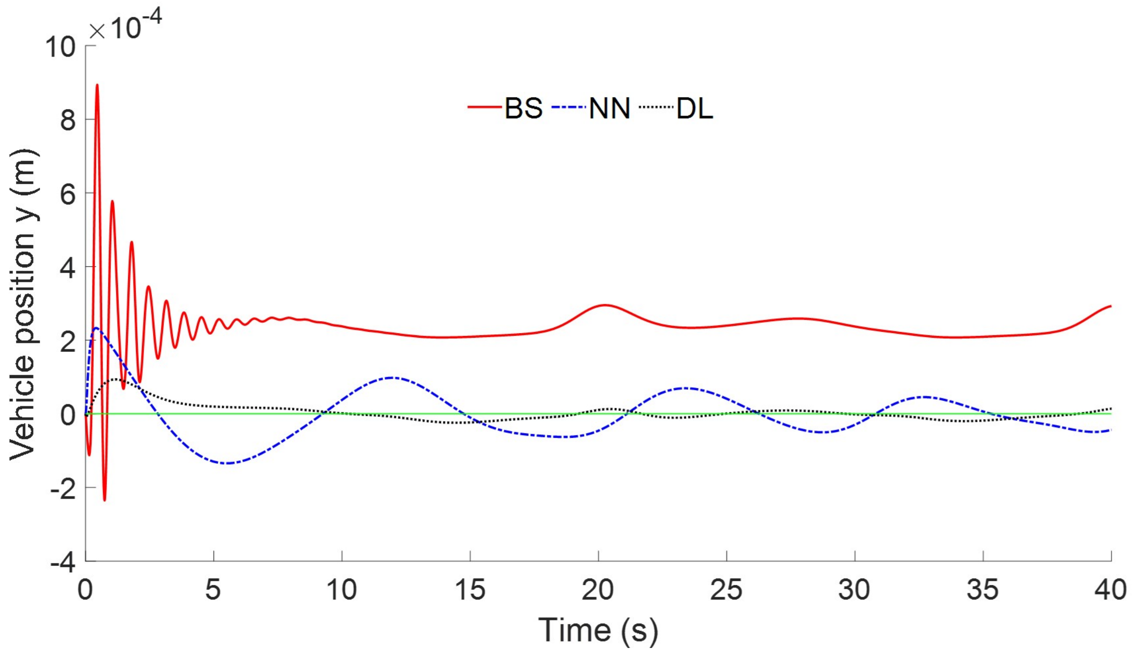


Figure 3.9: Vehicle position in y direction

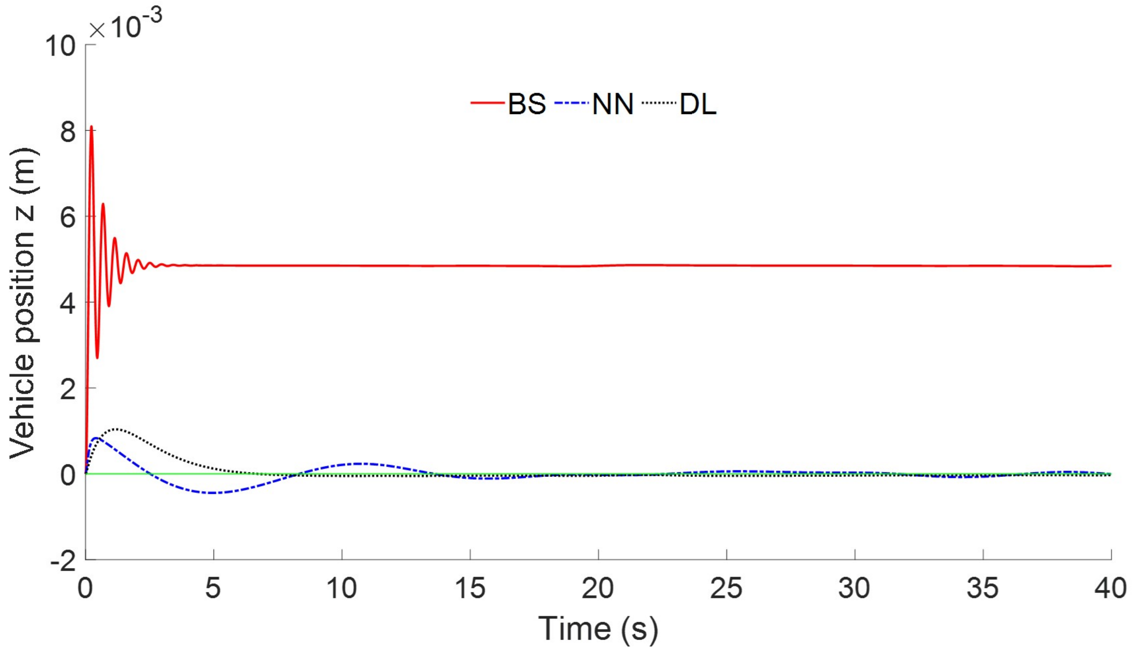


Figure 3.10: Vehicle position in z direction

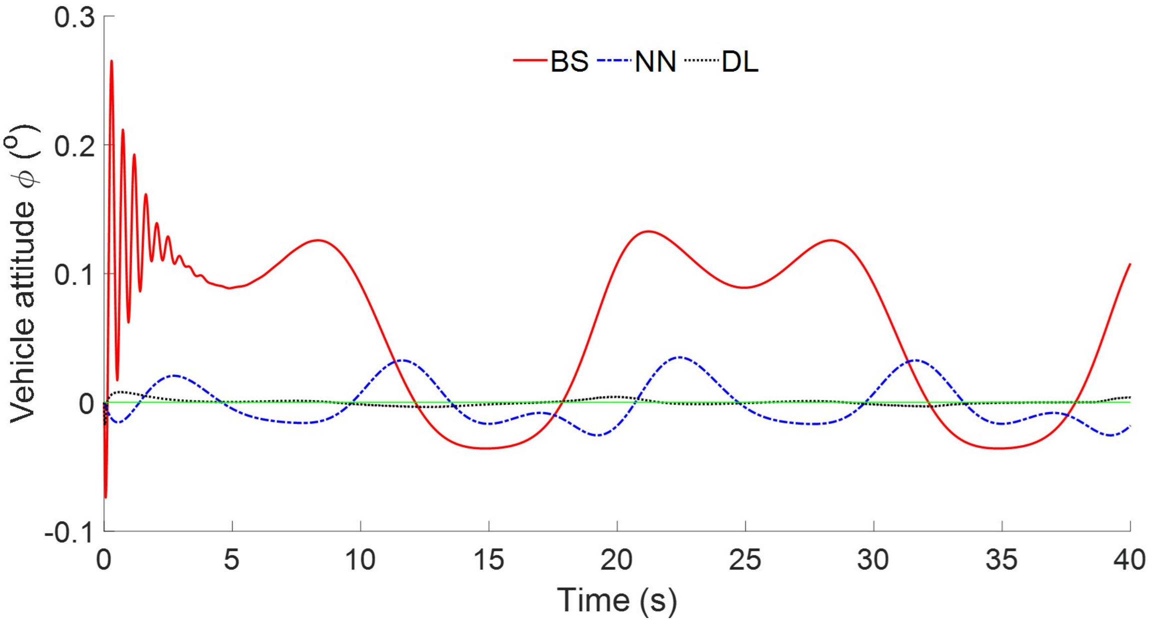


Figure 3.11: Vehicle attitude along x-axis

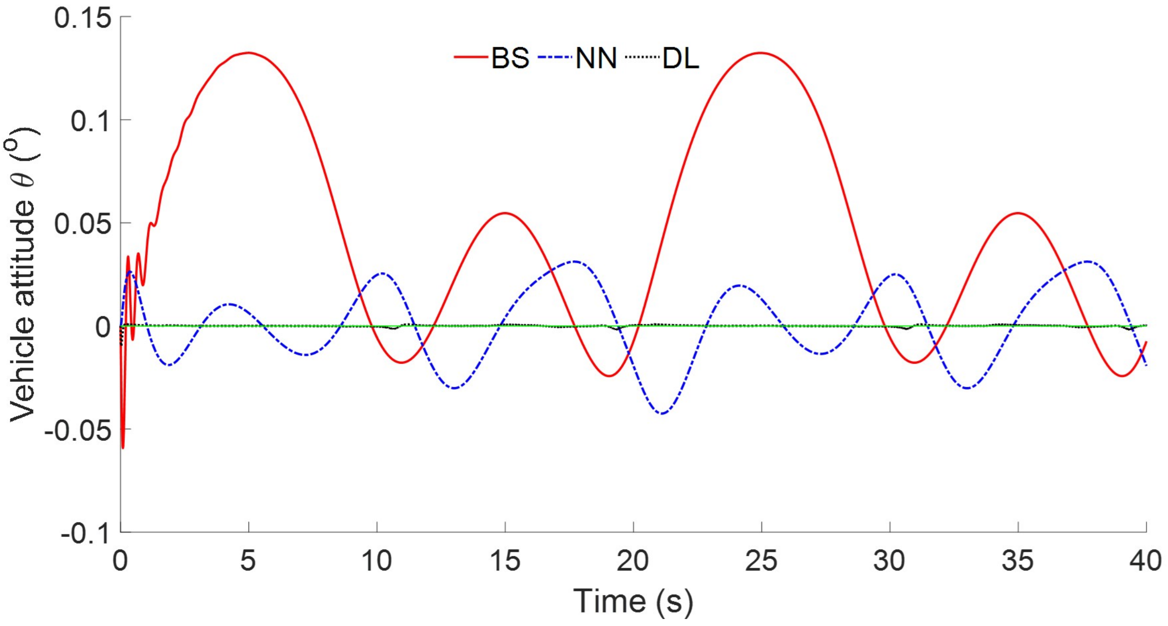


Figure 3.12: Vehicle attitude along y-axis

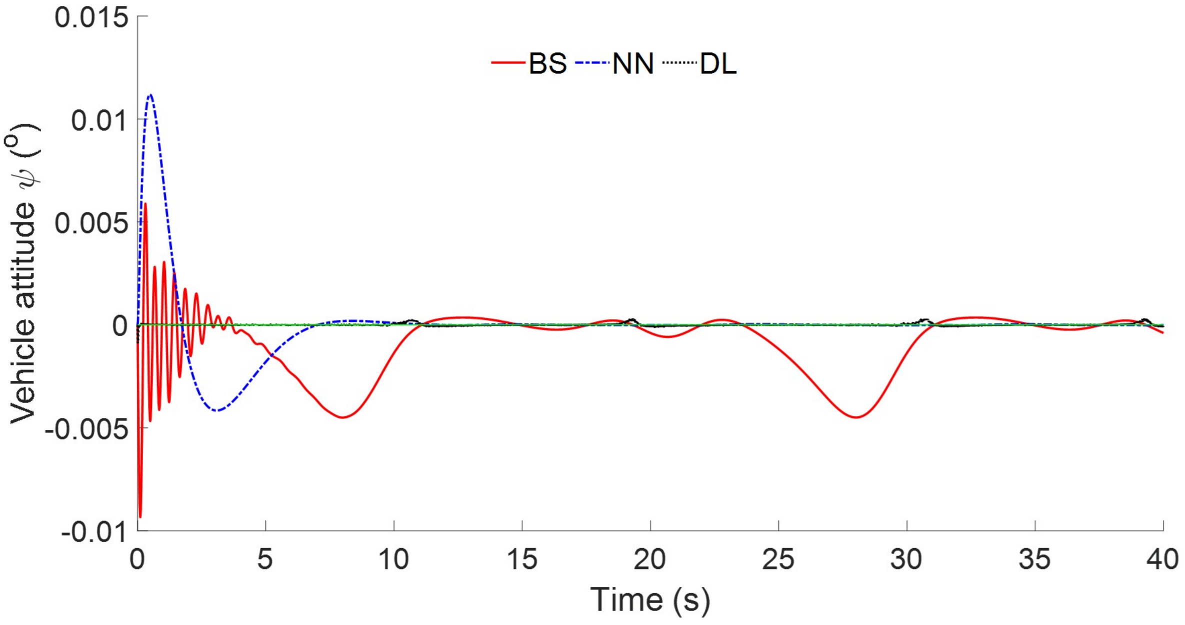


Figure 3.13: Vehicle attitude along z-axis

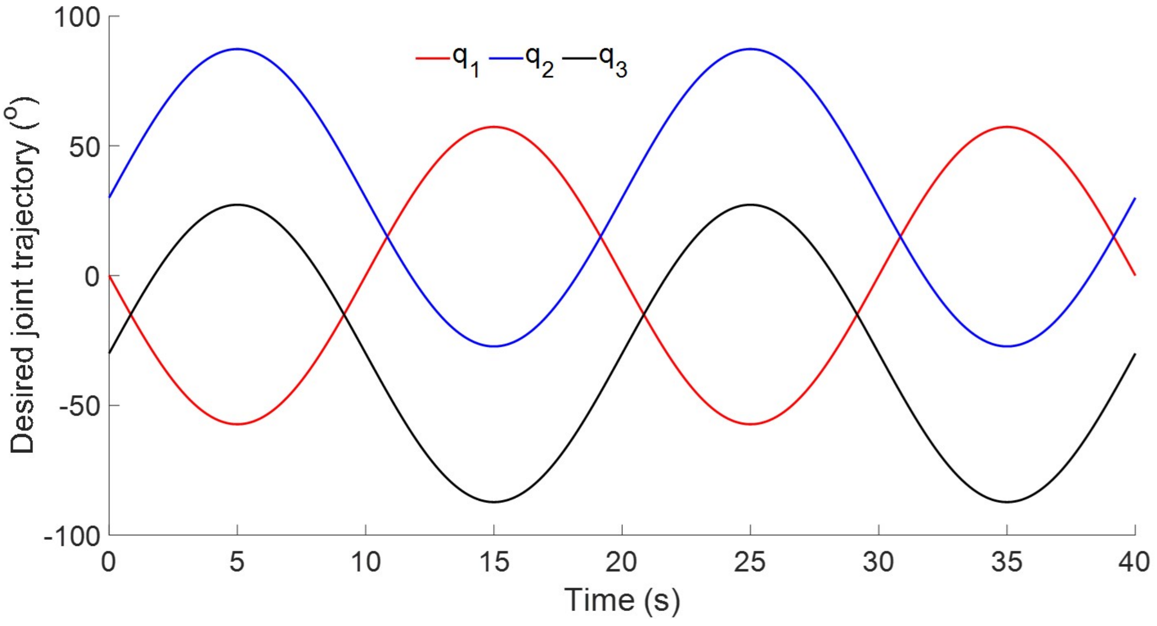


Figure 3.14: Desired joint trajectory

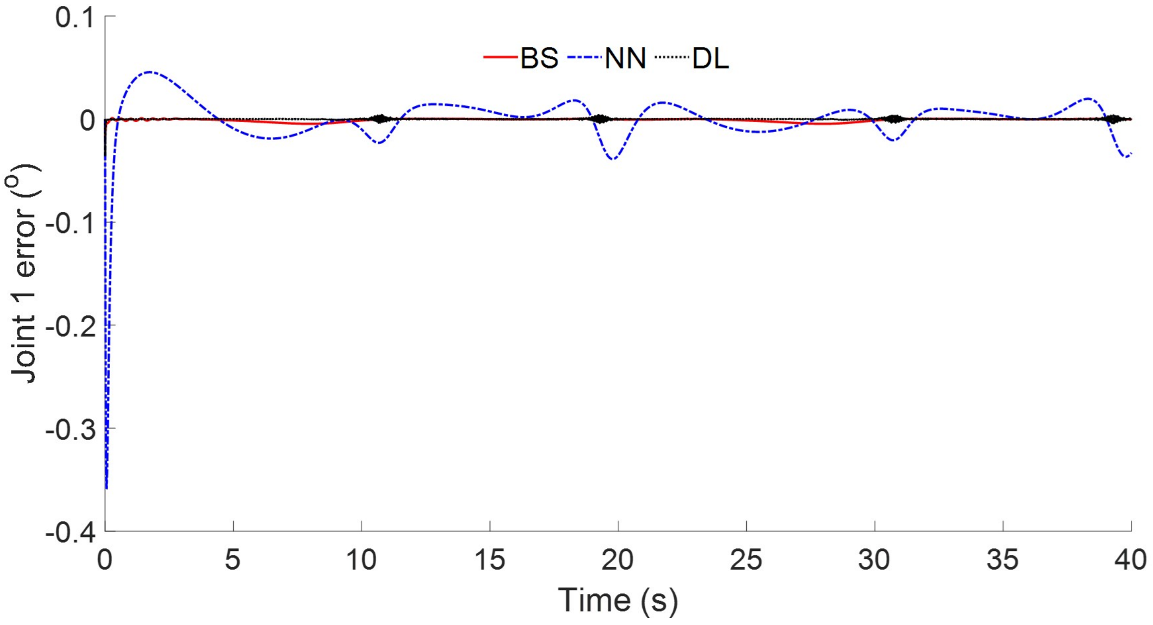


Figure 3.15: Joint 1 error

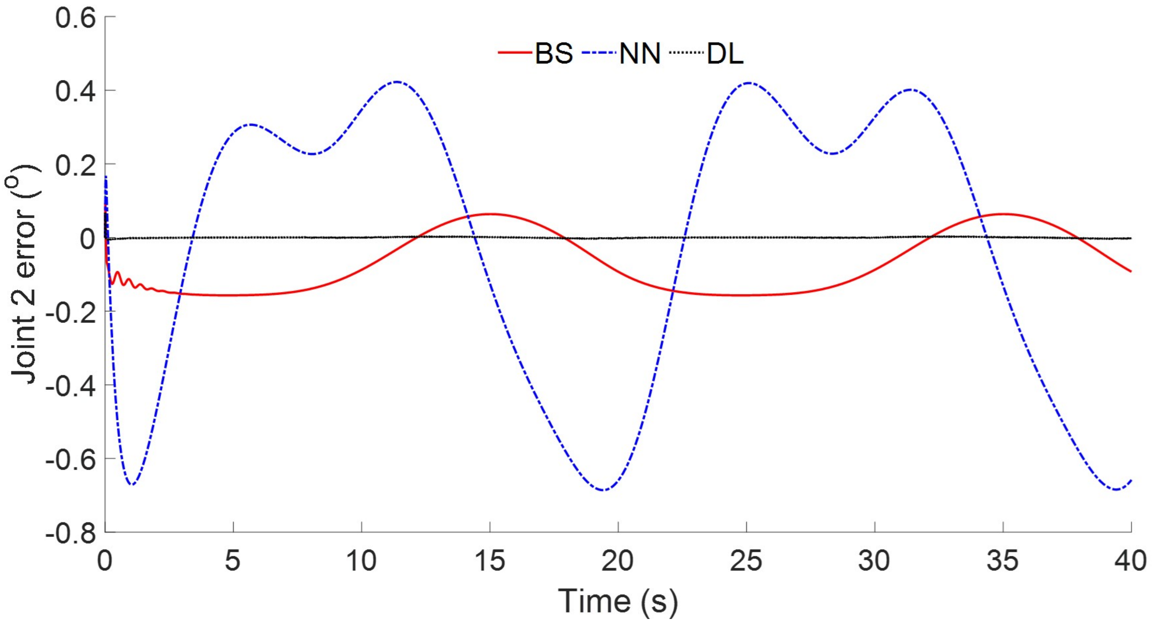


Figure 3.16: Joint 2 error

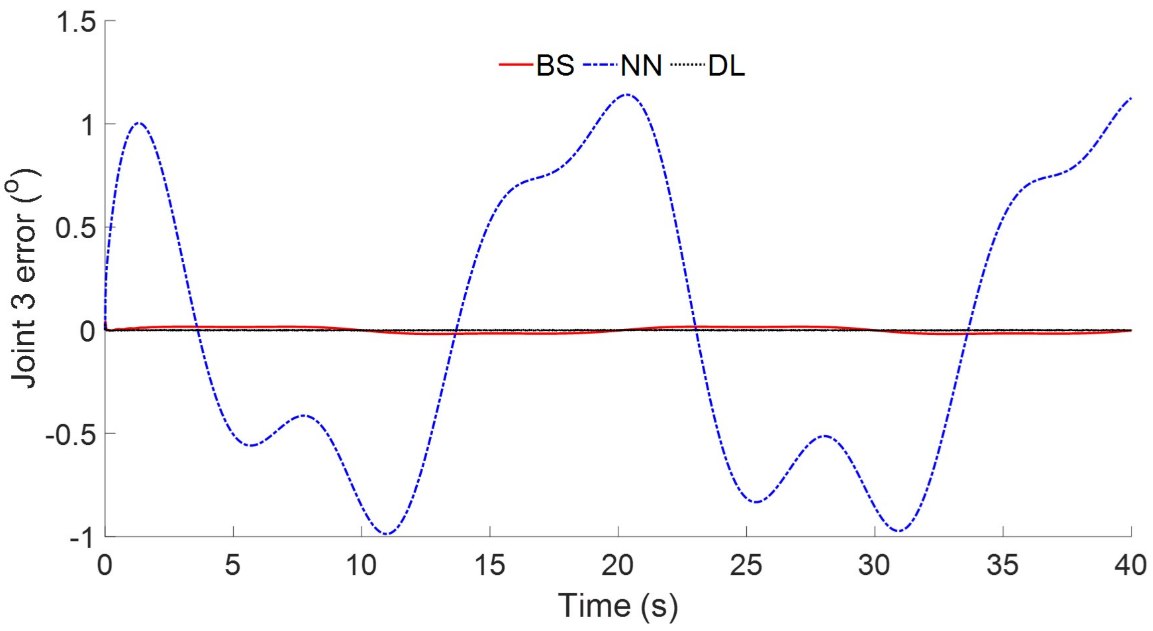


Figure 3.17: Joint 3 error

combination of the two parts are utilized to control the UVMS and the performance shows high robustness. The advantages and disadvantages are summarized as below.

Advantages:

Backstepping control: Dynamic model is not necessary, which means it is a good choice for complex models; It is a decentralized control scheme with a systematic recursive design procedure; Tracking errors can be made as small as possible by adjusting controller parameters.

Neural network control: Steady state tracking errors are eliminated; Neural network is used to approximate system uncertainties.

Dual integral sliding mode control: This method shows high robustness when used to complete tasks.

Disadvantages:

Backstepping control: Steady state tracking errors exist and can not be eliminated; There are fluctuations at the beginning.

Neural network control: Periodic tracking errors exist for periodic joint motion.

Dual integral sliding mode control: The design process is complex.

Chapter 4

Task space trajectory tracking and force control

Robot dynamics are typically modeled in joint space coordinates, but ultimately, such systems must complete their work in the task space. In the previous chapter, an inverse kinematics model is necessary for joint space control, and joint space controllers can be applied to finish simple tasks. Inverse kinematics can be difficult to derive for complex tasks, however, and may notbe unique, especially for redundant manipulators, including UVMS. To address such challenges, a task space trajectory tracking and force control strategy is proposed in this chapter and computer simulations explore its effectiveness.

4.1 Problem specification

The dynamic model of (2.38) can be written as:

$$\hat{\boldsymbol{M}}\dot{\boldsymbol{\xi}} + \boldsymbol{H} + \boldsymbol{\delta} = \boldsymbol{\tau}. \tag{4.1}$$

where $\boldsymbol{H} = \hat{\boldsymbol{C}}\boldsymbol{\xi} + \hat{\boldsymbol{D}} + \hat{\boldsymbol{G}}$. $\boldsymbol{\delta}$ is a clumped uncertainty term. The inertia matrix $\hat{\boldsymbol{M}}$ is symmetric and positive definete. Therefore, we have:

$$\dot{\boldsymbol{\xi}} + \hat{\boldsymbol{M}}^{-1}(\boldsymbol{H} + \boldsymbol{\delta}) = \hat{\boldsymbol{M}}^{-1}\boldsymbol{\tau}. \tag{4.2}$$

For a robot system, the force in joint space can be seen as a mapping from forces in task space. The relationship is expressed as

$$\boldsymbol{\tau} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{F}_{ee}. \tag{4.3}$$

where $\boldsymbol{J}$ is the Jacobian matrix mapping joint velocities into task space. Vector $\boldsymbol{F}_{ee}$ is an equivalent external force excerted on the end-effector.

The velocity of end-effector expressed in task space is

$$\dot{\boldsymbol{x}}_e = \boldsymbol{J}\boldsymbol{\xi}. \tag{4.4}$$

Differentiating both sides, we have

$$\ddot{\boldsymbol{x}}_e = \boldsymbol{J}\dot{\boldsymbol{\xi}} + \dot{\boldsymbol{J}}\boldsymbol{\xi}. \tag{4.5}$$

Substituting Eq. (4.3) and (4.5) into Eq. (4.2), we have

$$\boldsymbol{\Lambda}\ddot{\boldsymbol{x}}_e + \boldsymbol{\Lambda}\boldsymbol{J}\hat{\boldsymbol{M}}^{-1}(\boldsymbol{H} + \boldsymbol{\delta}) - \boldsymbol{\Lambda}\dot{\boldsymbol{J}}\boldsymbol{\xi} = \boldsymbol{F}_{ee}. \tag{4.6}$$

where $\boldsymbol{\Lambda} = (\boldsymbol{J}\hat{\boldsymbol{M}}^{-1}\boldsymbol{J}^{\mathrm{T}})^{-1}$. In order to control the trajectory of end-effector in task space, one classical strategy commonly adopted is

$$\boldsymbol{F}_{ee} = \boldsymbol{\Lambda}(\ddot{\boldsymbol{x}}_{ed} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_i \int \boldsymbol{e}) + \boldsymbol{\Lambda}\boldsymbol{J}\hat{\boldsymbol{M}}^{-1}\boldsymbol{H} - \boldsymbol{\Lambda}\dot{\boldsymbol{J}}\boldsymbol{\xi}. \tag{4.7}$$

where $\boldsymbol{e} = \boldsymbol{x}_{ed} - \boldsymbol{x}_e$ and $\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_{ed} - \dot{\boldsymbol{x}}_e$. Vectors $\boldsymbol{x}_{ed}$ and $\dot{\boldsymbol{x}}_{ed}$ are the desired end-effector position and velocity. Diagonal matrices $\boldsymbol{K}_v$, $\boldsymbol{K}_p$, and $\boldsymbol{K}_i$ are the corresponding gains, respectively. The integral term is used to eliminate steady-state tracking errors. Substituting Eq. (4.7) into Eq. (4.6), we have

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_i \int \boldsymbol{e} = \boldsymbol{\Delta}. \tag{4.8}$$

where $\boldsymbol{\Delta} = \boldsymbol{J}\hat{\boldsymbol{M}}^{\mathrm{T}}\boldsymbol{\delta}$. The existance of uncertainty term $\boldsymbol{\Delta}$ can affect the tracking accuracy.

For a redundant UVMS, there are infinite joint velocity solutions corresponding to one specific velocity vector in task space. Similarly, due to the redundancy, there are infinite joint space force solutions corresponding to one specific task space force vector as well. Therefore, one risk of using Eq. (4.3) as control input is that the system is not guaranteed to be stable in joint space. The relationship between control inputs in joint space and task space is shown in Fig. 4.1. A stable task space force vector corresponds to infinite joint space forces, which cover both the stable and unstable areas. An unstable task space force vector corresponds an unstable force area in joint space. Therefore, a stable task space input $\boldsymbol{F}_{ee}$ is not guaranteed to correspond to a stable area in joint space. Fig. 4.2 shows a case where the task space force input lies in the stable area while the corresponding joint space force input lies in the unstable area. The end-effector is able to track the desired task space, which means it is stable in task space. However, the roll angle ϕ increases exponentially, which means it is unstable in joint space.

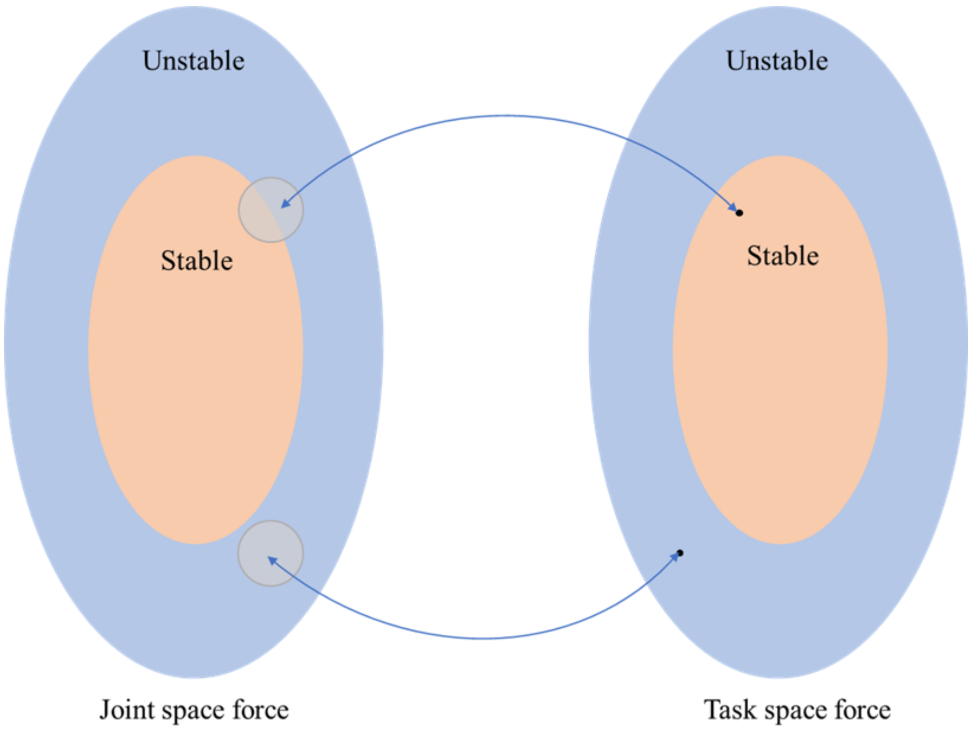
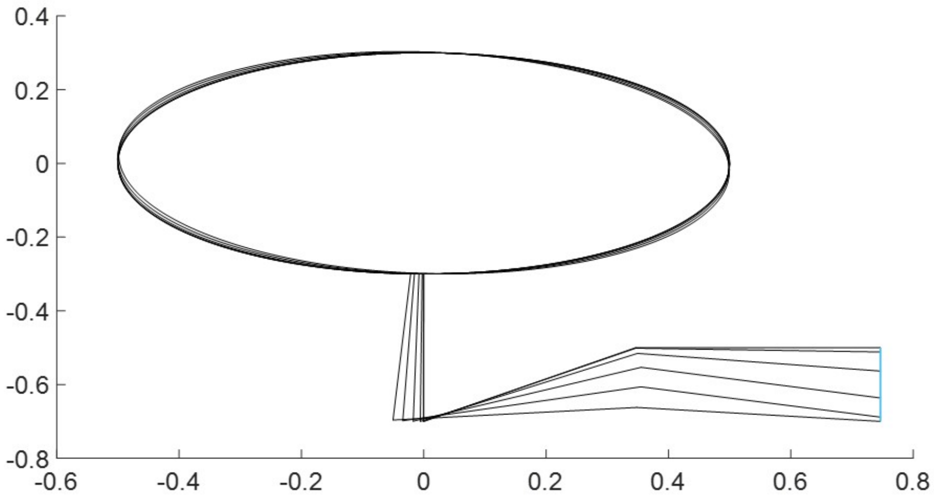


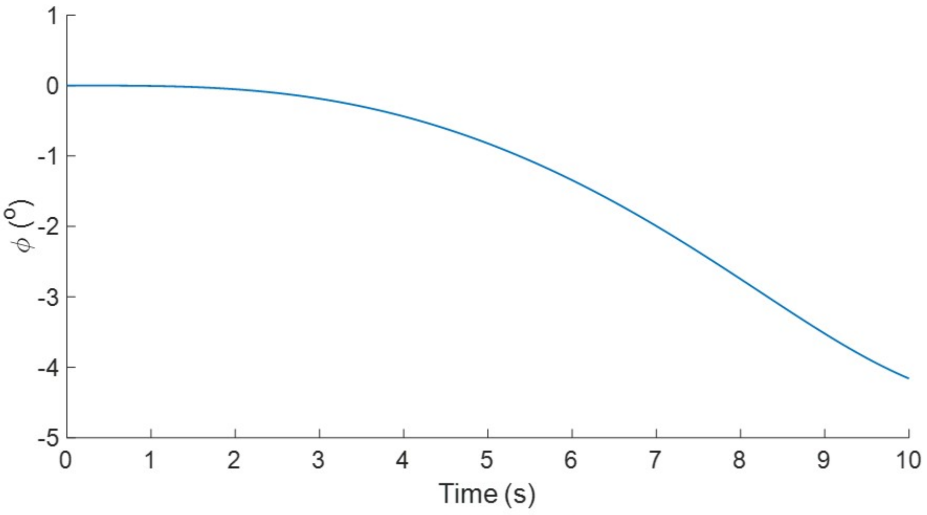
Figure 4.1: The relationship between joint and task space force

4.2 Task space trajectory tracking control

Considering the issues in section 4.1, we have the following concerns when designing a control strategy in task space:



(a). Motion process



(b). Roll angle

Figure 4.2: Unstable case

1. The mapping relationship of Eq. (4.3) should be modified to make sure a stable control input in task space lies in the stable area in joint space as well.
2. A compensator should be designed to eliminate the effect of uncertainty term $\boldsymbol{\Delta}$.
3. The stability should be verified in both joint space and task space.

4.2.1 Task space trajectory tracking controller design

The mapping relationship of Eq. (4.3) should be modified to guarantee system stability. Let us define:

$$\tilde{\boldsymbol{J}} = \hat{\boldsymbol{M}}^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\Lambda}. \tag{4.9}$$

$\tilde{\boldsymbol{J}}$ is a generalized inverse of the Jacobian matrix corresponding to the system that minimizes the instantaneous kinetic energy of the UVMS [74][75].

Lemma: The unconstrained end-effector is subjected to the task space force $\boldsymbol{F}_{ee}$ if and only if the generalized joint space control input is expressed as:

$$\boldsymbol{\tau} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{F}_E + \boldsymbol{\tau}_N. \tag{4.10}$$

where

$$\boldsymbol{\tau}_N = (\boldsymbol{I}_{(6+n)} - \boldsymbol{J}^{\mathrm{T}}\tilde{\boldsymbol{J}}^{\mathrm{T}})\boldsymbol{\tau}_0. \tag{4.11}$$

Here, $\boldsymbol{I}_{(6+n)}$ is a $(6+n)\times(6+n)$ identity matrix. Scalar n is the number of joints for the manipulator. Vector $\boldsymbol{\tau}_0$ is an arbitrary joint space input vector and $\boldsymbol{\tau}_N$ corresponds to a null task space input vector, which means any joint space input $\boldsymbol{\tau}_0$ has no effect on the task space control vector. An appropriate $\boldsymbol{\tau}_0$ should be selected to guarantee the stability in joint space. In this section, $\boldsymbol{\tau}_0$ is selected as:

$$\boldsymbol{\tau}_0 = -k_\xi\hat{\boldsymbol{M}}\boldsymbol{\xi}. \tag{4.12}$$

where k_ξ is a positive coefficient. Substituting Eq. (4.12) into Eq. (4.11), we have

$$\boldsymbol{\tau}_N = k_\xi \boldsymbol{J}^{\mathrm{T}} \boldsymbol{\Lambda} \dot{\boldsymbol{x}} - k_\xi \hat{\boldsymbol{M}} \boldsymbol{\xi}. \tag{4.13}$$

Finally, the control input in joint space is expressed as:

$$\boldsymbol{\tau} = \boldsymbol{J}^{\mathrm{T}} \boldsymbol{F}_{ee} + k_\xi \boldsymbol{J}^{\mathrm{T}} \boldsymbol{\Lambda} \dot{\boldsymbol{x}} - k_\xi \hat{\boldsymbol{M}} \boldsymbol{\xi}. \tag{4.14}$$

The stability of using Eq. (4.14) is analyzed in 4.2.3. To eliminate the effect of uncertainty term $\boldsymbol{\Delta}$, a same neural network as shown in section 3.2.2 is adopted in this part. The actual output of neural network is expressed as:

$$\hat{\boldsymbol{f}}_n = \hat{\boldsymbol{W}}^{\mathrm{T}} \boldsymbol{U}. \tag{4.15}$$

in which $\hat{\boldsymbol{W}}$ is the actual weights of the neural network and $\boldsymbol{U}$ is the radial basis vector. Therefore, the uncertainty term can be expressed as:

$$\boldsymbol{\Delta} = \boldsymbol{f}(\boldsymbol{W}^*) + \boldsymbol{\mu}_0. \tag{4.16}$$

where $\boldsymbol{\mu}_0$ is bounded and satisfies $||\boldsymbol{\mu}_0|| \leq \epsilon_0$

4.2.2 Adaptive law design and stability analysis in task space

Considering the neural network compensator, the control input in task space can be expressed as:

$$\boldsymbol{F}_E = \boldsymbol{\Lambda}(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_v \dot{\boldsymbol{e}} + \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{K}_i \int \boldsymbol{e}) + \boldsymbol{\Lambda}(\boldsymbol{J} \hat{\boldsymbol{M}}^{-1} \boldsymbol{H} - \dot{\boldsymbol{J}} \boldsymbol{\xi} + \boldsymbol{f}_n). \tag{4.17}$$

Then substituting Eq. (4.16) and (4.17) into Eq. (4.6) yields:

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_v \dot{\boldsymbol{e}} + \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{K}_i \int \boldsymbol{e} = -\widetilde{\boldsymbol{W}}^{\mathrm{T}} \boldsymbol{U} + \boldsymbol{\mu}_0. \tag{4.18}$$

where $\widetilde{\boldsymbol{W}}^{\mathrm{T}} = \hat{\boldsymbol{W}}^{\mathrm{T}} - \boldsymbol{W}^{*\mathrm{T}}$. Define the error state vector:

$$\boldsymbol{s} = \begin{bmatrix} \int \boldsymbol{e} \\ \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix}. \tag{4.19}$$

Then Eq. (4.18) can be expressed in term of state vector as:

$$\dot{\boldsymbol{s}} = \boldsymbol{A}\dot{\boldsymbol{s}} + \boldsymbol{F}(-\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{U} + \boldsymbol{\mu}_0). \tag{4.20}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K}_i & -\boldsymbol{K}_p & -\boldsymbol{K}_v \end{bmatrix}, \ \boldsymbol{F} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}. \tag{4.21}$$

Define the Lyapunov function:

$$V = \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s} + \frac{1}{2\gamma}||\widetilde{\boldsymbol{W}}||^2. \tag{4.22}$$

where $\gamma > 0$. Matrix $\boldsymbol{P}$ is symmetric positive definite, and satisfies the Lyapunov equation:

$$\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{P} = -\boldsymbol{Q}. \tag{4.23}$$

in which $\boldsymbol{Q} \geq 0$.

The time derivative of Lyapunov function is:

$$\begin{aligned} \dot{V} &= \frac{1}{2}[\boldsymbol{s}^{\mathrm{T}}\boldsymbol{P}\dot{\boldsymbol{s}} + \dot{\boldsymbol{s}}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s}] + \frac{1}{\gamma}tr(\dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}}) \\ &= \frac{1}{2}\{\boldsymbol{s}^{\mathrm{T}}\boldsymbol{P}[\boldsymbol{A}\boldsymbol{s} + \boldsymbol{F}(-\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{U} + \boldsymbol{\mu}_0)] + [\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}} + (-\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{U} + \boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}]\boldsymbol{P}\boldsymbol{s}\} \\ &\quad + \frac{1}{\gamma}tr(\dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}}) \\ &= \frac{1}{2}[\boldsymbol{s}^{\mathrm{T}}(\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{P})\boldsymbol{s}] + [-\widetilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{U} + \boldsymbol{\mu}_0]^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s} + \frac{1}{\gamma}tr(\dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}}) \\ &= -\frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{s} - \boldsymbol{U}^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s} + \boldsymbol{\mu}_0^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s} + \frac{1}{\gamma}tr(\dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}}). \end{aligned} \tag{4.24}$$

Because

$$\boldsymbol{U}^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s} = tr(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s}\boldsymbol{U}^{\mathrm{T}}\widetilde{\boldsymbol{W}}). \tag{4.25}$$

then Eq. (4.24) can be written as:

$$\begin{aligned}\dot{V} =& -\frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{s} + \boldsymbol{\mu}_0^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s} \\ &+ \frac{1}{\gamma}tr(-\gamma\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s}\boldsymbol{U}^{\mathrm{T}}\widetilde{\boldsymbol{W}} + \dot{\widetilde{\boldsymbol{W}}}^{\mathrm{T}}\widetilde{\boldsymbol{W}})\end{aligned}. \tag{4.26}$$

The adaptive law is selected as:

$$\dot{\hat{\boldsymbol{W}}} = \gamma\boldsymbol{U}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{F}. \tag{4.27}$$

Therefore,

$$\dot{V} = -\frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{s} + \boldsymbol{\mu}_0^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{s}. \tag{4.28}$$

As noted earlier, $||\boldsymbol{\mu}_0|| \leq \epsilon_0$ and $||\boldsymbol{F}|| = 1$. If $\lambda_{min}(\boldsymbol{Q})$ is the minimum eigenvalue of $\boldsymbol{Q}$ and $\lambda_{max}(\boldsymbol{P})$ is the maximum eigenvalue of $\boldsymbol{P}$, then

$$\begin{aligned}\dot{V} &\leq -\frac{1}{2}\lambda_{min}(\boldsymbol{Q})||\boldsymbol{s}||^2 + \epsilon_0\lambda_{max}(\boldsymbol{P})||\boldsymbol{s}|| \\ &= -\frac{1}{2}||\boldsymbol{s}||[\lambda_{min}(\boldsymbol{Q})||\boldsymbol{s}|| - 2\epsilon_0\lambda_{max}(\boldsymbol{P})]\end{aligned} \tag{4.29}$$

To make $\dot{V} \leq 0$, $\lambda_{min}(\boldsymbol{Q})$ should satisfy the condition:

$$\lambda_{min}(\boldsymbol{Q}) \geq \frac{2\epsilon_0\lambda_{max}(\boldsymbol{P})}{||\boldsymbol{s}||}. \tag{4.30}$$

Therefore, the error states and Lyapunov function V are both bounded, which means the controller designed in task space is stable.

4.2.3 Stability Analysis in Joint Space

The mapping relationship of Eq. (4.14) is used to implement the control input of Eq. (4.17). Considering the redundancy characteristic, the stability in joint space should be satisfied as well. Expanding Eq. (4.14), we have the control input in joint space as:

$$\begin{aligned}\boldsymbol{\tau} = \boldsymbol{J}^{\mathrm{T}}[\boldsymbol{\Lambda}(\ddot{\boldsymbol{x}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_i\int e) \\ + \boldsymbol{\Lambda}(\boldsymbol{J}\hat{M}^{-1}\boldsymbol{H} - \dot{\boldsymbol{J}}\boldsymbol{\xi} + \boldsymbol{f}_n)] + \boldsymbol{d}\boldsymbol{\xi}.\end{aligned} \tag{4.31}$$

where

$$\boldsymbol{d} = -[(\boldsymbol{K}_v - k_\xi\boldsymbol{I})\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{J} + k_\xi\hat{\boldsymbol{M}}]. \tag{4.32}$$

The UVMS can be considered as a conservative system subjected to the dissipative forces due to the velocity damping term $-\boldsymbol{K}_v\dot{\boldsymbol{x}}$ in Eq. (4.17). The forces are expressed as:

$$\boldsymbol{f}_{dis} = \boldsymbol{d}\boldsymbol{\xi}. \tag{4.33}$$

If $min(\boldsymbol{K}_v) \geq k_\xi$, then $\boldsymbol{d}$ is a negative definite matrix. Lyapunov stability analysis shows that

$$\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{d}\boldsymbol{\xi} < 0. \tag{4.34}$$

which means the system is asymptotically stable in joint space.

4.2.4 Simulation results and discussion

In this section, the performance of the proposed task space controller is analyzed by applying it to track a desired trajectory.

The structure parameters of the UVMS and the hydrodynamic parameters are the same as in Table 3.1 and 3.2. The initial configuration of the UVMS is $\boldsymbol{\zeta} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 90\ -30]$m, deg. The desired trajectory of end-effector is to track a triangular with three vertices $[0\ 0.7464\ -0.9]$m, $[0\ 0.7464\ -1]$m, and $[0\ 0.6464\ -1]$m. A fifth-order polynomial trajectory generation approach is used for each segment and the time period is 10s. The sampling frequency is 1000Hz.

The proportion-integration-differentiation parameters are selected as:

$$\begin{aligned}\boldsymbol{K}_p &= diag\{[10\ 10\ 10]\} \\ \boldsymbol{K}_v &= diag\{[5\ 5\ 5]\} \quad . \\ \boldsymbol{K}_i &= diag\{[5\ 5\ 5]\}\end{aligned} \tag{4.35}$$

For the mapping modification, $k_\xi = 20$. In the neural network, $\gamma = 20$, and $\boldsymbol{Q} = 100 \times \boldsymbol{I}_9$. The initial value of $\hat{\boldsymbol{W}}$ is set to be $0.001 \times \boldsymbol{I}_{5\times3}$. The corresponding Gaussian functions are shown in Fig. 4.3.

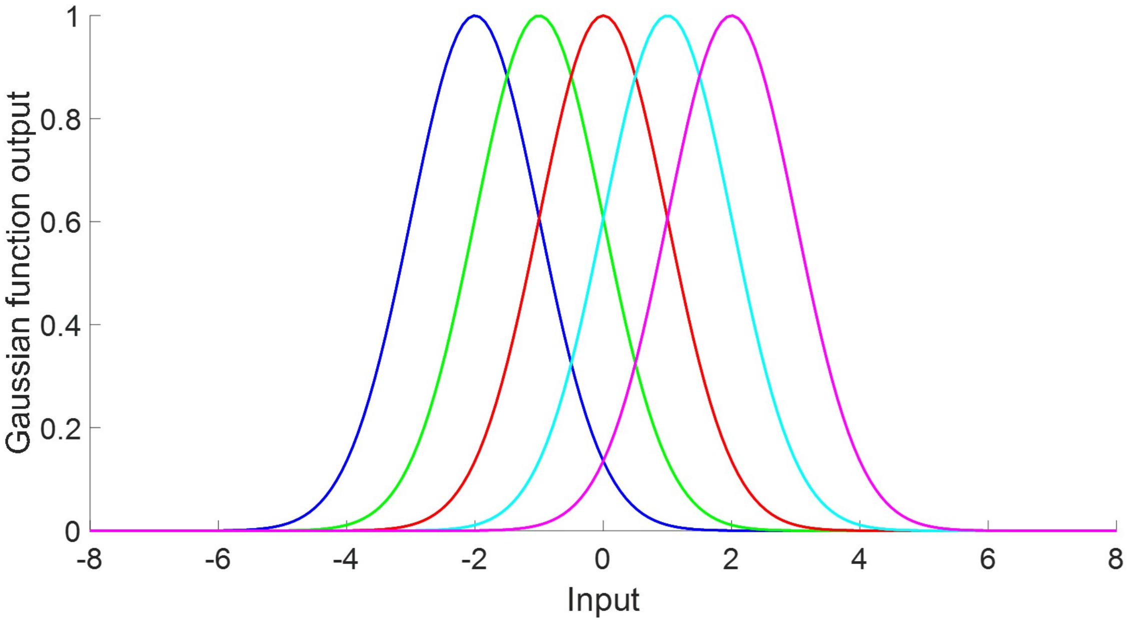


Figure 4.3: Neural network functions

Vehicle position and attitude responses are shown in Fig. 4.4. Despite the trajectory planning is in task space, the motion of vehicle is smooth with small linear and angular displacement. The vehicle velocities are shown in Fig. 4.5. It can be seen that the linear and angular velocities approach to zero as the end-effector approach to the destination, which means the UVMS stays stable in joint space. The manipulator joint angles and velocities are shown in Fig. 4.6. Compared to vehicle, the manipulator moves in a larger range, which shows that the trajectory of end-effector is realized mainly by the manipulator. Joint velocity response shows the manipulator is stable as well in joint space.

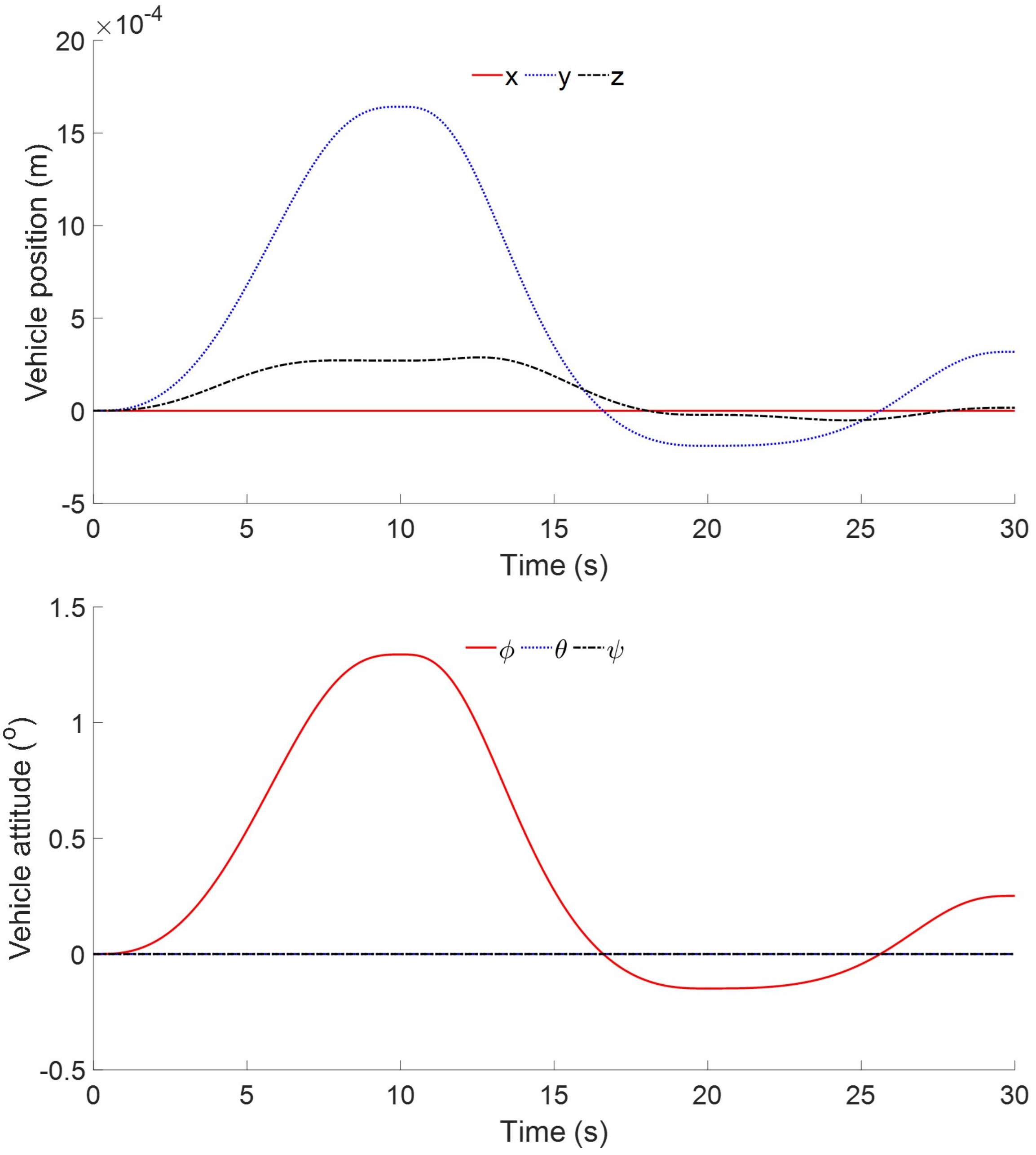


Figure 4.4: Vehicle displacement

In task space,the trajectory of end-effector is shown in Fig. 4.7. The tracking errors are shown in Fig. 4.8. It can be seen that the proposed controller is capable of tracking desired trajectory with high accuracy. There are some vibrations at the beginning because the neural

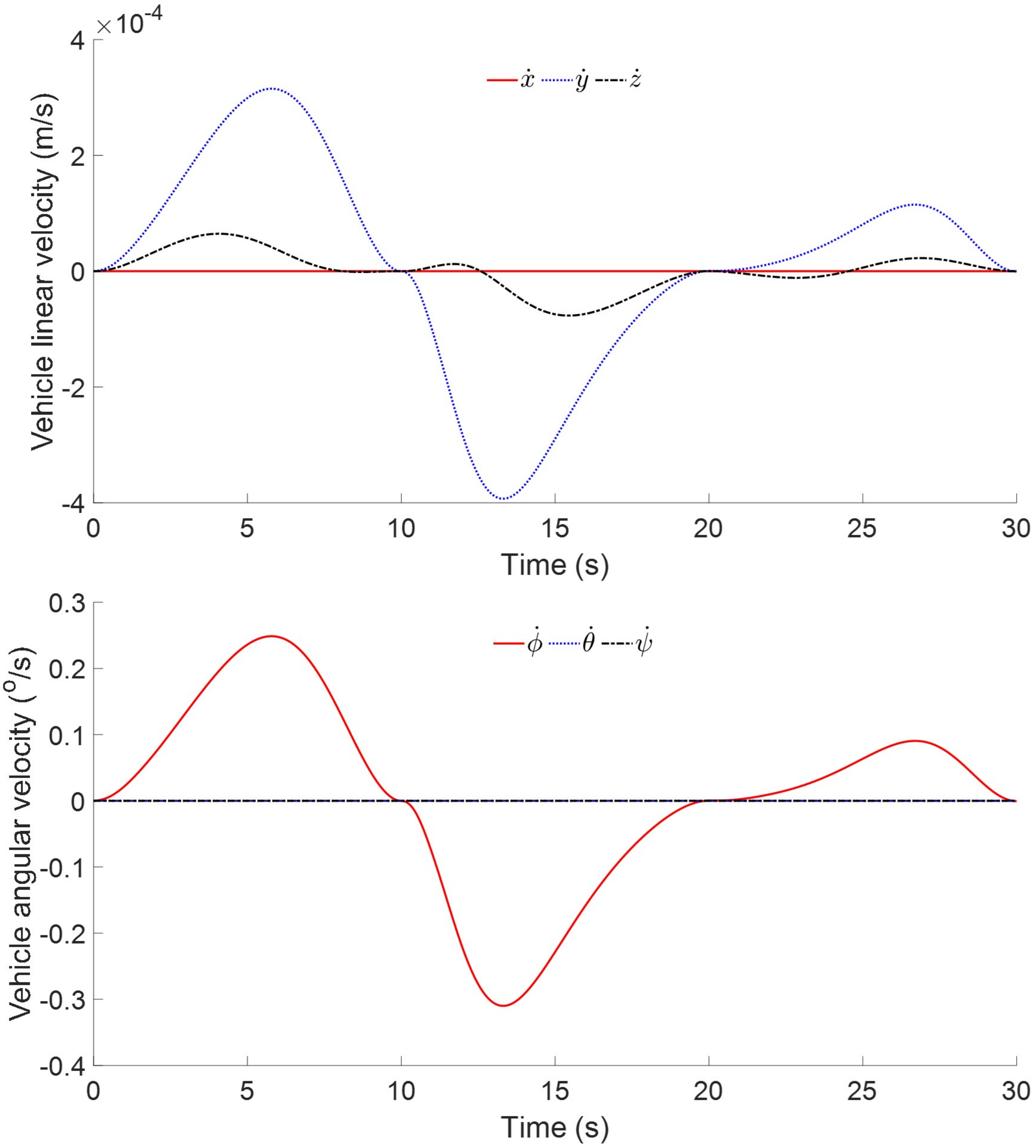


Figure 4.5: Vehicle velocity

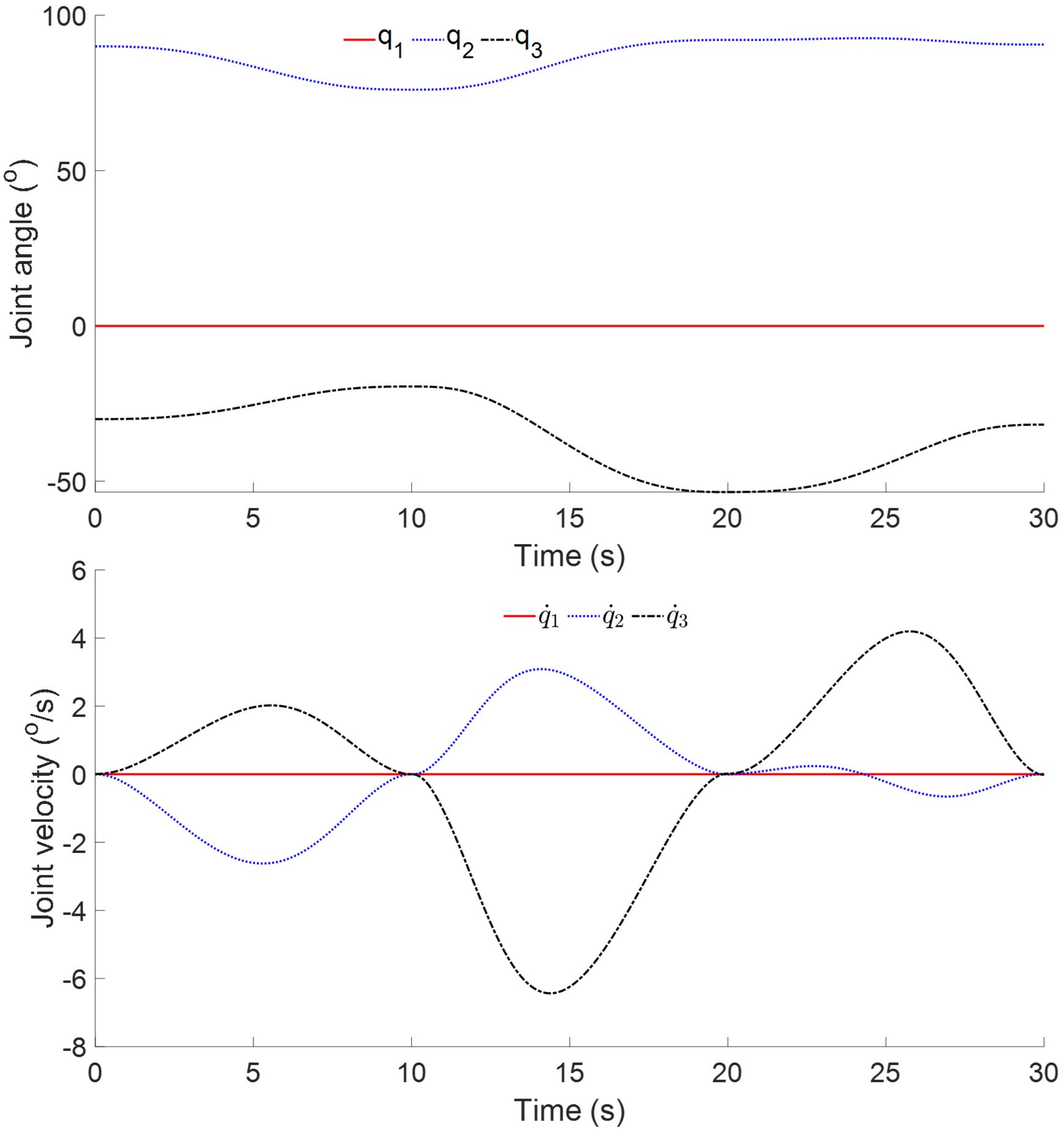


Figure 4.6: Joint response

network should adjust its parameters adaptively. After that, the tracking errors are smooth, showing that the system is stable in task space.

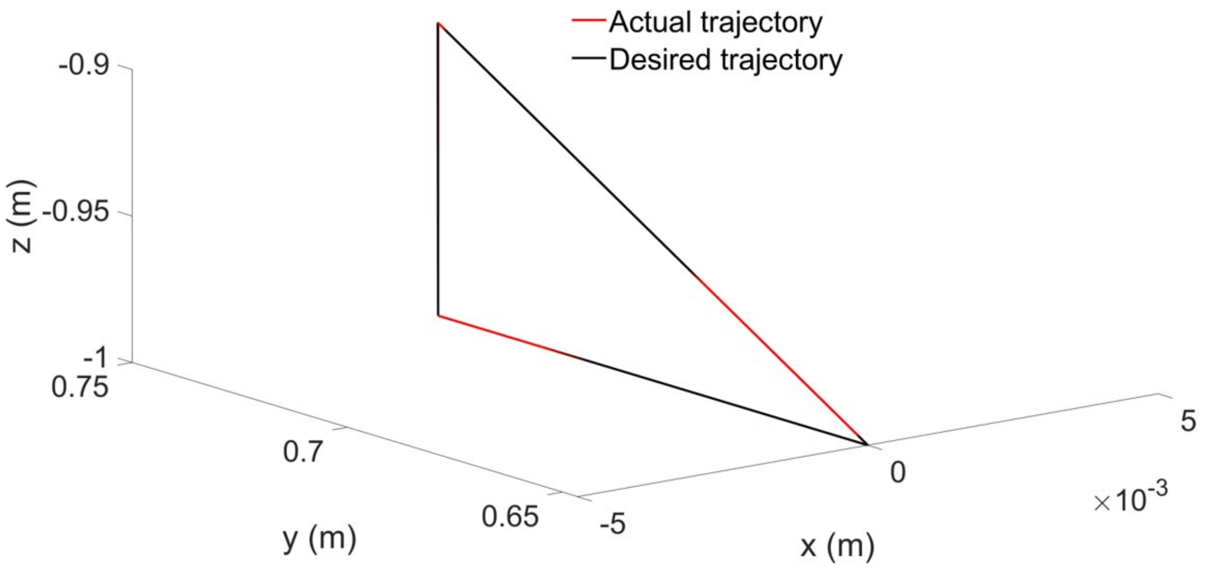


Figure 4.7: End-effector trajectory

4.2.5 Summary of task space trajectory tracking control

In this section, the mapping relationship between joint space and task space is modified to guarantee the stability in both joint and task space. The simulation results showed its effectiveness when applied to complete a trajectory tracking task.

4.3 Task space force control

Contact between the manipulator and the environment is usually difficult to model. In this section, a force control term is added to Eq. (4.31) to finish a force control task.

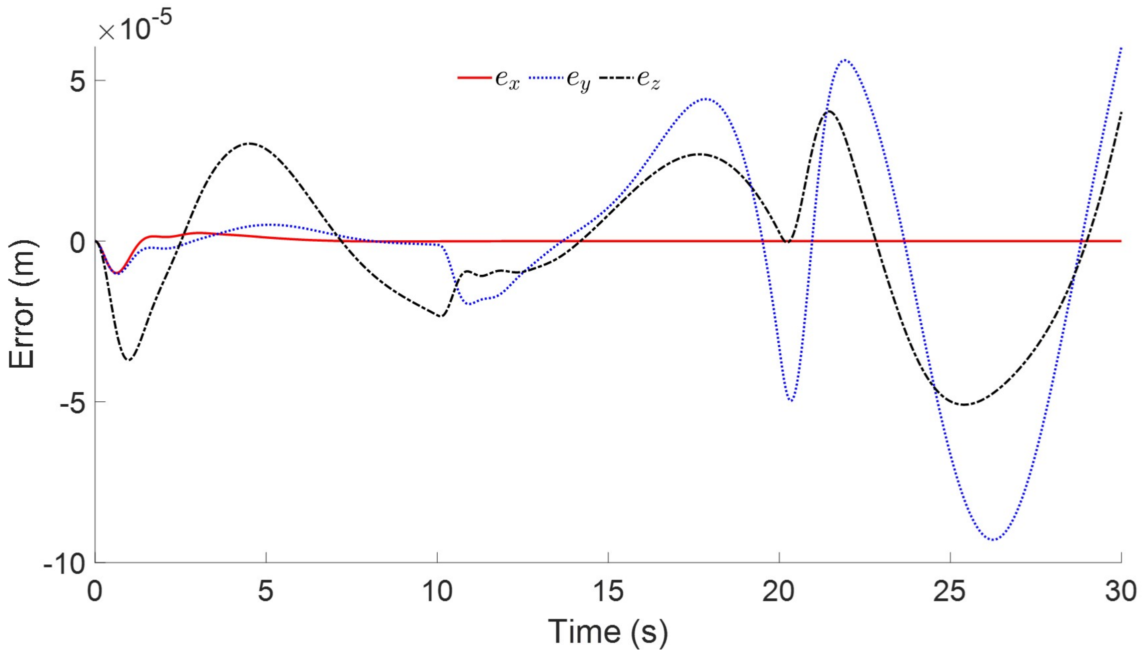


Figure 4.8: Tracking error

4.3.1 Contact with the environment

In this part, we will resort to the simple model constituted by a frictionless and elastically compliant plane. The force at the end-effector is then related to the deformation of the environment by the following simplified model [76]:

$$\boldsymbol{f}_e = \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}_c). \tag{4.36}$$

where $\boldsymbol{x}$ is the position of the end effector expressed in inertial frame. A constant vector $\boldsymbol{x}_c$ represents the position of the unperturbed environment expressed in the inertial frame, and

$$\boldsymbol{K} = k\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}. \tag{4.37}$$

with $k > 0$ is the stiffness of the environment. Vector $\boldsymbol{n}$ is normal to the plane. A planar view of the contact model is shown in Fig. 4.9. The range space of $\boldsymbol{K}$, $\mathbb{R}(\boldsymbol{K})$, is constituted by all the vectors parallel to $\boldsymbol{n}$, which are normal to the plane. Similarly, $\mathbb{R}(\boldsymbol{I}_{3\times 3} - \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}})$ spans the space of the vectors parallel to the contact plane.

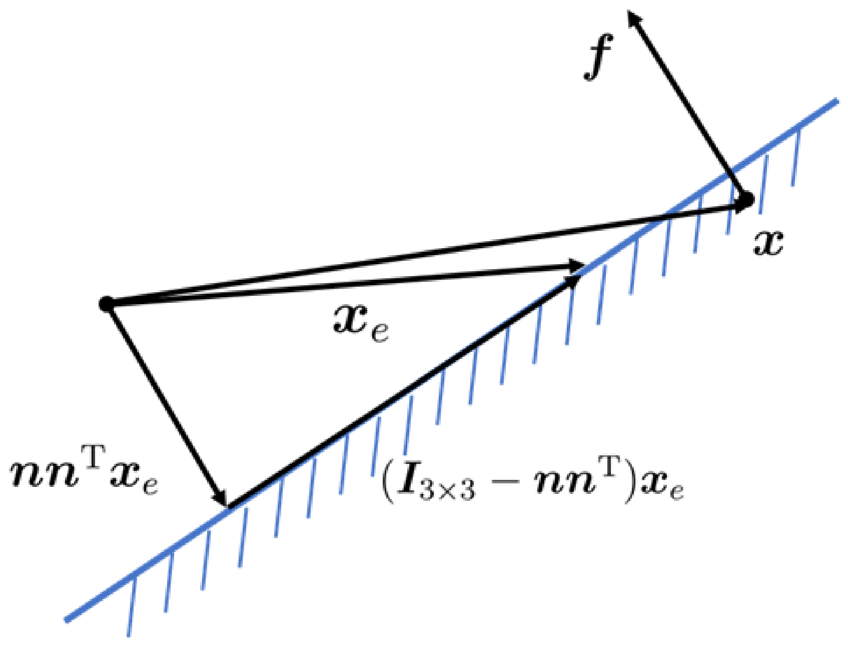

$\boldsymbol{f}$
$\boldsymbol{x}$
$\boldsymbol{x}_e$
$\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{x}_e$
$(\boldsymbol{I}_{3\times 3} - \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}})\boldsymbol{x}_e$

Figure 4.9: Planar view of the chosen contact model

4.3.2 Force controller design in task space

In order to implement the force control in task space, a simple force control term is given as:

$$\boldsymbol{f}_{fc} = -\boldsymbol{f}_{te} + K_{fp}\tilde{\boldsymbol{f}} + K_{fv}\dot{\boldsymbol{f}}_e + K_{fi}\int_0^t \tilde{\boldsymbol{f}} dt \tag{4.38}$$

where K_{fp}, K_{fv}, and K_{fi} are scalar positive gains. The desired force is $\boldsymbol{f}_d$ and the external force is $\boldsymbol{f}_{te}$. The force tracking error is written as $\tilde{\boldsymbol{f}} = \boldsymbol{f}_d - \boldsymbol{f}_e$.

Adding Eq. (4.38) into (4.31), we have the force control expression in joint space as:

$$\begin{aligned}\boldsymbol{\tau} = \boldsymbol{J}^{\mathrm{T}}[\boldsymbol{\Lambda}(\ddot{\boldsymbol{x}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_i\int \boldsymbol{e}) + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{f}_{fc} \\ + \boldsymbol{\Lambda}(\boldsymbol{J}\hat{\boldsymbol{M}}^{-1}\boldsymbol{H} - \dot{\boldsymbol{J}}\boldsymbol{\xi} + \boldsymbol{f}_n)] + \boldsymbol{d}\boldsymbol{\xi}.\end{aligned} \tag{4.39}$$

4.3.3 Force control task description

In this part, simulation results are presented to demonstrate the effectiveness of the proposed force control strategy. The initial configuration is:

$$\boldsymbol{\zeta} = [0\,0\,0\,0\,0\,0\,0\,120\ \ -30]^{\mathrm{T}}\ \mathrm{m},\ \mathrm{deg}. \tag{4.40}$$

which corresponds to the end-effector position $\boldsymbol{x}_e = [0\ 0.7464\ \ -0.5]^{\mathrm{T}}\mathrm{m}$. The position of the contact plane is set to be $z = -0.8\mathrm{m}$. The stiffness coefficient of the environment is $k = 10^4$N/m.

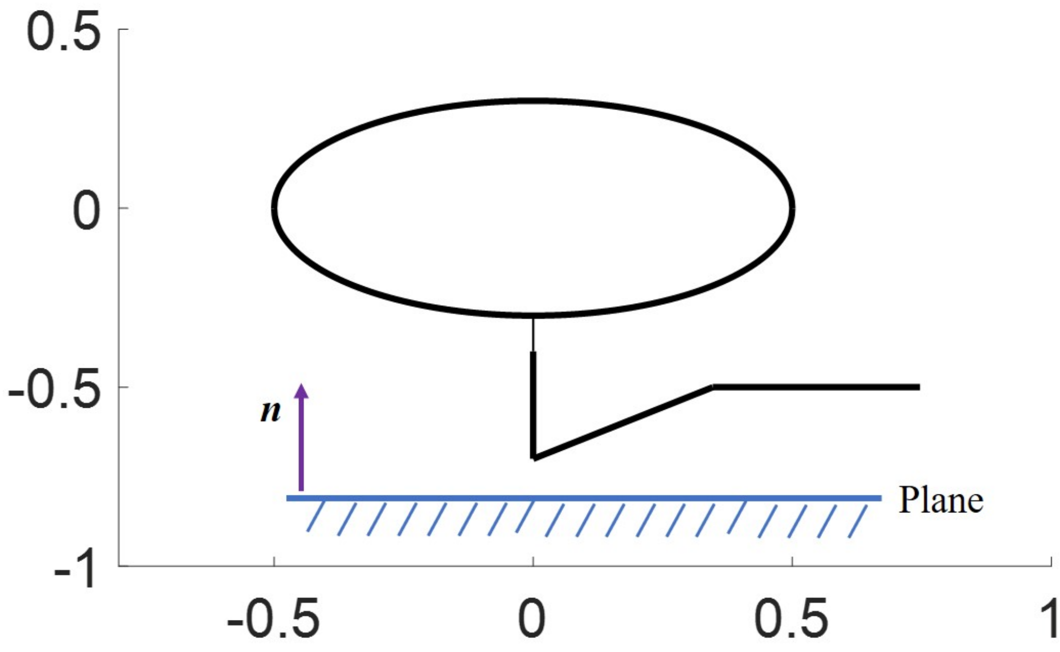


Figure 4.10: Initial configuration for force control

The task of the end-effector is to move directly to the plane and apply a constant force on this plane. The end-effector moves towards the plane in 10 sec. Once the contact is detected, the force control strategy is used to control the motion of the end-effector. The desired force is $\boldsymbol{f}_d = [0\ 0\ 200]^{\mathrm{T}}$N. The force control parameters are $K_{fp} = 0.01$, $K_{fv} = 0.005$, and $K_{fi} = 0.005$, respectively.

4.3.4 Simulation results and discussion for force control

The vehicle and manipulator response are shown in Fig. 4.11 - 4.13. At $t = 10$s, the end-effector contacts with a surface and the force control is activated. It can be seen that the velocity of vehicle and manipulator approaches to zero, which means the system is stable. Fig. 4.14 - 4.16 shows the position of end-effector. In the x direction, there are small fluctuations due to the coupling in the neural network. The contact between the surface and end-effector causes the motion in y direction. However, the end-effector recovers quickly and tends to be stable. Fig. 4.16 shows the position of end-effector in z direction and Fig. 4.17 shows the force of end-effector. It can be seen that the proposed control approach is able to complete force tasks effectively.

4.3.5 Summary of force control

In this section, a force control term is added to the proposed control model in task space trajectory tracking control. A constant force task is given and the simulation results confirm the effectiveness of the proposed force control strategy.

4.4 Chapter summary

In this chapter, a task space trajectory/force control strategy is proposed. A mapping relationship from task space to joint space using a Jacobian matrix can not guarantee system stability. Considering this issue, the mapping relationship is modified and system stability is analyzed using Lyapunov function. The simulation results show its effectiveness when applied to trajectory tracking tasks. A PID type force term is added to the controller designed for task space trajectory tracking. The end-effector is able to apply a constant force on a surface and stays stable for both the vehicle and manipulator.

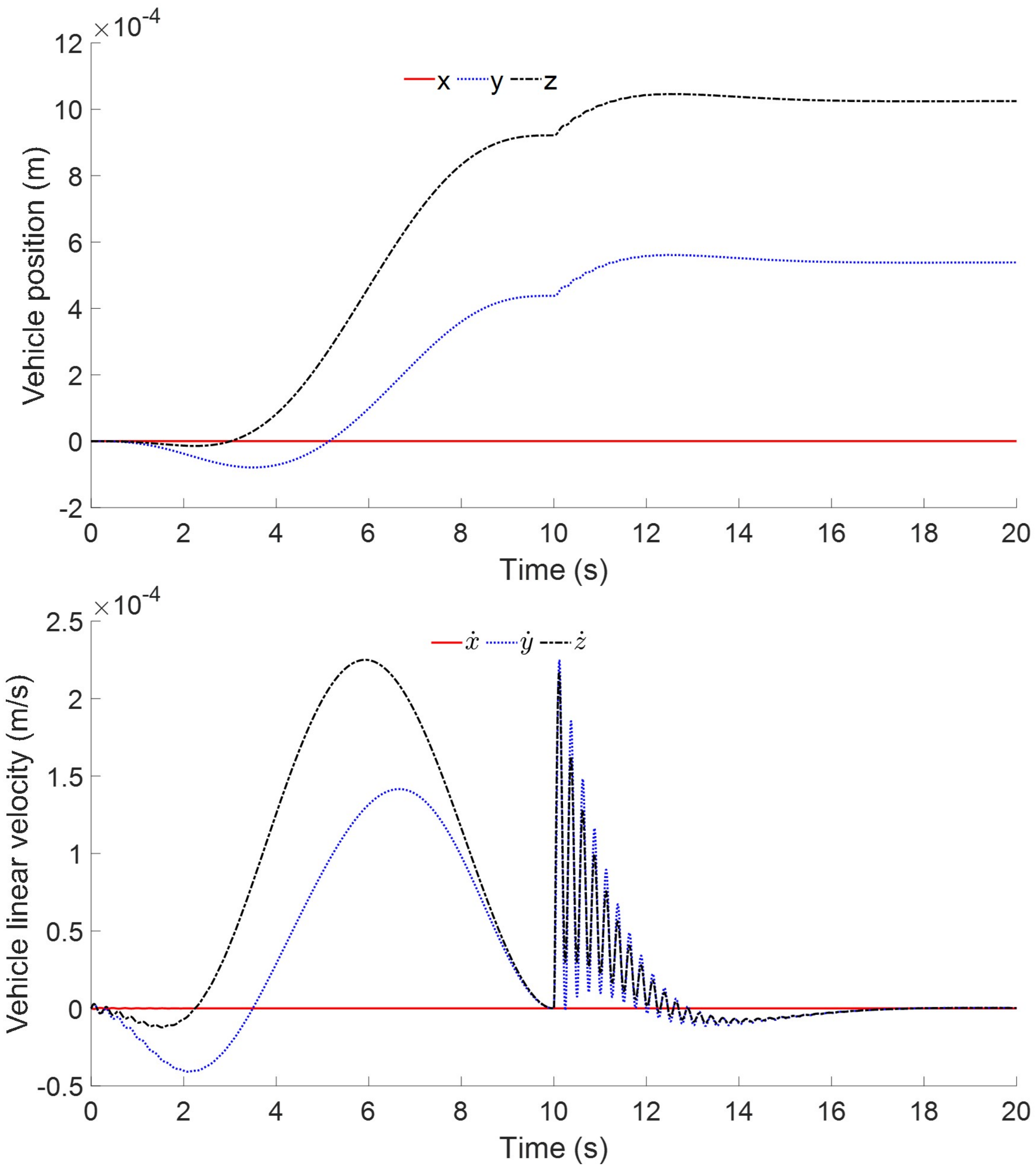


Figure 4.11: Vehicle position and linear velocity for force control

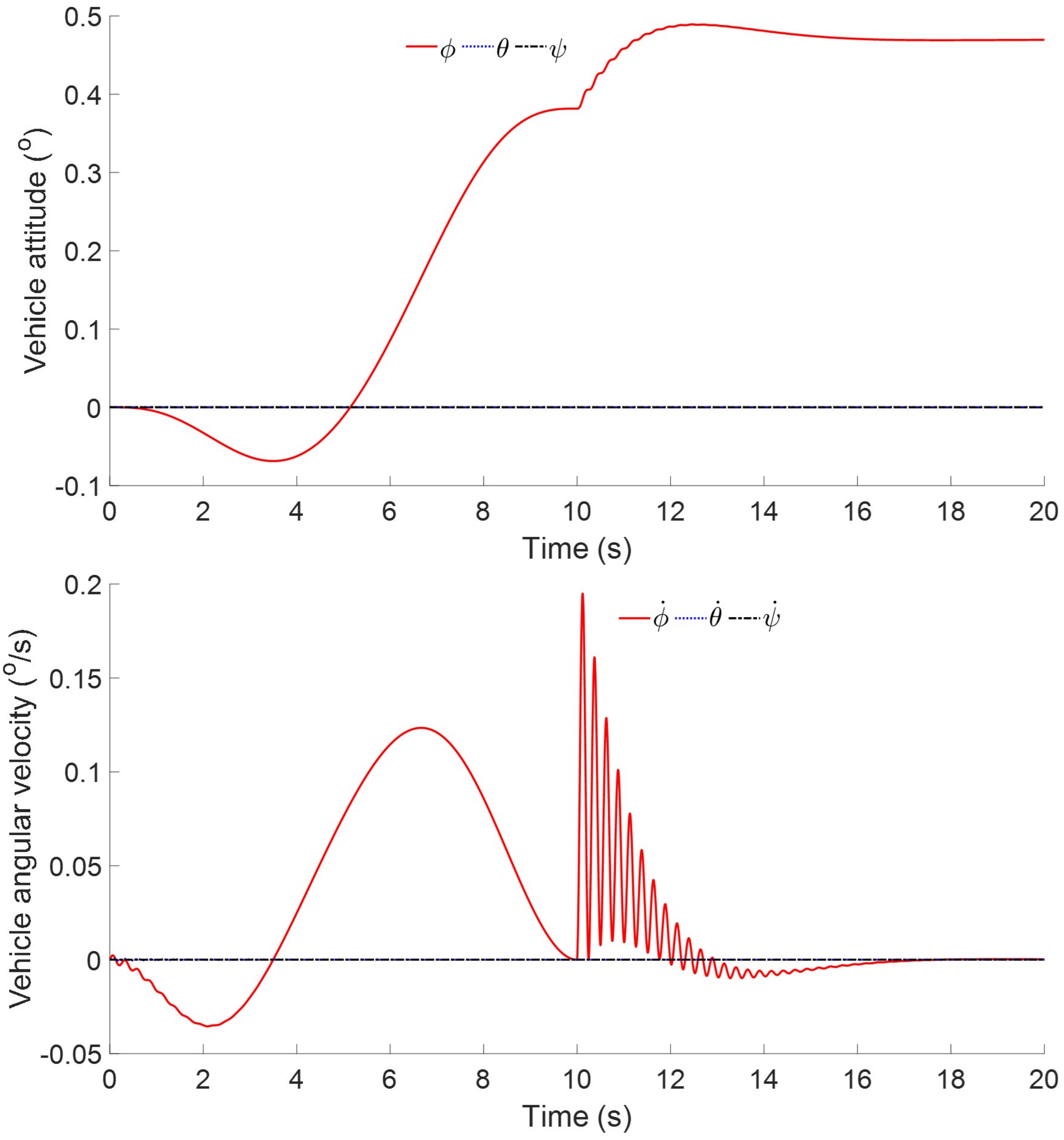


Figure 4.12: Vehicle attitude and angular velocity for force control

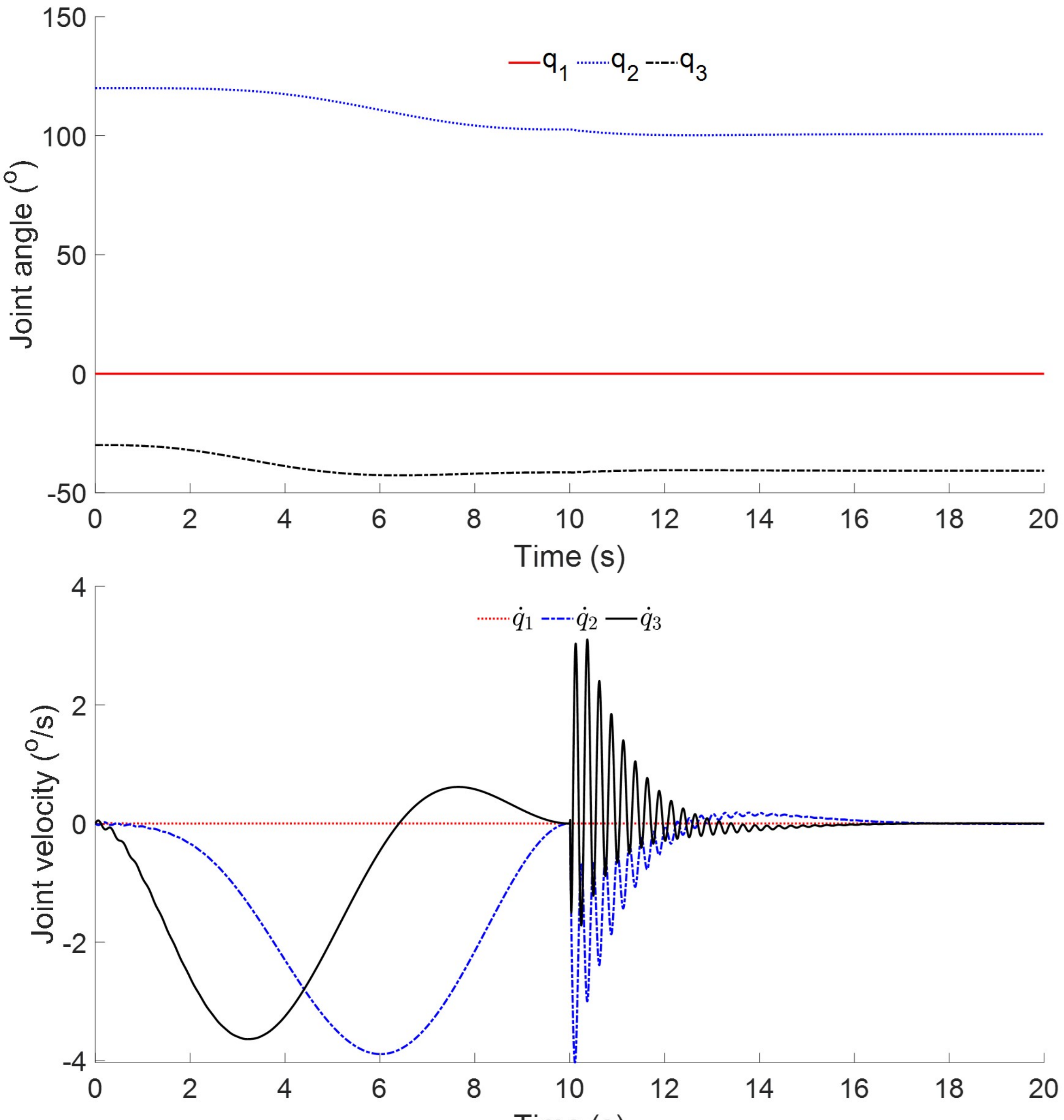


Figure 4.13: Joint response for force control

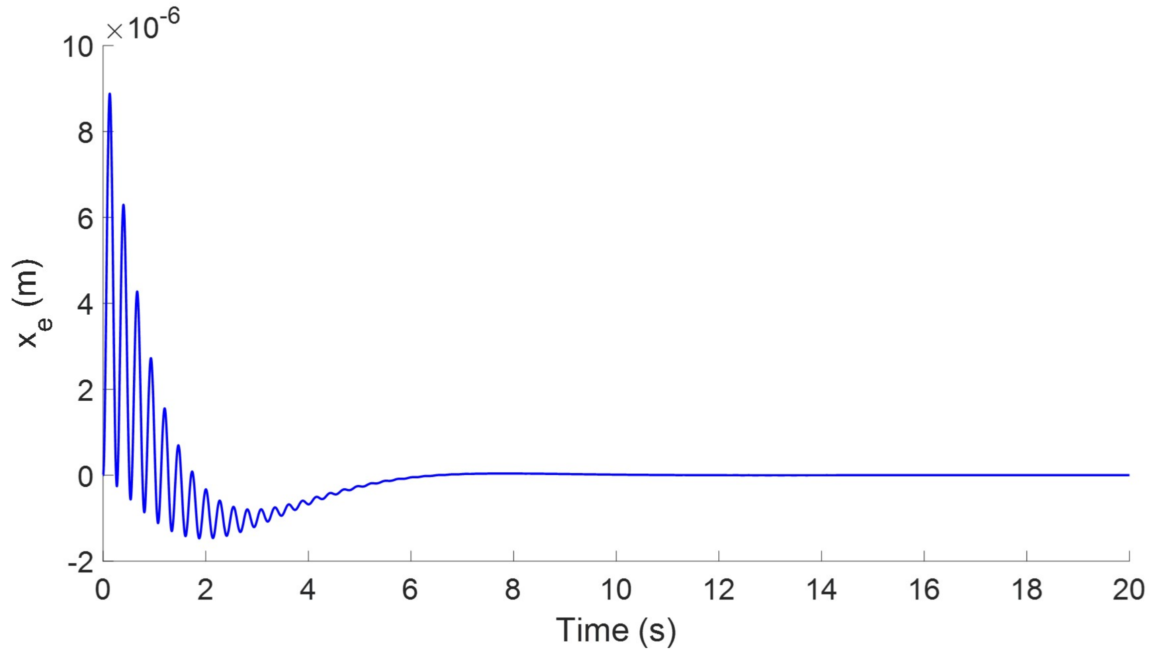


Figure 4.14: End-effector position in x idrection

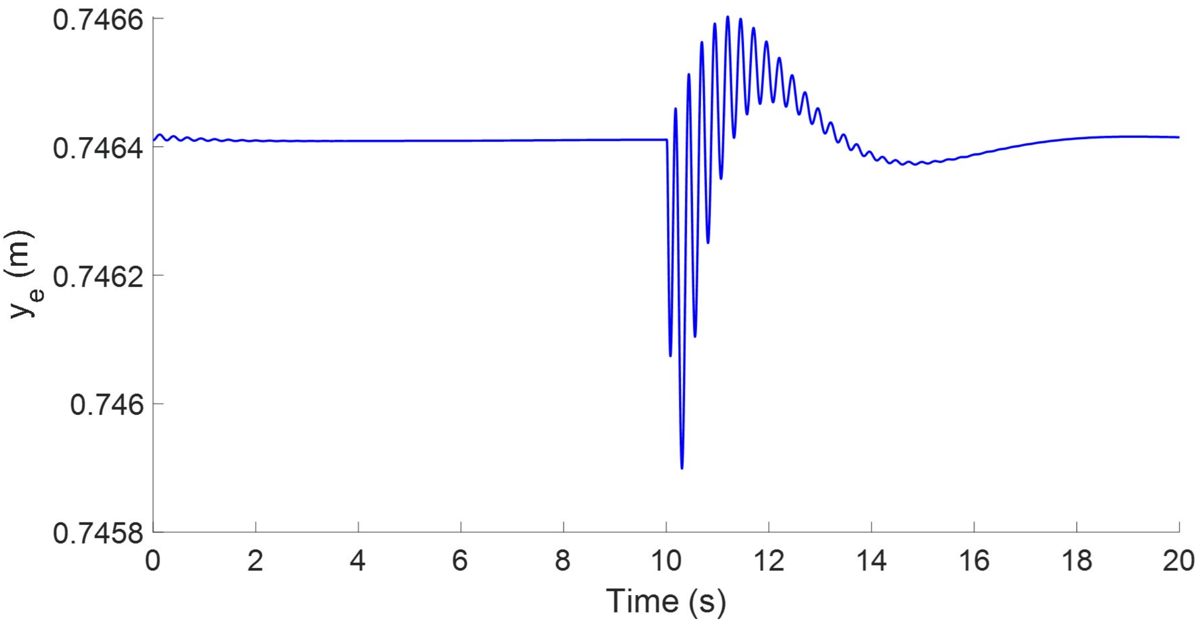


Figure 4.15: End-effector position in y idrection

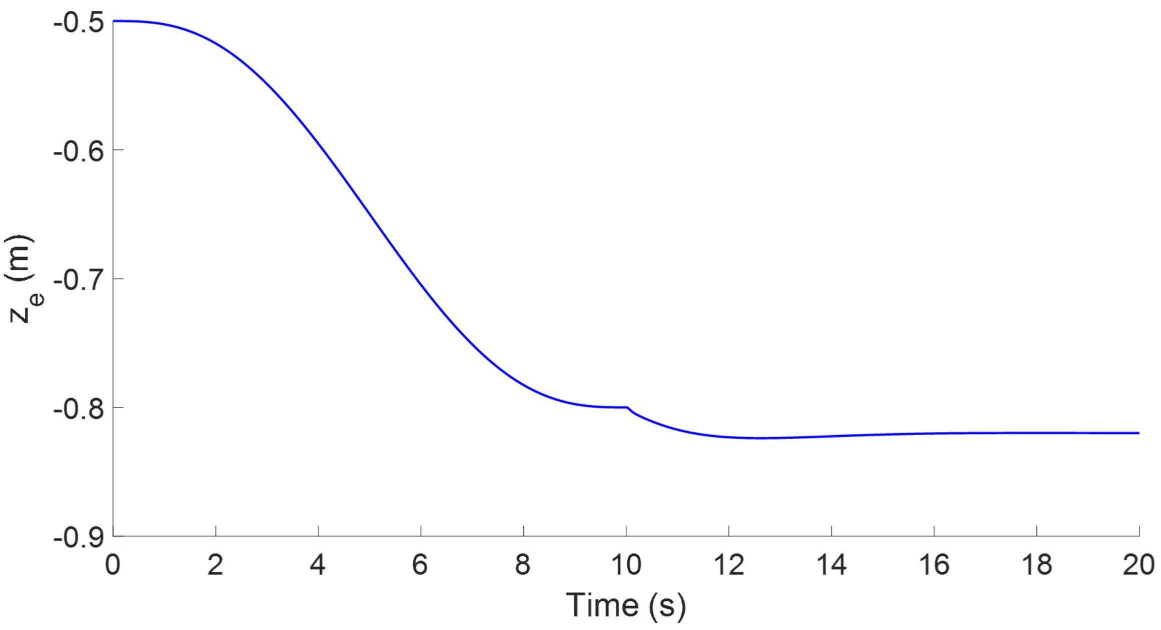


Figure 4.16: End-effector position in z idrection

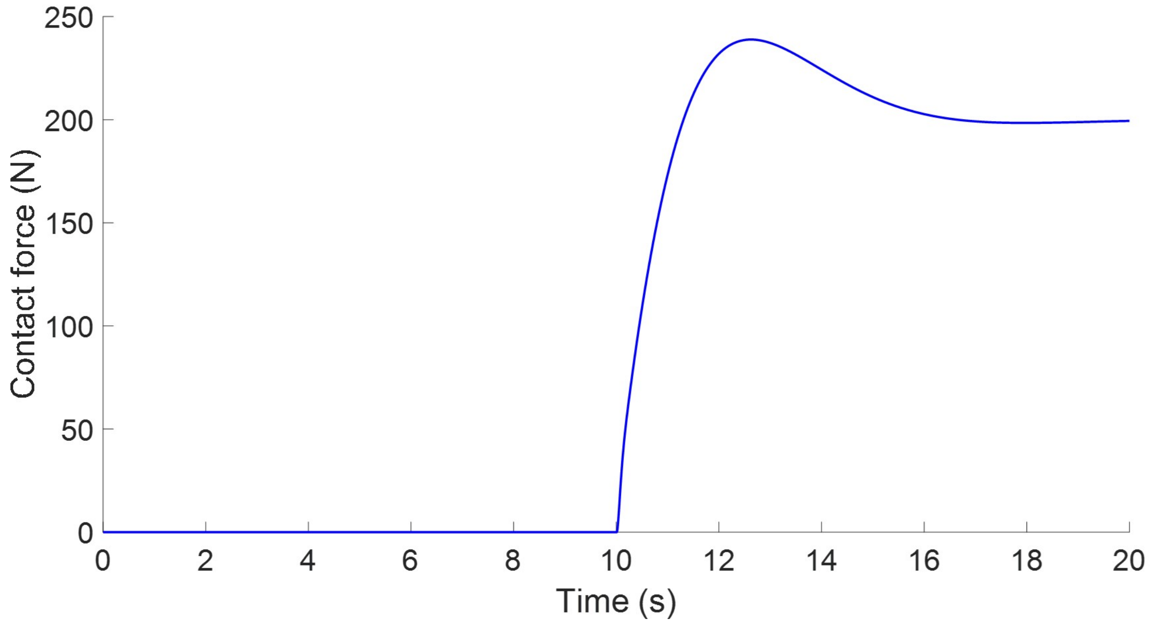


Figure 4.17: End-effector force

Chapter 5

Trajectory tracking control experiment using neural network

In the joint space and task space trajectory tracking control, a neural network is adopted to eliminate system uncertainties. Simulation results show its effectiveness when applied to different tasks. In this chapter, two control strategies using an adaptive neural network, torque control and position control, are tested on a four-link robot arm. Practical concerns are given and the performance of the controllers are analyzed.

5.1 Servo characteristics

In the following experiments, servos are used as actuators. Therefore, it is necessary to introduce the characteristics of the servo. In this work, DYNAMIXEL AX-12A servos are used. The output range is $[0,\ 300]$ degrees, and the input range is $[0,\ 1023]$ counts, which yields an approximate position resolution of $0.293°$/count. The stall torque at 12V, 1.5A is 1.5N·m. The front view is shown in Fig. 5.1.

This type of servo is intelligent, which allows us to read the current position and velocity information and write our desired position. If a present moving speed value is in the range of $0 - 1023$, it means that the motor rotates to the counter-clockwise (CCW) direction. If a value is in the range of $1024 - 2047$, it means that the motor rotates to the clockwise (CW) direction. The unit of speed value is 0.111 rpm. Torque limit can be changed as well. The output torque range is $[0,\ 1.5]$ N·m, and the input range is $[0,\ 1023]$ counts, which yields an approximate torque resolution of 1.46×10^{-3} N·m/count.

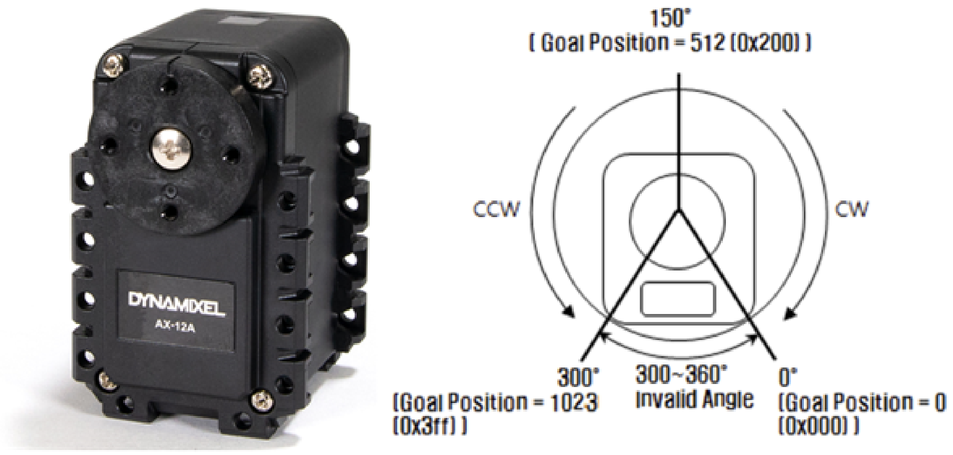

150°
[Goal Position = 512 (0x200)]
CCW
CW
300°
[Goal Position = 1023 (0x3ff)]
300~360°
Invalid Angle
0°
[Goal Position = 0 (0x000)]
DYNAMIXEL
AX-12A

Figure 5.1: The front view of AX-12A

Compliance slope exists in each direction of CW/CCW and sets the level of torque near the goal position. Compliance is to set the control flexibility of the motor. Fig. 5.2 shows the relationship between output torque and position of the motor.

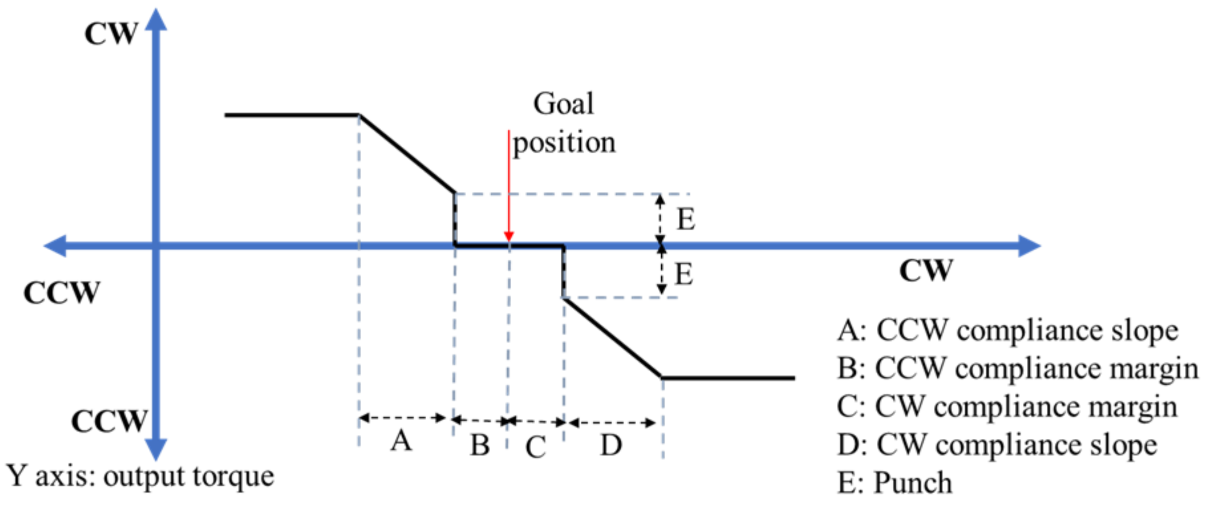


Figure 5.2: Compliance slope of AX-12A

5.2 Robot design

In this experiment, a four-joint robot is designed. The structure of the robot manipulator is shown in Fig. 5.3. The simplified robot structure with assigned frames is shown in 5.4.

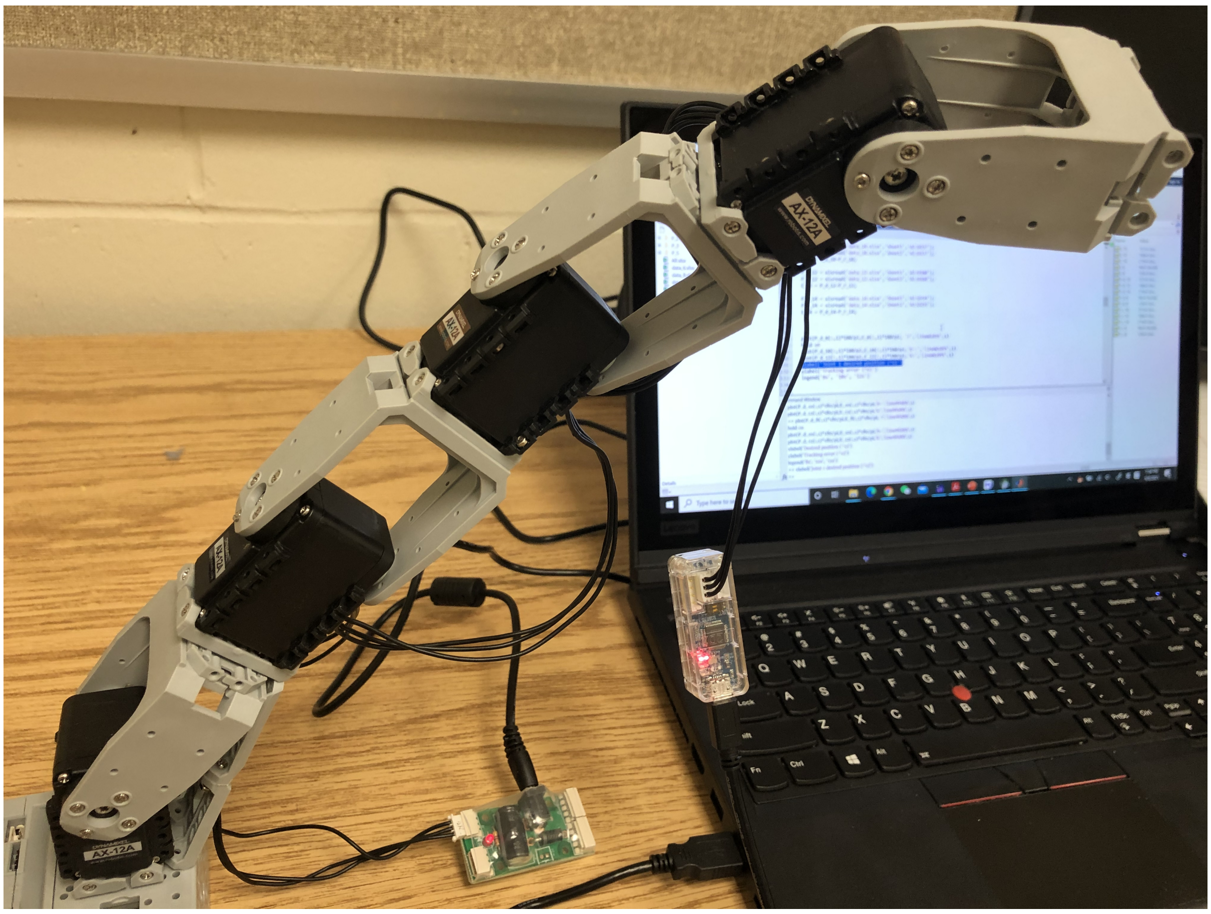


Figure 5.3: Assembled robot manipulator

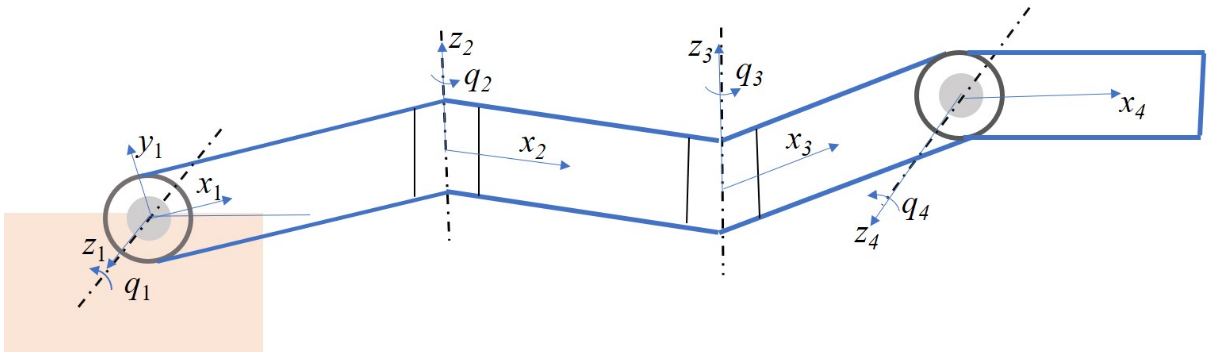
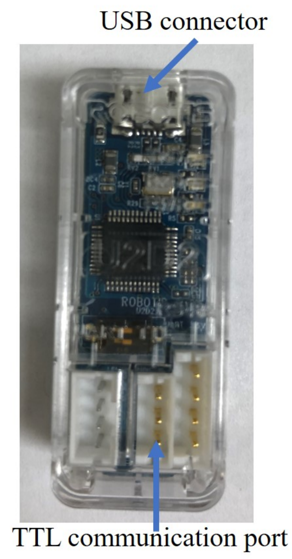
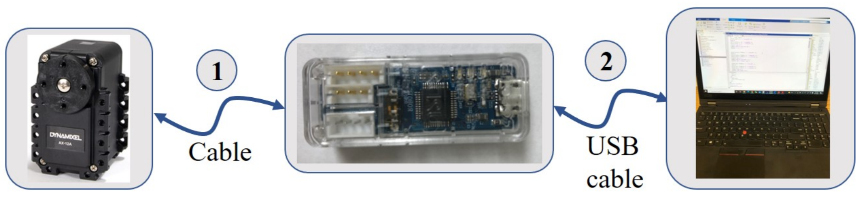


Figure 5.4: Simplified robot with assigned frames

Four servos are connected in series and fixed on a base. A U2D2 adapter is used to connect the servos to the laptop. U2D2 is a small size USB communication converter that enables to control and to operate the DYNAMIXEL with the laptop. It uses the USB cable to connect to the laptop and prevents damage of the USB terminals. It has both JST 3Pin connectors for TTL communication and JST 4Pin connectors for RS-485 communication. In this experiment, the TTL communication is used. A photograph of U2D2 and the corresponding block diagram model that describes the input and output signals are shown in Fig. 5.5. A switched-mode power supply (SMPS) is used to provide power. However, an adapter, SMPS2Dynamixel, should be used between the SMPS and the servo. The SMPS is connected to the DC terminal and then connect the servo using cable. The diagram is shown in Fig. 5.6.



(a). A U2D2 photograph



1. Read and write servo information package.
2. Read servo information based on address and send desired instructions, such as position, velocity, and torque.

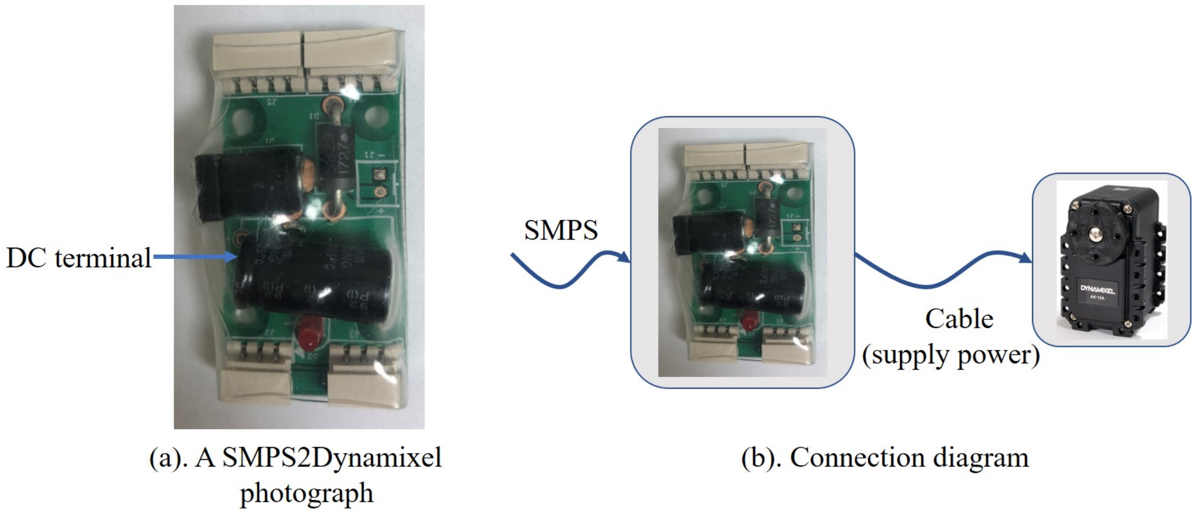
(b). Connection diagram

Figure 5.5: U2D2

5.3 Dynamics modeling

A simplified robot structure with assigned frames is shown in Fig. 5.4. The dynamic model can be expressed as:

$$\boldsymbol{M}(\boldsymbol{q},\dot{\boldsymbol{q}})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{f}_{fric} + \boldsymbol{d} = \boldsymbol{\tau} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{F}_e. \tag{5.1}$$



(a). A SMPS2Dynamixel photograph

(b). Connection diagram

Figure 5.6: SMPS2DYNAMIXEL

where $\boldsymbol{q}$ is defined as $\boldsymbol{q} = [q_1\ q_2\ q_3\ q_4]^{\mathrm{T}}$. Matrices $\boldsymbol{M}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are the inertia matrix and the Coriolis and centripetal matrix, respectively. Vector $\boldsymbol{G}(\boldsymbol{q})$ is the gravity term. The friction is written as $\boldsymbol{f}_{fric}$ and $\boldsymbol{d}$ describes the disturbance. The process input torque is $\boldsymbol{\tau}$. Vector $\boldsymbol{F}_e$ is the external force excerted on the manipulator and $\boldsymbol{J}$ is the corresponding jacobian matrix.

Considering system uncertainties, the dynamic model can be written as:

$$\hat{\boldsymbol{M}}\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}\dot{\boldsymbol{q}} + \hat{\boldsymbol{G}} + \hat{\boldsymbol{f}}_{fric} + \boldsymbol{\delta} = \boldsymbol{\tau} + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{F}_e. \tag{5.2}$$

where

$$\begin{aligned} \delta\boldsymbol{M} &= \boldsymbol{M}(\boldsymbol{q}, \hat{\boldsymbol{q}}) - \hat{\boldsymbol{M}} \\ \delta\boldsymbol{C} &= \boldsymbol{C}(\boldsymbol{q}, \hat{\boldsymbol{q}}) - \hat{\boldsymbol{C}} \\ \delta\boldsymbol{G} &= \boldsymbol{G}(\boldsymbol{q}) - \hat{\boldsymbol{G}} \\ \delta\boldsymbol{f}_{fric} &= \boldsymbol{f}_{fric} - \hat{\boldsymbol{f}}_{fric} \\ \boldsymbol{\delta} &= \delta\boldsymbol{M}\ddot{\boldsymbol{q}} + \delta\boldsymbol{C}\dot{\boldsymbol{q}} + \delta\boldsymbol{G} + \delta\boldsymbol{f}_{fric} + \boldsymbol{d}. \end{aligned} \tag{5.3}$$

In Eq. (5.3), $\hat{\boldsymbol{M}}$, $\hat{\boldsymbol{C}}$, $\hat{\boldsymbol{G}}$, and $\hat{\boldsymbol{f}}_{fric}$ are the nominal model parameters of the inertial matrix, Coriolis and centripetal matrix, gravity, and friction term, respectively. $\boldsymbol{\delta}$ is the clumped uncertainty term. In this research, system uncertainties are assumed to be continuous with respect to time.

5.4 Control strategies

Neural network is able to fitting smooth and continuous functions online theoretically. Based on this characteristics, two control methods are proposed and tested to demonstrate the performance of neural network when applied to robot control. The first controller is designed based on dynamic model and joint torque is used as the input. While the second controller uses joint position as the input to get desired trajectory.

5.4.1 Torque control

In this part, joint torque is selected as the input. The control scheme is shown in Fig. 5.7 and he control law is designed as:

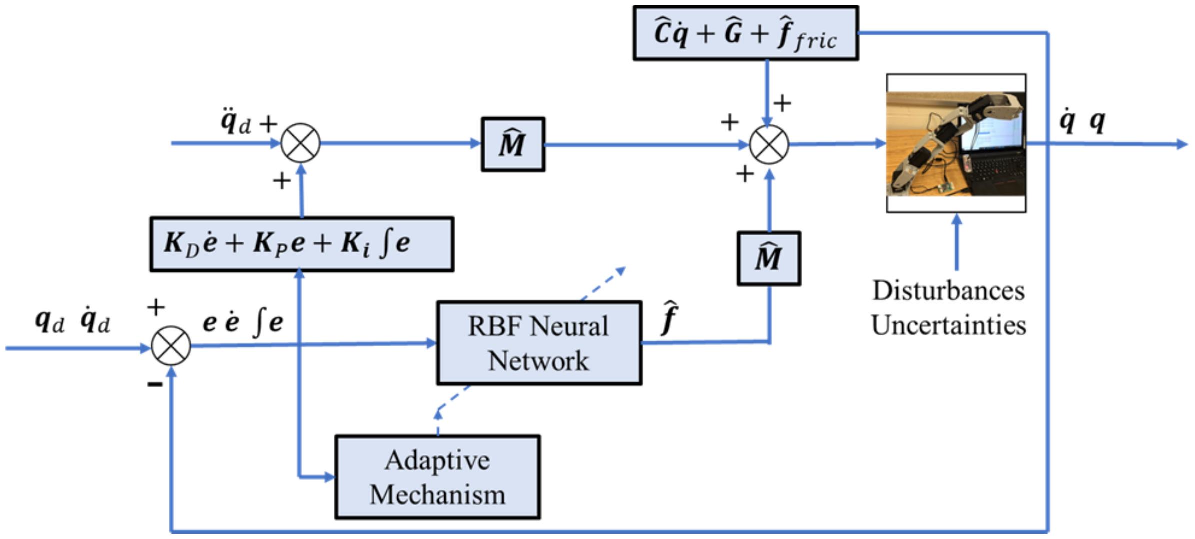


Figure 5.7: Torque control scheme

$$\boldsymbol{\tau} = \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2. \tag{5.4}$$

where

$$\begin{aligned} \boldsymbol{\tau}_1 &= \hat{\boldsymbol{M}}(\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_d\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_i\int \boldsymbol{e}) + \hat{\boldsymbol{C}}\dot{\boldsymbol{q}} + \hat{\boldsymbol{G}} + \hat{\boldsymbol{f}}_{fric} \\ \boldsymbol{\tau}_2 &= \hat{\boldsymbol{M}}\hat{\boldsymbol{f}}. \end{aligned} \tag{5.5}$$

In Eq. (5.5), $\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$. $\boldsymbol{K}_p$, $\boldsymbol{K}_d$, and $\boldsymbol{K}_i$ are the proportional, derivative, and integral gains. $\hat{\boldsymbol{f}}$ is the estimation of system uncertainties, which is the output of the neural network.

For the torque control, the input is chosen as $\boldsymbol{s} = [\int \boldsymbol{e}^{\mathrm{T}}\ \boldsymbol{e}^{\mathrm{T}}\ \dot{\boldsymbol{e}}^{\mathrm{T}}]^{\mathrm{T}}$. The adaptive law of the neural network is selected as:

$$\dot{\boldsymbol{W}} = \gamma \boldsymbol{H}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{F}. \tag{5.6}$$

where γ is a positive constant. $\boldsymbol{P}$ is a symmetric positive definite matrix. $\boldsymbol{F}$ is defined as:

$$\boldsymbol{F} = [\mathbf{0}_{1\times 8}\ \mathbf{1}_{1\times 4}]^{\mathrm{T}}. \tag{5.7}$$

5.4.2 Position control

The output position of the servo is related to the input position and a torque value, which is used to move the servo to desired position. Considering the unknown control methods inside the servo, one strategy is to give a constant torque value and position in every loop. The scheme is shown in Fig. 5.8. However, due to the compliance slope in Fig. 5.2, tracking error exists because the torque value decreases as the servo moves close to the goal position. Internal noise affects the performance as well. Therefore, it is necessary to reduce system nonlinear uncertainties and improve the performance of servos.

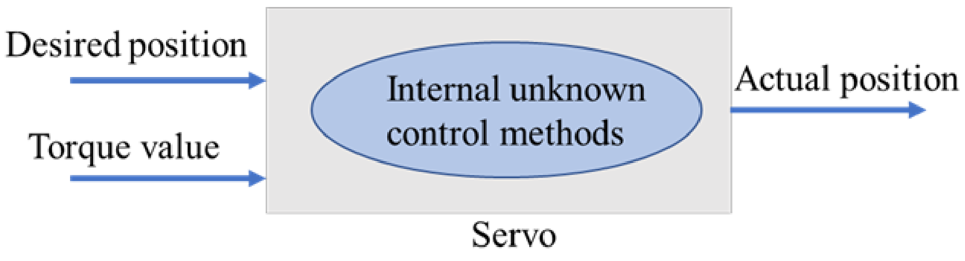
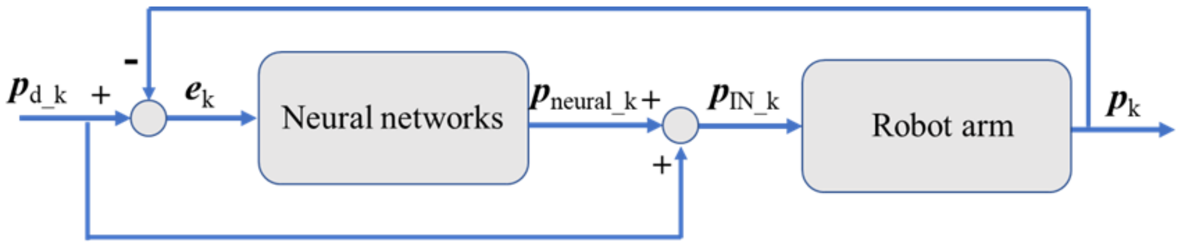


Figure 5.8: Direct servo motion

As shown in Fig. 5.8, there are two inputs for the position control method. One strategy is to give a constant torque value and change the desired position. The reason is due to the servo characteristics and experiment results will show the rationality of this choice.

System uncertainties are assumed to be continuous for the robot arm and a neural network is utilized to adjust the input position. The structure of the control scheme is shown in Fig. 5.9. The tracking error at time t_k is



$\boldsymbol{p}_{\mathrm{d\_k}}$
+
-
$\boldsymbol{e}_{\mathrm{k}}$
Neural networks
$\boldsymbol{p}_{\mathrm{neural\_k}}$
+
+
$\boldsymbol{p}_{\mathrm{IN\_k}}$
Robot arm
$\boldsymbol{p}_{\mathrm{k}}$

Figure 5.9: Control scheme

$$\boldsymbol{e}_k = \boldsymbol{p}_{d\_k} - \boldsymbol{p}_k. \tag{5.8}$$

Vector $\boldsymbol{p}_{d\_k} = [p_{d1\_k}\ p_{d2\_k}\ p_{d3\_k}\ p_{d4\_k}]^{\mathrm{T}}$ is the desired position and $\boldsymbol{p}_k = [p_{1\_k}\ p_{2\_k}\ p_{3\_k}\ p_{4\_k}]^{\mathrm{T}}$ is the output at t_k. Considering system uncertainties, $\boldsymbol{p}_k$ can be expressed as:

$$\boldsymbol{p}_k = f(\boldsymbol{p}_{IN\_k}). \tag{5.9}$$

where $\boldsymbol{p}_{IN\_k} = [p_{IN1\_k}\ p_{IN2\_k}\ p_{IN3\_k}\ p_{IN4\_k}]^{\mathrm{T}}$ is the input at t_k and $f(\cdot)$ is the nonlinear function of the system. As the input, $\boldsymbol{p}_{IN\_k}$ can be written as:

$$\boldsymbol{p}_{IN\_k} = \boldsymbol{p}_{d\_k} + \boldsymbol{p}_{neural\_k}. \tag{5.10}$$

where $\boldsymbol{p}_{neural\_k}$ is the output of the neural network. The input of the neural network is set as $\boldsymbol{s} = \boldsymbol{e}_k$ and the weight matrix at t_k is $\boldsymbol{W}_k^{\mathrm{T}}$. Therefore, the output of the neural network is expressed as:

$$\boldsymbol{p}_{neural\_k} = [p_{N1}\ p_{N2}\ p_{N3}\ p_{N4}]^{\mathrm{T}} = \boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{H}. \tag{5.11}$$

In order to adjust the weights adaptively, the total error at t_k is defined as:

$$Err = \frac{1}{2}\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{e}_k. \tag{5.12}$$

The gradient of Err along a single weight is expressed as:

$$\begin{aligned}
\frac{dErr}{dw_{11}} &= e_1 \cdot \frac{d(p_{d1\_k} - p_{1\_k})}{dw_{11}} = -e_1 \cdot \frac{dp_{1\_k}}{dp_{IN1\_k}} \cdot h_1 \\
\frac{dErr}{dw_{21}} &= e_1 \cdot \frac{d(p_{d1\_k} - p_{1\_k})}{dw_{21}} = -e_1 \cdot \frac{dp_{1\_k}}{dp_{IN2\_k}} \cdot h_2 \\
&. \\
&. \\
&. \\
\frac{dErr}{dw_{m4}} &= e_4 \cdot \frac{d(p_{d4\_k} - p_{4\_k})}{dw_{m4}} = -e_4 \cdot \frac{dp_{4\_k}}{dp_{IN4\_k}} \cdot h_m
\end{aligned} \tag{5.13}$$

The model is a bounded input bounded output (BIBO) system, which means $\frac{dp_{1\_k}}{dp_{IN1\_k}}$, $\frac{dp_{1\_k}}{dp_{IN2\_k}}$, ..., and $\frac{dp_{4\_k}}{dp_{IN4\_k}}$ are restricted to a boundary. To move toward the minimum of Err, the weight change should be in opposite direction. Therefore, the weight change using steepest descent direction can be written as:

$$\Delta\boldsymbol{W}_{k,k+1}^{\mathrm{T}} = \gamma\boldsymbol{e}_k\boldsymbol{H}^{\mathrm{T}}. \tag{5.14}$$

where γ is a positive constant. Then the updated weight matrix at t_{k+1} is:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \Delta\boldsymbol{W}_{k,k+1}. \tag{5.15}$$

5.5 Experiment results and discussion

In this section, a series of experiments based on the above two control strategies are tested and compared. Differences between theoratical analysis and practical experiment results are analyzed and reasons are given to explain the phenomenon.

5.5.1 Desired trajectory

The following experiments are based on a same desired trajectory. There are a few factors that affect the generation of desired trajectory.

(1). There is a max output torque for the servo, which means the acceleration of joint should not be too large. If the trajectory is not planned appropriately, the calculated torque may be larger than the max torque of the servo, and the performance of the controller can not be evaluated properly.

(2). The initial position and destination position should not be too close to the motion boundary of the servo, especially for the torque control. If the control parameters are not selected properly, the trajectory of every joint has a chance to get stuck at the boundary.

Due to those concerns, the initial position is set as $\boldsymbol{q}_s = [40^\circ \; 10^\circ \; -40^\circ \; 0^\circ]$. The destination position is $\boldsymbol{q}_e = [90^\circ \; -60^\circ \; 10^\circ \; -50^\circ]$. The robot arm moves from the initial position to the destination position in 10 seconds and moves back in the same way. The joint motion trajectory is generated using a fifth order polynomial, which has the advantage that the velocity and acceleration at the initial and end point are zero. The sampling frequency is 16Hz by setting the latency time of the laptop USB port.

5.5.2 Robot parameters

The robot parameters are shown in Table. 5.1.

	Mass (kg)	Length (m)	Center (m)	I_{xx} (kg $\cdot$ m^2)	I_{yy} (kg$\cdot$ m^2)	I_{zz} (kg$\cdot$ m^2)
Link 1	0.06	0.095	0.08	1.25×10^{-5}	2.05×10^{-5}	1.7×10^{-5}
Link 2	0.06	0.095	0.08	1.25×10^{-5}	2.05×10^{-5}	1.7×10^{-5}
Link 3	0.06	0.095	0.08	1.25×10^{-5}	2.05×10^{-5}	1.7×10^{-5}
Link 4	0.03	0.08	0.06	6.25×10^{-6}	1.025×10^{-5}	8.5×10^{-6}

Table 5.1: Physical parameters of the robot

5.5.3 Experiment based on torque control

As shown in Fig. 5.8, each servo receives two signals in one loop, desired position and output torque value. However, the servo only accepts a torque value without any directions for the torque control. Considering the structure of the servo in Fig. 5.1 and the compliance slope in Fig. 5.2, we use a pair of position and torque as our inputs. If the torque is negative, then the input is $(0,\ |\mathrm{torque}|)$. If the torque is positive, then the input is set as $(1023,\ |\mathrm{torque}|)$.

The static friction of the joints is set as $\hat{\boldsymbol{f}}_{fric} = [80\ 80\ 100\ 100]^{\mathrm{T}}$, which corresponds to $\hat{\boldsymbol{f}}_{fric} = [0.117\ 0.117\ 0.147\ 0.147]^{\mathrm{T}}$ N·m. The proportional, derivative, and integral gains are selected as $\boldsymbol{K}_p = diag([480\ 540\ 2010\ 2340])$, $\boldsymbol{K}_d = diag([20\ 20\ 30\ 30])$, and $\boldsymbol{K}_i = diag([40\ 40\ 60\ 60])$, respectively.

For the neural network, the center point vector matrix is set as:

$$\boldsymbol{c} = [\boldsymbol{c}_1\ \boldsymbol{c}_2\ \boldsymbol{c}_3\ \boldsymbol{c}_4\ \boldsymbol{c}_5] = [-\mathbf{2}_{12\times1}\ -\mathbf{1}_{12\times1}\ \mathbf{0}_{12\times1}\ \mathbf{1}_{12\times1}\ \mathbf{2}_{12\times1}]. \tag{5.16}$$

The stretch constant vector is selected as

$$\boldsymbol{b} = \mathbf{1}_{5\times1} \tag{5.17}$$

The initial weight matrix is:

$$\boldsymbol{W}^{\mathrm{T}} = \mathbf{1}_{4\times5}. \tag{5.18}$$

The trajectory of the four joints is shown in Fig. 5.10 - 5.13 and the tracking errors are shown in Fig. 5.14 - 5.17. The joint torques are shown in Fig. 5.18 - 5.21.

Theoretically, the control methods using neural network should have better performance than that without neural network. However, the experiment shows different conclusion compared to the theory. It is difficult to tell which one is better or inferior. Some reasons are given below.

(1). We assumed that system uncertainties are continuous before we design the neural network. However, uncertainties are not continuous in practice. Static friction is an important disturbance for this model. In different positions, static friction has different values, which means a constant estimation is not accurate.

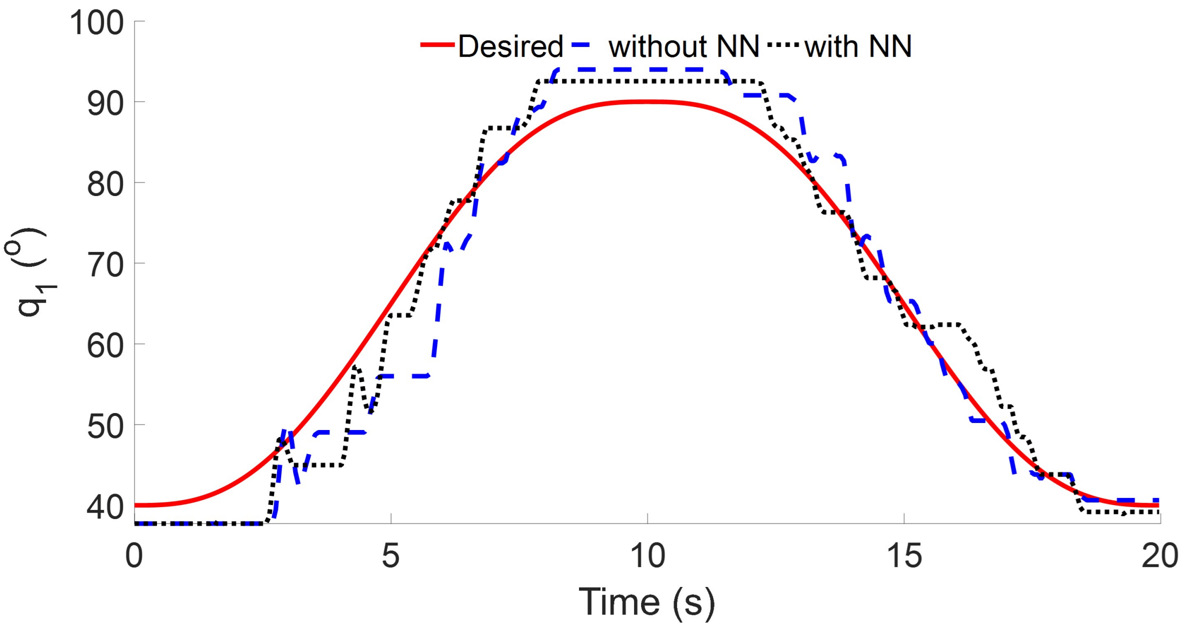


Figure 5.10: Joint 1 response using torque control

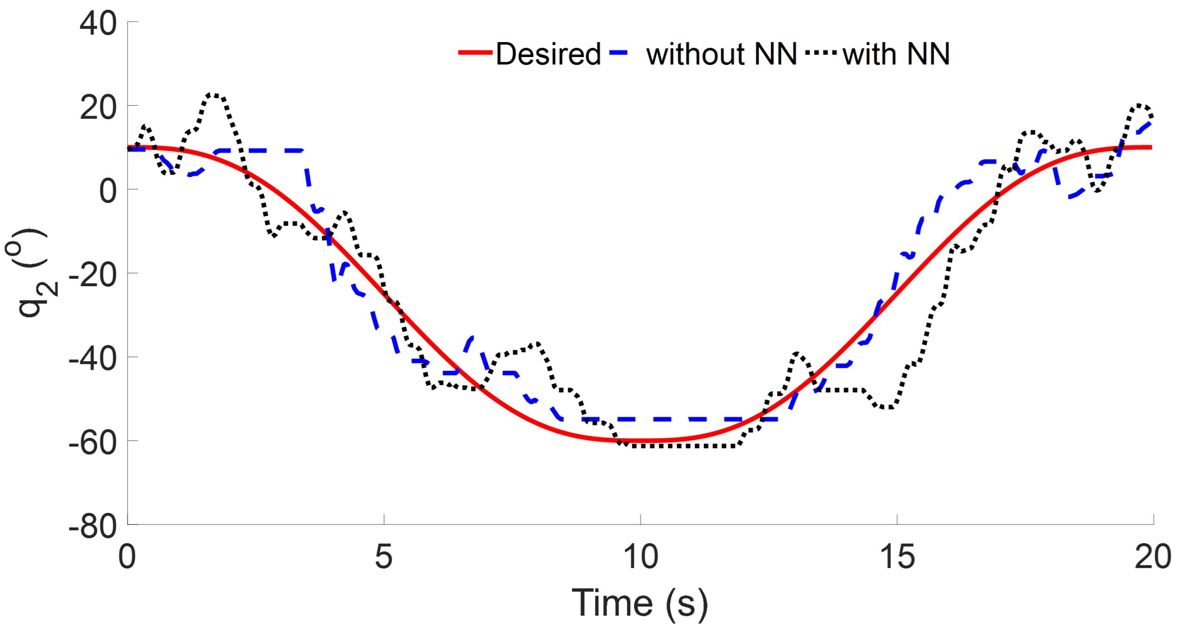


Figure 5.11: Joint 2 response using torque control

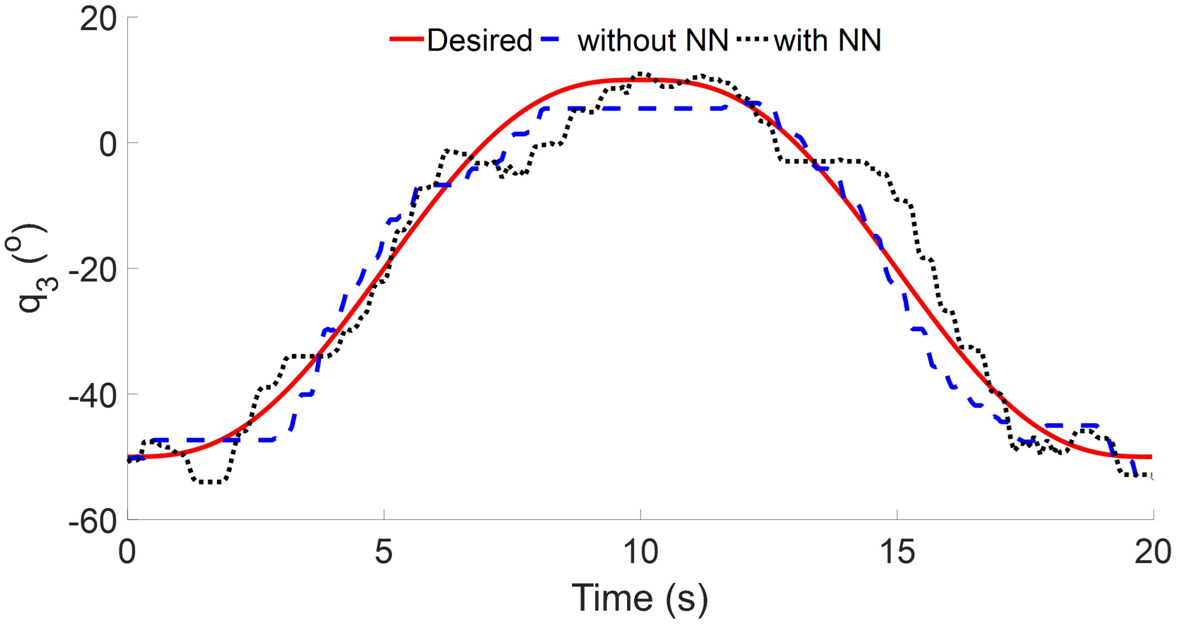


Figure 5.12: Joint 3 response using torque control

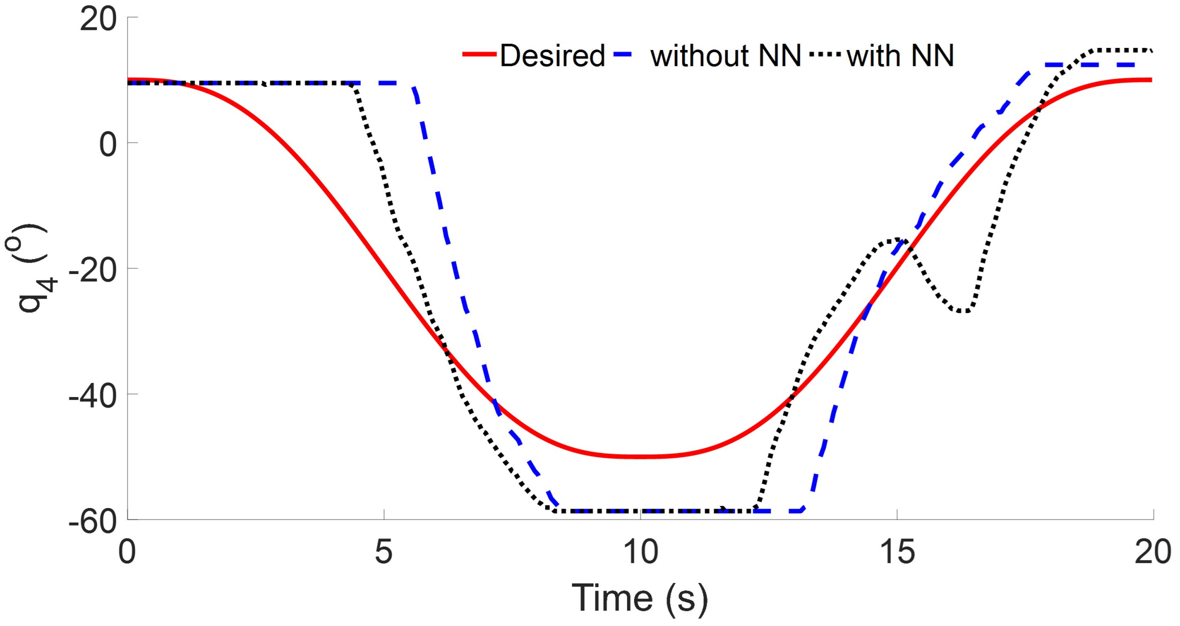


Figure 5.13: Joint 4 response using torque control

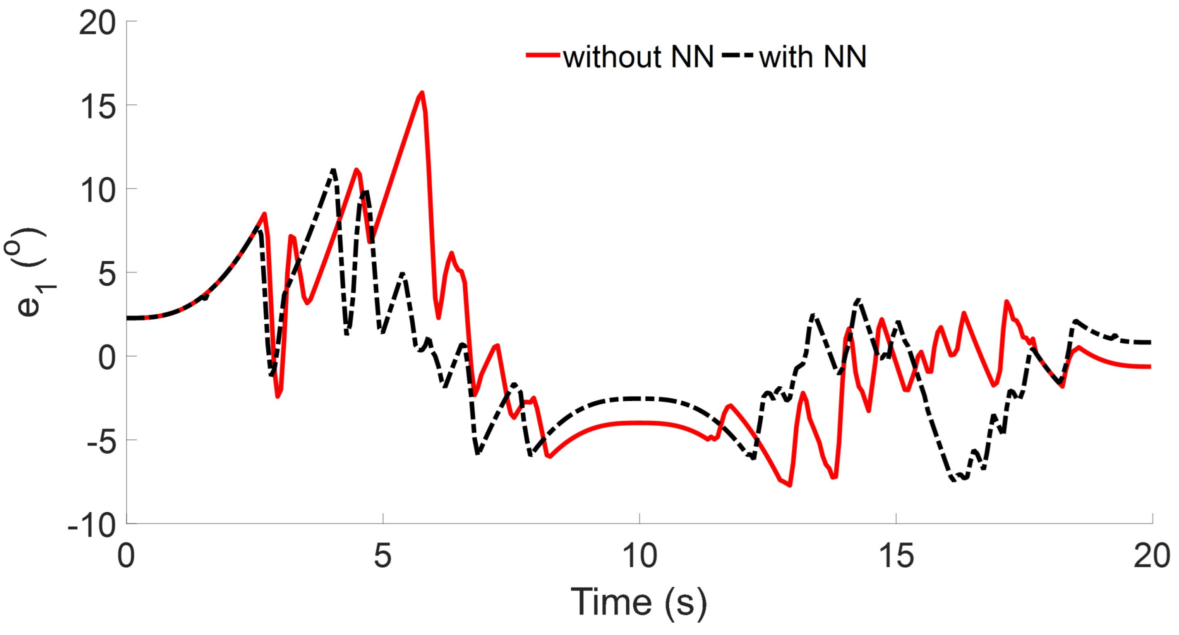


Figure 5.14: Joint 1 error using torque control

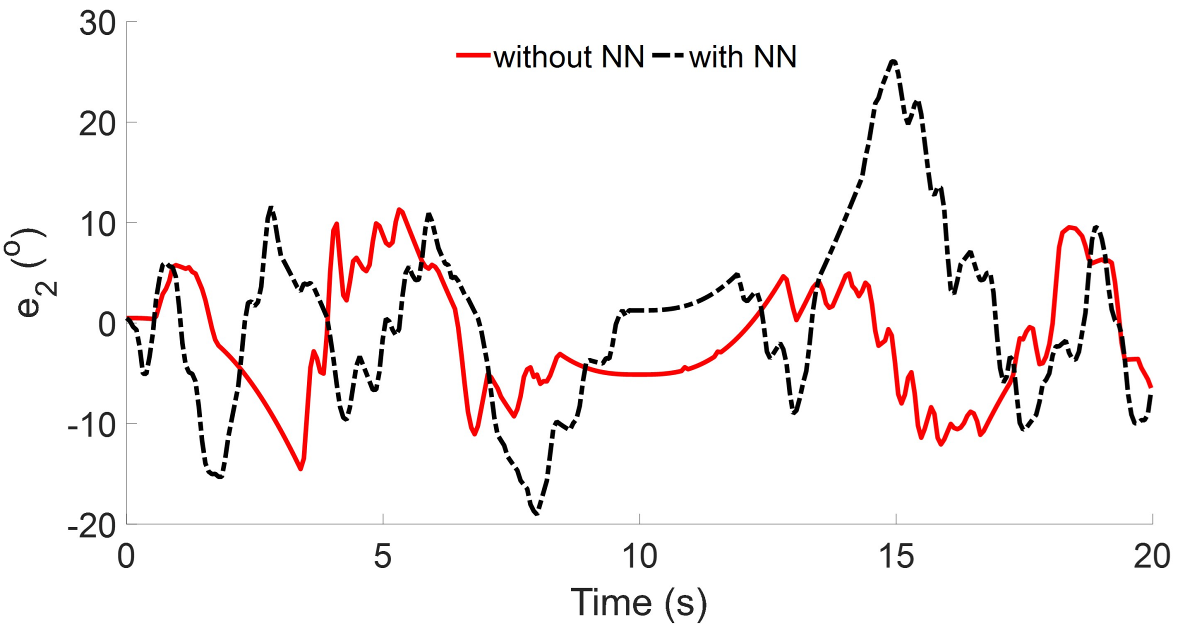


Figure 5.15: Joint 2 error using torque control

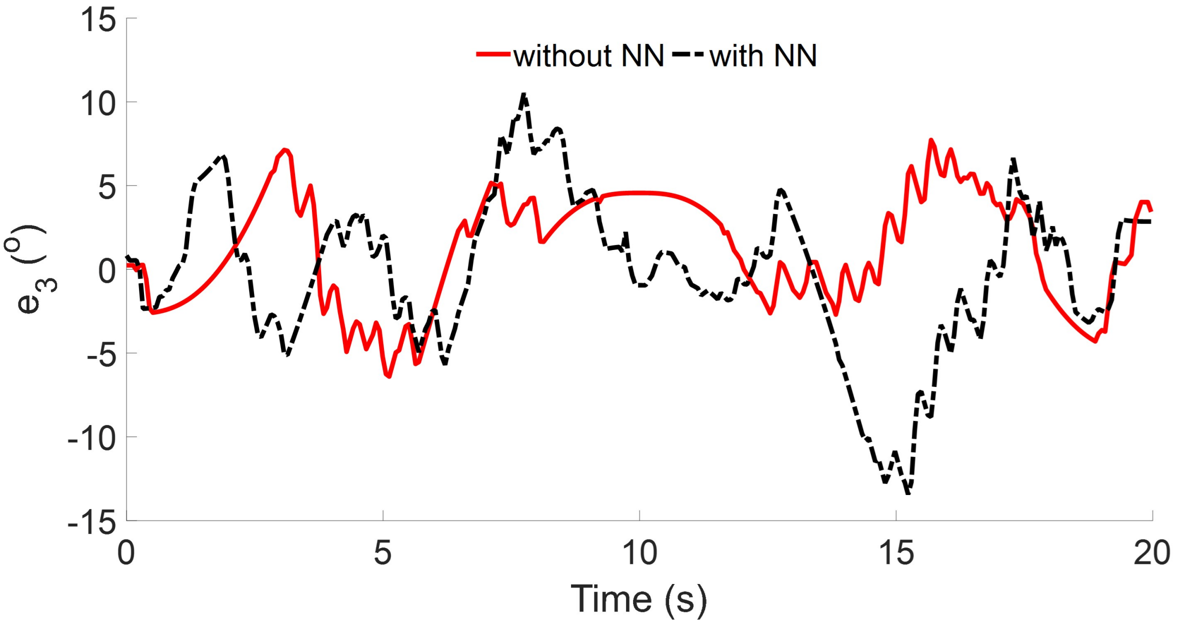


Figure 5.16: Joint 3 error using torque control

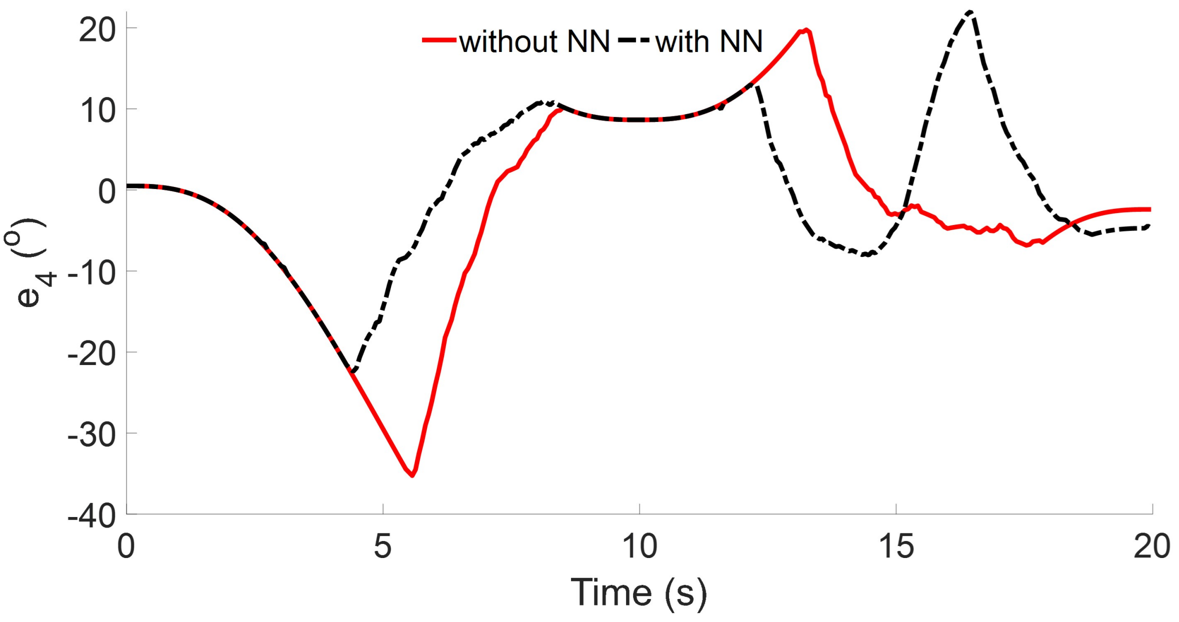


Figure 5.17: Joint 4 error using torque control

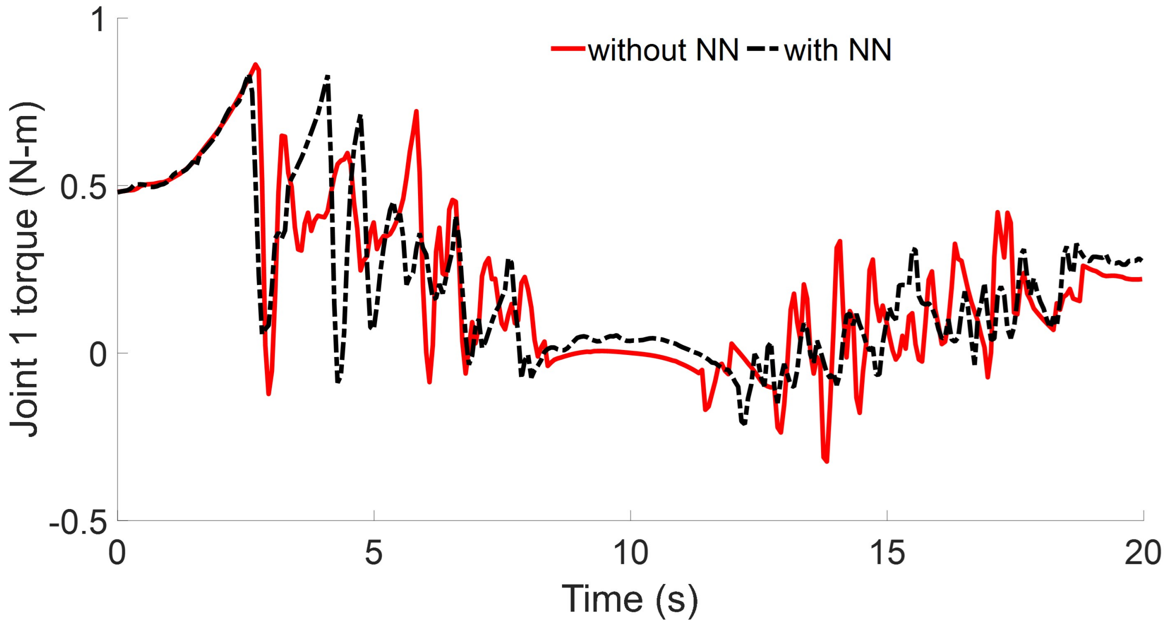


Figure 5.18: Joint 1 torque using torque control

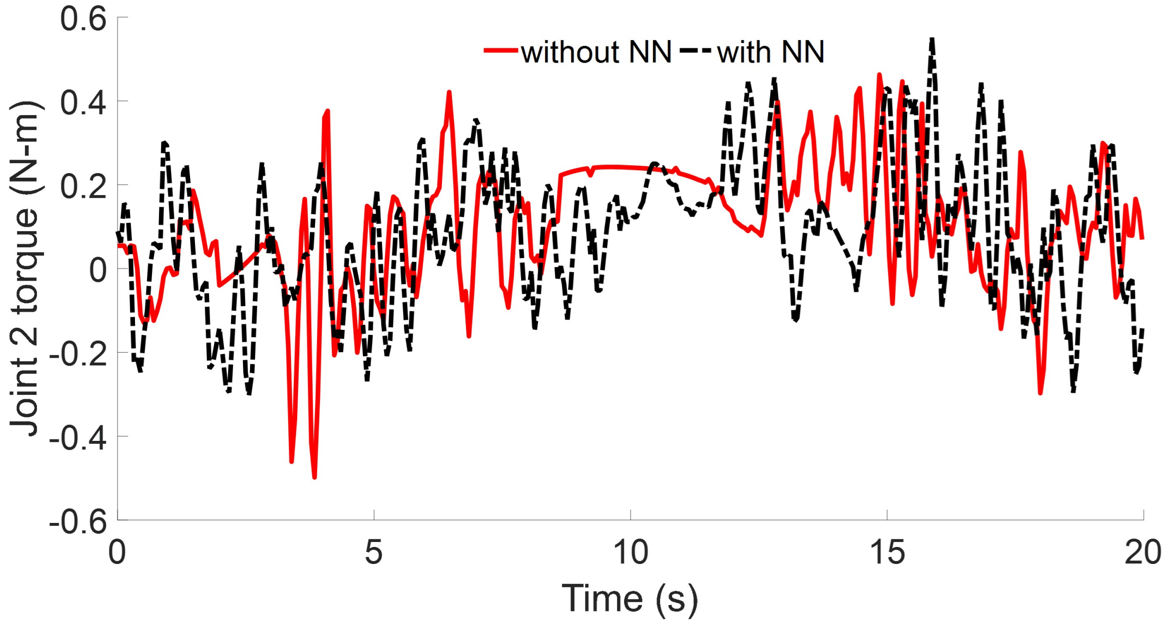


Figure 5.19: Joint 2 torque using torque control

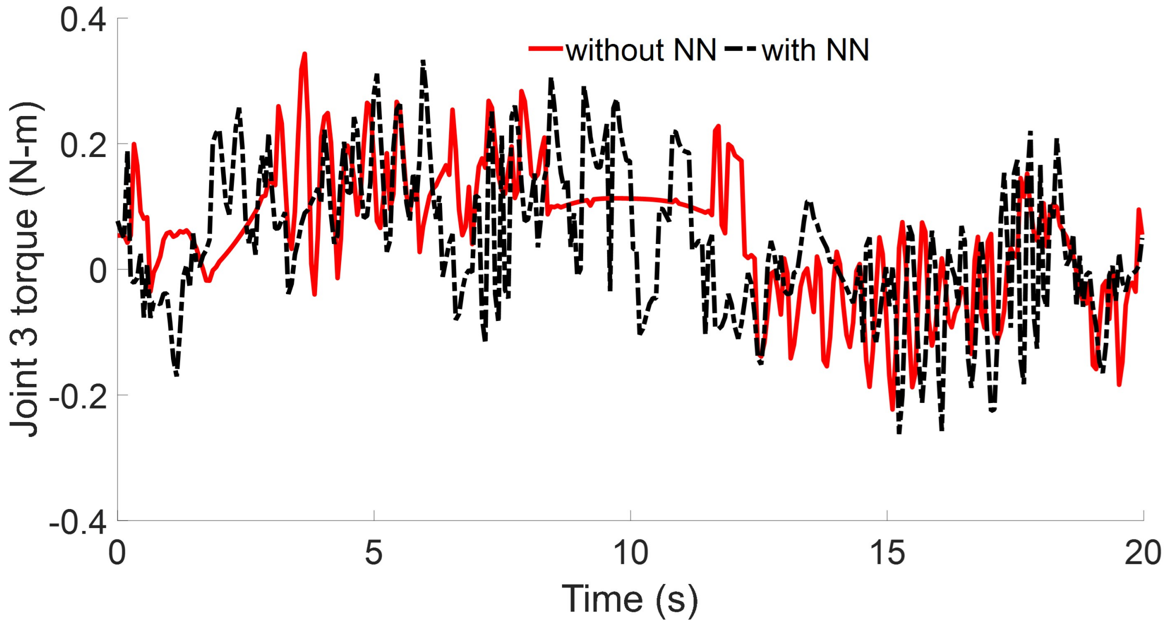


Figure 5.20: Joint 3 torque using torque control

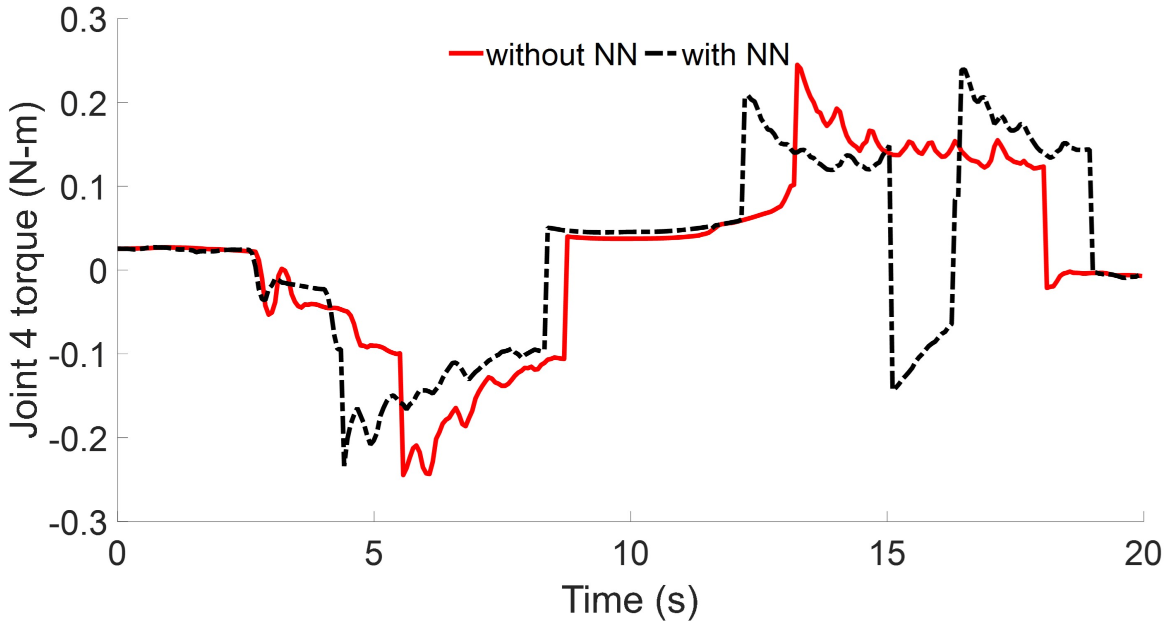


Figure 5.21: Joint 4 torque using torque control

(2). The torque control scheme inside the servo is unkonwn. We can write a torque value derived from the control method into the servo every loop. However, the output torque value may not be the same as that we write.

(3). The joint trajectory and input torque are not smooth as shown in Fig. 5.10 - 5.13 and Fig. 5.18 - 5.21. The trajectory shows chattering as well. These factors contribute to incontinuous system uncertainties.

In addition to the unknown torque control scheme in the servo and incontinuous uncertainties, there are some drawbacks when we use torque as our input.

(1). The control parameters are determined through many experiments. If the parameters are not selected properly, the system can become unstable easily.

(2). The control parameters only applies to this specific trajectory. We need to adjust the control parameters every time if we change the desired trajectory, which means this control method losts its generality.

(3). A linear control method is tested as well for this model. However, it is hard find a group of control parameters to follow the desired trajectory and the performance is even worse compared to the nonlinear controller described above.

5.5.4 Experiment based on position control

In order to avoid the unknown troque control scheme inside the servo for the position control, we write a constant torque value every loop. The torque is set as the maximum to afford the joint motion, which means the input pair is (position, 1023).

Firstly, we test the nonlinearity of the model. The desired trajectory is to move from the initial position to the destination at a constant speed. The time is set to be 8s, 10s, and 12s, respectively. The tracking error of the first joint with respect to desired position is shown in Fig. 5.22. It can be seen that joint velocity has little effect on the tracking errors, which means the model is a nonlinear system under position control.

For the neural network used in position control, the stretch constant vector is selected as:

$$\boldsymbol{b} = \mathbf{1}_{5\times 1}. \tag{5.19}$$

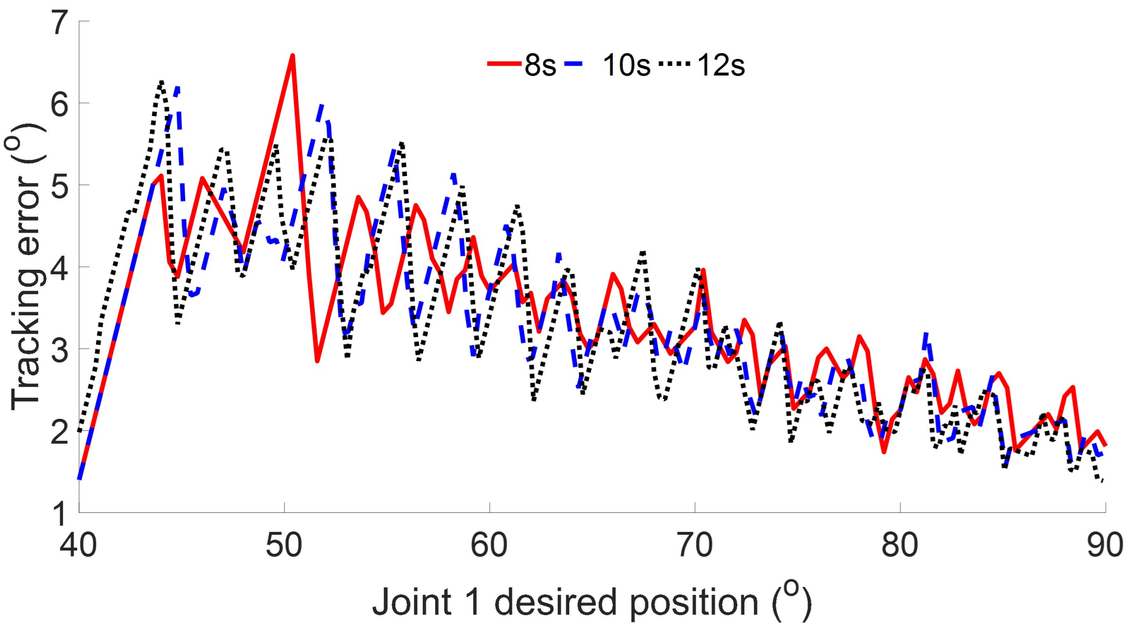


Figure 5.22: Nonlinearity test

The initial weight matrix is selected as:

$$\boldsymbol{W}_0^{\mathrm{T}} = 10^{-4} \times \mathbf{1}_{4\times 5}. \tag{5.20}$$

The constant coefficient γ is selected as $\gamma = 0.03$. In order to show the effectiveness of the neural network, a linear PID controller is used to test the robot arm as well. The corresponding parameters are $\boldsymbol{K}_p = diag([0.1\ 0.1\ 0.1\ 0.1])$, $\boldsymbol{K}_i = diag([0.1\ 0.1\ 0.1\ 0.1])$, and $\boldsymbol{K}_d = diag([0.05\ 0.05\ 0.05\ 0.05])$.

Three experiments are tested to show the performance of the neural network, open loop, PID control, and neural network control. The tracking trajectory is shown in Fig. 5.23 - 5.26 and the corresponding tracking error is shown in Fig. 5.27 - 5.30. It can be seen that the tracking error using neural network is reduced significantly compared to that using open loop and PID control.

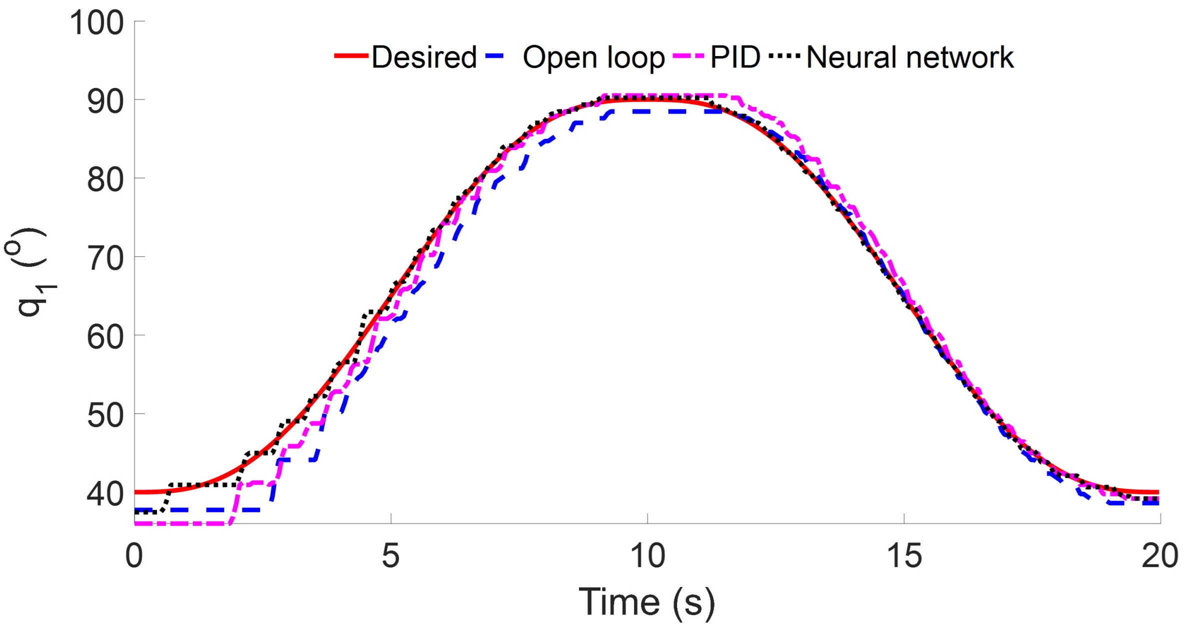


Figure 5.23: Joint 1 response using position control

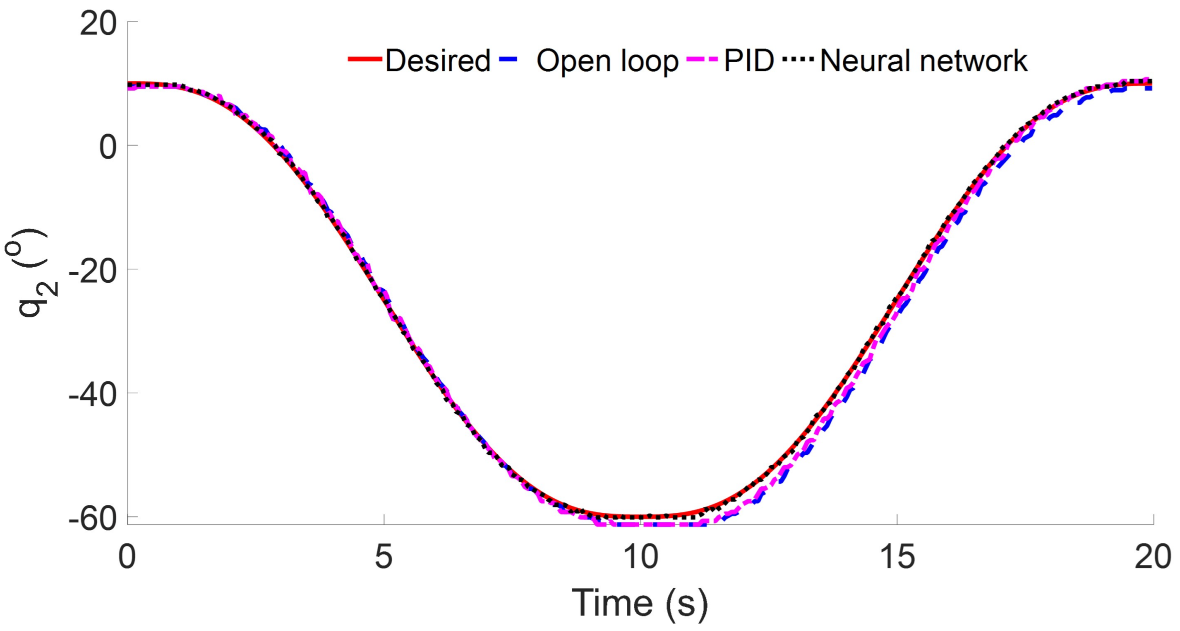


Figure 5.24: Joint 2 response using position control

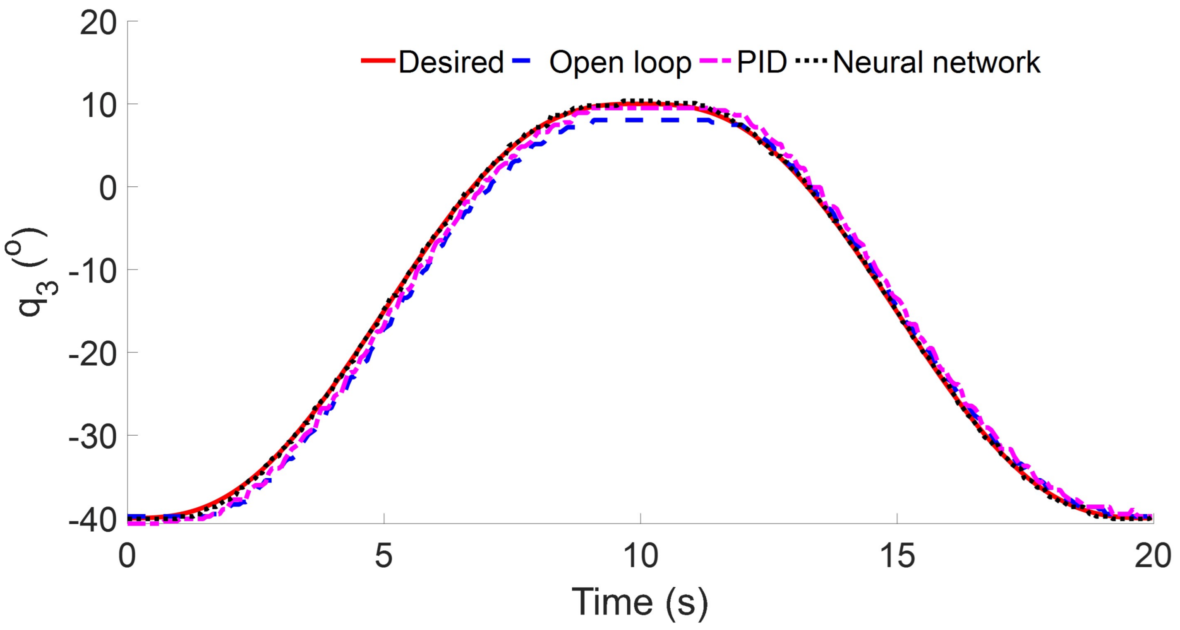


Figure 5.25: Joint 3 response using position control

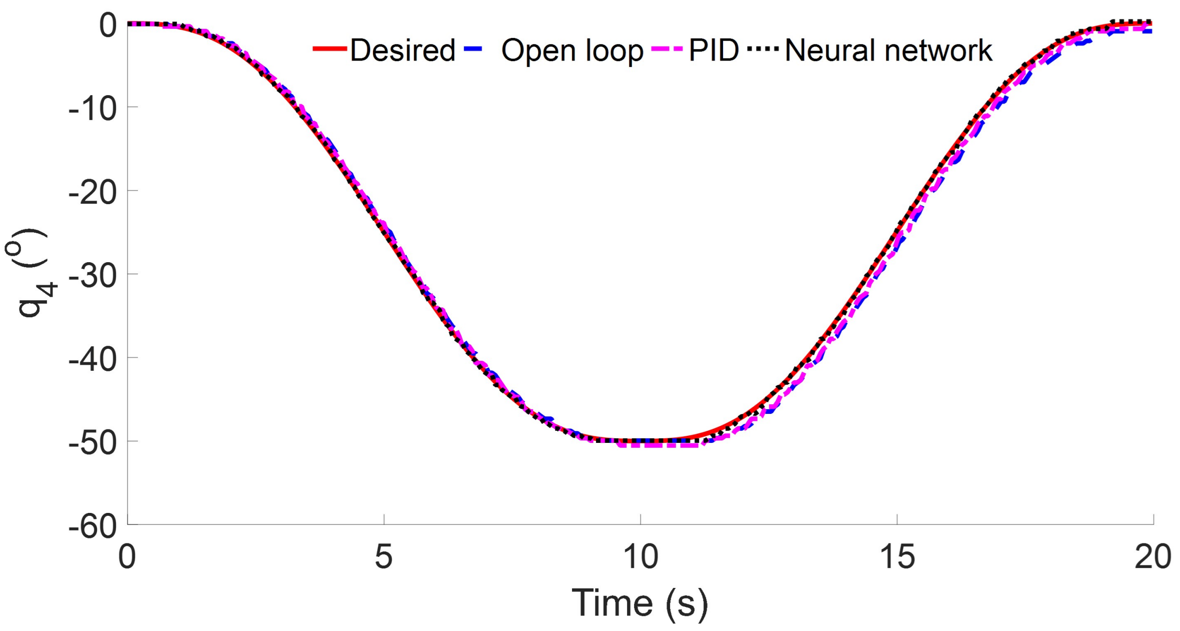


Figure 5.26: Joint 4 response using position control

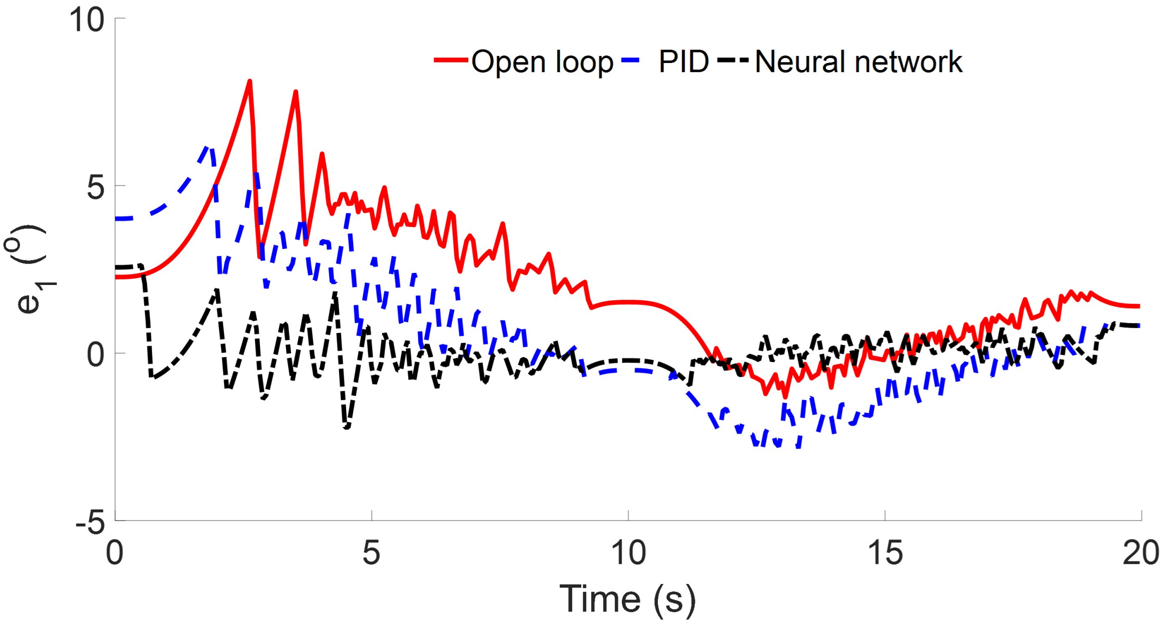


Figure 5.27: Joint 1 error using position control

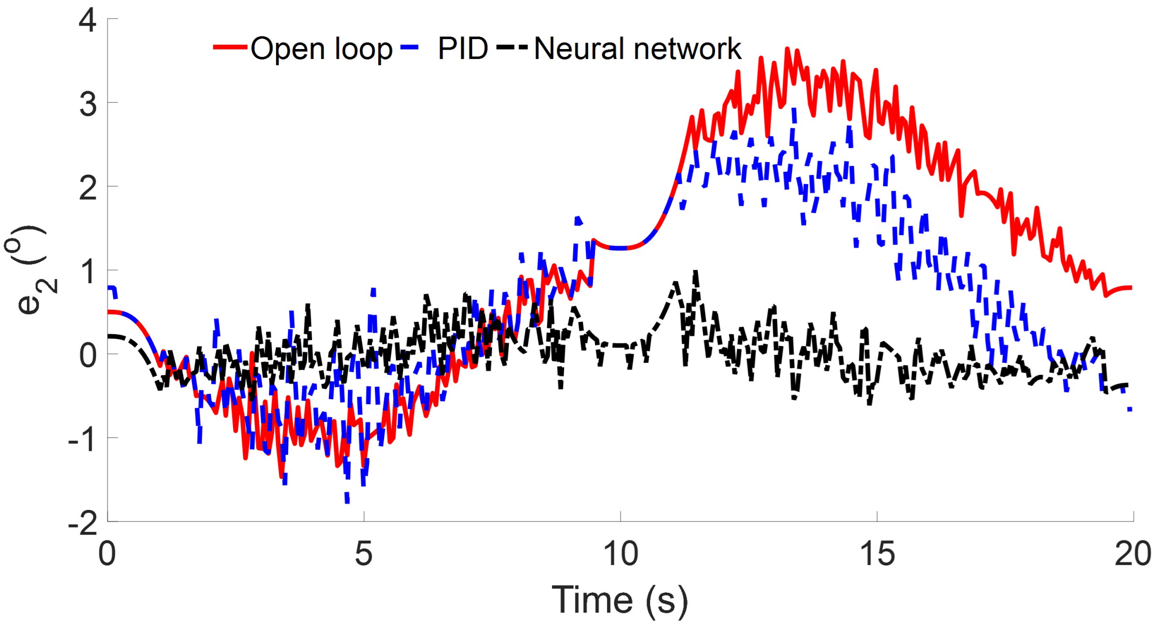


Figure 5.28: Joint 2 error using position control

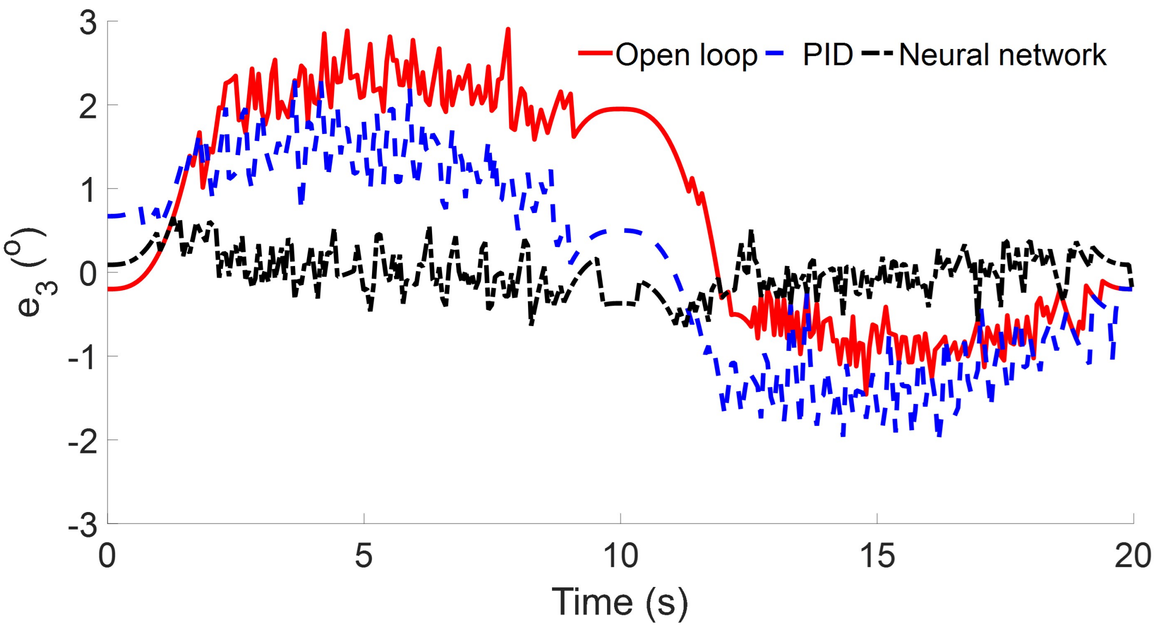


Figure 5.29: Joint 3 error using position control

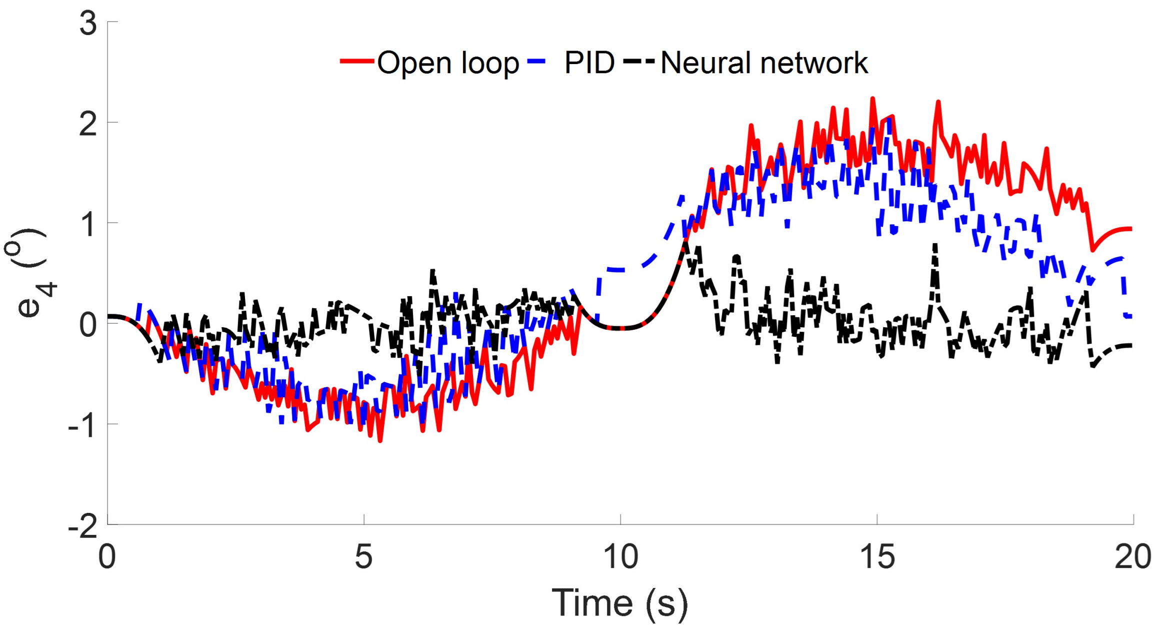


Figure 5.30: Joint 4 error using position control

In order to more quantitatively compare the performances of these three methods, Integral of Squared Error (ISE), which is commonly used to measure the quality of system response over the whole range of time, is taken into account. The defination is written as

$$\text{ISE} = \int_0^\infty e^2 dt \tag{5.21}$$

The values of these performance indices for the errors of each joint are computed and shown in Table. 5.2. Clearly, the ISE which emphasizes transient errors has smaller values by the neural network control than by open loop and PID control. The neural network compensation is able to reduce the ISE by more than 15x that achieved by the uncompensated ("open-loop") commercial controller, and 8x-20x better than the PID compensated system. That means the neural network control shows high effectiveness to keep errors near zero during the whole task.

Joint	Open loop	PID	Neural network
# 1	0.0497	0.0304	0.0031
# 2	0.0187	0.0092	0.0005
# 3	0.0142	0.0086	0.0004
# 4	0.0073	0.0043	0.0003

Table 5.2: The values of the ISE of the joint errors

5.6 Summary

In this section, two control strategies using neural networks based on torque and position are proposed and tested on a four-joint robot arm. For the torque control, there are large tracking errors due to the incontinuous uncertainties. For the position control, the neural network is able to compensate for the system uncertainties and reduce the tracking errors significantly compared to the open loop and linear PID control methods. The ISE values are compared to show the effectiveness of neural network control. In addition, the position control strategy using a neural network can be applied to random trajectories without changing the adaptive law and control parameters.

Chapter 6

Conclusion and future work

6.1 Conclusion

In this dissertation, trajectory tracking and force control strategies are proposed for a UVMS. The main contributions are listed below.

First, three joint space trajectory tracking methods are proposed to follow desired joint path. As a decentralized control strategy, backstepping control can be used directly to a nonlinear system and there is no need to derive its dynamic model. A fuzzy logic is added to eliminate the uncertain terms. Computer simulation shows its effectiveness when applied to follow desired joint path. However, steady state tracking errors exist. Considering this problem, a controller with an integral term is designed based on nominal dynamic model and an adaptive neural network is adopted to eliminate the effect of system uncertainties. But there are periodic tracking errors for periodic tasks. Integral sliding mode control has the advantage of remaining robust against uncertainties. Therefore, a dual integral sliding mode controller is proposed and it shows good performance when used to follow desired joint trajectories.

Due to system redundancy, it is difficult to calculate the inverse kinematics. To follow a desired trajectory for an end-effector, a task space trajectory tracking controller is designed. A neural network is used to eliminate the uncertain term in task space and the mapping relationship between joint space and task space is modified to guarantee system stabilities. In addition, a force control task is analyzed as well.

Finally, experiments are carried out to explore the application of neural networks. Since neural networks are used to approximate continuous functions, the performance of the torque control method with a neural network shows no obvious improvement due to the discontinuity of uncertain terms. While a compensator using a network is adopted for the position control and the performance is compared to that of the uncompensated controller and the PID compensated method. The experimental results show that the neural network compensation is able to reduce the ISE by more than 15x that achieved by the uncompensated commercial controller, and 8x-20x better than the PID compensated system.

6.2 Future work

In this work, simple tasks are tested for the joint space control strategies. Inverse kinematics should be explored to use the control methods.

For the controllers designed in task space, the trajectory of the vehicle and manipulator is unpredictable, which may arise the following problems.

1. The manipulator is in a singular configuration.
2. Moving the vehicle consumes more energy than moving the manipulator, which causes energy waste.

So it is necessary to find a way to solve the above problems. Furthermore, time delay is not considered in this work, which can affect tracking accuracy significantly in some cases. It's interesting to introduce a time delay compensator and explore its stability.

References

[1] Ellery Alex. Tutorial review on space manipulators for space debris mitigation. *Robotics*, 8(2): 1-56, 2019.

[2] G. Heredia, A.E. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J.A. Acosta and A. Ollero. Control of a multirotor outdoor aerial manipulator. In *2014 IEEE/RSJ international conference on intelligent robots and systems*, pages 3417-3422, IEEE, 2014.

[3] David Ribas, Pere Ridao, Alessio Turetta, Claudio Melchiorri, Gianluca Palli, José Javier Fernández, and Pedro José Sanz. I-AUV mechatronics integration for the TRIDENT FP7 project. *IEEE/ASME Transactions on Mechatronics*, 20(5): 2583-2592, 2015.

[4] Manuel Cardona, Fernando Cortez, Andrés Palacios, Kevin Cerros. Mobile Robots Application Against Covid-19 Pandemic. In *2020 IEEE ANDESCON*, pages 1-5, IEEE, 2020.

[5] Hiroyuki NAGAMATSU, Takashi KUBOTA, and Ichiro NAKATANI. Capture strategy for retrieval of a tumbling satellite by a space robotic manipulator. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 70-75. IEEE, 1996.

[6] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters*, 3(3): 1957-1964, 2018.

[7] Jonathan Evans, Paul Redmond, Costas Plakas, Kelvin Hamilton, and David Lane. Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation. In *Oceans 2003. Celebrating the Past... Teaming Toward the Future (IEEE Cat. No. 03CH37492)*, pages 2201-2210. IEEE, 2003.

[8] Qingxin Meng, Hua Wang, Ping Li, Liquan Wang, and Ze He. Dexterous underwater robot hand: HEU hand II. In *2006 international conference on mechatronics and automation*, pages 1477-1482. IEEE, 2006.

[9] Hua Wang, Xiaodiao Huang, Xindan Qi, and Qingxin Meng. Development of underwater robot hand and its finger tracking control. In *2007 IEEE International Conference on Automation and Logistics*, pages 2973-2977. IEEE, 2007.

[10] Gianluca Antonelli and Stefano Chiaverini. Task-priority redundancy resolution for underwater vehicle-manipulator systems. In *1998 IEEE international conference on robotics and automation*, pages 768-773. IEEE, 1998.

[11] Gianluca Antonelli and Stefano Chiaverini. Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems. *IEEE Transactions on Fuzzy Systems*, 11(1): 109-120, 2003.

[12] Yang Hu, Bidan Huang, and Guang-Zhong Yang. Task-priority redundancy resolution for co-operative control under task conflicts and joint constraints. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2398-2405. IEEE, 2015.

[13] Qirong Tang, Le Liang, Jianhua Xie, Yinghao Li, and Zhenqiang Deng. Task-priority redundancy resolution on acceleration level for underwater vehicle-manipulator system. *International Journal of Advanced Robotic Systems*, 14(4): 1-9, 2017.

[14] Enrico Simetti and Giuseppe Casalino. Whole body control of a dual arm underwater vehicle manipulator system. *Annual Reviews in Control*, 40(1): 191-200, 2015.

[15] Enrico Simetti, Giuseppe Casalino, Sandro Torelli, Alessandro Sperinde, and Alessio Turetta. Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project. *Journal of Field Robotics*, 31(3): 364-385, 2014.

[16] Bruno Siciliano and Jean-Jacques E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *Proceeding of 5th International Conference on Advanced Robotics*, vol. 2, pages 1211-1216, 1991.

[17] Paolo Di Lillo, Filippo Arrichiello, Gianluca Antonelli, and Stefano Chiaverini. Safety-related tasks within the set-based task-priority inverse kinematics framework. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6130-6135, IEEE, 2018.

[18] Roberto Simoni, Pere Ridao Rodríguez, Patryk Cieślak, and Dina Youakim. A novel approach to obstacle avoidance for an I-AUV. In *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, pages 1-6. IEEE, 2018.

[19] Patryk Cieślak, Roberto Simoni, Pere Ridao Rodríguez, and Dina Youakim. Practical formulation of obstacle avoidance in the Task-Priority framework for use in robotic inspection and intervention scenarios. *Robotics and Autonomous Systems*, 124 (2020): 103396, 2020.

[20] Zonggao Mu, Wenfu Xu, and Bin Liang. Avoidance of multiple moving obstacles during active debris removal using a redundant space manipulator. *International Journal of Control, Automation and Systems*, 15(2): 815-826, 2017.

[21] Abdullah Mohiuddin, Taha Tarek, Yahya Zweiri, and Dongming Gan. A survey of single and multi-UAV aerial manipulation. *Unmanned Systems*, 02(2020): 119-147, 2020.

[22] Bin Yang, Yuqing He, Jianda Han, and Guangjun Liu. Rotor-flying manipulator: modeling, analysis, and control. *Mathematical Problems in Engineering*, 2014: 1-14, 2014.

[23] Suseong Kim, Seungwon Choi, and H. Jin Kim. Aerial manipulation using a quadrotor with a two dof robotic arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4990-4995, IEEE, 2013.

[24] Guillermo Heredia, A. E. Jimenez-Cano, I. Sanchez, Domingo Llorente, V. Vega, J. Braga, J. A. Acosta, and Aníbal Ollero. Control of a multirotor outdoor aerial manipulator. In *2014 IEEE/RSJ international conference on intelligent robots and systems*, pages 3417-3422, IEEE, 2014.

[25] G Arleo, Fabrizio Caccavale, Giuseppe Muscio, and Francesco Pierri. Control of quadrotor aerial vehicles equipped with a robotic arm. In *21St mediterranean conference on control and automation*, pages 1174-1180, IEEE, 2013.

[26] Santhakumar Mohan and Jinwhan Kim. Indirect adaptive control of an autonomous underwater vehicle-manipulator system for underwater manipulation tasks. *Ocean Engineering*, 54: 233-243, 2012.

[27] Yuichiro Taira, Shinichi Sagara, and Masahiro Oya. A robust controller with integral action for underwater vehicle-manipulator systems including thruster dynamics. In *Proceedings of the 2014 International Conference on Advanced Mechatronic Systems*, pages 415-420, IEEE, 2014.

[28] Guohua Xu, Ying Guo, Xianbo Xiang, and Zhihu Xiao. Motion control and computer simulation for underwater vehicle-manipulator systems. In *2007 International Conference on Mechatronics and Automation*, pages 1368-1373, IEEE, 2007.

[29] Fabio Ruggiero, Miguel Angel Trujillo, Raul Cano, H. Ascorbe, Antidio Viguria, C. Peréz, Vincenzo Lippiello, Aníbal Ollero, and Bruno Siciliano. A multilayer control for multirotor UAVs equipped with a servo robot arm. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4014-4020, IEEE, 2015.

[30] Fabio Ruggiero, Jonathan Cacace, Hamid Sadeghian, and Vincenzo Lippiello. Impedance control of VToL UAVs with a momentum-based external generalized forces estimator. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 2093-2099, IEEE, 2014.

[31] Fabio Ruggiero, Jonathan Cacace, Hamid Sadeghian, and Vincenzo Lippiello. Passivity-based control of VToL UAVs with a momentum-based estimator of external wrench and unmodeled dynamics. *Robotics and Autonomous Systems*, 72 (2015): 139-151, 2015.

[32] Bin Xu, Shunmugham R. Pandian, Norimitsu Sakagami, and Fred Petry. Neuro-fuzzy control of underwater vehicle-manipulator systems. *Journal of the Franklin Institute*, 349(3): 1125-1138, 2012.

[33] Corina Barbălată, Matthew W. Dunnigan, and Yvan Pétillot. Dynamic coupling and control issues for a lightweight underwater vehicle manipulator system. In *2014 Oceans-St. John's*, pages 1-6, IEEE, 2014.

[34] Corina Barbalata, Matthew W. Dunnigan, and Yvan Petillot. Coupled and decoupled force/motion controllers for an underwater vehicle-manipulator system. *Journal of Marine Science and Engineering*, 6(3): 1-23, 2018.

[35] Han Han, Yanhui Wei, Lianwu Guan, Xiufen Ye, and Anqi Wang. Trajectory tracking control of underwater vehicle-manipulator systems using uncertainty and disturbance estimator. In *OCEANS 2018 MTS/IEEE Charleston*, pages 1-6, IEEE, 2018.

[36] Noboru Sugiyama and Masayoshi Toda. A nonlinear disturbance observer using delayed estimates-its application to motion control of an underwater vehicle-manipulator system. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2007-2013, IEEE, 2016.

[37] Jiyong Li, Hai Huang, Lei Wan, Zexing Zhou, and Yang Xu. Hybrid Strategy-based Coordinate Controller for an Underwater Vehicle Manipulator System Using Nonlinear Disturbance Observer. *Robotica*, 37(10): 1710-1731, 2019.

[38] Sibo Yang, T. O. Ting, Ka Lok Man, and Sheng-Uei Guan. Investigation of neural networks for function approximation. *Procedia Computer Science*, 17: 586-594, 2013.

[39] Qudrat Khan and Rini Akmeliawati. Neuro-adaptive dynamic integral sliding mode control design with output differentiation observer for uncertain higher order MIMO nonlinear systems. *Neurocomputing*, 226: 126-134, 2017.

[40] Hai Huang, Jiyong Li, Guocheng Zhang, Qirong Tang, and Lei Wan. Adaptive recurrent neural network motion control for observation class remotely operated vehicle manipulator system with modeling uncertainty. *Advances in Mechanical Engineering*, 10(10): 1-16, 2018.

[41] Panagiotis Sotiropoulos and Nikos Aspragathos. Neural networks to determine task oriented dexterity indices for an underwater vehicle-manipulator system. *Applied Soft Computing*, 49: 352-364, 2016.

[42] Pandurang S. Londhe, Mohan Santhakumar, Balasaheb M. Patre, and Laxman M. Waghmare. Task space control of an autonomous underwater vehicle manipulator system by robust single-input fuzzy logic control scheme. *IEEE Journal of oceanic engineering*, 42(1): 13-28, 2016.

[43] Yonghyun Kim, Santhakumar Mohan, and Jinwhan Kim. Task space-based control of an underwater robotic system for position keeping in ocean currents. *Advanced Robotics*, 28(16): 1109-1119, 2014.

[44] Santhakumar Mohan, Jinwhan Kim, and Yogesh Singh. A robust task space position tracking control of an underwater vehicle manipulator system. In *Proceedings of the 2015 Conference on Advances In Robotics*, pages 1-6, 2015.

[45] Santhakumar Mohan. Task space trajectory tracking control of an underwater vehicle-manipulator system under ocean currents. *NISCAIR-CSIR*, 10: 675-683, 2013.

[46] Pandurang S. Londhe, Mohan Santhakumar, Balasaheb M. Patre, and Laxman M. Waghmare. Robust nonlinear task space position tracking control of an autonomous underwater vehicle-manipulator system. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1713-1718, IEEE, 2015.

[47] Elisabetta Cataldi, Giuseppe Muscio, Miguel Angel Trujillo, Yamnia Rodríguez, Francesco Pierri, Gianluca Antonelli, Fabrizio Caccavale, Antidio Viguria, Stefano Chiaverini, and Aníbal Ollero. Impedance control of an aerial-manipulator: Preliminary results. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3848-3853, 2016.

[48] Joseph Jean-Baptiste Mvogo Ahanda, Jean Bosco Mbede, Achille Melingui, and Bernard Essimbi. Robust adaptive control for robot manipulators: Support vector regression-based command filtered adaptive backstepping approach. *Robotica*, 36(4): 8208-8213, 2018.

[49] T. K. Roy, L. C. Paul, M. F. Pervej, M. I. Sarkar, and F. K. Tumpa. Nonlinear adaptive backstepping controller design for trajectory flight control of uahs. In *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 69-74, IEEE, 2017.

[50] Fouad Yacef, Omar Bouhali, and Mustapha Hamerlain. Adaptive fuzzy backstepping control for trajectory tracking of unmanned aerial quadrotor. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 920-927, IEEE, 2014.

[51] Barmak Baigzadehnoe, Zahra Rahmani, Alireza Khosravi, and Behrooz Rezaie. On position/force tracking control problem of cooperative robot manipulators using adaptive fuzzy backstepping approach. *ISA transactions*, 70: 432-446, 2017.

[52] Seyed Mohammad Ahmadi, and Mohammad Mehdi Fateh. Task-space control of robots using an adaptive Taylor series uncertainty estimator. *International Journal of Control*, 92(9): 2159-2169, 2019.

[53] Donghyeon Lee, Woongyong Lee, Jonghoon Park, and Wan Kyun Chung. Task Space Control of Articulated Robot Near Kinematic Singularity: Forward Dynamics Approach. *IEEE Robotics and Automation Letters*, 2(5): 752-759, 2020.

[54] Reza Gholipour, and Mohammad Mehdi Fateh. Adaptive task-space control of robot manipulators using the Fourier series expansion without task-space velocity measurements. *Measurement*, 123: 285-292, 2018.

[55] Douglas R. Isenberg. A Task-Space Control Law for Free-Floating Space Robots. In *2017 25th International Conference on Systems Engineering (ICSEng)*, pages 33-38, 2017.

[56] S. Ali A. Moosavian and Evangelos Papadopoulos. Explicit dynamics of space free-flyers with multiple manipulators via SPACEMAPLE. *Advanced robotics*, 18(2): 223-244, 2004.

[57] Thor I Fossen. *Guidance and control of ocean vehicles*. John Wiley & Sons Inc, 1994.

[58] Gianluca Antonelli. *Underwater robots*. Switzerland: Springer International Publishing, 2014.

[59] G.E. Schubak, and D.S. Scott. A Techno-Economic Comparison of Power Systems for Autonomous Underwater Vehicles. *IEEE Journal Oceanic Engineering*, 20: 94-100, 1995.

[60] Scott McMillan, David E. Orin, and Robert B. McGhee. Efficient dynamic simulation of an underwater vehicle with a robotic manipulator. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(8): 1194-1206, 1995.

[61] Jiliang Wang, and John Y. Hung. Adaptive backstepping control for an underwater vehicle manipulator system using fuzzy logic. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 5600-5606, IEEE, 2018.

[62] Shao-Cheng Tong, Yong-Ming Li, Gang Feng, and Tie-Shan Li. Observer-based adaptive fuzzy backstepping dynamic surface control for a class of MIMO nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics)*, 41(4): 1124-1135, 2011.

[63] Qi Zhou, Peng Shi, Jinjun Lu, and Shengyuan Xu. Adaptive output-feedback fuzzy tracking control for a class of nonlinear systems. *IEEE Transactions on Fuzzy Systems*, 19(5): 972-982, 2011.

[64] Jiliang Wang, Edmon Perkins, and John Y. Hung. Trajectory Tracking Control for an Underwater Vehicle Manipulator System Using a Neural-adaptive Network. In *2019 SoutheastCon*, pages 1-6, IEEE, 2019.

[65] Vadim I. Utkin. *Sliding modes in control and optimization*. Springer Science & Business Media, 2013.

[66] Gianluca Antonelli, and Underwater Robots. *Motion and force control of vehicle-manipulator systems*. Springer Tracts in Advanced Robotics. Napoli, 2002.

[67] D. Swaroop, J. Karl Hedrick, Patrick P. Yip, and J. Christian Gerdes. Dynamic surface control for a class of nonlinear systems. *IEEE transactions on automatic control*, 45(10): 1893-1899, 2000.

[68] Wen-Jun Cao, and Jian-Xin Xu. Nonlinear integral-type sliding surface for both matched and unmatched uncertain systems. *IEEE Transactions on Automatic Control*, 49(8): 1355-1360, 2004.

[69] Francisco Javier Bejarano, Leonid Fridman, and Alex Poznyak. Output integral sliding mode control based on algebraic hierarchical observer. *International Journal of Control*, 80(3): 443-453, 2007.

[70] Francisco Javier Bejarano, Leonid Fridman, and Alex Poznyak. Output integral sliding mode for min-max optimization of multi-plant linear uncertain systems. *IEEE Transactions on Automatic Control*, 54(11): 2611-2620, 2009.

[71] Qudrat Khan, and Rini Akmeliawati. Neuro-adaptive dynamic integral sliding mode control design with output differentiation observer for uncertain higher order MIMO nonlinear systems. *Neurocomputing*, 226: 126-134, 2017.

[72] Dan Wang, and Jie Huang. Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form. *IEEE transactions on neural networks*, 16(1): 195-202, 2005.

[73] Tairen Sun, Hailong Pei, Yongping Pan, Hongbo Zhou, and Caihong Zhang. Neural network-based sliding mode adaptive control for robot manipulators. *Neurocomputing*, 74(14-15): 2377-2384, 2011.

[74] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1): 43-53, 1987.

[75] Michael Mistry, and Ludovic Righetti. Operational space control of constrained and underactuated systems. *Robotics: Science and systems*, 7: 225-232, 2012.

[76] Gianluca Antonelli, Stefano Chiaverini, and Nilanjan Sarkar. External force control for underwater vehicle-manipulator systems. *IEEE Transactions on Robotics and Automation*, 17(6): 931-938, 2001.

Appendices

Appendix A

Experiment details

In the experiment, a ThinkPad workstation P53 is used. The computer configuration is shown below.

Processor: Intel(R) Core(TM) i9-9800H

Graphics card: NVIDIA Quadro RTX 4000

Operating system: Windows 10 Home

Another HP computer is used as well to see if the configuration affects the communication speed. The configuration is shown below.

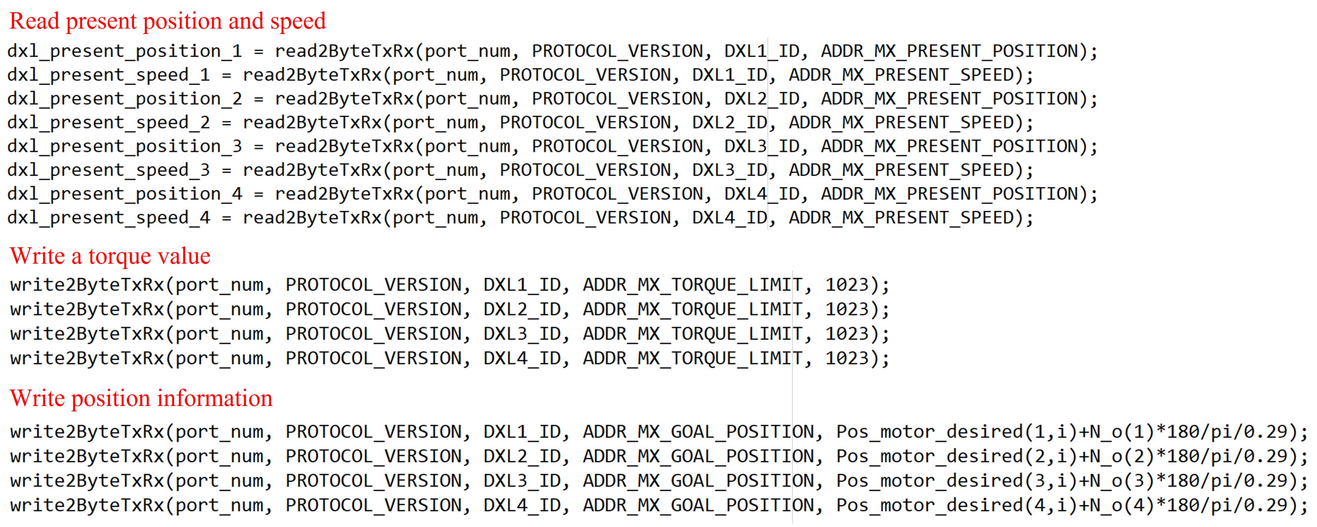
Processor: Intel i5-8250U

Graphics card: Intel UHD Graphics 620

Operating system: Windows 10 Home

The experiment results show that there is no difference between the two computers. Matlab R2019b is used for programming. In order to keep communication between the computer and servos, DYNAMIXEL SDK is added to the Matlab folder. DYNAMIXEL SDK is a software development kit that provides DYNAMIXEL control functions using packet communication, which can be downloaded from `https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_sdk/overview/`.

The sampling frequency is selected by adjusting the latency time of the USB port. In the experiments, the latency time is set to be 4 milliseconds. There are 16 read and write process in one loop, which corresponds to 64 milliseconds in one loop. Therefore, the sampling frequency is 16Hz. The read and write process are shown in Fig. A.1



Read present position and speed

```
dxl_present_position_1 = read2ByteTxRx(port_num, PROTOCOL_VERSION, DXL1_ID, ADDR_MX_PRESENT_POSITION);
dxl_present_speed_1 = read2ByteTxRx(port_num, PROTOCOL_VERSION, DXL1_ID, ADDR_MX_PRESENT_SPEED);
dxl_present_position_2 = read2ByteTxRx(port_num, PROTOCOL_VERSION, DXL2_ID, ADDR_MX_PRESENT_POSITION);
dxl_present_speed_2 = read2ByteTxRx(port_num, PROTOCOL_VERSION, DXL2_ID, ADDR_MX_PRESENT_SPEED);
dxl_present_position_3 = read2ByteTxRx(port_num, PROTOCOL_VERSION, DXL3_ID, ADDR_MX_PRESENT_POSITION);
dxl_present_speed_3 = read2ByteTxRx(port_num, PROTOCOL_VERSION, DXL3_ID, ADDR_MX_PRESENT_SPEED);
dxl_present_position_4 = read2ByteTxRx(port_num, PROTOCOL_VERSION, DXL4_ID, ADDR_MX_PRESENT_POSITION);
dxl_present_speed_4 = read2ByteTxRx(port_num, PROTOCOL_VERSION, DXL4_ID, ADDR_MX_PRESENT_SPEED);
```

Write a torque value

```
write2ByteTxRx(port_num, PROTOCOL_VERSION, DXL1_ID, ADDR_MX_TORQUE_LIMIT, 1023);
write2ByteTxRx(port_num, PROTOCOL_VERSION, DXL2_ID, ADDR_MX_TORQUE_LIMIT, 1023);
write2ByteTxRx(port_num, PROTOCOL_VERSION, DXL3_ID, ADDR_MX_TORQUE_LIMIT, 1023);
write2ByteTxRx(port_num, PROTOCOL_VERSION, DXL4_ID, ADDR_MX_TORQUE_LIMIT, 1023);
```

Write position information

```
write2ByteTxRx(port_num, PROTOCOL_VERSION, DXL1_ID, ADDR_MX_GOAL_POSITION, Pos_motor_desired(1,i)+N_o(1)*180/pi/0.29);
write2ByteTxRx(port_num, PROTOCOL_VERSION, DXL2_ID, ADDR_MX_GOAL_POSITION, Pos_motor_desired(2,i)+N_o(2)*180/pi/0.29);
write2ByteTxRx(port_num, PROTOCOL_VERSION, DXL3_ID, ADDR_MX_GOAL_POSITION, Pos_motor_desired(3,i)+N_o(3)*180/pi/0.29);
write2ByteTxRx(port_num, PROTOCOL_VERSION, DXL4_ID, ADDR_MX_GOAL_POSITION, Pos_motor_desired(4,i)+N_o(4)*180/pi/0.29);
```

Figure A.1: Read and write process