Motion Control of Autonomous Underwater Vehicles Using Advanced Model Predictive Control Strategy

by

Chao Shen
B.Eng., Northwestern Polytechnical University, 2009
M.A.Sc., Northwestern Polytechnical University, 2012

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Mechanical Engineering

Motion Control of Autonomous Underwater Vehicles Using Advanced Model Predictive Control Strategy

by

Chao Shen
B.Eng., Northwestern Polytechnical University, 2009
M.A.Sc., Northwestern Polytechnical University, 2012

Supervisory Committee

Dr. Yang Shi, Co-Supervisor
(Department of Mechanical Engineering)

Dr. Bradley Buckham, Co-Supervisor
(Department of Mechanical Engineering)

Dr. Lin Cai, Outside Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Yang Shi, Co-Supervisor
(Department of Mechanical Engineering)

Dr. Bradley Buckham, Co-Supervisor
(Department of Mechanical Engineering)

Dr. Lin Cai, Outside Member
(Department of Electrical and Computer Engineering)

ABSTRACT

The increasing reliance on oceans, rivers and waterways in a spectrum of human activities have demonstrated the large demand for advanced marine technologies that facilitate multifarious in-water services and tasks. The autonomous underwater vehicle (AUV) is a representative marine technology which has been contributing continuously to many ocean-related fields. An elaborate control system is essential to AUVs. However, AUVs present difficult control system design problems due to their nonlinear dynamics, the unpredictable environment and the poor knowledge about the hydrodynamic coupling of the vehicle degrees of freedom. When designing the motion controller, the practical constraints on the AUV system such as limited perceiving, computing and actuating capabilities should also be respected.

The model predictive control (MPC) is an advanced control technology that leverages optimization to calculate the control command. Thanks to the optimization nature, MPC can conveniently handle the complex nonlinearity in system dynamics as well as the state and control constraints. MPC takes the receding horizon control paradigm which gains satisfactory robustness against model uncertainties and external disturbances. Therefore, MPC is an ideal candidate for solving the AUV motion control problems. On the other hand, since the optimization is solved by iterative numerical algorithms, the obtained control signal is an implicit function of the system

state, which complicates the characterization of the closed-loop properties. Moreover, the nonlinear system dynamics makes the online optimization nonlinear programming (NLP) problems. The high computational complexity may cause an issue on the real-time control for embedded platforms with limited computing resources. In order to push the advanced MPC technology towards real-world AUV applications, this PhD dissertation is concerned with fundamental AUV motion control problems and attempts to address the aforementioned challenges and provide novel solutions.

This dissertation proceeds with Chapter 1 by providing state-of-the-art introductions to related research areas. The mathematical model used for the AUV motion control is elaborated in Chapter 2. In Chapter 3, we consider the AUV navigation and control problem in constrained workspace. A unified receding horizon optimization framework consisting of the dynamic path planning and the nonlinear model predictive control (NMPC) tracking control is developed. Although the NMPC tracking controller well accommodates the practical constraints on the AUV system, it presents a brand new design philosophy compared with the existing control systems that are implemented on real AUVs. Since the existing AUV control systems are reliable controllers, AUV practitioners tend not to fully replace them but to improve the control performance by adding features. By considering this, in Chapter 4, we develop the Lyapunov-based model predictive control (LMPC) scheme which builds on the existing AUV control system and invoke online optimization to improve the control performance. Chapter 5 focuses on the path following (PF) problem. Unlike the trajectory tracking control which equally emphasizes the spatial and temporal control objectives, the PF control often prioritizes the path convergence over the speed assignment. To incorporate this objective prioritization into the controller design, a novel multi-objective model predictive control (MOMPC) scheme is developed. While the MPC technique provides several salient features (e.g., optimality, constraints handling, objective prioritization, robustness, etc.), those features come at a price: a computational bottleneck is formed by the heavy burden of solving online optimizations in real time. To explicitly address this issue, in Chapter 6, the computational complexity of the MPC algorithms is particularly emphasized. Two novel strategies which potentially alleviate the computational burden of the MPC-based AUV tracking control are proposed. In Chapter 7, some conclusive remarks are provided and a few avenues for future research are identified.

Contents

Supervisory Committee ii

Abstract iii

Table of Contents v

List of Tables ix

List of Figures x

Acknowledgements xii

Dedication xiii

Acronyms xiv

1 Introduction 1

1.1 Autonomous Underwater Vehicle (AUV) 1

1.1.1 Overview 1

1.1.2 The Motion Control Problems 4

1.1.3 Literature review on AUV Motion Control 6

1.2 Model Predictive Control (MPC) 11

1.2.1 The Receding Horizon Control Strategy 11

1.2.2 Stability of MPC 12

1.3 Research Motivations and Contributions 14

1.4 Organizations of the Dissertation 17

2 AUV Modeling 19

2.1 Kinematics 19

2.1.1 Reference Frames 19

2.1.2 Transformation between Reference Frames 20
2.2 Nonlinear Dynamics of AUVs 22
2.2.1 Rigid-Body Dynamics 22
2.2.2 Hydrodynamic Forces and Moments 23
2.3 AUV Model for Motion Control 26
2.4 Conclusion 29

3 Receding Horizon Optimization for Integrated Path Planning and Tracking Control of an AUV 30
3.1 Introduction 30
3.1.1 Research Background and Contributions 30
3.1.2 Chapter Organization 32
3.2 Problem Statement 32
3.3 Minimum Curvature Receding Horizon Path Planning 34
3.4 Nonlinear Model Predictive Tracking Control 37
3.5 Integrated Receding Horizon Path Planning and Tracking Control: Algorithm Description 46
3.6 Simulation Results 48
3.6.1 Parameter Selection 48
3.6.2 Tracking Performance 49
3.7 Conclusion 51

4 Lyapunov-based Model Predictive Control for Dynamic Positioning and Trajectory Tracking Control of an AUV 52
4.1 Introduction 52
4.1.1 Research Background and Contributions 52
4.1.2 Chapter Organization 55
4.2 LMPC Design for Dynamic Positioning Control 55
4.2.1 Problem Formulation 55
4.2.2 Main Results 57
4.2.3 Simulation Results 62
4.3 LMPC Design for Trajectory Tracking Control 64
4.3.1 Problem Formulation 65
4.3.2 Main Results 69
4.3.3 Simulation Results 74

4.4 Conclusion 79

5 Multi-Objective Model Predictive Control for Path Following Control of an AUV 82

5.1 Introduction 82
5.1.1 Research Background and Contributions 82
5.1.2 Chapter Organization 85
5.2 Problem Formulation 85
5.2.1 The Path Following Problem 86
5.2.2 Zero-Path-Error Manifold 87
5.2.3 The MOMPC Formulation 88
5.3 Solve the MOMPC Problem 90
5.3.1 Weighted Sum Method 90
5.3.2 Lexicographic Ordering Method 92
5.4 Convergence Analysis 95
5.5 Simulation Results 101
5.5.1 PF Control Using WS-MOMPC 102
5.5.2 PF Control Using LO-MOMPC 103
5.5.3 Robustness Test 106
5.6 Conclusion 108

6 Efficient Implementation Algorithms for NMPC Trajectory Tracking Control of an AUV 111

6.1 Introduction 111
6.1.1 Research Background and Contributions 111
6.1.2 Chapter Organization 113
6.2 Modified C/GMRES Algorithm 114
6.2.1 Problem Formulation 114
6.2.2 Solving the NMPC Problem 116
6.2.3 Modified C/GMRES Algorithm 117
6.2.4 Simulation Results 122
6.3 Distributed Implementation Strategy 128
6.3.1 Problem Formulation 129
6.3.2 Solving the NMPC Problem 130
6.3.3 Distributed NMPC Implementation 132

6.3.4 Simulation Results 138
6.4 Conclusion 142

7 Conclusions and Future Work **144**
7.1 Conclusions 144
7.2 Future Work 146

A Forward Difference Generalized Minimal Residual Method **148**

B Derivation of Jacobians **150**

C Publications **152**

Bibliography **154**

List of Tables

Table 2.1 Hydrodynamic coefficient summary. 28

Table 4.1 Mean square errors for AUV tracking with disturbances - Case I. 81
Table 4.2 Mean square errors for AUV tracking with disturbances - Case II. 81

Table 6.1 Average computation time (*sec.*) per update - Case I. 129
Table 6.2 Average computation time (*sec.*) per update - Case II. 129
Table 6.3 Average computation time (*sec.*) per update - Case III. 129
Table 6.4 Average computation time (*sec.*) per update. 140
Table 6.5 Mean square errors for AUV tracking with disturbances. 143

List of Figures

Figure 1.1 The REMUS AUV [7]. 3
Figure 1.2 The Ictineu AUV [114]. 4
Figure 1.3 The motion variables of an AUV. 5

Figure 2.1 The reference frames for AUV motion control. 20

Figure 3.1 Illustration of the combined AUV motion control problem. . . . 33
Figure 3.2 The closed-loop control block diagram. 48
Figure 3.3 Simulation results of the combined AUV problem. 49
Figure 3.4 The generalized control signal. 50
Figure 3.5 The real thrust forces. 50

Figure 4.1 The AUV trajectory in local level plane. 62
Figure 4.2 The state trajectories. 63
Figure 4.3 The control input signals. 64
Figure 4.4 The AUV trajectory in local level plane (with disturbance). . . 64
Figure 4.5 The state trajectories (with disturbance). 65
Figure 4.6 The control input signals (with disturbance). 65
Figure 4.7 The AUV trajectory in local level plane - Case I. 76
Figure 4.8 The state trajectories - Case I. 76
Figure 4.9 The control input signals - Case I. 77
Figure 4.10The AUV trajectory in local level plane - Case II. 77
Figure 4.11The state trajectories - Case II. 78
Figure 4.12The control input signals - Case II. 78
Figure 4.13The AUV trajectory in local level plane (with disturbance) - Case I. 79
Figure 4.14The state trajectories (with disturbance) - Case I. 79
Figure 4.15The AUV trajectory in local level plane (with disturbance) - Case II. 80
Figure 4.16The state trajectories (with disturbance) - Case II. 80

Figure 5.1 The AUV PF results with WS-MOMPC. 102
Figure 5.2 Surge velocity of the AUV (WS-MOMPC). 103
Figure 5.3 Control inputs of the augmented system (WS-MOMPC). 104
Figure 5.4 The AUV PF results using LO-MOMPC. 104
Figure 5.5 Surge velocity of the AUV (LO-MOMPC). 105
Figure 5.6 Control inputs of the augmented system (LO-MOMPC). 105
Figure 5.7 The optimal value functions. 106
Figure 5.8 The AUV PF results (with parametric uncertainties). 107
Figure 5.9 Surge velocity of the AUV (with parametric uncertainties). . . . 107
Figure 5.10PF results with non-differentiable path. 108
Figure 5.11Surge velocity of the AUV (non-differentiable). 109
Figure 5.12Control inputs of the augmented system (non-differentiable). . . 109

Figure 6.1 Simulated AUV trajectories - Case I. 124
Figure 6.2 The control forces and moments - Case I. 125
Figure 6.3 The position error - Case I. 125
Figure 6.4 Simulated AUV trajectories - Case II. 126
Figure 6.5 The control forces and moments - Case II. 126
Figure 6.6 The position error - Case II. 127
Figure 6.7 Simulated AUV trajectories - Case III. 127
Figure 6.8 The control forces and moments - Case III. 128
Figure 6.9 The position error - Case III. 128
Figure 6.10The AUV trajectory in the local level plane. 139
Figure 6.11The state trajectories. 140
Figure 6.12The control input signals. 141
Figure 6.13The AUV trajectory in the local level plane (with disturbance). 141
Figure 6.14The state trajectories (with disturbance). 142
Figure 6.15The control input signals (with disturbance). 142

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my supervisors Dr. Yang Shi and Dr. Bradley Buckham who have been continuously helping me on my research, through their expertise in the areas of control theory and marine technology. They have been supporting me by efficiently dealing with practical issues such as paper revision and financing, by providing various opportunities, and by being fantastic friends. They set excellent examples of the way to work professionally and selflessly. What I have learnt from them grew me not only as a researcher but also as a person. The experience of working together with them benefited and will continue to benefit me in the rest of my life.

I would also like to thank the supervisory committee member Dr. Lin Cai and the external member Peter X. Liu for their time and patience reviewing this dissertation and providing constructive comments and suggestions.

I would like to give my appreciation to the present and former colleagues in the Applied Control and Information Processing Lab at the University of Victoria. Looking back, I feel very lucky to be a teammate of Ji Huang, Jian Wu, Huiping Li, Xiaotao Liu, Mingxi Liu, Bingxian Mu, Yuanye Chen, Yiming Zhao, Jicheng Chen, Kunwu Zhang, Xiang Sheng, Qian Zhang, Qi Sun, Zhang Zhang, Yuan Yang, Chen Ma, Huaiyuan Sheng, Henglai Wei and Tingting Yu. Those wonderful days that I had with you guys will be the most precious memories and will never fade away.

Special thanks goes to my beautiful wife Binbin, my lovely son Charlie and my little cat Yin. Having them accompanied, this journey was much less stressful. I want to thank my parents, parents-in-law and all the family members for their constant understanding, encouragement and support. They have given me the complete freedom to set my own path in life, and to them I dedicate this dissertation.

Finally, I gratefully acknowledge the financial support from the Chinese Scholarship Council, the Natural Science and Engineering Council of Canada, Canada Foundation for Innovation, the Department of Mechanical Engineering and the Faculty of Graduate Studies at the University of Victoria.

Victoria, BC, Canada
November, 2017

To my family

Acronyms

AUV autonomous underwater vehicle

BRF body-fixed reference frame

BSC backstepping control

CB center of buoyancy

CG center of gravity

C/GMRES continuation/generalized minimal residual

DMPC distributed model predictive control

DOB disturbance observer

DOF degree of freedom

DP dynamic positioning

DSC dynamic surface control

ECEF Earth-centered Earth-fixed

ECI Earth-centered inertial

FDGMRES forward difference generalized minimal residual

flop floating-point operation

GMRES generalized minimal residual

IRF inertial reference frame

IP interior-point

KKT Karush-Kuhn-Tucker

LICQ linear independent constraint qualification

LMPC Lyapunov-based model predictive control

LO lexicographic ordering

LO-MOMPC lexicographic ordering based multi-objective model predictive control

LOS line-of-sight

LQG linear quadratic Gaussian

LQR linear quadratic regulator

LTR loop transfer recovery

mC/GMRES modified continuation/generalized minimal residual

mDMPC modified distributed model predictive control

MIMO multiple-input multiple-output

MOMPC multi-objective model predictive control

MPC model predictive control

MSE mean square error

NED North-East-Down

NLP nonlinear programming

NMPC nonlinear model predictive control

NN neural network

OCP optimal control problem

PD proportional-derivative

PF path following

PID proportional-integral-derivative

PMP Pontryagin's minimum principle

PO Pareto optimal

QP quadratic programming

RHO receding horizon optimization

ROA region of attraction

ROV remotely operated vehicle

SMC sliding mode control

SQP sequential quadratic programming

TA thrust allocation

TRD trust-region-dogleg

UAV unmanned aerial vehicle

UUB uniform ultimate bounded

UUV unmanned underwater vehicle

WS weighted sum

WS-MOMPC weighted sum based multi-objective model predictive control

ZPE zero-path-error

Chapter 1

Introduction

This chapter provides some introductory knowledge about the autonomous underwater vehicle (AUV), a literature review of motion control of AUVs and a brief review of model predictive control (MPC). It also summarizes the motivations and main contributions of this dissertation. The organization of the dissertation is presented at the end of this chapter.

1.1 Autonomous Underwater Vehicle (AUV)

1.1.1 Overview

Oceans cover two thirds of the Earth and have a huge impact on our ecosystem. Traditionally they act as the source of food, provide warmth and natural resources, and sustain the ocean ecosystem by maintaining biodiversity. With the development of ocean science, their ecological, economic and social importance are now better understood. On the other hand, ocean activities are closely related to some deadly natural phenomenons such as tsunami, earthquake and hurricane. Hence the constant monitoring of the ocean state becomes an urgent necessity and will definitely benefit mankind in terms of minimizing the loss due to natural disasters, maximizing the harvest from the oceans, and more.

Underwater vehicles present advanced tools that enable the ocean monitoring to go far beneath the ocean surface, collect diverse first-hand data and see how the oceans behave. Underwater vehicles can be manned or unmanned. Clearly, the manned submarine technology was firstly focused. Since 1962 when the first submarine was constructed [116], dramatic progress has been made in the design and manufacturing

of manned underwater vehicles. However, the intrinsic weakness of reliance on human pilots limits its applications. In contrast, advances in navigation, control, computer, sensor and communication technologies have turned the idea of unmanned underwater vehicle into reality.

The unmanned underwater vehicles (UUVs) can be categorized into two groups:

- Remotely operated vehicles (ROVs) are tethered underwater robotic vehicles. ROVs require instructions from human operators who locate in the support vessel during the execution of tasks. An umbilical cable is therefore needed to carry power, relay control signals, and transmit sensor data.

- Autonomous underwater vehicles (AUVs) are tether-free underwater robotic vehicles. AUVs are powered by onboard batteries or fuel cells, equipped with navigation sensors, and execute preprogrammed missions without being constantly supervised or controlled by humans.

In recent years, UUVs have achieved great success in many ocean-related scientific fields such as marine geoscience [139, 142, 143, 11], offshore industry [138, 100, 37] and deep-sea archaeology [16, 117]. Compared to ROVs, AUVs have higher level of autonomy and demonstrate the following strengths:

- AUVs have much wider reachable scope. AUVs are more mobile platforms and can execute oceanic missions that need to travel a long distance, e.g., polar region survey beneath the ice sheet, or that need to be performed in dangerous areas, e.g., submarine volcanism data acquisition.

- AUVs avoid many technical issues related to the tether cable. The chaotic drag force induced by the cable makes the vehicle difficult to control. The drag force will become unmanageable as the tether length increases. Moreover, the communication latency greatly influences the control of the vehicle. In contrast, the absence the umbilical cable enables the AUV to achieve real-time control.

- AUVs reduce the operational cost. Unlike ROVs which need human operators to perform the task, AUVs require the minimum amount of human intervention. Therefore, it is likely to cut a large portion of the operational cost as the number of staff needed is reduced.

On the other hand, the absence of tether cable brings challenges in the power supply, underwater navigation and automatic control aspects. With the developments of new technologies in these areas, AUVs have extensive application prospect.

There are two configurations for the shape design of AUVs:

- The conventional slender body AUVs have efficient hydrodynamic properties and are best suited for oceanic missions that need to travel with a high speed or a long distance. These conventional AUVs are usually equipped with propellers to drive in the direction of the principal axis and with control surfaces (rudders) to perform maneuvers. Therefore, they have lower number of control variables than the motion DOF (i.e., underactuated). Hence they are easy to control only along straight lines. Examples of AUVs with this configuration include REMUS [7] and ODYSSEY [34].

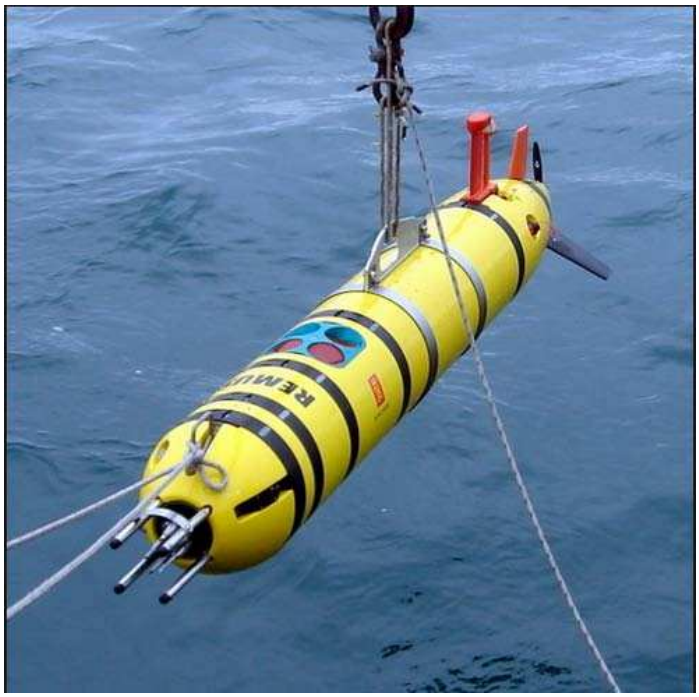

Figure 1.1: The REMUS AUV [7].

- The design of open-frame AUVs was recently borrowed from ROVs in order to enable the omni-directional mobility. This configuration usually contains redundant thrusters to provide more control DOF than the motion, hence the AUV can perform good low-speed maneuvers in cluttered environments. With the omni-directional mobility and the possible artificial intelligence, AUVs are potentially capable of performing complicated jobs. Examples of AUVs with this configuration include Ictineu [114] and Smart-E [91],

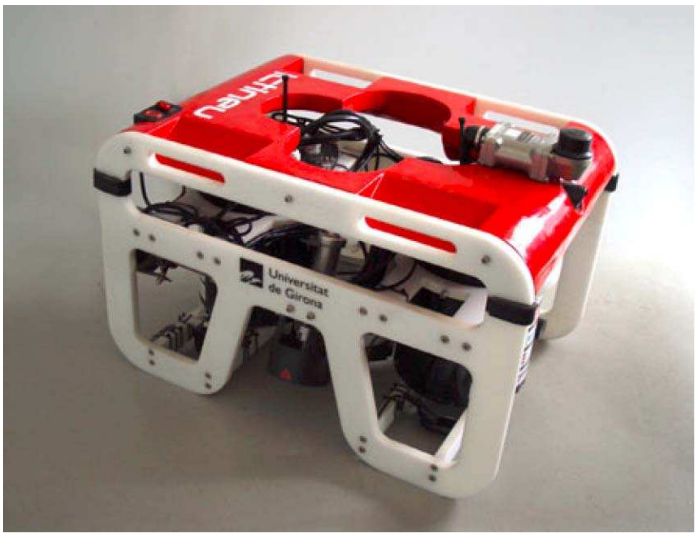

Figure 1.2: The Ictineu AUV [114].

The control system acts as the brain of the AUV and is responsible for the autonomy of the vehicle. The term *control* here actually has a broad sense, including but not limited to (i) motion control: the low-level system control, focusing on input/output of the vehicle and the closed-loop properties; (ii) mission control: the high-level behavioral control which is usually predefined and triggered by sensor measurement; (iii) power management: the control that aims at optimally distributing the onboard power, and even recharging from solar power [63]; (iv) cooperative control of multiple vehicles: the control that emphasizes on coordinated behaviors among a group of AUVs, e.g. formation control. Among all of these control categories, although it is hard to distinguish one specific type to be more important than the others, there exists little controversy over the statement: *Motion control is the most fundamental research study for the control of AUVs*. High-level mission controls or cooperative controls could only be realized through the motion control of each individual AUV.

1.1.2 The Motion Control Problems

The motion of an AUV in the three dimensional workspace can be described in six degrees of freedom (DOFs). The six independent variables which uniquely determine the motion of the vehicle are known as 'surge', 'sway', 'heave', 'roll', 'pitch' and 'yaw' (see Figure 1.3). The motion control of the AUV aims to regulate the motion variables to the desired values, i.e., the set-points which are determined by the high-level motion planning system. According to different types of set-points the motion

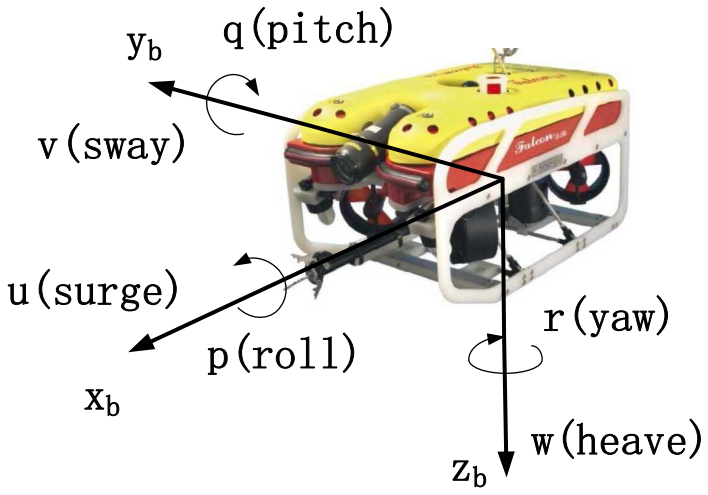

Figure 1.3: The motion variables of an AUV.

control problems can be classified into three categories:

- When the set-points are time-invariant, it is a point stabilization problem. In context of AUV motion control, the heading control, depth control and dynamic positioning control belong to this category. Specifically, the dynamic positioning refers to the automatic control of AUV to reach and then maintain its position and orientation at a fixed point, possibly with external disturbances such as waves and ocean currents.

- When the set-points are time-varying, it is a trajectory tracking problem. The trajectory tracking controller steers the AUV state (pose and velocity) to converge and then track the desired trajectory which is calculated by the high-level motion planner.

- When the set-points are time independent but only describe the geometric relationship among them, it is a path following problem. The path following control refers to the automatic control that moves the AUV along a specified path in the workspace, but there is no requirement on when should the AUV be where.

The AUV motion control problems can be solved at two levels:

- The velocity level solution. The velocities of the AUV are viewed as the control inputs. The motion controller determines the desired linear and angular velocities that will achieve the control objective, and the control of thrusters that

generate the desired velocities is assumed to be solved readily. At this level, only the kinematic equations of the AUV motion are considered.

- The force level solution. The forces and moments that cause the AUV motion are regarded as the control inputs. The motion controller determines the desired propulsive force for each thruster according to the AUV state and the control objective. At this level, both kinematic equations and dynamic equations of the AUV motion are considered.

The two levels of solutions reflect the design trade-off between precision and complexity. Since the AUV dynamics are not considered, the control algorithm design for the velocity level solution can be simplified significantly. For the same reason, however, the velocity level solution is not precise especially for the trajectory tracking and path following applications. In contrast, the force level solution has high precision. However, since it considers the dynamic equations of motion, the modeling work is required for the AUV. Moreover, the identified AUV dynamic model is highly nonlinear and possibly time-varying, which makes the control algorithm design very complicated or even intractable.

1.1.3 Literature review on AUV Motion Control

Motion control of marine vehicles has been an active field of research since 1911 in which year the first autopilot was constructed by Elmber Sperry [46]. Early automatic control systems employed empirical proportion-derivative (PD) and subsequently proportion-integral-derivative (PID) control to steer the marine vessel on the desired course. Not until 1970s, when the underwater navigation technology had become mature enough such that 1 meter positioning precision and 1 Hz update rate could be achieved [62], was the closed-loop motion control enabled for the underwater vehicles.

The AUVs present very challenging control system design problems. The technical challenges mainly come from the following aspects:

- The highly nonlinear system dynamics and the multiple input multiple output (MIMO) nature of the motion control problem.

- The considerable parametric uncertainties caused by poor knowledge of the hydrodynamic coefficients.

- The unpredictable external disturbances in terms of end-effector payloads, waves and ocean currents.
- The practical constraints on the real AUV system such as underactuation, over-actuation, finite sensing and actuating capabilities, etc.

Therefore, advanced AUV motion control designs attempt to tackle these issues. Some of the recent progress are reviewed in the following.

1. Dealing with the nonlinearity

Linear control theory has evolved a variety of powerful tools, and the design challenge due to the nonlinearity can be circumvented by applying the linear control methods to a linearized AUV model provided that the conditions for linearization can be satisfied: the pitch and roll angels are small, and the forward speed is constant.

In 1990s, Healey and Marco [59] proposed a decoupled design paradigm which claims that the 6 DOF linearized equations of motion can be divided into three weakly-interacted subsystems for the forward speed, steering and diving control, respectively. This design paradigm dramatically simplifies the linear control design for the AUV motion control and inversely directs the AUV mechanical design. The PID control is universally applied to AUVs due to its good robustness and easy implementation [46]. Acceleration feedback technique which enables the inertia shaping can be incorporated into the conventional PID control design in an attempt to get better control stability [81]. The linear quadratic regulator (LQR) control is a common alternative to PID control and is theoretically optimal in some sense (specifically, with respect to the performance index). In [99], the LQR controller is designed in combination with the Kalman filtering technique to solve the line tracking problem for the AUV.

However, the linearized model only approximates well the nonlinear behavior of the AUV motion around the predefined working point. In applications that involve tracking of curved trajectories, the linear control methods appear inappropriate as the curved trajectory itself emphasizes the nonlinearity in the AUV motion. In these cases, the nonlinear control methods should be applied.

Feedback linearization [127] is a powerful tool to deal with the nonlinearity. Since the AUV dynamic model can be arranged in the control-affine form, i.e., $\dot{x} = f(x) + g(x)u$, the feedback linearization technique can be employed by setting $u = -f(x)/g(x) + \bar{u}/g(x)$. In [140], the feedback linearization based controller is designed for the AUV to track the subsea cables. The main problem associated with

this control method is that it requires a high fidelity dynamic model of the AUV, however, the hydrodynamic coefficients are rarely accurately estimated. Therefore, the Lyapunov-based backstepping control (BSC) becomes the mainstream nonlinear control method for the design of AUV motion controller. The developed BSC control law exploits the good nonlinearities in the system dynamics, such as the damping term, to gain additional robustness. Examples include [36] and [113]. In [113], the velocity level tracking controller is firstly designed for the AUV, and the force level control law is then derived from it via the backstepping technique. The output feedback control variant is presented in [36]. The Lyapunov-based BSC sometimes suffers from the problem of "explosion of terms" in the derived control law, which motivates the designs of dynamic surface controllers (DSC) for AUVs [53].

2. Dealing with the parametric uncertainties

The sliding mode control (SMC) is another well-studied nonlinear control method for AUVs. The charm of SMC lies in its insensitivity to parameter uncertainties in the system model. By forcing the nonlinear system to slide along a predesigned reduced-order subspace, the tracking error can be eliminated in finite time. In [58], the precise diving and steering control is achieved using SMC controllers. However, SMC controllers apply discontinuous control laws, which results in the main drawback of SMC: the chattering problem (i.e, control signals switch signs too frequently). The chattering asks for an infinite communication bandwidth and wears out the actuator parts. Therefore, this issue must be addressed in real world AUV applications. In [131] an adaptive term is designed and added to the conventional SMC control law so that the chattering can be mitigated. The idea of using the higher order sliding mode to eliminate the chattering is reported in [119]. In [132], the trajectory tracking control integrates the SMC, PID and robust control techniques, and enhanced tracking performance is obtained.

Another effective treatment to deal with the model mismatch is to incorporate an adaptation mechanism which online corrects the parameters. This method has been successfully applied to the AUV motion control. In [48], the model uncertainty due to partly known nonlinear thruster dynamics is considered. An adaptive passivity-based controller and a combined adaptive and sliding mode controller are proposed. The simulation results demonstrate satisfactory tracking performance. In [85], the depth and pitch control design using L_1 adaptive control is reported, and the improvement

of the robustness and the adaptation rate are shown. Neural network (NN) based control schemes integrate the parameter identification with the control and can be viewed as a special type of adaptive control technique. In [148], Yuh proposed a multilayered forward network. The position and velocity error signals are used as the inputs of the NN, and the outputs are the propulsive forces and moments. Simulation results demonstrate the better tracking performance compared to an adaptive SMC controller. In [25], a similar approach based on reinforcement learning is exploited for the motion control of AUV. With the fast development of artificial intelligence, the deep network and deep learning based control continues to be an appealing option for the AUV motion control [52]. From a control theoretic point of view, however, the main drawback of the NN-based control design goes to the difficulty in characterizing the closed-loop system's behavior. As the NN does not take the kinematic and dynamic equations of motion as the AUV model, the validation of control design can only be demonstrated experimentally, but without a theoretical guarantee.

3. Dealing with the external disturbances

Although the PID, BSC and DSC controllers have moderate robustness margin against external disturbances, some specific AUV applications may emphasize the explicit rejection of external disturbances. In this case, robust control methods can be applied. The H_∞ loop shaping minimizes the sensitivity of a system and ensures that the closed-loop system will not deviate too much from expected trajectories in the presence of external disturbances. In [96], the H_2 and H_∞ design is applied to the diving and heading control of an AUV. The first- and second-order wave force disturbances are considered and rejected. The H_∞ control design optimizes the disturbance rejection assuming that the disturbance is bounded. This is a rather simplified assumption. The sea disturbances are usually stochastic, and if the probability distribution can be captured it is expected to have better disturbance rejection in the motion control. The well-known linear-quadratic-Gaussian (LQG) control deals with the stochastic disturbances. In [135], the loop transfer recovery (LTR) technique is applied to enhance the robustness of the LQG control. The effectiveness of the LQG/LTR design is verified through experiments.

When dealing with the external disturbance, an important alternative to robust control is to use a disturbance observer (DOB) [26]. The DOB-based control estimates the external disturbance explicitly and then compensate it in the control design. In

[97] a nonlinear DOB-based PD controller is proposed for the tracking control of the AUV. The robust stability under the observer-controller structure is proved. In principle, the DOB can be incorporated in any of the aforementioned control method to enhance the robustness. In context of AUV motion control, the examples can be found in [151] for DOB-based adaptive control, in [83] for DOB-based BSC control, and in [27] for DOB-based SMC control.

4. Dealing with practical constraints

The conventional slender body AUVs are underactuated. They are subject to nonholonomic constraints [104], which brings additional challenges to AUV motion control problems. For underactuated AUVs, the dynamic model cannot be fully feedback linearized, so the controllers are designed via the Lyapunov direct method and usually with the backstepping procedure (BSC). The three dimensional trajectory tracking problem is well studied for underactuated AUVs in [1], and the result has been extended in [2] by adding an adaptive supervisory control to the BSC controller to tackle the parameter variation. In [141], the combination of BSC and SMC enhances the robustness of the AUV tracking control in the presence of parametric uncertainties and environmental disturbances. For the point stabilization of an underactuated AUV, a hybrid control law with a logic-based switching is proposed in [4]. Global uniform stability is obtained. In [3], a non-smooth coordinate transformation is introduced and followed by backstepping procedure to design a smooth control law in the new coordinate system. An adaptive control law is then provided to make the controller robust against parametric uncertainties.

The open-frame AUVs are typically with redundant thruster arrangement which makes them overactuated. The thrust allocation (TA) has to be considered to deal with the overactuation. One prominent approach is the 2-norm based optimization. The pseudo-inverse method [50] is cheap in computation but barely adequate to guarantee the feasibility. Therefore, the TA is usually formulated as quadratic programming (QP) problems which explicitly take into account the individual limit on each thruster. To alleviate the computational burden, parametric QP solution can be adopted [65]. When the thrusters are rotatable, the TA optimization is nonlinear. The direct nonlinear programming (NLP) solution is studied in [107]. In [66], a piecewise linear approximation is made for the NLP, and the results obtained in [65] is extended with the azimuth angle considered as another decision variable. Other ap-

proaches include the 1-norm minimization [28] and infinity-norm minimization [131]. A noticeable variant can be found in [64] where a dynamic update law is proposed, instead of static optimization. The asymptotical stability is proved given that an exponentially stable trajectory tracking control law is working.

Other practical issues such as limited sensing and actuating capability [123], limited computing resources [121], actuator faults [130], lack of velocity measurements [112] and communication delay [40] are also addressed in the AUV motion control.

1.2 Model Predictive Control (MPC)

Optimal control is an important research direction in control engineering and applied mathematics. Early theocratical results mainly include Bellman's principle of optimality [12], Pontryagin's minimum principle [21] and linear quadratic regulator [69]. However, these control methods cannot handle system constraints on state and/or control variables, which evokes the keen interest in studying the model predictive control (MPC). On the other hand, in many industries such as petrochemical industry, the requirement of optimal process control to chase a maximum profit stimulates the growth of MPC since the optimum can often be obtained near or on the boundary of the operational region, and the system constraints have to be considered.

1.2.1 The Receding Horizon Control Strategy

Generally speaking, MPC is a control strategy which determines the control action by recursively solving finite horizon optimal control problems (OCPs) and respects the system constraints during the control [88, 87]. Consider a general nonlinear system:

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \tag{1.1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the underlying nonlinear dynamics. Without loss of generality, the origin is assumed to be the equilibrium of interest, and the control objective is to steer the system state to the origin.

The finite horizon OCP is defined as follows,

$$\begin{aligned} \min_{u} . \quad & J(x,u) = \int_0^T \ell(x(s),u(s))ds + g(x(T)) \\ \text{s.t.} \quad & \dot{x}(s) = f(x(s),u(s)) \\ & x(0) = x_0 \\ & x(s) \in \mathbb{X} \\ & u(s) \in \mathbb{U} \end{aligned} \tag{1.2}$$

where $J(x,u)$ is the cost function consisting of the stage cost $\ell(x,u)$ and the terminal cost $g(x)$; T is the prediction horizon and x_0 is the current measured system state; $\mathbb{X}$ and $\mathbb{U}$ are compact sets, representing the system constraints on the state and on the control, respectively.

The MPC is implemented in a receding horizon control paradigm which can be briefly described as follows:

- At the sampling time instant, the OCP (1.2) is solved, which obtains the solution curve $u^*(s)$, $s \in [0,T]$.
- The first portion of the solution curve, $u^*(s)$ for $s \in [0,\Delta t]$, is actually implemented to control the nonlinear system, where Δt is the sampling period.
- At the next sampling instant, the OCP (1.2) will be solved again with the system state measured and used as the new initial condition.

Since MPC is realized using digital computers, the OCPs need to be discretized and then solved by iterative numerical algorithms.

1.2.2 Stability of MPC

The MPC is implemented by recursively solving finite horizon OCPs. While the finite time horizon makes the solving of OCPs numerically tractable, it throws the closed-loop stability into question. Optimality does not necessarily lead to stability. As shown in [21], even for linear systems with no constraints, the finite horizon LQR can be destabilizing. Similar situation occurs in MPC as well.

1. A brief review of stability results in general MPC

Early days, however, the closed-loop stability was obtained in most process control applications, certainly after tuning. This is because the prediction horizon was nor-

mally sufficiently long. In [108], Primbs *et al.* showed that for linear systems, the length of a stabilizing prediction horizon can be pre-computed even in the presence of state and control constraints. For nonlinear constrained systems, Alamir *et al.* [5] established the asymptotical stability for a given region of attraction with a sufficiently long prediction horizon and a shorter control horizon.

On the other hand, the computational complexity of the OCP increases exponentially as the prediction horizon increases. In many applications such as AUV motion control, it is not affordable to use a too long prediction horizon. Since the closed-loop stability is an overriding requirement, during the past forty years, control theorists and practitioners have been devoting a significant amount of effort to the development stable MPC schemes that do not rely on long prediction horizons. Among the existing results, seeking stabilizing conditions using the terminal cost, terminal constraints, and an associated terminal controller (known as the terminal triple) is the most popular approach [88]. Terminal equality constraint was firstly imposed. Keerthi *et al.* demonstrated in [72] that the optimal value function of the infinite horizon OCP can be approached by that of a finite receding horizon approximation with the terminal state constraint $x(T) = 0$. Imposing the terminal equality constraint, however, adversely affects the feasibility of the OCP, and in turn requires a long prediction horizon, which is undesired. Later, therefore, the dual-mode MPC scheme [29, 120] were proposed. In the dual-mode MPC scheme a terminal set constraint $x(T) \in X_f$ is used in place of the former equality constraint. Once the system state x enters the terminal set X_f, a local feedback controller $\kappa_f(x)$ will take over the stabilization task and steer the system state to the origin. However, the implementation of the dual-mode MPC is complicated and it also loses some degree of optimality. Therefore, in the most recent MPC proposals, the terminal triple is used together to establish the stability condition. In an excellent MPC review paper [88], Mayne *et al.* summarized the stabilizing conditions in most MPC proposals and distilled the widely accepted principle of stability in terms of four mild assumptions. Once the four assumptions are satisfied, the optimal value function of the OCP can be shown a valid Lyapunov function for the nonlinear system, hence guarantees the closed-loop stability.

In the new century, with the significant development of stabilizing conditions without imposing a terminal constraint [87], there emerges an interesting discussion inside the MPC community: on the one hand, the conservativeness of MPC can be fairly relaxed [55] if there is no terminal constraints; on the other hand, the necessity of a terminal constraint is emphasized in [86] for guaranteeing the recursive feasibility.

Other MPC schemes which do not exploit the terminal triple mainly include variable horizon MPC [94], contractive MPC [84] and Lyapunov-based MPC [82].

2. Stability results in MPC for AUV Motion Control

While a number of MPC solutions have been proposed for AUV motion control problems, e.g., [102],[106],[136],[100], none of them include the stability analysis. This is partly due to the complicated nonlinear dynamic model of the AUV, and partly due to the fact that the MPC theoretical papers focus exclusively on the point stabilization problems while for AUV motion control we may be more interested in the trajectory tracking and path following problems.

In our published work [123], the stability of an MPC based trajectory tracking controller is explicitly discussed for the first time. By defining an appropriate reference system, we converted the trajectory tracking control problem to the stabilization problem of the error dynamics and then followed the principle of MPC stability [88] to derive the sufficient conditions for guaranteeing the closed-loop stability. In [126], we took a Lyapunov-based MPC strategy to deal with AUV trajectory tracking control. The closed-loop stability is guaranteed by imposing a contraction constraint derived from a Lyapunov-based control law. The design of a stable MPC controller for the AUV dynamic positioning problem is studied in [124], and the MPC solution to the AUV path following control problem is discussed in [125].

1.3 Research Motivations and Contributions

Although a significant amount of effort has been devoted to the study of AUV motion control, due to the complexity of the control problem itself, there still exist many unsolved issues in this area. One prominent issue is that the practical constraints on the real AUV system such as limited perceiving, computing and actuating capabilities are seldom considered in the controller design, because of the inherent limitations of those conventional control methods.

The model predictive control (MPC) presents a powerful framework for solving a broad spectrum of control engineering problems and has the capability of handling practical constraints in a systematic manner. However, due to its heavy computational burden, for many years, MPC had been applied to process control problems with slow dynamics only. With the development of computer technology and opti-

mization theory, now the computational barriers have been largely removed. The successful real-time implementation of MPC for unmanned aerial vehicles (UAVs) [10] and mobile robots [61] strongly suggests that the MPC can be applied to AUVs to address the practical constraint issue. Besides, the MPC owns good inherent robustness against model uncertainties and external disturbances, which makes it a perfect solution to AUV control problems. On the other hand, due to the implicit nature of the optimization procedure, the characterization of closed-loop stability for the MPC-based AUV control is challenging and complicated. There are no existing control theoretical results on the MPC-based AUV motion control in the literature. To fill this gap and push forward the application of the advanced model predictive control technology to AUV control systems, this dissertation focuses on the study of the MPC-based AUV motion control problems and attempts to lay the theoretical foundation for the application of MPC to the marine vehicle systems. The main contributions of this dissertation are summarized as follows.

- **Design of nonlinear model predictive control (NMPC) for the integrated path planning and tracking control.** A unified receding horizon optimization (RHO) framework is proposed for solving the integrated path planning and tracking control problem of an AUV. The RHO framework consists of a spline-based path planner and an NMPC tracking controller. The path planning is formulated into receding horizon optimization problems which accommodates the practically finite perceiving capability of the AUV. Once the reference path is planned, with a predetermined timing law, it is augmented in order to provide the reference trajectory for each state of the AUV. Then an NMPC tracking controller is designed for the vehicle to precisely track the reference trajectory. Sufficient conditions for closed-loop stability are derived. Finally, an implementation algorithm which seamlessly integrates the path planning and the NMPC tracking control is proposed. With the implementation algorithm the obtained closed-loop stability of the NMPC tracking control can be preserved. To the best of our knowledge, it is the first time to explicitly conduct the stabilizing conditions of the MPC-based trajectory tracking control for marine vehicles.

- **Design of Lyapunov-based model predictive control (LMPC) for the dynamic positioning and trajectory tracking control.** Firstly, an LMPC-based dynamic positioning (DP) control algorithm is proposed for an AUV. A nonlinear proportional-derivative (PD) control law is exploited to construct

the contraction constraint in optimization problem that is associated with the LMPC. A quasi-global stability property can be claimed for the closed-loop LMPC-based DP control system. Secondly, the LMPC is applied to solve the AUV trajectory tracking control problem. An auxiliary nonlinear tracking control law is designed using the backstepping technique and then used to construct the contraction constraint. Conditions for recursive feasibility and closed-loop stability are derived. In both DP and tracking control, the thrust allocation (TA) subproblem is solved simultaneously with the LMPC control, which reduces the conservativeness brought by conventional (TA) solutions. Essentially, the proposed LMPC method builds on the existing AUV control system and incorporates online optimization to improve the control performance. Since the closed-loop stability does not rely on the exact solution of the optimization, the LMPC creates a trade-off between computational complexity and control performance. We can easily control the computational complexity by specifying the maximum iteration number meanwhile guarantee the control performance no worse than the existing AUV motion controller.

- **Design of multi-objective model predictive control (MOMPC) for the path following control.** A novel MOMPC method is proposed to solve the path following (PF) control problem of an AUV. Two performance indexes which reflect the path convergence requirement and the speed assignment are designed. Then the PF problem can be formulated into the MOMPC framework with the two performance indexes as the objective function. Since the path convergence is usually more important than the speed assignment, two methods which handle objective prioritization are proposed to solve the associated vector-valued optimization problem. The internal relationship between the two methods are explored and the conditions for closed-loop stability are provided. The proposed MOMPC method not only provides a novel scheme to solve the AUV PF control problem, but also lays a foundation for the study of AUV motion control problems with multiple control objectives.

- **Design of efficient implementation algorithms for the NMPC trajectory tracking control.** Two distinct fast implementation strategies are proposed for the NMPC-based trajectory tracking control of an AUV. The first strategy is based on the numerical continuation method. Assuming that the solution of the associated optimization problem is not obtained at singular points, the

NMPC control signals can be approximated without undergoing the successive linearization step which is inevitable in off-the-shelf NLP algorithms, therefore, the computational complexity can be significantly reduced. The convergence of the solution is proved. The second strategy exploits the dynamic properties of the AUV motion, and solves the optimization problems in a distributed fashion. The recursive feasibility and closed-loop stability of the distributed implementation are proved. The proposed fast implementation strategies considerably alleviate the heavy computational burden hence greatly increases the possibility of implementing NMPC-based motion control on various AUVs including those with limited onboard computing resources.

1.4 Organizations of the Dissertation

This section provides a map of the dissertation to show the readers where and how it validates the claims previously made.

Chapter 1 contains the fundamentals and literature reviews of the closely related research fields. It also presents the research background, motivations and main contributions of this PhD dissertation.

Chapter 2 develops the mathematical model of AUV that will be used throughout the dissertation. Several important properties associated with the developed model are also explored in this chapter.

Chapter 3 studies the path planning and tracking control of an AUV. A unified receding horizon optimization (RHO) framework is proposed with a novel spline-based path planning method and the nonlinear model predictive tracking controller design.

Chapter 4 presents a Lyapunov-based model predictive control (LMPC) framework for the motion control of an AUV. The LMPC controller designs for dynamic positioning and trajectory tracking are detailed.

Chapter 5 considers the path following control problem of an AUV. A novel multi-objective model predictive control (MOMPC) framework is proposed to handle the objective prioritization.

Chapter 6 focuses on the computational complexity of the nonlinear model predictive control (NMPC) algorithms. Two numerically efficient implementation strategies, namely, modified C/GMRES and distributed NMPC, are proposed for the AUV trajectory tracking control.

Chapter 7 summarizes the work in this dissertation, and discusses some potential future research directions.

Chapter 2

AUV Modeling

The study of AUV motion can be split into two groups: Kinematics, which only deals with geometrical aspects of the motion, and Dynamics, which analyzes the forces and moments causing the motion. In this chapter, we elaborate the kinematic and dynamic equations of AUV motion, and based on which we establish the control system model that is adopted in the study of AUV motion control.

2.1 Kinematics

2.1.1 Reference Frames

The sensors provide their measurements with respect to different reference frames. When studying the motion control problems, it is convenient to use two reference frames (see Figure 2.1) to describe the AUV motion state and the control objective:

- The body-fixed reference frame (BRF) is affixed to the vehicle with the origin selected to be the center of gravity (CG). The body axes are defined such that they coincide with the principal axes of inertia: The longitude axis which points from aft to fore is often referred to as the x_b axis; the transversal axis which points from port to starboard is the y_b axis; the z_b axis is defined orthogonal to both x_b and y_b axes and obeys the right-hand rule.

- Then the motion of the AUV can be described as the BRF motion relative to an inertial reference frame (IRF) which is used to record the footprints of the vehicle and to specify the control objectives. Usual selections of IRF include

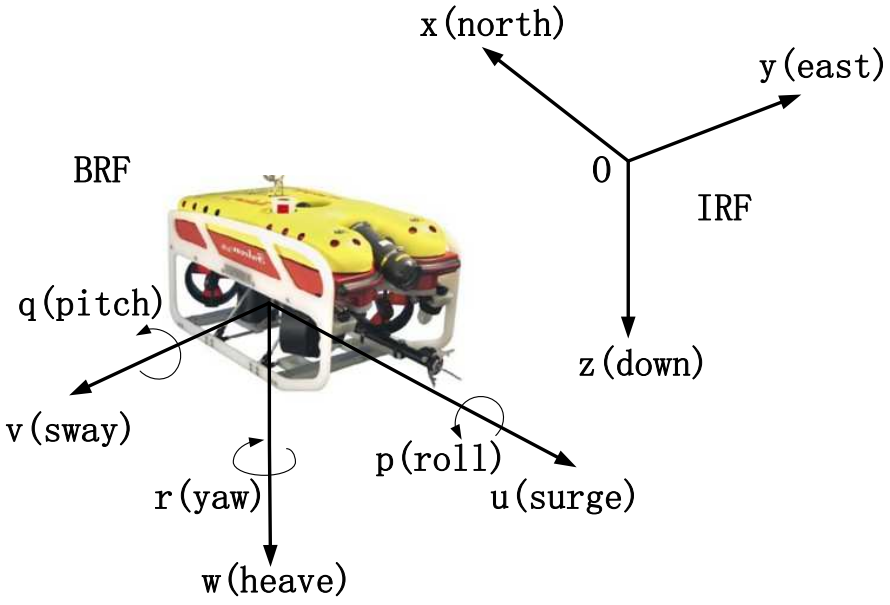

Figure 2.1: The reference frames for AUV motion control.

the Earth-centered inertial (ECI) frame, the Earth-centered Earth-fixed (ECEF) reference frame and the North-East-Down (NED) coordinate system [46].

The linear and angular velocities of the vehicle are expressed in the BRF while the position and orientation are described with respect to the IRF. The vectorial forms of these expressions are as follows:

$\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^{\mathrm{T}}$, the position and orientation vector represented in IRF

$\mathbf{v} = [u, v, w, p, q, r]^{\mathrm{T}}$, the velocity vector represented in BRF

Since in many navigation applications, the position vector is decomposed in NED coordinates, the IRF is selected to be coincident with the North-East-Down coordinate system in this dissertation.

2.1.2 Transformation between Reference Frames

Rotation matrices are essential in deriving the kinematic equations of motion for an AUV. The rotation matrix between the BRF and the IRF is denoted as $\mathbf{R}_b^i$ which belongs to the special orthogonal group of order three $SO(3)$:

$$SO(3) = \{\mathbf{R} \mid \mathbf{R} \in \mathbb{R}^{3\times 3},\ \mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I},\ \det \mathbf{R} = 1\} \tag{2.1}$$

Let $\mathbf{v}_o^b = [u, v, w]^{\mathrm{T}}$ denote the linear velocity vector fixed in BRF and $\mathbf{v}_o^i$ denote this velocity vector decomposed in IRF. Then the relation between them can be expressed using the following equation:

$$\mathbf{v}_o^i = \mathbf{R}_b^i(\Theta)\mathbf{v}_o^b \tag{2.2}$$

where $\Theta = [\phi, \theta, \psi]^{\mathrm{T}}$ encloses the Euler angles: roll (ϕ), pitch (θ) and yaw (ψ). In navigation, guidance and control applications, the zyx-convention is commonly adopted to describe the rotation matrix $\mathbf{R}_b^i(\Theta)$:

$$\mathbf{R}_b^i(\Theta) = \mathbf{R}_{z,\psi}\mathbf{R}_{y,\theta}\mathbf{R}_{x,\phi} \tag{2.3}$$

with

$$\mathbf{R}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix},\ \mathbf{R}_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix},\ \mathbf{R}_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Here, $s\cdot, c\cdot$ are shorthand for trigonometric functions $\sin(\cdot), \cos(\cdot)$. The inverse transformation satisfies

$$\mathbf{R}_b^i(\Theta)^{-1} = \mathbf{R}_i^b(\Theta) = \mathbf{R}_{x,\phi}^{\mathrm{T}}\mathbf{R}_{y,\theta}^{\mathrm{T}}\mathbf{R}_{z,\psi}^{\mathrm{T}} \tag{2.4}$$

Expanding (2.3) we have

$$\mathbf{R}_b^i(\Theta) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi c\theta s\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta c\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2.5}$$

Let $\boldsymbol{\omega}_{ib}^b = [p, q, r]^{\mathrm{T}}$ denote the angular velocity of BRF relative to IRF decomposed in BRF and $\dot{\Theta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$ denote the Euler-angle rate. They are related by the following equation:

$$\dot{\Theta} = \mathbf{T}(\Theta)\boldsymbol{\omega}_{ib}^b \tag{2.6}$$

The transformation matrix $\mathbf{T}(\Theta)$ can be derived through the following relation:

$$\boldsymbol{\omega}_{ib}^b = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{x,\phi}^{\mathrm{T}} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_{x,\phi}^{\mathrm{T}}\mathbf{R}_{y,\theta}^{\mathrm{T}} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{2.7}$$

Expanding (2.7) we have

$$\mathbf{T}(\Theta) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \tag{2.8}$$

where $s\cdot, c\cdot, t\cdot$ are shorthand for $\sin(\cdot), \cos(\cdot)$ and $\tan(\cdot)$, respectively. Combining (2.2) and (2.6), we can have the 6 DOF kinematic equations of motion expressed in the vectorial form

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^i(\Theta) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{T}(\Theta) \end{bmatrix} \begin{bmatrix} \mathbf{v}_o^b \\ \boldsymbol{\omega}_{ib}^b \end{bmatrix} = \mathbf{J}(\boldsymbol{\eta})\mathbf{v} \tag{2.9}$$

where $\mathbf{p} = [x, y, z]^{\mathrm{T}}$ is the position vector of the vehicle represented in IRF.

2.2 Nonlinear Dynamics of AUVs

2.2.1 Rigid-Body Dynamics

To facilitate the derivation of the dynamic equations of AUV motion, it is common and reasonable to assume that the vehicle is a rigid body, which eliminates the need of analyzing the interactions between individual elements of mass.

The rigid-body dynamics of the AUV can be derived by applying the Newtonian mechanics [46]:

$$\mathbf{M}_{RB}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_{RB} \tag{2.10}$$

where $\boldsymbol{\tau}_{RB} = [X, Y, Z, K, M, N]^{\mathrm{T}}$ is the generalized external force and moment vector expressed in BRF. Since the origin of BRF is coincident with the CG of AUV, the rigid-body inertia matrix $\mathbf{M}_{RB}$ can be simplified as

$$\mathbf{M}_{RB} = \begin{bmatrix} m\mathbf{I}_{3\times3} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_o \end{bmatrix} \tag{2.11}$$

where m is the mass of the vehicle and $\mathbf{I}_o$ is the inertia tensor defined as

$$\mathbf{I}_o = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{2.12}$$

and the rigid-body Coriolis and centripetal matrix $\mathbf{C}_{RB}$ is

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} \mathbf{0}_{3\times 3} & -m\mathbf{S}(\mathbf{v}_o^b) \\ -m\mathbf{S}(\mathbf{v}_o^b) & -\mathbf{S}(\mathbf{I}_0\boldsymbol{\omega}_{ib}^b) \end{bmatrix} \tag{2.13}$$

where $\mathbf{S}(\cdot)$ is the cross product operator.

Definition 1 (**Cross Product Operator**). *The cross product of two vectors* $\mathbf{a}\times\mathbf{b}$ *can be expressed using normal matrix multiplication:*

$$\mathbf{a}\times\mathbf{b} = \mathbf{S}(\mathbf{a})\mathbf{b} \tag{2.14}$$

and the operator $\mathbf{S}(\cdot)$ *is defined as*

$$\mathbf{S}(\mathbf{a}) = -\mathbf{S}^{\mathrm{T}}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \ \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{2.15}$$

2.2.2 Hydrodynamic Forces and Moments

The hydrodynamics should be considered in calculating the total external forces and moments $\boldsymbol{\tau}_{RB}$. Several main contributions of the hydrodynamic forces and moments include the radiation-induced forces, skin friction damping, wave drift damping, damping due to vortex shedding and environmental disturbances. They are treated separately based on the principle of superposition.

The radiation-induced forces and moments include three components, namely, added mass, potential damping and restoring forces. They can be expressed mathematically as follows:

$$\boldsymbol{\tau}_R = -\mathbf{M}_A\dot{\mathbf{v}} - \mathbf{C}_A(\mathbf{v})\mathbf{v} - \mathbf{D}_P(\mathbf{v})\mathbf{v} - \mathbf{g}(\boldsymbol{\eta}) \tag{2.16}$$

where $-\mathbf{M}_A\dot{\mathbf{v}} - \mathbf{C}_A(\mathbf{v})\mathbf{v}$ is the added mass term, $-\mathbf{D}_P(\mathbf{v})\mathbf{v}$ is the potential damping term and $-\mathbf{g}(\boldsymbol{\eta})$ is the restoring force term. The inertia matrix of the added mass is

defined as

$$\mathbf{M}_A = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \tag{2.17}$$

where the hydrodynamic coefficients are defined as partial derivative of the added mass force over the corresponding acceleration. For example, the added mass force X_A along the x-axis due to the acceleration w is $X_{A1} = X_{\dot{w}}\dot{w}$, and $X_{\dot{w}} = \partial X_A/\partial \dot{w}$.

The hydrodynamic Coriolis and centripetal matrix C_A can be calculated using into the following formula [46]:

$$\mathbf{C}_A(\mathbf{v}) = \begin{bmatrix} \mathbf{0}_{3\times 3} & -\mathbf{S}(\mathbf{A_{11}}\mathbf{v}_o^b + \mathbf{A_{12}}\boldsymbol{\omega}_{ib}^b) \\ -\mathbf{S}(\mathbf{A_{11}}\mathbf{v}_o^b + \mathbf{A_{12}}\boldsymbol{\omega}_{ib}^b) & -\mathbf{S}(\mathbf{A_{21}}\mathbf{v}_o^b + \mathbf{A_{22}}\boldsymbol{\omega}_{ib}^b) \end{bmatrix} \tag{2.18}$$

where $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$ defined as

$$\mathbf{A} = \frac{1}{2}(\mathbf{M}_A + \mathbf{M}_A^{\mathrm{T}}),\ \mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},\ \mathbf{A}_{ij} \in \mathbb{R}^{3\times 3} \tag{2.19}$$

In addition to potential damping the skin friction damping, wave drift damping, damping due to vortex shedding need to be included, and those damping forces and moments can be expressed as

$$\boldsymbol{\tau}_D = -\mathbf{D}_S(\mathbf{v})\mathbf{v} - \mathbf{D}_W(\mathbf{v})\mathbf{v} - \mathbf{D}_M(\mathbf{v})\mathbf{v} \tag{2.20}$$

Defining the total hydrodynamic damping matrix as

$$\mathbf{D}(\mathbf{v}) = \mathbf{D}_P(\mathbf{v}) + \mathbf{D}_S(\mathbf{v}) + \mathbf{D}_W(\mathbf{v}) + \mathbf{D}_M(\mathbf{v}) \tag{2.21}$$

we have the hydrodynamic force and moment vector $\boldsymbol{\tau}_H$ written as the sum of $\boldsymbol{\tau}_R$ and $\boldsymbol{\tau}_D$, i.e.,

$$\boldsymbol{\tau}_H = -\mathbf{M}_A\dot{\mathbf{v}} - \mathbf{C}_A(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\boldsymbol{\eta}) \tag{2.22}$$

The total hydrodynamic damping forces are composed of linear damping terms and

quadratic damping terms and can be conveniently expressed as

$$\mathbf{D}(\mathbf{v})\mathbf{v} = \mathbf{D}_L\mathbf{v} + \begin{bmatrix} |\mathbf{v}|^{\mathrm{T}}\mathbf{D}_{n1}\mathbf{v} \\ |\mathbf{v}|^{\mathrm{T}}\mathbf{D}_{n2}\mathbf{v} \\ |\mathbf{v}|^{\mathrm{T}}\mathbf{D}_{n3}\mathbf{v} \\ |\mathbf{v}|^{\mathrm{T}}\mathbf{D}_{n4}\mathbf{v} \\ |\mathbf{v}|^{\mathrm{T}}\mathbf{D}_{n5}\mathbf{v} \\ |\mathbf{v}|^{\mathrm{T}}\mathbf{D}_{n6}\mathbf{v} \end{bmatrix} \tag{2.23}$$

where $\mathbf{D}_L$ is the linear damping matrix, and $\mathbf{D}_{ni}, i = 1, 2, ..., 6$ are quadratic damping matrices. The restoring forces and moments are calculated as follows:

$$\mathbf{g}(\boldsymbol{\eta}) = \begin{bmatrix} (W-B)s\theta \\ -(W-B)c\theta s\phi \\ -(W-B)c\theta c\phi \\ y_B Bc\theta c\phi - z_B Bc\theta s\phi \\ -z_B Bs\theta - x_B Bc\theta c\phi \\ x_B Bc\theta s\phi + y_B Bs\theta \end{bmatrix} \tag{2.24}$$

where $W = mg$ is the gravity, $B = bg$ is the buoyancy and $[x_B, y_B, z_B]$ is the coordinates of the center of buoyancy (CB) with respect to BRF.

Let $\mathbf{w}$ denote the environmental disturbances which exist due to waves and ocean currents. The total external forces and moments $\boldsymbol{\tau}_{RB}$ can be expressed as

$$\boldsymbol{\tau}_{RB} = \boldsymbol{\tau}_H + \mathbf{w} + \boldsymbol{\tau} \tag{2.25}$$

where $\boldsymbol{\tau}$ represents the propulsive forces and moments. Then we can have the 6 DOF dynamic equations of motion arranged in the following form

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \mathbf{w} \tag{2.26}$$

where

$$\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A, \quad \mathbf{C}(\mathbf{v}) = \mathbf{C}_{RB}(\mathbf{v}) + \mathbf{C}_A(\mathbf{v}) \tag{2.27}$$

2.3 AUV Model for Motion Control

For our experimental platform, the Saab SeaEye Falcon open-frame ROV/AUV (Figure 2.1), the thruster layout does not allow active control on roll and pitch. In this dissertation, therefore, we consider the motion of the Falcon in the local level plane.

Three mild assumptions can be satisfied for the low-speed motion of Falcon: (i) the vehicle is with three planes of symmetry; (ii) the mass distribution is homogeneous; (iii) the pitch and roll motions are neglected. As a result, for the motion control in the local level plane, the system matrices in (2.26) can be simplified. The inertia matrix becomes

$$\mathbf{M} = \begin{bmatrix} M_{\dot{u}} & 0 & 0 \\ 0 & M_{\dot{v}} & 0 \\ 0 & 0 & M_{\dot{r}} \end{bmatrix} \tag{2.28}$$

where $M_{\dot{u}} = m - X_{\dot{u}}$, $M_{\dot{v}} = m - Y_{\dot{v}}$ and $M_{\dot{r}} = I_z - N_{\dot{r}}$ are the inertia terms including add mass. The restoring force is neglected $\mathbf{g}(\boldsymbol{\eta}) = \mathbf{0}$, and the damping matrix is

$$\mathbf{D}(\mathbf{v}) = \begin{bmatrix} X_u + D_u|u| & 0 & 0 \\ 0 & Y_v + D_v|v| & 0 \\ 0 & 0 & N_r + D_r|r| \end{bmatrix} \tag{2.29}$$

where X_u, Y_v, N_r are linear drag coefficients, and D_u, D_v, D_r are the quadratic drag coefficients. The Coriolis and centripetal matrix becomes

$$\mathbf{C}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & -M_{\dot{v}}v \\ 0 & 0 & M_{\dot{u}}u \\ M_{\dot{v}}v & -M_{\dot{u}}u & 0 \end{bmatrix} \tag{2.30}$$

In the local level plane, the velocity vector $\mathbf{v} = [u, v, r]^{\mathrm{T}}$ encloses the surge, sway and yaw velocities, and the position and orientation vector $\boldsymbol{\eta} = [x, y, \psi]^{\mathrm{T}}$ includes the position and heading of the vehicle.

In the AUV motion controller design, we assume that the disturbances are small, i.e., $\mathbf{w} \approx \mathbf{0}$. Then the dynamic equations of motion under consideration is:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{2.31}$$

where $\boldsymbol{\tau} = [F_u, F_v, F_r]^{\mathrm{T}}$ denotes the generalized thrust forces and moments. Further expanding the dynamic equations (2.31) into the element-wise expression, we have

the following equations:

$$\dot{u} = \frac{M_{\dot{v}}}{M_{\dot{u}}}vr - \frac{X_u}{M_{\dot{u}}}u - \frac{D_u}{M_{\dot{u}}}u|u| + \frac{F_u}{M_{\dot{u}}} \tag{2.32a}$$

$$\dot{v} = -\frac{M_{\dot{u}}}{M_{\dot{v}}}ur - \frac{Y_v}{M_{\dot{v}}}v - \frac{D_v}{M_{\dot{v}}}v|v| + \frac{F_v}{M_{\dot{v}}} \tag{2.32b}$$

$$\dot{r} = \frac{M_{\dot{u}} - M_{\dot{v}}}{M_{\dot{r}}}uv - \frac{N_r}{M_{\dot{r}}}r - \frac{D_r}{M_{\dot{r}}}r|r| + \frac{F_r}{M_{\dot{r}}} \tag{2.32c}$$

The kinematic equations (2.9) can also be simplified as follows:

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} = \mathbf{R}(\psi)\mathbf{v} \tag{2.33}$$

Further expanding the kinematic equations (2.33) into the element-wise expression, we have the following equations:

$$\dot{x} = u\cos\psi - v\sin\psi \tag{2.34a}$$

$$\dot{y} = u\sin\psi + v\cos\psi \tag{2.34b}$$

$$\dot{\psi} = r \tag{2.34c}$$

Defining the system state $\mathbf{x} = [\boldsymbol{\eta}^{\mathrm{T}}, \mathbf{v}^{\mathrm{T}}]^{\mathrm{T}}$ and view $\boldsymbol{\tau}$ as the generalized control input. From (2.9) and (2.31), we can have the general form of the AUV model

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{R}(\psi)\mathbf{v} \\ \mathbf{M^{-1}}(\boldsymbol{\tau} - \mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\boldsymbol{\eta})) \end{bmatrix} = \bar{\mathbf{f}}(\mathbf{x},\boldsymbol{\tau}) \tag{2.35}$$

The generalized control input $\boldsymbol{\tau}$ is the resulting force of the thrusters. For the Falcon, four thrusters are effective in the local level plane. The relationship between them is described by the following thrust distribution function:

$$\boldsymbol{\tau} = \mathbf{Bu} \tag{2.36}$$

where $\mathbf{u} = [u_1, u_2, u_3, u_4]^{\mathrm{T}}$ denotes the force provided by each thruster; $\mathbf{B}$ is the input matrix. Then the control system model that is used for the AUV motion control can be established by simply binding the kinematic equations, dynamics equations and

the thrust distribution function:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{R}(\psi)\mathbf{v} \\ \mathbf{M^{-1}}(\mathbf{Bu} - \mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\boldsymbol{\eta})) \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{2.37}$$

The hydrodynamic coefficients for the Falcon model (2.37) are summarized in Table 2.1 which are extracted from the previous modeling experiments based on [109]. The the input matrix is

$$\mathbf{B} = \begin{bmatrix} 0.7974 & 0.8643 & 0.8127 & 0.8270 \\ 0.6032 & 0.5029 & -0.5824 & -0.5610 \\ 0.2945 & -0.3302 & -0.2847 & 0.3505 \end{bmatrix} \tag{2.38}$$

Table 2.1: Hydrodynamic coefficient summary.

Inertia Term	Linear Drag	Quadratic Drag
$M_{\dot{u}} = 283.6\ kg$	$X_u = 26.9\ kg/s$	$D_u = 241.3\ kg/m$
$M_{\dot{v}} = 593.2\ kg$	$Y_v = 35.8\ kg/s$	$D_v = 503.8\ kg/m$
$M_{\dot{r}} = 29.0\ kgm^2$	$N_r = 3.5\ kgm^2/s$	$D_r = 76.9\ kgm^2$

For the established AUV model (2.37), the following important properties can be easily explored and will be exploited in the controller design:

- □ P-1: The initial matrix is symmetric positive definite and upper bounded: $\infty > \bar{m}\mathbf{I} \geq \mathbf{M} = \mathbf{M}^{\mathrm{T}} > 0$
- □ P-2: The Coriolis and centripetal matrix is skew-symmetric: $\mathbf{C}(\mathbf{v}) = -\mathbf{C}^{\mathrm{T}}(\mathbf{v})$
- □ P-3: The inverse of rotation matrix satisfies: $\mathbf{R}^{-1}(\psi) = \mathbf{R}^{\mathrm{T}}(\psi)$ and it preserves length $\|\mathbf{R}^{\mathrm{T}}(\psi)\dot{\boldsymbol{\eta}}\|_2 = \|\dot{\boldsymbol{\eta}}\|_2$.
- □ P-4: The damping matrix is positive definite: $\mathbf{D}(\mathbf{v}) > 0$
- □ P-5: The input matrix satisfies that $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ is non-singular
- □ P-6: The restoring force $\mathbf{g}(\boldsymbol{\eta})$ is bounded: $\|\mathbf{g}(\boldsymbol{\eta})\|_\infty \leq \bar{g}$

2.4 Conclusion

In this chapter, we have briefly discussed the kinematics and dynamics of the AUV motion. The mathematical model for our experimental platform, the Saab SeaEye Falcon open-frame ROV/AUV, was established based on the kinematic equations of motion, the dynamic equations of motion and the thrust distribution function. The hydrodynamic coefficients were provided and several important model properties that will be exploited in the motion controller design were explored.

Chapter 3

Receding Horizon Optimization for Integrated Path Planning and Tracking Control of an AUV

3.1 Introduction

3.1.1 Research Background and Contributions

Trajectory tracking, being a basic robotic control problem, has been extensively studied for AUVs in the past several decades. For the tracking of piecewise linear paths, the line-of-sight (LOS) scheme is often used [47]. To stabilize the cross-track error in the LOS scheme, conventional PID [46], LQG [60] and nonlinear PID control techniques [81] have been applied to AUVs. For tracking of time-parameterized curves, the Lyapunov-based backstepping technique can be applied [113]. Due to its insensitivity to parametric uncertainty, the sliding mode control [137] is suitable for the AUV tracking control as well. However, the aforementioned control methods lack the capability of handling system constraints which are ubiquitous, typically in terms of actuator limits. This motivates control theorists and practitioners to investigate the model predictive control (MPC) for the AUV trajectory tracking problem. The beauty of MPC lies in the fact that it can conveniently handle nonlinear multiple input multiple output (MIMO) system control problems and explicitly take system constraints into consideration [88]. Linear MPC formulation of AUV tracking control has been investigated based on the linearization of the AUV model [100]. Linear MPC inherits the merit of the convex optimization problem which can be efficiently solved

by off-the-shelf numerical algorithms [9] to realize real-time control implementation. On the other hand, demerit of this configuration is obviously drawn from the approximation error of linearization. Naturally, the AUV tracking is extended to nonlinear MPC formulation [23, 32] with the sacrifice of algorithm efficiency to model accuracy. Nevertheless, among all of the existing results of MPC tracking control, they either only take the kinematic equations as the vehicle model, e.g., [56, 23], or consider both kinematics and dynamics but no rigorous proof of stability is conducted in the controller design, e.g., [100, 32]. This motivates the main objective of the work in this chapter: To design an NMPC-based tracking controller considering both dynamics and kinematics of the AUV, and to provide the stability analysis for the closed-loop control system.

The path planning plays an important role in the tracking control. The conventional marine vessel guidance system [46] which generates whole-journey way-points does not work for AUVs, because oftentimes the global oceanic information is not available *a priori*. Instead, the fast, reactive, and dynamic path planning methods are desirable [144, 68]. Recently, a spline-based path planning method is reported in [14] for autonomous mining vehicle applications. In view of the similar essentials of autonomous vehicles, the basic principle is probably applicable to AUVs as well. The global environment information is assumed to be known in [14]. Considering the practical operation of AUVs with a limited sensing range, the second objective of the work in this chapter appears: To accommodate the spline-based path planning method to the limited perceiving capability of AUVs in the tracking control application.

In this chapter, by introducing a virtual reference system, the trajectory tracking control problem can be transformed to the regulation problem of the error dynamics. The optimal value function of the associated optimal control problem solved at each sampling instant can be shown nonincreasing along the trajectory of the closed-loop system, which guarantees the asymptotic stability of the closed-loop control system. Furthermore, a dynamic path planning method is developed based on the receding horizon optimizations, which enables the natural integration of the path planning and the tracking control.

The main contributions of this chapter are three-fold,

- A novel receding horizon optimization formulation is proposed for the spline-based path planning method which accommodates to the practically limited perceiving capability of AUVs.

- An NMPC-based tracking controller is designed to track the planned reference path. Sufficient conditions for closed-loop stability are conducted.
- A unified receding horizon optimization scheme, seamlessly integrating the path planning and the MPC tracking control, is proposed for the AUV application.

3.1.2 Chapter Organization

The remainder of this chapter is organized as follows. In Section 3.2, the problem statement is presented. In Section 3.3, we develop a receding horizon path planing method for the vehicle. Section 3.4 provides the formulation of the tracking control problem into the NMPC scheme as well as the detailed controller design. In Section 3.5, the overall motion control algorithm which combines the NMPC tracking control and the path planning is depicted. Simulation results are demonstrated in Section 3.6. Section 3.7 provides the concluding remarks.

The notations adopted in this chapter are explained as follows. The symbol $\|\cdot\|$ refers to the Euclidean norm for vectors and the induced 2-norm for matrices. For a function $f(x)$, the time derivative is denoted by $\dot{f}(x)$ while the derivative with respect to x is denoted by $f'(x)$. The diagonal operation is abbreviated by $\mathrm{diag}(\cdot)$.

3.2 Problem Statement

The combined problem of path planning and tracking control arises in many AUV applications. For example, in the marine source seeking applications [20] or deep-sea archaeological exploration applications [16], the workspace is constrained or cluttered with obstacles. As illustrated in Figure 3.1, the workspace of AUVs is confined by two polygonal chains $c_1(x)$ and $c_2(x)$ which reflect the topography of the seafloor or the structure of an underwater shipwreck. Inevitably, the path planning needs to be taken into account for safety reasons.

Definitions of reference paths may be diverse for various applications. Basically, a reference path should be continuous, feasible and maybe with some performance index optimized, e.g., the way-point straight path for marine vessles [46]. More advanced path planing requires higher order of smoothness, i.e., the resultant path is continuous with its derivatives. Now that the second-order derivative of the path is proportional to the vehicle acceleration, a reference path with continuous second-order derivative is particularly pursued.

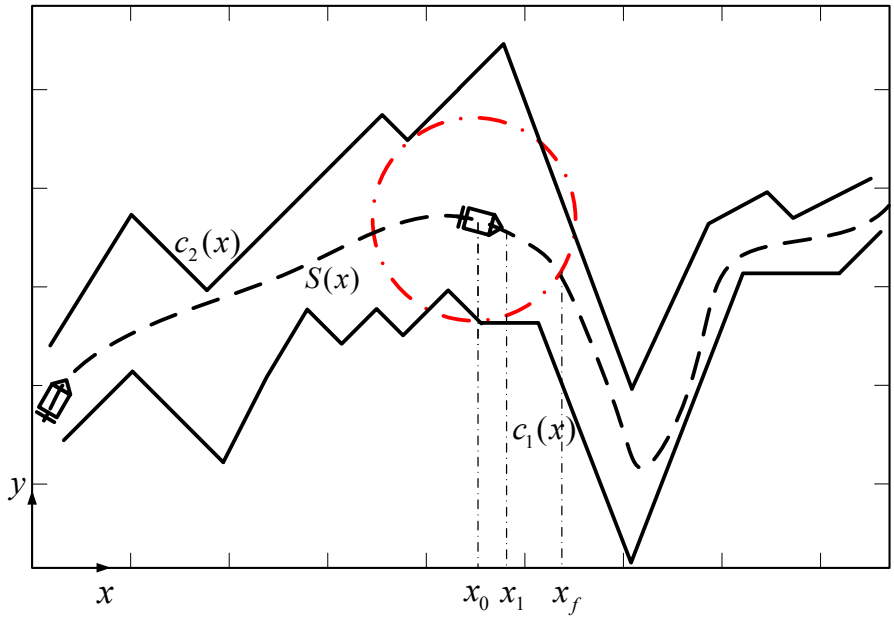

Figure 3.1: Illustration of the combined AUV motion control problem.

Since the AUV sway velocity is considerably smaller than its surge counterpart, the following nonholonomic kinematic equations well approximate the vehicle motion [46] at the velocity level:

$$\begin{aligned} \dot{x} &= u\cos\psi - v\sin\psi \approx u_0\cos\psi \\ \dot{y} &= u\sin\psi + v\cos\psi \approx u_0\sin\psi \\ \dot{\psi} &= r \end{aligned} \tag{3.1}$$

where $[x, y, \psi]^{\mathrm{T}}$ denote the position and orientation; $[u, v, r]^{\mathrm{T}}$ stand for the surge, sway and yaw velocities while u_0 denotes the resultant tangential velocity of the vehicle.

To precisely track a desired parameterized path $p(s) = [x_d(s), y_d(s)]^{\mathrm{T}}$, the required open-loop tangential velocity u_{oc} and yaw rate r_{oc} can be calculated for the kinematic model (3.1) in an explicit manner, provided that there is no disturbance or initial error

$$u_{oc}(t) = \sqrt{x_d'^2 + y_d'^2}\,\dot{s} \tag{3.2a}$$

$$r_{oc}(t) = \frac{x_d' y_d'' - y_d' x_d''}{x_d'^2 + y_d'^2}\dot{s} = u_{oc}(t)\kappa(s) \tag{3.2b}$$

where $\kappa(s)$ represents the curvature along the path. From (3.2) it is readily seen that

the smaller curvature will make the steering easier. In this sense, a reference path with minimum curvature is practically desirable for AUVs.

However, due to large decaying rate of acoustic signals in water, the effective sensing range is practically short (as described by the dashed circle in Figure. 3.1). The reference path needs to be dynamically generated as the vehicle moves forward, i.e., the path planning and tracking control are inherently coupled. Therefore, the AUV control task contains the following two aspects:

i). Dynamic Path Planning: Based on updated local information dynamically plan a minimum curvature reference path for the AUV.

ii). Precise Tracking Control: Based on the planned reference path steer the AUV to converge and precisely track the path.

3.3 Minimum Curvature Receding Horizon Path Planning

Inspired by [67] and [14], we take advantage of spline functions as the path template. Spline functions [18] have many desirable properties that make them suitable for path planning problems in the engineering practice. A spline can be conveniently selected to provide enough order of smoothness and is parameterized to facilitate the reference path computation. A spline function represented in b-form is defined as follows:

$$S(x) = \sum_{i=1}^{n} \alpha_i S_{i,k}(x) \tag{3.3}$$

where n is the number of knots, α_i are the control parameters and $S_{i,k}(x)$ are kth order basis splines defined recursively,

$$S_{i,1}(x) = \begin{cases} 1, & x_i \leq x < x_{i+1} \\ 0, & otherwise \end{cases}$$

and

$$S_{i,k}(x) = \frac{(x - x_i)S_{i,k-1}(x)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - x)S_{i+1,k-1}(x)}{x_{i+k} - x_{i+1}}$$

For a spline $S(x)$, the measure of smoothness is chosen as (3.4), which is the objective function in the reference path planning calculation.

$$F(x) = \int_{s(x_0)}^{s(x_f)} \kappa'(s)^2 ds \tag{3.4}$$

Considering the relationship between the arc-length s and the Cartesian coordinate x, the objective function in (3.4) is equivalent to

$$F(x) = \int_{x_0}^{x_f} \frac{\kappa'(x)^2}{\sqrt{1 + S'(x)^2}} dx \tag{3.5}$$

where $\kappa(x)$ represents the curvature of the spline

$$\kappa(x) = \frac{S''(x)}{(1 + S'(x)^2)^{3/2}}$$

For simplicity, we assume that the yaw plane environment is restricted by two polygonal chains (see Figure. 3.1). Then the path planning problem can be formulated as the following optimization problem.

Problem 1. *Given the restriction polygonal chains $c_1(x) \leq c_2(x)$, with proper numbers of knots n, order o and end multiplicity l, compute the spline function $S(\alpha, x)$ defined on $\Omega = [x_0, x_f]$ such that*

$$\begin{aligned} \min_{\alpha} . \quad & F(\alpha, x) \\ s.t. \quad & c_1(x) \leq S(\alpha, x) \leq c_2(x) \end{aligned}$$

where $\alpha = [\alpha_1, ..., \alpha_n]^{\mathrm{T}}$. Here, we denote the spline as $S(\alpha, x)$ and objective function as $F(\alpha, x)$ to explicitly point out the dependence on the control parameters which are the decision variables in the optimization. In other places, when the spline is already determined, we simply use the notation $S(x)$ as per established convention [18]. Also we shall notice that the order of the spline template should be at least three to guarantee the continuity of the acceleration.

Remark 1. *It is worth noting that the number of control parameters has an explicit relation to the number of knots, thus the selection of knots appears a significant issue. Extensive study on knot placement can be found in [80]. For the AUV application, the larger number of knots makes the description of the environment more accurate, but*

it also makes the calculation of reference path $S(x)$ more time-consuming. In view of the limited computational resource and memory size on chip, usually the number of knots is required to be as small as possible without losing main characteristics of the environment. A possible way is to choose the local extrema of the environment and connect these extrema sequentially. A safe margin concept can be used [14].

Remark 2. *Another issue regarding computational complexity is the inequality constraints $c_1(x) \leq S(\alpha, x) \leq c_2(x)$. The constraints are enforced numerically, i.e., we first sample these functions and then impose the inequality constraints at discrete points. Therefore, to increase the number of samples is to enlarge the size of the optimization problem. However, b-splines form a stable basis for splines and Runge's phenomenon in which oscillation may occur between knots can be avoided. In practice, a small number of samples are usually sufficient.*

Remark 3. *Although we simplify the yaw plane environment as the area bounded by two polygonal chains, more complicated environment can be tackled by transforming it into constraints of the optimization problem or equivalently feasible regions for the nonlinear programming (NLP) algorithm. The NLP algorithms such as SQP method and interior-point method [9] usually require a connected feasible region. However, complicated environment may be characterized as several disjoint feasible sub-regions. This will result in more than one time NLP to calculate the reference path.*

Because the sensing radius is short in practice, the global optimization cannot be performed for the AUV path planning. Instead, the receding horizon scheme can help to approximate the minimum curvature reference path. This idea can be explained through the following example: Suppose the sensing range is 5-meter (illustrated as the dashed circle in Figure 3.1), at a specific position, a local reference path (illustrated as the dashed line segment between x_0 and x_f) is generated based on the current 5-meter data; when the AUV moves 1 meter forward to x_1, a new reference path, between the new start point $\bar{x}_0 = x_1$ and new end point $\bar{x}_f$, will be generated based on the new 5-meter measurement data. In this case, the continuity between the neighbor splines has to be taken into consideration. Therefore, we impose the continuity conditions as constraints in the optimization formulation.

Problem 2. *Given measured environment information $c_1(x) \leq c_2(x)$ only defined on $\Omega = [x_0, x_f]$ and the last calculated reference path $S_0(x)$, with proper numbers of knots n, order o and end multiplicity l, compute the spline function $S(\alpha, x)$ defined on Ω*

such that

$$\begin{aligned} \min_{\alpha} . \quad & F(\alpha, x) \\ s.t. \quad & c_1(x) \leq S(\alpha, x) \leq c_2(x) \\ & \parallel S(\alpha, x_0) - S_0(x_0) \parallel \leq \epsilon_1 \\ & \parallel S'(\alpha, x_0) - S_0'(x_0) \parallel \leq \epsilon_2 \\ & \parallel S''(\alpha, x_0) - S_0''(x_0) \parallel \leq \epsilon_3 \end{aligned}$$

where $\alpha = [\alpha_1, ..., \alpha_n]^{\mathrm{T}}$ and ϵ_i are the tolerances for discontinuity. We should notice that the start point of the current spline x_0 is coincident with the second knot of the previous calculated spline.

3.4 Nonlinear Model Predictive Tracking Control

In this section, we propose the design of the NMPC tracking control law so that the AUV can precisely track the reference trajectory generated in Section 3.3.

The AUV model studied for the tracking control is the Falcon dynamic model which we have discussed in Chapter 2:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{R}(\psi)\mathbf{v} \\ \mathbf{M}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\boldsymbol{\eta})) \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{3.6}$$

where the state vector $\mathbf{x} = [x, y, \psi, u, v, r]^{\mathrm{T}}$ is consisted of the pose and velocity of the vehicle, and the control vector $\mathbf{u} = [u_1, u_2, u_3, u_4]^{\mathrm{T}}$ is consisted of the forces generated by the four thrusters. The detailed expression can be found in (2.32) and (2.34).

From a control point of view, the trajectory tracking control is challenging due to the nonlinearity and coupled dynamics in the model (3.6). Since the vehicle surge velocity does not keep constant for the trajectory tracking tasks, the assumptions for model linearization cannot hold and linear control methods are inappropriate to apply. In this regard, nonlinear model predictive control, which is capable of dealing with complex nonlinearity of the dynamics, seems an attractive and practical option.

The reference path $S(x)$ can be viewed as the state trajectory $\boldsymbol{\eta}_d = [x_d, y_d, \psi_d]^{\mathrm{T}}$ generated by a reference system which owns the same kinematic equations of motion

as the real vehicle's:

$$\begin{aligned} \dot{x}_d &= u_d \cos\psi_d - v_d \sin\psi_d && (3.7a)\\ \dot{y}_d &= u_d \sin\psi_d + v_d \cos\psi_d && (3.7b)\\ \dot{\psi}_d &= r_d && (3.7c) \end{aligned}$$

Decompose the kinematic state error in the vessel parallel reference frame [70, 46]:

$$\boldsymbol{\eta}_e = \begin{bmatrix} x_e \\ y_e \\ \psi_e \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \psi_d - \psi \end{bmatrix} \tag{3.8}$$

We further partition the velocity $\mathbf{v} = \mathbf{v}_f + \mathbf{v}_e$, where $\mathbf{v}_f = [u_d \cos\psi_e - v_d \sin\psi_e, u_d \sin\psi_e + v_d \cos\psi_e, r_d]^{\mathrm{T}}$ can be regarded as a feedforward control action, and $\mathbf{v}_e = [u_e, v_e, r_e]^{\mathrm{T}}$ represents the feedback control action. Differentiating both sides of (3.13) and substituting (2.34) (3.7), we derive the kinematic error equations:

$$\begin{aligned} \dot{x}_e &= y_e r_d - u_e + y_e r_e && (3.9a)\\ \dot{y}_e &= -x_e r_d - v_e - x_e r_e && (3.9b)\\ \dot{\psi}_e &= -r_e && (3.9c) \end{aligned}$$

Analogously, we view $\mathbf{v}_d = [u_d, v_d, r_d]^{\mathrm{T}}$ as the state trajectory of the reference system that is with the same dynamic equations of motion:

$$\begin{aligned} \dot{u}_d &= \frac{M_{\dot{v}}}{M_{\dot{u}}} v_d r_d - \frac{X_u}{M_{\dot{u}}} u_d - \frac{D_u}{M_{\dot{u}}} u_d |u_d| + \frac{F_{ud}}{M_{\dot{u}}} && (3.10a)\\ \dot{v}_d &= -\frac{M_{\dot{u}}}{M_{\dot{v}}} u_d r_d - \frac{Y_v}{M_{\dot{v}}} v_d - \frac{D_v}{M_{\dot{v}}} v_d |v_d| + \frac{F_{vd}}{M_{\dot{v}}} && (3.10b)\\ \dot{r}_d &= \frac{M_{\dot{u}} - M_{\dot{v}}}{M_{\dot{r}}} u_d v_d - \frac{N_r}{M_{\dot{r}}} r_d - \frac{D_r}{M_{\dot{r}}} r_d |r_d| + \frac{F_{rd}}{M_{\dot{r}}} && (3.10c) \end{aligned}$$

Differentiating both sides of $\mathbf{v}_e = \mathbf{v} - \mathbf{v}_f$ yields

$$\begin{aligned} \dot{u}_e &= \dot{u} - \dot{u}_d \cos\psi_e - u_d \sin\psi_e r_e + \dot{v}_d \sin\psi_e - v_d \cos\psi_e r_e && (3.11a)\\ \dot{v}_e &= \dot{v} - \dot{u}_d \sin\psi_e + u_d \cos\psi_e r_e - \dot{v}_d \cos\psi_e - v_d \sin\psi_e r_e && (3.11b)\\ \dot{r}_e &= \dot{r} - \dot{r}_d && (3.11c) \end{aligned}$$

Substituting (2.32) into (3.11), we have

$$\begin{aligned}\dot{u}_e &= \tfrac{M_{\dot{v}}}{M_{\dot{u}}}(v_e + u_d \sin\psi_e + v_d \cos\psi_e)(r_e + r_d) - \tfrac{X_u}{M_{\dot{u}}}(u_e + u_d \cos\psi_e \\ &\quad -v_d \sin\psi_e) - \tfrac{D_u}{M_{\dot{u}}}(u_e + u_d \cos\psi_e - v_d \sin\psi_e)|u_e + u_d \cos\psi_e - v_d \sin\psi_e| \\ &\quad -\dot{u}_d \cos\psi_e - u_d \sin\psi_e r_e + \dot{v}_d \sin\psi_e - v_d \cos\psi_e r_e + \tfrac{F_u}{M_{\dot{u}}}\end{aligned} \tag{3.12a}$$

$$\begin{aligned}\dot{v}_e &= -\tfrac{M_{\dot{u}}}{M_{\dot{v}}}(u_e + u_d \cos\psi_e - v_d \sin\psi_e)(r_e + r_d) - \tfrac{Y_v}{M_{\dot{v}}}(v_e + u_d \sin\psi_e \\ &\quad +v_d \cos\psi_e) - \tfrac{D_v}{M_{\dot{v}}}(v_e + u_d \sin\psi_e + v_d \cos\psi_e)|v_e + u_d \sin\psi_e + v_d \cos\psi_e| \\ &\quad -\dot{u}_d \sin\psi_e + u_d \cos\psi_e r_e - \dot{v}_d \cos\psi_e - v_d \sin\psi_e r_e + \tfrac{F_v}{M_{\dot{v}}}\end{aligned} \tag{3.12b}$$

$$\begin{aligned}\dot{r}_e &= \tfrac{M_{\dot{u}} - M_{\dot{v}}}{M_{\dot{r}}}(u_e + u_d \cos\psi_e - v_d \sin\psi_e)(v_e + u_d \sin\psi_e + v_d \cos\psi_e) \\ &\quad -\tfrac{N_r}{M_{\dot{r}}}(r_e + r_d) - \tfrac{D_r}{M_{\dot{r}}}(r_e + r_d)|r_e + r_d| - \dot{r}_d + \tfrac{F_r}{M_{\dot{r}}}\end{aligned} \tag{3.12c}$$

Design F_u, F_v, and F_r as follows:

$$\begin{aligned}F_u &= -M_{\dot{v}}(v_e + u_d \sin\psi_e + v_d \cos\psi_e)(r_e + r_d) + X_u(u_e + u_d \cos\psi_e \\ &\quad -v_d \sin\psi_e) + D_u(u_e + u_d \cos\psi_e - v_d \sin\psi_e)|u_e + u_d \cos\psi_e - v_d \sin\psi_e| \\ &\quad +M_{\dot{u}}(\dot{u}_d \cos\psi_e + u_d \sin\psi_e r_e - \dot{v}_d \sin\psi_e + v_d \cos\psi_e r_e) + M_{\dot{u}}\tau_u\end{aligned} \tag{3.13a}$$

$$\begin{aligned}F_v &= M_{\dot{u}}(u_e + u_d \cos\psi_e - v_d \sin\psi_e)(r_e + r_d) + Y_v(v_e + u_d \sin\psi_e \\ &\quad +v_d \cos\psi_e) + D_v(v_e + u_d \sin\psi_e + v_d \cos\psi_e)|v_e + u_d \sin\psi_e + v_d \cos\psi_e| \\ &\quad +M_{\dot{v}}(\dot{u}_d \sin\psi_e - u_d \cos\psi_e r_e + \dot{v}_d \cos\psi_e + v_d \sin\psi_e r_e) + M_{\dot{v}}\tau_v\end{aligned} \tag{3.13b}$$

$$\begin{aligned}F_r &= (M_{\dot{v}} - M_{\dot{u}})(u_e + u_d \cos\psi_e - v_d \sin\psi_e)(v_e + u_d \sin\psi_e \\ &\quad +v_d \cos\psi_e) + N_r(r_e + r_d) + D_r(r_e + r_d)|r_e + r_d| + M_{\dot{r}}\dot{r}_d + M_{\dot{r}}\tau_r\end{aligned} \tag{3.13c}$$

By defining $\boldsymbol{\tau}_e = [\tau_u, \tau_v, \tau_r]^{\mathrm{T}}$ and substituting (3.13) into (3.12), together with

(3.9) we have the following error dynamics for the AUV tracking control:

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\psi}_e \\ \dot{u}_e \\ \dot{v}_e \\ \dot{r}_e \end{bmatrix} = \begin{bmatrix} y_e r_d - u_e + y_e r_e \\ -x_e r_d - v_e - x_e r_e \\ -r_e \\ \tau_u \\ \tau_v \\ \tau_r \end{bmatrix} = \mathbf{f}_e(\mathbf{x}_e, \boldsymbol{\tau}_e) \tag{3.14}$$

Obviously, $(\mathbf{0}, \mathbf{0})$ is an equilibrium point of (3.14). Comparing (3.10) and (3.13) at $(\mathbf{0}, \mathbf{0})$, we find that $F_u = F_{ud}$, $F_v = F_{vd}$ and $F_r = F_{rd}$, which means that the tracking of the reference trajectory is equivalent to the stabilizing the error system (3.14) to the equilibrium point.

To stabilize the error dynamics, the nonlinear model predictive control technique is used. The stabilization is realized through the minimization of the cost function:

$$J(t, \mathbf{x}_e(t), \boldsymbol{\tau}_e(t)) = g(\mathbf{x}_e(t+T)) + \int_t^{t+T} \ell(\mathbf{x}_e(s), \boldsymbol{\tau}_e(s)) ds \tag{3.15}$$

where the initial time is viewed as 0, $g(\cdot)$ is the terminal state penalty satisfying $g(\mathbf{0}) = 0$ and $g(\mathbf{x}_e) > 0$ for any $\mathbf{x}_e \neq \mathbf{0}$; $\ell(\mathbf{x}_e, \boldsymbol{\tau}_e) = \mathbf{x}_e^{\mathrm{T}} \mathbf{Q} \mathbf{x}_e + \boldsymbol{\tau}_e^{\mathrm{T}} \mathbf{R} \boldsymbol{\tau}_e$ is the stage cost with $\mathbf{Q} > 0$, $\mathbf{R} \geq 0$. Then the optimal control problem to be solved at each sampling time instant can be formulated as follows.

Problem 3. *Given the current error state $\mathbf{x}_e$, the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$, compute the optimal control force $\boldsymbol{\tau}_e(t)$ by solving the following optimization problem:*

$$\begin{aligned} \min_{\boldsymbol{\tau}_e} . \quad & J(t, \mathbf{x}_e(t), \boldsymbol{\tau}_e(t)) \\ s.t. \quad & \dot{\mathbf{x}}_e(t) = \mathbf{f}_e(\mathbf{x}_e(t), \boldsymbol{\tau}_e(t)), \ \textit{for } t \in [0, T] \\ & \mathbf{x}_e(t) \in X_e, \ \textit{for } t \in [0, T] \\ & \boldsymbol{\tau}_e(t) \in T_e, \ \textit{for } t \in [0, T] \\ & \mathbf{x}_e(T) \in X_f \subset X_e \end{aligned} \tag{3.16}$$

where X_e is a closed set representing the constraints on the error state; T_e is a compact set containing all the allowed control inputs; and X_f the terminal constraint set.

The standard NMPC algorithm for the AUV tracking control problem can be briefly described as follows:

- □ The optimization problem (3.16) is solved with the current error state $\mathbf{x}_e(t_0)$ as the initial condition. Let $\bar{\boldsymbol{\tau}}_e(t)$ denote the solution.
- □ The system uses $\bar{\boldsymbol{\tau}}_e(t)$ to calculate $\boldsymbol{\tau}$ using (3.13) for only one sampling period: $\boldsymbol{\tau}_e(t) = \bar{\boldsymbol{\tau}}_e(t)$ for $[t_0, t_0 + \delta]$.
- □ At time $t_0 + \delta$, the optimization problem (3.16) is solved again using the latest measured state data $\mathbf{x}_e(t_0 + \delta)$.

The above procedure will repeat until accomplishing the AUV tracking task. However, it is well known that optimality does not automatically guarantee the closed-loop stability due to the finite prediction horizon. To solve this problem, we need to appropriately design the local controller $\boldsymbol{\tau}_e^L$, the terminal state penalty $h(\cdot)$ and the terminal state constraint X_f [88].

Theorem 1. *The closed-loop system (3.14) controlled by the NMPC algorithm is asymptotically stable if the following conditions are satisfied:*

C1: $\mathbf{0} \in X_f$ *and* $\mathbf{0} \in T_e$

C2: There exists a local controller $\boldsymbol{\tau}_e^L(t)$ *satisfying*

(a) $\boldsymbol{\tau}_e^L(t) \in T_e$*, for all* $\mathbf{x}_e \in X_f$

(b) $\mathbf{f}_e(\mathbf{x}_e, \boldsymbol{\tau}_e^L(t)) \in X_f$*, for all* $\mathbf{x}_e \in X_f$

(c) $\dot{g}(\mathbf{x}_e) + \ell(\mathbf{x}_e, \boldsymbol{\tau}_e^L(t)) \leq 0$ *for all* $\mathbf{x}_e \in X_f$*.*

C3: Let $\bar{\boldsymbol{\tau}}_e(t|t_0)$ *denote the solution of the optimization problem (3.16) at time* t_0*, the initial guess for the optimization at the next sampling time instant* $t_0 + \delta$ *is chosen as*

$$\hat{\boldsymbol{\tau}}_e(t|t_0 + \delta) = \begin{cases} \bar{\boldsymbol{\tau}}_e(t|t_0) & \text{for } t_0 + \delta \leq t \leq t_0 + T \\ \boldsymbol{\tau}_e^L(t) & \text{for } t_0 + T \leq t \leq t_0 + T + \delta \end{cases} \tag{3.17}$$

Proof. To prove the stability of the NMPC algorithm, we need to find a Lyapunov function such that this function is nonincreasing along the trajectory of (3.14) controlled by the NMPC algorithm. We take the optimal value function V of the performance index (3.15) as a Lyapunov function candidate .

In the following, the value function is denoted as $V(t, \mathbf{x}_e(t_0))$ implying that the optimal value function depends on the initial error; the state trajectory determined by $\bar{\boldsymbol{\tau}}_e(t)$ is denoted by $\bar{\mathbf{x}}_e(t)$; the state trajectory determined by the initial guess (3.17) is $\hat{\mathbf{x}}_e(t)$ and the corresponding cost is $\hat{V}(t, \mathbf{x}_e)$.

Case I: For two time instants that are within one sampling period, the optimization problem (3.16) is solved only once, then for $0 \leq t_1 \leq t_2 < \delta$, V is nonincreasing since

$$\begin{aligned} V(t_2, \mathbf{x}_e(t_0)) &= V(t_1, \mathbf{x}_e(t_0)) - \textstyle\int_{t_1}^{t_2} \ell(\mathbf{x}_e(s), \boldsymbol{\tau}_e(s))ds \\ &\leq V(t_1, \mathbf{x}_e(t_0)) \end{aligned}$$

Case II: For two time instants that locate in two successive sampling instants, the optimization problem (3.16) is solved twice, then for $0 \leq t_1 < \delta$, and $\delta \leq t_2 < 2\delta$, the following holds:

$$\begin{aligned} & V(t_2, \mathbf{x}_e(t_0+\delta)) - V(t_1, \mathbf{x}_e(t_0)) \\ = \;& V(t_2, \bar{\mathbf{x}}_e(t_0+\delta)) - V(t_1, \mathbf{x}_e(t_0)) \\ \leq \;& \hat{V}(t_2, \bar{\mathbf{x}}_e(t_0+\delta)) - V(t_1, \mathbf{x}_e(t_0)) \\ = \;& g(\hat{\mathbf{x}}_e(T+\delta)) - g(\bar{\mathbf{x}}_e(T)) + \textstyle\int_{t_2}^{T+\delta} \ell(\hat{\mathbf{x}}_e(s), \hat{\boldsymbol{\tau}}_e(s))ds \\ & - \textstyle\int_{t_1}^{T} \ell(\bar{\mathbf{x}}_e(s), \bar{\boldsymbol{\tau}}_e(s))ds \end{aligned}$$

Observing (3.17), we notice that for $s \in [t_2, T]$, $\hat{\mathbf{x}}_e(s) = \bar{\mathbf{x}}_e(s)$ and $\hat{\boldsymbol{\tau}}_e(s) = \bar{\boldsymbol{\tau}}_e(s)$, thus we have

$$\begin{aligned} & V(t_2, \mathbf{x}_e(t_0+\delta)) - V(t_1, \mathbf{x}_e(t_0)) \\ \leq \;& g(\hat{\mathbf{x}}_e(T+\delta)) - g(\hat{\mathbf{x}}_e(T)) + \textstyle\int_{T}^{T+\delta} \ell(\hat{\mathbf{x}}_e(s), \hat{\boldsymbol{\tau}}_e(s))ds \\ & - \textstyle\int_{t_1}^{t_2} \ell(\bar{\mathbf{x}}_e(s), \bar{\boldsymbol{\tau}}_e(s))ds \end{aligned}$$

From $C2$-(c), we have $\dot{g}(\hat{\mathbf{x}}_e) + \ell(\hat{\mathbf{x}}_e, \hat{\boldsymbol{\tau}}_e) \leq 0$, then integrating both sides from T to $T+\delta$ we have

$$g(\hat{\mathbf{x}}_e(T+\delta)) - g(\hat{\mathbf{x}}_e(T)) + \textstyle\int_{T}^{T+\delta} \ell(\hat{\mathbf{x}}_e(s), \hat{\boldsymbol{\tau}}_e(s))ds \leq 0$$

Therefore, we have $V(t_2, \mathbf{x}_e(t_0+\delta)) \leq V(t_1, \mathbf{x}_e(t_0))$.

Case III: For two time instants that locate across several sampling instants, the optimization problem (3.16) is solved several times, then for $k_1\delta \leq t_1 < (k_1+1)\delta$ and $k_2\delta \leq t_2 < (k_2+1)\delta$, with $k_1 < k_2$, by recursively using the result in Case II we have the following inequality:

$$V(t_2) \leq V(k_2\delta) \cdots \leq V((k_1+1)\delta) \leq V(t_1)$$

Furthermore, we have C1, C2-(a) and C2-(b) to guarantee the positively invariance of the system under NMPC algorithm. Now, we can conclude that the optimal value

function V is a valid Lyapunov function for the closed-loop system, and the system (3.14) will be steered to the equilibrium point $(\mathbf{0}, \mathbf{0})$ by the NMPC controller. $\square$

Now, we design the NMPC tracking controller based on **Theorem 1**. First, we construct the initial guess using the way in C3. Second, we select the local controller to be a linear state feedback $\boldsymbol{\tau}_e^L(t) = \mathrm{K}\mathbf{x}_e$. Third, we set the terminal state penalty $h(\mathbf{x}_e) \triangleq \frac{1}{2}\mathbf{x}_e^{\mathrm{T}}\mathbf{x}_e$. Specifically, let $\mathrm{K} \triangleq [\mathrm{K}_1\ \mathrm{K}_2]$ where $\mathrm{K}_1 = \mathrm{diag}(\alpha_1, \beta_1, \gamma_1)$ and $\mathrm{K}_2 = -\mathrm{diag}(\alpha_2, \beta_2, \gamma_2)$ with $\alpha_i > 0, \beta_i > 0$, $\gamma_i > 0$; $\mathbf{Q} = \mathrm{diag}(\mathrm{q}_{11}, \mathrm{q}_{22}, \mathrm{q}_{33}, \mathrm{q}_{44}, \mathrm{q}_{55}, \mathrm{q}_{66})$ and $\mathbf{R} = \mathrm{diag}(\mathrm{r}_{11}, \mathrm{r}_{22}, \mathrm{r}_{33})$ with $q_{ii} > 0, r_{ii} \geq 0$, then we calculate the following:

$$\begin{aligned}
\dot{h}(\mathbf{x}_e) &= \dot{x}_e x_e + \dot{y}_e y_e + \dot{\psi}_e \psi_e + \dot{u}_e u_e + \dot{v}_e v_e + \dot{r}_e r_e \\
&= (y_e r_d - u_e + y_e r_e) x_e + (-x_e r_d - v_e - x_e r_e) y_e \\
&\quad -\psi_e r_e + u_e \tau_u^L + v_e \tau_v^L + r_e \tau_r^L \\
&= -x_e u_e - y_e v_e - \psi_e r_e + u_e \tau_u^L + v_e \tau_v^L + r_e \tau_r^L
\end{aligned}$$

and

$$\begin{aligned}
\ell(\mathbf{x}_e, \boldsymbol{\tau}_e^L(t)) &= q_{11} x_e^2 + q_{22} y_e^2 + q_{33} \psi_e^2 + q_{44} u_e^2 \\
&\quad + q_{55} v_e^2 + q_{66} r_e^2 + r_{11} (\tau_u^L)^2 + r_{22} (\tau_v^L)^2 + r_{33} (\tau_r^L)^2
\end{aligned}$$

Substituting $\boldsymbol{\tau}_e^L(t) = \mathrm{K}\mathbf{x}_e$, we have

$$\begin{aligned}
& \dot{h}(\mathbf{x}_e) + \ell(\mathbf{x}_e, \boldsymbol{\tau}_e^L(t)) \\
= \ & -x_e u_e - v_e y_e - \psi_e r_e + \alpha_1 x_e u_e - \alpha_2 u_e^2 + \beta_1 y_e v_e \\
& -\beta_2 v_e^2 + \gamma_1 \psi_e r_e - \gamma_2 r_e^2 + q_{11} x_e^2 + q_{22} y_e^2 + q_{33} \psi_e^2 \\
& + q_{44} u_e^2 + q_{55} v_e^2 + q_{66} r_e^2 + r_{11} (\alpha_1 x_e - \alpha_2 u_e)^2 \\
& + r_{22} (\beta_1 y_e - \beta_2 v_e)^2 + r_{33} (\gamma_1 \psi_e - \gamma_2 r_e)^2 \\
\\
= \ & (q_{44} - \alpha_2 + r_{11} \alpha_2^2) u_e^2 + (q_{55} - \beta_2 + r_{22} \beta_2^2) v_e^2 \\
& + (q_{66} - \gamma_2 + r_{33} \gamma_2^2) r_e^2 + (q_{11} + r_{11} \alpha_1^2) x_e^2 \\
& + (\alpha_1 - 1 - 2\alpha_1 \alpha_2 r_{11}) x_e u_e + (q_{22} + r_{22} \beta_1^2) y_e^2 \\
& + (\beta_1 - 1 - 2\beta_1 \beta_2 r_{22}) y_e v_e + (q_{33} + r_{33} \gamma_1^2) \psi_e^2 \\
& + (\gamma_1 - 1 - 2\gamma_1 \gamma_2 r_{33}) \psi_e r_e
\end{aligned}$$

We select these parameters which satisfy the following inequalities

$$\begin{aligned} q_{11} + r_{11}\alpha_1^2 + \alpha_1 - 1 - 2\alpha_1\alpha_2 r_{11} &\leq 0 \\ q_{22} + r_{22}\beta_1^2 + \beta_1 - 1 - 2\beta_1\beta_2 r_{22} &\leq 0 \\ q_{33} + r_{33}\gamma_1^2 + \gamma_1 - 1 - 2\gamma_1\gamma_2 r_{33} &\leq 0 \\ q_{44} - \alpha_2 + r_{11}\alpha_2^2 &\leq 0 \\ q_{55} - \beta_2 + r_{22}\beta_2^2 &\leq 0 \\ q_{66} - \gamma_2 + r_{33}\gamma_2^2 &\leq 0 \end{aligned} \tag{3.18}$$

and select the terminal constraint to satisfy

$$\begin{aligned} &|u_e| \geq |x_e|,\ x_e u_e \geq 0 \\ &|v_e| \geq |y_e|,\ y_e v_e \geq 0 \\ &|r_e| \geq |\psi_e|,\ \psi_e r_e \geq 0 \end{aligned} \tag{3.19}$$

We further choose the terminal constraint set to be the sublevel set of $h(\cdot)$

$$X_{f1} = \{\mathbf{x}_e \in \mathbb{R}^6 \mid \frac{1}{2}\mathbf{x}_e^{\mathrm{T}}\mathbf{x}_e \leq \sigma\} \tag{3.20}$$

As a result, the condition C2 is satisfied. Define $X_f = X_{f1} \cap X_{f2}$, where

$$X_{f2} = \{\mathbf{x}_e \in \mathbb{R}^6 \mid (3.19)\} \tag{3.21}$$

Here, we assume $\mathbf{0} \in T_e$, then C1 can be satisfied in X_f.

Remark 4. *The inequalities in (3.18) are easy to satisfy. In fact, these parameters can be divided into three independent groups:* $(q_{11}, q_{44}, r_{11}, \alpha_1, \alpha_2)$, $(q_{22}, q_{55}, r_{22}, \beta_1, \beta_2)$ *and* $(q_{33}, q_{66}, r_{33}, \gamma_1, \gamma_2)$. *The selection among these groups are not coupled. Then for each group only two inequality conditions need to be satisfied.*

Remark 5. *Back to (3.7) and (3.10), we shall notice that the reference system plays a crucial role in the NMPC tracking control. Generally speaking, finding such a reference system is non-trivial, and it might be even more difficult than the stabilization problem itself. Fortunately, for the AUV tracking problem the reference pair* $(\mathbf{x}_d, \boldsymbol{\tau}_d)$ *can be derived directly from the reference path* $S(x)$. *In particular, one possible choice can be determined in the following way:*

□ *Make the calculated spline* $S(x)$ *a reference trajectory of the kinematic state* $\boldsymbol{\eta}_d$

by employing a predetermined timing law:

$$\begin{aligned} x_d &= \rho\ t \\ y_d &= S(x_d) \\ \psi_d &= atan2(\dot{y}_d, \dot{x}_d) \end{aligned}$$

where $\rho > 0$ and $atan2(\cdot,\cdot)$ is the four-quadrant inverse tangent operator.

- □ *Choose the reference velocity $\mathbf{v}_d$ to be $u_d = u_{oc}, v_d = 0$ and $r_d = r_{oc}$.*
- □ *Calculate the reference control input $\boldsymbol{\tau}_d$ through (3.10).*

Remark 6. *For the AUV tracking control, the constraints on control inputs are usually given as follows,*

$$F_{u,min} \leq F_u \leq F_{u,max} \tag{3.22a}$$

$$F_{v,min} \leq F_v \leq F_{v,max} \tag{3.22b}$$

$$F_{r,min} \leq F_r \leq F_{r,max} \tag{3.22c}$$

Then we can derive the control error constraints as

$$\frac{F_{u,min} - F_1}{M_{\dot{u}}} \leq \tau_u \leq \frac{F_{u,max} - F_1}{M_{\dot{u}}} \tag{3.23a}$$

$$\frac{F_{v,min} - F_2}{M_{\dot{v}}} \leq \tau_v \leq \frac{F_{v,max} - F_2}{M_{\dot{v}}} \tag{3.23b}$$

$$\frac{F_{r,min} - F_3}{M_{\dot{r}}} \leq \tau_r \leq \frac{F_{r,max} - F_3}{M_{\dot{r}}} \tag{3.23c}$$

Here, $F_1 = F_u - M_{\dot{u}}\tau_u$, $F_2 = F_v - M_{\dot{v}}\tau_v$, and $F_3 = F_r - M_{\dot{r}}\tau_r$ which can be derived from (3.13a)-(3.13c). Although (3.23) is time-varying with the reference system, it is not necessary to explicitly calculate (3.23). The optimization problem (3.16) is defined to facilitate the description of the stability theorem. In the NMPC implementation, we can directly use the polyhedral input constraint (3.22) to calculate $\boldsymbol{\tau}$ but construct the cost function (3.15) using the elaborately defined error state $\mathbf{x}_e$ and control $\boldsymbol{\tau}_e$. Then the equivalent $\boldsymbol{\tau}_e = \boldsymbol{\tau} - F_F$ with $F_F = [F_1, F_2, F_3]^{\mathrm{T}}$.

Notice that the predetermined timing law can be carefully designed [51, 118] such that the reference system chosen in **Remark 5** fits the actual system. Specifically, when the AUV is moving along the reference spline $S(x)$, the reference control input

can be expressed in an explicit manner

$$F_{ud} = M_{\dot{u}}\dot{u}_d + X_u u_d + D_u u_d|u_d| \tag{3.24a}$$
$$F_{vd} = M_{\dot{u}} u_d r_d \tag{3.24b}$$
$$F_{rd} = M_{\dot{r}}\dot{r}_d + N_r r_d + D_r r_d|r_d| \tag{3.24c}$$

where $\dot{u}_d$ and $\dot{r}_d$ can also be explicitly calculated by

$$\dot{u}_d = S'(x_d)S''(x_d)(1+S'(x_d)^2)^{-\frac{1}{2}}\dot{x}_d^2 + (1+S'(x_d)^2)^{\frac{1}{2}}\ddot{x}_d \tag{3.25a}$$
$$\dot{r}_d = \frac{(1+S'(x_d)^2)S'''(x_d) - 2S'(x_d)S''(x_d)^2}{(1+S'(x_d)^2)^2}\dot{x}_d^2 + \frac{S''(x_d)}{(1+S'(x_d)^2)}\ddot{x}_d \tag{3.25b}$$

Together with the reference state defined in **Remark 5**, it can be observed that we can always adjust x_d, i.e., the value of ρ to satisfy the input constraints. In the simulations, we use a fixed ρ as a simple example.

In this chapter, the thrust allocation problem is solved by conventional pseudo-inverse method, i.e., the calculated generalized generalized thrust forces and moments $\boldsymbol{\tau} = [F_u, F_v, F_r]^{\mathrm{T}}$ will be transformed into real thrusts by $\mathbf{u} = \mathbf{B}^{+}\boldsymbol{\tau}$, where $\mathbf{B}^{+}$ is the Moore-Penrose pseudo-inverse.

3.5 Integrated Receding Horizon Path Planning and Tracking Control: Algorithm Description

In view of the same receding horizon optimization nature, the NMPC tracking control can be conveniently integrated with the proposed path planning method.

Let $\mathcal{D}_i$ denote the measured data for local environment information. The dimension of sampled reference path $S(x)$ between the first and second knot is $\mathcal{M}$. And the prediction horizon $T = \mathcal{N}\delta$ with $\mathcal{N} < \mathcal{M}$. Based on **Problem 1 - 3**, the integrated path planning and tracking control algorithm is summarized in **Algorithm 1**.

In **Algorithm 1** two things need to be clarified. First, within the **for** iterations, we simply use $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ to denote the control process that to apply the calculated control signal for one sampling period and to measure the system state at next sampling instant. Second, since the prediction horizon $\mathcal{N}$ is fixed and the path planning is performed by receding horizon optimizations, when the first segment of the current spline between the first knot and second knot cannot provide enough samples

Algorithm 1 : Integrated Path Planning and Tracking Algorithm

1: $i = 0$.
2: Let $\mathbf{x}_0$ denote the initial state of the AUV.
3: Given sensor data $\mathcal{D}_i$, calculate the reference path by solving **Problem 1**. Let $S_i(x)$ be the solution.
4: **procedure**
5: $k = 0$.
6: **for** $k \leq \mathcal{M} - \mathcal{N}$ **do**
7: Given $S_i(x)$ calculate the reference pair $(\mathbf{x}_d, \boldsymbol{\tau}_d)$ according to **Remark 5**.
8: Solve **Problem 3** and use (3.13) to calculate $\bar{\boldsymbol{\tau}}(t)$; $\mathbf{u} = \mathbf{B}^{+}\bar{\boldsymbol{\tau}}$.
9: Implement the control: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ for $k\delta \leq t \leq (k+1)\delta$.
10: $k = k + 1$.
11: **end for**
12: $i = i + 1$.
13: Collect another set of sensor data $\mathcal{D}_i$.
14: Calculate reference path by solving **Problem 2**; let $S_i(x)$ be the solution.
15: **for** $k \leq \mathcal{M}$ **do**
16: Given $S_{i-1}(x)$ and $S_i(x)$ calculate $(\mathbf{x}_d, \boldsymbol{\tau}_d)$ according to **Remark 5**.
17: Solve **Problem 3** and use (3.13) to calculate $\bar{\boldsymbol{\tau}}(t)$; $\mathbf{u} = \mathbf{B}^{+}\bar{\boldsymbol{\tau}}$.
18: Implement the control: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ for $k\delta \leq t \leq (k+1)\delta$.
19: Let $\mathbf{x}_f$ denote the system state at time $(k+1)\delta$.
20: $k = k + 1$.
21: **end for**
22: **end procedure**
23: $\mathbf{x}_0 = \mathbf{x}_f$, repeat **procedure.**

of reference state $\mathbf{x}_d(k)$ from $k = 1$ to $\mathcal{N}$, i.e., $\mathcal{M} - k < \mathcal{N}$, another round of path planning needs to be performed starting from the second knot of the current spline.

The block diagram which represents the integrated control scheme consisting of the path planner and the NMPC tracking controller, is depicted in Figure. 3.2. With the aid of block diagram, we can explain the **procedure** in **Algorithm 1** as follows:

- The local environment $\mathcal{D}_i$ is detected. The mathematical representation of the workspace (c_1, c_2) is calculated according to current state $\mathbf{x}$ of the AUV, based on which the path planner constructs a receding horizon optimization problem, i.e., **Problem 2** to determine the minimum curvature spline path $S_i(x)$.

- The path $S_i(x)$ is then augmented and viewed as the trajectory of a virtual reference system which provides the reference state $(\mathbf{x}_d, \boldsymbol{\tau}_d)$ for the NMPC tracking controller. A feed-forward channel exists to calculate the control effort F_F corresponding to the compensation of the time-varying portion of the reference

path. Based on an elaborately defined error dynamics $(\mathbf{x}_e, \boldsymbol{\tau}_e)$, the receding horizon optimization problem, i.e., **Problem 3** is constructed and then solved to obtain the tracking control signal $\mathbf{u} = \mathbf{B}^{+}\bar{\boldsymbol{\tau}}$.

□ As the AUV moves, the newly measured data will be collected and used to generate an updated reference path. Repeat **procedure**.

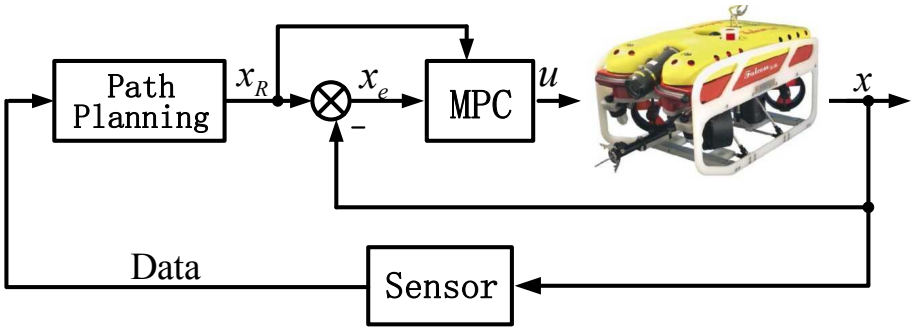

Figure 3.2: The closed-loop control block diagram.

For the NMPC tracking control, it is worth noting that at each sampling instant, only $\mathcal{N}$ steps of the reference states are used to construct the optimization problem. According to **Algorithm 1**, we have $\mathcal{N} < \mathcal{M}$ and every time (except for the first time) the reference path is planned starting from $\mathcal{N}$ steps ahead of the vehicle position. In this case, the reference path is equivalent to a predefined one although it is actually generated as the vehicle moves forward. The closed-loop stability of the tracking control will not be affected by the integration of the receding horizon path planning.

3.6 Simulation Results

3.6.1 Parameter Selection

For the path planning, it is assumed that the vehicle can sense the environment up to 5 meters ahead; we uniformly distribute 6 knots on the 5-meter range and insert 4 samples between knots; choose $o = 4$, $l = 4$ and update the reference path with current 5-meter data whenever the vehicle moves 1 meter forward in the x-axis. The tolerances for discontinuity are $\epsilon_1 = \epsilon_2 = \epsilon_3 = 10^{-2}$ (m).

For NMPC parameters, we use $\rho = 1$, $\delta = 0.1sec$, the prediction horizon $T = 8\delta$. The reference system is chosen according to **Remark 5**. The weighting matrices

are selected as $\mathbf{Q} = \text{diag}(q_{11}, q_{22}, q_{33}, q_{44}, q_{55}, q_{66}) = 0.4\ \mathbf{I}_6$, $\mathbf{R} = \text{diag}(r_{11}, r_{22}, r_{33}) = 0.01\mathbf{I}_3$, $K_1 = \text{diag}(\alpha_1, \beta_1, \gamma_1) = 0.5\mathbf{I}_3$ and $K_2 = -\text{diag}(\alpha_2, \beta_2, \gamma_2) = -\mathbf{I}_3$. It can be verified that these parameters satisfy (3.18).

3.6.2 Tracking Performance

In the simulations, the optimization **Problem 1 - 3** are solved by the embedded sequential quadratic programming (SQP) algorithm in the Matlab function *fmincon*.

The simulation results of the combined path planning and tracking control is shown in Figure 3.3. It can be observed that the generated reference path is smooth in

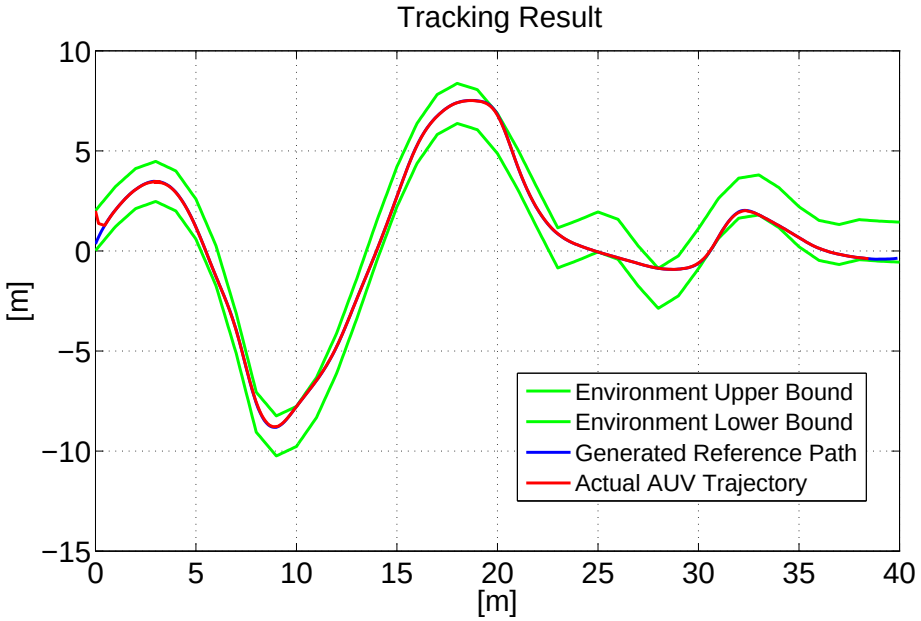

Figure 3.3: Simulation results of the combined AUV problem.

light of small curvature, which facilitates the tracking control of the AUV. The NMPC controller successfully steers the AUV asymptotically convergent to and then track the reference path in the presence of initial error. The average solution time per NMPC step of is 1.9642 second which needs to be improved for real-time implementation. The computational complexity issue will be further investigated in Chapter 6. Also one can refer to the excellent review paper [35] for possible solutions.

Figure 3.4 shows the generalized control forces and moments, and Figure 3.5 depicts the transformed real thrust forces. Here, we shall notice that the control inputs are not optimal in terms of fuel consumption (minimum control force) due

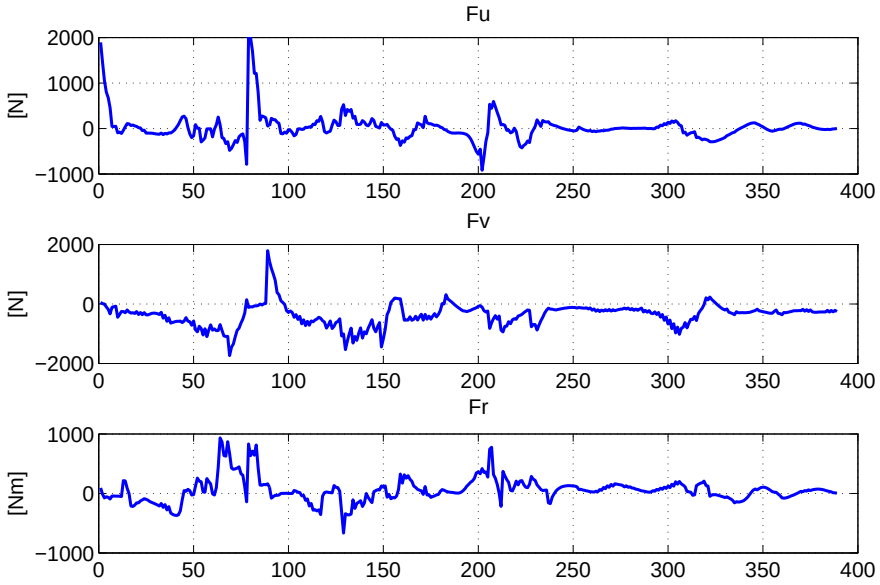

Figure 3.4: The generalized control signal.

to the selection of the reference system. As mentioned earlier, unlike the mobile robot whose sway velocity v is eliminated by friction, in water, the AUV will always slip sideways when taking turns. Therefore, although we are able to conveniently find the reference system as stated in **Remark 5**, the AUV has to compensate the sway velocity v all the time, which increases the control effort. Nevertheless, if we

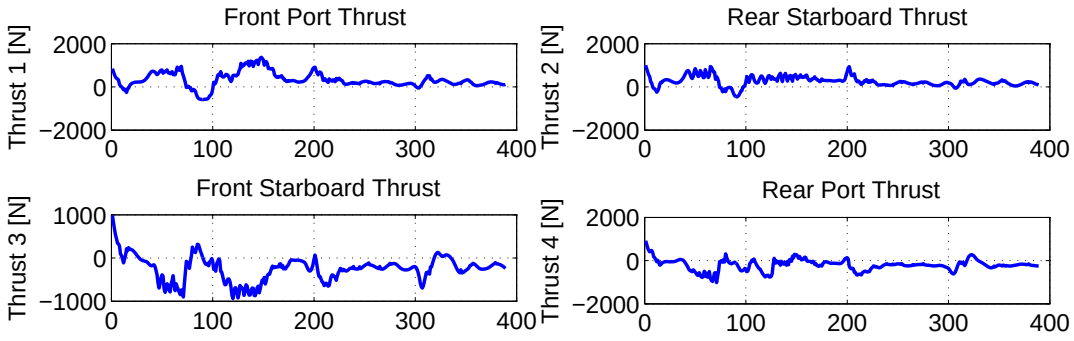

Figure 3.5: The real thrust forces.

view this point via another perspective, the compensation of v poses a request for the minimum curvature reference path, which is consistent with the proposed path

planning method.

3.7 Conclusion

In this chapter, we have presented a unified receding horizon optimization (RHO) framework for solving the combined path planning and tracking control problem of an AUV. Firstly, a novel RHO-based dynamic path planning method was proposed for the AUV to generate a minimum curvature reference path in the constrained workspace. With the RHO formulation, the dynamic path planning can well accommodate the finite perceiving capability of the vehicle. Then, the nonlinear model predictive control (NMPC) was employed for the AUV to precisely track the planned reference path. Sufficient conditions for closed-loop stability were explored and further applied to guide the NMPC controller design. Finally, an implementation algorithm inherently combining NMPC and the proposed path planning method was developed. With the implementation algorithm the obtained closed-loop stability of the NMPC tracking control can be ensured. Simulation results on the Falcon dynamic model with the identified model parameters revealed the effectiveness of the proposed control algorithm.

The main results in this chapter have been published in [123].

[123] **C. Shen**, Y. Shi and B. Buckham, "Integrated Path Planning and Tracking Control of an AUV: A Unified Receding Horizon Optimization Approach", *IEEE/ASME Transactions on Mechatronics*, vol.22, no.3, pp.1163-1173, 2017.

Chapter 4

Lyapunov-based Model Predictive Control for Dynamic Positioning and Trajectory Tracking Control of an AUV

4.1 Introduction

4.1.1 Research Background and Contributions

The dynamic positioning (DP) is an important part of the AUV control system. Traditionally, the DP functionality refers to the vehicle's capability of reaching and maintaining at a desired position with a desired attitude by exclusively means of active thrusters. In recent studies, it tends to encompass all of the low-speed maneuvering into the definition of DP [128]. The most widely used DP controllers are of the proportional-integral-derivative (PID) type since they are model free and easy to implement. Acceleration feedback can be articulated with the PID controller to get improved control stability [46]. For model-based control, in [129], the linear-quadratic-Gaussian (LQG) technique is applied to the DP control design with a linearized model. An H_∞ controller is proposed in [71] which makes trade-off between track keeping and station keeping. Obviously, the DP controllers based on linear control techniques only obtain local stability properties. Nonlinear DP control design, therefore, comes in for better closed-loop properties. The mainstream nonlinear control methods for

DP include the Lyapunov-based backstepping and sliding mode control. In [134], the observer-controller structure is investigated for the DP control, and a Lyapunov-based backstepping nonlinear control law is designed. Global exponential stability can be claimed. The nonlinear sliding model DP control can be found in [133]. Experiments are conducted to demonstrate the improved control performance comparing to the conventional PID controller. Most recent DP control designs begin to consider more practical issues. The system constraints such as control limit and safe operating region are inevitable in real applications. Therefore, it would be preferred if these constraints can be considered in the DP controller design.

Model predictive control (MPC), being an optimization-based time domain control technique, is known as one of few control methods that can explicitly incorporate constraints into the controller design phase [88, 87, 79]. In context of DP control, the MPC solution is reported in [136] where the simulation results indicate that MPC offers significantly improved control performance. In addition, the thrust allocation (TA) subproblem, i.e., the coordination of thrusters to generate the matched generalized force and moment requested by the DP control law, can be solved simultaneously with the LMPC-based DP control. Nevertheless, the big question associated with the MPC control design is how to guarantee the closed-loop stability, especially for nonlinear systems such as AUVs. The optimality is not necessarily bringing stability for the closed-loop system. To obtain this important closed-loop property, in standard MPC design precedure we need to employ additional terminal constraints to the formulated optimization problem, and to construct a local stabilizing control law via local linearization [88]. This essentially introduces considerable conservativeness and only local stability can be guaranteed. For DP of AUVs, since the system is highly nonlinear, often the stability obtained by local linearization is not practically useful. One potential solution is to circumvent this question by means of Lyapunov-based model predictive control (LMPC) [30, 92]. The LMPC method combines the merits of both Lyapunov-based control and model predictive control, which makes the MPC design relatively simple and intuitive. More importantly, the LMPC controller inherits the stability property of the Lyapunov-based control, which is attractive for the AUV DP control since many existing global DP controllers can be exploited.

In this chapter, we address the DP control problem for an AUV in the local level plane. To handle the highly nonlinear, cross-coupled dynamics as well as system constraints in terms of thrust limits, the Lyapunov-based nonlinear model predictive control strategy is proposed. Within the LMPC framework, we do not need to solve

the TA subproblem separately since the thruster distribution can be formulated into the optimal control problem directly. Considering that most DP controllers are of the PID type, the nonlinear PD control is adopted as the auxiliary control law. The recursive feasibility and closed-loop stability of the LMPC control system are explicitly analyzed. Since the nonlinear PD control is globally asymptotically stable, the LMPC DP control obtains a quasi-global closed-loop stability: the region of attraction (ROA) can be arbitrarily specified by the design parameters.

Furthermore, the obtained LMPC results in DP control are also extended to solve the trajectory tracking control problem for AUVs. To construct the contraction constraint in the LMPC problem formulation, the Lyapunov-based backstepping procedure is exploited. Sufficient conditions that ensure the recursive feasibility and closed-loop stability are provided analytically. Compared with the NMPC tracking control explored in Chapter 3, the main advantage of the LMPC method is that it builds on the existing AUV control system and utilizes onboard computing resources (online optimization) to improve the control performance. Since the closed-loop stability of LMPC-based control system does not rely on the exact solution of the optimization, we can easily control the computational complexity by specifying the maximum iteration number for the NLP solver. In addition, from the control-theoretic perspective, another advantage of LMPC is that using LMPC we are able to characterize an explicit ROA while with the NMPC we can only define the ROA in an implicit manner.

The main contributions of this chapter are summarized as follows,

- A novel LMPC framework is developed for the AUV motion control problems, providing a bridge connecting the modern optimization theory and the conventional marine control technology. The control performance and robustness can be significantly improved by the online optimization.

- Based on the LMPC framework, a dynamic positioning controller (DP) is designed for the AUV. Sufficient conditions for recursive feasibility and closed-loop stability are explicitly derived. With the nonlinear PD control law adopted as the auxiliary controller, we show that the region of attraction of the LMPC-based DP control can be specified arbitrarily large.

- An LMPC-based trajectory tracking controller is also designed. Sufficient conditions for recursive feasibility and closed-loop stability are provided. A guaranteed region of attraction is characterized analytically.

- With the LMPC framework the thrust allocation (TA) subproblem can be solved simultaneously with the AUV motion control, which reduces the conservativeness brought by conventional TA solutions.

4.1.2 Chapter Organization

The remaining part of this chapter is organized as follows: Section 4.2 proposes the LMPC framework and details the DP controller design. In Section 4.3, the LMPC method is applied to solve the AUV trajectory tracking problem; controller design and stability analysis are provided. In Section 4.4, the conclusion is made for this chapter.

Throughout this chapter, the following notations are used: the column operation $[\rho_1^{\mathrm{T}}, ...\rho_n^{\mathrm{T}}]^{\mathrm{T}}$ is denoted as $col(\rho_1, ..., \rho_n)$; the diagonal operation is denoted by $\mathrm{diag}(\cdot)$; the square of a weighted Euclidean norm $\rho^{\mathrm{T}} A \rho$ is abbreviated by $\|\rho\|_A^2$; and the infinity norm and 2 norm are denoted by $\|\cdot\|_\infty$ and $\|\cdot\|_2$, respectively. The symbol $\mathbf{1}$ is used to represent a column vector with all elements to be 1. The $\max\{\cdot\}$ is a function that returns the largest element within the brace. The absolute value operator applying on a vector $|\rho|$ takes the absolute value for each and every element.

4.2 LMPC Design for Dynamic Positioning Control

4.2.1 Problem Formulation

The AUV model studied for the DP control and the subsequent trajectory tracking control keeps the same as the one used in Chapter 3 which is established based on the kinematic equations, the dynamic equations and the thrust distribution:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{R}(\psi)\mathbf{v} \\ \mathbf{M}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\boldsymbol{\eta})) \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{4.1}$$

where the state vector $\mathbf{x} = [x, y, \psi, u, v, r]^{\mathrm{T}}$ is consisted of the pose and velocity of the vehicle, and the control vector $\mathbf{u} = [u_1, u_2, u_3, u_4]^{\mathrm{T}}$ is consisted of the forces generated by the four thrusters. The detailed expression can be found in (2.32) and (2.34).

The DP control refers to the feedback control of marine vehicles to reach and

maintain the desired position at the preferred orientation by exclusively means of thrusters. Consider the desired position and orientation given by $\boldsymbol{\eta}_d = [x_d, y_d, \psi_d]^{\mathrm{T}}$. An intuitive MPC formulation can be established for the DP control

$$\begin{aligned} \min_{\hat{\mathbf{u}} \in S(\delta)} \quad & J = \textstyle\int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + \|\hat{\mathbf{u}}(s)\|_R^2) ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \\ \text{s.t.} \quad & \dot{\hat{\mathbf{x}}}(s) = \mathbf{f}(\hat{\mathbf{x}}(s), \hat{\mathbf{u}}(s)) \\ & \hat{\mathbf{x}}(0) = \mathbf{x}(t_0) \\ & |\hat{\mathbf{u}}(s)| \leq \mathbf{u}_{\max} \end{aligned} \tag{4.2}$$

where $\hat{\mathbf{x}}(s)$ is the predicted state trajectory of the vehicle, evolving using the system model; $\tilde{\mathbf{x}} = col(\tilde{\boldsymbol{\eta}}, \hat{\mathbf{v}})$ denotes the error state in which $\tilde{\boldsymbol{\eta}} = \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d$; $S(\delta)$ denotes the family of piecewise constant functions characterized by the sampling period δ. $T = N\delta$ is the prediction horizon and Q, R, P are weighting matrices, positive definite.

However, even if the global optimal solution curve $\hat{\mathbf{u}}^*(s)$ can be obtained, the closed-loop stability is not guaranteed due to the finite prediction horizon. Complex offline design procedure appears to be a necessity. For nonlinear systems, basically, local linearization with respect to equilibrium point has to take place to facilitate the choice of weighting matrices as well as the construction of the auxiliary local feedback control [87]. Therefore, only local stability property can be claimed.

In view of the fact that many existing DP controllers are designed using the Lyapunov direct method and obtain the global stability property, we can explicitly take advantage of these existing controllers to formulate the LMPC problem (P_0) for the DP control

$$\min_{\hat{\mathbf{u}} \in S(\delta)} J = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + \|\hat{\mathbf{u}}(s)\|_R^2) ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{4.3a}$$

$$\text{s.t.} \quad \dot{\hat{\mathbf{x}}}(s) = \mathbf{f}(\hat{\mathbf{x}}(s), \hat{\mathbf{u}}(s)) \tag{4.3b}$$

$$\hat{\mathbf{x}}(0) = \mathbf{x}(t_0) \tag{4.3c}$$

$$|\hat{\mathbf{u}}(s)| \leq \mathbf{u}_{\max} \tag{4.3d}$$

$$\tfrac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\hat{\mathbf{x}}(0), \hat{\mathbf{u}}(0)) \leq \tfrac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\hat{\mathbf{x}}(0), h(\hat{\mathbf{x}}(0))) \tag{4.3e}$$

where $h(\cdot)$ is the existing DP controller and $V(\cdot)$ is the corresponding Lyapunov function. The presence of contraction constraint (4.3e) allows us to show that the LMPC

inherits the stability properties of the state feedback control $h(\mathbf{x})$ [30]. Therefore, if $h(\mathbf{x})$ brings the global stability of desired equilibrium point, the system under LMPC control is also globally stable. Meanwhile, thanks to the online optimization procedure, the LMPC controller can improve the DP control performance considerably.

The LMPC-based DP control algorithm will be implemented in the standard receding horizon control fashion: (i) at the current sampling instant t_0, given the system state $\mathbf{x}(t_0)$, the optimal control problem (P_0) is solved; let $\boldsymbol{\kappa}(s)$ denote the (sub-)optimal solution; (ii) the vehicle implements $\boldsymbol{\kappa}(s)$ for only one sampling period: $\mathbf{u}(t) = \boldsymbol{\kappa}(s)$ for $s \in [0, \delta]$; (iii) at next sampling instant $t_0 + \delta$, new measurement of system state $\mathbf{x}(t_0 + \delta)$ is fed back, and (P_0) will be solved again with the new initial condition $\mathbf{x}(t_0 + \delta)$. Then repeat from (i).

4.2.2 Main Results

Since the universally used DP controllers are of the PID type, we investigate the multivariable PD control for the purpose of constructing the contraction constraint in the LMPC problem (4.3). However, any other Lyapunov-based nonlinear controller can, in principle, be used.

Consider the following nonlinear PD control law:

$$\boldsymbol{\tau}(\mathbf{x}) = \mathbf{g}(\boldsymbol{\eta}) - \mathbf{R}^{\mathrm{T}}(\psi)\boldsymbol{\tau}_{PD} \tag{4.4}$$

$$\boldsymbol{\tau}_{PD} = \mathbf{K}_p\tilde{\boldsymbol{\eta}} + \mathbf{K}_d\dot{\boldsymbol{\eta}} \tag{4.5}$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are the user specified control gain matrices which are diagonal and positive definite.

The Lyapunov function candidate is suggested as follows:

$$V = \frac{1}{2}\mathbf{v}^{\mathrm{T}}\mathbf{M}\mathbf{v} + \frac{1}{2}\tilde{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} \tag{4.6}$$

Taking time derivative of V along the trajectory of the closed-loop system, we have

$$\dot{V} = \mathbf{v}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{v}} + \dot{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} \tag{4.7}$$

Substituting (2.31), (2.33), (4.4) and (4.5) into (4.7) yields

$$\dot{V} = -\mathbf{v}^{\mathrm{T}}[\mathbf{C}(\mathbf{v}) + \mathbf{D}(\mathbf{v}) + \mathbf{K}_d^*(\boldsymbol{\eta})]\mathbf{v} \tag{4.8}$$

where $\mathbf{K}_d^*(\boldsymbol{\eta}) = \mathbf{R}^{\mathrm{T}}(\psi)\mathbf{K}_d\mathbf{R}(\psi)$. Considering $\mathbf{v}^{\mathrm{T}}\mathbf{C}(\mathbf{v})\mathbf{v} = 0$ for all $\mathbf{v}$, we have

$$\dot{V} = -\mathbf{v}^{\mathrm{T}}[\mathbf{D}(\mathbf{v}) + \mathbf{K}_d^*(\boldsymbol{\eta})]\mathbf{v} \leq 0 \tag{4.9}$$

since it can be easily shown that $\mathbf{K}_d^*(\boldsymbol{\eta}) > 0$. By LaSalle's theorem [75], the closed-loop system with the nonlinear PD controller is globally asymptotically stable with respect to the equilibrium $[\tilde{\boldsymbol{\eta}}, \mathbf{v}] = [\mathbf{0}, \mathbf{0}]$.

Then the detailed expression of the contraction constraint (4.3e) corresponding to the nonlinear PD control is

$$\begin{aligned}&\hat{\mathbf{v}}(0)^{\mathrm{T}}(\hat{\mathbf{u}}(0) - \mathbf{C}(\hat{\mathbf{v}}(0))\hat{\mathbf{v}}(0) - \mathbf{D}(\hat{\mathbf{v}}(0))\hat{\mathbf{v}}(0) - \mathbf{g}(\hat{\boldsymbol{\eta}}(0)) + \mathbf{R}^{\mathrm{T}}(\hat{\psi}(0))\mathbf{K}_p\tilde{\boldsymbol{\eta}}(0)) \\ &\leq -\hat{\mathbf{v}}(0)^{\mathrm{T}}[\mathbf{D}(\hat{\mathbf{v}}(0)) + \mathbf{K}_d^*(\hat{\boldsymbol{\eta}}(0))]\hat{\mathbf{v}}(0)\end{aligned} \tag{4.10}$$

For the recursive feasibility, we notice that the PD controller $h(\hat{\mathbf{x}})$ is always feasible for the LMPC problem (4.3) provided that $|h(\hat{\mathbf{x}})| \leq \mathbf{u}_{\max}$ can be satisfied.

For calculation simplicity, in the following we make several reasonable and practical assumptions.

Assumption 1. *The thrusters have the same maximum capacity, i.e.,* $|u_i| \leq u_{max}$.

Note that **Assumption 1** is reasonable and often true in practice. Then we have the following proposition.

Proposition 1. *Consider the TA using the Moore - Penrose pseudoinverse implementation, i.e.,*

$$\mathbf{u} = (\mathbf{B}\mathbf{B}^{\mathrm{T}})^{-1}\mathbf{B}^{\mathrm{T}}\boldsymbol{\tau} = \mathbf{B}^{+}\boldsymbol{\tau} \tag{4.11}$$

and denote the maximum possible generalized thrust force by $\tau_{max} = \|\boldsymbol{\tau}_{max}\|_\infty$ *with* $\boldsymbol{\tau}_{max} = [F_{u,max}, F_{v,max}, F_{r,max}]^{\mathrm{T}}$.

If the following relation holds,

$$\tau_{max} \leq \frac{u_{max}}{\bar{b}^{+}} \tag{4.12}$$

where $\bar{b}^{+} = \|\mathbf{B}^{+}\|_\infty$, *then the TA is always feasible, i.e.,* $\|\mathbf{u}\|_\infty \leq u_{max}$.

Proof. Taking the infinity norm on both sides of (4.11) we have

$$\|\mathbf{u}\|_\infty = \|\mathbf{B}^{+}\boldsymbol{\tau}\|_\infty \leq \bar{b}^{+}\|\boldsymbol{\tau}\|_\infty \leq \bar{b}^{+}\tau_{\max} \tag{4.13}$$

Having (4.12) and **Assumption 1**, it follows

$$\|\mathbf{u}\|_\infty \leq \bar{b}^+ \tau_{\max} \leq u_{\max} \tag{4.14}$$

□

Assumption 2. *The restoring force* $\mathbf{g}(\boldsymbol{\eta})$ *is bounded and relatively small such that*

$$\|\mathbf{g}(\boldsymbol{\eta})\|_\infty \leq \bar{g} < \tau_{max} \tag{4.15}$$

where $\bar{g}$ *denotes the bound.*

The **Assumption 2** is also reasonable. The detailed expression of $\mathbf{g}(\boldsymbol{\eta})$ can be found in (2.24) which contains combinations of sine and cosine functions. Therefore, the boundedness of the restoring force can be guaranteed. Furthermore, the bound $\bar{g}$ is relatively small with respect to the maximum allowed thrust force $\tau_{\max}$. If not, from (4.4) we see that there would be no room for the feedback control, which is not considered in this dissertation. Then we have the following theorem.

Theorem 2. *Suppose the control gains* $\mathbf{K}_p = \mathrm{diag}\{k_{pi}\}$ *and* $\mathbf{K}_d = \mathrm{diag}\{k_{di}\}$. *Let* $\bar{k}_p = \max\{k_{pi}\}$ *denote the largest element in* $\mathbf{K}_p$ *and* $\bar{k}_d = \max\{k_{di}\}$ *denote the largest element in* $\mathbf{K}_d$. *Suppose* **Assumption 1** *and* **Assumption 2** *can hold and define* $h(\mathbf{x}) = \mathbf{B}^+ \boldsymbol{\tau}(\mathbf{x})$. *If the following relation holds,*

$$(\bar{k}_p + \sqrt{2}\bar{k}_d)\|\tilde{\mathbf{x}}(0)\|_2 \leq \frac{\tau_{max} - \bar{g}}{\sqrt{2}} \tag{4.16}$$

where $\tilde{\mathbf{x}}(0)$ *is the initial error and* τ_{max} *follows (4.12), then the LMPC* (P_0) *admits recursive feasibility, i.e.,* $|h(\hat{\mathbf{x}}(t))| \leq \mathbf{u}_{max}$ *for all* $t \geq 0$ *where* $\mathbf{u}_{max} = u_{max}\mathbf{1}$.

Proof. Taking infinity norm on both sides of (4.4) yields

$$\begin{aligned} \|\boldsymbol{\tau}\|_\infty &= \|\mathbf{g}(\boldsymbol{\eta}) - \mathbf{R}^{\mathrm{T}}(\psi)\boldsymbol{\tau}_{PD}\|_\infty \\ &\leq \|\mathbf{g}(\boldsymbol{\eta})\|_\infty + \|\mathbf{R}^{\mathrm{T}}(\psi)\|_\infty \|\boldsymbol{\tau}_{PD}\|_\infty \\ &\leq \bar{g} + \sqrt{2}\|\boldsymbol{\tau}_{PD}\|_\infty \end{aligned} \tag{4.17}$$

since $\|\mathbf{R}^{\mathrm{T}}(\psi)\|_\infty = \max\{\cos\psi - \sin\psi, \sin\psi + \cos\psi, 1\} \leq \sqrt{2}$.

From (2.33) and (4.5), we have

$$\begin{aligned} \|\boldsymbol{\tau}_{PD}\|_\infty &= \|\mathbf{K}_p\tilde{\boldsymbol{\eta}} + \mathbf{K}_d\dot{\boldsymbol{\eta}}\|_\infty = \|\mathbf{K}_p\tilde{\boldsymbol{\eta}} + \mathbf{K}_d\mathbf{R}(\psi)\mathbf{v}\|_\infty \\ &\leq \bar{k}_p\|\tilde{\boldsymbol{\eta}}\|_\infty + \sqrt{2}\bar{k}_d\|\mathbf{v}\|_\infty \\ &\leq (\bar{k}_p + \sqrt{2}\bar{k}_d)\|\tilde{\mathbf{x}}\|_\infty \end{aligned} \tag{4.18}$$

Since (4.3e) is satisfied, it admits $\dot{V} \leq 0$. Therefore, $\|\tilde{\mathbf{x}}\|_2 \leq \|\tilde{\mathbf{x}}(0)\|_2$. Considering $\|\tilde{\mathbf{x}}\|_\infty \leq \|\tilde{\mathbf{x}}\|_2$, we have

$$\|\boldsymbol{\tau}_{PD}\|_\infty \leq (\bar{k}_p + \sqrt{2}\bar{k}_d)\|\tilde{\mathbf{x}}(0)\|_2 \tag{4.19}$$

Together with (4.17), we have

$$\|\boldsymbol{\tau}\|_\infty \leq \bar{g} + \sqrt{2}(\bar{k}_p + \sqrt{2}\bar{k}_d)\|\tilde{\mathbf{x}}(0)\|_2 \tag{4.20}$$

If (4.16) can be satisfied, then the following relation holds

$$\|\boldsymbol{\tau}\|_\infty \leq \bar{g} + \sqrt{2}(\bar{k}_p + \sqrt{2}\bar{k}_d)\|\tilde{\mathbf{x}}(0)\|_2 \leq \tau_{\max} \tag{4.21}$$

With (4.12), we can guarantee that $\|h(\hat{\mathbf{x}}(t))\|_\infty \leq u_{\max}$ is satisfied all the time, which completes the proof. □

We notice that the (4.16) can be easily satisfied since $\bar{k}_p$ and $\bar{k}_d$ can be specified as arbitrarily small positive numbers. As the recursive feasibility implies the closed-loop stability, the region of attraction can be arbitrarily large.

Definition 2 (**Class $\mathcal{K}_\infty$ Function**). *A continuous function $\alpha : [0, \infty) \to [0, \infty)$ is said to belong to $\mathcal{K}_\infty$ if*

- *it is strictly increasing*
- *it is such that $\alpha(0) = 0$ and $\lim_{r\to\infty} \alpha(r) = \infty$*

Theorem 3. *Suppose* **Assumption 1** *and* **Assumption 2** *can hold, then the LMPC-based DP control makes the desired equilibrium point $[\tilde{\boldsymbol{\eta}}, \mathbf{v}] = [\mathbf{0}, \mathbf{0}]$ asymptotically stable. Furthermore, the region of attraction can be arbitrarily large with enough small control gains $\bar{k}_p$ and $\bar{k}_d$.*

Proof. The proof first shows that the equilibrium is asymptotically stable, and then illustrates that the region of attraction can be arbitrarily large.

Since we have already found a Lyapunov function $V(\mathbf{x})$ in (4.6), continuously differentiable and radically unbounded, by converse Lyapunov theorems [75], there exist functions $\alpha_i(\cdot)$, $i = 1, 2, 3$ belonging to class $\mathcal{K}_\infty$ such that the following inequalities can hold:

$$\alpha_1(\|\mathbf{x}\|) \leq V(\mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|) \tag{4.22a}$$

$$\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, h(\mathbf{x})) \leq -\alpha_3(\|\mathbf{x}\|) \tag{4.22b}$$

Considering (4.3e) and that only the first element of $\boldsymbol{\kappa}(\mathbf{x})$ will be implemented for each sampling period, we have

$$\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})) \leq \frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, h(\mathbf{x})) \leq -\alpha_3(\|\mathbf{x}\|) \tag{4.23}$$

By standard Lyapunov arguments (e.g., Theorem 4.8 in [75]) we claim that the closed-loop system with the LMPC control $\mathbf{u}(\mathbf{x})$ is asymptotically stable with a region of attraction

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid (\bar{k}_p + \sqrt{2}\bar{k}_d)\|\tilde{\mathbf{x}}\|_2 \leq \frac{\tau_{\max} - \bar{g}}{\sqrt{2}}\} \tag{4.24}$$

where $\tilde{\mathbf{x}} = col(\tilde{\boldsymbol{\eta}}, \mathbf{v})$ denotes the error state.

Obviously, with arbitrarily large initial error $\tilde{\mathbf{x}}$, we choose the control gains $\mathbf{K}_p > 0$ and $\mathbf{K}_d > 0$ satisfying

$$(\bar{k}_p + \sqrt{2}\bar{k}_d) \leq \frac{\tau_{\max} - \bar{g}}{\sqrt{2}\|\tilde{\mathbf{x}}\|_2} \tag{4.25}$$

Then the LMPC problem is feasible and the closed-loop system is stable. Since there are no other constraints on $\bar{k}_p$ and $\bar{k}_d$, the region of attraction can be arbitrarily large with enough small control gains satisfying (4.25). □

Remark 7. *Although the asymptotical stability relies only on the positive definiteness of the control gain matrices $\mathbf{K}_p$ and $\mathbf{K}_d$, the control performance of the PD controller is determined by the magnitude of the control gains. Smaller control gains will result in slower convergence. However, for the proposed LMPC-based DP control, due to the optimization procedure, it can automatically make full use of the thrust capability to generate the best achievable control performance, with respect to the objective function (4.3a), even though we have selected very small control gains for a large region of attraction.*

4.2.3 Simulation Results

In this section, the proposed LMPC-based DP control is simulated on the dynamic model of Falcon which were identified through experimental data [109]. The AUV model parameters are the same as those in the previous chapter and the details have been summarized in Table 2.1.

Without loss of generality, the desired position is chosen to be the origin of the IRF, i.e., $\boldsymbol{\eta}_d = [0, 0, 0]^{\mathrm{T}}$. The LMPC problem (4.3) is discretized and then solved by the sequential quadratic programming (SQP) method.

The controller parameters are chosen as follows: The sampling period $\delta = 0.1sec$, prediction horizon $T = 5\delta$, and the weighting matrices $Q = \mathrm{diag}(10^5,\ 10^5,\ 10^4,\ 10^3,\ 10^3,\ 10^3)$, $R = \mathrm{diag}(10^{-3},\ 10^{-3},\ 10^{-3},\ 10^{-3})$ and $P = \mathrm{diag}(10^3,\ 10^3,\ 10^2,\ 10,\ 10,\ 10)$. The nonlinear PD control gains $\mathbf{K}_p = \mathbf{K}_d = \mathrm{diag}(10, 10, 10)$. The initial condition is $\mathbf{x}(0) = [5, 5, -\pi/2, 0, 0, 0]^{\mathrm{T}}$.

The simulated AUV trajectories are shown in Figure 4.1 and the state trajectories with respect to time are illustrated in Figure 4.2.

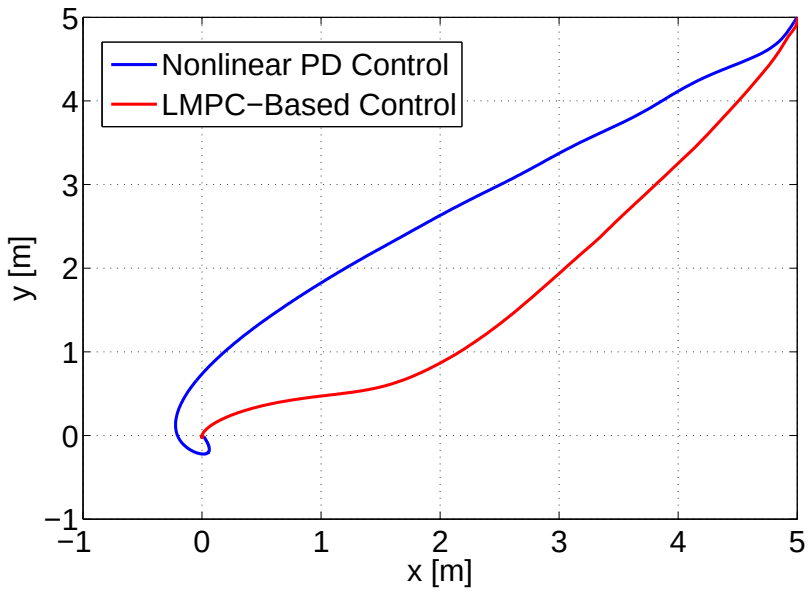

Figure 4.1: The AUV trajectory in local level plane.

As we can see, both LMPC and PD controller steer the AUV successfully to the desired set-point. However, the LMPC-based DP control converges much faster (in about 20 sec.) than the PD control (in about 40 sec.). This is because we have selected relatively small control gain matrices $\mathbf{K}_p$ and $\mathbf{K}_d$ for a large region of attraction. The

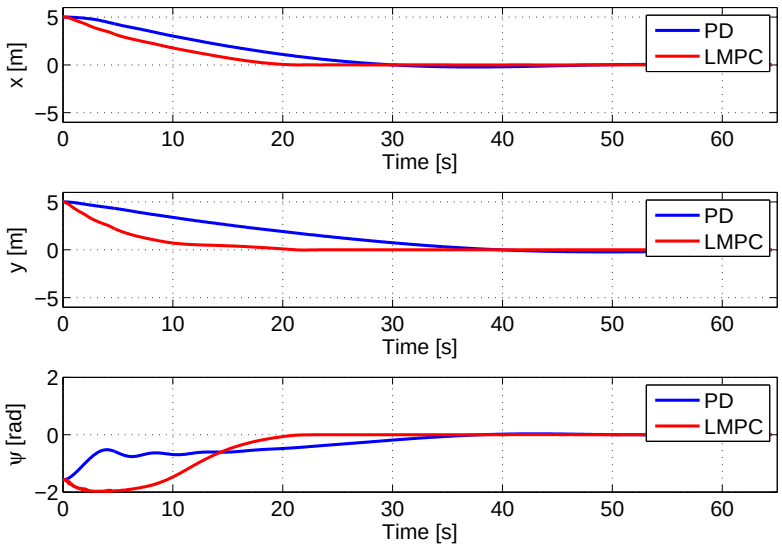

Figure 4.2: The state trajectories.

simulation results demonstrate the superior DP control performance; the resulting improved performance is mainly due to the online optimization. The thrust forces generated by each thruster are plotted in Figure 4.3. We shall notice that in the beginning of the LMPC control, the thrusters are operated with full speed in order to get fastest possible convergence rate.

The receding horizon mechanism introduces feedback into the closed-loop system. Therefore, an additional merit of LMPC-based DP control system is the inherent robustness to model uncertainties and external disturbances [105], which is attractive especially for marine control systems. The robustness of the LMPC-based DP control is investigated through simulations. In the following simulations, we assume that the system model is subject to 20% model error, and there is a disturbance $\mathbf{w} = [10(N), 10(N), 0(Nm)]^{\mathrm{T}}$ exerting on the vehicle all the time, caused by an irrotational ocean current.

From the simulation results illustrated in Figure 4.4 - Figure 4.6, we see that the LMPC-based DP control still gets converged to the desired position. In contrast, it can be proved that the PD control only ensures uniform ultimate boundedness (UUB) of the position error [49]. This has demonstrated that the robustness of DP control can be improved by the introduced online optimization.

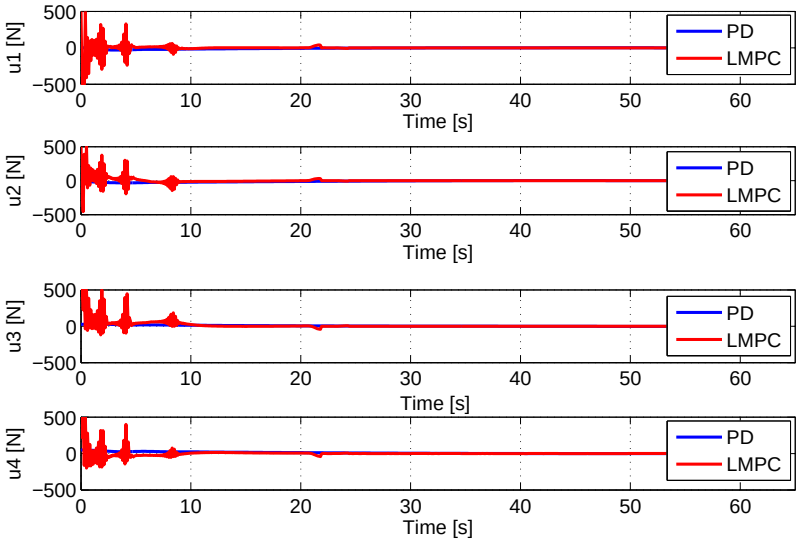

Figure 4.3: The control input signals.

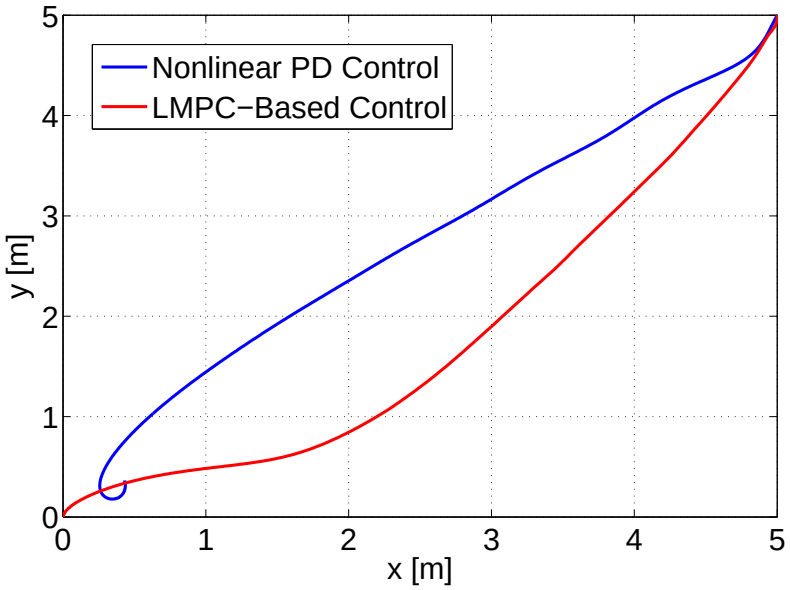

Figure 4.4: The AUV trajectory in local level plane (with disturbance).

4.3 LMPC Design for Trajectory Tracking Control

In this section, we extend the main results obtained in Section 4.2 to solve the trajectory tracking control problem of an AUV.

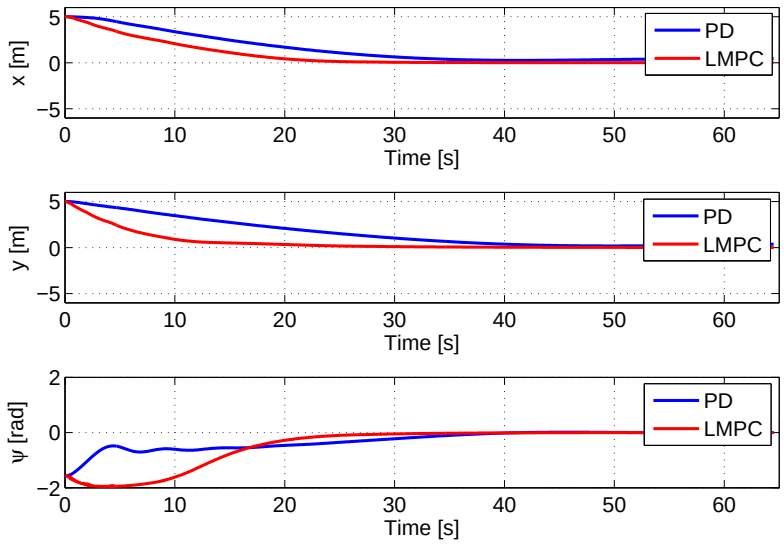

Figure 4.5: The state trajectories (with disturbance).

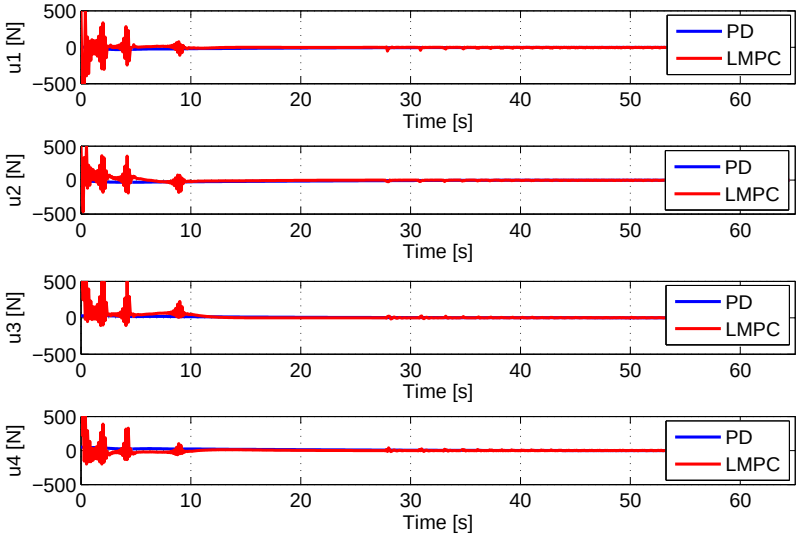

Figure 4.6: The control input signals (with disturbance).

4.3.1 Problem Formulation

Consider a desired trajectory $p(t) = [x_d(t), y_d(t)]^{\mathrm{T}}$ which defines the position of the AUV in the local level plane. Here, we assume that the following conditions hold:

Assumption 3. *The desired trajectory $p(t)$ and its derivatives are smooth and bounded, satisfying that:* $0 \leq \underline{p} \leq \|p(t)\|_\infty \leq \bar{p} < \infty$, $0 < \underline{p_1} \leq \|\dot{p}(t)\|_\infty \leq \bar{p}_1 < \infty$, $0 \leq \underline{p_2} \leq \|\ddot{p}(t)\|_\infty \leq \bar{p}_2 < \infty$ *and* $0 \leq \underline{p_3} \leq \|\dddot{p}(t)\|_\infty \leq \bar{p}_3 < \infty$.

We augment $p(t)$ to a reference system so that each state of the AUV system (4.1) has a feasible reference. Let $\mathbf{x}_d(t) = [x_d(t), y_d(t), \psi_d(t), u_d(t), v_d(t), r_d(t)]^{\mathrm{T}}$ with

$$\begin{aligned} \psi_d(t) &= \operatorname{atan2}(\dot{y}_d(t), \dot{x}_d(t)) \\ u_d(t) &= \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \\ v_d(t) &= 0 \\ r_d(t) &= (\dot{x}_d(t)\ddot{y}_d(t) - \dot{y}_d(t)\ddot{x}_d(t))/(\dot{x}_d^2(t) + \dot{y}_d^2(t)) \end{aligned} \tag{4.26}$$

where atan2 is the four-quadrant inverse tangent operator. Then it can be verified that $\mathbf{x}_d(t)$ satisfies the kinematic equations (2.33). Similarly, we assume $\mathbf{v}_d = [u_d, v_d, r_d]^{\mathrm{T}}$ obeys the dynamic equations (2.31), and the reference control forces $\boldsymbol{\tau}_d = [F_{ud}, F_{vd}, F_{rd}]^{\mathrm{T}}$ can be obtained by

$$\boldsymbol{\tau}_d = \mathbf{M}\dot{\mathbf{v}}_d + \mathbf{C}(\mathbf{v}_d)\mathbf{v}_d + \mathbf{D}(\mathbf{v}_d)\mathbf{v}_d + \mathbf{g}(\boldsymbol{\eta}_d) \tag{4.27}$$

where $\boldsymbol{\eta}_d = [x_d, y_d, \psi_d]^{\mathrm{T}}$ and $\dot{\mathbf{v}}_d$ can be calculated by taking time derivative of (4.26). Furthermore, since the model property P-5 is satisfied we take advantage of the Moore-Penrose pseudoinverse implementation to solve the TA subproblem and we can get the reference control for each thruster:

$$\mathbf{u}_d = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\boldsymbol{\tau}_d = \mathbf{B}^{+}\boldsymbol{\tau}_d \tag{4.28}$$

Proposition 2. *Provided that* **Assumption 3** *holds and the reference signals are chosen as (4.26) then the desired pose $\boldsymbol{\eta}_d$ and its derivatives $\dot{\boldsymbol{\eta}}_d$ and $\ddot{\boldsymbol{\eta}}_d$ are upper bounded, i.e., $\|\boldsymbol{\eta}_d(t)\|_\infty \leq \bar{\eta}_d$, $\|\dot{\boldsymbol{\eta}}_d(t)\|_\infty \leq \bar{\eta}_{d1}$ and $\|\ddot{\boldsymbol{\eta}}_d(t)\|_\infty \leq \bar{\eta}_{d2}$ for some positive numbers $\bar{\eta}_d$, $\bar{\eta}_{d1}$ and $\bar{\eta}_{d2}$.*

Proof. With **Assumption 3**, we notice that the boundedness of $\boldsymbol{\eta}_d$, $\dot{\boldsymbol{\eta}}_d$ and $\ddot{\boldsymbol{\eta}}_d$ depend on the boundedness of ψ_d, $\dot{\psi}_d$ and $\ddot{\psi}_d$, respectively.

By definition, we have $|\psi_d| \leq \pi$. Since $\dot{\psi}_d = r_d$, we have $|\dot{\psi}_d| \leq \bar{p}_1\bar{p}_2/\underline{p_1^2}$. We explicit $\ddot{\psi}_d$ as follows

$$\ddot{\psi}_d = \frac{\dot{x}_d\dddot{y}_d - \dot{y}_d\dddot{x}_d}{\dot{x}_d^2 + \dot{y}_d^2} - \frac{2(\dot{x}_d\ddot{x}_d + \dot{y}_d\ddot{y}_d)(\dot{x}_d\ddot{y}_d - \dot{y}_d\ddot{x}_d)}{(\dot{x}_d^2 + \dot{y}_d^2)^2} \tag{4.29}$$

Obviously, we have $|\ddot{\psi}_d| \leq \bar{p}_1\bar{p}_3/\underline{p_1^2} + 2\bar{p}_1^2\bar{p}_2^2/\underline{p_1^2}$. Therefore, the upper bounds can be calculated by

$$\bar{\eta}_d = \max\{\bar{p}, \pi\} \tag{4.30a}$$

$$\bar{\eta}_{d1} = \max\{\bar{p}_1,\ \bar{p}_1\bar{p}_2/\underline{p_1^2}\} \tag{4.30b}$$

$$\bar{\eta}_{d2} = \max\{\bar{p}_2,\ \bar{p}_1\bar{p}_3/\underline{p_1^2} + 2\bar{p}_1^2\bar{p}_2^2/\underline{p_1^2}\} \tag{4.30c}$$

□

Since $\dot{\boldsymbol{\eta}}_d = \mathbf{R}(\psi)\mathbf{v}_d$, together with (4.27), (4.28) and the model property P-6, it can be easily shown that the reference state $\mathbf{x}_d$ and control $\mathbf{u}_d$ are also finite and upper bounded.

Then the intuitive MPC formulation for the AUV trajectory tracking control can be safely established

$$\begin{aligned} \min_{\hat{\mathbf{u}}\in S(\delta)} \quad & J = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + \|\tilde{\mathbf{u}}(s)\|_R^2)ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \\ \text{s.t.} \quad & \dot{\hat{\mathbf{x}}}(s) = \mathbf{f}(\hat{\mathbf{x}}(s), \hat{\mathbf{u}}(s)) \\ & \hat{\mathbf{x}}(0) = \mathbf{x}(t_0) \\ & |\hat{\mathbf{u}}(s)| \leq \mathbf{u}_{\max} \end{aligned} \tag{4.31}$$

where $\hat{\mathbf{x}}(s)$ is the predicted state trajectory of the vehicle with respect to the predictive control $\hat{\mathbf{u}}(s)$, evolving from $\mathbf{x}(t_0)$ using the system model; $\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}_d$ is the error state and $\tilde{\mathbf{u}} = \hat{\mathbf{u}} - \mathbf{u}_d$ is the control error; $S(\delta)$ denotes the family of piecewise constant functions characterized by the sampling period δ and $T = N\delta$ is the prediction horizon; The weighting matrices Q, R and P are positive definite.

As usual, the closed-loop stability cannot be automatically guaranteed by the optimality of the solution due to the finite prediction horizon. To ensure this important closed-loop property, complex offline design procedure should take place. For nonlinear systems such as the AUV, with standard MPC design technique, local linearization needs to be performed in order to select appropriate weighting matrices and to construct an auxiliary local feedback control law. In this way, the region of attraction will be defined implicitly [111].

As mentioned in Section 1.1.3, for the AUV tracking of a curved reference trajectory the local linearization appears inappropriate to apply. To circumvent the local linearizarion while ensuring the closed-loop stability of the MPC tracking control, we exploit an auxiliary Lyapunov-based nonlinear tracking control law and formulate the

LMPC problem by adding a contraction constraint to the original MPC formulation:

$$\min_{\hat{\mathbf{u}}\in S(\delta)} J = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + \|\tilde{\mathbf{u}}(s)\|_R^2)ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{4.32a}$$

$$\text{s.t.} \quad \dot{\hat{\mathbf{x}}}(s) = \mathbf{f}(\hat{\mathbf{x}}(s), \hat{\mathbf{u}}(s)) \tag{4.32b}$$

$$\hat{\mathbf{x}}(0) = \mathbf{x}(t_0) \tag{4.32c}$$

$$|\hat{\mathbf{u}}(s)| \leq \mathbf{u}_{\max} \tag{4.32d}$$

$$\tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\hat{\mathbf{x}}(0), \hat{\mathbf{u}}(0)) \leq \tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\hat{\mathbf{x}}(0), h(\hat{\mathbf{x}}(0))) \tag{4.32e}$$

where $h(\cdot)$ is the auxiliary Lyapunov-based nonlinear tracking control law and $V(\cdot)$ is the corresponding Lyapunov function. The presence of contraction constraint (4.32e) allows us to show that the LMPC controller inherits the stability properties of the state feedback control $h(\mathbf{x})$ and a guaranteed ROA can be explicitly characterized. Furthermore, thanks to the online optimization procedure, the LMPC controller will automatically perform the best achievable tracking control and respect the physical limitation of the system.

The LMPC-based trajectory tracking control will be implemented in the standard receding horizon fashion and the control algorithm is summarized in **Algorithm 2**.

Algorithm 2 : LMPC Algorithm

1: Input the objective function J in (4.32a).
2: Measure the current state $\mathbf{x}(t)$.
3: Solve the LMPC problem (4.32) with $\mathbf{x}(t_0) = \mathbf{x}(t)$; let $\boldsymbol{\kappa}(s)$ denote the optimal solution.
4: Implement $\boldsymbol{\kappa}(s)$ for only one sampling period: $\mathbf{u}(t) = \boldsymbol{\kappa}(s)$ for $s \in [0, \delta]$;
5: At next sampling time instant, set $t = t + \delta$, then repeat from step 2.

Remark 8. *As will be seen shortly in the next section, neither the recursive feasibility nor the closed-loop stability relies on the exact solution of the optimization. In* **Algorithm 2** *suboptimal solutions are practically acceptable, which is highly desirable for any nonlinear MPC algorithm by considering the following facts: First, since the system model (4.1) is nonlinear, using iterative methods, the best guaranteed solution to (4.32) is a local optimum. More importantly, for the implementation on embedded systems with limited computational resource, the iteration number may be restricted for real-time control. In other words, the compatibility with suboptimal solutions introduces the flexibility between numerical efficiency and control performance. We can*

conveniently make the trade-off by specifying the maximum iteration number without destabilizing the tracking control.

4.3.2 Main Results

In this section, we first construct the auxiliary tracking controller using Lyapunov-based backstepping technique and then analyze the recursive feasibility and closed-loop stability of the LMPC under **Algorithm 2**.

To construct the contraction constraint in (4.32e), we need to find a state feedback controller together with the corresponding Lyapunov function. For the AUV trajectory tracking, the Lyapunov-based nonlinear controller can be designed via the backstepping technique.

Define the following change of variables:

$$\dot{\boldsymbol{\eta}}_r = \dot{\boldsymbol{\eta}}_d - \tilde{\boldsymbol{\eta}} \tag{4.33a}$$

$$\mathbf{v}_r = \mathbf{R}^{\mathrm{T}}(\psi)\dot{\boldsymbol{\eta}}_r \tag{4.33b}$$

$$\mathbf{s} = \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_r \tag{4.33c}$$

where $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta} - \boldsymbol{\eta}_d$ is the position tracking error. Considering the kinematic equations (2.33), we have

$$\dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_d = \mathbf{R}(\psi)(\mathbf{v} - \mathbf{v}_d) \tag{4.34}$$

We view $\mathbf{v}$ as a virtual control that stabilizes the trajectory tracking control:

$$\mathbf{R}(\psi)\mathbf{v} = \mathbf{s} + \boldsymbol{\alpha}_1 \tag{4.35}$$

Choosing $\boldsymbol{\alpha}_1 = \dot{\boldsymbol{\eta}}_r$ and substituting it into (4.34) - (4.35) yields

$$\dot{\tilde{\boldsymbol{\eta}}} = \mathbf{s} + \boldsymbol{\alpha}_1 - \mathbf{R}(\psi)\mathbf{v}_d = -\tilde{\boldsymbol{\eta}} + \mathbf{s} \tag{4.36}$$

Consider the following function:

$$V_1 = \frac{1}{2}\tilde{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} \tag{4.37}$$

where $\mathbf{K}_p = \mathbf{K}_p^{\mathrm{T}} > 0$ is a specified control gain matrix. Then the time derivative of V_1 becomes:

$$\dot{V}_1 = \tilde{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{K}_p\dot{\tilde{\boldsymbol{\eta}}} = -\tilde{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} + \mathbf{s}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} \tag{4.38}$$

Further construct the Lyapunov function candidate:

$$V_2 = \frac{1}{2}\mathbf{s}^{\mathrm{T}}\mathbf{M}^*(\psi)\mathbf{s} + V_1 \tag{4.39}$$

where $\mathbf{M}^*(\psi) = \mathbf{R}(\psi)\mathbf{M}\mathbf{R}^{\mathrm{T}}(\psi)$.

Taking time derivative of V_2 results in:

$$\dot{V}_2 \;=\; \mathbf{s}^{\mathrm{T}}\mathbf{M}^*(\psi)\dot{\mathbf{s}} + \tfrac{1}{2}\mathbf{s}^{\mathrm{T}}\dot{\mathbf{M}}^*(\psi)\mathbf{s} + \dot{V}_1 \tag{4.40}$$

Substituting the dynamic equations (2.31), we have

$$\begin{aligned}\dot{V}_2 \;=\;& -\mathbf{s}^{\mathrm{T}}[\mathbf{C}^*(\mathbf{v},\psi) + \mathbf{D}^*(\mathbf{v},\psi)]\mathbf{s} \\ & +\mathbf{s}^{\mathrm{T}}\mathbf{R}(\psi)[\boldsymbol{\tau} - \mathbf{M}\dot{\mathbf{v}}_r - \mathbf{C}(\mathbf{v})\mathbf{v}_r - \mathbf{D}(\mathbf{v})\mathbf{v}_r - \mathbf{g}(\boldsymbol{\eta})] \\ & +\tfrac{1}{2}\mathbf{s}^{\mathrm{T}}\dot{\mathbf{M}}^*(\psi)\mathbf{s} - \tilde{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} + \mathbf{s}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}}\end{aligned} \tag{4.41}$$

where $\mathbf{C}^*(\mathbf{v},\psi) = \mathbf{R}(\psi)[\mathbf{C}(\mathbf{v}) - \mathbf{M}\mathbf{R}^{\mathrm{T}}(\psi)\dot{\mathbf{R}}(\psi)]\mathbf{R}^{\mathrm{T}}(\psi)$ and $\mathbf{D}(\mathbf{v},\psi)^* = \mathbf{R}(\psi)\mathbf{D}(\mathbf{v})\mathbf{R}^{\mathrm{T}}(\psi)$.

From model property P-2, it can be verified that

$$\mathbf{s}^{\mathrm{T}}(\dot{\mathbf{M}}^*(\psi) - 2\mathbf{C}^*(\mathbf{v},\psi))\mathbf{s} = 0, \quad \forall\ \mathbf{v},\psi,\mathbf{s} \tag{4.42}$$

Therefore, if we choose the following control law:

$$\boldsymbol{\tau}(\mathbf{x}) = \mathbf{M}\dot{\mathbf{v}}_r + \mathbf{C}\mathbf{v}_r + \mathbf{D}\mathbf{v}_r + \mathbf{g} - \mathbf{R}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} - \mathbf{R}^{\mathrm{T}}\mathbf{K}_d\mathbf{s} \tag{4.43}$$

where $\mathbf{K}_d > 0$ is another user specified control gain matrix, Eqn. (4.41) becomes

$$\dot{V}_2 = -\mathbf{s}^{\mathrm{T}}[\mathbf{D}^*(\mathbf{v},\psi) + \mathbf{K}_d]\mathbf{s} - \tilde{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} \tag{4.44}$$

From model property P-4, we have $\dot{V}_2 \leq 0$. Then by standard Lyapunov arguments, the closed-loop system under (4.43) is globally asymptotically stable with respect to the equilibrium $[\tilde{\boldsymbol{\eta}}, \mathbf{s}] = [\mathbf{0}, \mathbf{0}]$.

Therefore, the detailed expression of the contraction constraint (4.32e) correspond-

ing to (4.43) is

$$\begin{aligned} -\hat{\mathbf{s}}(0)^{\mathrm{T}}\mathbf{D}^{*}(\hat{\mathbf{v}}(0),\hat{\psi}(0))\hat{\mathbf{s}}(0)+\hat{\mathbf{s}}(0)^{\mathrm{T}}\mathbf{R}(\hat{\psi}(0))[\mathbf{B}\hat{\mathbf{u}}(0) \\ -\mathbf{M}\dot{\hat{\mathbf{v}}}_r(0)-\mathbf{C}(\hat{\mathbf{v}}(0))\hat{\mathbf{v}}_r(0)-\mathbf{D}(\hat{\mathbf{v}}(0))\hat{\mathbf{v}}_r(0) \\ -\mathbf{g}(\hat{\boldsymbol{\eta}}(0))]-\tilde{\boldsymbol{\eta}}(0)^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}}(0)+\hat{\mathbf{s}}(0)^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}}(0) \\ \leq -\hat{\mathbf{s}}(0)^{\mathrm{T}}[\mathbf{D}^{*}(\hat{\mathbf{v}}(0),\hat{\psi}(0))+\mathbf{K}_d]\hat{\mathbf{s}}(0)-\tilde{\boldsymbol{\eta}}(0)^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}}(0) \end{aligned} \tag{4.45}$$

Note that, here, we construct the contraction constraint with the help of the auxiliary nonlinear backstepping controller. However, in principle, any other Lyapunov-based tracking controller such as sliding model control and dynamic surface control, can be employed.

Now we analyze the recursive feasibility of the problem (4.32) and the closed-loop stability under **Algorithm 2**.

Lemma 1. *For the AUV system (4.1), suppose* **Assumption 3** *is satisfied. If the backstepping control law (4.43) is applied to the AUV, then the Coriolis and centripetal matrix* $\mathbf{C}(\mathbf{v})$ *and damping matrix* $\mathbf{D}(\mathbf{v})$ *are such that the following relations hold:*

$$\|\mathbf{C}(\mathbf{v})\|_{\infty}\leq\bar{c}=2\sqrt{2}\bar{m}\bar{\eta}_{d1}+4\sqrt{2}\bar{m}\|\gamma(0)\|_2 \tag{4.46}$$

$$\|\mathbf{D}(\mathbf{v})\|_{\infty}\leq\bar{d}=\bar{d}_1+\bar{d}_2(\sqrt{2}\bar{\eta}_{d1}+2\sqrt{2}\|\gamma(0)\|_2) \tag{4.47}$$

where $\gamma(t)=col(\tilde{\boldsymbol{\eta}}(t),\mathbf{s}(t))$, $\bar{d}_1=\max\{|X_u|,|Y_v|,|N_r|\}$ *and* $\bar{d}_2=\max\{D_u,D_v,D_r\}$.

Proof. Define $\gamma'=col(\tilde{\boldsymbol{\eta}},\mathbf{R}^{\mathrm{T}}(\psi)\mathbf{s})$ and we can reformulate the Lyapunov function (4.39) as $V_2=\frac{1}{2}\gamma'^{\mathrm{T}}\Pi\gamma'$ with $\Pi=\mathrm{diag}(\mathbf{K}_p,\mathbf{M})$. Since $\dot{V}_2\leq 0$, we have $\|\gamma'(t)\|_2\leq\|\gamma'(0)\|_2$. Furthermore, from the model property P-3 we have $\|\gamma'(t)\|_2=\|\gamma(t)\|_2$ followed by

$$\|\gamma(t)\|_2\leq\|\gamma(0)\|_2 \tag{4.48}$$

By definition, we have $\|\tilde{\boldsymbol{\eta}}\|_{\infty}\leq\|\gamma\|_{\infty}$ and $\|\mathbf{s}\|_{\infty}\leq\|\gamma\|_{\infty}$. Then the following holds

$$\|\dot{\tilde{\boldsymbol{\eta}}}\|_{\infty}=\|\mathbf{s}-\tilde{\boldsymbol{\eta}}\|_{\infty}\leq\|\mathbf{s}\|_{\infty}+\|\tilde{\boldsymbol{\eta}}\|_{\infty}\leq 2\|\gamma\|_{\infty} \tag{4.49}$$

Considering that $\|\gamma\|_{\infty}\leq\|\gamma\|_2$, we have

$$\|\dot{\tilde{\boldsymbol{\eta}}}(t)\|_{\infty}\leq 2\|\gamma(0)\|_2 \tag{4.50}$$

Since $\|\dot{\boldsymbol{\eta}}\|_{\infty}=\|\dot{\boldsymbol{\eta}}_d+\dot{\tilde{\boldsymbol{\eta}}}\|_{\infty}\leq\|\dot{\boldsymbol{\eta}}_d\|_{\infty}+\|\dot{\tilde{\boldsymbol{\eta}}}\|_{\infty}$, together with (4.50) and **Proposition**

2, it follows that

$$\|\dot{\boldsymbol{\eta}}(t)\|_\infty \leq \bar{\eta}_{d1} + 2\|\gamma(0)\|_2 \tag{4.51}$$

By the kinematic equations (2.33) and the model property P-3, we have $\|\mathbf{v}\|_\infty = \|\mathbf{R}^{\mathrm{T}}(\psi)\dot{\boldsymbol{\eta}}\|_\infty \leq \sqrt{2}\|\dot{\boldsymbol{\eta}}\|_\infty$ due to the fact that $\|\mathbf{R}^{\mathrm{T}}(\psi)\|_\infty = \max\{|\cos\psi|+|\sin\psi|, 1\} \leq \sqrt{2}$. Therefore, the following holds

$$\|\mathbf{v}(t)\|_\infty \leq \sqrt{2}\bar{\eta}_{d1} + 2\sqrt{2}\|\gamma(0)\|_2 \tag{4.52}$$

Having the model property P-1 and taking infinity norm on (2.30) yields

$$\|\mathbf{C}(\mathbf{v})\|_\infty \leq |M_{\dot{u}}u| + |M_{\dot{v}}v| \leq \bar{m}(|u| + |v|) \tag{4.53}$$

Further considering (4.52), we have

$$\|\mathbf{C}(\mathbf{v})\|_\infty \leq 2\sqrt{2}\bar{m}\bar{\eta}_{d1} + 4\sqrt{2}\bar{m}\|\gamma(0)\|_2 \tag{4.54}$$

From the detailed damping matrix expression (2.29), by definition, we have $\|\mathbf{D}(\mathbf{v})\|_\infty = \max\{|X_u + D_u|u||, |Y_v + D_v|v||, |N_r + D_r|r||\}$. Further considering (4.52), it yields

$$\|\mathbf{D}(\mathbf{v})\|_\infty \leq \bar{d}_1 + \bar{d}_2(\sqrt{2}\bar{\eta}_{d1} + 2\sqrt{2}\|\gamma(0)\|_2) \tag{4.55}$$

□

Theorem 4. *Suppose the control gain matrices are chosen as* $\mathbf{K}_p = \mathrm{diag}\{k_{pi}\}$ *and* $\mathbf{K}_d = \mathrm{diag}\{k_{di}\}$*, positive definite. Let* $\bar{k}_p = \max\{k_{pi}\}$ *denote the largest element in* $\mathbf{K}_p$ *and* $\bar{k}_d = \max\{k_{di}\}$ *denote the largest element in* $\mathbf{K}_d$*. Suppose* **Assumption 1** *and* **Assumption 3** *can hold. For* $h(\mathbf{x}) = \mathbf{B}^{+}\boldsymbol{\tau}(\mathbf{x})$*, if the following relation can be satisfied*

$$\bar{m}\bar{v}_{r1} + (\bar{c} + \bar{d})\bar{v}_r + \bar{g} + \sqrt{2}(\bar{k}_p + \bar{k}_d)\|\gamma(0)\|_2 \leq \tau_{max} \tag{4.56}$$

with $\bar{v}_{r1} = \sqrt{2}\bar{\eta}_{d2} + 2\bar{\eta}_{d1}^2 + (2\sqrt{2} + 6\bar{\eta}_{d1})\|\gamma(0)\|_2 + 4\|\gamma(0)\|_2^2$ *and* $\bar{v}_r = \sqrt{2}(\bar{\eta}_{d1} + \|\gamma(0)\|_2)$*, and* τ_{max} *follows (4.12), then the LMPC (4.32) admits recursive feasibility, i.e.,* $|h(\hat{\mathbf{x}}(t))| \leq \mathbf{u}_{max}$ *for all* $t \geq 0$ *where* $\mathbf{u}_{max} = u_{max}\mathbf{1}$*.*

Proof. We notice that given the current system state $\mathbf{x}(t)$, $h(\hat{\mathbf{x}})$ is always feasible for the LMPC problem (4.32) if $|h(\hat{\mathbf{x}})| \leq \mathbf{u}_{\max}$ can be satisfied.

Since we have **Proposition 2** and the fact that $\|\hat{\dot{\boldsymbol{\eta}}}\|_\infty \leq \|\hat{\gamma}\|_\infty \leq \|\gamma(0)\|_2$, it can be verified that $\|\hat{\mathbf{v}}_r\|_\infty \leq \bar{v}_r$ with $\bar{v}_r = \sqrt{2}(\bar{\eta}_{d1} + \|\gamma(0)\|_2)$ by taking infinity norm on

both sides of (4.33).

Taking time derivative on (4.33a) and on (4.33b) we have

$$\begin{aligned} \dot{\mathbf{v}}_r &= \mathbf{R}^{\mathrm{T}}(\psi)\ddot{\boldsymbol{\eta}}_r - \dot{\psi}\boldsymbol{\Omega}(\psi)\dot{\boldsymbol{\eta}}_r \\ &= \mathbf{R}^{\mathrm{T}}(\psi)(\ddot{\boldsymbol{\eta}}_d - \dot{\tilde{\boldsymbol{\eta}}}) - r\boldsymbol{\Omega}(\psi)(\dot{\boldsymbol{\eta}}_d - \tilde{\boldsymbol{\eta}}) \end{aligned} \tag{4.57}$$

where

$$\boldsymbol{\Omega}(\psi) = \begin{bmatrix} \sin\psi & -\cos\psi & 0 \\ \cos\psi & \sin\psi & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.58}$$

which admits $\|\boldsymbol{\Omega}(\psi)\|_\infty \leq \sqrt{2}$. Then taking infinity norm on (4.57) and combining (4.50) and (4.52), we have $\|\dot{\hat{\mathbf{v}}}_r\|_\infty \leq \bar{v}_{r1}$ with $\bar{v}_{r1} = \sqrt{2}\bar{\eta}_{d2} + 2\bar{\eta}_{d1}^2 + (2\sqrt{2} + 6\bar{\eta}_{d1})\|\gamma(0)\|_2 + 4\|\gamma(0)\|_2^2$.

Then taking infinity norm on both sides of (4.43), together with **Lemma 1** and the model property P-6, we have

$$\|\boldsymbol{\tau}(\hat{\mathbf{x}})\|_\infty \leq \bar{m}\bar{v}_{r1} + (\bar{c} + \bar{d})\bar{v}_r + \bar{g} + \sqrt{2}(\bar{k}_p + \bar{k}_d)\|\gamma(0)\|_2 \tag{4.59}$$

If (4.56) can be satisfied, then $\|\boldsymbol{\tau}(\hat{\mathbf{x}})\|_\infty \leq \tau_{\max}$ can hold. With (4.12), we can guarantee that $\|h(\hat{\mathbf{x}}(s))\|_\infty \leq u_{\max}$ will be always satisfied. This completes the proof. □

Note that the nonlinear backstepping controller acts as an initial guess for the optimization problem (4.32) but is never applied to the AUV system. As a result, the feasibility is guaranteed by the auxiliary controller and the control performance will be optimized by the LMPC controller.

Theorem 5. *Suppose* **Assumption 1** *and* **Assumption 3** *can hold, then the closed-loop system under* **Algorithm 2** *is asymptotically stable with respect to the equilibrium* $[\tilde{\boldsymbol{\eta}}, \mathbf{s}] = [\mathbf{0}, \mathbf{0}]$*, i.e., the AUV will converge to the desired trajectory* $p(t)$ *with the LMPC-based trajectory tracking control.*

Proof. According to the Lyapunov function $V_2(\mathbf{x})$ in (4.39), continuously differentiable and radically unbounded, by converse Lyapunov theorems [75], there exist functions $\beta_i(\cdot)$, $i = 1, 2, 3$ which belong to class $\mathcal{K}_\infty$ such that the following inequalities hold:

$$\beta_1(\|\mathbf{x}\|) \leq V_2(\mathbf{x}) \leq \beta_2(\|\mathbf{x}\|) \tag{4.60a}$$

$$\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, h(\mathbf{x})) \leq -\beta_3(\|\mathbf{x}\|) \tag{4.60b}$$

Considering the contraction constraint (4.32e) and that the optimal solution $\boldsymbol{\kappa}(s)$ will be implemented for one sampling period, we have

$$\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})) \leq \frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, h(\mathbf{x})) \leq -\beta_3(\|\mathbf{x}\|) \tag{4.61}$$

By standard Lyapunov arguments we claim that the closed-loop system under **Algorithm 2** is asymptotically stable with a guaranteed region of attraction

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid (4.56)\} \tag{4.62}$$

Furthermore, $\mathcal{X}$ can be enlarged by shrinking the magnitude of the control gains $\bar{k}_p$ and $\bar{k}_d$. □

Remark 9. *Although the asymptotical stability relies only on the positive definiteness of the control gain matrices* $\mathbf{K}_p$ *and* $\mathbf{K}_d$*, the tracking control performance with the backstepping controller* $\boldsymbol{\tau}(\mathbf{x})$ *in (4.43) is determined by the magnitude of the control gains. From (4.44), we can see that smaller values of* $\bar{k}_p$ *and* $\bar{k}_d$ *result in slower convergence. However, for the proposed LMPC-based trajectory tracking control, thanks to the optimization procedure, the controller can automatically make full use of the thrust capability to generate the best possible tracking control with respect to the objective function (4.32a) even if we have selected small control gains for a large region of attraction.*

4.3.3 Simulation Results

In this section, we present simulation results of the AUV tracking control which highlight the advantages of the proposed LMPC method. All the simulations are based on the experimentally identified dynamic model of Falcon. The model parameter details can be found in Table 2.1.

Two desired trajectories are used to test the AUV tracking control. The first one (Case I) is a sinusoidal trajectory defined as follows,

$$p(t) = \begin{cases} x_d = 0.5t \\ y_d = \sin(0.5t) \end{cases} \tag{4.63}$$

and the second one (Case II) is an eight-shaped trajectory defined by

$$p(t) = \begin{cases} x_d = -\sin(0.5t) \\ y_d = \sin(0.25t) \end{cases} \tag{4.64}$$

For the LMPC tracking controller, the following parameters are used: The sampling period is $\delta = 0.1$ [sec]; the prediction horizon is $T = 5\delta$; the weighting matrices are chosen as $Q = \text{diag}(10^5, 10^5, 10^3, 10^2, 10^2, 10^2)$, $R = \text{diag}(10^{-4}, 10^{-4}, 10^{-4}, 10^{-4})$ and $P = \text{diag}(10^3, 10^3, 10^2, 10, 10, 10)$; and the limit on each thruster is 500 [N]. The control gains $\mathbf{K}_p = \mathbf{K}_d = \text{diag}(1, 1, 1)$. The initial condition is $\mathbf{x}(0) = [0.5, 0, 0, 0, 0, 0]^{\mathrm{T}}$.

To solve the LMPC problem (4.32) numerically, we need to discretize the problem and then solve the corresponding Karush-Kuhn-Tucker (KKT) conditions by the sequential quadratic programming (SQP) method [9].

The trajectory tracking results for Case I are shown in Figure 4.7 and Figure 4.8. The blue curve is the simulated AUV trajectories using backstepping control (BSC), and the red curve is the AUV trajectories with the proposed LMPC control, while the black curve is the desired sinusoidal trajectory. It can be observed that both BSC and LMPC drive the vehicle to the desired trajectory, which verifies the closed-loop stability. But obviously the LMPC controller generates a much faster convergence than the BSC controller. This is because we have selected small control gain matrices $\mathbf{K}_p$ and $\mathbf{K}_d$ for a large region of attraction. The simulation results demonstrate the enhanced tracking control performance brought by the online optimization.

The required control forces for each thruster are plotted in Figure 4.9. We observe that in the beginning of the tracking, the LMPC controller fully uses the onboard thrust capability in order to generate the fastest possible convergence while respecting the physical limit of thrusters. The magnitude of control commands stays within the permitted range as expected.

The simulation results for Case II are provided through Figure 4.10 to Figure 4.12. Similar observations can be made: The AUV converges faster to the desired trajectory when adopting the proposed LMPC-based tracking control and the calculated control commands are always feasible for the real system.

To test the robustness of the BSC and LMPC controllers, we simulate the AUV tracking control under the following test condition: There exist 30% model parameter error and an ocean current introduced disturbance $\mathbf{w} = [100(N), 100(N), 0(Nm)]^{\mathrm{T}}$. Note that this is a very strict test condition, and it is more of a test than introducing

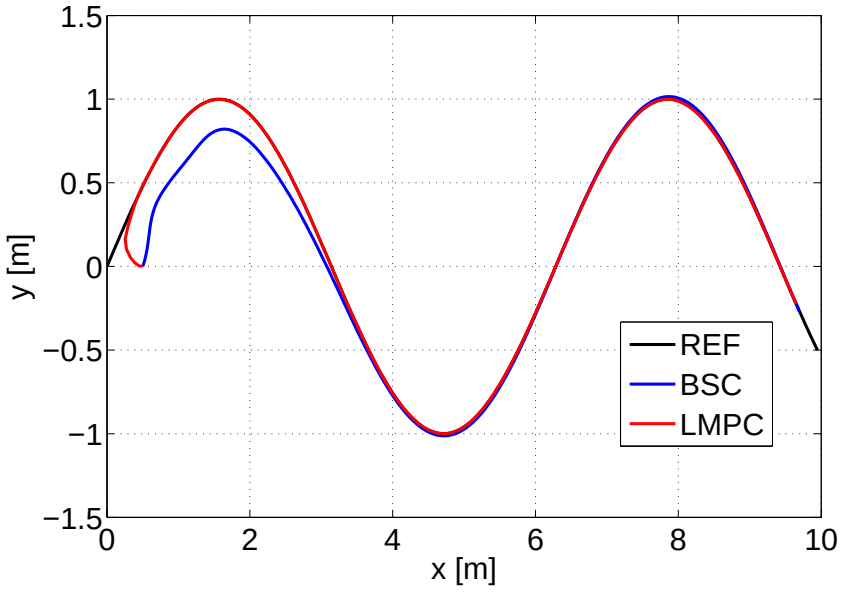

Figure 4.7: The AUV trajectory in local level plane - Case I.

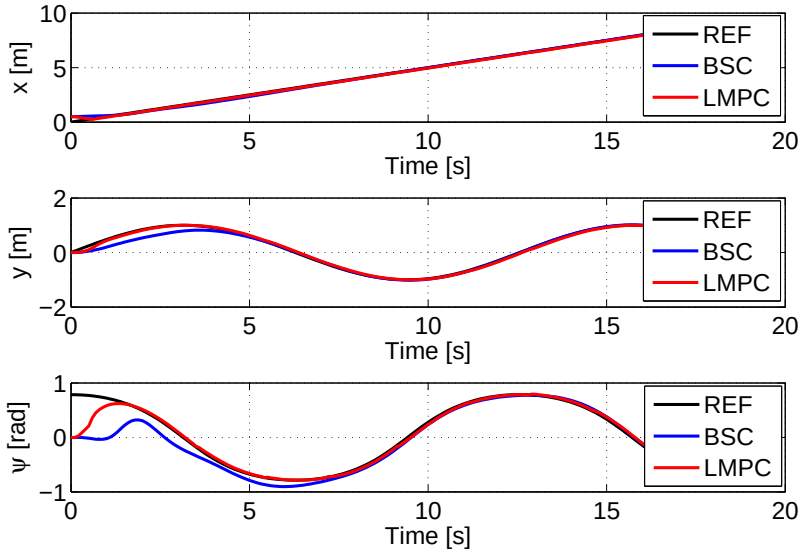

Figure 4.8: The state trajectories - Case I.

some obscure coupling of vehicle DOFs that may or may not exist on a real AUV.

From the simulation results illustrated in Figure 4.13 - Figure 4.16, we find that the LMPC-based tracking control still successfully leads the AUV well converged to the desired trajectory, while the tracking control with BSC suffers from large tracking

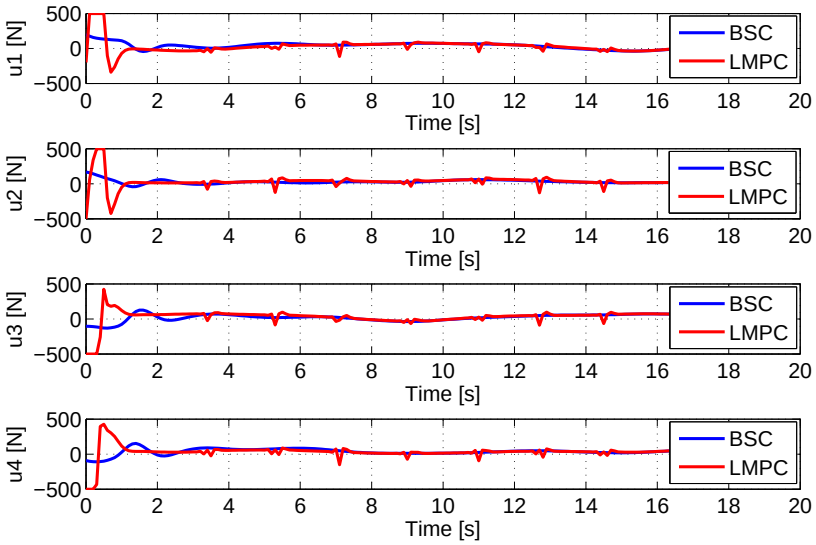

Figure 4.9: The control input signals - Case I.

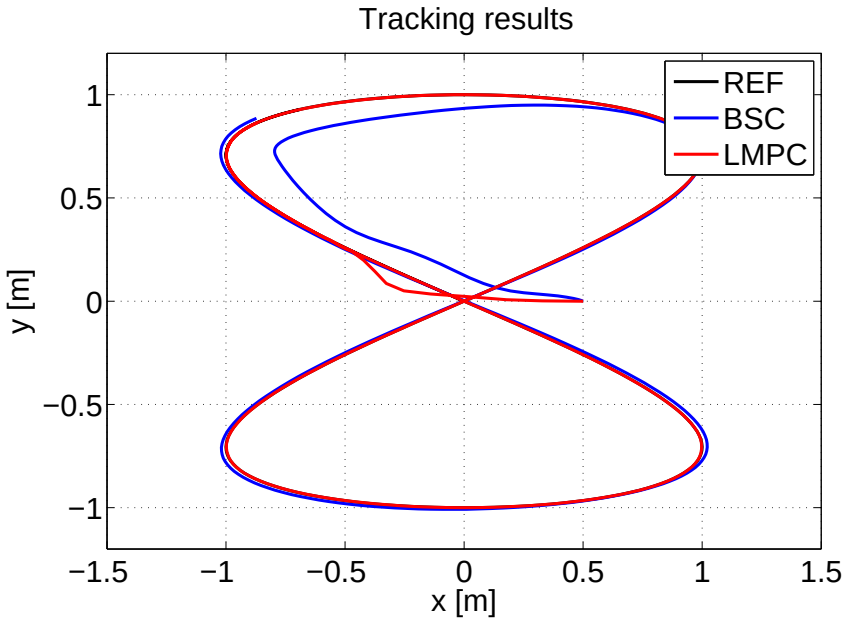

Figure 4.10: The AUV trajectory in local level plane - Case II.

errors. The quantitative comparisons for both cases are summarized in Table 4.1 and 4.2, where tremendous improvement in mean square errors (MSEs) can be witnessed. Roughly speaking, the MSEs are more than 20 times smaller with the LMPC, especially for Case II. This is because LMPC can leverage online optimization to schedule

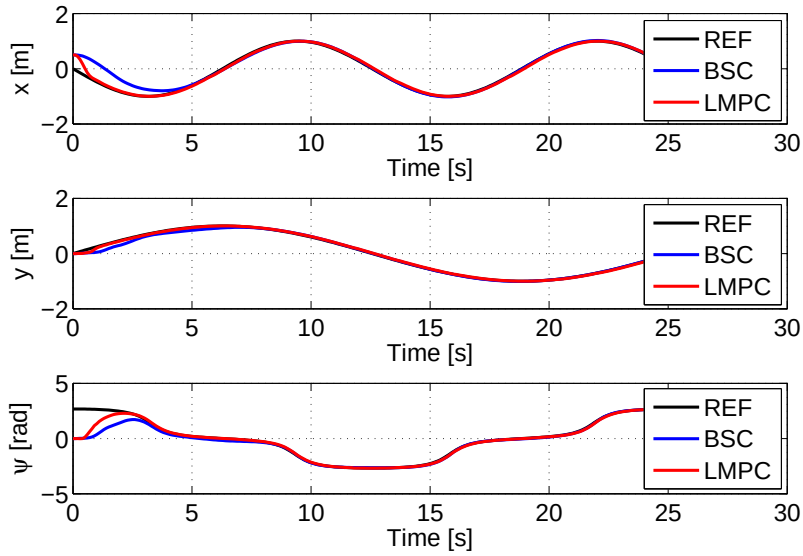

Figure 4.11: The state trajectories - Case II.

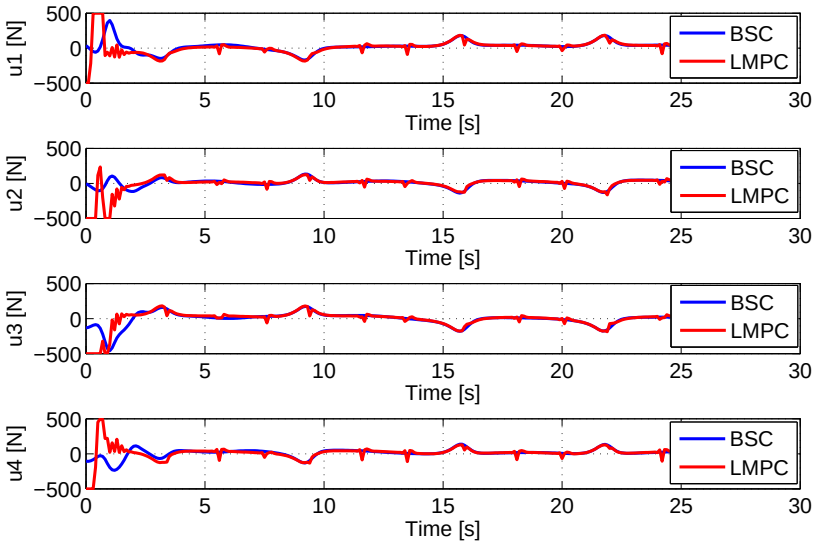

Figure 4.12: The control input signals - Case II.

an appropriate control gain to well compensate the disturbances, whereas BSC is a fixed gain controller and lacks such flexibility. The LMPC, therefore, is robust to model uncertainties and external disturbances, which exhibits great potential for the motion control of general marine vessels.

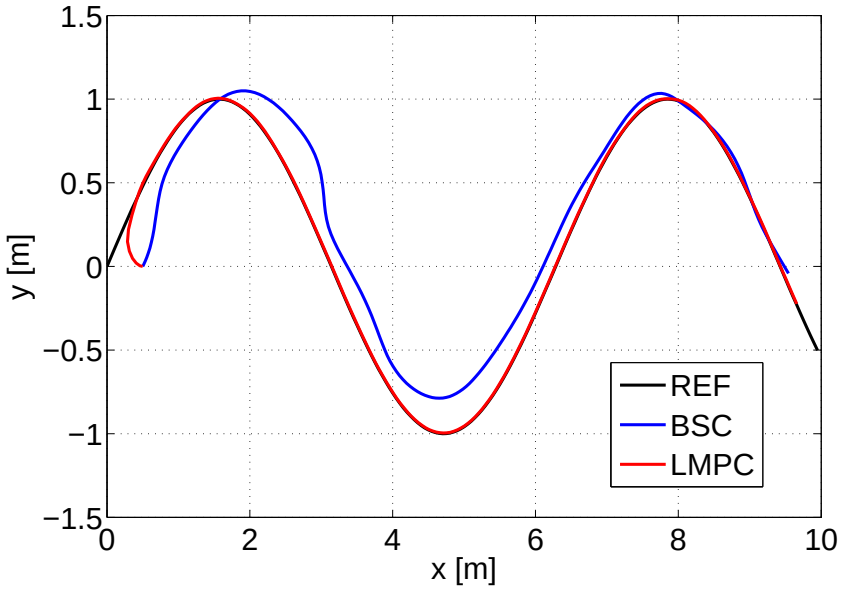

Figure 4.13: The AUV trajectory in local level plane (with disturbance) - Case I.

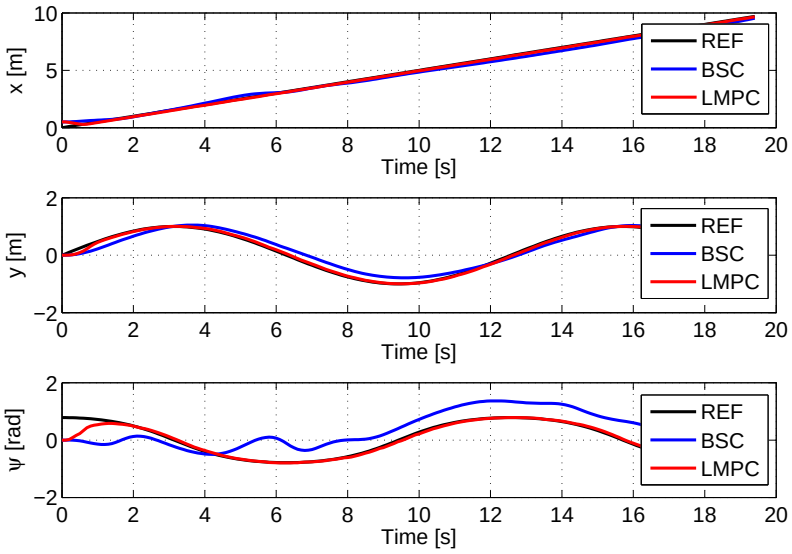

Figure 4.14: The state trajectories (with disturbance) - Case I.

4.4 Conclusion

In this chapter, we have proposed a novel LMPC framework to solve the AUV motion control problems. The LMPC presents a convenient and effective tool to allocate on-

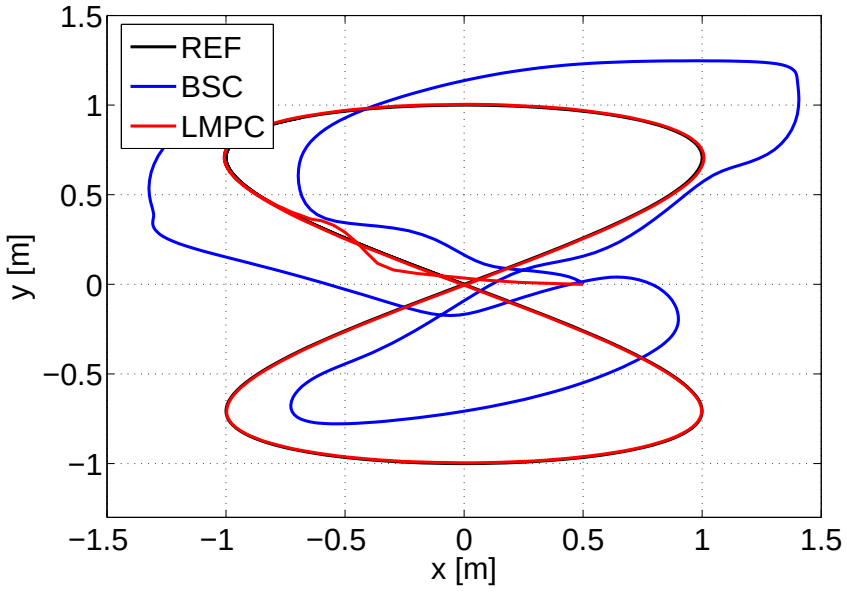

Figure 4.15: The AUV trajectory in local level plane (with disturbance) - Case II.

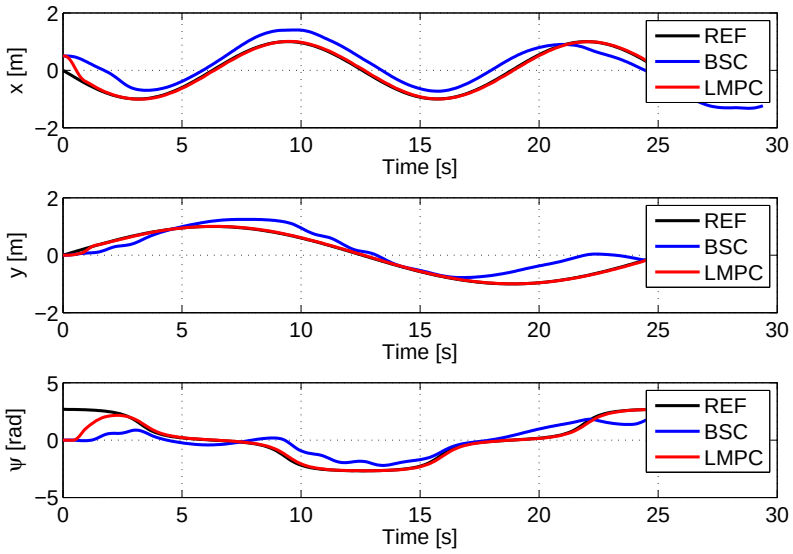

Figure 4.16: The state trajectories (with disturbance) - Case II.

board computing resources to improve the performance of the existing motion control system. Two fundamental AUV motion control problems, namely, the dynamic positioning and the trajectory tracking, were investigated under the LMPC framework and ended up with the controller design.

Table 4.1: Mean square errors for AUV tracking with disturbances - Case I.

MSE	Backstepping	LMPC	Improvement
x [m^2]	0.0418	0.0052	87.5%
y [m^2]	0.0297	0.0015	94.9%
ψ [rad^2]	0.3207	0.0184	94.3%

Table 4.2: Mean square errors for AUV tracking with disturbances - Case II.

MSE	Backstepping	LMPC	Improvement
x [m^2]	0.1483	0.0059	96.0%
y [m^2]	0.0977	0.0005	99.5%
ψ [rad^2]	1.0081	0.2480	75.4%

Using the nonlinear PD control law as the base DP controller, the conditions for recursive feasibility and closed-loop stability of the LMPC control system were derived. A quasi-global stability property was claimed for the LMPC-based DP control: The region of attraction can be specified arbitrarily large. Also, built on a nonlinear backstepping tracking control law, the LMPC-based trajectory tracking controller was designed with guaranteed feasibility and stability. A guaranteed region of attraction was characterized explicitly.

Simulation studies on the dynamic model of Falcon suggested that the dynamic positioning and trajectory tracking performance could be dramatically improved by the online optimization that was incorporated in the LMPC control. Meanwhile, through the robustness test, the superior robustness of the LMPC control system was demonstrated, which indicated that the proposed LMPC method was very promising and could be applied to real AUVs (large model mismatch) under real working condition (large disturbances).

The main results in this chapter have been published in [124] and [126].

[124] **C. Shen**, Y. Shi and B. Buckham, "Lyapunov-based Model Predictive Control for Dynamic Positioning of Autonomous Underwater Vehicles", in *Proceedings of the 2017 IEEE International Conference on Unmanned Systems (ICUS)*, Beijing, China, 2017, pp. 1-6.

[126] **C. Shen**, Y. Shi and B. Buckham, "Trajectory Tracking Control of an Autonomous Underwater Vehicle using Lyapunov-based Model Predictive Control", *IEEE Transactions on Industrial Electronics*, accepted, 2017, DOI: 10.1109/TIE.2017.2779442

Chapter 5

Multi-Objective Model Predictive Control for Path Following Control of an AUV

5.1 Introduction

5.1.1 Research Background and Contributions

In the previous chapters, we have studied the trajectory tracking problem of the AUV. In trajectory tracking problems, the reference trajectory defines the desired footprints with precise time stamps, i.e., it specifies when should the vehicle be where. However, in many circumstances, we may be interested in following a specific geometric pattern in the workspace, and the temporal requirement can be relaxed, i.e., we mainly care about the footprints but not the exact time the vehicle gets there. If this is the case, we are considering the path following (PF) problem.

The PF control problem has been extensively studied in the past several decades. The work reported in [93] presents a powerful PF control structure for land robots. The control structure, however, exploits the kinematic characteristics only. The techniques therein cannot be directly applied to AUVs. In [38], Encarnacao *et al.* took advantage of the backstepping technique and recursively designed the PF controller for the underwater vehicle. In [39], Encarnacao *et al.* further considered more practical scenarios and extended the PF controller design to compensate the ocean currents. However, the controllers in [38, 39] are designed based on that the target point is selected to be the closest point on the path relative to the vehicle, hence these

methods share an inherent limitation: There exist singularities for certain initial positions of the vehicle, located exactly at the centers of path curvature. To solve this serious problem, in [77] Lapierre *et al.* proposed a novel PF control strategy. The target point was changed to a virtual particle moving along the path and possessing independent dynamic properties. The novel strategy essentially created an extra degree of freedom (DOF) which could be exploited to achieve additional specifications such as a desired speed assignment. Following this strategy, in [76] Lapierre *et al.* further developed the robust PF controller by involving an adaptive law which well handles the parametric uncertainty in the AUV model. Unfortunately, the preceding PF controller designs have no capability to incorporate the system constraints. Accommodating such constraints has been the focus of recent studies of model predictive control (MPC) for PF problems. [43, 42, 17].

The MPC provides a flexible framework that accommodates the complicated system dynamics and nonlinear system constraints. These features make MPC desirable for a variety of control problems [87, 110]. In the context of path following control, there exist several pioneering works in the literature. In [106], the PF problem was studied for marine vessels, in which MPC was used to optimize the lookahead distance in the line-of-sight (LOS) guidance law. In [102], a complete linear MPC structure was developed for the LOS-based PF control rather than just adding an optimization of the lookahead parameter. However, the LOS guidance law is only valid for PF of line segments (waypoint routes). To solve the PF problem with a desired curved path, Faulwasser *et al.* [43] provided the standard MPC formulation with an augmented system consisting of the vehicle dynamics and the path dynamics. Sufficient stabilizing conditions were derived. An extended work in terms of output MPC version was presented in [42]. The PF control designs in [43, 42] only guarantee a nonzero forward motion and cannot easily accommodate a desired speed assignment. Therefore, the incorporation of the speed assignment into the MPC formulation was studied in [6] for underactuated vehicles. In [145] Yu *et al.* took another approach that transforms the PF problem into a regulation problem where the initial state of the path dynamics was also viewed as a decision variable. The PF task can also be accomplished with a trajectory tracking controller by explicitly assigning a timing law to the path dynamics. In this way, however, it loses the flexibility of adjusting the velocity of the vehicle for possible better path following performance. In [123], the coupled path planning and tracking control problem was solved with a model predictive trajectory tracking control formulation. The MPC-based PF control applied to industrial robots

and tower cranes can be found in [44] and [17], respectively.

In practice, however, the PF control always prioritizes the path convergence requirement over the speed assignment, therefore, the MPC framework proposed in [6] provides no flexibility to adjust the relative priorities of the path convergence and speed assignment objectives. To systematically address the practical priority issue among several expected objectives, it is natural to resort to the multi-objective model predictive control (MOMPC) method. The MOMPC has been receiving increasing attentions from the system control community [13, 150, 74], because many practical applications involve meaningful multiple control objectives. Early results in MOMPC mainly focus on the problem modeling, i.e., how to formulate the interested control problem into the MOMPC framework. Examples include the sewer network application [101] and the solar cooling plant application [149]. In recent years, the MOMPC studies begin to emphasize the closed-loop properties such as stability and optimality. In [13], Bemporad *et al.* adopted the weighted sum method to solve the multi-objective optimization problem and ensured the closed-loop stability for a special class of linear systems by adding an intermediate procedure which determines appropriate weights. In [150], the utopia-tracking strategy was employed for the MOMPC problem. By constructing the partial Lagrange function and under assumptions of strong duality and Lipschitz continuity, the utopia-tracking cost can be exploited to establish the non-increasing property, which guarantees the closed-loop stability. The lexicographic ordering method which is used to solve the multi-objective optimization was studied for the MOMPC problem in [57]. The optimal value function for the most important objective can be shown a valid Lyapunov function, and thus guarantees the closed-loop stability. An interesting variant in terms of switched cost MPC was proposed in [98]. The average dwell time was imposed between switches so that the cost function can be monotonically decreasing no matter how arbitrary the switches will be made.

The MOMPC has an enlarged capacity and exhibits an attractive framework to solve AUV path following control problem. In this chapter, we propose a novel MOMPC formulation for the PF problem of an AUV so that the PF requirement prioritization can be explicitly incorporated. Regarding the competing objectives, the weighted sum (WS) method and the lexicographic ordering (LO) methods will be investigated. For the WS-MOMPC method, a logistic function is introduced to automate the weight selection. To implement the WS-MOMPC PF control, the Pontryagin minimum principle is applied. For the LO-MOMPC method, the controller

design that guarantees the path convergence is provided.

The main contributions of this chapter include the following:

- A novel MOMPC framework is developed for the AUV PF problem to explicitly incorporate the objective prioritization in the PF task.
- The WS and LO methods are proposed to solve the MOMPC problem; the implicit relation between them is explored.
- A novel logistic function is introduced to deal with the weights associated with the WS-MOMPC method; and with several properly defined barrier functions, the Pontryagin minimum principle (PMP) is applied to implement the WS-MOMPC algorithm.
- Sufficient conditions that guarantee the convergence of the MOMPC-based PF control are derived; and a feasible solution to satisfy the conditions is provided for the LO-MOMPC implementation.

5.1.2 Chapter Organization

The remaining part of this chapter is organized as follows. In Section 5.2, the detailed problem formulation is presented. Two methods that solves the MOMPC problem are proposed in Section 5.3. In Section 5.4, the conditions that guarantees the convergence of the solution are explicitly derived. Section 5.5 presents the simulation results of the proposed MOMPC PF control, and some investigation into robustness of the MOMPC method is provided. Section 5.6 concludes the entire chapter.

The notations are explained as follows. The column operation $[\rho_1^{\mathrm{T}}, ...\rho_n^{\mathrm{T}}]^{\mathrm{T}}$ is written as $col(\rho_1, ..., \rho_n)$; diagonal operation is abbreviated by $\mathrm{diag}(\cdot)$; and the weighted norm $\sqrt{\rho^{\mathrm{T}} Q \rho}$ is denoted by $\|\rho\|_Q$. For a function $p(s)$ the derivative with respect to s is represented by $p'(s)$ while the time derivative is denoted by $\dot{p}(s)$.

5.2 Problem Formulation

The AUV model studied for the PF control problem is established using the kinematic equations and the dynamic equations:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{R}(\psi)\mathbf{v} \\ \mathbf{M}^{-1}(\mathbf{u} - \mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\boldsymbol{\eta})) \end{bmatrix} = f(\mathbf{x}, \mathbf{u}) \tag{5.1}$$

where the state vector $\mathbf{x} = [x, y, \psi, u, v, r]^{\mathrm{T}}$ is consisted of the pose and velocity of the vehicle, and the control vector $\mathbf{u} = [F_u, F_v, F_r]^{\mathrm{T}}$ is generalized thrust forces and moments. The detailed expression can be found in (2.32) and (2.34).

To facilitate the following mathematical derivation, we will directly use the discretized version of (5.1) in the MOMPC problem formulation

$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k) \tag{5.2}$$

Note that the detailed expression of (5.2) is determined by the detailed discretization method. For a small sampling period, the Euler method or the Runge - Kutta method is practically sufficient.

5.2.1 The Path Following Problem

In the following, we define the PF problem. The mapping,

$$s \mapsto p(s) \in \mathbb{R}^d, \ s \in [s_0, s_1] \subset \mathbb{R} \tag{5.3}$$

describes the geometric path

$$\mathcal{P} = \{\bar{p} \in \mathbb{R}^d \mid \bar{p} = p(s), s \in [s_0, s_1]\} \tag{5.4}$$

in a d-dimensional output space. The scalar variable s is referred to as the path parameter and the domain should be closed but may be unbounded, i.e., $s_1 = +\infty$. The mapping p is assumed to be smooth and bounded. The path $\mathcal{P}$ can be interpreted as the output of the path dynamics governed by

$$\dot{s} = g(s, v_s), \ \bar{p} = p(s) \tag{5.5}$$

where v_s is the control input for the path dynamics.

Alternatively, the desired path $\mathcal{P}$ might also be defined in the n-dimensional state space by the mapping

$$s \mapsto \bar{p}_{\mathbf{x}}(s) \in \mathbb{R}^n, \ s \in [s_0, s_1] \subset \mathbb{R} \tag{5.6}$$

Usually the components of $\bar{p}_{\mathbf{x}}(s)$ are coupled according to the dynamics of the controlled system, hence cannot be freely chosen.

More precisely, the output path (5.4) defines a zero-path-error (ZPE) manifold [41] in the state space, described by the following mapping

$$s \mapsto p_{\mathbf{x}}(s, \dot{s}, \ddot{s}, ...) \in \mathbb{R}^n, \ s \in [s_0, s_1] \subset \mathbb{R} \tag{5.7}$$

We will particularly exploit the ZPE in the subsequent MOMPC formulation.

Then the PF problem that under study in this chapter can be formulated as determining the control signal $\mathbf{u}(t)$ such that the following requirements are met:

- Path Convergence. The AUV system states converge to the ZPE manifold:

$$\lim_{t\to\infty} \|\mathbf{x}(t) - p_{\mathbf{x}}(s(t), \dot{s}(t), ...)\| = 0 \tag{5.8}$$

- Forward Motion. The AUV follows the path in the direction of $\dot{s} \geq 0$.
- Speed Assignment. When moving along the path, the desired speed profile is pursued: $\dot{s} \to \dot{s}_r$.
- Constraint Satisfaction. The AUV system constraints such as thrust limits are always respected.

Note that the Path Convergence should be identified as the primary task and need to be guaranteed, and the Speed Assignment is the secondary task which can be sacrificed at times, in lieu of better performance on the primary one.

5.2.2 Zero-Path-Error Manifold

In general, finding an explicit parametrization of the ZPE manifold (5.7) could be difficult. Fortunately, for the AUV system, each state has its physical meaning, which provides a guideline for the parametrization.

Consider a desired path defined in the output space

$$\mathcal{P} = \{\bar{p} \in \mathbb{R}^2 \mid \bar{p} = [p_x(s), p_y(s)]^{\mathrm{T}}, \ s \in [s_0, s_1]\} \tag{5.9}$$

Regarding the kinematic equations (2.34), we choose the AUV surge velocity u always tangent to the desired path, then the ZPE manifold can be explicitly parameterized in the following way

$$p_{\mathbf{x}}(s, \dot{s}) = [p_x(s), p_y(s), p_\psi(s), p_u(s, \dot{s}), p_v(s), p_r(s, \dot{s})]^{\mathrm{T}} \tag{5.10}$$

with

$$
\begin{aligned}
p_\psi(s) &= \operatorname{atan2}(p'_y, p'_x) && \text{(5.11a)}\\
p_u(s,\dot{s}) &= \sqrt{p'^2_x + p'^2_y}\,\dot{s} && \text{(5.11b)}\\
p_v(s) &= 0 && \text{(5.11c)}\\
p_r(s,\dot{s}) &= \frac{p'_x p''_y - p'_y p''_x}{p'^2_x + p'^2_y}\,\dot{s} && \text{(5.11d)}
\end{aligned}
$$

Note that since we have additionally confined the velocity of the AUV, Eqn. (5.10)-(5.11) do not cover the entire ZPE manifold defined by the output path (5.9) but define a subset of the manifold which is convenient to parameterize. It specifies the unique reference for each state of the AUV to track, which facilitates the following convergence analysis.

For the manifold (5.10)-(5.11), only the first-order time derivative of the path parameter is needed. Therefore, we can choose the single integrator model to describe the path dynamics

$$\dot{s} = g(s, v_s) = v_s \tag{5.12}$$

This choice may facilitate the implementation of the MOMPC algorithm in the sense that the Forward Motion requirement will be formulated as inequality constraints on the input variables rather than on the state variables. Then those indirect methods which are based on solving the Pontryagin minimum principle [35, 54] can be applied.

5.2.3 The MOMPC Formulation

Basically, the AUV PF problem considers two aspects: Path Convergence and Speed Assignment. The Path Convergence requires the AUV to converge to $\mathcal{P}$ as fast as possible, while the Speed Assignment requires the path parameter to move in a preferred pace. Since $\dot{s}_r$ is known, stringent fulfilment of the Speed Assignment results in the determined reference states $p_{\mathbf{x}}(s_r, \dot{s}_r)$ at each time instant. This substantially degrades the PF problem to a trajectory tracking problem where the flexibility of adjusting forward speed may be lost.

To maintain this flexibility, the MOMPC framework is investigated. Consider the following multi-objective optimization problem:

$$\min_{U} \quad J(U, \xi) \tag{5.13a}$$

$$\begin{aligned} &\text{subject to} \\ &\xi_{k+1} = h(\xi_k, \omega_k),\ \xi_0 = \xi \\ &\xi_k \in \Xi,\ k = 1, 2, ..., N \\ &\omega_k \in \Omega, k = 0, 1, ..., N-1 \end{aligned} \tag{5.13b}$$

with

$$\xi = \begin{bmatrix} \mathbf{x} \\ s \end{bmatrix}, \omega = \begin{bmatrix} \mathbf{u} \\ v_s \end{bmatrix}, h(\xi, \omega) = \begin{bmatrix} f_d(\mathbf{x}, \mathbf{u}) \\ g_d(s, v_s) \end{bmatrix} \tag{5.14}$$

Here, (5.14) describes the discretized augmented system; $J(U, \xi) = [J_1(U, \xi), J_2(U, \xi)]^{\mathrm{T}}$ is a vector-valued objective function constructed for the two PF requirements; $U = col(\omega_0, \omega_1, ..., \omega_{N-1})$ is the sequence of control inputs to be optimized; ξ_k denotes the k-step predicted state from initial condition $\xi_0 = \xi$; Ξ and Ω represent the constraints on state and input, respectively.

Each objective function is in the following form:

$$J_i(U, \xi) = \sum_{k=0}^{N-1} L_i(\xi_k, \omega_k) + E_i(\xi_N) \tag{5.15}$$

where

$$L_1(\xi, \omega) = \|\xi - \xi_p\|^2_{Q_1} + \|\omega\|^2_{R_1} \tag{5.16a}$$

$$L_2(\xi, \omega) = \|\xi - \xi_t\|^2_{Q_2} + \|\omega\|^2_{R_2} \tag{5.16b}$$

$$E_1(\xi) = \|\xi - \xi_p\|^2_{P_1} \tag{5.16c}$$

$$E_2(\xi) = \|\xi - \xi_t\|^2_{P_2} \tag{5.16d}$$

Here, $\xi_p = col(p_{\mathbf{x}}(s), s)$ can be viewed as the reference for Path Convergence while $\xi_t = col(p_{\mathbf{x}}(s_r), s_r)$ is the reference for Speed Assignment with s_r generated by integrating $\dot{s}_r$ from s_0; Q_i, R_i and P_i are weighting matrices, positive definite.

Since $J_i(U, \xi)$ are in general conflicting with each other, we take advantage of the notion of Pareto optimality to measure the efficiency of a solution in the multi-objective optimization problem [33]:

Definition 3 (**Pareto Optimality**). *Let $\bar{U}$ be a feasible point for (5.13). Then, $\bar{U}$ is said to be Pareto optimal (PO) if there is no other feasible point U such that $J_i(U, \xi) \leq J_i(\bar{U}, \xi)$ for all i, and $J_i(U, \xi) < J_i(\bar{U}, \xi)$ for at least one i; and $\bar{U}$ is said to be weakly PO if $J_i(U, \xi) < J_i(\bar{U}, \xi)$ for all i.*

By definition, we shall notice that there usually exist more than one PO points. More specifically, all of the PO points constitute the so-called Pareto frontier. The complete solution of (5.13) is to determine the Pareto frontier using evolutionary methods [33] or classical methods [95]. However, for the PF problem, it is unnecessary to calculate the entire frontier. What we need is just to find one preferred point on the Pareto frontier. Even a weakly PO point may be acceptable. In this regard, two methods which are capable of handling objective prioritization are investigated to solve the MOMPC PF problem.

5.3 Solve the MOMPC Problem

5.3.1 Weighted Sum Method

The weighted sum (WS) method [95] scalarizes the vector-valued objective function by assigning a weight to each objective. Instead of solving (5.13) directly, the WS method solves the following single objective problem:

$$\begin{aligned} \min_{U} \quad & J_W = a^{\mathrm{T}} J(U, \xi) \\ \text{s.t.} \quad & \text{(5.13b)} \end{aligned} \tag{5.17}$$

where $a = [\alpha, 1-\alpha]^{\mathrm{T}}$ with $0 \leq \alpha \leq 1$.

Remark 10. *There are some comments on the WS method. First of all, the solution U^* of (5.17) is always weakly PO, or PO if $0 < \alpha < 1$ for (5.13) without any further assumptions [95]. This provides a solid theoretical support for use of the WS method. Second, by scalarizing the original problem, we can choose from a wide class of optimization algorithms, namely, the gradient descent methods [19], to efficiently solve the optimization problem. Recent developments of fast MPC implementation algorithms are exclusively designed for single objective optimizations [35]. Hence the use of WS method serves as a link between the MOMPC and a vast majority of results for single objective MPC. Finally, through the use of WS method we can incorporate some priori knowledge into the MOMPC formulation by selecting preferred weights for each objective.*

To automatically choose the appropriate weight for each objective function, we

propose the following logistic function:

$$\alpha(Er) = \frac{1}{1 + e^{-\beta Er}} \tag{5.18}$$

where $Er = \|\boldsymbol{\eta} - \boldsymbol{\eta}_p\|_K^2$ with $\boldsymbol{\eta}_p = [p_x(s), p_y(s), p_\psi(s)]^{\mathrm{T}}$ serves as an indicator of Path Convergence; $\beta > 0$ controls the change rate of the function; K is a positive definite weighting matrix. The logistic function (5.18) is smooth and monotonic. The range of (5.18) is $[0.5, 1)$ corresponding to the domain $[0, +\infty)$. Since the Path Convergence is more important than the Speed Assignment, the logistic function (5.18) always puts more weights on J_1.

We have selected the path dynamics to be a single integrator (5.12), therefore, the Forward Motion requirement can be formulated as constraints on the input of the augmented system. Further considering AUV thrust limits, we have the input constraint set Ω polyhedral:

$$\Omega = \{\omega \in \mathbb{R}^4 |\ \underline{\omega}(j) \leq \omega(j) \leq \bar{\omega}(j),\ j = 1, ..., 4\} \tag{5.19}$$

where $\underline{\omega}(j)$ and $\bar{\omega}(j)$ represent the lower and upper bounds on the jth component. Suppose $s_0 = 0$ and $s_1 = +\infty$, and there is no constraint on the system state, i.e., $\Xi = \mathbb{R}^4$. The scalarized problem (5.17) can be solved by applying Pontryagin minimum principle (PMP) with the help of barrier functions.

Define the Hamiltonian by

$$H(\xi, \lambda, \omega) = \ell(\xi, \omega) + \lambda^{\mathrm{T}} h(\xi, \omega) - \gamma(b_1(\omega) + b_2(\omega)) \tag{5.20}$$

with

$$\ell(\xi, \omega) = \alpha L_1(\xi, \omega) + (1 - \alpha) L_2(\xi, \omega) \tag{5.21}$$

$$b_1(\omega) = \sum_{j=1}^{4} \log(\omega(j) - \underline{\omega}(j)) \tag{5.22}$$

$$b_2(\omega) = \sum_{j=1}^{4} \log(\bar{\omega}(j) - \omega(j)) \tag{5.23}$$

Here, λ is called the costate, and $b_1(\omega)$, $b_2(\omega)$ are barriers for the inequality (5.19); α is the weight determined by (5.18) and γ is a small positive number. Define $E(\xi) = \alpha E_1(\xi) + (1 - \alpha) E_2(\xi)$, then PMP claims that for a local optimal control $\{\omega_i^*\}_{i=0}^{N-1}$,

there exist $\{\lambda_i^*\}_{i=0}^N$ satisfying the following optimality conditions:

$$\begin{aligned}
\xi_{k+1}^* &= h(\xi_k^*, \omega_k^*) && \text{(5.24a)}\\
\lambda_i^* &= \lambda_{i+1}^* + H_\xi^{\mathrm{T}}(\xi_i^*, \lambda_{i+1}^*, \omega_i^*)\Delta t && \text{(5.24b)}\\
\lambda_N^* &= E_\xi^{\mathrm{T}}(\xi_N^*) && \text{(5.24c)}\\
\xi_0^* &= \xi && \text{(5.24d)}\\
H_\omega(\xi_i^*, \lambda_{i+1}^*, \omega_i^*) &= 0 && \text{(5.24e)}
\end{aligned}$$

which sufficiently solves the KKT conditions by imposing N boundary conditions (5.24b)-(5.24c). Observe that given the initial state ξ and a control input $U = col(\omega_0, ..., \omega_{N-1})$, the states $\{\xi_i^*\}_{i=0}^N$ and costates $\{\lambda_i^*\}_{i=0}^N$ can be determined via recurrence relations (5.24a)-(5.24b). Therefore, at each sampling instant, we only need to solve equations (5.24e), which avoids expensive numerical operations in terms of successive linearizations in solving the KKT conditions [19]. Furthermore, fast MPC implementation algorithms that are based on solving the PMP system (5.24) such as C/GMRES [103, 121] and Gradient Projection [54] can, in principle, be used.

The proposed weighted sum based multi-objective model predictive control (WS-MOMPC) algorithm is summarized in **Algorithm 3**.

Algorithm 3 : WS-MOMPC PF Algorithm

1: Input the objective functions $J_i(U, \xi)$ in (5.15).
2: Measure current state $\xi(t)$.
3: Evaluate the value of (5.18) using $Er(t)$.
4: Solve the scalarized problem (5.17) with $\xi = \xi(t)$.
5: Let $U^* = col(\omega_0^*, ..., \omega_{N-1}^*)$ denote the solution.
6: Implement ω_0^* to the augmented system for one sampling period.
7: At next sampling time instant, set $t = t + \Delta t$, then repeat from step 2.

5.3.2 Lexicographic Ordering Method

The WS method is an indirect method which presumes that the value of each objective function at the solution is negatively correlated to its weight. Solving the nonlinear problem (5.17), however, only obtains a local minimum, which makes the negative correlation not strictly monotonic. Therefore, a direct method called lexicographic ordering (LO) is also studied for the MOMPC PF problem.

To solve the MOMPC problem (5.13), the LO method creates a procedure which considers the two objectives, one at a time, ordered by their priority [95, 57]. At each sampling time instant the following optimization problems are solved sequentially:

$$J_1^*(\xi) = \min_U \{J_1(U,\xi) \mid (5.13b)\} \tag{5.25a}$$

$$J_2^*(\xi) = \min_U \{J_2(U,\xi) \mid (5.13b), J_1(U,\xi) = J_1^*(\xi)\} \tag{5.25b}$$

Then we obtain the solution

$$U^* = \arg\min_U \{J_2(U,\xi) \mid (5.13b), J_1(U,\xi) = J_1^*(\xi)\} \tag{5.26}$$

To prevent the numerical algorithm from stalling and to improve the computational efficiency, the equality constraint $J_1(U,\xi) = J_1^*(\xi)$ in (5.25b) is often replaced by the following inequality constraint

$$J_1(U,\xi) \leq J_1^*(\xi) + \epsilon \tag{5.27}$$

where $\epsilon \geq 0$ is the tolerance.

Remark 11. *There are some comments on the LO method. Firstly, the solution (5.26) of the standard LO procedure is always PO [95]. When the relaxation (5.27) is included, the solution U^* becomes weakly PO. However, the weak Pareto optimality is practically acceptable for the AUV PF control problem. Also, we will see shortly in the next section that the relaxation is totally compatible with the path convergence conditions. Secondly, due to the imposed additional constraint (5.27) in the second layer optimization (5.25b), the PMP conditions can hardly be applied. Instead, we resort to direct methods which solve the KKT conditions.*

The LO method and WS method are closely related. To explore the internal relationship, we simplify the mathematical expression of (5.17) as follows:

$$\begin{aligned} \min_U \quad & J_W(U) = \alpha J_1(U) + (1-\alpha) J_2(U) \\ \text{s.t.} \quad & c_i(U) = 0, \quad m_j(U) \leq 0 \end{aligned} \tag{5.28}$$

Also simplify the mathematical expression of (5.25b) with the relaxation (5.27) as

$$\begin{aligned} \min_U \quad & J_2(U) \\ \text{s.t.} \quad & c_i(U) = 0, \quad m_j(U) \leq 0, \quad J_1(U) \leq \delta \end{aligned} \tag{5.29}$$

where $c_i(U)$ are the equality constraints including system dynamics and boundary conditions; $m_j(U)$ represent the inequality constraints including state constraints and control constraints; and $\delta = J_1^*(\xi) + \epsilon$.

Then we show the relationship between the two methods by explicating the KKT conditions.

Theorem 6. *If the weights for the WS method are selected precisely the same as the optimal values of dual variables for the LO method, the two methods yield the same optimal solution.*

Proof. Let $\bar{J}_W(U) = \bar{\alpha}J_1(U) + J_2(U)$ with $\bar{\alpha} = \frac{\alpha}{1-\alpha}$. Obviously, minimizing J_W is equivalent to minimizing $\bar{J}_W$. Define the Lagrangian for (5.28) as

$$\bar{L}_W = J_2(U) + \Sigma\bar{\lambda}_i c_i(U) + \Sigma\bar{\mu}_j m_j(U) + \bar{\alpha}J_1(U)$$

where $\bar{\lambda}_i$ and $\bar{\mu}_j$ are dual variables.

Define the Lagrangian for (5.29) as

$$L_\delta = J_2(U) + \Sigma\lambda_i c_i(U) + \Sigma\mu_j m_j(U) + \nu(J_1(U) - \delta)$$

where λ_i, μ_j and ν are dual variables. Then we list the detailed KKT conditions for (5.28) as follows:

$$\nabla_U \bar{L}_W(U^*) = 0, \quad c_i(U^*) = 0, \quad m_j(U^*) \leq 0 \tag{5.30a}$$

$$\bar{\mu}_j^* m_j(U^*) = 0, \quad \bar{\mu}_j^* \geq 0 \tag{5.30b}$$

where ∇_U represents the gradient with respect to U. Let us also list the detailed KKT conditions for problem (5.29)

$$\nabla_U L_\delta(U^*) = 0, \quad c_i(U^*) = 0, \quad m_j(U^*) \leq 0 \tag{5.31a}$$

$$\mu_j^* m_j(U^*) = 0, \quad \mu_j^* \geq 0 \tag{5.31b}$$

$$J_1(U^*) - \delta \leq 0 \tag{5.31c}$$

$$\nu^*(J_1(U^*) - \delta) = 0, \quad \nu^* \geq 0 \tag{5.31d}$$

Further expanding $\nabla_U \bar{L}_W(U^*)$ and $\nabla_U L_\delta(U^*)$, we have

$$\nabla J_2(U^*) + \Sigma \bar{\lambda}_i^* \nabla c_i(U^*) + \Sigma \bar{\mu}_j^* \nabla m_j(U^*) + \bar{\alpha} \nabla J_1(U^*) = 0$$

$$\nabla J_2(U^*) + \Sigma \lambda_i^* \nabla c_i(U^*) + \Sigma \mu_j^* \nabla m_j(U^*) + \nu^* \nabla J_1(U^*) = 0$$

We notice that the main differences between the above two sets of KKT conditions are the presence of (5.31c) and (5.31d). If $\bar{\alpha} = \nu^*$, (5.30) and (5.31) yield the same optimal solution U^* with $\bar{\lambda}_i^* = \lambda_i^*$ and $\bar{\mu}_j^* = \mu_j^*$ due to the fact that:

Case 1: The last inequality is active, i.e., $J_1(U^*) = \delta$. Then (5.31c) and (5.31d) become solely $\nu^* \geq 0$. By definition we have $\bar{\alpha} \geq 0$, therefore, in this case the two sets of KKT conditions are exactly the same.

Case2: The last inequality is inactive, i.e., $J_1(U^*) < \delta$. Then (5.31c) and (5.31d) become $\nu^* = 0$. So we have $\bar{\alpha} = \nu^* = 0$, which make the two sets of KKT conditions identical. □

Remark 12. *As seen from the above proof, the WS method and LO method are implicitly related by the optimal value of dual variables. Therefore, it is difficult to find the corresponding ϵ which generates the same solution as that using WS method with the change of α obeying (5.18).*

The proposed lexicographic ordering based multi-objective model predictive control (LO-MOMPC) algorithm is summarized in **Algorithm 4**.

Algorithm 4 : LO-MOMPC PF Algorithm

1: Input the objective functions $J_i(U, \xi)$ in (5.15).
2: Measure current state $\xi(t)$.
3: Sequentially solve the lexicographic ordering subproblems (5.25) with $\xi = \xi(t)$.
4: Let $U^* = col(\omega_0^*, ..., \omega_{N-1}^*)$ be the solution of the second layer problem (5.25b).
5: Implement ω_0^* to the augmented system for one sampling period.
6: At next sampling time instant, set $t = t + \Delta t$, then repeat from step 2.

5.4 Convergence Analysis

In order to follow the standard analysis procedure used in conventional MPC [111], the original MOMPC problem (5.13) needs to be modified and reformulated as the

regulation problem of a well-defined error dynamics:

$$\min_{\tilde{U}} \quad J(\tilde{U}, \tilde{\xi}) \tag{5.32a}$$

$$\begin{aligned} &\text{subject to} \\ &\tilde{\xi}_{k+1} = \tilde{h}(\tilde{\xi}_k, \tilde{\omega}_k),\ \tilde{\xi}_0 = \tilde{\xi} \\ &\tilde{\xi}_k \in \tilde{\Xi},\ k = 1, 2, ..., N \\ &\tilde{\omega}_k \in \tilde{\Omega}, k = 0, 1, ..., N-1 \end{aligned} \tag{5.32b}$$

where $\tilde{h}$ denotes the error dynamics; $J(\tilde{U}, \tilde{\xi}) = [J_1, J_2]^{\mathrm{T}}$ with $J_i = \sum_{k=0}^{N-1} L_i(\tilde{\xi}_k, \tilde{\omega}_k) + E_i(\tilde{\xi}_N)$; $\tilde{\Xi}$ and $\tilde{\Omega}$ denote system constraints.

Then the convergence of the MOMPC solution can be analyzed with the following assumptions.

Assumption 4. *The functions $\tilde{h}(\cdot)$, $L_i(\cdot)$ and $E_i(\cdot)$ are continuous; $\tilde{h}(0,0) = 0$, $L_i(0,0) = 0$ and $E_i(0) = 0$.*

Assumption 5. *The sets $\tilde{\Xi}$ and $\tilde{\Pi}$ are closed, and $\tilde{\Omega}$ is compact. Each set contains the origin in its interior.*

Assumption 6. *There exists an invariant set $\tilde{\Pi}$ for the error dynamics $\tilde{h}$, and a local controller $\kappa(\tilde{\xi})$ such that*

$$\begin{aligned} &\kappa(\tilde{\xi}) \in \tilde{\Omega}, \quad \tilde{h}(\tilde{\xi}, \kappa(\tilde{\xi})) \in \tilde{\Pi}, \quad i = 1, 2 \\ &E_i(\tilde{h}(\tilde{\xi}, \kappa(\tilde{\xi}))) - E_i(\tilde{\xi}) + L_i(\tilde{\xi}, \kappa(\tilde{\xi})) \leq 0. \end{aligned} \tag{5.33}$$

for any $\tilde{\xi} \in \tilde{\Pi}$.

For the WS-MOMPC scheme, we need to modify the problem by imposing the terminal constraints as follows:

$$J_W^*(\tilde{\xi}, \alpha) = \min_{\tilde{U}} \{a^{\mathrm{T}} J(\tilde{U}, \tilde{\xi}) \mid (5.32\text{b}), \tilde{\xi}_N \in \tilde{\Pi}\} \tag{5.34}$$

Also, before solving (5.34), at each sampling instant (except for the first time $t = 0$), a subproblem that determines the appropriate weight α has to be solved:

$$\alpha^*(\tilde{\xi}, \alpha_d, J_\alpha) = \arg\min_{\alpha} \quad f_\alpha(\alpha - \alpha_d) \tag{5.35a}$$

$$\begin{aligned} \text{s.t.} \quad & J_W^*(\tilde{\xi}, \alpha) \leq J_\alpha \\ & 0 \leq \alpha \leq 1 \end{aligned} \tag{5.35b}$$

where α_d is the target weight calculated by (5.18); f_a is a convex function that measures the distance between α and α_d; Let $\tilde{U}^*_{-1} = col(\tilde{\omega}^*_{-1,0}, ..., \tilde{\omega}^*_{-1,N-1})$ and a^*_{-1} be the previous time solutions for (5.34) and (5.35), respectively. Then $J_\alpha = a^{*\mathrm{T}}_{-1} J(\tilde{U}^0, \tilde{\xi})$ with $\tilde{U}^0 = col(\tilde{\omega}^*_{-1,1}, ..., \tilde{\omega}^*_{-1,N-1}, \kappa(\tilde{\xi}_{-1,N}))$.

Then we can have the following results on the convergence of the solution:

Theorem 7. *Suppose* **Assumptions 4 - 6** *are satisfied, and the problem (5.34) is feasible at time $t = 0$. Then with the WS-MOMPC algorithm, the error state converges to the origin, i.e., $\tilde{\xi} \to 0$ as $t \to \infty$.*

Proof. The recursive feasibility can be ensured because the shifted control sequence $\tilde{U}^0$ is always feasible for (5.34) and $a^{*\mathrm{T}}_{-1}$ is always feasible for (5.35) at the next time instant.

Evaluating the optimal value function of J^*_W at two successive time instants, we have

$$\begin{aligned}
\Delta J_W &= J_W(\tilde{U}^*, \tilde{\xi}) - J_W(\tilde{U}^*_{-1}, \tilde{\xi}_{-1,0}) \\
&= a^{*\mathrm{T}} J(\tilde{U}^*, \tilde{\xi}) - a^{*\mathrm{T}}_{-1} J(\tilde{U}^*_{-1}, \tilde{\xi}_{-1,0}) \\
&\leq a^{*\mathrm{T}}_{-1} J(\tilde{U}^0, \tilde{\xi}) - a^{*\mathrm{T}}_{-1} J(\tilde{U}^*_{-1}, \tilde{\xi}_{-1,0}) \\
&= \alpha^*_{-1}[E_1(\tilde{h}(\tilde{\xi}_{-1,N}, \kappa(\tilde{\xi}_{-1,N}))) - E_1(\tilde{\xi}_{-1,N}) \\
&\quad + L_1(\tilde{\xi}_{-1,N}, \kappa(\tilde{\xi}_{-1,N})) - L_1(\tilde{\xi}_{-1,0}, \tilde{\omega}^*_{-1,0})] \\
&\quad + (1 - \alpha^*_{-1})[E_2(\tilde{h}(\tilde{\xi}_{-1,N}, \kappa(\tilde{\xi}_{-1,N}))) - E_2(\tilde{\xi}_{-1,N}) \\
&\quad + L_2(\tilde{\xi}_{-1,N}, \kappa(\tilde{\xi}_{-1,N})) - L_2(\tilde{\xi}_{-1,0}, \tilde{\omega}^*_{-1,0})]
\end{aligned}$$

By **Assumption 6**, $\Delta J_W \leq -\alpha^*_{-1} L_1(\tilde{\xi}_{-1,0}, \tilde{\omega}^*_{-1,0}) - (1 - \alpha^*_{-1}) L_2(\tilde{\xi}_{-1,0}, \tilde{\omega}^*_{-1,0}) \leq 0$. By construction, we have $J_W \geq 0$. Therefore, $J^*_W(t)$ is a non-increasing sequence and lower bounded by zero. By contradiction, we have $-\alpha^*_{-1} L_1(\tilde{\xi}_{-1,0}, \tilde{\omega}^*_{-1,0}) - (1 - \alpha^*_{-1}) L_2(\tilde{\xi}_{-1,0}, \tilde{\omega}^*_{-1,0}) \to 0$ as $t \to \infty$. Since $Q_i > 0$ and $R_i > 0$ the convergence of the error state can be guaranteed, i.e., $\tilde{\xi} \to 0$ as $t \to \infty$. □

For the LO-MOMPC PF scheme, the convergence of the MOMPC solution can be guaranteed with the following assumption.

Assumption 7. *There exist an invariant set $\tilde{\Pi}$ for the error dynamics $\tilde{h}$, containing the origin in the interior, and a local feedback control $\kappa_1(\tilde{\xi})$ such that*

$$\begin{aligned}
&\kappa_1(\tilde{\xi}) \in \tilde{\Omega}, \quad \tilde{h}(\tilde{\xi}, \kappa_1(\tilde{\xi})) \in \tilde{\Pi}, \\
&E_1(\tilde{h}(\tilde{\xi}, \kappa_1(\tilde{\xi}))) - E_1(\tilde{\xi}) + L_1(\tilde{\xi}, \kappa_1(\tilde{\xi})) \leq 0.
\end{aligned} \tag{5.36}$$

for any $\tilde{\xi} \in \tilde{\Pi}$.

We modify the LO-MOMPC problem by imposing the terminal constraints:

$$J_1^*(\tilde{\xi}) = \min_{\tilde{U}}\{J_1(\tilde{U},\tilde{\xi}) \mid (5.32b), \tilde{\xi}_N \in \tilde{\Pi}\} \tag{5.37a}$$

$$J_2^*(\tilde{\xi}) = \min_{\tilde{U}} \left\{ J_2(\tilde{U},\tilde{\xi}) \middle| \begin{array}{c} J_1(\tilde{U},\tilde{\xi}) \leq J_1^*(\tilde{\xi}) + \epsilon, \\ (5.32b), \quad \tilde{\xi}_N \in \tilde{\Pi} \end{array} \right\} \tag{5.37b}$$

Then we have the following results on the convergence of the solution:

Theorem 8. *Suppose* **Assumptions 4, 5 and 7** *are satisfied, and for $\epsilon = 0$ the problem (5.37) is feasible at time $t = 0$. Let $\tilde{\xi}_{-1,0}$ denote the previous system state and $\tilde{\omega}^*_{-1,0}$ denote the first element of the previous time solution. For $t > 0$, if we choose $\epsilon = L_1(\tilde{\xi}_{-1,0}, \tilde{\omega}^*_{-1,0}) - \tilde{\xi}^{\mathrm{T}} Q_\epsilon \tilde{\xi}$ with $Q_1 > Q_\epsilon > 0$, then the error state converges to the origin, i.e., $\tilde{\xi} \to 0$ as $t \to \infty$.*

Proof. Let $\tilde{U}^*_{-1} = col(\tilde{\omega}^*_{-1,0}, ..., \tilde{\omega}^*_{-1,N-1})$ denote the solution for the previous sampling time instant, and $\tilde{\xi}_{-1,i}$ denote the corresponding state prediction for $i = 0, 1, ..., N$. An initial guess $\tilde{U}^0 = col(\tilde{\omega}^*_{-1,1}, ..., \tilde{\omega}^*_{-1,N-1}, \kappa_1(\tilde{\xi}_{-1,N}))$ can be constructed for the first layer subproblem (5.37a), which admits the horizontal feasibility. The hierarchical feasibility is also preserved since the solution of (5.37a) is always feasible for (5.37b).

Evaluating the optimal value function of J_1^* at these two successive sampling time instants, we have

$$\begin{array}{rcl} \Delta J_1 & = & J_1(\tilde{U}^*, \tilde{\xi}) - J_1(\tilde{U}^*_{-1}, \tilde{\xi}_{-1,0}) \\ & \leq & J_1(\tilde{U}^0, \tilde{\xi}) - J_1(\tilde{U}^*_{-1}, \tilde{\xi}_{-1,0}) + \epsilon \\ & = & E_1(\tilde{h}(\tilde{\xi}_{-1,N}, \kappa_1(\tilde{\xi}_{-1,N}))) - E_1(\tilde{\xi}_{-1,N}) \\ & & + L_1(\tilde{\xi}_{-1,N}, \kappa_1(\tilde{\xi}_{-1,N})) - \tilde{\xi}^{\mathrm{T}} Q_\epsilon \tilde{\xi} \end{array} \tag{5.38}$$

With Assumption 7 we have $\Delta J_1 \leq -\tilde{\xi}^{\mathrm{T}} Q_\epsilon \tilde{\xi} \leq 0$. By construction, we know that $J_1 \geq 0$. Therefore, $J_1^*(t)$ is a non-increasing sequence and lower bounded by zero. By contradiction, we have $-\tilde{\xi}^{\mathrm{T}} Q_\epsilon \tilde{\xi} \to 0$ as $t \to \infty$. Since $Q_\epsilon > 0$, the convergence of the error state trajectory to the origin can be guaranteed, i.e., $\tilde{\xi} \to 0$ as $t \to \infty$. □

Remark 13. *The differences between* **Assumption 6** *and* **Assumption 7** *reflect the underlying differences between the WS-MOMPC and LO-MOMPC schemes.* **Assumption 6** *requires that for each and every objective J_i we need to find the same local feedback control law $\kappa(\tilde{\xi})$ and terminal region $\tilde{\Pi}$ such that the inequalities in*

(5.33) can hold. This will guarantee both path convergence and speed assignment but complicate the MPC design at the same time. In terms of the path following control problem, one possible reformulation that satisfies **Assumption 6** *is the stringent fulfilment of the speed assignment along the ZPE manifold. In this way, however, the PF problem is solved as a trajectory tracking problem with two performance indices that emphasize different error states (position vs. forward speed), and thus the extra degree of freedom of adjusting $s(t)$ is lost. In contrast,* **Assumption 7** *requires to find the local feedback control law and terminal region for the most important objective J_1 such that the inequality in (5.36) can hold. This will only rigorously guarantee the path convergence which is the primary task in the PF problem. However, since the path convergence and speed assignment are physically achievable and the relaxation parameter ϵ is incorporated, the control on speed assignment is often satisfactory. Adopting LO-MOMPC scheme simplifies the MPC design and retain the flexibility of adjusting $s(t)$ on the fly.*

Remark 14. *There are different ways to reformulate the problem as the regulation of the error dynamics and to satisfy the assumptions. As an example, we can take advantage of the ZPE manifold. Since the ZPE manifold (5.11) serves as a reference for the AUV to track, the following relation holds:*

$$\dot{p}_x = p_u \cos p_\psi - p_v \sin p_\psi \tag{5.39a}$$

$$\dot{p}_y = p_u \sin p_\psi + p_v \cos p_\psi \tag{5.39b}$$

$$\dot{p}_\psi = p_r \tag{5.39c}$$

Decomposing the kinematic error $\tilde{\boldsymbol{\eta}}$ in the vessel parallel reference frame, we have

$$\tilde{\boldsymbol{\eta}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\psi} \end{bmatrix} = \begin{bmatrix} -\cos\psi & -\sin\psi & 0 \\ \sin\psi & -\cos\psi & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x - p_x \\ y - p_y \\ \psi - p_\psi \end{bmatrix} \tag{5.40}$$

Define the velocity error $\tilde{\mathbf{v}}$ in the following way,

$$\tilde{\mathbf{v}} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{r} \end{bmatrix} = \begin{bmatrix} u - p_u \cos\tilde{\psi} + p_v \sin\tilde{\psi} \\ v - p_u \sin\tilde{\psi} - p_v \cos\tilde{\psi} \\ r - p_r \end{bmatrix} \tag{5.41}$$

Further define $\tilde{\boldsymbol{\tau}} = col(\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\tau}_3)$ *in which*

$$\begin{aligned}\tilde{\tau}_1 &= \tfrac{F_u}{M_{\dot{u}}} + \tfrac{M_{\dot{v}}}{M_{\dot{u}}}(\tilde{v} + p_u \sin\tilde{\psi} + p_v \cos\tilde{\psi})(\tilde{r} + p_r) \\ &\quad -\tfrac{X_u}{M_{\dot{u}}}(\tilde{u} + p_u \cos\tilde{\psi} - p_v \sin\tilde{\psi}) - \tfrac{D_u}{M_{\dot{u}}}(\tilde{u} + p_u \cos\tilde{\psi} \\ &\quad -p_v \sin\tilde{\psi})|\tilde{u} + p_u \cos\tilde{\psi} - p_v \sin\tilde{\psi}| - (\dot{p}_u \cos\tilde{\psi} \\ &\quad +p_u \sin\tilde{\psi}\tilde{r} - \dot{p}_v \sin\tilde{\psi} + p_v \cos\tilde{\psi}\tilde{r})\end{aligned} \tag{5.42}$$

$$\begin{aligned}\tilde{\tau}_2 &= \tfrac{F_v}{M_{\dot{v}}} - \tfrac{M_{\dot{u}}}{M_{\dot{v}}}(\tilde{u} + p_u \cos\tilde{\psi} - p_v \sin\tilde{\psi})(\tilde{r} + p_r) \\ &\quad -\tfrac{Y_v}{M_{\dot{v}}}(\tilde{v} + p_u \sin\tilde{\psi} + p_v \cos\tilde{\psi}) - \tfrac{D_v}{M_{\dot{v}}}(\tilde{v} + p_u \sin\tilde{\psi} \\ &\quad +p_v \cos\tilde{\psi})|\tilde{v} + p_u \sin\tilde{\psi} + p_v \cos\tilde{\psi}| - (\dot{p}_u \sin\tilde{\psi} \\ &\quad -p_u \cos\tilde{\psi}\tilde{r} + \dot{p}_v \cos\tilde{\psi} + p_v \sin\tilde{\psi}\tilde{r})\end{aligned} \tag{5.43}$$

$$\begin{aligned}\tilde{\tau}_3 &= \tfrac{F_r}{M_{\dot{r}}} - \tfrac{M_{\dot{v}} - M_{\dot{u}}}{M_{\dot{r}}}(\tilde{u} + p_u \cos\tilde{\psi} - p_v \sin\tilde{\psi})(\tilde{v} + p_u \sin\tilde{\psi} \\ &\quad +p_v \cos\tilde{\psi}) - \tfrac{N_r}{M_{\dot{r}}}(\tilde{r} + p_r) - \tfrac{D_r}{M_{\dot{r}}}(\tilde{r} + p_r)|\tilde{r} + p_r| - \dot{p}_r\end{aligned} \tag{5.44}$$

Then it can be shown that

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{\psi}} \\ \dot{\tilde{u}} \\ \dot{\tilde{v}} \\ \dot{\tilde{r}} \end{bmatrix} = \begin{bmatrix} p_r\tilde{y} - \tilde{u} + \tilde{y}\tilde{r} \\ -p_r\tilde{x} - \tilde{v} - \tilde{x}\tilde{r} \\ -\tilde{r} \\ \tilde{\tau}_1 \\ \tilde{\tau}_2 \\ \tilde{\tau}_3 \end{bmatrix} \triangleq \tilde{f}(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\tau}}) \tag{5.45}$$

and at the origin $(\mathbf{0}, \mathbf{0})$*, it admits that*

$$\dot{p}_u = \frac{M_{\dot{v}}}{M_{\dot{u}}} p_v p_r - \frac{X_u}{M_{\dot{u}}} p_u - \frac{D_u}{M_{\dot{u}}} p_u |p_u| + \frac{F_u}{M_{\dot{u}}} \tag{5.46a}$$

$$\dot{p}_v = -\frac{M_{\dot{u}}}{M_{\dot{v}}} p_u p_r - \frac{Y_v}{M_{\dot{v}}} p_v - \frac{D_v}{M_{\dot{v}}} p_v |p_v| + \frac{F_v}{M_{\dot{v}}} \tag{5.46b}$$

$$\dot{p}_r = \frac{M_{\dot{u}} - M_{\dot{v}}}{M_{\dot{r}}} p_u p_v - \frac{N_r}{M_{\dot{r}}} p_r - \frac{D_r}{M_{\dot{r}}} p_r |p_r| + \frac{F_r}{M_{\dot{r}}} \tag{5.46c}$$

Comparing (5.39) and (5.46) with (2.34) and (2.32), we find that the ZPE manifold can be viewed as the state trajectory generated by a virtual AUV owning exactly the same kinematic and dynamic properties as the real AUV. Since $(\mathbf{0}, \mathbf{0})$ *indicates that the AUV is already on the manifold, the control input* F_u*,* F_v *and* F_r *will make the AUV stay on the manifold.*

Detailed derivation is similar to (3.7) - (3.14).

Define $\tilde{\xi} = col(\tilde{\boldsymbol{\eta}}, \tilde{\mathbf{v}}, 0)$, $\tilde{\omega} = col(\tilde{\boldsymbol{\tau}}, 0)$ *and the error dynamics* $\tilde{h} = col(\tilde{f}, g)$. *We modify* J_1 *with*

$$L_1(\tilde{\xi}, \tilde{\omega}) = \|\tilde{\xi}\|^2_{Q_1} + \|\tilde{\omega}\|^2_{R_1}, \quad E_1(\tilde{\xi}) = \|\tilde{\xi}\|^2_{P_1} \tag{5.47}$$

Leave L_2 *and* E_2 *the same as in (5.16). In this way,* **Assumption 4** *can be satisfied.*

Notice that the error control signal construction (5.42)-(5.44) needs the values of $\dot{p}_u$, $\dot{p}_v$ *and* $\dot{p}_r$. *By (5.11) we have* $\dot{p}_v = 0$ *and*

$$\dot{p}_u = (p'_x p''_x + p'_y p''_y)(p'^2_x + p'^2_y)^{-\frac{1}{2}} \dot{s}^2 + (p'^2_x + p'^2_y)^{\frac{1}{2}} \ddot{s} \tag{5.48a}$$

$$\dot{p}_r = \frac{(p'^2_x + p'^2_y)(p'_x p'''_y - p'_y p'''_x) - 2(p'_x p''_y - p'_y p''_x)(p'_x p''_x + p'_y p''_y)}{(p'^2_x + p'^2_y)^2} \dot{s}^2 + \frac{p'_x p''_y - p'_y p''_x}{p'^2_x + p'^2_y} \ddot{s} \tag{5.48b}$$

Since the above calculations require the information of $\ddot{s}$, *the path dynamics need to be modeled as a second order integrator, i.e.,*

$$\dot{z} = \tilde{A} z + \tilde{B} v_s, \ \tilde{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \ \tilde{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.49}$$

where $z = [s, \dot{s}]^{\mathrm{T}}$. *Accordingly, the Forward Motion requirement will be formulated as* $z_2 = \dot{s} \geq 0$. *We further employ the manifold (5.11) as the terminal constraint, i.e.,* $\tilde{\Pi} = \{0\}$. *Then* **Assumption 5** *and* **Assumption 7** *can be satisfied with* $\kappa_1(0) = 0$.

5.5 Simulation Results

In this section, we present the simulation results of the Falcon AUV to follow a sinusoidal path $p_x(s) = s$ and $p_y(s) = \sin(s)$ with $s \geq 0$ in the local level plane. For the Speed Assignment, instead of directly setting a preferred path velocity $\dot{s}_r$, we set a desired surge velocity $u_r = 1$ (m/s) for the vehicle. This is a common selection in engineering practice because for AUVs with control surfaces (fins), the surge velocity effectively influences the actuation efficiency of the control surface (higher speed with higher reactivity). Then, according to (5.11b), we have the following explicit expression

$$\dot{s}_r = u_r (p'^2_x + p'^2_y)^{-\frac{1}{2}} \tag{5.50}$$

to calculate ξ_t at each sampling instant.

All the simulations are based on the experimentally identified dynamic model

of Falcon. The model parameter details can be found in Table 2.1. To solve the PMP equation (5.24e) the Trust-Region-Dogleg algorithm [147] is adopted. To direct solve the NLP problem with KKT conditions, the embedded sequential quadratic programming algorithm [19] of Matlab function *fmincon* is used.

5.5.1 PF Control Using WS-MOMPC

For the WS-MOMPC PF control, we select the parameters as follows: The sampling period $\Delta t = 0.1sec$, prediction horizon $N = 10$, $\gamma = 10^{-4}$, $K = I$ and the weighting matrices $Q_1 = \text{diag}([10^5, 10^5, 10^2, 0.1, 0.1, 0.1, 0.1])$, $Q_2 = \text{diag}([1, 1, 1, 10^3, 0.1, 0.1, 0.1])$, $P_1 = \text{diag}([100, 100, 10, 10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}])$, $P_2 = \text{diag}([0.1, 0.1, 0.1, 100, 10^{-3}, 10^{-3}, 10^{-3}])$, $R_1 = R_2 = \text{diag}([10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}])$. The control limits $F_{u,\max} = 500$ (N), $F_{v,\max} = 500$ (N) and $F_{r,\max} = 500$ (Nm), and initial conditions $\mathbf{x}(0) = [0.5, 0, 0, 0, 0, 0]^{\mathrm{T}}$ and $s(0) = 0$.

The AUV path-following results with different β values are shown in Figure 5.1. It can be observed that (i) for all of the three cases, the AUV trajectory successfully converges to the desired sinusoidal path, which validates the effectiveness of the proposed WS-MOMPC PF method; (ii) for each case, at the beginning, the AUV

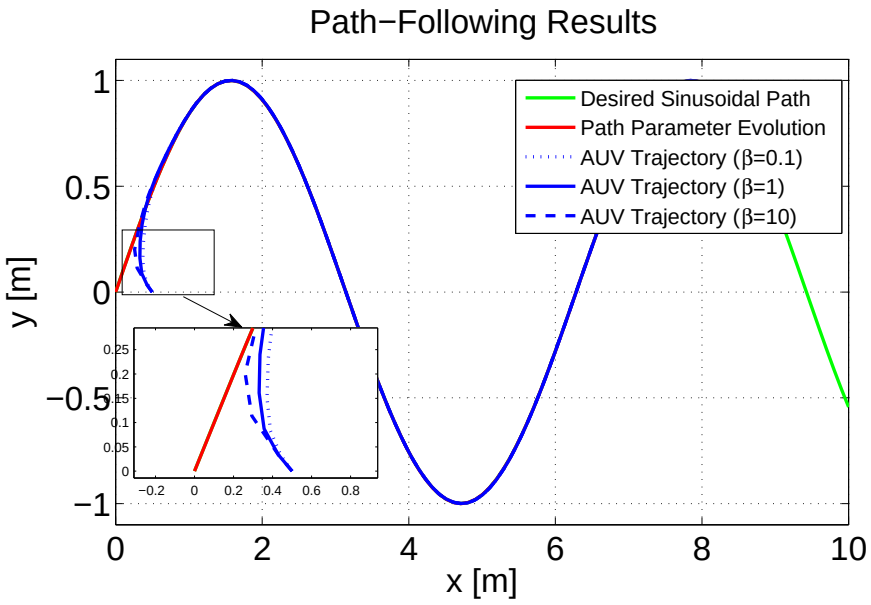

Figure 5.1: The AUV PF results with WS-MOMPC.

moves in the direction that is almost perpendicular to the path in order to get the fastest convergence, which is a desirable property; (iii) the larger β value results in

the faster path convergence as the β value accounts for the sensitivity (slope) of the logistic function.

Figure 5.2 records the surge velocity of the AUV during the simulation (for the case of $\beta = 1$). As we can see, the surge velocity keeps very well at the desired speed

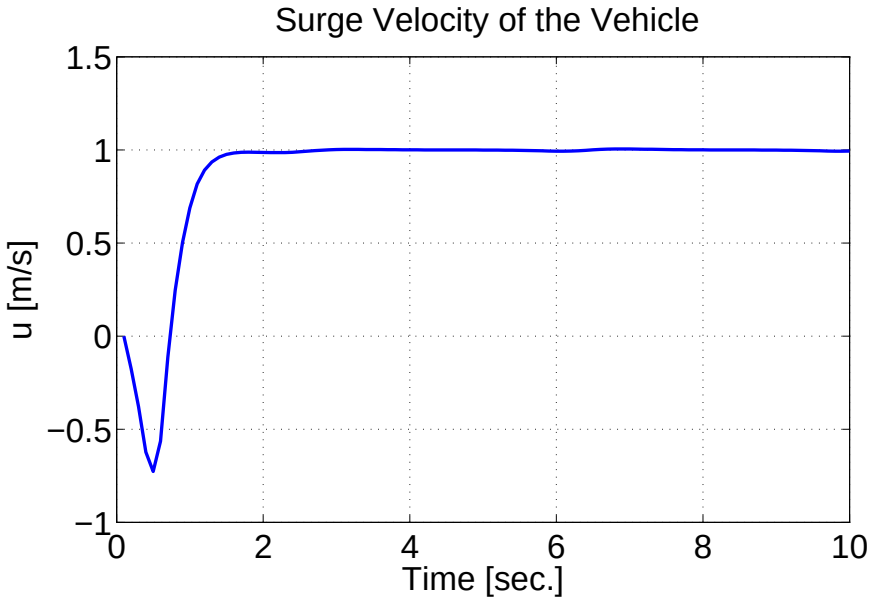

Figure 5.2: Surge velocity of the AUV (WS-MOMPC).

after the sacrifice for Path Convergence in the beginning. In Figure 5.3, the general thrust forces and moments as well as the imaginary path control inputs are plotted. As expected, they are all within the corresponding ranges of permitted values, which validates the effectiveness of barrier functions in the PMP-based implementation.

5.5.2 PF Control Using LO-MOMPC

For the LO-MOMPC scheme, we simulate the AUV PF control using the way suggested in **Remark 14**. The parameters are chosen as follows: The sampling period $\Delta t = 0.1sec$, prediction horizon $N = 10$ and weighting matrices $Q_1 = \text{diag}([10^5, 10^5, 10^3, 10^3, 10^{-3}, 10^{-3}, 10^{-3}])$, $Q_2 = \text{diag}([10^4, 10^4, 10^2, 10^3, 10^{-3}, 10^{-3}, 10^3])$, $P_1 = \text{diag}([100, 100, 10, 100, 10^{-3}, 10^{-3}, 10^{-3}])$, $P_2 = \text{diag}([100, 100, 10, 100, 10^{-3}, 10^{-3}, 10^2])$, $R_1 = R_2 = \text{diag}([10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}])$. The control limits $F_{u,\max} = 500$ (N), $F_{v,\max} = 500$ (N) and $F_{r,\max} = 500$ (Nm), and initial conditions $\mathbf{x}(0) = [0.5, 0, 0, 0, 0, 0]^{\mathrm{T}}$ and $s(0) = 0$.

The AUV path-following results are shown in Figure 5.4. Similar observations can be made: (i) the AUV trajectory successfully converge to the desired sinusoidal path,

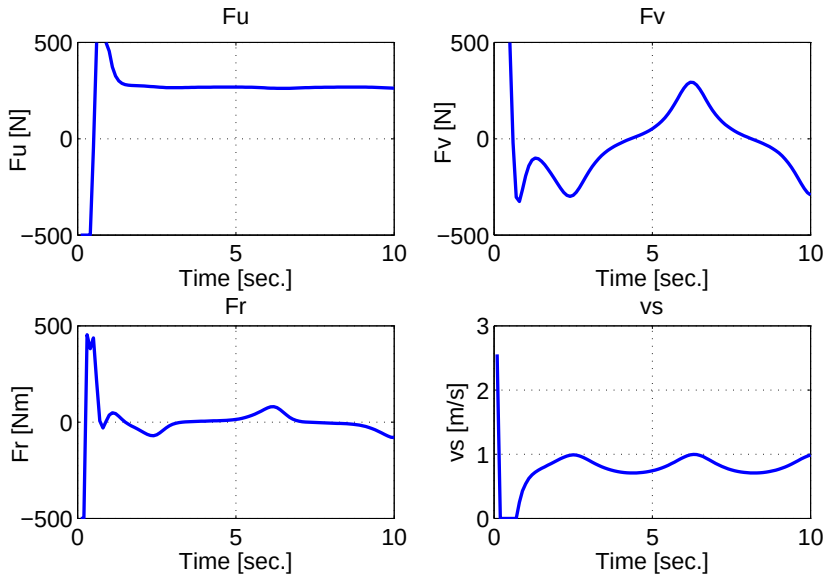

Figure 5.3: Control inputs of the augmented system (WS-MOMPC).

which validates the effectiveness of the proposed LO-MOMPC PF method; (ii) in the

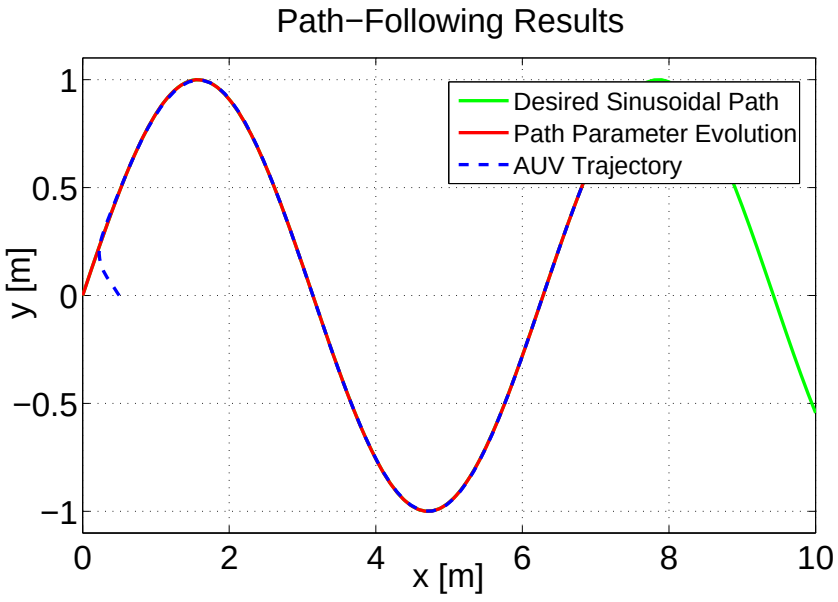

Figure 5.4: The AUV PF results using LO-MOMPC.

beginning, the AUV moves in the direction almost perpendicular to the desired path in order to get the fastest convergence, which reflects the different priorities of the two PF requirements.

Figure 5.5 records the surge velocity of the AUV during the simulation. The

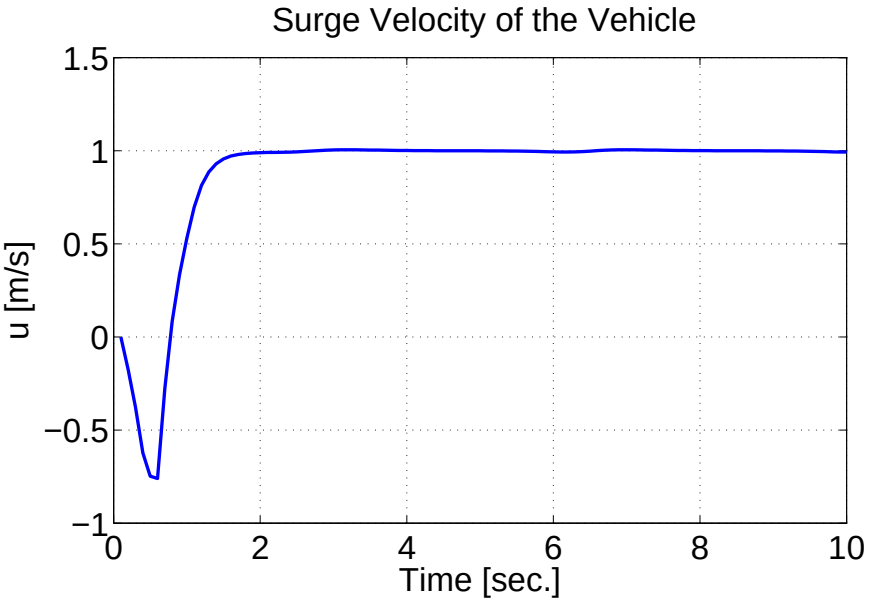

Figure 5.5: Surge velocity of the AUV (LO-MOMPC).

surge velocity keeps very well at the desired speed after the initial sacrifice for path convergence. In Figure 5.6, the general thrust forces and moments as well as the imaginary path control input are plotted.

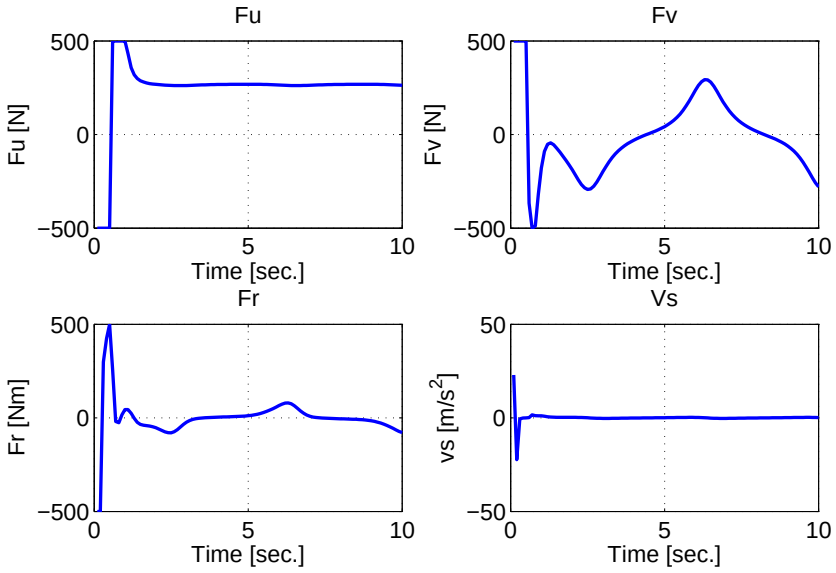

Figure 5.6: Control inputs of the augmented system (LO-MOMPC).

Also the optimal values for the two objective functions are plotted in Figure 5.7. It can be seen that J_1^* decreases monotonically, which indicates that the path convergence can be obtained; while J_2^* admits an initial hike, which conforms to the strict prioritization between the two control objectives.

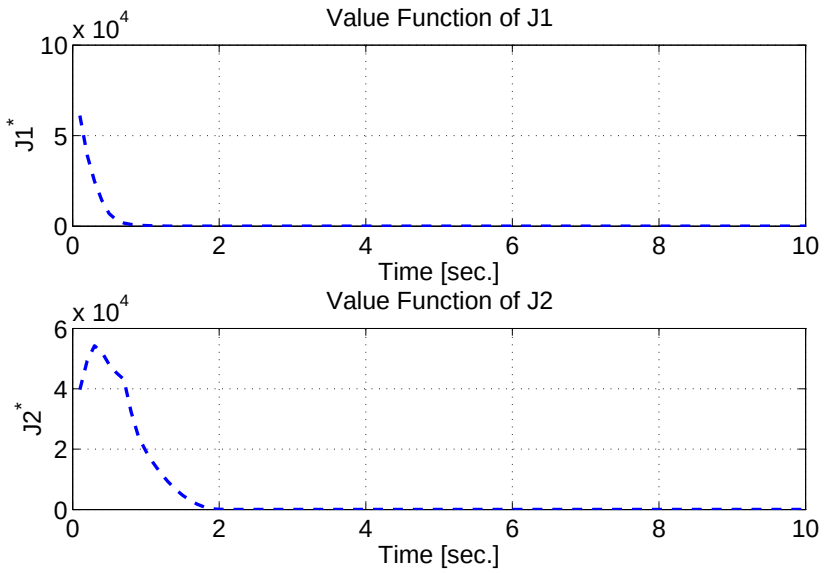

Figure 5.7: The optimal value functions.

5.5.3 Robustness Test

The receding horizon control paradigm makes MPC robust to parametric uncertainty of the system model. This feature is extremely suitable for control of underwater vehicles whose dynamic model is always with uncertainties due to the poor knowledge of hydrodynamic coefficients. In Figure 5.8, the path-following control is performed using a dynamic model with 20% model errors. As we can see, the AUV still closely follows the desired path, and the control performance is quite satisfactory. The surge velocity of the vehicle is plotted in Figure 5.9, which well demonstrates the robustness of the proposed MOMPC method.

In addition, the robustness is also tested through the path following task of non-differentiable paths since these types of path (e.g., the lawn mower pattern) represent actual oceanographic survey applications.

Notice that in Section 5.2.1, the desired path is assumed to be smooth. This is to facilitate the unified expression of parametrization of the ZPE manifold in (5.4) -

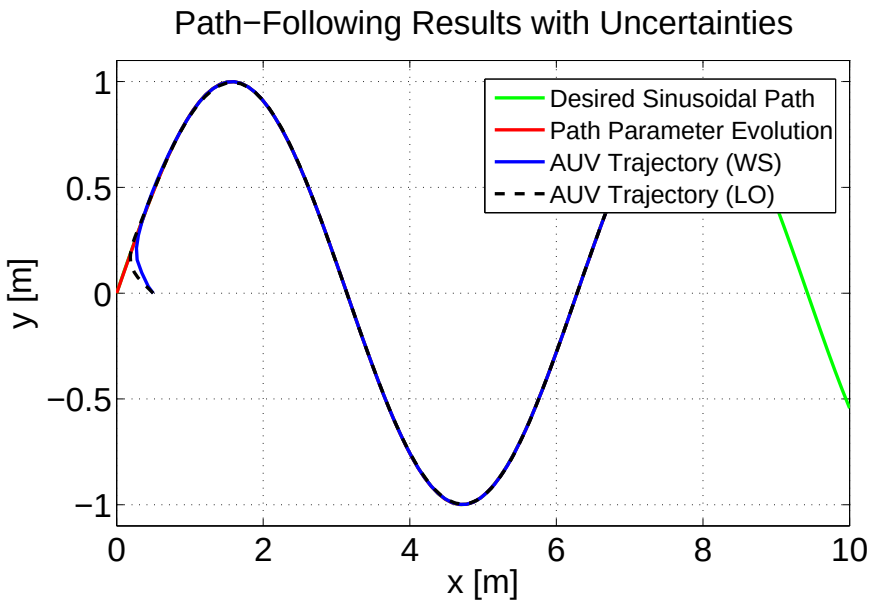

Figure 5.8: The AUV PF results (with parametric uncertainties).

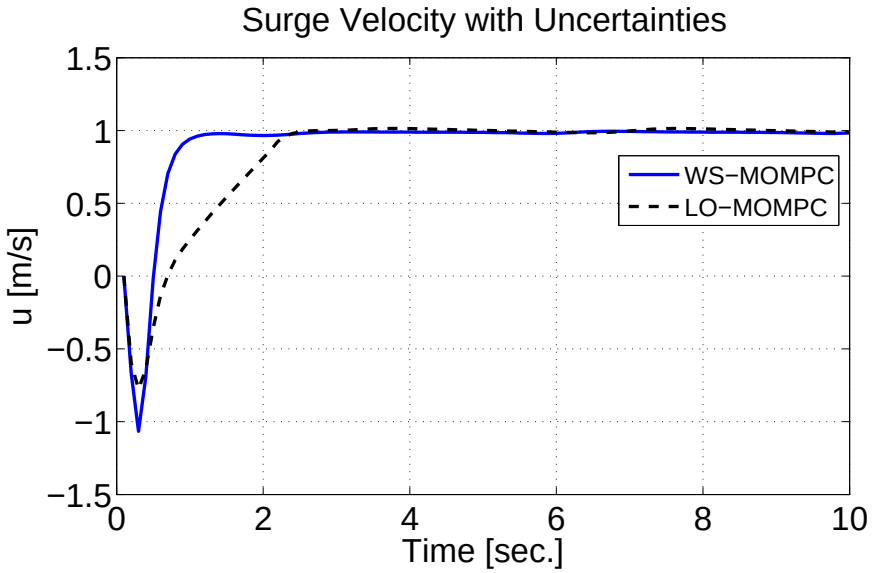

Figure 5.9: Surge velocity of the AUV (with parametric uncertainties).

(5.10). For a non-differentiable desired path such as the following example:

$$p(s) = \begin{cases} p_x(s) = s, \quad p_y(s) = s & s \in [0, 1) \\ p_x(s) = s, \quad p_y(s) = 2 - s & s \in [1, 2) \\ p_x(s) = 4 - s, \quad p_y(s) = 0 & s \in [2, 4) \end{cases}$$

we need to specify the $[p_\psi(s), p_u(s), p_v(s), p_r(s)]$ at the non-differentiable points. In the example, we adopt the following choice:

$$p(s) = \begin{cases} p_\psi(s) = \pi/4, \quad p_u(s) = \sqrt{2}\dot{s}, \quad p_v(s) = 0, \quad p_r(s) = 0 & s = 0 \\ p_\psi(s) = -\pi/4, \quad p_u(s) = \sqrt{2}\dot{s}, \quad p_v(s) = 0, \quad p_r(s) = 0 & s = 1 \\ p_\psi(s) = -\pi/2, \quad p_u(s) = \dot{s}, \quad p_v(s) = 0, \quad p_r(s) = 0 & s = 2 \end{cases}$$

In the simulation, the AUV, subject to 20% model error, is expected to follow the above non-differentiable path; the Speed Assignment to be $u_r = 0.2$ (m/s). As the simulation results (Figure 5.10 - Figure 5.12) indicate, the proposed MOMPC method can well handle the non-differentiable path and the parametric uncertainty. Furthermore, we can observe obvious decrease in surge velocity at these non-differentiable points. The decrease in surge velocity conforms with the objective prioritization we set for the path following control and also largely increases the control stability.

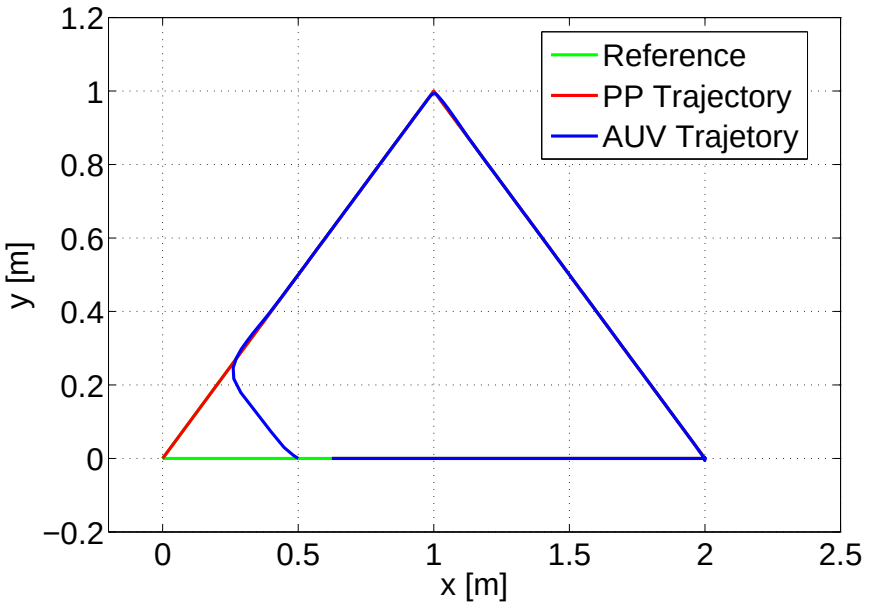

Figure 5.10: PF results with non-differentiable path.

5.6 Conclusion

In this chapter, the path following control problem was studied for the autonomous underwater vehicles. To handle the prioritization between PF requirements, we have

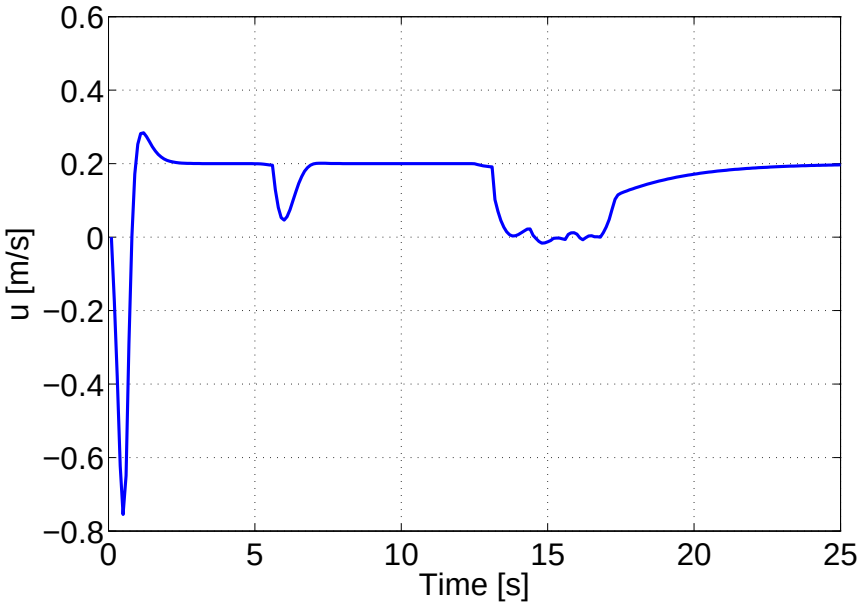

Figure 5.11: Surge velocity of the AUV (non-differentiable).

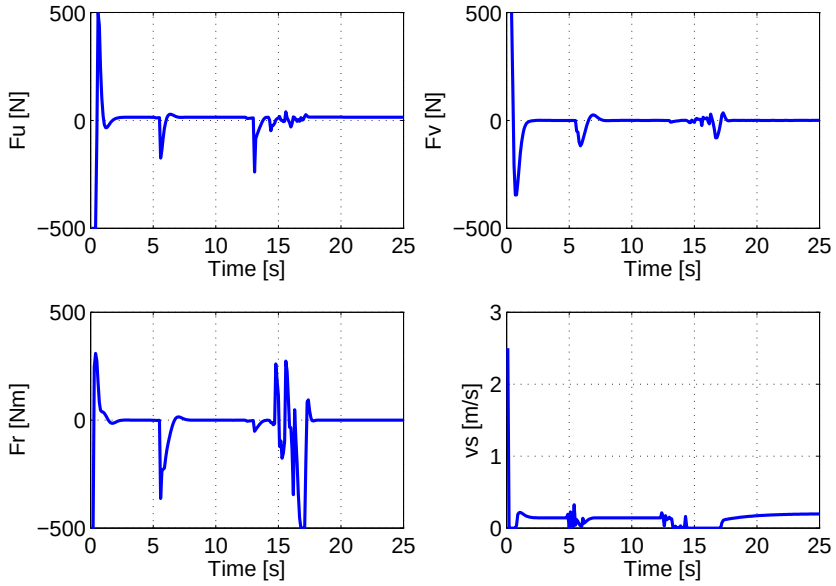

Figure 5.12: Control inputs of the augmented system (non-differentiable).

proposed a novel MOMPC method supported by two implementation algorithms: The weighted sum based and the lexicographic ordering based algorithms. An explicit parametrization of the zero-path-error manifold was constructed facilitating the

MOMPC formulation for the AUV PF control problem. For the WS-MOMPC scheme, a logistic function was proposed to automate the weight selection for the objectives. The Pontryagin minimum principle was applied with several well-defined barrier functions. The sufficient conditions that guarantee the convergence of the closed-loop trajectory were derived. The internal relationship between the WS-MOMPC and the LO-MOMPC were explored. Simulation results on the Falcon dynamic model demonstrated the effectiveness of the proposed method: The objective prioritization could be well incorporated into the PF control. Furthermore, the robustness test results indicated that the MOMPC method had the inherent robustness to some extent, which potentially presented a useful and powerful framework for the AUVs to solve a broad spectrum of motion control problems.

The main results in this chapter have been published in [125].

[125] **C. Shen**, Y. Shi and B. Buckham, "Path-Following Control of an AUV: A Multi-Objective Model Predictive Control Approach", *IEEE Transactions on Control Systems Technology*, accepted, 2017, DOI: 10.1109/TCST.2018.2789440

Chapter 6

Efficient Implementation Algorithms for NMPC Trajectory Tracking Control of an AUV

6.1 Introduction

6.1.1 Research Background and Contributions

In Chapters 3 - 5, we have discussed the model predictive control for the trajectory tracking, dynamic positioning and path following problems for the AUVs. The purpose of the previous chapters is to propose novel and useful frameworks that bring salient features, such as constraint handling and multiple control objective incorporation, in solving these AUV motion control problems. The main focus is hence on the design of the optimization problems such that the closed-loop system is theoretically stable with respect to the equilibrium of interest. The simulation results indicate that the MPC solution is very promising for the AUV applications. However, there may be an issue that potentially prohibits the MPC from being implemented on real AUV systems: the heavy computational burden.

As a feedback control technique, the MPC realizes the feedback mechanism by recursively solving open-loop optimal control problems (OCPs) with real-time state measurement as initial conditions. In theory, the solutions of the OCPs are assumed to be obtained instantly. In practice, they can only be provided by iterative methods. The OCPs need to be solved within strict time constraints. Violation of the time constraints possibly leads to performance degradation or even instability [45]. This

is exactly the reason why the most successful area for MPC is the chemical industry [110] where the sampling period is sufficiently long for standard numerical algorithms. However, the AUV system is with fast dynamics, and the permitted sampling period may be less than tenth of a second. Computation time is no longer negligible. Linear MPC, inheriting the good structure of the quadratic problem, might be efficiently solved by off-the-shelf numerical algorithms, while nonlinear MPC faces an intense conflict between the comparatively long computation time and the short sampling period. Due to nonlinear dynamics of the AUV motion, the OCPs are essentially generic nonlinear programming (NLP) problems. The computational complexity will raise exponentially as the problem size increases. Consequently, different control strategies such as numerical continuation [103], event-triggered control [78], off-line precomputation [22] and distributed implementation [24, 31] have been proposed attempting to shorten the computing time. Although the LMPC framework proposed in Chapter 4 admits the trade-off between computational complexity and control performance, which partially alleviates the heavy burden by circumventing this technical issue, it is desired to develop new tools to explicitly tackle the complexity of the control algorithms.

To move one step further toward real AUV applications, in this chapter, we particularly focus on the computational issue and propose two distinct implementation strategies to reduce the computational complexity of the NMPC-based AUV trajectory tracking control algorithm while maintaining the comparable control performance. The first strategy is based on Ohtsuka's C/GMRES algorithm [103]. The C/GMRES algorithm is appealing since it entertains both merits of the numerical continuation method [8] and the Krylov subspace method [73]. Applying the numerical continuation method, the NLP needs not to be solved. Instead, an approximate solution can be traced from the previous value using the solution of a linear equation. The linear equation is solved by Krylov subspace method which is well-known for its high efficiency: The solution typically converges in a few iterations. However, the original C/GMRES algorithm was proposed for the equality constrained NMPC problems. To incorporate the AUV thrust limits (which are essentially inequality constraints) in the NMPC formulation, we borrow barrier functions from the interior-point method and modify the C/GMRES algorithm. The sufficiency of Pontryagin's minimum principle (PMP) points to solve the Karush-Kuhn-Tucker (KKT) conditions is exploited. As a result, under differentiability and constraint qualifications, the validity of the modification can be claimed. Since the linear independence constraint qualification (LICQ)

is satisfied in the AUV tracking control problem, the proposed modified C/GMRES algorithm provide a viable solution with high efficiency.

The second strategy exploits the dynamic properties of the AUV motion and solves the OCPs in a distributed fashion. To facilitate the distributed implementation of the NMPC tracking control, a reference augmentation is performed. With the reference augmentation the interaction between surge, sway and yaw dynamics is weakened. By exploiting this property two novel distributed implementation algorithms are developed in an attempt to relieve the computation burden. By solving three subproblems with smaller size, the computational complexity drops significantly. The proposed parallel implementation minimizes the computational complexity while the sequential implementation sacrifices some efficiency for the guaranteed stability. Recursive feasibility and closed-loop stability are rigorously proved. The warm start strategy is used to accelerate the convergence.

The main contributions of this chapter are summarized as follows,

- The modified C/GMRES algorithm is proposed to implement the NMPC for the AUV trajectory tracking control with significantly improved efficiency.
- The validity of the incorporation of the barrier functions into the numerical continuation approximations is proved based on the Pontryagin's minimum principle and the KKT conditions.
- Two novel distributed implementation algorithms are developed. The computational complexity of the NMPC tracking control can be dramatically reduced by adopting the developed distributed implementation while the tracking performance can be well maintained.
- The closed-loop properties are explicitly analyzed for the distributed implementation. Sufficient conditions to guarantee the recursive feasibility and the closed-loop stability are provided.

6.1.2 Chapter Organization

The remaining part of this chapter is organized as follows: In Section 6.2 the modified C/GMRES algorithm is presented with technical details. Section 6.3 introduces the novel distributed implementation strategy and elaborates the control algorithm design and stability analysis. In Section 6.4, several conclusive remarks are provided.

In this chapter, the following notations are used: The column operation $[\rho_1^{\mathrm{T}}, ...\rho_n^{\mathrm{T}}]^{\mathrm{T}}$ is denoted as $col(\rho_1, ..., \rho_n)$; the diagonal operation is abbreviated by $\mathrm{diag}(\cdot)$; the square of a weighted Euclidean norm $\rho^{\mathrm{T}} A \rho$ is denoted by $\|\rho\|_A^2$. The $\max\{\cdot\}$ function returns the largest value from the numbers provided in the brace. The absolute value on a vector $|\rho|$ applies the absolute value operation to each element. For a scalar-valued function f, ∇f represents the gradient while F_x denotes the Jacobian of the vector-valued function F with respect to x.

6.2 Modified C/GMRES Algorithm

6.2.1 Problem Formulation

The AUV model studied for the trajectory tracking control is established using the kinematic equations and the dynamic equations:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{R}(\psi)\mathbf{v} \\ \mathbf{M}^{-1}(\mathbf{u} - \mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\boldsymbol{\eta})) \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{6.1}$$

where the state vector $\mathbf{x} = [x, y, \psi, u, v, r]^{\mathrm{T}}$ is consisted of the pose and velocity of the vehicle, and the control vector $\mathbf{u} = [F_u, F_v, F_r]^{\mathrm{T}}$ is generalized thrust forces and moments. The detailed expression can be found in (2.32) and (2.34).

Consider a reference trajectory $p(t) = [x_d(t), y_d(t)]^{\mathrm{T}}$ which defines the desired positions for the vehicle. We assume that the reference trajectory is appropriate by considering physical limits of a vehicle. We view $p(t)$ as the output trajectory of a reference system which owns the same kinematic properties

$$\begin{aligned} \dot{x}_d &= u_d \cos\psi_d - v_d \sin\psi_d \\ \dot{y}_d &= u_d \sin\psi_d + v_d \cos\psi_d \\ \dot{\psi}_d &= r_d \end{aligned} \tag{6.2}$$

Then the kinematic equations in (6.2) can be used as a guideline to augment $p(t)$.

A valid reference system for Falcon is $\mathbf{x}_d = [x_d, y_d, \psi_d, u_d, v_d, r_d]^{\mathrm{T}}$ with

$$\begin{cases} \psi_d = \mathrm{atan2}(\dot{y}_d, \dot{x}_d) \\ u_d = \sqrt{\dot{x}_d^2 + \dot{y}_d^2} \\ v_d = 0 \\ r_d = (\dot{x}_d \ddot{y}_d - \dot{y}_d \ddot{x}_d)/(\dot{x}_d^2 + \dot{y}_d^2) \end{cases} \tag{6.3}$$

where atan2 is the four-quadrant inverse tangent function. In this way, potential numerical difficulties associated with positive semi-definite weighting matrices induced singularity or rank deficiency (when the tracking control is formulated as an output MPC problem) can be avoided. This is important because in the following we will take advantage of the numerical continuation to approximate the optimal control signal, in which the regularity of the solution, or equivalently, local full rank of the Hessian of the Hamiltonian is required by implicit function theorem [8]. Now we define the optimal control problem (P_0) that needs to be solved online:

$$\begin{aligned} (\mathrm{P}_0): \quad \min_{\hat{\mathbf{u}}(s,t_0)} . \quad J = \; & \int_0^T (\|\hat{\mathbf{x}}(s,t_0) - \mathbf{x}_d(t_0+s)\|_Q^2 \\ & + \|\hat{\mathbf{u}}(s,t_0)\|_R^2) ds + \|\hat{\mathbf{x}}(T,t_0) - \mathbf{x}_d(t_0+T)\|_{Q_f}^2 \\ \text{s.t.} \quad & \dot{\hat{\mathbf{x}}}(s,t_0) = \mathbf{f}(\hat{\mathbf{x}}(s,t_0), \hat{\mathbf{u}}(s,t_0)) \\ & \hat{\mathbf{x}}(0,t_0) = \mathbf{x}(t_0) \\ & |\hat{\mathbf{u}}(s,t_0)| \le \mathbf{u}_{\max} \end{aligned} \tag{6.4}$$

where $\hat{\mathbf{x}}(s,t_0)$ denotes the predicted state evolution starting from $\mathbf{x}(t_0)$ at the origin of the fictitious time axis $s = 0$. The optimal control signal $\hat{\mathbf{u}}^*(s,t_0)$ is determined by solving the above optimization. T is the prediction horizon, and Q_f, Q, R are corresponding weighting matrices, positive definite.

The NMPC algorithm for the AUV tracking can be briefly described as follows:

- At sampling instant t_0, the optimization (P_0) is performed given the current system state $\mathbf{x}(t_0)$, and let $\hat{\mathbf{u}}^*(s,t_0)$ denote the solution.
- The AUV uses $\hat{\mathbf{u}}^*(s,t_0)$ for only one sampling period: $\mathbf{u}(t) = \hat{\mathbf{u}}^*(s,t_0)$ for $s \in [0, \Delta t]$.
- At next sampling instant $t_0 + \Delta t$, new measurement of system state $\mathbf{x}(t_0 + \Delta t)$ is fed back, then (P_0) is solved again substituting t_0 with $t_0 + \Delta t$.

The above procedure will repeat until accomplishing the tracking task.

We notice that in (P_0) the objective function J includes an integral operation and the constraints include the control system dynamics which are essentially derivative operations. However, in real-world implementation, both the integral and the derivative have to be performed numerically. Therefore, we solve the discretized version of (P_0) at each sampling instant.

For the convenience of the description of C/GMRES algorithm, let us slightly abuse the notations. In the following, u represents the control input of a general nonlinear dynamical system $\dot{x} = f(x, u, p)$ rather than AUV surge velocity; x represents the state rather than AUV position; and p is the time-varying parameter. We generalize (P_0), and the discretized version (P) using Euler forward approximation is defined as follows:

$$\begin{aligned} (\mathrm{P}): \quad \min_{u_i} . \quad J = \quad & \textstyle\sum_{i=0}^{N-1} \ell(x_i, u_i, p_i)\Delta t + g(x_N, p_N) \\ \text{s.t.} \quad & x_{i+1} = x_i + f(x_i, u_i, p_i)\Delta t \\ & x_0 = x(t_0) \\ & c(x_i, u_i, p_i) = 0 \\ & h(x_i, u_i, p_i) \leq 0 \end{aligned} \tag{6.5}$$

Here, we divide the prediction horizon T into N steps with step size Δt (which is known as the sampling period). The continuous-time system dynamics $\dot{x} = f(x, u, p)$ is also discretized, and the initial condition is given by the current system state $x(t_0)$. $c(x_i, u_i, p_i)$, $h(x_i, u_i, p_i)$ represent vector-valued equality and inequality constraints, respectively.

6.2.2 Solving the NMPC Problem

Generally, solving the optimization problem (P) is a generic nonlinear programming (NLP) problem. Under differentiability and constraint qualifications, the Karush-Kuhn-Tucker (KKT) theorem can be applied. The KKT theorem states the first-order necessary conditions for a solution $\{u_i^*\}_{i=0}^{N-1}$ of (P) to be optimal [9]. Since the system dynamics and initial condition of (P) are essentially equality constraints, we combine them with rest of the equalities and denote them by $\bar{c}(x_i, u_i, p_i) = 0$. We simplify the notation of (P) as follows:

$$\min_{\bar{U}} . \quad J(\bar{U}) \quad \text{s.t.} \quad \bar{c}(\bar{U}) = 0, \ h(\bar{U}) \leq 0$$

where $\bar{U} = col(u_0, u_1, ..., u_{N-1})$. The KKT theorem states: For a locally optimal solution $\bar{U}^*$, there exist vector-valued multipliers λ^* and μ^* such that the following conditions hold

$$\begin{aligned} &\nabla J(\bar{U}^*) + \textstyle\sum_i \lambda_i^* \nabla \bar{c}_i(\bar{U}^*) + \sum_j \mu_j^* \nabla h_j(\bar{U}^*) = 0 \\ &\bar{c}_i(\bar{U}^*) = 0 \\ &h_j(\bar{U}^*) \leq 0 \\ &\mu_j^* \geq 0 \\ &\mu_j^* h_j(\bar{U}^*) = 0 \end{aligned} \tag{6.6}$$

To solve (P) is actually to solve the KKT system (6.6) using Newton's methods such as Sequential Quadratic Programming (SQP) methods and Interior-Point (IP) methods [9]. However, conventional Newton type numerical algorithms basically rely on successive linearizations, essentially making it computationally expensive in solving the problem. To alleviate the computational burden, many efficient approximate algorithms for MPC have been developed [35]. For the AUV tracking control, Ohtsuka's C/GMRES algorithm [103] is tailored and modified to incorporate the physical limits of the thrusters.

6.2.3 Modified C/GMRES Algorithm

Instead of solving the generic NMPC problem (P), the original C/GMRES Algorithm considers the less general problem (P_1) in which no inequality constraints exist

$$\begin{aligned} (\mathrm{P}_1): \quad \min_{u_i} . \quad J = \quad & \textstyle\sum_{i=0}^{N-1} \ell(x_i, u_i, p_i)\Delta t + g(x_N, p_N) \\ \text{s.t.} \quad & x_{i+1} = x_i + f(x_i, u_i, p_i)\Delta t \\ & x_0 = x(t_0) \\ & c(x_i, u_i, p_i) = 0 \end{aligned}$$

The C/GMRES is based on solving the discretized necessary condition for the optimal control, i.e., the Pontryagin's Minimum Principle (PMP) [103]. Define the Hamiltonian by

$$H(x, \bar{\lambda}, u, \nu, p) = \ell(x, u, p) + \bar{\lambda}^{\mathrm{T}} f(x, u, p) + \nu^{\mathrm{T}} c(x, u, p)$$

Here, we call $\bar{\lambda}$ the costate, and ν the Language multiplier. The discretized PMP claims that for a local optimal control $\{u_i^*\}_{i=0}^{N-1}$, there exist $\{\bar{\lambda}_i^*\}_{i=0}^{N}$ and $\{\nu_i^*\}_{i=0}^{N-1}$

satisfying the following conditions:

$$
\begin{aligned}
x_{i+1}^* &= x_i^* + f(x_i^*, u_i^*, p_i)\Delta t && \text{(6.7a)}\\
\bar{\lambda}_i^* &= \bar{\lambda}_{i+1}^* + H_x^{\mathrm{T}}(x_i^*, \bar{\lambda}_{i+1}^*, u_i^*, \nu_i^*, p_i)\Delta t && \text{(6.7b)}\\
\bar{\lambda}_N^* &= g_x^{\mathrm{T}}(x_N^*, p_N) && \text{(6.7c)}\\
x_0^* &= x(t_0) && \text{(6.7d)}\\
H_u(x_i^*, \bar{\lambda}_{i+1}^*, u_i^*, \nu_i^*, p_i) &= 0 && \text{(6.7e)}\\
c(x_i^*, u_i^*, p_i) &= 0 && \text{(6.7f)}
\end{aligned}
$$

Observe that (6.7a)-(6.7b) are recurrence relation on x^* and $\bar{\lambda}^*$, and (6.7c)-(6.7d) can be viewed as boundary conditions. Define $U = col(u_0^*, \nu_0^*, \ldots, u_{N-1}^*, \nu_{N-1}^*)$, then the state trajectory $\{x_i^*\}_{i=0}^N$ and the costate trajectory $\{\bar{\lambda}_i^*\}_{i=0}^N$ are explicit functions of U. We substitute (6.7a)-(6.7d) into (6.7e)-(6.7f) and construct the equation system

$$
F(U,x) = \begin{bmatrix}
H_u^{\mathrm{T}}(x_0^*, \bar{\lambda}_1^*, u_0^*, \nu_0^*, p_0)\\
c(x_0^*, u_0^*, p_0)\\
\vdots\\
H_u^{\mathrm{T}}(x_{N-1}^*, \bar{\lambda}_N^*, u_{N-1}^*, \nu_{N-1}^*, p_{N-1})\\
c(x_{N-1}^*, u_{N-1}^*, p_{N-1})
\end{bmatrix} = 0 \qquad \text{(6.8)}
$$

Normally, we need to solve (6.8) at each sampling instant. However, solving (6.8) using iterative methods is computationally expensive in the sense of evaluating Jacobians, Hessian and inverses. Instead, C/GMRES takes the advantage of numerical continuation method [8] that views the equation system $F(U,x)$ as a dynamical system governed by

$$
\dot{F}(U,x,t) = A_s F(U,x,t) \qquad \text{(6.9)}
$$

Here, we explicitly point out the time dependence of the dynamical system (6.9) by denoting the state as $F(U,x,t)$ rather than solely $F(U,x)$. A_s is an introduced stable matrix to stabilize $F(U,x,t)$ at the origin. Hence, if F_U is nonsingular, Eqn. (6.9) is equivalent to

$$
\dot{U} = F_U^{-1}(A_s F - F_x \dot{x} - F_t) \qquad \text{(6.10)}
$$

The derivative of U can be computed by (6.10) using the derivative of x. Then, if an initial solution $U(0)$ satisfying $F(U(0), x(0), 0) = 0$ can be found, the solution curve $U(t)$ of (6.8) can be traced by integrating (6.10) from $U(0)$.

To further relieve the computational load, the action of Jacobians on vectors is approximated by forward difference:

$$\begin{aligned} & F_U(U,x,t)W + F_x(U,x,t)w + F_t(U,x,t)\omega \\ & \approx \delta^{-1}(F(U+\delta W, x+\delta w, t+\delta\omega) - F(U,x,t)) \\ & = D_\delta F(U,x,t:W,w,\omega) \end{aligned}$$

where δ is a small positive number. To further avoid the expensive operation of matrix inverse F_U^{-1}, Generalized Minimal Residual method (GMRES) is applied to solve the linear equation $F_U\dot{U} = AsF - F_x\dot{x} - F_t$. The combination of forward difference approximation and GMRES is called FDGMRES introduced in [73]. To simplify the description, we can simply view FDGMRES as a function

$$\dot{U} = \text{FDGMRES}(U, x, \dot{x}, t, \hat{\dot{U}}, \delta, k_{\max})$$

where $\hat{\dot{U}}$ is an initial guess and $k_{\max}$ is the allowed maximum iteration number. Defining a transformation $T_0(U) = u_0^*$, the C/GMRES algorithm can be depicted in **Algorithm 5**.

Algorithm 5 : C/GMRES Algorithm

1: Initialize $t = 0$, $k = 0$, initial state $x_0 = x(0)$ and find U_0 numerically such that $F(U_0, x_0, 0) = 0$.
2: For $s \in [t, t + \Delta t)$, the real control input is computed by $u(s) = T_0(U_k)$.
3: At next sampling instant $t + \Delta t$, the system state $x_{k+1} = x(t + \Delta t)$ is fed back. Compute the state difference $\Delta x_k = x_{k+1} - x_k$.
4: $\dot{U}_k$=FDGMRES$(U_k, x_k, \Delta x_k/\Delta t, t, \hat{\dot{U}}_k, \delta, k_{\max})$
5: Set $U_{k+1} = U_k + \dot{U}_k \Delta t$
6: Update $t = t + \Delta t$, $k = k + 1$
7: Go to Step 2.

Since several approximations are made in the C/GMRES algorithm, the approximation error should be bounded. An error analysis can be found in [103].

The AUV tracking problem (P_0), however, is constrained by inequality constraints in terms of actuator limits. We modify the C/GMRES algorithm appropriately so that it can incorporate the inequality constraints. By introducing the barrier functions [19], we approximately solve (P) by actually solving another optimization ($\bar{P}$) defined

as follows:

$$(\bar{\mathrm{P}}): \quad \min_{\bar{U}} . \quad J(\bar{U}) - \gamma_l \sum_j \log(-h_j(\bar{U})) \quad \text{s.t.} \quad \bar{c}(\bar{U}) = 0 \tag{6.11}$$

where γ_l is a positive number. Since the C/GMRES is based on solving PMP system rather than KKT system, in the following, we first explore the sufficiency of PMP points to satisfy the KKT conditions, then exploit the convergence of the solution of $(\bar{\mathrm{P}})$ to that of (P).

Lemma 2. *A point that satisfies the PMP of* (P_1) *sufficiently solves the corresponding KKT conditions.*

Proof. The detailed KKT conditions of (P_1) are derived in the following:

$$\nabla \bar{J} = 0 \tag{6.12a}$$

$$\bar{c}(x_i, u_i, p_i) = 0 \tag{6.12b}$$

with

$$\begin{aligned} \bar{J} = \; & \textstyle\sum_{i=0}^{N-1} L(x_i, u_i, p_i) + g(x_N, p_N) + \sum_{i=0}^{N-1} \bar{\lambda}_{j+1}^{\mathrm{T}} (f_d(x_j, u_j, p_j) \\ & - x_{j+1}) + \tilde{\lambda}_0^{\mathrm{T}} (x(t_0) - x_0) + \textstyle\sum_{k=0}^{N-1} \nu_k^{\mathrm{T}} c(x_k, u_k, p_k) \Delta t \end{aligned}$$

where $L(x_i, u_i, p_i) = \ell(x_i, u_i, p_i)\Delta t$ and $f_d(x_i, u_i, p_i) = x_i + f(x_i, u_i, p_i)\Delta t$. Define Hamiltonian sequence by

$$H^i = L(x_i, u_i, p_i) + \bar{\lambda}_{i+1}^{\mathrm{T}} f_d(x_i, u_i, p_i) + \nu_i^{\mathrm{T}} c(x_i, u_i, p_i) \Delta t$$

Then we can simplify the notation of $\bar{J}$

$$\bar{J} = g(x_N, p_N) - \bar{\lambda}_N^{\mathrm{T}} x_N + \textstyle\sum_{i=1}^{N-1} (H^i - \bar{\lambda}_i^{\mathrm{T}} x_i) + H^0 + \tilde{\lambda}_0^{\mathrm{T}} (x(t_0) - x_0)$$

Since given x_0, the system state x_i is merely dependent on $\bar{U}$, the gradient

$$\nabla \bar{J} = \frac{\partial \bar{J}}{\partial X} \frac{dX}{d\bar{U}} + \frac{\partial \bar{J}}{\partial \bar{U}}$$

where $X = col(x_0, ...x_N)$, then $\nabla \bar{J} = 0$ is equivalent to

$$d\bar{J} = \frac{\partial \bar{J}}{\partial X} dX + \frac{\partial \bar{J}}{\partial \bar{U}} d\bar{U} = 0$$

By explicitly computing $d\bar{J}$, we have

$$\begin{aligned} d\bar{J} = \; & (\tfrac{\partial g}{\partial x_N} - \bar{\lambda}_N^{\mathrm{T}})dx_N + \textstyle\sum_{i=1}^{N-1}\{(\tfrac{\partial H^i}{\partial x_i} - \bar{\lambda}_i^{\mathrm{T}})dx_i \\ & + \tfrac{\partial H^i}{\partial u_i}du_i\} + \tfrac{\partial H^0}{\partial x_0}dx_0 + \tfrac{\partial H^0}{\partial u_0}du_0 - \tilde{\lambda}_0^{\mathrm{T}}dx_0 = 0 \end{aligned} \tag{6.13}$$

If we select the costate sequence satisfying

$$\bar{\lambda}_i^{\mathrm{T}} - \frac{\partial H^i}{\partial x_i} = 0 \tag{6.14a}$$

$$\frac{\partial g}{\partial x_N} - \bar{\lambda}_N^{\mathrm{T}} = 0 \tag{6.14b}$$

and $x_0 = x(t_0)$ is given, then $d\bar{J} = 0$ is equivalent to

$$d\bar{J} = \sum_{i=0}^{N-1} \frac{\partial H^i}{\partial u_i}du_i = 0 \tag{6.15}$$

We notice that (6.14a)-(6.14b) are identical to (6.7b)-(6.7c) and (6.7e) sufficiently solves (6.15). Eqn. (6.12b) states the rest of PMP conditions. Now, the sufficiency of PMP points to solve the KKT conditions has been shown. □

Lemma 3. *Suppose differentiability and constraint qualifications hold. Let a sequence* $\{\gamma_l\}$ *satisfy* $0 < \gamma_{l+1} < \gamma_l$, *and* $\gamma_l \to 0$ *as* $l \to \infty$. *Suppose the solution of* (P) *and* $(\bar{P})$ *exist and denoted by* $\bar{U}^*$ *and* $\bar{U}^l$, *then* $\bar{U}^l$ *converge to* $\bar{U}^*$ *as* $\gamma_l \to 0$.

Proof. See [19] ch. 11 sec. 3.3. □

Definition 4 (Linear Independence Constraint Qualification). *Let* $\bar{U}$ *be feasible for* (P) *and* $I(\bar{U}) = \{j | h_j(\bar{U}) = 0\}$, *then we say that the linear independence constraint qualification (LICQ) holds at* $\bar{U}$, *if for* $j \in I(\bar{U})$

$$\nabla h_j(\bar{U}), \nabla \bar{c}_i(\bar{U})$$

are linearly independent. Then strong duality gap holds for (P) *and KKT theorem is applicable.*

Theorem 9. *For the AUV tracking problem (6.4), the modified C/GMRES with log barriers approximately solves* (P_0) *with a small positive* γ_l.

Proof. Since the system dynamics $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ imposes constraints on $\mathbf{x}$ not on $\mathbf{u}$, the only constraints on $\mathbf{u}$ are $|\mathbf{u}(t)| \leq \mathbf{u}_{\max}$. For the discretized version, the

constraints are $|\mathbf{u}_i| \leq \mathbf{u}_{\max}$, or equivalently, $\mathbf{u}_i \leq \mathbf{u}_{\max}$ and $\mathbf{u}_i \geq -\mathbf{u}_{\max}$, which obviously satisfy LICQ at all feasible points. Therefore, **Theorem 9** is followed by **Lemma 2** and **Lemma 3**. □

The modified C/GMRES algorithm can be depicted in **Algorithm 6**.

Algorithm 6 : mC/GMRES Algorithm

1: Initialize $t = 0$, $k = 0$ and $l = 0$. Given initial state $x_0 = x(0)$, γ_0, α and $l_{\max}$, find U_0^l numerically such that $F(U_0^l, x_0, \gamma_l, 0) = 0$.
2: Set $l = l + 1$. If $l < l_{\max}$, go to Step 3. Otherwise, go to Step 4.
3: Set $\gamma_l = \alpha\gamma_{l-1}$. Then find U_0^l numerically such that $F(U_0^l, x_0, \gamma_l, 0) = 0$ with initial guess $\hat{U}_0^l = U_0^{l-1}$. Go to Step 2.
4: For $s \in [t, t + \Delta t)$, the real control input is computed by $u(s) = T_0(U_k)$.
5: At next sampling instant $t + \Delta t$, the system state $x_{k+1} = x(t + \Delta t)$ is fed back. Compute the state difference $\Delta x_k = x_{k+1} - x_k$.
6: $\dot{U}_k$=FDGMRES$(U_k, x_k, \Delta x_k/\Delta t, t, \hat{\dot{U}}_k, \delta, k_{\max})$
7: Set $U_{k+1} = U_k + \dot{U}_k \Delta t$
8: Update $t = t + \Delta t$, $k = k + 1$
9: Go to Step 4.

Note that the conventional C/GMRES algorithm only solves the PMP system (6.7) once at the initialization, and the modified version changes the initialization by solving $F(U_0, x_0, \gamma_l, 0) = 0$ several times with a shrinking γ_l and initial guesses $\hat{U}_0^{l+1} = U_0^l$ in order to obtain good approximations while circumventing numerical difficulties [19]. Therefore, the modification preserves the efficiency of the fast algorithm.

6.2.4 Simulation Results

In this section, the AUV tracking control with three different reference trajectories are simulated. We demonstrate the effectiveness and highlight the efficiency of the modified C/GMRES by comparisons to other numerical algorithms. All the simulations are performed on a personal computer (CPU: Intel(R) Core(TM) i7-3520M: 2.90GHz 2.90GHz; RAM: 4.00GB).

The first reference trajectory (Case I) to be tested is a circle defined as follows,

$$p(t) = \begin{cases} x_d = 0.8\cos(0.5t) \\ y_d = 0.8\sin(0.5t) \end{cases} \tag{6.16}$$

The circle trajectory is a typical test trajectory for tracking control problems. It is

relative easier to track because the reference angular and tangential velocities keep constant values.

The second test reference trajectory (Case II) is a sinusoidal curve defined as follows,

$$p(t) = \begin{cases} x_d = 0.5t \\ y_d = \sin(0.5t) \end{cases} \tag{6.17}$$

The sinusoidal trajectory is another typical test trajectory for tracking control whose reference tangential and angular velocities are no longer constant.

The third test reference trajectory (Case III) is an eight-shaped trajectory defined as follows,

$$p(t) = \begin{cases} x_d = \sin(0.5t) \\ y_d = \sin(0.25t) \end{cases} \tag{6.18}$$

The eight-shaped trajectory has sharp changes in its reference velocities, which makes the tracking task more challenging.

We notice that in the augmented reference system (6.3), the first and second derivatives of $p(t)$ are needed. It could be tedious to analytically calculate these functions. Instead, in the implementation, we approximate the discrete values by forward differences:

$$\begin{aligned} \dot{p}(n\Delta t) &\approx \frac{p((n+1)\Delta t) - p(n\Delta t)}{\Delta t} \\ \ddot{p}(n\Delta t) &\approx \frac{p((n+2)\Delta t) - 2p((n+1)\Delta t) + p(n\Delta t)}{(\Delta t)^2} \end{aligned}$$

Note that the above process will introduce some approximation errors, so there exists a trade-off between computational complexity and solution accuracy.

The AUV system parameters are from the identified dynamic model of Falcon [109]. The model parameter details are summarized in Table 2.1. Thrust limits $F_{u,\max} = 500$ (N), $F_{v,\max} = 500$ (N) and $F_{r,\max} = 500$ (Nm) and initial conditions $\mathbf{x}(0) = [0.3, 0, 0, 0, 0, 0]^{\mathrm{T}}$. For the NMPC parameters, sampling period $\Delta t = 0.1sec$, weighting matrices $Q = \mathrm{diag}(10^4, 10^4,\ 10, 10, 1, 10)$, $R = \mathrm{diag}(10^{-4}, 10^{-4}, 10^{-2})$ and $Q_f = \mathrm{diag}(10, 10, 1, 1, 1, 1)$. For the modified C/GMRES algorithm, $A_s = -\zeta I$ with $\zeta = 1/\Delta t$, $k_{\max} = 4$, $\delta = 10^{-5}$. In the initialization $\gamma_0 = 10^2$ and iteratively solve $F(U_0, x_0, \gamma_l, 0) = 0$ with $\gamma_{l+1} = \alpha\gamma_l = \gamma_l/10$ and $\hat{U}_0^{l+1} = U_0^l$ until $l = l_{\max} = 6$.

In all of the cases, at each sampling instant, the optimal control signal of the NMPC tracking is computed by (i) the proposed modified C/GMRES algorithm (red

curves), (ii) the Trust-Region-Dogleg (TRD) algorithm [147] to solve $F(U, x, t) = 0$ (black curves), and (iii) the embedded SQP [9] of Matlab function $fmincon$ to directly solve the KKT system of (P_0) (blue curves), respectively. The green curves are the reference trajectories $p(t)$.

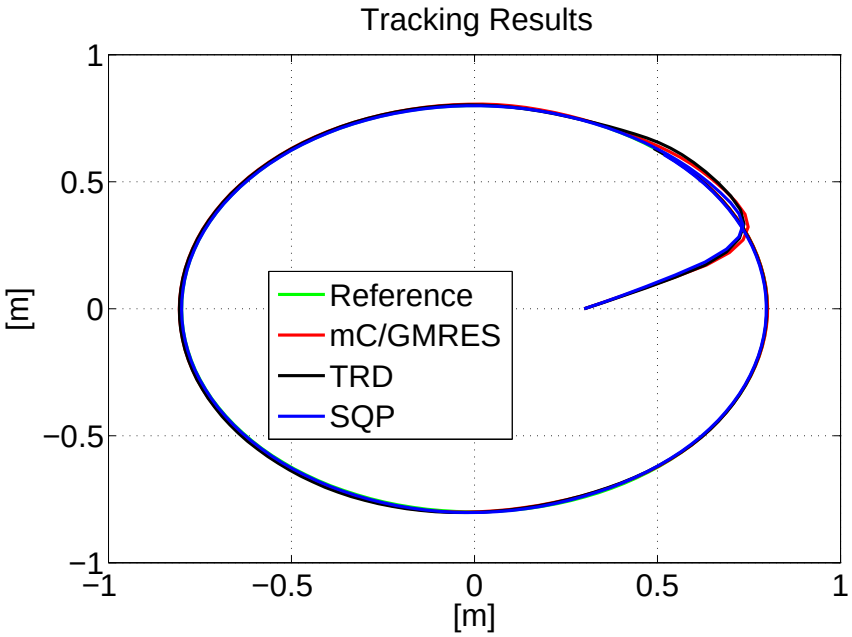

Figure 6.1: Simulated AUV trajectories - Case I.

The NMPC tracking control results of a circle are shown in Figure 6.1. The red, black and blue curves are simulated AUV trajectories. The control forces and moments are recorded in Figure 6.2, and the position errors are plotted in Figure 6.3 with respect to time. Generally speaking, the AUV tracking control are comparably well with different implementations. Position errors are bounded. Also, as expected, the control signals are within their intervals of permitted values, which validates the usefulness of barrier functions.

The simulated AUV trajectories to track a sinusoidal reference are illustrated in Figure 6.4. The control forces and moments are plotted in Figure 6.5, and the position errors are recorded in Figure 6.6. The same as Case I, we can observe that the AUV tracking performance are comparably well with three algorithms, and the control signals never go beyond their limits.

The simulation results of tracking the eight-shape reference trajectory are demonstrated in Figure 6.7. The control signals are shown in Figure 6.8, and the position errors are plotted in Figure 6.9. Similar observations can be made that the NMPC

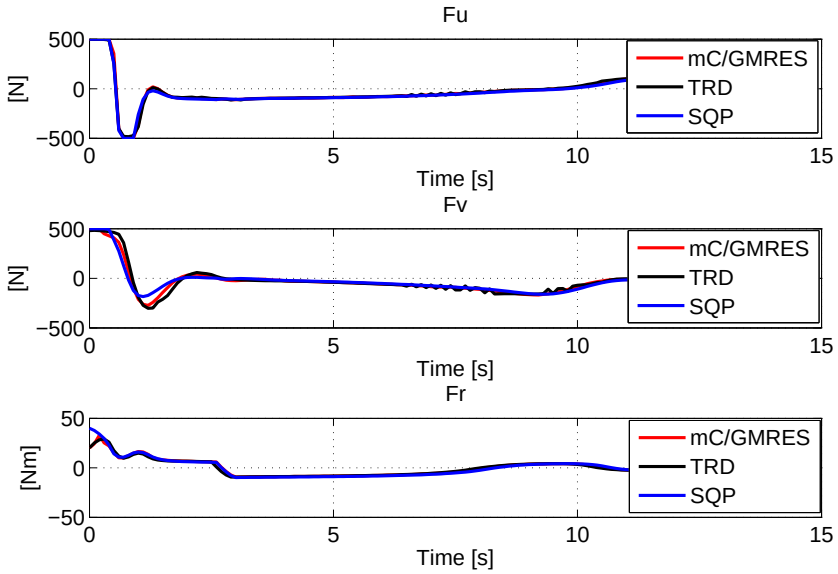

Figure 6.2: The control forces and moments - Case I.

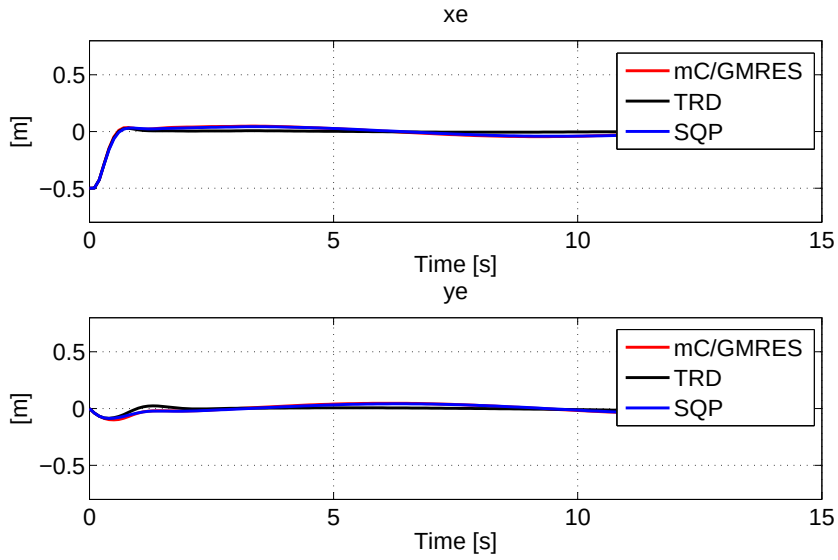

Figure 6.3: The position error - Case I.

tracking control seem comparably well using different numerical algorithms. Position errors are bounded, and the control signals never transcend the boundary.

Although the tracking control performance are comparably acceptable with different implementation, if we check the average computation time, as listed in Table 6.1 - 6.3, we can easily distinguish the efficiencies of these algorithms. Remember

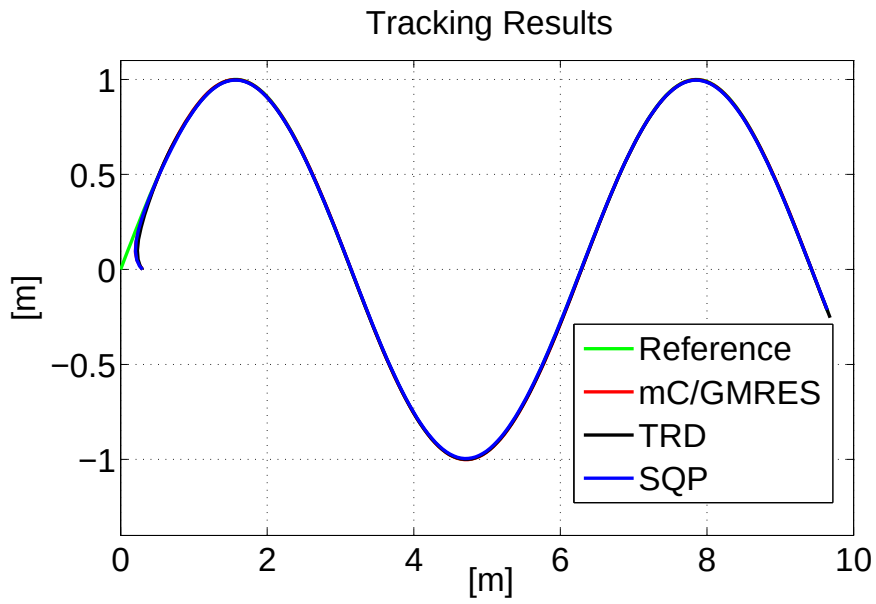

Figure 6.4: Simulated AUV trajectories - Case II.

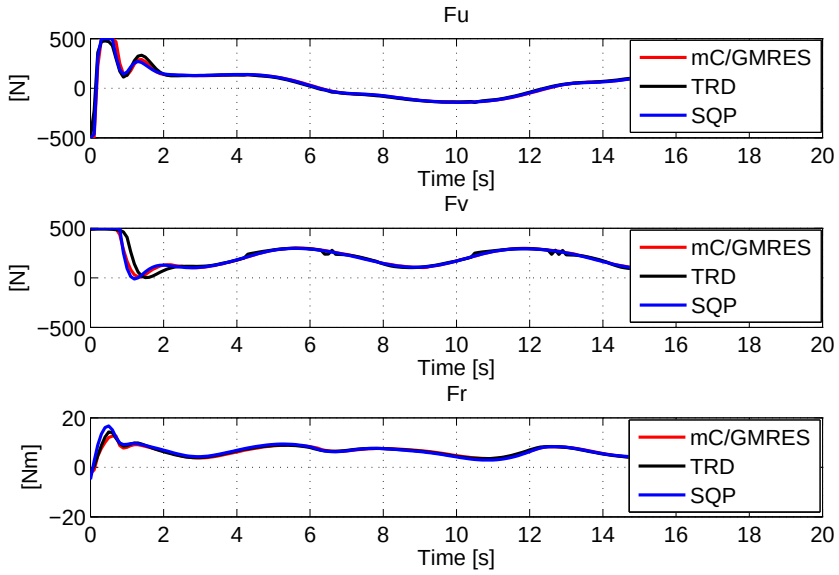

Figure 6.5: The control forces and moments - Case II.

that, the sampling period $\Delta t = 0.1sec$, which means at each sampling instant the optimal control signals have to be obtained within $0.1sec$. Obviously, the embedded SQP using conventional Newton failed to solve (6.6) within the time constraint, and TRD only marginally satisfied the time constraint with small prediction horizons. In

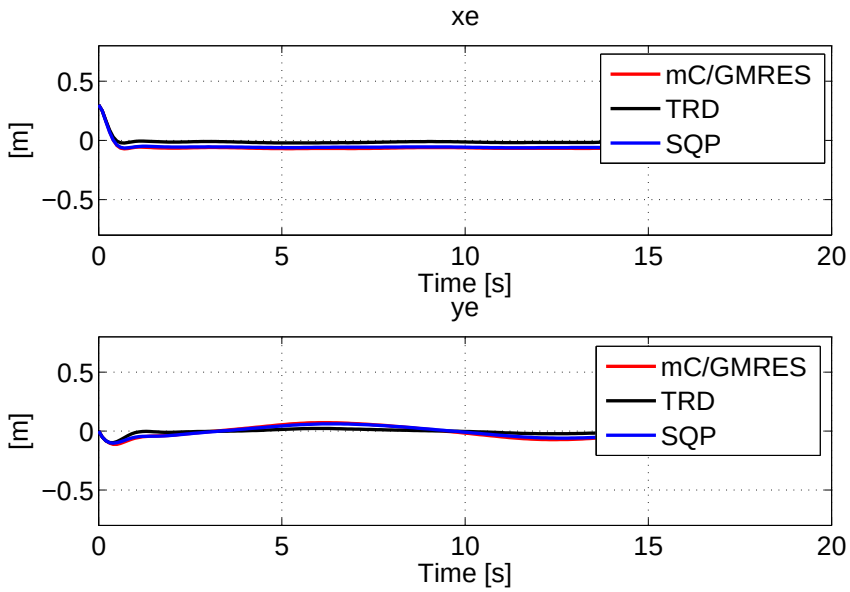

Figure 6.6: The position error - Case II.

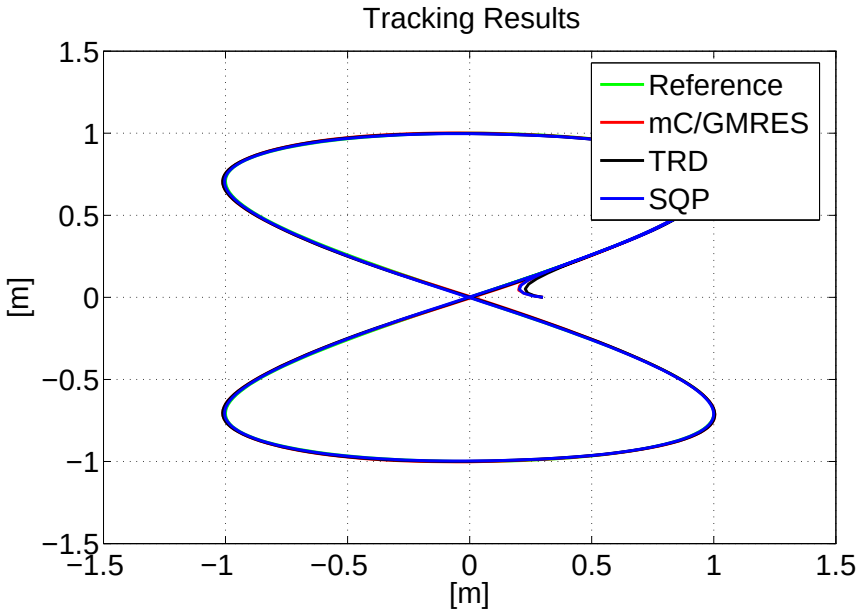

Figure 6.7: Simulated AUV trajectories - Case III.

contrast, modified C/GMRES efficiently solved the NMPC problem within 10% of Δt, which is very promising for real-time implementation of the NMPC-based AUV tracking control.

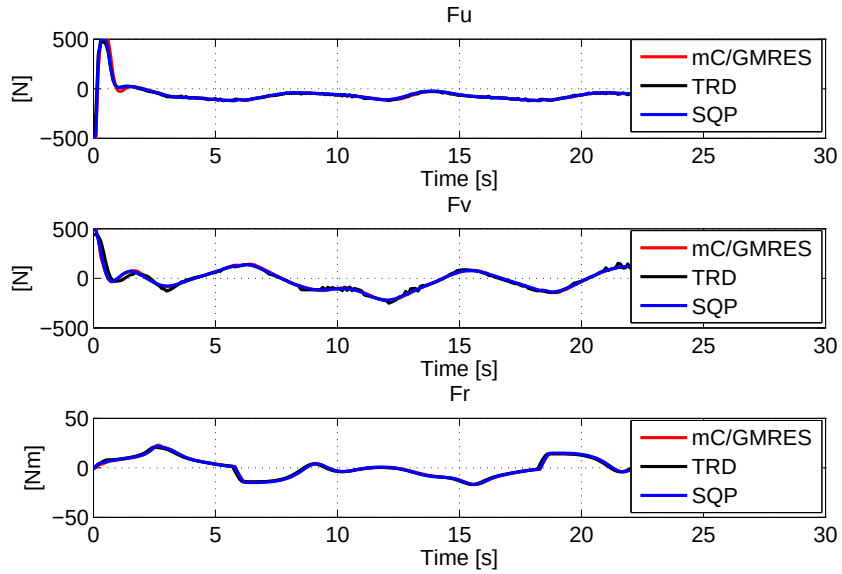

Figure 6.8: The control forces and moments - Case III.

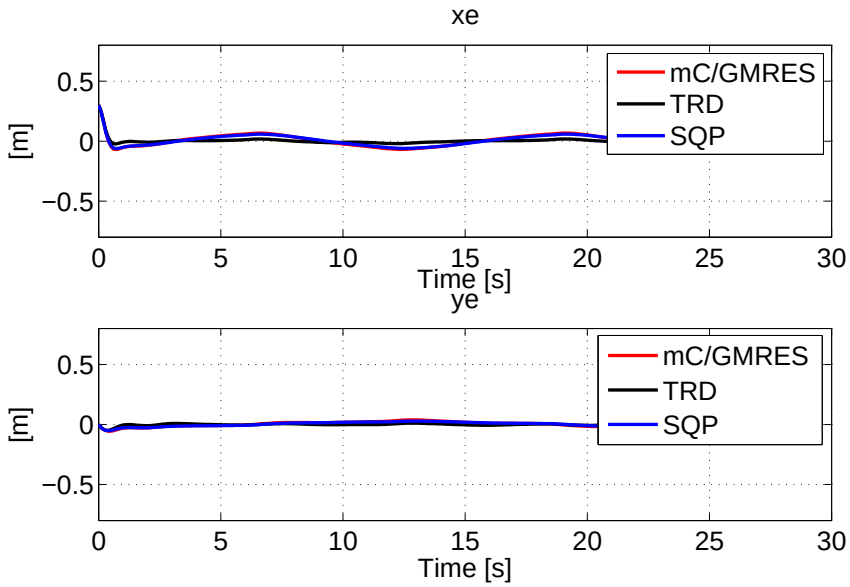

Figure 6.9: The position error - Case III.

6.3 Distributed Implementation Strategy

In the previous section, we have witnessed the dramatically improved numerical efficiency of the proposed modified C/GMRES algorithm. Due to the numerical continuation step involved in the algorithm, the optimal solution at each sampling time

Table 6.1: Average computation time (*sec.*) per update - Case I.

N	mC/GMRES	Trust-Region-Dogleg	SQP
5	0.0038	0.0274	0.1741
10	0.0058	0.0564	0.5881
20	0.0098	0.2168	2.0969

Table 6.2: Average computation time (*sec.*) per update - Case II.

N	mC/GMRES	Trust-Region-Dogleg	SQP
5	0.0033	0.0276	0.2395
10	0.0055	0.0927	0.5523
20	0.0097	0.2507	1.8901

Table 6.3: Average computation time (*sec.*) per update - Case III.

N	mC/GMRES	Trust-Region-Dogleg	SQP
5	0.0040	0.0309	0.1990
10	0.0064	0.0605	0.6167
20	0.0100	0.2218	2.2646

instant can be traced from the previous solution without truly solving the NLP problem. On one side, this can be extremely efficient in obtaining the optimal control signal. On the flip side, however, the approximation error will accumulate over time. The large approximation error may lead to highly degraded control performance or even instability. Therefore, in this section, we are going to take another fast implementation strategy which reduces the overall computational complexity but preserves the closed-loop stability.

6.3.1 Problem Formulation

The AUV model considered in this section is established based on the kinematic equations, the dynamic equations and the thrust distribution:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{R}(\psi)\mathbf{v} \\ \mathbf{M}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{g}(\boldsymbol{\eta})) \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{6.19}$$

where the state vector $\mathbf{x} = [x, y, \psi, u, v, r]^{\mathrm{T}}$ is consisted of the pose and velocity of the vehicle, and the control vector $\mathbf{u} = [u_1, u_2, u_3, u_4]^{\mathrm{T}}$ is consisted of the forces generated by the four thrusters. The detailed expression can be found in (2.32) and (2.34).

The trajectory of interest is defined in the output space: $p(t) = [x_d(t), y_d(t)]^{\mathrm{T}}$

which describes the desired position in the IRF. To avoid singularities, we assume that **Assumption 3** can hold. Then taking advantage of **Proposition 2**, we can safely formulate the NMPC tracking control problem (P_0) as follows,

$$\min_{\mathbf{u}\in S(\delta)} J = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + \|\mathbf{u}(s)\|_R^2)ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{6.20a}$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(s) = \mathbf{f}(\mathbf{x}(s), \mathbf{u}(s)) \tag{6.20b}$$

$$\mathbf{x}(0) = \mathbf{x}(t_0) \tag{6.20c}$$

$$|\mathbf{u}(s)| \leq \mathbf{u}_{\max} \tag{6.20d}$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d$ with $\mathbf{x}_d = [x_d, y_d, \psi_d, u_d, v_d, r_d]^{\mathrm{T}}$ defined in (6.3) denotes the state error; $S(\delta)$ is the family of piecewise constant functions characterized by the sampling period δ; $T = N\delta$ is known as the prediction horizon, and $Q = \mathrm{diag}\{q_{ii}\}$, $R = \mathrm{diag}\{r_{ii}\}$, $P = \mathrm{diag}\{p_{ii}\}$ are weighting matrices, positive definite.

To establish a feedback mechanism, the NMPC tracking control needs to be implemented in the receding horizon fashion, i.e., we solve (P_0) for the current sampling time instant, and implement the solution for one sampling period, and then repeat the procedure at next sampling time. The NMPC tracking control algorithm is summarized in **Algorithm 7**.

Algorithm 7 : Centralized Implementation Algorithm

1: Initialization: Input the weighting matrices and prediction horizon in (6.20).
2: Fetch the state measurement $\mathbf{x}(t)$.
3: Solve the NMPC problem (6.20) with $\mathbf{x}(t_0) = \mathbf{x}(t)$, and let $\mathbf{u}^*(s)$ denote the solution.
4: Use $\mathbf{u}^*(s)$ for only one sampling period: $\mathbf{u}(t) = \mathbf{u}^*(s)$ for $s \in [0, \delta]$;
5: At next sampling time, set $t = t + \delta$; Goto step 2.

6.3.2 Solving the NMPC Problem

In the NMPC problem (P_0), the performance index J includes an integral operation and the constraints encompass the system model which are essentially derivative operations. In engineering practice, both the integral and derivative operations need to be performed numerically. Equivalently, instead of solving (P_0) in continuous-time

setting, we solve the discretized problem:

$$\min_{U} \; J = \sum_{k=0}^{N-1} (\|\tilde{\mathbf{x}}(k)\|_Q^2 + \|\mathbf{u}(k)\|_R^2) + \|\tilde{\mathbf{x}}(N)\|_P^2 \tag{6.21a}$$

$$\text{s.t.} \quad \mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)) \tag{6.21b}$$

$$\mathbf{x}(0) = \mathbf{x}(t_0) \tag{6.21c}$$

$$|\mathbf{u}(k)| \leq \mathbf{u}_{\max} \tag{6.21d}$$

Here, the system model is discretized and represented by $\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k))$ with $\mathbf{x}(i) = \mathbf{x}(i\delta)$ and $\mathbf{u}(i) = \mathbf{u}(i\delta)$; the sequence of control inputs $U = col(\mathbf{u}(0), \mathbf{u}(1), ..., \mathbf{u}(N-1))$ are the decision variables.

Since f is non-convex and is imposed in the constraints, solving (6.21) is a generic NLP problem. Solving a non-convex NLP problem is computationally expensive and unrealistic in real-time control if the problem size is too large. Therefore, it is desirable to (approximately) inspect the computational complexity. The sequential quadratic programming (SQP) method represents one of the most widely-used and effective NLP solvers, and we take SQP for example.

To solve the NLP, SQP generates a sequence of iterates which satisfies the first order KKT conditions. Each iterate is the solution of a QP subproblem that linearly approximates the local behavior of the original NLP problem in a neighborhood of the initial feasible point. Compared to the active-set counterpart which is of exponential complexity, the interior-point method has the polynomial complexity and is suggested to solve the QP subproblem. As a benchmark, the Mehrotra's predictor-corrector algorithm [90] is investigated.

The worst-case number of floating-point operations (flops) for the QP subproblem is given by the following formula [115]

$$\#\text{flops}_{\text{IP}} = i_{\text{IP}}(2/3(Nm)^3 + 2(Nm)^2) \tag{6.22}$$

for the condense formulation. Here, N is the prediction horizon and m is the number of control inputs, hence Nm is the number of decision variables for the optimization problem; i_{IP} is the number of interior point iterations which is expected to be $\mathcal{O}(\sqrt{Nm}\log(1/\epsilon))$ [89] for the ϵ-accurate solution, or simply a specified maximum iteration number $i_{\text{IP,max}}$. Neglecting minor operations in function evaluations, gradient

and Hessian updating, and line search in one major SQP iteration, a rough approximation of the computational complexity for numerically solving (P_0) using SQP can be given as

$$\#\text{flops}_{\text{SQP}} \approx i_{\text{SQP}} \times (\#\text{flops}_{\text{IP}}) \tag{6.23}$$

where i_{SQP} denotes the numbers of SQP iterations which is specified by the maximum iteration number $i_{\text{SQP,max}}$. From (6.23) we know that shrinking the sizes of m and N effectively reduces the computational complexity.

6.3.3 Distributed NMPC Implementation

For the model predictive control, it is well known that the prediction horizon affects not only the control performance but also the stability margin. Usually, we want to leave the prediction horizon a free parameter for tuning. Decreasing the number of control inputs, therefore, becomes the realistic option to alleviate the computational burden. In view of this point, the very first idea is to separate the thrust allocation (TA) from the tracking control.

Since P-5 is satisfied, the Moore - Penrose pseudoinverse method can be adopted as the closed-form solution for the TA:

$$\mathbf{u} = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\boldsymbol{\tau} = \mathbf{B}^{+}\boldsymbol{\tau} \tag{6.24}$$

Then we choose $\boldsymbol{\tau} = [F_u, F_v, F_r]^{\mathrm{T}}$ as the decision variables in (P_0). As a result, the number of control inputs m reduces from 4 to 3.

To facilitate the following derivations, instead of using $|\mathbf{B}^{+}\boldsymbol{\tau}(s)| \leq \mathbf{u}_{\max}$ in place of (6.20d), we use $|\boldsymbol{\tau}(s)| \leq \boldsymbol{\tau}_{\max}$ with $\boldsymbol{\tau}_{\max} = [F_{u,\max}, F_{v,\max}, F_{r,\max}]^{\mathrm{T}}$, which is a direct bound constraint on the decision variables. Suppose **Assumption 1** can be satisfied. Then we can take advantage of **Proposition 1** to guarantee that the TA is always feasible for the real AUV system. According to (6.22) the expected number of flops has already dropped by approximately 1/2 of that of the original problem (6.21).

Remark 15. *Although choosing τ_{max} that satisfies (4.12) guarantees the TA will fit the real system, it introduces some degree of conservativeness. Using $|\boldsymbol{\tau}(s)| \leq \boldsymbol{\tau}_{max}$ tightens the constraints in (P_0) since (4.12) is just a sufficient condition rather than the sufficient and necessary condition. However, the constraints tightening is important for the following distributed implementation because the original constraints $|\mathbf{B}^{+}\boldsymbol{\tau}(s)| \leq \mathbf{u}_{max}$ couples the decision variables. With the tightened constraints the*

decision variables $\boldsymbol{\tau} = [F_u, F_v, F_r]^T$ *are independent.*

From (2.32) and (6.3) we find that when the AUV exactly tracks the reference the dynamic equations of motion are loosely coupled. In fact, the surge dynamics (2.32a) and yaw dynamics (2.32c) are totally decoupled since $v = 0$. This observation inspires the idea of distributed implementation for solving the NMPC problem (6.21) approximately.

In the distributed control paradigm, the tracking control signals are calculated in parallel by three subsystems. The surge subsystem only considers the surge dynamics and the kinematics, and determines the control input F_u by solving the following subproblem:

$$\min_{F_u \in S(\delta)} J_1 = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{11} F_u^2(s)) ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{6.25a}$$

$$\text{s.t.} \quad \dot{\xi}(s) = f_1(\xi(s), \hat{v}(s), \hat{r}(s), F_u(s)) \tag{6.25b}$$

$$\xi(0) = \xi(t_0) \tag{6.25c}$$

$$|F_u(s)| \leq F_{\text{u,max}} \tag{6.25d}$$

where $\xi = [x, y, \psi, u]^{\text{T}}$ is the state of the surge subsystem and the subsystem model f_1 can be elaborated by taking the corresponding columns and rows in (6.19). The f_1 also contains the information of v and r but we can view them as known coefficients for some assumed trajectories $\hat{v}(s)$ and $\hat{r}(s)$.

Similarly, the sway subsystem includes the sway dynamics and the kinematics, and determines the control input F_v by solving the subproblem:

$$\min_{F_v \in S(\delta)} J_2 = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{22} F_v^2(s)) ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{6.26a}$$

$$\text{s.t.} \quad \dot{\zeta}(s) = f_2(\zeta(s), \hat{u}(s), \hat{r}(s), F_v(s)) \tag{6.26b}$$

$$\zeta(0) = \zeta(t_0) \tag{6.26c}$$

$$|F_v(s)| \leq F_{\text{v,max}} \tag{6.26d}$$

where $\zeta = [x, y, \psi, v]^{\text{T}}$ is the state of the sway subsystem and the model f_2 can be obtained by taking the corresponding columns and rows in (6.19).

The yaw subsystem encompasses the yaw dynamics and the kinematics, and de-

termines the control input F_r by solving the following optimization problem:

$$\min_{F_r \in S(\delta)} J_3 = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{33} F_r^2(s)) ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{6.27a}$$

$$\text{s.t.} \quad \dot{\omega}(s) = f_3(\omega(s), \hat{u}(s), \hat{v}(s), F_r(s)) \tag{6.27b}$$

$$\omega(0) = \omega(t_0) \tag{6.27c}$$

$$|F_r(s)| \leq F_{\text{r,max}} \tag{6.27d}$$

where $\omega = [x, y, \psi, r]^{\mathrm{T}}$ is the state of the yaw subsystem and the dynamics f_3 can be explicated by picking out the corresponding columns and rows in (6.19).

The assumed state trajectories $\hat{u}$, $\hat{v}$ and $\hat{r}$ can be determined using the following way: Let $\mathbf{F}_u^* = [F_u^*(0), ..., F_u^*(N-1)]^{\mathrm{T}}$, $\mathbf{F}_v^* = [F_v^*(0), ..., F_v^*(N-1)]^{\mathrm{T}}$, $\mathbf{F}_r^* = [F_r^*(0), ..., F_r^*(N-1)]^{\mathrm{T}}$ be the optimal solutions of the subproblems (6.25), (6.26) and (6.27) for the previous time. At current time, construct the assumed control signals using the previous solution as

$$\hat{\mathbf{F}}_u = [F_u^*(1), ..., F_u^*(N-1), F_u^*(N-1)]^{\mathrm{T}} \tag{6.28a}$$

$$\hat{\mathbf{F}}_v = [F_v^*(1), ..., F_v^*(N-1), F_v^*(N-1)]^{\mathrm{T}} \tag{6.28b}$$

$$\hat{\mathbf{F}}_r = [F_r^*(1), ..., F_r^*(N-1), F_r^*(N-1)]^{\mathrm{T}} \tag{6.28c}$$

Then the assumed state trajectories $\hat{u}$, $\hat{v}$ and $\hat{r}$ can be obtained via state evolution through the dynamic equations (2.32) and kinematic equations (2.34) starting from the updated system state $\mathbf{x}(t)$ for the current time.

By solving the subproblems in parallel, the numerical efficiency can be improved significantly. The worst-case number of flops is down to approximately 1/30 of that using the centralized implementation.

Remark 16. *In general, the optimal control obtained by the above distributed implementation is not a local minimum of the original problem. To converge to a local minimum the parallel computation needs to iterate [15], which is obviously undesired for the additional communication and computation burden. Fortunately, for the AUV tracking control the optimum is usually not a necessity. In MPC applications, it is well known that using the warm start (6.28), the optimal solutions won't differ too much from the initial guess. Therefore, we particularly exploit this property and avoid*

subsystem-wide iterations during the distributed implementation.

An initialization procedure is presented to facilitate the warm start: At the very first sampling instant, solve the centralized problem (6.20) once to obtain the assumed control trajectories $\hat{\boldsymbol{\tau}}(s) = \boldsymbol{\tau}^*(s)$ for $t = 0$. The distributed control algorithm is summarized in **Algorithm 8**.

Algorithm 8 : Distributed Implementation Algorithm

1: Initialization: Input the weighting matrices and prediction horizon; solve the NMPC problem (6.20) at t=0, let $\mathbf{u}^*(s)$ denote the solution; set $\hat{\boldsymbol{\tau}}(s) = \boldsymbol{\tau}^*(s) = \mathbf{B}\mathbf{u}^*(s)$.
2: Fetch the state measurement $\mathbf{x}(t)$.
3: Calculate the assumed state trajectories by $\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{B}^+\hat{\boldsymbol{\tau}})$ with $\hat{\mathbf{x}}(0) = \mathbf{x}(t)$.
4: Solve the subproblems (6.25), (6.26) and (6.27) in parallel; let $\boldsymbol{\tau}^*(s) = [F_u^*(s), F_v^*(s), F_r^*(s)]^{\mathrm{T}}$ denote the solution.
5: Set $\mathbf{u}^*(s) = \mathbf{B}^+\boldsymbol{\tau}^*(s)$; construct $\hat{\boldsymbol{\tau}}(s)$ using (6.28).
6: Use $\mathbf{u}^*(s)$ for only one sampling period: $\mathbf{u}(t) = \mathbf{u}^*(s)$ for $s \in [0, \delta]$.
7: At next sampling time, set $t = t + \delta$; Goto step 2.

So far, what we focused was only the computational complexity of the algorithm, and we had no explicit consideration of the closed-loop stability. Although with enough long prediction horizon the stability can always be obtained, it is preferable to provide some analytical means to guarantee this important closed-loop property.

The optimality of the solutions does not indicate the stability for the distributed implementation in **Algorithm 8**, not even for the centralized NMPC. To ensure it, we need to modify the subproblems and solve them sequentially.

Assume that $\mathbf{v}_d = [u_d, v_d, r_d]^{\mathrm{T}}$ obeys the dynamic equations of motion, then we can have the reference control forces $\boldsymbol{\tau}_d = [F_{ud}, F_{vd}, F_{rd}]^{\mathrm{T}}$ given by

$$\boldsymbol{\tau}_d = \mathbf{M}\dot{\mathbf{v}}_d + \mathbf{C}(\mathbf{v}_d)\mathbf{v}_d + \mathbf{D}(\mathbf{v}_d)\mathbf{v}_d + \mathbf{g}(\boldsymbol{\eta}_d) \tag{6.29}$$

where $\dot{\mathbf{v}}_d$ can be calculated by taking time derivative of (6.3).

Define $\tilde{F}_u = F_u - F_{ud}$, $\tilde{F}_v = F_v - F_{vd}$ and $\tilde{F}_r = F_r - F_{rd}$ and the surge subproblem is modified as follows:

$$\min_{F_u \in S(\delta)} J_1 = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{11}\tilde{F}_u^2(s))ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{6.30a}$$

$$\text{s.t.} \quad \dot{\xi}(s) = f_1(\xi(s), \hat{v}(s), \hat{r}(s), F_u(s)) \tag{6.30b}$$

$$\xi(0) = \xi(t_0) \tag{6.30c}$$

$$|F_u(s)| \leq F_{\text{u,max}} \tag{6.30d}$$

$$\tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), \lambda(0)) \leq \tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), h(\mathbf{x}(0))) \tag{6.30e}$$

where $h(\mathbf{x}) = \mathbf{B}^{+}\bar{h}(\mathbf{x})$ with $\bar{h}(\mathbf{x}) = [\bar{h}_u(\mathbf{x}), \bar{h}_v(\mathbf{x}), \bar{h}_r(\mathbf{x})]^{\mathrm{T}}$ is an auxiliary tracking controller and $V(\mathbf{x})$ is the corresponding Lyapunov function; $\lambda(0) = \mathbf{B}^{+}\bar{\lambda}(0)$ with $\bar{\lambda}(0) = [F_u(0), \bar{h}_v(\mathbf{x}(0)), \bar{h}_r(\mathbf{x}(0))]^T$.

The modified subproblem for the sway subsystem is constructed as follows:

$$\min_{F_v \in S(\delta)} J_2 = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{22}\tilde{F}_v^2(s))ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{6.31a}$$

$$\text{s.t.} \quad \dot{\zeta}(s) = f_2(\zeta(s), \hat{u}(s), \hat{r}(s), F_v(s)) \tag{6.31b}$$

$$\zeta(0) = \zeta(t_0) \tag{6.31c}$$

$$|F_v(s)| \leq F_{\text{v,max}} \tag{6.31d}$$

$$\tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), \pi(0)) \leq \tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), h(\mathbf{x}(0))) \tag{6.31e}$$

where $\pi(0) = \mathbf{B}^{+}\bar{\pi}(0)$ and $\bar{\pi}(0) = [F_u^*(0), F_v(0), \bar{h}_r(\mathbf{x}(0))]^{\mathrm{T}}$; $F_u^*(s)$ is the solution for (6.30) passed from the surge subsystem.

And the optimization problem for the yaw subsystem is modified as follows:

$$\min_{F_r \in S(\delta)} J_3 = \int_0^T (\|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{33}\tilde{F}_r^2(s))ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{6.32a}$$

$$\text{s.t.} \quad \dot{\omega}(s) = f_3(\omega(s), \hat{u}(s), \hat{v}(s), F_r(s)) \tag{6.32b}$$

$$\omega(0) = \omega(t_0) \tag{6.32c}$$

$$|F_r(s)| \leq F_{\text{r,max}} \tag{6.32d}$$

$$\tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), \mathbf{u}(0)) \leq \tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), h(\mathbf{x}(0))) \tag{6.32e}$$

where $\mathbf{u}(0) = \mathbf{B}^{+}\boldsymbol{\tau}(0)$ and $\boldsymbol{\tau}(0) = [F_u^*(0), F_v^*(0), F_r(0)]^{\mathrm{T}}$; $F_u^*(s)$, $F_v^*(s)$ are the solutions for (6.30) and (6.31), respectively.

The modified distributed control algorithm is summarized in **Algorithm 9**.

Since the subproblems are solved in sequence, the computational complexity for the modified distributed implementation is about 1/10 of that for the centralized

Algorithm 9 : Modified Distributed Implementation Algorithm

1: Initialization: Input the weighting matrices and prediction horizon; solve the NMPC problem (6.20) at t=0, let $\mathbf{u}^*(s)$ denote the solution; set $\hat{\boldsymbol{\tau}}(s) = \boldsymbol{\tau}^*(s) = \mathbf{B}\mathbf{u}^*(s)$.
2: Fetch the state measurement $\mathbf{x}(t)$.
3: Calculate the assumed state trajectories by $\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{B}^+\hat{\boldsymbol{\tau}})$ with $\hat{\mathbf{x}}(0) = \mathbf{x}(t)$.
4: Solve the subproblem (6.30) and send the solution $F_u^*(s)$ to the sway and yaw subsystems.
5: Solve the subproblem (6.31) and send the solution $F_v^*(s)$ to yaw subsystems.
6: Solve the subproblem (6.32) and let the solution be $F_r^*(s)$.
7: Set $\mathbf{u}^*(s) = \mathbf{B}^+\boldsymbol{\tau}^*(s)$ with $\boldsymbol{\tau}^*(s) = [F_u^*(s), F_v^*(s), F_r^*(s)]^{\mathrm{T}}$; construct $\hat{\boldsymbol{\tau}}(s)$ using (6.28).
8: Use $\mathbf{u}^*(s)$ for only one sampling period: $\mathbf{u}(t) = \mathbf{u}^*(s)$ for $s \in [0, \delta]$.
9: At next sampling time, set $t = t + \delta$; Goto step 2.

implementation. Although it is not as efficient as **Algorithm 8**, we can establish the feasibility theorem and stability theorem for **Algorithm 9**.

Theorem 10. *Choosing $F_{u,max} = F_{v,max} = F_{r,max} = \tau_{max}$ which satisfies (4.12), if $\|\bar{h}(\mathbf{x})\|_\infty \leq \tau_{max}$ can hold, then **Algorithm 9** admits recursive feasibility, i.e., we can always find an initial feasible solution for each of the optimizations (6.30)-(6.32) to start with.*

Proof. As $\|\bar{h}(\mathbf{x})\|_\infty \leq \tau_{\max}$, we have $|\bar{h}_u(\mathbf{x})| \leq F_{\mathrm{u,max}}$. Then the $\bar{h}_u(\mathbf{x})$ can be used as a feasible initial solution for the surge subproblem (6.30).

For the sway subsystem, since $|\bar{h}_v(\mathbf{x})| \leq F_{\mathrm{v,max}}$ and $F_u^*(s)$ satisfies (6.30e), it can be easily verified that $\bar{h}_v(\mathbf{x})$ must be a feasible solution for the subproblem (6.31).

Likewise, we have $|\bar{h}_r(\mathbf{x})| \leq F_{\mathrm{r,max}}$ and $F_u^*(s)$, $F_v^*(s)$ satisfying (6.31e). Therefore, $\bar{h}_r(\mathbf{x})$ is feasible for (6.32). □

Theorem 11. *Suppose that there exists an auxiliary control law $\bar{h}(\mathbf{x})$ such that $\tilde{\mathbf{x}} = \mathbf{0}$ is asymptotically stable for the closed-loop system controlled by $h(\mathbf{x}) = \mathbf{B}^+\bar{h}(\mathbf{x})$; $V(\mathbf{x})$ is the corresponding Lyapunov function. Provided that the recursive feasibility can be guaranteed, then the closed-loop system under **Algorithm 9** is asymptotically stable and the AUV converges to the desired trajectory.*

Proof. By converse Lyapunov theorems [75], there must be some functions $\gamma_i(\cdot)$, $i = 1, 2, 3$ which belong to class $\mathcal{K}_\infty$ such that the following inequalities hold:

$$\gamma_1(\|\mathbf{x}\|) \leq V(\mathbf{x}) \leq \gamma_2(\|\mathbf{x}\|) \tag{6.33a}$$

$$\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, h(\mathbf{x})) \leq -\gamma_3(\|\mathbf{x}\|) \tag{6.33b}$$

Since we have the contraction constraint (6.32e) and $\mathbf{u}(t) = \mathbf{u}^*(s)$ only for $s \in [0, \delta]$, the following condition is true:

$$\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})) \leq \frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, h(\mathbf{x})) \leq -\beta_3(\|\mathbf{x}\|) \tag{6.34}$$

Then by standard Lyapunov arguments (Theorem 4.8 in [75]) we claim that $\tilde{\mathbf{x}} = \mathbf{0}$ is asymptotically stable for closed-loop system controlled by **Algorithm 9**. □

Theorem 10 and **Theorem 11** depend on the auxiliary control law $\bar{h}(\mathbf{x})$. In principle, any Lyapunov-based tracking control law can be exploited. As an example, we can use the following backstepping control law

$$\bar{h}(\mathbf{x}) = \mathbf{M}\dot{\mathbf{v}}_r + \mathbf{C}\mathbf{v}_r + \mathbf{D}\mathbf{v}_r + \mathbf{g} - \mathbf{R}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} - \mathbf{R}^{\mathrm{T}}\mathbf{K}_d\mathbf{s} \tag{6.35}$$

which is derived in Section 4.3.2 and with the region of attraction (4.62) .

6.3.4 Simulation Results

In this section, we simulate the AUV to track a desired sinusoidal trajectory in order to demonstrate the effectiveness of the proposed distributed NMPC tracking control method. The AUV model parameters can be found in Table 2.1. All the simulations are performed on a personal laptop (CPU: Intel(R) Core(TM) i7-3520M: 2.90GHz 2.90GHz; RAM: 4.00GB).

The desired trajectory represents a sinusoidal path in the local level plane and is defined as follows:

$$p(t) = \begin{cases} x_d = 0.5t \\ y_d = \sin(0.5t) \end{cases} \tag{6.36}$$

For the NMPC tracking controller we use the following parameters: The sampling period $\delta = 0.1sec$; the prediction horizon $T = 5\delta$; the weighting matrices $Q = \mathrm{diag}(10^5, 10^5, 10^3, 10^2, 10^2, 10^2)$, $R = \mathrm{diag}(10^{-4}, 10^{-4}, 10^{-4}, 10^{-4})$ and $P = \mathrm{diag}(10^3, 10^3, 10^2, 10, 10, 10)$; the limit on each thruster is 500 (N). The control gains $\mathbf{K}_p = \mathbf{K}_d = \mathrm{diag}(1, 1, 1)$; and the initial condition $\mathbf{x}(0) = [0.5, 0, 0, 0, 0, 0]^{\mathrm{T}}$.

To solve the optimization problems, we adopt the SQP method and use the embedded Matlab function *fmincon* as the NLP solver in the simulations.

The trajectory tracking results are shown in Figure 6.10 and Figure 6.11. The blue curve is the simulated AUV trajectory using the auxiliary backstepping control (BSC); the green curve is the AUV trajectory with the centralized NMPC implementation; the magenta curve is the vehicle trajectory with the parallel distributed implementation (DMPC); and the red curve is with the modified sequential distributed implementation (mDMPC) while the black curve is the desired sinusoidal trajectory (REF). As can be seen, all the tracking controllers can drive the vehicle convergent to the desired trajectory, which shows the closed-loop stability.

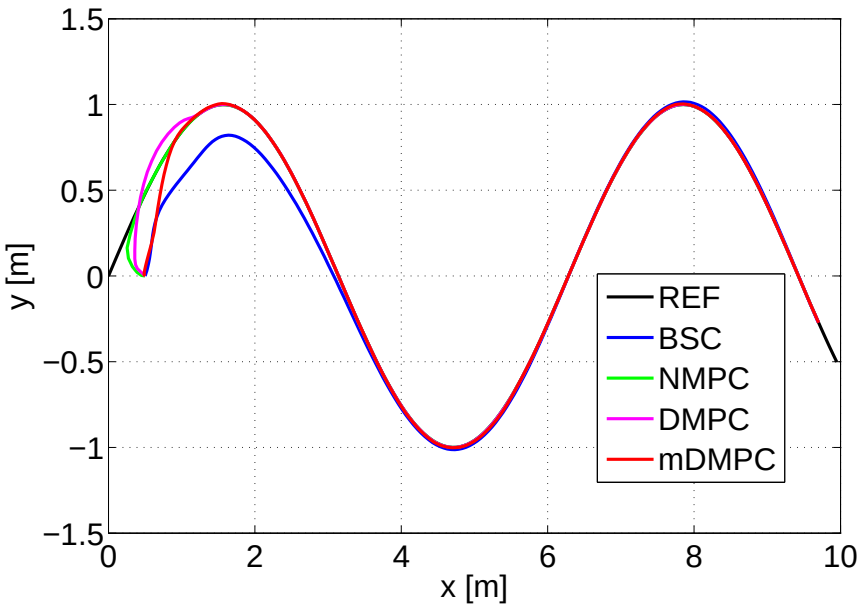

Figure 6.10: The AUV trajectory in the local level plane.

Comparing the tracking performance, we find that with the NMPC controller the AUV can converge faster than using the auxiliary BSC controller. This is because the NMPC automatically invokes online optimization to search for the best possible solution, while the control gains $\mathbf{K}_p$ and $\mathbf{K}_d$ for BSC are selected to be relatively small for a large guaranteed ROA. As discussed in **Remark 15**, since we introduce certain conservativeness into the distributed implementation, the centralized NMPC should outperform the DMPC, which can be observed from Figure 6.10 and Figure 6.11. However, if we compare the average computation time for each implementation, as listed in Table 6.4, the two distributed implementations demonstrate significant improvement in terms of numerical efficiency. This verifies the effectiveness and efficiency of the proposed DMPC implementation.

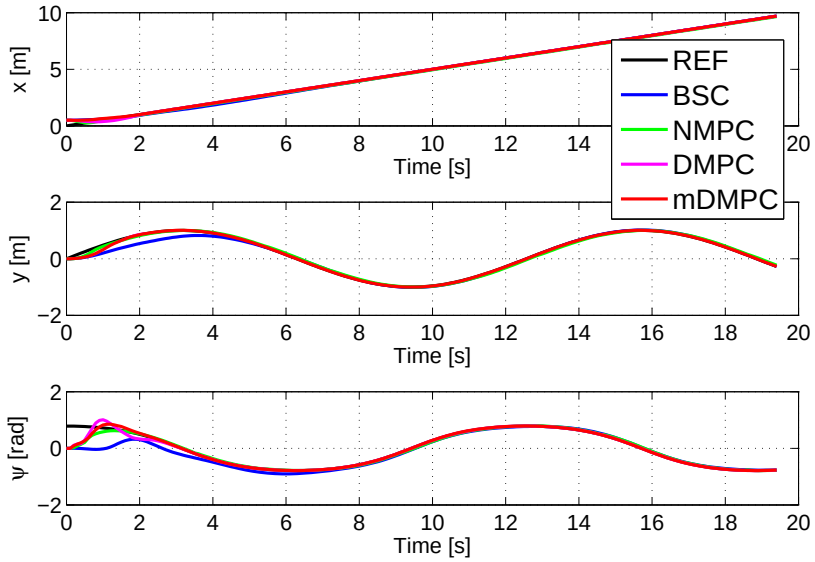

Figure 6.11: The state trajectories.

Table 6.4: Average computation time (*sec.*) per update.

N	NMPC	DMPC	mDMPC
3	0.1397	0.0081	0.0361
5	0.3641	0.0194	0.0960
10	1.3172	0.1512	0.5082

The control commands for the thrusters are recorded in Fig. 6.12. As we can see, in the beginning of the tracking, the NMPC tracking controller fully uses the onboard thrust capability to generate the fastest possible convergence. The magnitude of control commands stay within the permitted range as expected.

An extra merit of the NMPC tracking control refers to its inherent robustness against uncertainties and disturbances. We also test the robustness of the proposed tracking control in the simulations. It is assumed that there exists 30% model error in the obtained dynamic model of Falcon, then we simulate the tracking control with a disturbance $\mathbf{w} = [20(N), 20(N), 0(Nm)]^{\mathrm{T}}$ exerting on the vehicle.

From the simulation results illustrated in Figure 6.13 - Figure 6.15, it can be seen that the NMPC tracking control still leads the AUV converging to the desired trajectory in the presence of parametric uncertainties and external disturbances. The mean square errors (MSE) are provided in Table 6.5. The MSE with the NMPC (either centralized implementation or distributed implementation) is much smaller

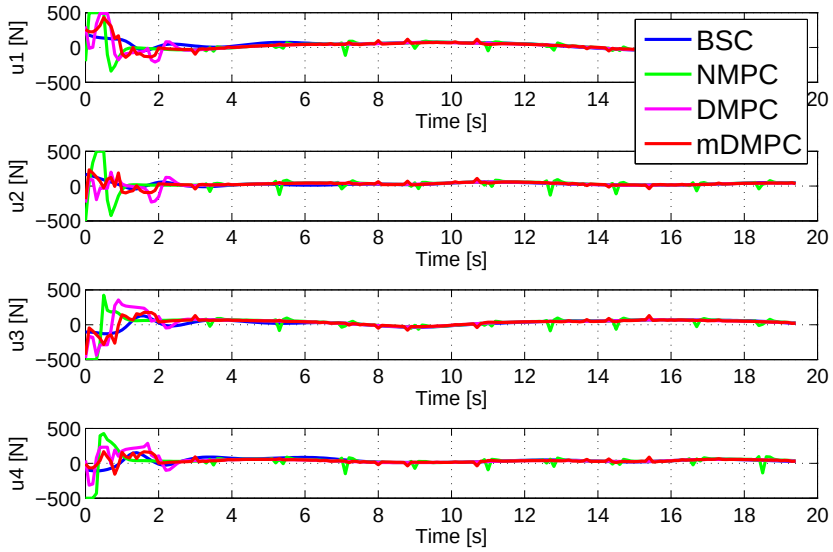

Figure 6.12: The control input signals.

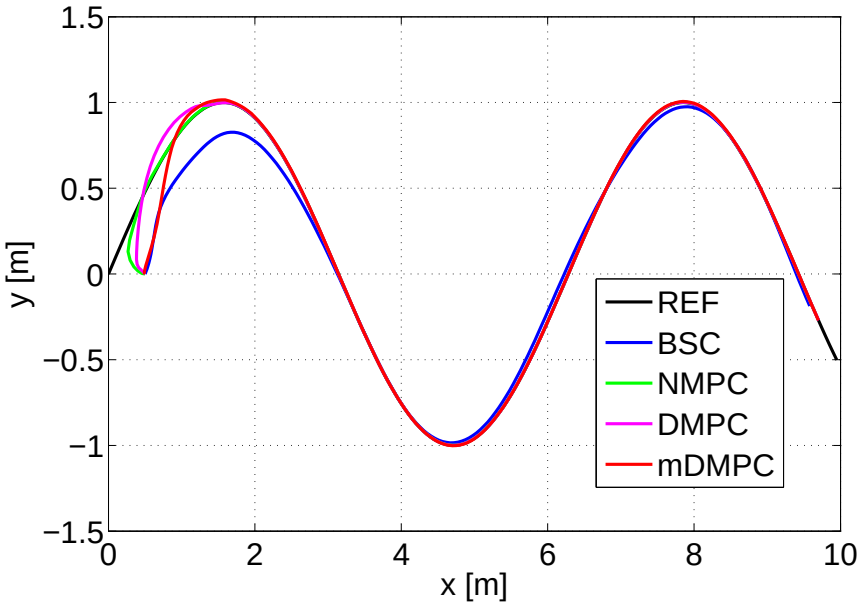

Figure 6.13: The AUV trajectory in the local level plane (with disturbance).

than that with BSC, which demonstrates the good robustness of the proposed method.

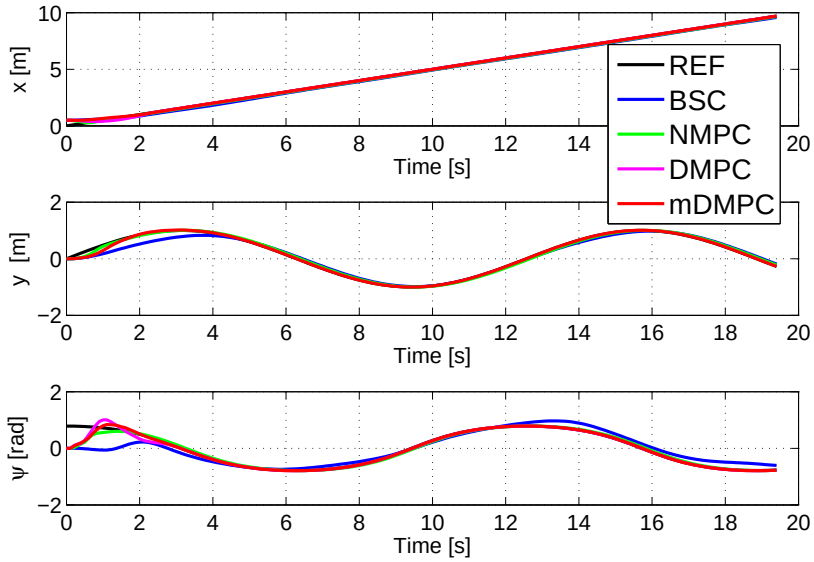

Figure 6.14: The state trajectories (with disturbance).

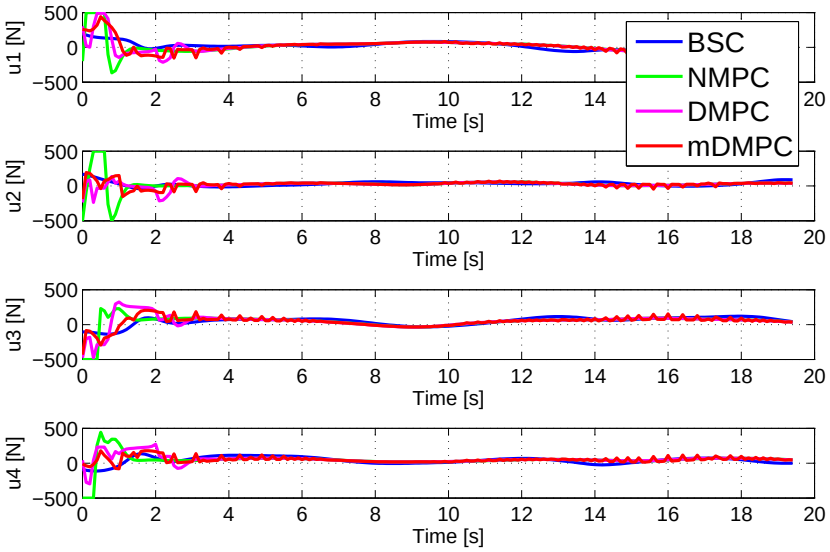

Figure 6.15: The control input signals (with disturbance).

6.4 Conclusion

In this chapter, we have investigated the nonlinear model predictive control for the trajectory tracking application of an autonomous underwater vehicle. To remove the

Table 6.5: Mean square errors for AUV tracking with disturbances.

MSE	BSC	NMPC	DMPC	mDMPC
x [m^2]	0.0161	0.0058	0.0054	0.0058
y [m^2]	0.0158	0.0019	0.0026	0.0020
ψ [rad^2]	0.0711	0.0184	0.0173	0.0161

computational barrier and increase the possibility of applying the NMPC to real-world AUV systems, we proposed two fast implementation strategies that largely reduce the computational complexity while maintaining good tracking control performance. Firstly, the modified C/GMRES algorithm was presented. By exploiting the relationship between Pontryagin minimum principle and KKT conditions, the barrier functions were successfully incorporated into the numerical continuation approximation. The convergence of the approximate solution was proved. Extensive simulation studies revealed the effectiveness and efficiency of the proposed modified C/GMRES algorithm for the AUV trajectory tracking control. Secondly, a distributed implementation paradigm was developed to enhance the control stability in the fast implementation of the NMPC tracking control. By exploiting the dynamic properties of the AUV motion, the optimal control signals can be determined by solving several subproblems with smaller size. Since the computational complexity increases exponentially with the problem size, the numerical efficiency can be significantly improved. A parallel implementation algorithm and a sequential implementation algorithm were proposed. The parallel implementation can minimize the computational complexity while the sequential implementation can theoretically guarantee the closed-loop stability. Simulation results suggested that the distributed implementation method was not only numerically efficient but also robust against disturbance and uncertainties.

The main results in Section 6.2 have been published in [121], and the main results in Section 6.3 have been summarized in [122] and submitted for possible publication.

[121] **C. Shen**, B. Buckham and Y. Shi, "Modified C/GMRES Algorithm for Fast Nonlinear Model Predictive Tracking Control of an AUV", *IEEE Transactions on Control Systems Technology*, vol.25, no.5, pp.1896-1904, 2017.

[122] **C. Shen**, Y. Shi and B. Buckham, "Distributed implementation for nonlinear model predictive tracking control of an AUV", *IEEE/ASME Transactions on Mechatronics*, under review, 2017.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This dissertation studies the MPC controller design problem for the AUVs. Three fundamental motion control problems, namely, the trajectory tracking, dynamic positioning and path following control, are investigated and end up with effective and inspiring MPC solutions.

Chapter 3 was concerned with the combined path planning and tracking control problem for the AUV to navigate in the constrained workspace. A novel RHO framework consisting of the spline-based dynamic path planning and the NMPC trajectory tracking control was presented. The dynamic path planning approximated a minimum curvature reference path by repeatedly solving local optimization problems with boundary conditions imposed from the previous solution, which accommodated the practically finite perceiving capability of the AUV. The planned path was then assigned a timing law and augmented to provide valid reference for each state of the AUV. The error system was elaborately designed, which facilitated the stability analysis of the closed-loop tracking control systems. Sufficient conditions that guarantee the closed-loop stability were derived.

In Chapter 4, the dynamic positioning and the trajectory tracking control problems were studied for the AUV. An LMPC framework was developed for the AUV to draw computing power (online optimization) to improve the motion control performance. Based on a nonlinear PD type auxiliary control law, the LMPC-based dynamic positioning controller was designed. Recursive feasibility of the OCP and the stability of the closed-loop control system were analyzed. A quasi-global asymp-

totic stability could be claimed. The theoretical results obtained for the dynamic positioning control was further extended to solve the trajectory tracking problem. With the help of a nonlinear backstepping tracking control law, the contraction constraint was constructed and imposed in the OCP, which ensured the stability of the closed-loop LMPC tracking control system. The unique feature of the LMPC framework is that the design procedure is constructive and concise. By exploiting the existing Lyapunov-based control law, the closed-loop stability can be conveniently guaranteed. Since the stability is not determined by the quality of the optimal solution, it essentially creates a trade-off between computational complexity and control performance for the designers to allocate the computing resources.

Chapter 5 was devoted to the AUV path following control problem. A novel MOMPC framework was proposed. Within the MOMPC framework, the different priorities between the Path Convergence and the Speed Assignment could be fully respected. Taking advantage of an explicit parametrization of the ZPE manifold, we formulated the path following control problem into vector-valued OCPs. To solve the OCPs, two solution methods that handle objective prioritization were investigated and followed by the development of the WS-MOMPC algorithm and the LO-MOMPC algorithm, respectively. Conditions that guarantee the closed-loop stability under the two algorithms were provided, and the internal relationship between the two solution methods were explored. The proposed MOMPC method not only presents a novel framework to address the AUV path following problem, but more importantly, it lays a theoretical foundation for the study of AUV motion control problems which involve multiple control objectives.

Different from the previous chapters which emphasize the closed-loop property of the proposed MPC solution, in Chapter 6, the computational complexity of the MPC algorithm was particularly investigated. Two distinct fast implementation strategies, namely, the modified C/GMRES algorithm and the distributed NMPC implementation, were proposed in an attempt to remove the potential computational barrier in applying the MPC to real-world AUV systems. The modified C/GMRES algorithm took advantage of the numerical continuation method to approximate the optimal control signals without solving the NLP problem at each sampling time instant, hence dramatically reduced the computing time. The barrier function was proposed to handle the inequality constraints in terms of thrust limits. Under differentiability and LICQ, the validity of the incorporation of the barrier functions into the numerical continuation approximations could be shown. To further preserve the closed-loop

stability in the fast implementation, in Chapter 6, another strategy called distributed NMPC implementation was also proposed for the AUV tracking control. The distributed implementation exploited the dynamic properties of the AUV motion and calculated one control command at a time by solving well-defined subproblems with smaller size. Since the computational complexity increases exponentially with the problem size, the proposed distributed implementation method can be numerically efficient. By imposing additional constraints in these subproblems and by solving them sequentially, the stability of the closed-loop system under the distributed NMPC implementation could be guaranteed. The work in Chapter 6 reveals the possibility of eliminating the computational cost of implementing NMPC while guaranteeing the closed-loop stability theoretically. Thus, it is expected that more novel and efficient MPC solutions will be inspired for various AUV applications.

7.2 Future Work

The work presented in this dissertation focuses on the design of MPC controllers to solve the AUV motion control problems and the design of efficient numerical algorithms to implement these MPC controllers. The derived results try to provide some theoretical insights into the application of MPC to the AUV systems. However, the results have only been verified through simulation examples; their experimental validation needs to be carried out in the near future. Furthermore, since the MPC solution is new to the AUVs, there are many interesting open problems deserving studies. Specifically, the following closely related research topics are suggested for possible future work.

- **MPC-based motion controller design with incremental input constraints.** All the MPC controllers designed in this dissertation only consider the bound input constraint which represents the thrust limits on the real AUV system. However, this is a rather simplified problem formulation. As we can see from Figure (4.3), for the LMPC-based DP control, the control signal varies too rapidly, which is undesired in real control applications. The fast varying control command is either forbidden by the real thruster dynamics or wearing out the actuators. In practice, the change rate of the control input should be considered as well. As shown in [146], despite the assumptions, the incremental input constraints can also be handled by the NMPC technique. Therefore, it is

of great interest to extend the results obtained in this dissertation to incorporate this realistic type of control inputs.

- **6 DOF motion controller design using LMPC.** The control problems addressed in this dissertation are concerned with the AUV motion in the local level plane. This is partially due to the physical limitation of the experimental platform Falcon and partially due to the complexity of the motion control problem itself. When considering the full 6 DOF dynamic model, the couplings among the DOFs may be too complicated for the standard NMPC to handle. However, as demonstrated in Chapter 4, with the LMPC framework the closed-loop stability can be conveniently guaranteed with the help of an auxiliary Lyapunov-based control law. As there are many existing results can be exploited, it is quite interesting to explore the LMPC-based control structure, possibly with distributed implementation algorithms, for the AUV motion control problem in the general 6 DOF settings.

- **Distributed MPC for cooperative control of multiple AUVs.** The cooperative control of multi-agent system has been a hot research topic for decades. The distributed model predictive control (DMPC) presents an effective and promising tool to design the coordination control laws among the agents [87]. The distributed implementation proposed in Chapter 6 is essentially part of the DMPC technique. The surge, sway, yaw subsystems considered in the distributed implementation algorithm can be analogous to several independent AUVs in a cooperative control task. The design procedure may be inspiring for the cooperative control law design. Compared with a single AUV, a fleet of AUVs can perform more tasks with higher efficiency. Therefore, the DMPC control design with guaranteed recursive feasibility and closed-loop feasibility is practically desirable for the AUVs.

Appendix A

Forward Difference Generalized Minimal Residual Method

In Section 6.2, the forward difference generalized minimal residual (FDGMRES) method is applied in **Algorithm 6** to numerically solve the linear equation $F_U\dot{U} = A_sF - F_x\dot{x} - F_t$ which is approximated by the following equation:

$$D_\delta F(U, x + \delta\dot{x}, t + \delta : \dot{U}, 0, 0) = b(U, x, \dot{x}, t) \tag{A.1}$$

where

$$b(U, x, \dot{x}, t) = A_sF(U, x, t) - D_\delta F(U, x, t : 0, \dot{x}, 1) \tag{A.2}$$

The detailed FDGMRES is summarized in **Algorithm 10**. It is worth noting that the generalized minimal residual method (GMRES) is a kind of Krylov subspace method to iteratively solve linear equation systems in forms of $Ax = b$. At kth iteration, GMRES searches for $x \in x_0 + \mathcal{K}_k$ such that the residual $\rho = \| b - Ax \|$ is minimized, where $\mathcal{K}_k = \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$ with $r_0 = b - Ax_0$. To facilitate the residual minimization, Gram-Schmidt process (STEP 5 - STEP 11 in **Algorithm 10**) is performed, so that the original minimization in Krylov subspace is transformed into an unconstrained least squares problem (STEP 13 in **Algorithm 10**). The least squares problem can be solved efficiently with QR decomposition or singular value decomposition. One important feature of GMRES is that for a large linear equation system, usually a small $k_{\max}$ is enough to give an satisfactory approximate solution with a specified error tolerance.

Algorithm 10 : $\dot{U} = \text{FDGMRES}(U, x, \dot{x}, t, \hat{\dot{U}}, \delta, k_{\max})$

$\hat{r} = b(U, x, \dot{x}, t) - D_\delta F(U, x + \delta\dot{x}, t + \delta : \hat{\dot{U}}, 0, 0)$.
$v_1 = \hat{r} / \parallel \hat{r} \parallel$, $\rho = \parallel \hat{r} \parallel$, $\beta = \rho$, $k = 0$.
for $k < k_{\max}$ **do**
    $k = k + 1$
    $v_{k+1} = D_\delta F(U, x + \delta\dot{x}, t + \delta : v_k, 0, 0)$
    for $j = 1, \ldots, k$ **do**
        $h_{jk} = v_{k+1}^{\mathrm{T}} v_j$
        $v_{k+1} = v_{k+1} - h_{jk} v_j$
    end for
    $h_{k+1,k} = \parallel v_{k+1} \parallel$
    $v_{k+1} = v_{k+1} / \parallel v_{k+1} \parallel$
    $e_1 = [1, 0, \ldots, 0]^{\mathrm{T}} \in \mathbb{R}^{k+1}$, $H_k = \{h_{ij}\} \in \mathbb{R}^{(k+1)\times k}$ ($h_{ij} = 0$ for $i > j + 1$)
    Minimize $\parallel \beta e_1 - H_k y^k \parallel$ to determine $y^k \in \mathbb{R}^k$
    $\rho = \parallel \beta e_1 - H_k y^k \parallel$.
end for
$\dot{U} = \hat{\dot{U}} + V_k y^k$, where $V_k = [v_1, \ldots, v_k] \in \mathbb{R}^{mN\times k}$.

Appendix B

Derivation of Jacobians

In Section 6.2, the Jacobian matrices need to be calculated for the execution of mC/GMRES algorithm. Specifically, for the NMPC tracjectory tracking control problem (6.4), we have inequality constraints $|F_u| \leq F_{u,\max}$, $|F_v| \leq F_{v,\max}$ and $|F_r| \leq F_{r,\max}$, which are equivalent to

$$F_u^2 - F_{u,\max}^2 \leq 0 \tag{B.1a}$$
$$F_v^2 - F_{v,\max}^2 \leq 0 \tag{B.1b}$$
$$F_r^2 - F_{r,\max}^2 \leq 0 \tag{B.1c}$$

Hence we have the modified Hamiltonian

$$\begin{aligned} H = \; & q_{11}(x-x_d)^2 + q_{22}(y-y_d)^2 + q_{33}(\psi-\psi_d)^2 + q_{44}(u-u_d)^2 + q_{55}(v-v_d)^2 \\ & + q_{66}(r-r_d)^2 + r_{11}F_u^2 + r_{22}F_v^2 + r_{33}F_r^2 - \gamma_k \log(F_{u,\max}^2 - F_u^2) - \gamma_k \log(F_{v,\max}^2 \\ & - F_v^2) - \gamma_k \log(F_{r,\max}^2 - F_r^2) + \bar{\lambda}_1 u \cos\psi - \bar{\lambda}_1 v \sin\psi + \bar{\lambda}_2 u \sin\psi + \bar{\lambda}_2 v \cos\psi \\ & + \bar{\lambda}_3 r + \bar{\lambda}_4 \tfrac{F_u}{M_{\dot{u}}} + \bar{\lambda}_4 \tfrac{M_{\dot{v}}}{M_{\dot{u}}} vr - \bar{\lambda}_4 \tfrac{X_u}{M_{\dot{u}}} u - \bar{\lambda}_4 \tfrac{D_u}{M_{\dot{u}}} |u|u + \bar{\lambda}_5 \tfrac{F_v}{M_{\dot{v}}} - \bar{\lambda}_5 \tfrac{M_{\dot{u}}}{M_{\dot{v}}} ur \\ & - \bar{\lambda}_5 \tfrac{Y_v}{M_{\dot{v}}} v - \bar{\lambda}_5 \tfrac{D_v}{M_{\dot{v}}} |v|v + \bar{\lambda}_6 \tfrac{F_r}{M_{\dot{r}}} + \bar{\lambda}_6 \tfrac{M_{\dot{u}} - M_{\dot{v}}}{M_{\dot{r}}} uv - \bar{\lambda}_6 \tfrac{N_r}{M_{\dot{r}}} r - \bar{\lambda}_6 \tfrac{D_r}{M_{\dot{r}}} |r|r \end{aligned}$$

Then we calculate the Jacobians $H_{\mathbf{x}}$ and $H_{\mathbf{u}}$ in the following:

$$H_{\mathbf{x}} = \begin{bmatrix} 2q_{11}(x-x_d) \\ 2q_{22}(y-y_d) \\ 2q_{33}(\psi-\psi_d) - \bar{\lambda}_1 u \sin\psi - \bar{\lambda}_1 v \cos\psi + \bar{\lambda}_2 u \cos\psi - \bar{\lambda}_2 v \sin\psi \\ 2q_{44}(u-u_d) + \bar{\lambda}_1 \cos\psi + \bar{\lambda}_2 \sin\psi - \bar{\lambda}_4 \tfrac{X_u}{M_{\dot{u}}} - \bar{\lambda}_5 \tfrac{M_{\dot{u}}}{M_{\dot{v}}} r + \bar{\lambda}_6 \tfrac{M_{\dot{u}} - M_{\dot{v}}}{M_{\dot{r}}} v - 2\bar{\lambda}_4 \tfrac{D_u}{M_{\dot{u}}} |u| \\ 2q_{55}(v-v_d) - \bar{\lambda}_1 \sin\psi + \bar{\lambda}_2 \cos\psi + \bar{\lambda}_4 \tfrac{M_{\dot{v}}}{M_{\dot{u}}} r - \bar{\lambda}_5 \tfrac{Y_v}{M_{\dot{v}}} + \bar{\lambda}_6 \tfrac{M_{\dot{u}} - M_{\dot{v}}}{M_{\dot{r}}} u - 2\bar{\lambda}_5 \tfrac{D_v}{M_{\dot{v}}} |v| \\ 2q_{66}(r-r_d) + \bar{\lambda}_3 + \bar{\lambda}_4 \tfrac{M_{\dot{v}}}{M_{\dot{u}}} v - \bar{\lambda}_5 \tfrac{M_{\dot{u}}}{M_{\dot{v}}} u - \bar{\lambda}_6 \tfrac{N_r}{M_{\dot{r}}} - 2\bar{\lambda}_6 \tfrac{D_r}{M_{\dot{r}}} |r| \end{bmatrix}$$

$$H_{\mathbf{u}} = \begin{bmatrix} 2r_{11}F_u + \frac{\bar{\lambda}_4}{M_{\dot{u}}} + \gamma_k \frac{2F_u}{F_{u,\max}^2 - F_u^2} \\ 2r_{22}F_v + \frac{\bar{\lambda}_5}{M_{\dot{v}}} + \gamma_k \frac{2F_v}{F_{v,\max}^2 - F_v^2} \\ 2r_{33}F_r + \frac{\bar{\lambda}_6}{M_{\dot{r}}} + \gamma_k \frac{2F_r}{F_{r,\max}^2 - F_r^2} \end{bmatrix}$$

Also we have $g_{\mathbf{x}} = [2q_{f11}(x - x_d), 2q_{f22}(y - y_d), 2q_{f33}(\psi - \psi_d), 2q_{f44}(u - u_d), 2q_{f55}(v - v_d), 2q_{f66}(r - r_d)]^{\mathrm{T}}$, where q_{ii}, r_{ii} and q_{fii} are the corresponding diagonal elements of weighting matrices Q, R and Q_f.

Appendix C

Publications

- **Refereed journal papers that have been published or accepted**

J1. **C. Shen**, Y. Shi and B. Buckham, "Integrated Path Planning and Tracking Control of an AUV: A Unified Receding Horizon Optimization Approach", *IEEE/ASME Transactions on Mechatronics*, vol.22, no.3, pp.1163-1173, 2017.

J2. **C. Shen**, B. Buckham and Y. Shi, "Modified C/GMRES Algorithm for Fast Nonlinear Model Predictive Tracking Control of an AUV", *IEEE Transactions on Control Systems Technology*, vol.25, no.5, pp.1896-1904, 2017.

J3. Y. Shi, **C. Shen**, H. Fang and H. Li, "Advanced Control in Marine Mechatronic Systems: A Survey", *IEEE/ASME Transactions on Mechatronics*, vol.22, no.3, pp.1121-1131, 2017.

J4. **C. Shen**, Y. Shi and B. Buckham, "Trajectory Tracking Control of an Autonomous Underwater Vehicle using Lyapunov-based Model Predictive Control", *IEEE Transactions on Industrial Electronics*, accepted, 2017, DOI: 10.1109/TIE.2017.2779442

J5. **C. Shen**, Y. Shi and B. Buckham, "Path-Following Control of an AUV: A Multi-Objective Model Predictive Control Approach", *IEEE Transactions on Control Systems Technology*, accepted, 2017, DOI: 10.1109/TCST.2018.2789440

- **Refereed journal papers that are under review**

J6. **C. Shen**, Y. Shi and B. Buckham, "Distributed implementation for nonlinear model predictive tracking control of an AUV", *IEEE/ASME Transactions on Mechatronics*, under review, 2017.

- **Refereed conference papers that have been published**

C1. **C. Shen**, Y. Shi and B. Buckham, "Lyapunov-based Model Predictive Control for Dynamic Positioning of Autonomous Underwater Vehicles", in *Proceedings of the 2017 IEEE International Conference on Unmanned Systems (ICUS)*, Beijing, China, 2017, pp. 1-6. (Best Paper Award)

C2. **C. Shen**, K. Zhang, Y. Shi and B. Buckham, "Dynamic Positioning of an AUV: A Lyapunov-based Model Predictive Control Approach", in *Proceedings of the 26th Canadian Congress of Applied Mechanics*, Victoria, BC, Canada, 2017, pp. 1-4.

C3. **C. Shen**, Y. Shi and B. Buckham, "Nonlinear Model Predictive Control for Trajectory Tracking of an AUV: A Distributed Implementation", in *Proceeding of the 55th IEEE Conference on Decision and Control*, Las Vegas, NV, USA, 2016, pp. 5998-6003

C4. **C. Shen**, Y. Shi and B. Buckham, "Path-Following Control of an AUV using Multi-Objective Model Predictive Control", in *Proceedings of the 2016 American Control Conference*, Boston, MA, USA, 2016, pp. 4507-4512.

C5. **C. Shen**, Y. Shi and B. Buckham, "Model Predictive Control for an AUV with Dynamic Path Planning", in *Proceedings of the Joint 34th Chinese Control Conference and SICE Annual Conference 2015*, Hangzhou, China, 2015, pp. 475-480.

Bibliography

[1] A. Aguiar and J. Hespanha. Position tracking of underactuated vehicles. In *Proceedings of American Control Conference*, Danver, CO, USA, 2003.

[2] A. Aguiar and J. Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertanty. *IEEE Transactions on Automatic Control*, 52:1362–1379, 2007.

[3] A. Aguiar and A. Pascoal. Regulation of a nonholonomic autonomous underwater vehicle with parametric modeling uncertainty using Lyapunov functions. In *Proceedings of the 40st IEEE Conference on Decision and Control*, Orlando, FL, USA, 2001.

[4] A. Aguiar and A. Pascoal. Global stabilization of an underactuated autonomous underwater vehicle via logic-based switching. In *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, USA, 2002.

[5] M. Alamir and G. Bornard. Stability of a truncated infinite constrained receding horizon scheme: the general discrete nonlinear case. *Automatica*, 31(9):1353–1356, 1995.

[6] A. Alessandretti, P. Aguiar, C. Cunha, and C. Jones. Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predicitive control. In *Proceedings of European Control Conference*, Zurich, Switzerland, 2013.

[7] B. Allen, R. Stokey, N. Forrester, R. Gouldsborough, M. Purcell, and C. von Alt. REMUS: A small, low cost AUV; system description, field trials and performance results. In *Proceedings of IEEE/MTS OCEANS'97*, Halifax, Canada, 1997.

[8] L. Allgower and K. Georg. *Numerical Continuation Methods*. Springer, Heidelberg, 1990.

[9] A. Antoniou and W. Lu. *Practical Optimization: Algorithms and Engineering Applications*. Springer Science & Business Media, 2007.

[10] T. Baca, G. Loianno, and M. Saska. Embedded model predictive control of unmanned micro aerial vehicles. In *The 21st International Conference on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, 2016.

[11] R. Bachmayer, S. Humphris, D. Fornari, C. Van Dover, J. Howland, A. Bowen, R. Elder, T. Crook, D. Gleason, W. Sellers, and S. Lerner. Oceanographic research using remotely operated underwater robotic vehicles: Exploration of hydrothermal vent sites on the mid-atlantic ridge at 37^o north 32^o west. *Marine Technology Society Journal*, 32(3):37–47, 1998.

[12] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[13] A. Bemporad and D. Munoz de la Pena. Multiobjective model predictive control. *Automatica*, 45:2823–2830, 2009.

[14] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist. Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles. *IEEE Transactions on Automation Science and Engineering*, 7:167–172, 2010.

[15] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[16] B. Bingham, B. Foley, H. Singh, R. Camilli, K. Dellaporta, R. Eustice, A. Mallios, D. Mindell, C. Roman, and D. Sakellariou. Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *Journal of Field Robotics*, 27:702–717, 2010.

[17] M. Bock and A. Kugi. Real-time nonlinear model predictive path-following control of a laboratory tower crane. *IEEE Transactions on Control Systems Technology*, 22:1461–1474, 2014.

[18] C. Boor. *A Practical Guide to Splines*. Springer, 2001.

[19] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[20] L. Brinon-Arranz, A. Seuret, and C. Canudas de Wit. Cooperative control design for time-varying formations of multi-agent systems. *IEEE Transactions on Automatic Control*, 59:2283–2288, 2014.

[21] A. Bryson and Y. Ho. *Applied Optimal Control: Optimization Estimation and Control*. Taylor & Francis Group, New York, NJ, 1975.

[22] C. Buskens and H. Maurer. SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control. *Journal of Computational and Applied Mathematics*, 120:85–108, 2000.

[23] C. Caldwell, E. Collins, and D. Dunlap. Motion planning for an autonomous underwater vehicle via sampling based model predictive control. In *OCEANS 2010 MTS/IEEE*, Seattle, USA, 2010.

[24] E. Camponogara, D. Jia, B. Kroph, and S. Talukdar. Distributed model predictive control. *IEEE Control System Magazine*, 22:44–52, 2002.

[25] M. Carreras, J. Batlle, and P. Ridao. Hybrid coordination of reinforcement learning-based behaviors for auv control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, USA, 2001.

[26] W. Chen. Disturbance observer based control for nonlinear systems. *IEEE/ASME Transactions on Mechatronics*, 9(4):706–710, 2004.

[27] W. Chen, Y. Wei, J. Zeng, H. Han, and X. Jia. Adaptive terminal sliding mode NDO-based control of underactuated AUV in vertical plane. *Discrete Dynamics in Nature and Society*, 2016:1–9, 2016.

[28] C. Chin, M. Lau, E. Low, and G. Seet. Design of thrusters configuration and thrust allocation control for a remotely operated vehicle. In *Proceedings of IEEE International Conference on Robotics*, Thailand, 2006.

[29] L. Chisci, A. Lombardi, and E. Mosca. Dual receding horizon control of constrained discrete-time systems. *European Journal of Control*, 2:278–285, 1996.

[30] P. Christofides, J. Liu, and D. Munoz de la Pena. *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications*. Springer-Verlag, London, England, 2011.

[31] P. Christofides, R. Scattolini, D. Munoz de la Pena, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers and Chemical Engineering*, 51:21–41, 2013.

[32] L. Cooney. Dynamic response and maneuvering strategies of a hybrid autonomous underwater vehicle in hovering. Master's thesis, Massachusetts Institute of Technology, 2006.

[33] K. Deb. *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, NY, 2001.

[34] S. Desset, R. Damus, F. Hover, J. Morash, and V. Polidoro. Closer to deep underwater science with ODYSSEY IV class hovering autonomous underwater vehicle HAUV. In *Proceedings of IEEE/MTS OCEANS 2005*, Brest, France, 2005.

[35] D. Diehl, H. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. *Nonlinear Model Predictive Control*, 384:391–417, 2009.

[36] K. Do, Z. Jiang, J. Pan, and H. Nijmeijer. A global output-feedback controller for stabilization and tracking of underactuated ODIN: A spherical underwater vehicle. *Automatica*, 40:117–124, 2004.

[37] J. Elvander and G. Hawkes. ROVs and AUVs in support of marine renewable technologies. In *OCEANS 2012 MTS/IEEE*, Hampton Roads, USA, 2012.

[38] P. Encarnacao and A. Pascoal. 3D path-following for autonomous underwater vehicles. In *Proceedings of 39th IEEE Conference on Desicion and Control*, Sydney, Australia, 2000.

[39] P. Encarnacao, A. Pascoal, and M. Arcak. Path-following for marine vehicles in the presence of unknown currents. In *Proceedings of 6th IFAC Symposium on Robot Control*, Vienna, Austria, 2000.

[40] H. Esfahani, V. Azimirad, and M. Danesh. A time delay controller included terminal sliding mode and fuzzy gain tuning for underwater vehicle-manipulator systems. *Ocean Engineering*, 107:97–107, 2015.

[41] T. Faulwasser. *Optimization-based Soultions to Constrained Trajectory-tracking and Path-following Problems*. PhD thesis, Otto-von-Guericke-Universitat, Magdeburg, Germany, 2012.

[42] T. Faulwasser and R. Findeisen. Constrained output path-following for nonlinear systems using predictive control. In *Proceedings of 8th IFAC Symposium on Nonlinear Control Systems*, Bologna, Italy, 2010.

[43] T. Faulwasser, B. Kern, and R. Findeisen. Model predictive path-following for constrained nonlinear systems. In *Proceedings of 48th IEEE Conference on Decision and Control*, Shanghai, China, 2009.

[44] T. Faulwasser, T. Weber, P. Zometa, and R. Findeisen. Implementation of nonlinear model predictive path-following control for an industrial robot. *arXiv:1506.09084 [cs.SY]*, 2015.

[45] R. Findeisen and F. Allgower. Computational delay in nonlinear model predictive control. In *Proceedings of International Symposium on Advanced Control of Chemical Processes*, Hong Kong, 2003.

[46] T. Fossen. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, Trondheim, Norway, 2002.

[47] T. Fossen and K. Pettersen. On uniform semiglobal exponential stability (USGES) of propotional line-of-sight guidance law. *Automatica*, 50(10):2912–2917, 2014.

[48] T. Fossen and S. Sagatun. Adaptive control of nonlinear underwater robotic systems. In *Proceedings of IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, 1991.

[49] T I Fossen. *Marine Control Systems: guidance, navigation and control of ships,rigs and underwater vehicles*. Marine Cybernetics AS. Trondlheim, Norway, 2002.

[50] Thor I Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley & Sons, 1994.

[51] R. Frezza. Path following for air vehicles in coordinated flight. In *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, USA, 1999.

[52] J. Gaya, L. Goncalves, A. Duarte, B. Zanchetta, P. Drews, and S. Botelho. Vision-based obstacle avoidance using deep learning. In *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, Recife, Brazil, 2016.

[53] R. Gomes, J. Sousa, and F. Pereira. Integrated maneuver and control design for ROV operations. In *OCEANS 2003 MTS/IEEE*, San Diego, CA, USA, 2003.

[54] K. Graichen. A fixed-point iteration scheme for real-time model predictive control. *Automatica*, 48:1300–1305, 2012.

[55] L. Grune. NMPC without terminal constraints. In *Proceedings of IFAC conference on nonlinear model predictive control*, Noordwijkerhout, The Netherlands, 2012.

[56] D. Gu and H. Hu. Receding horizon tracking control of wheeled mobile robots. *IEEE Transactions on Control Systems Technology*, 14(4):743–749, 2006.

[57] D. He, L. Wang, and J. Sun. On stability of multiobjective NMPC with objective prioritization. *Automatica*, 57:189–198, 2015.

[58] A. J. Healey and D. Lienard. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, 18(3):327–339, 1993.

[59] J. Healey and D. Marco. Slow speed flight control of autonomous underwater vehicles: experimental results with NPS AUV II. In *Proceedings of the 2nd International Offshore and Polar Engineering Conference*, San Francisco, CA, USA, 1992.

[60] T. Holzhuter. LQG approach for high-precision track control of ships. 144(2):121–127, 1997.

[61] T. Howard, C. Green, and A. Kally. Receding horizon model-predictive control for mobile robot navigation of intricate paths. In *The 7th International Conference on Field and Service Robotics*, Cambridge, Massachusetts, USA, 2009.

[62] M. Hunt, W. Marquet, D. Moller, K. Peal, W. Smith, and R. Spindell. An acoustic navigation system. Technical report, Woods Hole Oceanographic Institution, 1974.

[63] J. Jalbert, J. Baker, J. Duchesney, P. Pietryka, W. Dalton, D. Bidberg, S. Chappell, R. Nitzel, and K. Holappa. Solar powered autonomous underwater vehicle development. In *Proceedings of the 13th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, USA, 2003.

[64] T. Johansen. Optimizing nonlinear control allocation. In *Proceedings of the 43rd IEEE Conference on Desicion and Control*, Atlantis, Paradise Island, Bahamas, 2004.

[65] T. Johansen, T. Fossen, and P. Tondel. Efficient optimal constrained control allocation via multi-parametric programming. *AIAA Journal of Guidance, Control and Dynamics*, 28:506–515, 2005.

[66] T. Johansen, T. Fuglseth, P. Tondel, and T. Fossen. Optimal constrained control allocation in marine surface vessels with rudders. In *Proceedings of the IFAC Conference on Manoeuvering and Control of Marine Craft*, Girona, Spain, 2003.

[67] D. Jung and P. Tsiotras. On-line path generation for small unmanned aerial vehicles using b-spline. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, USA, 2008.

[68] E. Jung, B. Yi, and W. Kim. Kinematic analysis and motion planning for a planar multiarticulated omnidirectional mobile robot. *IEEE/ASME Transactions on Mechatronics*, 20:2983–2995, 2015.

[69] R. Kalman. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mex.*, 5:102–119, 1960.

[70] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *IEEE International Conference on Robotics and Automation*, Cincinnati, OH, 1990.

[71] M. Katebi, M. Grimble, and Y. Zhang. H_∞ robust control design for dynamic ship positioning. *IEE Proceedings - Control Theory and Applications*, 144:110–120, 1997.

[72] S. Keerthi and E.Gilbert. Optimal, infinite horizon feedback laws for a general class of constrained discrete time systems: Stability and moving-horiozon approximations. *Journal of Optimization Theory and Application*, 57:265–293, 1988.

[73] C. Kelley. *Iterative Methods for Linear and Nonlinear Equations.* SIAM, Philadelphia, PA, 1995.

[74] E. Kerrigan and J. Maciejowski. Designing model predictive controllers with prioritised contraints and objectives. In *Proceedings of IEEE International Symposium on Computer Aided Control System Design 2002*, Darmstadt, Germany, 2002.

[75] H. Khalil. *Nonlinear Systems.* Prentice Hall, New York, 1996.

[76] L. Lapierre and B. Jouvencel. Robust nonlinear path-following control of an AUV. *IEEE Journal of Oceanic Engineering*, 33, 2008.

[77] L. Lapierre and D. Soetanto. Nonlinear path-following control of an AUV. *Ocean Engineering*, 34, 2007.

[78] H. Li and Y. Shi. Event-triggered robust model predictive control of continous-time nonlinear systems. *Automatica*, 50(4):1507–1513, 2014.

[79] H. Li and Y. Shi. *Robust Receding Horizon Control for Networked and Distributed Nonlinear Systems.* Studies in Systems, Decision and Control. Springer, 2016.

[80] W. Li, S. Xu, G. Zhao, and L.Goh. Adaptive knot placement in b-spline curve approximation. *Computer-Aided Design*, 37:791–797, 2005.

[81] K. Lindegaard. *Acceleration Feedback in Dynamic Positioning Systems.* PhD thesis, Norwegian University of Science and Technology, 2003.

[82] J. Liu, D. Munoz de la Pena, P. Christofides, and J. Davis. Lyapunov-based model predictive control of nonlinear systems subject to time-varying measurement delays. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

[83] S. Liu, Y. Liu, and N. Wang. Nonlinear disturbance observer-based backstepping finite-time sliding mode tracking control of underwater vehicles with system uncertainties and external disturbances. *Nonlinear Dynamics*, 88(1):465–476, 2017.

[84] X. Liu, Y. Shi, and D. Constantinescu. Robust contrained model predictive control using contraction theory. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, Los Angeles, CA, USA, 2014.

[85] D. Maalouf, A. Chemori, and V. Creuze. L_1 adaptive depth and pitch control of an underwater vehicle with real-time experiments. *Ocean Engineering*, 98:66–77, 2015.

[86] D. Mayne. An apologia for stabilising terminal conditions in model predictive control. *International Journal of Control*, 86(11):2090–2095, 2013.

[87] D. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50:2967–2986, 2014.

[88] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–841, 2000.

[89] L. McGovern. *Computational analysis of real-time convex optimization for control system*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2000.

[90] S. Mchrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2:575–601, 1992.

[91] B. Meyer, K. Ehlers, C. Osterloh, and E. Maehle. Smart-E an autonomous omnidirectional underwater robot. *Jounral of Behavioral Robotics*, 4:204–210, 2013.

[92] P. Mhaskar, N. El-Farra, and P. Christofides. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *System & Control Letters*, 55:650–659, 2006.

[93] A. Micaelli and C. Samson. Trajectory tracking for a unicycle-type and two steering wheels mobile robots. Technical Report 2097, INRIA, Sophia-Antipolis, France, 1993.

[94] H. Michalska. A new formulation of receding horizon control without a terminal constraint on the state. *European Journal of Control*, 3(1):2–14, 1997.

[95] K. Miettinen. *Nonlinear multiobjective optimization.* Kluwer Academic Publishers, 1999.

[96] L. Moreira and C. Guedes Soares. H_2 and H_∞ design for diving and course control of an autonomous underwater vehicle in presence of waves. *IEEE Journal of Oceanic Engineering*, 33(2):69–88, 2008.

[97] K. Mukherjee, I. Kar, and R. Bhatt. Disturbance observer based tracking controller for an autonomous underwater vehicle. In *2015 IEEE Underwater Technology (UT)*, Chennai, India, 2015.

[98] M. Muller and F. Allgower. Improving performance in model predictive control: Switching cost functions under average dwell-time. *Automatica*, 48:402–409, 2012.

[99] W. Naeem, R. Sutton, and M. Ahmad. LQG/LTR control of an autonomous underwater vehicle using hybrid guidance law. In *Proceedings of the GCUV'03 Conference*, Newport, UK, 2003.

[100] W. Naeem, R. Sutton, and S. Ahmad. Pure pursuit guidance and model predictive control of an autonomous underwater vehicle for cable/pipeline tracking. In *World Maritime Technology Conference*, San Francisco, Calnifornia, USA, 2003.

[101] C. Ocampo-Martinez, A. Ingimundarson, V. Puig, and J. Quevedo. Objective prioritization using lexicographic minimizers for MPC of sewer networks. *IEEE Transactions on Control Systems Technology*, 16:113–121, 2008.

[102] S. Oh and J. Sun. Path following of underactuated marine vessels using line-of-sight based model predictive control. *Ocean Eng.*, 37:289–295, 2010.

[103] T. Ohtsuka. A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4):563–574, 2004.

[104] G. Oriolo and Y. Nakamura. Control of mechanical systems with second-order nonholonomic constraits: underactuated manipulators. In *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, England, 1991.

[105] G. Pannocchia, J. Rawlings, and S. Wright. Inherently robust suboptimal nonlinear MPC: Theory and application. In *Proceedings of the IEEE 50th Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, 2011.

[106] A. Pavlov, H. Nordahl, and M. Breivik. MPC-based optimal path following for underactuated vessels. In *Proc. 8th IFAC Int. Conf. Manoeuving Control Marine Craft*, pages 340–345, Guaruja, Brazil, 2009.

[107] V. Poonamallee, S. Yurkovich, A. Serrani, D. Doman, and M. Oppenheimer. A nonlinear programming approach for control allocation. In *Proceedings of American Control Conference*, Boston, MA, USA, 2004.

[108] J. Primbs and V. Nevistic. Constrained finite receding horizon linear quadratic control. Technical report, California Institute of Technology, 1997.

[109] A. Proctor. *Semi-automous Guidance and Control of a Saab SeaEye Falcon ROV*. PhD thesis, University of Victoria, 2014.

[110] S. Qin and T. Badgwell. An overview of nonlinear model predictive control applications. In *Nonlinear Model Predictive Control*, pages 369–392. Birkhauser, 2000.

[111] J. Rawlings and D. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI, 2009.

[112] J. Refsnes, A. Sorensen, and K. Pettersen. Model-based output feedback control of slender-body underactuated AUVs: Theory and experiments. *IEEE Transactions on Automatic Control*, 16:930–946, 2008.

[113] F. Repoulias and E. Papadopoulos. Plannar trajectory planning and tracking control design for underactuated AUVs. *Ocean Engineering*, 34(11-12):1650–1667, 2007.

[114] David Ribas, Pere Ridao, and Jose Neira. *Underwater SLAM for Structured Environments Using an Imaging Sonar*, chapter 3: Design and Development of the Ictineu AUV, pages 23–35. Springer Berlin Heidelberg, 2010.

[115] S. Richter, C. Jones, and M. Morari. Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. *IEEE Transactions on Automatic Control*, 57:1391–1403, 2012.

[116] G. Roberts and R. Sutton. *Advances in Unmanned Marine Vehicles*. Stevenage, United Kingdom: The Institution of Electrical Engineers, 2006.

[117] C. Roman and R. Mather. Autonomous underwater vehicles as tools for deep-submergence archaeology. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for Maritime Environment*, 224:327–340, 2010.

[118] A. Saccon, J. Hauser, and A. Beghi. Trajectory exploration of a rigid motorcycle model. *IEEE Transactions on Control Systems Technology*, 20:424–437, 2012.

[119] T. Salgado-Jimenez, J. Spiewak, P. Fraisse, and B. Jouvencel. A robust control algorithm for AUV: based on a high order sliding mode. In *Proceedings of MTS/IEEE Techno-Oceans'04*, Kobe, Japan, 2004.

[120] P. Scokaert, D. Mayne, and J. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648–654, 1999.

[121] C. Shen, B. Buckham, and Y. Shi. Modified C/GMRES algorithm for fast nonlinear model predictive tracking control of AUVs. *IEEE Transactions on Control Systems Technology*, 25(5):1896 – 1904, 2017.

[122] C. Shen, Y. Shi, and B. Buckham. Distributed implementation for nonlinear model predictive tracking control of an AUV. *IEEE/ASME Transactions on Mechatronics*, under review, 2017.

[123] C. Shen, Y. Shi, and B. Buckham. Integrated path planning and tracking control of an AUV: A unified receding horizon optimization approach. *IEEE/ASME Transactions on Mechatronics*, 22(3):1163 – 1173, 2017.

[124] C. Shen, Y. Shi, and B. Buckham. Lyapunov-based model predictive control for dynamic positioning of autonomous underwater vehicles. In *Proceedings of the 2017 IEEE International Conference on Unmanned Systems (ICUS)*, Beijing, China, 2017.

[125] C. Shen, Y. Shi, and B. Buckham. Path-following control of an AUV: A multi-objective model predictive control approach. *IEEE Transactions on Control Systems Technology*, accepted, 2017, DOI: 10.1109/TCST.2018.2789440.

[126] C. Shen, Y. Shi, and B. Buckham. Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control. *IEEE Transactions on Industrial Electronics*, accepted, 2017, DOI: 10.1109/TIE.2017.2779442.

[127] J. Slotine and W. Li. *Applied Nonlinear Control*. NJ: Prentice-Hall, 1991.

[128] A. Sørensen. A survey of dynamic positioning control systems. *Annual Reviews in Control*, 35:123–136, 2011.

[129] A. Sørensen, S. Sagatun, and T. Fossen. Design of a dynamic positioning system using model-based control. *Control Engineering Practice*, 4:359–368, 1996.

[130] S. Soylu, B. Buckham, and R. Podhorodeski. Robust control of underwater vehicles with fault-tolerant infinity-norm thruster force allocation. In *OCEANS 2007 MTS/IEEE*, Vancouver, BC, Canada, 2007.

[131] S. Soylu, B. Buckham, and R. Podhorodeski. A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation. *Ocean Engineering*, 35(16):1647–1659, 2008.

[132] S. Soylu, A. Proctor, R. Podhorodeski, C. Bradley, and B. Buckham. Precise trajectory control for an inspection class ROV. *Ocean Engineering*, 111:508–523, 2015.

[133] E. Tannuri, A. Agostinho, H. Morishita, and L. Moratelli. Dynamic positioning systems: An experimental analysis of sliding mode control. *Control Engineering Practice*, 18:1121–1132, 2010.

[134] T.Fossen and A. Grøvlen. Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping. *IEEE Transactions on Control Systems Technology*, 6:121–128, 1998.

[135] M. Triantafyllou and M. Grosenbaugh. Robust control for underwater vehicle systems with time delays. *IEEE Journal of Oceanic Engineering*, 16:146–151, 1991.

[136] A. Veksler, T. Johansen, F. Borrelli, and B. Realfsen. Dynamic positioning with model predictive control. *IEEE Transactions on Control Systems Technology*, 4:1340–1353, 2016.

[137] L. Wang, H. Jia, L. Zhang, and H. Wang. Horizontal tracking control for AUV based on nonlinear sliding mode. In *Proceedings of IEEE International Conference on Information and Automation*, Shenyang, China, 2012.

[138] R. Wernli. AUVs commercialization - who's leading the pack? In *OCEANS 2001 MTS/IEEE*, Honolulu, HI, USA, 2001.

[139] R. Wynn, V. Huvenne, T. Le Bas, B. Murton, D. Connelly, B. Bett, H. Ruhl, K. Morris, J. Peakall, D. Parsons, E. Sumner, S. Darby, R. Dorrell, and J. Hunt. Autonomous underwater vehicle (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352:451–468, 2014.

[140] X. Xiang, C. Yu, Z. Niu, and Q. Zhang. Subsea cable tracking by autonomous underwater vehicle with magnetic sensing guidance. *Sensors*, 16:1335, 2016.

[141] J. Xu, M. Wang, and L. Qiao. Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles. *Ocean Engineering*, 105:54–63, 2015.

[142] D. Yoerger, M. Jakuba, A. Bradley, and B. Bingham. Techniques for deep sea near bottom survey using an autonomous underwater vehicle. *International Journal of Robotics Research*, 26:41–54, 2007.

[143] S. Yoon and C. Qiao. Cooperative search and survey using autonomous underwater vehicles (AUVs). *IEEE Transaction on Parallel and Distributed Systems*, 22(3):364–379, 2011.

[144] H. Yu, K. Meier, M. Argyle, and R. Beard. Cooperative path planning for target tracking in urban environments using unmanned air and groud vehicles. *IEEE/ASME Transactions on Mechatronics*, 20:541–552, 2015.

[145] S. Yu, X. Li, H. Chen, and F. Allgower. Nonlinear model predictive control for path following problems. *International Journal of Robust and Nonlinear Control*, 25:1168–1182, 2015.

[146] S. Yu, T. Qu, F. Xu, H. Chen, and Y. Hu. Stability of finite horizon model predictive control with incremental input constraits. *Automatica*, 79:265–272, 2017.

[147] Y. Yuan. *ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics*, chapter A review of trust region algorithms for optimization, pages 271–282. Oxford University Press, 2000.

[148] J. Yuh. Learning control for underwater robotic vehicles. *IEEE Control Systems*, 4:39–46, 1994.

[149] D. Zambrano and E. Camacho. Applications of MPC with multiple objectives for a solar refrigeration plant. In *Proceedings of the IEEE conference on control applications*, Glasgow, Scotland, 2002.

[150] V. Zavala and A. Flores-Tlacuahuac. Stability of multiobjective predictive control: a utopia-tracking approach. *Automatica*, 48:2627–2632, 2012.

[151] S. Zhao, J. Yuh, and S. Choi. Adaptive DOB control for AUVs. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004.