Signe Moe

Guidance and Control of Robot Manipulators and Autonomous Marine Robots

Thesis for the degree of philosophiae doctor
Trondheim, November 2016

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Innovation and Creativity

NTNU
Norwegian University of Science and Technology

Thesis for the degree of philosophiae doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-326-1982-5 (printed version)
ISBN 978-82-326-1983-2 (electronic version)
ISSN 1503-8181
ITK report: 2016-15-W

Doctoral theses at NTNU, 2016:322

Printed by Skipnes Kommunikasjon as

In memory of B.

Summary

This thesis is motivated by the increasing use of robots within numerous fields and in a vast range of applications. The use of robots provides several advantages, e.g. reduced labor costs, increased production and productivity, high precision and flexibility. Fully autonomous robotic systems operate independently of humans, and there is a strong desire to utilize robotic systems to completely automate processes, also outside of the production industry.

The singularity-robust multiple task-priority inverse kinematics framework is a widely used kinematic control method for a general robotic system. This framework allows several equality tasks, i.e. tasks with a defined desired value (for instance end effector position), to be defined, prioritized and achieved simultaneously. However, for a general robotic system, several goals may be described as set-based tasks, which are tasks that have a desired interval of values rather than one exact desired value, e.g. collision or joint limit avoidance. In this thesis, a systematic approach to include set-based tasks into this framework is presented. It is proven that set-based tasks given high priority, i.e. above the highest priority equality task, are satisfied at all times, but that the same cannot be guaranteed for low-priority set-based tasks. The resulting closed-loop dynamic system can be described as a discontinuous differential equation, and by using switched control systems theory it is proven that the equality task errors converge asymptotically to zero when including set-based tasks into the framework given that certain, specified conditions are satisfied. Simulation and experimental results are presented which confirm the correctness and effectiveness of the proposed method.

An unmanned marine vehicle may be considered a type of robotic system. However, these vehicles pose some additional challenges due to hydrodynamic effects, underactuation, thrust allocation, communication and environmental influence. A cornerstone ability of autonomous marine vehicles, both surface and underwater, is to converge to and follow a general path under the influence of ocean currents. This thesis presents two approaches to achieve this for surface vehicles, one of which is based on absolute velocity measurements only, and one which assumes that relative velocity measurements are available. The latter has also been extended to the underwater case. Common for these results are the fact that the control system is designed to make the vehicle side-slip such that the total vehicle velocity is aligned with the desired path, even if the heading is not. This is necessary to counteract both for the curvature of the path and the ocean current. Convergence to the desired path is achieved with asymptotic stability properties under explicit conditions. For surface vehicles the results hold globally, whereas for underwater vehicles they hold for a defined domain (due to the well-known singularity of Euler angle representation).

Another highly important ability of autonomous marine vehicles is to avoid collisions, and in nautical navigation all surface vessels are subject to the International Regulations for Preventing Collisions at Sea (COLREGs). This thesis introduces a switched guidance system for underactuated surface vehicles to achieve collision avoidance and path following. The proposed method is based on set-based control, is adapted to the underactuation of marine vehicles, is highly generic, and may be used for any combination of methods for path following and collision avoidance. A specific guidance law for the collision avoidance mode is suggested that will, if satisfied, make the surface vessel track a circle with a constant safe radius about the moving obstacle center in a COLREGs compliant manner. The set-based guidance system is proven to prevent collisions given that certain, specified assumptions on the obstacle velocity are satisfied and the references provided by the guidance system are tracked. Furthermore, path following is achieved as long as the surface vessel can safely follow the path without colliding.

Contents

Summary **iii**

Contents **v**

List of figures **ix**

List of tables **xi**

List of abbreviations **xiii**

Preface **xv**

Publications **xvii**

1 Introduction **1**
- 1.1 Background and Motivation 1
- 1.2 Modeling and Control of Robotic Systems 2
- 1.3 Modeling and Control of Marine Vehicles 6
- 1.4 Thesis Outline and Contributions 8

2 Mathematical Preliminaries and Models **13**
- 2.1 Notation 13
- 2.2 Trigonometric Formulas 13
- 2.3 Stability Theory 14
- 2.4 Modeling of Marine Surface Vessels and Underwater Vehicles . . . 17
- 2.5 The Singularity-robust Multiple Task-priority Inverse Kinematics Framework 23

I Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework **29**

---

3 Preliminary Result: Kinematic Control of a Dual-Arm UVMS **31**
- 3.1 Vehicle Kinematics 32
- 3.2 Vehicle Tasks 34
- 3.3 Guidance System 37

3.4 Simulation Results 39
3.5 The Need for Set-based Tasks 39
3.6 Conclusions 41

4 High-priority Set-based Tasks 43
4.1 Definition of Set-based Tasks 43
4.2 Set-based Singularity-robust Multiple Task-priority Inverse Kinematics 44
4.3 Stability Analysis 51
4.4 Simulation Results 52
4.5 Conclusions 53

5 Set-based Tasks: Low-priority and combination of high- and low-priority 55
5.1 Set-based singularity-robust task-priority inverse kinematics 55
5.2 Stability analysis 63
5.3 Conclusions 65

6 Set-based Tasks: Implementation and Experimental Results 67
6.1 Implementation 67
6.2 Experimental Setup 70
6.3 Experimental Results 72
6.4 Conclusions 80

II Curved Path Following of Marine Vehicles in the Presence of Unknown Ocean Currents 83

7 A Serret-Frenet approach for Surface Vessels 85
7.1 Control Objectives 85
7.2 The Control System 86
7.3 Main Result 89
7.4 Simulation Results 91
7.5 Conclusions 93

8 A Serret-Frenet approach for Underwater Vehicles 95
8.1 Control Objectives 95
8.2 The Control System 97
8.3 Main Result 101
8.4 Simulation Results 104
8.5 Conclusions 105

9 Line-of-Sight with side-slip for Surface Vessels and AUVs in the horizontal plane 109
9.1 Control Objectives 109
9.2 The Control System 110
9.3 Main Result 115

9.4 Simulation Results 118
9.5 Conclusions 120

III A Set-based Approach for Path Following of Marine Vehicles with Collision Avoidance 121

10 Set-based Line-of-Sight Path Following with Collision Avoidance 123
10.1 Control Objectives 124
10.2 The Control System 124
10.3 Main Result 132
10.4 Simulation Results: Straight Line Path 133
10.5 Conclusions 134

11 Conclusions and Future Work 139

Appendices 143

A Appendix Chapter 7 145

B Appendix Chapter 8, I 151

C Appendix Chapter 8, II 155

References 165

List of figures

1.1 Examples of different robots. 2
1.2 A typical control structure for a robotic system with joint configuration $\boldsymbol{q}$ and joint velocities and accelerations $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$. 3

2.1 Illustration of (2.7). 14

3.1 Illustration of vehicle base with two manipulator arms with position and orientation of the vehicle body and the manipulator end effectors. . . 33
3.2 Illustration of $\boldsymbol{a}_{\text{des}}$ and $\boldsymbol{a}$ for relative field of view. 37
3.3 Block diagram of the proposed guidance system when one task for the leader and one task for the follower is implemented. 38
3.4 Simulation results: The green dashed line shows the desired trajectory for end effector 1. 40
3.5 Simulation results: Task errors for manipulator 1 and vehicle body (a) and manipulator 2 (b). 41

4.1 Illustration of valid set D. The set-based task σ is satisfied in D and violated outside of D. 44
4.2 Simulation results: The desired position is marked as a red cross. . . . 54
4.3 Equality task errors and the set-based task with its lower limit. 54

6.1 Graphic illustration of the function `in_T_RC` with return value `True` shown in green and `False` in red. 68
6.2 Coordinate frames corresponding to the D-H parameters in Table 6.2. 70
6.3 The control structure of the experiments. The tested algorithm is implemented in the kinematic controller block. 71
6.4 Logged data from experimental results, Example 1-3. 77
6.5 Pictures from simulation and actual experiments, Example 1. 78
6.6 Desired and actual end trajectory plotted in red and blue respectively. 80
6.7 Actual vs. desired end effector trajectory from Example 4 plotted in the xy-, xz- and yz-plane. 80
6.8 Logged data from Example 4. 81

7.1 Serret-Frenet path representation. 87
7.2 Desired and actual path C_1. 92
7.3 Path following errors and actual vs. estimated ocean current for C_1 . . 93

7.4 Desired vs. actual surge velocity, yaw and yaw rate for C_1. 93
7.5 Desired and actual path C_2. 94
7.6 Path following errors and actual vs. estimated ocean current for C_2 . . 94
7.7 Desired vs. actual surge velocity, yaw and yaw rate for C_2. 94

8.1 Desired and actual path C_1. 106
8.2 Path following of C_1. 106
8.3 Path following errors and actual vs. estimated ocean current for C_1 . . 107
8.4 Desired vs. actual surge velocity, orientation and pitch/yaw rate for C_1. 107
8.5 Desired and actual path C_2. 107
8.6 Path following of C_2. 108
8.7 Path following errors and actual vs. estimated ocean current for C_2 . . 108
8.8 Desired vs. actual surge velocity, orientation and pitch/yaw rate for C_2. 108

9.1 Desired path C, path-tangential reference frame with orientation $\gamma_p(\theta)$ and cross-track error y_e illustrated. 111
9.2 Desired and actual path C_1. 119
9.3 Cross-track error and desired vs. actual surge velocity, yaw and yaw rate for C_1. 119
9.4 Desired and actual path C_2. 119
9.5 Cross-track error and desired vs. actual surge velocity, yaw and yaw rate for C_2. 119

10.1 Illustration of parameters used for collision avoidance. 127
10.2 COLREGs scenarioes and the correct behavior of the involved vessels. 129
10.3 The different COLREGs scenarios as function of ω. 129
10.4 The set $D = [\min(R_\mathrm{m}, \max(\sigma, R_\mathrm{o})), \infty)$ used in Algorithm 10.1 illustrated in green. 131
10.5 Trajectory for straight line path following with collision avoidance. . . 136
10.6 Distance between the USV and the three obstacles over time. 137
10.7 Path following error over time (top) and reference vs. actual surge velocity (center) and heading (bottom). 137

List of tables

4.1 System equations for the resulting 2^j modes for j high-priority set-based tasks. . . . 50

5.1 System equations for the resulting 2^j modes for j low-priority set-based tasks. . . . 60

6.1 Table illustrating the activation of mode based on the boolean variables as defined in (6.2). . . . 69
6.2 Table of the D-H parameters of the UR5. The corresponding coordinate systems can be seen in Figure 6.2. . . . 70
6.3 Implemented tasks in Example 1 sorted by decreasing priority. . . . 73
6.4 Table illustrating the activation of mode in Example 1. . . . 74
6.5 Implemented tasks in Example 2 sorted by decreasing priority. . . . 74
6.6 Table illustrating the activation of mode in Example 2. . . . 75
6.7 Implemented tasks in Example 3 sorted by decreasing priority. . . . 75
6.8 Implemented tasks in Example 4 sorted by decreasing priority. . . . 79
6.9 Table illustrating the activation of mode in Example 4. . . . 79

10.1 Table of implemented obstacles in simulation. . . . 134

List of abbreviations

AUV	Autonomous Underwater Vehicle
CB	Center of Buoyancy
CG	Center of Gravity
CLIK	Closed-Loop Inverse Kinematics
COLREGS	International Regulations for Preventing Collisions at Sea
DOF	Degree of Freedom
DP	Dynamic Positioning
FOV	Field of View
LOS	Line-of-Sight
NSB	Null-Space-Based
ROV	Remotely Operated Vehicle
QP	Quadratic Programming
USV	Unmanned Surface Vehicle
UVMS	Underwater Vehicle Manipulator System
VO	Velocity Obstacle

AS	Asymptotically Stable
US	Uniformly Stable
UAS	Uniformly Asymptotically Stable
UGAS	Uniformly Globally Asymptotically Stable
ULES	Uniformly Locally Exponentially Stable
USGES	Uniformly Semi-Globally Exponentially Stable
UGES	Uniformly Globally Exponentially Stable

Preface

The work presented in this thesis is the result of my doctoral studies at the Department of Engineering Cybernetics (ITK) at the Norwegian University of Science and Technology (NTNU) from August 2013 to 2016 under the supervision of Professor Kristin Y. Pettersen (Dept. Engineering Cybernetics, NTNU) and co-supervisors Professor Jan Tommy Gravdahl (Dept. Engineering Cybernetics, NTNU), Professor Ingrid Schjølberg (Dept. Marine Technology, NTNU), Professor Hefeng Dong (Dept. Electronics and Telecommunications, NTNU) and Professor Gianluca Antonelli (Dept. Electrical and Information Engineering, University of Cassino and Southern Lazio). During this time, I have been funded by an NTNU SO scholarship from the IME faculty through the Lighthouse Robotics Project and by NTNU AMOS (Centre for Autonomous Marine Operations and Systems).

Acknowledgments

When given the chance to remain at ITK as a doctoral candidate after five wonderful years as a master student, I immediately jumped at the opportunity. My time in Trondheim and at this department has been exciting, challenging and very educational, both on a professional and a personal level, and I hope and believe I am a better person because of it.

As I am nearing the end of my doctoral studies, I would like to thank my main supervisor, Professor Kristin Y. Pettersen, for her support and encouragement, for her positive being, and for always having a new idea when an old one fails. She has found the perfect balance between providing help and guidance when needed and encouraging me to make my own decisions, for which I am truly grateful.

I would also like to thank all my co-supervisors for always being eager to help. In particular, Jan Tommy Gravdahl for giving great ideas and input and for fighting (and winning) to get me a position at ITK after completing my PhD, Ingrid Schjølberg for setting me on a path leading to my commencing doctoral studies and for being a great supervisor, and Gianluca Antonelli for hosting me at the University of Cassino and Southern Lazio with great hospitality and for his many and helpful ideas.

In addition to my team of official supervisors, Professor Andrew Teel at the University of California, Santa Barbara deserves to be thanked for a great and fruitful collaboration, as do Professor Thor Inge Fossen from the Department of Engineering Cybernetics at NTNU. I would also like to thank Professor Oussama

Khatib at the Artificial Intelligence Lab at Stanford University for hosting me for six months during my doctoral studies.

Furthermore, I have been lucky enough to be supported by the best administration, workshop and technical staff imaginable. Bente, Tove, Eva, Unni, Janne and Anne Sofie are always able to fix any problem, all the while taking the time to ask you how you are doing. Their smiles and "can-do" spirit have meant a great deal to me, both as a master student and a doctoral candidate. Stefano, Terje, Glenn and Daniel are always up for a challenge, and they deserve a special thanks for the amazing help they gave when I had the slightly crazy idea of making a coffee table as a Christmas present (it turned out really great).

My fellow PhD candidates and postdocs all deserve credit for being such great colleagues and friends. Coffee and lunch breaks have been spent having interesting, funny and at times extremely strange conversations with you, and I truly appreciate it. I would especially like to thank my officemates, Ida, Mikkel, Thomas and Anna, and the PhD girls for our semi-monthly gatherings with good food and great company.

I also want to thank my friends, Ingrid, Jenny, Katrine, Silje and Marie for countless taco-Thursdays, candy-induced stomach pains and hour-long phone calls, and Hilde Marie, Hedvik, Ylva, Marianne, Kristian and Øystein for being that amazing type of friend that you may only see twice a year, but when you do it feels like no time has passed at all. Kasper and Jakob, my colleagues and roommates in what has come to be known as "the danish asylum", deserve a thank you for great companionship, encouragement and for making Mondays the best day of the week with our weekly "Game of Thrones"-nights.

Finally, I would like to thank my loving family for always believing in me. My aunts and uncles, for encouragement and support. My grandparents, for always calling me the day before an exam or presentation, telling me not to be nervous because they will be nervous for me, and for watching cross-country skiing with me every winter. Lis and Henning, for always greeting me at the airport with great smiles and welcoming me into their home with open arms. My parents, for telling me I could accomplish whatever goal I set and for always being proud of me. And finally Kim, my soon to be husband, best friend and travel mate, for making me smile every single day. You mean the world to me.

Publications

Journal Papers

- S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen and J. Schrimpf, "Set-based Tasks within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework: General Formulation, Stability Analysis and Experimental Results" in Frontiers in Robotics and AI, Volume 3, Number 16, 2016.

Conference Papers

- S. Moe, K. Y. Pettersen, T. I. Fossen and J. T. Gravdahl, "Line-of-Sight Curved Path Following for Underactuated USVs and AUVs in the Horizontal Plane under the influence of Ocean Currents", in Proc. 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, June 21-24, 2016.
- S. Moe, K. Y. Pettersen, "Set-Based Line-of-Sight (LOS) Path Following with Collision Avoidance for Underactuated Unmanned Surface Vessel", in Proc. 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, June 21-24, 2016.
- S. Moe, A. R. Teel, G. Antonelli and K. Y. Pettersen, "Stability analysis for Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework", in Proc. 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, December 15-18, 2015.
- S. Moe, G. Antonelli, K. Y. Pettersen and J. Schrimpf, "Experimental Results for Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework", in Proc. 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, December 6 – 9, 2015.
- G. Antonelli, S. Moe and K. Y. Pettersen, "Incorporating set-based control within the singularity-robust multiple task-priority inverse kinematics", in Proc. 23rd Mediterranean Conference on Control and Automation (MED), Torremolinos, Spain, June 16-19, 2015.
- S. Moe, G. Antonelli and K. Y. Pettersen, "Null-space-based behavior guidance of planar dual-arm UVMS", in Proc. 2014 International Conference on Robotics and Biomimetics (ROBIO), Bali, Indonesia, Dec. 5-10, 2014.

- S. Moe, W. Caharija, K. Y. Pettersen and I. Schjølberg, "Path Following of Underactuated Underwater Vehicles in the Presence of Unknown Ocean Currents", in Proc. 33th International Conference on Ocean, Offshore and Arctic Engineering (OMAE), San Fransisco, CA, June 8-13, 2014.
- S. Moe, W. Caharija, K. Y. Pettersen and I. Schjølberg, "Path Following of Underactuated Marine Surface Vessels in the Presence of Unknown Ocean Currents", in Proc. 2014 American Control Conference (ACC), Portland, Oregon, June 4-6, 2014.

Other Publications, not part of this thesis

- S. Moe and I. Schjølberg, "Real-time Hand Guiding of Industrial Manipulator in 5DOF using Microsoft Kinect and accelerometer", in Proc. 22nd International Symposium and Robot and Human Interactive Communication (RO-MAN), Gyeongju, Korea, August 26-29, 2013.

Chapter 1

Introduction

1.1 Background and Motivation

The term “robot” was initially introduced by playwriter Karel Capek in a science fiction play in 1920, and stems from the Czech word for work; robota [86]. Although the robots in this play, namely artificial people constructed out of synthetic organic matter, are quite different from what most people associate with robots today, the phrase persevered and is now a common term for numerous mechanical devices with a wide range of applications.

The field of robotics as we know it today emerged as a part of the industrial revolution in the shape of manipulators (Figure 1.1(a)) along conveyor belts. Today, the term is applied to a great variety of mechanical devices, including teleoperators, unmanned marine and aerial vehicles and humanoids [2, 71, 86, 95]. Robots have even become a part of our everyday life in the form of autonomous vacuum cleaners and lawn mowers (Figure 1.1(b)). In this thesis, we consider a robot as a machine designed to execute one or more tasks automatically with speed, precision and repeatability. Possible applications are vast, ranging from space exploration to deep sea inspections. Robots are used for production and manufacturing, inspection and maintenance of underwater oil and gas structures, military purposes, environmental surveillance and archeology [2, 72]. They are even used for social purposes and elderly care [16] and as intelligent prosthetics [12].

The use of robots provides several advantages. In manufacturing, reduced labor costs, increased production and productivity and high precision and flexibility are compelling arguments. For instance, in the spraypaint industry, the best performing robots are able to transfer up to 90% of the paint to the spray surface, whereas a human manually spraypainting the surface achieves only around 30-40% transfer efficiency[1]. Teleoperators, i.e. remotely operated robots, can operate in environments that are dangerous or simply off limits to humans, such as under water, in space or in spaces where there is a risk of radiation.

Fully autonomous robotic systems are able to operate completely independent of humans, using a variety of sensors and communication systems to assess the situation and make decisions and changes during operation [75]. Today there are

---

[1] Øyvind A. Landsnes, ABB Bryne

over 800 000 industrial robots in operation world wide with varying levels of autonomy [86], ranging from constant repetition of a predefined motion to systems that are able to detect objects, avoid obstacles, map its surroundings, learn based on previous experience, etc. However, there is a strong desire to utilize robotic systems to completely automate processes also outside of the production industry [36].

This thesis focuses on the control of marine vehicles in particular, in addition to completely general robotic systems. Marine robotics has been an important branch of robotics since its beginning in the early 1970s [97], and in 2015 the world's commercial fleet consisted of 89 464 vessels [89]. Imagine the decrease in shipping costs if these vehicles were unmanned and able to safely navigate to their destination through changing environmental forces without colliding with other ships, not to mentioned the reduced risk of losing human lives at sea. Under the sea surface, the use of remotely operated vehicles (ROVs, see Figure 1.1(c)) and autonomous underwater vehicles (AUVs, see Figure 1.1(d)) is rapidly increasing for pipeline surveys, cable maintenance and other inspection and intervention work on underwater structures, in addition to fields such as marine biology, environmental monitoring, seafloor mapping, oceanography and military use [2]. ROVs are tethered to, powered by and operated from a surface vessel, making the operation costs high in addition to requiring an experienced human operator, and although companies like Bluefin Robotics and Kongsberg Maritime deliver AUVs to the offshore survey market, there are still challenges related to making them fully autonomous [24].

(a) The KR 270 R2700 ultra Industrial Robot, produced by KUKA.

(b) The Robomow lawn mower.

(c) The Spectrum ROV, produced by Oceaneering.

(d) The HUGIN AUV, produced by Kongsberg Maritime.

Figure 1.1: Examples of different robots.

1.2 Modeling and Control of Robotic Systems

The field of robotics involves elements from numerous different engineering directions, e.g. mechanical, electrical, control, computer science, and applications [86]. This thesis will mostly focus on the control aspect, although Chapter 6 also presents implementation and gives examples of possible applications of the proposed control theory.

The mechanical structure of a robot manipulator consists of a series of rigid bodies (links) interconnected by means of articulations (revolute and prismatic joints). Typically, the manipulator has an arm that accounts for most of the movement, a wrist that ensures dexterity, and an end effector that performs the task required of the robot [81].

A complete model of a robotic system consists of the kinematics and the dynamics. Kinematics describe the motion of a robotic system without considering the forces and torques that cause the motion. Since robotic mechanisms are designed for motion, kinematics is the most fundamental aspect of robot design, analysis, control, and simulation, and the robotics research community has focused on efficiently applying different representations of position and orientation and their derivatives to solve foundational kinematics problems [80]. The kinematics can be derived by use of the Denavit-Hartenberg convention [86]. The dynamic equations of motion describe the relationship between actuation and contact forces and the resulting acceleration and motion trajectories. These equations may be derived using for instance Lagrange's, Newton's and Euler's equations of motion for a rigid body and can be expressed both in joint and operational space [26, 80].

Traditionally, robotic systems are controlled in their joint space. A typical robotic control system for joint space control is illustrated in Figure 1.2. The kinematic controller calculates reference states based on the desired behavior, often specified in operational space, and the current state of the system. The reference states are the input of the dynamic controller, which calculates and imposes forces and torques on the actual system through the actuators. There exists a variety of controller types that may be applied, ranging from model-based feedback linearizing controllers to robust sliding mode, adaptive and standard PID-controllers [81]. Note that in kinematic control it is common to assume that the reference state is tracked and simply use this as feedback to the kinematic controller rather than the actual state [3]. This is illustrated by the dashed arrow in Figure 1.2.

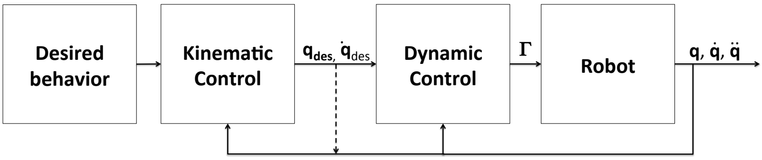


Figure 1.2: A typical control structure for a robotic system with joint configuration $\boldsymbol{q}$ and joint velocities and accelerations $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$. Based on the desired behavior and the current state, the kinematic controller calculates reference states and the dynamic controller calculates actuator forces and torques $\boldsymbol{\Gamma}$ which are imposed on the actual system.

Note that operational space control [45] is an alternative to joint space control. In this control scheme, the measured joint space variables can be transformed into the corresponding operational space variables through direct kinematics, thus removing the need for computationally heavy inverse kinematics calculations. This approach works directly with forces and torques, and is thus highly suitable for robotic systems that interact with the environment such as walking or grasping robots, but not for non-intervention marine vehicles, which in addition have a very simple kinematics. Furthermore, all operational space control schemes result in considerable computational requirements which are somewhat equal to inverse kinematics functions [81]. For these reasons, in this thesis we will focus on joint space control, and on kinematic control in particular for general robotic systems.

1.2.1 Kinematic Control

Robotic systems may be required to perform one or several tasks which are often given in the operational space, for instance obtaining a certain desired end effector position and/or orientation. Hence, a variety of inverse kinematics algorithms have been developed to map tasks from the operational space to the joint space and thus generate reference trajectories for the controllers, all the while being able to handle singularities and non-square matrices. The most common approach is to use a Jacobian-based method [18, 27, 69], such as the Jacobian transpose, damped least squares or pseudo-inverse. The Jacobian matrix defines the connection between joint space velocities and operational space velocities and is an essential part of kinematic control. Examples of various tasks and their corresponding kinematics and Jacobian matrices for the underwater case are given in [2].

These different approaches have various pros and cons. Using the transpose as an approximation of the inverse of the Jacobian matrix is not accurate, but computationally very efficient. This approach was widely used when the available computational power was lower than it is today [10, 93]. The damped least squares-approach is defined for systems that are not of full rank and non-square systems. Furthermore, the damping effect ensures good behavior near singularities by preventing the desired joint velocities from becoming too large [92]. However, the damped least-squares solution lacks the nice, mathematical properties of the pseudo-inverse, which is defined by four criteria [76].

In particular, the pseudo-inverse Jacobian is, like the damped least-squares solution, defined for systems that are not square nor have full rank and is a widely used solution to the inverse kinematics problem [17, 47, 80]. Furthermore, for full-rank systems, the Jacobian times the corresponding pseudo-inverse is equal to the identity matrix, and for all systems it is trivial to calculate a null-space matrix which may be used to achieve secondary tasks. This attribute is essential in Part I of this thesis.

A robotic system is said to be kinematically redundant if it possesses more degrees of freedom (DOFs) than those required to perform a certain task [81]. In this case, the “excess” DOFs can be utilized in order to perform several tasks using null-space-based (NSB) behavioral control. Furthermore, it is useful to sort tasks in a prioritized order to handle potentially conflicting tasks. Initial results solves the problem for two tasks with open loop control [39, 56]. However, the proposed approach is prone to algorithmic singularities, and the open loop control does not guarantee convergence of the tasks. An algorithmic singularity can arise if two tasks are linearly dependent and in conflict, in which case the secondary task asks for very high joint velocities. This problem is solved for open loop control in [23] and expanded to a general number of tasks with closed-loop control in [3, 6]. It is proven that the task errors converge to zero given that certain, specified assumptions are satisfied. This framework is known as the singularity-robust multiple task-priority inverse kinematics framework, and the approach may be utilized on a number of different robotic systems for a wide range of applications [5, 56, 79, 87, 88].

The singularity-robust multiple task-priority inverse kinematics framework, which is described in more detail in Chapter 2, has been developed for *equality tasks*. Equality tasks specify exactly one desired value for given states of the system, for instance the position and orientation of the end effector. However, for a general robotic system, several goals may not be described as equality tasks, but rather as *set-based tasks*, which are tasks that have a desired interval of values rather than one exact desired value. Such tasks are also referred to as inequality constraints. Examples of such tasks are staying within joint limits [59], collision/obstacle avoidance [39] and field of view (FOV). As recognized in [42], the multiple task-priority inverse kinematics algorithm is not suitable to handle set-based tasks directly, and these tasks are therefore usually transformed into more and unnecessary restrictive equality constraints through potential fields or cost functions [30, 45].

Set-based Tasks

An approach to systematically include set-based tasks in a prioritized task regulation framework is proposed in [42] and further improved in [29]. To handle the set-based tasks, the algorithms in [29, 42] transform the inverse kinematics problem into a quadratic programming (QP) problem. The optimization problem is solved through an iterative algorithm which may be computationally heavy and slow. Furthermore, by defining the problem as a QP problem, the tasks can not be directly included in the multiple task-priority inverse kinematics framework. In [82], set-based tasks are handled by resorting to proper activation and regularization functions, but no analytical framework to prove the convergence/satisfaction of the tasks is provided. In [57, 58] set-based tasks are considered in a prioritized order and an algorithm is developed to ensure a smooth control law when (de)activating a set-based task. However, during transitions, the strict priority of the tasks is lost.

In [25], set-based tasks can be considered only as high priority tasks, and in [9] the inequality constrains are transformed to an equality constraint by the use of proper slack variables. The optimization problem is then modified in order to minimize the defined slack variables together with the task errors in a task-priority architecture.

Set-based tasks may also be embedded in the control problem by assigning virtual forces that push the robot away from the set boundaries. This idea was first proposed in [44] and then used in several following approaches. However, when resorting to virtual forces/potentials fields, satisfaction of the boundaries can not be analytically guaranteed and often the overall control architecture may be experience the considerable drawbacks described in for instance [6, 48], such as oscillating behavior.

In this thesis, the singularity-robust multiple task-priority inverse kinematics framework is extended to systematically handle set-based tasks in addition to equality tasks, thereby expanding an already generic, widely used method to handle all types of tasks in a consistent, unified manner. Note that an approach similar to this framework may also be applied for operational space control to simultaneously achieve multiple tasks in a prioritized order. Hence, the results presented in this thesis may be extended also to operational space control.

1.3 Modeling and Control of Marine Vehicles

As mentioned, unmanned marine vehicles may be considered as a type of robotic system, and the equations of motion consist of kinematics and dynamics. However, these crafts pose some additional challenges due to hydrodynamic effects, underactuation, thrust allocation, communication and environmental influence [31]. Even so, a typical control system for a marine vehicle has the same structure as that of a robotic system (Figure 1.2), where kinematic control is referred to as guidance. Furthermore, navigation, i.e. the transformation from sensor output and navigation data to state estimation, is essential for control of marine vehicles. This thesis focuses on the guidance and control part, and assumes that the navigation system provides correct estimates of the vehicle states.

The dynamics of marine vehicles include hydrodynamic damping, gravity and buoyancy effects, mass and inertia effects of the rigid body, added mass due to the movement through fluid, Coriolis and centripetal forces and actuator forces. Unlike a traditional industrial manipulator, a marine vehicle is typically underactuated, meaning it has fewer control inputs than it has DOFs. For example, the state of a surface vessel is given by a two-dimensional position and a heading, i.e. three DOF, and it typically has 2 control inputs, e.g. a propeller and a rudder. This must be accounted for by the guidance and control system to achieve the desired behavior.

A marine vehicle may have several objectives. The most common are dynamic positioning (DP), target tracking, trajectory tracking and path following [31]. In addition, additional tasks or motivations may be considered, such as collision avoidance and fuel/energy consumption.

The goal of DP is to keep the vessel position and heading constant, which is useful for mobile offshore drilling units, research vessels, cruise ships etc. Target tracking aims to track a moving object for which no future motion information is available, and is applicable for for instance missiles.

The objective of trajectory tracking systems is to track a predefined path with a temporal constraint, i.e. the vessel should be on a specific point of the path at a specific time. Path following corresponds to trajectory tracking without the time demand: the vessel should simply converge to and follow a specified path. In some cases the vessel should move along the path at a desired speed, and the control problem is then a maneuvering control problem. In this thesis, the main focus is on path following and maneuvering control of marine vehicles in the presence of ocean currents and collision avoidance.

1.3.1 Path Following in the Presence of Ocean Currents

A commonly used approach for path following and maneuvering control is the line-of-sight (LOS) method [28, 31, 40, 51, 52, 77, 83]. LOS is a guidance method providing attitude references to the marine vehicle that, if satisfied, will ensure convergence to the desired path. However, this control approach is susceptible to environmental disturbances such as ocean currents, waves and wind: path deviation and convergence problems will occur if the vessel is affected by environmental disturbances [15]. For straight line paths, LOS can be used with integral action [15, 20, 21] or adaptive estimation techniques [52] by allowing the vehicle to side-slip to compensate for the effect of the ocean current. In [20] the steady state of the vessel is used to estimate the ocean current magnitude and direction, and in [52] the estimates of the ocean current is used directly in the LOS guidance law.

For curved path following, the vehicle has to side-slip even without ocean currents to achieve path following due to the non-zero curvature of the path. Rigorous stability properties for LOS with side-slip used on a general, curved path in the plane when ocean current is not considered is given in [33]. The side-slip is calculated based on velocity measurements. A similar approach is used in [13], where curved path following of general paths for both surface and underwater vehicles is achieved by analyzing Serret-Frenet equations. In [98], an approach to achieve a spiraling motion for an underwater glider is presented. Experimental results show that the results are applicable also when ocean currents influences the glider, but the current must be known beforehand. A glider differs from standard AUVs in the sense that it has a single internal movable and rotatable mass to steer the motion.

Side-slipping is required both for curved path following and for ocean current compensation, and thus also in the case when these two scenarios are combined. This thesis presents two approaches to achieve this, where one is based on absolute velocity measurements [68], and the other relies on relative velocity measurements [63, 64]. A similar approach as in [63] was suggested later on [32], where a LOS path following approach is used with current estimation techniques and measurements of relative velocities, allowing the vehicle to follow curved paths while compensating for the influence of ocean currents.

1.3.2 Collision Avoidance

In nautical navigation, all surface vessels are subject to COLREGs [49], and unmanned surface vessels (USVs) must be able to abide by these regulations without humans in the control loop. There exists several path planning algorithms to compute a safe path to avoid obstacles, such as A*, RRT and HBug [41]. However, these global path planning methods are not suitable for unknown and dynamic environments, and must be complemented by a local guidance system that may is able to avoid small, unforeseen and dynamic obstacles while following the global path. A variety of such local approaches have been proposed, both for the general and maritime case, such as potential fields [44], dynamic window [34, 54] and velocity obstacles [49, 90]. However, these approaches have several drawbacks. Potential fields may suffer from oscillating behavior and other limitations [48]. The dynamic window approach assumes no sideways velocity, and is therefore not suitable for marine vehicles since these are able to and will glide sideways while moving through water. Furthermore, the dynamic window approach can be computationally heavy, but it has the advantage that it can easily be modified to comply with COLREGs [54]. Finally, the velocity obstacle (VO) approach has good mathematical qualities, is computationally simple and does easily comply with COLREGs. However, it assumes linear (i.e. non-rotational) and constant velocities and is not straight-forward to implement. Furthermore, it is not obvious how to combine these collision avoidance methods with existing guidance methods such as LOS.

In [96], the proposed guidance system switches between a path following task and a collision avoidance task. Only stationary obstacles are considered, making the method insufficient for avoiding collisions with other vessels. Furthermore, the switching between path following and collision avoidance is given as a smooth transition function dependent on the distance between the USV and the obstacle center, which results in a potential field approach with known drawbacks as discussed above.

1.4 Thesis Outline and Contributions

This thesis is divided into three main parts: Part I, which consists of Chapter 3-6, describes a method for incorporating set-based tasks into the Singularity-robust Multiple Task-priority Inverse Kinematics Framework using switched systems theory, whereas Part II considers curved path following of marine vehicles in the presence of ocean currents. This is covered by Chapter 7-9. Finally, Part III combines results from Part I and II in the form of a set-based path following system for surface vessels to avoid static and dynamic obstacles, which is presented in Chapter 10.

This chapter is followed by a brief introduction to the notation, mathematical preliminaries and models used throughout this thesis, which is given in Chapter 2.

In the following, the topic and the contributions of each chapter are presented.

Part I - Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework

Chapter 3

Topic: The widely used singularity-robust multiple task-priority inverse kinematics framework is applied to a dual-arm underwater vehicle manipulator system (UVMS).

Contribution: This chapter proposes to use the singularity-robust multiple task-priority inverse kinematics scheme [3] for a dual-arm UVMS, which calculates references for both the manipulator arms and the vehicle base itself, in order to implement several concurrent tasks. A UVMS may be considered as a manipulator arm mounted on a floating base, and a two-manipulator system can use the two arms to cooperate and thereby perform more complex tasks and pick up larger/heavier objects. A lot of research has been done on fixed dual-arm systems regarding coordinated and cooperative control, leader/follower control, force control, collision detection and avoidance etc. [19, 91], but little has been done for a two-manipulator floating base system. A variety of tasks can be implemented and included in a prioritized order. Furthermore, it is proposed to consider one manipulator and the vehicle base as a leader and the second manipulator as the follower. This division ensures that conflicting tasks do not attempt to move the vehicle base in different directions and that the tasks are achieved using the desired DOFs. An example scenario in which the framework can be applied is presented.

Chapter 4

Topic: We extend the singularity-robust multiple task-priority inverse kinematics framework to incorporate high-priority set-based tasks. The chapter includes a stability analysis and simulation results.

Contribution: The singularity-robust multiple task-priority inverse kinematics framework, which is described in detail in Section 2.5, allows equality tasks to be considered in a prioritized order by projecting task velocities through the null-spaces of higher priority tasks. Set-based tasks, i.e. tasks with a range of valid values, are normally handled through over-constraining QP-problems [29], artificial potential fields which make a stability analysis infeasible [45], or smooth switched systems in which the strict task priority is lost during transitions [58]. This chapter extends the singularity-robust multiple task-priority inverse kinematics framework to handle high-priority set-based tasks in addition to equality tasks. The proposed method is proven to ensure asymptotic convergence of the equality task errors and the satisfaction of all high-priority set-based tasks. Furthermore, the priority of the tasks are maintained at all times.

Chapter 5

Topic: The results of Chapter 4 are extended to include low-priority and a combination of high- and low-priority set-based tasks in the singularity-robust multiple task-priority inverse kinematics framework. The chapter includes a stability analysis and simulation results.

Contribution: This chapter builds on Chapter 4 and presents a similar analysis for the case of low-priority set-based tasks, which are tasks that have priority after at least one equality task, and for a combination of high- and low-priority set-based tasks. The proposed method is proven to ensure asymptotic convergence of the equality task errors and the satisfaction of all high-priority set-based tasks. Low-priority set-based tasks can not be guaranteed to be satisfied at all times due to the influence of the higher priority equality tasks. However, if the higher priority equality task errors are zero, the low-priority set-based tasks will also be satisfied.

Chapter 6

Topic: A practical implementation of the results in Chapter 4 and 5 is presented along with experimental results that confirm the correctness of the theory and the effectiveness of the method.

Contribution: This chapter presents the proposed implementation of the results from Chapter 4 and 5 and discusses the resulting computational load. Experiments were run on a six DOF UR5 manipulator. Several set-based tasks are implemented, including collision avoidance, field of view, and limited workspace, and the experimental results confirm the previously presented theory.

Part II - Curved Path Following of Marine Vehicles in the Presence of Unknown Ocean Currents

Chapter 7

Topic: In this chapter, an existing curved path following method [13] for surface vehicles is presented in order to achieve convergence to the path also in the presence of unknown ocean currents. The proposed method is suitable for USVs that are equipped with sensors to measure relative velocities.

Contribution: This chapter extends the curved path following method in [13] to obtain path following of USVs also under the influence of unknown ocean currents. This is achieved by expanding the guidance law in [13] and combining it with an ocean current observer described in [1], allowing the USV to side-slip to compensate both for the curvature of the path and the current effects. The closed-loop system includes feedback linearizing controllers, and is dependent on relative velocity measurements. The path is represented by a Serret-Frenet reference frame. The results are also applicable for AUVs restricted to the horizontal plane.

Chapter 8

Topic: In this chapter, an existing curved path following method [13] for AUVs is presented in order to achieve convergence to the path also when ocean currents affect the vehicle. The proposed method is suitable for vehicles that are equipped with sensors to measure relative velocities.

Contribution: This chapter extends the result of Chapter 7 to AUVs and is motivated by [13], in which curved path following is achieved for underwater vehicles when no ocean currents affect the vehicle. The path is represented by a Serret-Frenet reference frame, and the key components of the closed-loop system are guidance laws, which serve as input to the control system, an ocean current observer and an update law to drive the reference frame along the path. As in Chapter 7, the AUV side-slips to compensate for the combined effect of the curvature of the path and the ocean current.

Chapter 9

Topic: A guidance and control system is developed to achieve curved path following for USVs under the influence of ocean currents. The proposed method relies on absolute velocity measurements.

Contribution: This chapter proves that the LOS guidance law suggested in [33] for curved path following is suitable also for curved path following in combination with ocean current compensation by using adaptive feedback linearization combined with sliding mode. The proposed guidance and control system requires measurements of absolute velocities only, and is therefore suitable when the vehicle in question lacks sensors to measure relative velocities. Furthermore, the guidance law is not based on Serret-Frenet frames, which allows for any parametrization of the path and removes the need for an update law for the Serret-Frenet frame and an ocean current observer. The results are applicable for USVs and AUVs restricted to the horizontal plane.

Part III - A Set-based Approach for Path Following of Marine Vehicles with Collision Avoidance

Chapter 10

Topic: In this chapter, we consider a USV whose objectives are path following and collision avoidance. Results from Part I are adapted to the underactuated USV and a specific guidance law for obstacle avoidance is presented.

Contribution: The contribution of this chapter is twofold. Firstly, it is suggested to apply the set-based theory presented in Chapter 4-6 to satisfy two objectives: Collision avoidance and path following for an unmanned surface vehicle. The approach defines collision avoidance as a high-priority set-based task and path following as a low-priority equality task, and is adapted to the underactuated USV by switching between two predefined guidance laws rather than combining them using the Null-Space-Based Inverse Kinematics approach for fully actuated systems as in Part I of this thesis. The guidance laws, if satisfied, will ensure path following and collision avoidance respectively. Thus the system is equipped with one path following mode and one collision avoidance mode, in addition to a defined and deterministic method for switching between these two. This method can be used for any combination of path following and collision avoidance guidance laws, making it a highly generic solution. Secondly, a specific LOS-based guidance law for collision avoidance is suggested. This guidance law, if satisfied, will ensure that the USV tracks a circle with constant radius about the obstacle center, which may be stationary or moving, and is specifically designed to assure collision avoidance while abiding by the COLREGs. The guidance system is especially suitable to avoid collisions with small and dynamic obstacles, and can easily be combined with existing global path planning methods that handle topography [41].

Chapter 2

Mathematical Preliminaries and Models

This chapter presents notation, stability theory and relevant background theory that are of relevance to the analysis and results presented in this thesis.

2.1 Notation

In this thesis, the space $\mathbb{R}^n$ denotes the Euclidian space of dimension n, and vectors and matrices are expressed in bold face.

Rotations are described using the Euler angle representation ϕ, θ and ψ, where the transformation from reference frame $\{a\}$ to $\{b\}$ is given as

$$\boldsymbol{R}_a^b(\boldsymbol{\Theta}_{ba}) = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) \\ -s(y) & c(y)s(\phi) & c(y)c(\phi) \end{bmatrix}, \tag{2.1}$$

where $s(\cdot) \triangleq \sin(\cdot)$, $c(\cdot) \triangleq \cos(\cdot)$ and $\boldsymbol{\Theta}_{ba} = [\phi, \theta, \psi]^T$ is the roll, pitch and yaw angle, respectively. The rotation matrix has the following properties [31].

$$\det(\boldsymbol{R}_a^b(\boldsymbol{\Theta}_{ba})) = 1, \tag{2.2}$$

$$\boldsymbol{R}_a^b(\boldsymbol{\Theta}_{ba})\boldsymbol{R}_a^b(\boldsymbol{\Theta}_{ba})^T = \boldsymbol{R}_a^b(\boldsymbol{\Theta}_{ba})^T\boldsymbol{R}_a^b(\boldsymbol{\Theta}_{ba}) = \boldsymbol{I}_{3\times 3}, \tag{2.3}$$

$$\boldsymbol{R}_b^a(\boldsymbol{\Theta}_{ba}) = \left[\boldsymbol{R}_a^b(\boldsymbol{\Theta}_{ba})\right]^{-1} = \boldsymbol{R}_a^b(\boldsymbol{\Theta}_{ba})^T, \tag{2.4}$$

$$\boldsymbol{R}_a^c(\boldsymbol{\Theta}_{ca}) = \boldsymbol{R}_b^c(\boldsymbol{\Theta}_{cb})\boldsymbol{R}_a^b(\boldsymbol{\Theta}_{ba}). \tag{2.5}$$

The pseudo-inverse of a matrix $\boldsymbol{J}$ is denoted $\boldsymbol{J}^\dagger$.

2.2 Trigonometric Formulas

The addition formula for arctan-terms states that

$$\arctan(x) + \arctan(y) = \arctan\left(\frac{x+y}{1-xy}\right). \tag{2.6}$$

Furthermore, the sine and cosine of an arctan term is defined as

$$\begin{aligned}\alpha &= \arctan\left(\frac{a}{b}\right),\\ \cos(\alpha) &= \cos\left(\arctan\left(\frac{a}{b}\right)\right)\\ &= \frac{b}{\sqrt{a^2+b^2}},\\ \sin(\alpha) &= \sin\left(\arctan\left(\frac{a}{b}\right)\right)\\ &= \frac{a}{\sqrt{a^2+b^2}}.\end{aligned} \tag{2.7}$$

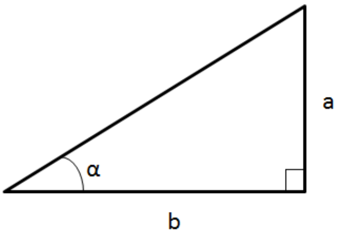


Figure 2.1: Illustration of (2.7).

2.3 Stability Theory

2.3.1 General Stability

Consider the non-autonomous system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^n \tag{2.8}$$

The vector function $\boldsymbol{f}$ is assumed to be piecewise continuous in t and locally Lipschitz in $\boldsymbol{x} \in \mathbb{R}^n$. The equilibrium point $\boldsymbol{x} = \boldsymbol{0}$ of the system (2.8) is

Stable if for each $\epsilon > 0$, there exists a $\delta(\epsilon, t_0) > 0$ such that $||\boldsymbol{x}(t_0)|| < \delta \Rightarrow ||\boldsymbol{x}(t)|| < \epsilon, \forall\, t \geq t_0 \geq 0$.

Uniformly stable (US) if for each $\epsilon > 0$, there exists a $\delta(\epsilon) > 0$ such that $||\boldsymbol{x}(t_0)|| < \delta \Rightarrow ||\boldsymbol{x}(t)|| < \epsilon, \forall\, t \geq t_0 \geq 0$.

Unstable if it is not stable.

Asymptotically stable (AS) if it is stable and there is a positive constant $c = c(t_0)$ such that $\boldsymbol{x}(t) \to 0$ as $t \to \infty, \forall\, ||\boldsymbol{x}(t_0)|| < c$.

Uniformly asymptotically stable (UAS) if it is uniformly stable and there is a positive constant c, independent of t_0, such that $\forall\ ||\boldsymbol{x}(t_0)|| < c$, $\boldsymbol{x}(t) \to 0$ as $t \to \infty$, uniformly in t_0.

Uniformly globally asymptotically stable (UGAS) if it is uniformly stable, $\delta(\epsilon)$ can be chosen to satisfy $\lim_{\epsilon\to\infty} \delta(\epsilon) = \infty$, and, for each pair of positive numbers η and c, there is a $T = T(\eta, c) > 0$ such that $||\boldsymbol{x}(t)|| < \eta, \forall\ t \geq t_0 + T(\eta, c), \forall\ ||\boldsymbol{x}(t_0)|| < c$.

Uniformly locally exponentially stable (ULES) if there exist constants γ_1, γ_2 and $r > 0$ such that for all $(t_0, \boldsymbol{x}(t_0)) \in \mathbb{R}_{\geq 0} \times B_r$, $||\boldsymbol{x}(t)|| \leq \gamma_1 ||\boldsymbol{x}(t_0)|| e^{-\gamma_2(t-t_0)}$ $\forall\ t \geq t_0$, where B_r is the open ball $B_r := \{\boldsymbol{x} \in \mathbb{R}^n : ||\boldsymbol{x}|| < r\}$.

Uniformly semi-global exponentially stable (USGES) if for each $r > 0$ there exist constants γ_1, $\gamma_2 > 0$ such that $(t_0, \boldsymbol{x}(t_0)) \in \mathbb{R}_{\geq 0} \times B_r$, $||\boldsymbol{x}(t)|| \leq \gamma_1 ||\boldsymbol{x}(t_0)|| e^{-\gamma_2(t-t_0)}\ \ \forall\ t \geq t_0$.

Uniformly globally exponentially stable (UGES) if there exist constants γ_1, $\gamma_2 > 0$ such that for all $(t_0, \boldsymbol{x}(t_0)) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^n$, $||\boldsymbol{x}(t)|| \leq \gamma_1 ||\boldsymbol{x}(t_0)|| e^{-\gamma_2(t-t_0)}\ \ \forall\ t \geq t_0$.

κ-exponentially stable if there exists a neighborhood $\Omega_p \subset \mathbb{R}^n$, a positive constant λ, and a function $h(\cdot)$ of class κ such that all solutions of $\boldsymbol{x}(t)$ satisfy $||\boldsymbol{x}(t)|| \leq h(||\boldsymbol{x}(t_0)||) e^{-\lambda(t-t_0)}\ \ \ \forall\ t \geq t_0, \forall\ ||\boldsymbol{x}(t_0)|| \in \Omega_p$.

Globally κ-exponentially stable if it is κ-exponentially stable with $\Omega_p = \mathbb{R}^n$.

Definitions from [43], [50] and [85].

2.3.2 Hurwitz matrix

Theorem 4.5 [43] : Consider the linear system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}. \tag{2.9}$$

The matrix $\boldsymbol{A}$ is Hurwitz if and only if all eigenvalues λ_i of $\boldsymbol{A}$ satisfy $Re(\lambda_i) < 0$. If $\boldsymbol{A}$ is Hurwitz, the equilibrium point $\boldsymbol{x}^* = \boldsymbol{0}$ is globally asymptotically (exponentially) stable.

Theorem 4.6 [43] : Consider a quadratic Lyapunov function candidate $V(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}$ where $\boldsymbol{P} = \boldsymbol{P}^T > 0$ is a real symmetric positive definite matrix. The derivative of V along the trajectories of the linear system in Equation (2.9) is given by

$$\dot{V}(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{P}\dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}^T\boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}^T(\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P})\boldsymbol{x} = -\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}, \tag{2.10}$$

where $\boldsymbol{Q}$ is a symmetric matrix defined by $\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} = -\boldsymbol{Q}$.

The matrix $\boldsymbol{A}$ is Hurwitz if and only if for any given positive definite symmetric matrix $\boldsymbol{Q}$ there exists a positive definite symmetric matrix $\boldsymbol{P}$ that satisfies $\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} = -\boldsymbol{Q}$. Moreover, if $\boldsymbol{A}$ is Hurwitz, then $\boldsymbol{P}$ is the unique solution of $\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} = -\boldsymbol{Q}$.

2.3.3 Lyapunov stability

Stability of the system in Equation (2.8) can be proven using Lyapunov's direct method [43]. This is done by considering Lyapunov function candidates, often denoted V. A time-varying function $V(t, \boldsymbol{x})$ is said to be

Positive definite if $V(t, \boldsymbol{x}) \geq W_1(\boldsymbol{x})$ for some positive definite $W_1(\boldsymbol{x})$ which satisfies $W_1(\boldsymbol{0}) = 0$ and $W_1(\boldsymbol{x}) > 0$ for $\boldsymbol{x} \neq \boldsymbol{0}$. Similarly, $V(t, \boldsymbol{x})$ is positive semidefinite and radially unbounded if $W_1(\boldsymbol{x})$ is positive semidefinite and radially unbounded.

Decrescent if $V(t, \boldsymbol{x}) \leq W_2(\boldsymbol{x})$ for all $t \geq 0$ for some positive definite $W_2(\boldsymbol{x})$.

Theorem 4.8 and 4.9 [43] : The equilibrium point $\boldsymbol{x}^* = 0$ is

	Stable	US	UAS	UGAS
V	Positive definite	Positive definite, decrescent	Positive definite, decrescent	Positive definite, decrescent, radially unbounded
$\dot{V}$	Negative semidefinite	Negative semidefinite	Negative definite	Negative definite

Theorem 4.10 [43] : Let $\boldsymbol{x}^* = 0$ be an equilibrium point for the system in (2.8) and $\mathbb{D} \subset \mathbb{R}^n$ be a domain containing $\boldsymbol{x}^*$. Let $V : [0, \infty) \times \mathbb{D} \to \mathbb{R}$ be a continuously differentiable function such that

$$k_1 ||\boldsymbol{x}||^a \leq V(t, x) \leq k_2 ||\boldsymbol{x}||^a, \tag{2.11}$$

$$\frac{\delta V}{\delta t} + \frac{\delta V}{\delta x} \boldsymbol{f}(t, \boldsymbol{x}) \leq -k_3 ||\boldsymbol{x}||^a, \tag{2.12}$$

$\forall\ t \geq 0$ and $\forall\ \boldsymbol{x} \in \mathbb{D}$, where k_1, k_2, k_3 and a are positive constants. Then, $\boldsymbol{x}^*$ is exponentially stable. If the assumptions hold globally, then $\boldsymbol{x}^*$ is globally exponentially stable.

2.3.4 Stability of Cascades

Consider the nonlinear time-varying cascaded system

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}_1(t, \boldsymbol{x}) + \boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{y})\boldsymbol{y}, \\ \dot{\boldsymbol{y}} &= \boldsymbol{f}_2(t, \boldsymbol{y}), \end{aligned} \tag{2.13}$$

where $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are continuously differentiable in their arguments.

Lemma 2.2 [74]: The cascaded system in equation (2.13) is UGAS if both $\dot{\boldsymbol{x}} = \boldsymbol{f}_1(t, \boldsymbol{x})$ and $\dot{\boldsymbol{y}} = \boldsymbol{f}_2(t, \boldsymbol{y})$ are UGAS and the solutions of (2.13) are globally, uniformly bounded.

Theorem 2 [73]: The cascaded system in Equation (2.13) is UGAS if the following assumptions are satisfied:

- ***A1*** - The system $\dot{\boldsymbol{x}} = \boldsymbol{f}_1(t, \boldsymbol{x})$ is UGAS with a Lyapunov function satisfying $||\frac{\delta V}{\delta \boldsymbol{x}}|| \, ||\boldsymbol{x}|| \leq c_1 V(t, \boldsymbol{x}) \quad \forall \, ||\boldsymbol{x}|| \geq \eta$, where $c_1 > 0$ and $\eta > 0$.
- ***A2*** - The function $\boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{y})$ satisfies $||\boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{y})|| \leq \theta_1(||\boldsymbol{y}||) + \theta_2(||\boldsymbol{y}||)||\boldsymbol{x}||$, where $\theta_1, \theta_2 : \mathbb{R}^+ \to \mathbb{R}^+$ are continuous.
- ***A3*** - The system $\dot{\boldsymbol{y}} = \boldsymbol{f}_2(t, \boldsymbol{y})$ is UGAS and for all $t_0 \geq 0$, $\int_{t_0}^{\infty} ||\boldsymbol{y}(s, t_0, \boldsymbol{y}(t_0))||ds \leq \phi(||\boldsymbol{y}(t_0)||)$, where the function $\phi(\cdot)$ is a class $\mathcal{K}$ function.

Lemma 8 [73]: If, in addition to the assumptions in Theorem 2 [73], both $\dot{\boldsymbol{x}} = f_1(t, \boldsymbol{x})$ and $\dot{\boldsymbol{y}} = \boldsymbol{f}_2(t, \boldsymbol{y})$ are globally $\mathcal{K}$-exponentially stable, then the cascaded system in equation (2.13) is globally $\mathcal{K}$-exponentially stable.

2.4 Modeling of Marine Surface Vessels and Underwater Vehicles

This section introduces theory on modeling of marine vehicles and presents the simulation models that have been used in the work presented in Part II and III.

2.4.1 Marine Surface Vessels

In general, the equations of motion of a USV under the influence of ocean currents can be described with 3 DOFs. The model is given in (2.14) [31]:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{R}_b^i(\boldsymbol{\Theta}_{ib})\boldsymbol{\nu}_r + \boldsymbol{V}_c, \\ \boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{M}_A\dot{\boldsymbol{\nu}}_{\boldsymbol{r}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r &= \boldsymbol{B}\boldsymbol{f}, \end{aligned} \tag{2.14}$$

where $\boldsymbol{\eta} = [x, y, \psi]^T$ is the position and heading in the inertial frame $\{i\}$, $\boldsymbol{\nu} = [u, v, r]^T$ is the surge velocity, sway velocity and yaw rate in body frame $\{b\}$ and $\boldsymbol{\Theta}_{ib} = [0, 0, \psi]^T$. Furthermore, $\boldsymbol{V}_c = [V_x, V_y, 0]^T$ is the ocean current in the inertial frame and $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{R}^T(\psi)\boldsymbol{V}_c = [u_r, v_r, r]^T$ is the body velocities relative to the ocean current. The rotation matrix $\boldsymbol{R}_b^i(\boldsymbol{\Theta}_{ib})$ is defined in (2.1) and simplifies to

$$\boldsymbol{R}_b^i(\psi) \triangleq \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.15}$$

and the controlled input $\boldsymbol{f} = [T, \delta]^T$ where T is the propeller force and δ is the rudder angle. The matrix $\boldsymbol{M}_{RB} = \boldsymbol{M}_{RB}^T > 0$ is the rigid-body mass and inertia matrix and $\boldsymbol{C}_{RB}$ is the rigid-body Coriolis and centripetal matrix. Similarly, $\boldsymbol{M}_A = \boldsymbol{M}_A^T > 0$ and $\boldsymbol{C}_A$ are mass and Coriolis matrices for hydrodynamic added mass. The strictly positive hydrodynamic matrix is given by $\boldsymbol{D}$ and $\boldsymbol{B} \in \mathbb{R}^{3\times 2}$ is the actuator configuration matrix.

Assumption 2.1: The USV is port-starboard symmetric.

Assumption 2.2: The body-fixed coordinate frame b is located at a distance $(x_g^\star, 0)$ from the USV's center of gravity (CG) along the center-line of the USV, where $x_g^\star$ is to be defined later.

Remark 2.1*:* The body-fixed coordinate system can always be translated to the required location $x_g^\star$ [31].

Assumption 2.3: The ocean current in the inertial frame $\boldsymbol{V}_c \triangleq [V_x, V_y, 0]^T$ is constant, irrotational and bounded. Hence there exists a constant $V_{\max} > 0$ such that $V_{\max} > \sqrt{V_x^2 + V_y^2}$.

Remark 2.2*:* Note that the model (2.14) does not depend on wave frequency. Hence, the parameters in $\boldsymbol{M}_A$ and $\boldsymbol{D}$ can be considered constant.

Remark 2.3*:* It is shown in [31] that since the ocean current is constant and irrotational in i, the USV can be described by the 3-DOF maneuvering model in (2.14).

Given the assumptions above, the matrices have the following structure:

$$\boldsymbol{M}_x \triangleq \begin{bmatrix} m_{11}^x & 0 & 0 \\ 0 & m_{22}^x & m_{23}^x \\ 0 & m_{23}^x & m_{33}^x \end{bmatrix}, \quad \boldsymbol{D}(\boldsymbol{\nu}_r) \triangleq \begin{bmatrix} d_{11} + d_{11}^q u_r & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix}, \tag{2.16}$$
$$\boldsymbol{B} \triangleq \begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \\ 0 & b_{32} \end{bmatrix}, \quad \boldsymbol{C}_x(\boldsymbol{z}) \triangleq \begin{bmatrix} 0 & 0 & -m_{22}^x z_2 - m_{23}^x z_3 \\ 0 & 0 & m_{11}^x z_1 \\ m_{22}^x z_2 + m_{23}^x z_3 & -m_{11}^x z_1 & 0 \end{bmatrix},$$

for $x \in \{RB, A\}$. Assumption 2.1-2.2 justify the structure of the matrices $\boldsymbol{M}_x$ and $\boldsymbol{D}$. The structure of $\boldsymbol{C}$ is obtained as described in [31]. Furthermore, the distance $x_g^\star$ from Assumption 2.2 is chosen so that $\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{f} = [\tau_u, 0, \tau_r]^T$. This point $(x_g^\star, 0)$ exists for all port-starboard symmetric ships [15], where

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{23} & m_{33} \end{bmatrix} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A. \tag{2.17}$$

For control purposes it is useful to express the USV model in component form. Below are two alternative rewrites of the model (2.14), which, depending on application and available on-board sensors, are used throughout this thesis.

Relative velocities

The model below expresses the motion of the ship through relative velocities only and is suitable for control schemes where the USV in question possesses sensors that provide these velocities as feedback. To derive these expressions, the property $\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} \equiv \boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}}_{\boldsymbol{r}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r$ which follows from Assumption 2.3

has been used [31].

$$\begin{aligned}
\dot{x} &= \cos(\psi)u_r - \sin(\psi)v_r + V_x, \\
\dot{y} &= \sin(\psi)u_r + \cos(\psi)v_y + V_y, \\
\dot{\psi} &= r, \\
\dot{u_r} &= F_{u_r}(v_r, r) - \frac{d_{11} + d_{11}^q u_r}{m_{11}} u_r + \tau_u, \\
\dot{v_r} &= X(u_r)r + Y(u_r)v_r, \\
\dot{r} &= F_r(u_r, v_r, r) + \tau_r,
\end{aligned} \tag{2.18}$$

where

$$F_{u_r}(v_r, r) = \frac{m_{22}v_r + m_{23}r}{m_{11}} r, \tag{2.19}$$

$$X(u_r) = \frac{{m_{23}}^2 - m_{11}m_{33}}{m_{22}m_{33} - {m_{23}}^2} u_r + \frac{d_{33}m_{23} - d_{23}m_{33}}{m_{22}m_{33} - {m_{23}}^2}, \tag{2.20}$$

$$Y(u_r) = \frac{m_{22}m_{23} - m_{11}m_{23}}{m_{22}m_{33} - {m_{23}}^2} u_r - \frac{d_{22}m_{33} - d_{32}m_{23}}{m_{22}m_{33} - {m_{23}}^2}, \tag{2.21}$$

$$\begin{aligned}
F_r(u_r, v_r, r) = {} & \frac{m_{23}d_{22} - m_{22}(d_{32} + (m_{22} - m_{11})u_r)}{m_{22}m_{33} - {m_{23}}^2} v_r \\
& + \frac{m_{23}(d_{23} + m_{11}u_r) - m_{22}(d_{33} + m_{23}u_r)}{m_{22}m_{33} - {m_{23}}^2} r.
\end{aligned} \tag{2.22}$$

Absolute velocities

The model below expresses the motion of the ship by separating the effects of the ocean current and expressing the remainder of the model through absolute velocities. In deriving the expressions below, it has been used that $m_{11}^{RB} - m_{22}^{RB} = 0$, which follows from the fact that $m_{11}^{RB} = m_{22}^{RB} = m$, where m is the mass of the vessel [31].

$$\begin{aligned}
\dot{x} &= \cos(\psi)u - \sin(\psi)v, \\
\dot{y} &= \sin(\psi)u + \cos(\psi)v, \\
\dot{\psi} &= r, \\
\dot{u} &= -\frac{d_{11} + d_{11}^q u}{m_{11}} u + \frac{(m_{22}v + m_{23}r)r}{m_{11}} + \boldsymbol{\phi}_u^T(\psi, r)\boldsymbol{\theta}_u + \tau_u, \\
\dot{v} &= X(u_r, u_c)r + Y(u_r)v_r, \\
\dot{r} &= F_r(u, v, r) + \boldsymbol{\phi}_r^T(u, v, r, \psi)\boldsymbol{\theta}_r + \tau_r,
\end{aligned} \tag{2.23}$$

where

$$\boldsymbol{\theta}_u = \boldsymbol{\theta}_r = \begin{bmatrix} V_x & V_y & V_x^2 & V_y^2 & V_x V_y \end{bmatrix}^T, \tag{2.24}$$

$$\boldsymbol{\phi}_u(\psi, r) = \begin{bmatrix} \frac{d_{11}+2d_{11}^q u}{m_{11}} \cos(\psi) - \frac{m_{11}^A - m_{22}^A}{m_{11}} r \sin(\psi) \\ \frac{d_{11}+2d_{11}^q u}{m_{11}} \sin(\psi) + \frac{m_{11}^A - m_{22}^A}{m_{11}} r \cos(\psi) \\ -d_{11}^q \cos^2(\psi) \\ -d_{11}^q \sin^2(\psi) \\ -2d_{11}^q \cos(\psi)\sin(\psi) \end{bmatrix}, \tag{2.25}$$

$$\begin{aligned} X(u_r, u_c) = \frac{1}{\Gamma}\Big(&m_{33}(-d_{23} - m_{11}u_r - m_{11}^{RB}u_c) \\ &+ m_{23}d_{33} + m_{23}(m_{23}u_r + m_{23}^{RB}u_c + m_{22}^A u_c)\Big), \end{aligned} \tag{2.26}$$

$$Y(u_r) = \frac{1}{\Gamma}\left(-m_{33}d_{22} + m_{23}d_{32} + m_{23}(m_{22}^A - m_{11}^A u_r)\right), \tag{2.27}$$

$$\begin{aligned} F_r(u, v, r) = \frac{m_{22}}{\Gamma}&\left(-(m_{22}v - m_{23}r)u + m_{11}uv - d_{32}v - d_{33}r\right) \\ &- \frac{m_{23}}{\Gamma}\left(-m_{11}ur - d_{22}v - d_{23}r\right). \end{aligned} \tag{2.28}$$

Here, $\Gamma = m_{22}m_{33} - m_{23}^2 > 0$. Furthermore, the function $\boldsymbol{\phi}_r(u, v, r, \psi) = [\phi_{r1}, ..., \phi_{r5}]^T$ is defined by

$$\begin{aligned} \begin{bmatrix} \phi_{r1} \\ \phi_{r2} \end{bmatrix} &= \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \\ \phi_{r3} &= -\frac{m_{22}}{\Gamma}(m_{11}^A - m_{22}^A)\sin(\psi)\cos(\psi), \\ \phi_{r4} &= \frac{m_{22}}{\Gamma}(m_{11}^A - m_{22}^A)\sin(\psi)\cos(\psi), \\ \phi_{r5} &= \frac{m_{22}}{\Gamma}(m_{11}^A - m_{22}^A)(1 - 2\sin^2(\psi)), \end{aligned} \tag{2.29}$$

where

$$a_1 = -\frac{m_{22}}{\Gamma}\left((m_{11}^A - m_{22}^A)v + (m_{23}^A - m_{22}^A)r\right) - \frac{m_{23}}{\Gamma} m_{11}^A r, \tag{2.30}$$

$$a_2 = \frac{m_{22}}{\Gamma}\left(d_{32} - (m_{11}^A - m_{22}^A)u\right) - \frac{m_{23}}{\Gamma} d_{22}. \tag{2.31}$$

2.4.2 Underwater Vehicles

The equations of motion of an AUV under the influence of ocean currents is similar in structure to that of the surface vessel (2.14) [31]. Note that a neutrally buoyant AUV moving in the horizontal plane may be described by the same models as a surface vessel, and that the model presented in this section is relevant for 3D motion.

Assumption 2.4: The roll angle ϕ of the AUV is assumed passively stabilized due to the effects of gravity and/or fins.

Remark 2.4*:* Assumption 2.4 is a common assumption for slender-body underwater vehicles such as HUGIN [31].

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{J}(\boldsymbol{\Theta})\boldsymbol{\nu}_r + \boldsymbol{V}_c, \\ \boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\nu_r + \boldsymbol{g}(\boldsymbol{\eta}) &= \boldsymbol{B}\boldsymbol{f}. \end{aligned} \tag{2.32}$$

The state of the surface vessel is given by the vector $\boldsymbol{\eta} = [x, y, z, \theta, \psi]^T$, which describes the position $\boldsymbol{p} = [x, y, z]^T$ and the orientation $\boldsymbol{\Theta} = [\theta, \psi]^T$, i.e. the pitch and way angles, of the AUV with respect to the inertial frame $\{i\}$. The vector $\boldsymbol{\nu} = [u, v, w, q, r]^T$ contains the linear and angular velocities of the underwater vehicle defined in the body-fixed frame $\{b\}$, where u is the surge velocity, v is the sway velocity, w is the heave velocity and q and r are the pitch and yaw rates, respectively. The ocean current velocity in the body frame, $\boldsymbol{\nu}_c = [u_c, v_c, w_c, 0, 0]^T$, is obtained from $[u_c, v_c, w_c]^T = \boldsymbol{R}_b^i(\boldsymbol{\Theta}_{ib})^T\boldsymbol{V}_c = \boldsymbol{R}_b^i(\boldsymbol{\Theta}_{ib})^T[V_x, V_y, V_z]^T$, where the rotation matrix $\boldsymbol{R}(\boldsymbol{\Theta}_{ib})$ is defined in 2.1 and $\boldsymbol{\Theta}_{ib} = [0, \theta, \psi]^T$. Hence, $\boldsymbol{R}(\boldsymbol{\Theta}_{ib})$ simplifies to

$$\boldsymbol{R}_b^i(\boldsymbol{\Theta}_{ib}) = \begin{bmatrix} \cos(\psi)\cos(\theta) & -\sin(\psi) & \cos(\psi)\sin(\theta) \\ \sin(\psi)\cos(\theta) & \cos(\psi) & \sin(\psi)\sin(\theta) \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}. \tag{2.33}$$

Furthermore,

$$\begin{aligned} \boldsymbol{J}(\boldsymbol{\Theta}_{ib}) &\triangleq \begin{bmatrix} \boldsymbol{R}_b^i(\boldsymbol{\Theta}_b^i) & \boldsymbol{0}_{3\times 2} \\ \boldsymbol{0}_{2\times 3} & \boldsymbol{T}(\boldsymbol{\Theta}_{ib}) \end{bmatrix}, \\ \boldsymbol{T}(\boldsymbol{\Theta}_{ib}) &\triangleq \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\cos(\theta)} \end{bmatrix}. \end{aligned} \tag{2.34}$$

Similar to (2.14), $\boldsymbol{M}$, $\boldsymbol{D}(\boldsymbol{\nu}_r)$ and $\boldsymbol{C}(\boldsymbol{\nu}_r)$ describe the mass, damping and Coriolis and centripetal effects, respectively, where $\boldsymbol{M}$ and $\boldsymbol{C}$ include the added mass effect. In addition, $\boldsymbol{g}(\boldsymbol{\eta})$ contains gravity and buoyancy forces, and the controlled input $\boldsymbol{f} = [T, \delta_q, \delta_r]^T$ consists of the thruster force and the rudder angles of the side and top/bottom rudders, respectively.

Assumption 2.5: The AUV is port-starboard symmetric.

Assumption 2.6: The body-fixed coordinate frame b is anchored in a point $(x_g^\star, 0, 0)$ from the vehicle's CG along the center-line of the AUV, where $x_g^\star$ is to be defined later.

Remark 2.5*:* The body-fixed coordinate system can always be translated to the required location $x_g^\star$ [31].

Assumption 2.7: The vehicle is neutrally buoyant and the CG and the center of buoyancy (CB) are located along the same vertical axis in the body frame.

Remark 2.6*:* CG and CB are chosen through design of the underwater vehicle using weights and buoyancy elements.

Assumption 2.8: The ocean current in the inertial frame $\boldsymbol{V}_c \triangleq [V_x, V_y, V_z, 0, 0]^T$ is constant, irrotational and bounded. Hence there exists a constant $V_{\max} > 0$ such that $V_{\max} > \sqrt{V_x^2 + V_y^2 + V_z^2}$.

Given the assumptions above, the matrices have the following structure:

$$\begin{aligned}
\boldsymbol{M} &\triangleq \begin{bmatrix} m_{11} & 0 & 0 & 0 & 0 \\ 0 & m_{22} & 0 & 0 & m_{25} \\ 0 & 0 & m_{33} & m_{34} & 0 \\ 0 & 0 & m_{34} & m_{44} & 0 \\ 0 & m_{25} & 0 & 0 & m_{55} \end{bmatrix}, \quad \boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ BGzW\sin(\theta) \\ 0 \end{bmatrix}, \\
\boldsymbol{D}(\boldsymbol{\nu}_r) &\triangleq \begin{bmatrix} d_{11} + d_{11}^q|u_r| & 0 & 0 & 0 & 0 \\ 0 & d_{22} & 0 & 0 & d_{25} \\ 0 & 0 & d_{33} & d_{34} & 0 \\ 0 & 0 & d_{43} & d_{44} & 0 \\ 0 & d_{52} & 0 & 0 & d_{55} \end{bmatrix}, \quad \boldsymbol{B} \triangleq \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & 0 & b_{23} \\ 0 & b_{32} & 0 \\ 0 & b_{42} & 0 \\ 0 & 0 & b_{53} \end{bmatrix}, \\
\boldsymbol{C}(\boldsymbol{\nu}_r) &\triangleq \begin{bmatrix} 0 & 0 & 0 & m_{33}w_r + m_{34}q & -m_{22}v_r - m_{25}r \\ 0 & 0 & 0 & 0 & m_{11}u_r \\ 0 & 0 & 0 & -m_{11}u_r & 0 \\ -m_{33}w_r - m_{34}q & 0 & m_{11}u_r & 0 & 0 \\ m_{22}v_r + m_{25}r & -m_{11}u_r & 0 & 0 & 0 \end{bmatrix},
\end{aligned} \tag{2.35}$$

where BG_z is the vertical distance between CG and CG, and W is the weight of the AUV. Assumption 2.5-2.7 justify the structure of the matrices $\boldsymbol{M}$ and $\boldsymbol{D}$. The structure of $\boldsymbol{C}$ is obtained as described in [31]. Furthermore, the distance $x_g^\star$ from Assumption 2.6 is chosen so that $\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{f} = [\tau_u, 0, 0, \tau_q, \tau_r]^T$. τ_u, τ_q and τ_r are proportional to T, δ_q and δ_r respectively and are the controlled inputs in the system. The point $(x_g^\star, 0, 0)$ exists for AUVs of cylindrical shape employing symmetric steering and diving control surfaces [14].

In this thesis, the following relative velocity component model has been used:

$$\begin{aligned}
\dot{x} &= \cos(\theta)\cos(\psi)u_r - \sin(\psi)v_r + \sin(\theta)\cos(\psi)w_r + V_x, \\
\dot{y} &= \cos(\theta)\sin(\psi)u_r + \cos(\psi)v_r + \sin(\theta)\sin(\psi)w_r + V_y, \\
\dot{z} &= -\sin(\theta)u_r + \cos(\theta)w_r + V_z, \\
\dot{\theta} &= q, \\
\dot{\psi} &= \frac{r}{\cos(\theta)}, \\
\dot{u}_r &= F_{u_r}(u_r, v_r, w_r, r, q) + \tau_u, \\
\dot{v}_r &= X_{v_r}(u_r)r + Y_{v_r}(u_r)v_r, \\
\dot{w}_r &= X_{w_r}(u_r)q + Y_{w_r}(u_r)w_r + Z_{w_r}\sin(\theta), \\
\dot{q} &= F_q(\theta, u_r, w_r, q) + \tau_q, \\
\dot{r} &= F_r(u_r, v_r, r) + \tau_r,
\end{aligned} \tag{2.36}$$

where

$$
\begin{aligned}
F_{u_r}(u_r, v_r, w_r, r, q) &= \frac{(m_{22}v_r + m_{25}r)r - (m_{33}w_r + m_{34}q)q}{m_{11}} \\
&\quad - \frac{d_{11}u_r - d_{11}^{q}|u_r|u_r}{m_{11}}, && (2.37) \\
X_{v_r}(u_r) &= \frac{{m_{25}}^2 - m_{11}m_{55}}{m_{22}m_{55} - {m_{25}}^2}u_r + \frac{d_{55}m_{25} - d_{25}m_{55}}{m_{22}m_{55} - {m_{25}}^2}, && (2.38) \\
Y_{v_r}(u_r) &= \frac{(m_{22} - m_{11})m_{25}}{m_{22}m_{55} - {m_{25}}^2}u_r - \frac{d_{22}m_{55} - d_{52}m_{25}}{m_{22}m_{55} - {m_{25}}^2}, && (2.39) \\
X_{w_r}(u_r) &= \frac{-{m_{34}}^2 + m_{11}m_{44}}{m_{33}m_{44} - {m_{34}}^2}u_r + \frac{d_{44}m_{34} - d_{34}m_{44}}{m_{33}m_{44} - {m_{34}}^2}, && (2.40) \\
Y_{w_r}(u_r) &= \frac{(m_{11} - m_{33})m_{34}}{m_{33}m_{44} - {m_{34}}^2}u_r - \frac{d_{33}m_{44} - d_{43}m_{34}}{m_{33}m_{44} - {m_{34}}^2}, && (2.41) \\
Z_{w_r} &= \frac{BG_zWm_{34}}{m_{33}m_{44} - {m_{34}}^2}, && (2.42) \\
F_q(\theta, u_r, w_r, q) &= \frac{m_{34}d_{33} - m_{33}(d_{43} - (m_{33} - m_{11})u_r)}{m_{33}m_{44} - {m_{34}}^2}w_r \\
&\quad + \frac{m_{34}(d_{34} - m_{11}u_r) - m_{33}(d_{44} - m_{34}u_r)}{m_{33}m_{44} - {m_{34}}^2}q \\
&\quad - \frac{BG_zWm_{33}\sin(\theta)}{m_{33}m_{44} - {m_{34}}^2}, && (2.43) \\
F_r(u_r, w_r, r) &= \frac{m_{25}d_{22} - m_{22}(d_{52} + (m_{22} - m_{11})u_r)}{m_{22}m_{55} - {m_{25}}^2}v_r \\
&\quad + \frac{m_{25}(d_{25} + m_{11}u_r) - m_{22}(d_{55} + m_{25}u_r)}{m_{22}m_{55} - {m_{25}}^2}r. && (2.44)
\end{aligned}
$$

2.5 The Singularity-robust Multiple Task-priority Inverse Kinematics Framework

This section describes the singularity-robust multiple task-priority inverse kinematics framework, which is a basis for the work presented in Part I in this thesis.

Traditionally, robotic systems are controlled in their joint space. However, the tasks they are required to perform are often given in the operational space, for instance given by the desired end effector position or orientation. As such, a variety of inverse kinematics and dynamics algorithms have been developed to map tasks from the operational space to the joint space and thus generate reference trajectories for the controllers. On the kinematic level, the most common approach is to use a Jacobian-based method [69]. The singularity-robust multiple task-priority inverse kinematics framework calculates a reference for the joint velocities of a robotic system that, if fulfilled, fulfills several tasks in a prioritized order. It is robust with respect to algorithmic singularities, which can arise when two tasks are in conflict with each other [23].

A general robotic system has n DOFs. Its configuration is given by the joint values $\boldsymbol{q} = [q_1, q_2, \ldots, q_n]^T$. Tasks and task velocities in the operational space can be expressed through forward kinematics and the task Jacobian matrix. Define an m-dimensional task $\boldsymbol{\sigma}(t) \in \mathbb{R}^m$ as

$$\boldsymbol{\sigma}(t) = \boldsymbol{f}(\boldsymbol{q}(t)), \tag{2.45}$$

with the corresponding differential relationship:

$$\dot{\boldsymbol{\sigma}}(t) = \frac{\partial \boldsymbol{f}(\boldsymbol{q}(t))}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}}(t) = \boldsymbol{J}(\boldsymbol{q}(t))\dot{\boldsymbol{q}}(t), \tag{2.46}$$

where $\boldsymbol{J}(\boldsymbol{q}(t)) \in \mathbb{R}^{m \times n}$ is the configuration-dependent task Jacobian matrix and $\dot{\boldsymbol{q}}(t) \in \mathbb{R}^n$ is the system velocity. For compactness, the argument $\boldsymbol{q}$ of tasks and Jacobians are omitted from the equations in this thesis.

Consider a single m-dimensional task to be followed, with a defined desired trajectory $\boldsymbol{\sigma}_{\text{des}}(t) \in \mathbb{R}^m$. The corresponding joint references $\boldsymbol{q}_{\text{des}}(t) \in \mathbb{R}^n$ for the robotic system may be computed by integrating the locally inverse mapping of (2.46). The least-squares solution is given as

$$\dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{J}^{\dagger} \dot{\boldsymbol{\sigma}}_{\text{des}} = \boldsymbol{J}^T \left(\boldsymbol{J}\boldsymbol{J}^T\right)^{-1} \dot{\boldsymbol{\sigma}}_{\text{des}}, \tag{2.47}$$

where $\boldsymbol{J}^{\dagger}$, implicitly defined in the above equation for full row rank matrices, is the right pseudo-inverse of $\boldsymbol{J}$. In the general case, the pseudo-inverse is the matrix that satisfies the four Moore-Penrose conditions (2.48)-(2.51)[38], and it is defined for systems that are not square ($m \neq n$) nor have full rank [17]:

$$\boldsymbol{J}\boldsymbol{J}^{\dagger}\boldsymbol{J} = \boldsymbol{J}, \tag{2.48}$$

$$\boldsymbol{J}^{\dagger}\boldsymbol{J}\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\dagger}, \tag{2.49}$$

$$(\boldsymbol{J}\boldsymbol{J}^{\dagger})^{\star} = \boldsymbol{J}\boldsymbol{J}^{\dagger}, \tag{2.50}$$

$$(\boldsymbol{J}^{\dagger}\boldsymbol{J})^{\star} = \boldsymbol{J}^{\dagger}\boldsymbol{J}. \tag{2.51}$$

Here, $\boldsymbol{J}^{\star}$ denotes the complex-conjugate of $\boldsymbol{J}$.

The vector $\boldsymbol{q}_{\text{des}}$ achieved by taking the time integral of (2.47) is prone to drifting. To handle this, a closed-loop inverse kinematics (CLIK) version of the algorithm is usually implemented [23], where

$$\dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{J}^{\dagger} \left(\dot{\boldsymbol{\sigma}}_{\text{des}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}}\right) = \boldsymbol{J}^{\dagger} \dot{\boldsymbol{\sigma}}_{\text{ref}}. \tag{2.52}$$

Here, $\tilde{\boldsymbol{\sigma}} \in \mathbb{R}^m$ is the task error defined as

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_{\text{des}} - \boldsymbol{\sigma} \tag{2.53}$$

and $\boldsymbol{\Lambda} \in \mathbb{R}^{m \times m}$ is a positive-definite matrix of gains. This feedback approach reduces the error dynamics to

$$\begin{aligned} \dot{\tilde{\boldsymbol{\sigma}}} &= \dot{\boldsymbol{\sigma}}_{\text{des}} - \dot{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}}_{\text{des}} - \boldsymbol{J}\dot{\boldsymbol{q}} \\ &= \dot{\boldsymbol{\sigma}}_{\text{des}} - \boldsymbol{J}\boldsymbol{J}^{\dagger}(\dot{\boldsymbol{\sigma}}_{\text{des}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}}) \\ &= -\boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}}, \end{aligned} \tag{2.54}$$

if $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{\text{des}}$ and $\boldsymbol{J}$ has full rank, implying that $\boldsymbol{J}\boldsymbol{J}^{\dagger} = \boldsymbol{I}$. Equation (2.54) describes a linear system with a globally exponentially stable equilibrium point at the equilibrium $\tilde{\boldsymbol{\sigma}} = \mathbf{0}$. It is worth noticing that the assumption $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{\text{des}}$ is common to all inverse kinematics algorithms [3]. For practical applications, it requires that the low level dynamic control loop is faster than the kinematic one.

In case of system redundancy, i.e., if $n > m$, the classic general solution contains a null projector operator [53]:

$$\dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{ref}} + \left(\boldsymbol{I}_n - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right)\dot{\boldsymbol{q}}_{\text{null}}, \tag{2.55}$$

where $\boldsymbol{I}_n$ is the $(n \times n)$ identity matrix and the vector $\dot{\boldsymbol{q}}_{\text{null}} \in \mathbb{R}^n$ is an arbitrary system velocity vector. It can be recognized that the operator $\left(\boldsymbol{I}_n - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right)$ projects $\dot{\boldsymbol{q}}_{\text{null}}$ in the null-space of the Jacobian matrix. This corresponds to generating a motion of the robotic system that does not affect that of the given task.

For highly redundant systems, multiple tasks can be arranged in priority. Consider three tasks that will be denoted with the subscripts 1, 2 and 3, respectively:

$$\boldsymbol{\sigma}_1 = \boldsymbol{f}_1(\boldsymbol{q}) \in \mathbb{R}^{m_1}, \tag{2.56}$$

$$\boldsymbol{\sigma}_2 = \boldsymbol{f}_2(\boldsymbol{q}) \in \mathbb{R}^{m_2}, \tag{2.57}$$

$$\boldsymbol{\sigma}_3 = \boldsymbol{f}_3(\boldsymbol{q}) \in \mathbb{R}^{m_3}. \tag{2.58}$$

For each of the tasks a corresponding Jacobian matrix can be defined, denoted $\boldsymbol{J}_1 \in \mathbb{R}^{m_1 \times n}$, $\boldsymbol{J}_2 \in \mathbb{R}^{m_2 \times n}$ and $\boldsymbol{J}_3 \in \mathbb{R}^{m_3 \times n}$, respectively. Let us further define the corresponding null-space projectors for the first two tasks as

$$\boldsymbol{N}_1 = \left(\boldsymbol{I}_n - \boldsymbol{J}_1^{\dagger}\boldsymbol{J}_1\right), \tag{2.59}$$

$$\boldsymbol{N}_2 = \left(\boldsymbol{I}_n - \boldsymbol{J}_2^{\dagger}\boldsymbol{J}_2\right). \tag{2.60}$$

The *augmented Jacobian* of tasks 1 and 2 is given by stacking the two independent task Jacobians:

$$\boldsymbol{J}_{12}^{\text{A}} = \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} \tag{2.61}$$

The common null-space for tasks 1 and 2 is then defined as

$$\boldsymbol{N}_{12}^{\text{A}} = \left(\boldsymbol{I}_n - \boldsymbol{J}_{12}^{\text{A}\,\dagger}\boldsymbol{J}_{12}^{\text{A}}\right), \tag{2.62}$$

where $\boldsymbol{J}_{12}^{\text{A}\,\dagger}$ is the pseudo-inverse of $\boldsymbol{J}_{12}^{\text{A}}$ that satisfies the four Moore-Penrose conditions (2.48)-(2.51). By expanding the expression for $\boldsymbol{J}_{12}^{\text{A}}$, we see that

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} \boldsymbol{N}_{12}^{\text{A}} &= \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} \left(\boldsymbol{I}_n - \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix}^{\dagger} \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix}\right) \\
&= \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} - \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix}^{\dagger} \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} - \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix},
\end{aligned} \tag{2.63}$$

so in general,

$$\boldsymbol{J}_i \boldsymbol{N}^{\mathrm{A}}_{12..k} = \boldsymbol{0} \text{ for } i \in \{1, ..., k\}\,. \tag{2.64}$$

The following equation then defines the desired joint velocities:

$$\dot{\boldsymbol{q}}_{\mathrm{des}} = \underbrace{\boldsymbol{J}_1^{\dagger} \dot{\boldsymbol{\sigma}}_{1,\mathrm{ref}}}_{\dot{\boldsymbol{q}}_{1,\mathrm{des}}} + \boldsymbol{N}_1 \underbrace{\boldsymbol{J}_2^{\dagger} \dot{\boldsymbol{\sigma}}_{2,\mathrm{ref}}}_{\dot{\boldsymbol{q}}_{2,\mathrm{des}}} + \boldsymbol{N}^{\mathrm{A}}_{12} \underbrace{\boldsymbol{J}_3^{\dagger} \dot{\boldsymbol{\sigma}}_{3,\mathrm{ref}}}_{\dot{\boldsymbol{q}}_{3,\mathrm{des}}}, \tag{2.65}$$

where the definition of $\dot{\boldsymbol{\sigma}}_{x,\mathrm{ref}}$ can be easily extrapolated from (2.52) for each task with the corresponding positive definite matrix $\boldsymbol{\Lambda}_x \in \mathbb{R}^{m_x \times m_x}$. The priority of the tasks follows the numerical order, with $\boldsymbol{\sigma}_1$ being the highest-priority task. Equation (2.65) also implicitly defines the joint velocities $\dot{\boldsymbol{q}}_{x,\mathrm{des}} \in \mathbb{R}^n$ that represent the desired joint velocity corresponding to task $\boldsymbol{\sigma}_x$ if this was the sole task.

The generalization to k tasks is straightforward: Equation (2.65) can be expanded as follows:

$$\dot{\boldsymbol{q}}_{\mathrm{des}} = \boldsymbol{J}_1^{\dagger} \dot{\boldsymbol{\sigma}}_{1,\mathrm{ref}} + \boldsymbol{N}_1 \boldsymbol{J}_2^{\dagger} \dot{\boldsymbol{\sigma}}_{2,\mathrm{ref}} + \cdots + \boldsymbol{N}^{\mathrm{A}}_{12\ldots(k-1)} \boldsymbol{J}_k^{\dagger} \dot{\boldsymbol{\sigma}}_{k,\mathrm{ref}}, \tag{2.66}$$

where $\boldsymbol{N}^{\mathrm{A}}_{12\ldots(k-1)}$ is the null-space of the augmented Jacobian matrix

$$\boldsymbol{J}^{\mathrm{A}}_{12\ldots(k-1)} = \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \\ \vdots \\ \boldsymbol{J}_{k-1} \end{bmatrix}. \tag{2.67}$$

Equation (2.65) represents the singularity-robust multiple task-priority inverse kinematics framework. A formal stability for this method exists for a general number of *regulation tasks*, i.e. tasks where $\dot{\boldsymbol{\sigma}}_{\mathrm{des}} \equiv \boldsymbol{0}$ [3]. The error dynamics in this case are reduced to

$$\begin{aligned}
\dot{\tilde{\boldsymbol{\sigma}}} = -\dot{\boldsymbol{\sigma}}_{\mathrm{eb}} = -\begin{bmatrix} \dot{\boldsymbol{\sigma}}_1 \\ \dot{\boldsymbol{\sigma}}_2 \\ \vdots \\ \dot{\boldsymbol{\sigma}}_k \end{bmatrix} &= -\begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \\ \vdots \\ \boldsymbol{J}_k \end{bmatrix} \dot{\boldsymbol{q}} \\
= -\begin{bmatrix} \boldsymbol{\Lambda}_1 & \boldsymbol{0}_{m_1 \times m_2} & \cdots & \boldsymbol{0}_{m_1 \times m_k} \\ \boldsymbol{J}_2 \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 & \boldsymbol{J}_2 \boldsymbol{N}_1 \boldsymbol{J}_2^{\dagger} \boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{0}_{m_2 \times m_k} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{J}_k \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 & \boldsymbol{J}_k \boldsymbol{N}_1 \boldsymbol{J}_2^{\dagger} \boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{J}_k \boldsymbol{N}^{\mathrm{A}}_{12..(k-1)} \boldsymbol{J}_k^{\dagger} \boldsymbol{\Lambda}_k \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\sigma}}_1 \\ \tilde{\boldsymbol{\sigma}}_2 \\ \vdots \\ \tilde{\boldsymbol{\sigma}}_k \end{bmatrix} &= -\boldsymbol{M} \tilde{\boldsymbol{\sigma}}
\end{aligned} \tag{2.68}$$

if $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{\mathrm{des}}$. The matrix $\boldsymbol{M}$ is positive definite if Assumption 2.9-2.10 are satisfied [3].

Assumption 2.9: When an additional task is considered, the task Jacobian is independent with respect to the Jacobian obtained by stacking all the higher priority tasks, i.e.

$$\rho({\boldsymbol{J}^{\mathrm{A}}_{12..(i-1)}}^{\dagger}) + \rho(\boldsymbol{J}_i^{\dagger}) = \rho(\begin{bmatrix} {\boldsymbol{J}^{\mathrm{A}}_{12..(i-1)}}^{\dagger} & \boldsymbol{J}_i^{\dagger} \end{bmatrix}) \tag{2.69}$$

for $i \in \{2, ..., k\}$ where, $\rho(\cdot)$ is the rank of the matrix and k is the total number of tasks.

Remark 2.7: Assumption 2.9 requires that the tasks are compatible. For instance, if the system is given one end effector position tracking task and one collision avoidance task, and the desired trajectory moves through the obstacle, the tasks are clearly not compatible. In this case, Assumption 2.9 is not satisfied, and the system will fulfill the highest priority task.

Assumption 2.10: The task gains are chosen according to [3]. For the specific case of $k = 3$, the task gains are chosen as $\boldsymbol{\Lambda}_1 = \lambda_1 \boldsymbol{I}_{m_1}$, $\boldsymbol{\Lambda}_2 = \lambda_2 \boldsymbol{I}_{m_2}$ and $\boldsymbol{\Lambda}_3 = \lambda_3 \boldsymbol{I}_{m_3}$ for the first, second and third priority task respectively, with

$$\lambda_1 > 0 \tag{2.70}$$

$$\lambda_2 > \max(0, \frac{\overline{\lambda}_{21} - \underline{\lambda}_{11}}{\underline{\lambda}_{22}} \lambda_1) \tag{2.71}$$

$$\lambda_3 > \max(0, \frac{\overline{\lambda}_{31} - \underline{\lambda}_{11}}{\underline{\lambda}_{33}} \lambda_1, \frac{\overline{\lambda}_{32} - \underline{\lambda}_{22}}{\underline{\lambda}_{33}} \lambda_2), \tag{2.72}$$

where $\overline{\lambda}_{ij}$ and $\underline{\lambda}_{ij}$ denote the largest and smallest singular value of the matrix $\boldsymbol{P}_{ij}$, respectively, and

$$\begin{array}{ll} \boldsymbol{P}_{11} = \boldsymbol{I}_{m_1}, & \boldsymbol{P}_{22} = \boldsymbol{J}_2 \boldsymbol{N}_1 \boldsymbol{J}_2^{\dagger}, \\ \boldsymbol{P}_{21} = \boldsymbol{J}_2 \boldsymbol{J}_1^{\dagger}, & \boldsymbol{P}_{32} = \boldsymbol{J}_3 \boldsymbol{N}_1 \boldsymbol{J}_2^{\dagger}, \\ \boldsymbol{P}_{31} = \boldsymbol{J}_3 \boldsymbol{J}_1^{\dagger}, & \boldsymbol{P}_{33} = \boldsymbol{J}_3 \boldsymbol{N}_{12} \boldsymbol{J}_3^{\dagger}. \end{array} \tag{2.73}$$

Part I

Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework

Chapter 3

Preliminary Result: Kinematic Control of a Dual-Arm UVMS

Contributions of this chapter: This chapter is a preliminary result to the remainder of Part I of this thesis in the sense that the work presented in this chapter served as motivation to extend the singularity-robust multiple task-priority inverse kinematics framework to include set-based tasks.

Section 2.5 described the singularity-robust multiple task-priority inverse kinematics framework, which is a type of kinematic control that allows redundant system to perform several tasks in a prioritized order. This chapter considers a dual-arm UVMS, which can be considered as a manipulator arm mounted on a floating base. A dual-manipulator system can use the two arms to cooperate and thereby perform more complex tasks and pick up larger/heavier objects. A lot of research has been done on fixed dual-arm systems regarding coordinated and cooperative control, leader/follower control, force control, collision detection and avoidance etc. [19, 91], but very little has been done for a two-manipulator floating base system.

This chapter proposes to use the singularity-robust multiple task-priority inverse kinematics scheme for a dual-arm UVMS, which calculates references for both the manipulator arms and the vehicle base itself, in order to implement several concurrent tasks. A variety of tasks can be implemented and included in a prioritized order. Furthermore, it is proposed to consider one manipulator and the vehicle base as a leader and the second manipulator as the follower. This division ensures that conflicting tasks do not attempt to move the vehicle base in different directions and that the tasks are achieved using the desired DOFs. An example scenario in which the framework can be applied is presented.

Organization of this chapter: Section 3.1 presents the vehicle kinematics and the proposed division of the leader-follower states. Section 3.2 describes the implemented tasks and the corresponding task Jacobians and Section 3.3 contains the proposed guidance system based on the vehicle tasks. The simulation results are given in Section 3.4. A brief discussion about the motivation for set-based tasks are given in Section 3.5 and conclusions in Section 3.6 .

Publications: The results in this chapter are based on [62].

3.1 Vehicle Kinematics

A complete model of an UVMS consists of the kinematics and the dynamics. The kinematics is relatively straight forward and described in [4, 46]. The forward kinematics of the manipulator arm(s) can be derived, for instance, by the Denavit-Hartenberg convention [86]. The dynamics of such a vehicle, on the other hand, is very complex, highly non-linear and contains several cross-terms because of the interaction between the vehicle body and the manipulator arm [4, 78]. In addition, the numeric values of the hydrodynamical parameters are difficult to identify precisely [60, 70, 84]. Hence, the singularity-robust multiple task-priority inverse kinematics framework, which is a type of kinematic control (Section 2.5), may successfully be used to generate reference states for this type of redundant vehicles in combination with controllers that are not model-based (e.g. a standard PID-controller).

In this chapter, the singularity-robust multiple task-priority inverse kinematics framework described in Section 2.5 is applied to a two-manipulator underwater vehicle in the plane. The results presented here illustrate the implementation and properties of the framework that forms the basis for the remainder of Part I in this thesis. The proposed approach can be followed for manipulators with any number of links. In this chapter, a specific example is presented where the two manipulators have two and three links, respectively. The vector $\boldsymbol{\eta}_{\mathrm{b}} = [x_{\mathrm{b}}, y_{\mathrm{b}}, \psi_{\mathrm{b}}]^T$ is the vehicle position and orientation relative to the inertial frame. Similarly, $\boldsymbol{\eta}_1 = [x_{\mathrm{ee1}}, y_{\mathrm{ee1}}, \psi_{\mathrm{ee1}}]^T$ and $\boldsymbol{\eta}_2 = [x_{\mathrm{ee2}}, y_{\mathrm{ee2}}, \psi_{\mathrm{ee2}}]^T$ describe the position and orientation of the manipulator end effectors in inertial frame, see Figure 3.1.

The vehicle's velocity is defined in body-frame as $\boldsymbol{\nu} = [u, v, r]^T$, where u and v are the linear velocities and r is the angular rate. Equation (3.1) describes the relationship between the velocities in the inertial and body frame.

$$\dot{\boldsymbol{\eta}}_{\mathrm{b}} = \boldsymbol{R}(\psi_{\mathrm{b}})\boldsymbol{\nu} = \begin{bmatrix} \cos(\psi_{\mathrm{b}}) & -\sin(\psi_{\mathrm{b}}) & 0 \\ \sin(\psi_{\mathrm{b}}) & \cos(\psi_{\mathrm{b}}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\nu}. \tag{3.1}$$

Furthermore, the manipulator arms have joint angles $\boldsymbol{q}_1 = [q_{11}, q_{12}, q_{13}]^T$ and $\boldsymbol{q}_2 = [q_{21}, q_{22}]^T$ and joint angular velocities $\dot{\boldsymbol{q}}_1$ and $\dot{\boldsymbol{q}}_2$.

The forward kinematics of the manipulator arms are straight forward. Defining the vehicle length and height as L and H respectively and the length of the jth link of manipulator i as l_{ij}, the position and orientation of the manipulator arms in the inertial frame are given as follows:

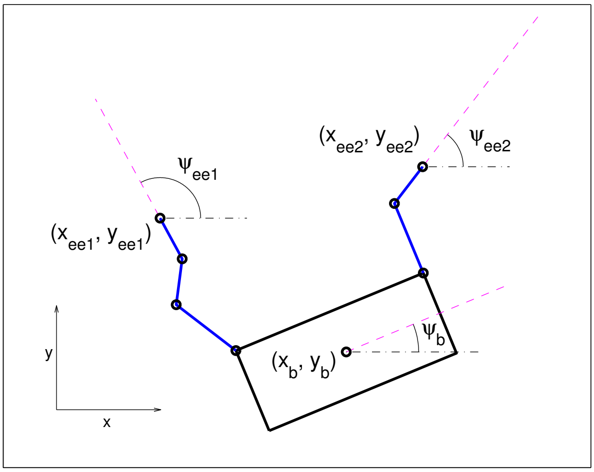


Figure 3.1: Illustration of vehicle base with two manipulator arms with position and orientation of the vehicle body and the manipulator end effectors.

$$\boldsymbol{\eta}_1(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1) = \boldsymbol{\eta}_{\mathrm{b}} + \boldsymbol{R}(\psi_{\mathrm{b}})\boldsymbol{k}_1(\boldsymbol{q}_1), \tag{3.2}$$

$$\boldsymbol{\eta}_2(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_2) = \boldsymbol{\eta}_{\mathrm{b}} + \boldsymbol{R}(\psi_{\mathrm{b}})\boldsymbol{k}_2(\boldsymbol{q}_2), \tag{3.3}$$

$$\boldsymbol{k}_1(\boldsymbol{q}_1) = \begin{bmatrix} -L/2 + l_{11}\cos(q_{11}) + l_{12}\cos(q_{11}+q_{12}) + l_{13}\cos(q_{11}+q_{12}+q_{13}) \\ H/2 + l_{11}\sin(q_{11}) + l_{12}\sin(q_{11}+q_{12}) + l_{13}\sin(q_{11}+q_{12}+q_{13}) \\ q_{11}+q_{12}+q_{13} \end{bmatrix}, \tag{3.4}$$

$$\boldsymbol{k}_2(\boldsymbol{q}_2) = \begin{bmatrix} L/2 + l_{21}\cos(q_{21}) + l_{22}\cos(q_{21}+q_{12}) \\ H/2 + l_{21}\sin(q_{21}) + l_{22}\sin(q_{21}+q_{22}) \\ q_{21}+q_{22} \end{bmatrix}. \tag{3.5}$$

Similarly, the forward kinematics of the manipulator arms relative to the body frame is given as

$$\boldsymbol{\eta}_1^{\mathrm{b}} = \boldsymbol{k}_1(\boldsymbol{q}_1), \tag{3.6}$$

$$\boldsymbol{\eta}_2^{\mathrm{b}} = \boldsymbol{k}_2(\boldsymbol{q}_2). \tag{3.7}$$

When considering the control of a one-manipulator UVMS, it is common to consider the vector $\boldsymbol{\zeta} = [\boldsymbol{\nu}^T, \dot{\boldsymbol{q}}^T]^T$ where $\boldsymbol{q}$ is a n-dimensional vector containing the angles of the manipulator arm. The guidance system then aims to find desired values for $\boldsymbol{\zeta}$ that, if fulfilled, will result in the vehicle tasks (see Section 3.2) being fulfilled. As such, one option for the two-manipulator UVMS considered in this chapter is to choose $\boldsymbol{\zeta} = [\boldsymbol{\nu}^T, \dot{\boldsymbol{q}}_1^T, \dot{\boldsymbol{q}}_2^T]^T$. With this choice one would find references for the entire system as a whole since the two manipulators share one floating base. However, if the manipulators have conflicting tasks, they will try to pull the vehicle base in different directions. As such, this chapter proposes to choose one of the manipulators as the main manipulator and any others as additional manipulators. These are only considered relative to the vehicle base and their tasks as such do not affect the desired velocity/position of the vehicle itself. Take for instance the case where the main task of manipulator two is to point the camera towards the main manipulator arm to provide the human operator/supervisor with a good visual input. To achieve this goal, only the joint angles of manipulator 2 should be changed. In particular, the position and orientation of the vehicle base to be affected in order to achieve this goal. Consequently, we propose to consider the two following independent vectors:

$$\boldsymbol{\zeta}_1 = \begin{bmatrix} \boldsymbol{\nu} \\ \dot{\boldsymbol{q}}_1 \end{bmatrix} \tag{3.8}$$

$$\boldsymbol{\zeta}_2 = \dot{\boldsymbol{q}}_2 \tag{3.9}$$

3.2 Vehicle Tasks

This chapter utilizes the singularity-robust multiple task-priority inverse kinematics behavior control to perform several tasks at once. If the UVMS has more DOFs than those required to execute a given task, the system is redundant with respect to that specific task and kinematic redundancy can be exploited to achieve additional tasks. The tasks are sorted by priority: The secondary task is given lower priority with respect to the primary task by projecting the relative actions through the null-space of the primary task Jacobian. The tertiary task is given lower priority with respect to the secondary task by projecting the relative actions through the null-space of the primary and secondary task Jacobian and so on (see Section 3.3). To illustrate the general method proposed in this chapter, a certain scenario has been implemented and simulated. The different tasks are defined in this section.

Task j for $\boldsymbol{\zeta}_i$ is denoted $\boldsymbol{\sigma}_{ij}$ and the corresponding task Jacobian $\boldsymbol{J}_{ij}$. In general for a two-manipulator UVMS modeled as proposed in this chapter, for a m DOF task, $\boldsymbol{\sigma}_{1j}$ is a vector of length m and the corresponding Jacobian is an $m \times (n_\mathrm{b}+n_1)$ matrix, and $\boldsymbol{\sigma}_{2j}$ is a vector of length m and the corresponding Jacobian is a $m \times n_2$ matrix, where n_b, n_1 and n_2 are the DOFs of the vehicle itself, manipulator 1 and manipulator 2, respectively. In this chapter we will develop the equations for the particular case where $n_\mathrm{b} = n_1 = 3$ and $n_2 = 2$. This implies that the main manipulator and vehicle can be given tasks that require up to 6 DOFs and the secondary manipulator can be given tasks that require 2 DOFs. For the particular case where manipulator 2 carries a camera and manipulator 1 is for intervention, the following tasks are proposed:

ζ_1 **- Task 1 - End effector trajectory and orientation - 3 DOFs**: The end effector of manipulator 1 should track a given trajectory with a given orientation:

$$\boldsymbol{\sigma}_{11}(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1) = \boldsymbol{\eta}_1(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1) = \begin{bmatrix} x_{\mathrm{ee1}} \\ y_{\mathrm{ee1}} \\ \psi_{\mathrm{ee1}} \end{bmatrix}, \tag{3.10}$$

$$\boldsymbol{\sigma}_{11,\mathrm{des}} = \boldsymbol{\eta}_{1,\mathrm{des}} = \begin{bmatrix} x_{\mathrm{ee1,des}} \\ y_{\mathrm{ee1,des}} \\ \psi_{\mathrm{ee1,des}} \end{bmatrix}. \tag{3.11}$$

The expression for $\boldsymbol{\sigma}_{11}$ is given by (3.2), and the Jacobian

$$\dot{\boldsymbol{\sigma}}_{11} = \boldsymbol{J}_{11}(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1)\boldsymbol{\zeta}_1 \tag{3.12}$$

is derived by taking the time-derivative of (3.2) and inserting (3.1).

ζ_1 - Task 2 - Orientation of vehicle - 1 DOF: The vehicle should have a constant orientation of $\psi_{\mathrm{b}} = 0$.

$$\boldsymbol{\sigma}_{12}(\boldsymbol{\eta}_{\mathrm{b}}) = \psi_{\mathrm{b}}, \tag{3.13}$$

$$\boldsymbol{\sigma}_{12,\mathrm{des}} = 0, \tag{3.14}$$

$$\dot{\boldsymbol{\sigma}}_{12} = \dot{\psi}_{\mathrm{b}} = r = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\zeta}_1 = \boldsymbol{J}_{12}\boldsymbol{\zeta}_1. \tag{3.15}$$

ζ_1 - Task 3 - Vertical distance between the vehicle and the end effector - 1 DOF: The vertical distance between the vehicle center and the end effector should be constant and positive to ensure that the manipulator is operating over the vehicle and that the vehicle is not blocking the view of end effector 2.

$$\boldsymbol{\sigma}_{13}(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1) = y_{\mathrm{ee1}} - y_{\mathrm{b}}, \tag{3.16}$$

$$\boldsymbol{\sigma}_{13,\mathrm{des}} = C, \tag{3.17}$$

for some positive, constant C. Similarly to Task 1, the Jacobian is derived by taking the time derivative of σ_{13} using (3.2) and (3.1):

$$\dot{\boldsymbol{\sigma}}_{13} = \boldsymbol{J}_{13}(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1)\boldsymbol{\zeta}_1. \tag{3.18}$$

ζ_2 - Task 1 - Relative Field of View - 1 DOF: End effector 2 should always point towards end effector 1. In this case, the task is chosen as the error between the desired and actual direction of the manipulator, so the desired task value is 0:

$$\boldsymbol{\sigma}_{21}(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1, \boldsymbol{q}_2) = \sqrt{(\boldsymbol{a}_{\mathrm{des}} - \boldsymbol{a})^T(\boldsymbol{a}_{\mathrm{des}} - \boldsymbol{a})}, \tag{3.19}$$

$$\boldsymbol{\sigma}_{21,\mathrm{des}} = 0, \tag{3.20}$$

where $\boldsymbol{a}_{\mathrm{des}}$ and $\boldsymbol{a}$ are unit vectors illustrated in Figure 3.2 and are defined as

$$\boldsymbol{a}_{\mathrm{des}} = \frac{1}{\sqrt{(x_{\mathrm{ee1}} - x_{\mathrm{ee2}})^2 + (y_{\mathrm{ee1}} - y_{\mathrm{ee2}})^2}} \underbrace{\begin{bmatrix} x_{\mathrm{ee1}} - x_{\mathrm{ee2}} \\ y_{\mathrm{ee1}} - y_{\mathrm{ee2}} \\ 0 \end{bmatrix}}_{\triangleq \boldsymbol{p}_e} \tag{3.21}$$

$$\boldsymbol{a} = \begin{bmatrix} \cos(\psi_{\mathrm{ee2}}) & -\sin(\psi_{\mathrm{ee2}}) & 0 \\ \sin(\psi_{\mathrm{ee2}}) & \cos(\psi_{\mathrm{ee2}}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{3.22}$$

The corresponding 1×2 Jacobian is defined in [4] as

$$\dot{\boldsymbol{\sigma}}_{21} = \boldsymbol{J}_{21}(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1, \boldsymbol{q}_2)\boldsymbol{\zeta}_2, \tag{3.23}$$

$$\boldsymbol{J}_{21} = \frac{(\boldsymbol{a}_{des} - \boldsymbol{a})^T}{\|\boldsymbol{a}_{des} - \boldsymbol{a}\|} \left(-\boldsymbol{S}(\boldsymbol{a}_{des})\boldsymbol{S}(\boldsymbol{p}_e)^{\dagger}\boldsymbol{J}_p + \boldsymbol{S}(\boldsymbol{a})\boldsymbol{J}_o \right), \tag{3.24}$$

where $\boldsymbol{S}(\cdot)$ is the matrix cross product operator and $\boldsymbol{X}^{\dagger}$ denotes the Moore-Penrose inverse of the matrix $\boldsymbol{X}$. $\boldsymbol{J}_p$ and $\boldsymbol{J}_o$ denote the position and orientation Jacobian matrices of end effector 2. With the proposed choice of $\boldsymbol{\zeta}_1$ and $\boldsymbol{\zeta}_2$, the position and orientation is that of end effector 2 relative to the body frame. By taking the time derivative of (3.7) $\boldsymbol{J}_p$ and $\boldsymbol{J}_o$ can be derived:

$$\begin{aligned} \dot{\boldsymbol{\eta}}_{\boldsymbol{2}}^{\mathrm{b}}(\boldsymbol{q}_2) &= \frac{\delta}{\delta t}\boldsymbol{k}_2(\boldsymbol{q}_2) \\ &= \begin{bmatrix} -l_{21}\sin(q_{21})\dot{q}_{21} - l_{22}\sin(q_{21} + q_{12})(\dot{q}_{21} + \dot{q}_{22}) \\ l_{21}\cos(q_{21})\dot{q}_{21} + l_{22}\cos(q_{21} + q_{22})(\dot{q}_{21} + \dot{q}_{22}) \\ \dot{q}_{21} + \dot{q}_{22} \end{bmatrix} \\ &= \begin{bmatrix} -l_{21}\sin(q_{21}) - l_{22}\sin(q_{21} + q_{12}) & -l_{22}\sin(q_{21} + q_{12}) \\ l_{21}\cos(q_{21}) + l_{22}\cos(q_{21} + q_{22}) & l_{22}\cos(q_{21} + q_{22}) \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_{21} \\ \dot{q}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{J}_p \\ \boldsymbol{J}_o \end{bmatrix} \boldsymbol{\zeta}_2. \end{aligned} \tag{3.25}$$

Remark 3.1*:* Note that the vectors $\boldsymbol{a}$ and $\boldsymbol{a}_{des}$ have been expanded to three dimensions even though movement in only considered in the plane. This is due to the fact that the matrix cross product operator is a 3×3 matrix, and the dimensions must fit to carry out the matrix multiplications. Similarly, (3.25) shows that $\boldsymbol{J}_p$ and $\boldsymbol{J}_o$ are 2×2 and 1×2 matrices, respectively. However, in (3.24) they are expanded to be 3×2 matrices by adding one zero row at the bottom of $\boldsymbol{J}_p$ and two zero rows at the top of $\boldsymbol{J}_o$.

Remark 2. The Jacobian includes a singularity that occurs when $\boldsymbol{a}_{des} = \boldsymbol{a}$. This is resolved in the implementation by dividing by a small number ϵ rather than the error norm if the norm is smaller than ϵ [4].

ζ_2 - Task 2 - First joint - 1 DOF: The first joint of manipulator 2 should be kept constant: $q_{21} = \pi/2$. This ensures that manipulator 2 operates over the

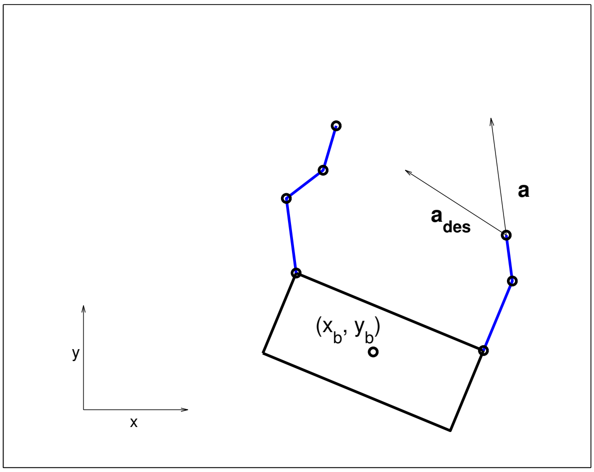


Figure 3.2: Illustration of $\boldsymbol{a}_{\mathrm{des}}$ and $\boldsymbol{a}$ for relative field of view.

vehicle, similarly to Task 3 for manipulator 1:

$$\boldsymbol{\sigma}_{22}(\boldsymbol{q}_2) = q_{21}, \tag{3.26}$$

$$\boldsymbol{\sigma}_{22,\mathrm{des}} = \frac{\pi}{2}, \tag{3.27}$$

$$\dot{\boldsymbol{\sigma}}_{22} = \dot{q}_{21} = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{\zeta}_2 = \boldsymbol{J}_{22}\boldsymbol{\zeta}_2. \tag{3.28}$$

3.3 Guidance System

This section presents the guidance system that calculates the desired system trajectories. With the proposed division of $\boldsymbol{\zeta}_1$ and $\boldsymbol{\zeta}_2$ a general task and the corresponding Jacobian can be expressed as

$$\boldsymbol{\sigma}_1 = \boldsymbol{f}(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1) \tag{3.29}$$

$$\dot{\boldsymbol{\sigma}}_1 = \boldsymbol{J}_1(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1)\boldsymbol{\zeta}_1 \tag{3.30}$$

$$\boldsymbol{\sigma}_2 = \boldsymbol{g}(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1, \boldsymbol{q}_2) \tag{3.31}$$

$$\dot{\boldsymbol{\sigma}}_2 = \boldsymbol{J}_2(\boldsymbol{\eta}_{\mathrm{b}}, \boldsymbol{q}_1, \boldsymbol{q}_2)\boldsymbol{\zeta}_2. \tag{3.32}$$

Note that tasks related to the vehicle/manipulator 1 are completely independent on manipulator 2 ($\boldsymbol{q}_2$) due to the fact that this is considered the main manipulator/leader. Tasks related to manipulator 2, however, may generally depend on the configuration of the vehicle, manipulator 1 and 2.

For a single task, the desired $\boldsymbol{\zeta}$ can be calculated as

$$\boldsymbol{\zeta}_{\mathbf{1},\text{des}} = \boldsymbol{J}_1^{\dagger}(\dot{\boldsymbol{\sigma}}_{1,\text{des}} + \boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1), \tag{3.33}$$

$$\boldsymbol{\zeta}_{\mathbf{2},\text{des}} = \boldsymbol{J}_2^{\dagger}(\dot{\boldsymbol{\sigma}}_{2,\text{des}} + \boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2), \tag{3.34}$$

where $\boldsymbol{\Lambda}$ is a positive definite gain matrix and $\tilde{\boldsymbol{\sigma}} \triangleq \boldsymbol{\sigma}_{\text{des}} - \boldsymbol{\sigma}$. This is illustrated in Figure 3.3. Note that the calculated, desired state of the UVMS is used as feedback in the guidance system rather than the actual state of the UVMS. In other words, the guidance system is independent of the actual behavior of the UVMS and only generates trajectories that, if fulfilled, will result in the best possible achievement of the given tasks. It is a common assumption in kinematic control to assume that the generated reference state is tracked [3].

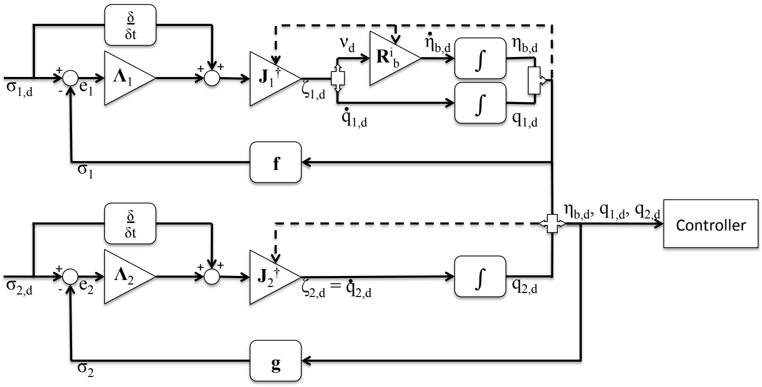


Figure 3.3: Block diagram of the proposed guidance system when one task for the leader and one task for the follower is implemented ((3.33) and (3.34)). Triangular blocks are gains and rectangular blocks are functions. Based on the desired tasks a reference for $\boldsymbol{\eta}_{\text{b}}$, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ is calculated and sent to the controller system.

In Section 3.2, several tasks have been defined with a certain priority. As such, it is desirable to find a $\boldsymbol{\zeta}_{1,\text{des}}$ and $\boldsymbol{\zeta}_{2,\text{des}}$ that, if fulfilled, will result in achievement of all tasks. In case of conflicting tasks this might not be possible, in which case the goal is to find the best possible solution with respect to the defined priorities. This is done with the singularity-robust multiple task-priority inverse kinematics framework. Before adding the contribution of a lower-priority task to the overall desired vehicle velocity, it is projected onto the null-space of the immediately higher-priority task so as to remove velocity components that would conflict with it. The total proposed guidance law for the considered system is thus given by (3.35) and (3.36).

$$\begin{aligned}\boldsymbol{\zeta}_{\mathbf{1},\text{des}} = {} & \boldsymbol{J}_{11}^{\dagger}(\dot{\boldsymbol{\sigma}}_{11,\text{des}} + \boldsymbol{\Lambda}_{11}\tilde{\boldsymbol{\sigma}}_{11}) + (\boldsymbol{I} - \boldsymbol{J}_{11}^{\dagger}\boldsymbol{J}_{11})\boldsymbol{J}_{12}^{\dagger}(\dot{\boldsymbol{\sigma}}_{12,\text{des}} + \boldsymbol{\Lambda}_{12}\tilde{\boldsymbol{\sigma}}_{12}) \\ & + (\boldsymbol{I} - \begin{bmatrix}\boldsymbol{J}_{11}\\ \boldsymbol{J}_{12}\end{bmatrix}^{\dagger}\begin{bmatrix}\boldsymbol{J}_{11}\\ \boldsymbol{J}_{12}\end{bmatrix})\boldsymbol{J}_{13}^{\dagger}(\dot{\boldsymbol{\sigma}}_{13,\text{des}} + \boldsymbol{\Lambda}_{13}\tilde{\boldsymbol{\sigma}}_{13}),\end{aligned} \tag{3.35}$$

$$\boldsymbol{\zeta}_{\mathbf{2},\text{des}} = \boldsymbol{J}_{21}^{\dagger}(\dot{\boldsymbol{\sigma}}_{21,\text{des}} + \boldsymbol{\Lambda}_{21}\tilde{\boldsymbol{\sigma}}_{11}) + (\boldsymbol{I} - \boldsymbol{J}_{21}^{\dagger}\boldsymbol{J}_{21})\boldsymbol{J}_{22}^{\dagger}(\dot{\boldsymbol{\sigma}}_{22,\text{des}} + \boldsymbol{\Lambda}_{22}\tilde{\boldsymbol{\sigma}}_{22}). \tag{3.36}$$

3.4 Simulation Results

The UVMS described in Section 3.1, the tasks listed in Section 3.2 and the guidance laws (3.35) and (3.36) have been implemented using Matlab. This section presents simulation results of the UVMS behavior if the calculated desired states are tracked perfectly by the controller.

In the presented simulation, manipulator 1 (task 1) has been tasked with tracking a straight line trajectory with a constant velocity between a defined start and end point. Furthermore, the desired orientation is constant and normal to the line. The desired vertical distance between the vehicle and end effector 1 (task 3) has been chosen as $C = \frac{H}{2} + \frac{2}{3}\left(l_{11} + l_{12} + l_{13}\right)$. The guidance law error gains have been chosen as follows:

$$\mathbf{\Lambda}_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{\Lambda}_{12} = 3, \mathbf{\Lambda}_{13} = 2, \mathbf{\Lambda}_{21} = 1, \mathbf{\Lambda}_{22} = 3. \tag{3.37}$$

Finally, the UVMS has been implemented with a saturation on the linear and angular velocities. A method for ensuring that the saturation is not reached is described in [8]. This approach has not been implemented in this chapter.

$$|u| \leq 3\frac{\text{m}}{\text{s}}, \quad |v| \leq 2\frac{\text{m}}{\text{s}}, \quad |r| \leq 3\frac{\text{deg}}{\text{s}}, \quad |\dot{q}_{ij}| \leq 10\frac{\text{deg}}{\text{s}}. \tag{3.38}$$

The simulation results are shown in Figure 3.4(a)-3.4(d) and confirm that the calculated reference values will in fact, if fulfilled, result in achievement of the implemented tasks. Furthermore, Figure 3.5(a) and 3.5(b) show that the task errors all converge to zero. Note that higher priority tasks converge before the tasks with lower priority.

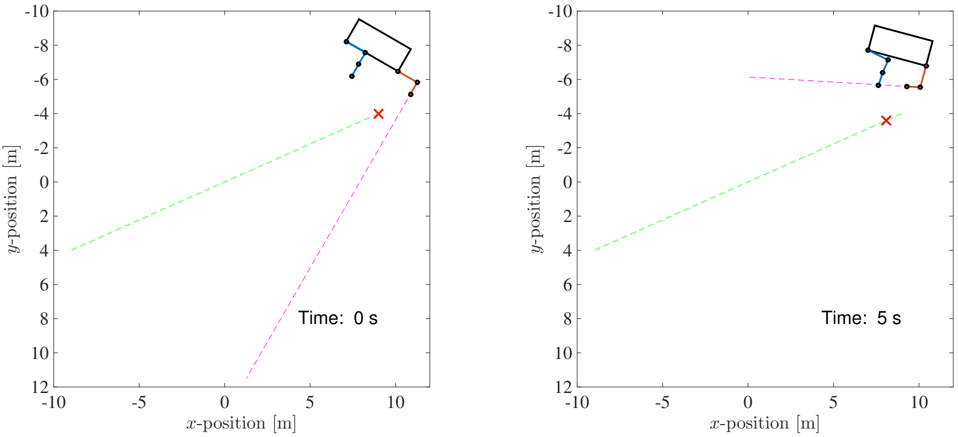
3.5 The Need for Set-based Tasks

This chapter has illustrated the implementation and behavior of the singularity-robust multiple task-priority inverse kinematics framework. Furthermore, a control scheme for a dual-manipulator floating base system has been proposed. In doing so, the need for set-based tasks has become apparent. The existing framework is based on equality tasks, i.e. tasks with a specific desired value. For certain scenarios, defining a $\boldsymbol{\sigma}_{\text{des}}$ might not desirable or even possible to do. Rather, the task should be specified with a certain valid set of values. From here on, we refer to this type of task as a set-based task.

Consider, for instance, task 3 for $\boldsymbol{\zeta}_1$ (σ_{13}) given in Section 3.2. In this particular example, the desired vertical distance between the body center and the end effector is defined as some positive constant C to ensure that the manipulator operates above the vehicle body. However, this could also be achieved by enforcing

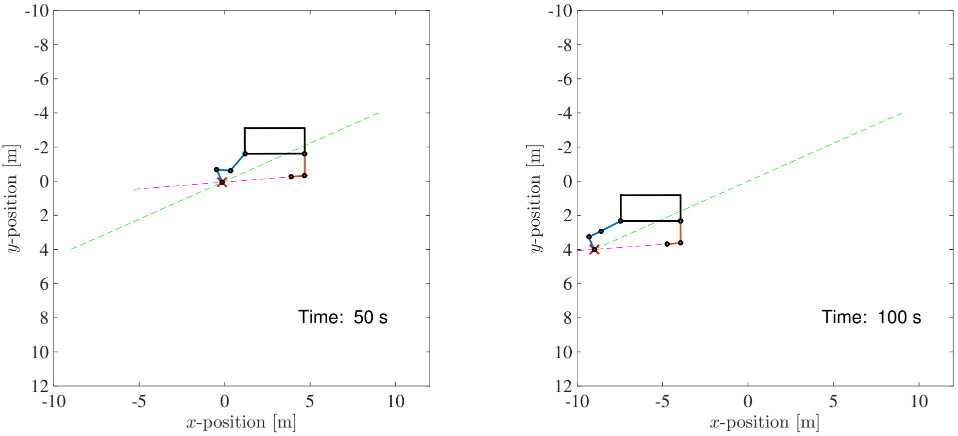
$$\sigma_{13} \in [C_{\min}, C_{\max}] \tag{3.39}$$

for some $C_{\max} > C_{\min} > 0$. This provides the system with more freedom to achieve the other tasks: by defining a valid interval, there exists numerous (in principle an



(a) Time = 0 s.

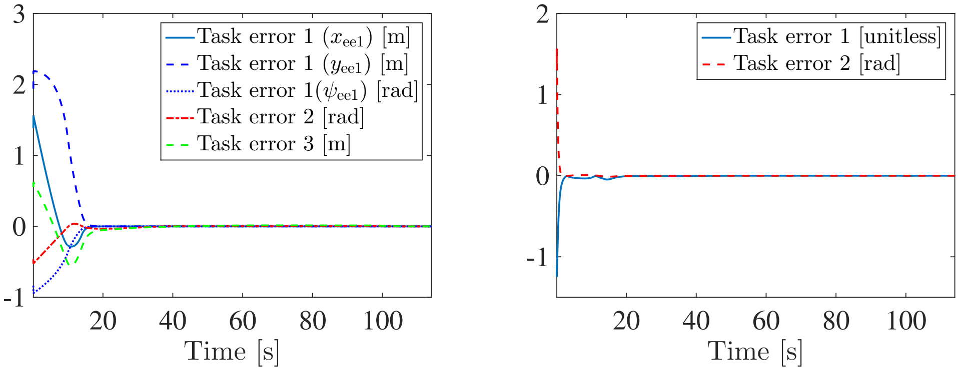
(b) Time = 5 s.



(c) Time = 50 s.

(d) Time = 100 s.

Figure 3.4: Simulation results: The green dashed line shows the desired trajectory for end effector 1. The red cross is moving along the line and is the reference position. The purple dashed line shows the pointing direction of manipulator 2 $\boldsymbol{a}_{\text{des}}$. The results show that all the tasks are fulfilled during the simulation.



(a) **(b)**

Figure 3.5: Simulation results: Task errors for manipulator 1 and vehicle body (a) and manipulator 2 (b).

infinite) amount of valid configurations for this task, which again implies numerous possible means to achieve the remaining tasks.

Similarly, task 1 for ζ_2 (σ_{21}) is defined as the Field of View, where manipulator 2 is tasked to point straight towards the end effector of manipulator 1. This is for instance a relevant task if manipulator 2 has a camera to provide visual feedback of the intervention of the main manipulator. In this case, however, it is not necessary to point the camera directly at the end effector by enforcing $\boldsymbol{a} = \boldsymbol{a}_{\mathrm{des}}$ (see Figure 3.2), but allowing $\sigma_{21} = \sqrt{(\boldsymbol{a}_{\mathrm{des}} - \boldsymbol{a})^T(\boldsymbol{a}_{\mathrm{des}} - \boldsymbol{a})}$ to be in a set

$$\sigma_{13} \in [0, C_{\mathrm{max}}], \tag{3.40}$$

where $C_{\mathrm{max}} > 0$ can be constructed to allow a certain number of degrees between the two vectors to ensure that end effector 1 is always in the camera frame, but not necessarily in the center.

Other tasks that are suitable as set-based tasks are collision and joint limit avoidance. It is not necessary nor desirable to actively control the distance to an obstacle or the value of a joint. These tasks should only affect the behavior of the system when it is necessary to handle them as to prevent them from being violated, which leaves the system with greater freedom to accomplish other tasks. With this in mind, we have extended the singularity-robust multiple task-priority inverse kinematics framework to systematically include set-based tasks. The resulting theory and experimental results are presented in Chapters 4, 5 and 6.

3.6 Conclusions

This chapter presents a method for generating reference trajectories for a two-manipulator Underwater Vehicle manipulator System by considering the main manipulator and the vehicle base as a leader unit and the secondary manipulator as a follower unit. This ensures that the two manipulators do not attempt to drive the vehicle base in opposing directions/velocities due to conflicting tasks. The references are calculated using singularity-robust multiple task-priority inverse kinematics framework. The proposed approach has been implemented for a particular UVMS working in the plane and simulated for a defined set of tasks. In particular, the case where one manipulator is dedicated to intervention operations and the other is holding a camera to provide the operator/supervisor a good view of the intervention, has been considered. Relevant tasks for this scenario have been defined, and simulations illustrate the performance of the proposed guidance approach. During this work the need for and usefulness of set-based tasks has become apparent, and an extension of the framework to include these are presented in Chapters 4, 5 and 6.

Chapter 4

High-priority Set-based Tasks

Contributions of this chapter: The singularity-robust multiple task-priority inverse kinematics framework described in Section 2.5 allows equality tasks to be considered in a prioritized order by projecting task velocities through the null-spaces of higher priority tasks. This chapter extends this framework to handle high-priority set-based tasks, i.e. high-priority tasks with a range of valid values, in addition to equality tasks, which have a specific desired value. The proposed method is proven to ensure asymptotic convergence of the equality task errors and the satisfaction of all high-priority set-based tasks. Simulation and experimental results confirm the effectiveness of the proposed approach.

Organization of this chapter: Section 4.1 gives a closer definition of set-based tasks. The proposed method is described in Section 4.2 for high-priority set-based tasks and is analyzed with respect to stability in Section 4.3. Simulation results are given in Section 4.4 and conclusions in Section 4.5.

Publications: The results in this chapter are based on [7, 66, 67].

4.1 Definition of Set-based Tasks

Section 2.5 introduced the concept of multiple task-priority inverse kinematic control for a robotic system as a method to generate reference trajectories for the system configuration that, if satisfied, will result in the successful achievement of several tasks. However, this framework has been developed for equality tasks that have a specific desired value $\boldsymbol{\sigma}_{\mathrm{des}}(t)$, e.g. the desired end effector position. This chapter proposes a method to extend the existing framework to handle high-priority set-based tasks such as the avoidance of joint limits and obstacles, field of view etc.

A set-based task is still expressed through forward kinematics (2.45), but the objective is to keep the task in a defined set D rather than controlling it to a desired value. Mathematically, this can be expressed as

$$\boldsymbol{\sigma}(t) \in D \ \forall \ t, \tag{4.1}$$

rather than

$$\boldsymbol{\sigma}(t) = \boldsymbol{\sigma}_{\text{des}}(t). \tag{4.2}$$

Thus, set-based tasks cannot be directly inserted into the singularity-robust multiple task-priority inverse kinematics framework (2.66) as they have no exact desired value. In Section 4.2 we will present a method that allows a general number of scalar high-priority set-based tasks to be handled with a given priority within a number of equality tasks.

From here on, equality tasks are denoted with number subscripts and set-based tasks with letters. Furthermore, while equality tasks in general can be multidimensional and are thus described as vectors, set-based tasks are scalar and are therefore not expressed in bold-face, e.g. $\boldsymbol{\sigma}_1$ and σ_{a}. Finally, only regulation equality tasks are considered, that is equality tasks to guide the system to a stationary value ($\dot{\boldsymbol{\sigma}}_{\text{des}} \equiv 0$). Finally, it is assumed that the desired joint velocities $\dot{\boldsymbol{q}}_{\text{des}}$ are tracked perfectly by the system, so that $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{\text{des}}$.

Definition. Consider Figure 4.1. A set-based task σ is defined as *satisfied* when it is contained in its valid set, i.e. $\sigma \in D = [\sigma_{\text{min}}, \sigma_{\text{max}}]$, and *violated* if it is not satisfied. Note that on the boundary of D the task is still satisfied.

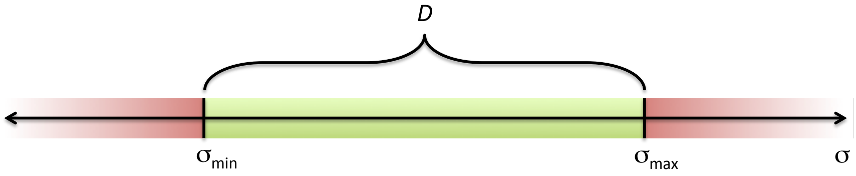


Figure 4.1: Illustration of valid set D. The set-based task σ is satisfied in D and violated outside of D.

4.2 Set-based Singularity-robust Multiple Task-priority Inverse Kinematics

This section presents the proposed method for incorporating high-priority set-based tasks in the singularity-robust multiple task-priority inverse kinematics framework. *High-priority set-based tasks* are defined as all set-based tasks with a higher priority than the highest priority equality task, whereas *low-priority set-based tasks* have priority after at least one equality task. The latter offer some additional challenges and are further described in Chapter 5. For simplicity, a system with a single set-based task is considered first, followed by two, and finally the general case of j set-based tasks.

When a set-based task is in the interior of D (Figure 4.1) it should not affect the behavior of the system. All the system DOFs can then be used to fulfill the equality tasks without being limited in any way by the set-based task, which is *inactive*. The proposed algorithm therefore considers only the system's equality tasks according to the singularity-robust multiple task-priority inverse kinematics framework (2.66) as long as the resulting solution stays within this desired set. If this is not the case, the set-based task is actively inserted into the task priority to be handled. A key aspect of the proposed solution is the *tangent cone* to a set. The tangent cone to the set D in Figure 4.1 at the point $\sigma \in D$ is defined as

$$T_D(\sigma) = \begin{cases} [\,0, \infty\,) & \sigma = \sigma_{\text{min}} \\ \mathbb{R} & \sigma \in (\sigma_{\text{min}}, \sigma_{\text{max}}) \\ (\,-\infty, 0\,] & \sigma = \sigma_{\text{max}} \end{cases} . \tag{4.3}$$

Note that if $\dot{\sigma}(t) \in T_D(\sigma(t)) \ \forall \ t \geq t_0$, then this implies that $\sigma(t) \in \ D \ \forall \ t \geq t_0$. If σ is in the interior of D, the derivative is always in the tangent cone, as this is defined as $\mathbb{R}$. If $\sigma = \sigma_{\text{min}}$, the task is at the lower border of the set. In this case, if $\dot{\sigma} \in [0, \infty)$, then σ will either stay on the border, or move into the interior of the set. Similarly, if $\sigma = \sigma_{\text{max}}$ and $\dot{\sigma} \in (-\infty, 0]$, σ will not leave D.

Due to the fact that a set-based task can be either active or inactive, a system with j set-based tasks has 2^j possible combinations of set-based tasks being active/inactive. These combinations are referred to as "modes" of the system, and the proposed algorithm must switch between modes to fulfill the equality tasks while ensuring that the set-based tasks are not violated. The modes are sorted by increasing restrictiveness. The more set-based tasks that are active in a mode, the more restrictive it is. Hence, in the first mode no set-based tasks are active. This corresponds to considering only the system equality tasks (2.66). In the 2^jth mode all set-based tasks are active.

Throughout this section we consider a robotic system with n DOFs and k equality tasks of m_i DOFs each for $i \in \{1, ...k\}$. Equality task i is denoted $\boldsymbol{\sigma}_i$ and the task error is defined as $\tilde{\boldsymbol{\sigma}}_i = \boldsymbol{\sigma}_{i,\text{des}} - \boldsymbol{\sigma}_i$. Furthermore, the system has j set-based tasks. The first and jth set-based tasks are denoted σ_{a} and σ_{x}, respectively (where x represents the jth letter of the alphabet). We consider the state-vector $\boldsymbol{z} \in \mathbb{R}^l$, where

$$l = n + j + \sum_{i=1}^{k} m_i, \tag{4.4}$$

and

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\sigma}_{\text{sb}} \\ \tilde{\boldsymbol{\sigma}}_{\text{eb}} \end{bmatrix} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \\ \sigma_{\text{a}} \\ \vdots \\ \sigma_{\text{x}} \\ \tilde{\boldsymbol{\sigma}}_1 \\ \vdots \\ \tilde{\boldsymbol{\sigma}}_k \end{bmatrix} . \tag{4.5}$$

Here, $\boldsymbol{\sigma}_{\text{sb}}$ and $\tilde{\boldsymbol{\sigma}}_{\text{eb}}$ are defined as vectors containing the set-based tasks and the corresponding valid set for $\boldsymbol{z}$ in which all set-based tasks are satisfied is defined as

$$C := \mathbb{R}^n \times C_1 \times C_2 \times ... \times C_j \times \mathbb{R}^{l-n-j}, \tag{4.6}$$

where

$$C_1 := [\sigma_{\mathrm{a,min}}, \sigma_{\mathrm{a,max}}], \tag{4.7}$$

$$\vdots$$

$$C_j := [\sigma_{\mathrm{x,min}}, \sigma_{\mathrm{x,max}}]. \tag{4.8}$$

Furthermore, we make the same assumptions as in [3] on the independence of tasks and task gains:

Assumption 4.1: When an additional task is considered, the task Jacobian is independent with respect to the Jacobian obtained by stacking all the higher priority tasks, i.e.

$$\rho\left(\boldsymbol{J}^{\mathrm{A}}_{12..(i-1)}{}^{\dagger}\right) + \rho\left(\boldsymbol{J}_i^{\dagger}\right) = \rho\left(\begin{bmatrix}\boldsymbol{J}^{\mathrm{A}}_{12..(i-1)}{}^{\dagger} & \boldsymbol{J}_i^{\dagger}\end{bmatrix}\right) \tag{4.9}$$

for $i \in \{2, ..., (j+k)\}$.

Remark 4.1*:* Assumption 4.1 requires that the tasks are compatible. For instance, if the system is given one end effector position tracking task and one collision avoidance task, and the desired trajectory moves through the obstacle, the tasks are clearly not compatible. In this case, Assumption 4.1 is not satisfied, and the system will fulfill the highest priority task.

Assumption 4.2: The task gains are chosen according to [3]. For the specific case of $(j+k) = 3$, the task gains are chosen as $\boldsymbol{\Lambda}_1 = \lambda_1 \boldsymbol{I}_{m_1}$, $\boldsymbol{\Lambda}_2 = \lambda_2 \boldsymbol{I}_{m_2}$ and $\boldsymbol{\Lambda}_3 = \lambda_3 \boldsymbol{I}_{m_3}$ for the first, second and third priority task respectively, with

$$\lambda_1 > 0, \tag{4.10}$$

$$\lambda_2 > \max\left(0, \frac{\overline{\lambda}_{21} - \underline{\lambda}_{11}}{\underline{\lambda}_{22}}\lambda_1\right), \tag{4.11}$$

$$\lambda_3 > \max\left(0, \frac{\overline{\lambda}_{31} - \underline{\lambda}_{11}}{\underline{\lambda}_{33}}\lambda_1, \frac{\overline{\lambda}_{32} - \underline{\lambda}_{22}}{\underline{\lambda}_{33}}\lambda_2\right), \tag{4.12}$$

where $\overline{\lambda}_{ij}$ and $\underline{\lambda}_{ij}$ denote the largest and smallest singular value of the matrix $\boldsymbol{P}_{ij}$, respectively, and

$$\begin{array}{ll} \boldsymbol{P}_{11} = \boldsymbol{I}_{m_1}, & \boldsymbol{P}_{22} = \boldsymbol{J}_2 \boldsymbol{N}_1 \boldsymbol{J}_2^{\dagger}, \\ \boldsymbol{P}_{21} = \boldsymbol{J}_2 \boldsymbol{J}_1^{\dagger}, & \boldsymbol{P}_{32} = \boldsymbol{J}_3 \boldsymbol{N}_1 \boldsymbol{J}_2^{\dagger}, \\ \boldsymbol{P}_{31} = \boldsymbol{J}_3 \boldsymbol{J}_1^{\dagger}, & \boldsymbol{P}_{33} = \boldsymbol{J}_3 \boldsymbol{N}_{12}^{A} \boldsymbol{J}_3^{\dagger}. \end{array} \tag{4.13}$$

4.2.1 One set-based task, k equality tasks

For simplicity we first consider a robotic system with a single high-priority set-based task $\sigma_{\mathrm{a}} \in \mathbb{R}$. In this section, we choose σ_{a} as a collision avoidance task as an example, with the goal of avoiding a circular obstacle at a constant position $\boldsymbol{p}_{\mathrm{o}}$ with radius $r > 0$. The task is defined as the distance between the end effector and the obstacle center. The kinematics and Jacobian of this task are given in [2]:

$$\sigma_{\mathrm{a}} = \sqrt{(\boldsymbol{p}_{\mathrm{o}} - \boldsymbol{p}_{\mathrm{e}})^T (\boldsymbol{p}_{\mathrm{o}} - \boldsymbol{p}_{\mathrm{e}})}, \tag{4.14}$$

$$\dot{\sigma}_{\mathrm{a}} = \boldsymbol{J}_{\mathrm{a}} \dot{\boldsymbol{q}} = -\frac{(\boldsymbol{p}_{\mathrm{o}} - \boldsymbol{p}_{\mathrm{e}})^T}{||\boldsymbol{p}_{\mathrm{o}} - \boldsymbol{p}_{\mathrm{e}}||} \boldsymbol{J} \dot{\boldsymbol{q}}. \tag{4.15}$$

Here, $\boldsymbol{p}_{\mathrm{e}}$ denotes the position of the end effector and $\boldsymbol{J}$ is the corresponding position Jacobian. For this specific example we define

$$C_1 = [\epsilon, \infty) \tag{4.16}$$

for an $\epsilon > r$ and the set C as in (4.6) with $j = 1$ and C_1 as defined above. In C, the set-based task is limited to $[\epsilon, \infty)$. Thus, the set-based task is always satisfied for $\boldsymbol{z} \in C$.

For a system with one high-priority set-based task, two modes must be considered:

1. Ignoring the set-based task and considering only the equality tasks.
2. Freezing the set-based task as first priority and considering the equality tasks as second priority.

Mode 1 is the "default" solution, whereas mode 2 should be activated only when it is necessary to prevent the set-based task from being violated. Using the multiple task-priority inverse kinematics framework presented in Section 2.5, mode 1 corresponds to (2.66) and results in the following system:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \boldsymbol{N}_{12}^{\mathrm{A}}\boldsymbol{J}_3^{\dagger}\boldsymbol{\Lambda}_3\tilde{\boldsymbol{\sigma}}_3 + \ldots + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k, \tag{4.17}$$

$$\Downarrow$$

$$\dot{\sigma}_{\mathrm{a}} = \boldsymbol{J}_{\mathrm{a}}\dot{\boldsymbol{q}} = \boldsymbol{J}_{\mathrm{a}}(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \ldots + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k), \tag{4.18}$$

$$\begin{aligned}
\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} &= -\dot{\boldsymbol{\sigma}}_{\mathrm{eb}} = -\begin{bmatrix} \dot{\boldsymbol{\sigma}}_1 \\ \dot{\boldsymbol{\sigma}}_2 \\ \vdots \\ \dot{\boldsymbol{\sigma}}_k \end{bmatrix} = -\begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \\ \vdots \\ \boldsymbol{J}_k \end{bmatrix}\dot{\boldsymbol{q}} \\
&= -\begin{bmatrix} \boldsymbol{J}_1(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k) \\ \boldsymbol{J}_2(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k) \\ \vdots \\ \boldsymbol{J}_k(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k) \end{bmatrix} \\
&= -\begin{bmatrix} \boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 \\ \boldsymbol{J}_2\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{J}_2\boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 \\ \vdots \\ \boldsymbol{J}_k\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{J}_k\boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{J}_k\boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k \end{bmatrix} \\
&= -\begin{bmatrix} \boldsymbol{\Lambda}_1 & \boldsymbol{0}_{m_1\times m_2} & \cdots & \boldsymbol{0}_{m_1\times m_k} \\ \boldsymbol{J}_2\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1 & \boldsymbol{J}_2\boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{0}_{m_2\times m_k} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{J}_k\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1 & \boldsymbol{J}_k\boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{J}_k\boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k \end{bmatrix}\begin{bmatrix} \tilde{\boldsymbol{\sigma}}_1 \\ \tilde{\boldsymbol{\sigma}}_2 \\ \vdots \\ \tilde{\boldsymbol{\sigma}}_k \end{bmatrix} \\
&= -\boldsymbol{M}_1\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}, \qquad (4.19)
\end{aligned}$$

$$\dot{\boldsymbol{z}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\sigma}_{\mathrm{a}} \\ \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{f}_{11}(\boldsymbol{z}) \\ f_{12}(\boldsymbol{z}) \\ \boldsymbol{f}_{13}(\boldsymbol{z}) \end{bmatrix} = \boldsymbol{f}_1(\boldsymbol{z}). \tag{4.20}$$

The matrix $\boldsymbol{M}_1$ is identical to the matrix $\boldsymbol{M}$ (2.68), and is positive definite by Assumption 4.1-4.2 [3].

In mode 1, the set-based task evolves freely according to $\dot{\sigma}_{\mathrm{a}} = f_{12}(\boldsymbol{z})$, and the time evolution of $\boldsymbol{z}$ should follow the vector field $\boldsymbol{f}_1(\boldsymbol{z})$ as long as the solution $\boldsymbol{z}$ stays in C. If following $\boldsymbol{f}_1(\boldsymbol{z})$ would result in $\boldsymbol{z}$ leaving the set C, avoiding the obstacle is considered the first priority task and mode 2 is activated. This can only occur on the border of C, that is for $\sigma_{\mathrm{a}} = \epsilon$, and when $\dot{\sigma}_{\mathrm{a}} = f_{12}(\boldsymbol{z}) < 0$. To avoid the obstacle, an equality task with the goal of keeping the current distance to the obstacle is added as the highest priority task. In this case, the desired task value $\sigma_{\mathrm{a,des}}$ is equal to the current task value $\sigma_{\mathrm{a}} = \epsilon$. Thus, the task error $\tilde{\sigma}_{\mathrm{a}} = \sigma_{\mathrm{a,des}} - \sigma_{\mathrm{a}} \equiv 0$. The system is then defined by the following equations:

$$
\begin{aligned}
\dot{\boldsymbol{q}} &= \boldsymbol{J}_{\mathrm{a}}^{\dagger}\tilde{\sigma}_{\mathrm{a}} + \boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{\mathrm{a}1}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \boldsymbol{N}_{\mathrm{a}12}^{\mathrm{A}}\boldsymbol{J}_3^{\dagger}\boldsymbol{\Lambda}_3\tilde{\boldsymbol{\sigma}}_3 \\
&\qquad + \ldots + \boldsymbol{N}_{\mathrm{a}12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k \\
&= \boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{\mathrm{a}1}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{\mathrm{a}12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k, \qquad (4.21)\\
&\Downarrow \\
\dot{\sigma}_{\mathrm{a}} &= \boldsymbol{J}_{\mathrm{a}}\dot{\boldsymbol{q}} = \boldsymbol{J}_{\mathrm{a}}(\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 \ldots + \boldsymbol{N}_{\mathrm{a}12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k) = 0, \qquad (4.22)\\
\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} &= -\dot{\boldsymbol{\sigma}}_{\mathrm{eb}} = -\begin{bmatrix}\dot{\boldsymbol{\sigma}}_1\\ \dot{\boldsymbol{\sigma}}_2\\ \vdots\\ \dot{\boldsymbol{\sigma}}_k\end{bmatrix} = -\begin{bmatrix}\boldsymbol{J}_1\\ \boldsymbol{J}_2\\ \vdots\\ \boldsymbol{J}_k\end{bmatrix}\dot{\boldsymbol{q}} \\
&= -\begin{bmatrix}
\boldsymbol{J}_1(\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{\mathrm{a}1}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{\mathrm{a}12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k)\\
\boldsymbol{J}_2(\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{\mathrm{a}1}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{\mathrm{a}12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k)\\
\vdots\\
\boldsymbol{J}_k(\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{\mathrm{a}1}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{\mathrm{a}12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k)
\end{bmatrix} \\
&= -\begin{bmatrix}
\boldsymbol{J}_1\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1\\
\boldsymbol{J}_2\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{J}_2\boldsymbol{N}_{\mathrm{a}1}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2\\
\vdots\\
\boldsymbol{J}_k\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{J}_k\boldsymbol{N}_{\mathrm{a}1}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{J}_k\boldsymbol{N}_{\mathrm{a}12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k
\end{bmatrix} \\
&= -\begin{bmatrix}
\boldsymbol{J}_1\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1 & \boldsymbol{0}_{m_1\times m_2} & \cdots & \boldsymbol{0}_{m_1\times m_k}\\
\boldsymbol{J}_2\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1 & \boldsymbol{J}_2\boldsymbol{N}_{\mathrm{a}1}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{0}_{m_2\times m_k}\\
\vdots & \vdots & & \vdots\\
\boldsymbol{J}_k\boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1 & \boldsymbol{J}_k\boldsymbol{N}_{\mathrm{a}1}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{J}_k\boldsymbol{N}_{\mathrm{a}12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k
\end{bmatrix}
\begin{bmatrix}\tilde{\boldsymbol{\sigma}}_1\\ \tilde{\boldsymbol{\sigma}}_2\\ \vdots\\ \tilde{\boldsymbol{\sigma}}_k\end{bmatrix} \\
&= -\boldsymbol{M}_2\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}, \qquad (4.23)\\
\dot{\boldsymbol{z}} &= \begin{bmatrix}\dot{\boldsymbol{q}}\\ \dot{\sigma}_{\mathrm{a}}\\ \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}}\end{bmatrix} \triangleq \begin{bmatrix}\boldsymbol{f}_{21}(\boldsymbol{z})\\ 0\\ \boldsymbol{f}_{23}(\boldsymbol{z})\end{bmatrix} = \boldsymbol{f}_2(\boldsymbol{z}). \qquad (4.24)
\end{aligned}
$$

The matrix $\boldsymbol{M}_2$ can be seen as a principal submatrix of the general matrix $\boldsymbol{M}$ in (2.68). Thus, given Assumption 4.1-4.2, $\boldsymbol{M}_2$ is also positive definite. Furthermore, as can be seen by (4.22), the joint velocities (4.21) ensure that the distance to the obstacle is kept constant ($\dot{\sigma}_{\mathrm{a}} = 0$).

Let $\boldsymbol{T}_C(\boldsymbol{z})$ denote the tangent cone to C at the point $\boldsymbol{z} \in C$ and define the set

P as the interior of C.

$$\boldsymbol{T}_C(\boldsymbol{z}) = \begin{cases} \mathbb{R}^l & \boldsymbol{z} \in P \\ \mathbb{R}^n \times [\,0, \infty\,) \times \mathbb{R}^{l-n-1} & \boldsymbol{z} = C \backslash P. \end{cases} \tag{4.25}$$

Consider the continuous functions $\boldsymbol{f}_1, \boldsymbol{f}_2 : C \to \mathbb{R}^l$ as defined in (4.20) and (4.24). The time evolution of $\boldsymbol{z}$ follows the vector field $\boldsymbol{f}_1(\boldsymbol{z})$ as long as the solution $\boldsymbol{z}$ stays in C (mode 1). Using Lemma 5.26 [37] on the system $\dot{\boldsymbol{z}} = \boldsymbol{f}_1(\boldsymbol{z})$, we know that such a solution exists when $\boldsymbol{f}_1(\boldsymbol{x}) \cap \boldsymbol{T}_C(\boldsymbol{x}) \neq \emptyset$ for all $\boldsymbol{x}$ near $\boldsymbol{z}$ (restricting $\boldsymbol{x}$ to C). Hence, we define the set

$$S := \{\boldsymbol{z} \in C : \exists \text{ a neighborhood U of } \boldsymbol{z} : \boldsymbol{f}_1(\boldsymbol{x}) \in \boldsymbol{T}_C(\boldsymbol{x}) \;\; \forall \boldsymbol{x} \in C \cap U\}\,. \tag{4.26}$$

The discontinuous function $\boldsymbol{f} : C \to \mathbb{R}^l$

$$\boldsymbol{f}(\boldsymbol{z}) := \begin{cases} \boldsymbol{f}_1(\boldsymbol{z}) & \boldsymbol{z} \in S \\ \boldsymbol{f}_2(\boldsymbol{z}) & \boldsymbol{z} \in C \backslash S \end{cases} \tag{4.27}$$

then describes our system. The differential equation $\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z})$ then corresponds to following $\boldsymbol{f}_1$ (mode 1) as long as $\boldsymbol{z}$ stays in C and following $\boldsymbol{f}_2$ (mode 2) otherwise. In mode 2, σ_a is frozen, so the $(n+1)$th element in $\boldsymbol{f}_2(\boldsymbol{z}) \equiv 0$. Consequently, $\boldsymbol{f}_2(\boldsymbol{z}) \in \boldsymbol{T}_C(\boldsymbol{z}) \; \forall \boldsymbol{z} \in C$. This implies that C is strongly forward invariant for $\dot{\boldsymbol{z}} = \boldsymbol{f}_2(\boldsymbol{z})$, i.e., that $\boldsymbol{z}(t_0) \in C \Rightarrow \boldsymbol{z}(t) \in C \; \forall t \geq t_0$.

In other words, the set S contains the points $\boldsymbol{z}$ in C such that $\boldsymbol{f}_1(\boldsymbol{x}) \in \boldsymbol{T}_C(\boldsymbol{x})$ for $\boldsymbol{x}$ in C that are near $\boldsymbol{z}$. At the border of C, $\sigma_a = \epsilon$ and the distance between the end effector and the obstacle center is at the minimum allowed value. Therefore, the $(n+1)$th element of $\boldsymbol{f}_1(\boldsymbol{z})$, corresponding to $\dot{\sigma}_\mathrm{a}$, must be zero or positive for $\boldsymbol{z}$ to stay in S. If it is not, mode 2 is activated, which freezes the distance to the obstacle at the border of C. This remains the solution until following $\boldsymbol{f}_1(\boldsymbol{z})$ will result in $\dot{\sigma}_\mathrm{a} \geq 0$, i.e. that the distance between the end effector and obstacle will remain constant or increase.

4.2.2 j set-based tasks, k equality tasks

We now consider the general case of j high-priority set-based tasks and consider the state vector $\boldsymbol{z} \in \mathbb{R}^l$ and the corresponding valid set as defined in (4.5) and (4.6), respectively. Note that in the case that a set-based task σ is a joint limit for joint i, $i \in \{1, ..., n\}$, then $\sigma = q_i$ and should therefore not be included in the vector $\boldsymbol{\sigma}_\mathrm{sb}$ as it is already included in the state vector through $\boldsymbol{q}$. With j set-based tasks, it is necessary to consider the 2^j solutions resulting from all combinations of activating and deactivating every set-based task. These modes are listed in Table 4.1.
All modes have certain commonalities:

1. All active set-based tasks are frozen.
2. Inactive set-based tasks do not affect the behavior of the system.
3. All matrices $\boldsymbol{M}_i$ for $i \in \{1, \ldots, 2^j\}$ are either equal to or a principal submatrix of the general matrix $\boldsymbol{M}$ (2.68). Consequently, given Assumption 4.1-4.2, the matrices $\boldsymbol{M}_i$ are positive definite.

Mode	Description	Equations
1	No set-based tasks active	$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + ... + \boldsymbol{N}^{\mathrm{A}}_{12..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_1\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_1(\boldsymbol{z}).$
2	σ_{a} active	$\dot{\boldsymbol{q}} = \boldsymbol{N}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + ... + \boldsymbol{N}^{\mathrm{A}}_{\mathrm{a}12..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\sigma}_{\mathrm{a}} = 0,$ $\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_2\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_2(\boldsymbol{z}).$
3	σ_{b} active	$\dot{\boldsymbol{q}} = \boldsymbol{N}_{\mathrm{b}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + ... + \boldsymbol{N}^{\mathrm{A}}_{\mathrm{b}12..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\sigma}_{\mathrm{b}} = 0,$ $\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_3\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_3(\boldsymbol{z}).$
$\ddots$	$\ddots$	$\ddots$
j+1	σ_{x} active	$\dot{\boldsymbol{q}} = \boldsymbol{N}_{\mathrm{x}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + ... + \boldsymbol{N}^{\mathrm{A}}_{\mathrm{x}12..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\sigma}_{\mathrm{x}} = 0,$ $\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_{(j+1)}\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_{(j+1)}(\boldsymbol{z}).$
j+2	σ_{a} and σ_{b} active	$\dot{\boldsymbol{q}} = \boldsymbol{N}_{\mathrm{ab}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + ... + \boldsymbol{N}^{\mathrm{A}}_{\mathrm{ab}12..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\sigma}_{\mathrm{a}} = 0,\ \sigma_{\mathrm{b}} = 0$ $\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_{(j+2)}\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_{(j+2)}(\boldsymbol{z}).$
$\ddots$	$\ddots$	$\ddots$
2^j	All set-based tasks active	$\dot{\boldsymbol{q}} = \boldsymbol{N}^{\mathrm{A}}_{\mathrm{ab..x}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + ... + \boldsymbol{N}^{\mathrm{A}}_{\mathrm{ab..x}12..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\boldsymbol{\sigma}}_{\mathrm{sb}} = \boldsymbol{0},$ $\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_{2^j}\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_{2^j}(\boldsymbol{z}).$

Table 4.1: System equations for the resulting 2^j modes for j high-priority set-based tasks.

Consider the corresponding tangent cone to the set C in (4.6).

$$\boldsymbol{T}_C(\boldsymbol{z}) = \mathbb{R}^n \times T_{C_1}(\boldsymbol{z}) \times T_{C_2}(\boldsymbol{z}) \times ... \times T_{C_j}(\boldsymbol{z}) \times \mathbb{R}^{l-n-j}, \tag{4.28}$$

where $T_{C_i}(\boldsymbol{z})$ for $i \in \{1, ..., j\}$ is defined as in (4.3). Defining the sets

$$S_1 := \{\boldsymbol{z} \in C : \exists \text{ a neighborhood U of } \boldsymbol{z} : \boldsymbol{f}_1(\boldsymbol{x}) \in \boldsymbol{T}_C(\boldsymbol{x}) \forall \boldsymbol{x} \in C \cap U\}, \tag{4.29}$$

$$S_2 := \{\boldsymbol{z} \in C \backslash S_1 : \exists \text{ a nbhd U of } \boldsymbol{z} : \boldsymbol{f}_2(\boldsymbol{x}) \in \boldsymbol{T}_C(\boldsymbol{x}) \forall \boldsymbol{x} \in C \cap U\}, \tag{4.30}$$

$$S_3 := \{\boldsymbol{z} \in C \backslash (S_1 \cup S_2) : \exists \text{ a nbhd U of } \boldsymbol{z} : \boldsymbol{f}_3(\boldsymbol{x}) \in \boldsymbol{T}_C(\boldsymbol{x}) \forall \boldsymbol{x} \in C \cap U\}, \tag{4.31}$$

$$\vdots$$

$$S_{2^j} := C \backslash \left(S_1 \cup S_2 \cup S_3 \cup ... \cup S_{2^j - 1}\right), \tag{4.32}$$

the discontinuous equation $\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z})$ with $\boldsymbol{f} : C \to \mathbb{R}^l$ defined as

$$\boldsymbol{f}(\boldsymbol{z}) := \begin{cases} \boldsymbol{f}_1(\boldsymbol{z}) & \boldsymbol{z} \in S_1 \\ \boldsymbol{f}_2(\boldsymbol{z}) & \boldsymbol{z} \in S_2 \\ \boldsymbol{f}_3(\boldsymbol{z}) & \boldsymbol{z} \in S_3 \\ \vdots & \vdots \\ \boldsymbol{f}_{2^j}(\boldsymbol{z}) & \boldsymbol{z} \in S_{2^j} \end{cases} \tag{4.33}$$

then defines our system.

4.3 Stability Analysis

To study the behavior of the discontinuous differential equation (4.33), we look at the corresponding constrained differential inclusion

$$\boldsymbol{z} \in C \quad \dot{\boldsymbol{z}} \in \boldsymbol{F}(\boldsymbol{z}), \tag{4.34}$$

where

$$\boldsymbol{F}(\boldsymbol{z}) := \bigcap_{\delta > 0} \overline{\mathrm{co} \boldsymbol{f}((\boldsymbol{z} + \delta \mathbb{B}) \cap C)} \quad \forall \boldsymbol{z} \in C. \tag{4.35}$$

Here, $\mathbb{B}$ is a unit ball in $\mathbb{R}^{\dim(\boldsymbol{z})}$ centered at the origin and $\overline{\mathrm{co}P}$ denotes the closed convex hull of the set P. In other words, the smallest closed convex set containing P, and $\boldsymbol{z}$ is the Krasovskii solution of the discontinuous differential equation (4.33) (Definition 4.2 [37]).

4.3.1 Convergence of equality tasks

The first part of the stability proof considers the convergence of the system equality tasks when switching between 2^j possible modes of the system.

Theorem 4.1. *Given a switched system defined by (4.33) and Table 4.1. If Assumption 4.1-4.2 hold, the equality task errors $\tilde{\boldsymbol{\sigma}}_{eb}$ converge asymptotically to zero.*

Proof. If V is a continuously differentiable Lyapunov function for $\dot{\boldsymbol{z}} = \boldsymbol{f}_i(\boldsymbol{z})$ for all $i \in \{1, \ldots, 2^j\}$, then V is a Lyapunov function for $\dot{\boldsymbol{z}} \in \boldsymbol{F}(\boldsymbol{z})$. The following equation holds as all $\boldsymbol{f}_i(\boldsymbol{z})$ are continuous functions:

$$\boldsymbol{f} \in \boldsymbol{F}(\boldsymbol{z}) \Rightarrow \boldsymbol{f} = \lambda_1 \boldsymbol{f}_1(\boldsymbol{z}) + \lambda_2 \boldsymbol{f}_2(\boldsymbol{z}) + \ldots + \lambda_{2^j} \boldsymbol{f}_{2^j}(\boldsymbol{z}) \tag{4.36}$$

for some $\sum_{i=1}^{2^j} \lambda_i = 1, \lambda_i \geq 0$. Consider the Lyapunov function candidate for the equality task errors:

$$V(\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}) = \frac{1}{2} \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}^T \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}. \tag{4.37}$$

Using (4.36) and the system equations given in Table 4.1, we find that the time derivative of V is given by

$$\begin{aligned} \dot{V} &= \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}^T (\lambda_1(-\boldsymbol{M}_1 \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}) + \lambda_2(-\boldsymbol{M}_2 \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}) + \ldots + \lambda_{2^j}(-\boldsymbol{M}_{2^j} \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}})) \\ &= -\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}^T (\lambda_1 \boldsymbol{M}_1 + \lambda_2 \boldsymbol{M}_2 + \ldots + \lambda_{2^j} \boldsymbol{M}_{2^j}) \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}} \\ &= -\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}^T \boldsymbol{Q} \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}. \end{aligned} \tag{4.38}$$

The convex combination $\boldsymbol{Q}$ of positive definite matrices is also positive definite. Therefore, $\dot{V}$ is negative definite and $\tilde{\boldsymbol{\sigma}}_{\text{eb}} = \boldsymbol{0}$ is a globally asymptotically stable equilibrium point in all modes. Thus, the equality task errors $\tilde{\boldsymbol{\sigma}}_{\text{eb}}$ asymptotically converge to zero when switching between modes. Furthermore, if $\boldsymbol{q}$ belongs to a compact set, the equilibrium point $\tilde{\boldsymbol{\sigma}}_{\text{eb}} = \boldsymbol{0}$ is exponentially stable. ☐

4.3.2 Satisfaction of set-based tasks and existence of solution

The second part of the stability proof considers the satisfaction of the set-based tasks and the existence of a valid solution.

Theorem 4.2. *Given a switched system defined by (4.33) and Table 4.1, all set-based tasks $\boldsymbol{\sigma}_{sb}$ will be satisfied at all times.*

Proof. We have defined a closed set C in (4.6) within which all set-based tasks are satisfied at all times. As long as the system solution $\boldsymbol{z} \in C$, the set-based tasks are not violated.

For a system with only high-priority set-based tasks, (4.33) defines the system: $\dot{\boldsymbol{z}} = \boldsymbol{f}_1(\boldsymbol{z})$ as long as the solution $\boldsymbol{z}$ stays in C. If the system reaches the boundary of C and remaining in mode 1 would cause $\boldsymbol{z}$ to leave C, another mode is activated. If neither of the vector fields $\boldsymbol{f}_1$-$\boldsymbol{f}_{2^j-1}$ will result in $\boldsymbol{z}$ staying in C, the chosen solution is $\dot{\boldsymbol{z}} = \boldsymbol{f}_{2^j}(\boldsymbol{z})$, for which it has been shown that $\dot{\boldsymbol{\sigma}}_{\text{sb}} \equiv \boldsymbol{0}$. Therefore, $\boldsymbol{f}_{2^j}(\boldsymbol{z}) \in \boldsymbol{T}_C(\boldsymbol{z}) \forall \boldsymbol{z} \in C$, and C is strongly forward invariant for $\dot{\boldsymbol{z}} = \boldsymbol{f}_{2^j}(\boldsymbol{z})$. Thus, there will always exist a solution $\boldsymbol{z} \in C$ and the set-based tasks are consequently always satisfied. ☐

4.4 Simulation Results

To illustrate the effectiveness of the proposed method, a simple example has been implemented in Matlab for a planar three-link manipulator. This section presents the simulations results. Details about the implementation is given in Chapter 6 along with several experimental results.

The simulated example is the case presented in Section 4.2.1: The manipulator has one set-based task σ_{a}, which is to avoid a circular obstacle with center in $(0, 2.6)$ m and radius $r = 0.65$ m. As described above, an $\epsilon > r$ is chosen as a lower limit for the distance between the end effector and the obstacle center. In this simulation, $\epsilon = 0.75$ m. Furthermore, the system has one equality task $\boldsymbol{\sigma}_1 \in \mathbb{R}^3 = [x_{\text{ee}}, y_{\text{ee}}, \psi_{\text{ee}}]^T$ to guide the end effector to a certain position and orientation. Hence the state vector $\boldsymbol{z} \in \mathbb{R}^7$ is defined as

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{q} \\ \sigma_{\text{a}} \\ \tilde{\boldsymbol{\sigma}}_1 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \sigma_{\text{a}} \\ \tilde{x}_{\text{ee}} \\ \tilde{y}_{\text{ee}} \\ \tilde{\psi}_{\text{ee}} \end{bmatrix} \tag{4.39}$$

and the valid set C as

$$C = \mathbb{R}^3 \times [\epsilon, \infty) \times \mathbb{R}^3, \tag{4.40}$$

with the corresponding tangent cone as in (4.25).

In this example, the desired position is $(-2, 3)$ m and the desired orientation is $\frac{\pi}{2}$ radians, so

$$\boldsymbol{\sigma}_{1,\text{des}} = \begin{bmatrix} -2 & 3 & \frac{\pi}{2} \end{bmatrix}^T. \tag{4.41}$$

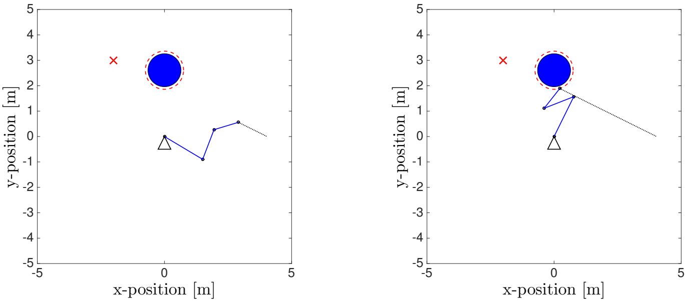
The equality task gain matrix is chosen as the identity matrix, $\boldsymbol{\Lambda}_1 = \boldsymbol{I}$. The manipulator links have length 1.75 m, 1.25 m and 1.0 m, respectively. Furthermore, the manipulator has been implemented with a saturation on the joint velocities of $10 \frac{\text{deg}}{\text{s}}$.

The simulation results are shown in Figure 4.2. The system starts in mode 1, ignoring the obstacle and moving in a straight line towards the target (Figure 4.2(a)). However, by doing so the end effector approaches the obstacle and soon reaches the minimum allowed distance ϵ to the obstacle center (Figure 4.2(b)). Obviously, by continuing to follow the straight line in mode 1, this distance would decrease further. Thus, the system switches to mode 2 and freezes the distance at $\sigma_{\text{a}} = \epsilon$. The end effector then moves towards the desired end effector position along the circle with center in the obstacle and radius ϵ. Eventually it reaches a point where the shortest path between the end effector and desired position does not bring the manipulator closer to the obstacle (Figure 4.2(c)). The system then changes back to mode 1 and converges to the desired position and orientation (Figure 4.2(d) and 4.3(a)). Figure 4.3(b) confirms that the set-based task is satisfied at all times.

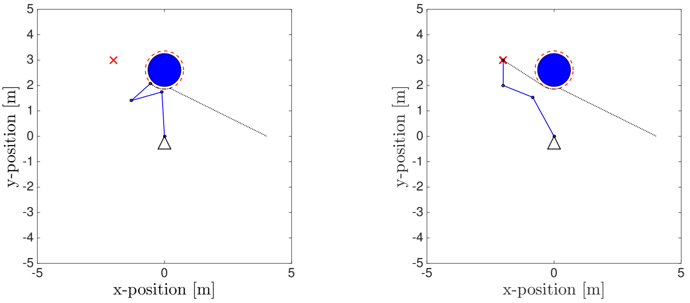
The theoretical results in this chapter are further confirmed by experiments presented in Chapter 6.

4.5 Conclusions

This chapter presents an approach to include high-priority set-based tasks in the multiple task-priority inverse kinematics framework. As a default solution, only the system equality tasks are considered and implemented with the desired priority by projecting the low-priority task velocities through the null-spaces of the high-priority tasks. However, if this solution would result in a set-based task (e.g. a joint limit) being violated, this task is included in the task hierarchy with a certain priority with the goal of freezing the task at its current value. It is shown that set-based tasks given high priority, i.e. above the highest priority equality task, are fulfilled at all times. Due to the switching between set-based tasks being active/inactive, the resulting closed-loop dynamic system can be described as a discontinuous differential equation. Using switched control systems theory it is proven that the equality task errors converge asymptotically to zero when including set-based tasks into the framework given that certain, specified conditions are fulfilled. Furthermore, presented simulation results illustrate the effectiveness of the proposed method.

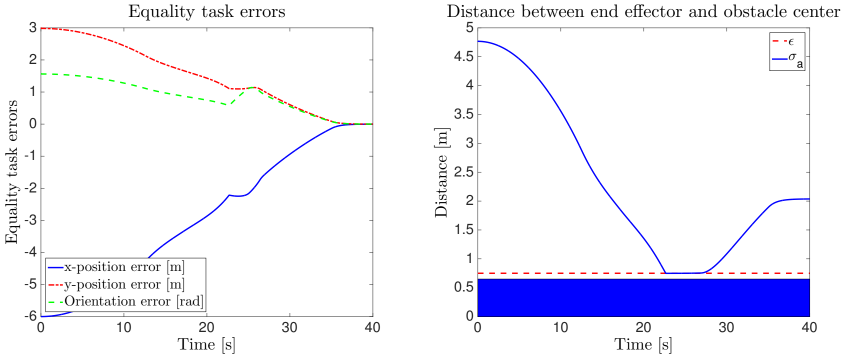


(a) Time = 10 s **(b)** Time = 23 s



(c) Time = 28 s **(d)** Time = 40 s

Figure 4.2: Simulation results: The desired position is marked as a red cross at $(-2, 3)$ m. The obstacle is the blue circle, and the red dashed line around it marks the area within which the end effector is not allowed to ensure avoidance of the obstacle. End effector trajectory is drawn as a dotted black line.



(a) Equality task errors. Both the position and orientation errors converge to 0.

(b) Evolution of distance between end effector and obstacle center. The simulations confirm that the chosen task limit of ϵ is not violated at any time.

Figure 4.3: Equality task errors and the set-based task with its lower limit.

Chapter 5

Set-based Tasks: Low-priority and combination of high- and low-priority

Contributions of this chapter: The singularity-robust multiple task-priority inverse kinematics framework described in Section 2.5 allows equality tasks to be considered in a prioritized order by projecting task velocities through the null-spaces of higher priority tasks. Chapter 4 extended this framework to handle high-priority set-based tasks, i.e. high priority tasks with a range of valid values, in addition to equality tasks, which have a specific desired value. This chapter presents a similar analysis for the case of low-priority set-based tasks, which are tasks that have priority after at least one equality task, and for a combination of high- and low-priority set-based tasks. The proposed method is proven to ensure asymptotic convergence of the equality task errors and the satisfaction of all high-priority set-based tasks. Low-priority set-based tasks can not be guaranteed to be satisfied at all times due to the influence of the higher priority equality tasks. However, if the higher priority equality task errors are zero, the low-priority set-based tasks will also be satisfied. Experimental results that confirm the theory in this and the previous chapter are presented in Chapter 6.

Organization of this chapter: The proposed method is described in Section 5.1 for low-priority (Section 5.1.1) and a combination of high- and low-priority set-based tasks (Section 5.1.2), and is analyzed with respect to stability in Section 5.2. Conclusions are given in Section 5.3.

Publications: The results of this chapter are based on [7, 67].

5.1 Set-based singularity-robust task-priority inverse kinematics

This section presents the proposed method for incorporating set-based tasks in the singularity-robust multiple task-priority inverse kinematics framework. Low-priority set-based tasks offer some additional challenges and are further described in Section 5.1.1, for the case of 1 and j set-based tasks, respectively. In Section 5.1.2 the framework is presented for the completely general case of a combination of high- and low-priority set-based tasks.

In this chapter, the definition of set-based tasks presented in Section 4.1 and the tangent cone T_D (4.3) to a closed set $D = [\sigma_{\text{min}}, \sigma_{\text{max}}]$ are applied also in this chapter. In addition, we define the extended tangent cone to D at the point $\sigma \in \mathbb{R}$ as

$$T_{\mathbb{R},D}(\sigma) = \begin{cases} [\,0, \infty\,) & \sigma \leq \sigma_{\text{min}} \\ \mathbb{R} & \sigma \in (\sigma_{\text{min}}, \sigma_{\text{max}}) \\ (\,-\infty, 0\,] & \sigma \geq \sigma_{\text{max}} \end{cases} . \tag{5.1}$$

The definition of the extended tangent cone is very similar to the tangent cone, but it is defined for $\sigma \in \mathbb{R}$, not just $\sigma \in D$. For $\sigma \notin D$, $\dot{\sigma} \in T_{\mathbb{R},D}$ implies that σ either stays constant or moves closer to D.

Similar to Chapter 4, a system with j set-based tasks has 2^j possible combinations of set-based tasks being active/inactive, and thus 2^j modes. The modes are sorted by increasing restrictiveness.

Throughout this Section we consider a robotic system with n DOFs and k equality tasks of m_i DOFs each for $i \in \{1, ...k\}$. Equality task i is denoted $\boldsymbol{\sigma}_i$ and the task error is defined as $\tilde{\boldsymbol{\sigma}}_i = \boldsymbol{\sigma}_{i,\text{des}} - \boldsymbol{\sigma}_i$. Furthermore, the system has j set-based tasks. The first and jth set-based tasks are denoted σ_{a} and σ_{x}, respectively (where x represents the jth letter of the alphabet). We consider the state-vector $\boldsymbol{z} \in \mathbb{R}^l$, where

$$l = n + j + \sum_{i=1}^{k} m_i, \tag{5.2}$$

and

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\sigma}_{\text{sb}} \\ \tilde{\boldsymbol{\sigma}}_{\text{eb}} \end{bmatrix} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \\ \sigma_{\text{a}} \\ \vdots \\ \sigma_{\text{x}} \\ \tilde{\boldsymbol{\sigma}}_1 \\ \vdots \\ \tilde{\boldsymbol{\sigma}}_k \end{bmatrix} . \tag{5.3}$$

Here, $\boldsymbol{\sigma}_{\text{sb}}$ and $\tilde{\boldsymbol{\sigma}}_{\text{eb}}$ are defined as vectors containing the set-based tasks and the corresponding valid set for $\boldsymbol{z}$ in which all set-based tasks are satisfied is defined as

$$C := \mathbb{R}^n \times C_1 \times C_2 \times ... \times C_j \times \mathbb{R}^{l-n-j}, \tag{5.4}$$

where

$$C_1 := [\sigma_{\text{a,min}}, \sigma_{\text{a,max}}], \tag{5.5}$$

$$\vdots$$

$$C_j := [\sigma_{\text{x,min}}, \sigma_{\text{x,max}}]. \tag{5.6}$$

Furthermore, we make the same assumptions as in [3] on the independence of tasks and task gains, given by Assumpstions 4.1-4.2.

5.1.1 Low-priority set-based tasks

One set-based task, k equality tasks

For simplicity we first consider a robotic system with a single low-priority set-based task $\sigma_{\mathrm{a}} \in \mathbb{R}$ with priority between equality tasks $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ and a valid set $\sigma_{\mathrm{a}} \in [\sigma_{\mathrm{a,min}}, \sigma_{\mathrm{a,max}}]$.

Consider the state-vector $\boldsymbol{z} \in \mathbb{R}^l$ and the closed set C, where l, $\boldsymbol{z}$ and C are defined in (5.2), (5.3) and (5.4), respectively.

Similarly to Section 4.2.1, one set-based task leads to two modes, and in the first mode only the equality tasks are considered. Hence, mode 1 and results in the following system:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \boldsymbol{N}_{12}^{\mathrm{A}}\boldsymbol{J}_3^{\dagger}\boldsymbol{\Lambda}_3\tilde{\boldsymbol{\sigma}}_3 + ... + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k, \tag{5.7}$$

$$\Downarrow$$

$$\dot{\sigma}_{\mathrm{a}} = \boldsymbol{J}_{\mathrm{a}}\dot{\boldsymbol{q}} = \boldsymbol{J}_{\mathrm{a}}(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + ... + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k), \tag{5.8}$$

$$\begin{aligned}
\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} &= -\dot{\boldsymbol{\sigma}}_{\mathrm{eb}} = -\begin{bmatrix}\dot{\boldsymbol{\sigma}}_1\\ \dot{\boldsymbol{\sigma}}_2\\ \vdots\\ \dot{\boldsymbol{\sigma}}_k\end{bmatrix} = -\begin{bmatrix}\boldsymbol{J}_1\\ \boldsymbol{J}_2\\ \vdots\\ \boldsymbol{J}_k\end{bmatrix}\dot{\boldsymbol{q}}\\
&= -\begin{bmatrix}\boldsymbol{J}_1(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + ... + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k)\\ \boldsymbol{J}_2(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + ... + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k)\\ \vdots\\ \boldsymbol{J}_k(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + ... + \boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k)\end{bmatrix}\\
&= -\begin{bmatrix}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1\\ \boldsymbol{J}_2\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{J}_2\boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2\\ \vdots\\ \boldsymbol{J}_k\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{J}_k\boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + ... + \boldsymbol{J}_k\boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k\end{bmatrix}\\
&= -\begin{bmatrix}\boldsymbol{\Lambda}_1 & \boldsymbol{0}_{m_1\times m_2} & \cdots & \boldsymbol{0}_{m_1\times m_k}\\ \boldsymbol{J}_2\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1 & \boldsymbol{J}_2\boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{0}_{m_2\times m_k}\\ \vdots & \vdots & & \vdots\\ \boldsymbol{J}_k\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1 & \boldsymbol{J}_k\boldsymbol{N}_1\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{J}_k\boldsymbol{N}_{12..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{\sigma}}_1\\ \tilde{\boldsymbol{\sigma}}_2\\ \vdots\\ \tilde{\boldsymbol{\sigma}}_k\end{bmatrix}\\
&= -\boldsymbol{M}_1\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},
\end{aligned} \tag{5.9}$$

$$\dot{\boldsymbol{z}} = \begin{bmatrix}\dot{\boldsymbol{q}}\\ \dot{\sigma}_{\mathrm{a}}\\ \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}}\end{bmatrix} \triangleq \begin{bmatrix}\boldsymbol{f}_{11}(\boldsymbol{z})\\ f_{12}(\boldsymbol{z})\\ \boldsymbol{f}_{13}(\boldsymbol{z})\end{bmatrix} = \boldsymbol{f}_1(\boldsymbol{z}). \tag{5.10}$$

The matrix $\boldsymbol{M}_1$ is identical to the matrix $\boldsymbol{M}$ (2.68), and is positive definite by Assumption 4.1-4.2 [3].

In mode 2, the set-based task is actively handled, and as for the higher-priority set-based tasks in Section 4.2.1 the goal is to freeze it at its current value. Hence, $\tilde{\sigma}_{\mathrm{a}} \equiv 0$, and the joint velocities for mode 2 are found by inserting σ_{a} into the framework (2.66) at the given priority level:

$$\begin{aligned}\dot{\boldsymbol{q}} &= \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1\boldsymbol{J}_{\mathrm{a}}^{\dagger}\Lambda_{\mathrm{a}}\tilde{\sigma}_{\mathrm{a}} + \boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{1\mathrm{a}2..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k \\ &= \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{1\mathrm{a}2..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k. \end{aligned} \tag{5.11}$$

In Section 4.2.1, it was shown that mode 2 for high-priority tasks implies $\dot{\sigma}_{\mathrm{a}} = 0$, so if this mode is activated on the border of C ($\sigma_{\mathrm{a}} = \sigma_{\mathrm{a,min}}$ or $\sigma_{\mathrm{a}} = \sigma_{\mathrm{a,max}}$), σ_{a} will indeed be frozen and thereby remain in C. However, when σ_{a} is a low-priority task, the same guarantee cannot be made. Consider the evolution of σ_{a} in case of (5.11):

$$\begin{aligned}\dot{\sigma}_{\mathrm{a}} &= \boldsymbol{J}_{\mathrm{a}}\dot{\boldsymbol{q}} = \boldsymbol{J}_{\mathrm{a}}(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{1\mathrm{a}2..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k) \\ &= \boldsymbol{J}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1. \end{aligned} \tag{5.12}$$

Equation (5.12) is not exactly equal to zero. Rather, the evolution of σ_{a} is influenced by the higher priority equality task. Thus, lower-priority set-based tasks cannot be guaranteed to be satisfied as they can not be guaranteed to be frozen at any given time. Even so, they should still be actively handled by attempting to freeze them, as this might result in these tasks deviating less from their valid set than if they are ignored completely. Thus, we define the system mode 2 as (5.11):

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{1\mathrm{a}2..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k, \tag{5.13}$$

$$\Downarrow$$

$$\dot{\sigma}_{\mathrm{a}} = \boldsymbol{J}_{\mathrm{a}}\dot{\boldsymbol{q}} = \boldsymbol{J}_{\mathrm{a}}\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1, \tag{5.14}$$

$$\begin{aligned}\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} &= -\dot{\boldsymbol{\sigma}}_{\mathrm{eb}} = -\begin{bmatrix}\dot{\boldsymbol{\sigma}}_1 \\ \dot{\boldsymbol{\sigma}}_2 \\ \vdots \\ \dot{\boldsymbol{\sigma}}_k\end{bmatrix} = -\begin{bmatrix}\boldsymbol{J}_1 \\ \boldsymbol{J}_2 \\ \vdots \\ \boldsymbol{J}_k\end{bmatrix}\dot{\boldsymbol{q}} \\
&= -\begin{bmatrix}\boldsymbol{J}_1(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{1\mathrm{a}2..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k) \\ \boldsymbol{J}_2(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{1\mathrm{a}2..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k) \\ \vdots \\ \boldsymbol{J}_k(\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 + \ldots + \boldsymbol{N}_{1\mathrm{a}2..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k)\end{bmatrix} \\
&= -\begin{bmatrix}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 \\ \boldsymbol{J}_2\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{J}_2\boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2\tilde{\boldsymbol{\sigma}}_2 \\ \vdots \\ \boldsymbol{J}_k\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + +\ldots + \boldsymbol{J}_k\boldsymbol{N}_{1\mathrm{a}2..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k\end{bmatrix} \\
&= -\begin{bmatrix}\boldsymbol{\Lambda}_1 & \boldsymbol{0}_{m_1\times m_2} & \cdots & \boldsymbol{0}_{m_1\times m_k} \\ \boldsymbol{J}_2\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1 & \boldsymbol{J}_2\boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{0}_{m_2\times m_k} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{J}_k\boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1 & \boldsymbol{J}_k\boldsymbol{N}_{1\mathrm{a}}^{\mathrm{A}}\boldsymbol{J}_2^{\dagger}\boldsymbol{\Lambda}_2 & \cdots & \boldsymbol{J}_k\boldsymbol{N}_{1\mathrm{a}2..(k-1)}^{\mathrm{A}}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{\sigma}}_1 \\ \tilde{\boldsymbol{\sigma}}_2 \\ \vdots \\ \tilde{\boldsymbol{\sigma}}_k\end{bmatrix} \qquad (5.15) \\
&= -\boldsymbol{M}_2\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}}, \qquad (5.16)\end{aligned}$$

$$\dot{\boldsymbol{z}} = \begin{bmatrix}\dot{\boldsymbol{q}} \\ \dot{\sigma}_{\mathrm{a}} \\ \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}}\end{bmatrix} \triangleq \begin{bmatrix}\boldsymbol{f}_{21}(\boldsymbol{z}) \\ f_{22}(\boldsymbol{z}) \\ \boldsymbol{f}_{23}(\boldsymbol{z})\end{bmatrix} = \boldsymbol{f}_2(\boldsymbol{z}). \tag{5.17}$$

The matrix $\boldsymbol{M}_2$ can be seen as a principal submatrix of the general matrix $\boldsymbol{M}$ in (2.68). Thus, given Assumption 4.1-4.2, $\boldsymbol{M}_2$ is also positive definite.

Consider the continuous functions $\boldsymbol{f}_1, \boldsymbol{f}_2 : C \to \mathbb{R}^l$ as defined in (5.10) and (5.17). The time evolution of $\boldsymbol{z}$ should follow the vector field $\boldsymbol{f}_1(\boldsymbol{z})$ as long as the solution $\boldsymbol{z}$ stays in C (mode 1). However, since it cannot be guaranteed that $\boldsymbol{z}$ stays in C, mode 1 should also be active if $\boldsymbol{z} \notin C$ and following $\boldsymbol{f}_1(\boldsymbol{z})$ will result in $\boldsymbol{z}$ keeping its current distance to or moving closer to C. This is mathematically expressed for the system $\dot{\boldsymbol{z}} = \boldsymbol{f}_1(\boldsymbol{z})$ as $\boldsymbol{f}_1(\boldsymbol{x}) \cap \boldsymbol{T}_{\mathbb{R},C}(\boldsymbol{x}) \neq \emptyset$ for all $\boldsymbol{x}$ near $\boldsymbol{z}$, where

$$\boldsymbol{T}_{\mathbb{R},C}(\boldsymbol{z}) = \mathbb{R}^n \times T_{\mathbb{R},C_1}(\boldsymbol{z}) \times \mathbb{R}^{l-n-1}. \tag{5.18}$$

and $T_{\mathbb{R},C_1}(\boldsymbol{z})$ is the extended tangent cone to C_1 in (5.5) at the point σ_a as defined in (5.1). Hence, we define

$$S := \left\{ \boldsymbol{z} \in \mathbb{R}^l : \exists \text{ a neighborhood U of } \boldsymbol{z} : \boldsymbol{f}_1(\boldsymbol{x}) \in \boldsymbol{T}_{\mathbb{R},C}(\boldsymbol{x}) \ \forall \boldsymbol{x} \in U \right\}. \tag{5.19}$$

The discontinuous function $\boldsymbol{f} : C \to \mathbb{R}^l$

$$\boldsymbol{f}(\boldsymbol{z}) := \begin{cases} \boldsymbol{f}_1(\boldsymbol{z}) & \boldsymbol{z} \in S \\ \boldsymbol{f}_2(\boldsymbol{z}) & \boldsymbol{z} \in \mathbb{R}^l \backslash S \end{cases} \tag{5.20}$$

then describes our system. The differential equation $\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z})$ then corresponds to following $\boldsymbol{f}_1(\boldsymbol{z})$ (mode 1) if one of three conditions are satisfied:

1. $\sigma_\mathrm{a} \in (\sigma_{\mathrm{a,min}}, \sigma_{\mathrm{a,max}})$
2. $\sigma_\mathrm{a} \geq \sigma_{\mathrm{a,max}}$ and $\dot{\boldsymbol{z}} = \boldsymbol{f}_1(\boldsymbol{z}) \Rightarrow \dot{\sigma}_\mathrm{a} \leq 0$
3. $\sigma_\mathrm{a} \leq \sigma_{\mathrm{a,min}}$ and $\dot{\boldsymbol{z}} = \boldsymbol{f}_1(\boldsymbol{z}) \Rightarrow \dot{\sigma}_\mathrm{a} \geq 0$

If none of these conditions hold, mode 2 is activated and $\dot{\boldsymbol{z}} = \boldsymbol{f}_2(\boldsymbol{z})$. In mode 2, the set-based task is actively handled by attempting to freeze it, but since the $(n+1)$th element in $\boldsymbol{f}_2(\boldsymbol{z})$ is not identically equal to 0, this cannot be guaranteed. However, as shown by (5.14), as soon as the higher-priority equality task $\boldsymbol{\sigma}_1$ converges to its desired value, σ_a is indeed frozen in mode 2 and is not affected by the lower-priority equality tasks.

j set-based task, k equality tasks

In this section we consider a system with j low-priority set-based tasks. Consider the state-vector $\boldsymbol{z} \in \mathbb{R}^l$, where l and $\boldsymbol{z}$ are defined in (5.2) and (5.3), respectively. We assume that the set-based tasks are labeled so that σ_b always has lower priority than σ_a etc. Furthermore, assume that σ_a has priority after equality task $\boldsymbol{\sigma}_{p_\mathrm{a}}$ for some $p_\mathrm{a} = \{1, \ldots, k\}$, σ_b has priority after equality task $\boldsymbol{\sigma}_{p_\mathrm{b}}$ for some $p_\mathrm{b} = \{p_a, \ldots, k\}$ and so forth. The resulting 2^j modes of the system are presented in Table 5.1.

All matrices $\boldsymbol{M}_i$ in Table 5.1 for $i \in \left\{1, \ldots, 2^j\right\}$ are either equal to or a principal submatrix of the general matrix $\boldsymbol{M}$ (2.68). Consequently, given Assumption 4.1-4.2, the matrices $\boldsymbol{M}_i$ are positive definite.

Mode	Description	Equations
1	No set-based tasks active	$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + .. + \boldsymbol{N}^{\mathrm{A}}_{12..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_1\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_1(\boldsymbol{z}).$
2	σ_{a} active	$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + .. + \boldsymbol{N}^{\mathrm{A}}_{12..(p_{\mathrm{a}}-1)}\boldsymbol{J}_{p_{\mathrm{a}}}^{\dagger}\boldsymbol{\Lambda}_{p_{\mathrm{a}}}\tilde{\boldsymbol{\sigma}}_{p_{\mathrm{a}}} + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{a}}\mathrm{a}}\boldsymbol{J}_{(p_{\mathrm{a}}+1)}^{\dagger}\boldsymbol{\Lambda}_{(p_{\mathrm{a}}+1)}\tilde{\boldsymbol{\sigma}}_{(p_{\mathrm{a}}+1)} + .. + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{a}}\mathrm{a}(p_{\mathrm{a}}+1)..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_2\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_2(\boldsymbol{z}).$
3	σ_{b} active	$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + .. + \boldsymbol{N}^{\mathrm{A}}_{12..(p_{\mathrm{b}}-1)}\boldsymbol{J}_{p_{\mathrm{b}}}^{\dagger}\boldsymbol{\Lambda}_{p_{\mathrm{b}}}\tilde{\boldsymbol{\sigma}}_{p_{\mathrm{b}}} + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{b}}\mathrm{b}}\boldsymbol{J}_{(p_{\mathrm{b}}+1)}^{\dagger}\boldsymbol{\Lambda}_{(p_{\mathrm{b}}+1)}\tilde{\boldsymbol{\sigma}}_{(p_{\mathrm{b}}+1)} + .. + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{b}}\mathrm{b}(p_{\mathrm{b}}+1)..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_3\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_3(\boldsymbol{z}).$
$\ddots$	$\ddots$	$\ddots$
$j+1$	σ_{x} active	$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + .. + \boldsymbol{N}^{\mathrm{A}}_{12..(p_{\mathrm{x}}-1)}\boldsymbol{J}_{p_{\mathrm{x}}}^{\dagger}\boldsymbol{\Lambda}_{p_{\mathrm{x}}}\tilde{\boldsymbol{\sigma}}_{p_{\mathrm{x}}} + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{x}}\mathrm{x}}\boldsymbol{J}_{(p_{\mathrm{x}}+1)}^{\dagger}\boldsymbol{\Lambda}_{(p_{\mathrm{x}}+1)}\tilde{\boldsymbol{\sigma}}_{(p_{\mathrm{x}}+1)} + .. + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{x}}\mathrm{x}(p_{\mathrm{x}}+1)..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_{(j+1)}\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_{(j+1)}(\boldsymbol{z}).$
$j+2$	σ_{a} and σ_{b} active	$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + .. + \boldsymbol{N}^{\mathrm{A}}_{12..(p_{\mathrm{a}}-1)}\boldsymbol{J}_{p_{\mathrm{a}}}^{\dagger}\Lambda_{p_{\mathrm{a}}}\tilde{\boldsymbol{\sigma}}_{p_{\mathrm{a}}} + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{a}}\mathrm{a}}\boldsymbol{J}_{p_{\mathrm{a}}+1}^{\dagger}\boldsymbol{\Lambda}_{(p_{\mathrm{a}}+1)}\tilde{\boldsymbol{\sigma}}_{(p_{\mathrm{a}}+1)} + .. + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{a}}\mathrm{a}(p_{\mathrm{a}}+1)..(p_{\mathrm{b}}-1)}\boldsymbol{J}_{p_{\mathrm{b}}}^{\dagger}\boldsymbol{\Lambda}_{p_{\mathrm{b}}}\tilde{\boldsymbol{\sigma}}_{p_{\mathrm{b}}} + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{a}}\mathrm{a}(p_{\mathrm{a}}+1)..p_{\mathrm{b}}\mathrm{b}}\boldsymbol{J}_{(p_{\mathrm{b}}+1)}^{\dagger}\boldsymbol{\Lambda}_{(p_{\mathrm{b}}+1)}\tilde{\boldsymbol{\sigma}}_{(p_{\mathrm{b}}+1)} + .. + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{a}}\mathrm{a}(p_{\mathrm{a}}+1)..p_{\mathrm{b}}\mathrm{b}(p_{\mathrm{b}}+1)....(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_{(j+2)}\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_{(j+2)}(\boldsymbol{z}).$
$\ddots$	$\ddots$	$\ddots$
2^j	All set-based tasks active	$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\dagger}\boldsymbol{\Lambda}_1\tilde{\boldsymbol{\sigma}}_1 + .. + \boldsymbol{N}^{\mathrm{A}}_{12..(p_{\mathrm{a}}-1)}\boldsymbol{J}_{p_{\mathrm{a}}}^{\dagger}\Lambda_{p_{\mathrm{a}}}\tilde{\boldsymbol{\sigma}}_{p_{\mathrm{a}}} + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{a}}\mathrm{a}}\boldsymbol{J}_{(p_{\mathrm{a}}+1)}^{\dagger}\boldsymbol{\Lambda}_{(p_{\mathrm{a}}+1)}\tilde{\boldsymbol{\sigma}}_{(p_{\mathrm{a}}+1)} + .. + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{a}}\mathrm{a}(p_{\mathrm{a}}+1)..p_{\mathrm{b}}\mathrm{b}}\boldsymbol{J}_{(p_{\mathrm{b}}+1)}^{\dagger}\boldsymbol{\Lambda}_{(p_{\mathrm{b}}+1)}\tilde{\boldsymbol{\sigma}}_{(p_{\mathrm{b}}+1)} + .. + \boldsymbol{N}^{\mathrm{A}}_{12..p_{\mathrm{a}}\mathrm{a}(p_{\mathrm{a}}+1)..p_{\mathrm{b}}\mathrm{b}(p_{\mathrm{b}}+1)....p_{\mathrm{x}}\mathrm{x}(p_{\mathrm{x}}+1)..(k-1)}\boldsymbol{J}_k^{\dagger}\boldsymbol{\Lambda}_k\tilde{\boldsymbol{\sigma}}_k,$ $\Rightarrow \dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_{2^j}\tilde{\boldsymbol{\sigma}}_{\mathrm{eb}},$ $\dot{\boldsymbol{z}} = \boldsymbol{f}_{2^j}(\boldsymbol{z}).$

Table 5.1: System equations for the resulting 2^j modes for j low-priority set-based tasks.

For example, for a system with $j = 3$ set-based tasks and $k = 6$ equality tasks, $p_\mathrm{a} = 3$ and $p_\mathrm{b} = p_\mathrm{c} = 5$, the $2^j = 8$th mode would be

$$\begin{aligned}\dot{\boldsymbol{q}} = \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1 \boldsymbol{J}_2^\dagger \boldsymbol{\Lambda}_2 \tilde{\boldsymbol{\sigma}}_2 + \boldsymbol{N}_{12}^\mathrm{A} \boldsymbol{J}_3^\dagger \boldsymbol{\Lambda}_3 \tilde{\boldsymbol{\sigma}}_3 \\ + \boldsymbol{N}_{123\mathrm{a}}^\mathrm{A} \boldsymbol{J}_4^\dagger \boldsymbol{\Lambda}_4 \tilde{\boldsymbol{\sigma}}_4 + \boldsymbol{N}_{123\mathrm{a}4}^\mathrm{A} \boldsymbol{J}_5^\dagger \boldsymbol{\Lambda}_5 \tilde{\boldsymbol{\sigma}}_5 + \boldsymbol{N}_{123\mathrm{a}45\mathrm{bc}}^\mathrm{A} \boldsymbol{J}_6^\dagger \boldsymbol{\Lambda}_6 \tilde{\boldsymbol{\sigma}}_6.\end{aligned} \tag{5.21}$$

Consider the sets C_1-C_j defined by (5.5)-(5.6) with corresponding extended tangent cones as defined in (5.1). If the ith set-based task is inactive in a given mode $\boldsymbol{f}$, the corresponding $(n+i)$th element of vector field $\boldsymbol{f}(\boldsymbol{x})$ must be in $T_{\mathbb{R},C_i}(\boldsymbol{x})$ for all $\boldsymbol{x}$ near $\boldsymbol{z}$. If the same task is active, there is no requirement to the same element of the vector field. In mode 1, all the set-based tasks are inactive, thus all the $(n+1)$ to $(n+j)$ elements of $\boldsymbol{f}_1(\boldsymbol{z})$ must be in their respective extended tangent cones for this mode to be chosen. In mode 2, only σ_a is active, so in this case all the $(n+2)$ to $(n+j)$ elements of $\boldsymbol{f}_2(\boldsymbol{z})$ must be in their respective extended tangent cones if mode 2 should be the active mode. Thus, we define

$$\begin{aligned} S_1 :=& \left\{\boldsymbol{z} \in \mathbb{R}^l : \exists \text{ a neighborhood U of } \boldsymbol{z} : \right. \\ & \left. \boldsymbol{f}_1(\boldsymbol{x}) \in \left\{\mathbb{R}^n \times T_{\mathbb{R},C_1}(\boldsymbol{x}) \times ... \times T_{\mathbb{R},C_j}(\boldsymbol{x}) \times \mathbb{R}^{l-n-j}\right\} \forall \boldsymbol{x} \in U\right\}, \end{aligned} \tag{5.22}$$

$$\begin{aligned} S_2 :=& \left\{\boldsymbol{z} \in \mathbb{R}^l \backslash S_1 : \exists \text{ a nbhd U of } \boldsymbol{z} : \right. \\ & \left. \boldsymbol{f}_2(\boldsymbol{x}) \in \left\{\mathbb{R}^n \times \mathbb{R} \times T_{\mathbb{R},C_2}(\boldsymbol{x}) \times ... \times T_{\mathbb{R},C_j}(\boldsymbol{x}) \times \mathbb{R}^{l-n-j}\right\} \forall \boldsymbol{x} \in U\right\}, \end{aligned} \tag{5.23}$$

$$\begin{aligned} S_3 :=& \left\{\boldsymbol{z} \in \mathbb{R}^l \backslash (S_1 \cup S_2) : \exists \text{ a nbhd U of } \boldsymbol{z} : \right. \\ & \left. \boldsymbol{f}_3(\boldsymbol{x}) \in \left\{\mathbb{R}^n \times T_{\mathbb{R},C_1}(\boldsymbol{x}) \times \mathbb{R} \times T_{\mathbb{R},C_3} \times ... \times T_{\mathbb{R},C_j}(\boldsymbol{x}) \times \mathbb{R}^{l-n-j}\right\} \forall \boldsymbol{x} \in U\right\}, \end{aligned} \tag{5.24}$$

$$\vdots$$

$$S_{2^j} := C \backslash \left(S_1 \cup S_2 \cup S_3 \cup ... \cup S_{2^j - 1}\right). \tag{5.25}$$

The discontinuous equation $\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z})$ with $\boldsymbol{f} : C \to \mathbb{R}^l$ defined as

$$\boldsymbol{f}(\boldsymbol{z}) := \begin{cases} \boldsymbol{f}_1(\boldsymbol{z}) & \boldsymbol{z} \in S_1 \\ \boldsymbol{f}_2(\boldsymbol{z}) & \boldsymbol{z} \in S_2 \\ \boldsymbol{f}_3(\boldsymbol{z}) & \boldsymbol{z} \in S_3 \\ \vdots & \vdots \\ \boldsymbol{f}_{2^j}(\boldsymbol{z}) & \boldsymbol{z} \in S_{2^j} \end{cases} \tag{5.26}$$

then defines our system.

5.1.2 Combination of high- and low-priority set-based tasks

Consider an n DOF robotic system with k equality tasks of m_i DOFs for $i \in \{1, \dots, k\}$ and j set-based tasks $\in \mathbb{R}$, where $j_x \leq j$ of them are high priority and $j - j_x$ are low-priority at any given priority level. Denote the j_xth, (j_x+1)th and jth set-based task as σ_x, σ_y and σ_z, respectively, where σ_x is the last high-priority set-based task, σ_y is the first low-priority set-based task and σ_z is the lowest priority set-based task (x, y and z represent the j_xth, $(j_x + 1)$th and jth letter of the alphabet respectively). Consider the state-vector $\boldsymbol{z} \in \mathbb{R}^l$, where

$$l = n + j + \sum_{i=1}^{k} m_i, \tag{5.27}$$

and

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\sigma}_{\mathrm{sb}} \\ \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}} \end{bmatrix} = \left[q_1, .., q_n, \sigma_{\mathrm{a}}, .., \sigma_{\mathrm{x}}, \sigma_{\mathrm{y}}, .., \sigma_{\mathrm{z}}, \tilde{\boldsymbol{\sigma}}_1^T, .., \tilde{\boldsymbol{\sigma}}_k^T\right]^T. \tag{5.28}$$

It can be shown that the 2^j resulting modes from all possible combinations of active and inactive set-based tasks will reduce the error dynamics of the equality tasks to the form

$$\dot{\tilde{\boldsymbol{\sigma}}}_{\mathrm{eb}} = -\boldsymbol{M}_i \tilde{\boldsymbol{\sigma}}_{\mathrm{eb}} \tag{5.29}$$

for $i = \{1, ..., 2^j\}$. The analysis is similar to Section 4.2.2 and 5.1.1. All matrices $\boldsymbol{M}_i$ are either equal to or a principal submatrix of the general matrix $\boldsymbol{M}$ (2.68). Therefore, by Assumption 4.1-4.2, the matrices $\boldsymbol{M}_i$ are positive definite. Furthermore, as described in detail in Section 4.2.2, all the active high-priority set-based tasks are frozen in all modes.

Consider the sets

$$C_1 := [\sigma_{\mathrm{a,min}}, \sigma_{\mathrm{a,max}}], \tag{5.30}$$

$$\vdots$$

$$C_{j_x} := [\sigma_{\mathrm{x,min}}, \sigma_{\mathrm{x,max}}], \tag{5.31}$$

$$C_{j_x+1} := [\sigma_{\mathrm{y,min}}, \sigma_{\mathrm{y,max}}], \tag{5.32}$$

$$\vdots$$

$$C_j := [\sigma_{\mathrm{z,min}}, \sigma_{\mathrm{z,max}}], \tag{5.33}$$

$$C := \mathbb{R}^n \times C_1 \times \cdots \times C_{j_x} \times \mathbb{R}^{j-j_x} \times \mathbb{R}^{l-n-j}, \tag{5.34}$$

and

$$T_{C_1}(\boldsymbol{z}) = \begin{cases} [\,0, \infty\,) & \sigma_{\mathrm{a}} = \sigma_{\mathrm{a},min} \\ \mathbb{R} & \sigma_{\mathrm{a}} \in (\sigma_{\mathrm{a},min}, \sigma_{\mathrm{a},max}) \\ (\,-\infty, 0\,] & \sigma_{\mathrm{a}} = \sigma_{\mathrm{a},max} \end{cases}, \tag{5.35}$$

$$\vdots$$

$$T_{C_{j_x}}(\boldsymbol{z}) = \begin{cases} [\,0, \infty\,) & \sigma_{\mathrm{x}} = \sigma_{\mathrm{x,min}} \\ \mathbb{R} & \sigma_{\mathrm{x}} \in (\sigma_{\mathrm{x,min}}, \sigma_{\mathrm{x,max}}) \\ (\,-\infty, 0\,] & \sigma_{\mathrm{x}} = \sigma_{\mathrm{j,max}} \end{cases}, \tag{5.36}$$

$$T_{\mathbb{R},C_{j_x+1}}(\boldsymbol{z}) = \begin{cases} [\,0, \infty\,) & \sigma_{\mathrm{y}} \leq \sigma_{\mathrm{y,min}} \\ \mathbb{R} & \sigma_{\mathrm{y}} \in (\sigma_{\mathrm{y,min}}, \sigma_{\mathrm{y,max}}) \\ (\,-\infty, 0\,] & \sigma_{\mathrm{y}} \geq \sigma_{\mathrm{y,max}} \end{cases}, \tag{5.37}$$

$$\vdots$$

$$T_{\mathbb{R},C_j}(\boldsymbol{z}) = \begin{cases} [\,0, \infty\,) & \sigma_{\mathrm{z}} \leq \sigma_{\mathrm{z,min}} \\ \mathbb{R} & \sigma_{\mathrm{z}} \in (\sigma_{\mathrm{z,min}}, \sigma_{\mathrm{z,max}}) \\ (\,-\infty, 0\,] & \sigma_{\mathrm{z}} \geq \sigma_{\mathrm{z,max}} \end{cases}, \tag{5.38}$$

$$\boldsymbol{T}_{C_{\mathrm{h}}}(\boldsymbol{z}) = T_{C_1}(\boldsymbol{z}) \times T_{C_2}(\boldsymbol{z}) \times \cdots \times T_{C_{j_x}}(\boldsymbol{z}), \tag{5.39}$$

$$\boldsymbol{T}_C(\boldsymbol{z}) = \mathbb{R}^n \times \boldsymbol{T}_{C_{\mathrm{h}}}(\boldsymbol{z}) \times \mathbb{R}^{j-j_x} \times \mathbb{R}^{l-n-j}. \tag{5.40}$$

Note that $\boldsymbol{z} \in C$ only implies that all the high-priority set-based tasks are within their respective desired sets. Furthermore, the tangent cones are considered for the high-priority set-based tasks, and the extended tangent cones for the low-priority ones. We then define

$$
\begin{aligned}
S_1 := \{\boldsymbol{z} \in C : \exists \text{ a neighborhood U of } \boldsymbol{z} : \boldsymbol{f}_1(\boldsymbol{x}) \in & \qquad (5.41)\\
\Big\{\mathbb{R}^n \times \boldsymbol{T}_{C_{\mathrm{h}}}(\boldsymbol{x}) \times T_{\mathbb{R},C_{j_x+1}}(\boldsymbol{x}) \times .. \times T_{\mathbb{R},C_j}(\boldsymbol{x}) \times \mathbb{R}^{l-n-j}\Big\} \, \forall \boldsymbol{x} \in C \cap U\Big\}, &\\
S_2 := \{\boldsymbol{z} \in C \backslash S_1 : \exists \text{ a nbhd U of } \boldsymbol{z} : \boldsymbol{f}_2(\boldsymbol{x}) \in & \qquad (5.42)\\
\Big\{\mathbb{R}^n \times \boldsymbol{T}_{C_{\mathrm{h}}}(\boldsymbol{x}) \times T_{\mathbb{R},C_{j_x+1}}(\boldsymbol{x}) \times .. \times T_{\mathbb{R},C_j}(\boldsymbol{x}) \times \mathbb{R}^{l-n-j}\Big\} \, \forall \boldsymbol{x} \in C \cap U\Big\}, &\\
\vdots &\\
S_{j_x+1} := \{\boldsymbol{z} \in C \backslash (S_1 \cup .. \cup S_{j_x}) : \exists \text{ a nbhd U of } \boldsymbol{z} : \boldsymbol{f}_{j_x+1}(\boldsymbol{x}) \in & \qquad (5.43)\\
\Big\{\mathbb{R}^n \times \boldsymbol{T}_{C_{\mathrm{h}}}(\boldsymbol{x}) \times T_{\mathbb{R},C_{j_x+1}}(\boldsymbol{x}) \times .. \times T_{\mathbb{R},C_j}(\boldsymbol{x}) \times \mathbb{R}^{l-n-j}\Big\} \, \forall \boldsymbol{x} \in C \cap U\Big\}, &\\
S_{j_x+2} := \{\boldsymbol{z} \in C \backslash (S_1 \cup .. \cup S_{j_x+1}) : \exists \text{ a nbhd U of } \boldsymbol{z} : \boldsymbol{f}_{j_x+2}(\boldsymbol{x}) \in & \qquad (5.44)\\
\Big\{\mathbb{R}^n \times \boldsymbol{T}_{C_{\mathrm{h}}}(\boldsymbol{x}) \times \mathbb{R} \times T_{\mathbb{R},C_{j_x+2}}(\boldsymbol{x}) \times .. \times T_{\mathbb{R},C_j}(\boldsymbol{x}) \times \mathbb{R}^{l-n-j}\Big\} \, \forall \boldsymbol{x} \in C \cap U\Big\}, &\\
S_{j_x+3} := \{\boldsymbol{z} \in C \backslash (S_1 \cup .. \cup S_{j_x+2}) : \exists \text{ a nbhd U of } \boldsymbol{z} : \boldsymbol{f}_{j_x+3}(\boldsymbol{x}) \in & \qquad (5.45)\\
\Big\{\mathbb{R}^n \times \boldsymbol{T}_{C_{\mathrm{h}}}(\boldsymbol{x}) \times T_{\mathbb{R},C_{j_x+1}}(\boldsymbol{x}) \times \mathbb{R} \times T_{\mathbb{R},C_{j_x+3}}(\boldsymbol{x}) \times .. \times T_{\mathbb{R},C_j}(\boldsymbol{x}) \times \mathbb{R}^{l-n-j}\Big\} \, \forall \boldsymbol{x} \in C \cap U\Big\}, &\\
\vdots &\\
S_{2^j} := C \backslash \left(S_1 \cup S_2 \cup S_3 \cup .. \cup S_{2^j-1}\right). & \qquad (5.46)
\end{aligned}
$$

The discontinuous equation $\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z})$ with $\boldsymbol{f} : C \to \mathbb{R}^l$ defined as

$$
\boldsymbol{f}(\boldsymbol{z}) := \begin{cases} \boldsymbol{f}_1(\boldsymbol{z}) & \boldsymbol{z} \in S_1 \\ \boldsymbol{f}_2(\boldsymbol{z}) & \boldsymbol{z} \in S_2 \\ \boldsymbol{f}_3(\boldsymbol{z}) & \boldsymbol{z} \in S_3 \\ \quad \vdots & \quad \vdots \\ \boldsymbol{f}_{2^j}(\boldsymbol{z}) & \boldsymbol{z} \in S_{2^j} \end{cases} \tag{5.47}
$$

then defines our system. Similarly to the analysis of the previous sections, in the sets S_i, $i \in \{1, ..., 2^j\}$, the elements of the vector flows $\boldsymbol{f}_i(\boldsymbol{z})$ corresponding to the high-priority set-based tasks are required to be in their respective tangent cone in all modes, and the elements corresponding to the low-priority set-based tasks are required to be in the corresponding extended tangent cone only in the modes where that task is inactive.

5.2 Stability analysis

To study the behavior of the discontinuous differential equations (5.47), we look at the corresponding constrained differential inclusion:

$$
\boldsymbol{z} \in C \quad \dot{\boldsymbol{z}} \in \boldsymbol{F}(\boldsymbol{z}), \tag{5.48}
$$

where

$$
\boldsymbol{F}(\boldsymbol{z}) := \bigcap_{\delta > 0} \overline{\operatorname{co} \boldsymbol{f}((\boldsymbol{z} + \delta \mathbb{B}) \cap C)} \qquad \forall \boldsymbol{z} \in C. \tag{5.49}
$$

Here, $\mathbb{B}$ is a unit ball in $\mathbb{R}^{\dim(\boldsymbol{z})}$ centered at the origin and $\overline{\operatorname{co} P}$ denotes the closed convex hull of the set P, or in other words, the smallest closed convex set containing P. The state evolution of $\boldsymbol{z}$ is the Krasovskii solution of the discontinuous differential equation (5.47) (Definition 4.2 [37]).

5.2.1 Convergence of equality tasks

The first part of the stability proof considers the convergence of the system equality tasks when switching between 2^j possible modes of the system.

Theorem 5.1. *Given a switched system defined by (5.47). If Assumption 4.1-4.2 hold, the equality task errors $\tilde{\boldsymbol{\sigma}}_{eb}$ converge asymptotically to zero.*

Proof. If V is a continuously differentiable Lyapunov function for $\dot{\boldsymbol{z}} = \boldsymbol{f}_i(\boldsymbol{z})$ for all $i \in \{1, \dots, 2^j\}$, then V is a Lyapunov function for $\dot{\boldsymbol{z}} \in \boldsymbol{F}(\boldsymbol{z})$. The following equation holds as all $\boldsymbol{f}_i(\boldsymbol{z})$ are continuous functions:

$$\boldsymbol{f} \in \boldsymbol{F}(\boldsymbol{z}) \Rightarrow \boldsymbol{f} = \lambda_1 \boldsymbol{f}_1(\boldsymbol{z}) + \lambda_2 \boldsymbol{f}_2(\boldsymbol{z}) + \dots + \lambda_{2^j} \boldsymbol{f}_{2^j}(\boldsymbol{z}) \tag{5.50}$$

for some $\sum_{i=1}^{2^j} \lambda_i = 1, \lambda_i \geq 0$. Consider the Lyapunov function candidate for the equality task errors:

$$V(\tilde{\boldsymbol{\sigma}}_{\text{eb}}) = \frac{1}{2} \tilde{\boldsymbol{\sigma}}_{\text{eb}}^T \tilde{\boldsymbol{\sigma}}_{\text{eb}}. \tag{5.51}$$

Using (5.50) and the system equations (given in Table 4.1 and 5.1 for high-priority and low-priority set-based tasks respectively), we find that the time derivative of V is given by

$$\begin{aligned} \dot{V} &= \tilde{\boldsymbol{\sigma}}_{\text{eb}}^T (\lambda_1(-\boldsymbol{M}_1 \tilde{\boldsymbol{\sigma}}_{\text{eb}}) + \lambda_2(-\boldsymbol{M}_2 \tilde{\boldsymbol{\sigma}}_{\text{eb}}) + \dots + \lambda_{2^j}(-\boldsymbol{M}_{2^j} \tilde{\boldsymbol{\sigma}}_{\text{eb}})) \\ &= -\tilde{\boldsymbol{\sigma}}_{\text{eb}}^T (\lambda_1 \boldsymbol{M}_1 + \lambda_2 \boldsymbol{M}_2 + \dots + \lambda_{2^j} \boldsymbol{M}_{2^j}) \tilde{\boldsymbol{\sigma}}_{\text{eb}} \\ &= -\tilde{\boldsymbol{\sigma}}_{\text{eb}}^T \boldsymbol{Q} \tilde{\boldsymbol{\sigma}}_{\text{eb}}. \end{aligned} \tag{5.52}$$

The convex combination $\boldsymbol{Q}$ of positive definite matrices is also positive definite. Therefore, $\dot{V}$ is negative definite and $\tilde{\boldsymbol{\sigma}}_{\text{eb}} = \boldsymbol{0}$ is a globally asymptotically stable equilibrium point in all modes. Thus, the equality task errors $\tilde{\boldsymbol{\sigma}}_{\text{eb}}$ asymptotically converge to zero when switching between modes. Furthermore, if $\boldsymbol{q}$ belongs to a compact set, the equilibrium point $\tilde{\boldsymbol{\sigma}}_{\text{eb}} = \boldsymbol{0}$ is exponentially stable. ☐

5.2.2 Satisfaction of set-based tasks and existence of solution

The second part of the stability proof considers the satisfaction of the set-based tasks and the existence of a valid solution.

The analysis made in Section 5.1.1 concluded that the evolution of active low-priority set-based tasks are influenced by the higher priority equality tasks. Unlike the high-priority set-based tasks they cannot be frozen at any given time, and therefore they cannot be guaranteed to be satisfied. We have defined a closed set C in (5.34) within which all high-priority set-based tasks are satisfied at all times. As long as the system solution $\boldsymbol{z} \in C$, these tasks are not violated.

Theorem 5.2. *Given a switched system defined by (5.47), all high-priority set-based tasks $\sigma_a - \sigma_x$ will be satisfied at all times.*

Proof. For a system described by (5.47), $\dot{\boldsymbol{z}} = \boldsymbol{f}_1(\boldsymbol{z})$ as long as the solution $\boldsymbol{z} \in C$ and the lower-priority set-based tasks outside of their respective desired sets do not move further away from them. If the system reaches the boundary of C and remaining in mode 1 would cause $\boldsymbol{z}$ to leave C, another mode is activated. If neither of the vector fields $\boldsymbol{f}_1$-$\boldsymbol{f}_{2^j-1}$ will result in $\boldsymbol{z}$ staying in C, the chosen solution is $\dot{\boldsymbol{z}} = \boldsymbol{f}_{2^j}(\boldsymbol{z})$, for which it has been shown that $\dot{\sigma}_{\mathrm{a}}$ to $\dot{\sigma}_{\mathrm{x}} \equiv 0$. Therefore, $\boldsymbol{f}_{2^j}(\boldsymbol{z}) \in \boldsymbol{T}_C(\boldsymbol{z})\ \forall \boldsymbol{z} \in C$, and C is strongly forward invariant for $\dot{\boldsymbol{z}} = \boldsymbol{f}_{2^j}(\boldsymbol{z})$. Thus, there will always exist a solution $\boldsymbol{z} \in C$ and the high-priority set-based tasks are consequently always satisfied. ☐

5.3 Conclusions

This chapter presents an extension to the singularity-robust multiple task-priority inverse kinematics framework that enables set-based tasks to be handled directly. The proposed method allows a general number of scalar set-based tasks to be handled with a given priority within a number of equality tasks. The main purpose of this method is to fulfill the system's equality tasks while ensuring that the set-based tasks are always satisfied, i.e. contained in their valid set. A mathematical framework is presented and concluded with a stability analysis, in which it is proven that high-priority set-based tasks remain in their valid set at all times, whereas lower-priority set-based tasks cannot be guaranteed to be satisfied due to the influence of the higher-priority equality tasks. Furthermore, it is proven that the equality task errors converge asymptotically to zero given certain assumptions.

Chapter 6

Set-based Tasks: Implementation and Experimental Results

Contributions of this chapter: The singularity-robust multiple task-priority inverse kinematics framework described in Section 2.5 allows equality tasks to be considered in a prioritized order by projecting task velocities through the null-spaces of higher priority tasks. In Chapter 4 and 5 this framework was extended to handle set-based tasks, i.e. tasks with a range of valid values, in addition to equality tasks. This chapter presents the proposed implementation of the method. In addition, several experimental results are given that confirm the previously presented theory.

Organization of this chapter: Section 6.1 presents the proposed implementation of the set-based framework described in Chapter 4 and 5. In Section 6.2 the experimental setup is described, and experimental results are presented in Section 6.3. Conclusions are given in Section 6.4.

Publications: The results in this chapter are based on [65, 67].

6.1 Implementation

This section presents the practical implementation of the proposed algorithm and discusses the computational load of running it.

In the stability analysis in Section 5.2 only regulation equality tasks are considered, i.e. tasks with $\dot{\boldsymbol{\sigma}}_{\mathrm{des}} \equiv \mathbf{0}$. For practical purposes, one can also apply this algorithm for time-varying equality tasks. An example of this is presented in Section 6.3.4. Furthermore, in Chapter 5, the lower-priority set-based tasks are handled the same way as the high-priority ones and are attempted frozen. However, as this can not be guaranteed, they might exceed their desired set. With this in mind, for practical purposes these tasks can be handled by defining $\sigma_{\mathrm{des}} = \sigma_{\mathrm{max}}$ or $\sigma_{\mathrm{des}} = \sigma_{\mathrm{min}}$ if $\sigma \geq \sigma_{\mathrm{max}}$ or $\leq \sigma_{\mathrm{min}}$, respectively. Should σ leave the set, the solution will actively attempt to bring σ back to the border of its valid set (rather than simply freezing it outside the set) by having a non-zero error feedback $\tilde{\sigma}$ as it does in for instance (5.11).

In the implementation, the boolean function `in_T_RC` defined in Algorithm 6.1 is used to check if the time-derivative of a set-based task σ with a valid set

$$C = [\sigma_{\min}, \sigma_{\max}] \tag{6.1}$$

is in the extended tangent cone of C, i.e. if $\dot{\sigma} \in T_{\mathbb{R},C}(\sigma)$, where $T_{\mathbb{R},C}(\sigma)$ is defined in (5.1). The algorithm `in_T_RC` is illustrated in Figure 6.1.

Algorithm 6.1: The boolean function `in_T_RC`.

```
Input: $\dot{\sigma}$, $\sigma$, $\sigma_{\min}$, $\sigma_{\max}$
if $\sigma_{\min} < \sigma < \sigma_{\max}$ then
    return True;
else if $\sigma \leq \sigma_{\min}$ and $\dot{\sigma} \geq 0$ then
    return True;
else if $\sigma \leq \sigma_{\min}$ and $\dot{\sigma} < 0$ then
    return False;
else if $\sigma \geq \sigma_{\max}$ and $\dot{\sigma} \leq 0$ then
    return True;
else
    return False;
end
```

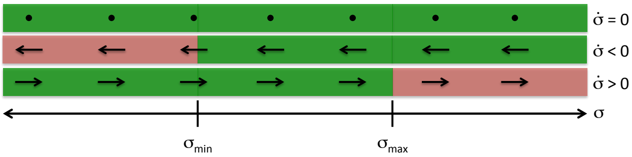


Figure 6.1: Graphic illustration of the function `in_T_RC` with return value `True` shown in green and `False` in red. The function only returns `False` when σ is outside or on the border of its valid set and the derivative points away from the valid set.

Note that in the implementation of the algorithm, `in_T_RC` is used as a check both for the high-priority and low-priority set-based task. This is to handle small numerical inaccuracies that result from discretization of a continuous system. Furthermore, given initial conditions outside the valid set C, the chosen implementation is still well-defined for all set-based tasks.

For a system with j set-based tasks, the 2^j modes must be calculated every timestep. For practical purposes, it is only necessary to calculate the part of the state vector $\boldsymbol{z}$ (5.28)that corresponds to the system configuration $\boldsymbol{q}$, so we calculate this using $\dot{\boldsymbol{q}} = \boldsymbol{f}_i(\boldsymbol{q})$ for $i \in \{1, ..., 2^j\}$, where $\boldsymbol{f}_i(\boldsymbol{q})$ is calculated as described in Chapter 4 and 5. Then, to decide what mode to activate, `in_T_RC` is used to check if every inactive set-based task of that mode returns `True`. If it does, that mode is picked. If not, the next mode is considered.

Recall that $\dot{\boldsymbol{\sigma}}_\mathrm{k} = \boldsymbol{J}_\mathrm{k}\dot{\boldsymbol{q}}$. Consider a system with j set-based tasks and denote the jth set-based task as σ_z. Define the boolean variables ϕ_i

$$\begin{aligned} \phi_i \triangleq \texttt{in\_T\_RC}\left(\boldsymbol{J}_\phi \boldsymbol{f}_i, \sigma_\phi, \sigma_{\phi,\mathrm{min}}, \sigma_{\phi,\mathrm{max}}\right) \\ \text{for } i \in \left\{1, ..., 2^j\right\} \text{ and } \phi \in \{\mathrm{a}, ..., \mathrm{z}\} \end{aligned} \tag{6.2}$$

The active mode is chosen according to Table 6.1. The algorithm iterates through priority 1 to 2^j and activates the first mode that satisfies the criteria presented in Table 6.1. The modes are assigned priority according to their restrictiveness. The more set-based tasks are active, the more restrictive it is. Therefore, the least restrictive mode is $\boldsymbol{f}_1$, where no set-based tasks are active. From there, $\boldsymbol{f}_2 - \boldsymbol{f}_{(j+1)}$ are the modes where one set-based task is active, etc. In Table 6.1 a hyphen indicates that the boolean value does not need to be checked because in that mode, the corresponding set-based task is active and therefore satisfied by definition (in case of a high-priority set-based task) or actively handled to the best of the system's ability (in case of a low-priority set-based task).

Priorities (i)	a_i	b_i	c_i	d_i	...	z_i	Active mode
1	`True`	`True`	`True`	`True`	...	`True`	$\boldsymbol{f}_1$
2	-	`True`	`True`	`True`	...	`True`	$\boldsymbol{f}_2$
3	`True`	-	`True`	`True`	...	`True`	$\boldsymbol{f}_3$
4	`True`	`True`	-	`True`	...	`True`	$\boldsymbol{f}_4$
5	`True`	`True`	`True`	-	...	`True`	$\boldsymbol{f}_5$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$j+1$	`True`	`True`	`True`	`True`	...	-	$\boldsymbol{f}_{(j+1)}$
$j+2$	-	-	`True`	`True`	...	`True`	$\boldsymbol{f}_{(j+2)}$
$j+3$	-	`True`	-	`True`	...	`True`	$\boldsymbol{f}_{(j+3)}$
$j+4$	-	`True`	`True`	-	...	`True`	$\boldsymbol{f}_{(j+4)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$2j$	-	`True`	`True`	`True`	...	-	$\boldsymbol{f}_{(2j)}$
$2j+1$	`True`	-	-	`True`	...	`True`	$\boldsymbol{f}_{(2j+1)}$
$2j+2$	`True`	-	`True`	-	...	`True`	$\boldsymbol{f}_{(2j+2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
2^j	-	-	-	-	$\ddots$	-	$\boldsymbol{f}_{2^j}$

Table 6.1: Table illustrating the activation of mode based on the boolean variables as defined in (6.2). The algorithm iterates through the priorities and checks if the boolean variables satisfy the criteria for activating that mode. The highest priority possible is activated.

6.2 Experimental Setup

This section presents the platform, control structure and the implemented tasks that form the basis for the experimental results presented in Section 6.3.

6.2.1 UR5 and Control Setup

The UR5 is a manipulator with 6 revolute joints, and the joint angles are denoted $\boldsymbol{q} \triangleq \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{bmatrix}^{\mathrm{T}}$. In this chapter, the Denavit-Hartenberg (D-H) parameters are used to calculate the forward kinematics. The parameters are given in [94] and are presented Table 6.2 with the corresponding coordinate frames illustrated in Figure 6.2. The resulting forward kinematics has been experimentally verified to confirm the correctness of the parameters.

Joint	a_i [m]	α_i [rad]	d_i [m]	θ_i [rad]
1	0	$\pi/2$	0.089	q_1
2	-0.425	0	0	q_2
3	-0.392	0	0	q_3
4	0	$\pi/2$	0.109	q_4
5	0	$-\pi/2$	0.095	q_5
6	0	0	0.082	q_6

Table 6.2: Table of the D-H parameters of the UR5. The corresponding coordinate systems can be seen in Figure 6.2.

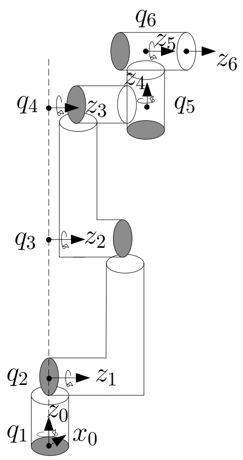


Figure 6.2: Coordinate frames corresponding to the D-H parameters in Table 6.2. Illustration from [94].

The UR5 is equipped with a high-level controller that can control the robot both in joint and Cartesian space. In the experiments presented here, a calculated reference $\boldsymbol{q}_{\text{des}}$ is sent to the high-level controller, which is assumed to function nominally such that

$$\boldsymbol{q} \approx \boldsymbol{q}_{\text{des}}. \tag{6.3}$$

From this reference, $\dot{\boldsymbol{q}}_{\text{des}}$ and $\ddot{\boldsymbol{q}}_{\text{des}}$ are extrapolated and sent with $\boldsymbol{q}_{\text{des}}$ to the low-level controller.

The structure of the system is illustrated in Figure 6.3. The algorithm described in Section 6.1 is implemented in the kinematic controller block. Every timestep, a reference for the joint velocities is calculated and integrated to desired joint angles $\boldsymbol{q}_{\text{des}}$. This is used as input to the dynamic controller, which in turn applies torques to the joint motors. Note that the actual state $\boldsymbol{q}$ is not used for feedback to the kinematic control block. When the current state is used as input for the kinematic controller, the kinematic and dynamic loops are coupled and the gains designed for the kinematic control alone 1 can not be used. This results in uneven motion, and therefore the kinematic control block receives the previous reference as feedback, which leads to much nicer behavior, and is a good approximation because the dynamic controller tracks the reference with very high precision. This is a standard method of implementation for industry robots when kinematic control is used.

The communication between the implemented algorithm and the industrial manipulator system occurs through a TCP/IP connection which operates at 125 Hz. The kinematic control block is implemented using python, which is a highly suitable programming language for the task. The TCP/IP connection is very simple to set up in python. Furthermore, python has several libraries that can handle different math and matrix operations.

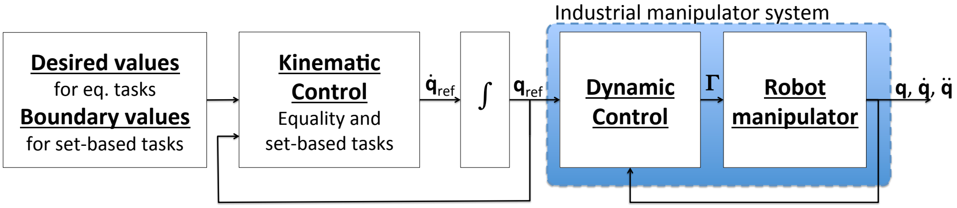


Figure 6.3: The control structure of the experiments. The tested algorithm is implemented in the kinematic controller block.

6.2.2 Implemented Tasks

Three tasks make up the basis for the experiments: Position control, collision avoidance and FOV. Position control and FOV is implemented as both an equality and set-based tasks and collision avoidance as a set-based task.

Position Control

The position of the end effector relative to the base coordinate frame is given by the forward kinematics. The analytical expression can be found through the homogeneous transformation matrix [86] using the D-H parameters given in Table 6.2. The task is then defined by

$$\boldsymbol{\sigma}_{\text{pos}} = \boldsymbol{f}(\boldsymbol{q}) \in \mathbb{R}^3, \tag{6.4}$$

$$\dot{\boldsymbol{\sigma}}_{\text{pos}} = \boldsymbol{J}_{\text{pos}}(\boldsymbol{q})\dot{\boldsymbol{q}} = \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{q}}\dot{\boldsymbol{q}}, \tag{6.5}$$

where the function $\boldsymbol{f}(\boldsymbol{q})$ is given by the forward kinematics.

Collision Avoidance

To avoid a collision between the end effector and an object at position $\boldsymbol{p}_o \in \mathbb{R}^3$, the distance between them is used as a task:

$$\sigma_{\mathrm{CA}} = \sqrt{(\boldsymbol{p}_o - \boldsymbol{\sigma}_{\mathrm{pos}})^{\mathrm{T}}(\boldsymbol{p}_o - \boldsymbol{\sigma}_{\mathrm{pos}})} \in \mathbb{R} \tag{6.6}$$

$$\dot{\sigma}_{\mathrm{CA}} = \boldsymbol{J}_{\mathrm{CA}}(\boldsymbol{q})\dot{\boldsymbol{q}} = -\frac{(\boldsymbol{p}_o - \boldsymbol{\sigma}_{\mathrm{pos}})^{\mathrm{T}}}{\sigma_{\mathrm{CA}}}\boldsymbol{J}_{\mathrm{pos}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{6.7}$$

Field of View

The field of view is defined as the outgoing vector of the end effector, i.e. the z_6-axis in Figure 6.2. This vector expressed in base coordinates is denoted $\boldsymbol{a} \in \mathbb{R}^3$, and can be found through the homogeneous transformation matrix using the D-H parameters:

$$\boldsymbol{a} = \boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^3 \tag{6.8}$$

$$\dot{\boldsymbol{a}} = \boldsymbol{J}_{\mathrm{FOV,3DOF}}(\boldsymbol{q})\dot{\boldsymbol{q}} = \frac{\mathrm{d}\boldsymbol{g}}{\mathrm{d}\boldsymbol{q}}\dot{\boldsymbol{q}} \tag{6.9}$$

FOV is a useful task when directional devices or sensors are mounted on the end-effector and they are desired to point in a certain direction $\boldsymbol{a}_{\mathrm{des}} \in \mathbb{R}^3$. The task is defined as the norm of the error between $\boldsymbol{a}$ and $\boldsymbol{a}_{\mathrm{des}}$:

$$\sigma_{\mathrm{FOV}} = \sqrt{(\boldsymbol{a}_{\mathrm{des}} - \boldsymbol{a})^{\mathrm{T}}(\boldsymbol{a}_{\mathrm{des}} - \boldsymbol{a})} \in \mathbb{R} \tag{6.10}$$

$$\dot{\sigma}_{\mathrm{FOV}} = \boldsymbol{J}_{\mathrm{FOV}}(\boldsymbol{q})\dot{\boldsymbol{q}} = -\frac{(\boldsymbol{a}_{\mathrm{des}} - \boldsymbol{a})^{\mathrm{T}}}{\sigma_{\mathrm{FOV}}}\boldsymbol{J}_{\mathrm{FOV,3DOF}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{6.11}$$

Note that in (6.7) and (6.11) that $\boldsymbol{J}_{\mathrm{CA}}$ and $\boldsymbol{J}_{\mathrm{FOV}}$ are not defined for $\sigma_{\mathrm{CA}} = 0$ and $\sigma_{\mathrm{FOV}} = 0$, respectively. In the implementation, this is solved by adding a small $\epsilon > 0$ to the denominator of these two Jacobians thereby ensuring that division by zero does not occur. Furthermore, the method presented in this chapter ensures collision avoidance, hence σ_{CA} will never be zero. Also note that alternative FOV functions exist which do not suffer from this representation singularity.

6.3 Experimental Results

This section presents four experimental examples to illustrate the effectiveness and the correctness of the theory presented in Chapter 4 and 5.

In Example 1-3, the system has been given two waypoints for the end effector to reach. Here, position control is always the first priority equality task, namely $\boldsymbol{\sigma}_1$.

$$\boldsymbol{p}_{\mathrm{w1}} = \begin{bmatrix}0.486 \text{ m} & -0.066 \text{ m} & -0.250 \text{ m}\end{bmatrix}^{\mathrm{T}}_{,} \quad \text{and} \tag{6.12}$$

$$\boldsymbol{p}_{\mathrm{w2}} = \begin{bmatrix}0.320 \text{ m} & 0.370 \text{ m} & -0.250 \text{ m}\end{bmatrix}^{\mathrm{T}}_{.} \tag{6.13}$$

A circle of acceptance (COA) of 0.02 m is implemented for switching from $\boldsymbol{\sigma}_{1,\mathrm{des}} = \boldsymbol{\sigma}_{\mathrm{pos,des}} = \boldsymbol{p}_{w1}$ to $\boldsymbol{\sigma}_{1,\mathrm{des}} = \boldsymbol{\sigma}_{\mathrm{pos,des}} = \boldsymbol{p}_{w2}$. The task gain matrix has been chosen as

$$\boldsymbol{\Lambda}_1 = \mathrm{diag}(\begin{bmatrix}0.3 & 0.3 & 0.3\end{bmatrix}). \tag{6.14}$$

Furthermore, two obstacles have been introduced, hence two collision avoidance tasks are necessary. In this example, these tasks are high-priority and are denoted σ_a and σ_b, respectively. The obstacles are positioned at

$$\boldsymbol{p}_{\mathrm{o}1} = \begin{bmatrix}0.40 \text{ m} & -0.25 \text{ m} & -0.33 \text{ m}\end{bmatrix}^\mathrm{T} \quad \text{and} \tag{6.15}$$

$$\boldsymbol{p}_{\mathrm{o}2} = \begin{bmatrix}0.40 \text{ m} & 0.15 \text{ m} & -0.33 \text{ m}\end{bmatrix}^\mathrm{T}_{,} \tag{6.16}$$

and have a radius of 0.18 m and 0.15 m, respectively. This radius is used as the minimum value of the set-based collision avoidance task to ensure that the end effector is never closer to the obstacle center than the allowed radius, see Table 6.3, 6.5 and 6.7. Because the task is only considered as a high-priority set-based task, it is not necessary to choose a task gain.

FOV is implemented both as an equality task, a high-priority and a low-priority set-based task in Example 1, 2 and 3, respectively. In these experiments,

$$\boldsymbol{a}_\mathrm{des} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^\mathrm{T}, \tag{6.17}$$

and σ_FOV is defined as in (6.10). In Example 1, FOV is implemented as an equality task. Since σ_FOV is defined as the norm of the error, $\sigma_\mathrm{FOV,des} = 0$. In Examples 2 and 3, FOV is implemented as a set-based task with a maximum value to limit the error between $\boldsymbol{a}$ and $\boldsymbol{a}_\mathrm{des}$. Here, the maximum value for the set-based FOV task is set as 0.2622. This corresponds to allowing the angle between $\boldsymbol{a}_\mathrm{des}$ and $\boldsymbol{a}$ being 15° or less. The gain for this task is chosen as

$$\Lambda_\mathrm{FOV} = 1. \tag{6.18}$$

6.3.1 Example 1

In Example 1, FOV is implemented as an equality task, and the system has two set-based tasks. The task priorities are given in Table 6.3.

Name	Task description	Type	Valid set C
σ_a	Collision avoidance	Set-based	$C_\mathrm{a} = [0.18, \infty)$
σ_b	Collision avoidance	Set-based	$C_\mathrm{b} = [0.15, \infty)$
$\boldsymbol{\sigma}_1$	Position	Equality	-
σ_2	Field of view	Equality	-

Table 6.3: Implemented tasks in Example 1 sorted by decreasing priority.

According to Chapter 4, a system with 2 high-priority set-based tasks have $2^2 = 4$ modes to consider: One containing only the equality tasks, one where σ_a is active, one where σ_b is active and one where σ_a and σ_b are active. However, in this case, the two obstacles have no points of intersection. Hence, it will never be necessary to activate both σ_a and σ_b, and thus the system has three modes:

$$\text{Mode 1:} \quad \dot{\boldsymbol{q}}_\mathrm{ref} = \boldsymbol{f}_1 \triangleq \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1 \boldsymbol{J}_2^\dagger \Lambda_2 \tilde{\sigma}_2 \tag{6.19}$$

$$\text{Mode 2:} \quad \dot{\boldsymbol{q}}_\mathrm{ref} = \boldsymbol{f}_2 \triangleq \boldsymbol{N}_\mathrm{a} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{\mathrm{a}1} \boldsymbol{J}_2^\dagger \Lambda_2 \tilde{\sigma}_2 \tag{6.20}$$

$$\text{Mode 3:} \quad \dot{\boldsymbol{q}}_\mathrm{ref} = \boldsymbol{f}_3 \triangleq \boldsymbol{N}_\mathrm{b} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{\mathrm{b}1} \boldsymbol{J}_2^\dagger \Lambda_2 \tilde{\sigma}_2 \tag{6.21}$$

Denote the boolean variables

$$\phi_i = \texttt{in\_T\_RC}\left(\boldsymbol{J}_\phi \boldsymbol{f}_i, \sigma_\phi, \sigma_{\phi,\min}, \sigma_{\phi,\max}\right) \qquad \text{for } i \in \{1,2,3\} \text{ and } \phi \in \{\mathrm{a},\mathrm{b}\}\,. \tag{6.22}$$

The active mode is then chosen by Table 6.4. Figure 6.5 displays the end effector trajectory and the active mode, the distance between the end effector and the obstacle centers and the FOV task over time.

Priorities (i)	a_i	b_i	Active mode
1	`True`	`True`	$\boldsymbol{f}_1$
2	-	`True`	$\boldsymbol{f}_2$
3	`True`	-	$\boldsymbol{f}_3$

Table 6.4: Table illustrating the activation of mode in Example 1. Note that the tasks σ_a and σ_b are never active in the same mode (an active task is indicated by a hyphen) because the tasks are collision avoidance tasks in the case where the obstacles are not overlapping.

6.3.2 Example 2

In Example 2, FOV is implemented as a high-priority set-based task, and the system has three set-based tasks in total. Table 6.5 displays the system tasks in prioritized order. According to Chapter 4, 3 set-based tasks should result in $2^3 = 8$ modes to consider. However, as in Example 1, we can discard the two modes where both σ_a and σ_b are active. Thus, 6 modes have to be considered:

Name	Task description	Type	Valid set C
σ_a	Collision avoidance	Set-based	$C_\mathrm{a} = [0.18, \infty)$
σ_b	Collision avoidance	Set-based	$C_\mathrm{b} = [0.15, \infty)$
σ_c	Field of view	Set-based	$C_\mathrm{c} = (-\infty, 0.2622]$
$\boldsymbol{\sigma}_1$	Position	Equality	-

Table 6.5: Implemented tasks in Example 2 sorted by decreasing priority.

$$\text{Mode 1:} \qquad \dot{\boldsymbol{q}}_\mathrm{ref} = \boldsymbol{f}_1 \triangleq \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.23}$$

$$\text{Mode 2:} \qquad \dot{\boldsymbol{q}}_\mathrm{ref} = \boldsymbol{f}_2 \triangleq \boldsymbol{N}_\mathrm{a} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.24}$$

$$\text{Mode 3:} \qquad \dot{\boldsymbol{q}}_\mathrm{ref} = \boldsymbol{f}_3 \triangleq \boldsymbol{N}_\mathrm{b} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.25}$$

$$\text{Mode 4:} \qquad \dot{\boldsymbol{q}}_\mathrm{ref} = \boldsymbol{f}_4 \triangleq \boldsymbol{N}_\mathrm{c} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.26}$$

$$\text{Mode 5:} \qquad \dot{\boldsymbol{q}}_\mathrm{ref} = \boldsymbol{f}_5 \triangleq \boldsymbol{N}_\mathrm{ac} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.27}$$

$$\text{Mode 6:} \qquad \dot{\boldsymbol{q}}_\mathrm{ref} = \boldsymbol{f}_6 \triangleq \boldsymbol{N}_\mathrm{bc} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.28}$$

We define the boolean variables

$$\phi_i = \texttt{in\_T\_RC}\left(\boldsymbol{J}_\phi \boldsymbol{f}_i, \sigma_\phi, \sigma_{\phi,\min}, \sigma_{\phi,\max}\right) \qquad \text{for } i \in \{1,...,6\} \text{ and } \phi \in \{\mathrm{a},...,\mathrm{c}\}\,. \tag{6.29}$$

The active mode is then chosen by Table 6.6. Figure 6.5 displays the end effector trajectory and the active mode, the distance between the end effector and the obstacle centers and the FOV task over time.

Priorities (i)	a_i	b_i	c_i	Active mode
1	`True`	`True`	`True`	$\boldsymbol{f}_1$
2	-	`True`	`True`	$\boldsymbol{f}_2$
3	`True`	-	`True`	$\boldsymbol{f}_3$
4	`True`	`True`	-	$\boldsymbol{f}_4$
5	-	`True`	-	$\boldsymbol{f}_5$
6	`True`	-	-	$\boldsymbol{f}_6$

Table 6.6: Table illustrating the activation of mode in Example 2. Note that the tasks σ_{a} and σ_{b} are never active in the same mode (an active task is indicated by a hyphen) because the tasks are collision avoidance tasks in the case where the obstacles are not overlapping.

6.3.3 Example 3

In Example 3, FOV is implemented as a low-priority set-based task (see Table 6.7). The implementation is very similar to Example 2. However, as stated in Chapter 5, lower-priority set-based tasks can not be guaranteed to be satisfied at all times. Hence, if the FOV error exceeds the maximum value of 0.2622, rather than attempting to freeze the task at its current value, an effort is made to push it back to the boundary of the valid set:

Name	Task description	Type	Valid set C
σ_{a}	Collision avoidance	Set-based	$C_{\mathrm{a}} = [0.18, \infty)$
σ_{b}	Collision avoidance	Set-based	$C_{\mathrm{b}} = [0.15, \infty)$
$\boldsymbol{\sigma}_1$	Position	Equality	-
σ_{c}	Field of view	Set-based	$C_{\mathrm{c}} = (-\infty, 0.2622]$

Table 6.7: Implemented tasks in Example 3 sorted by decreasing priority.

$$\tilde{\sigma}_{\mathrm{c}} = \sigma_{\mathrm{FOV,max}} - \sigma_{\mathrm{c}} = 0.2622 - \sigma_{\mathrm{c}} \tag{6.30}$$

$$\text{Mode 1:} \quad \dot{\boldsymbol{q}}_{\mathrm{ref}} = \boldsymbol{f}_1 \triangleq \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.31}$$

$$\text{Mode 2:} \quad \dot{\boldsymbol{q}}_{\mathrm{ref}} = \boldsymbol{f}_2 \triangleq \boldsymbol{N}_{\mathrm{a}} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.32}$$

$$\text{Mode 3:} \quad \dot{\boldsymbol{q}}_{\mathrm{ref}} = \boldsymbol{f}_3 \triangleq \boldsymbol{N}_{\mathrm{b}} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.33}$$

$$\text{Mode 4:} \quad \dot{\boldsymbol{q}}_{\mathrm{ref}} = \boldsymbol{f}_4 \triangleq \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_1 \boldsymbol{J}_{\mathrm{c}}^\dagger \Lambda_{\mathrm{c}} \tilde{\sigma}_{\mathrm{c}} \tag{6.34}$$

$$\text{Mode 5:} \quad \dot{\boldsymbol{q}}_{\mathrm{ref}} = \boldsymbol{f}_5 \triangleq \boldsymbol{N}_{\mathrm{a}} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{\mathrm{a}1} \boldsymbol{J}_{\mathrm{c}}^\dagger \Lambda_{\mathrm{c}} \tilde{\sigma}_{\mathrm{c}} \tag{6.35}$$

$$\text{Mode 6:} \quad \dot{\boldsymbol{q}}_{\mathrm{ref}} = \boldsymbol{f}_6 \triangleq \boldsymbol{N}_{\mathrm{b}} \boldsymbol{J}_1^\dagger \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{N}_{\mathrm{b}1} \boldsymbol{J}_{\mathrm{c}}^\dagger \Lambda_{\mathrm{c}} \tilde{\sigma}_{\mathrm{c}} \tag{6.36}$$

The implementation is identical to Example 2 with modes defined by (6.31)-(6.36), so the active mode is chosen by Table 6.5.

The results from running Examples 1-3 on the UR5 manipulator are illustrated in Figure 6.4, and Figure 6.5 displays screenshots and images from simulation and actual experiment from Example 1.

In all examples, the position task is fulfilled as predicted by the theory presented in Chapter'4 and 5, i.e. the two waypoints are reached by the end effector. Furthermore, the end effector avoids the two obstacles by locking the distance to the obstacle center at the obstacle radius until the other active tasks drive the end effector away from the obstacle center on their own accord. This can be seen in Figure 6.4(a)-6.4(c), and is also confirmed by Figure 6.4(g)-6.4(i): The set-based collision avoidance tasks never exceed the valid sets C_{a} and C_{b}, but freeze on the boundary of these sets.

Figure 6.4(d)-6.4(f) display the active mode over time, and confirm that mode changes coincide with set-based tasks either being activated (frozen on boundary/leaving valid set) or deactivated (unfrozen/approaching valid set). An increase in mode means a new set-based task has been activated and vice versa.

In Example 1, FOV is implemented as an equality task with lower priority than the position task with the goal of aligning the FOV vector $\boldsymbol{a}$ with $\boldsymbol{a}_{\mathrm{des}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$. This corresponds to the z-axis of the end effector being parallel to the x-axis of the base coordinate system. As can be seen in Figure 6.4(j), the norm of the error between $\boldsymbol{a}$ and $\boldsymbol{a}_{\mathrm{des}}$ converges to zero at about $t = 30$ s, and Figure 6.5 shows that in the end configuration, these two vectors are indeed parallel.

In Example 2, FOV is a high-priority set-based task with a maximum value of 0.2622, corresponding to the angle between $\boldsymbol{a}$ and $\boldsymbol{a}_{\mathrm{des}}$ not exceeding 15°. The task has initial conditions outside its valid set C_{c} (Figure 6.4(k)). However, the other active tasks naturally bring the FOV closer to and eventually (at around $t = 4$ s) into C_{c}, and thus it is not necessary to freeze σ_{c}. Once σ_{c} enters C_{c}, the task will always stay in this set. At around $t = 13$ s, the system enters mode 4 and σ_{c} is frozen because the error between the actual and desired FOV vectors has reached its upper limit and keeping the task deactivated would result in the maximum value being violated. Shortly after, the end effector reaches the second obstacle, and so mode 6 is activated where both σ_{b} and σ_{c} are frozen. Once the end effector has moved around the obstacle, σ_{b} is released. σ_{c}, however, can not be released without leaving C_{c}, and so the system goes back to mode 4 and remains there for the duration of the example.

In Example 3, FOV is a low-priority set-based task with the same maximum value as Example 2. By comparing Figure 6.4(k) and 6.4(l) it is evident that these implementations behave similarly until $t = 13$ s, when the system enters mode 4 and activates σ_{c}. In Example 2, the task is frozen on the boundary, which is guaranteed due to the fact it is high priority. As explained in Chapter 5, low priority set-based tasks can not be guaranteed to actually freeze on the boundary, and they are therefore activated with the goal of pushing the task back to its boundary. This is confirmed by Figure 6.4(l). In this example, σ_{c} does indeed exceed its maximum value in spite of the system activating the task. However, eventually σ_{c} converges back to the boundary of C_{c}.

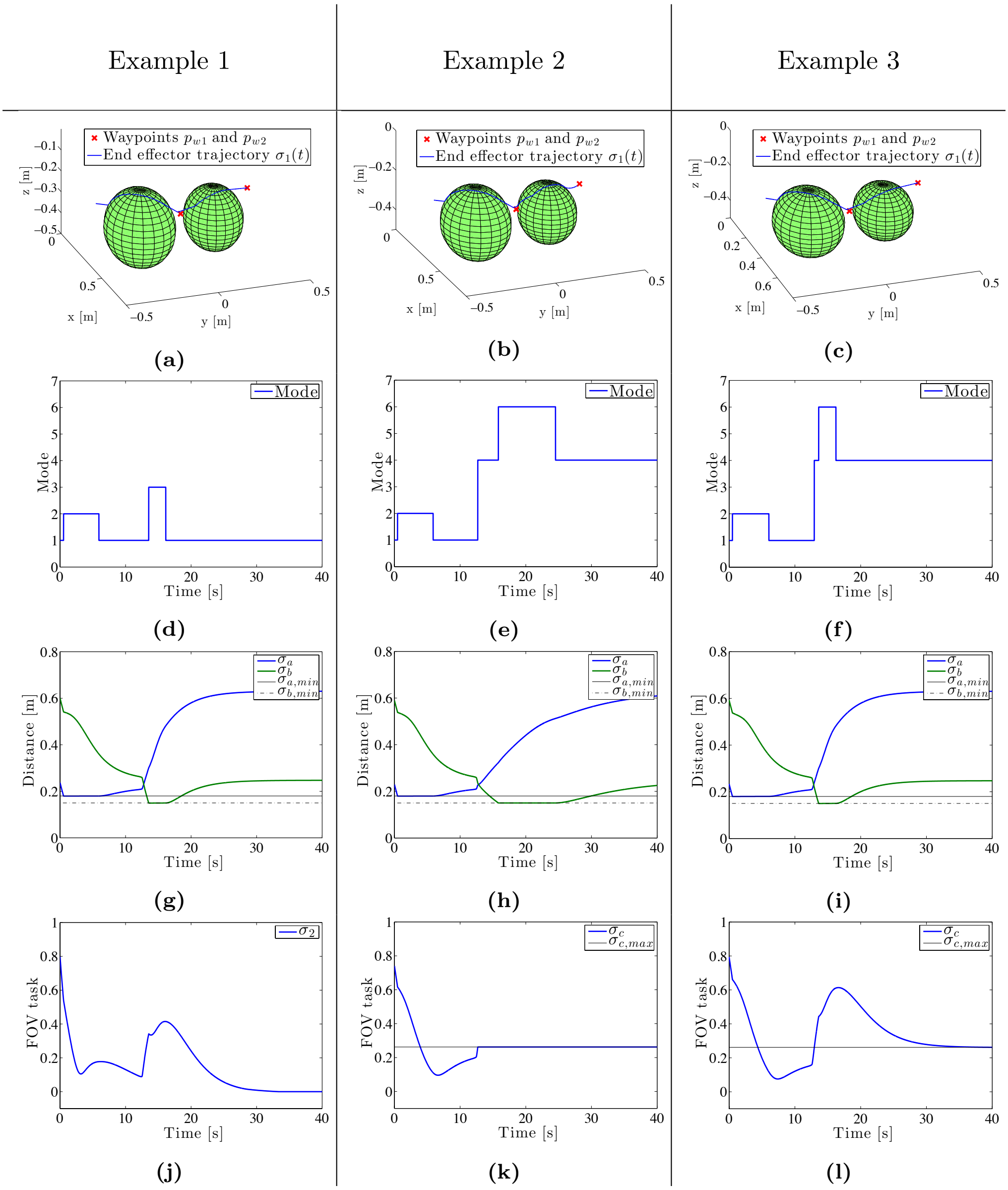
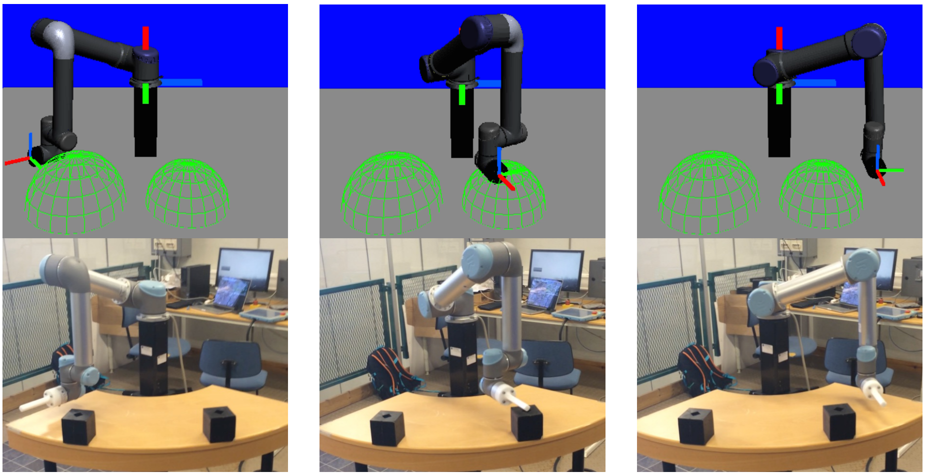


Figure 6.4: Logged data from experimental results, Example 1-3. In Figure (a)-(c) the end effector trajectory is displayed. Obstacles shown as spheres and waypoints as red dots. Figure (d)-(f) illustrate the active mode over time, and Figure (g)-(i) the distance between the end effector and the obstacle centers. Finally, the FOV task is shown in Figure (j)-(l).

Figure 6.4(j) and 6.4(l) show that implementing FOV as a lower priority equality and set-based render similar results. As expected, the equality task converges to the exact desired value and the set-based to the boundary of the valid set. Even so, in the case that the system has several other tasks with even lower priority, it might be beneficial to implement FOV as a set-based task as this imposes less constraint on the lower-priority tasks when inactive.



(a) $t = 0$ s **(b)** $t = 16$ s **(c)** $t = 40$ s

Figure 6.5: Pictures from simulation and actual experiments, Example 1. In the simulation, the base and end effector coordinate system is illustrated with green, blue and red axes for the x-, y- and z-axes respectively. These correspond to the coordinate frames of the actual robot.

6.3.4 Example 4

In this example, the system has been given a time-varying trajectory for the end effector to track:

$$\boldsymbol{\sigma}_{1,\text{des}}(t) = \boldsymbol{\sigma}_{\text{pos,des}}(t) = \begin{bmatrix} 0.5\sin^2(0.1t) + 0.2 \\ 0.5\cos(0.1t) + 0.25\sin(0.1t) \\ 0.5\sin(0.1t)\cos(0.1t) + 0.1 \end{bmatrix} \tag{6.37}$$

The task gain matrix has been chosen as

$$\boldsymbol{\Lambda}_{\text{pos}} = \text{diag}(\begin{bmatrix} 0.15 & 0.15 & 0.15 \end{bmatrix}). \tag{6.38}$$

In addition, the system is also given a valid workspace: A minimum and maximum value for the x, y and z position of the end effector. This is implemented as three set-based tasks that are equal to the first, second and third element of $\boldsymbol{\sigma}_{\text{pos}}$ respectively. These set-based tasks are all implemented as high-priority as shown in Table 6.8.

A system with 3 set-based tasks has $2^3 = 8$ modes to consider. These are listed below and activated according to Table 6.9, where a hyphen indicates that

Name	Task description	Type	Valid set C
σ_{a}	Limited workspace x-direction	Set-based	$C_{\text{a}} = [0.1, 0.6]$
σ_{b}	Limited workspace y-direction	Set-based	$C_{\text{b}} = [-0.5, 0.4]$
σ_{c}	Limited workspace z-direction	Set-based	$C_{\text{c}} = [-0.3, 0.25]$
$\boldsymbol{\sigma}_1$	Position	Equality	-

Table 6.8: Implemented tasks in Example 4 sorted by decreasing priority.

a set-based task is active in that mode.

$$\text{Mode 1: } \dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{f}_1 \triangleq \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.39}$$

$$\text{Mode 2: } \dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{f}_2 \triangleq \boldsymbol{N}_{\text{a}} \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.40}$$

$$\text{Mode 3: } \dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{f}_3 \triangleq \boldsymbol{N}_{\text{b}} \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.41}$$

$$\text{Mode 4: } \dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{f}_4 \triangleq \boldsymbol{N}_{\text{c}} \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.42}$$

$$\text{Mode 5: } \dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{f}_5 \triangleq \boldsymbol{N}_{\text{ab}}^{\text{A}} \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.43}$$

$$\text{Mode 6: } \dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{f}_6 \triangleq \boldsymbol{N}_{\text{ac}}^{\text{A}} \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.44}$$

$$\text{Mode 7: } \dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{f}_7 \triangleq \boldsymbol{N}_{\text{bc}}^{\text{A}} \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.45}$$

$$\text{Mode 8: } \dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{f}_8 \triangleq \boldsymbol{N}_{\text{abc}}^{\text{A}} \boldsymbol{J}_1^{\dagger} \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1 \tag{6.46}$$

Denote the boolean variables

$$\phi_i \triangleq \texttt{in\_T\_RC}\left(\boldsymbol{J}_\phi \boldsymbol{f}_i, \sigma_\phi, \sigma_{\phi,\min}, \sigma_{\phi,\max}\right) \qquad \text{for } i \in \{1, ..., 8\} \text{ and } \phi \in \{\text{a}, ..., \text{c}\}\,. \tag{6.47}$$

Priorities (i)	a_i	b_i	c_i	Active mode
1	`True`	`True`	`True`	$\boldsymbol{f}_1$
2	-	`True`	`True`	$\boldsymbol{f}_2$
3	`True`	-	`True`	$\boldsymbol{f}_3$
4	`True`	`True`	-	$\boldsymbol{f}_4$
5	-	-	`True`	$\boldsymbol{f}_5$
6	-	`True`	-	$\boldsymbol{f}_6$
7	`True`	-	-	$\boldsymbol{f}_7$
8	-	-	-	$\boldsymbol{f}_8$

Table 6.9: Table illustrating the activation of mode in Example 4.

The results of Example 4 are shown in Figure 6.6, 6.8 and 6.7. The system is unable to track the equality task perfectly as the desired trajectory moves in and out of the allowed workspace (Figure 6.6 and 6.7). Mathematically, this is explained by the fact that the set-based tasks and the position equality task are not linearly independent (in fact, they are equal to each other). Thus Assumption 4.1 in Chapter 5 is not satisfied, and the equality task errors can no longer be guaranteed to converge to zero. However, as seen in Figure 6.8(b), the end effector stays within

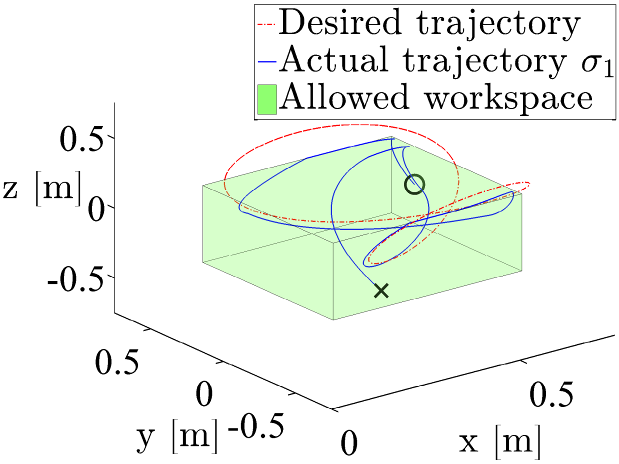


Figure 6.6: Desired and actual end trajectory plotted in red and blue respectively. The allowed workspace is illustrated by the green box. Initial and end position marked by x and o, respectively.

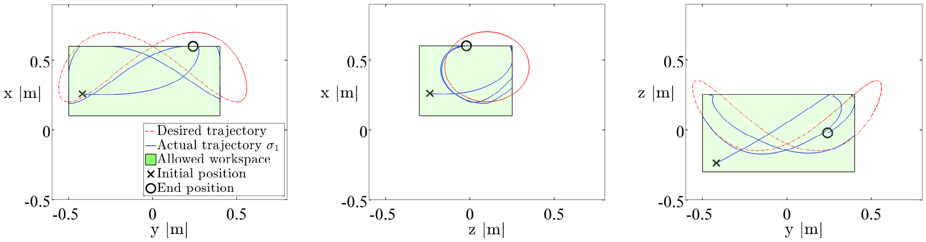


Figure 6.7: Actual vs. desired end effector trajectory from Example 4 plotted in the xy-, xz- and yz-plane, respectively. The end effector trajectory is contained in the allowed workspace at all times even though the reference trajectory moves in and out of this workspace.

the valid workspace at all times by freezing on the boundary when following the trajectory would bring it outside of the valid workspace. Figure 6.8(a) displays the active mode over time. Mode changes clearly correspond to set-based tasks being activated/deactivated on the border of their valid sets.

Videos of the experimental results can be viewed online[1].

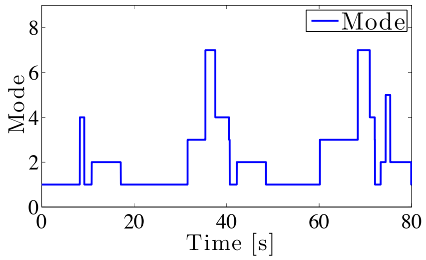
6.4 Conclusions

The method proposed in Chapter 4 and 5 for incorporating set-based tasks in the singularity-robust multiple task-priority inverse kinematics framework has been illustrated and validated in this chapter. In particular, the method has been implemented on a 6 DOF UR5 manipulator. Four examples have been constructed to

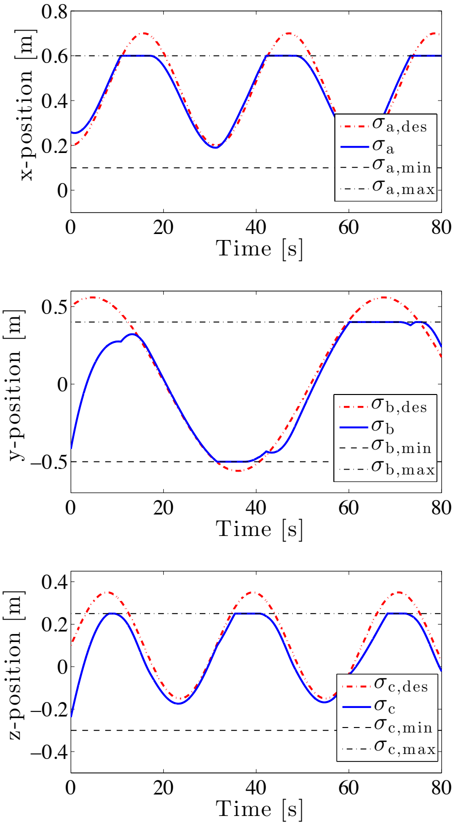
---

[1] https://www.dropbox.com/sh/x92agg4n7ly9hdc/AAC-xh8DyJM5X6jknJhyoJdca?dl=0

test various qualities and the performance of the algorithm. In summary, the experimental results confirm the following properties:



(a) Active mode over time.



(b) Reference and actual end effector position in x, y and z-direction with valid workspace limits.

Figure 6.8: Logged data from Example 4.

- All equality tasks converge to their desired value, given that they are not in direct conflict with a higher-priority set-based task
- All high-priority set-based tasks with initial conditions in their valid set C stay in this set $\forall t \geq 0$.
- All high-priority set-based tasks with initial conditions outside C can only 1) freeze at the current value, or 2) move closer to C. Hence, the initial condition is the maximum deviation from C.
- If a high-priority set-based task with initial conditions outside its valid set eventually enters C, it will stay in this set $\forall t \geq t_{\mathrm{e}}$, where t_{e} is the time the task entered the set.
- Lower-priority set-based tasks are not necessarily satisfied.

Furthermore, it is suggested that implementing a low-priority task as set-based rather than as an equality task is less restrictive on even lower priority tasks as they are not affected by the set-based task when it is not active.

Part II

Curved Path Following of Marine Vehicles in the Presence of Unknown Ocean Currents

Chapter 7

A Serret-Frenet approach for Surface Vessels

In this chapter, the vehicle model is given by (2.18). We assume that the vessel in question is equipped with sensors to measure position, heading (yaw), yaw rate and relative velocities. Note that this chapter assumes linear damping, i.e. the quadratic damping coefficient d_{11}^q in (2.18) is equal to zero.

Contributions of this chapter: This chapter extends the control method in [13] to obtain path following of curved paths for USVs also under the influence of unknown ocean currents. This is achieved by expanding the guidance law in [13] and combining it with an ocean current observer described in [1]. The closed-loop system consists of feedback linearizing controllers, and is dependent on relative velocity measurements. The path is represented by a Serret-Frenet reference frame.

Organization of this chapter: Section 7.1 defines the control objective. The suggested control system is then introduced in Section 7.2 followed by the main result in Section 7.3. Simulations results and conclusions are presented in Sections 7.4 and 7.5, respectively.

Publications: The results of this chapter are based on [63].

7.1 Control Objectives

This section formalizes the control problem solved in this chapter. The control system should make the vessel follow a given smooth path C and maintain a desired constant relative surge velocity $u_{r,\text{des}} > 0$ in the presence of unknown constant irrotational ocean current $\boldsymbol{V}_c = [V_x, V_y, 0]^T$. By Assumption 2.3 (Section 2.4.1) the current is bounded by some $V_{\max} > 0$.

The path C is parametrized with respect to the North-East-Down frame $\{i\}$ as a function of the arc length s. To accomplish this, a virtual reference Serret-Frenet frame $\{f\}$ is introduced, and the path is given by $C \triangleq \{(x_f(s), y_f(s), \psi_f(s))\}$. The

Serret-Frenet frame is anchored in and propagates along C with instantaneous speed $\dot{s}$.

Denoting $x_{b/f}$ and $y_{b/f}$ as the position of the body frame $\{b\}$ relative to the Serret-Frenet frame $\{f\}$ and $\psi_{b/f} \triangleq \psi - \psi_f$ as the relative orientation of the vessel relative to the $\{f\}$-frame, the control objectives can be formalized as follows:

$$\begin{aligned} \lim_{t\to\infty} x_{b/f}(t) &= 0, \\ \lim_{t\to\infty} y_{b/f}(t) &= 0, \\ \lim_{t\to\infty} u_r(t) &= u_{r,\text{des}}. \end{aligned} \tag{7.1}$$

The reference frames, $x_{b/f}$ and $y_{b/f}$ are illustrated in Figure 7.1. Clearly, ensuring that $x_{b/f}$ and $y_{b/f}$ converge to zero will result in the vessel converging to the path. Since the Serret-Frenet frame is virtual, we are free to choose the evolution $\dot{s}$ of this frame along the path.

The dynamics of the body frame relative to the Serret-Frenet can be expressed as follows [13]:

$$\begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \end{bmatrix} = \begin{bmatrix} \cos(\psi_{b/f}) & -\sin(\psi_{b/f}) \\ \sin(\psi_{b/f}) & \cos(\psi_{b/f}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} \dot{s} \\ 0 \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa \\ \kappa & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \end{bmatrix} \tag{7.2}$$

Here, κ is the curvature of the path C.

In this chapter the relative surge velocity u_r is controlled. As such, the total path following speed $U = \sqrt{u^2 + v^2}$ is unconstrained. For speed profile planning/tracking scenarios, this is not ideal. On the other hand, controlling the relative velocity of the vessel gives direct control of energy consumption.

Assumption 7.1: The propulsion system is rated with power and thrust capasity such that $u_{r,\text{des}}$ satisfies $0 \leq V_{\max} < u_{r,\text{des}}$.

Remark 7.1*:* Assumption 7.1 is necessary to prevent the vessel from drifting away while following the desired path.

7.2 The Control System

This section presents a control system that solves the control objectives in (7.1). The system consists of an ocean current observer, a guidance law, an update law for the Serret-Frenet frame, and surge and yaw controllers.

7.2.1 Ocean Current Observer

The current estimator for a surface ship is given in [1], and is a Luenberger type observer based on the model in (2.18). In addition to estimating the current components V_x and V_y, it gives estimates of the known states x and y that can be

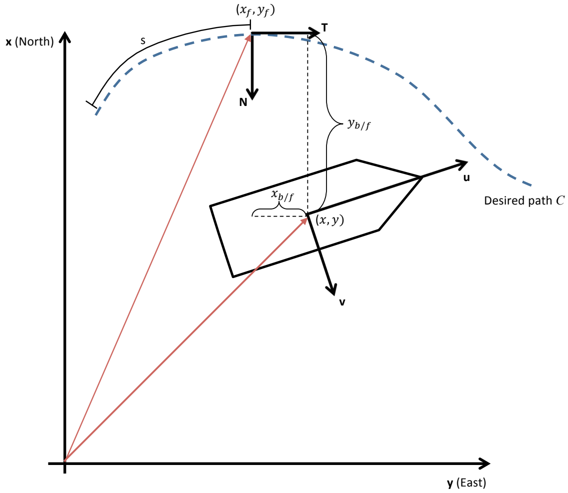


Figure 7.1: The inertial frame x-axis points north and the y-axis points east. The Serret-Frenet frame has axes denoted T and N and is anchored in the desired path. The position of this frame relative to the inertial frame is $(x_f(s), y_f(s)$. The body frame is fixed on the marine surface vessel. The position of the body frame relative to the inertial frame and Serret-Frenet frame is denoted (x, y) and $(x_{b/f}, y_{b/f})$ respectively.

compared to the actual, known states. The observer is given below:

$$\begin{aligned}
\dot{\hat{x}} &= \cos(\psi)u_r - \sin(\psi)v_r + \hat{V}_x + k_{x_1}\tilde{x} \\
\dot{\hat{y}} &= \sin(\psi)u_r + \cos(\psi)v_r + \hat{V}_y + k_{y_1}\tilde{y} \\
\dot{\hat{V}}_x &= k_{x_2}\tilde{x} \\
\dot{\hat{V}}_y &= k_{y_2}\tilde{y}
\end{aligned} \tag{7.3}$$

Here, $\hat{x}$, $\hat{y}$, $\hat{V}_x$ and $\hat{V}_y$ are the estimates of x, y, V_x and V_y, and $\tilde{x} \triangleq x - \hat{x}$, $\tilde{y} \triangleq y - \hat{y}$, $\tilde{V}_x \triangleq V_x - \hat{V}_x$ and $\tilde{V}_y \triangleq V_y - \hat{V}_y$ are the estimation errors. If the constant gain parameters k_{x_1}, k_{y_1}, k_{x_2} and k_{y_2} are greater than zero, the errors $\tilde{x}$, $\tilde{y}$, $\tilde{V}_x$ and $\tilde{V}_y$ will globally exponentially converge to zero (see Section 7.3).

Assumption 7.2: The current estimates are saturated in accordance with Assumption 2.3 in Section 2.4.1: $\sqrt{\hat{V}_x^2 + \hat{V}_y^2} \leq V_{\max} < u_{r,\text{des}}$.

Remark 7.2*:* The saturation is placed on the estimated current after the feedback loop so the saturation does not affect the stability of the observer.

In this chapter, the following notation is used for the ocean current and ocean current estimate in the $\{f\}$-frame:

$$\begin{aligned}V_x^f &= \cos(\psi_f)V_x + \sin(\psi_f)V_y,\\ V_y^f &= -\sin(\psi_f)V_x + \cos(\psi_f)V_y,\\ \hat{V}_x^f &= \cos(\psi_f)\hat{V}_x + \sin(\psi_f)\hat{V}_y,\\ \hat{V}_y^f &= -\sin(\psi_f)\hat{V}_x + \cos(\psi_f)\hat{V}_y.\end{aligned} \tag{7.4}$$

7.2.2 Guidance and Update Laws

Equation (7.5) and (7.6) contain the update law used to drive the $\{f\}$-frame forward along the desired path and the guidance law providing the yaw controller with its reference, respectively.

$$\dot{s} = \underbrace{\sqrt{u_{r,\mathrm{des}}^2 + v_r^2}}_{\triangleq U_{\mathrm{des}}} \frac{\sqrt{\Delta^2 + x_{b/f}^2} + x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} + \hat{V}_x^f, \tag{7.5}$$

$$\psi_{\mathrm{des}} = \psi_f - \arctan\left(\frac{v_r}{u_{r,\mathrm{des}}}\right) - \arctan\left(\frac{y_{b/f} + g}{\sqrt{\Delta^2 + x_{b/f}^2}}\right), \tag{7.6}$$

where g is the solution to the second-order equation

$$\underbrace{(\hat{V}_y^{f^2} - U_{\mathrm{des}}^2)}_{a} g^2 + 2\underbrace{\hat{V}_y^{f^2} y_{b/f}}_{b} g + \underbrace{\hat{V}_y^{f^2}\left(\Delta^2 + x_{b/f}^2 + y_{b/f}^2\right)}_{c} = 0, \tag{7.7}$$

and $\Delta > 0$ is a design parameter.

Remark 7.3: Assumption 7.1-7.2 ensure that the solution(s) of (7.7) are real and finite.

Equation (7.7) is a second order equation with parameters $(a, 2b, c)$ and thus it has two possible solutions:

$$\begin{aligned}g_1 &\triangleq \frac{-b - \sqrt{b^2 - ac}}{a},\\ g_2 &\triangleq \frac{-b + \sqrt{b^2 - ac}}{a}.\end{aligned} \tag{7.8}$$

To prove stability, the term g must have the same sign as $\hat{V}_y^f$ (this is shown in Appendix A). By simple analysis and Assumption 7.1-7.2, $a < 0$ and $c \geq 0$. Furthermore, since $\sqrt{b^2 - ac} \geq b$, g_1 will always be positive (≥ 0) and g_2 will always be negative (≤ 0). Thus, g is chosen as follows:

$$g = \begin{cases} g_1 & \hat{V}_y^f \geq 0 \\ g_2 & \hat{V}_y^f < 0 \end{cases} \tag{7.9}$$

7.2.3 Surge and Yaw Controllers

A feedback linearizing P-controller is used to ensure tracking of the desired relative surge velocity $u_{r,\text{des}}(t)$.

$$\tau_u = -F_{u_r}(v_r, r) + \frac{d_{11}}{m_{11}} u_{r,\text{des}} + \dot{u}_{r\text{des}} - k_{u_r}(u_r - u_{r,\text{des}}). \tag{7.10}$$

The gain $k_{u_r} > 0$ is constant, and the expression for F_{u_r} is given in (2.19). The terms m_{11} and d_{11} are mass and damping coefficients as defined in (2.16). The controller (7.10) guarantees exponential tracking of $u_{r,\text{des}}(t)$, see Section 7.3. Note that part of the damping is not canceled in order to guarantee some robustness with respect to model uncertainties.

Similarly, a feedback linearizing PD-controller is used to track the desired yaw angle ψ_{des}. In this case $\psi_{\text{des}}(t)$ is provided by the guidance law (7.6) and $\dot{\psi}_{\text{des}}(t)$ is calculated by taking the time derivative of $\psi_{\text{des}}(t)$.

$$\tau_r = -F_r(u_r, v_r, r) + \ddot{\psi}_{\text{des}} - k_\psi(\psi - \psi_{\text{des}}) - k_r(\dot{\psi} - \dot{\psi}_{\text{des}}). \tag{7.11}$$

k_ψ and k_r are strictly positive constant controller gains, and the expression for $F_r(u_r, v_r, r)$ is given in (2.22). This controller ensures exponential tracking of ψ_{des} and $\dot{\psi}_{\text{des}}$ (proven in Section 7.3).

7.2.4 State Measurements

The control system proposed in this chapter assumes that $\boldsymbol{\eta}$ (position and heading/yaw) and $\boldsymbol{\nu_r}$ (relative surge and sway velocity and yaw rate) are measured. Ships are usually equipped with a large variety of sensors that combined provide sensor data to estimate the vessel state [31]. For instance, Global Navigation Satellite System (GNSS) receivers can provide position and velocity measurements and a gyrocompass to measure yaw ψ yaw rate r.

To measure relative velocity, Acoustic Doppler Current Profilers (ADCP), Pitometer Logs and Paddle meters can be used [20]. ADCP uses acoustic measurements to capture the relative velocity. The Pitometer log compares the dynamic and static pressures of the fluid, and the Paddle meter measures spin velocity of a paddle driven by the flow itself. We assume that the measurements are filtered to remove first-order wave-induced motions and measurement noise.

7.3 Main Result

This section presents the conditions under which the proposed control system achieves the control objectives (7.1). First, consider the error vector

$$\boldsymbol{\xi} = \begin{bmatrix} \tilde{u}_r & \tilde{\psi} & \dot{\tilde{\psi}} & \tilde{V}_x & \tilde{V}_y & \tilde{x} & \tilde{y} \end{bmatrix}^T, \tag{7.12}$$

where $\tilde{u}_r \triangleq u_r - u_{r,\text{des}}$, $\tilde{\psi} \triangleq \psi - \psi_{\text{des}}$, $\dot{\tilde{\psi}} \triangleq r - \dot{\psi}_{\text{des}}$, $\tilde{V}_x \triangleq V_x - \hat{V}_x$, $\tilde{V}_y \triangleq V_y - \hat{V}_y$, $\tilde{x} \triangleq x - \hat{x}$ and $\tilde{y} \triangleq y - \hat{y}$ are the controller and observer errors.

Proposition 7.1. *Given an underactuated surface vessel described by the dynamical system (2.18). If Assumption 2.1-2.3 hold, the ocean current observer (7.3) and feedback linearizing controllers (7.10) and (7.11) ensure that* $\boldsymbol{\xi} = \mathbf{0}$ *is a UGES equilibrium point and that the references provided by the guidance system are exponentially tracked.*

Proof. By combining the ocean current observer (7.3) and feedback linearizing controllers (7.10) and (7.11) with the USV model (2.18), the error dynamics is reduced to

$$\dot{\boldsymbol{\xi}} = \underbrace{\begin{bmatrix} -(k_{u_r} + \frac{d_{11}}{m_{11}}) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -k_\psi & -k_r & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_{x_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -k_{y_2} \\ 0 & 0 & 0 & 1 & 0 & -k_{x_1} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -k_{y_1} \end{bmatrix}}_{\triangleq \boldsymbol{\Lambda}} \boldsymbol{\xi}. \tag{7.13}$$

The system (7.13) is linear and time-invariant. Furthermore, all controller and observer gains and $\frac{d_{11}}{m_{11}}$ are strictly positive, and it is straightforward to show that all eigenvalues of $\boldsymbol{\Lambda}$ are strictly in the left half plane. Hence, $\boldsymbol{\Lambda}$ is Hurwitz and the origin $\boldsymbol{\xi} = \mathbf{0}$ of (7.13) is UGES. ☐

Assumption 7.3: The controllers (7.11) and (7.10) are tuned such that the error dynamics (7.13) is fast compared to the cross-track error dynamics (7.2). Hence, $\boldsymbol{\xi}$ can be assumed to be zero.

Theorem 7.1. *Given an underactuated surface vessel described by the dynamical system (2.18). If Assumption 2.1-2.3 and 7.1-7.3 hold, the current observer (7.3) and the controllers given by (7.10)-(7.11) in combination with the update and guidance law (7.5) and (7.6) give a cross-track error dynamics for which the equilibrium point* $(x_{b/f}, y_{b/f}) = (0, 0)$ *is UGAS, and the control objectives (7.1) are achieved.*

Proof. The dynamics of the body frame relative to the Serret-Frenet is given in (7.2) and can be rewritten as

$$\begin{aligned} \begin{bmatrix} \dot{x}_{b/f} \\ y_{b/f} \end{bmatrix} &= \begin{bmatrix} \cos(\psi_{b/f}) & -\sin(\psi_{b/f}) \\ \sin(\psi_{b/f}) & \cos(\psi_{b/f}) \end{bmatrix} \begin{bmatrix} u_r \\ v_r \end{bmatrix} - \begin{bmatrix} \dot{s} \\ 0 \end{bmatrix} \\ &\quad - \dot{s} \begin{bmatrix} 0 & -\kappa \\ \kappa & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \end{bmatrix} + \begin{bmatrix} \cos(\psi_f) & \sin(\psi_f) \\ -\sin(\psi_f) & \cos(\psi_f) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix}. \end{aligned} \tag{7.14}$$

Using the expressions for the current observer (7.3), the update law (7.5), guidance law (7.6) and controllers (7.10)-(7.11), (7.14) can be expressed as

$$\begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \end{bmatrix} = \begin{bmatrix} -U_{\text{des}} \frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} \\ -U_{\text{des}} \frac{y_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa \\ \kappa & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \end{bmatrix} + \boldsymbol{H}(t, U_{\text{des}}, \boldsymbol{\xi})\boldsymbol{\xi}, \tag{7.15}$$

where

$$\boldsymbol{H}(t, U_{\mathrm{des}}, \boldsymbol{\xi}) = \begin{bmatrix} \cos(\psi_{b/f}) & h_1(t, U_{\mathrm{des}}, \boldsymbol{\xi}) & 0 & \cos(\psi_f) & \sin(\psi_f) & 0 & 0 \\ \sin(\psi_{b/f}) & h_2(t, U_{\mathrm{des}}, \boldsymbol{\xi}) & 0 & -\sin(\psi_f) & \cos(\psi_f) & 0 & 0 \end{bmatrix}^T \tag{7.16}$$

and

$$\begin{aligned} h_1(t, U_{\mathrm{des}}, \boldsymbol{\xi}) &= \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}} U_{\mathrm{des}} \frac{\sqrt{\Delta^2 + x_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} \\ &\quad + \frac{\sin(\tilde{\psi})}{\tilde{\psi}} U_{\mathrm{des}} \frac{y_{b/f} + g}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}}, \\ h_2(t, U_{\mathrm{des}}, \boldsymbol{\xi}) &= \frac{\sin(\tilde{\psi})}{\tilde{\psi}} U_{\mathrm{des}} \frac{\sqrt{\Delta^2 + x_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} \\ &\quad - \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}} U_{\mathrm{des}} \frac{y_{b/f} + g}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}}. \end{aligned} \tag{7.17}$$

The details of the above calculation are given in Appendix A.

By Assumption 7.3, $\boldsymbol{\xi} = \mathbf{0}$, reducing the error dynamics to

$$\begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \end{bmatrix} = \begin{bmatrix} -U_{\mathrm{des}} \frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} \\ -U_{\mathrm{des}} \frac{y_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa \\ \kappa & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \end{bmatrix}. \tag{7.18}$$

The stability of the system (7.18) can be proven using the quadratic positive definite Lyapunov function $V = \frac{1}{2}(x_{b/f}^2 + y_{b/f}^2)$.

$$\begin{aligned} \dot{V} &= \dot{x}_{b/f} x_{b/f} + \dot{y}_{b/f} y_{b/f} \\ &= -U_{\mathrm{des}} \frac{x_{b/f}^2 + y_{b/f}^2}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} + \dot{s}\kappa x_{b/f} y_{b/f} - \dot{s}\kappa x_{b/f} y_{b/f} \\ &\triangleq W(x_{b/f}, y_{b/f}, g) < 0 \end{aligned} \tag{7.19}$$

$\dot{V}$ is negative definite and thus the nominal system (7.18) is UGAS and $x_{b/f}$ and $y_{b/f}$ converge to zero. Furthermore, by Proposition 1, and $\tilde{u}_r$ exponentially to zero. Thus the control objectives are satisfied. □

7.4 Simulation Results

This section presents simulation results when the desired path is a straight line and a circular path, respectively. Numeric values for the ship model are given in [35]. In all simulations, the desired relative surge velocity $u_{r,\mathrm{des}} = 5$ m/s and the ocean current is $\boldsymbol{V}_c = [-1, 1.2]^T$ m/s. The look-ahead distance $\Delta = 50$ m and the controller and observer gains are $k_{u_r} = 0.1\ \mathrm{s}^{-1}$, $k_\psi = 0.04\ \mathrm{s}^{-2}$, $k_r = 0.1\ \mathrm{s}^{-1}$, $k_{x_1} = 4\ \mathrm{s}^{-1}$, $k_{x_2} = 0.05\ \mathrm{s}^{-2}$, $k_{y_1} = 4\ \mathrm{s}^{-1}$ and $k_{y_2} = 0.05\ \mathrm{s}^{-2}$.

Two desired paths have been defined and simulated:

$$C_1 := \begin{cases} x_f(s) & = s\cos(\psi_f(s)) \\ y_f(s) & = s\sin(\psi_f(s)) \\ \psi_f(s) & = 40° \end{cases} \tag{7.20}$$

$$C_2 := \begin{cases} x_f(s) & = R\cos(\frac{s}{R}) + C_1 \\ y_f(s) & = R\sin(\frac{s}{R}) + C_2 \\ \psi_f(s) & = \frac{s}{R} + \frac{\pi}{2} \end{cases} \tag{7.21}$$

In C_2, $R = 400$ m, $C_1 = 0$ m and $C_2 = 800$ m.

Simulation results are shown in Figure 7.2-7.7, and confirm that the control objectives are fulfilled. The ship converges to the desired path and $x_{b/f}$ and $y_{b/f}$ converge to zero. Furthermore, the controllers ensure that $u_r(t)$, $\psi(t)$ and $r(t)$ converge to and track their respective references, as shown in Figure 7.4 and 7.7. Finally, the current observer correctly estimates the ocean currents (Figure 7.3 and 7.6). Finally, notice in Figure 7.2 and 7.5 that when the USV is on the path, the heading is not aligned with the tangent of the path, but side-slips to counteract the effects of the ocean current and the curvature of the path.

All simulations were conducted with an appropriate saturation on the thruster force T and the rudder angle δ, confirming that the control system is applicable in a real-life scenario.

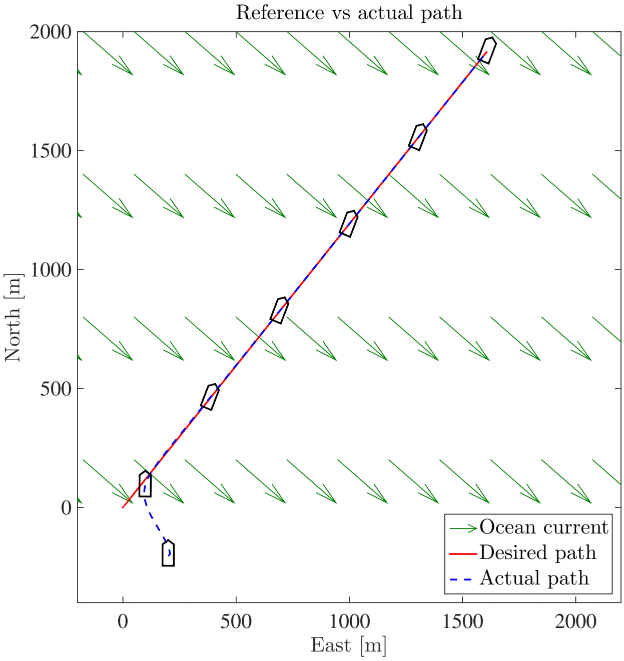


Figure 7.2: Desired and actual path C_1.

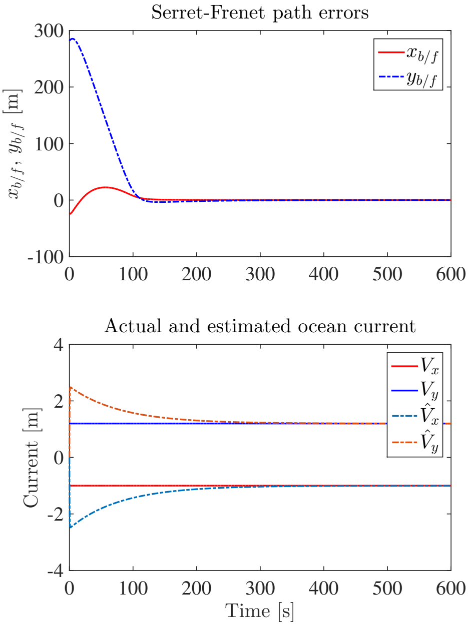


Figure 7.3: Path following errors and actual vs. estimated ocean current for C_1.

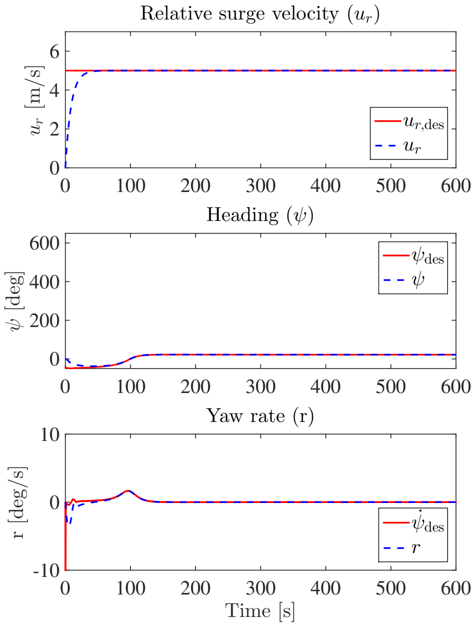


Figure 7.4: Desired vs. actual surge velocity, yaw and yaw rate for C_1.

7.5 Conclusions

In this chapter a guidance and control system for an underactuated surface vessel is developed to solve the control objective of making the vessel follow a general path in the presence of unknown ocean currents. The results are motivated by the path following methods of [13], and by expanding these guidance and update laws and combining them with an ocean current observer [1], convergence to the desired path is achieved with UGAS stability properties under explicit conditions. Simulation results verify the theoretical results.

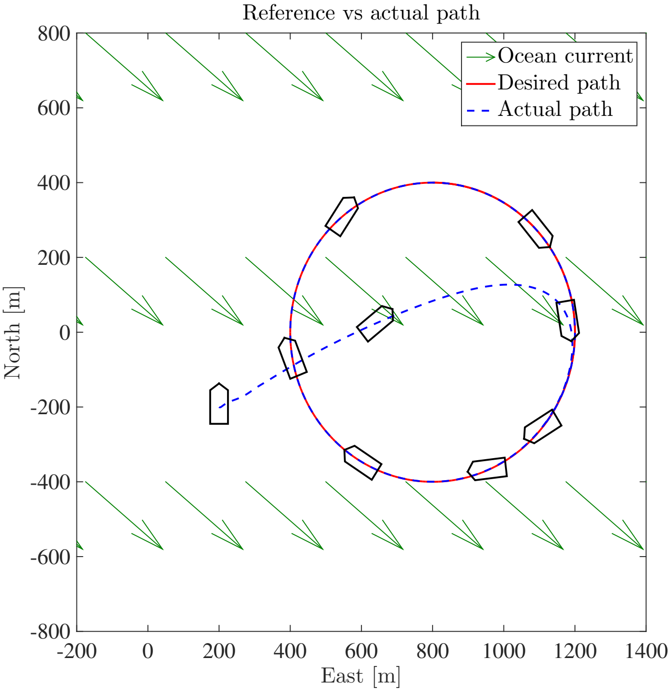
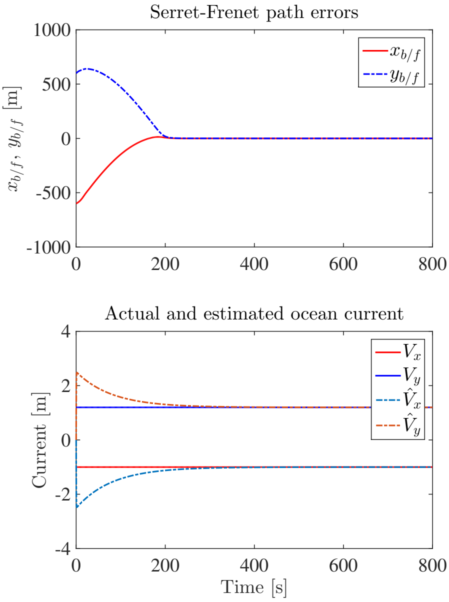


Figure 7.5: Desired and actual path C_2.



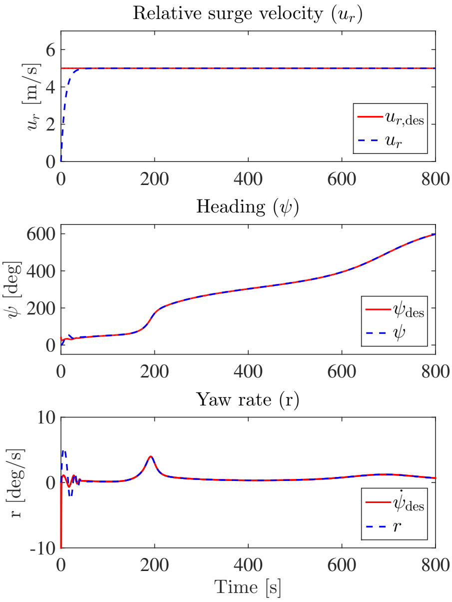


Figure 7.6: Path following errors and actual vs. estimated ocean current for C_2.

Figure 7.7: Desired vs. actual surge velocity, yaw and yaw rate for C_2.

Chapter 8

A Serret-Frenet approach for Underwater Vehicles

In this chapter, the vehicle model is given by (2.36). We assume that the vessel in question is equipped with sensors to measure position, attitude, relative velocities and pitch and yaw rate.

Contributions of this chapter: Chapter 7 presented a path following control method which allows underactuated surface vehicles to follow a predefined curved path under the influence of ocean currents. This chapter extends the result to AUVs, and is motivated by [13], in which curved path following is achieved for underwater vehicles when no ocean currents affect the vehicle.

Organization of this chapter: This chapter is organized as follows. Section 8.1 defines the control objectives. In Section 8.2 the control system that is proposed to solve the path following task is presented followed by a stability analysis of the closed-loop system in Section 8.3. Simulation results and conclusions are given in Section 8.4 and 8.5, respectively.

Publications: The results of this chapter are based on [64].

8.1 Control Objectives

This section specifies the control problem considered in this chapter. The control system should ensure that the vehicle converges to and follows a given continuously differentiable path C while maintaining a desired constant relative surge velocity $u_{r,\text{des}} > 0$ in the presence of unknown constant irrotational ocean currents. By Assumption 2.8 (Section 2.4.2), the current is bounded by some $V_{\max} > 0$.

The path C is parametrized with respect to the inertial frame $\{i\}$ as a function of the arc length s: $C \triangleq \{(x_f(s), y_f(s), z(s), \theta_f(s), \psi_f(s))\}$. To achieve the control objectives a virtual reference Serret-Frenet frame $\{f\}$ is introduced. This frame is anchored in and propagates along C with instantaneous speed $\dot{s}$.

Denoting $x_{b/f}$, $y_{b/f}$ and $z_{b/f}$ as the position of the body frame $\{b\}$ relative to the Serret-Frenet frame $\{f\}$, the control objectives can be formalized as follows:

$$\begin{aligned}\lim_{t\to\infty} x_{b/f}(t) &= 0,\\ \lim_{t\to\infty} y_{b/f}(t) &= 0,\\ \lim_{t\to\infty} z_{b/f}(t) &= 0,\\ \lim_{t\to\infty} u_r(t) &= u_{r,\text{des}}.\end{aligned} \tag{8.1}$$

The Serret-Frenet frame is illustrated in two dimensions in Figure 7.1. When $x_{b/f}$, $y_{b/f}$ and $z_{f/b}$ are zero, the vehicle is on the desired path. Since the Serret-Frenet frame is virtual, we are free to choose the evolution $\dot{s}$ of this frame along the path, and can use this to control the AUV velocity along the path. In this chapter the objective is to control the relative surge velocity u_r.

Assumption 8.1: The propulsion system is rated with power and thrust capasity such that choosing $u_{r,\text{des}}$ to satisfy $0 \leq V_{\max} \leq k u_{r,\text{des}}$, $0 < k < 1$, is feasible.

Remark 8.1*:* For most marine vehicles Assumption 8.1 is easily fulfilled since their propulsion systems are designed to give much more than 5 meters per second of relative speed $u_{r,\text{des}}$. The ocean current usually has an intensity of less than 1 meter per second [20]. In addition, we are free to choose the value of the design parameter k, which may be chosen close to one to make Assumption 8.1 less strict.

The dynamics of the body frame relative to the Serret-Frenet can be expressed as follows [13]:

$$\begin{bmatrix}\dot{x}_{b/f}\\ \dot{y}_{b/f}\\ \dot{y}_{b/f}\end{bmatrix} = \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb})\begin{bmatrix}u\\ v\\ w\end{bmatrix} - \begin{bmatrix}\dot{s}\\ 0\\ 0\end{bmatrix} - \dot{s}\begin{bmatrix}0 & -\kappa & 0\\ \kappa & 0 & -\tau\\ 0 & \tau & 0\end{bmatrix}\begin{bmatrix}x_{b/f}\\ y_{b/f}\\ z_{b/f}\end{bmatrix} \tag{8.2}$$

Here, κ and τ describe the path curvature and torsion, respectively, and the rotation matrix $\boldsymbol{R}_b^f(\boldsymbol{\theta}_{fb})$ is defined in 2.1.

Since the orientation of underwater vehicles includes rotations about several axes, it is not straightforward to define a desired absolute rotation for the vehicle. Therefore, a fourth reference frame $\{c\}$ is introduced. This is related to the $\{b\}$-frame through a rotation of an angle $-\beta_c$ about the body z-axis and an angle α_c about the rotated y-axis, where

$$\alpha_c \triangleq \arctan\left(\frac{w_r}{u_{r,\text{des}}}\right), \tag{8.3}$$

$$\beta_c \triangleq \arctan\left(\frac{v_r}{\sqrt{u_{r,\text{des}}^2 + w_r^2}}\right), \tag{8.4}$$

$$\boldsymbol{\Theta}_{cb} = \begin{bmatrix}0\\ \alpha_c\\ -\beta_c\end{bmatrix}, \tag{8.5}$$

describe the angle of attack and side-slip of the AUV, respectively [31]. The orientation of the $\{c\}$-frame relative to the $\{f\}$-frame is then described by

$$\boldsymbol{\Theta}_{fc} = \begin{bmatrix} \phi_{fc} \\ \theta_{fc} \\ \psi_{fc} \end{bmatrix}, \tag{8.6}$$

which can be calculated based on the rotation matrix

$$\boldsymbol{R}_c^f(\boldsymbol{\Theta}_{fc}) = \boldsymbol{R}_f^i(\boldsymbol{\Theta}_{if})^T \boldsymbol{R}_b^i(\boldsymbol{\Theta}_{ib}) \boldsymbol{R}_b^c(\boldsymbol{\Theta}_{cb})^T. \tag{8.7}$$

Here,

$$\boldsymbol{\Theta}_{ib} = \begin{bmatrix} 0 \\ \theta \\ \psi \end{bmatrix} \tag{8.8}$$

is the AUV pitch and yaw angle relative to the inertial frame defined in Section 2.4.2, and

$$\boldsymbol{\Theta}_{if} = \begin{bmatrix} 0 \\ \theta_f \\ \psi_f \end{bmatrix} \tag{8.9}$$

is the pitch and yaw angle of the Serret-Frenet reference frame relative to the inertial frame. The relative orientation $\boldsymbol{\Theta}_{fc}$ (not the absolute orientation $\boldsymbol{\Theta}_{ib}$) is then the controlled state as explained in Section 8.2.

8.2 The Control System

In this section a control system that aims to solve the control objectives (8.1) is proposed. This includes an ocean current observer, an update law for the reference frame $\{f\}$, guidance laws for the vehicle attitude and surge, pitch and yaw controllers.

8.2.1 Ocean Current Observer

A current estimator for a surface vessel is given in [1] and used in Chapter 7. The observer is a Luenberger type observer based on the kinematics of a surface vessel model. In this chapter this observer is extended from two to three by including the kinematics in the dynamical model in (2.36). In addition to estimating the current components V_x, V_y and V_z, it provides estimates of the known position states x, y and z that can be compared to the actual states:

$$\begin{aligned}
\dot{\hat{x}} &= \cos(\theta)\cos(\psi)u_r - \sin(\psi)v_r + \sin(\theta)\cos(\psi)w_r + \hat{V}_x + k_{x_1}\tilde{x} \\
\dot{\hat{y}} &= \cos(\theta)\sin(\psi)u_r + \cos(\psi)v_r + \sin(\theta)\sin(\psi)w_r + \hat{V}_y + k_{y_1}\tilde{y} \\
\dot{\hat{z}} &= -\sin(\theta)u_r + \cos(\theta)w_r + \hat{V}_z + k_{z_1}\tilde{z} \\
\dot{\hat{V}}_x &= k_{x_2}\tilde{x} \\
\dot{\hat{V}}_y &= k_{y_2}\tilde{y} \\
\dot{\hat{V}}_z &= k_{z_2}\tilde{z}
\end{aligned} \tag{8.10}$$

Here, $\hat{x}$, $\hat{y}$, $\hat{z}$, $\hat{V}_x$, $\hat{V}_y$ and $\hat{V}_z$ are the estimates of x, y, z, V_x, V_y and V_z, respectively. Furthermore, $\tilde{x} = x - \hat{x}$, $\tilde{y} = y - \hat{y}$, $\tilde{z} = z - \hat{z}$, $\tilde{V}_x = V_x - \hat{V}_x$, $\tilde{V}_y = V_y - \hat{V}_y$ and $\tilde{V}_z = V_z - \hat{V}_z$. are the observer error signals. If the constant gain parameters k_{x_1}, k_{y_1}, k_{z_1}, k_{x_2}, k_{y_2} and k_{z_2} are chosen greater than zero, the errors $\tilde{x}$, $\tilde{y}$, $\tilde{z}$, $\tilde{V}_x$, $\tilde{V}_y$ and $\tilde{V}_z$ will globally exponentially converge to zero (see Section 8.3).

Assumption 8.2: The current estimates are saturated in accordance with Assumption 2.8, Section 2.4.2: $\sqrt{\hat{V}_x^2 + \hat{V}_y^2 + \hat{V}_z^2} \leq V_{\max} < u_{r,\text{des}}$.

Remark 8.2*:* The saturation is placed on the estimated current after the feedback loop so the saturation does not affect the stability of the observer.

In this chapter, the following notation is used for the ocean current and ocean current estimates expressed in the $\{f\}$-frame.

$$\boldsymbol{V}_c^f = \begin{bmatrix} V_x^f \\ V_y^f \\ V_z^f \end{bmatrix} = \boldsymbol{R}_f^i(\boldsymbol{\Theta}_{if})^T \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}, \tag{8.11}$$

$$\hat{\boldsymbol{V}}_c^f = \begin{bmatrix} \hat{V}_x \\ \hat{V}_y \\ \hat{V}_z \end{bmatrix} = \boldsymbol{R}_f^i(\boldsymbol{\Theta}_{if})^T \begin{bmatrix} \hat{V}_x^f \\ \hat{V}_y^f \\ \hat{V}_z^f \end{bmatrix}. \tag{8.12}$$

8.2.2 Guidance and Update Laws

The update law used to drive the Serret-Frenet frame forward along the desired path can be freely chosen as the $\{f\}$-frame is a virtual reference frame. This chapter proposes the following update law:

$$\begin{aligned} \dot{s} = {} & \underbrace{\sqrt{u_{r,\text{des}}^2 + v_r^2 + w_r^2}}_{\triangleq U_{\text{des}}} \frac{\sqrt{\Delta^2 + y_{b/f}^2}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f} + f)^2}} \frac{\sqrt{\Delta^2 + x_{b/f}^2 + z_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2 + z_{b/f}^2}} \\ & + U_{\text{des}} \frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2}} + \hat{V}_x^f \end{aligned} \tag{8.13}$$

The guidance laws providing the pitch and yaw controller with their references are chosen as follows:

$$\theta_{fc,\text{des}} = \arctan\left(\frac{z_{b/f} + f}{\sqrt{\Delta^2 + y_{b/f}^2}}\right) \tag{8.14}$$

$$\psi_{fc,\text{des}} = -\arctan\left(\frac{y_{b/f} + g}{\sqrt{\Delta^2 + x_{b/f}^2 + z_{b/f}^2}}\right), \tag{8.15}$$

where f is the solution to the second order equation

$$\underbrace{\left(\hat{V}_z^{f^2} - U_{\text{des}}^2\right)}_{a_f} f^2 + 2\underbrace{\hat{V}_z^{f^2} z_{b/f}}_{b_f} f + \underbrace{\hat{V}_z^{f^2}\left(\Delta^2 + y_{b/f}^2 + z_{b/f}^2\right)}_{c_f} = 0, \tag{8.16}$$

and g is the solution to the second order equation

$$\underbrace{\left(\hat{V}_y^{f^2} - \cos^2(\theta_{fc,\text{des}})U_{\text{des}}^2\right)}_{a_g} g^2 + 2\underbrace{\hat{V}_y^{f^2} y_{b/f}}_{b_g} g + \underbrace{\hat{V}_y^{f^2}\left(\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2\right)}_{c_g} = 0. \tag{8.17}$$

Here, $\Delta > 0$ is a design parameter. Note that the guidance laws are defined for the relative orientations θ_{fc} and ψ_{fc} as defined in Section 8.1.

In this chapter the analysis is restricted to a domain $\mathbb{D}$ where $\theta_{fc,\text{des}}$ is bounded by the same design parameter k as the desired relative surge velocity in Assumption 8.1:

$$\mathbb{D} = \left\{(\theta, \theta_{fc}, \theta_{fc,\text{des}}) \in \mathbb{R}^3 : |\theta| < \pi/2, |\theta_{fc}| < \pi/2, |\theta_{fc,\text{des}}| < \arccos(k)\right\} \tag{8.18}$$

Remark 8.3*:* $|\theta_{fc,\text{des}}| < \arccos(k)$ and Assumption 8.1-8.2 ensure that the solutions of f and g are real and finite.

Remark 8.4*:* If k is chosen close to 1, it greatly limits the valid values of $\theta_{fc,\text{des}}$, but the control system will work for ocean currents that has a maximum magnitude close to the desired relative surge velocity $u_{r\text{des}}$. Similarly, if k is chosen close to 0, $\theta_{fc,\text{des}}$ can take on values close to $\pm\pi/2$, but the current has to be very limited compared to the desired relative surge velocity. If the current is relatively strong, this can be overcome by choosing a large $u_{r\text{des}}$.

Remark 8.5*:* The restrictions on θ and θ_{fc} are to avoid the singularities that occur due to Euler angle representation.

Remark 8.6*:* Deriving explicit conditions that ensure that the domain $\mathbb{D}$ is invariant is not straight forward and will remain a topic of future work.

Equation (8.16) and (8.17) are second order equations with parameters $(a_f, 2b_f, c_f)$ and $(a_g, 2b_g, c_g)$. Thus, these equations have two possible solutions:

$$\begin{aligned} i_1 &\triangleq \frac{-b_i - \sqrt{b_i^2 - a_i c_i}}{a_i}, \\ i_2 &\triangleq \frac{-b_i + \sqrt{b_i^2 - a_i c}}{a_i}, \end{aligned} \tag{8.19}$$

for $i = \{f, g\}$. To compensate for the ocean current, f must have the same sign as $\hat{V}_z^f$ and g must have the same sign as $\hat{V}_y^f$ (this will be shown in Section 8.3). In $\mathbb{D}$ and by Assumption 8.2, $a_i < 0$ and $c_i \geq 0$. Furthermore, since $\sqrt{b_i^2 - a_i c_i} \geq b_i$, i_1 will always be positive (≥ 0) and i_2 will always be negative (≤ 0). Hence, f and g are thus chosen as follows:

$$f = \begin{cases} f_1 & \hat{V}_z^f \geq 0 \\ f_2 & \hat{V}_z^f < 0 \end{cases} \tag{8.20}$$

$$g = \begin{cases} g_1 & \hat{V}_y^f \geq 0 \\ g_2 & \hat{V}_y^f < 0 \end{cases} \tag{8.21}$$

8.2.3 Controllers

A feedback linearizing P-controller is used to ensure tracking of the desired relative surge velocity $u_{r,\mathrm{des}}(t)$.

$$\tau_u = -F_{u_r}(u_r, v_r, w_r, r, q) + \dot{u}_{r,\mathrm{des}} - k_{u_r}(u_r - u_{r,\mathrm{des}}) \tag{8.22}$$

The gain $k_{u_r} > 0$ is constant and $F_{u_r}(u_r, v_r, r, q)$ is defined in (2.37).

In addition to $u_r(t)$, the controlled states are the relative orientation $\theta_{fc}(t)$, $\psi_{fc}(t)$ and the rotational velocities q and r. The controller is an integrator backstepping controller that ensures that the difference between the actual and desired states converge to zero. The errors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are defined as

$$\boldsymbol{z}_1 \triangleq \begin{bmatrix} \tilde{\theta}_{fc} \\ \tilde{\psi}_{fc} \end{bmatrix} \triangleq \begin{bmatrix} \theta_{fc} \\ \psi_{fc} \end{bmatrix} - \begin{bmatrix} \theta_{fc,\mathrm{des}} \\ \psi_{fc,\mathrm{des}} \end{bmatrix}, \tag{8.23}$$

$$\boldsymbol{z}_2 \triangleq \begin{bmatrix} \tilde{q} \\ \tilde{r} \end{bmatrix} \triangleq \begin{bmatrix} q \\ r \end{bmatrix} - \begin{bmatrix} q_{\mathrm{des}} \\ r_{\mathrm{des}} \end{bmatrix}. \tag{8.24}$$

The controller is then given as

$$\begin{bmatrix} \tau_q \\ \tau_r \end{bmatrix} = \begin{bmatrix} \dot{q}_{\mathrm{des}} \\ \dot{r}_{\mathrm{des}} \end{bmatrix} - \begin{bmatrix} F_q(\theta, u_r, w_r, q) \\ F_r(u_r, v_r, r) \end{bmatrix} - \boldsymbol{A}^T \boldsymbol{z}_1 - \boldsymbol{K}_2 \boldsymbol{z}_2, \tag{8.25}$$

where

$$\boldsymbol{A} \triangleq \begin{bmatrix} \cos(\phi_{fc}) & -\sin(\phi_{fc}) \\ \frac{\sin(\phi_{fc})}{\cos(\theta_{fc})} & \frac{\cos(\phi_{fc})}{\cos(\theta_{fc})} \end{bmatrix} \begin{bmatrix} \cos(\beta_c) & -\sin(\beta_c)\sin(\alpha_c) \\ 0 & \cos(\alpha_c) \end{bmatrix}, \tag{8.26}$$

$$\begin{bmatrix} q_{\mathrm{des}} \\ r_{\mathrm{des}} \end{bmatrix} \triangleq -\boldsymbol{A}^{-1}(\boldsymbol{\Phi} + \boldsymbol{K}_1 \boldsymbol{z}_1), \tag{8.27}$$

$$\boldsymbol{\Phi} = -\begin{bmatrix} \dot{\theta}_{fc,\mathrm{des}} \\ \dot{\psi}_{fc,\mathrm{des}} \end{bmatrix} + \begin{bmatrix} \cos(\phi_{fc}) & -\sin(\phi_{fc}) \\ \frac{\sin(\phi_{fc})}{\cos(\theta_{fc})} & \frac{\cos(\phi_{fc})}{\cos(\theta_{fc})} \end{bmatrix} \boldsymbol{L}(\boldsymbol{\omega}_{bc}^c - [\boldsymbol{R}_c^f(\boldsymbol{\Theta}_{fc})]^T \boldsymbol{\omega}_{if}^f), \tag{8.28}$$

$$\boldsymbol{L} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{8.29}$$

$\boldsymbol{\omega}_{bc}^c = [-\dot{\beta}_c \sin(\alpha_c), -\dot{\alpha}_c, \dot{\beta}_c \cos(\alpha_c)]^T$, $\boldsymbol{\omega}_{if}^f = \dot{s}[\tau(s), 0, \kappa(s)]^T$ and $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are positive definite, symmetric gain matrices. Furthermore, $\tau(s)$ and $\kappa(s)$ are the path torsion and curvature, and $\theta_{fc,\mathrm{des}}$ and $\psi_{fc,\mathrm{des}}$ are given by (8.14) and (8.15). The functions $F_q(\theta, u_r, w_r, q)$ and $F_r(u_r, v_r, r)$ are defined in (2.43)-(2.44).

Remark 8.7*:* $\boldsymbol{A}$ is not defined for $\theta_{fc} = \pm\pi/2$.This is due to the well-known singularity of Euler angle representation. The singularity is not contained the considered domain $\mathbb{D}$ (8.18). The guidance law (8.14) is designed so the desired value $|\theta_{fc,\mathrm{des}}| < \pi/2$. Furthermore, unless the singularity $\theta_{fc} = \pm\pi/2$ is reached, θ_{fc} will exponentially converge to $\theta_{fc,\mathrm{des}}$ (see Proposition 8.1). Thus, unless $\theta_{fc}(0) = \pi/2$ as an initial condition, the singularity will never be reached in the physical system.

8.2.4 State Measurements

The control system proposed in this chapter requires that $\boldsymbol{\eta}$ and $\boldsymbol{\nu_r}$ are measured. Underwater vehicles are usually equipped with a large variety of sensors that can be used to estimate the vessel state [31]. For instance, absolute position and velocity can be estimated using an Inertial Measurement Unit (IMU) and a LongBase Line transponder system. Acoustic Doppler Current Profilers (ADCP), Pitometer Logs and Paddle meters can be used to measure relative velocity. ADCP estimates relative velocity using acoustic measurements, the Pitometer log compares the dynamic and static pressures of the fluid, and the Paddle meter measures spin velocity of a paddle driven by the flow itself [22].

Since the controllers (8.22) and (8.25) depend on the time derivative of the reference signals, they are susceptible to measurement noise. Thus, it is important that the AUV is equipped with reliable sensors with minimal measurement noise. The effects of measurement noise can also be minimized by utilizing a low-pass filter on the sensor data.

8.3 Main Result

This section presents the conditions under which the proposed control system achieves the control objectives (8.1). First, consider the error vector

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \end{bmatrix}, \tag{8.30}$$

$$\boldsymbol{\xi}_1 = \begin{bmatrix} \tilde{u}_r \\ \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix}, \tag{8.31}$$

$$\boldsymbol{\xi}_2 = \begin{bmatrix} \tilde{V}_x & \tilde{V}_y & \tilde{V}_z & \tilde{x} & \tilde{y} & \tilde{z} \end{bmatrix}^T, \tag{8.32}$$

where $\tilde{u}_r \triangleq u_r - u_{r,\text{des}}$, $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are defined in (8.23)-(8.24) and the observer errors $\boldsymbol{\xi}_2$ are defined in Section 8.2.1.

Proposition 8.1. *Given an underactuated surface vessel described by the dynamical system (2.36). If Assumption 2.5-2.8 (Section 2.4.2) hold, the ocean current observer (8.10) and feedback linearizing controllers (8.22) and (8.25) ensure that* $\boldsymbol{\xi} = \mathbf{0}$ *is an exponentially stable equilibrium point and that the references provided by the guidance system are exponentially tracked.*

Proof. By combining the controllers (8.22)-(8.25) with the USV model (2.36), the error dynamics is reduced to

$$\dot{\boldsymbol{\xi}}_1 = \begin{bmatrix} -k_{u_r} & \mathbf{0} & \mathbf{0} \\ 0 & -\boldsymbol{K}_1 & \boldsymbol{A} \\ 0 & -\boldsymbol{A}^T & -\boldsymbol{K}_2 \end{bmatrix} \boldsymbol{\xi}_1. \tag{8.33}$$

The details of the above calculation are given in Appendix B. Consider the positive definite and radially unbounded Lyapunov function candidate

$$V = \frac{1}{2}\boldsymbol{\xi}_1^T \boldsymbol{\xi}_1 \tag{8.34}$$

along the solutions of the system (8.33), which results in

$$\begin{aligned}\dot{V} &= -k_{u_r}\tilde{u}_r^2 + \boldsymbol{z}_1^T(-\boldsymbol{K}_1\boldsymbol{z}_1 + \boldsymbol{A}\boldsymbol{z}_2) + \boldsymbol{z}_2^T(-\boldsymbol{A}^T\boldsymbol{z}_1 - \boldsymbol{K}_2\boldsymbol{z}_2) \\ &= -k_{u_r}\tilde{u}_r^2 - \boldsymbol{z}_1^T\boldsymbol{K}_1\boldsymbol{z}_1 - \boldsymbol{z}_2^T\boldsymbol{K}_2\boldsymbol{z}_2 \\ &\leq -\lambda||\boldsymbol{\xi}_1||^2, \end{aligned} \tag{8.35}$$

where $\lambda = \min\left(k_{u_r}, \mathrm{eig}(\boldsymbol{K}_1), \mathrm{eig}(\boldsymbol{K}_2)\right) > 0$. Hence, the equilibrium point $\boldsymbol{\xi}_1 = \boldsymbol{0}$ is exponentially stable.

Remark 8.8*:* Note that the equilibrium point is not globally stable, due to the singularity in $\boldsymbol{A}$.

Furthermore, by inserting the USV kinematics (2.36) into the ocean current observer dynamics (8.10), the error dynamics $\boldsymbol{\xi}_2$ can be expressed as

$$\dot{\boldsymbol{\xi}}_2 = \begin{bmatrix} 0 & 0 & 0 & -k_{x2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -k_{y2} & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_{z2} \\ 1 & 0 & 0 & -k_{x1} & 0 & 0 \\ 0 & 1 & 0 & 0 & -k_{y1} & 0 \\ 0 & 0 & 1 & 0 & 0 & -k_{z1} \end{bmatrix} \boldsymbol{\xi}_2. \tag{8.36}$$

The system (8.36) is linear and time-invariant. Furthermore, all observer gains are strictly positive, and it is straightforward to show that all eigenvalues of $\boldsymbol{\Lambda}$ are strictly in the left half plane. Hence, $\boldsymbol{\Lambda}$ is Hurwitz and the origin $\boldsymbol{\xi}_2 = \boldsymbol{0}$ of (8.36) is UGES. □

Assumption 8.3: The current observer (8.10) and controllers (8.22) and (8.25) are tuned such that the error dynamics (8.33) and (8.36) are fast compared to the cross-track error dynamics (8.2). Hence, $\boldsymbol{\xi}$ can be assumed to be zero.

Theorem 8.1. *Given an underactuated underwater vehicle described by the dynamical system (2.36) and* $(\theta(t), \theta_{fc}(t), \theta_{fc,des}(t)) \in \mathbb{D}, \forall t \geq t_0$. *If Assumption 2.5-2.8 and 8.1-8.3 hold, then the ocean current observer (8.10), update law (8.13), guidance laws (8.14)-(8.15) and controllers (8.22) and (8.25) result in cross-track error dynamics for which the equilibrium point* $(x_{b/f}, y_{b/f}, z_{b/f}) = (0, 0, 0)$ *is asymptotically stable, and the control objectives (8.1) are achieved.*

Proof. The error dynamics are given in (8.2) and can be rewritten as

$$\begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \\ \dot{z}_{b/f} \end{bmatrix} = \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} - \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \\ z_{b/f} \end{bmatrix} + [\boldsymbol{R}_f^i(\boldsymbol{\Theta}_{if})]^T \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}. \tag{8.37}$$

Inserting the expressions for the current observer (8.10), the update law (8.13), guidance laws (8.14)-(8.15) and controllers (8.22) and (8.25), this can be further rewritten as

$$\begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \\ \dot{z}_{b/f} \end{bmatrix} = \begin{bmatrix} -U_{\text{des}} \frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2}} \\ -U_{\text{des}} \frac{\sqrt{\Delta^2 + y_{b/f}^2}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f} + f)^2}} \frac{y_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2 + z_{b/f}^2}} \\ -U_{\text{des}} \frac{z_{b/f}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f} + f)^2}} \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \\ z_{b/f} \end{bmatrix} + \boldsymbol{H}(t, U_{\text{des}}, \boldsymbol{\xi})\boldsymbol{\xi}, \tag{8.38}$$

where

$$\boldsymbol{H}(t, U_{\text{des}}, \boldsymbol{\xi})^T = \tag{8.39}$$
$$\begin{bmatrix} c(\theta_{fb})c(\psi_{fb}) & c(\theta_{fb})s(\psi_{fb}) & -s(\theta_{fb}) \\ h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ c(\psi_f)c(\theta_f) & c(\psi_f)s(\phi_f)s(\theta_f) - c(\phi_f)s(\psi_f) & c(\phi_f)c(\psi_f)s(\theta_f) + s(\phi_f)s(\psi_f) \\ c(\theta_f)s(\psi_f) & s(\phi_f)s(\psi_f)s(\theta_f) + c(\phi_f)c(\psi_f) & c(\phi_f)s(\psi_f)s(\theta_f) - c(\psi_f)s(\phi_f) \\ -s(\theta_f) & c(\theta_f)s(\phi_f) & c(\phi_f)c(\theta_f) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and

$$h_{11} = U_{\text{des}} \left(\frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}} \sin(\theta_{fc,\text{des}}) \left(-\cos(\psi_{fc,\text{des}}) \cos(\tilde{\psi}_{fc}) + \sin(\psi_{fc,\text{des}}) \sin(\tilde{\psi}_{fc}) \right) + \frac{\cos(\tilde{\theta}_{fc}) - 1}{\tilde{\theta}_{fc}} \cos(\theta_{fc,\text{des}}) \cos(\psi_{fc,\text{des}}) \right),$$

$$h_{12} = U_{\text{des}} \cos(\theta_{fc,\text{des}}) \cos(\tilde{\theta}_{fc}) \left(-\frac{\sin(\tilde{\psi}_{fc})}{\tilde{\psi}_{fc}} \sin(\psi_{fc,\text{des}}) + \frac{\cos(\tilde{\psi}_{fc}) - 1}{\tilde{\psi}_{fc}} \cos(\psi_{fc,\text{des}}) \right),$$

$$h_{21} = U_{\text{des}} \left(\frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}} \sin(\theta_{fc,\text{des}}) \left(-\sin(\psi_{fc,\text{des}}) \cos(\tilde{\psi}_{fc}) - \cos(\psi_{fc,\text{des}}) \sin(\tilde{\psi}_{fc}) \right) + \frac{\cos(\tilde{\theta}_{fc}) - 1}{\tilde{\theta}_{fc}} \cos(\theta_{fc,\text{des}}) \sin(\psi_{fc,\text{des}}) \right),$$

$$h_{22} = U_{\text{des}} \cos(\theta_{fc,\text{des}}) \cos(\tilde{\theta}_{fc}) \left(\frac{\sin(\tilde{\psi}_{fc})}{\tilde{\psi}_{fc}} \cos(\psi_{fc,\text{des}}) + \frac{\cos(\tilde{\psi}_{fc}) - 1}{\tilde{\psi}_{fc}} \sin(\psi_{fc,\text{des}}) \right),$$

$$h_{31} = U_{\text{des}} \left(-\frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}} \cos(\theta_{fc,\text{des}}) - \frac{\cos(\tilde{\theta}_{fc}) - 1}{\tilde{\theta}_{fc}} \sin(\theta_{fc,\text{des}}) \right).$$

Rewriting (8.37) into (8.38) is quite extensive, and the details are presented in Appendix C. The matrix $\boldsymbol{H}(t, U_{\text{des}}, \boldsymbol{\xi})$ is defined and finite for all $\boldsymbol{\xi}$. Applying Assumption 8.3, we can assume that $\boldsymbol{\xi} = \mathbf{0}$, which reduces the dynamics to

$$\begin{aligned}\begin{bmatrix}\dot{x}_{b/f}\\ \dot{y}_{b/f}\\ \dot{z}_{b/f}\end{bmatrix} = &\begin{bmatrix} -U_{\text{des}}\frac{x_{b/f}}{\sqrt{\Delta^2+x_{b/f}^2+y_{b/f}^2+z_{b/f}^2}} \\ -U_{\text{des}}\frac{\sqrt{\Delta^2+y_{b/f}^2}}{\sqrt{\Delta^2+y_{b/f}^2+(z_{b/f}+f)^2}}\frac{y_{b/f}}{\sqrt{\Delta^2+x_{b/f}^2+(y_{b/f}+g)^2+z_{b/f}^2}} \\ -U_{\text{des}}\frac{z_{b/f}}{\sqrt{\Delta^2+y_{b/f}^2+(z_{b/f}+f)^2}} \end{bmatrix} \\ &- \dot{s}\begin{bmatrix}0 & -\kappa & 0\\ \kappa & 0 & -\tau\\ 0 & \tau & 0\end{bmatrix}\begin{bmatrix}x_{b/f}\\ y_{b/f}\\ z_{b/f}\end{bmatrix}.\end{aligned} \tag{8.40}$$

The stability of the system (8.40) can be proven using the quadratic positive definite Lyapunov function $V = \frac{1}{2}(x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2)$.

$$\begin{aligned}\dot{V} &= \dot{x}_{b/f}x_{b/f} + \dot{y}_{b/f}y_{b/f} + \dot{z}_{b/f}z_{b/f} \\ &= -U_{\text{des}}\left(\frac{x_{b/f}^2}{\sqrt{\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2}} + \frac{z_{b/f}^2}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f}+f)^2}}\right. \\ &\quad \left. + \frac{\sqrt{\Delta^2 + y_{b/f}^2}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f}+f)^2}}\frac{y_{b/f}^2}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2 + z_{b/f}^2}}\right) \\ &\leq -u_{r,\text{des}}\left(\frac{x_{b/f}^2}{\sqrt{\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2}} + \frac{z_{b/f}^2}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f}+f)^2}}\right. \\ &\quad \left. + \frac{\sqrt{\Delta^2 + y_{b/f}^2}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f}+f)^2}}\frac{y_{b/f}^2}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2 + z_{b/f}^2}}\right) \\ &\triangleq W(x_{b/f}, y_{b/f}, z_{b/f}) < 0\end{aligned} \tag{8.41}$$

$\dot{V}$ is negative definite and thus the system (8.40) is asymptotically stable in $\mathbb{D}$. Hence, we can conclude that the control objectives are satisfied in this region. Note that global stability can not be claimed due to the singularities of the Euler angle representation. $\square$

8.4 Simulation Results

This section presents simulation results when the desired path is a straight line and a helix path, respectively. The simulated vehicle is the HUGIN AUV, which is produced by Kongsberg Maritime. In all simulations, the desired relative surge velocity $u_{r,\text{des}} = 5$ m/s and the ocean current is $\boldsymbol{V}_c = [0.5, -0.5, 0.5]^T$ m/s. The look-ahead distance $\Delta = 50$ m and the controller and observer gains are $k_{u_r} = 0.1$

s^{-1}, $\boldsymbol{K}_1 = \boldsymbol{I}$ s^{-1}, $\boldsymbol{K}_2 = \boldsymbol{I}$ s^{-1}, $k_{x_1} = k_{y_1} = k_{z_1} = 4$ s^{-1}, and $k_{x_2} = k_{y_2} = k_{z_2} = 0.05$ s^{-2}.

Two desired paths have been defined and simulated:

$$C_1 := \begin{cases} x_f(s) & = s \\ y_f(s) & = 0 \\ z_f(s) & = s\sin(\theta_f(s)) \\ \theta_f(s) & = -\frac{\pi}{4} \\ \psi_f(s) & = 0 \\ \kappa(s) & = 0 \\ \tau(s) & = 0 \end{cases} \tag{8.42}$$

$$C_2 := \begin{cases} x_f(s) & = R\cos(\frac{s}{\sqrt{R^2+(\frac{h}{2\pi})^2}}) \\ y_f(s) & = R\sin(\frac{s}{\sqrt{R^2+(\frac{h}{2\pi})^2}}) \\ z_f(s) & = \frac{h}{2\pi}\frac{s}{\sqrt{R^2+(\frac{h}{2\pi})^2}} \\ \theta_f(s) & = -\arctan(\frac{h}{2\pi R}) \\ \psi_f(s) & = \frac{s\cos(\theta_f(s))}{R} + \frac{\pi}{2} \\ \kappa(s) & = \frac{R}{R^2+(\frac{h}{2\pi})^2} \\ \tau(s) & = \frac{\frac{h}{2\pi}}{R^2+(\frac{h}{2\pi})^2} \end{cases} \tag{8.43}$$

In C_2, $R = 200$ m and $h = 100$ m.

It is trivial to verify that all Assumption are fulfilled. For instance, given a $V_{\max} = 1$ m/s and k=0.2, then $|\boldsymbol{V}_c| = 0.866\frac{m}{s} < V_{max} = ku_{r,\text{des}}$. Hence, Assumption 8.1 is satisfied. The ocean current observer is implemented with the same saturation, $|\hat{\boldsymbol{V}}_c| < V_{max}$. Furthermore, the simulations results confirm that $|\theta_{fc,\text{des}}| < \arccos(k)$ and that $|\theta|$ and $|\theta_{fc}| < \pi/2$, hence the system is always in the considered domain $\mathbb{D}$ (8.18).

The simulation results are shown in Figure 8.1-8.6, and confirm that the control objectives are fulfilled. The AUV converges to the desired path, so the path errors $x_{b/f}$, $y_{b/f}$, $z_{b/f}$ converge to 0. The controller ensure that the controlled states approach their reference values (Figure 8.4 and 8.8). Furthermore, the current observer correctly estimates the ocean currents (Figure 8.3 and 8.7).

Figure 8.2 and 8.6 display the movement of the AUV in the xy- and xz-plane. The vehicle is not aligned with the tangent of the path, but side-slips to counteract the influence of the ocean current, resulting in a total velocity that is aligned with the desired path.

All simulations were conducted with an appropriate saturation on the thruster force T and the rudder angles δ_q and δ_r, confirming that the control system is applicable in a real-life scenario.

8.5 Conclusions

In this chapter a guidance and control system for an underactuated underwater vehicle is developed to solve the control objective of making the vehicle follow a general path in the presence of unknown ocean currents. The result is motivated by the path following methods of [13], and by expanding these guidance and update laws and combining them with an ocean current observer, convergence to the desired path is achieved with asymptotic stability properties under explicit conditions. Simulation results verify the theoretical results.

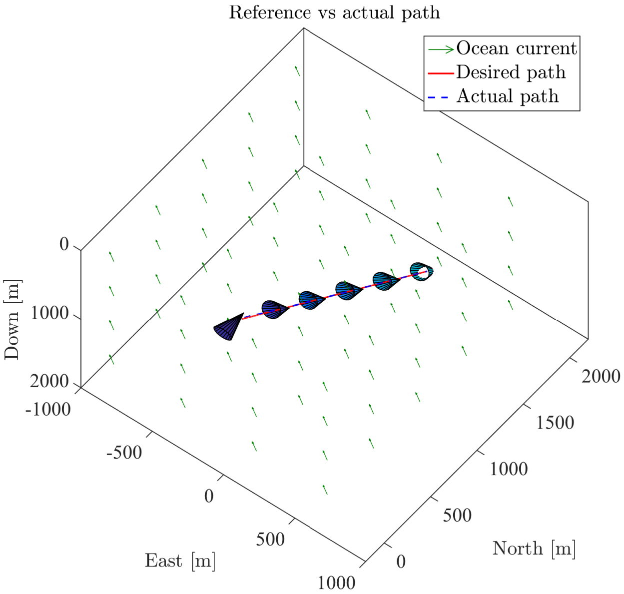
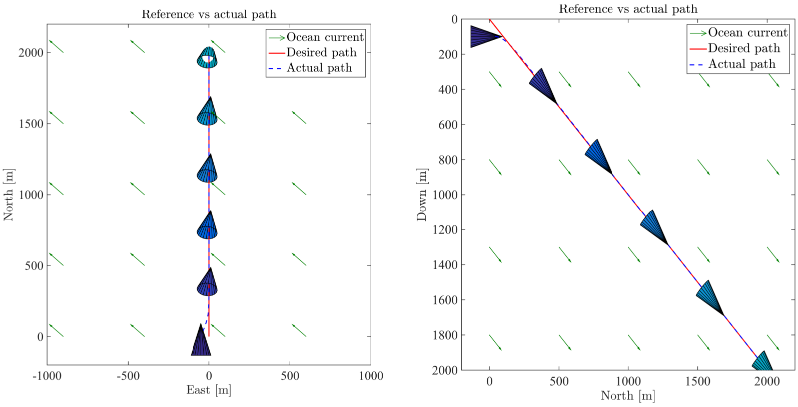


Figure 8.1: Desired and actual path C_1.



(a) xy-plane

(b) xz-plane

Figure 8.2: Path following of C_1. The vehicle side-slips to counteract the effects of the ocean current, so the total velocity is aligned with the path. Note in 8.2(b), the current direction is equal of that of the path, and so there is no side-slip in this direction.

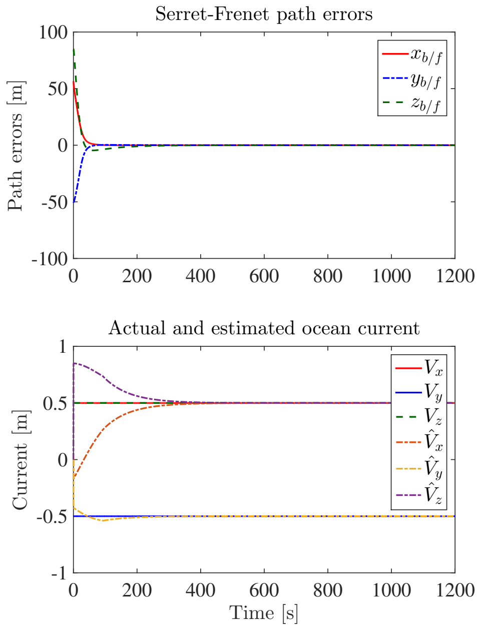


Figure 8.3: Path following errors and actual vs. estimated ocean current for C_1.

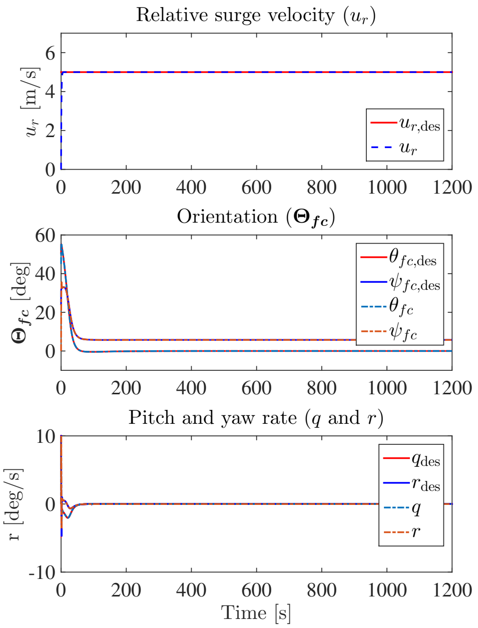


Figure 8.4: Desired vs. actual surge velocity, orientation and pitch/yaw rate for C_1.

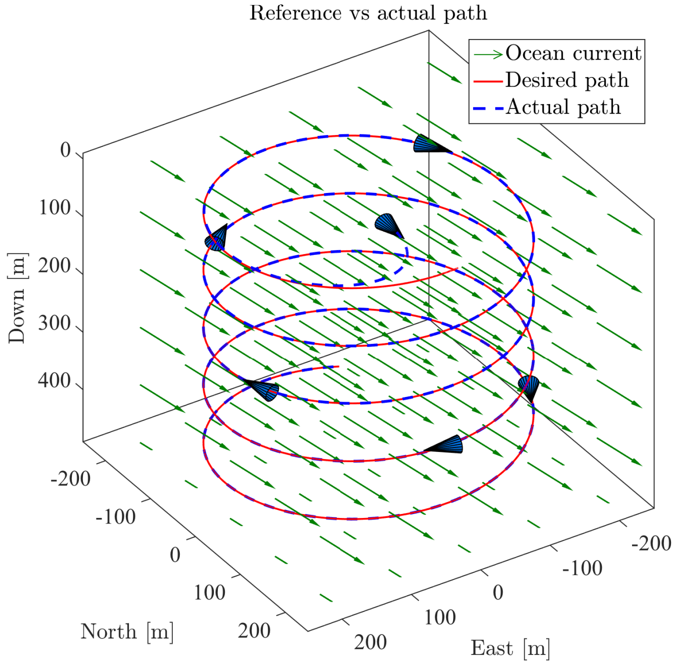
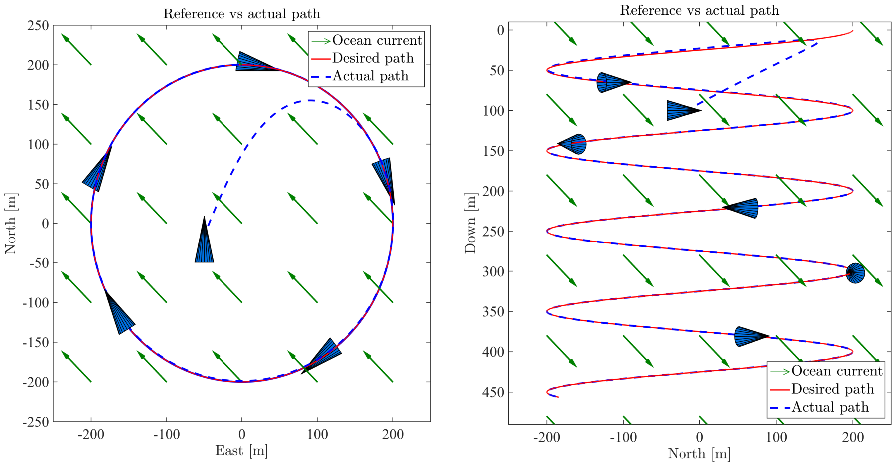


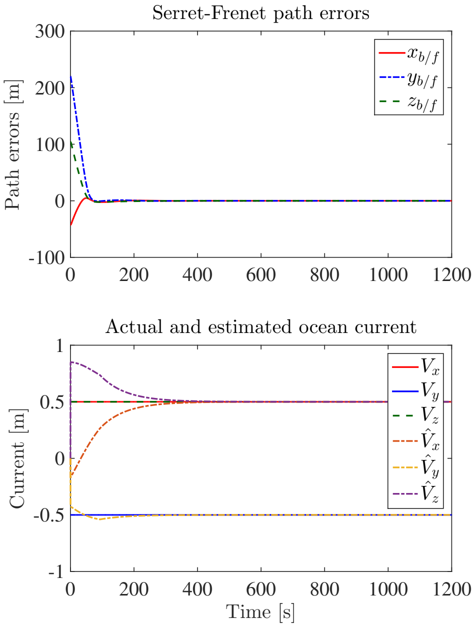
Figure 8.5: Desired and actual path C_2.



(a) xy-plane

(b) xz-plane

Figure 8.6: Path following of C_2. The vehicle side-slips to counteract the effects of the ocean current, so the total velocity is aligned with the path.



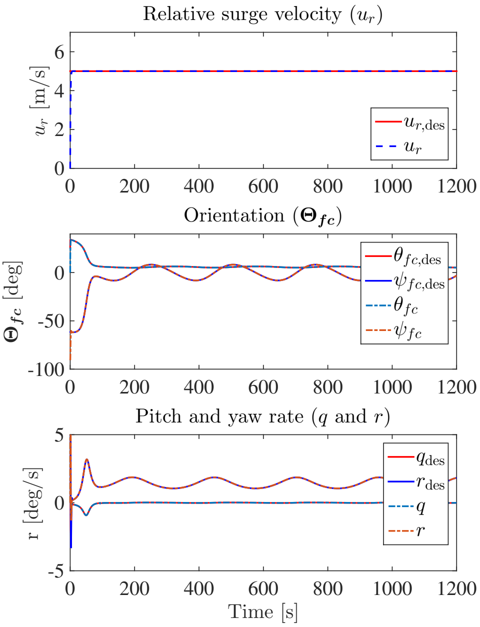


Figure 8.7: Path following errors and actual vs. estimated ocean current for C_2.

Figure 8.8: Desired vs. actual surge velocity, orientation and pitch/yaw rate for C_2.

Chapter 9

Line-of-Sight with side-slip for Surface Vessels and AUVs in the horizontal plane

In this chapter, the vehicle model is given by (2.23). We assume that the vessel in question is equipped with sensors to measure position, heading (yaw), yaw rate and absolute velocities.

Contributions of this chapter: This chapter proves that the LOS guidance law suggested in [33] for curved path following is suitable also for curved path following in combination with ocean current compensation by using adaptive feedback linearization combined with sliding mode. The proposed guidance and control system requires measurements of absolute velocities only, and is therefore more suitable than the control system in Chapter 7 and 8 when the vehicle in question lacks sensors to measure relative velocities. Furthermore, the guidance law is not based on Serret-Frenet frames, which allows for any parametrization of the path and removes the need for an update law for the Serret-Frenet frame and an ocean current observer. The results are applicable for USVs and AUVs restricted to the horizontal plane.

Organization of this chapter: This chapter is organized as follows. The control objectives are formalized in Section 9.1. The suggested control system is then introduced in Section 9.2 followed by the main result in Section 9.3. Finally, simulations results and conclusions are presented in Sections 9.4 and 9.5, respectively.

Publications: The results of this chapter are based on [68].

9.1 Control Objectives

This section formalizes the control problem solved in this chapter: The control system should make the vessel follow a given smooth path C and maintain a desired constant surge velocity u_{des} in the presence of unknown constant irrotational ocean currents. C is parametrized with respect to the inertial frame i. Let $\theta \geq 0$ denote the path variable. Then, the path C is parametrized by $(x_p(\theta), y_p(\theta))$.

The cross-track error y_e is computed as the orthogonal distance between the vessel position (x, y) to the path-tangential reference frame defined by the point $(x_p(\theta), y_p(\theta))$. The path variables are illustrated in Figure 9.1. Note that the position of the path-tangential reference frame is always such that the along-track error $x_e = 0$. The path-tangential frame corresponds to the inertial frame rotated by $\gamma_p(\theta)$, where

$$\gamma_p(\theta) = \arctan\left(\frac{y_p'(\theta)}{x_p'(\theta)}\right). \tag{9.1}$$

Hence,

$$\begin{bmatrix} 0 \\ y_e \end{bmatrix} = \boldsymbol{R}^T(\gamma_p(\theta)) \begin{bmatrix} x - x_p(\theta) \\ y - y_p(\theta) \end{bmatrix}, \tag{9.2}$$

where $\boldsymbol{R}$ is the rotation matrix defined in (2.1). Thus, the cross-track error is given as

$$y_e = -(x - x_p(\theta))\sin(\gamma_p(\theta)) + (y - y_p(\theta))\cos(\gamma_p(\theta)), \tag{9.3}$$

where the propagation of θ is given by [31]:

$$\dot{\theta} = \frac{U}{\sqrt{x_p'(\theta)^2 + y_p'(\theta)^2}} > 0, \tag{9.4}$$

and

$$U = \sqrt{u^2 + v^2} \tag{9.5}$$

is the total speed of the USV. Note that it is assumed that the path is an open curve, i.e. the end point is different from the start point. Definition 1 [33] guarantees that there is a unique solution for the cross-track error y_e obtained by minimizing θ. It can be shown that the cross-track error dynamics can be expressed as [33]

$$\begin{aligned} \dot{y}_e = &-(u\cos(\psi) - v\sin(\psi))\sin(\gamma_p(\theta)) \\ &+ (u\sin(\psi) + v\cos(\psi))\cos(\gamma_p(\theta)). \end{aligned} \tag{9.6}$$

It is obvious from Figure 9.1 that $y_e = 0$ implies that the vessel is on the desired path. Hence, we define the control objectives as

$$\begin{aligned} \lim_{t\to\infty} u(t) &= u_{\text{des}}, \\ \lim_{t\to\infty} y_e(t) &= 0. \end{aligned} \tag{9.7}$$

9.2 The Control System

This section presents the proposed guidance law and controllers. The system uses absolute velocity measurements only, making it unnecessary to buy expensive sensors to measure relative velocities. The reason why only absolute velocity measurements are required, is that the absolute velocity is the sum of relative velocity and

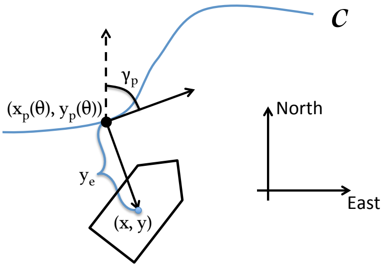


Figure 9.1: Desired path C, path-tangential reference frame with orientation $\gamma_p(\theta)$ and cross-track error y_e illustrated.

ocean currents, and thus implicitly contains information about these two components although the exact composition of relative velocities and ocean currents are unknown. Thus, by the design of the control system, the vessel is able to compensate both for the curvature of the path and the ocean current without knowing the exact components of these measurements.

9.2.1 Guidance Laws

The desired surge velocity is chosen to be constant and positive.

$$u_{\text{des}}(t) > 0, \dot{u}_{\text{des}}(t) \equiv 0. \tag{9.8}$$

The guidance law for the USV heading is chosen as

$$\psi_{\text{des}} = \gamma_{\text{p}}(\theta) \underbrace{- \arctan\left(\frac{v}{u_{\text{des}}}\right)}_{\triangleq \beta_{\text{des}}} - \arctan\left(\frac{y_e}{\Delta}\right). \tag{9.9}$$

9.2.2 Surge and Yaw Controllers

We define the following error signals:

$$\tilde{u} = u - u_{\text{des}}, \tag{9.10}$$

$$\tilde{\psi} = \psi - \psi_{\text{des}}, \tag{9.11}$$

$$\dot{\tilde{\psi}} = r - \dot{\psi}_{\text{des}}, \tag{9.12}$$

$$\boldsymbol{\xi} = \begin{bmatrix} \tilde{u} & \tilde{\psi} & \dot{\tilde{\psi}} \end{bmatrix}^T_. \tag{9.13}$$

An adaptive feedback linearizing PD-controller is used to ensure tracking of the desired heading ψ_{des}:

$$\begin{aligned} \tau_r &= -F_r(u,v,r) - \boldsymbol{\phi}_r^T(u,v,r,\psi)\hat{\boldsymbol{\theta}}_r + \ddot{\psi}_{\text{des}} \\ &\quad - (k_\psi + \lambda k_r)\tilde{\psi} - (k_r + \lambda)\dot{\tilde{\psi}} - k_d \text{sign}(\dot{\tilde{\psi}} + \lambda\tilde{\psi}) \qquad (9.14) \\ \dot{\hat{\boldsymbol{\theta}}}_r &= \gamma_r \boldsymbol{\phi}_r(u,v,r,\psi)\left(\dot{\tilde{\psi}} + \lambda\tilde{\psi}\right) \qquad (9.15) \end{aligned}$$

The gains $k_\psi, k_r, \lambda, k_d, \gamma_r > 0$ are constant and positive, and the function $\text{sign}(x)$ returns 1, 0 and -1 when x is positive, zero or negative, respectively. The expressions for $F_r(u,v,r)$ and $\boldsymbol{\phi}_r(u,v,r,\psi)$ are given in (2.28) and (2.29), respectively.

Similarly, an adaptive feedback linearizing P-controller is used to ensure tracking of the desired surge velocity u_{des}:

$$\begin{aligned} \tau_u &= -\frac{1}{m_{11}}(m_{22}v + m_{23}r)r + \frac{d_{11}}{m_{11}}u_{\text{des}} - \boldsymbol{\phi}_u^T(\psi,r)\hat{\boldsymbol{\theta}}_u \\ &\quad + \frac{d_{11}^q}{m_{11}}u^2 + \dot{u}_{\text{des}} - k_u\tilde{u} - k_e\text{sign}(\tilde{u}) \qquad (9.16) \\ \dot{\hat{\boldsymbol{\theta}}}_u &= \gamma_u \boldsymbol{\phi}_u(\psi,r)\tilde{u} \qquad (9.17) \end{aligned}$$

The gains k_u, k_e, γ_u are strictly positive constant controller gains. The expression for $\boldsymbol{\phi}_u(\psi,r)$ is given in (2.25), and m_{xy} and d_{xy} denote mass and damping parameters as defined in (2.16). Furthermore, $\hat{\boldsymbol{\theta}}_r$ and $\hat{\boldsymbol{\theta}}_u$ are estimates of the ocean current vectors $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_r$, given in (2.24).

The proposed controllers are similar to the controllers in [15], but in this chapter the terms $k_d\text{sign}(\dot{\tilde{\psi}} + \lambda\tilde{\psi})$ and $k_e\text{sign}(\tilde{u})$ have been added to increase the robustness of the controller with respect to model uncertainties. Note that the controllers (9.14) and (9.16) rely only on absolute velocity measurements, as relative velocities are not available for feedback.

Proposition 9.1. *Given an underactuated surface vessel described by the dynamical system (2.23). If Assumption 2.1-2.3 (Section 2.4.1) hold, the adaptive controllers (9.14)-(9.15) and (9.16)-(9.17) ensure that* $\boldsymbol{\xi} = \mathbf{0}$ *is a UGES equilibrium point and that the references provided by the guidance system are exponentially tracked.*

Proof. Defining

$$\tilde{\boldsymbol{\theta}}_r = \hat{\boldsymbol{\theta}}_r - \boldsymbol{\theta}_r, \qquad (9.18)$$

$$s = \dot{\tilde{\psi}} + \lambda\tilde{\psi}, \qquad (9.19)$$

the error dynamics of the heading controller system can be expressed as

$$\begin{bmatrix} \dot{\tilde{\psi}} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ -k_\psi & -k_r \end{bmatrix} \begin{bmatrix} \tilde{\psi} \\ s \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\phi}_r^T \end{bmatrix} \tilde{\boldsymbol{\theta}}_r - \begin{bmatrix} 0 \\ k_d\text{sign}(s) \end{bmatrix}, \qquad (9.20)$$

$$\dot{\tilde{\boldsymbol{\theta}}}_r = \gamma_r \boldsymbol{\phi}_r(u,v,r,\psi)s. \qquad (9.21)$$

To analyze the stability properties of the interconnected system (9.20)-(9.21), we consider the positive definite and radially unbounded Lyapunov function candidate

$$
\begin{aligned}
V &= \frac{1}{2}k_\psi\tilde{\psi}^2 + \frac{1}{2}s^2 + \frac{1}{2\gamma_r}\tilde{\boldsymbol{\theta}}_r^T\tilde{\boldsymbol{\theta}}_r, && (9.22)\\
&\Downarrow \\
\dot{V} &= -\lambda k_\psi\tilde{\psi}^2 - k_r s^2 - k_d|s| \leq 0. && (9.23)
\end{aligned}
$$

$\dot{V}$ is negative semi-definite, and thus the origin $(\tilde{\psi}, s, \tilde{\boldsymbol{\theta}}_r) = (0, 0, \mathbf{0})$ is a UGS equilibrium point of the interconnected system (9.20)-(9.21) [55]. Since $V(t) \leq V(0)$, $\tilde{\psi}$, s and $\tilde{\boldsymbol{\theta}}_r$ are bounded. Thus, there exists some bounds α, β, ζ such that

$$
\begin{aligned}
|\tilde{\psi}(t)| &\leq \alpha, && && (9.24)\\
|s(t)| &\leq \beta, && \forall\, t \geq 0 && (9.25)\\
||\tilde{\boldsymbol{\theta}}_r(t)|| &\leq \zeta. && && (9.26)
\end{aligned}
$$

It is straightforward to show that

$$
\begin{aligned}
\left|\tilde{V}_x\right| &\leq \sqrt{\tilde{V}_x^2 + \tilde{V}_y^2 + \tilde{V_x^2}^2 + \tilde{V_y^2}^2 + \widetilde{V_xV_y}^2} = ||\tilde{\boldsymbol{\theta}}_r(t)|| < \zeta, && (9.27)\\
\left|\tilde{V}_y\right| &\leq ||\tilde{\boldsymbol{\theta}}_r(t)|| < \zeta, && (9.28)\\
&\vdots \\
\left|\widetilde{V_xV_y}\right| &\leq ||\tilde{\boldsymbol{\theta}}_r(t)|| < \zeta. && (9.29)
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\left|\boldsymbol{\phi}_r^T\tilde{\boldsymbol{\theta}}_r\right| &= |\phi_1\tilde{V}_x + \phi_2\tilde{V}_y + \phi_3\tilde{V_x^2} + \phi_4\tilde{V_y^2} + \phi_5\widetilde{V_xV_y}|\\
&\leq |\phi_1||\tilde{V}_x| + |\phi_2||\tilde{V}_y| + |\phi_3||\tilde{V_x^2} + |\phi_4||\tilde{V_y^2}| + |\phi_5||\widetilde{V_xV_y}|\\
&\leq \zeta\left(|\phi_1| + |\phi_2| + |\phi_3| + |\phi_4| + |\phi_5|\right) = \zeta|\boldsymbol{\phi}_r|. && (9.30)
\end{aligned}
$$

Since the equilibrium point $(\tilde{\psi}, s, \tilde{\boldsymbol{\theta}}_r) = (0, 0, \mathbf{0})$ is UGS, the states have no finite escape time and the system (9.20)-(9.21) is forward complete. Thus, we can analyze (9.20) under the assumption that $\tilde{\boldsymbol{\theta}}_r$ is a bounded time-varying signal. Consider the positive definite and radially unbounded Lyapunov function candidate

$$
\begin{aligned}
V_\psi &= \frac{1}{2}k_\psi\tilde{\psi}^2 + \frac{1}{2}s^2, && (9.31)\\
&\Downarrow \\
\dot{V}_\psi &= -\lambda k_\psi\tilde{\psi}^2 - k_r s^2 - \boldsymbol{\phi}_r^T\tilde{\boldsymbol{\theta}}s - k_d|s|\\
&\leq -\lambda k_\psi\tilde{\psi}^2 - k_r s^2 + (|\boldsymbol{\phi}_r^T\tilde{\boldsymbol{\theta}}| - k_d)|s|\\
&\leq -\lambda k_\psi\tilde{\psi}^2 - k_r s^2 + (\zeta|\boldsymbol{\phi}_r| - k_d)|s|.\\
& && (9.32)
\end{aligned}
$$

We choose the gain $k_d > 2\zeta|\boldsymbol{\phi}_r|$. Thus,

$$\begin{aligned} \dot{V}_\psi &\leq -\lambda k_\psi \tilde{\psi}^2 - k_r s^2 - \zeta|\boldsymbol{\phi}_r||s| \\ &\leq -\lambda k_\psi \tilde{\psi}^2 - k_r s^2 < 0. \end{aligned} \tag{9.33}$$

$\dot{V}_\psi$ is negative definite, and thus the origin $(\tilde{\psi}, s) = (0,0)$ is a UGAS equilibrium point of the system (9.20). Furthermore, Theorem 4.10 [43] (given in Section 2.3) is satisfied with $k_1 = \min(k_\psi/2, 1/2)$, $k_2 = \max(k_\psi/2, 1/2)$, $k_3 = \min(\lambda k_\psi, k_r)$ and $a = 2$. This implies that the origin $(\tilde{\psi}, s) = (0,0)$ is a UGES equilibrium point of the system (9.20), and

$$\lim_{t\to\infty} \tilde{\psi} = 0, \tag{9.34}$$

$$\lim_{t\to\infty} \dot{\tilde{\psi}} = \lim_{t\to\infty} s - \lambda\tilde{\psi} = 0. \tag{9.35}$$

Similarly, defining

$$\tilde{\boldsymbol{\theta}}_u = \hat{\boldsymbol{\theta}}_u - \boldsymbol{\theta}_u, \tag{9.36}$$

the error dynamics of the surge controller system is given by

$$\dot{\tilde{u}} = -\left(\frac{d_{11}}{m_{11}} + k_u\right)\tilde{u} - \boldsymbol{\phi}_u^T(\psi, r)\tilde{\boldsymbol{\theta}}_u - k_e \text{sign}(\tilde{u}), \tag{9.37}$$

$$\dot{\tilde{\boldsymbol{\theta}}}_u = \gamma_u \boldsymbol{\phi}_y(\psi, r)\tilde{u}. \tag{9.38}$$

To analyze the stability properties of the interconnected system (9.37)-(9.38), we consider the positive definite and radially unbounded Lyapunov function candidate

$$V = \frac{1}{2}\tilde{u}^2 + \frac{1}{2\gamma_u}\tilde{\boldsymbol{\theta}}_u^T\tilde{\boldsymbol{\theta}}_u, \tag{9.39}$$

$$\Downarrow$$

$$\dot{V} = -\left(\frac{d_{11}}{m_{11}} + k_u\right)\tilde{u}^2 - k_e|\tilde{u}| \leq 0. \tag{9.40}$$

$\dot{V}$ is negative semi-definite, and thus the origin $(\tilde{u}, \tilde{\boldsymbol{\theta}}_u) = (0, \mathbf{0})$ is an UGS equilibrium point of the interconnected system (9.37)-(9.38). Since $V(t) \leq V(0)$, $\tilde{u}$, and $\tilde{\boldsymbol{\theta}}_u$ are bounded. Thus, there exists some bounds α, β, ζ such that

$$|\tilde{u}(t)| \leq \iota, \qquad \forall\, t \geq 0 \tag{9.41}$$

$$||\tilde{\boldsymbol{\theta}}_u(t)|| \leq \kappa. \tag{9.42}$$

Similarly as the analysis for system (9.20), we can analyze (9.37) under the assumption that $\tilde{\boldsymbol{\theta}}_u$ is a bounded time-varying signal. Consider the positive definite and radially unbounded Lyapunov function candidate

$$\begin{aligned} V_u &= \frac{1}{2}\tilde{u}^2, \qquad (9.43) \\ &\Downarrow \\ \dot{V}_u &= -\left(\frac{d_{11}}{m_{11}} + k_u\right)\tilde{u}^2 - \boldsymbol{\phi}_t^T\tilde{\boldsymbol{\theta}}_u\tilde{u} - k_e|\tilde{u}| \\ &\leq -\left(\frac{d_{11}}{m_{11}} + k_u\right)\tilde{u}^2 + (|\boldsymbol{\phi}_t^T\tilde{\boldsymbol{\theta}}_u| - k_e)|\tilde{u}| \\ &\leq -\left(\frac{d_{11}}{m_{11}} + k_u\right)\tilde{u}^2 + (\kappa|\boldsymbol{\phi}_t| - k_e)|\tilde{u}|. \qquad (9.44) \end{aligned}$$

We choose the gain $k_e > 2\kappa|\boldsymbol{\phi}_u|$. Thus,

$$\begin{aligned} \dot{V}_\psi &\leq -\left(\frac{d_{11}}{m_{11}} + k_u\right)\tilde{u}^2 - \kappa|\boldsymbol{\phi}_t||\tilde{u}| \\ &\leq -\left(\frac{d_{11}}{m_{11}} + k_u\right)\tilde{u}^2 < 0. \qquad (9.45) \end{aligned}$$

$\dot{V}_u$ is negative definite, and thus the origin $\tilde{u} = 0$ is a UGAS equilibrium point of the system (9.37). Furthermore, Theorem 4.10 [43] is satisfied with $k_1 = k_2 = 1/2$, $k_3 = d_{11}/m_{11} + k_u$ and $a = 2$. This implies that the origin $\tilde{u} = 0$ is a UGES equilibrium point of the system (9.37), and

$$\lim_{t\to\infty} \tilde{u} = 0. \qquad (9.46)$$

Thus, the errors $\boldsymbol{\xi} = \begin{bmatrix} \tilde{u} & \tilde{\psi} & \dot{\tilde{\psi}} \end{bmatrix}^T$ converge exponentially to zero. □

Assumption 9.1: The controllers (9.14) and (9.16) are tuned such that the error dynamics in (9.20) and (9.37) is fast compared to the cross-track error dynamics (9.6). Hence, $\boldsymbol{\xi}$ can be assumed to be zero.

9.2.3 State Measurements

The control system proposed in this chapter assumes that $\boldsymbol{\eta}$ (position and heading/yaw) and $\boldsymbol{\nu}$ (surge and sway velocity and yaw rate) are measured. Ships are usually equipped with a large variety of sensors that combined provide sensor data to estimate the vessel state [31]. We assume that the ship is equipped with Global Navigation Satellite System (GNSS) receivers to provide position and velocity measurements and a gyrocompass to measure yaw ψ yaw rate r. Furthermore, we assume that the measurements are filtered to remove first-order wave-induced motions and measurement noise.

9.3 Main Result

This section presents the conditions under which the proposed control system achieves the control objectives (9.7).

Theorem 9.1. *Given an underactuated surface vessel described by the dynamical system (2.23). If Assumption 2.1-2.3 and 9.1 hold, the controllers given by (9.14) and (9.16), and the guidance laws given by (9.8) and (9.9), give a cross-track error dynamics for which the equilibrium point $y_e = 0$ is UGAS and USGES, and the control objectives (9.7) are achieved.*

Proof. The dynamics of the path cross track error is given in (9.6), and can be rewritten as follows:

$$\begin{aligned}
\dot{y}_e &= -\left(u\cos(\psi) - v\sin(\psi)\right)\sin\left(\gamma_{\mathrm{p}}(\theta)\right) \\
&\qquad + \left(u\sin(\psi) + v\cos(\psi)\right)\cos\left(\gamma_{\mathrm{p}}(\theta)\right) \\
&= u\left(\sin(\psi)\cos(\gamma_{\mathrm{p}}(\theta)) - \cos(\psi)\sin(\gamma_{\mathrm{p}}(\theta))\right) \\
&\qquad + v\left(\cos(\psi)\sin(\gamma_{\mathrm{p}}(\theta)) + \sin(\psi)\cos(\gamma_{\mathrm{p}}(\theta))\right) \\
&= (u_{\mathrm{des}} + \tilde{u})\sin(\psi - \gamma_{\mathrm{p}}(\theta)) + v\cos(\psi - \gamma_{\mathrm{p}}(\theta)) \\
&= U_{\mathrm{des}}\sin\left(\psi - \gamma_{\mathrm{p}}(\theta) + \beta_{c,\mathrm{des}}\right) + \tilde{u}\sin(\psi - \gamma_{\mathrm{p}}(\theta)),
\end{aligned} \tag{9.47}$$

where

$$U_{\mathrm{des}} = \sqrt{u_{\mathrm{des}}^2 + v^2}. \tag{9.48}$$

Inserting the expressions for the guidance law in yaw (9.9) and error signal (9.11), the sine terms in (9.47) can be rewritten as below:

$$\begin{aligned}
\sin(\psi - \gamma_{\mathrm{p}}(\theta) + \beta_{\mathrm{des}}) &= \sin\left(\tilde{\psi} + \psi_{\mathrm{des}} - \gamma_{\mathrm{p}}(\theta) + \beta_{\mathrm{des}}\right) \\
&= \sin\left(\tilde{\psi} - \arctan\left(\frac{y_e}{\Delta}\right)\right) \\
&= \sin(\tilde{\psi})\cos\left(\arctan\left(\frac{y_e}{\Delta}\right)\right) - \cos(\tilde{\psi})\sin\left(\arctan\left(\frac{y_e}{\Delta}\right)\right) \\
&= -\sin\left(\arctan\left(\frac{y_e}{\Delta}\right)\right) + \sin(\tilde{\psi})\cos\left(\arctan\left(\frac{y_e}{\Delta}\right)\right) \\
&\qquad - \left(\cos(\tilde{\psi}) - 1\right)\sin\left(\arctan\left(\frac{y_e}{\Delta}\right)\right) \\
&= -\frac{y_e}{\sqrt{\Delta^2 + y_e^2}} + \sin(\tilde{\psi})\frac{\Delta}{\sqrt{\Delta^2 + y_e^2}} - \left(\cos(\tilde{\psi}) - 1\right)\frac{y_e}{\sqrt{\Delta^2 + y_e^2}},
\end{aligned} \tag{9.49}$$

$$\begin{aligned}
\sin\left(\psi - \gamma_{\mathrm{p}}(\theta)\right) &= \sin\left(\tilde{\psi} + \psi_{\mathrm{des}} - \gamma_{\mathrm{p}}(\theta)\right) \\
&= \sin\left(\tilde{\psi} - \beta_{\mathrm{des}} - \arctan\left(\frac{y_e}{\Delta}\right)\right) \\
&= \sin\left(\tilde{\psi} - \arctan\left(\frac{\Delta v + y_e u_{\mathrm{des}}}{\Delta u_{\mathrm{des}} - y_e v}\right)\right).
\end{aligned} \tag{9.50}$$

Inserting (9.49)-(9.50) into (9.47) yields

$$\begin{aligned}\dot{y}_e = U_{\text{des}} \Bigg(&-\frac{y_e}{\sqrt{\Delta^2 + y_e^2}} + \sin(\tilde{\psi})\frac{\Delta}{\sqrt{\Delta^2 + y_e^2}} \\ &- \left(\cos(\tilde{\psi}) - 1\right)\frac{y_e}{\sqrt{\Delta^2 + y_e^2}}\Bigg) + \tilde{u}\sin\left(\tilde{\psi} - \arctan\left(\frac{\Delta v + y_e u_{\text{des}}}{\Delta u_{\text{des}} - y_e v}\right)\right) \\ = f_1(t, y_e) + \boldsymbol{g}(t, y_e, \boldsymbol{\xi})\boldsymbol{\xi}, \end{aligned} \tag{9.51}$$

with

$$f_1(t, y_e) = -U_{\text{des}}\frac{y_e}{\sqrt{\Delta^2 + y_e^2}}, \tag{9.52}$$

$$\boldsymbol{g}(t, y_e, \tilde{\psi}) = \begin{bmatrix} \sin\left(\tilde{\psi} - \arctan\left(\frac{\Delta v + y_e u_{\text{des}}}{\Delta u_{\text{des}} - y_e v}\right)\right) \\ U_{\text{des}} g_1(t, y_e, \tilde{\psi}) \\ 0 \end{bmatrix}^T, \tag{9.53}$$

where

$$g_1(t, y_e, \tilde{\psi}) = \frac{\sin(\tilde{\psi})}{\tilde{\psi}}\frac{\Delta}{\sqrt{\Delta^2 + y_e^2}} - \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}}\frac{y_e}{\sqrt{\Delta^2 + y_e^2}}. \tag{9.54}$$

By Assumption 9.1, $\boldsymbol{\xi} = \mathbf{0}$, reducing the error dynamics to

$$\dot{y}_e = f_1(t, y_e) = -U_{\text{des}}\frac{y_e}{\sqrt{\Delta^2 + y_e^2}}. \tag{9.55}$$

Stability of the nominal system can be shown using the quadratic, positive definite, decrescent and radially unbounded Lyapunov function $V = \frac{1}{2}y_e^2$.

$$\dot{V} = y_e\dot{y}_e = -\frac{U_{\text{des}}}{\sqrt{\Delta^2 + y_e^2}}y_e^2. \tag{9.56}$$

Since

$$U_{\text{des}} = \sqrt{u_{\text{des}}^2 + v^2} \geq u_{\text{des}} > 0 \tag{9.57}$$

by (9.8), $\dot{V}$ is negative definite along the trajectories of the system (9.55), which is UGAS (see Theorem 4.8/4.9 in Section 2.3), and

$$|y_e(t)| \leq |y_e(t_0)| \; \forall \; t \geq t_0. \tag{9.58}$$

Furthermore, using the same approach as the proof in Appendix A in [33], we define

$$\phi(t, y_e) = \frac{U_{\text{des}}}{\sqrt{\Delta^2 + y_e^2}}. \tag{9.59}$$

For each $r > 0$ and $|y_e(t)| \leq r$, we have

$$\phi(t, y_e) \geq \frac{u_{\text{des}}}{\sqrt{\Delta^2 + r^2}} \triangleq c(r). \tag{9.60}$$

Consequently,

$$\dot{V}(y, y_e) = -2\phi(t, y_e)V(t, y_e) \leq -2c(r)V(t, y_e) \quad \forall\ |y_e(t)| \leq r. \tag{9.61}$$

Given (9.58), the above holds for all trajectories generated by the initial conditions $y_e(t_0)$. Therefore, the comparison lemma (Lemma 3.4 [43]) can be used to prove that the equilibrium point $y_e = 0$ is USGES. The differential equation $\dot{z} = -2c(r)z$ has the solution $z(t) = e^{-2c(r)(t-t_0)}z(t_0)$, which implies that

$$w(t) \leq e^{-2c(r)(t-t_0)}w(t_0), \qquad w(t) = V(t, y_e(t)). \tag{9.62}$$

Furthermore,

$$y_e(t) = 2\sqrt{w(t)}, \tag{9.63}$$

which implies that

$$y_e(t) \leq e^{-c(r)(t-t_0)}y_e(t_0). \tag{9.64}$$

for all $t \geq t_0$, $|y_e(t_0)| \leq r$ and $r > 0$. Thus, we can conclude that the equilibrium point $y_e = 0$ is USGES (Definition 2.7. [50], see Section 2.3).

We have thus proved that the cross track error dynamics (9.6) have global convergence to the equilibrium point $y_e = 0$. In addition, the local convergence properties are even stronger than asymptotic. In particular, it is shown that $y_e = 0$ is USGES and that the control objectives (9.7) are satisfied. ☐

9.4 Simulation Results

This section presents simulation results for two different curved paths. The simulated vehicle is a HUGIN AUV, produced by Kongsberg Maritime, restricted to movement in the horizontal plane. In all simulations, the desired relative surge velocity is chosen as $u_{\text{des}} = 3$ m/s and the lookahead distance Δ as $\Delta = 50$ m. The ocean current $\boldsymbol{V}_c = [0.3, 0.3]^T$ m/s, and the controller gains are chosen as $k_\psi = 1.2\text{s}^{-2}$, $k_r = 1.3\text{s}^{-1}$, $\lambda = 0.5\text{s}^{-1}$, $k_d = 0.1\text{s}^{-2}$, $k_u = 0.1\text{s}^{-1}$ and $k_e = 0.1\text{ms}^{-2}$. The dimensionless adaptive gains are chosen as $\gamma_u = \gamma_r = 0.1$. Note that to avoid chattering about the equilibrium point, in the simulations the discontinuous term $\text{sign}(z)$ in the controllers (9.14) and (9.16) has been replaced with the continuous function $\tanh(10z)$.

Two desired paths have been defined and simulated:

$$C_1 := \begin{cases} x_p(\theta) = & \theta \\ y_p(\theta) = & 30\sin(0.005\theta) \end{cases} \tag{9.65}$$

$$C_2 := \begin{cases} x_p(\theta) = & 1.2\theta\sin(0.005\theta) \\ y_p(\theta) = & 600\cos(0.005\theta) - 650 \end{cases} \tag{9.66}$$

Simulations results for the two paths are presented in Figure 9.2-9.3 and 9.4-9.5, respectively. In both simulations, the cross-track error converges to zero and the relative surge velocity to the constant desired value. Thus, the control objectives are satisfied, and the controllers achieve tracking of the references ψ_{des} and $\dot{\psi}_{\text{des}}$.

Figure 9.2 and 9.4 both illustrate that the USV side-slips to follow the desired path. The AUV heading is not aligned with the path heading, but the AUV velocity is aligned with the path, ensuring that the vessel follows the curved path without knowledge of the ocean current direction or magnitude.

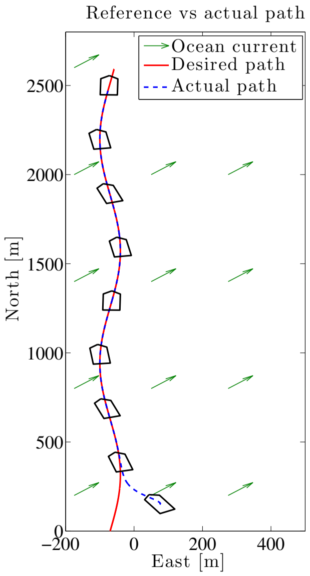


Figure 9.2: Desired and actual path C_1.

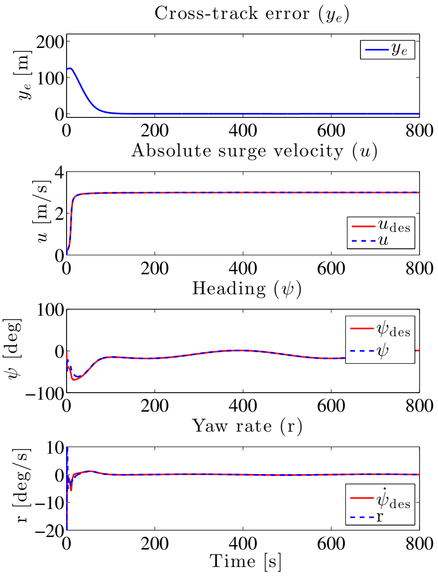


Figure 9.3: Cross-track error and desired vs. actual surge velocity, yaw and yaw rate for C_1.

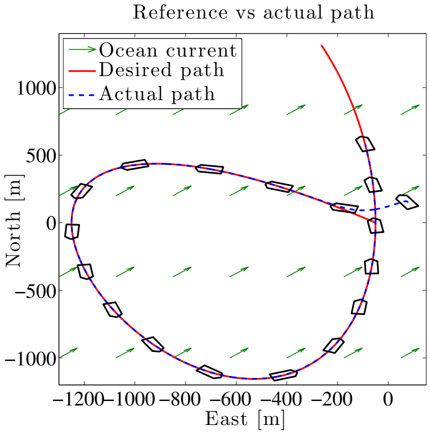


Figure 9.4: Desired and actual path C_2.

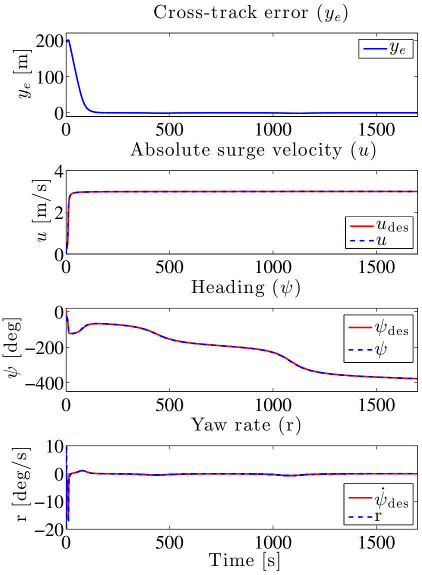


Figure 9.5: Cross-track error and desired vs. actual surge velocity, yaw and yaw rate for C_2.

9.5 Conclusions

In this chapter a guidance and control system for underactuated surface vessels is developed to solve the control objective of making the vessel follow a curved path in the presence of unknown ocean currents. The results are also applicable to AUVs moving in the horizontal plane.

The chapter is motivated by the guidance law suggested in [33] designed for curved path following when no ocean currents are considered. Furthermore, the guidance and control system presented in this chapter does not depend on the relative velocity sensors required by the control system in Chapter 7 and 8. It is proven that a similar guidance law combined with an adaptive feedback linearizing controller with sliding mode, achieves convergence to the desired path with UGAS and USGES stability properties under explicit conditions. Only absolute velocity measurement are required. Simulation results verify the theoretical results.

Part III

A Set-based Approach for Path Following of Marine Vehicles with Collision Avoidance

Chapter 10

Set-based Line-of-Sight Path Following with Collision Avoidance

This chapter considers guidance and control of a USV described by (2.18) when no ocean currents are present. Furthermore, we assume linear damping in surge, so the constant d_{11}^q in (2.18) is zero.

Contributions of this chapter: The contribution of this chapter is twofold. Firstly, it is suggested to apply the set-based theory presented in Chapter 4-6 to satisfy two objectives: Collision avoidance and path following for an unmanned surface vehicle. The approach defines collision avoidance as a high-priority set-based task and path following as a low-priority equality task, and is adapted to the underactuated USV by switching between two predefined guidance laws rather than combining them using the Null-Space-Based Inverse Kinematics approach for fully actuated systems as in Part I in this thesis. The guidance laws, if satisfied, will ensure path following and collision avoidance respectively. Thus the system is equipped with one path following mode and one collision avoidance mode, in addition to a defined and deterministic method for switching between these two. This results in a tighter coupling between collision avoidance, path planning and guidance than standard VO implementations. Furthermore, this method can be used for any combination of path following and collision avoidance guidance laws, making it a highly generic solution. Secondly, a specific LOS-based guidance law for collision avoidance is suggested. This guidance law, if satisfied, will ensure that the USV tracks a circle with constant radius about the obstacle center, which may be stationary or moving, and is specifically designed to assure collision avoidance while abiding by the COLREGs. The guidance system is especially suitable to avoid collisions with small and dynamic obstacles, and can easily be combined with existing global path planning methods that handle topography [41].

Organization of this chapter: The control objectives are formalized in Section 10.1. The suggested guidance and control system is presented in Section 10.2 and the main results in Section 10.3 before simulation results are given in Section 10.4. Conclusions are given in Section 10.5.

Publications: The results of this chapter are based on [61].

10.1 Control Objectives

This section formalizes the control problem solved in this chapter. The USV has several objectives: avoid all obstacles in a COLREGs compliant manner, follow a predefined path C and keep a given surge velocity along the path. It is clear that the two first objectives may be in conflict. Hence, in the case both cannot be achieved, collision avoidance needs to have first priority to ensure safe passage of the USV.

The path cross-track error y_e is computed as the shortest distance between the USV and any point on the path, and is defined so that $y_e = 0$ implies that the USV is on the desired path. In the case of a path parametrized as a function of θ it is the orthogonal distance between the USV position (x, y) to the path-tangential reference frame defined by the point $(x_p(\theta), y_p(\theta))$. It is assumed that the path is an open curve, i.e. the end point is different from the start point.

The control objectives are formalized in prioritized order below:

1. The distance between the USV with position $\boldsymbol{p}(t)$ and every obstacle with position $\boldsymbol{p}_\mathrm{o}(t)$ should always be greater than or equal to some safe distance R_o:
$$|\boldsymbol{p}(t) - \boldsymbol{p}_\mathrm{o}(t)| \geq R_\mathrm{o} \;\; \forall \;\; t \geq t_0 \tag{10.1}$$
2. The USV should converge to the desired path.
$$\lim_{t \to \infty} y_e(t) = 0 \tag{10.2}$$
3. The USV surge velocity should track to some desired, positive velocity.
$$\lim_{t \to \infty} u(t) = u_\mathrm{des}(t) \tag{10.3}$$

Note that $u_\mathrm{des}(t)$ will be provided by the set-based guidance system and will depend on the mode of the system, i.e. if the USV is in path following mode or collision avoidance mode.

10.2 The Control System

This section presents the guidance and control system consisting of guidance laws and controllers. The guidance system consists of separate guidance laws for path following and collision avoidance, and an algorithm to switch between these two.

10.2.1 Guidance Laws for Path Following

The desired surge velocity u_des for path following is constant, positive and is denoted u_pf. Furthermore, the desired heading for path following is denoted ψ_pf, where ψ_pf is a suitable guidance law for following the desired path in question. For instance, in the case of a straight line path, a suitable choice for ψ_pf would be a LOS guidance law designed for straight paths as given in [35], which is proven to give UGAS and ULES, and thus makes the path cross-track error converge to zero.

Assumption 10.1: The path following guidance law is chosen so that, if satisfied, the path cross-track error will converge to zero.

Remark 10.1*:* The above is not a strict assumption as it provides freedom to choose between a variety of stable path following schemes, such as for instance Line-of-Sight.

10.2.2 Guidance Laws for collision avoidance

This chapter suggests a specific guidance law to safely avoid obstacles while abiding COLREGs. Note that this guidance law is completely independent of the set-based algorithm presented in the next subsection, and that it may be replaced by any other guidance law to ensure collision avoidance.

In the case of collision avoidance, the goal is to track a safe radius around the object center. If this radius is maintained, a collision will never occur. Denote the obstacle safe radius as R_o, the obstacle center as

$$\boldsymbol{p}_\mathrm{o}(t) = \begin{bmatrix} x_\mathrm{o}(t) \\ y_\mathrm{o}(t) \end{bmatrix} \tag{10.4}$$

and the obstacle velocity as

$$U_\mathrm{o} = \sqrt{\dot{x}_\mathrm{o}^2 + \dot{y}_\mathrm{o}^2}. \tag{10.5}$$

Assumption 10.2: The obstacle speed is upper bounded by $U_{\mathrm{o,max}}$:

$$U_\mathrm{o} \leq U_{\mathrm{o,max}} \tag{10.6}$$

Denote

$$\phi = \arctan\left(\frac{y - y_\mathrm{o}}{x - x_\mathrm{o}}\right), \tag{10.7}$$

$$\beta_\mathrm{o} = \arctan\left(\frac{\dot{y}_\mathrm{o}}{\dot{x}_\mathrm{o}}\right), \tag{10.8}$$

$$V_\mathrm{o} = U_\mathrm{o} \cos(\phi - \beta_\mathrm{o}), \tag{10.9}$$

where ϕ and β_o are illustrated in Figure 10.1. The velocity V_o describes the velocity of the obstacle relative to the position of the USV, where a positive V_o suggests that the obstacle is moving closer to the USV. It reaches its maximum value of U_o when the obstacle is moving straight towards the current position of the USV (not taking the USV velocity or heading into account).

The desired surge velocity u_{des} for collision avoidance is constant, positive and is denoted u_{oa}.

Assumption 10.3: We assume that the obstacle speed is lower than the desired surge velocity, i.e.

$$u_{\mathrm{oa}} > U_{\mathrm{o,max}}. \tag{10.10}$$

Remark 10.2*:* It is natural to assume that the USV moves sufficiently fast to avoid the obstacle by moving around it. Furthermore, Assumption 10.2-10.3 ensure that the term k in the collision avoidance heading guidance law (10.11) is real.

The following guidance law giving the desired heading for collision avoidance is proposed:

$$\psi_{\mathrm{oa}} = \phi + \lambda \left(\frac{\pi}{2} - \arctan\left(\frac{e+k}{\Delta} \right) \right) - \arctan\left(\frac{v}{u_{\mathrm{oa}}} \right), \tag{10.11}$$

where $\lambda = 1$ corresponds to clock-wise motion and $\lambda = -1$ to counter-clockwise motion. Note that λ should be chosen in accordance with COLREGs, see Section 10.2.3. The guidance parameter $\Delta > 0$ is a design parameter corresponding to the look-ahead distance, e is the cross-track error of the circular path defined as

$$e = R_{\mathrm{o}} - \rho = R_{\mathrm{o}} - \sqrt{(x-x_{\mathrm{o}})^2 + (y-y_{\mathrm{o}})^2}, \tag{10.12}$$

and k is defined as

$$k = \begin{cases} k_1 & V_{\mathrm{o}} \geq 0 \\ k_2 & V_{\mathrm{o}} < 0, \end{cases} \tag{10.13}$$

$$k_{\{1,2\}} = \frac{-b\,\{+,-\}\,\sqrt{b^2-4ac}}{2a}, \tag{10.14}$$

where

$$a = U_{\mathrm{oa}}^2 - V_{\mathrm{o}}^2, \tag{10.15}$$

$$b = -2V_{\mathrm{o}}^2 e, \tag{10.16}$$

$$c = -V_{\mathrm{o}}^2(\Delta^2 + e^2), \tag{10.17}$$

and

$$U_{\mathrm{oa}} = \sqrt{u_{\mathrm{oa}}^2 + v^2}. \tag{10.18}$$

By Assumption 10.2-10.32 and 3, it is easy to verify that $a > 0$ and $c \leq 0$. Hence $k_1 \geq 0$ and $k_2 \leq 0$. The parameter k is designed to compensate for the movement of the obstacle, and thus the sign of the compensation shifts as the USV traverses the obstacle radius and the movement of the obstacle relative to the USV changes.

Theorem 10.1. *Given Assumption 10.2-10.3, if the guidance laws* $u_{des} = u_{oa}$ *and* $\psi_{des} = \psi_{oa}$ *are satisfied, the cross-track error* e *will asymptotically converge to zero and the USV (2.18) will track the radius* R_o *about the obstacle center* $\boldsymbol{p}_o(t)$*.*

Proof. The error dynamics is given by

$$\dot{e} = \dot{R}_{\mathrm{o}} - \dot{\rho} = -\dot{\rho}, \tag{10.19}$$

where

$$\rho = \sqrt{(x-x_{\mathrm{o}})^2 + (y-y_{\mathrm{o}})^2} \tag{10.20}$$

Rewriting into polar coordinates,

$$x - x_{\mathrm{o}} = \rho\cos(\phi) \tag{10.21}$$

$$y - y_{\mathrm{o}} = \rho\sin(\phi), \tag{10.22}$$

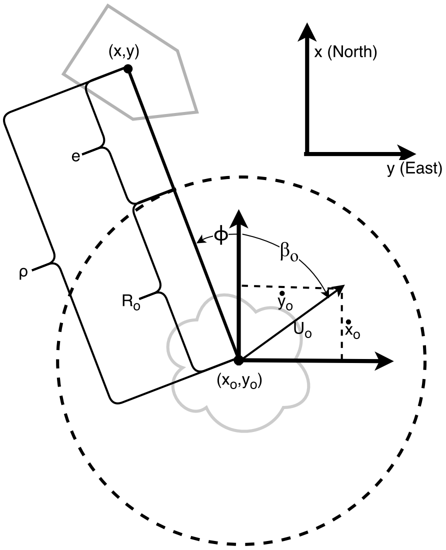


Figure 10.1: Illustration of parameters used for collision avoidance.

where ϕ is defined in (10.7), the derivative is given as

$$\begin{aligned}
\dot{\rho} &= \frac{2(x-x_\text{o})(\dot{x}-\dot{x}_\text{o}) + 2(y-y_\text{o})(\dot{y}-\dot{y}_\text{o})}{2\sqrt{(x-x_\text{o})^2 + (y-y_\text{o})^2}} \\
&= \frac{\rho\cos(\phi)(\dot{x}-\dot{x}_\text{o}) + \rho\sin(\phi)(\dot{y}-\dot{y}_\text{o})}{\rho} \\
&= u\cos(\phi-\psi) + v\sin(\phi-\psi) - \sqrt{\dot{x}_\text{o}^2+\dot{y}_\text{o}^2}\cos\left(\phi - \arctan\left(\frac{\dot{y}_\text{o}}{\dot{x}_\text{o}}\right)\right) \\
&= U\cos\left(\phi-\psi-\beta\right) - V_\text{o}. \qquad (10.23)
\end{aligned}$$

Under the conditions of Theorem 1, $u = u_\text{oa}$ (10.10) and $\psi = \psi_\text{oa}$ (10.11). It can be shown that this reduces the error dynamics to

$$\dot{e} = -\frac{U_\text{oa}}{\sqrt{\Delta^2+(e+k)^2}}e - \frac{U_\text{oa}}{\sqrt{\Delta^2+(e+k)^2}}k + V_\text{o}. \qquad (10.24)$$

Furthermore, given the proposed solution of k (10.13)-(10.14),

$$-\frac{U_\text{oa}}{\sqrt{\Delta^2+(e+k)^2}}k + V_\text{o} \equiv 0,$$

so

$$\dot{e} = -\frac{U_\text{oa}}{\sqrt{\Delta^2+(e+k)^2}}e. \qquad (10.25)$$

Using the positive definite Lyapunov function $V(e) = 0.5e^2$,

$$\dot{V} = -\frac{U_\text{oa}}{\sqrt{\Delta^2+(e+k)^2}}e^2 \qquad (10.26)$$

is negative definite and hence the equilibrium point $e = 0$ of (10.24) is UGAS [43]. □

10.2.3 Choosing λ

In the guidance law for collision avoidance (10.11), λ determines whether the USV circumvents an obstacle with a clockwise or counterclockwise motion. COLREGs provides definitions and rules to the various collision avoidance scenarios, see Figure 10.2 [49]. A formal classification is illustrated in Figure 10.3.

- Overtaking: Overtaking is allowed on either side of the obstacle as long as the USV keeps a sufficient distance to the obstacle. COLREGs defines overtaking as approaching another vessel at more than 22.5 degrees abaft her beam. This corresponds to the angle ω (illustrated in Figure 10.3) between the heading of the obstacle and the position of the USV is larger than 112.5° or less than −112.5°. This chapter suggests considering both clockwise and counter-clockwise motion by calculating ψ_{oa} (10.11) for both $\lambda = -1$ (counterclockwise) and $\lambda = 1$ (clockwise), and denoting this as $\psi_{\mathrm{oa,cc}}$ and $\psi_{\mathrm{oa,c}}$ respectively. The direction closest to the current heading ψ of the USV is then chosen, thereby avoiding a sharp turn. This is formalized as

$$\lambda = \begin{cases} -1 & |\psi - \psi_{\mathrm{oa,cc}}| \leq |\psi - \psi_{\mathrm{oa,c}}| \\ 1 & |\psi - \psi_{\mathrm{oa,cc}}| > |\psi - \psi_{\mathrm{oa,c}}|. \end{cases} \tag{10.27}$$

- Crossing from left: In this case, the USV has the right of way. Technically, no collision avoidance should be activated. However, an autonomous system should be able to avoid a collision even in the case that the other vehicle does not abide by the rules. Hence, this chapter suggests activating collision avoidance with $\lambda = -1$, corresponding to counterclockwise motion. Thus, if the obstacle complies with COLREGs, the USV's deviation of the path is minimized. Unlike overtaking, COLREGs does not specify a specific angle for this premise. Thus, this chapter defines this scenario as $\omega \in [\alpha, 112.5°)$ for some positive angle α. In all simulations, $\alpha = 15°$ [11].

- Crossing from right: In this situation, the USV is the give-way vessel and must avoid the obstacle by moving in a counterclockwise motion, indicating $\lambda = -1$. This scenario is defined as $\omega \in [-112.5°, -\alpha)$.

- Head-on: When two vessels meet head-on, both vehicles should alter course so that they will pass with the other vessel on their port side. This corresponds to a counterclockwise motion, therefore $\lambda = -1$. This case is activated when $\omega \in [-\alpha, \alpha)$.

Note that as the USV circumvents the obstacle and ω changes, the setting is not reclassified. Hence, as the set-based algorithm enters collision avoidance mode, it is determined what collision avoidance scenario is applicable and λ is determined as described above. This value for λ is kept until the next time the system enters obstacle mode.

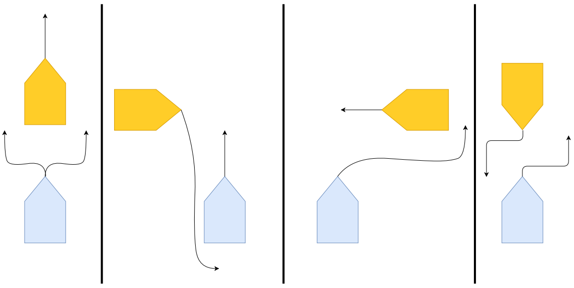


Figure 10.2: COLREGs scenarioes and the correct behavior of the involved vessels. USV in shown in blue and obstacle in orange. From left to right: Overtaking, crossing from left, crossing from right, head-on.

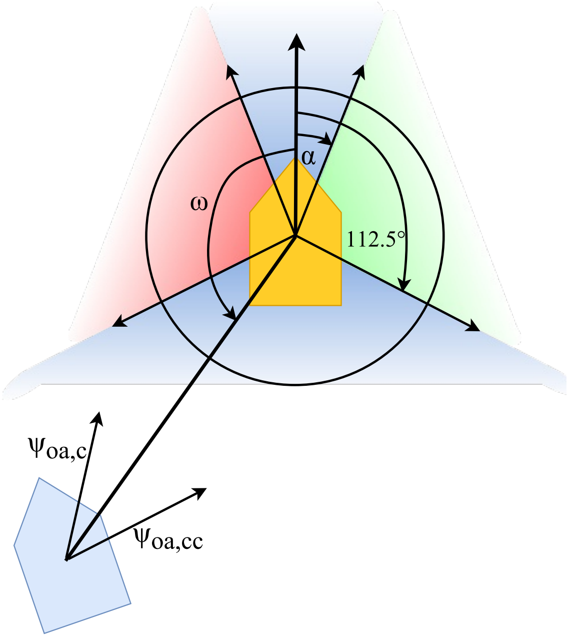


Figure 10.3: The different COLREGs scenarios as function of ω. From the top and clockwise: Head-on, crossing from left, overtaking, crossing from right. This specific illustration displays an overtaking situation, which is the only scenario where the required direction of the motion (clockwise or counterclockwise) is not strictly specified by COLREGs. Therefore, the motion most aligned with the current heading is chosen, in this case clockwise, given by $\psi_{\text{oa,c}}$.

10.2.4 Set-Based Guidance

The two previous subsections have presented guidance laws for surge velocity and heading for path following and collision avoidance. These two different scenarios are considered the two modes of the system, and we will in the following, based on the set-based control approach [66] presented in Chapter 4, develop a method for switching between these two. Note that this approach can be used with any combination of methods for path following and collision avoidance and is not limited to the collision avoidance approach presented in the previous subsection.

We define the task σ as the distance between an obstacle center and the USV, which is given by ρ (10.12):

$$\sigma = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{10.28}$$

The task derivative is defined in the proof of Theorem 10.1 as

$$\dot{\sigma} = U \cos(\phi - \psi - \beta) - V_\mathrm{o}. \tag{10.29}$$

Furthermore, we define a mode change radius $R_\mathrm{m} > R_\mathrm{o}$ around the obstacle.

Assumption 10.4: The radius R_m is chosen sufficiently large that in case of a switch to collision avoidance modus, the USV can converge to the radius R_o without overshoot. This will depend on the velocities of the USV and obstacle, the look-ahead distance Δ and the maximum turning radius of the USV.

Given our specified control objectives (10.1)-(10.2), the desired behavior of the USV is to follow the desired path C as long as this is possible while avoiding collisions. Path following is therefore considered the default mode. If the USV is outside the radius R_m, path following is always active. However, in agreement with the method in [66], we allow the path following mode to be active inside R_m under the condition that this will increase or maintain the current distance between the obstacle center and the USV, i.e. if $\dot{\sigma} \geq 0$ with $u = u_\mathrm{pf}$ and $\psi = \psi_\mathrm{pf}$. In other words, inside R_m collision avoidance is active as long as the desired behavior in path following mode will result in the USV decreasing the distance to the obstacle. In this case, the guidance laws for collision avoidance will ensure that the USV converges to the safe distance R_o from the obstacle center until such a time that the path following guidance law will take the USV further away from the obstacle. This switching behavior can be captured by the *tangent cone*. The tangent cone to the set $D = [\sigma_\mathrm{min}, \sigma_\mathrm{max}]$ at the point $\sigma \in D$ is defined as

$$T_D(\sigma) = \begin{cases} [\,0, \infty\,) & \sigma = \sigma_\mathrm{min} \\ \mathbb{R} & \sigma \in (\sigma_\mathrm{min}, \sigma_\mathrm{max}) \\ (\,-\infty, 0\,] & \sigma = \sigma_\mathrm{max} \end{cases}. \tag{10.30}$$

Note that $\dot{\sigma}(t) \in T_D(\sigma(t)) \; \forall \; t \geq t_0$ implies that $\sigma(t) \in \; D \; \forall \; t \geq t_0$. Thus, we can define a valid set D for our collision avoidance task and remain in path following mode as long as our collision avoidance task σ and the corresponding $\dot{\sigma}$ is in the tangent cone of D. We suggest defining

$$D = [\min(R_\mathrm{m}, \max(\sigma, R_\mathrm{o}))\,, \infty), \tag{10.31}$$

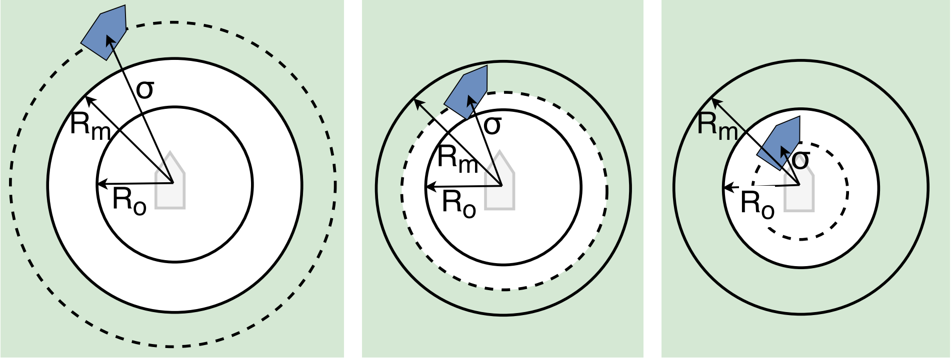


Figure 10.4: The set $D = [\min(R_\mathrm{m}, \max(\sigma, R_\mathrm{o})), \infty)$ used in Algorithm 10.1 illustrated in green. Outside R_m (left), the USV is always in the tangent cone of D and thus path following mode is active. For $R_\mathrm{o} \leq \sigma \leq R_\mathrm{m}$ (center), σ is always on the lower border of D by definition. Hence, path following is active only if the corresponding $\dot{\sigma} \geq 0$. For $\sigma < R_\mathrm{o}$ (right), the USV is outside the set D and the collision avoidance control objective is violated.

which is illustrated in Figure 10.4. As long as $\sigma \in D$, the collision avoidance objective (10.1) is satisfied. A practical implementation of the tangent cone is given in Algorithm 6.1.

Thus, we define our control algorithm as in Algorithm 10.1 where `in_T_RC` is the tangent cone function defined in Algorithm 6.1 and σ and $\dot{\sigma}$ are defined in (10.28) and (10.29) with $u = u_\mathrm{pf}$ and $\psi = \psi_\mathrm{pf}$. Note that in case of multiple obstacles, one has to consider one task per obstacle and check which obstacle, if any, is not in the tangent cone and thereby requires circumvention. The problem of overlapping obstacles is a topic for future work.

10.2.5 Surge and Yaw Controllers

This section presents surge and yaw controllers to ensure tracking of the desired surge velocity $u_\mathrm{des}(t)$ and heading $\psi_\mathrm{des}(t)$ provided by the set-based guidance system presented in the previous subsection. A feedback linearizing P-controller is used to ensure tracking of the desired relative surge velocity $u_\mathrm{des}(t)$:

$$\tau_u = -F_u(v, r) + \frac{d_{11}}{m_{11}} u_\mathrm{des} + \dot{u}_\mathrm{des} - k_u(u - u_\mathrm{des}) \tag{10.32}$$

The gain $k_u > 0$ is constant. Part of the damping is not canceled in order to guarantee some robustness with respect to model uncertainties. Similarly, a feedback linearizing PD-controller is used to track the desired yaw angle ψ_des. In this case $\dot{\psi}_\mathrm{des}(t)$ is calculated by taking the time derivative of $\psi_\mathrm{des}(t)$. Note that to prevent $\dot{\psi}_\mathrm{des}$ growing very large when switching between path following and collision avoidance, a smoothing function may be applied (see Section 10.4).

$$\tau_r = -F_r(u, v, r) + \ddot{\psi}_\mathrm{des} - k_\psi(\psi - \psi_\mathrm{des}) - k_r(\dot{\psi} - \dot{\psi}_\mathrm{des}), \tag{10.33}$$

where k_ψ and k_r are strictly positive constant controller gains.

Algorithm 10.1: Set-based guidance algorithm.

```
Initialize:
last_mode = path_following;
$\lambda = -1$;
while True do
    a = in_T_RC($\sigma$, $\dot{\sigma}$, min($R_\mathrm{m}$, max($\sigma$, $R_\mathrm{o}$)), $\infty$);
    if a is True then
        $u_\mathrm{des} = u_\mathrm{pf}$;
        $\psi_\mathrm{des} = \psi_\mathrm{pf}$;
        mode = path_following;
    else
        if last_mode is path_following then
            choose $\lambda$ in accordance with COLREGs
        end
        $u_\mathrm{des} = u_\mathrm{oa}$;
        $\psi_\mathrm{des} = \psi_\mathrm{oa}(\lambda)$;
        mode = obstacle_avoidance;
    end
    last_mode = mode
end
```

10.3 Main Result

This section presents the conditions under which the proposed control system achieves the control objectives (10.1)-(10.3).

Theorem 10.2. *Given an underactuated USV described by the dynamical system (2.18) with* $V_x = V_y = d^q_{11} = 0$. *If Assumption 2.1-2.3 and 10.1-10.4 hold, and the surge and yaw references provided by the set-based guidance system in Algorithm 10.1 are tracked, the control objective (10.1) is satisfied. Furthermore, as long as the system is in path following mode, the control objective (10.2) is also fulfilled.*

Proof. In collision avoidance mode, the guidance laws for surge and yaw (10.11) ensure that the distance between the USV and an obstacle at position $\boldsymbol{p}_\mathrm{o}(t)$, denoted ρ, converges to a constant value R_o (Theorem 1). Given Assumption 10.4, this mode is always activated at a distance large enough that $\rho \rightarrow R_\mathrm{o}$ without overshoot. Hence, we can apply the proof in Chapter 4 regarding satisfaction of set-based tasks with a valid set defined in (10.31). Thus, the first control objective is satisfied. Furthermore, the control objective (10.2) is fulfilled in path following mode by Assumption 10.1. ☐

Proposition 10.1. *Given an underactuated USV described by the dynamical system (2.18) with* $V_x = V_y = d^q_{11} = 0$. *If Assumption 2.1-2.3 and 10.1-10.4 hold, the* *controllers (10.32) and (10.33) ensure that the references provided by the set-based guidance system in Algorithm 10.1 are tracked, and the control objective (10.3) is satisfied.*

Proof. Define the following error signals:

$$\tilde{u} = u - u_{\text{des}}, \tag{10.34}$$

$$\tilde{\psi} = \psi - \psi_{\text{des}}, \tag{10.35}$$

$$\tilde{r} = r - \dot{\psi}_{\text{des}}. \tag{10.36}$$

The surge and yaw controllers (10.32) and (10.33) reduce the error dynamics to

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} \dot{\tilde{u}}_r \\ \dot{\tilde{\psi}} \\ \ddot{\tilde{\psi}} \end{bmatrix} = \begin{bmatrix} -(k_{u_r} + \frac{d_{11}}{m_{11}}) & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -k_{\psi} & -k_r \end{bmatrix} \begin{bmatrix} \tilde{u}_r \\ \tilde{\psi} \\ \dot{\tilde{\psi}} \end{bmatrix} = \boldsymbol{\Lambda}\boldsymbol{\xi}. \tag{10.37}$$

The system is linear and time-invariant. All controller gains and d_{11}/m_{11} are strictly positive, meaning that $\boldsymbol{\Lambda}$ is Hurwitz and the origin $\boldsymbol{\xi} = 0$ is UGES. Thus the error signals converge to zero and the control objective (10.3) is satisfied. ☐

10.4 Simulation Results: Straight Line Path

This section presents simulation results in the case where the desired path C is a straight line path, and where a traditional LOS guidance law is used for path following. Without loss of generality, we assume that the inertial frame has been rotated so the path is aligned with the inertial x-axis. Thus, the cross-track error of the path is given by the vehicle y-position and the LOS guidance law for path following is

$$\psi_{\text{pf}} = \arctan\left(-\frac{y}{\Delta}\right). \tag{10.38}$$

In the simulations, we use the vehicle model described by (2.18) with numeric values given in [35]. The desired surge velocity is chosen to be 4 m/s both during path following and obstacle avoidance, i.e. $u_{\text{pf}} = u_{\text{oa}} = 4$ m/s. The look-ahead distance Δ is chosen as $\Delta = 75$ m, which is in accordance with the condition given in [35] for stability of the path following guidance control system. Furthermore, the controller gains are chosen as $k_u = 0.1\ \text{s}^{-1}$, $k_{\psi} = 0.04\ \text{s}^{-2}$ and $k_r = 0.9\ \text{s}^{-1}$. Three obstacles moving at constant speed have been added to the simulation, see Table 10.1. Furthermore, a smoothing function has been implemented to prevent jumps in the desired heading when switching between path following and collision avoidance. Note that this smoothing function ensures a smooth transition over time between ψ_{pf} and ψ_{oa} when switching between modes, but is not active in deciding whether or not a switch in the set-based guidance occurs. Hence, it does not suffer from the same drawbacks as the potential field approach as discussed in Chapter 1. The implemented smoothing function has been chosen as

$$\alpha(t, t_{\text{switch}}) = (1/\pi)\arctan\left(5(t - t_{\text{switch}} - 2)\right) + (1/2). \tag{10.39}$$

Obstacle	**Details**
Obstacle 1	• $R_{\mathrm{o}} = 75$ m • $R_{\mathrm{m}} = 220$ m • Initial position: $\boldsymbol{p}_{\mathrm{o}}(0) = [300, 25]^T$ m • Velocity: $\dot{\boldsymbol{p}}_{\mathrm{c}}(t) = [-0.3, 0.0]^T$ m/s
Obstacle 2	• $R_{\mathrm{o}} = 100$ m • $R_{\mathrm{m}} = 250$ m • Initial position: $\boldsymbol{p}_{\mathrm{o}}(0) = [780, 125]^T$ m • Velocity: $\dot{\boldsymbol{p}}_{\mathrm{c}}(t) = [0.0, -0.5]^T$ m/s
Obstacle 3	• $R_{\mathrm{o}} = 100$ m • $R_{\mathrm{m}} = 200$ m • Initial position: $\boldsymbol{p}_{\mathrm{o}}(0) = [960, 50]^T$ m • Velocity: $\dot{\boldsymbol{p}}_{\mathrm{c}}(t) = [1.0, 0.0]^T$ m/s

Table 10.1: Table of implemented obstacles in simulation.

To test robustness, these simulations consider only obstacles that do not abide by COLREGs, meaning that they do nothing to prevent a collision with the USV.

The simulation results are illustrated in Fig 10.5. The vessel successfully avoids the obstacles (Figure 10.6) and converges to the desired straight line path when it is possible to do so. Figure 10.7 illustrates the path following cross-track error, which converges to zero in path following mode, and the desired and actual surge velocity and heading. The controllers (10.32) and (10.33) are able to track their references well. Thus, the control objectives (10.1)-(10.3) are satisfied.

10.5 Conclusions

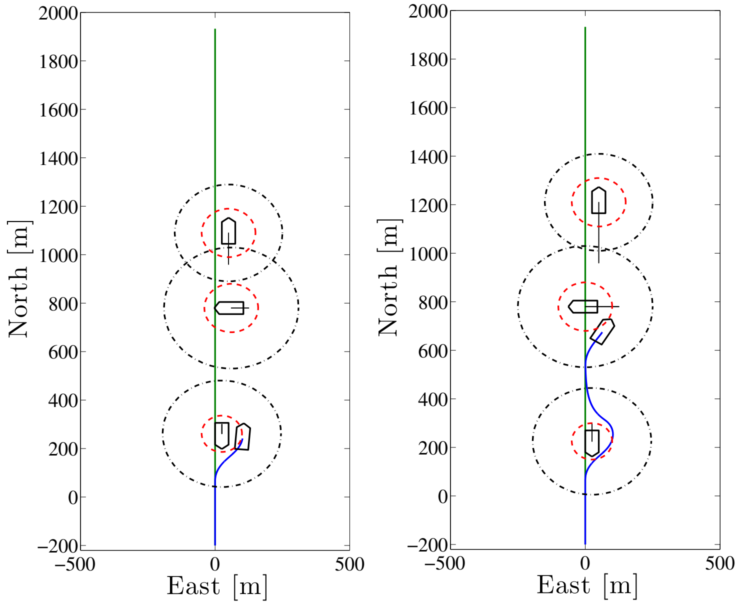
This chapter presents a guidance and control system for a USV that allows for collision avoidance of moving obstacles while following a predefined desired path. Based on recent results in set-based guidance theory, this chapter develops a collision avoidance method that take the underactuation of the USV into account. The proposed system switches between two guidance laws that, if satisfied, ensure path following and collision avoidance, respectively. Note that this approach can be used with any combination of methods for path following and collision avoidance.

Furthermore, this chapter has suggested a specific guidance law for the collision avoidance mode that will, if satisfied, make the USV track a circle with a constant safe radius about the moving obstacle center. The suggested guidance law for collision avoidance has a parameter $\lambda = \pm 1$ corresponding to clockwise and counterclockwise motion. A method for choosing λ has been presented to ensure that the USV abides by COLREGs while avoiding the obstacles.

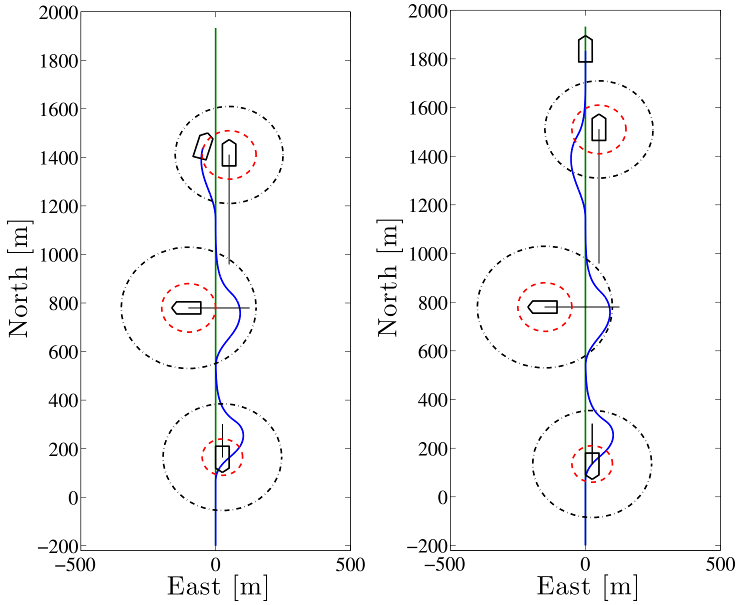
The set-based guidance system has been proved to prevent collisions given that certain, specified assumptions on the obstacle velocity are satisfied and the references provided by the guidance system are tracked. Furthermore, the proposed controllers ensure exponential tracking of the references.

Presented simulation results illustrate the effectiveness of the proposed method. To prove robustness, the simulated obstacles do not abide by COLREGs, and the USV still successfully circumvents them in a COLREGs manner and converges back to the desired path as soon as this is safe.

Future work includes expanding the results to the case of overlapping obstacles and experimental verification.



(a) Time = 130 s **(b)** Time = 250 s



(c) Time = 450 s **(d)** Time = 550 s

Figure 10.5: Trajectory for straight line path following with collision avoidance. Desired path in green, USV path in blue and the radii R_o and R_m of the obstacles in dashed red and black, respectively. In this simulation, the USV encounters three obstacles corresponding to a head-on, crossing from right and overtaking situation. All obstacles are successfully circumvented in a COLREGs manner and the USV converges back to the path as soon as this is safe.

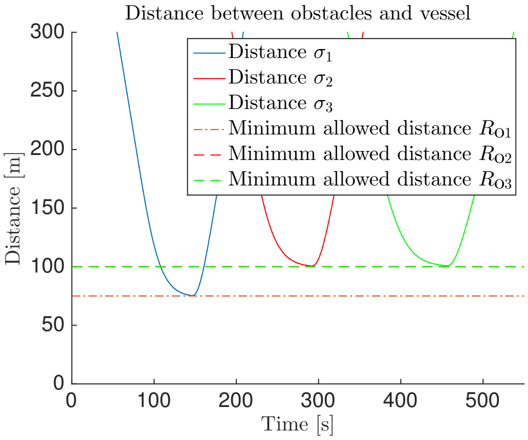


Figure 10.6: Distance between the USV and the three obstacles over time. The distance is always greater or equal to the minimum allowed safe distance R_o, confirming that control objective 1 is satisfied.

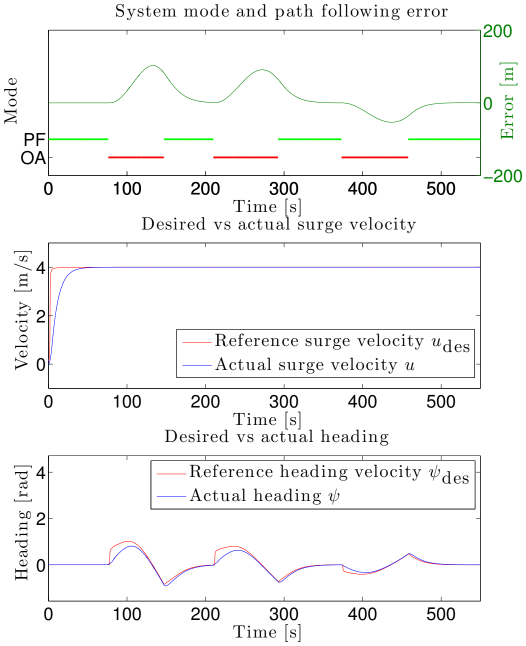


Figure 10.7: The path following error over time (top) and the reference vs. actual surge velocity (center) and heading (bottom). The UVS follows the path as long as this is possible without colliding and deviates from it when it is necessary to avoid obstacles. When the system returns to path following mode the USV converges back to the desired path. The controllers are able to track the references for u and ψ.

Chapter 11

Conclusions and Future Work

In this thesis, a detailed conclusion is given at the end of each chapter. This chapter presents a more general overview, conclusions and topics of future work.

In Part I of this thesis, an extension of the singularity-robust multiple task-priority inverse kinematics framework is presented to systematically handle set-based tasks in addition to equality tasks, for which the system has been developed. As a default solution, only the equality tasks are considered and implemented with the desired priority by projecting the low-priority task velocities through the null-spaces of the high-priority tasks. However, if this solution would result in a set-based task (e.g. a joint limit) being violated, this task is included in the task hierarchy with a certain priority with the goal of freezing the task at its current value. It is shown that set-based tasks given high priority, i.e. above the highest priority equality task, are satisfied at all times, but that the same cannot be guaranteed for low-priority set-based tasks. However, several tasks might be suitable as low-priority tasks in the case that satisfaction of said tasks is not crucial for the operation, such as field of view, manipulability, etc. Furthermore, by defining lower-priority set-based tasks, the overall system will ignore the task when it is satisfied, thereby resulting in more freedom to achieve the remaining tasks. Due to the switching between set-based tasks being active/inactive, the resulting closed-loop dynamic system can be described as a discontinuous differential equation. Using switched control systems theory it is proven that the equality task errors converge asymptotically to zero when including set-based tasks into the framework given that certain, specified conditions are satisfied. Should these conditions not be satisfied, the proposed system respects the given priority of the tasks (e.g. if the desired trajectory goes through an obstacle and the collision avoidance task has high priority, the system will deviate from the trajectory to avoid the obstacle). Simulations and experimental results are presented that validate the theory and illustrate the effectiveness of the method, in addition to a practical implementation of the proposed algorithm.

Due to the switched system, the proposed method may ask for very high joint accelerations when switching from one solution to another. Therefore, it is a topic of future work to develop a smooth switching method that also maintains the strict priority of the tasks. Furthermore, this highly generic method may be utilized in a number of applications, some of which may require certain adaptations. For collision avoidance purposes, closing the loop to also include obstacle detection would be highly relevant for both industrial purposes and marine applications. By applying set-based tasks for manipulability and singularity avoidance, the performance of manipulator systems may be enhanced, and set-based orientation control may be useful for spraypaint, welding, pick and place movements, etc. Further exploration of these possibilities will be conducted in the immediate future.

Part II presents guidance and control systems for curved path following of underactuated marine vehicles in the presence of unknown ocean currents. Different approaches are suggested based on the available sensors on the vehicle. In particular, the results of Chapter 7 and 8 are applicable when relative velocity measurements are obtainable, whereas Chapter 9 is based on absolute measurements only. Common for all chapters in Part II is that path following is achieved by allowing the vehicle to side-slip. This ensures that the total vehicle velocity is aligned with the desired path, even if the heading is not. When relative velocity measurements are avaliable, the ocean current must be estimated to achieve the correct side-slip angle. In the case of absolute velocity measurements, however, the total effect of the curvature of the path and the current is captured by the absolute velocity measurement and thus there is no need to estimate the current. For all chapters, convergence to the desired path is achieved with asymptotic stability properties under explicit conditions. For surface vehicles (Chapter 7 and 9) the results hold globally, whereas for underwater vehicles (Chapter 8) they hold for a defined domain (due to the well-known singularity of Euler angle representation).

Future work for this part consist of running experimental results to validate the correctness of the proposed systems, and to extend the results of Chapter 9 to the underwater case.

Finally, Part III combines Part I and II to make a switched guidance system for underactuated surface vehicles to achieve collision avoidance and path following. The proposed method is generic and may be used for any combination of methods for path following and collision avoidance. Furthermore, a specific guidance law for the collision avoidance mode if presented that will, if satisfied, make the USV track a circle with a constant safe radius about the moving obstacle center. The suggested guidance law for collision avoidance has a parameter $\lambda = \pm 1$ corresponding to clockwise and counterclockwise motion. A method for choosing λ is suggested to ensure that the surface vessel abides by COLREGs while avoiding the obstacles. The set-based guidance system is proven to prevent collisions given that certain, specified assumptions on the obstacle velocity are satisfied and the references provided by the guidance system are tracked. Furthermore, path following is achieved as long as the surface vessel can safely follow the path without colliding. Presented simulation results illustrate the effectiveness of the proposed method.

The results presented in Chapter 10 are valid for the case of obstacles that do not overlap, and for obstacles that do not themselves abide by COLREGs. The case of overlapping obstacles (e.g. numerous vehicles moving in a close formation) remains to be investigated, as does the case where all vessels are implemented with the proposed guidance system. Furthermore, although the developed theory is applicable also for curved path following, the initial simulation is for a straight line path. Finally, the proposed obstacle avoidance method should be expanded to ensure tracking of a safe radius about an obstacle center also in the case of ocean currents, and the results should be extended to the 3D case to be applicable for underwater vehicles and unmanned aerial vehicles.

Appendices

Appendix A

This appendix is related to Chapter 7 and describes in detail the step by step calculations to rewrite (7.14) into (7.15) given the expressions for the current observer (7.3), the update law (7.5), guidance law (7.6) and controllers (7.10)-(7.11). The equations are reprinted below for readability.

The ocean current observer is given as

$$\begin{aligned}
\dot{\hat{x}} &= \cos(\psi)u_r - \sin(\psi)v_r + \hat{V}_x + k_{x_1}\tilde{x}, \\
\dot{\hat{y}} &= \sin(\psi)u_r + \cos(\psi)v_r + \hat{V}_y + k_{y_1}\tilde{y}, \\
\dot{\hat{V}}_x &= k_{x_2}\tilde{x}, \\
\dot{\hat{V}}_y &= k_{y_2}\tilde{y},
\end{aligned} \tag{A.1}$$

where $\hat{x}$, $\hat{y}$, $\hat{V}_x$ and $\hat{V}_y$ are the estimates of x, y, V_x and V_y, and $\tilde{x} \triangleq x - \hat{x}$, $\tilde{y} \triangleq y - \hat{y}$, $\tilde{V}_x \triangleq V_x - \hat{V}_x$ and $\tilde{V}_y \triangleq V_y - \hat{V}_y$ are the estimation errors. The ocean current and the corresponding estimates in the $\{f\}$-frame are denoted

$$\begin{aligned}
V_x^f &= \cos(\psi_f)V_x + \sin(\psi_f)V_y, \\
V_y^f &= -\sin(\psi_f)V_x + \cos(\psi_f)V_y, \\
\hat{V}_x^f &= \cos(\psi_f)\hat{V}_x + \sin(\psi_f)\hat{V}_y, \\
\hat{V}_y^f &= -\sin(\psi_f)\hat{V}_x + \cos(\psi_f)\hat{V}_y,
\end{aligned} \tag{A.2}$$

and the update law is defined as

$$\dot{s} = \underbrace{\sqrt{u_{r,\text{des}}^2 + v_r^2}}_{\triangleq U_{\text{des}}} \frac{\sqrt{\Delta^2 + x_{b/f}^2} + x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} + \hat{V}_x^f. \tag{A.3}$$

Furthermore, the guidance law is given as

$$\psi_{\text{des}} = \psi_f - \arctan\left(\frac{v_r}{u_{r,\text{des}}}\right) - \arctan\left(\frac{y_{b/f} + g}{\sqrt{\Delta^2 + x_{b/f}^2}}\right), \tag{A.4}$$

where g is the solution to the second-order equation

$$\underbrace{(\hat{V}_y^{f^2} - U_{\text{des}}^2)}_{a} g^2 + 2 \underbrace{\hat{V}_y^{f^2} y_{b/f}}_{b} g + \underbrace{\hat{V}_y^{f^2} \left(\Delta^2 + x_{b/f}^2 + y_{b/f}^2\right)}_{c} = 0, \tag{A.5}$$

and $\Delta > 0$ is a design parameter. By Assumption 7.1-7.2, the solution(s) of (A.5) are real and finite. Equation (A.5) is a second order equation with parameters $(a, 2b, c)$ and thus it has two possible solutions:

$$\begin{aligned} g_1 &\triangleq \frac{-b - \sqrt{b^2 - ac}}{a}, \\ g_2 &\triangleq \frac{-b + \sqrt{b^2 - ac}}{a}. \end{aligned} \tag{A.6}$$

By simple analysis and Assumption 7.1-7.2, $a < 0$ and $c \geq 0$. Furthermore, since $\sqrt{b^2 - ac} \geq b$, g_1 will always be positive (≥ 0) and g_2 will always be negative (≤ 0). Thus, g is chosen as follows:

$$g = \begin{cases} g_1 & \hat{V}_y^f \geq 0 \\ g_2 & \hat{V}_y^f < 0 \end{cases} \tag{A.7}$$

Finally, the feedback linearizing controllers are given as

$$\tau_u = -F_{u_r}(v_r, r) + \frac{d_{11}}{m_{11}} u_{r,\text{des}} + \dot{u}_{r\text{des}} - k_{u_r}(u_r - u_{r,\text{des}}), \tag{A.8}$$

$$\tau_r = -F_r(u_r, v_r, r) + \ddot{\psi}_{\text{des}} - k_\psi(\psi - \psi_{\text{des}}) - k_r(\dot{\psi} - \dot{\psi}_{\text{des}}), \tag{A.9}$$

where the controller gains $k_{u_r} > 0$, k_ψ and k_r are constant and strictly positive, and the expressions for F_{u_r} and $F_r(u_r, v_r, r)$ are given in (2.19). The terms m_{11} and d_{11} are mass and damping coefficients as defined in (2.16). The controller errors are defined as $\tilde{u}_r \triangleq u_r - u_{r,\text{des}}$, $\tilde{\psi} \triangleq \psi - \psi_{\text{des}}$ and $\dot{\tilde{\psi}} \triangleq r - \dot{\psi}_{\text{des}}$.

Equation (7.14) describes the path error dynamics and is given as

$$\begin{aligned} \begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \end{bmatrix} = &\begin{bmatrix} \cos(\psi_{b/f}) & -\sin(\psi_{b/f}) \\ \sin(\psi_{b/f}) & \cos(\psi_{b/f}) \end{bmatrix} \begin{bmatrix} u_r \\ v_r \end{bmatrix} - \begin{bmatrix} \dot{s} \\ 0 \end{bmatrix} \\ &- \dot{s} \begin{bmatrix} 0 & -\kappa \\ \kappa & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \end{bmatrix} + \begin{bmatrix} \cos(\psi_f) & \sin(\psi_f) \\ -\sin(\psi_f) & \cos(\psi_f) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix}. \end{aligned} \tag{A.10}$$

Here, ψ_f denotes the heading of the Serret-Frenet reference frame and

$$\begin{aligned} \psi_{b/f} &= \psi - \psi_f \\ &= \tilde{\psi} + \psi_{\text{des}} - \psi_f. \end{aligned} \tag{A.11}$$

Inserting (A.4) into (A.11) yields

$$\begin{aligned}\psi_{b/f} &= \tilde{\psi} + \psi_f - \arctan\left(\frac{v_r}{u_{r,\text{des}}}\right) - \arctan\left(\frac{y_{b/f}+g}{\sqrt{\Delta^2 + x_{b/f}^2}}\right) - \psi_f \\ &= \tilde{\psi} \underbrace{- \arctan\left(\frac{v_r\sqrt{\Delta^2 + x_{b/f}^2} + u_{r,\text{des}}(y_{b/f}+g)}{u_{r,\text{des}}\sqrt{\Delta^2 + x_{b/f}^2} - v_r(y_{b/f}+g)}\right).}_{\triangleq \alpha}\end{aligned} \tag{A.12}$$

In the above equation, the trigonometric formula (2.6) has been used. Thus,

$$\begin{aligned}\cos(\psi_{b/f}) &= \cos(\tilde{\psi} - \alpha) \\ &= \cos(\tilde{\psi})\cos(\alpha) + \sin(\tilde{\psi})\sin(\alpha) \\ &= \cos(\alpha) + \cos(\alpha)\left(\cos(\tilde{\psi}) - 1\right) + \sin(\tilde{\psi})\sin(\alpha) \\ &= \cos(\alpha) + \tilde{\psi}\left[\frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}}\cos(\alpha) + \frac{\sin(\tilde{\psi})}{\tilde{\psi}}\sin(\alpha)\right],\end{aligned} \tag{A.13}$$

and

$$\begin{aligned}\sin(\psi_{b/f}) &= \sin(\tilde{\psi} - \alpha) \\ &= \sin(\tilde{\psi})\cos(\alpha) - \cos(\tilde{\psi})\sin(\alpha) \\ &= -\sin(\alpha) - \sin(\alpha)\left(\cos(\tilde{\psi}) - 1\right) + \sin(\tilde{\psi})\cos(\alpha) \\ &= -\sin(\alpha) + \tilde{\psi}\left[\frac{\sin(\tilde{\psi})}{\tilde{\psi}}\cos(\alpha) - \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}}\sin(\alpha)\right].\end{aligned} \tag{A.14}$$

Using the formulas stated in (2.7), the following relationship is clear:

$$\begin{aligned}\cos(\alpha) &= \frac{u_{r,\text{des}}\sqrt{\Delta^2 + x_{b/f}^2} - v_r(y_{b/f}+g)}{U_{\text{des}}\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}}, \\ \sin(\alpha) &= \frac{v_r\sqrt{\Delta^2 + x_{b/f}^2} + u_{r,\text{des}}(y_{b/f}+g)}{U_{\text{des}}\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}}.\end{aligned} \tag{A.15}$$

This can be inserted into (A.10).

$$\begin{aligned}
\dot{x}_{b/f} &= \cos(\psi_{b/f})u_r - \sin(\psi_{b/f})v_r - \dot{s} + \dot{s}\kappa y_{b/f} + \cos(\psi_f)V_x + \sin(\psi_f)V_y \\
&= \cos(\psi_{b/f})(\tilde{u_r} + u_{r,\text{des}}) - \sin(\psi_{b/f})v_r - \dot{s} + \dot{s}\kappa y_{b/f} + \cos(\psi_f)V_x + \sin(\psi_f)V_y \\
&= u_{r,\text{des}}\cos(\alpha) + v_r\sin(\alpha) - \dot{s} + \dot{s}\kappa y_{b/f} + \cos(\psi_f)V_x + \sin(\psi_f)V_y + \cos(\psi_{b/f})\tilde{u_r} \\
&\quad + \tilde{\psi}\left[\frac{\cos(\tilde{\psi})-1}{\tilde{\psi}}\left(u_{r,\text{des}}\cos(\alpha) + v_r\sin(\alpha)\right) + \frac{\sin(\tilde{\psi})}{\tilde{\psi}}\left(u_{r,\text{des}}\sin(\alpha) - v_r\cos(\alpha)\right)\right] \\
&= U_{\text{des}}\frac{\sqrt{\Delta^2 + x_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} - \dot{s} + \dot{s}\kappa y_{b/f} + \cos(\psi_f)V_x + \sin(\psi_f)V_y + \cos(\psi_{b/f})\tilde{u_r} \\
&\quad + \tilde{\psi}\left[\frac{\cos(\tilde{\psi})-1}{\tilde{\psi}}\frac{U_{\text{des}}\sqrt{\Delta^2 + x_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} + \frac{\sin(\tilde{\psi})}{\tilde{\psi}}\frac{U_{\text{des}}(y_{b/f}+g)}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}}\right].
\end{aligned} \tag{A.16}$$

Inserting for $\dot{s}$ defined in (A.3), (A.16) reduces to

$$\begin{aligned}
\dot{x}_{b/f} &= -U_{\text{des}}\frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} + \dot{s}\kappa y_{b/f} + \cos(\psi_{b/f})\tilde{u_r} + \cos(\psi_f)\tilde{V_x} + \sin(\psi_f)\tilde{V_y} \\
&\quad \underbrace{+ \tilde{\psi}\left[\frac{\cos(\tilde{\psi})-1}{\tilde{\psi}}\frac{U_{\text{des}}\sqrt{\Delta^2 + x_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} + \frac{\sin(\tilde{\psi})}{\tilde{\psi}}\frac{U_{\text{des}}(y_{b/f}+g)}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}}\right]}_{\triangleq h_1(t, U_{\text{des}}, \boldsymbol{\xi})}.
\end{aligned} \tag{A.17}$$

Similarily,

$$\begin{aligned}
\dot{y}_{b/f} &= \sin(\psi_{b/f})u_r + \cos(\psi_{b/f})v_r - \dot{s}\kappa x_{b/f} - \sin(\psi_f)V_x + \cos(\psi_f)V_y \\
&= \sin(\psi_{b/f})(\tilde{u_r} + u_{r,\text{des}}) + \cos(\psi_{b/f})v_r - \dot{s}\kappa x_{b/f} - \sin(\psi_f)V_x + \cos(\psi_f)V_y \\
&= v_r\cos(\alpha) - u_{r,\text{des}}\sin(\alpha) - \dot{s}\kappa x_{b/f} - \sin(\psi_f)V_x + \cos(\psi_f)V_y + \sin(\psi_{b/f})\tilde{u_r} \\
&\quad + \tilde{\psi}\left[\frac{\cos(\tilde{\psi})-1}{\tilde{\psi}}\left(v_r\cos(\alpha) - u_{r,\text{des}}\sin(\alpha)\right) + \frac{\sin(\tilde{\psi})}{\tilde{\psi}}\left(u_{r,\text{des}}\cos(\alpha) + v_r\sin(\alpha)\right)\right] \\
&= -U_{\text{des}}\frac{y_{b/f}+g}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} - \dot{s}\kappa x_{b/f} - \sin(\psi_f)V_x + \cos(\psi_f)V_y + \sin(\psi_{b/f})\tilde{u_r} \\
&\quad + \tilde{\psi}\left[\frac{\sin(\tilde{\psi})}{\tilde{\psi}}\frac{U_{\text{des}}\sqrt{\Delta^2 + x_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} - \frac{\cos(\tilde{\psi})-1}{\tilde{\psi}}\frac{U_{\text{des}}(y_{b/f}+g)}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}}\right]
\end{aligned} \tag{A.18}$$

Note that the term g is chosen so that

$$\begin{aligned}
&-U_{\text{des}}\frac{g}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} + \hat{V}_y^f \\
&\qquad = -U_{\text{des}}\frac{g}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} - \sin(\psi_f)\hat{V}_x + \cos(\psi_f)\hat{V}_y = 0.
\end{aligned} \tag{A.19}$$

Solving the above equation with respect to g leads to the second-order equation (A.5), which has been proven to have real solutions, $g_1 \geq 0$ and $g_2 \leq 0$. Thus, by choosing g as in (A.7), (A.18) reduces to

$$\begin{aligned}\dot{y}_{b/f} = &-U_{\text{des}} \frac{y_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} - \dot{s}\kappa x_{b/f} + \sin(\psi_{b/f})\tilde{u_r} - \sin(\psi_f)\tilde{V}_x + \cos(\psi_f)\tilde{V}_y \\ &+ \tilde{\psi} \underbrace{\left[\frac{\sin(\tilde{\psi})}{\tilde{\psi}} \frac{\sqrt{U_{\text{des}}\Delta^2 + x_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} - \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}} \frac{U_{\text{des}}(y_{b/f} + g)}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2}} \right]}_{\triangleq h_2(t, U_{\text{des}}, \boldsymbol{\xi})}. \end{aligned} \tag{A.20}$$

This can be expressed as (7.15), thus concluding this appendix:

$$\begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \end{bmatrix} = \begin{bmatrix} -U_{\text{des}} \frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} \\ -U_{\text{des}} \frac{y_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f}+g)^2}} \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa \\ \kappa & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \end{bmatrix} + \boldsymbol{H}(t, U_{\text{des}}, \boldsymbol{\xi})\boldsymbol{\xi} \tag{A.21}$$

$$\boldsymbol{H}(t, U_{\text{des}}, \boldsymbol{\xi}) = \begin{bmatrix} \cos(\psi_{b/f}) & h_1(t, U_{\text{des}}, \boldsymbol{\xi}) & 0 & \cos(\psi_f) & \sin(\psi_f) & 0 & 0 \\ \sin(\psi_{b/f}) & h_2(t, U_{\text{des}}, \boldsymbol{\xi}) & 0 & -\sin(\psi_f) & \cos(\psi_f) & 0 & 0 \end{bmatrix} \tag{A.22}$$

$$\boldsymbol{\xi} = \begin{bmatrix} \tilde{u_r} & \tilde{\psi} & \dot{\tilde{\psi}} & \tilde{V}_x & \tilde{V}_y & \tilde{x} & \tilde{y} \end{bmatrix}^T \tag{A.23}$$

Appendix B

This appendix is related to Chapter 8 and describes in detail the step by step calculations to derive the expression (8.33) for the controller error dynamics given the controllers (8.22)-(8.25) with the AUV model (2.36). The equations are reprinted below for readability.

The AUV model in component form is defined as

$$\begin{aligned}
\dot{x} &= \cos(\theta)\cos(\psi)u_r - \sin(\psi)v_r + \sin(\theta)\cos(\psi)w_r + V_x, \\
\dot{y} &= \cos(\theta)\sin(\psi)u_r + \cos(\psi)v_r + \sin(\theta)\sin(\psi)w_r + V_y, \\
\dot{z} &= -\sin(\theta)u_r + \cos(\theta)w_r + V_z, \\
\dot{\theta} &= q, \\
\dot{\psi} &= \frac{r}{\cos(\theta)}, \\
\dot{u}_r &= F_{u_r}(u_r, v_r, w_r, r, q) + \tau_u, \\
\dot{v}_r &= X_{v_r}(u_r)r + Y_{v_r}(u_r)v_r, \\
\dot{w}_r &= X_{w_r}(u_r)q + Y_{w_r}(u_r)w_r + Z_{w_r}\sin(\theta), \\
\dot{q} &= F_q(\theta, u_r, w_r, q) + \tau_q, \\
\dot{r} &= F_r(u_r, v_r, r) + \tau_r,
\end{aligned} \tag{B.1}$$

and the controllers are given below.

$$\tau_u = -F_{u_r}(u_r, v_r, w_r, r, q) + \dot{u}_{r,\text{des}} - k_{u_r}(u_r - u_{r,\text{des}}) \tag{B.2}$$

The gain $k_{u_r} > 0$ is constant and $F_{u_r}(u_r, v_r, r, q)$ is defined in (2.37). The controller error is denoted $\tilde{u}_r \triangleq u_r - u_{r,\text{des}}$. Thus,

$$\begin{aligned}
\dot{\tilde{u}}_r &= \dot{u}_r - \dot{u}_{r,\text{des}} \\
&= F_{u_r}(u_r, v_r, w_r, r, q) + \tau_u - \dot{u}_{r,\text{des}} \\
&= F_{u_r}(u_r, v_r, w_r, r, q) - F_{u_r}(u_r, v_r, w_r, r, q) + \dot{u}_{r,\text{des}} - k_{u_r}(u_r - u_{r,\text{des}}) - \dot{u}_{r,\text{des}} \\
&= -k_{u_r}\tilde{u}_r.
\end{aligned} \tag{B.3}$$

In addition to $u_r(t)$, the controlled states are the relative orientation $\theta_{fc}(t)$, $\psi_{fc}(t)$ and the rotational velocities q and r. The rudder controller is an integrator backstepping controller that ensures that the difference between the actual and desired states converge to zero. The errors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are defined as

$$\boldsymbol{z}_1 \triangleq \begin{bmatrix} \tilde{\theta}_{fc} \\ \tilde{\psi}_{fc} \end{bmatrix} \triangleq \begin{bmatrix} \theta_{fc} \\ \psi_{fc} \end{bmatrix} - \begin{bmatrix} \theta_{fc,\mathrm{des}} \\ \psi_{fc,\mathrm{des}} \end{bmatrix}, \tag{B.4}$$

$$\boldsymbol{z}_2 \triangleq \begin{bmatrix} \tilde{q} \\ \tilde{r} \end{bmatrix} \triangleq \begin{bmatrix} q \\ r \end{bmatrix} - \begin{bmatrix} q_{\mathrm{des}} \\ r_{\mathrm{des}} \end{bmatrix}. \tag{B.5}$$

The controller is then given as

$$\begin{bmatrix} \tau_q \\ \tau_r \end{bmatrix} = \begin{bmatrix} \dot{q}_{\mathrm{des}} \\ \dot{r}_{\mathrm{des}} \end{bmatrix} - \begin{bmatrix} F_q(\theta, u_r, w_r, q) \\ F_r(u_r, v_r, r) \end{bmatrix} - \boldsymbol{A}^T \boldsymbol{z}_1 - \boldsymbol{K}_2 \boldsymbol{z}_2, \tag{B.6}$$

where

$$\boldsymbol{A} \triangleq \begin{bmatrix} \cos(\phi_{fc}) & -\sin(\phi_{fc}) \\ \frac{\sin(\phi_{fc})}{\cos(\theta_{fc})} & \frac{\cos(\phi_{fc})}{\cos(\theta_{fc})} \end{bmatrix} \begin{bmatrix} \cos(\beta_c) & -\sin(\beta_c)\sin(\alpha_c) \\ 0 & \cos(\alpha_c) \end{bmatrix}, \tag{B.7}$$

$$\begin{bmatrix} q_{\mathrm{des}} \\ r_{\mathrm{des}} \end{bmatrix} \triangleq -\boldsymbol{A}^{-1}(\boldsymbol{\Phi} + \boldsymbol{K}_1 \boldsymbol{z}_1), \tag{B.8}$$

$$\boldsymbol{\Phi} = -\begin{bmatrix} \dot{\theta}_{fc,\mathrm{des}} \\ \dot{\psi}_{fc,\mathrm{des}} \end{bmatrix} + \begin{bmatrix} \cos(\phi_{fc}) & -\sin(\phi_{fc}) \\ \frac{\sin(\phi_{fc})}{\cos(\theta_{fc})} & \frac{\cos(\phi_{fc})}{\cos(\theta_{fc})} \end{bmatrix} \boldsymbol{L}(\boldsymbol{\omega}_{bc}^c - [\boldsymbol{R}_c^f(\boldsymbol{\Theta}_{fc})]^T \boldsymbol{\omega}_{if}^f), \tag{B.9}$$

$$\boldsymbol{L} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{B.10}$$

$\boldsymbol{\omega}_{bc}^c = [-\dot{\beta}_c \sin(\alpha_c), -\dot{\alpha}_c, \dot{\beta}_c \cos(\alpha_c)]^T$, $\boldsymbol{\omega}_{if}^f = \dot{s}[\tau(s), 0, \kappa(s)]^T$ and $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are positive definite, symmetric gain matrices. Furthermore, $\tau(s)$ and $\kappa(s)$ are the path torsion and curvature, and the functions $F_q(\theta, u_r, w_r, q)$ and $F_r(u_r, v_r, r)$ are defined in (2.43)-(2.44). Furthermore,

$$\alpha_c \triangleq \arctan\left(\frac{w_r}{u_{r,\mathrm{des}}}\right), \tag{B.11}$$

$$\beta_c \triangleq \arctan\left(\frac{v_r}{\sqrt{u_{r,\mathrm{des}}^2 + w_r^2}}\right), \tag{B.12}$$

$$\boldsymbol{\Theta}_{cb} = \begin{bmatrix} 0 \\ \alpha_c \\ -\beta_c \end{bmatrix}. \tag{B.13}$$

The time-derivative of $\boldsymbol{z}_1$ is given as

$$\dot{\boldsymbol{z}}_1 = \begin{bmatrix} \dot{\theta}_{fc} \\ \dot{\psi}_{fc} \end{bmatrix} - \begin{bmatrix} \dot{\theta}_{fc,\mathrm{des}} \\ \dot{\psi}_{fc,\mathrm{des}} \end{bmatrix}. \tag{B.14}$$

Using standard rules for differntial kinematics from [31],

$$\begin{bmatrix} \dot{\phi}_{fc} \\ \dot{\theta}_{fc} \\ \dot{\psi}_{fc} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi_{fc})\tan(\theta_{fc}) & \cos(\phi_{fc})\tan(\theta_{fc}) \\ 0 & \cos(\phi_{fc}) & -\sin(\phi_{fc}) \\ 0 & \sin(\phi_{fc})/\cos(\theta_{fc}) & \cos(\phi_{fc})/\cos(\theta_{fc}) \end{bmatrix} \boldsymbol{\omega}_{fc}^c. \tag{B.15}$$

Here $\boldsymbol{\omega}^c_{fc}$ denotes the angular velocity of frame $\{c\}$ relative to $\{f\}$ expressed in $\{c\}$, and can be decomposed as

$$\boldsymbol{\omega}^c_{fc} = \boldsymbol{\omega}^c_{ic} - \boldsymbol{\omega}^c_{if} = \boldsymbol{\omega}^c_{ib} + \boldsymbol{\omega}^c_{bc} - \boldsymbol{\omega}^c_{if} = \boldsymbol{R}^c_b(\boldsymbol{\Theta}_{cb})\boldsymbol{\omega}^b_{ib} + \boldsymbol{\omega}^c_{bc} - \boldsymbol{R}^c_f(\boldsymbol{\Theta}_{fc})\boldsymbol{\omega}^f_{if}. \tag{B.16}$$

Here, $\boldsymbol{\omega}^b_{ib}$ is the rotational velocity of the body relative to the inertial frame and is given by the AUV states $\boldsymbol{\omega}^b_{ib} = [0, q, r]^T$, $\boldsymbol{\omega}^c_{bc} = -\boldsymbol{\omega}^c_{cb} = [-\dot{\beta}_c \sin(\alpha_c), -\dot{\alpha}_c, \dot{\beta}_c \cos(\alpha_c)]^T$ through a similar transformation as (B.15) and $\boldsymbol{\omega}^f_{if} = \dot{s}[\tau(s), 0, \kappa(s)]^T$ by definition of the Serret-Frenet reference frame [26]. Thus,

$$\begin{aligned}
\dot{\boldsymbol{z}}_1 &= \underbrace{\begin{bmatrix} \cos(\phi_{fc}) & -\sin(\phi_{fc}) \\ \sin(\phi_{fc})/\cos(\theta_{fc}) & \cos(\phi_{fc})/\cos(\theta_{fc}) \end{bmatrix}}_{\triangleq \boldsymbol{B}_{fc}} \boldsymbol{L}\boldsymbol{\omega}^c_{fc} - \begin{bmatrix} \dot{\theta}_{fc,\text{des}} \\ \dot{\psi}_{fc,\text{des}} \end{bmatrix}, \\
&= \boldsymbol{B}_{fc}\boldsymbol{L}\boldsymbol{\omega}^c_{fc} - \begin{bmatrix} \dot{\theta}_{fc,\text{des}} \\ \dot{\psi}_{fc,\text{des}} \end{bmatrix} \\
&= \boldsymbol{B}_{fc}\boldsymbol{L}(\boldsymbol{R}^c_b(\boldsymbol{\Theta}_{cb})\boldsymbol{\omega}^b_{ib} + \boldsymbol{\omega}^c_{bc} - \boldsymbol{R}^c_f(\boldsymbol{\Theta}_{fc})\boldsymbol{\omega}^f_{if}) - \begin{bmatrix} \dot{\theta}_{fc,\text{des}} \\ \dot{\psi}_{fc,\text{des}} \end{bmatrix} \\
&= \boldsymbol{B}_{fc}\boldsymbol{L}\boldsymbol{R}^c_b(\boldsymbol{\Theta}_{cb}) \begin{bmatrix} 0 \\ q \\ r \end{bmatrix} - \begin{bmatrix} \dot{\theta}_{fc,\text{des}} \\ \dot{\psi}_{fc,\text{des}} \end{bmatrix} + \boldsymbol{B}_{fc}\boldsymbol{L}(\boldsymbol{\omega}^c_{bc} - \boldsymbol{R}^c_f(\boldsymbol{\Theta}_{fc})\boldsymbol{\omega}^f_{if}) \\
&= \boldsymbol{B}_{fc} \begin{bmatrix} \cos(\beta_c) & -\sin(\beta_c)\sin(\alpha_c) \\ 0 & \cos(\alpha_c) \end{bmatrix} \begin{bmatrix} q \\ r \end{bmatrix} - \begin{bmatrix} \dot{\theta}_{fc,\text{des}} \\ \dot{\psi}_{fc,\text{des}} \end{bmatrix} + \boldsymbol{B}_{fc}\boldsymbol{L}(\boldsymbol{\omega}^c_{bc} - \boldsymbol{R}^c_f(\boldsymbol{\Theta}_{fc})\boldsymbol{\omega}^f_{if}) \\
&= \boldsymbol{A} \begin{bmatrix} q \\ r \end{bmatrix} + \boldsymbol{\Phi},
\end{aligned} \tag{B.17}$$

where $\boldsymbol{A}$, $\boldsymbol{\Phi}$ and $\boldsymbol{L}$ are defined in (B.7), (B.9) and (B.10), respectively. Furthermore, by inserting (B.5) and (B.8), the $\boldsymbol{z}_1$ dynamics can be further rewritten into

$$\begin{aligned}
\dot{\boldsymbol{z}}_1 &= \boldsymbol{A}(\boldsymbol{z}_2 + \begin{bmatrix} q_\text{des} \\ r_\text{des} \end{bmatrix}) + \boldsymbol{\Phi} \\
&= \boldsymbol{A}\boldsymbol{z}_2 + \boldsymbol{A}\left(-\boldsymbol{A}^{-1}(\boldsymbol{\Phi} + \boldsymbol{K}_1\boldsymbol{z}_1)\right) + \boldsymbol{\Phi} \\
&= \boldsymbol{A}\boldsymbol{z}_2 - \boldsymbol{K}_1\boldsymbol{z}_1.
\end{aligned} \tag{B.18}$$

Similarly,

$$\begin{aligned}
\dot{\boldsymbol{z}}_2 &= \begin{bmatrix} \dot{q} \\ \dot{r} \end{bmatrix} - \begin{bmatrix} \dot{q}_\text{des} \\ \dot{r}_\text{des} \end{bmatrix} \\
&= \begin{bmatrix} F_q(\theta, u_r, w_r, q) \\ F_r(u_r, v_r, r) \end{bmatrix} + \begin{bmatrix} \tau_q \\ \tau_r \end{bmatrix} - \begin{bmatrix} \dot{q}_\text{des} \\ \dot{r}_\text{des} \end{bmatrix} \\
&= \begin{bmatrix} F_q(\theta, u_r, w_r, q) \\ F_r(u_r, v_r, r) \end{bmatrix} + \begin{bmatrix} \dot{q}_\text{des} \\ \dot{r}_\text{des} \end{bmatrix} - \begin{bmatrix} F_q(\theta, u_r, w_r, q) \\ F_r(u_r, v_r, r) \end{bmatrix} - \boldsymbol{A}^T\boldsymbol{z}_1 - \boldsymbol{K}_2\boldsymbol{z}_2 - \begin{bmatrix} \dot{q}_\text{des} \\ \dot{r}_\text{des} \end{bmatrix} \\
&= -\boldsymbol{A}^T\boldsymbol{z}_1 - \boldsymbol{K}_2\boldsymbol{z}_2.
\end{aligned} \tag{B.19}$$

when inserting (B.1) and (B.6) into the time derivative of the definition (B.5). Thus, the error dynamics for $\boldsymbol{\xi}_1 = \begin{bmatrix} \tilde{u}_r & \boldsymbol{z}_1^T & \boldsymbol{z}_2^T \end{bmatrix}^T$ can be expressed as in (8.33), Q.E.D.

$$\dot{\boldsymbol{\xi}}_1 = \begin{bmatrix} -k_{u_r} & \mathbf{0} & \mathbf{0} \\ 0 & -\boldsymbol{K}_1 & \boldsymbol{A} \\ 0 & -\boldsymbol{A}^T & -\boldsymbol{K}_2 \end{bmatrix} \boldsymbol{\xi}_1 \tag{B.20}$$

Appendix C

This appendix is related to Chapter 8 and describes in detail the step by step calculations to rewrite (8.37) into (8.38) given the expressions for the current observer (8.10), the update law (8.13) and guidance laws (8.14)-(8.15). The equations are reprinted below for readability.

The ocean current observer is given as

$$\begin{aligned}
\dot{\hat{x}} &= \cos(\theta)\cos(\psi)u_r - \sin(\psi)v_r + \sin(\theta)\cos(\psi)w_r + \hat{V}_x + k_{x_1}\tilde{x}, \\
\dot{\hat{y}} &= \cos(\theta)\sin(\psi)u_r + \cos(\psi)v_r + \sin(\theta)\sin(\psi)w_r + \hat{V}_y + k_{y_1}\tilde{y}, \\
\dot{\hat{z}} &= -\sin(\theta)u_r + \cos(\theta)w_r + \hat{V}_z + k_{z_1}\tilde{z}, \\
\dot{\hat{V}}_x &= k_{x_2}\tilde{x}, \\
\dot{\hat{V}}_y &= k_{y_2}\tilde{y}, \\
\dot{\hat{V}}_z &= k_{z_2}\tilde{z},
\end{aligned} \tag{C.1}$$

where $\hat{x}$, $\hat{y}$, $\hat{z}$, $\hat{V}_x$, $\hat{V}_y$ and $\hat{V}_z$ are the estimates of x, y, z, V_x, V_y and V_z, respectively. Furthermore, $\tilde{x} = x - \hat{x}$, $\tilde{y} = y - \hat{y}$, $\tilde{z} = z - \hat{z}$, $\tilde{V}_x = V_x - \hat{V}_x$, $\tilde{V}_y = V_y - \hat{V}_y$ and $\tilde{V}_z = V_z - \hat{V}_z$. are the observer error signals. The ocean current and the corresponding estimates in the $\{f\}$-frame are denoted

$$\boldsymbol{V}_c^f = \begin{bmatrix} V_x^f \\ V_y^f \\ V_z^f \end{bmatrix} = \boldsymbol{R}_f^i(\boldsymbol{\Theta}_{if})^T \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}, \tag{C.2}$$

$$\hat{\boldsymbol{V}}_c^f = \begin{bmatrix} \hat{V}_x \\ \hat{V}_y \\ \hat{V}_z \end{bmatrix} = \boldsymbol{R}_f^i(\boldsymbol{\Theta}_{if})^T \begin{bmatrix} \hat{V}_x^f \\ \hat{V}_y^f \\ \hat{V}_z^f \end{bmatrix}. \tag{C.3}$$

and the update law is defined as

$$\begin{aligned}
\dot{s} = {} & \underbrace{\sqrt{u_{r,\text{des}}^2 + v_r^2 + w_r^2}}_{\triangleq U_{\text{des}}} \frac{\sqrt{\Delta^2 + y_{b/f}^2}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f} + f)^2}} \frac{\sqrt{\Delta^2 + x_{b/f}^2 + z_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2 + z_{b/f}^2}} \\
& + U_{\text{des}} \frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2}} + \hat{V}_x^f
\end{aligned} \tag{C.4}$$

Furthermore, the guidance laws are given as

$$\theta_{fc,\text{des}} = \arctan\left(\frac{z_{b/f} + f}{\sqrt{\Delta^2 + y_{b/f}^2}}\right) \tag{C.5}$$

$$\psi_{fc,\text{des}} = -\arctan\left(\frac{y_{b/f} + g}{\sqrt{\Delta^2 + x_{b/f}^2 + z_{b/f}^2}}\right), \tag{C.6}$$

where f is the solution to the second order equation

$$\underbrace{\left(\hat{V}_z^{f^2} - U_{\text{des}}^2\right)}_{a_f} f^2 + 2\underbrace{\hat{V}_z^{f^2} z_{b/f}}_{b_f} f + \underbrace{\hat{V}_z^{f^2}\left(\Delta^2 + y_{b/f}^2 + z_{b/f}^2\right)}_{c_f} = 0, \tag{C.7}$$

and g is the solution to the second order equation

$$\underbrace{\left(\hat{V}_y^{f^2} - \cos^2(\theta_{fc,\text{des}})U_{\text{des}}^2\right)}_{a_g} g^2 + 2\underbrace{\hat{V}_y^{f^2} y_{b/f}}_{b_g} g + \underbrace{\hat{V}_y^{f^2}\left(\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2\right)}_{c_g} = 0. \tag{C.8}$$

Here, $\Delta > 0$ is a design parameter. In Chapter 8, the analysis is restricted to a domain $\mathbb{D}$ where $\theta_{fc,\text{des}}$ is bounded by a parameter $0 < k < 1$ (see Assumption 8.1):

$$\mathbb{D} = \left\{(\theta, \theta_{fc}, \theta_{fc,\text{des}}) \in \mathbb{R}^3 : |\theta| < \pi/2, |\theta_{fc}| < \pi/2, |\theta_{fc,\text{des}}| < \arccos(k)\right\} \tag{C.9}$$

Assumption 8.1-8.2 and $|\theta_{fc,\text{des}}| < \arccos(k)$ ensure that the solutions of f and g are real and finite. Equation (C.7) and (C.8) are second order equations with parameters $(a_f, 2b_f, c_f)$ and $(a_g, 2b_g, c_g)$. Thus, these equations have two possible solutions:

$$\begin{aligned} i_1 &\triangleq \frac{-b_i - \sqrt{b_i^2 - a_i c_i}}{a_i}, \\ i_2 &\triangleq \frac{-b_i + \sqrt{b_i^2 - a_i c}}{a_i}, \end{aligned} \tag{C.10}$$

for $i = \{f, g\}$. In $\mathbb{D}$ and by Assumption 8.2, $a_i < 0$ and $c_i \geq 0$. Furthermore, since $\sqrt{b_i^2 - a_i c_i} \geq b_i$, i_1 will always be positive (≥ 0) and i_2 will always be negative (≤ 0). Hence, f and g are thus chosen as follows:

$$f = \begin{cases} f_1 & \hat{V}_z^f \geq 0 \\ f_2 & \hat{V}_z^f < 0 \end{cases} \tag{C.11}$$

$$g = \begin{cases} g_1 & \hat{V}_y^f \geq 0 \\ g_2 & \hat{V}_y^f < 0 \end{cases} \tag{C.12}$$

Finally, the controller errors are defined as

$$\tilde{u}_r = u_r - u_{r,\text{des}}, \tag{C.13}$$

$$\boldsymbol{z}_1 = \begin{bmatrix} \tilde{\theta}_{fc} \\ \tilde{\psi}_{fc} \end{bmatrix} \triangleq \begin{bmatrix} \theta_{fc} \\ \psi_{fc} \end{bmatrix} - \begin{bmatrix} \theta_{fc,\text{des}} \\ \psi_{fc,\text{des}} \end{bmatrix}, \tag{C.14}$$

$$\boldsymbol{z}_2 = \begin{bmatrix} \tilde{q} \\ \tilde{r} \end{bmatrix} \triangleq \begin{bmatrix} q \\ r \end{bmatrix} - \begin{bmatrix} q_{\text{des}} \\ r_{\text{des}} \end{bmatrix}. \tag{C.15}$$

Equation (8.37) describes the path error dynamics and is given as

$$\begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \\ \dot{z}_{b/f} \end{bmatrix} = \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} - \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \\ z_{b/f} \end{bmatrix} + [\boldsymbol{R}_f^i(\boldsymbol{\Theta}_{if})]^T \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}. \tag{C.16}$$

Here, $\boldsymbol{\Theta}_{if} = [0, \theta_f, \psi_f]$ denotes the orientation of the Serret-Frenet reference frame relative to the inertial frame.

The rotation matrix $\boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb})$ can be rewritten as

$$\boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) = \boldsymbol{R}_c^f(\boldsymbol{\Theta}_{fc})\boldsymbol{R}_b^c(\boldsymbol{\Theta}_{cb}), \tag{C.17}$$

where $\boldsymbol{\Theta}_{fc} = [\phi_{fc}, \theta_{fc}, \psi_{fc}]^T$ is calculated from the rotation matrix in (8.7), $\boldsymbol{\Theta}_{cb} = [0, \alpha_c, -\beta_c]^T$ and

$$\begin{aligned} \alpha_c &\triangleq \arctan\left(\frac{w_r}{u_{r,\text{des}}}\right), \\ \beta_c &\triangleq \arctan\left(\frac{v_r}{\sqrt{u_{r,\text{des}}^2 + w_r^2}}\right). \end{aligned} \tag{C.18}$$

Using the formula (2.7),

$$\begin{aligned} \sin(\alpha_c) &= \frac{w_r}{\sqrt{u_{r,\text{des}}^2 + w_r^2}}, \\ \cos(\alpha_c) &= \frac{u_{r,\text{des}}}{\sqrt{u_{r,\text{des}}^2 + w_r^2}}, \\ \sin(\beta_c) &= \frac{v_r}{\sqrt{u_{r,\text{des}}^2 + v_r^2 + w_r^2}} = \frac{v_r}{U_{\text{des}}}, \\ \cos(\beta_c) &= \frac{\sqrt{u_{r,\text{des}}^2 + w_r^2}}{U_{\text{des}}}. \end{aligned} \tag{C.19}$$

Hence,

$$
\begin{aligned}
\boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} &= \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} u_{r,\mathrm{des}} \\ v_r \\ w_r \end{bmatrix} + \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} \tilde{u} \\ 0 \\ 0 \end{bmatrix} \\
&= \boldsymbol{R}_c^f(\boldsymbol{\Theta}_{fc}) \boldsymbol{R}_b^c(\boldsymbol{\Theta}_{cb}) \begin{bmatrix} u_{r,\mathrm{des}} \\ v_r \\ w_r \end{bmatrix} + \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} \tilde{u} \\ 0 \\ 0 \end{bmatrix} \\
&= \boldsymbol{R}_c^f(\boldsymbol{\Theta}_{fc}) \begin{bmatrix} \cos(\alpha_c)\cos(\beta_c) & \sin(\beta_c) & \sin(\alpha_c)\cos(\beta_c) \\ -\cos(\alpha_c)\sin(\beta_c) & \cos(\beta_c) & -\sin(\alpha_c)\sin(\beta_c) \\ -\sin(\alpha_c) & 0 & \cos(\alpha_c) \end{bmatrix} \begin{bmatrix} u_{r,\mathrm{des}} \\ v_r \\ w_r \end{bmatrix} \\
&\qquad + \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} \tilde{u} \\ 0 \\ 0 \end{bmatrix} \\
&= \boldsymbol{R}_c^f(\boldsymbol{\Theta}_{fc}) \begin{bmatrix} \frac{u_{r,\mathrm{des}}}{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}} \frac{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}}{U_{\mathrm{des}}} & \frac{v_r}{U_{\mathrm{des}}} & \frac{w_r}{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}} \frac{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}}{U_{\mathrm{des}}} \\ -\frac{u_{r,\mathrm{des}}}{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}} \frac{v_r}{U_{\mathrm{des}}} & \frac{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}}{U_{\mathrm{des}}} & -\frac{w_r}{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}} \frac{v_r}{U_{\mathrm{des}}} \\ -\frac{w_r}{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}} & 0 & \frac{u_{r,\mathrm{des}}}{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}} \end{bmatrix} \begin{bmatrix} u_{r,\mathrm{des}} \\ v_r \\ w_r \end{bmatrix} \\
&\qquad + \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} \tilde{u} \\ 0 \\ 0 \end{bmatrix} \\
&= \boldsymbol{R}_c^f(\boldsymbol{\Theta}_{fc}) \begin{bmatrix} \frac{u_{r,\mathrm{des}}^2+v_r^2+w_r^2}{U_{\mathrm{des}}} \\ \frac{-u_{r,\mathrm{des}}^2 v_r+(u_{r,\mathrm{des}}^2+w_r^2)v_r-w_r^2 v_r}{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}U_{\mathrm{des}}} \\ \frac{-u_{r,\mathrm{des}}w_r+u_{r,\mathrm{des}}w_r}{\sqrt{u_{r,\mathrm{des}}^2+w_r^2}} \end{bmatrix} + \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} \tilde{u} \\ 0 \\ 0 \end{bmatrix} \\
&= \boldsymbol{R}_c^f(\boldsymbol{\Theta}_{fc}) \begin{bmatrix} U_{\mathrm{des}} \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} \tilde{u} \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} U_{\mathrm{des}}\cos(\theta_{fc})\cos(\psi_{fc}) \\ U_{\mathrm{des}}\cos(\theta_{fc})\sin(\psi_{fc}) \\ -U_{\mathrm{des}}\sin(\theta_{fc}) \end{bmatrix} + \begin{bmatrix} \tilde{u}_r\cos(\theta_{fb})\cos(\psi_{fb}) \\ \tilde{u}_r\cos(\theta_{fb})\sin(\psi_{fb}) \\ -\tilde{u}_r\sin(\theta_{fb}) \end{bmatrix}. \qquad \text{(C.20)}
\end{aligned}
$$

Furthermore,

$$
\begin{aligned}
\cos(\theta_{fc}) = \cos(\theta_{fc,\mathrm{des}} + \tilde{\theta}_{fc}) &= \cos(\theta_{fc,\mathrm{des}}) \\
&+ \cos(\theta_{fc,\mathrm{des}})\left(\cos(\tilde{\theta}_{fc}) - 1\right) - \sin(\theta_{fc,\mathrm{des}})\sin(\tilde{\theta}_{fc}), \qquad \text{(C.21)}
\end{aligned}
$$

$$
\begin{aligned}
\sin(\theta_{fc}) = \sin(\theta_{fc,\mathrm{des}} + \tilde{\theta}_{fc}) &= \sin(\theta_{fc,\mathrm{des}}) \\
&+ \sin(\theta_{fc,\mathrm{des}})\left(\cos(\tilde{\theta}_{fc}) - 1\right) + \cos(\theta_{fc,\mathrm{des}})\sin(\tilde{\theta}_{fc}), \qquad \text{(C.22)}
\end{aligned}
$$

$$
\begin{aligned}
\cos(\psi_{fc}) = \cos(\psi_{fc,\mathrm{des}} + \tilde{\psi}_{fc}) &= \cos(\psi_{fc,\mathrm{des}}) \\
&+ \cos(\psi_{fc,\mathrm{des}})\left(\cos(\tilde{\psi}_{fc}) - 1\right) - \sin(\psi_{fc,\mathrm{des}})\sin(\tilde{\psi}_{fc}), \qquad \text{(C.23)}
\end{aligned}
$$

$$
\begin{aligned}
\sin(\psi_{fc}) = \sin(\psi_{fc,\mathrm{des}} + \tilde{\psi}_{fc}) &= \sin(\psi_{fc,\mathrm{des}}) \\
&+ \sin(\psi_{fc,\mathrm{des}})\left(\cos(\tilde{\psi}_{fc}) - 1\right) + \cos(\psi_{fc,\mathrm{des}})\sin(\tilde{\psi}_{fc}), \qquad \text{(C.24)}
\end{aligned}
$$

and

$$
\begin{aligned}
&\cos(\theta_{fc})\cos(\psi_{fc}) = \cos(\theta_{fc,\mathrm{des}} + \tilde{\theta}_{fc})\cos(\psi_{fc,\mathrm{des}} + \tilde{\psi}_{fc}) \\
&\quad = \cos(\theta_{fc,\mathrm{des}})\cos(\psi_{fc,\mathrm{des}}) - \sin(\theta_{fc,\mathrm{des}})\sin(\tilde{\theta}_{fc})\cos(\psi_{fc,\mathrm{des}})\cos(\tilde{\psi}_{fc}) \\
&\qquad + \sin(\theta_{fc,\mathrm{des}})\sin(\tilde{\theta}_{fc})\sin(\psi_{fc,\mathrm{des}})\sin(\tilde{\psi}_{fc}) + \cos(\theta_{fc,\mathrm{des}})\cos(\psi_{fc,\mathrm{des}})\left(\cos(\tilde{\theta}_{fc}) - 1\right) \\
&\qquad - \cos(\theta_{fc,\mathrm{des}})\cos(\tilde{\theta}_{fc})\sin(\psi_{fc,\mathrm{des}})\sin(\tilde{\psi}_{fc}) \\
&\qquad + \cos(\theta_{fc,\mathrm{des}})\cos(\tilde{\theta}_{fc})\cos(\psi_{fc,\mathrm{des}})\left(\cos(\tilde{\psi}_{fc}) - 1\right) \\
&\quad = \cos(\theta_{fc,\mathrm{des}})\cos(\psi_{fc,\mathrm{des}}) + \tilde{\theta}_{fc}\left[-\frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}}\sin(\theta_{fc,\mathrm{des}})\cos(\psi_{fc,\mathrm{des}})\cos(\tilde{\psi}_{fc})\right. \\
&\qquad \left. + \frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}}\sin(\theta_{fc,\mathrm{des}})\sin(\psi_{fc,\mathrm{des}})\sin(\tilde{\psi}_{fc}) + \frac{\cos(\tilde{\theta}_{fc}) - 1}{\tilde{\theta}_{fc}}\cos(\theta_{fc,\mathrm{des}})\cos(\psi_{fc,\mathrm{des}})\right] \\
&\qquad + \tilde{\psi}_{fc}\left[-\frac{\sin(\tilde{\psi}_{fc})}{\tilde{\psi}_{fc}}\cos(\theta_{fc,\mathrm{des}})\cos(\tilde{\theta}_{fc})\sin(\psi_{fc,\mathrm{des}})\right. \\
&\qquad \left. + \frac{\cos(\tilde{\psi}_{fc}) - 1}{\tilde{\psi}_{fc}}\cos(\theta_{fc,\mathrm{des}})\cos(\tilde{\theta}_{fc})\cos(\psi_{fc,\mathrm{des}})\right]_{,}
\end{aligned}
\tag{C.25}
$$

$$
\begin{aligned}
&\cos(\theta_{fc})\sin(\psi_{fc}) = \cos(\theta_{fc,\mathrm{des}} + \tilde{\theta}_{fc})\sin(\psi_{fc,\mathrm{des}} + \tilde{\psi}_{fc}) \\
&\quad = \cos(\theta_{fc,\mathrm{des}})\sin(\psi_{fc,\mathrm{des}}) - \sin(\theta_{fc,\mathrm{des}})\sin(\tilde{\theta}_{fc})\sin(\psi_{fc,\mathrm{des}})\cos(\psi\theta_{fc}) \\
&\qquad - \sin(\theta_{fc,\mathrm{des}})\sin(\tilde{\theta}_{fc})\cos(\psi_{fc,\mathrm{des}})\sin(\psi\theta_{fc}) + \cos(\theta_{fc,\mathrm{des}})\sin(\psi_{fc,\mathrm{des}})\left(\cos(\tilde{\theta}_{fc}) - 1\right) \\
&\qquad + \cos(\theta_{fc,\mathrm{des}})\cos(\tilde{\theta}_{fc})\cos(\psi_{fc,\mathrm{des}})\sin(\psi\theta_{fc}) \\
&\qquad + \cos(\theta_{fc,\mathrm{des}})\cos(\tilde{\theta}_{fc})\sin(\psi_{fc,\mathrm{des}})\left(\cos(\tilde{\theta}_{fc}) - 1\right) \\
&\quad = \cos(\theta_{fc,\mathrm{des}})\sin(\psi_{fc,\mathrm{des}}) + \tilde{\theta}_{fc}\left[-\frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}}\sin(\theta_{fc,\mathrm{des}})\sin(\psi_{fc,\mathrm{des}})\cos(\tilde{\psi}_{fc})\right. \\
&\qquad \left. -\frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}}\sin(\theta_{fc,\mathrm{des}})\cos(\psi_{fc,\mathrm{des}})\sin(\psi\theta_{fc}) + \frac{\cos(\tilde{\theta}_{fc}) - 1}{\tilde{\theta}_{fc}}\cos(\theta_{fc,\mathrm{des}})\sin(\psi_{fc,\mathrm{des}})\right] \\
&\qquad + \tilde{\psi}_{fc}\left[\frac{\sin(\tilde{\psi}_{fc})}{\tilde{\psi}_{fc}}\cos(\theta_{fc,\mathrm{des}})\cos(\tilde{\theta}_{fc})\cos(\psi_{fc,\mathrm{des}})\right. \\
&\qquad \left. +\frac{\cos(\tilde{\psi}_{fc}) - 1}{\tilde{\psi}_{fc}}\cos(\theta_{fc,\mathrm{des}})\cos(\tilde{\theta}_{fc})\sin(\psi_{fc,\mathrm{des}})\right]_{,}
\end{aligned}
\tag{C.26}
$$

$$
\begin{aligned}
\sin(\theta_{fc}) &= \sin(\theta_{fc,\mathrm{des}} + \tilde{\theta}_{fc}) = \sin(\theta_{fc,\mathrm{des}}) + \sin(\theta_{fc,\mathrm{des}})\left(\cos(\tilde{\theta}_{fc}) - 1\right) + \cos(\theta_{fc,\mathrm{des}})\sin(\tilde{\theta}_{fc}) \\
&= \sin(\theta_{fc,\mathrm{des}}) + \tilde{\theta}_{fc}\left[\frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}}\cos(\theta_{fc,\mathrm{des}}) + \frac{\cos(\tilde{\theta}_{fc}) - 1}{\tilde{\theta}_{fc}}\sin(\theta_{fc,\mathrm{des}})\right]_{.}
\end{aligned}
\tag{C.27}
$$

Inserting the above equations into (C.20) yields the following result:

$$
\begin{aligned}
\boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} &= \begin{bmatrix} U_{\text{des}} \cos(\theta_{fc}) \cos(\psi_{fc}) \\ U_{\text{des}} \cos(\theta_{fc}) \sin(\psi_{fc}) \\ -U_{\text{des}} \sin(\theta_{fc}) \end{bmatrix} + \begin{bmatrix} \tilde{u}_r \cos(\theta_{fb}) \cos(\psi_{fb}) \\ \tilde{u}_r \cos(\theta_{fb}) \sin(\psi_{fb}) \\ -\tilde{u}_r \sin(\theta_{fb}) \end{bmatrix} \\
&= \begin{bmatrix} U_{\text{des}} \cos(\theta_{fc,\text{des}}) \cos(\psi_{fc,\text{des}}) \\ U_{\text{des}} \cos(\theta_{fc,\text{des}}) \sin(\psi_{fc,\text{des}}) \\ -U_{\text{des}} \sin(\theta_{fc,\text{des}}) \end{bmatrix} + \underbrace{\begin{bmatrix} \cos(\theta_{fb}) \cos(\psi_{fb}) & h_{11} & h_{12} \\ \cos(\theta_{fb}) \sin(\psi_{fb}) & h_{21} & h_{22} \\ -\sin(\theta_{fb}) & h_{31} & 0 \end{bmatrix}}_{\triangleq \boldsymbol{H}_1} \begin{bmatrix} \tilde{u}_r \\ \tilde{\theta}_{fc} \\ \tilde{\psi}_{fc} \end{bmatrix},
\end{aligned}
\tag{C.28}
$$

where

$$
\begin{aligned}
h_{11} = U_{\text{des}} \Bigg(& \frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}} \sin(\theta_{fc,\text{des}}) \left(-\cos(\psi_{fc,\text{des}}) \cos(\tilde{\psi}_{fc}) + \sin(\psi_{fc,\text{des}}) \sin(\tilde{\psi}_{fc}) \right) \\
& + \frac{\cos(\tilde{\theta}_{fc}) - 1}{\tilde{\theta}_{fc}} \cos(\theta_{fc,\text{des}}) \cos(\psi_{fc,\text{des}}) \Bigg),
\end{aligned}
\tag{C.29}
$$

$$
h_{12} = U_{\text{des}} \cos(\theta_{fc,\text{des}}) \cos(\tilde{\theta}_{fc}) \left(-\frac{\sin(\tilde{\psi}_{fc})}{\tilde{\psi}_{fc}} \sin(\psi_{fc,\text{des}}) + \frac{\cos(\tilde{\psi}_{fc}) - 1}{\tilde{\psi}_{fc}} \cos(\psi_{fc,\text{des}}) \right),
\tag{C.30}
$$

$$
\begin{aligned}
h_{21} = U_{\text{des}} \Bigg(& \frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}} \sin(\theta_{fc,\text{des}}) \left(-\sin(\psi_{fc,\text{des}}) \cos(\tilde{\psi}_{fc}) - \cos(\psi_{fc,\text{des}}) \sin(\tilde{\psi}_{fc}) \right) \\
& + \frac{\cos(\tilde{\theta}_{fc}) - 1}{\tilde{\theta}_{fc}} \cos(\theta_{fc,\text{des}}) \sin(\psi_{fc,\text{des}}) \Bigg),
\end{aligned}
\tag{C.31}
$$

$$
h_{22} = U_{\text{des}} \cos(\theta_{fc,\text{des}}) \cos(\tilde{\theta}_{fc}) \left(\frac{\sin(\tilde{\psi}_{fc})}{\tilde{\psi}_{fc}} \cos(\psi_{fc,\text{des}}) + \frac{\cos(\tilde{\psi}_{fc}) - 1}{\tilde{\psi}_{fc}} \sin(\psi_{fc,\text{des}}) \right),
\tag{C.32}
$$

$$
h_{31} = U_{\text{des}} \left(-\frac{\sin(\tilde{\theta}_{fc})}{\tilde{\theta}_{fc}} \cos(\theta_{fc,\text{des}}) - \frac{\cos(\tilde{\theta}_{fc}) - 1}{\tilde{\theta}_{fc}} \sin(\theta_{fc,\text{des}}) \right).
\tag{C.33}
$$

Thus, the dynamics of $x_{b/f}$, $y_{b/f}$ and $z_{b/f}$ in equation (C.16) can be rewritten as

$$
\begin{aligned}
\begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \\ \dot{z}_{b/f} \end{bmatrix} &= \boldsymbol{R}_b^f(\boldsymbol{\Theta}_{fb}) \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} - \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \\ z_{b/f} \end{bmatrix} + \boldsymbol{R}_f^i(\boldsymbol{\Theta}_{if})^T \boldsymbol{V}_c \\
&= \begin{bmatrix} U_{\text{des}} \cos(\theta_{fc,\text{des}}) \cos(\psi_{fc,\text{des}}) \\ U_{\text{des}} \cos(\theta_{fc,\text{des}}) \sin(\psi_{fc,\text{des}}) \\ -U_{\text{des}} \sin(\theta_{fc,\text{des}}) \end{bmatrix} + \boldsymbol{H}_1 \begin{bmatrix} \tilde{u}_r \\ \tilde{\theta}_{fc} \\ \tilde{\psi}_{fc} \end{bmatrix} - \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \\ z_{b/f} \end{bmatrix} + \boldsymbol{V}_c^f
\end{aligned}
\tag{C.34}
$$

By applying (2.7) and inserting the guidance laws (C.5)-(C.6),

$$\cos(\theta_{fc,\text{des}}) = \frac{\sqrt{\Delta^2 + y_{b/f}^2}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f} + f)^2}}, \tag{C.35}$$

$$\sin(\theta_{fc,\text{des}}) = \frac{z_{b/f} + f}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f} + f)^2}}, \tag{C.36}$$

$$\cos(\psi_{fc,\text{des}}) = \frac{\sqrt{\Delta^2 + x_{b/f}^2 + z_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2 + z_{b/f}^2}}, \tag{C.37}$$

$$\sin(\psi_{fc,\text{des}}) = -\frac{y_{b/f} + g}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2 + z_{b/f}^2}}. \tag{C.38}$$

This can be inserted into the $x_{b/f}$, $y_{b/f}$ and $z_{b/f}$ dynamics in (C.34).

$$\begin{aligned}
\dot{x}_{b/f} &= U_{\text{des}} \cos(\theta_{fc,\text{des}}) \cos(\psi_{fc,\text{des}}) + \cos(\theta_{fb}) \cos(\psi_{fb}) \tilde{u}_r + h_{11} \tilde{\theta}_{fc,\text{des}} + h_{12} \tilde{\psi}_{fc} \\
&\quad - \dot{s} + \kappa \dot{s} y_{b/f} + V_x^f \\
&= U_{\text{des}} \frac{\sqrt{\Delta^2 + y_{b/f}^2}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f} + f)^2}} \frac{\sqrt{\Delta^2 + x_{b/f}^2 + z_{b/f}^2}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2 + z_{b/f}^2}} \\
&\quad - \dot{s} + \kappa \dot{s} y_{b/f} + V_x^f + \cos(\theta_{fb}) \cos(\psi_{fb}) \tilde{u}_r + h_{11} \tilde{\theta}_{fc,\text{des}} + h_{12} \tilde{\psi}_{fc}
\end{aligned} \tag{C.39}$$

Inserting for $\dot{s}$ as defined in (C.4) and the current and current estimates as defined in (C.2)-(C.3) results in the following expression:

$$\begin{aligned}
\dot{x}_{b/f} &= -U_{\text{des}} \frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2}} + \kappa \dot{s} y_{b/f} + V_x^f - \hat{V}_x^f + \cos(\theta_{fb}) \cos(\psi_{fb}) \tilde{u}_r \\
&\quad + h_{11} \tilde{\theta}_{fc,\text{des}} + h_{12} \tilde{\psi}_{fc} \\
&= -U_{\text{des}} \frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2}} + \kappa \dot{s} y_{b/f} + \cos(\theta_{fb}) \cos(\psi_{fb}) \tilde{u}_r + h_{11} \tilde{\theta}_{fc,\text{des}} \\
&\quad + h_{12} \tilde{\psi}_{fc} + \cos(\psi_f) \cos(\theta_f) \tilde{V}_x + \cos(\theta_f) \sin(\psi_f) \tilde{V}_y - \sin(\theta_f) \tilde{V}_z
\end{aligned} \tag{C.40}$$

Similarly,

$$\begin{aligned}
\dot{y}_{b/f} &= U_{\text{des}} \cos(\theta_{fc,\text{des}}) \sin(\psi_{fc,\text{des}}) + \cos(\theta_{fb}) \sin(\psi_{fb}) \tilde{u}_r + h_{21} \tilde{\theta}_{fc} + h_{22} \tilde{\psi}_{fc} \\
&\quad - \kappa \dot{s} x_{b/f} + \tau \dot{s} z_{b/f} + V_y^f \\
&= -U_{\text{des}} \cos(\theta_{fc,\text{des}}) \frac{y_{b/f} + g}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2 + z_{b/f}^2}} - \kappa \dot{s} x_{b/f} + \tau \dot{s} z_{b/f} \\
&\quad + V_y^f + \cos(\theta_{fb}) \sin(\psi_{fb}) \tilde{u}_r + h_{21} \tilde{\theta}_{fc} + h_{22} \tilde{\psi}_{fc}
\end{aligned} \tag{C.41}$$

Note that the term g is chosen so that

$$-U_{\text{des}}\cos(\theta_{fc,\text{des}})\frac{g}{\sqrt{\Delta^2+x_{b/f}^2+(y_{b/f}+g)^2+z_{b/f}^2}}+\hat{V}_y^f=0. \tag{C.42}$$

Solving the above equation with respect to g leads to the second-order equation (C.8), which has been proven to have real solutions, $g_1 \geq 0$ and $g_2 \leq 0$. Thus, by choosing g as in (C.12), (C.41) reduces to

$$\begin{aligned}
\dot{y}_{b/f} &= -U_{\text{des}}\frac{\sqrt{\Delta^2+y_{b/f}^2}}{\sqrt{\Delta^2+y_{b/f}^2+(z_{b/f}+f)^2}}\frac{y_{b/f}}{\sqrt{\Delta^2+x_{b/f}^2+(y_{b/f}+g)^2+z_{b/f}^2}}\\
&\quad -\kappa\dot{s}x_{b/f}+\tau\dot{s}z_{b/f}+V_y^f-\hat{V}_y^f+\cos(\theta_{fb})\sin(\psi_{fb})\tilde{u}_r+h_{21}\tilde{\theta}_{fc}+h_{22}\tilde{\psi}_{fc}\\
&= -U_{\text{des}}\frac{\sqrt{\Delta^2+y_{b/f}^2}}{\sqrt{\Delta^2+y_{b/f}^2+(z_{b/f}+f)^2}}\frac{y_{b/f}}{\sqrt{\Delta^2+x_{b/f}^2+(y_{b/f}+g)^2+z_{b/f}^2}}\\
&\quad -\kappa\dot{s}x_{b/f}+\tau\dot{s}z_{b/f}+\cos(\theta_{fb})\sin(\psi_{fb})\tilde{u}_r+h_{21}\tilde{\theta}_{fc}+h_{22}\tilde{\psi}_{fc}\\
&\quad +\left(\cos(\psi_f)\sin(\phi_f)\sin(\theta_f)-\cos(\phi_f)\sin(\psi_f)\right)\tilde{V}_x+\cos(\theta_f)\sin(\phi_f)\tilde{V}_z\\
&\quad +\left(\cos(\phi_f)\cos(\psi_f)+\sin(\phi_f)\sin(\psi_f)\sin(\theta_f)\right)\tilde{V}_y.
\end{aligned} \tag{C.43}$$

Furthermore,

$$\begin{aligned}
\dot{z}_{b/f} &= -U_{\text{des}}\sin(\theta_{fc,\text{des}})-\sin(\theta_{fb})\tilde{u}_r+h_{31}\tilde{\theta}_{fc}+h_{32}\tilde{\psi}_{fc}-\tau\dot{s}y_{b/f}+V_z^f\\
&= -U_{\text{des}}\frac{z_{b/f}+f}{\sqrt{\Delta^2+y_{b/f}^2+(z_{b/f}+f)^2}}-\tau\dot{s}y_{b/f}+V_z^f-\sin(\theta_{fb})\tilde{u}_r\\
&\quad +h_{31}\tilde{\theta}_{fc}+h_{32}\tilde{\psi}_{fc},
\end{aligned} \tag{C.44}$$

where the term f has been chosen so

$$-U_{\text{des}}\frac{f}{\sqrt{\Delta^2+y_{b/f}^2+(z_{b/f}+f)^2}}+\hat{V}_z^f=0. \tag{C.45}$$

Thus,

$$\begin{aligned}
\dot{z}_{b/f} &= -U_{\text{des}}\frac{z_{b/f}}{\sqrt{\Delta^2+y_{b/f}^2+(z_{b/f}+f)^2}}-\tau\dot{s}y_{b/f}+\tilde{V}_z^f-\sin(\theta_{fb})\tilde{u}_r+h_{31}\tilde{\theta}_{fc}+h_{32}\tilde{\psi}_{fc}\\
&= -U_{\text{des}}\frac{z_{b/f}}{\sqrt{\Delta^2+y_{b/f}^2+(z_{b/f}+f)^2}}-\tau\dot{s}y_{b/f}-\sin(\theta_{fb})\tilde{u}_r+h_{31}\tilde{\theta}_{fc}+h_{32}\tilde{\psi}_{fc}\\
&\quad +\left(\sin(\phi_f)\sin(\psi_f)+\cos(\phi_f)\cos(\psi_f)\sin(\theta_f)\right)\tilde{V}_x+\cos(\phi_f)\cos(\theta_f)\tilde{V}_z\\
&\quad +\left(\cos(\phi_f)\sin(\psi_f)\sin(\theta_f)-\cos(\psi_f)\sin(\phi_f)\right)\tilde{V}_y.
\end{aligned} \tag{C.46}$$

This can be expressed as (8.38), thus concluding this appendix:

$$\begin{bmatrix} \dot{x}_{b/f} \\ \dot{y}_{b/f} \\ \dot{z}_{b/f} \end{bmatrix} = \begin{bmatrix} -U_{\text{des}} \frac{x_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + y_{b/f}^2 + z_{b/f}^2}} \\ -U_{\text{des}} \frac{\sqrt{\Delta^2 + y_{b/f}^2}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f} + f)^2}} \frac{y_{b/f}}{\sqrt{\Delta^2 + x_{b/f}^2 + (y_{b/f} + g)^2 + z_{b/f}^2}} \\ -U_{\text{des}} \frac{z_{b/f}}{\sqrt{\Delta^2 + y_{b/f}^2 + (z_{b/f} + f)^2}} \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} x_{b/f} \\ y_{b/f} \\ z_{b/f} \end{bmatrix} + \boldsymbol{H}(t, U_{\text{des}}, \boldsymbol{\xi}) \boldsymbol{\xi} \tag{C.47}$$

$$\boldsymbol{H}(t, U_{\text{des}}, \boldsymbol{\xi})^T = \begin{bmatrix} c(\theta_{fb})c(\psi_{fb}) & c(\theta_{fb})s(\psi_{fb}) & -s(\theta_{fb}) \\ h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ c(\psi_f)c(\theta_f) & c(\psi_f)s(\phi_f)s(\theta_f) - c(\phi_f)s(\psi_f) & c(\phi_f)c(\psi_f)s(\theta_f) + s(\phi_f)s(\psi_f) \\ c(\theta_f)s(\psi_f) & s(\phi_f)s(\psi_f)s(\theta_f) + c(\phi_f)c(\psi_f) & c(\phi_f)s(\psi_f)s(\theta_f) - c(\psi_f)s(\phi_f) \\ -s(\theta_f) & c(\theta_f)s(\phi_f) & c(\phi_f)c(\theta_f) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{C.48}$$

References

[1] A. P. Aguiar and A. M. Pascoal. Dynamic positioning and way-point tracking of underactuated AUVs in the presence of ocean currents. In *Proc. 41st IEEE Conference on Decision and Control*, volume 2, pages 2105–2110, Las Vegas, NV, USA, 2002.

[2] G. Antonelli. *Underwater Robots, Motion and Force Control of Vehicle-Manipulator Systems*. Springer-Verlag Berlin Heidelberg, 2008.

[3] G. Antonelli. Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems. *IEEE Transactions on Robotics*, 25(5):985–994, 2009.

[4] G. Antonelli. *Underwater Robots*, volume 96 of *Springer Tracts in Advanced Robotics*. Springer International Publishing, 2014.

[5] G. Antonelli and S. Chiaverini. Task-priority redundancy resolution for underwater vehicle-manipulator systems. In *Proc. IEEE International Conference on Robotics and Automation*, volume 1, pages 768–773, Leuven, Belgium, 1998.

[6] G. Antonelli, F. Arrichiello, and S. Chiaverini. The null-space-based behavioral control for autonomous robotic systems. *Intelligent Service Robotics*, 1(1):27–39, 2008.

[7] G. Antonelli, S. Moe, and K. Y. Pettersen. Incorporating Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics. In *Proc. 23rd Mediterranean Conference on Control and Automation*, pages 1092 – 1097, Torremolinos, Spain, 2015.

[8] G. Antonelli, G. Indiveri, and S. Chiaverini. Prioritized closed-loop inverse kinematic algorithms for redundant robotic systems with velocity saturations. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5892–5897, St. Louis, MO, USA, 2009.

[9] H. Azimian, T. Looi, and J. Drake. Closed-loop inverse kinematics under inequality constraints: Application to concentric-tube manipulators. In *Proc. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 498–503, Chicago, IL, USA, 2014. IEEE.

[10] A. Balestrino, G. De Maria, and L. Sciavicco. Robust control of robotic manipulators. In *Proc. 9th IFAC World Congress*, pages 2435–2440, Budapest, Hungary, 1984.

[11] M. R. Benjamin, J. J. Leonard, C. J. A., and P. M. Newman. A method for protocol-based collision avoidance between autonomous marine surface craft. *Journal of Field Robotics*, 23(5):333–346, 2006.

[12] R. Bogue. Exoskeletons and robotic prosthetics: a review of recent developments. *Industrial Robot: An International Journal*, 36(5):421–427, 2009.

[13] E. Børhaug and K. Y. Pettersen. LOS path following for underactuated underwater vehicle. In *Proc. 7th IFAC Conference on Manoeuvring and Control of Marine Craft*, Lisbon, Portugal, 2006.

[14] E. Børhaug, A. Pavlov, and K. Y. Pettersen. Straight line path following for formations of underactuated underwater vehicles. In *Proc. 46th IEEE Conference on Decision and Control*, pages 2905–2912, New Orleans, LA, USA, 2007.

[15] E. Børhaug, A. Pavlov, and K. Y. Pettersen. Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents. In *Proc. 47th IEEE Conference on Decision and Control*, pages 4984–4991, Cancun, Mexico, 2008.

[16] J. Broekens, M. Heerink, and H. Rosendal. Assistive social robots in elderly care: a review. *International Journal on the Fundamental Aspects of Technology to Serve the Ageing Community*, 8(2):94–103, 2009.

[17] S. R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, 2009. URL `https://www.math.ucsd.edu/sbuss/ResearchWeb/ikmethods/iksurvey.pdf`.

[18] F. Caccavale and B. Siciliano. Kinematic control of redundant free-floating robotic systems. *Advanced Robotics*, 15(4):429–448, 2001.

[19] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani. Six-DOF Impedance Control of Dual-Arm Cooperative Manipulators. *IEEE/ASME Transactions on Mechatronics*, 13(5):576–586, 2008.

[20] W. Caharija, M. Candeloro, K. Y. Pettersen, and A. J. Sørensen. Relative Velocity Control and Integral LOS for Path Following of Underactuated Surface Vessels. In *Proc. 9th IFAC Conference on Manoeuvring and Control of Marine Craft*, pages 380–385, Arenzano, Italy, 2012.

[21] W. Caharija, K. Y. Pettersen, J. T. Gravdahl, and E. Børhaug. Path following of underactuated autonomous underwater vehicles in the presence of ocean currents. In *Proc. 51st IEEE Conference on Decision and Control*, pages 528–535, Maui, HI, USA, 2012.

[22] W. Caharija, K. Y. Pettersen, A. J. Sørensen, M. Candeloro, and J. T. Gravdahl. Relative velocity control and integral line of sight for path following of autonomous surface vessels: Merging intuition with theory. *Journal of Engineering for the Maritime Environment*, 228(2):180–191, 2014.

[23] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.

[24] R. D. Christ and R. L. Wernli. *The ROV Manual: A User Guide for Remotely Operated Vehicles.* Elsevier Science, 2013.

[25] M. de Lasa, I. Mordatch, and A. Hertzmann. Feature-Based Locomotion Controllers. *ACM Transactions on Graphics*, 29(4):131:1–131:10, 2010.

[26] O. Egeland and J. T. Gravdahl. *Modeling and Simulation for Automatic Control.* Marine Cybernetics, Trondheim, Norway, 1st edition, 2002.

[27] O. Egeland and K. Y. Pettersen. *Free-floating robotic systems*, volume 230, pages 119–134. Springer Berlin Heidelberg, 1998.

[28] P. Encarnacao, A. M. Pascoal, and M. Arcak. Path Following for Marine Vehicles in the Presence of Unknown Currents. *Proc. 6th IFAC International Symposium on Robot Control*, pages 469–474, 2000.

[29] A. Escande, N. Mansard, and P. B. Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.

[30] B. Faverjon and P. Tournassoud. A local based approach for path planning of manipulators with a high number of degrees of freedom. In *Proc. EEE International Conference on Robotics and Automation*, volume 4, pages 1152–1159, Raleigh, NC, USA, 1987.

[31] T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* Wiley, 2011.

[32] T. I. Fossen and A. M. Lekkas. Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents. *International Journal of Adaptive Control and Signal Processing*, 2015.

[33] T. I. Fossen and K. Y. Pettersen. On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws. *Automatica*, 50(11): 2912–2917, 2014.

[34] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *Robotics Automation Magazine, IEEE*, 4(1):23–33, 1997.

[35] E. Fredriksen and K. Y. Pettersen. Global κ-exponential way-point manoeuvering of ships. In *Proc. 43rd IEEE Conference on Decicion and Control*, pages 5360–5367, Atlantis, Paradise Island, Bahamas, 2004.

[36] P. J. From, J. T. Gravdahl, and K. Y. Pettersen. *Vehicle-Manipulator Systems.* Advances in Industrial Control. Springer London, 2014.

[37] R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness.* Princeton University Press, 2012.

[38] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, MD, USA, 3 edition, 1996.

[39] H. Hanafusa, T. Yoshikawa, and Y. Nakamura. Analysis and control of articulated robot arms with redundancy. In *Proc. 8th IFAC World Congress*, Kyoto, Japan, 1981.

[40] 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.

[41] E. Hernandez, M. Carreras, and P. Ridao. A comparison of homotopic path planning algorithms for robotic applications. *Robotics and Autonomous Systems*, 64(C):44–58, 2015.

[42] O. Kanoun, F. Lamiraux, and P. B. Wieber. Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *Robotics, IEEE Transactions on*, 27(4):785–792, 2011.

[43] H. K. Khalil. *Nonlinear systems.* Prentice Hall PTR, 2002.

[44] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, St. Louis, MO, USA, 1985.

[45] O. Khatib. *The potential field approach and operational space formulation in robot control.* Springer US, 1986.

[46] J. Kim and W. K. Chung. Dynamic Analysis and Active Damping Control for Underwater Vehicle-Manipulator Systems. In *Proc. International Offshore and Polar Engineering Conference*, Honolulu, HI, USA,, 2003.

[47] C. A. Klein and C. Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(2):245–250, 1983.

[48] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 1398–1404, Sacramento, CA, USA, 1991.

[49] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger. Safe Maritime Autonomous Navigation With COLREGS, Using Velocity Obstacles. *IEEE Journal of Oceanic Engineering*, 39(1):110–119, 2014.

[50] F. Lamnabhi-Lagarrigue, A. Loria, and E. Panteley. Advanced topics in control systems theory II, Lecture notes from FAP 2005, 2004.

[51] A. M. Lekkas and T. I. Fossen. Minimization of Cross-track and Along-track Errors for Path Tracking of Marine Underactuated Vehicles. In *Proc. European Control Conference*, pages 3004–3010, Strasbourg, France, 2014.

[52] A. M. Lekkas and T. I. Fossen. Trajectory tracking and ocean current estimation for marine underactuated vehicles. In *Proc. 2014 IEEE Conference on Control Applications*, pages 905–910, Juan Les Antibes, France, 2014.

[53] A. Liegeois. Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, 7(12):868–871, 1977.

[54] Ø. A. G. Loe. Collision Avoidance for Unmanned Surface Vehicles. Master's thesis, Norwegian University of Science and Technology, 2008.

[55] A. Loria, E. Panteley, D. Popovic, and A. R. Teel. A nested Matrosov theorem and persistency of excitation for uniform convergence in stable nonautonomous systems. *IEEE Transactions on Automatic Control*, 50(2):183–198, 2005.

[56] A. A. Maciejewski and C. A. Klein. Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments. *The International Journal of Robotics Research*, 4(3):109–117, 1985.

[57] N. Mansard and O. Khatib. Continuous control law from unilateral constraints. In *Proc. 2008 IEEE International Conference on Robotics and Automation*, pages 3359–3364, Pasadena, CA, 2008.

[58] N. Mansard, O. Khatib, and A. Kheddar. A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks. *IEEE Transactions on Robotics*, 25(3):670–685, 2009.

[59] E. Marchand, F. Chaumette, and A. Rizzo. Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1083–1090, Osaka, Japan, 1996.

[60] D. B. Marco, A. Martins, and A. J. Healy. Surge motion parameter identification for the nps phoenix auv. In *Proc. IARP 1st International Workshop on Autonomous Underwater Vehicles for Shallow Water and Coastal Environment*, pages 197–210, Lafayette, LA, USA, 1998.

[61] S. Moe and K. Y. Pettersen. Set-Based Line-of-Sight (LOS) Path Following with Collision Avoidance for Underactuated Unmanned Surface Vessel. In *Proc. 24th Mediterranean Conference on Control and Automation*, Athens, Greece, 2016.

[62] S. Moe, G. Antonelli, and K. Y. Pettersen. Null-space-based behavior guidance of planar dual-arm UVMS. In *Proc. IEEE International Conference on Robotics and Biomimetics*, pages 735–740, Bali, Indonesia, 2014.

[63] S. Moe, W. Caharija, K. Y. Pettersen, and I. Schjølberg. Path following of underactuated marine surface vessels in the presence of unknown ocean currents. In *Proc. American Control Conference*, pages 3856–3861, Portland, OR, USA, 2014.

[64] S. Moe, W. Caharija, K. Y. Pettersen, and I. Schjølberg. Path Following of Underactuated Marine Underwater Vehicles in the Presence of Unknown Ocean Currents. In *Proc. 33rd International Conference on Offshore Mechanics and Arctic Engineering*, volume 7, San Francisco, CA, USA, 2014.

[65] S. Moe, G. Antonelli, K. Y. Pettersen, and J. Schrimpf. Experimental Results for Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework. In *Proc. IEEE International Conference on Robotics and Biomimetics*, pages 1233 – 1239, Zhuhai, China, 2015.

[66] S. Moe, A. R. Teel, G. Antonelli, and K. Y. Pettersen. Stability Analysis for Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework. In *Proc. 54th IEEE Conference on Decision and Control*, pages 171 – 178, Osaka, Japan, 2015.

[67] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf. Set-Based Tasks within the Singularity-Robust Multiple Task-Priority Inverse Kinematics Framework: General Formulation, Stability Analysis, and Experimental Results. *Frontiers in Robotics and AI*, 3(16):1–18, 2016.

[68] S. Moe, K. Y. Pettersen, T. I. Fossen, and J. T. Gravdahl. Line-of-Sight Curved Path Following for Underactuated USVs and AUVs in the Horizontal Plane under the influence of Ocean Currents. In *Proc. 24th Mediterranean Conference on Control and Automation*, Athens, Greece, 2016.

[69] D. Nenchev, Y. Umetani, and Kazuya Y. Analysis of a redundant free-flying spacecraft/manipulator system. *IEEE Transactions on Robotics and Automation*, 8(1):1–6, 1992.

[70] M. Nomoto and M. Hattori. A deep ROV Dolphin 3K: Design and performance analysis. *IEEE Journal of Oceanic Engineering*, 11(3):373–391, 1986.

[71] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa. *Autonomous Flying Robots*. Springer Japan, 2010.

[72] S. M. Nornes, M. Ludvigsen, Ø. Ødegard, and A. J. Sørensen. Underwater Photogrammetric Mapping of an Intact Standing Steel Wreck with ROV. *IFAC-PapersOnLine*, 48(2):206–211, 2015.

[73] E. Panteley and A. Loria. On global uniform asymptotic stability of nonlinear time-varying systems in cascade. *Systems & Control Letters*, 33(2):131–138, 1998.

[74] E. Panteley and A. Loria. Brief growth rate conditions for uniform asymptotic stability of cascaded time-varying systems. *Automatica*, 37(3):453–460, 2001.

[75] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 30(3):286–297, 2000.

[76] R. Penrose and J. A. Todd. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, July 1955.

[77] K. Y. Pettersen and E. Lefeber. Way-point tracking control of ships. In *Proc. IEEE Conference on Decision and Control*, volume 1, pages 940–945, Orlando, FL, USA, 2001.

[78] I. Schjølberg and T. I. Fossen. Modelling and Control of Underwater Vehicle-Manipulator Systems. In *Proc. 3rd Conference on Marine Craft maneuvering and control*, pages 45–57, Southampton, UK, 1994.

[79] D. Shiyou, Z. Xiaoping, and L. Guoqing. The Null-Space-Based Behavioral Control for Swarm Unmanned Aerial Vehicles. In *Proc. 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pages 1003–1006, Beijing, China, 2011.

[80] B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics.* Springer Berlin Heidelberg, 2008.

[81] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: modelling, planning and control.* Springer Verlag, 2009.

[82] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta. Floating Underwater Manipulation: Developed Control Methodology and Experimental Validation within the TRIDENT Project. *Journal of Field Robotics*, 31(3): 364–385, 2013.

[83] R. Skjetne, U. Jørgensen, and A. R. Teel. Line-of-sight path-following along regularly parametrized curves solved as a generic maneuvering problem. In *Proc. IEEE Conference on Decision and Control and European Control Conference*, pages 2467–2474, Orlando, FL, USA, 2011.

[84] D. A. Smallwood and L. L. Whitcomb. Adaptive identification of dynamically positioned underwater robotic vehicles. *IEEE Transactions on Control Systems Technology*, 11(4):505–515, 2003.

[85] O. J. Sørdalen and O. Egeland. Exponential Stabilization of Nonholonomic Chained Systems. *IEEE Transactions on Automatic Control*, 40(1):35–49, 1995.

[86] M. W. Spong and S. Hutchinson. *Robot Modeling and Control.* Wiley, 2005.

[87] G. Tevatia and S. Schaal. Inverse kinematics for humanoid robots. In *Proc. IEEE International Conference on Robotics and Automation*, volume 1, pages 294–299, San Francisco, CA, USA, 2000.

[88] Unicas. Modeling and control for the MARIS UVMS. Technical report.

[89] United Nations Council on Trade and Development (UNCTAD). Review of Maritime Transport 2015. Technical report, New York and Geneva: United Nations, 2015.

[90] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha. *Reciprocal n-Body Collision Avoidance*, pages 3–19. Springer Berlin Heidelberg, 2011.

[91] R. Vidal, O. Shakernia, and S. Sastry. Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation. In *Proc. IEEE International Conference on Robotics and Automation*, volume 1, pages 584–589, Taipei, Taiwan, 2003.

[92] C. W. Wampler and L. J. Leifer. Applications of Damped Least-Squares Methods to Resolved-Rate and Resolved-Acceleration Control of Manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 110(1):31–38, 1988.

[93] W. Wolovich and H. Elliott. A computational technique for inverse kinematics. In *Proc. 23rd IEEE Conference on Decision and Control*, pages 1359–1363, Las Vegas, NV, USA, 1984.

[94] H. Wu, W. Tizzano, T. T. Andersen, N. A. Andersen, and O. Ravn. *Hand-eye calibration and inverse kinematics of robot arm using neural network*, volume 274, pages 581–591. Springer International Publishing, Cham, Switzerland, 2014.

[95] R. Zaier, editor. *The Future of Humanoid Robots - Research and Applications.* InTech, 2012.

[96] E. Zereik, A. Sorbara, M. Bibuli, G. Bruzzone, and M. Caccia. Priority Task Approach for USVs' Path Following Missions with Obstacle Avoidance and Speed Regulation. In *Proc. 10th Conference of Manoeuvering and Control of Marine Craft*, Copenhagen, Denmark, 2015.

[97] F. Zhang, G. Marani, R. N. Smith, and H. T. Choi. Future trends in marine robotics. *IEEE Robotics and Automation Magazine*, 22(1):14 – 122, 2015.

[98] S. Zhang, J. Yu, A. Zhang, and F. Zhang. Spiraling motion of underwater gliders: Modeling, analysis, and experimental results. *Ocean Engineering*, 60: 1–13, 2013.