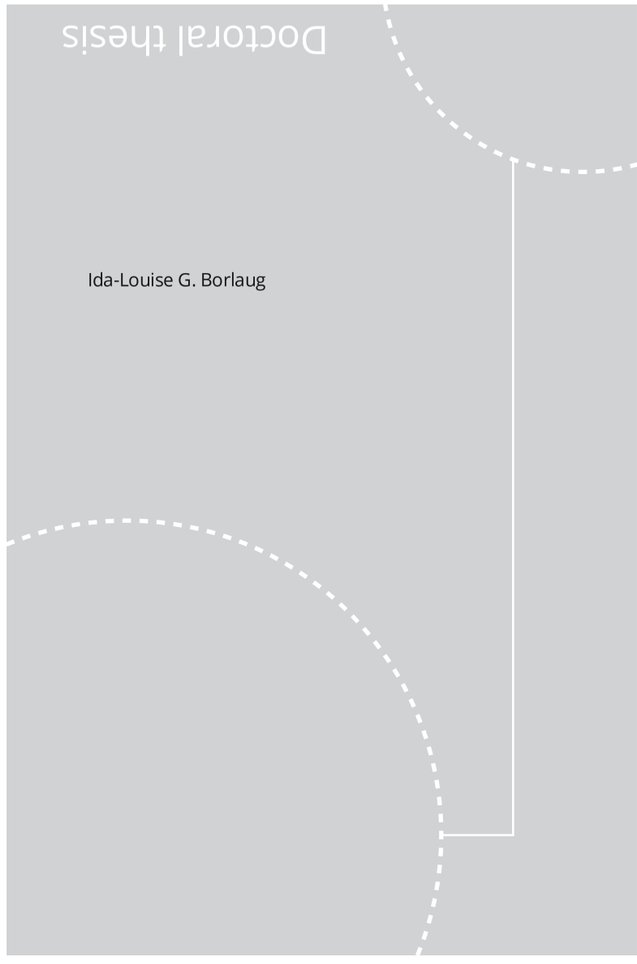
Doctoral theses at NTNU, 2020:345

Ida-Louise G. Borlaug

Robust Control of Articulated Intervention-AUVs Using Sliding Mode Control

Doctoral thesis



Ida-Louise G. Borlaug

ISBN 978-82-326-5036-1 (printed ver.)
ISBN 978-82-326-5037-8 (electronic ver.)
ISSN 1503-8181

NTNU
Norwegian University of Science and Technology

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Doctoral theses at NTNU, 2020:345

NTNU
Norwegian University of Science and Technology

Ida-Louise G. Borlaug

Robust Control of Articulated Intervention-AUVs Using Sliding Mode Control

Thesis for the Degree of Philosophiae Doctor

Trondheim, November 2020

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

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-326-5036-1 (printed ver.)
ISBN 978-82-326-5037-8 (electronic ver.)
ISSN 1503-8181

ITK-report 2020-8-W

Doctoral theses at NTNU, 2020:345

Printed by NTNU Grafisk senter

Summary

This thesis is concerned with the theoretical development of trajectory tracking control for an articulated intervention autonomous underwater vehicle (AUV) in 6DOF while using no or as little model knowledge as possible since an articulated intervention-AUV is subject to hydrodynamic and hydrostatic parameter uncertainties, uncertain thruster characteristics, unknown disturbances, and unmodelled dynamics, meaning that the control approach should be robust. To achieve this goal, robust algorithms based on sliding mode control are proposed. The stability of the proposed control approaches is analysed theoretically, and their performance is tested in simulations and experiments.

The first sliding mode control approach proposed for trajectory tracking control for an articulated intervention-AUV is the super-twisting algorithm with adaptive gains. At the start of this project, this algorithm was the most powerful second-order continuous sliding mode control algorithm available, as it attenuates chattering and does not require consideration of any conservative upper bound on the disturbance to maintain sliding because of the adaptive gains. Motivated by this characteristic, we solve the trajectory tracking control problem for an articulated intervention-AUV in 6DOF using the super-twisting algorithm with adaptive gains. We consider both the case where velocity measurements are available and the case where they are not. When velocity measurements are not available, we use a higher-order sliding mode observer to estimate the linear and angular velocities. The reason we choose the higher-order sliding mode observer is its strong stability properties. Specifically, the finite-time stability of the higher-order sliding mode observer enables us to prove that the closed-loop system is uniformly globally asymptotically stable. Finally, we demonstrate the applicability of the presented control schemes with comprehensive simulation and experimental results.

The second sliding mode control approach proposed for trajectory tracking control for an articulated intervention-AUV is the generalized super-twisting algorithm. The generalized super-twisting algorithm is an extension of the super-twisting algorithm that provides finite-time convergence in the case where both the perturbations and control coefficients are state- and time-dependent and the control coefficients are uncertain, which is essential for robust control of an articulated intervention-AUV. Motivated by this feature, we solve the trajectory tracking control problem for an articulated intervention-AUV in 6DOF using the generalized super-twisting algorithm. Furthermore, we prove the asymptotic convergence of the tracking errors. We also consider the case where velocity measurements are unavailable by solving the tracking control problem using the generalized super-twisting

algorithm in combination with a higher-order sliding mode observer. Furthermore, we prove the asymptotic convergence of the tracking errors when the higher-order sliding mode observer is included. We also present comprehensive simulation and experimental results that validate and demonstrate the applicability of both control schemes.

We then compare the previously proposed tracking control laws. Specifically, we compare the control law using the super-twisting algorithm with adaptive gains and the control law using the generalized super-twisting algorithm. We also compare the control laws when they are combined with a higher-order sliding mode observer. Furthermore, we solve the tracking problem using a PID controller to evaluate how the sliding mode control algorithms perform compared to this standard linear controller. In both simulations and experiments, the super-twisting algorithm with adaptive gains provided the best overall tracking performance, but the generalized super-twisting algorithm was not far from achieving the same tracking results. Thus, the fact that we had to tune the generalized super-twisting algorithm manually may be the reason that we achieved lower tracking performance with this algorithm. The super-twisting algorithm with adaptive gains is therefore better than the generalized super-twisting algorithm in practice, but the generalized super-twisting algorithm does have some theoretical advantages as it is proven to provide global finite-time stability for a larger class of systems. These additional theoretical properties make it possible to prove that the closed-loop system is uniformly globally asymptotically stable without the higher-order sliding mode observer when the generalized super-twisting algorithm is used. This is not possible for the super-twisting algorithm with adaptive gains.

Based on the results of the comparison, we develop an adaptive generalized super-twisting algorithm for a class of systems whose perturbations and uncertain control coefficients are time- and state-dependent, i.e., we combine the best properties of the super-twisting algorithm with adaptive gains and the generalized super-twisting algorithm. The proposed approach consists of using dynamically adapted control gains that ensure global finite-time convergence. The advantage of adaptive gains is that no conservative upper bound on the perturbations and control coefficients must be considered to maintain sliding. We prove that the resulting closed-loop system is globally finite-time stable. To demonstrate the effectiveness of the proposed adaptive generalized super-twisting algorithm, we use it to solve the trajectory tracking control problem for an articulated intervention-AUV in 6DOF. Additionally, we prove that the adaptive generalized super-twisting algorithm makes the tracking errors of the articulated intervention-AUV converge asymptotically to zero. We also present simulation and experimental results that validate and show the applicability of the proposed control law.

Finally, to utilize the redundancy of this highly flexible underwater vehicle, we propose a combined kinematic and dynamic control approach for vehicle-manipulator systems. The main idea is to combine the singularity-robust multiple task-priority framework with a robust sliding mode controller while simultaneously ensuring that the task errors remain bounded. *Any* controller can be used, as long as it is able to make the velocity vector converge to the velocity reference vector in finite time. The reference vector is chosen as the output from the singularity-robust multiple task-priority inverse kinematic controller. This novel approach allows us

to analyse the stability properties of the kinematic and dynamic subsystems together in the presence of model uncertainty while retaining the possibility of solving multiple tasks simultaneously. The finite-time convergence property of the sliding mode controller allows us to show that multiple set-point regulation tasks will converge asymptotically to zero without the strict requirement that the velocities be perfectly controlled. This novel approach thus avoids the assumption of perfect dynamic control that is common in kinematic stability analyses for robot manipulators and vehicle manipulator systems. We provide two examples of sliding mode controllers that are able to make the velocity vector converge to the velocity reference vector in finite time: a first-order sliding mode controller and the adaptive generalized super-twisting algorithm. The applicability of the proposed method is illustrated via a simulation study, where the primary task is to control the position and orientation of the centre link of an articulated intervention-AUV and the secondary tasks are to control the orientations of the front end and back end of the articulated intervention-AUV, and an experimental study, where the primary task is to control the position and orientation of the front end of the articulated intervention-AUV and the secondary task is to control the position and orientation of the back end of the articulated intervention-AUV.

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU). This work was conducted from August 2017 to 2020 at the Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics under the supervision of Professor Kristin Y. Pettersen and the co-supervision of Professor Jan Tommy Gravdahl.

Acknowledgements

This work was supported by the Research Council of Norway through its Centres of Excellence funding scheme, project no. 223254-NTNU AMOS.

The past three years during which the work for this thesis was conducted have gone by incredibly fast. It has been a great experience and I have learned a lot. I have had the opportunity to work with and get to know a lot of inspiring people, for which I am truly grateful.

First of all, I would like to thank my supervisors Professor Kristin Y. Pettersen and Professor Jan Tommy Gravdahl for giving me the opportunity to continue working on sliding mode control and articulated intervention-AUVs and for their scientific advice and excellent supervision during the past three years. Especially, I would like to thank Professor Kristin Y. Pettersen for letting me work independently but also providing new input and guidance when it was needed. She has also pushed me to come up with my own ideas and given me invaluable feedback on my work. In addition, I would like to thank Professor Jan Tommy Gravdahl for always having time for me, always being interested in what I was doing and giving me invaluable feedback. I would also like to thank him for motivating and encouraging me to take a PhD.

During the past three years I have also had the privilege to work alongside the most incredible colleagues. I would like to thank Dr Anna Kohl for welcoming me with open arms into the snake group and giving me invaluable support during my first experiments; they would truly not have been as successful without you. I would also like to thank Marianna Wrzos-Kaminska. It has been a pleasure being your office mate, and I am going to miss having an oracle of knowledge right behind me and also our daily discussions about everything and anything, both work related and non-work related. I would also like to thank you for invaluable support during my experiments. On that note, I would also like to thank the rest of the snake group, Jørgen Sverdrup-Thygeson, Henrik Schmidt-Didlaukies, Erlend A. Basso

and Carina Norvik, for the discussions and invaluable help during my experiments. I would also like to thank Dr Pål Liljeback for providing help and letting us use the Eelume facilities to check that everything was in order before the experiments. I would also like to thank the coffee break people and the Activity Thursdays people for the discussions, fun times and keeping me sane.

The environment that we are working in would certainly not be the same without the administrative and technical staff at ITK. It has been a privilege to be surrounded by so many friendly and competent people who have been incredibly supportive in every way. I want to thank the administrative personnel for making my life a lot easier during the past few years, especially Tove Kristin Blomset Johnsen. My thanks also go to Stefano Brevik Bertelli and Glenn Angell for their support during experiments, and the other technical staff at ITK for their support regarding IT and hardware.

Last but not least, I want to thank my family and friends for their unconditional support. I want to thank my new friends that I have made during my PhD, my friends obtained during my masters and my friends back home in Bergen, who always welcomed me back home as if no timed had passed. I would especially like to thank Malin and Ine who have taken the time to visit me multiple times in Trondheim, and Tuva and Hilde who have been invaluable in my spare time. I want to thank my parents, Eva and Thor-Arne, for the values with which they raised me, their support and encouragement. I also want to thank my brothers, Jan-Trygve and Ole-Henrik, who opened my eyes to the world of cybernetics, for motivating me to aim high and for all the hours you spent playing card games and strategic games with your little sister (even though you were getting crushed). I would also like to thank my aunts Berit and Janet and my uncles Johan-Fredrik and Rolf for their interests in my work and their support. I would also like to thank my nephew Thorbjørn for bringing so much joy to my life. Finally, I want to thank Andreas, for encouraging me to take on this journey, for the invaluable support during the stressful times and for always being there for me, I could not have done this without your support.

Contents

Summary i

Preface v

Contents vii

List of figures ix

List of tables xiii

Nomenclature xv

1 Introduction 1
1.1 Motivation . . . 1
1.2 Background and literature overview . . . 2
1.3 Contributions and organization of this thesis . . . 8

2 AIAUV: Mathematical Model, Simulation Model and Experimental Set-up 13
2.1 AIAUV model and problem statement . . . 13
2.2 Simulation model . . . 16
2.3 Experiments fall 2018 . . . 18
2.4 Experiments summer 2020 . . . 24

3 Tracking using the Super-Twisting Algorithm 29
3.1 The super-twisting algorithm with adaptive gains . . . 29
3.2 Tracking control laws . . . 30
3.3 Simulation results . . . 36
3.4 Experimental investigation . . . 40
3.5 Chapter summary . . . 43

4 Tracking using the Generalized Super-Twisting Algorithm 45
4.1 The generalized super-twisting algorithm . . . 46
4.2 Tracking control laws . . . 46
4.3 Simulation results . . . 56
4.4 Experimental investigation . . . 60
4.5 Chapter summary . . . 64

5 Comparison of Two Sliding Mode Control Algorithms **67**
5.1 PID controller . . . 68
5.2 Simulation results . . . 69
5.3 Experimental investigation . . . 75
5.4 Chapter summary . . . 82

6 The Generalized Super-Twisting Algorithm with Adaptive Gains **85**
6.1 Problem statement and main results . . . 86
6.2 Case Study: Articulated Intervention-AUV . . . 93
6.3 Experimental study: Articulated Intervention AUV . . . 97
6.4 Chapter summary . . . 112

7 Combined Kinematic and Dynamic Control **113**
7.1 Vehicle-manipulator model . . . 114
7.2 Inverse kinematic control . . . 114
7.3 Combined kinematic and dynamic stability analysis . . . 115
7.4 Simulation case study . . . 120
7.5 Experimental case study . . . 133
7.6 Chapter summary . . . 149

8 Conclusions and Future Work **151**

A Stability Definitions and Theorems **155**

List of figures

1.1 The Eelume vehicle inspecting (Courtesy: Eelume) 2
1.2 Overview of control structure for AIAUVs 3
1.3 Illustrating the reaching and sliding phase with $\sigma = x_1 + x_2$ (Credit: Arnt Erik Stene) 5

2.1 The Eelume 2016 vehicle (Courtesy: Eelume) 17
2.2 The Eelume 2020 vehicle (Courtesy: Eelume) 18
2.3 The Eelume 2016 vehicle with reflective markers attached to the base 19
2.4 Illustration of the system structure (experiments fall 2018) 20
2.5 Reference trajectories for position and orientation 21
2.6 Tracking results for a complete inspection 22
2.7 Configurations 22
2.8 Joint angles in the C-shape case 23
2.9 Joint angles in the C-shape with a moving head case 23
2.10 Joint angles in the I-shape case 24
2.11 The Eelume 2020 vehicle with reflective markers attached to the centre link 25
2.12 Illustration of the system structure (experiments summer 2020) 25
2.13 Reference trajectories for the inspection case 26
2.14 Reference trajectories for each task in the task-priority case 28

3.1 Optimal simulation results using the control law from Section 3.2.1 . . 37
3.2 Simulation results using the control law from Section 3.2.1 39
3.3 Simulation results using the control law from Section 3.2.2 39
3.4 Experimental results using the control law from Section 3.2.1 41
3.5 Experimental results using the control law from Section 3.2.2 42
3.6 Estimation errors for the HOSMO during C-shape (the first sub-plot shows $e_1 = x_1 - \hat{x}_1$ and the second sub-plot shows $e_2 = x_2 - \hat{x}_2$) . . . 43

4.1 Optimal simulation results using the control law from Section 4.2.1 . . 57
4.2 Simulation results using the control law from Section 4.2.1 59
4.3 Simulation results using the control law from Section 4.2.2 60
4.4 Error between desired control input and actual control input from the thruster allocation scheme 61
4.5 Experimental results using the control law from Section 4.2.1 62
4.6 Experimental results using the control law from Section 4.2.2 63

4.7 Estimation errors for the HOSMO during C-shape (the first sub-plot shows $e_1 = x_1 - \hat{x}_1$ and the second sub-plot shows $e_2 = x_2 - \hat{x}_2$) . . . 64

5.1 Simulation: Trajectory tracking results using the control laws without the HOSMO . . . 70
5.2 Simulation: Thruster forces used in the C-shape when the control laws without the HOSMO are used . . . 71
5.3 Simulation: Thruster forces used in the C-shape with a moving head when the control laws without the HOSMO are used . . . 71
5.4 Simulation: Trajectory tracking results using the control laws with the HOSMO . . . 73
5.5 Simulation: Thruster forces used in the C-shape when the control laws with the HOSMO are used . . . 74
5.6 Simulation: Thruster forces used in the C-shape with a moving head when the control laws with the HOSMO are used . . . 74
5.7 Experiments: Trajectory tracking results using the control laws without the HOSMO . . . 77
5.8 Experiments: Thruster forces used in the C-shape when the control laws without the HOSMO are used . . . 78
5.9 Experiments: Thruster forces used in the C-shape with a moving head when the control laws without the HOSMO are used . . . 78
5.10 Experiments: Trajectory tracking results using the control laws with the HOSMO . . . 79
5.11 Experiments: Thruster forces used in the C-shape when the control laws with the HOSMO are used . . . 80
5.12 Experiments: Thruster forces used in the C-shape with a moving head when the control laws with the HOSMO are used . . . 80

6.1 Tracking results using adaptive GSTA . . . 96
6.2 Sliding surface σ . . . 97
6.3 Control input . . . 98
6.4 Evolution of $k_1(t)$ over time . . . 98
6.5 Simulation results using the adaptive GSTA for the inspection case . . . 100
6.6 Simulation: Tracking errors for the inspection case . . . 102
6.7 Simulation: Thruster forces τ_{thr} and joint torques τ_q used for the inspection case . . . 103
6.8 Simulation: Evolution of the adaptive gains over time for the STA with adaptive gains for the inspection case . . . 104
6.9 Experimental results using the adaptive GSTA for the inspection case 106
6.10 Experiments: Tracking errors for the inspection case . . . 108
6.11 Experiments: Thruster forces τ_{thr} and joint torques τ_q used for the inspection case . . . 109
6.12 Experiments: Evolution of the adaptive gains over time for the STA with adaptive gains for the inspection case . . . 110

7.1 Results when the first-order SMC is used for the AIAUV described by Table 2.2 . . . 122

7.2 Results when the GSTA with adaptive gains is used for the AIAUV described by Table 2.2 . . . 124
7.3 Results when the PID controller is used for the AIAUV described by Table 2.2 . . . 125
7.4 Results when the feedback linearisation controller with thruster allocation is used for the AIAUV described by Table 2.2 . . . 126
7.5 Results when the feedback linearisation controller is used for the AIAUV described by Table 2.2 . . . 127
7.6 Results when the first-order SMC is used for the AIAUV described by Table 7.1 . . . 130
7.7 Results when the GSTA with adaptive gains is used for the AIAUV described by Table 7.1 . . . 131
7.8 Results when the feedback linearisation controller is used for the AIAUV described by Table 7.1 . . . 132
7.9 Simulation results using the adaptive GSTA for the task-priority case 135
7.10 Simulation: Evolution of the adaptive gains over time for the adaptive GSTA for the task-priority case . . . 136
7.11 Simulation: Tracking errors for the task-priority case . . . 138
7.12 Simulation: Thruster forces τ_{thr} and joint torques τ_q used for the task-priority case . . . 139
7.13 Simulation: Evolution of the adaptive gains over time for the STA with adaptive gains for the task-priority case . . . 140
7.14 Experimental results using the adaptive GSTA for the task-priority case 142
7.15 Experiments: Evolution of the adaptive gains over time for the adaptive GSTA for the task-priority case . . . 143
7.16 Experiments: Tracking errors for the task-priority case . . . 145
7.17 Experiments: Thruster forces τ_{thr} and joint torques τ_q used for the task-priority case . . . 146
7.18 Experiments: Evolution of the adaptive gains over time for the STA with adaptive gains for the task-priority case . . . 147

List of tables

2.1 Physical parameters used in the simulations 16
2.2 Eelume 2016 link properties 17
2.3 Eelume 2020 link properties 18

3.1 Simulation: Control gains for the STA with adaptive gains 38
3.2 Experiments: Control gains for the STA with adaptive gains 40

4.1 Simulations: Control gains for the GSTA 58
4.2 Experiments: Control gains for the GSTA 61

5.1 Simulation: Control gains for the PID controller 69
5.2 Simulation: Comparison of results when the control laws without the HOSMO are used 72
5.3 Simulation: Comparison of the thruster forces used when the control laws without the HOSMO are used 72
5.4 Simulation: Comparison of results when the control laws with the HOSMO are used 72
5.5 Simulation: Comparison of the thruster forces used when the control laws with the HOSMO are used 73
5.6 Experiments: Control gains for the PID controller 76
5.7 Experiments: Comparison of results when the control laws without the HOSMO are used 77
5.8 Experiments: Comparison of the thruster forces used when the control laws without the HOSMO are used 79
5.9 Experiments: Comparison of results when the control laws with the HOSMO are used 81
5.10 Experiments: Comparison of the thruster forces used when the control laws with the HOSMO are used 81

6.1 Control gains for the adaptive GSTA for the inspection case 99
6.2 Control gains for the STA with adaptive gains for the inspection case 101
6.3 Control gains for the GSTA for the inspection case 101
6.4 Control gains for the PID controller for the inspection case 101
6.5 Simulation: Comparison of the tracking results for the inspection case 104
6.6 Simulation: Comparison of the control inputs for the inspection case . 105
6.7 Experiments: Comparison of the tracking results for the inspection case 107
6.8 Experiments: Comparison of the control inputs for the inspection case 107

7.1 Eelume 2016 link properties (link lengths reduced by 20%) 133
7.2 Simulation: Control gains for the adaptive GSTA for the task-priority case 134
7.3 Simulation: Control gains for the STA with adaptive gains for the task-priority case 136
7.4 Simulation: Control gains for the GSTA for the task-priority case . . . 137
7.5 Simulation: Control gains for the PID controller for the task-priority case 137
7.6 Simulation: Comparison of the tracking results for the task-priority case 137
7.7 Simulation: Comparison of the control inputs for the task-priority case 137
7.8 Experiments: Control gains for the adaptive GSTA for the task-priority case 143
7.9 Experiments: Control gains for the STA with adaptive gains for the task-priority case 143
7.10 Experiments: Control gains for the GSTA for the task-priority case . . 144
7.11 Experiments: Control gains for the PID controller for the task-priority case 144
7.12 Experiments: Comparison of the tracking results for the task-priority case 144
7.13 Experiments: Comparison of the control inputs for the task-priority case 147

Nomenclature

Abbreviations

UV	Underwater Vehicle
ROV	Remotely Operated Vehicle
USR	Underwater Snake Robot
AUV	Autonomous Underwater Vehicle
AIAUV	Articulated Intervention Autonomous Underwater Vehicle
UVMS	Underwater Vehicle-Manipulator System
VMS	Vehicle-Manipulator System
IK	Inverse Kinematics
ODF	Operation Space Formulation
SRMTP	Singularity-Robust Multiple Task-Priority
SMC	Sliding Mode Control
STA	Super-Twisting Algorithm
GSTA	Generalized Super-Twisting Algorithm
SMO	Sliding Mode Observer
HOSMO	Higher-Order Sliding Mode Observer
RMSE	Root-Mean-Square Error
UGAS	Uniformly Globally Asymptotically Stable
GES	Globally Exponentially Stable
UGB	Uniformly Globally Bounded
FTS	Finite-Time Stable
GFT	Global Finite-Time
GFTS	Globally Finite-Time Stable
GUFTS	Globally Uniformly Finite-Time Stable
LFC	Lyapunov Function Candidate

Model parameters

n	Number of joints
m	Number of thrusters
C_a	Added mass coefficient for the cross section
C_{d_x}	Non-linear drag coefficient in surge
C_{d_ϕ}	Non-linear drag coefficient in roll
C_{d_c}	Non-linear crossflow drag coefficient
C_{d_l}	Linear cross-sectional drag coefficient
α	Added mass ratio in surge/heave for a link
β	Linear drag parameter in surge
γ	Linear drag parameter in roll

Model conventions

$q \in \mathbb{R}^{(n-1)}$	Vector representing the joint angles
$\tau(q) \in \mathbb{R}^{m+(n-1)}$	Vector representing the generalized forces
$\tau_{thr} \in \mathbb{R}^m$	Vector representing the thruster forces
$\tau_q \in \mathbb{R}^{(n-1)}$	Vector representing the joint torques
$\zeta \in \mathbb{R}^{6+(n-1)}$	Vector representing the body-fixed velocities
$v \in \mathbb{R}^3$	Vector representing the body-fixed linear velocities
$\omega \in \mathbb{R}^3$	Vector representing the angular velocities
$\dot{q} \in \mathbb{R}^{(n-1)}$	Vector representing the joint angle velocities
$\xi_q \in \mathbb{R}^{7+(n-1)}$	Complete state vector specifying the position, orientation, and shape of the AIAUV when quaternions are used
$\eta_1 \in \mathbb{R}^3$	Vector representing the position
$p \in \mathbb{R}^4$	Vector representing the unit quaternion describing the orientation in the inertial frame
$\xi_e \in \mathbb{R}^{6+(n-1)}$	Complete state vector specifying the position, orientation, and shape of the AIAUV when Euler angles are used
$\eta_2 \in \mathbb{R}^3$	Vector representing the Euler angles describing the orientation in the inertial frame

Control parameters

σ	Sliding surface
ε	Control gain for the adaptive gains
λ	Control gain for the adaptive gains
γ_1	Control gain for the adaptive gains
ω_1	Control gain for the adaptive gains
α_m	Design parameter for the adaptive gains α and k_1
L	HOSMO gain
k_1	Control gain for the GSTA
k_2	Control gain for the GSTA
β_{GSTA}	Control gain for the GSTA
k_p	Control gain for the PID controller
k_d	Control gain for the PID controller
k_i	Control gain for the PID controller
K_p	Gain matrix for Λ in sliding surface
K_q	Gain matrix for Λ in sliding surface
χ	Task variable for the SRMTP method
Λ_i	Gain matrix for the SRMTP method
K_d	Gain matrix for the first-order SMC
K	Gain matrix for the first-order SMC

Matrix and vector operators

$e_i \in \mathbb{R}^{1\times i}$	$1 \times i$ vector of ones
$R_I^B \in \mathbb{R}^{3\times 3}$	Rotation matrix expressing the transformation from the inertial frame to the body-fixed frame
$I_n \in \mathbb{R}^{n\times n}$	$(n \times n)$ identity matrix
$S(\cdot) \in \mathbb{R}^{3\times 3}$	Cross-product operator defined as in [30, Definition 2.2]
$J_{k,o} \in \mathbb{R}^{3\times 3}$	Jacobian matrix
$J_i^+ = J_i^T(J_i J_i^T)^{-1}$	Right Moore-Penrose pseudo-inverse of the task Jacobian

Chapter 1

Introduction

1.1 Motivation

Currently, subsea operations are often performed using remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) (Figure 1.1). ROVs are suited for short-term, short-distance inspection and intervention operations. These vehicles are typically human-operated from a surface support vessel; ROV operations are therefore expensive (100-200 kUSD/day) and have a large environmental footprint. Survey AUVs are hydro-dynamically efficient torpedo-shaped vehicles that are suited for long-distance operations without human supervision but usually cannot stand still (hover) or perform intervention operations. These vehicles also rely on a surface support vessel for energy provision. Articulated intervention-AUVs (AIAUVs) [51, 59] are novel bio-inspired marine robots that have a slender, multi-articulated body of an underwater snake robot (USR), which imparts the AIAUV with phenomenal accessibility and flexibility, but also multiple thrusters along its body. The thrusters enable the AIAUV to move forward without using an undulating gait pattern and provide the ability to hover. This property is especially important for station-keeping and trajectory tracking in narrow and confined spaces. These capabilities enable the AIAUV to operate as a floating base manipulator. Moreover, the AIAUV adopts the high-kinematic redundancy of the USR and the fully energy-efficient hydrodynamic properties and tether-less operation of AUVs. Compared to standard survey AUVs, AIAUVs have the advantage of full actuation and the ability to perform intervention tasks. Since an AIAUV can use its slender body to access narrow spaces, use its thrusters to keep itself stationary and then use its joints to perform intervention tasks, the AIAUV can exploit the full potential of the inherent kinematic redundancy. This ability has been addressed in detail in [77, 79]. Existing AIAUVs are either remotely controlled or used to perform mostly preplanned missions. However, the AIAUV concept is arguably the best foundation for the development of a truly autonomous and versatile underwater robot that can perform both observation and intervention operations in the same mission, e.g., mapping the seabed and collecting sediments, inspecting and repairing the net of an aquaculture fish cage, acting as a key instrument for the automation of sea weed production plants, and detecting and gathering plastic and

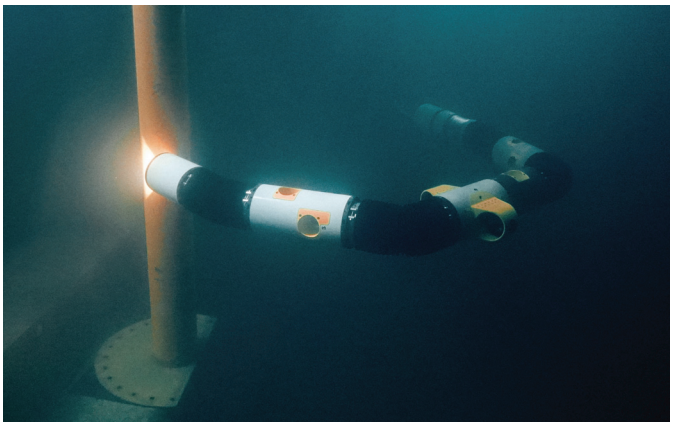


Figure 1.1: The Eelume vehicle inspecting (Courtesy: Eelume)

other debris polluting the oceans.

Station-keeping and trajectory tracking are essential for an AIAUV to be able to move in confined spaces and to perform intervention tasks. Since the AIAUV is a floating-base robot manipulator, standard control methods for fixed-base robot manipulators cannot be applied to the AIAUV. Moreover, the AIAUV is subject to hydrodynamic and hydrostatic parameter uncertainties, uncertain thruster characteristics, unknown disturbances, and unmodelled dynamics. Furthermore, since the coupling forces caused by the manipulator joint motion are even larger for the AIAUV than for ROVs because the AIAUV has no separate vehicle base while ROVs have a separate base with a larger inertia than that of the manipulator arm, it is essential for the control approach to be robust. The design of a robust trajectory tracking controller for the AIAUV is therefore the objective of this thesis. Sliding mode control (SMC) is a robust and versatile non-linear control approach that is particularly well suited for situations where unknown non-linearities affect the system, as in the case of AIAUVs. In this thesis we therefore investigate using SMC to obtain a robust trajectory tracking controller. The proposed methods are theoretically analysed, and simulation and experimental results that demonstrate their performance are presented.

1.2 Background and literature overview

This section provides an introduction to previous research that is relevant for this thesis. First, an overview of control methods used for AIAUVs is given. Then, some background on SMC is presented.

1.2.1 Control of the articulated intervention AUV

Control of AIAUVs or UVMSs (the class of vehicles that the AIAUV is a subgroup of) has usually been divided into two parts: motion planning and dynamic control (Figure 1.2). In this section, some previous related research on both motion

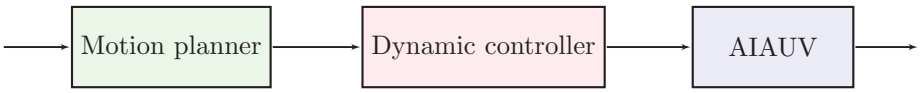


Figure 1.2: Overview of control structure for AIAUVs

planning and dynamic control is presented, although this thesis focuses mainly on dynamic control.

Motion planning

Motion planning is required to enable the AIAUV to move efficiently to solve the mission tasks and simultaneously assure the safety of the system. AIAUVs are redundant, as the many joints and thrusters provide more degrees of freedom than the mission task requires. Thus, the motion planning algorithm must include redundancy resolution. Methods have been developed for fixed-base robots and floating-base robots with a large base. The existing methods for motion planning with redundancy resolution include task-priority inverse kinematics (IK) control and operational space formulation (OSF). The two main classes of IK are based on [20] and [71]. While many of the derived methods have no stability analysis, the stability properties of the singularity-robust multiple task-priority (SRMTP) method based on [20] are analysed in [3]. Autonomous robots are safety-critical systems, and safety-related tasks are inherently set-based [61]. Set-based tasks do not fit naturally into the original task-priority IK framework. The authors in [54] and [55] extend the SRMTP method to include set-based tasks using hybrid systems theory. The authors in [73] add set-based tasks to a framework derived from [71]. For AIAUVs specifically, [77] applies the SRMTP method, and [78] applies the theory from [54]. The existing IK methods do not work well for AIAUVs: one main reason is that the dynamics are not considered. A fundamental assumption of IK is that the kinematics and dynamics can be decoupled; this assumption holds for ROVs with a heavy base and fixed-base robots. Furthermore, when analysing the resulting stability properties, it is common to neglect the dynamics and assume that the reference output is tracked perfectly by a dynamic controller. This approach makes sense for fixed-base manipulators or for VMS, where the mass of the vehicle is much larger than the mass of the manipulator arm such that joint motion does not have a significant impact on the overall motion of the whole vehicle. For AIAUVs, the coupling forces caused by joint motion are too large to rely on this assumption. Inertia, drag forces, and restoring forces and moments caused by joint motion have a significant effect on the overall motion of the whole mechanism, for which any joint motion leads to a corresponding change in the direction of the thrusters that are distributed along the robot. For this type of system, the stability properties of the kinematic and dynamic subsystems must be analysed together since the assumption that the references are perfectly tracked, on which the IK approach is based, is not valid for AIAUVs. Therefore, existing IK methods are not suitable for AIAUV motion planning.

In contrast to IK, OSF directly assigns a control torque to the redundant manipulator [23, 41]. In [7], a novel task-priority framework based on a hierarchy of control

Lyapunov function and a control barrier function based on quadratic programs are presented. The proposed method guarantees strict priority among different groups of tasks, such as safety-related, operational and optimization tasks. Moreover, a soft priority measure in the form of penalty parameters can be employed to prioritize tasks at the same priority level. In contrast to kinematic control schemes, the proposed framework is a holistic approach to the control of redundant robotic systems that solves the redundancy resolution, dynamic control and control allocation problems simultaneously. However, a drawback of OSF is its dependency on accurate knowledge of the dynamic model. This dependence was investigated in [6], where the effect of ROV mass matrix uncertainties when using OSF was analysed. In addition to these uncertainties, AIAUVs are strongly affected by forces with large uncertainties, for example, hydrodynamic parameter uncertainties, uncertain thruster characteristics, and unknown ocean current disturbances. Existing OSF methods are therefore not suitable for AIAUV motion planning.

Dynamic control

There are many similarities between AIAUVs, robot manipulators, USRs and more conventional underwater vehicles (UVs). The AIAUV can be seen as a free-floating robot manipulator since it is equipped with actuators to control the overall position and orientation. Modelling and control of traditional fixed-base robots is presented in textbooks on robotics, such as [70, 72, 76]. However, since the AIAUV is a floating-base robot manipulator, standard control methods for fixed-base robot manipulators cannot be applied to AIAUVs. Another type of system characterized as a free-floating robot manipulator is spacecraft manipulator systems, for which the only significant external forces are actuator forces. A review of such systems is given in [24]. These methods are not directly applicable since the AIAUV is also subject to hydrodynamic and hydrostatic parameter uncertainties and external disturbances such as currents. The AIAUV has the slender, multi-articulated body of a USR while also possessing hovering and intervention capabilities because of the thrusters. Path following and manoeuvring control of USRs are investigated in [36, 38, 42–45]; however, we do not want to apply these methods directly, as we would then not be able to utilize the AIAUV's hovering and intervention capabilities. Some methods for dynamic control of single-body UVs are presented in [30]. If the UV is equipped with a manipulator arm, we face the additional challenge of handling the interaction forces, also known as coupling forces, between the arm and the vehicle. Several different control approaches that address these challenges are summarized and presented in [4]. However, [4] considers mostly ROVs. Since the coupling forces caused by the manipulator joint motion are even larger for AIAUVs than for ROVs because the AIAUV has no separate vehicle base while ROVs have a separate base with large inertia compared to the manipulator arm, these methods cannot be directly applied.

For the AIAUV in particular, some methods for propulsion have been proposed in [37, 39], and energy-efficient methods for long-distance travel have been proposed in [83]. In [77, 79], a control framework for AIAUVs that uses IK and a dynamic controller is proposed. However, in both papers, IK is the main focus. The dynamic controller proposed in [77] is only a simple P-controller, and the

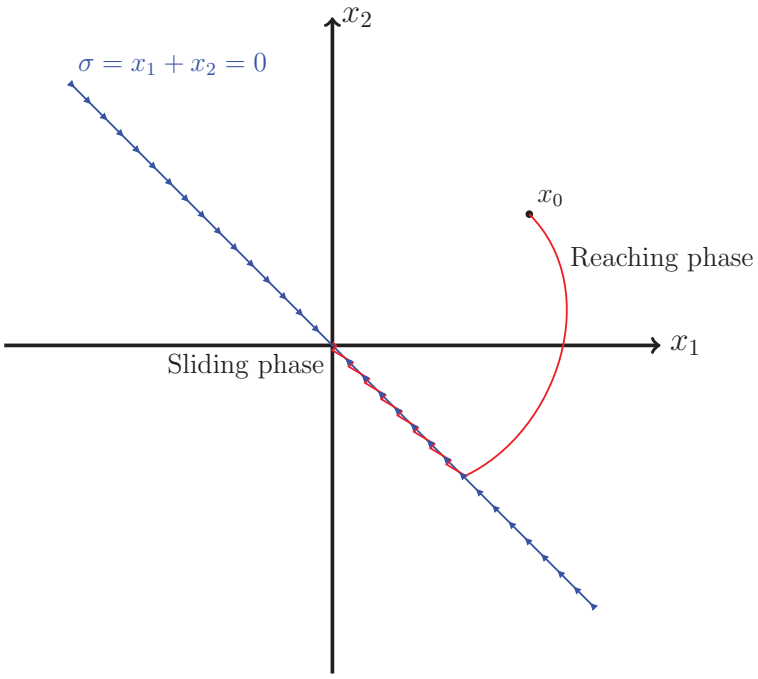


Figure 1.3: Illustrating the reaching and sliding phase with $\sigma = x_1 + x_2$ (Credit: Arnt Erik Stene)

controller proposed in [79] is a feedback-linearisation controller combined with a velocity-based PID controller. Moreover, none of these methods focus on robust control, which is essential for AIAUVs.

1.2.2 Sliding mode control

SMC systems are designed to drive system state trajectories onto a particular surface in the state space, named the sliding surface σ, in finite time. Once the sliding surface is reached, SMC keeps the states in the close neighbourhood of the sliding surface for all future time. Hence, SMC is a two-part controller design. The first part involves the design of a sliding surface, which should be designed such that when the sliding variable σ goes to zero, the state variables asymptotically converge to zero. The second part is concerned with selecting a control law that makes the sliding surface converge to zero in finite time. Thus, SMC can be divided into two phases, as shown in Figure 1.3. The first phase is called the reaching phase, where the state trajectory is driven towards the sliding surface, and the second phase is called the sliding phase, where the state trajectory moves towards the origin along the sliding surface.

SMC is a robust and versatile non-linear control approach that is particularly well suited for controlling perturbed control systems, specifically, control systems perturbed by matched uncertainties and disturbances [80]. We achieve these proper-

ties using a discontinuous control law. The discontinuous element provides robustness but also introduces chattering, i.e., high-frequency switching in the control input. Chattering can be avoided by using a saturation or sigmoid function instead of the discontinuous signum function [17, 74]. We then achieve continuous control input, but we restrict our sliding system's trajectory to a boundary around the sliding surface, thereby losing robustness to disturbances. However, chattering can also be avoided via higher-order sliding mode (HOSM) techniques [49, 50, 68]. We thus achieve a continuous control input without loss of robustness. The HOSM methods drive the sliding variable and its derivatives to zero in the presence of disturbances and uncertainties [69].

The super-twisting algorithm (STA) [47] is one of the most powerful second-order continuous SMC algorithms. The STA attenuates chattering by introducing a dynamic extension to the system such that the discontinuous term is hidden behind an integrator. In this way, the STA generates a smooth continuous control input that drives the sliding variable and its derivatives to zero in finite time in the presence of smooth matched disturbances with a bounded gradient. The main drawback of this approach is that the boundaries of the disturbance gradient must be known. This boundary is not always easily estimated, which often leads to overestimation that leads to unnecessarily large control gains. Therefore, in [69], an STA with adaptive gains was proposed. The approach continuously drives the sliding variable and its derivatives to zero in the presence of a bounded disturbance with unknown boundary, such that no conservative upper bound on the disturbance gradient has to be considered to maintain sliding because of the adaptive gains.

In recent years, various Lyapunov functions have been designed to obtain convergence conditions and estimates of the reaching time. However, these Lyapunov proofs are made under conservative assumptions. For example, the perturbations are dependent only on time [47, 58], the control coefficient is known [33, 35, 47, 58], or perturbations are dependent on state and time, but it is supposed that their total time derivative, i.e., the control signal, is a priori bounded by some constant [33, 66]. Therefore, [18] proposed a generalized super-twisting algorithm (GSTA) for a more general scenario, i.e., the case when both the perturbations and control coefficients are state- and time-dependent and the control coefficients are uncertain. This approach provides some additional theoretical properties over the regular STA proposed in [47]. However, in this case, we also need to know the boundaries of the perturbations and control coefficients to obtain bounds on the control gains that are not excessively conservative.

In [81] and [82], two adaptive GSTAs are proposed as alternatives to the GSTA proposed in [56] for SISO and MIMO systems, respectively. However, these two methods do not utilize the fact that the GSTA can handle unknown control coefficients. In [34], a third alternative adaptive GSTA is proposed for an underwater vehicle; however, this method does not utilize the fact that the GSTA can handle state-dependent disturbances.

Sliding mode control for underwater vehicles

In recent years, several results have been reported regarding the use of SMC for many complex dynamic systems. For UVs, in general, some relevant contributions

are as follows. In [5], an SMC approach for the regulation problem of an UVMS is developed. This control law is inspired by [27]; it avoids the inversion of the system Jacobian and is therefore singularity free. In [28] and [31], SMC is used to address input uncertainty due to partially known non-linear thruster characteristics. In [75], chattering-free SMC is proposed for trajectory control. The chattering-free approach is developed by combining SMC with adaptive PID controller gains and implementing an adaptive update of the upper bounds on the disturbance and parameter uncertainties. In [22], SMC is used to address the coupling effects between a manipulator and an UV. A combination of virtual velocity control and SMC is used to develop a hybrid control strategy in [87] for the trajectory tracking of an unmanned UV. In [84], the trajectory tracking problem of an under-actuated unmanned UV is studied by combining backstepping and SMC. An attitude controller for an AUV is designed in [21] using a sliding-mode-based adaptive controller. In [52], the trajectory tracking problem for an UV subject to unknown system uncertainties and time-varying external disturbances is considered, and the problem is solved using a non-linear disturbance observer-based backstepping finite-time SMC scheme. In [62], SMC is applied to land-based snake robots to achieve robust tracking of a desired gait pattern and under-actuated straight-line path following.

Other sliding mode control approaches

Many different SMC approaches exist; the ones mentioned above are the most relevant for this thesis. However, many other promising SMC approaches have not been mentioned. In this section, some of these approaches will be mentioned, and some examples of each will be given.

Integral SMC is one approach that produces promising results. SMC that retains the order of the compensated system's dynamics in the sliding mode is called integral SMC [67, Definition 1.7]. Integral SMC can compensate for disturbances without the presence of a reaching phase. This feature is desirable, as the system's compensated dynamics can become insensitive to matched disturbances and uncertainties when the sliding surface is reached and sliding motion is established [67]. In [63], an integral SMC approach is proposed to reduce the disturbance terms that act on non-linear systems with state-dependent drift and input matrix. The general case of both matched and unmatched disturbances affecting the system is addressed. In [26], a switching structure scheme for motion control of industrial robots manipulators was proposed. The scheme implements both a decentralized approach and an inverse dynamic-based centralized approach. In both cases, integral SMC is used to compensate matched disturbances and to estimate the unmodelled dynamics used for the switching decision mechanism.

A newer class of approaches is terminal SMC and non-singular terminal SMC. Terminal SMC includes SMC approaches where not only the sliding surface converges to zero in finite time but also the tracking errors converge to zero or a bounded set in finite time. In [86], the terminal SMC proposed for rigid robotic manipulators provides faster and higher-precision tracking than conventional continuous SMC methods. In [85], a non-singular terminal SMC is proposed for non-linear dynamic systems. This approach avoids the possible singularity during the control phase, which is a problem with the terminal SMC method. In [64], a robust

adaptive non-singular terminal SMC is proposed. The proposed method guarantees that the tracking errors converge to a bounded set in finite-time, even under unmodelled dynamics and external disturbances.

SMC introduces considerable energy into the system, and it is often necessary to minimise or limit the control input since physical systems have a limited amount of thrust or power. Methods that combine optimization with SMC have therefore been introduced to achieve energy-efficient robust control. For example, [25] propose an optimization-based adaptive SMC. The aim of the approach is to reduce the control effort while maintaining all the positive aspects of finite-time convergence and robustness to match uncertainty.

1.3 Contributions and organization of this thesis

In this section, the scope, contributions and outline of this thesis are presented.

1.3.1 Scope

The scope of this thesis, entitled "Robust Control of Articulated Intervention-AUVs Using Sliding Mode Control", is to solve the trajectory tracking problem for an AIAUV in 6DOF. This thesis is concerned with the theoretical development of a trajectory tracking controller for an AIAUV in 6DOF without using any or as little model knowledge as possible since the AIAUV is subject to hydrodynamic and hydrostatic parameter uncertainties, uncertain thruster characteristics, unknown disturbances, and unmodelled dynamics, i.e., the control approach should be robust. To achieve this goal, robust algorithms based on SMC are proposed. The stability properties of the proposed control approaches are analysed theoretically, and their performance is tested in simulations and experiments.

1.3.2 Contributions and outline

In this thesis, various SMC algorithms are proposed for the trajectory tracking problem for an AIAUV in 6DOF. We consider both the case where velocity measurements are available and the case where they are not. When velocity measurements are not available, we use a higher-order sliding mode observer (HOSMO) to estimate the linear and angular velocities. Moreover, a novel GSTA with adaptive gains that is particularly well suited for control of the AIAUV is developed. Additionally, we combine the SRMTP method with different SMC approaches that are finite-time stable (FTS, Definition A.7). The stability properties of the complete systems are analysed theoretically, and their performance is tested in simulations and experiments. The particular contributions and methods of the single chapters are outlined in the following.

Chapter 2

This chapter presents the AIAUV model represented in terms of both quaternions and Euler angles and the corresponding tracking control problem for each model.

We also present the simulation models used in this thesis and the experimental set-up used in experiments during the fall of 2018 and the summer of 2020.

Chapter 3

This chapter investigates the use of the STA with adaptive gains to solve the trajectory tracking problem for an AIAUV. The tracking problem is also solved when velocity measurements are not available by using the STA with adaptive gains in combination with a HOSMO. The closed-loop system is then analysed.

The STA with adaptive gains, proposed by [69], is the most powerful second-order continuous SMC algorithm, as it attenuates chattering and does not require the consideration of any conservative upper bound on the disturbance to maintain sliding because of the adaptive gains. Motivated by this approach, we solve the trajectory tracking control problem of an AIAUV in 6DOF using the STA with adaptive gains. We also solve the trajectory tracking control problem in the case that velocity measurements are not available by using the STA with adaptive gains in combination with a HOSMO [46]. We choose to use the HOSMO because of its strong stability. Specifically, the finite-time stability of the HOSMO enables us to show that the closed-loop system is uniformly globally asymptotically stable (UGAS, Definition A.2). The HOSMO is then used to estimate the linear and angular velocities. Additionally, we prove the asymptotic convergence of the tracking errors for the control law using the STA with adaptive gains in combination with the HOSMO. Finally, we demonstrate the applicability of the presented control laws with comprehensive simulation and experimental results. The purpose of the experiments is to validate the theory and the robustness of the control approaches by showing that the proposed approaches work in experiments and not only in the ideal case presented by simulations. This argument holds for both the STA with adaptive gains and the HOSMO, as, to the best of the author's knowledge, few experimental results exist for these methods.

Chapter 4

This chapter investigates the use of the GSTA to solve the trajectory tracking problem for an AIAUV. The tracking problem is also solved in the case that velocity measurements are not available by using the GSTA in combination with a HOSMO. The closed-loop systems are then analysed.

The GSTA, proposed by [18], is an extension of the STA that provides finite-time convergence in the case where both the perturbations and control coefficients are state- and time-dependent and the control coefficients are uncertain, which is essential for robust control of an AIAUV. Motivated by this scenario, we solve the trajectory tracking control problem of an AIAUV in 6DOF using the GSTA. Furthermore, we prove the asymptotic convergence of the tracking errors. Additionally, we prove that the GSTA is actually globally uniformly finite-time stable (GUFTS, Definition A.8), while in [18], the GSTA was proven to be only globally finite-time stable (GFTS, Definition A.7). We also consider the case where velocity measurements are unavailable by solving the tracking control problem using the GSTA in combination with a HOSMO [46]. The reason we choose to use the HOSMO is its

strong stability. Specifically, the finite-time stability of the HOSMO enables us to show that the closed-loop system is UGAS. Furthermore, we prove the asymptotic convergence of the tracking errors when the HOSMO is included. We also present comprehensive simulation and experimental results that validate and illustrate the applicability of both control schemes. The purpose of the experiments is to validate the theory and the robustness of the control approaches by showing that the proposed approaches perform well in experiments and not only in the ideal case presented by simulations. This argument holds for both the GSTA and the HOSMO, as, to the best of the author's knowledge, few experimental results exist for these methods.

Chapter 5

In this chapter, we compare the previously proposed tracking control laws presented in Chapter 3 and Chapter 4. Specifically, we compare the performance of the control law using the STA with adaptive gains presented in Section 3.2.1 and the control law using the GSTA presented in Section 4.2.1. We also compare the control scheme in which the STA with adaptive gains is combined with a HOSMO, which is presented in Section 3.2.2, with the control scheme in which the GSTA is combined with a HOSMO, which is presented in Section 4.2.2. In addition, we solve the tracking problem using a standard PID controller to evaluate how the SMC algorithms perform compared to a linear controller. The main objective and contribution of this chapter is the comparison between the STA with adaptive gains, the GSTA and a PID controller.

Chapter 6

In this chapter, we propose an adaptive GSTA for a class of systems whose perturbations and uncertain control coefficients are time- and state-dependent, i.e., we combine the best properties of the STA with adaptive gains [69] (used in Chapter 3) and the GSTA [18] (used in Chapter 4). The proposed approach consists of using dynamically adapted control gains that ensure global finite-time (GFT) convergence. The advantage of adaptive gains is that no conservative upper bound on the perturbations and control coefficients must be considered to maintain sliding. We prove that the resulting closed-loop system is GFTS.

In Chapter 3, we investigate the use of the STA with adaptive gains to control an AIAUV, and in Chapter 4, we investigate the use of the GSTA to control an AIAUV. In Chapter 5, we compare the two SMC algorithms via both simulations and experiments, and we observe that the STA with adaptive gains produces better tracking results than does the GSTA, even though the GSTA has better theoretical properties as it is proven to provide GFT stability for a larger class of systems. The adaptive gains are thus seen to be very practical and provide tuning advantages. Therefore, we find it desirable to combine the practical advantages of the adaptive gains with the theoretical advantages of the GSTA to control an AIAUV. A novel adaptive GSTA is thus proposed for a class of systems whose perturbations and uncertain control coefficients are time- and state-dependent. The proposed approach consists of using dynamically adapted control gains in a GSTA, which ensures GFT

convergence. A non-smooth strict Lyapunov function is used to obtain the conditions for the GFT stability. To illustrate the effectiveness of the proposed adaptive GSTA, we use the adaptive GSTA to solve the trajectory tracking control problem of an AIAUV in 6DOF. Additionally, we show that the adaptive GSTA makes the tracking errors of the AIAUV converge asymptotically to zero. We also present simulation and experimental results for the AIAUV that validate and demonstrate the applicability of the proposed control law.

Chapter 7

In this chapter, a novel combined kinematic and dynamic control approach is proposed for vehicle-manipulator systems (VMSs). The main idea is to combine the SRMTP framework with a robust SMC while simultaneously ensuring that the task errors remain bounded. This novel approach allows us to analyse the stability properties of the kinematic and dynamic subsystems together in the presence of model uncertainty while retaining the possibility of solving multiple tasks simultaneously. We present two examples of robust SMC that satisfy the required condition of making the velocity vector converge to the velocity reference vector in finite time.

Specifically, we combine the SRMTP framework [20] with a robust SMC while simultaneously ensuring that the task errors remain bounded. The kinematic stability analysis of the SRMTP method is based on the results in [3]. *Any* SMC can be used as long as it is able to make the velocity vector converge to the velocity reference vector in finite time. The reference vector is chosen as the output from the SRMTP inverse kinematic controller. The finite-time convergence property of the SMC enables us to show that the multiple set-point regulation tasks will converge asymptotically to zero without the strict requirement that the velocities are perfectly controlled. This novel approach thus avoids the assumption of perfect dynamic control that is common in kinematic stability analyses for vehicle manipulators. We provide two examples of SMCs that can make the velocity vector converge to the velocity reference vector in finite time: a first-order SMC and the GSTA with adaptive gains proposed in Chapter 6. We choose these two SMCs because we want to demonstrate how both a basic SMC, i.e., a first-order SMC, and a higher-order SMC fit into the control scheme. Part of the analysis of the first-order SMC is based on [5], who consider a set-point regulation problem for a UVMS. The control law proposed in [5] avoids the inversion of the Jacobian system, thus overcoming the occurrence of kinematic singularities, but the inverse kinematic problem is not considered. In this chapter, we extend the analysis in [5] to show the finite-time convergence of the velocity controller. A simulation and experimental study for the AIAUV are performed to demonstrate the effectiveness of the proposed method.

Chapter 8

This chapter presents concluding remarks and suggestions for future research.

1.3.3 Underlying publications

The underlying studies of this thesis have resulted in the following list of publications, including several journal articles and several peer-reviewed conference papers.

Journal articles

- I.-L. G. Borlaug, J. T. Gravdahl, J. Sverdrup-Thygeson, K. Y. Pettersen, and A. Loría. Trajectory tracking for underwater swimming manipulator using a super twisting algorithm. *Asian Journal of Control*, 21(1):208–223, 2019.
- I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Tracking control of an articulated intervention autonomous underwater vehicle in 6DOF using generalized super-twisting: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 2020.
- I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Comparison of two second-order sliding mode control algorithms for an articulated intervention AUV: Theory and experimental results. *Submitted to Ocean Engineering,* 2020.
- I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Combined kinematic and dynamic control of vehicle-manipulator systems. *Mechatronics*, 69:102380, Aug. 2020.
- I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. The generalized super-twisting algorithm with adaptive gains. *Submitted to International Journal of Robust and Nonlinear Control*, 2020.

Peer-reviewed conference papers

- I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Trajectory tracking for an articulated intervention AUV using a super-twisting algorithm in 6DOF. *IFAC PapersOnLine*, 51(29):311–316, Sep. 10-12, 2018. In: Proc. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, Opatija, Croatia.
- I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Tracking control of an articulated intervention AUV in 6DOF using the generalized super-twisting algorithm. In *Proc. American Control Conference*, pages 5705–5712, Philadelphia, USA, Jul. 10-12, 2019.
- I.-L. G. Borlaug, J. Sverdrup-Thygeson, K. Y. Pettersen, and J. T. Gravdahl. Combined kinematic and dynamic control of an underwater swimming manipulator. *IFAC PapersOnLine*, 52(21):8–13, Sep. 18-20, 2019. In: Proc. 12th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, Daejeon, Korea.
- I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. The generalized super-twisting algorithm with adaptive gains. In *Proc. European Control Conference*, pages 1624–1631, Saint Petersburg, Russia, Mai 12-15, 2020. Best Student Paper Award Finalist.

Chapter 2

AIAUV: Mathematical Model, Simulation Model and Experimental Set-up

This chapter presents the AIAUV model represented with both quaternions and Euler angles and the corresponding tracking problem for each model. We also present the simulation model used in this thesis and the experimental set-up used in the experiments conducted during the fall of 2018 and the summer of 2020.

2.1 AIAUV model and problem statement

In this section, we provide the model and the mathematical definition of the tracking control problem for the AIAUV. The AIAUV consists of n links connected by $n-1$ motorized joints. Each joint is treated as a one-dimensional Euclidean joint. To follow the convention used in the robotics community, the first link is referred to as the base of the manipulator. The base link is link 1, and the front link, where the end-effector is positioned, is link n. The joints are numbered from $i = 1$ to $n-1$ such that link i and link $i+1$ are connected by joint i. Furthermore, the AIAUV is equipped with m thrusters, including one or more thrusters acting along the body of the AIAUV to provide forward thrust and tunnel thrusters acting through the links to provide station-keeping capability. The AIAUV is considered to be a floating base manipulator operating in an underwater environment subject to added mass forces, dissipative drag forces, and gravity and buoyancy forces. This allows us to model the AIAUV as an UVMS, with dynamic equations given in matrix form by [4, 32],

$$M(q)\dot{\zeta} + C(q,\zeta)\zeta + D(q,\zeta)\zeta + g(q, R_B^I) = \tau(q) \tag{2.1}$$

where $q \in \mathbb{R}^{(n-1)}$ is the vector representing the joint angles, $M(q)$ is the inertia matrix including added mass terms, $C(q,\zeta)$ is the Coriolis-centripetal matrix, $D(q,\zeta)$ is the damping matrix, and $g(q, R_B^I)$ is the matrix of gravitational and buoyancy

forces. The control input is given by the generalized forces $\tau(q)$:

$$\tau(q) = \begin{bmatrix} T(q) & 0_{6\times(n-1)} \\ 0_{(n-1)\times m} & I_{(n-1)\times(n-1)} \end{bmatrix} \begin{bmatrix} \tau_{thr} \\ \tau_q \end{bmatrix} \tag{2.2}$$

where $T(q) \in \mathbb{R}^{6\times m}$ is the thruster configuration matrix, $\tau_{thr} \in \mathbb{R}^m$ is the vector of thruster forces, and $\tau_q \in \mathbb{R}^{(n-1)}$ represents the joint torques. To implement the control input $\tau(q)$, a thruster allocation scheme as proposed in [79] needs to be implemented to distribute the desired control inputs onto the m thrusters. The vector of body-fixed velocities, ζ, is defined as

$$\zeta = \begin{bmatrix} v^T & \omega^T & \dot{q}^T \end{bmatrix}^T \in \mathbb{R}^{6+(n-1)} \tag{2.3}$$

where v and ω are the body-fixed linear and angular velocities of the base of the AIAUV, respectively, and $\dot{q}$ is the vector of joint angle velocities. The desired velocities are denoted as

$$\zeta_d = \begin{bmatrix} v_d^T & \omega_d^T & \dot{q}_d^T \end{bmatrix}^T \tag{2.4}$$

in the body-fixed frame. To represent the orientation of the AIAUV, both Euler angles and quaternions can be used.

2.1.1 Model representation using quaternions

In this section, we will describe the model using quaternions. When we use quaternions to represent the model, we avoid the singularities in the Jacobian matrix that arise from the Euler angles. We therefore obtain the advantage of a well-defined Jacobian. The complete state vector specifying the position, orientation, and shape of the AIAUV when quaternions are used is defined as

$$\xi_q = \begin{bmatrix} \eta_1^T & p^T & q^T \end{bmatrix}^T \quad \in \mathbb{R}^{7+(n-1)} \tag{2.5}$$

where $\eta_1 = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$ is the position of the base and $p = \begin{bmatrix} \varepsilon^T & \eta \end{bmatrix}^T = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \varepsilon_3 & \eta \end{bmatrix}^T \in \mathbb{R}^4$ is the unit quaternion describing the orientation of the base in the inertial frame. The relationship between the body-fixed velocities and the complete state vector specifying the position, orientation, and shape of the AIAUV is given by the differential equation

$$\dot{\xi}_q = J_q(p)\zeta = \begin{bmatrix} R_B^I(p) & 0_{3\times3} & 0_{3\times(n-1)} \\ 0_{4\times3} & J_{k,oq}(p) & 0_{4\times(n-1)} \\ 0_{(n-1)\times3} & 0_{(n-1)\times3} & I_{(n-1)\times(n-1)} \end{bmatrix} \zeta \tag{2.6}$$

where R_I^B is the rotation matrix expressing the transformation from the inertial frame to the body-fixed frame and $J_{k,oq}(p) = \frac{1}{2}\begin{bmatrix} \eta I_3 + S(\varepsilon) \\ -\varepsilon^T \end{bmatrix}$, where I_3 is the (3×3) identity matrix and $S(\cdot)$ is the cross-product operator defined as in Definition A.1. By defining the desired trajectories as

$$\xi_{q,d} = \begin{bmatrix} \eta_{1,d}^T & p_d^T & q_d^T \end{bmatrix}^T \tag{2.7}$$

where $p_d = \begin{bmatrix} \varepsilon_d^T & \eta_d \end{bmatrix}^T$, the orientation error, which is computed via the composition (quaternion product) [4], is then given by

$$\tilde{p} = \begin{bmatrix} \tilde{\varepsilon} \\ \tilde{\eta} \end{bmatrix} = \begin{bmatrix} \eta\varepsilon_d - \eta_d\varepsilon + S(\varepsilon_d)\varepsilon \\ \eta\eta_d + \varepsilon^T\varepsilon_d \end{bmatrix} \tag{2.8}$$

The complete tracking error can then be defined as

$$\tilde{\xi}_q = \begin{bmatrix} \tilde{\eta}_1 \\ \tilde{\varepsilon} \\ \tilde{q} \end{bmatrix} = \begin{bmatrix} \eta_1 - \eta_{1,d} \\ \eta\varepsilon_d - \eta_d\varepsilon + S(\varepsilon_d)\varepsilon \\ q - q_d \end{bmatrix} \tag{2.9}$$

Remark 2.1. Note that for the orientation, the goal is to ensure that $p = \pm p_d$, which corresponds to $\tilde{p} = \begin{bmatrix} 0_{1\times 3} & \pm 1 \end{bmatrix}^T$. Now, since the Euler parameters satisfy $\eta^2 + \varepsilon^T\varepsilon = 1$, it is sufficient to make $\tilde{\varepsilon} \to 0$ because then $\tilde{p} = \begin{bmatrix} 0_{1\times 3} & \pm 1 \end{bmatrix}^T$. Therefore, $\tilde{\eta}$ is not included as an independent state in (2.9).

The goal of the tracking problem is to make the error vector (2.9) converge to zero. The tracking control objective is therefore to make $(\tilde{\xi}_q, \tilde{\zeta}) = (0, 0)$ an asymptotically stable equilibrium point of (2.1) and (2.6), which will ensure that the tracking error will converge to zero.

2.1.2 Model representation using Euler angles

In this section, we will describe the model using Euler angles. A well-known problem when we use the Euler angles is that the Jacobian can become singular at $\theta = \pm\pi/2$ (xyz-convention); however, if we stay away from that angle, the Jacobian is well defined and the inverse of the Jacobian matrix can be used. The complete state vector specifying the position, orientation, and shape of the AIAUV when Euler angles are used is defined as

$$\xi_e = \begin{bmatrix} \eta_1^T & \eta_2^T & q^T \end{bmatrix}^T \quad \in \mathbb{R}^{6+(n-1)} \tag{2.10}$$

where $\eta_1 = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$ is the position of the base and $\eta_2 = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \in \mathbb{R}^3$ are the Euler angles describing the orientation of the base in the inertial frame. The relationship between the body-fixed velocities and the complete state vector is given by the differential equation

$$\dot{\xi}_e = J_e(\eta_2)\zeta = \begin{bmatrix} R_B^I & 0_{3\times 3} & 0_{3\times(n-1)} \\ 0_{3\times 3} & J_{k,o}^{-1} & 0_{3\times(n-1)} \\ 0_{(n-1)\times 3} & 0_{(n-1)\times 3} & I_{(n-1)\times(n-1)} \end{bmatrix} \zeta \tag{2.11}$$

where $J_{k,o}$ is the Jacobian matrix. By defining the desired trajectory as $\xi_{e,d} = \begin{bmatrix} \eta_{1,d}^T & \eta_{2,d}^T & q_d^T \end{bmatrix}^T$, the tracking error vector can be defined as

$$\tilde{\xi}_e = \begin{bmatrix} \tilde{\eta}_1 \\ \tilde{\eta}_2 \\ \tilde{q} \end{bmatrix} = \begin{bmatrix} \eta_1 - \eta_{1,d} \\ \eta_2 - \eta_{2,d} \\ q - q_d \end{bmatrix} \tag{2.12}$$

The goal of the tracking problem is to make the error vector, $\tilde{\xi}_e$, converge to zero. The tracking control objective is therefore to make $(\tilde{\xi}_e, \tilde{\zeta}) = (0, 0)$ an asymptotically stable equilibrium point of (2.1) and (2.11), which will ensure that the tracking errors will converge to zero.

2.2 Simulation model

The simulation of the motion of the AIAUV is performed using MATLAB Simulink. The model is implemented via the method described in [65] and is based on Eelume robots. In this thesis, we use two different AIAUV models, one based on the 2016 version of the Eelume robot and one based on the 2020 version of the Eelume robot. A detailed description of each robot is given in the following sections. Note that the robots have only 5 visible links (Figure 2.1 and Figure 2.2); however, the links are interconnected by joint modules that allow rotation about two axes (y and z). Accordingly, we model each joint as two revolute joints interconnected by a small link; this is why we define the robot as possessing $n = 9$ links and $n-1 = 8$ revolute joints. In the simulations, both robots are assumed to be neutrally buoyant. The link frames are right-hand coordinate systems, in which the completely outstretched robot is placed such that the x-axes point forward and the z-axes point upwards. The physical parameters in the simulations are shown in Table 2.1.

Table 2.1: Physical parameters used in the simulations

Physical parameter	**Value**
C_a: Added mass coefficient for the cross section	1
C_{d_x}: Non-linear drag coefficient in surge	0.2
C_{d_ϕ}: Non-linear drag coefficient in roll	0.1
C_{d_c}: Non-linear crossflow drag coefficient	0.5
C_{d_l}: Linear cross-sectional drag coefficient	0.1
α: Added mass ratio in surge/heave for a link	0.2
β: Linear drag parameter in surge	0.1
γ: Linear drag parameter in roll	0.1

2.2.1 Eelume 2016 version

The 2016 version of the Eelume robot has a length of 3.37 m, weighs 85.6 kg, and is shown in Figure 2.1. The AIAUV has $n = 9$ cylindrical links with radius 0.09 m, $n - 1 = 8$ revolute joints and $m = 7$ thrusters. The properties of each link are presented in Table 2.2. In the thrusters column, "2: Z, Y" means that the link has 2 thrusters: one thruster works in the z-direction and one thruster work in the y-direction of the link frame. Joints 1, 3, 5 and 7 rotate about the z-axis, and joints 2, 4, 6 and 8 rotate about the y-axis. Joint rotation occurs in the link frame of the corresponding link, i.e., joint 1 rotates about the z-axis of link 1. Note that the thruster configuration of the simulation model is singular in roll when the AIAUV is straight. The thruster allocation matrix is implemented, as proposed in [79]. The maximum thrust of each thruster is approximately 50 N.

Figure 2.1: The Eelume 2016 vehicle (Courtesy: Eelume)

Table 2.2: Eelume 2016 link properties

Link nr.	Length [m]	Mass [kg]	Thrusters
1	0.62	15.8	None
2, 4, 6, 8	0.10	2.5	None
3	0.59	15.0	2: Z, Y
5	0.80	20.4	3: X, X, Z
7	0.59	15.0	2: Y, Z
9	0.37	9.4	None

2.2.2 Eelume 2020 version

The 2020 version of the Eelume robot has a length of 6.15 m, weighs 194.1 kg, and is shown in Figure 2.2. The AIAUV has $n = 9$ cylindrical links with radius 0.1 m, $n - 1 = 8$ revolute joints and $m = 12$ thrusters. The properties of each link are presented in Table 2.3. In the thrusters column, "4: (X,Z), (X,-Z), Y, -Y" means that the link has 4 thrusters: two thrusters work in both the x-direction and the positive/negative z-direction in the link frame, and two thrusters work in the positive/negative y-direction. The thrusters that work in both the x-direction and z-direction are tilted 45 deg such that they works in both directions. Joints 1, 3, 5 and 7 rotate about the z-axis, and joints 2, 4, 6 and 8 rotate about the y-axis. Joint rotation occurs in the link frame of the corresponding link, i.e., joint 1 rotates about the z-axis of link 1. The thruster allocation matrix is implemented, as proposed in [65]. The maximum thrust of each thruster is approximately 60 N, while the limit on the joint torques are 16 Nm.

Figure 2.2: The Eelume 2020 vehicle (Courtesy: Eelume)

Table 2.3: Eelume 2020 link properties

Link nr.	Length [m]	Mass [kg]	Thrusters
1	0.84	26.4	None
2, 4, 6, 8	0.08	2.6	None
3	1.38	43.5	4: (X,Z), (X,-Z), Y, -Y
5	1.63	51.3	4: (X,Z), (X,-Z), Y, -Y
7	1.38	43.5	4: (X,-Z), (X,Z), -Y, Y
9	0.60	19.0	None

2.3 Experiments fall 2018

This section describes the Eelume 2016 robot and the experimental set-up employed for validating the proposed algorithms for trajectory tracking. We also describe the test cases used in the experiments conducted during the fall of 2018.

2.3.1 The Eelume 2016 robot

The Eelume robot used during the experiments conducted during the fall of 2018 was the 2016 version of the robot, which is described by the simulation model in Section 2.2.1. A more detailed description of the robot can be found in [51]. The Eelume 2016 robot has internal joint controllers, which means that in the experiments, we could only provide a desired joint reference q_d rather than joint torque. This is also why we use a P-controller in the simulations performed to be compared with the experiments.

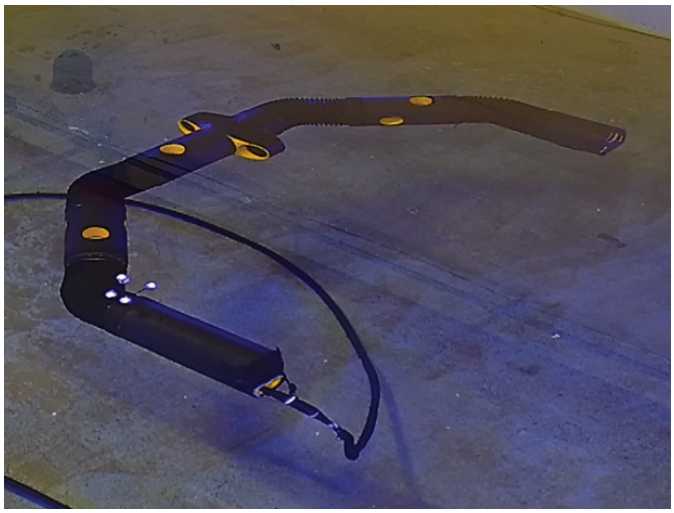


Figure 2.3: The Eelume 2016 vehicle with reflective markers attached to the base

2.3.2 Experimental set-up

We performed the experiments at the MC-lab at NTNU, Trondheim, Norway [1]. The tank located in the MC-lab has the following dimensions: length of 40 m, height of 1.5 m and width of 6.45 m. The underwater motion capture system from Qualisys [2] is installed in the basin to provide us with real-time measurements of the position and orientation of the base of the AIAUV. The Qualisys system has six identical cameras, allowing us to track reflective markers attached to the base of the AIAUV underwater, as shown in Figure 2.3. The cameras allow us to track the reflective markers inside a working area with dimensions of 10 m $\times$1.35 m $\times$5.45 m.

The system structure used during the experiments is illustrated in Figure 2.4. The Qualisys system or camera positioning system sent the measured position and orientation from an external computer, to which the Qualisys system was connected, through UDP to LabVIEW 2016. LabVIEW was then connected to the AIAUV through an optical fibre cable and through UDP to another computer running MATLAB Simulink with the dynamic controller; thruster allocation, which was implemented as in [79]; and the reference generator. The computer running MATLAB Simulink received the measured position and orientation, and if the control laws without the HOSMO were used, also an estimate of the linear and angular velocities, which were estimated by using an extended Kalman filter based on the kinematic model. This solution was chosen since only position measurements were available during the experiments. When the control laws with the HOSMO were used, the linear and angular velocities were estimated by the HOSMO in the dynamic controller. The computer running MATLAB Simulink then sent the thrust commands and the desired joint angles back to LabVIEW through UDP. These were then passed through to the AIAUV through the optical fibre cable. The trajectories for the position and orientation of the base of the AIAUV and the

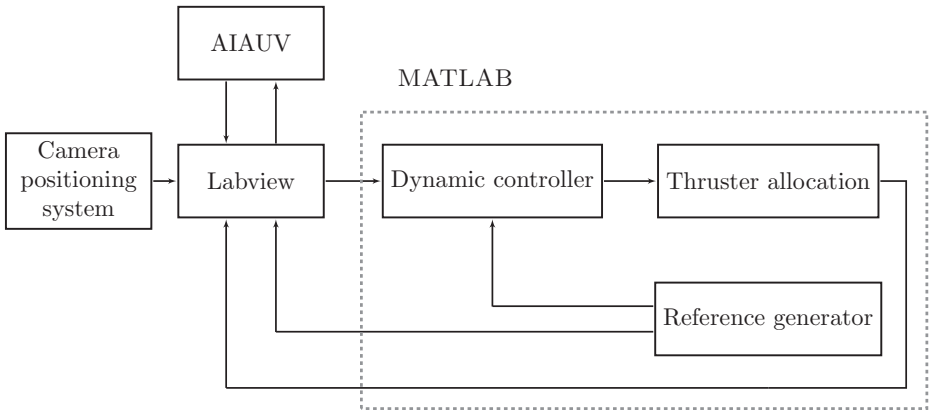


Figure 2.4: Illustration of the system structure (experiments fall 2018)

joints were generated by setting set-points and by then using a filter to create a continuous trajectory between them.

2.3.3 Test cases for simulations and experiments

In this section, we describe the different test cases considered in the simulations and experiments performed during the fall in 2018. The cases, or shapes, chosen to validate and compare the control laws presented in this thesis are C-shape, C-shape with a moving head and I-shape. The reason we chose these shapes is the two different modes, transport mode and operation mode, which is explained in detail in [79]. The transport mode is for long-distance travel, and the operation mode is for performing inspections and intervention tasks. In the transport mode, the accuracy of the tracking is not that important, and we therefore chose to use the I-shape for that case. In the operation mode, however, the accuracy is extremely important, and since the AIAUV is unactuated in roll when the AIAUV is in the I-shape, the I-shape cannot be used for inspection or intervention tasks. We therefore chose to test two other cases for the operation mode where all DOFs are actuated. The two cases chosen are the C-shape and the C-shape with a moving head.

In Figure 2.5 the reference trajectory for the position and orientation that is used for all the simulation cases is shown. At 100 s, we change the reference for x, y and ψ, and at 300 s, we change the reference for z and θ. We also change the reference for x, ψ and θ at 450 s and for x, y and z at 600 s. The reference values for z are, however, different for the different cases. For the C-shape, the reference for z starts at -1 and goes to -0.7, and for the C-shape with a moving head and the I-shape, the z reference starts at -0.7 and goes to -0.5. The reason for the difference is that during the experiments, we observed that some of the joints were peeking out of the water and creating problems for the measurements system when we started the trajectory at -1 for the C-shape with a moving head and the I-shape. The z reference for the C-shape with a moving head when the control law with the HOSMO is used is also changed at 100 s, starting at -1, going to -0.7 at 100 s, and then to -0.5 at 300 s.

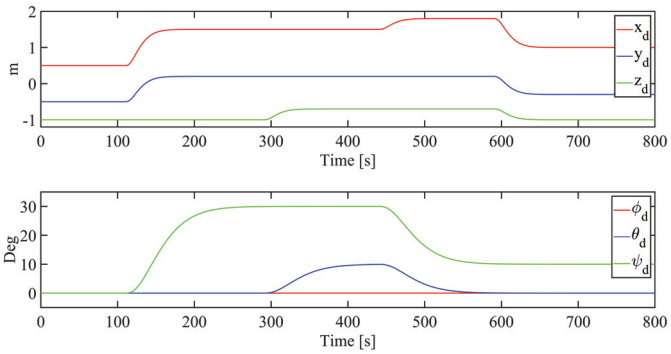
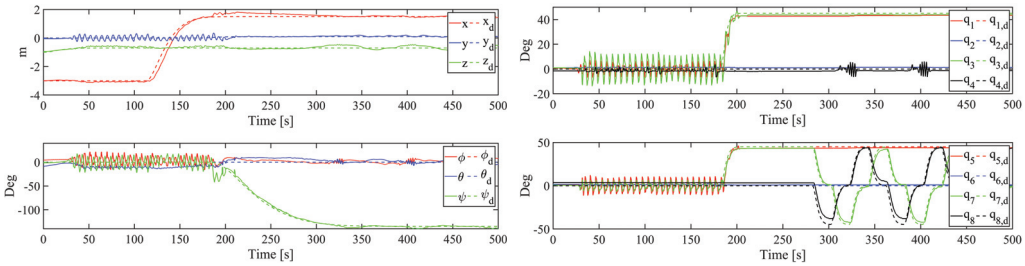


Figure 2.5: Reference trajectories for position and orientation

In the experiments, a test representing both modes was also performed, i.e. we started with the transport mode and then went into the operation mode. This test was mainly performed to highlight the reasoning for choosing the tests (i.e. the shapes). In the complete test, we first move forward in the x-direction in the transport mode before we turn the robot into the C-shape, and then we move in ψ to make the head face the direction that it originally was facing in the I-shape. The head is then moved around to make the test represent a complete inspection. In Figure 2.6a and Figure 2.6b, the movement of the joints, the position and the orientation are shown. Note the oscillations when the robot is in the I-shape. These oscillations occur because the robot is underactuated in roll, which means that we do not have any thrusters that can provide a torque to control roll. The robot will therefore move from side to side in roll. Now, since the controller will try to control roll anyway, the controller calculates a force that it needs to stabilize the roll angle. This force is given to the thruster allocation algorithm to distribute to the thrusters. Since the robot is underactuated in roll, the thruster allocation scheme should have given zero as the desired moment in roll, but that is not the case. Some of the desired moment in roll is therefore distributed to the thrusters, which cause an oscillation effect. From these results, we can find that performing an inspection or intervention in the I-shape would be very difficult, or at least the precision would be bad, and we therefore move into the C-shape where the robot is still before we perform the inspection.

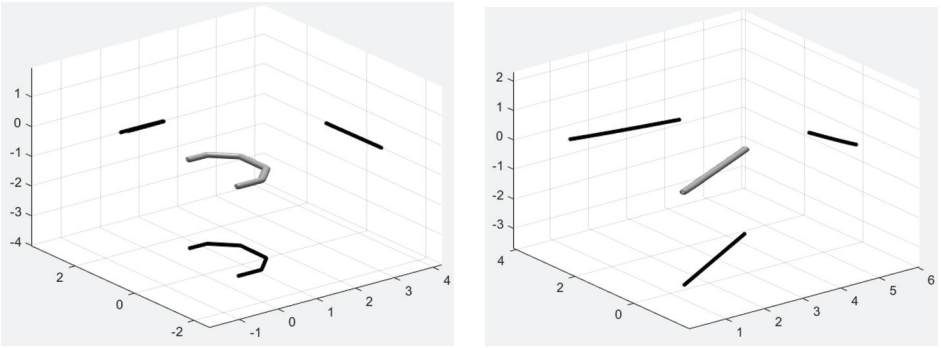
To make the simulations a valid comparison with the experiments, we use a P-controller for the joints. The reason for this choice is that the Eelume robot has an internal joint controller, which is a P-controller. We therefore include figures of the actual joint angles and their reference trajectories in the description of the test cases in the section below, both in simulations and experiments. This is to provide the reader with an idea of how well the internal P-controller for the joints performed.



(a) Position and orientation

(b) Joint angles

Figure 2.6: Tracking results for a complete inspection



(a) C-shape configuration

(b) I-shape configuration

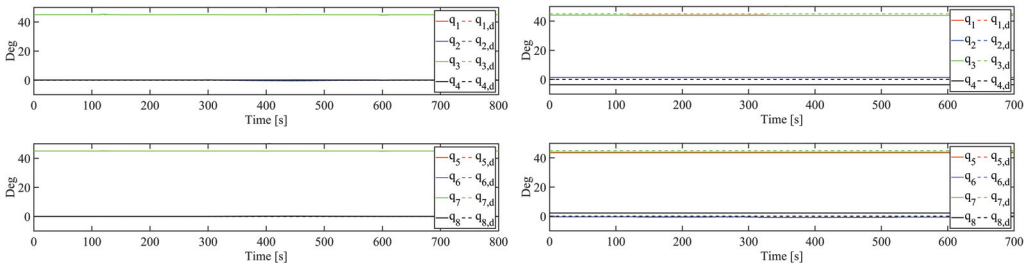
Figure 2.7: Configurations

C-shape

In the C-shape configuration, the AIAUV is shaped like a C and the AIAUV is actuated in every DOF, since the thrusters are oriented such that the robot can provide thrust in all 6DOFs. The C-shape is therefore ideal for operation mode. In C-shape the joint angles references are set to $q_d = [45\,\text{deg}\ \ 0\,\text{deg}\ \ 45\,\text{deg}\ \ 0\,\text{deg}\ \ 45\,\text{deg}\ \ 0\,\text{deg}\ \ 45\,\text{deg}\ \ 0\,\text{deg}]^T$, and in Figure 2.7a, the configuration of the robot is shown. In Figure 2.8a and Figure 2.8b, the joint angles are shown together with their reference trajectories for the C-shape for simulations and experiments, respectively.

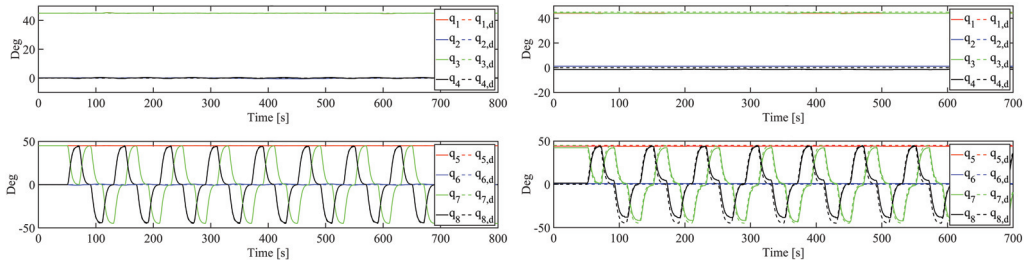
C-shape with a moving head

The C-shape with a moving head configuration of the AIAUV is similar to that of the C-shape configuration; the only difference is that the $n-1$-th and $n-2$-th joints are moving such that the camera positioned in the n-th joint of the AIAUV is looking around. The references for the joint angles at $t = 0$ are therefore set to $q_d(0) = [45\,\text{deg}\ \ 0\,\text{deg}\ \ 45\,\text{deg}\ \ 0\,\text{deg}\ \ 45\,\text{deg}\ \ 0\,\text{deg}\ \ 45\,\text{deg}\ \ 0\,\text{deg}]^T$, and then at 50 s q_7 and q_8 starts to move in a circular motion, while the other joints remain constant.



(a) Simulation results (b) Experimental results

Figure 2.8: Joint angles in the C-shape case



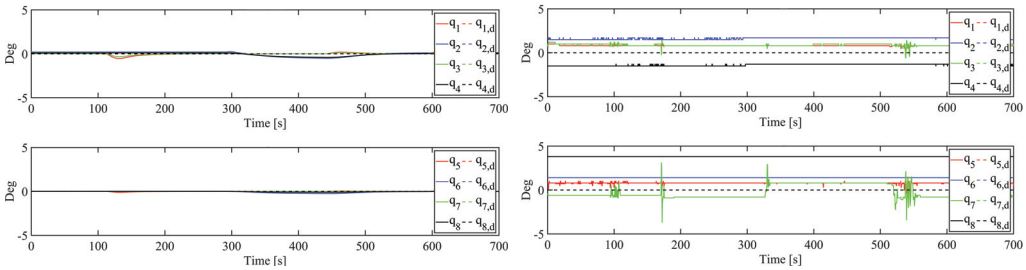
(a) Simulation results (b) Experimental results

Figure 2.9: Joint angles in the C-shape with a moving head case

The reference for q_7 is given by four set-points $0\,\text{deg} \rightarrow 45\,\text{deg} \rightarrow 0\,\text{deg} \rightarrow -45\,\text{deg}$. The set-point is changed every 50 s, and a filter is used to create a continuous trajectory between the set-points. The set-points are then repeated trough the simulation. The reference for q_8 is created in the same manner, however the set-points are then given by $45\,\text{deg} \rightarrow 0\,\text{deg} \rightarrow -45\,\text{deg} \rightarrow 0\,\text{deg}$. The joint angles together with their reference trajectories for the C-shape with a moving head are presented in Figure 2.9a for the simulations, and in Figure 2.9b for the experiments.

I-shape

In the I-shape, the joints are all set to zero such that the robot is straight, like an I, i.e. the joint angles references are set to $q_d = [0\,\text{deg}\ \ 0\,\text{deg}\ \ 0\,\text{deg}\ \ 0\,\text{deg}\ \ 0\,\text{deg}\ \ 0\,\text{deg}\ \ 0\,\text{deg}\ \ 0\,\text{deg}]^T$. The configuration of the robot is shown in Figure 2.7b. In Figure 2.10a and Figure 2.10b, the joint angles are shown together with their reference trajectories for the C-shape for simulations and experiments, respectively.



(a) Simulation results

(b) Experimental results

Figure 2.10: Joint angles in the I-shape case

2.4 Experiments summer 2020

This section describes the Eelume 2020 robot and the experimental set-up employed for validating the proposed trajectory tracking algorithms. We also describe the test cases used in the experiments conducted during the summer of 2020.

2.4.1 The Eelume 2020 robot

The Eelume robot used during the experiments conducted during the summer of 2020 was the 2020 version of the robot, which is described by the simulation model in Section 2.2.2. This new robot has torque control on the joints, which means that we could use our control algorithms to also control the joint angles by means of the torque, and not just the position and orientation of the vehicle by means of the thrusters.

2.4.2 Experimental set-up

The experiments were performed in the same location as the experiments conducted during the fall of 2018, i.e., at the MC-lab at NTNU, Trondheim, Norway [1]. Additionally, we used the underwater motion capture system from Qualisys [2], which allowed us to track reflective markers inside a working area with dimensions of 10 m $\times$1.35 m $\times$5.45 m. In Figure 2.11, the 2020 version of the robot can be seen, with the reflective markers attached to the centre link of the robot.

The system structure used during the experiments is illustrated in Figure 2.12. The Qualisys system or camera positioning system sent the measured position and orientation from an external computer, to which the Qualisys system was connected, via UDP to a different computer running LabVIEW 2016. The LabVIEW computer was then connected via UDP to another computer running MATLAB Simulink, where the reference generator or inverse kinematic scheme, the dynamic controllers and the thruster allocation were implemented. The computer running MATLAB Simulink also ran Eelume Suite, which is the program developed by Eelume to connect to the robot. From the Eelume Suite program, we obtained the orientation of the vehicle and the joint angles. We sent the joint torques and

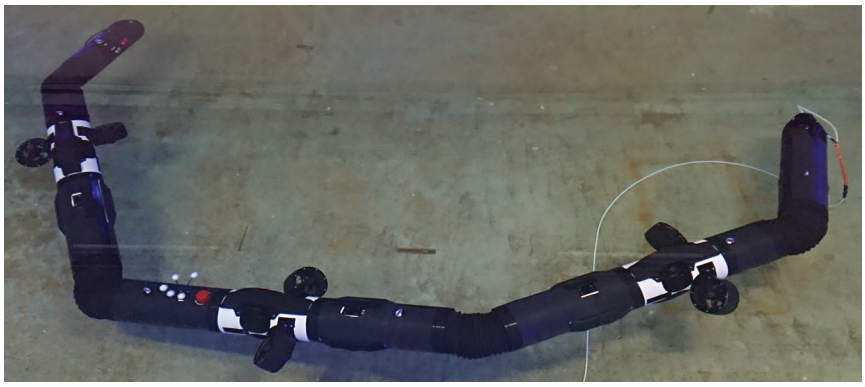


Figure 2.11: The Eelume 2020 vehicle with reflective markers attached to the centre link

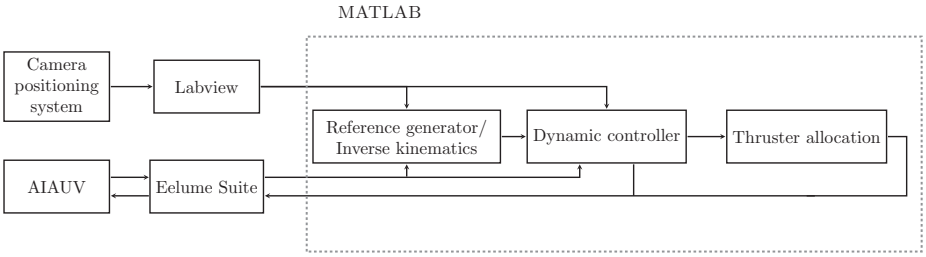
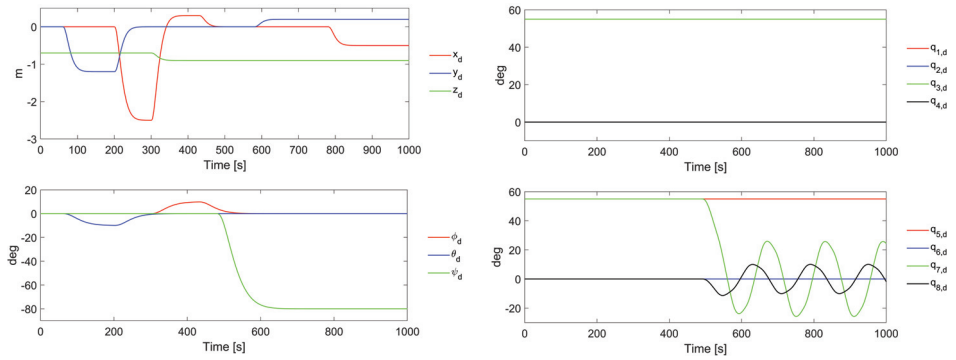


Figure 2.12: Illustration of the system structure (experiments summer 2020)

thrust commands from MATLAB Simulink to Eelume Suite. Eelume Suite sent and received information to and from the robot through an optical fibre cable. To estimate the linear and angular velocities of the vehicle and the joint velocities, we used an extended Kalman filter based on the kinematic model, which was also implemented in MATLAB Simulink. As inputs to the Kalman filter, we used the position measurements from Qualisys and the orientation and joint measurements from the robot, as the orientation measurements from the robot were more accurate than those from Qualisys. To control the thrusters, we used current control. The thruster reference was thus proportional to the motor current, which means that we used a linear mapping to calculate the commanded reference we sent from MATLAB Simulink from the desired force. The commanded reference was a number in the range of ± 100, which corresponded to ± 23 A on the motor and approximately ± 60 N.

2.4.3 Test cases for simulations and experiments

In this section, we describe the different test cases considered in the simulations and experiments performed during the summer of 2020.



(a) Position and orientation

(b) Joint angles

Figure 2.13: Reference trajectories for the inspection case

Inspection

In this case, we wanted to highlight two different modes, i.e., transport mode and operation mode, which are explained in detail in [79]. Transport mode is for long-distance travel, and operation mode is for performing inspections and intervention tasks. However, we noticed during the experiments that the 2020 version of the robot is very sensitive in roll when in the I-shape (in which all joint angles are equal to zero: $q = 0\,\text{deg}$), which is used for transport mode. We therefore needed to use a very small gain for roll control when in the I-shape. When we then tried to transition from the I-shape to the C-shape (in which every second joint angle is equal to $55\,\text{deg}$), we needed a much larger gain in roll to achieve good tracking. Keeping the same gain as for the I-shape resulted in a large deviation in roll, and the robot ended up touching the bottom of the basin during the transition. When we used a controller with an integrator, we were able to eventually achieve good tracking after the transition from the I-shape to the C-shape; however, it took a very long time. Since the robot is so sensitive in roll, it was also very difficult to achieve good tracking in the I-shape because of the dead-zone of the thrusters. We therefore ended up with a case that could highlight only the operation mode.

In this case, we performed an inspection by setting the AIAUV in the C-shape configuration and then, after some time, moving joints 7 and 8 in a circular motion such that link 9 was moved around. In Figure 2.13, the reference trajectory for the position, the orientation and the movement of the joints are shown. The trajectories for the position and orientation of the AIAUV and the joints were generated by setting set-points and then applying a filter to create a continuous trajectory between them.

Singularity-robust multiple task-priority

This case was chosen to test the combined kinematic and dynamic controller, which will be presented in Chapter 7. Therefore, we used the SRMTP method to create a continuous trajectory for the AIAUV to follow. The dynamic controller acts on the centre link of the AIAUV, and the output reference for the SRMTP method must

therefore be created for the centre link. To demonstrate the proposed approach, we used the following two set-point regulation tasks:

1. Control the position and orientation of the front end (link 9)
2. Control the position and orientation of the back end (link 1)

This combination of tasks illustrates the ability of the AIAUV to move to a position of interest and then perform a double observation task by simultaneously adjusting the position and orientation of its front and back ends.

The expressions for the task error and the task Jacobian for task 1 are as follows:

Task 1 - Position and orientation of the front end

$$\tilde{\chi}_1 = \left[(\tilde{\eta}^f_{1,If})^T, \tilde{\eta}^T_{2,f}\right]^T$$
$$J_1 = J_{cf},$$

where $\tilde{\eta}^f_{1,If}$ is the position deviation of the front end and $\tilde{\eta}_{2,f}$ is the orientation deviation of the frond end. The task Jacobian J_1 for task 1 is the front-end Jacobian J_{cf}, which relates the body-fixed velocities of the front end to the body-fixed velocity of the centre link and the joint velocities. The tasks can then be solved by using the position and orientation of the centre link and the two double-joint modules in front of the centre link.

The second task utilizes the position and orientation of the centre link and the two double-joint modules behind the centre link, so the expressions for the task error and the task Jacobian for task 2 are as follows:

Task 2 - Position and orientation of the back end

$$\tilde{\chi}_2 = \left[(\tilde{\eta}^b_{1,Ib})^T, \tilde{\eta}^T_{2,b}\right]^T$$
$$J_2 = J_{cb},$$

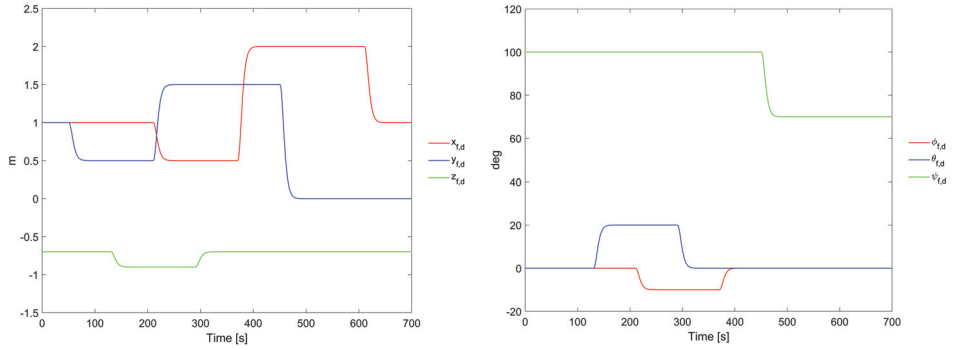
where $\tilde{\eta}^b_{1,Ib}$ is the position deviation of the back end and $\tilde{\eta}_{2,b}$ is the orientation deviation of the back end. The task Jacobian J_2 for task 2 is the back-end Jacobian J_{cb}, which relates the body-fixed velocities of the back end to the body-fixed velocity of the centre link and the joint velocities.

The reference velocities, ζ_r, are calculated according to

$$\zeta_r = J_1^+ \Lambda_1 \tilde{\chi}_1 + N_1 J_2^+ \Lambda_2 \tilde{\chi}_2 \tag{2.13}$$

where $J_i^+ = J_i^T (J_i J_i^T)^{-1}$ is the Right Moore-Penrose pseudo-inverse of the task Jacobian and N_1 is the null space projector of task 1. The set-points $\chi_{i,d}$ were set manually and filtered through a third-order reference filter to avoid discontinuities and large jumps in the calculated reference velocities. The gain parameters in (2.13) were set to $\Lambda_1 = 0.5 I_{6\times6}$ and $\Lambda_2 = 0.5 I_{6\times6}$.

The reference trajectories for each task are shown in Figure 2.14.



(a) Front-end position

(b) Front-end orientation

(c) Back-end position

(d) Back-end orientation

Figure 2.14: Reference trajectories for each task in the task-priority case

Chapter 3

Tracking using the Super-Twisting Algorithm

This chapter investigates the use of the STA with adaptive gains to solve the trajectory tracking problem for an AIAUV. The tracking problem is also solved when velocity measurements are not available by using the STA with adaptive gains in combination with a HOSMO. The closed-loop system is then analysed.

Contributions of this chapter

The contributions of this chapter can be summarized as follows. The trajectory tracking control problem of an AIAUV in 6DOF is solved using the STA with adaptive gains. The trajectory tracking control problem is also solved by using the algorithm in combination with a HOSMO. Additionally, the tracking errors for the control law using the STA with adaptive gains in combination with the HOSMO are proven to converge asymptotically to zero. We demonstrate the applicability of the presented control laws with comprehensive simulation and experimental results.

Organization of this chapter

This chapter is structured as follows. In Section 3.1 the STA with adaptive gains is given in detail. The control law and observer design for tracking the desired trajectory are presented and analysed in Section 3.2. The simulation results are presented in Section 3.3 and the experimental results are presented in Section 3.4. In Section 3.5 a chapter summary is presented.

Publications

This chapter is based on [8], [9] and [15].

3.1 The super-twisting algorithm with adaptive gains

In this section the STA with adaptive gains, which will be used for trajectory tracking in this chapter, will be introduced. In recent years, SMC has been further

developed into higher-order SMC schemes, which removes the chattering problem. The STA with adaptive gains, proposed by [69], is the most powerful second-order continuous SMC algorithm, as it attenuates chattering and does not require the consideration of any conservative upper bound on the disturbance in order to maintain sliding because of the adaptive gains. The STA with adaptive gains proposed in [69] can be written by the update law

$$\begin{aligned} u_{\text{STA}}(\sigma) &= -\alpha|\sigma|^{1/2}\,\text{sgn}(\sigma) + v \\ \dot{v} &= -\beta_{\text{STA}}\,\text{sgn}(\sigma) \end{aligned} \tag{3.1}$$

where σ is the sliding surface and the adaptive gains are defined as

$$\dot{\alpha} = \begin{cases} \omega_1\sqrt{\frac{\gamma_1}{2}} & \text{if } \sigma \neq 0 \\ 0, & \text{if } \sigma = 0 \end{cases} \tag{3.2}$$

and

$$\beta_{\text{STA}} = 2\varepsilon\alpha + \lambda + 4\varepsilon^2 \tag{3.3}$$

where ε, λ, γ_1 and ω_1 are positive constants and σ is the sliding surface.

3.2 Tracking control laws

In this section, we will present two tracking control laws for the AIAUV based on the STA with adaptive gains: one where only the STA with adaptive gains is used, and one where the STA with adaptive gains is combined with a HOSMO. The reason for this is that the STA is only applicable to systems where the control input appears in the equation for the first derivative of the sliding variable, both the position and velocity of the AIAUV must therefore be available for measurement. For the case when only the position measurements are available, we use a HOSMO, as proposed in [46], to estimate the states.

3.2.1 Control law based on the STA with adaptive gains

In this section, a control law based on the STA with adaptive gains is presented.

Sliding surface:

For SMC to be used, a sliding surface needs to be designed. The surface should be designed such that when the sliding variable σ goes to zero, the state variables asymptotically converge to zero, and such that the control input $\tau(q)$ appears in the first derivative of σ. Define $\tilde{x}_1 = \tilde{\xi}_q$, where $\tilde{\xi}_q$ is given by (2.9) and

$$\tilde{x}_2 = \begin{bmatrix} R_B^I(\tilde{p}) & 0_{3\times3} & 0_{3\times(n-1)} \\ 0_{3\times3} & \frac{1}{2}(\tilde{\eta}I_3 + S(\tilde{\varepsilon})) & 0_{3\times(n-1)} \\ 0_{(n-1)\times3} & 0_{(n-1)\times3} & I_{(n-1)\times(n-1)} \end{bmatrix} (\zeta - \zeta_d) = T_q(\tilde{p})\tilde{\zeta} \tag{3.4}$$

where ζ and ζ_d are defined in (2.3) and (2.4), respectively. The sliding surface can then be chosen as

$$\sigma_q = \tilde{x}_1 + \tilde{x}_2 \quad \in \mathbb{R}^{6+(n-1)} \tag{3.5}$$

If $\sigma_q = 0$, we will now have $\tilde{x}_1 + \tilde{x}_2 = 0$. Since $\tilde{x}_2 = \dot{\tilde{x}}_1$, we can write this as

$$\dot{\tilde{x}}_1 = -\tilde{x}_1 \tag{3.6}$$

which will ensure that $\tilde{x}_1$ globally exponentially converges to zero. Now, since $\tilde{x}_1 = \tilde{\xi}_q$, the state variables $\tilde{\xi}_q$ will also globally exponentially converge to zero if $\sigma_q = 0$. To drive the sliding surface to zero, we will use the STA with adaptive gains.

Control input:

Now, we want the control input $\tau(q)$ to be chosen such that $\dot{\sigma}_q = u_{\text{STA}}$, so that σ_q and $\dot{\sigma}_q$ will reach zero in finite time. However, since we do not know anything about the model, we choose our control input to be

$$\tau(q) = u_{\text{STA}}(\sigma_q) \tag{3.7}$$

with σ_q defined as in (3.5). With this choice, u_{STA} appears in $\dot{\sigma}_q$, as desired.

3.2.2 Control law based on the STA with adaptive gains combined with a higher-order sliding mode observer

In this section, a control law based on the STA with adaptive gains combined with a HOSMO will be derived. Using Lyapunov theory, we will show that $(\tilde{\xi}_e, \tilde{\zeta}) = (0, 0)$ is an asymptotically stable equilibrium point of (2.1) and (2.11) with the proposed control law. We here use the Euler angles representation of the model (Section 2.1.2) when we create the control law. The reason is that the HOSMO does not work with quaternions, as there is then a different number of states in position versus velocity.

State observer

Because velocity measurements are not available, a state observer has to be designed. We want to use the third-order sliding mode observer (SMO) presented in [46], as it has been proven to be FTS in [57]. To use this third-order SMO, we introduce a change of variables. Define $x_1 = \xi_e$, where ξ_e is given by (2.10), and $x_2 = J_e(\eta_2)\zeta$, where $J_e(\eta_2)\zeta$ is given by (2.11). The dynamics can then be written as

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{d}{dt}(J_e(\eta_2))J_e^{-1}(\eta_2)x_2 + M^{-1}(q)J_e(\eta_2)\Big(- C\big(q, J_e^{-1}(\eta_2)x_2\big)J_e^{-1}(\eta_2)x_2 \\
&\quad - D\big(q, J_e^{-1}(\eta_2)x_2\big)J_e^{-1}(\eta_2)x_2 - g(q, R_B^I) + \tau(q)\Big)
\end{aligned} \tag{3.8}$$

Assumption 3.1. We assume that $\frac{d}{dt}(J_e(\eta_2))J_e^{-1}(\eta_2)x_2$ is a small bounded disturbance, which we will call $d(t)$.

Remark 3.1. For all practical purposes, Assumption 3.1 will be satisfied since the AIAUV is a mechanical system and has a limited control input, which will cause

the velocities of the system to be bounded. If the velocities are bounded $\frac{d}{dt}(J_e(\eta_2))$ will be small and bounded. Furthermore, $J_e^{-1}(\eta_2)$ is bounded since it consists of elements with cos and sin. For theoretical purposes, it should be proven that this assumption hold; however, this proof will be left as a topic for future work.

We also define $f(\cdot) = -C\big(q, J_e^{-1}(\eta_2)x_2\big)J_e^{-1}(\eta_2)x_2 - D\big(q, J_e^{-1}(\eta_2)x_2\big)J_e^{-1}(\eta_2)x_2 - g(q, R_B^I)$ to reduce the space used to write the model. The model can then be written as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= d(t) + M^{-1}(q)J_e(\eta_2)\big(f(\cdot) + \tau(q)\big) \end{aligned} \tag{3.9}$$

By designing the third-order SMO structure as in [46], the third-order SMO can be written as

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 + z_1 \\ \dot{\hat{x}}_2 &= \hat{x}_3 + z_2 + M^{-1}(q)J_e(\eta_2)\tau(q) \\ \dot{\hat{x}}_3 &= z_3 \end{aligned} \tag{3.10}$$

where

$$\begin{aligned} z_1 &= k_1|e_1|^{2/3}\operatorname{sgn}(e_1) \\ z_2 &= k_2|e_1|^{1/3}\operatorname{sgn}(e_1) \\ z_3 &= k_3\operatorname{sgn}(e_1) \end{aligned} \tag{3.11}$$

and $k_1 \in \mathbb{R}^{6+(n-1)}$, $k_2 \in \mathbb{R}^{6+(n-1)}$ and $k_3 \in \mathbb{R}^{6+(n-1)}$ are gains to be chosen according to [48] and [49], where $e_1 = x_1 - \hat{x}_1 \in \mathbb{R}^{6+(n-1)}$. One choice of parameters that satisfies the requirements in [48] and [49] is, according to [19], $k_1 = 6L^{1/3}$, $k_2 = 11L^{1/2}$ and $k_3 = 6L$, where $L \in \mathbb{R}^{6+(n-1)}$ is a sufficiently large constant. Note that the mathematical operations in (3.10) and (3.11) are performed in an element-wise manner. By defining $e_2 = x_2 - \hat{x}_2$ and $e_3 = -\hat{x}_3 + F(\cdot)$, where $F(\cdot) = d(t) + M^{-1}(q)J_e(\eta_2)f(\cdot)$, the error dynamics of the HOSMO can be written as

$$\begin{aligned} \dot{e}_1 &= -k_1|e_1|^{2/3}\operatorname{sgn}(e_1) + e_2 \\ \dot{e}_2 &= -k_2|e_1|^{1/3}\operatorname{sgn}(e_1) + e_3. \\ \dot{e}_3 &= -k_3\operatorname{sgn}(e_1) + \dot{F}(\cdot) \end{aligned} \tag{3.12}$$

If $|\dot{F}(\cdot)| < \Delta$, then the third-order SMO errors go to zero in finite time [57]. Since $F(\cdot)$ is a combination of $d(t)$, $C(q, J_e^{-1}(\eta_2)x_2)J_e^{-1}(\eta_2)x_2$, $D(q, J_e^{-1}(\eta_2)x_2)J_e^{-1}(\eta_2)x_2$ and $g(q, R_B^I)$, and since the AIAUV is a mechanical system, these matrices will not change infinitely fast. It is therefore a valid assumption to assume that $\dot{F}(p, \zeta, \eta_2)$ is bounded.

Sliding surface

As mentioned previously, we have to design a sliding surface to use SMC. The sliding surface has to be designed such that when the sliding variable σ goes to

zero, the state variables asymptotically converge to zero, and such that the control input $\tau(q)$ appears in the first derivative of σ. When the velocity measurement is available, we choose the sliding surface to be

$$\sigma_e = \tilde{x}_1 + \tilde{x}_2 \quad \in \mathbb{R}^{6+(n-1)} \tag{3.13}$$

where $\tilde{x}_1 = \tilde{\xi}_e$, with $\tilde{\xi}_e$ defined as in (2.12), and $\tilde{x}_2 = J_e(\eta_2)\zeta - J_e(\eta_{2,d})\zeta_d$, where $J_e(\eta_2)\zeta$ is defined in (2.11). If $\sigma_e = 0$, we will now have $\tilde{x}_1 + \tilde{x}_2 = 0$. Since $\tilde{x}_2 = \dot{\tilde{x}}_1$, we can write this as

$$\dot{\tilde{x}}_1 = -\tilde{x}_1 \tag{3.14}$$

which will ensure that $\tilde{x}_1$ globally exponentially converges to zero. Now, since $\tilde{x}_1 = \tilde{\xi}_e$, the state variables $\tilde{\xi}_e$ will also globally exponentially converge to zero if $\sigma_e = 0$. When the velocity measurement is not available, the observed state values are used, and we can therefore write the sliding surface with the observed values as

$$\hat{\sigma}_e = \hat{\tilde{x}}_1 + \hat{\tilde{x}}_2 \quad \in \mathbb{R}^{6+(n-1)} \tag{3.15}$$

where $\hat{\tilde{x}}_1 = \hat{x}_1 - \xi_{e,d}$ and $\hat{\tilde{x}}_2 = \hat{x}_2 - J_e(\eta_{2,d})\zeta_d$. Since the third-order SMO errors in (3.12) go to zero in finite time, $\hat{\sigma}_e = \sigma_e$ after some finite time. Thus, if $\hat{\sigma}_e$ goes to zero, the tracking objective will be satisfied. To drive the sliding surface to zero, we will use the STA with adaptive gains.

Control input

By designing the control input $\tau(q)$ such that $\dot{\hat{\sigma}}_e = u_{\text{STA}}$, we achieve that $\hat{\sigma}_e$ and $\dot{\hat{\sigma}}_e$ reach zero in finite time since the STA is FTS. Taking the time derivative of (3.15) and substituting $\dot{\hat{x}}_1$ and $\dot{\hat{x}}_2$, defined in (3.10), we find that

$$\begin{aligned} \dot{\hat{\sigma}}_e &= \dot{\hat{\tilde{x}}}_1 + \dot{\hat{\tilde{x}}}_2 = \dot{\hat{x}}_1 - \dot{x}_{1,d} + \dot{\hat{x}}_2 - \dot{x}_{2,d} \\ &= \hat{x}_2 + z_1 - \dot{x}_{1,d} + \hat{x}_3 + z_2 + M(q)^{-1} J_e(\eta_2)\tau(q) - \dot{x}_{2,d} \end{aligned} \tag{3.16}$$

By choosing $\tau(q)$ to be

$$\tau(q) = J_e(\eta_2)^{-1} M(q)(-\hat{x}_2 - z_1 + \dot{x}_{1,d} - \hat{x}_3 - z_2 + \dot{x}_{2,d} + u_{\text{STA}}) \tag{3.17}$$

we obtain

$$\dot{\hat{\sigma}}_e = u_{\text{STA}}. \tag{3.18}$$

Stability analysis

In this section, we perform a stability analysis of the closed-loop system, and it is shown that the tracking error converges asymptotically to zero. We consider the closed-loop system defined by (2.1), (2.11) and (3.17). By using the fact that $\hat{x}_1 = x_1 - e_1$ and that $\hat{x}_2 = x_2 - e_2$, from Section 3.2.2, (3.15) can be written as

$$\hat{\sigma}_e = x_1 - e_1 - x_{1,d} + x_2 - e_2 - x_{2,d} = \tilde{\xi}_e - e_1 + \dot{\tilde{\xi}}_e - e_2 \tag{3.19}$$

Through rearranging, we obtain that the tracking error dynamics is

$$\dot{\tilde{\xi}}_e = -\tilde{\xi}_e + \hat{\sigma}_e + e_1 + e_2 \tag{3.20}$$

Furthermore, the velocity tracking error $\tilde{\zeta}$ is represented by the sliding variable $\hat{\sigma}_e$, cf., (3.8) and (3.15). The overall closed-loop dynamics with $\tau(q)$ given by (3.7) is thus given by $\dot{\hat{\sigma}}_e$ given in (3.18), $\dot{\tilde{\xi}}_e$ given in (3.20) and the state observer error given in (3.12). The closed-loop dynamics is thus

$$\begin{aligned} \sum_1 \begin{cases} \dot{\tilde{\xi}}_e = -\tilde{\xi}_e + \hat{\sigma}_e + e_1 + e_2 \\ \dot{\hat{\sigma}}_e = -\alpha|\hat{\sigma}_e|^{1/2}\operatorname{sgn}(\hat{\sigma}_e) + v \\ \dot{v} = -\beta_{\text{STA}}\operatorname{sgn}(\hat{\sigma}_e) \end{cases} \\ \sum_2 \begin{cases} \dot{e}_1 = -k_1|e_1|^{2/3}\operatorname{sgn}(e_1) + e_2 \\ \dot{e}_2 = -k_2|e_1|^{1/3}\operatorname{sgn}(e_1) + e_3 \\ \dot{e}_3 = -k_3\operatorname{sgn}(e_1) + \dot{F}(\cdot) \end{cases} \end{aligned} \tag{3.21}$$

Theorem 3.1. *Consider the closed-loop system (2.1), (2.11), and (3.17). Assume that the HOSMO in (3.10) and (3.11) is used to estimate x_1 and x_2, where it is assumed that $|\dot{F}(\cdot)| < \Delta$, and assume that the sliding surface is chosen as in (3.15). Then, the complete system is represented by the cascaded system in (3.21), and the origin of the cascaded system is UGAS, which ensures the asymptotic convergence of the tracking error.*

Proof. Analysis of subsystem 1, with $e_1 = 0$ and $e_2 = 0$: With $e_1 = 0$ and $e_2 = 0$, subsystem 1 can be written as

$$\sum_1 \begin{cases} \dot{\tilde{\xi}}_e = -\tilde{\xi}_e + \hat{\sigma}_e \\ \dot{\hat{\sigma}}_e = -\alpha|\hat{\sigma}_e|^{1/2}\operatorname{sgn}(\hat{\sigma}_e) + v \\ \dot{v} = -\beta_{\text{STA}}\operatorname{sgn}(\hat{\sigma}_e) \end{cases} \tag{3.22}$$

This can then be divided into two subsystems:

$$\begin{aligned} \sum_{11} \begin{cases} \dot{\tilde{\xi}}_e = -\tilde{\xi}_e + \hat{\sigma}_e \end{cases} \\ \sum_{12} \begin{cases} \dot{\hat{\sigma}}_e = -\alpha|\hat{\sigma}_e|^{1/2}\operatorname{sgn}(\hat{\sigma}_e) + v \\ \dot{v} = -\beta_{\text{STA}}\operatorname{sgn}(\hat{\sigma}_e) \end{cases} \end{aligned} \tag{3.23}$$

where Lemma A.7 can be used. Subsystem $\sum_{11}$ with $\hat{\sigma}_e = 0$ is analysed first. This is clearly a globally exponentially stable linear system, but since we will need a Lyapunov function to analyse this system when $\hat{\sigma}_e \neq 0$, we use the Lyapunov function candidate $V_{11}(\tilde{\xi}_e) = \frac{1}{2}\tilde{\xi}_e^2$ for the analysis. The derivative of V_{11} yields

$$\dot{V}_{11}(\tilde{\xi}_e) = \tilde{\xi}_e\dot{\tilde{\xi}}_e = \tilde{\xi}_e(-\tilde{\xi}_e) = -\tilde{\xi}_e^2 \leq -||\tilde{\xi}_e||^2 \tag{3.24}$$

This means that the Lyapunov function satisfies:

$$\begin{aligned} k_1||x||^a \leq V_{11}(x) \leq k_2||x||^a \\ \frac{\partial V_{11}}{\partial x} f_{11}(t,x) \leq -k_3||x||^a \end{aligned} \tag{3.25}$$

with $k_1 = k_2 = \frac{1}{2}$, $k_3 = 1$ and $a = 2$. Hence, by virtue of Theorem A.4, the origin for subsystem $\sum_{11}$ with $\hat{\sigma}_e = 0$ is globally exponentially stable (GES, Definition A.3).

Subsystem $\sum_{12}$ has the structure of the STA with adaptive gains. In [69], a Lyapunov function is proposed for systems with this structure. Here, it is proven that the Lyapunov function proposed is indeed a Lyapunov function for subsystem $\sum_{12}$ and that for any initial conditions, $\hat{\sigma}_e, \dot{\hat{\sigma}}_e \to 0$ in finite time by using the STA with adaptive gains given by Eq. (3.2) and Eq. (3.3), where ε, λ, γ_1 and ω_1 are arbitrary positive constant. It is also proven that the sliding surface $\hat{\sigma}_e = 0$ will be reached in finite time. Now, since the subsystem is GFTS and autonomous, it is also UGAS by Proposition A.1 and Proposition A.2, which also implies that $||\hat{\sigma}_e(t)|| < \beta_1 \forall t \geq 0$.

To verify that the solutions of $\sum_1$ are uniformly globally bounded (UGB, Definition A.9), subsystem $\sum_{11}$ must be analysed with $\hat{\sigma}_e \neq 0$. The derivative of the Lyapunov function V_{11} is then as follows:

$$\begin{aligned} \dot{V}_{11}(\tilde{\xi}_e) &= -||\tilde{\xi}_e||^2 + \hat{\sigma}_e \tilde{\xi}_e \\ &\leq -||\tilde{\xi}_e||^2 + \theta||\tilde{\xi}_e||^2 - \theta||\tilde{\xi}_e||^2 + \beta_1||\tilde{\xi}_e|| \\ &\leq -(1-\theta)||\tilde{\xi}_e||^2 \quad \forall \quad ||\tilde{\xi}_e|| \geq \frac{\beta_1}{\theta} \end{aligned} \tag{3.26}$$

where $0 < \theta < 1$. The solutions are then UGB because the conditions of Theorem A.5 are satisfied. Consequently, the conditions of Lemma A.7 are satisfied, which implies that the origin of subsystem $\sum_1$ is UGAS.

Analysis of subsystem 2: In [57] a Lyapunov function is proposed for a third-order observer. It is proven that the Lyapunov function is radially unbounded and positive definite and that it is a Lyapunov function for subsystem $\sum_2$, whose trajectories converge in finite time to the origin $e = 0$ for every value of $|\dot{F}(t)|$ as long as $\dot{F}(t)$ is bounded. Since $\dot{F}(t)$ is bounded by assumptions, the origin is GFTS for every value of $\dot{F}(t)$, which means that the origin is also UGAS by Proposition A.1 and Proposition A.2, which in turn implies $||e(t)|| \leq \beta_2 \forall t \geq 0$.

Analysis of the complete system: To analyse the complete system Lemma A.7 is used. To check if the solutions of the complete system are UGB, the boundedness of $\tilde{\xi}_e$ must be evaluated when $e_1 \neq 0$ and $e_2 \neq 0$, and for this, the Lyapunov function V_{11} is used. Note that the boundedness of $\hat{\sigma}_e$ follows from $\sum_{12}$ being UGAS because $\sum_{12}$ is not perturbed by $\sum_2$.

$$\begin{aligned} \dot{V}_{11}(\tilde{\xi}_e) &= -||\tilde{\xi}_e||^2 + (\hat{\sigma}_e + e_1 + e_2)\tilde{\xi}_e \\ &\leq -||\tilde{\xi}_e||^2 + \theta||\tilde{\xi}_e||^2 - \theta||\tilde{\xi}_e||^2 + (\beta_1 + 2\beta_2)||\tilde{\xi}_e|| \\ &\leq -(1-\theta)||\tilde{\xi}_e||^2 \quad \forall \quad ||\tilde{\xi}_e|| \geq \frac{\beta_1 + 2\beta_2}{\theta} \end{aligned} \tag{3.27}$$

where $0 < \theta < 1$. The solutions are then UGB because the conditions of Theorem A.5 are satisfied. Consequently, the conditions of Lemma A.7 are satisfied, which implies that the complete system is UGAS. □

3.3 Simulation results

In this section, we present simulation results for trajectory tracking using the control laws presented in Section 3.2. We present both results showing the optimal performance of the algorithm, but also realistic ones that are based on using the control gains obtained in experiments during the fall of 2018. We use the simulation model presented in Section 2.2 and the Eelume 2016 version of the robot presented in Section 2.2.1.

3.3.1 Adaptive gains

For implementation purposes, a small boundary is put on σ so the adaptive gains can be expressed as

$$\begin{aligned} \dot{\alpha} &= \begin{cases} \omega_1 \sqrt{\frac{\gamma_1}{2}}, & \text{if } |\sigma| > \alpha_m \\ 0, & \text{if } |\sigma| \leq \alpha_m \end{cases} \\ \beta &= 2\varepsilon\alpha + \lambda + 4\varepsilon^2 \end{aligned} \tag{3.28}$$

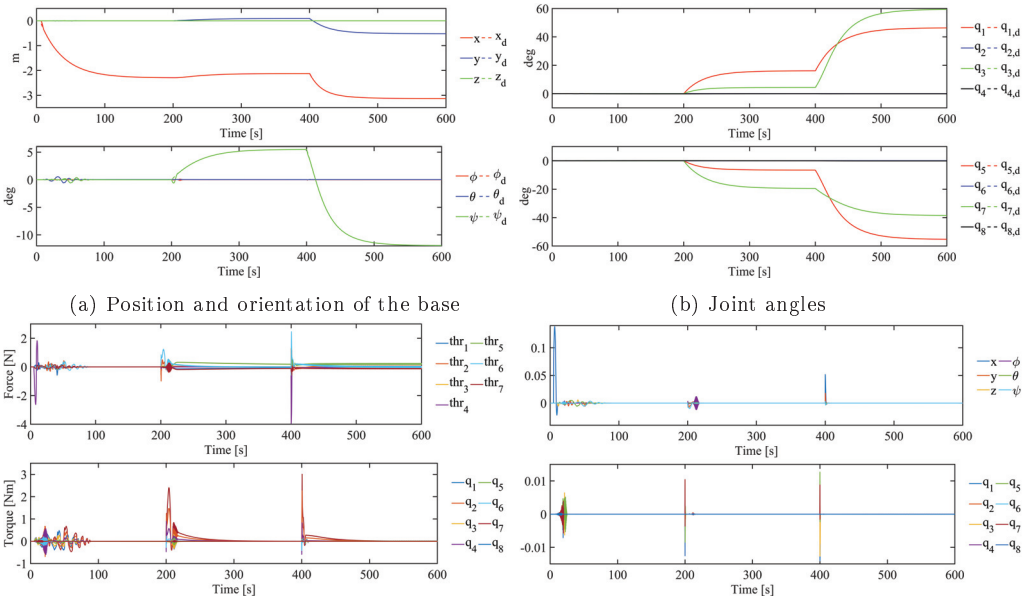
where the design parameter α_m is a small positive constant chosen empirically. This boundary was introduced because numerically σ will never be exactly zero, meaning that in practical implementations the second condition of (3.2) will never be met, which would make the adaptive gains increase to infinity.

3.3.2 Optimal performance

In this section the optimal performance case will be presented, by using the control law from Section 3.2.1. The task that is performed in the simulation is trajectory tracking for the base of the AIAUV. A suitable path for the base to follow is generated by giving set-points to an inverse kinematic controller. The set-points given are for the end-effector of the AIAUV, and the inverse kinematic then generates a reference trajectory for the base and joints, such that the end-effector reaches it target. Three different set-points are given to the inverse kinematic, and they change at 5, 200, 400 seconds. For the simulations, a fixed-step solver with a step size of 10^{-4} was used.

Simulations

Since the STA has an adaptive gain α, the choice of parameters is not that important for the STA. The choice of gains can impact how fast the adaptive gain reaches its optimal value, but it will always reach that value. The gains for the STA were therefore chosen by tuning them manually. Specifically, the gains in the super-twisting algorithm with adaptive gains were set to $\varepsilon_1 = [0.0001e_{14}]^T$, $\lambda_1 = [0.1e_6 \quad 5e_8]^T$, $\gamma_1 = [e_{14}]^T$, $\omega_1 = [8e_{14}]^T$ and $\alpha_m = [0.005e_{14}]^T$, where e_i is



(a) Position and orientation of the base

(b) Joint angles

(c) Thruster forces and joint torques

(d) Sliding surface

Figure 3.1: Optimal simulation results using the control law from Section 3.2.1

a $1 \times i$ vector of ones. In Figure 3.1a, the simulation results for the position and orientation of the base are presented and in Figure 3.1b, the simulation results for the joint angles are presented. In Figure 3.1c the thruster forces and joint torques applied are presented and in Figure 3.1d the sliding surface σ is shown.

Discussion

We can see from Figure 3.1a and Figure 3.1b that the AIAUV follows the given position, orientation and joints trajectories very well. This is also supported by Figure 3.1d, as σ is below 0.15 for the position and orientation and below 0.015 for the joints in absolute value. From Figure 3.1c we can see that the forces used is smooth, i.e. no chattering and below 50 N, which is the limit for the thrusters. That means that the forces used to control the AIAUV is indeed applicable.

3.3.3 Realistic simulations

In this section simulation results based on control gains obtained in experiments during the fall of 2018 will be presented, for both control laws presented in Section 3.2. The test cases used are the C-shape and the C-shape with a moving head. The test cases are explained in Section 2.3.3. To make the simulations a valid comparison with the experiments, we use a P-controller for the joints. The reason for this choice is that the Eelume 2016 robot has an internal joint controller, which is a P-controller. To create a continuous trajectory we use a filter to generate a

continuous trajectory between set-points. For the simulations, the ode3 fixed-step solver with a step size of 0.002 was used.

Simulations

To ensure that the comparison between the simulations and experiments is as fair as possible, the gains used during the simulations are the gains found during the experiments. The gains for the STA with adaptive gains and the HOSMO are presented in Table 3.1. The gain for the HOSMO, L, is different from the one used during the experiments for the C-shape and the C-shape with a moving head because we observed chattering in the simulations when we used $L = 0.01$, which was not experienced in the experiments.

Table 3.1: Simulation: Control gains for the STA with adaptive gains

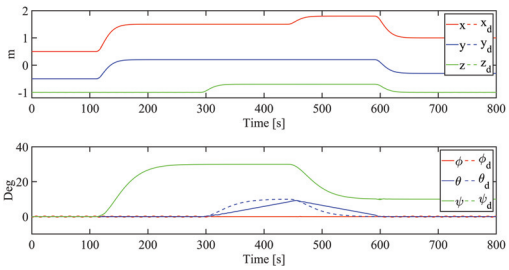
Gains	**Tests**		
	C-shape	C-shape w/moving head	C-shape w/moving head w/HOSMO
ε	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$
λ	0.05	0.05	0.05
γ_1	0.5	0.5	0.5
ω_1	0.4	0.4	0.2
L	0.1	HOSMO not used	0.1

The results when using the control law proposed in Section 3.2.1 are presented in Figure 3.2. Figure 3.3 shows the simulation results when velocity measurements are not available and the control law proposed in Section 3.2.2 is used. Note that if higher gains had been used in the simulations, the results would have improved.

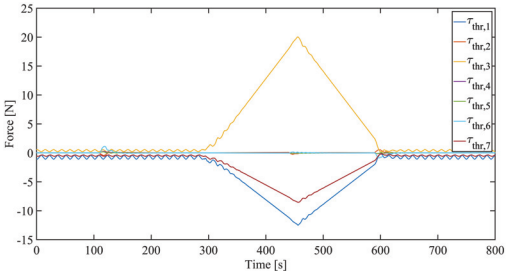
Discussion

As shown in Figure 3.2 and Figure 3.3, the AIAUV follows the given position and orientation very well for both simulation cases, both when the control law from Section 3.2.1 and when the control law with the HOSMO from Section 3.2.2 is used. The AIAUV does have some issues with following the θ reference in both cases, but that might be because there is a small deviation between the desired control input and the control input that we obtain from the thruster allocation scheme. Note that when we, in the simulations, gave the thruster allocation scheme more time to catch up with the reference before it was set to zero again, we observed that the reference was eventually followed. Since we want a back-to-back comparison between simulations and experiments, we kept the same timing in the simulations as in the experiments. We also observe some oscillations in θ and ψ in the C-shape with a moving head case, as shown in Figure 3.2c and Figure 3.3c, and the reason for the oscillations is the moving head; moreover, we observe that the oscillations in the states are consistent with the movement of the head.

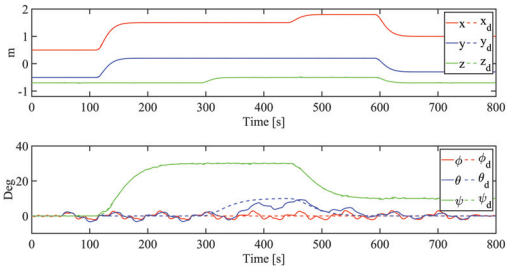
For the C-shape and the C-shape with a moving head, we do observe an improvement in the performance when the HOSMO is used, and the reason for this improvement can be found by comparing Figure 3.2b and Figure 3.3b, and Fig-



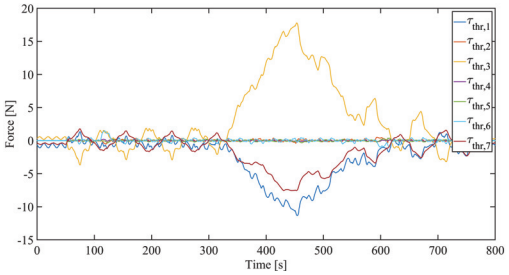
(a) Tracking in C-shape



(b) Thruster forces used in C-shape

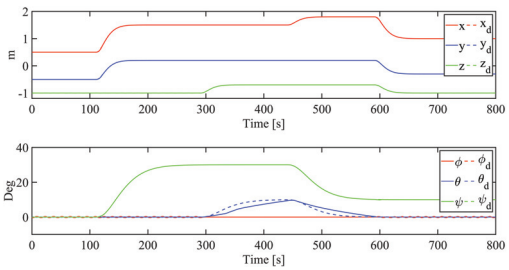


(c) Tracking in C-shape with a moving head

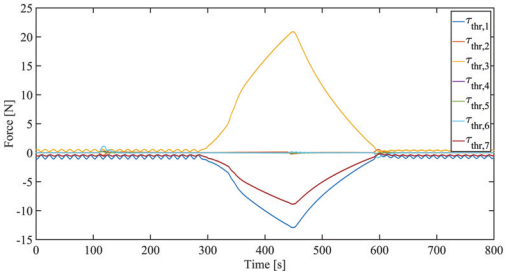


(d) Thruster forces used in C-shape with a moving head

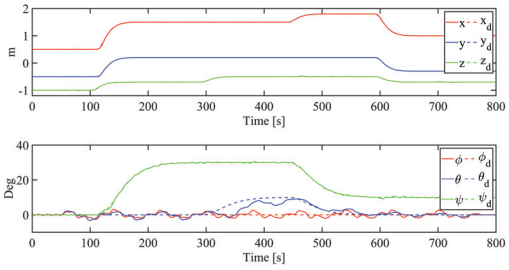
Figure 3.2: Simulation results using the control law from Section 3.2.1



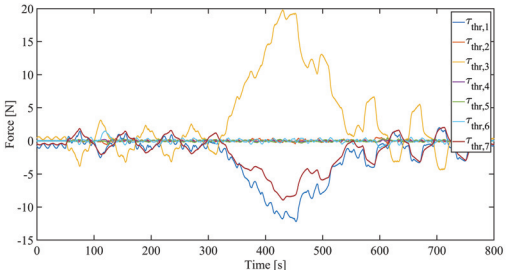
(a) Tracking in C-shape



(b) Thruster forces used in C-shape



(c) Tracking in C-shape with a moving head



(d) Thruster forces used in C-shape with a moving head

Figure 3.3: Simulation results using the control law from Section 3.2.2

ure 3.2d and Figure 3.3d. As shown in these plots, when the HOSMO is used, higher thruster forces are used.

Figure 3.2 and Figure 3.3 also show that for both control schemes the thruster forces used are below 50 N, which is the limit of the thrusters on the actual Eelume robot. This result means that the forces used to control the AIAUV are indeed applicable.

3.4 Experimental investigation

In this section, the experimental results obtained during the fall of 2018 are presented. The purpose of the experiments is to validate the theory and the robustness of the control approaches, by showing that the proposed approaches also work in experiments and not only in the ideal case of the simulations. The robot used is the Eelume 2016 version of the robot presented in Section 2.3.1 and the test cases used are the same as for the simulations, and are explained in Section 2.3.3.

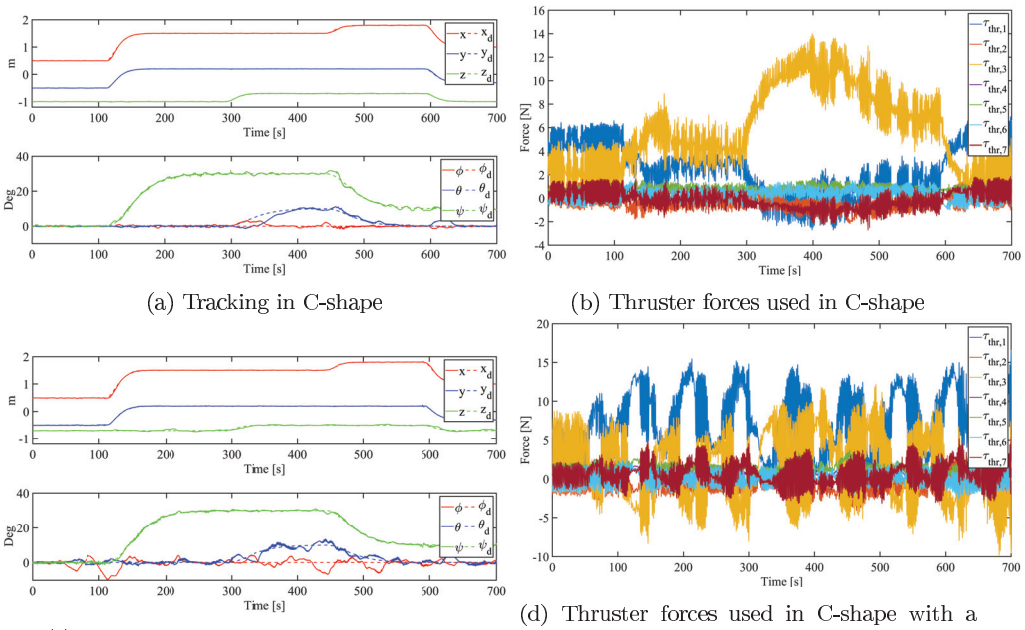
3.4.1 Results

The gains were increased until we observed that rather than following the reference, the AIAUV began to oscillate around the reference. We therefore chose gains that created a small deviation from the reference rather than gains that were more aggressive (higher), where the AIAUV would oscillate around the reference. These oscillations might be caused by delays in the thrusters. We attempted to use the same gains for both test cases, i.e., we attempted to find the best possible gains for the C-shape, and then we used the same for the C-shape with a moving head for a fair comparison. For the STA with adaptive gains, the choice of gains is not very important because the gains will adapt to their best possible values by themselves. We did, however, observe that if we chose the initial adaptive gains to be too high, the Eelume AIAUV started to oscillate. We therefore had to use lower gains when the HOSMO was used in the case of the C-shape with a moving head. The reason for this is probably because when the HOSMO was used, the thruster forces increased. The gains for the STA with adaptive gains and the HOSMO are presented in Table 3.2.

Table 3.2: Experiments: Control gains for the STA with adaptive gains

Gains	**Tests**		
	C-shape	C-shape w/moving head	C-shape w/moving head w/HOSMO
ε	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$
λ	0.05	0.05	0.05
γ_1	0.5	0.5	0.5
ω_1	0.4	0.4	0.2
L	0.01	HOSMO not used	0.01

In Figure 3.4 the results when using the control law proposed in Section 3.2.1 are presented. Note that here the velocity estimates come from the Kalman filter.



(a) Tracking in C-shape

(b) Thruster forces used in C-shape

(c) Tracking in C-shape with a moving head

(d) Thruster forces used in C-shape with a moving head

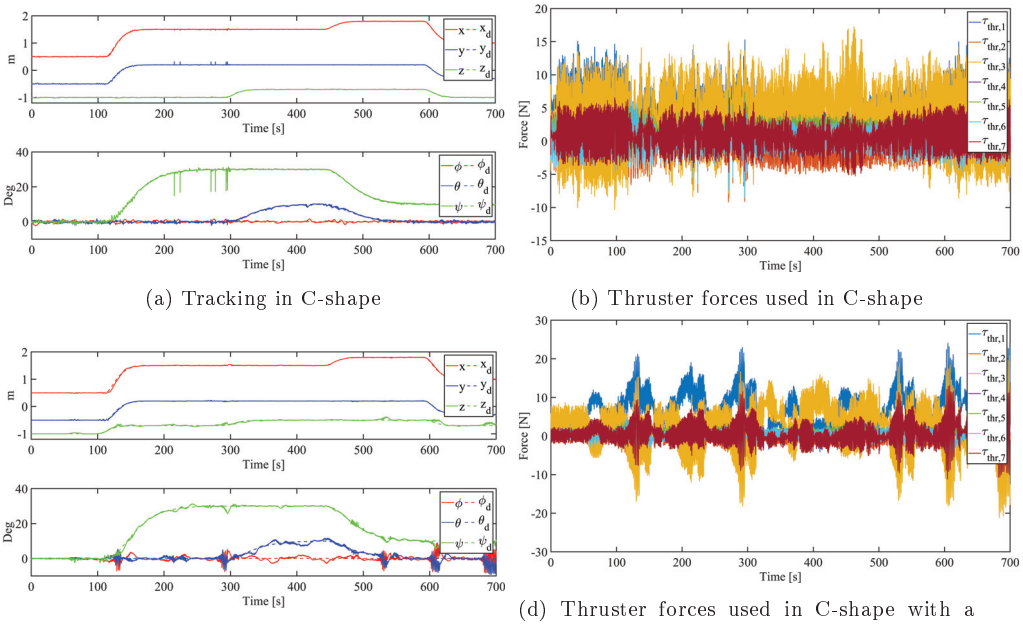
Figure 3.4: Experimental results using the control law from Section 3.2.1

In Figure 3.5 we see the results when velocity measurements are not available and the control law proposed in Section 3.2.2 were used.

3.4.2 Discussion

In the C-shape, we observe from Figure 3.4a that the reference was followed nicely, but we do observe a small deviation from the θ reference caused by a transient, i.e., when the θ reference is changed to 10 deg at 300 s. We also observe a small transient deviation in ϕ at 300 s, which is also caused by the change in θ reference. In the C-shape with a moving head, Figure 3.4c shows that the movement of the head caused oscillations in ϕ and θ, but the position reference was still followed. From Figure 3.4b and Figure 3.4d, we find that there is some chattering in the control input, which could potentially have been reduced by reducing the gains, but if we reduced the gains, we would not obtain the desired tracking performance.

We observed an improvement in the tracking performance when the HOSMO was used without a very large increase in thruster use; this can be seen by comparing Figure 3.4 and Figure 3.5. In the C-shape, we observe from Figure 3.5a that we have almost perfect tracking. In the C-shape with a moving head, we observe from Figure 3.5c some small oscillations in ϕ and θ caused by the movement of the head, but the movement is much smaller than for the case when the HOSMO is not used and is present almost only when the θ reference is changed to 10 deg at 300 s. We also see some rapid oscillations ϕ and θ, which are also caused by the moving head.



(a) Tracking in C-shape

(b) Thruster forces used in C-shape

(c) Tracking in C-shape with a moving head

(d) Thruster forces used in C-shape with a moving head

Figure 3.5: Experimental results using the control law from Section 3.2.2

Exactly why they appear is however, not clear. Additional measurement noise from Qualisys was observed in these tests, but the noise did not affect the performance. Thus, the conclusion would be that the HOSMO works well and creates better performance than the Kalman filter that we used. When we used the HOSMO, we observed an increase in chattering, most likely because some chattering exists in the estimated states from the HOSMO. From the HOSMO errors in Figure 3.6, where the first sub-plot shows $e_1 = x_1 - \hat{x}_1$ and the second sub-plot shows $e_2 = x_2 - \hat{x}_2$, we observe that the errors are quite small and that they appear to be noise, this means that the HOSMO is indeed applicable.

If we compare the results obtained in the simulations (Figure 3.2 and Figure 3.3) and the results obtained in the experiments (Figure 3.4 and Figure 3.5) for both control schemes, we find that they are quite similar. For the C-shape, the trajectories were almost perfectly tracked, whereas for the C-shape with a moving head, we observed some oscillations in some of the states, which were caused by the moving head. For the C-shape and the C-shape with a moving head, we observed an improvement in performance when the HOSMO was used. The tracking performance in the experiments is almost better than the tracking performance in the simulations. The reason for this might be because we used gains that were optimal for the experiments.

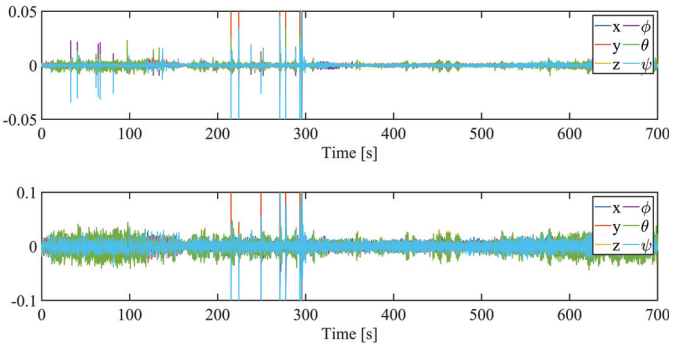


Figure 3.6: Estimation errors for the HOSMO during C-shape (the first sub-plot shows $e_1 = x_1 - \hat{x}_1$ and the second sub-plot shows $e_2 = x_2 - \hat{x}_2$)

3.5 Chapter summary

In this chapter, we have proposed the STA with adaptive gains for solving the trajectory tracking problem of AIAUVs in 6DOF. We have also proposed the STA with adaptive gains together with a HOSMO for solving the trajectory tracking problem of AIAUVs in 6DOF when velocity measurements are not available. Furthermore, we have proved the asymptotic convergence of the tracking error when the control law including the HOSMO is used. We have shown how well the algorithm can perform in an optimal performance example, and we have presented comprehensive simulation and experimental results for two different shapes, representing operation mode. The results validate and verify that the proposed approaches are well suited for control of an AIAUV.

The simulation and experimental results were almost equally good. Some small offsets and oscillations observed in the experiments were not present in the simulations, but that is to be expected since the experiments have outliers and measurement noise.

In the experiments, we observed an increase in performance when the HOSMO was used instead of the Kalman filter. Therefore, the HOSMO is indeed applicable for state estimation.

Chapter 4

Tracking using the Generalized Super-Twisting Algorithm

This chapter investigates the use of the GSTA to solve the trajectory tracking problem for AIAUVs. The tracking problem is also solved in the case that velocity measurements are not available by using the GSTA in combination with a HOSMO. The closed-loop systems are then analysed.

Contributions of this chapter

The contributions of this chapter can be summarized as follows. The trajectory tracking control problem of an AIAUV in 6DOF is solved using the GSTA. The tracking errors are proven to converge asymptotically to zero. Furthermore, the GSTA is proven to give global uniform finite-time stability, whereas in [18], the GSTA was only proven to give global finite-time stability. Comprehensive simulation and experimental results that validate that the approach is well suited for control of an AIAUV are presented. The trajectory tracking control problem of an AIAUV in 6DOF is also solved with the GSTA combined with a HOSMO. Additionally, the tracking errors are proven to converge asymptotically to zero, and comprehensive simulation and experimental results are presented that validate that the approach is well suited for control of an AIAUV.

Organization of this chapter

The chapter is structured as follows. In Section 4.1 the GSTA is given in detail. The control law and observer design for tracking the desired trajectory is presented and analysed in Section 4.2. The simulation results are presented in Section 4.3 and the experimental results are presented in Section 4.4. In Section 4.5 a chapter summary is presented.

Publications

This chapter is based on [10], [15] and [16].

4.1 The generalized super-twisting algorithm

In this section, the equations describing the GSTA are presented in detail. The GSTA is an extension of the STA that provides finite-time convergence in the presence of time- and state-dependent perturbations, which is essential for robust control of the AIAUV. The GSTA proposed in [18] can be written as

$$\begin{aligned} u_{\text{GSTA}} &= -k_1\phi_1(\sigma) + z \\ \dot{z} &= -k_2\phi_2(\sigma) \end{aligned} \tag{4.1}$$

where

$$\begin{aligned} \phi_1(\sigma) &= \lceil\sigma\rfloor^{\frac{1}{2}} + \beta_{\text{GSTA}}\sigma \\ \phi_2(\sigma) &= \frac{1}{2}\lceil\sigma\rfloor^0 + \frac{3}{2}\beta_{\text{GSTA}}\lceil\sigma\rfloor^{\frac{1}{2}} + \beta_{\text{GSTA}}^2\sigma \end{aligned} \tag{4.2}$$

where $\lceil a\rfloor^b = |a|^b \operatorname{sgn}(a)$ and k_1, k_2 and β_{GSTA} are controller gains. With the extra linear term, compared to the STA, three degrees of freedom are obtained in the design of the GSTA gains: k_1, k_2 and β_{GSTA}. The linear growth term $\beta_{\text{GSTA}}\sigma$ in ϕ_1 helps to counteract the effects of state-dependent perturbations, which can exponentially increase in time. By choosing the gains as described in [18], the algorithm is proven to make σ go to zero, globally and in finite time, in the presence of state- and time-dependent uncertain control coefficients and perturbations. Note that the gains, when chosen as described in [18], are defined based on bounds on the perturbations.

4.2 Tracking control laws

In this section, we propose two tracking control laws for the AIAUV based on the GSTA: first, we present one control law where we use the GSTA; then, we present a control law where the GSTA is combined with a HOSMO.

4.2.1 Control law based on the GSTA

In this subsection, the tracking control law for the AIAUV using the GSTA will be presented. Using Lyapunov theory, we will show that $(\tilde{\xi}_q, \tilde{\zeta}) = (0, 0)$ is an asymptotically stable equilibrium point of (2.1) and (2.6) with the proposed control law.

Error dynamics

Define $\tilde{x}_1 = \tilde{\xi}_q$, where $\tilde{\xi}_q$ is given by (2.9) and

$$\tilde{x}_2 = \begin{bmatrix} R_B^I(\tilde{p}) & 0_{3\times3} & 0_{3\times(n-1)} \\ 0_{3\times3} & \frac{1}{2}(\tilde{\eta}I_3 + S(\tilde{\varepsilon})) & 0_{3\times(n-1)} \\ 0_{(n-1)\times3} & 0_{(n-1)\times3} & I_{(n-1)\times(n-1)} \end{bmatrix} (\zeta - \zeta_d) = T(\tilde{p})\tilde{\zeta} \tag{4.3}$$

where ζ and ζ_d are defined in (2.3) and (2.4), respectively. The reason for choosing $\tilde{x}_2 = T(\cdot)\tilde{\zeta}$ is that this makes $\tilde{x}_2 = \dot{\tilde{x}}_1$, and by using that, we can prove that the

error variables asymptotically converge to zero when the sliding surface is equal to zero (see Section 4.2.1 for the proof), which is a requirement when designing the sliding surface. If $\tilde{x}_2$ was chosen to be equal to $\tilde{\zeta}$, then this would not have been the case. By differentiating (2.9) and (4.3) and using $\dot{\zeta}$ from (2.1) and $\dot{\tilde{\xi}}_q$ from (2.6), the error dynamics is

$$\begin{aligned}
\dot{\tilde{x}}_1 &= \tilde{x}_2 \\
\dot{\tilde{x}}_2 &= \frac{d}{dt}\big(T(\tilde{p})\big)T^{-1}(\tilde{p})\tilde{x}_2 + M^{-1}(\tilde{q}+q_d)T(\tilde{p})\Big(- C\big(\tilde{q}+q_d, T^{-1}(\tilde{p})\tilde{x}_2+\zeta_d\big) \\
&\big(T^{-1}(\tilde{p})\tilde{x}_2+\zeta_d\big) - D\big(\tilde{q}+q_d, T^{-1}(\tilde{p})\tilde{x}_2+\zeta_d\big)\big(T^{-1}(\tilde{p})\tilde{x}_2+\zeta_d\big) \\
&- g(\tilde{q}+q_d, R_B^I) + \tau(\tilde{q}+q_d) - M(\tilde{q}+q_d)\dot{\zeta}_d\Big)
\end{aligned} \tag{4.4}$$

To reduce the space used to write the error dynamics, we will introduce some new functions, $f_1(\cdot) = \frac{d}{dt}(T(\cdot))T^{-1}(\cdot)$ and $f_2(\cdot) = -C(\cdot)(T^{-1}(\cdot)\tilde{x}_2+\zeta_d) - D(\cdot)(T^{-1}(\cdot)\tilde{x}_2+\zeta_d) - g(\cdot) - M(\cdot)\dot{\zeta}_d$, such that the error dynamics can be written as

$$\begin{aligned}
\dot{\tilde{x}}_1 &= \tilde{x}_2 \\
\dot{\tilde{x}}_2 &= f_1(\cdot)\tilde{x}_2 + M^{-1}(\cdot)T(\cdot)\big(f_2(\cdot)+\tau(\cdot)\big)
\end{aligned} \tag{4.5}$$

Sliding surface

To use an SMC approach, we must first design a sliding surface. The sliding surface should be designed such that when the sliding variable σ goes to zero, the error variables asymptotically converge to zero and such that the control input $\tau(\cdot)$ appears in the first derivative of σ. The sliding surface is chosen as

$$\sigma_q = \tilde{x}_1 + \tilde{x}_2 \quad \in \mathbb{R}^{6+(n-1)}. \tag{4.6}$$

If $\sigma_q = 0$, we will have $\tilde{x}_1 + \tilde{x}_2 = 0$. Since $\tilde{x}_2 = \dot{\tilde{x}}_1$, we can write this as

$$\dot{\tilde{x}}_1 = -\tilde{x}_1 \tag{4.7}$$

which ensures that $\tilde{x}_1$ globally exponentially converges to zero. Since $\tilde{x}_1 = \tilde{\xi}_q$, the original state variable $\tilde{\xi}_q$ will also globally exponentially converge to zero if $\sigma_q = 0$. The GSTA, which is described in Section 4.1, is then used to drive the sliding surface to zero.

Control input

In this section, we design a control law based on the GSTA described in Section 4.1, which we show achieves the tracking control objective in Section 4.2.1. By using the fact that $\dot{\tilde{x}}_1 = \tilde{x}_2$ from (4.4), (4.6) can be written as

$$\dot{\tilde{x}}_1 = \sigma_q - \tilde{x}_1. \tag{4.8}$$

By differentiating (4.6), we obtain

$$\dot{\sigma}_q = \dot{\tilde{x}}_1 + \dot{\tilde{x}}_2 = \tilde{x}_2 + f_1(\cdot)\tilde{x}_2 + M^{-1}(\cdot)T(\cdot)\big(f_2(\cdot)+\tau(\cdot)\big) \tag{4.9}$$

and by using $\tilde{x}_2 = \sigma_q - \tilde{x}_1$, we obtain

$$\dot{\sigma}_q = \sigma_q - \tilde{x}_1 + f_1(\cdot)(\sigma_q + \tilde{x}_1) + M^{-1}(\cdot)T(\cdot)\big(f_2(\cdot) + \tau(\cdot)\big). \tag{4.10}$$

Now, we want the control input $\tau(\cdot)$ to be chosen such that $\dot{\sigma}_q = u_{\text{GSTA}}$, such that σ_q and $\dot{\sigma}_q$ reach zero in finite time. However, since we do not know $f_1(\cdot)$ and $f_2(\cdot)$, we choose our control input to be

$$\tau(\cdot) = T^{-1}(\cdot)u_{\text{GSTA}} \tag{4.11}$$

and in Section 4.2.1, we will show that this control input achieves the tracking control objective.

Stability analysis

In this section, we will analyse the closed-loop system resulting from the control law proposed and we show that the tracking error converges asymptotically to zero. Parts of the analysis build upon the results obtained in [18], but we extend these results by proving uniformity for the GSTA, which makes it possible to show that the tracking errors for the complete system converge asymptotically to zero.

By introducing $\varphi(\sigma_q, \tilde{x}_1, t) = \varphi_1(\sigma_q, \tilde{x}_1, t) + \varphi_2(\sigma_q, \tilde{x}_1, t)$, where $\varphi_1(0, \tilde{x}_1, t) = 0$, $\Gamma(\cdot) = M^{-1}(\cdot)$, and using $\tau(\cdot)$ as in (4.11), we can rewrite (4.10) as

$$\dot{\sigma}_q = -k_1\Gamma(\cdot)\phi_1(\sigma_q) + \varphi_1(\sigma_q, \tilde{x}_1, t) + \Gamma(\cdot)\big(z + \Gamma^{-1}(\cdot)\varphi_2(\sigma_q, \tilde{x}_1, t)\big) \tag{4.12}$$

where $\varphi_1(\sigma_q, \tilde{x}_1, t) = \sigma_q + f_1(\cdot)\sigma_q + \Gamma(\cdot)(-C(\cdot)\sigma_q - D(\cdot)\sigma_q)$ and $\varphi_2(\sigma_q, \tilde{x}_1, t) = -\tilde{x}_1 + f_1(\cdot)\tilde{x}_1 + \Gamma(\cdot)(-C(\cdot)(\tilde{x}_1 + T(\cdot)\zeta_d) - D(\cdot)(\tilde{x}_1 + T(\cdot)\zeta_d) - T(\cdot)g(\cdot) - T(\cdot)M(\cdot)\dot{\zeta}_d)$. Then, by means of (4.8) and setting $\sigma_{q,1} = \sigma_q$ and $\sigma_{q,2} = z + \Gamma^{-1}(\cdot)\varphi_2(\sigma_q, \tilde{x}_1, t)$, we can write the overall closed-loop dynamics as

$$\begin{aligned} \sum_1 \Big\{ \quad & \dot{\tilde{x}}_1 = -\tilde{x}_1 + \sigma_{q,1} \\ \sum_2 \Big\{ \quad & \dot{\sigma}_{q,1} = -k_1\Gamma(\cdot)\phi_1(\sigma_{q,1}) + \varphi_1(\sigma_{q,1}, \tilde{x}_1, t) + \Gamma(\cdot)\sigma_{q,2} \\ & \dot{\sigma}_{q,2} = -k_2\phi_2(\sigma_{q,1}) + \tfrac{d}{dt}\big(\Gamma^{-1}(\cdot)\varphi_2(\sigma_{q,1}, \tilde{x}_1, t)\big) \end{aligned} \tag{4.13}$$

Theorem 4.1. *Consider the error dynamics given by (4.4), and let the sliding surface σ_q be defined by (4.6). Let the control input be given by (4.11). Then, the closed-loop dynamics are described by (4.13), and the origin of this cascade system is UGAS, which ensures the asymptotic convergence of the tracking error when $0 < k_m I \leq \Gamma(\cdot) \leq k_M I$, $|\varphi_1(\cdot)| \leq \alpha|\phi_1(\sigma_q)|$ and $|\frac{d}{dt}(\Gamma^{-1}(\cdot)\varphi_2(\cdot))| \leq \Delta$, where k_m, k_M, α and Δ are positive constants.*

Proof. To analyse the cascade system (4.13), Lemma A.7 will be used. Note that the system is actually interconnected, but since subsystem 1 is well behaved as long as σ_q does not explode, i.e. $\tilde{x}_1$ is bounded, the system can be analysed with cascaded theory by analysing along the trajectories with $\tilde{x}_1(t)$ bounded. When analysing the complete system, we will prove that this is indeed the case, i.e. prove that $\tilde{x}_1(t)$ is UGB. We first start by analysing subsystem 1 without perturbations.

Analysis of subsystem 1 with $\sigma_{q,1} = 0$: With $\sigma_{q,1} = 0$, subsystem 1 can be written as

$$\sum_1 \Big\{ \quad \dot{\tilde{x}}_1 = -\tilde{x}_1 \tag{4.14}$$

This is clearly a globally exponentially stable linear system, but since we will need a Lyapunov function to analyse this system when $\sigma_{q,1} \neq 0$, we use the Lyapunov function candidate $V_1(\tilde{x}_1) = \frac{1}{2}\tilde{x}_1^2$ for the analysis. The derivative of V_1 yields

$$\dot{V}_1(\tilde{x}_1) = \tilde{x}_1\dot{\tilde{x}}_1 = \tilde{x}_1(-\tilde{x}_1) = -\tilde{x}_1^2 \leq -||\tilde{x}_1||^2 \tag{4.15}$$

This means that the Lyapunov function satisfies

$$\begin{aligned} &k_1||\tilde{x}_1||^a \leq V_1(\tilde{x}_1) \leq k_2||\tilde{x}_1||^a \\ &\frac{\partial V_1}{\partial x} f(t,x) \leq -k_3||\tilde{x}_1||^a \end{aligned} \tag{4.16}$$

with $k_1 = k_2 = \frac{1}{2}$, $k_3 = 1$ and $a = 2$. Hence, by virtue of Theorem A.4, the origin for subsystem $\sum_1$ with $\sigma_{q,1} = 0$ is GES.

Analysis of subsystem 2: Subsystem $\sum_2$ has the same structure as the system in [18]; thus, we can use the results obtained in [18]. For completeness, and because we will base the further analysis on the Lyapunov function obtained here, we recall the results from [18]. In [18], the origin of the system is proven to be GFTS if $0 < k_m I \leq \Gamma(\cdot) \leq k_M I$, $|\varphi_1(\cdot)| \leq \alpha|\phi_1(\sigma_q)|$ and $|\frac{d}{dt}(\Gamma^{-1}(\cdot)\varphi_2(\cdot))| \leq \Delta$, where k_m, k_M, α and Δ are positive constants. Since the system is GFTS, it is also globally asymptotically stable (Definition A.2) by Proposition A.2. To prove that the origin of subsystem 2 is UGAS or GUFTS, we will use Theorem A.6. In the analysis, we will use the Lyapunov function found in [18], which was used to prove GFTS, but the analysis itself, i.e., proving uniformity by using Theorem A.6 is novel. The function $V = \xi^T P \xi$, where $\xi^T = [\phi_1(\sigma_{q,1}) \quad \sigma_{q,2}]$ and $P = \begin{bmatrix} p_1 & -1 \\ -1 & p_2 \end{bmatrix}$, $p_1 p_2 > 1$, is the generalized Lyapunov function for subsystem 2; see [18] for details. This function is globally proper and continuous (but not Lipschitz continuous on the line $\sigma_{q,1} = 0$). For $\sigma_{q,1} \neq 0$, this function is differentiable and

$$DV_{F(\sigma_{q,1},\sigma_{q,2})}(\sigma_{q,1},\sigma_{q,2}) \leq -\mu_1\sqrt{V(\sigma_{q,1},\sigma_{q,2})} \tag{4.17}$$

where $\mu_1 > 0$ and

$$\begin{pmatrix} \dot{\sigma}_{q,1} \\ \dot{\sigma}_{q,2} \end{pmatrix} \in F(\sigma_{q,1},\sigma_{q,2}) = \begin{pmatrix} -k_1\Gamma(\cdot)\phi_1(\sigma_{q,1}) + \varphi_1(\sigma_{q,1},\tilde{x}_1,t) + \Gamma(\cdot)\sigma_{q,2} \\ -k_2\phi_2(\sigma_{q,1}) + \frac{d}{dt}\big(\Gamma^{-1}(\cdot)\varphi_2(\sigma_{q,1},\tilde{x}_1,t)\big) \end{pmatrix} \tag{4.18}$$

For $\sigma_{q,1}1 = 0$ and $\sigma_{q,2} \neq 0$, we need to calculate a generalized directional derivative. Thus, consider the limit

$$D_{\{h_n\},\{u_n\}}V(0,\sigma_{q,2}) = \lim_{n\to\infty} \frac{V(h_n u_n^{\sigma_{q,1}}, \sigma_{q,2} + h_n u_n^{\sigma_{q,2}}) - V(0,\sigma_{q,2})}{h_n} \tag{4.19}$$

where $\{h_n\} \in \mathbb{K}$ ($\mathbb{K}$ is a set of all sequences of real numbers converging to zero, i.e., $\{h_n\} \in \mathbb{K} \Leftrightarrow h_n \to 0, h_n \neq 0$), $u_n = (u_n^{\sigma_{q,1}}, u_n^{\sigma_{q,2}})^T$, $\{u_n\} \in \mathbb{M}(d)$ ($\mathbb{M}(d)$ is a

set of all sequences of real vectors converging to $d \in \mathbb{R}^n$, i.e. $\{u_n\} \in \mathbb{M}(d) \Leftrightarrow u_n \to d, u_n \in \mathbb{R}^n)$, and $d \in F(0, \sigma_{q,2})$. In this case, $u_n^{\sigma_{q,1}} \to \sigma_{q,2}$ and $u_n^{\sigma_{q,2}} \to q$, where $q \in [-\frac{1}{2}k_2 \pm \Delta, \frac{1}{2}k_2 \pm \Delta]$. Hence,

$$\begin{aligned} D_{\{h_n\},\{u_n\}}V(0,\sigma_{q,2}) &= \lim_{n\to\infty} \frac{V(h_n\sigma_{q,2}, \sigma_{q,2}+h_nq) - V(0,\sigma_{q,2})}{h_n} \\ &= \lim_{n\to\infty} \Big(p_1\big(|h_n\sigma_{q,2}|^{(1/2)}\operatorname{sgn}(h_n\sigma_{q,2}) + \beta h_n\sigma_{q,2}\big)^2 - 2\big(|h_n\sigma_{q,2}|^{(1/2)}\operatorname{sgn}(h_n\sigma_{q,2}) \\ &+ \beta h_n\sigma_{q,2}\big)(\sigma_{q,2}+h_nq) + p_2(\sigma_{q,2}+h_nq)^2 - p_2\sigma_{q,2}^2\Big)/h_n = -\infty \end{aligned} \tag{4.20}$$

Therefore,

$$D_{F(\sigma_{q,1},\sigma_{q,2})}V(0,\sigma_{q,2}) = \{-\infty\} \leq -\mu_1\sqrt{V(0,\sigma_{q,2})} \text{ for } \sigma_{q,2} \neq 0 \tag{4.21}$$

and the origin of subsystem 2 is therefore GUFTS Theorem A.6, and therefore, it is also UGAS. This result implies that $||\sigma_q(t)|| < \beta \ \forall t \geq 0$.

Analysis of the complete system: To analyse the complete system, Lemma A.7 is used. To check whether the solutions of the complete system are UGB, the boundedness of $\tilde{x}_1$ must be evaluated when $\sigma_{q,1} \neq 0$. The derivative of the Lyapunov function V_1 is then as follows:

$$\begin{aligned} \dot{V}_1(\tilde{x}_1) &= -||\tilde{x}_1||^2 + \sigma_{q,1}\tilde{x}_1 \\ &\leq -||\tilde{x}_1||^2 + \theta||\tilde{x}_1||^2 - \theta||\tilde{x}_1||^2 + \beta||\tilde{x}_1|| \\ &\leq -(1-\theta)||\tilde{x}_1||^2 \quad \forall \quad ||\tilde{x}_1|| \geq \frac{\beta}{\theta} \end{aligned} \tag{4.22}$$

where $0 < \theta < 1$. The solutions are then UGB because the conditions of Theorem A.5 are satisfied. Consequently, the conditions of Lemma A.7 are satisfied, which implies that the origin of the complete system is UGAS. □

One way for the inequalities in Theorem 4.1 to be satisfied is if the assumptions in the following theorem are satisfied.

Theorem 4.2. *Consider the closed-loop system in* (4.13)*. If the following assumptions are satisfied:*

Assumption 4.1. The AIAUV is neutrally buoyant.

Assumption 4.2. The AIAUV has only revolute joints.

Assumption 4.3. The reference trajectory and its derivatives are continuous and bounded by design.

Assumption 4.4. The matrix $||\frac{d^2}{dt^2}T(\cdot)|| \leq T_M$ is bounded, where $T(\cdot)$ is defined in (4.3), the Coriolis-centripetal matrix is bounded by $||C(\cdot)|| \leq C_M||x_2||$ and $||\frac{d}{dt}C(\cdot)|| \leq C_m||x_2||$, the damping matrix is bounded by $||D(\cdot)|| \leq D_M||x_2||$ and $||\frac{d}{dt}D(\cdot)|| \leq D_m||x_2||$, and the matrix of gravitational and buoyancy forces is bounded by $||\frac{d}{dt}g(\cdot)|| \leq g_M||x_2||$.

Assumption 4.5. $\tilde{x}_2(t)$ is bounded.

then positive constants k_m, k_M, α and Δ exist such that

1. *Inequality 1:* $0 < k_m I \leq \Gamma(\cdot) \leq k_M I$
2. *Inequality 2:* $|\varphi_1(\cdot)| \leq \alpha|\phi_1(\sigma)|$
3. *Inequality 3:* $|\frac{d}{dt}\big(\Gamma^{-1}(\cdot)\varphi_2(\cdot)\big)| \leq \Delta$

are satisfied.

Remark 4.1. Assumptions 1-3 are valid due to the design of AIAUVs and their reference trajectories. Assumptions 4-5 are valid since the AIAUV is a mechanical system.

Proof. To be able to prove that the above inequalities are satisfied, we first note some properties that arise from having revolute joints [4]:

1. Property 1: $0 < \lambda_{\min}(M) \leq ||M|| \leq \lambda_{\max}(M)$
2. Property 2: $M = M^T > 0$
3. Property 3: $\dot{M} = C + C^T$ and $\zeta^T(\dot{M} - 2C)\zeta = 0 \quad \forall \quad \zeta \in \mathbb{R}^{6+(n-1)}$

Proof of Inequality 1: $0 < k_m I \leq \Gamma(\cdot) \leq k_M I$

Since $\Gamma(\cdot) = M^{-1}(\cdot)$, we need to prove that

$$0 < k_m I \leq M^{-1}(\cdot) \leq k_M I \tag{4.23}$$

where k_m and k_M are positive constants. From Property 1, we have that M is lower and upper bounded; the inverse will therefore also be lower and upper bounded by $0 < 1/\lambda_{\max}(M) \leq ||M^{-1}|| \leq 1/\lambda_{\min}(M)$, which means that $k_m = 1/\lambda_{\max}(M)$ and $k_M = 1/\lambda_{\min}(M)$. Inequality 1 is therefore satisfied.

Proof of Inequality 2: $|\varphi_1(\cdot)| \leq \alpha|\phi_1(\sigma)|$

Since $\varphi_1(\sigma, \tilde{x}_1, t) = \sigma + f_1(\cdot)\sigma + \Gamma(\cdot)(-C(\cdot)\sigma - D(\cdot)\sigma)$ with $f_1(\cdot) = \frac{d}{dt}(T(\cdot))T^{-1}(\cdot)$, we need to prove that

$$|\sigma + f_1(\cdot)\sigma + \Gamma(\cdot)\big(-C(\cdot)\sigma - D(\cdot)\sigma\big)| \leq \alpha|\phi_1(\sigma)| = \alpha|\lceil\sigma\rfloor^{\frac{1}{2}} + \beta_{\text{GSTA}}\sigma| \tag{4.24}$$

By rewriting

$$|1 + f_1(\cdot) + \Gamma(\cdot)\big(-C(\cdot) - D(\cdot)\big)||\sigma| \leq \alpha|\phi_1(\sigma)| = \alpha|\lceil\sigma\rfloor^{\frac{1}{2}} + \beta_{\text{GSTA}}\sigma| \tag{4.25}$$

we find that if

$$|1 + f_1(\cdot) + \Gamma(\cdot)\big(-C(\cdot) - D(\cdot)\big)| \leq \alpha, \tag{4.26}$$

the inequality holds. Now, $T(\cdot)$ is a matrix that contains the rotation matrix R_B^I, the identity matrix and the expression $(1/2)(\tilde{\eta}I_3 + S(\tilde{\varepsilon}))$, which comes from $J_{k,oq}(\tilde{p})$. Since they are all bounded, the matrix $T(\cdot)$ will also be bounded. The matrix $T(\cdot)$ is also nonsingular since quaternions are used, which means that $T^{-1}(\cdot)$ exists and will also be bounded. By taking the derivative of $T(\cdot)$, we find that for $\frac{d}{dt}(T(\cdot))$ to be bounded, $\tilde{x}_2(t)$ needs to be bounded, which it is by assumption. The function $f_1(\cdot)$ is therefore a function of bounded signals, and $f_1(\cdot)$ is thus bounded. The function $\Gamma(\cdot)$ is found to be bounded in the proof of Inequality 1. The matrices

$C(\cdot)$ and $D(\cdot)$ are bounded by assumption as long as $\tilde{x}_2(t)$ is bounded, which is bounded by assumption. The matrices $C(\cdot)$ and $D(\cdot)$ are therefore bounded, and since all the functions on the right-hand side of (4.26) are bounded, the inequality holds. Equation (4.24) is therefore satisfied, and thus, Inequality 2 is satisfied.

Proof of Inequality 3: $|\frac{d}{dt}\big(\Gamma^{-1}(\cdot)\varphi_2(\cdot)\big)| \leq \Delta$
Since $\varphi_2(\sigma, \tilde{x}_1, t) = -\tilde{x}_1 + f_1(\cdot)\tilde{x}_1 + \Gamma(\cdot)(-C(\cdot)(\tilde{x}_1 + T(\cdot)\zeta_d) - D(\cdot)(\tilde{x}_1 + T(\cdot)\zeta_d) - T(\cdot)g(\cdot) - T(\cdot)M(\cdot)\dot{\zeta}_d)$, we need to prove

$$
\begin{aligned}
|\frac{d}{dt}\Big(\Gamma^{-1}(\cdot)\Big(&- \tilde{x}_1 + f_1(\cdot)\tilde{x}_1 + \Gamma(\cdot)\big(- C(\cdot)(\tilde{x}_1 + T(\cdot)\zeta_d) \\
&- D(\cdot)(\tilde{x}_1 + T(\cdot)\zeta_d) - T(\cdot)g(\cdot) - T(\cdot)M(\cdot)\dot{\zeta}_d\big)\Big)\Big)| \leq \Delta
\end{aligned}
\tag{4.27}
$$

By differentiating, we find that for the above to hold, we need $\frac{d}{dt}f_1(\cdot)$, $\frac{d}{dt}\Gamma(\cdot)$, $\frac{d}{dt}C(\cdot)$, $\frac{d}{dt}D(\cdot)$, $g(\cdot)$, $\frac{d}{dt}g(\cdot)$ and $\frac{d}{dt}M(\cdot)$ to be bounded since $\tilde{x}_1(t)$, $\tilde{x}_2(t)$, $f_1(\cdot)$, $\Gamma(\cdot)$, $T(\cdot)$, $M(\cdot)$, $C(\cdot)$, $D(\cdot)$, $\frac{d}{dt}(T(\cdot))$, ζ_d, $\dot{\zeta}_d$ and $\ddot{\zeta}_d$ have previously been proven to be bounded or are bounded by assumption. For the function $\frac{d}{dt}f_1(\cdot)$ to be bounded, we need for the matrix $\frac{d^2}{dt^2}T(\cdot)$ to be bounded, which it is by assumption; thus, $\frac{d}{dt}f_1(\cdot)$ is bounded. The time derivative $\frac{d}{dt}\Gamma(\cdot)$ is bounded if $\frac{d}{dt}M(\cdot)$ and $M(\cdot)$ are bounded. Since $C(\cdot)$ is bounded, $\frac{d}{dt}M(\cdot)$ is bounded (from Property 3), and $M(\cdot)$ is bounded by Property 1. The function $\frac{d}{dt}\Gamma(\cdot)$ is therefore bounded. Furthermore, $\frac{d}{dt}C(\cdot)$ and $\frac{d}{dt}D(\cdot)$ are bounded since $\tilde{x}_2(t)$ is bounded by assumption. The matrix $g(\cdot)$ is bounded since the AIAUV is neutrally buoyant, and $\frac{d}{dt}g(\cdot)$ is bounded by assumption since $\tilde{x}_2(t)$ is bounded. Now, since (4.27) is satisfied, Inequality 3 is satisfied. □

Remark 4.2. If the assumptions in Theorem 4.2 are satisfied, the inequalities in Theorem 4.1 are satisfied if the positive parameters α, k_m, k_M and Δ are chosen according to inequalities (4.23), (4.26) and (4.27) given in the proof, and the procedure in [18] can then be used to choose the gains k_1, k_2 and β_{GSTA} in (4.1) and (4.2), which will ensure the finite-time convergence.

4.2.2 Control law based on the GSTA combined with a higher-order sliding mode observer

In this subsection, the tracking control law for the AIAUV using the GSTA combined with the HOSMO will be presented. Using Lyapunov theory, we will show that $(\tilde{\xi}_e, \tilde{\zeta}) = (0, 0)$ is an asymptotically stable equilibrium point of (2.1) and (2.11) with the proposed control law. We here use the Euler angles representation of the model (Section 2.1.2) when we create the control law. The reason is that the HOSMO does not work with quaternions, as there is then a different number of states in position versus velocity.

Third-order sliding mode observer

Because velocity measurements are unavailable, a HOSMO will be developed in this subsection for state estimation. We want to use the third-order SMO presented in

[46] because the third-order SMO has been proven to be FTS in [57]. To use this HOSMO, we introduce a change of variables. Define $x_1 = \xi_e$, where ξ_e is given by (2.10), and $x_2 = J(\eta_2)\zeta$, where $J(\eta_2)\zeta$ is given by (2.11); then, the dynamics can be written as

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{d}{dt}(J(\eta_2))J^{-1}(\eta_2)x_2 + M^{-1}(q)J(\eta_2)\big(-C\big(q, J^{-1}(\eta_2)x_2\big)J^{-1}(\eta_2)x_2 \\ &- D\big(q, J^{-1}(\eta_2)x_2\big)J^{-1}(\eta_2)x_2 - g(q, R_B^I) + \tau(q)\big)\end{aligned} \tag{4.28}$$

Assumption 4.6. We assume that $\frac{d}{dt}(J(\cdot))J^{-1}(\cdot)x_2$ is a small bounded disturbance, which we will call $d(t)$.

Remark 4.3. For all practical purposes, Assumption 4.6 will be satisfied since the AIAUV is a mechanical system and has a limited control input, which will cause the velocities of the system to be bounded. If the velocities are bounded $\frac{d}{dt}(J(\cdot))$ will be small and bounded. Furthermore, $J^{-1}(\cdot)$ is bounded since it consists of elements with cos and sin. For theoretical purposes, it should be proven that this assumption hold; however, this proof will be left as a topic for future work.

We also introduce a new function $f(\cdot) = -C(\cdot)J^{-1}(\cdot)x_2 - D(\cdot)J^{-1}(\cdot)x_2 - g(\cdot)$ to reduce the space used to write the model. The model can then be written as

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= d(t) + M^{-1}(\cdot)J(\cdot)\big(f(\cdot) + \tau(\cdot)\big)\end{aligned} \tag{4.29}$$

Now that we have introduced the change of variables, the HOSMO can be introduced. By designing the HOSMO structure as in [46], the HOSMO is chosen as

$$\begin{aligned}\dot{\hat{x}}_1 &= \hat{x}_2 + z_1, \\ \dot{\hat{x}}_2 &= \hat{x}_3 + z_2 + M^{-1}(\cdot)J(\cdot)\tau(\cdot), \\ \dot{\hat{x}}_3 &= z_3,\end{aligned} \tag{4.30}$$

where

$$\begin{aligned}z_1 &= k_1|e_1|^{2/3}\operatorname{sgn}(e_1) \\ z_2 &= k_2|e_1|^{1/3}\operatorname{sgn}(e_1) \\ z_3 &= k_3\operatorname{sgn}(e_1)\end{aligned} \tag{4.31}$$

and $k_1 \in \mathbb{R}^{6+(n-1)}$, $k_2 \in \mathbb{R}^{6+(n-1)}$ and $k_3 \in \mathbb{R}^{6+(n-1)}$ are gains to be chosen according to [48] and [49], where $e_1 = x_1 - \hat{x}_1 \in \mathbb{R}^{6+(n-1)}$. One choice of parameters that satisfies the requirements in [48] and [49] is, according to [19], $k_1 = 6L^{1/3}$, $k_2 = 11L^{1/2}$ and $k_3 = 6L$, where $L \in \mathbb{R}^{6+(n-1)}$ is a sufficiently large constant. By defining $e_2 = x_2 - \hat{x}_2$ and $e_3 = -\hat{x}_3 + F(\cdot)$, where $F(\cdot) = d(t) + M^{-1}(\cdot)J(\cdot)f(\cdot)$, the error dynamics of the HOSMO can be written as

$$\begin{aligned}\dot{e}_1 &= -k_1|e_1|^{2/3}\operatorname{sgn}(e_1) + e_2 \\ \dot{e}_2 &= -k_2|e_1|^{1/3}\operatorname{sgn}(e_1) + e_3. \\ \dot{e}_3 &= -k_3\operatorname{sgn}(e_1) + \dot{F}(\cdot)\end{aligned} \tag{4.32}$$

If $|\dot{F}(\cdot)| < \Delta$, then the HOSMO errors go to zero in finite time [57]. Since $F(\cdot)$ is a combination of $d(t)$, $C(\cdot)J^{-1}(\cdot)x_2$, $D(\cdot)J^{-1}(\cdot)x_2$ and $g(\cdot)$, and since the AIAUV is a mechanical system, these matrices will not change infinitely fast. Therefore, assuming that $\dot{F}(\cdot)$ is bounded is a valid assumption.

Sliding surface

As previously discussed, to use an SMC approach, we must first design a sliding surface. The sliding surface should be designed such that when the sliding variable σ goes to zero, the state variables asymptotically converge to zero and such that the control input $\tau(\cdot)$ appears in the first derivative of σ. If we had velocity measurements available, we would choose the sliding surface as

$$\sigma_e = \tilde{x}_1 + \tilde{x}_2, \quad \in \mathbb{R}^{6+(n-1)} \tag{4.33}$$

where $\tilde{x}_1 = \tilde{\xi}_e$, with $\tilde{\xi}_e$ defined as in (2.12), and $\tilde{x}_2 = J_e(\eta_2)\zeta - J_e(\eta_{2,d})\zeta_d$, where $J_e(\eta_2)\zeta$ is defined in (2.11). Now, if $\sigma_e = 0$, we will have $\tilde{x}_1 + \tilde{x}_2 = 0$. Since $\tilde{x}_2 = x_2 - x_{2,d} = \dot{x}_1 - \dot{x}_{1,d} = \dot{\tilde{x}}_1$, we can write this as

$$\dot{\tilde{x}}_1 = -\tilde{x}_1 \tag{4.34}$$

which will ensure that $\tilde{x}_1$ globally exponentially converges to zero. Now, since $\tilde{x}_1 = \tilde{\xi}_e$, the state variables $\tilde{\xi}_e$ will also globally exponentially converge to zero if $\sigma_e = 0$. Since the velocity measurement is not available, the observed state values are used, and we can therefore write the sliding surface with the observed values as

$$\hat{\sigma}_e = \hat{\tilde{x}}_1 + \hat{\tilde{x}}_2, \quad \in \mathbb{R}^{6+(n-1)} \tag{4.35}$$

where $\hat{\tilde{x}}_1 = \hat{x}_1 - \xi_{e,d}$ and $\hat{\tilde{x}}_2 = \hat{x}_2 - J_e(\eta_{2,d})\zeta_d$. Since the HOSMO errors in (4.32) go to zero in finite time, $\hat{\sigma}_e = \sigma_e$ after some finite time. Thus, if $\hat{\sigma}_e = 0$, the tracking objective will be satisfied.

Control input

In this section, we will design a control law based on the GSTA described in Section 4.1, which we will show achieves the tracking control objective in Section 4.2.2. By designing the control input $\tau(\cdot)$ such that $\dot{\hat{\sigma}}_e = u_{\text{GSTA}}$, we thus achieve that $\hat{\sigma}_e$ and $\dot{\hat{\sigma}}_e$ reach zero in finite time since the GSTA is FTS. Taking the time derivative of (4.35) and substituting $\dot{\hat{x}}_1$ and $\dot{\hat{x}}_2$, defined in (4.30), we find that

$$\begin{aligned} \dot{\hat{\sigma}}_e &= \dot{\hat{\tilde{x}}}_1 + \dot{\hat{\tilde{x}}}_2 = \dot{\hat{x}}_1 - \dot{x}_{1,d} + \dot{\hat{x}}_2 - \dot{x}_{2,d} \\ &= \hat{x}_2 + z_1 - \dot{x}_{1,d} + \hat{x}_3 + z_2 + M^{-1}(\cdot)J(\cdot)\tau(\cdot) - \dot{x}_{2,d} \end{aligned} \tag{4.36}$$

By choosing $\tau(\cdot)$ to be

$$\tau(\cdot) = J^{-1}(\cdot)M(\cdot)\big(-\hat{x}_2 - z_1 + \dot{x}_{1,d} - \hat{x}_3 - z_2 + \dot{x}_{2,d} + u_{\text{GSTA}}\big) \tag{4.37}$$

we obtain

$$\dot{\hat{\sigma}}_e = u_{\text{GSTA}}. \tag{4.38}$$

Stability analysis

In this section, we will analyse the closed-loop system resulting from the control law proposed and we show that the tracking error converges asymptotically to zero. We consider the closed-loop system (2.1), (2.11), (4.37). By using the facts that $\hat{x}_1 = x_1 - e_1$ and that $\hat{x}_2 = x_2 - e_2$, from Section 4.2.2, (4.35) is

$$\hat{\sigma}_e = x_1 - e_1 - x_{1,d} + x_2 - e_2 - x_{2,d} = \tilde{\xi}_e - e_1 + \dot{\tilde{\xi}}_e - e_2 \tag{4.39}$$

By rearranging, we obtain that the tracking error dynamics is

$$\dot{\tilde{\xi}}_e = -\tilde{\xi}_e + \hat{\sigma}_e + e_1 + e_2. \tag{4.40}$$

Furthermore, the velocity tracking error $\tilde{\zeta}$ is represented by the sliding variable $\hat{\sigma}_e$; cf. (4.29) and (4.35). The overall closed-loop dynamics with $\tau(\cdot)$ given by (4.37) is thus given by $\dot{\hat{\sigma}}_e$ given in (4.38), $\dot{\tilde{\xi}}_e$ given in (4.40) and the HOSMO error given in (4.32). The closed-loop dynamics is thus

$$\begin{aligned} \sum_1 \begin{cases} \dot{\tilde{\xi}}_e = -\tilde{\xi}_e + \hat{\sigma}_e + e_1 + e_2 \\ \dot{\hat{\sigma}}_e = -k_1\phi_1(\hat{\sigma})_e + z \\ \dot{z} = -k_2\phi_2(\hat{\sigma})_e \end{cases} \\ \sum_2 \begin{cases} \dot{e}_1 = -k_1|e_1|^{2/3}\operatorname{sgn}(e_1) + e_2 \\ \dot{e}_2 = -k_2|e_1|^{1/3}\operatorname{sgn}(e_1) + e_3 \\ \dot{e}_3 = -k_3\operatorname{sgn}(e_1) + \dot{F}(\cdot) \end{cases} \end{aligned} \tag{4.41}$$

Theorem 4.3. *Consider the closed-loop system (2.1), (2.11), and (4.37). Assume that the HOSMO in (4.30) and (4.31) is used to estimate x_1 and x_2, where it is assumed that $|\dot{F}(\cdot)| < \Delta$, is used to estimate x_1 and x_2, and assume that the sliding surface is chosen as in (4.35). Then, the complete system is represented by the cascaded system in (4.41), and the origin of the cascaded system is UGAS, which ensures the asymptotic convergence of the tracking error.*

Proof. Analysis of subsystem 1 with $e_1 = 0$ and $e_2 = 0$: With $e_1 = 0$ and $e_2 = 0$, subsystem 1 can be written as

$$\sum_1 \begin{cases} \dot{\tilde{\xi}}_e = -\tilde{\xi}_e + \hat{\sigma}_e \\ \dot{\hat{\sigma}}_e = -k_1\phi_1(\hat{\sigma})_e + z \\ \dot{z} = -k_2\phi_2(\hat{\sigma}_e) \end{cases} \tag{4.42}$$

This subsystem is then equal to the system analysed in Section 4.2.1 with $\Gamma(\cdot) = 1$, $\varphi_1(\cdot) = 0$ and $\varphi_2(\cdot) = 0$, which satisfy the inequalities in Theorem 4.1. We can therefore conclude that subsystem 1 is UGAS.

Analysis of subsystem 2: In [57], a Lyapunov function is proposed for a third-order SMO. It is proven that the Lyapunov function is radially unbounded and

positive definite and that it is a Lyapunov function for subsystem $\sum_2$, whose trajectories converge in finite time to the origin $e = 0$ for every value of $|\dot{F}(\cdot)|$ as long as $\dot{F}(\cdot)$ is bounded. Since $\dot{F}(\cdot)$ is bounded by assumptions, the origin is GFTS for every value of $\dot{F}(\cdot)$, which means that the origin is also UGAS by Proposition A.1 and Proposition A.2, in turn implying that $||e(t)|| \leq \delta \ \forall t \geq 0$.

Analysis of the complete system: To analyse the complete system, Lemma A.7 is used. To check whether the solutions of the complete system are UGB, the boundedness of $\tilde{\xi}_e$ must be evaluated when $e_1 \neq 0$ and $e_2 \neq 0$, and for this purpose, the Lyapunov function candidate $V(\tilde{\xi}_e) = \frac{1}{2}\tilde{\xi}_e^2$ is used. Note that the boundedness of $\hat{\sigma}_e$ was proven in the proof of Theorem 4.1. The derivative of the Lyapunov function candidate V is then as follows:

$$\begin{aligned} \dot{V}(\tilde{\xi}_e) &= -||\tilde{\xi}_e||^2 + (\hat{\sigma}_e + e_1 + e_2)\tilde{\xi}_e \\ &\leq -||\tilde{\xi}_e||^2 + \theta||\tilde{\xi}_e||^2 - \theta||\tilde{\xi}_e||^2 + (\beta + 2\delta)||\tilde{\xi}_e|| \\ &\leq -(1-\theta)||\tilde{\xi}_e||^2 \quad \forall \quad ||\tilde{\xi}_e|| \geq \frac{\beta + 2\delta}{\theta} \end{aligned} \tag{4.43}$$

where $0 < \theta < 1$. The solutions are then UGB because the conditions of Theorem A.5 are satisfied. Consequently, the conditions of Lemma A.7 are satisfied, which implies that the complete system is UGAS. □

4.3 Simulation results

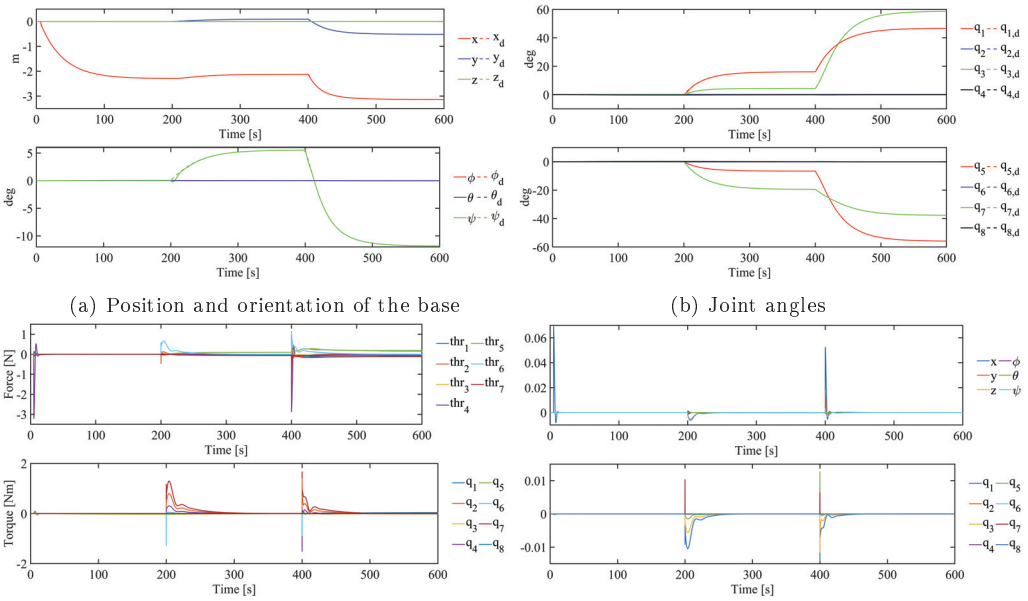
In this section, we present simulation results for trajectory tracking using the control laws presented in Section 4.2. We present both results showing the optimal performance of the algorithm, but also realistic ones that are based on using the control gains obtained in experiments during the fall of 2018. We use the simulation model presented in Section 2.2 and the Eelume 2016 version of the robot presented in Section 2.2.1.

4.3.1 Optimal performance

In this section the optimal performance case will be presented, by using the control law from Section 4.2.1. The task that is performed in the simulation is trajectory tracking for the base of the AIAUV. A suitable path for the base to follow is generated by giving set-points to an inverse kinematic controller. The set-points given are for the end-effector of the AIAUV, and the inverse kinematic then generates a reference trajectory for the base and joints, such that the end-effector reaches it target. Three different set-points are given to the inverse kinematic, and they change at 5, 200, 400 seconds. For the simulations, a fixed-step solver with a step size of 10^{-4} was used.

Simulations

The gains for the GSTA were chosen as $k_1 = [5e_{14}]^T$, $k_2 = [0.0002e_{14}]^T$ and $\beta_{\text{GSTA}} = [15e_{14}]^T$ where e_i is a $1 \times i$ vector of ones. In Figure 4.1a, the simulation results for the position and orientation of the base are presented and in Figure 4.1b,



(a) Position and orientation of the base

(b) Joint angles

(c) Thruster forces and joint torques

(d) Sliding surface

Figure 4.1: Optimal simulation results using the control law from Section 4.2.1

the simulation results for the joint angles are presented. In Figure 4.1c the thruster forces and joint torques applied are presented and in Figure 4.1d the sliding surface σ is shown.

Discussion

We can see from Figure 4.1a and Figure 4.1b that the AIAUV follows the given position, orientation and joints trajectories very well. This is also supported by Figure 4.1d, as σ is below 0.07 for the position and orientation and below 0.015 for the joints in absolute value. From Figure 4.1c we can see that the forces used is smooth, i.e. no chattering and below 50 N, which is the limit for the thrusters. That means that the forces used to control the AIAUV is indeed applicable.

4.3.2 Realistic simulations

In this section simulation results based on control gains obtained in experiments will be presented, for both control laws presented in Section 4.2. The test cases used are the C-shape, the C-shape with a moving head and the I-shape. The test cases are explained in Section 2.3.3. To make the simulations a valid comparison with the experiments, we use a P-controller for the joints. The reason for this choice is that the Eelume 2016 robot has an internal joint controller, which is a P-controller. To create a continuous trajectory we use a filter to generate a continuous trajectory

between set-points. For the simulations, the ode3 fixed-step solver with a step size of 0.002 was used.

Simulations

The gains used for the simulations are those found during the experiments to ensure that the comparison between the simulations and experiments is as correct as possible. The gains for the GSTA and the HOSMO were chosen as in Table 4.1. Note that we used lower gains for the I-shape than for the two other shapes, and we also used different gains for the two control schemes. The reason for that is that when the control scheme with the HOSMO was used the control input was more aggressive, which can be seen from the results. This led to higher forces being used and therefore we had to use lower gains to not get oscillations in the experiments. Also note that the gains for the HOSMO, L, are different from those used during the experiments for the C-shape and the C-shape with a moving head, because we observed chattering in the simulations when we used $L = 0.01$, which was not experienced in the experiments.

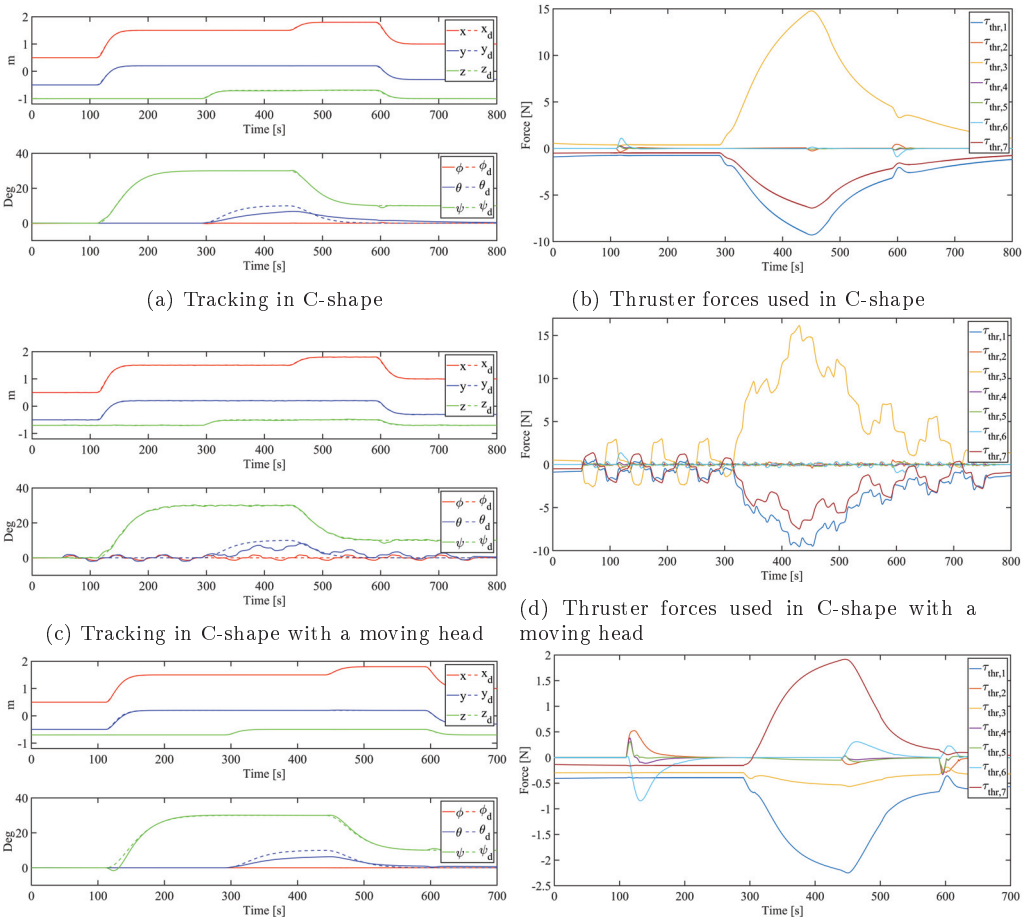
Table 4.1: Simulations: Control gains for the GSTA

Gains	**Tests**			
	C-shape	C-shape w/moving head	I-shape	I-shape
k_1	1	1	2	1
k_2	0.0006	0.0006	0.0006	0.0006
β	24	24	12	3
L	0.1	0.1	HOSMO not used	0.1

The results when using the control law proposed in Section 4.2.1 are presented in Figure 4.2. Figure 4.3 shows the simulation results when velocity measurements are not available and the control law proposed in Section 4.2.2 is used. Note that if higher gains had been used in the simulations, the results would have improved.

4.3.3 Discussion

As shown in Figure 4.2 and Figure 4.3, the AIAUV follows the given position and orientation very well for all three shapes, both when the control law from Section 4.2.1 and when the control law with the HOSMO from Section 4.2.2 is used. The AIAUV does have some issues with following the θ reference in all cases, but that might be because there is a small deviation between the desired control input and the control input that we obtain from the thruster allocation scheme, as shown in Figure 4.4. Note that when we, in the simulations, gave the thruster allocation scheme more time to catch up with the reference before it was set to zero again, we observed that the reference was eventually followed. Since we want a back-to-back comparison between simulations and experiments, we kept the same timing in the simulations as in the experiments. We can also observe a small deviation in ψ at 600 s for all cases, which is a transient because of the changes in position references that occur at 600 s. In the I-shape, there is a transient deviation in ψ



(a) Tracking in C-shape

(b) Thruster forces used in C-shape

(c) Tracking in C-shape with a moving head

(d) Thruster forces used in C-shape with a moving head

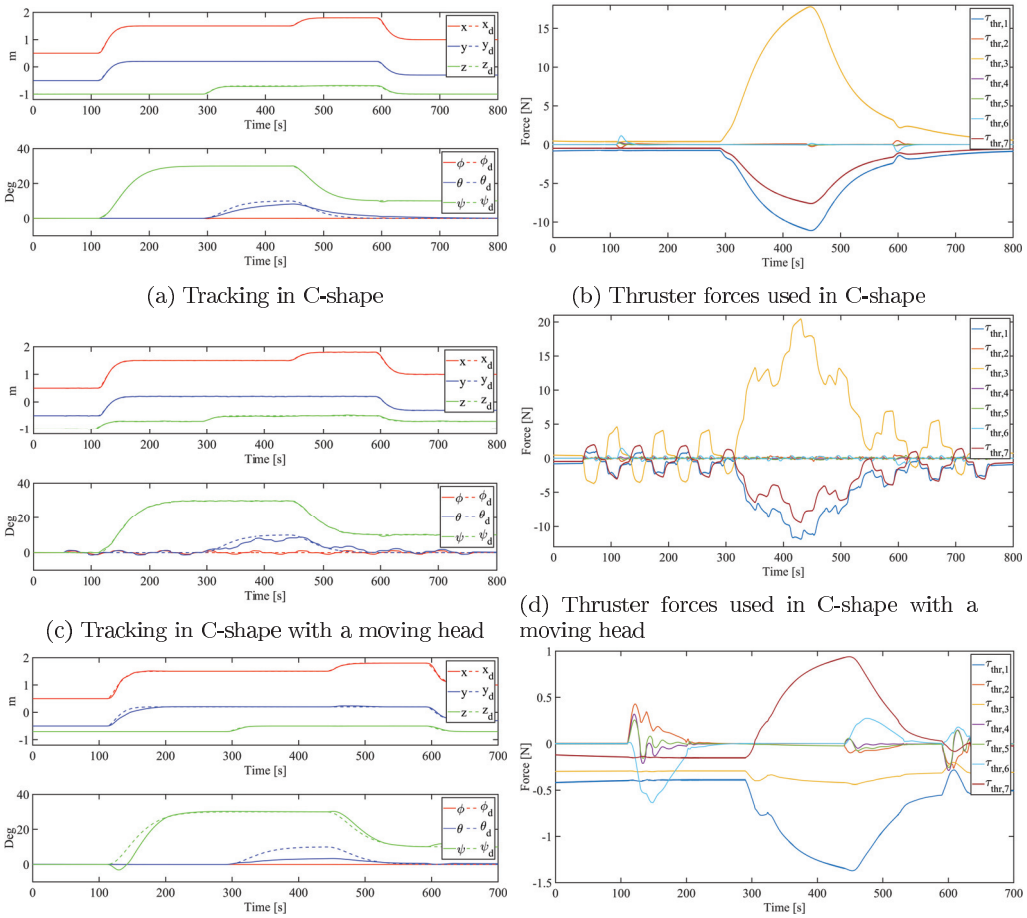
(e) Tracking in I-shape

(f) Thruster forces used in I-shape

Figure 4.2: Simulation results using the control law from Section 4.2.1

also after 100 s, and the reason for this is probably the lower gains. We also observe some oscillations in θ and ψ in the C-shape with a moving head case, as shown in Figure 4.2c and Figure 4.3c, and the reason for the oscillations is the moving head; moreover, we observe that the oscillations in the states are consistent with the movement of the head.

For the C-shape and the C-shape with a moving head, we do observe an improvement in the performance when the HOSMO is used, and the reason for this improvement can be found by comparing Figure 4.2b and Figure 4.3b, and Figure 4.2d and Figure 4.3d. As shown in these plots, when the HOSMO is used, higher thruster forces are used. However, for the I-shape, comparing Figure 4.2f and Figure 4.3f shows that less force is used, thus making the performance in the I-shape when the HOSMO is used poorer, which is logical since lower gains are



(a) Tracking in C-shape

(b) Thruster forces used in C-shape

(c) Tracking in C-shape with a moving head

(d) Thruster forces used in C-shape with a moving head

(e) Tracking in I-shape

(f) Thruster forces used in I-shape

Figure 4.3: Simulation results using the control law from Section 4.2.2

used when the HOSMO is used.

Figure 4.2 and Figure 4.3 also show that for both control schemes the thruster forces used are below 50 N, which is the limit of the thrusters on the actual Eelume robot. This result means that the forces used to control the AIAUV are indeed applicable.

4.4 Experimental investigation

In this section, the experimental results obtained during the fall of 2018 are presented. The purpose of the experiments is to validate the theory and the robustness of the control approaches, by showing that the proposed approaches also work in experiments and not only in the ideal case of the simulations. The robot used is

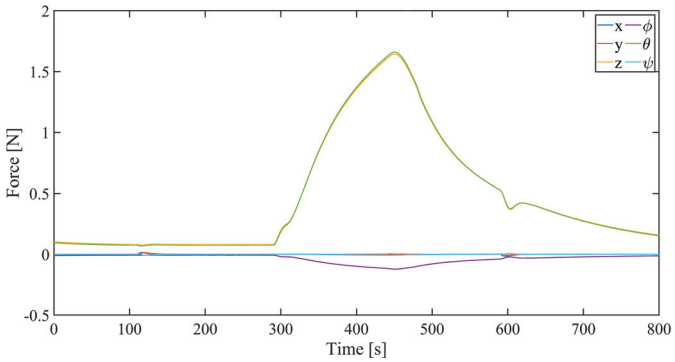


Figure 4.4: Error between desired control input and actual control input from the thruster allocation scheme

the Eelume 2016 version of the robot presented in Section 2.3.1 and the test cases used are the same as for the simulations, and are explained in Section 2.3.3.

4.4.1 Results

In this section, the experimental results from the performed tests are presented. The GSTA was easy to make work in the experiments, and it did not require much effort in tuning. The gains were increased until we observed that rather than following the reference, the AIAUV began to oscillate around the reference. We therefore chose gains that created a small deviation from the reference rather than gains that were more aggressive (higher), where the AIAUV would oscillate around the reference. These oscillations might be caused by delays in the thrusters. The gains for the GSTA and the HOSMO were chosen as shown in Table 4.2.

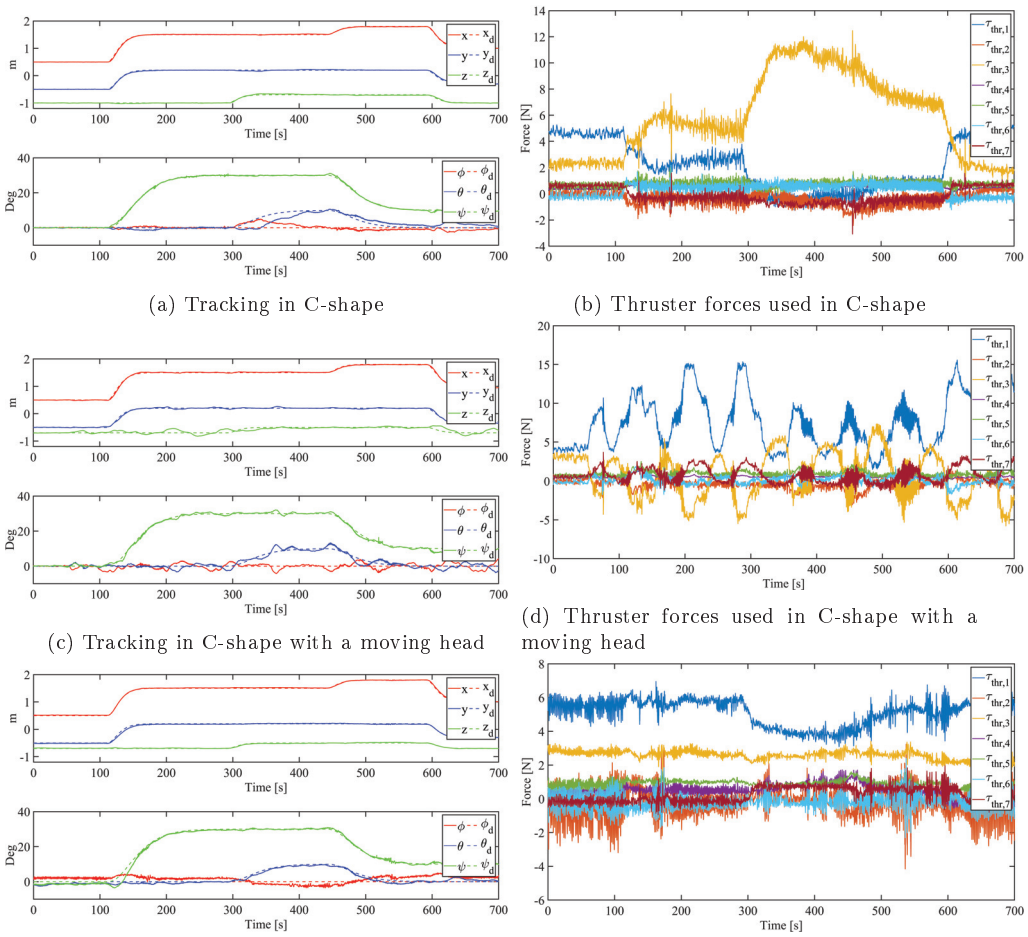
Table 4.2: Experiments: Control gains for the GSTA

Gains	**Tests**			
	C-shape	C-shape w/moving head	I-shape	I-shape
k_1	1	1	2	1
k_2	0.0006	0.0006	0.0006	0.0006
β	24	24	12	3
L	0.01	0.01	HOSMO not used	0.1

In Figure 4.5 the results when using the control law proposed in Section 4.2.1 are presented. Note that here the velocity estimates come from the Kalman filter. In Figure 4.6 we see the results when velocity measurements are not available and the control law proposed in Section 4.2.2 were used.

4.4.2 Discussion

In the C-shape, we observe from Figure 4.5a that the reference was followed nicely, but we do observe a small deviation from the θ reference caused by a transient, i.e.,



(a) Tracking in C-shape

(b) Thruster forces used in C-shape

(c) Tracking in C-shape with a moving head

(d) Thruster forces used in C-shape with a moving head

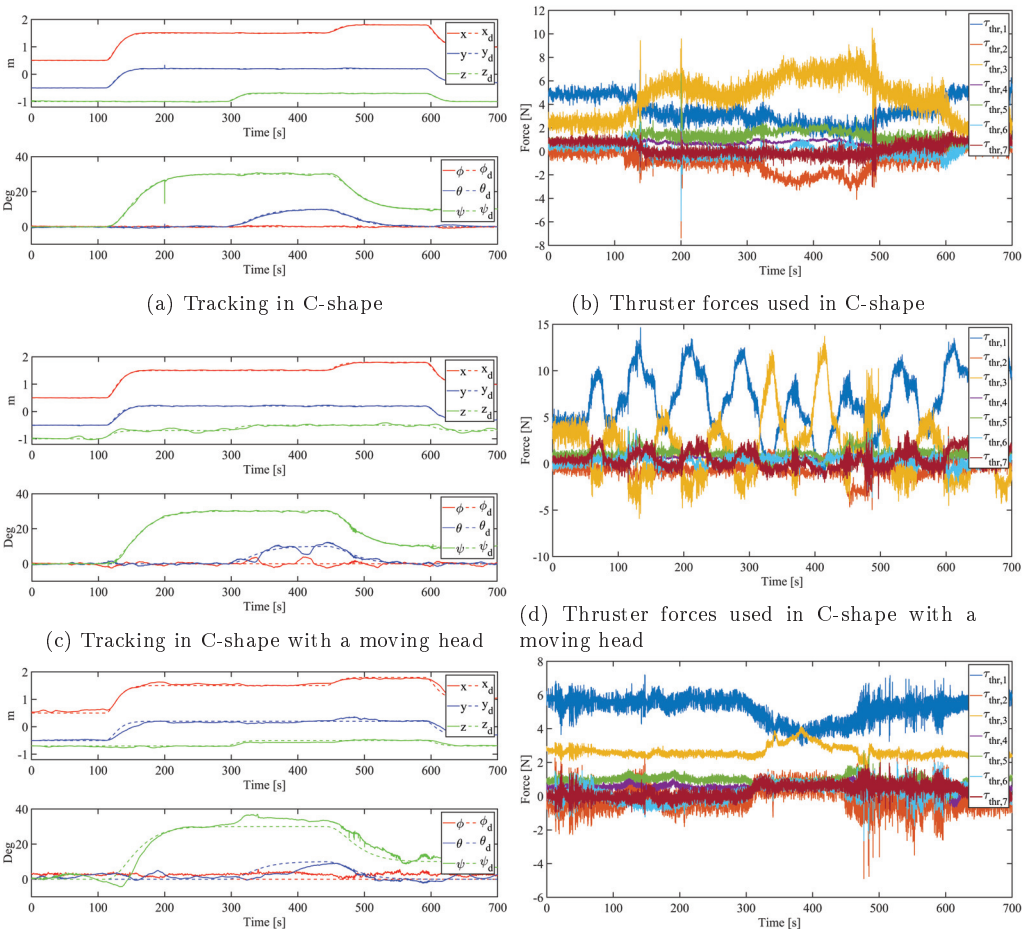
(e) Tracking in I-shape

(f) Thruster forces used in I-shape

Figure 4.5: Experimental results using the control law from Section 4.2.1

when the θ reference is changed to 10 deg at 300 s. We also observe small transient deviations in ϕ and θ at 100 s, 300 s and 600 s, which are when the position reference is changed. These are similar to the deviations observed in the simulations. In the C-shape with a moving head, Figure 4.5c shows that the movement of the head caused oscillations in z, ϕ, θ and ψ, but the position reference was still followed. In the I-shape shown in Figure 4.5e, we observe that the reference was followed nicely for all DOFs except ϕ, which is logical since ϕ is unactuated in the I-shape. From Figure 4.5b, Figure 4.5d and Figure 4.5f, we find that there is some chattering in the control input, which could potentially have been reduced by reducing the gains, but if we reduced the gains, we would not obtain the desired tracking performance.

We observed an improvement in the tracking performance when the HOSMO was used without an increase in thruster use; this can be seen by comparing Fig-



(a) Tracking in C-shape

(b) Thruster forces used in C-shape

(c) Tracking in C-shape with a moving head

(d) Thruster forces used in C-shape with a moving head

(e) Tracking in I-shape

(f) Thruster forces used in I-shape

Figure 4.6: Experimental results using the control law from Section 4.2.2

ure 4.5 and Figure 4.6. In the C-shape, we observe from Figure 4.6a that we have almost perfect tracking. In the C-shape with a moving head, we observe from Figure 4.6c some small oscillations in z, ϕ and θ caused by the movement of the head, but the movement is much smaller than for the case when the HOSMO is not used and is present almost only when the θ reference is changed to 10 deg at 300 s. In the I-shape shown in Figure 4.6e, we observe some deviation from the reference trajectory, but we do see that the trajectory is essentially followed. The I-shape is therefore the only case where we have worse performance with the control scheme with the HOSMO. The reason for this result is probably the lower gains, as we had oscillations in the AIAUV when we used the same gains as we did for the control law without the HOSMO, i.e., the control law from Section 4.2.1. Additional measurement noise from Qualisys was observed in these tests, but the noise did

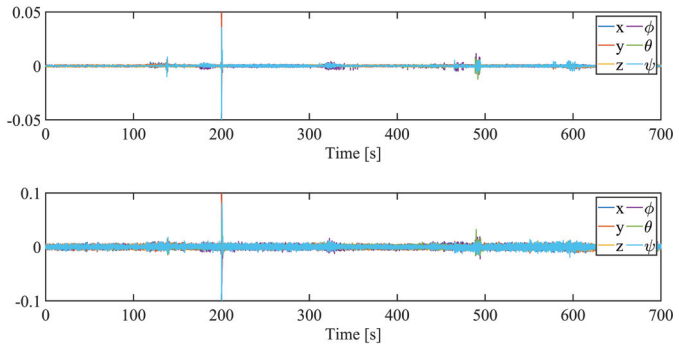


Figure 4.7: Estimation errors for the HOSMO during C-shape (the first sub-plot shows $e_1 = x_1 - \hat{x}_1$ and the second sub-plot shows $e_2 = x_2 - \hat{x}_2$)

not affect the performance. Thus, the conclusion would be that the HOSMO works well and creates better performance than the Kalman filter that we used. When we used the HOSMO, we observed an increase in chattering, most likely because some chattering exists in the estimated states from the HOSMO. From the HOSMO errors in Figure 4.7, where the first sub-plot shows $e_1 = x_1 - \hat{x}_1$ and the second sub-plot shows $e_2 = x_2 - \hat{x}_2$, we observe that the errors are quite small and that they appear to be noise, this means that the HOSMO is indeed applicable.

If we compare the results obtained in the simulations (Figure 4.2 and Figure 4.3) and the results obtained in the experiments (Figure 4.5 and Figure 4.6) for both control schemes, we find that they are quite similar. For the C-shape, the trajectories were almost perfectly tracked, whereas for the C-shape with a moving head, we observed some oscillations in some of the states, which were caused by the moving head. For the I-shape, as expected, the tracking performance was not as good as for the other shapes since the gains used for the I-shape were lower and the AIAUV is underactuated in ϕ in this shape. For the C-shape and the C-shape with a moving head, we observed an improvement in performance when the HOSMO was used, whereas for the I-shape, we found a decrease in performance. The reason for this result was probably the lower gains. For the experiments in general, we observed that the tracking performance was slightly worse than in the simulation results; however, this result is expected since in the experiments, we had outliers and errors in the position measurements. The fact that we did not have feedback from the thrusters probably also affected the errors, because we were not necessarily obtaining the forces from the thrusters that we were asking for. If we consider the difference in thruster use, we find that the main difference was more chattering in the thruster forces in the experiments. The reason for this result was probably thruster delays and measurement noise.

4.5 Chapter summary

In this chapter, we have proposed the GSTA for solving the trajectory tracking problem of AIAUVs in 6DOF. We have proven that the closed-loop system is UGAS

and that the GSTA gives GUFTS, not only GFTS. We have shown how well the algorithm can perform in an optimal performance example, and we have presented comprehensive simulation and experimental results for three different shapes, representing both transport mode and operation mode. The results validate and verify that the proposed approach is well suited for control of an AIAUV. Furthermore, we have also solved the trajectory tracking problem in 6DOF using the GSTA in combination with a HOSMO. Additionally, we have proven that the closed-loop system is UGAS. The same three shapes were used to obtain comprehensive simulation and experimental results, and these validate and verify that the proposed approach is well suited for control of an AIAUV.

The simulation and experimental results were almost equally good. Some small offsets and oscillations observed in the experiments were not present in the simulations, but that is to be expected since the experiments have outliers and measurement noise.

In the experiments, we observed an increase in performance when the higher-order sliding mode observer was used instead of the Kalman filter. Therefore, the higher-order sliding mode observer is indeed applicable for state estimation.

Chapter 5

Comparison of Two Sliding Mode Control Algorithms

In this chapter, we will compare the previously proposed tracking control laws presented in Chapter 3 and Chapter 4. Specifically, we will compare the control law using the STA with adaptive gains presented in Section 3.2.1 and the control law using the GSTA presented in Section 4.2.1. We will also compare the control law where the STA with adaptive gains is combined with a HOSMO, which is presented in Section 3.2.2 with the control law where the GSTA is combined with a HOSMO, which is presented in Section 4.2.2. We also solve the tracking problem using a PID controller to evaluate how the SMC algorithms perform compared to a standard linear controller. The main objective of this chapter is a comparison between the STA with adaptive gains, the GSTA and a PID controller.

Contributions of this chapter

The contributions of this chapter can be summarized as follows. We demonstrate the applicability of the proposed control laws from Chapter 3 and Chapter 4, and we compare these two different algorithms for solving the trajectory tracking problem for AIAUVs in 6DOF: the STA with adaptive gains and the GSTA. Furthermore, their performance is compared with that of a PID controller

Organization of this chapter

The chapter is structured as follows. In Section 5.1, we present two tracking control laws for the AIAUV based on the PID controller: one where we assume that the velocity is known, and one where we use the estimated velocity from a HOSMO. The simulation results are presented in Section 5.2 and the experimental results are presented in Section 5.3. In Section 5.4 a chapter summary is presented.

Publications

This chapter is based on [15].

5.1 PID controller

To benchmark the two SMC algorithms, we also include a PID controller in the comparison. Therefore, in this section, we propose two tracking control laws for the AIAUV based on the PID controller: one where we assume that the velocity is known, and one where we use the estimated velocity from a HOSMO.

5.1.1 Control law based on the PID controller

In this subsection, the tracking control law for the AIAUV using the PID controller is presented. The control input when the PID controller is used is defined as

$$\tau(q) = -k_p\tilde{\xi}_e - k_d\tilde{\zeta} - k_i\int\tilde{\xi}_e dt \tag{5.1}$$

where $\tilde{\xi}_e$ is defined as in (2.12) and $\tilde{\zeta} = \zeta - \zeta_d$, where ζ and ζ_d are defined as in (2.3) and (2.4), respectively. The constants k_p, k_d and k_i are controller gains.

5.1.2 Control law based on the PID controller combined with a HOSMO

In this subsection, the tracking control law for the AIAUV using the PID controller with the estimated velocity from the HOSMO is presented.

State observer

Because velocity measurements are not available, a state observer has to be designed. We want to use the third-order SMO presented in [46], as it has been proven to be FTS in [57]. By designing the third-order SMO structure as in [46], the third-order SMO can be written as

$$\begin{aligned}\dot{\hat{x}}_1 &= \hat{x}_2 + z_1\\ \dot{\hat{x}}_2 &= \hat{x}_3 + z_2 + M^{-1}(q)J_e(\eta_2)\tau(q)\\ \dot{\hat{x}}_3 &= z_3\end{aligned} \tag{5.2}$$

where

$$\begin{aligned}z_1 &= k_1|e_1|^{2/3}\operatorname{sgn}(e_1)\\ z_2 &= k_2|e_1|^{1/3}\operatorname{sgn}(e_1)\\ z_3 &= k_3\operatorname{sgn}(e_1)\end{aligned} \tag{5.3}$$

and $k_1 \in \mathbb{R}^{6+(n-1)}$, $k_2 \in \mathbb{R}^{6+(n-1)}$ and $k_3 \in \mathbb{R}^{6+(n-1)}$ are gains to be chosen according to [48] and [49], where $e_1 = \xi_e - \hat{x}_1 \in \mathbb{R}^{6+(n-1)}$ with ξ_e defined as in (2.10). One choice of parameters that satisfies the requirements in [48] and [49] is, according to [19], $k_1 = 6L^{1/3}$, $k_2 = 11L^{1/2}$ and $k_3 = 6L$, where $L \in \mathbb{R}^{6+(n-1)}$ is a sufficiently large constant. Note that the mathematical operations in (5.2) and (5.3) are performed in an element-wise manner.

Control input

The control input when the PID controller with the velocity from the third-order SMO from (5.2) and (5.3) are used can be found as

$$\tau(q) = -k_p\tilde{\xi}_e - k_d\hat{\dot{\tilde{\zeta}}} - k_i \int \tilde{\xi}_e dt \tag{5.4}$$

where $\tilde{\xi}_e$ is defined as in (2.12) and $\hat{\dot{\tilde{\zeta}}} = J_e(\eta_2)^{-1}\hat{x}_2 - \zeta_d$, where $\hat{x}_2$ and ζ_d are defined as in (5.2) and (2.4), respectively. The constants k_p, k_d and k_i are controller gains.

5.2 Simulation results

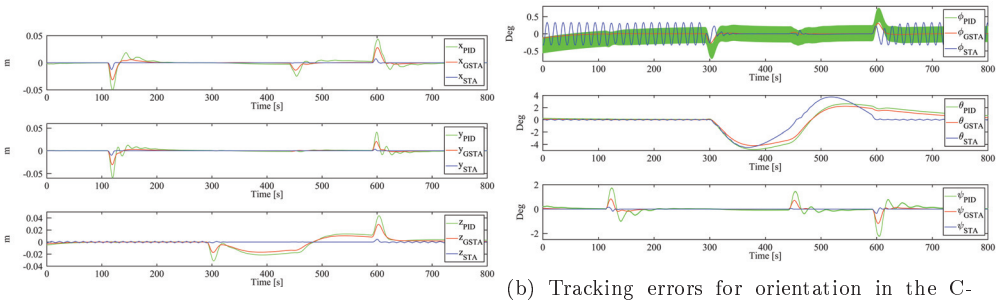
In this section, we present simulation results for trajectory tracking using the control laws presented in Section 3.2, Section 4.2 and Section 5.1. We use the simulation model presented in Section 2.2 and the Eelume 2016 version of the robot presented in Section 2.2.1.

The test cases used are the C-shape and the C-shape with a moving head. The test cases are explained in Section 2.3.3. To make the simulations a valid comparison with the experiments, we use a P-controller for the joints. The reason for this choice is that the Eelume 2016 robot has an internal joint controller, which is a P-controller. To create a continuous trajectory we use a filter to generate a continuous trajectory between set-points. For the simulations, the ode3 fixed-step solver with a step size of 0.002 was used. To ensure that the comparison between the simulations and experiments is as fair as possible, the gains used during the simulations are the gains found during the experiments. The gains for the STA with adaptive gains and the HOSMO are presented in Table 3.1; for the GSTA and the HOSMO, the gains are presented in Table 4.1; and for the PID controller, the gains are presented in Table 5.1. For the SMC algorithms, the gains for the HOSMO, L, are different from those used during the experiments for the C-shape and the C-shape with a moving head because we observed chattering in the simulations when we used $L = 0.01$, which was not experienced in the experiments.

Table 5.1: Simulation: Control gains for the PID controller

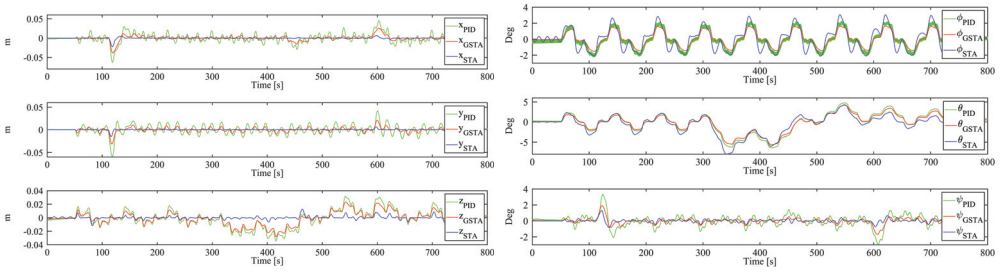
Gains	**Tests**	
	C-shape	C-shape w/moving head
k_p	20	20
k_d	0.4	0.4
k_i	0.1	0.1
L	0.01	0.01

The trajectory tracking results using the control laws from Section 3.2.1, Section 4.2.1 and Section 5.1.1, i.e., without the HOSMO, are presented in Figure 5.1. Note that the orientation errors are presented in terms of Euler angles for visualization, even though the control laws use quaternions. This is also the case for the rest of the results presented in this chapter. The corresponding thruster forces



(a) Tracking errors for position in the C-shape

(b) Tracking errors for orientation in the C-shape

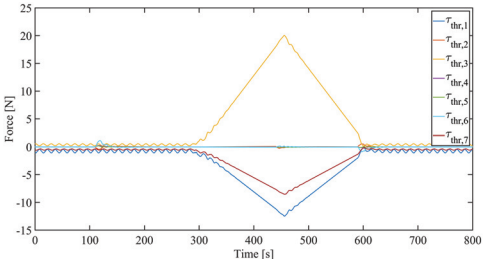


(c) Tracking errors for position in the C-shape with a moving head

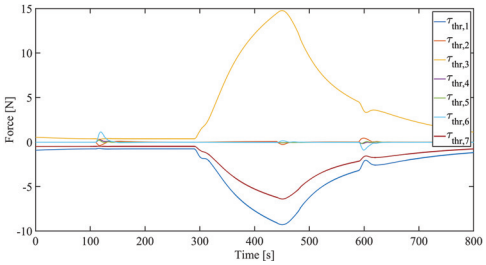
(d) Tracking errors for orientation in the C-shape with a moving head

Figure 5.1: Simulation: Trajectory tracking results using the control laws without the HOSMO

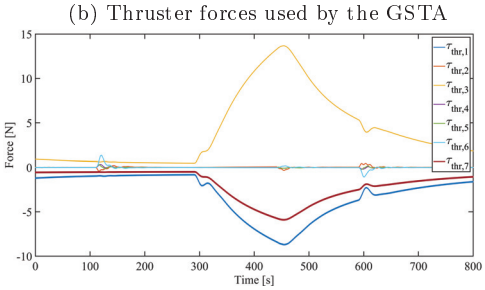
are shown for the C-shape in Figure 5.2 and for the C-shape with a moving head in Figure 5.3. In Table 5.2, the root mean square error (RMSE) and maximum error for the position and orientation are given for each control law without the HOSMO and for both simulation test cases, i.e., the C-shape and the C-shape with a moving head. In Table 5.3, the root mean square (RMS) for the thruster forces and the derivative of the thruster forces are given for each control law without the HOSMO and for both simulation test cases. The RMS of the thruster forces provides an idea of how much force is used, and the RMS of the derivative of the thruster forces provides a measure of how much chattering is present in the thruster forces. The trajectory tracking results using the control laws from Section 3.2.2, Section 4.2.2 and Section 5.1.2, i.e., with the HOSMO, are presented in Figure 5.4. The corresponding thruster forces are shown for the C-shape in Figure 5.5 and for the C-shape with a moving head in Figure 5.6. In Table 5.4, the RMSE and maximum error for the position and orientation are given for each control law with the HOSMO and for both simulation test cases. In Table 5.5, the RMS for the thruster forces and the derivative of the thruster forces are given for each control law with the HOSMO and for both simulation test cases. Note that if higher gains had been used in the simulations, the results would have improved, at least for the GSTA.



(a) Thruster forces used by the STA with adaptive gains

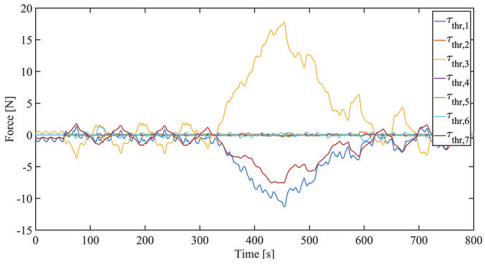


(b) Thruster forces used by the GSTA

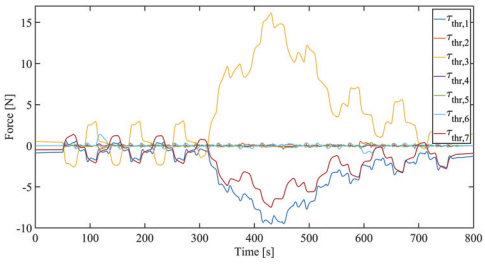


(c) Thruster forces used by the PID controller

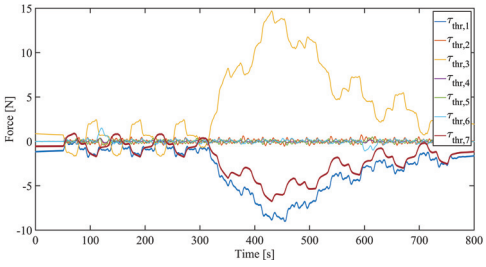
Figure 5.2: Simulation: Thruster forces used in the C-shape when the control laws without the HOSMO are used



(a) Thruster forces used by the STA with adaptive gains



(b) Thruster forces used by the GSTA



(c) Thruster forces used by the PID controller

Figure 5.3: Simulation: Thruster forces used in the C-shape with a moving head when the control laws without the HOSMO are used

5.2.1 Discussion

As shown in Figure 5.1, the AIAUV follows the given position and orientation very well for both test cases as all errors are quite small when the control laws from Section 3.2.1, Section 4.2.1 and Section 5.1.1, i.e., without the HOSMO, are used. This result indicates that all the control laws are applicable. As shown in Figures 5.1a and 5.1b, in the C-shape case, the STA with adaptive gains provides the best result overall, i.e., the lowest error, and the GSTA provides the second best. The reason why the STA with adaptive gains provides better results than the GSTA is probably that we have to tune the GSTA manually, while the STA with adaptive gains adaptively finds the best gains. However, if we more closely examine some of the results for ϕ and θ, we do find that for ϕ, the GSTA provides the lowest

Table 5.2: Simulation: Comparison of results when the control laws without the HOSMO are used

Algorithm		**Root Mean Square Error (RMSE)**		**Maximum error**	
		C-shape	C-shape w/moving head	C-shape	C-shape w/moving head
STA	Position	0.0008	0.0019	0.0090	0.0167
	Orientation	0.0116	0.0228	0.0301	0.0720
GSTA	Position	0.0054	0.0068	0.0303	0.0327
	Orientation	0.0114	0.0193	0.0330	0.0535
PID	Position	0.0087	0.0115	0.0509	0.0521
	Orientation	0.0154	0.0249	0.0455	0.0699

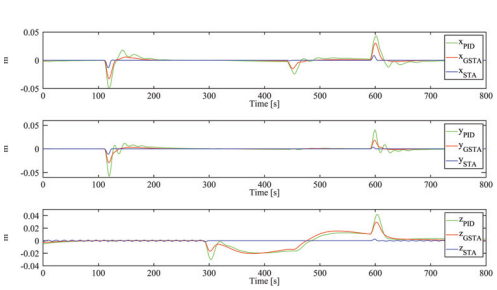
Table 5.3: Simulation: Comparison of the thruster forces used when the control laws without the HOSMO are used

Algorithm	**Root Mean Square (RMS)**			
	τ_{thr}		$\dot{\tau}_{thr}$	
	C-shape	C-shape w/moving head	C-shape	C-shape w/moving head
STA	2.16	1.89	5.32	2.19
GSTA	1.84	1.91	5.05	0.31
PID	1.80	1.87	0.74	0.67

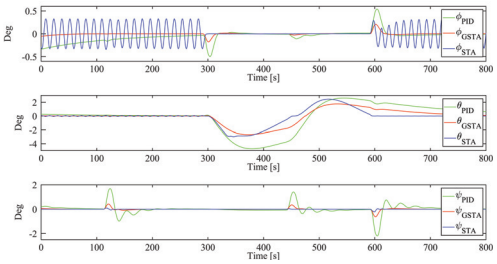
Table 5.4: Simulation: Comparison of results when the control laws with the HOSMO are used

Algorithm		**Root Mean Square Error (RMSE)**		**Maximum error**	
		C-shape	C-shape w/moving head	C-shape	C-shape w/moving head
STA	Position	0.0009	0.0032	0.0091	0.0271
	Orientation	0.0078	0.0198	0.0206	0.0676
GSTA	Position	0.0059	0.0075	0.0304	0.0341
	Orientation	0.0072	0.0125	0.0205	0.0348
PID	Position	0.0083	0.0112	0.0497	0.0524
	Orientation	0.0147	0.0245	0.0434	0.0692

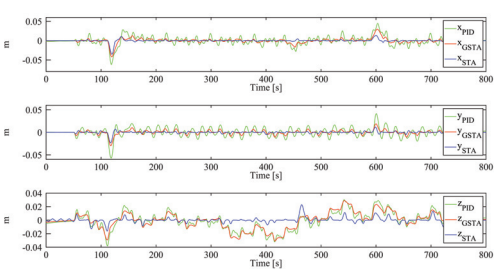
error, and for θ, GSTA and PID provide a slightly lower error than the STA with adaptive gains. From Figures. 5.1c and 5.1d, we observe the same tendencies for the C-shape with a moving head as for the C-shape case, i.e., in x, y, z and ψ, the STA with adaptive gains provides the smallest error, and the GSTA is the second best. However, for ϕ and θ, the errors are much more similar. Examining Table 5.2 confirms what we can observe from Figure 5.1, i.e., that the STA with adaptive gains provides the smallest error for the position, and that the GSTA is the second best, whereas for orientation, the GSTA and the STA with adaptive gains have much more similar results, and the GSTA actually provides a slightly smaller error. The PID controller clearly offers the worst tracking performance for both position and orientation. If we take the thruster use of the different algorithms into consideration, i.e., Figures 5.2 and 5.3, we find that for both test cases, the STA with adaptive gains uses more force than the GSTA and the PID controller, while the GSTA and the PID use very similar amounts of force. However, the RMS of τ_{thr} in Table 5.3 shows that over time, there is actually not a large difference in how much force is used. The STA with adaptive gains uses slightly more force in



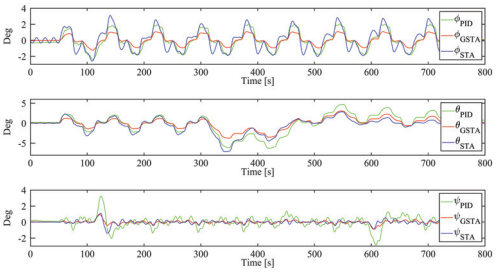
(a) Tracking errors for position in the C-shape



(b) Tracking errors for orientation in the C-shape



(c) Tracking errors for position in the C-shape with a moving head



(d) Tracking errors for orientation in the C-shape with a moving head

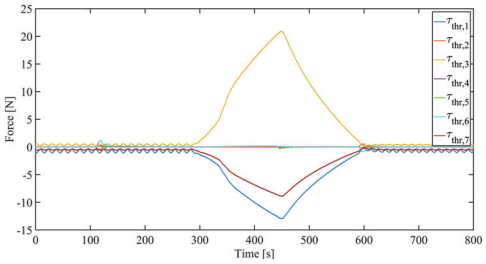
Figure 5.4: Simulation: Trajectory tracking results using the control laws with the HOSMO

Table 5.5: Simulation: Comparison of the thruster forces used when the control laws with the HOSMO are used

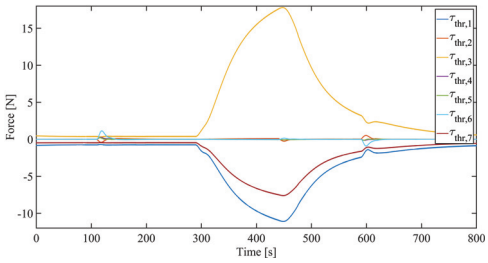
Algorithm	**Root Mean Square (RMS)**			
	τ_{thr}		$\dot{\tau}_{thr}$	
	C-shape	C-shape w/moving head	C-shape	C-shape w/moving head
STA	2.25	2.05	15.78	9.48
GSTA	2.11	2.22	10.99	7.90
PID	1.82	1.89	1.43	1.37

the C-shape case, but for the case of the C-shape with a moving head, it is very similar. However, from the RMS of $\dot{\tau}_{thr}$, we find that there is the most chattering in the control input from the STA with adaptive gains, there is less in that from the GSTA, and the control input from the PID controller shows the least rapid changes in the thrust forces. However, the value of the RMS of $\dot{\tau}_{thr}$ is so small that the chattering that is introduced by the STA with adaptive gains is negligible.

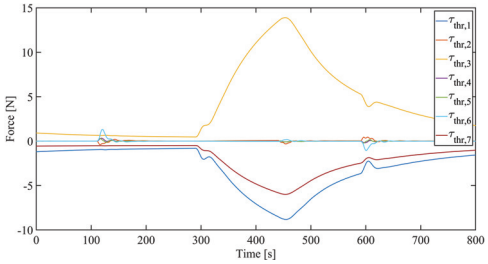
When the control laws with the HOSMO are used, i.e., the control laws from Section 3.2.2, Section 4.2.2 and Section 5.1.2, we observe the same tendencies in Figure 5.4 as when the control laws without the HOSMO are used. As shown in Figures 5.4a and 5.4b, the STA with adaptive gains provides the smallest errors for x, y, z and ψ, and the GSTA gives the second smallest errors. For ϕ, however, the GSTA provides the smallest error, while the STA with adaptive gains results



(a) Thruster forces used by the STA with adaptive gains

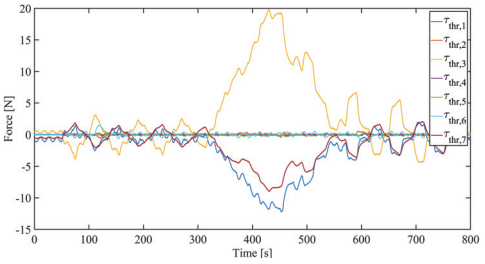


(b) Thruster forces used by the GSTA

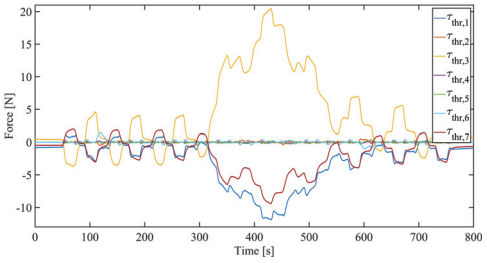


(c) Thruster forces used by the PID controller

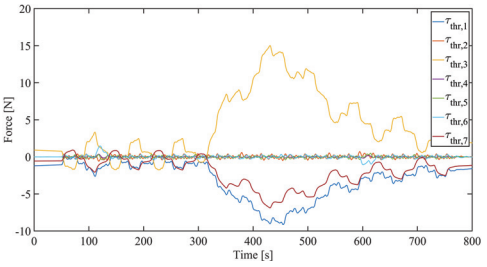
Figure 5.5: Simulation: Thruster forces used in the C-shape when the control laws with the HOSMO are used



(a) Thruster forces used by the STA with adaptive gains



(b) Thruster forces used by the GSTA



(c) Thruster forces used by the PID controller

Figure 5.6: Simulation: Thruster forces used in the C-shape with a moving head when the control laws with the HOSMO are used

in oscillations. For θ, the results are more similar between the STA with adaptive gains and the GSTA, while the PID controller gives rise to a larger error. As shown in Figures 5.4c and 5.4d, the results for the C-shape with a moving head are the same as those for the C-shape case, except that the results for ϕ and θ are now more similar. Examining Table 5.4 confirms what we can observe from Figure 5.4, i.e., that the STA with adaptive gains provides the best results for position, while the GSTA gives slightly better results for orientation. The PID controller exhibits the worst tracking performance. If we take the thruster use of the different algorithms into consideration, i.e., Figures 5.5 and 5.6, we find that the STA with adaptive gains also uses more force for both test cases, whereas the GSTA and the PID controller use quite similar amounts of force. Over time,

however, as seen in Table 5.5 from the RMS of τ_{thr}, there is actually not a large difference in how much force is used. From the RMS of $\dot{\tau}_{thr}$, we find that there is the most chattering in the control input from the STA with adaptive gains, there is less in that from the GSTA, and the PID controller shows the least rapid changes in the thrust forces. However, the value of the RMS of $\dot{\tau}_{thr}$ is so small that the chattering that is introduced by the STA with adaptive gains is negligible.

If we compare the results obtained with the control laws from Section 3.2.1, Section 4.2.1 and Section 5.1.1, i.e., without the HOSMO, and the results obtained with the control laws from Section 3.2.2, Section 4.2.2 and Section 5.1.2, i.e., when the HOSMO is used, by comparing Table 5.2 with Table 5.4, Table 5.3 and Table 5.5, we find that when the HOSMO is used, we obtain better tracking results for the orientation for all the algorithms, as well as better tracking for position when the PID controller is used. Additionally, note that the difference between the position errors, i.e., with and without the HOSMO, is smaller than the difference between the orientation errors, i.e., the control laws with the HOSMO provide the best performance overall. The reason for the better tracking performance is probably due to the increase in thruster use. The thruster use is, however, well within the boundaries of what the Eelume AIAUV can provide, which is 50 N.

To conclude, we find that we obtain the best tracking performance in position by using the STA with adaptive gains, and the GSTA provides the best tracking performance in orientation. If we simultaneously consider the errors for position and orientation, we find that the STA with adaptive gains provides the best tracking performance, but there is not a large difference between the STA with adaptive gains and the GSTA. The reason for this difference is probably that we have to tune the GSTA manually, whereas the method with adaptive gains adaptively finds the best gains and thus has an advantage. If we use the HOSMO to estimate the linear and angular velocities in the control laws, we improve our results.

5.3 Experimental investigation

In this section, experimental results obtained during the fall of 2018 are presented. In Section 3.4 experimental results for the STA with adaptive gains was presented, and in Section 4.4 experimental results for the GSTA was presented. In this section we will present these results together with the results obtained for the PID controller to perform a comparison between the STA with adaptive gains, the GSTA and the PID controller. The robot used is the Eelume 2016 version of the robot presented in Section 2.3.1 and the test cases used are the same as for the simulations, and are explained in Section 2.3.3.

5.3.1 Results

In this section, the experimental results from the performed tests are presented. All three algorithms, i.e., the STA with adaptive gains, the GSTA and the PID controller, were easy to apply successfully in the experiments, and none of them required much effort in tuning. In the experiments, we increased the gains until we observed that the AIAUV started to oscillate around the reference instead of

following it. The gains that we chose therefore create a small deviation from the reference rather than oscillations around the reference. If we would have chosen more aggressive (higher) gains, the AIAUV would have oscillated. These oscillations might be caused by delays in the thrusters. For all the algorithms, we attempted to use the same gains for both test cases, i.e., we attempted to find the best possible gains for the C-shape, and then we used the same for the C-shape with a moving head for a fair comparison. For the STA with adaptive gains, the choice of gains is not very important because the gains will adapt to their best possible values by themselves. We did, however, observe that if we chose the initial adaptive gains to be too high, the Eelume AIAUV started to oscillate. We therefore had to use lower gains when the HOSMO was used in the case of the C-shape with a moving head. The reason for this is probably because when the HOSMO was used, the thruster forces increased. The gains for the STA with adaptive gains and the HOSMO are presented in Table 3.2; for the GSTA and the HOSMO, the gains are presented in Table 4.2; and for the PID controller, the gains are listed in Table 5.6.

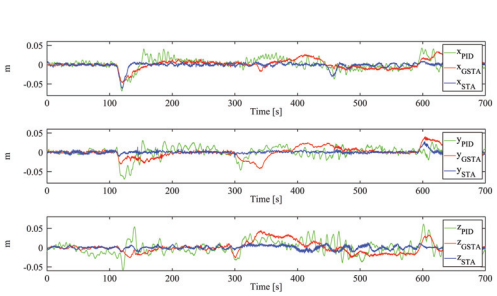
Table 5.6: Experiments: Control gains for the PID controller

Gains	**Tests**	
	C-shape	C-shape w/moving head
k_p	20	20
k_d	0.4	0.4
k_i	0.1	0.1
L	0.01	0.01

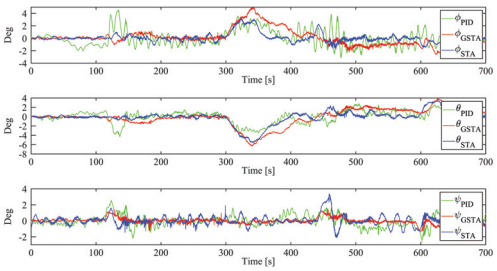
The trajectory tracking results using the control laws from Section 3.2.1, Section 4.2.1 and Section 5.1.1, i.e., without the HOSMO, are presented in Figure 5.7. The corresponding thruster forces are shown for the C-shape in Figure 5.8 and for the C-shape with a moving head in Figure 5.9. In Table 5.7, the RMSE and maximum error for the position and orientation are given for each control law without the HOSMO and for both test cases. In Table 5.8, the RMS for the thruster forces and the derivative of the thruster forces are given for each control law without the HOSMO and for both test cases. The trajectory tracking results using the control laws from Section 3.2.2, Section 4.2.2 and Section 5.1.2, i.e., with the HOSMO, are presented in Figure 5.10. The corresponding thruster forces are shown for the C-shape in Figure 5.11 and for the C-shape with a moving head in Figure 5.12. In Table 5.9, the RMSE and maximum error for the position and orientation are given for each control law with the HOSMO and for both test cases. In Table 5.10, the RMS for the thruster forces and the derivative of the thruster forces are given for each control law with the HOSMO and for both test cases. Note that the simulation study indicates that if higher gains could have been used without causing oscillations, then the tracking accuracy would have been improved.

5.3.2 Discussion

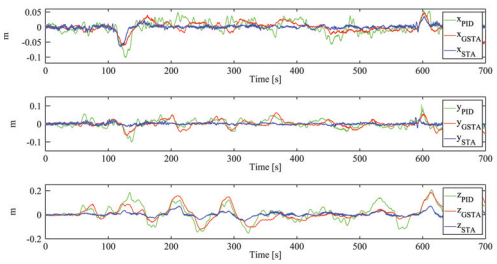
As shown in Figure 5.7, the AIAUV follows the given position and orientation very well for both test cases, as all errors are quite small when the control laws from



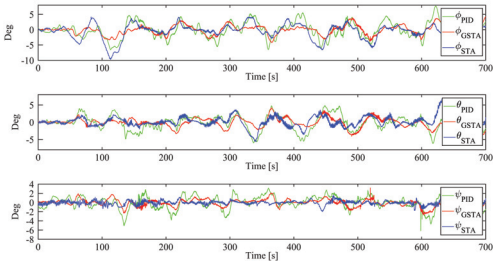
(a) Tracking errors for position in the C-shape



(b) Tracking errors for orientation in the C-shape



(c) Tracking errors for position in the C-shape with a moving head



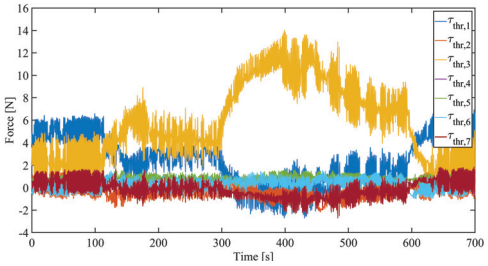
(d) Tracking errors for orientation in the C-shape with a moving head

Figure 5.7: Experiments: Trajectory tracking results using the control laws without the HOSMO

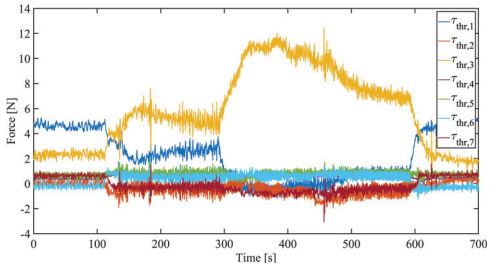
Table 5.7: Experiments: Comparison of results when the control laws without the HOSMO are used

Algorithm		**Root Mean Square Error (RMSE)**		**Maximum error**	
		C-shape	C-shape w/moving head	C-shape	C-shape w/moving head
STA	Position	0.0054	0.0121	0.0345	0.0579
	Orientation	0.0163	0.0255	0.0707	0.1074
GSTA	Position	0.0133	0.0327	0.0446	0.1039
	Orientation	0.0210	0.0216	0.0740	0.0674
PID	Position	0.0158	0.0392	0.0663	0.1382
	Orientation	0.0194	0.0362	0.0669	0.1167

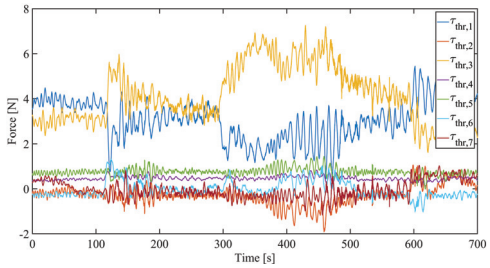
Section 3.2.1, Section 4.2.1 and Section 5.1.1, i.e., without the HOSMO, are used. As shown in Figure 5.7a, in the C-shape case, the STA with adaptive gains provides the best results for the position. The GSTA and PID controller have much more similar results for the position as the GSTA has almost as large position errors as the PID controller; however, we do observe that the position errors for the PID controller oscillate more than the position errors for the GSTA. As shown in Figure 5.7b, for the orientation, the results are more similar for all algorithms. From Figures 5.7c and 5.7d, the same tendencies are observed for the C-shape with a moving head as for the C-shape case, i.e., for the position, the STA with adaptive gains gives the smallest error, whereas for the orientation, the tracking performance is more similar. Examining Table 5.7 confirms what we find from



(a) Thruster forces used by the STA with adaptive gains

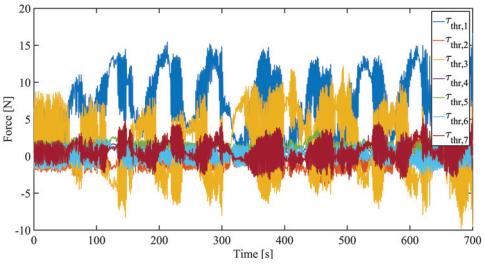


(b) Thruster forces used by the GSTA

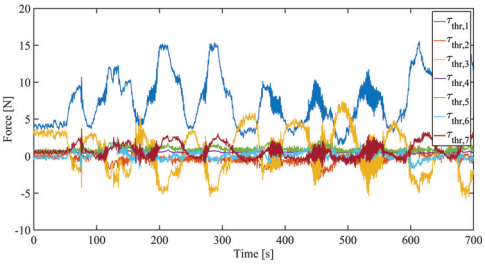


(c) Thruster forces used by the PID controller

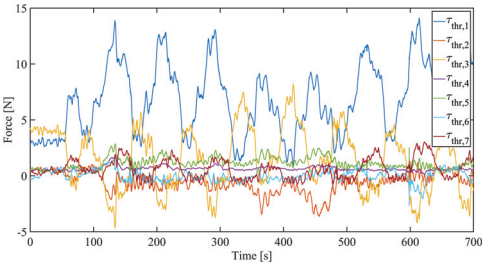
Figure 5.8: Experiments: Thruster forces used in the C-shape when the control laws without the HOSMO are used



(a) Thruster forces used by the STA with adaptive gains



(b) Thruster forces used by the GSTA



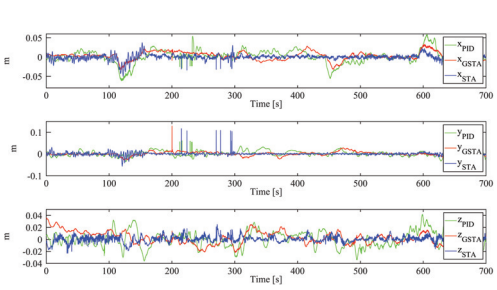
(c) Thruster forces used by the PID controller

Figure 5.9: Experiments: Thruster forces used in the C-shape with a moving head when the control laws without the HOSMO are used

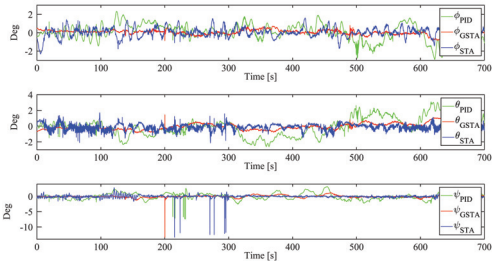
Figure 5.7, i.e., that the STA with adaptive gains gives the smallest error for the position, whereas for the orientation, which algorithm performs the best depends on the case, i.e., C-shape or C-shape with a moving head. The reason why the STA with adaptive gains outperforms the two other algorithms is probably because it has adaptive gains and is therefore robust against tuning. Table 5.7 also shows that the GSTA does outperform the PID controller by a small margin. If we take the thruster use of the different algorithms into consideration, by looking at Figures 5.8 and 5.9, we find that for both test cases, the STA with adaptive gains uses more force than the GSTA and the PID controller and that the GSTA uses more force than the PID controller. However, based on the RMS of τ_{thr}, Table 5.8 shows that over time, there is actually not a large difference in how much force is used. In the

Table 5.8: Experiments: Comparison of the thruster forces used when the control laws without the HOSMO are used

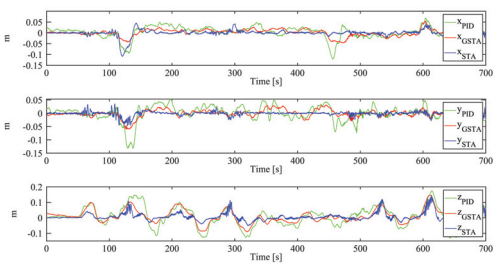
Algorithm	**Root Mean Square (RMS)**			
	τ_{thr}		$\dot{\tau}_{thr}$	
	C-shape	C-shape w/moving head	C-shape	C-shape w/moving head
STA	1.94	2.76	2.95	7.24
GSTA	1.82	2.18	1.46	1.95
PID	1.44	2.14	0.75	0.93



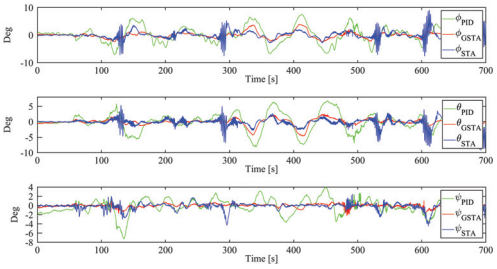
(a) Tracking errors for position in the C-shape



(b) Tracking errors for orientation in the C-shape



(c) Tracking errors for position in the C-shape with a moving head

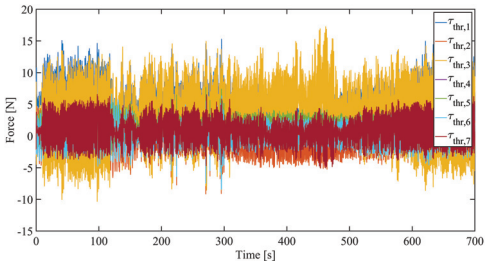


(d) Tracking errors for orientation in the C-shape with a moving head

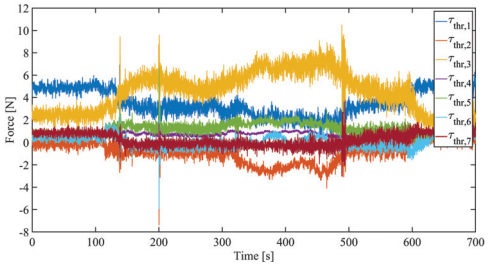
Figure 5.10: Experiments: Trajectory tracking results using the control laws with the HOSMO

C-shape case, the STA with adaptive gains uses slightly more force than the GSTA, and the GSTA uses more force than the PID controller. In the case of the C-shape with a moving head, however, the PID controller and the GSTA use very similar amounts of force, while the STA with adaptive gains uses more. Nevertheless, from the RMS of $\dot{\tau}_{thr}$, we can however that there is the most chattering in the control input from the STA with adaptive gains, there is less in that from the GSTA, and the PID controller shows the least rapid changes in the thrust forces. However, the value of the RMS of $\dot{\tau}_{thr}$ is so small that the chattering that is introduced by the STA with adaptive gains is almost negligible. The reason why the chattering may appear to be considerable in the plots is because of the time scale used.

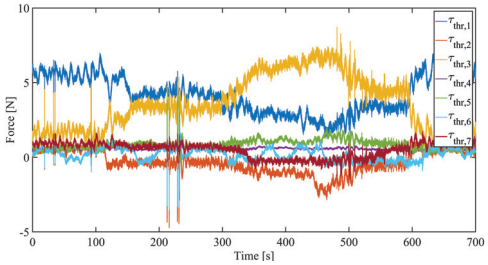
When the control laws with the HOSMO are used, i.e., from Section 3.2.2, Section 4.2.2 and Section 5.1.2, we observe the same tendencies in Figure 5.10 as



(a) Thruster forces used by the STA with adaptive gains

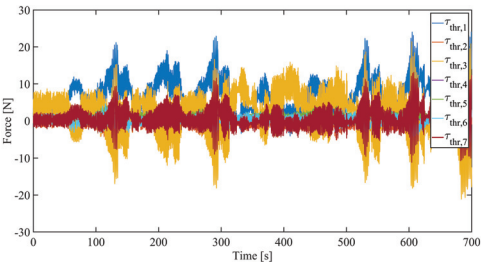


(b) Thruster forces used by the GSTA

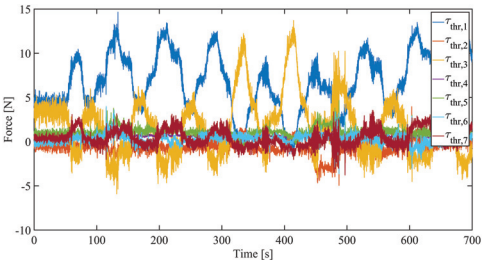


(c) Thruster forces used by the PID controller

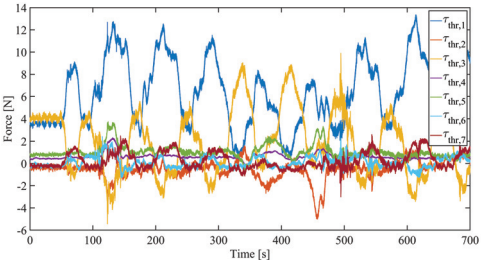
Figure 5.11: Experiments: Thruster forces used in the C-shape when the control laws with the HOSMO are used



(a) Thruster forces used by the STA with adaptive gains



(b) Thruster forces used by the GSTA



(c) Thruster forces used by the PID controller

Figure 5.12: Experiments: Thruster forces used in the C-shape with a moving head when the control laws with the HOSMO are used

when the control laws without the HOSMO are used, but we find from Figures 5.10a and 5.10b and from Figures 5.10c and 5.10d that the STA with adaptive gains is no longer clearly better. We also observe that we have some more measurement noise in the measurements from the measurement system, but the algorithms are robust against this noise. As shown in Table 5.9, we have the same results as those when the HOSMO is not used, i.e., the STA with adaptive gains provides the smallest error for position, whereas for the orientation, the GSTA gives the smallest error. The PID controller provides the worst tracking performance. From the maximum error in Table 5.9, the measurement noise impacts the maximum error results since if we would have filtered the position measurements such that the peaks in the measurements were not there, we would have obtained a lower maximum error for

Table 5.9: Experiments: Comparison of results when the control laws with the HOSMO are used

Algorithm		**Root Mean Square Error (RMSE)**		**Maximum error**	
		C-shape	C-shape w/moving head	C-shape	C-shape w/moving head
STA	Position	0.0064	0.0177	0.0640	0.1007
	Orientation	0.0087	0.0215	0.1035	0.1321
GSTA	Position	0.0100	0.0252	0.0661	0.0871
	Orientation	0.0062	0.0147	0.0914	0.0618
PID	Position	0.0143	0.0394	0.0549	0.1428
	Orientation	0.0181	0.0431	0.0779	0.1327

Table 5.10: Experiments: Comparison of the thruster forces used when the control laws with the HOSMO are used

Algorithm	**Root Mean Square (RMS)**			
	τ_{thr}		$\dot{\tau}_{thr}$	
	C-shape	C-shape w/moving head	C-shape	C-shape w/moving head
STA	3.02	3.34	17.49	13.51
GSTA	1.85	2.23	4.77	5.05
PID	1.71	2.09	4.27	4.64

all algorithms, particularly the STA with adaptive gains and the GSTA, as there are some large peaks in those measurements. If we take the thruster use of the different algorithms into consideration, i.e., Figures 5.11 and 5.12, we find that the STA with adaptive gains uses more force for both test cases and that the GSTA uses more force than the PID controller in the C-shape case, whereas in the case of the C-shape with a moving head, the PID controller and the GSTA use almost the same amount of force. However, as shown in Table 5.10, the RMS of τ_{thr} indicates that the STA does use more force than the GSTA and the PID controller in both test cases, whereas the PID controller and the GSTA use almost the same amount of force. From the RMS of $\dot{\tau}_{thr}$, we can also observe that there is the most chattering in the control input from the STA with adaptive gains, there is less in that from the GSTA, and the PID controller shows the least rapid changes in the thruster forces. However, the value of the RMS of $\dot{\tau}_{thr}$ is so small that the chattering that is introduced by the STA with adaptive gains is negligible. The reason why there may appear to be considerable chattering in the plots is again because of the time scale used.

If we compare the results obtained with the control laws from Section 3.2.1, Section 4.2.1 and Section 5.1.1, i.e., without the HOSMO, and the results obtained with the control laws from Section 3.2.2, Section 4.2.2 and Section 5.1.2, i.e., when the HOSMO is used, by comparing Table 5.7 with Table 5.9, Table 5.8 and Table 5.10, we find that when the HOSMO is used, we obtain better tracking results for the orientation for all the algorithms, as well as better tracking for position when the GSTA is used. Additionally, note that the difference between the position errors, i.e., with and without the HOSMO, is smaller than the difference between the orientation errors, i.e., the control laws with the HOSMO provide the overall best performance. The reason for the better tracking performance is prob-

ably due to the increase in thruster use. The thruster use is, however, well within the boundaries of what the Eelume AIAUV can provide, which is 50 N.

If we compare the simulation results (Figures. 5.1 and 5.4) and the experimental results (Figures. 5.7 and 5.10) for both test cases, we find that the results are quite similar and that all algorithms are applicable with and without the HOSMO to control the AIAUV. We also find that it is the same algorithms that provide the best tracking results in the simulations and the experiments, i.e., the STA with adaptive gains and the GSTA in a good second place. We also observed an improvement in the tracking performance for all algorithms when the HOSMO was used in both simulations and experiments. Of course, the tracking errors, i.e., the position and orientation errors, are larger in the experiments than in the simulations, but that is to be expected because of measurement noise, thruster dynamics and other unmodelled dynamics. The fact that we did not have feedback from the thrusters likely also affected the errors because we were not necessarily obtaining the forces from the thrusters that we were asking for.

5.4 Chapter summary

In this chapter, we have compared three different algorithms for solving the trajectory tracking problem for AIAUVs in 6DOF: the STA with adaptive gains, the GSTA and a PID controller. Comprehensive simulation and experimental results for two different test cases, i.e., the C-shape and the C-shape with a moving head, representing the operation mode, have been presented and used to compare the performance of the three mentioned algorithms. The results indicate that we obtain the best tracking performance in position by using the STA with adaptive gains, and the GSTA gives the best tracking performance in orientation. If the errors for position and orientation are added together, we find that the STA with adaptive gains provides the best tracking performance overall, but there is not a large difference between the STA with adaptive gains and the GSTA. The reason for this difference is probably that we have to tune the GSTA manually, whereas the STA adaptively finds the best gains. The PID controller offers the worst tracking performance. The same two test cases were used to obtain comprehensive simulation and experimental results for the three algorithms in combination with the HOSMO. The results have been used to compare the control laws and to study the performance of the HOSMO. When the HOSMO was used to estimate the linear and angular velocities in the control laws, we improved our results. This means that the HOSMO is indeed applicable for state estimation.

In both the simulations and in the experiments, it was the STA with adaptive gains that provided the best overall tracking performance, but the GSTA was not far from achieving the same tracking results. Thus, the fact that we had to tune the GSTA manually is probably the reason that we achieved lower tracking performance with this algorithm. The STA with adaptive gains is therefore better in practice than the GSTA, but the GSTA does have some theoretical advantages as it is proven to provide GFT stability for a larger class of systems. These additional theoretical properties make it possible to prove that the closed-loop system is UGAS without the HOSMO when the GSTA is used (proven in [16] or Chapter 4). This is not

possible for the STA with adaptive gains.

The simulation and experimental results were almost equally good. Of course, the tracking errors, i.e., the position and orientation errors, are larger in the experiments than in the simulations, but that is to be expected because of outliers, measurement noise and because of the accuracy of the measurement system.

Chapter 6

The Generalized Super-Twisting Algorithm with Adaptive Gains

In this chapter we propose an adaptive GSTA for a class of system whose perturbations and uncertain control coefficients are time- and state-dependent, i.e. we combine the best properties of the STA with adaptive gains [69] (used in Chapter 3) and the GSTA [18] (used in Chapter 4). The proposed approach consists in using dynamically adapted control gains that ensure global finite-time convergence. The theoretical advantage of the GSTA is that GFT stability is proven for a larger class of systems, i.e., systems for which both the perturbations and the control coefficients may depend on both state and time and the control coefficients are uncertain. The advantage with adaptive gains is that no conservative upper bound has to be considered on the perturbations and control coefficients to maintain sliding. We prove that the resulting closed-loop system is GFTS.

Contributions of this chapter

The contributions of the chapter can be summarized as follows. A novel adaptive GSTA is proposed for a class of systems whose perturbations and uncertain control coefficients are time- and state-dependent. The proposed approach consists in using dynamically adapted control gains in a GSTA, which ensures global finite-time convergence. A non-smooth strict Lyapunov function is used to obtain the conditions for the global finite-time stability. It is also shown that the adaptive GSTA makes the tracking errors of the AIAUV converge asymptotically to zero. A simulation and experimental study for the AIAUV is performed to show the effectiveness of the proposed algorithm.

Organization of this chapter

The remainder of the chapter is organized as follows. In Section 6.1, the problem statement and main results are given. The case study for the AIAUV is presented in Section 6.2 and the experimental study is presented in Section 6.3. In Section 6.4 a chapter summary is presented.

Publications

This chapter is based on [12] and [13].

6.1 Problem statement and main results

In this section, we prove that the proposed GSTA with adaptive gains causes the system trajectories to globally converge to zero in finite time while accounting for unknown bounds on the uncertain control coefficient and perturbation term.

6.1.1 System dynamics

Consider the dynamic system represented by the differential equation

$$\dot{\sigma} = \gamma(\sigma, t)u + \varphi(\sigma, t) \tag{6.1}$$

where $\sigma \in \mathbb{R}$ is the state vector and $u \in \mathbb{R}$ is the control input vector. The functions $\gamma(\sigma, t)$ and $\varphi(\sigma, t)$ are uncertain functions depending on the state and time. Following [18], we adopt the following assumptions:

Assumption 6.1. The functions $\gamma(\sigma, t)$ and $\varphi(\sigma, t)$ are Lipschitz continuous functions with respect to t, and $\gamma(\sigma, t), \varphi(\sigma, t) \in \mathcal{C}^1$ with respect to σ.

Assumption 6.2. The uncertain control coefficient function is bounded by

$$0 < k_m \leq \gamma(\sigma, t) \leq k_M \tag{6.2}$$

where k_m and k_M are positive constants.

Assumption 6.3. The perturbation term $\varphi(\sigma, t)$ can be split into two components:

$$\varphi(\sigma, t) = \varphi_1(\sigma, t) + \varphi_2(\sigma, t) \tag{6.3}$$

where the first component vanishes at the origin, i.e., $\varphi_1(0, t) = 0 \; \forall \; t \geq 0$, and is bounded by

$$|\varphi_1(\sigma, t)| \leq \alpha |\phi_1(\sigma)|, \qquad \alpha > 0 \tag{6.4}$$

Remark 6.1. For the adaptive GSTA which will be presented below, if we insert (6.8) for $\phi_1(\sigma)$ in (6.4), we have that $|\varphi_1(\sigma, t)| \leq \alpha ||\sigma|^{\frac{1}{2}} \operatorname{sgn}(\sigma) + \beta\sigma| \leq \alpha(|\sigma|^{\frac{1}{2}} + \beta|\sigma|)$. This means that the perturbation term $\varphi_1(\sigma, t)$ has to be contained in the sector limited by the curves $\pm\alpha(|\sigma|^{\frac{1}{2}} + \beta|\sigma|)$, and can only grow at most linearly in σ.

Assumption 6.4. The total time derivative of the non-vanishing component of the perturbation term divided by the control coefficient $\gamma(\sigma, t)$ can be represented as

$$\begin{aligned} \frac{d}{dt}\left(\gamma^{-1}(\sigma, t)\varphi_2(\sigma, t)\right) &= \underbrace{\gamma^{-1}\frac{\partial \varphi_2}{\partial t} - \gamma^{-2}\varphi_2\frac{\partial \gamma}{\partial t}}_{\delta_1(\sigma,t)} + \underbrace{\left(\gamma^{-1}\frac{\partial \varphi_2}{\partial \sigma} - \gamma^{-2}\varphi_2\frac{\partial \gamma}{\partial \sigma}\right)}_{\delta_2(\sigma,t)}\dot{\sigma} \\ &= \delta_1(\sigma, t) + \delta_2(\sigma, t)\dot{\sigma} \end{aligned} \tag{6.5}$$

where $\delta_1(\sigma,t)$ and $\delta_2(\sigma,t)$ are bounded by positive constants:

$$|\delta_1(\sigma,t)| \leq \bar{\delta}_1, \qquad |\delta_2(\sigma,t)| \leq \bar{\delta}_2 \tag{6.6}$$

Remark 6.2. This means that the non-vanishing term divided by the control coefficient $\gamma^{-1}(\sigma,t)\varphi_2(\sigma,t)$ can only grow linearly in σ. Since $\gamma(\sigma,t)$ has to be lower and upper bounded by constants (Assumptions 6.2), this means that the non-vanishing term $\varphi_2(\sigma,t)$ can only grow at most linearly in σ.

Remark 6.3. Note that Assumptions 6.2-6.4 are strongly related with the proof of the positive definiteness of the matrix $Q(t)$ in (6.13), as they are needed to be able to bound the variables in (6.15), where the bounds are defined as in (6.16).

6.1.2 Generalized super-twisting algorithm with adaptive gains

In this section, the equations describing the adaptive GSTA are presented. The GSTA proposed in [18] can be written as

$$\begin{aligned} u_{\text{AGSTA}} &= -k_1\phi_1(\sigma) + z \quad \in \mathbb{R} \\ \dot{z} &= -k_2\phi_2(\sigma) \end{aligned} \tag{6.7}$$

with

$$\begin{aligned} \phi_1(\sigma) &= \lceil\sigma\rfloor^{\frac{1}{2}} + \beta\sigma \\ \phi_2(\sigma) &= \frac{1}{2}\lceil\sigma\rfloor^0 + \frac{3}{2}\beta\lceil\sigma\rfloor^{\frac{1}{2}} + \beta^2\sigma \end{aligned} \tag{6.8}$$

where $\lceil a\rfloor^b = |a|^b \operatorname{sgn}(a)$ and $k_1 \in \mathbb{R}$, $k_2 \in \mathbb{R}$ and $\beta \in \mathbb{R}$ are constant controller gains. Motivated by [69], we propose to instead let k_1 and k_2 be adaptive gains defined by the following update laws:

$$\dot{k}_1 = \begin{cases} \omega_1\sqrt{\frac{\gamma_1}{2}}, & \text{if } \sigma \neq 0 \\ 0, & \text{if } \sigma = 0 \end{cases} \tag{6.9a}$$

$$k_2 = 2\varepsilon k_1 + \lambda + 4\varepsilon^2 \tag{6.9b}$$

where $\varepsilon \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $\gamma_1 \in \mathbb{R}$ and $\omega_1 \in \mathbb{R}$ are positive constants.

6.1.3 Closed-loop dynamics

The closed-loop dynamics are obtained by substituting (6.3) and (6.7) into (6.1), yielding

$$\dot{\sigma} = -k_1\gamma(\sigma,t)\phi_1(\sigma) + \varphi_1(\sigma,t) + \gamma(\sigma,t)\big(z + \gamma^{-1}(\sigma,t)\varphi_2(\sigma,t)\big) \tag{6.10}$$

By defining $\sigma_1 = \sigma$ and $\sigma_2 = z + \gamma^{-1}(\sigma_1,t)\varphi_2(\sigma_1,t)$, we can represent the closed-loop dynamics as

$$\dot{\sigma}_1 = \gamma(\sigma_1,t)\big(-k_1\phi_1(\sigma_1) + \gamma^{-1}(\sigma_1,t)\varphi_1(\sigma_1,t) + \sigma_2\big) \tag{6.11a}$$

$$\dot{\sigma}_2 = -k_2\phi_2(\sigma_1) + \frac{d}{dt}\big(\gamma^{-1}(\sigma_1,t)\varphi_2(\sigma_1,t)\big) \tag{6.11b}$$

Theorem 6.1. *Suppose that $\gamma(\sigma_1, t)$ and $\varphi(\sigma_1, t)$ in system (6.1) satisfy Assumptions 6.1-6.4. Then, the closed-loop dynamics in (6.11) are GFTS, such that the states σ_1 and σ_2 converge to zero and z converges to $-\gamma^{-1}(0,t)\varphi_2(0,t)$, globally and in finite time, if the gains k_1 and k_2 are designed as expressed in (6.9), $\beta > 0$, $\lambda > 0$, $\omega_1 > 0$, $\gamma_1 > 0$ and $\varepsilon = \frac{\omega_2}{2\omega_1}\sqrt{\frac{\gamma_2}{\gamma_1}}$, where $\omega_2 > 0$ and $\gamma_2 > 0$.*

Remark 6.4. Note that the proof is for a one-dimensional case; however, since SMC approaches do not use model information, the n dimensions can simply be separated into n one-dimensional cases. The proof thus holds for n dimensions as long as Assumptions 6.1-6.4 hold for each dimension. This will be demonstrated in the case study presented in Section 6.2.

Proof. From [18], we have that the closed-loop system in (6.11) is GFTS when constant values of k_1, k_2 and $\beta > 0$ are used in (6.7) and the gains are chosen in accordance with [18, Theorem 2.1]. This is proven using the Lyapunov function candidate

$$V_0 = \xi^T P \xi, \quad P = \begin{bmatrix} p_1 & -1 \\ -1 & p_2 \end{bmatrix} \tag{6.12}$$

where $p_1 p_2 > 1$ and $\xi^T = \begin{bmatrix} \phi_1(\sigma_1) & \sigma_2 \end{bmatrix}$. It is shown that the derivative along the trajectory of the system is

$$\begin{aligned} \dot{V}_0 &\leq -2\gamma(\sigma_1, t)\phi_1'(\sigma_1)\xi^T Q(t)\xi \\ &\leq -\mu_1 V_0^{\frac{1}{2}}(\sigma_1, \sigma_2) - \mu_2 V_0(\sigma_1, \sigma_2) \end{aligned} \tag{6.13}$$

where

$$\mu_1 = \frac{k_m \epsilon \lambda_{\min}^{\frac{1}{2}}\{P\}}{\lambda_{\max}\{P\}}, \qquad \mu_2 = \beta \frac{2k_m\epsilon}{\lambda_{\max}\{P\}} \tag{6.14}$$

and $Q(t)$ is positive definite if the gains are chosen in accordance with [18, Theorem 2.1]. For the proposed adaptive GSTA, however, k_1 and k_2 are not constants. Instead, k_1 and k_2 are time-varying functions given by (6.9). Motivated by [69], we use the Lyapunov function candidate defined in (6.12) to find a k_1 that satisfies (6.9a) such that $Q(t)$ is positive definite when k_2 is chosen as expressed in (6.9b). From [18], the elements of $Q(t)$ are

$$Q(t) = \begin{bmatrix} q_1(t) & q_2(t) \\ q_2(t) & q_3(t) \end{bmatrix} = \begin{bmatrix} \tilde{k}_1\tilde{p}_1 - \tilde{k}_2 & \frac{1}{2}(p_2\tilde{k}_2 - (\tilde{k}_1\tilde{h} + \tilde{p}_1)) \\ \frac{1}{2}(p_2\tilde{k}_2 - (\tilde{k}_1\tilde{h} + \tilde{p}_1)) & \tilde{h} \end{bmatrix} \tag{6.15}$$

with

$$\begin{aligned} \tilde{p}_1 &= \left(p_1 - \frac{\delta_2(\sigma_1,t)}{\phi_1'(\sigma_1)}\right) \Rightarrow \tilde{p}_1 \in [\underline{p}_1, \bar{p}_1] = \left[p_1 - \frac{\bar{\delta}_2}{\beta}, p_1 + \frac{\bar{\delta}_2}{\beta}\right] \\ \tilde{k}_2 &= \gamma^{-1}(\sigma_1,t)\left(k_2 - \frac{\delta_1(\sigma_1,t)}{\phi_2(\sigma_1)}\right) \Rightarrow \tilde{k}_2 \in [\underline{k}_2, \bar{k}_2] = \left[\frac{1}{k_M}(k_2 - 2\bar{\delta}_1), \frac{1}{k_m}(k_2 + 2\bar{\delta}_1)\right] \\ \tilde{k}_1 &= \left(k_1 - \gamma^{-1}(\sigma_1,t)\frac{\varphi_1(\sigma_1,t)}{\phi_1(\sigma_1)}\right) \Rightarrow \tilde{k}_1 \in [\underline{k}_1, \bar{k}_1] = \left[k_1 - \frac{\alpha}{k_m}, k_1 + \frac{\alpha}{k_m}\right] \\ \tilde{h} &= \left(1 - \frac{p_2\delta_2(\sigma_1,t)}{\phi_1'(\sigma_1)}\right) \Rightarrow \tilde{h} \in [\underline{h}, \bar{h}] = \left[1 - \frac{p_2\bar{\delta}_2}{\beta}, 1 + \frac{p_2\bar{\delta}_2}{\beta}\right] \end{aligned} \tag{6.16}$$

If we choose $\tilde{k}_1 = k_1 - \frac{\alpha}{k_m}$ and $\tilde{k}_2 = \frac{1}{k_m}(k_2 + 2\bar{\delta}_1)$, such that $Q(t)$ is as negative as possible, we can rewrite $Q(t)$ as

$$Q(t) = \begin{bmatrix} (k_1 - \frac{\alpha}{k_m})\tilde{p}_1 - \frac{1}{k_m}(k_2 + 2\bar{\delta}_1) & q_2(t) \\ \frac{1}{2}\Big(\frac{p_2}{k_m}(k_2 + 2\bar{\delta}_1) - \big((k_1 - \frac{\alpha}{k_m})\tilde{h} + \tilde{p}_1\big)\Big) & \tilde{h} \end{bmatrix} \tag{6.17}$$

For the matrix $Q(t)$ in (6.17) to be positive definite, we need $q_1(t) > 0$ and $\det(Q(t)) > 0$. By using k_2 as given in (6.9b) and calculating the determinant of $Q(t)$ in (6.17), we obtain

$$\begin{aligned} &\det(Q(t)) = q_1(t)q_3(t) - q_2^2(t) \\ &= \Big((k_1 - \frac{\alpha}{k_m})\tilde{p}_1 - \frac{1}{k_m}\Big(2\varepsilon k_1 + \lambda + 4\varepsilon^2 + 2\bar{\delta}_1\Big)\Big)\tilde{h} \\ &- \frac{1}{4}\Big(\frac{p_2}{k_m}\Big(2\varepsilon k_1 + \lambda + 4\varepsilon^2 + 2\bar{\delta}_1\Big) - \Big((k_1 - \frac{\alpha}{k_m})\tilde{h} + \tilde{p}_1\Big)\Big)^2 \\ &= \tilde{h}\tilde{p}_1 k_1 - \frac{\alpha\tilde{h}\tilde{p}_1}{k_m} - \frac{2\varepsilon\tilde{h}}{k_m}k_1 - \frac{\lambda\tilde{h}}{k_m} - \frac{4\varepsilon^2\tilde{h}}{k_m} - \frac{2\bar{\delta}_1\tilde{h}}{k_m} \\ &- \frac{1}{4}\Big(\frac{2\varepsilon p_2}{k_m}k_1 + \frac{\lambda p_2}{k_m} + \frac{4\varepsilon^2 p_2}{k_m} + \frac{2\bar{\delta}_1 p_2}{k_m} - \tilde{h}k_1 + \frac{\alpha\tilde{h}}{k_m} - \tilde{p}_1\Big)^2 \end{aligned} \tag{6.18}$$

By introducing $k_a = \frac{\alpha\tilde{h}\tilde{p}_1}{k_m} + \frac{\lambda\tilde{h}}{k_m} + \frac{4\varepsilon^2\tilde{h}}{k_m} + \frac{2\bar{\delta}_1\tilde{h}}{k_m} > 0$ and $k_b = \frac{\lambda p_2}{k_m} + \frac{4\varepsilon^2 p_2}{k_m} + \frac{2\bar{\delta}_1 p_2}{k_m} + \frac{\alpha\tilde{h}}{k_m} > 0$, we can rewrite (6.18) as

$$\begin{aligned} &\det(Q(t)) = \Big(\Big(\tilde{h}\tilde{p}_1 - \frac{2\varepsilon\tilde{h}}{k_m}\Big)k_1 - k_a\Big) - \frac{1}{4}\Big(\Big(\frac{2\varepsilon p_2}{k_m} - \tilde{h}\Big)k_1 + k_b - \tilde{p}_1\Big)^2 \\ &= \Big(\Big(\tilde{h}\tilde{p}_1 - \frac{2\varepsilon\tilde{h}}{k_m}\Big)k_1 - k_a\Big) - \Big(\Big(\frac{\varepsilon^2 p_2^2}{k_m^2} - \frac{\varepsilon\tilde{h}p_2}{k_m} + \frac{1}{4}\tilde{h}^2\Big)k_1^2 \\ &+ \Big(\frac{\varepsilon k_b p_2}{k_m} - \frac{\varepsilon\tilde{p}_1 p_2}{k_m} - \frac{1}{2}\tilde{h}k_b + \frac{1}{2}\tilde{h}\tilde{p}_1\Big)k_1 + \frac{1}{4}k_b^2 - \frac{1}{2}k_b\tilde{p}_1 + \frac{1}{4}\tilde{p}_1^2\Big) \\ &= \Big(-\frac{\varepsilon^2 p_2^2}{k_m^2} + \frac{\varepsilon\tilde{h}p_2}{k_m} - \frac{1}{4}\tilde{h}^2\Big)k_1^2 + \Big(\tilde{h}\tilde{p}_1 - \frac{2\varepsilon\tilde{h}}{k_m} - \frac{\varepsilon k_b p_2}{k_m} + \frac{\varepsilon\tilde{p}_1 p_2}{k_m} \\ &+ \frac{1}{2}\tilde{h}k_b - \frac{1}{2}\tilde{h}\tilde{p}_1\Big)k_1 - \frac{1}{4}k_b^2 + \frac{1}{2}k_b\tilde{p}_1 - \frac{1}{4}\tilde{p}_1^2 - k_a \\ &= (k_d - k_c)k_1^2 + (k_e - k_f)k_1 + k_h - k_g \end{aligned} \tag{6.19}$$

with

$$k_c = \frac{\varepsilon^2 p_2^2}{k_m^2} + \frac{1}{4}\tilde{h}^2 > 0 \tag{6.20a}$$

$$k_d = \frac{\varepsilon \tilde{h} p_2}{k_m} > 0 \tag{6.20b}$$

$$k_e = \tilde{h}\tilde{p}_1 + \frac{\varepsilon \tilde{p}_1 p_2}{k_m} + \frac{1}{2}\tilde{h}k_b > 0 \tag{6.20c}$$

$$k_f = \frac{2\varepsilon\tilde{h}}{k_m} + \frac{\varepsilon k_b p_2}{k_m} + \frac{1}{2}\tilde{h}\tilde{p}_1 > 0 \tag{6.20d}$$

$$k_g = \frac{1}{4}k_b^2 + \frac{1}{4}\tilde{p}_1^2 + k_a > 0 \tag{6.20e}$$

$$k_h = \frac{1}{2}k_b\tilde{p}_1 > 0 \tag{6.20f}$$

A solution to

$$\det(Q(t)) = (k_d - k_c)k_1^2 + (k_e - k_f)k_1 + k_h - k_g > 0 \tag{6.21}$$

is then

$$k_1 > \frac{k_g - k_h}{k_e - k_f} \tag{6.22}$$

where we must choose p_1 and p_2 such that $k_f < k_e$ and $k_d = k_c$. To ensure that $k_d = k_c$, we choose

$$p_2 = \frac{\tilde{h}k_m}{2\varepsilon} \tag{6.23}$$

By substituting (6.23) into (6.20a) and (6.20b), we obtain

$$\begin{aligned} k_c &= \frac{\varepsilon^2(\frac{\tilde{h}k_m}{2\varepsilon})^2}{k_m^2} + \frac{1}{4}\tilde{h}^2 = \frac{1}{2}\tilde{h}^2 \\ k_d &= \frac{\varepsilon\tilde{h}\frac{\tilde{h}k_m}{2\varepsilon}}{k_m} = \frac{1}{2}\tilde{h}^2 \end{aligned} \tag{6.24}$$

thus showing that $k_d = k_c$ is ensured. To ensure that $k_f < k_e$, we calculate

$$\begin{aligned} k_f &< k_e \\ \frac{2\varepsilon\tilde{h}}{k_m} + \frac{\varepsilon k_b p_2}{k_m} + \frac{1}{2}\tilde{h}\tilde{p}_1 &< \tilde{h}\tilde{p}_1 + \frac{\varepsilon\tilde{p}_1 p_2}{k_m} + \frac{1}{2}\tilde{h}k_b \end{aligned} \tag{6.25}$$

By substituting (6.23) into (6.25), we obtain

$$\begin{aligned} \frac{2\varepsilon\tilde{h}}{k_m} + \frac{\varepsilon k_b}{k_m}\frac{\tilde{h}k_m}{2\varepsilon} + \frac{1}{2}\tilde{h}\tilde{p}_1 &< \tilde{h}\tilde{p}_1 + \frac{\varepsilon\tilde{p}_1}{k_m}\frac{\tilde{h}k_m}{2\varepsilon} + \frac{1}{2}\tilde{h}k_b \\ \frac{2\varepsilon\tilde{h}}{k_m} + \frac{\tilde{h}k_b}{2} + \frac{1}{2}\tilde{h}\tilde{p}_1 &< \tilde{h}\tilde{p}_1 + \frac{\tilde{h}\tilde{p}_1}{2} + \frac{1}{2}\tilde{h}k_b \\ \frac{2\varepsilon}{k_m} &< \tilde{p}_1 \end{aligned} \tag{6.26}$$

This means that by choosing $\tilde{p}_1 > \frac{2\varepsilon}{k_m}$ or

$$p_1 > \frac{2\varepsilon}{k_m} + \frac{\bar{\delta}_2}{\beta} \tag{6.27}$$

we can ensure that $k_f < k_e$. Therefore, by choosing k_1 as in (6.22), p_1 as in (6.27) and p_2 as in (6.23), we obtain $\det(Q(t)) > 0$. Now, to ensure that $q_1(t) > 0$, we calculate

$$\begin{aligned} q_1(t) &> 0 \\ \Big(k_1 - \frac{\alpha}{k_m}\Big)\tilde{p}_1 - \frac{1}{k_m}(2\varepsilon k_1 + \lambda + 4\varepsilon^2 + 2\bar{\delta}_1) &> 0 \\ \Big(\tilde{p}_1 - \frac{2\varepsilon}{k_m}\Big)k_1 - \frac{\alpha}{k_m}\tilde{p}_1 - \frac{1}{k_m}(\lambda + 4\varepsilon^2 + 2\bar{\delta}_1) &> 0 \end{aligned} \tag{6.28}$$

By choosing

$$k_1 > k_q = \Big(\tilde{p}_1 - \frac{2\varepsilon}{k_m}\Big)^{-1}\Big(\frac{\alpha}{k_m}\tilde{p}_1 + \frac{1}{k_m}(\lambda + 4\varepsilon^2 + 2\bar{\delta}_1)\Big) \tag{6.29}$$

we can ensure that $q_1(t) > 0$. By combining (6.22) and (6.29), we obtain

$$k_1 > \frac{k_g}{k_e - k_f} + k_q \tag{6.30}$$

which will ensure both that $\det(Q(t)) > 0$ and that $q_1(t) > 0$. Note that the term $-k_h$ that appears in the numerator in (6.22) is removed since $k_h > k_g$ could lead to $q_1(t) < 0$. Also note that this does not affect $\det(Q(t)) > 0$ since $\frac{k_g}{k_e - k_f} > \frac{k_g - k_h}{k_e - k_f}$ because $k_h > 0$. From this, we can conclude that if we choose k_1 as in (6.30), p_1 as in (6.27) and p_2 as in (6.23), then the matrix $Q(t)$ will be positive definite, thus ensuring that the closed-loop system in (6.11) will be GFTS when constant gains are used; in other words, we have proven that there exists a gain k_1 such that $Q(t)$ is positive definite. We now need to prove that when the adaptive gains defined in (6.9) are adopted, k_1 will converge such that (6.30) is satisfied.

Now, we will use the Lyapunov function candidate

$$V = V_0 + \frac{1}{2\gamma_1}(k_1 - k_1^\star)^2 + \frac{1}{2\gamma_2}(k_2 - k_2^\star)^2 \tag{6.31}$$

where $k_1^\star > 0$ and $k_2^\star > 0$ are constants, to prove that the closed-loop dynamics in (6.11) are also GFTS with the adaptive gains given in (6.9). By taking the derivative of (6.31), we obtain

$$\dot{V} = \dot{V}_0 + \frac{1}{\gamma_1}(k_1 - k_1^\star)\dot{k}_1 + \frac{1}{\gamma_2}(k_2 - k_2^\star)\dot{k}_2 \tag{6.32}$$

By using the fact that $\dot{V}_0 \leq -\mu_1 V_0^{\frac{1}{2}}(\sigma_1, \sigma_2)$ and subtracting and adding $\frac{\omega_1}{\sqrt{2\gamma_1}}|k_1 - k_1^\star| + \frac{\omega_2}{\sqrt{2\gamma_2}}|k_2 - k_2^\star|$, we can rewrite (6.32) as

$$\begin{aligned} \dot{V} \leq\ & -\mu_1 V_0^{\frac{1}{2}} - \frac{\omega_1}{\sqrt{2\gamma_1}}|k_1 - k_1^\star| - \frac{\omega_2}{\sqrt{2\gamma_2}}|k_2 - k_2^\star| \\ & + \frac{1}{\gamma_1}(k_1 - k_1^\star)\dot{k}_1 + \frac{1}{\gamma_2}(k_2 - k_2^\star)\dot{k}_2 + \frac{\omega_1}{\sqrt{2\gamma_1}}|k_1 - k_1^\star| + \frac{\omega_2}{\sqrt{2\gamma_2}}|k_2 - k_2^\star| \end{aligned} \tag{6.33}$$

By using the well-known inequality

$$(x^2 + y^2 + z^2)^{1/2} \leq |x| + |y| + |z| \tag{6.34}$$

in (6.31), we obtain

$$\begin{aligned} \sqrt{V} &= \left(V_0 + \frac{1}{2\gamma_1}(k_1 - k_1^\star)^2 + \frac{1}{2\gamma_2}(k_2 - k_2^\star)^2\right)^{\frac{1}{2}} \\ &\leq V_0^{\frac{1}{2}} + \frac{1}{\sqrt{2\gamma_1}}|k_1 - k_1^\star| + \frac{1}{\sqrt{2\gamma_2}}|k_2 - k_2^\star| \end{aligned} \tag{6.35}$$

We can then derive

$$-\mu_1 V_0^{\frac{1}{2}} - \frac{\omega_1}{\sqrt{2\gamma_1}}|k_1 - k_1^\star| - \frac{\omega_2}{\sqrt{2\gamma_2}}|k_2 - k_2^\star| \leq -\eta\sqrt{V} \tag{6.36}$$

where $\eta = \min(\mu_1, \omega_1, \omega_2)$. Considering (6.36), we can rewrite (6.33) as

$$\begin{aligned} \dot{V} \leq &-\eta V^{\frac{1}{2}} + \frac{1}{\gamma_1}(k_1 - k_1^\star)\dot{k}_1 + \frac{1}{\gamma_2}(k_2 - k_2^\star)\dot{k}_2 \\ &+ \frac{\omega_1}{\sqrt{2\gamma_1}}|k_1 - k_1^\star| + \frac{\omega_2}{\sqrt{2\gamma_2}}|k_2 - k_2^\star| \end{aligned} \tag{6.37}$$

By [69, Proposition 1], we have that the adaptation law given in (6.9) causes the adaptive gains k_1 and k_2 to be bounded. Then, there exist positive constants $k_1^\star$ and $k_2^\star$ such that

$$k_1(t) - k_1^\star < 0, \qquad k_2(t) - k_2^\star < 0 \qquad \forall t \geq 0. \tag{6.38}$$

We can therefore reduce (6.37) to

$$\dot{V} \leq -\eta V^{\frac{1}{2}} - |k_1 - k_1^\star|\left(\frac{1}{\gamma_1}\dot{k}_1 - \frac{\omega_1}{\sqrt{2\gamma_1}}\right) - |k_2 - k_2^\star|\left(\frac{1}{\gamma_2}\dot{k}_2 - \frac{\omega_2}{\sqrt{2\gamma_2}}\right) \tag{6.39}$$

where we must ensure that

$$-|k_1 - k_1^\star|\left(\frac{1}{\gamma_1}\dot{k}_1 - \frac{\omega_1}{\sqrt{2\gamma_1}}\right) - |k_2 - k_2^\star|\left(\frac{1}{\gamma_2}\dot{k}_2 - \frac{\omega_2}{\sqrt{2\gamma_2}}\right) = 0 \tag{6.40}$$

to achieve finite-time convergence. The satisfaction of (6.40) should be achieved through the adaptation of the gains k_1 and k_2, i.e.,

$$\dot{k}_1 = \omega_1\sqrt{\tfrac{\gamma_1}{2}} \tag{6.41a}$$

$$\dot{k}_2 = \omega_2\sqrt{\tfrac{\gamma_2}{2}} \tag{6.41b}$$

If we select $\varepsilon = \frac{\omega_2}{2\omega_1}\sqrt{\frac{\gamma_2}{\gamma_1}}$, then (6.9b) and (6.41b) are equal, since

$$k_2 = 2\varepsilon k_1 + \lambda + 4\varepsilon^2 \quad \Rightarrow \quad \dot{k}_2 = 2\varepsilon\dot{k}_1 \quad \Rightarrow \quad \dot{k}_2 = \varepsilon\omega_1\sqrt{2\gamma_1} = \omega_2\sqrt{\tfrac{\gamma_2}{2}} \tag{6.42}$$

For finite-time convergence, $k_1(t)$ must satisfy (6.30). This means that $k_1(t)$ must increase in accordance with (6.41a) until (6.30) is satisfied; since $k_1(t)$ increases linearly, (6.30) will be satisfied in finite time. This guarantees the positive definiteness of the matrix $Q(t)$. Once (6.30) is satisfied, finite-time convergence is guaranteed according to (6.39), and as nicely described in the Introduction of [67], this implies that the closed-loop system in (6.11) is GFTS. □

6.2 Case Study: Articulated Intervention-AUV

In this section, we apply the theoretical results of Section 6.1 for the tracking control of an AIAUV in a case study to show the effectiveness of the proposed control algorithm.

6.2.1 Tracking control law using the adaptive GSTA

In this section, we develop a tracking control law for the AIAUV based on the adaptive GSTA and show that the tracking errors converge asymptotically to zero.

Define the virtual reference vector $\zeta_r = \zeta_d - \Lambda\tilde{\xi}_q$, where ζ_d is defined as in (2.4), $\tilde{\xi}_q$ is defined as in (2.9) and $\Lambda = \text{diag}(K_p R_I^B(p) \quad \text{sgn}(\tilde{\eta})I_3 \quad K_q)$, where $\tilde{\eta}$ is defined as in (2.8) and K_p and K_q are constant, positive definite gain matrices. Let the sliding surface be defined as

$$\sigma = \zeta - \zeta_r \quad \in \mathbb{R}^{6+(n-1)} \tag{6.43}$$

where ζ is defined as in (2.3) and let the control input be given by

$$\tau(q) = u_{\text{AGSTA}} \quad \in \mathbb{R}^{6+(n-1)} \tag{6.44}$$

where u_{AGSTA} is given in (6.7)-(6.9). By differentiating (6.43) and inserting (2.1), we obtain

$$\dot{\sigma} = \dot{\zeta} - \dot{\zeta}_r = M^{-1}(\cdot)(-C(\cdot)\zeta - D(\cdot)\zeta - g(\cdot) + \tau(\cdot)) - \dot{\zeta}_r \tag{6.45}$$

and by using the fact that $\zeta = \zeta_r + \sigma$ from (6.43), we obtain the following equation describing the dynamics of σ:

$$\dot{\sigma} = M^{-1}(\cdot)(-C(\cdot)(\zeta_r + \sigma) - D(\cdot)(\zeta_r + \sigma) - g(\cdot) + \tau(\cdot)) - \dot{\zeta}_r. \tag{6.46}$$

Now, we can separate (6.46) into $6 + (n-1)$ one-dimensional equations such that Theorem 6.1 can be used. The dynamics of σ_i can then be described by

$$\begin{aligned}\dot{\sigma}_i = {} & m_i^{-1}(\cdot)(-c_i(\cdot)(\zeta_{r,i} + \sigma_i) - d_i(\cdot)(\zeta_{r,i} + \sigma_i) - g_i(\cdot) + \tau_i(\cdot)) - \dot{\zeta}_{r,i} \\ & + \boldsymbol{m}_{\neq i}^{-1}(\cdot)(-\boldsymbol{c}_{\neq i}(\cdot)(\zeta_{r,\neq i} + \sigma_{\neq i}) - \boldsymbol{d}_{\neq i}(\cdot)(\zeta_{r,\neq i} + \sigma_{\neq i}) - g_{\neq i}(\cdot))\end{aligned} \tag{6.47}$$

where $\zeta_{r,i}$ and $\dot{\zeta}_{r,i}$ are the velocity reference and its derivative, respectively, corresponding to $\sigma_i = \zeta_i - \zeta_{r,i}$; $m_i^{-1}(\cdot)$, $c_i(\cdot)$ and $d_i(\cdot)$ are the ith elements on the diagonals of the matrices $M^{-1}(\cdot)$, $C(\cdot)$ and $D(\cdot)$, respectively; g_i is element i in the vector $g(\cdot)$; and $\tau_i(\cdot)$ is element i in the control input $\tau(\cdot)$. The vectors $\boldsymbol{m}_{\neq i}^{-1}(\cdot) \in \mathbb{R}^{6+(n-1)-1}$, $\boldsymbol{c}_{\neq i}(\cdot) \in \mathbb{R}^{6+(n-1)-1}$ and $\boldsymbol{d}_{\neq i}(\cdot) \in \mathbb{R}^{6+(n-1)-1}$ consist of the elements (i,j), where $j = 1 : 6 + (n-1)$ but $j \neq i$, in the matrices $M^{-1}(\cdot)$, $C(\cdot)$ and $D(\cdot)$, respectively. The vectors $\zeta_{r,\neq i} \in \mathbb{R}^{6+(n-1)-1}$, $\sigma_{\neq i} \in \mathbb{R}^{6+(n-1)-1}$ and $g_{\neq i}(\cdot) \in \mathbb{R}^{6+(n-1)-1}$ consists of the elements j, where $j = 1 : 6 + (n-1)$ but $j \neq i$, in the velocity reference vector ζ_r, the state vector σ and the vector $g(\cdot)$, respectively.

Now, by introducing $\varphi_i(\sigma_i,t) = \varphi_{1,i}(\sigma_i,t)+\varphi_{2,i}(\sigma_i,t)$, where $\varphi_{1,i}(0,t)=0$, and $\gamma_i(\cdot) = m_i^{-1}(\cdot)$ and substituting $\tau_i(\cdot)$ from (6.44) and (6.7)-(6.9) into the above expression, we obtain

$$\dot{\sigma}_i = -k_{1,i}\gamma_i(\cdot)\phi_1(\sigma_i) + \varphi_{1,i}(\sigma_i,t) + \gamma_i(\cdot)(z_i + \gamma_i^{-1}(\cdot)\varphi_{2,i}(\sigma_i,t)) \tag{6.48}$$

where $\varphi_{1,i}(\sigma_i,t) = \gamma_i(\cdot)(-c_i(\cdot)\sigma_i - d_i(\cdot)\sigma_i)$ and $\varphi_{2,i}(\sigma_i,t) = \gamma_i(\cdot)(-c_i(\cdot)\zeta_{r,i} - d_i(\cdot)\zeta_{r,i} - g_i(\cdot) - m_i(\cdot)\dot{\zeta}_{r,i}) + \boldsymbol{m}_{\neq i}^{-1}(\cdot)(-\boldsymbol{c}_{\neq i}(\cdot)(\zeta_{r,\neq i}+\sigma_{\neq i}) - \boldsymbol{d}_{\neq i}(\cdot)(\zeta_{r,\neq i}+\sigma_{\neq i}) - g_{\neq i}(\cdot))$. By setting $\sigma_{1,i} = \sigma_i$ and $\sigma_{2,i} = z_i + \gamma_i^{-1}(\cdot)\varphi_{2,i}(\sigma_i,t)$, we can then write the dynamics as

$$\begin{aligned} \dot{\sigma}_{1,i} &= -k_{1,i}\gamma_i(\cdot)\phi_1(\sigma_{1,i}) + \varphi_{1,i}(\sigma_{1,i},t) + \gamma_i(\cdot)\sigma_{2,i} \\ \dot{\sigma}_{2,i} &= -k_{2,i}\phi_2(\sigma_{1,i}) + \frac{d}{dt}\big(\gamma_i^{-1}(\cdot)\varphi_{2,i}(\sigma_{1,i},t)\big) \end{aligned} \tag{6.49}$$

Now, if $\gamma(\cdot)$, $\varphi_{1,i}(\cdot)$ and $\varphi_{2,i}(\cdot)$ satisfy Assumptions 6.1-6.4, then Theorem 6.1 will be satisfied, and the dynamics in (6.49) will be GFTS. The sliding surface $\sigma = 0$ is then a GFTS equilibrium point, which means that σ converges to zero in finite time.

Now, to prove that $\gamma(\cdot)$, $\varphi_{1,i}(\cdot)$ and $\varphi_{2,i}(\cdot)$ satisfy Assumptions 6.1-6.4, we need to adopt the following assumptions regarding the AIAUV:

Assumption 6.5. The AIAUV is neutrally buoyant.

Assumption 6.6. The AIAUV has only revolute joints. The following properties then hold [4]:

1. Property 1: $0 < \lambda_{\min}(M) \leq ||M|| \leq \lambda_{\max}(M)$.
2. Property 2: $M = M^T > 0$.
3. Property 3: $\dot{M} = C + C^T$ and $\zeta^T(\dot{M} - 2C)\zeta = 0 \quad \forall \quad \zeta \in \mathbb{R}^{6+(n-1)}$.

Assumption 6.7. The reference trajectory and its derivatives are continuous and bounded by design.

Remark 6.5. Assumptions 6.5-6.7 are valid due to the design of the AIAUV and the reference trajectories.

Assumption 6.8. The following conditions hold:

1. $|\gamma_i(\cdot)c_i(\cdot)| \leq k_{\alpha,c}$, where $k_{\alpha,c}$ is a positive constant.
2. $|\gamma_i(\cdot)d_i(\cdot)| \leq k_{\alpha,d}$, where $k_{\alpha,d}$ is a positive constant.

Assumption 6.9. The following conditions hold:

1. $|\frac{d}{dt}(c_i(\cdot))| \leq k_{\delta_1,c_1} + k_{\delta_2,c_1}|\dot{\sigma}_i|$, where k_{δ_1,c_1} and k_{δ_2,c_1} are positive constants.
2. $|\frac{d}{dt}(d_i(\cdot))| \leq k_{\delta_1,d_1} + k_{\delta_2,d_1}|\dot{\sigma}_i|$, where k_{δ_1,d_1} and k_{δ_2,d_1} are positive constants.
3. $|\frac{d}{dt}(g_i(\cdot))| \leq k_{\delta_1,g} + k_{\delta_2,g}|\dot{\sigma}_i|$, where k_{δ_1,g_1} and k_{δ_2,g_1} are positive constants.
4. $|\frac{d}{dt}\big(\gamma_i^{-1}(\cdot)(\boldsymbol{m}_{\neq i}^{-1}(\cdot)\boldsymbol{c}_{\neq i}(\cdot))\big)| \leq k_{\delta_1,c_2} + k_{\delta_2,c_2}|\dot{\sigma}_i|$, where k_{δ_1,c_2} and k_{δ_2,c_2} are positive constants.

5. $|\frac{d}{dt}(\gamma_i^{-1}(\cdot)(\boldsymbol{m}_{\neq i}^{-1}(\cdot)\boldsymbol{d}_{\neq i}(\cdot)))| \leq k_{\delta_1,d_2} + k_{\delta_2,d_2}|\dot{\sigma}_i|$, where k_{δ_1,d_2} and k_{δ_2,d_2} are positive constants.
6. $|\frac{d}{dt}(\gamma_i^{-1}(\cdot)(\boldsymbol{m}_{\neq i}^{-1}(\cdot)g_{\neq i}(\cdot)))| \leq k_{\delta_1,g_2} + k_{\delta_2,g_2}|\dot{\sigma}_i|$, where k_{δ_1,g_2} and k_{δ_2,g_2} are positive constants.

Remark 6.6. For all practical purposes, Assumptions 6.8 and 6.9 will be satisfied since the AIAUV is a mechanical system and has a limited control input, which will cause the velocities of the system to also be bounded. For theoretical purposes, it should be proven that these assumptions hold; however, this proof will be left as a topic for future work. Some of the assumptions can also be discarded when the adaptive GSTA is extended to MIMO systems; this is also a topic for future work.

Now, Assumption 6.2 holds because of Property 1 in Assumption 6.6, while Assumptions 6.3 and 6.4 hold because of Assumptions 6.8 and 6.9, respectively. The conditions of Theorem 6.1 are thus satisfied in each dimension, and therefore, by Theorem 6.1, the dynamics in (6.49) are GFTS. The sliding surface $\sigma = 0$ is therefore a GFTS equilibrium point, which means that σ converges to zero in finite time.

Once the system trajectories are confined to $\sigma = 0$, the tracking error dynamics are given by

$$\zeta - \zeta_d = -\Lambda\tilde{\xi}_q \quad \Leftrightarrow \quad \tilde{v} = -K_p R_I^B(p)\tilde{\eta}_1 \tag{6.50a}$$

$$\tilde{\omega} = -\operatorname{sgn}(\tilde{\eta})\tilde{\varepsilon} \tag{6.50b}$$

$$\dot{\tilde{q}} = -K_q\tilde{q} \tag{6.50c}$$

For the position error (6.50a) it is proven in [27] that the equilibrium point $\tilde{\eta}_1$ is UGAS, and for the orientation error (6.50b) it is proven in [27] that there are two uniformly asymptotically stable equilibrium points, $[\tilde{\varepsilon}, \tilde{\eta}] = [0, \pm 1]$, both representing perfect alignment between the desired and the actual orientation of the AIAUV. Finally, (6.50c), shows that the joint angle errors, $\tilde{q}$, converge uniformly globally exponentially to zero, since K_q is chosen positive definite.

6.2.2 Simulation results

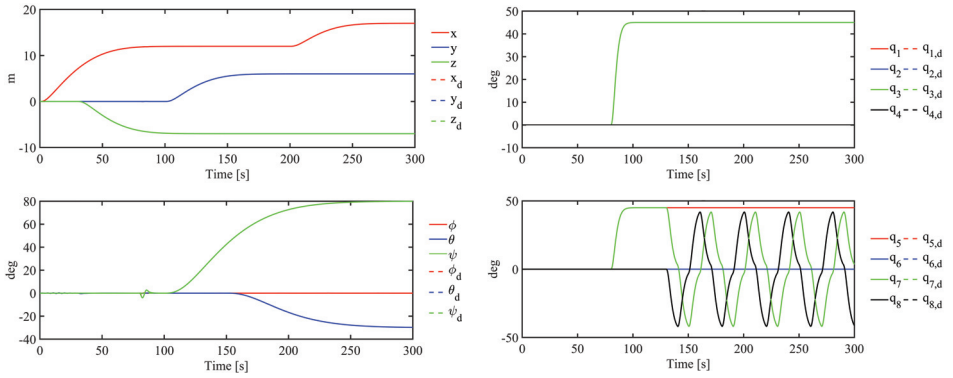
In this section, we present simulation results for trajectory tracking using the control law presented in Section 6.2.1. We present results showing the optimal performance of the algorithm.

Adaptive gains

For implementation purposes, a small boundary is put on σ so the adaptive gains can be expressed as

$$\dot{k}_{1,i} = \begin{cases} \omega_{1,i}\sqrt{\frac{\gamma_{1,i}}{2}}, & \text{if } |\sigma_i| > \alpha_{m,i} \\ 0, & \text{if } |\sigma_i| \leq \alpha_{m,i} \end{cases} \tag{6.51a}$$

$$k_{2,i} = 2\varepsilon_i k_{1,i} + \lambda_i + 4\varepsilon_i^2 \tag{6.51b}$$



(a) Tracking of position and orientation

(b) Tracking of joint angles

Figure 6.1: Tracking results using adaptive GSTA

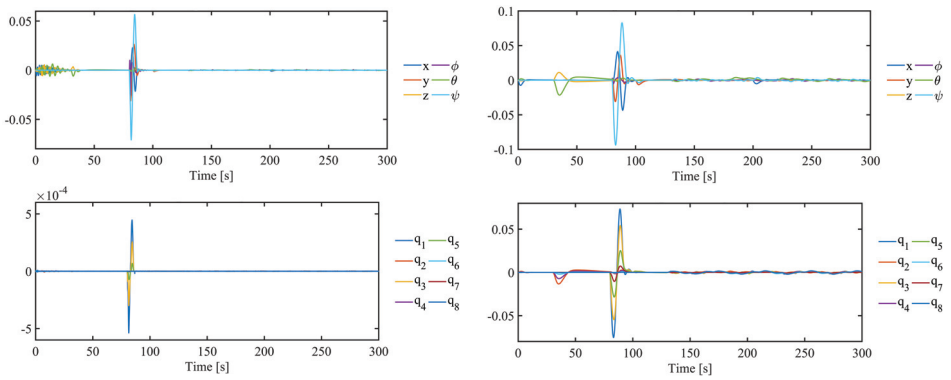
where the design parameter $\alpha_m \in \mathbb{R}^{6+(n-1)}$ is a small positive constant chosen empirically. This boundary was introduced because numerically σ will never be exactly zero, meaning that in practical implementations the second condition of (6.9a) will never be met, which would make the adaptive gains increase to infinity.

Optimal performance

In this section we present results showing the optimal performance of the algorithm, and compare them with results obtained using the GSTA. We use the simulation model presented in Section 2.2 and the Eelume 2016 version of the robot presented in Section 2.2.1.

The simulation case chosen highlights the two different modes, transport mode and operation mode, which are explained in detail in [79]. Transport mode is for long-distance travel, and operation mode is for performing inspections and intervention tasks. In this case we start in transport mode, i.e. start with the robot in an I-shape (all the joint angles are equal to zero: $q = 0\,\text{deg}$), moving forward in the x-direction and downward in the z-direction, before moving into operation mode at 80s. In operation mode we perform an inspection, by setting the AIAUV in C-shape (every second joint angle is equal to 45 deg), and then moving joint 7 and 8 in a circular motion such that link 9 is moving around, performing an inspection. For the simulations, a fixed-step solver with a step size of 10^{-4} was used. The gains were chosen as: $\beta_{\text{AGSTA}} = [80e_{14}]^T$, $\lambda = [0.1e_6 \quad 10e_8]^T$, $\varepsilon = \frac{\omega_2}{2\omega_1}\sqrt{\frac{\gamma_2}{\gamma_1}}[e_{14}]^T$, $\gamma_1 = [e_{14}]^T$, $\omega_1 = [2e_{14}]^T$, $\gamma_2 = [4\cdot 10^{-10}e_3 \quad 8\cdot 10^{-9}e_3 \quad 4\cdot 10^{-10}e_8]^T$, $\omega_2 = [2e_{14}]^T$ and $\alpha_m = [5\cdot 10^{-3}e_6 \quad 5\cdot 10^{-6}e_8]^T$, where e_i is a $1\times i$ vector of ones. In Figure 6.1 the reference trajectory for the position, the orientation and the movement of the joints together with the states are shown. In Figure 6.2a the sliding surface σ is shown, in Figure 6.3a the thruster forces and joint torques used are shown, and in Figure 6.4 the evolution of the gain $k_1(t)$ over time is shown.

From Figure 6.1 we can see that the desired trajectories for position, orientation and the joints are tracked very well. This is also supported by Figure 6.2, as σ is



(a) Sliding surface σ using adaptive GSTA

(b) Sliding surface σ using GSTA

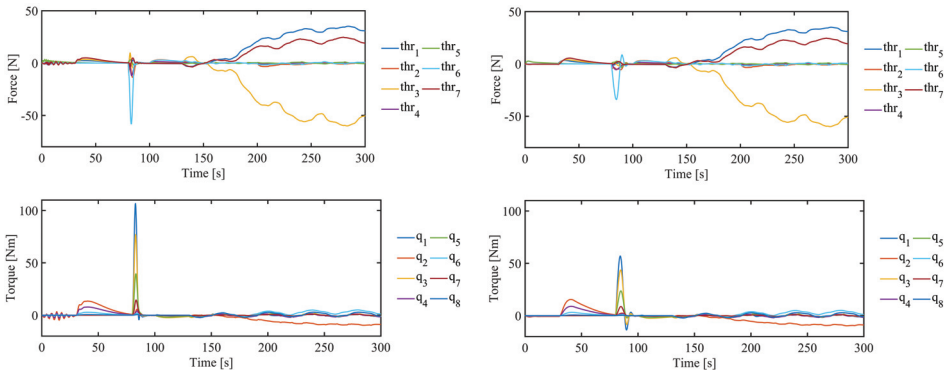
Figure 6.2: Sliding surface σ

below 0.07 for the position and orientation and below $5 \cdot 10^{-4}$ for the joints in absolute value. From Figure 6.3 we can see that the forces used is smooth, i.e. no chattering and applicable. In Figure 6.4 we can see that the gain $k_1(t)$ increases linearly and converges to a suitable value, this fits what we found in Section 6.1.

For comparison, in Figure 6.2b the sliding surface σ and in Figure 6.3b the thruster forces and joint torques are shown, when the GSTA with constant gains is used, with gains $k_1 = [5e_{14}]^T$, $k_2 = [0.02e_{14}]^T$ and $\beta_{\text{GSTA}} = [80e_{14}]^T$. By comparing the sliding surfaces in Figure 6.2 we can see that the adaptive GSTA gives considerably smaller sliding surface values than the GSTA with constant gains. By comparing the thruster forces and joint torques in Figure 6.3 we can see that when the joints move at 80 s (because the AIAUV is moving into operation mode) the adaptive GSTA provides higher thruster forces and joint torques. The rest of the time the thruster forces and joint torques used are similar. This is probably some of the reason why the adaptive GSTA gives a much smaller sliding surface. However, we do also see that the sliding surface is smaller for the adaptive GSTA when the thruster forces and joint torques used are similar. We can therefore conclude that the adaptive GSTA here gives better tracking performance than the GSTA with constant gains.

6.3 Experimental study: Articulated Intervention AUV

In this section we will describe an experimental study with an AIAUV to show how the proposed control algorithm performs in experiments. We first show realistic simulation results where the control gains used will be those obtained in experiments during the summer of 2020. We then present the experimental results obtained during the summer of 2020. We also present a comparison with the STA with adaptive gains (3.1) and the original GSTA (4.1) to evaluate whether adding adaptive gains to the GSTA actually improves the resulting tracking capabilities, as we then get both the theoretical advantages afforded by the GSTA and the practical advantages afforded by adaptive gains. Additionally, we compare the results



(a) Control input using adaptive GSTA (b) Control input using GSTA

Figure 6.3: Control input

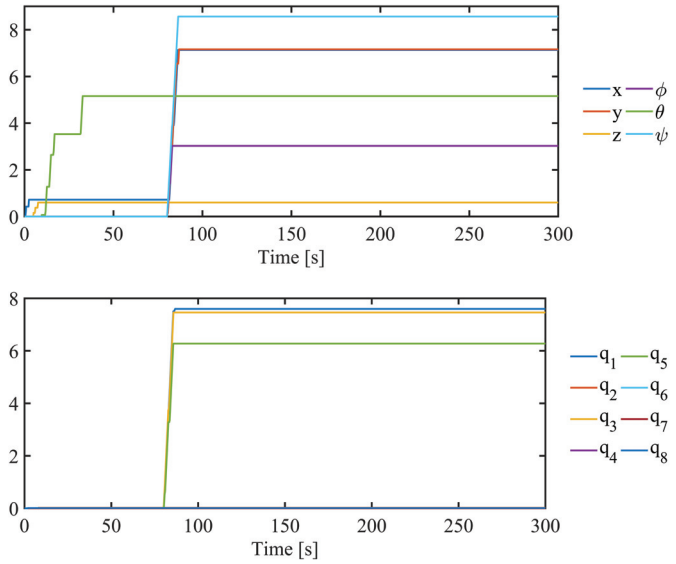


Figure 6.4: Evolution of $k_1(t)$ over time

with those of a standard PID controller.

6.3.1 Realistic simulations

In this section, simulation results with control gains obtained in experiments during the summer of 2020 will be presented. We used the simulation model presented in Section 2.2 and the Eelume 2020 version of the robot presented in Section 2.2.2. The test case used is the inspection case explained in Section 2.4.3. For the simulations, the ode3 fixed-step solver with a step size of 0.01 was used.

Results AGSTA

The gains used for the GSTA with adaptive gains are presented in Table 6.1. The

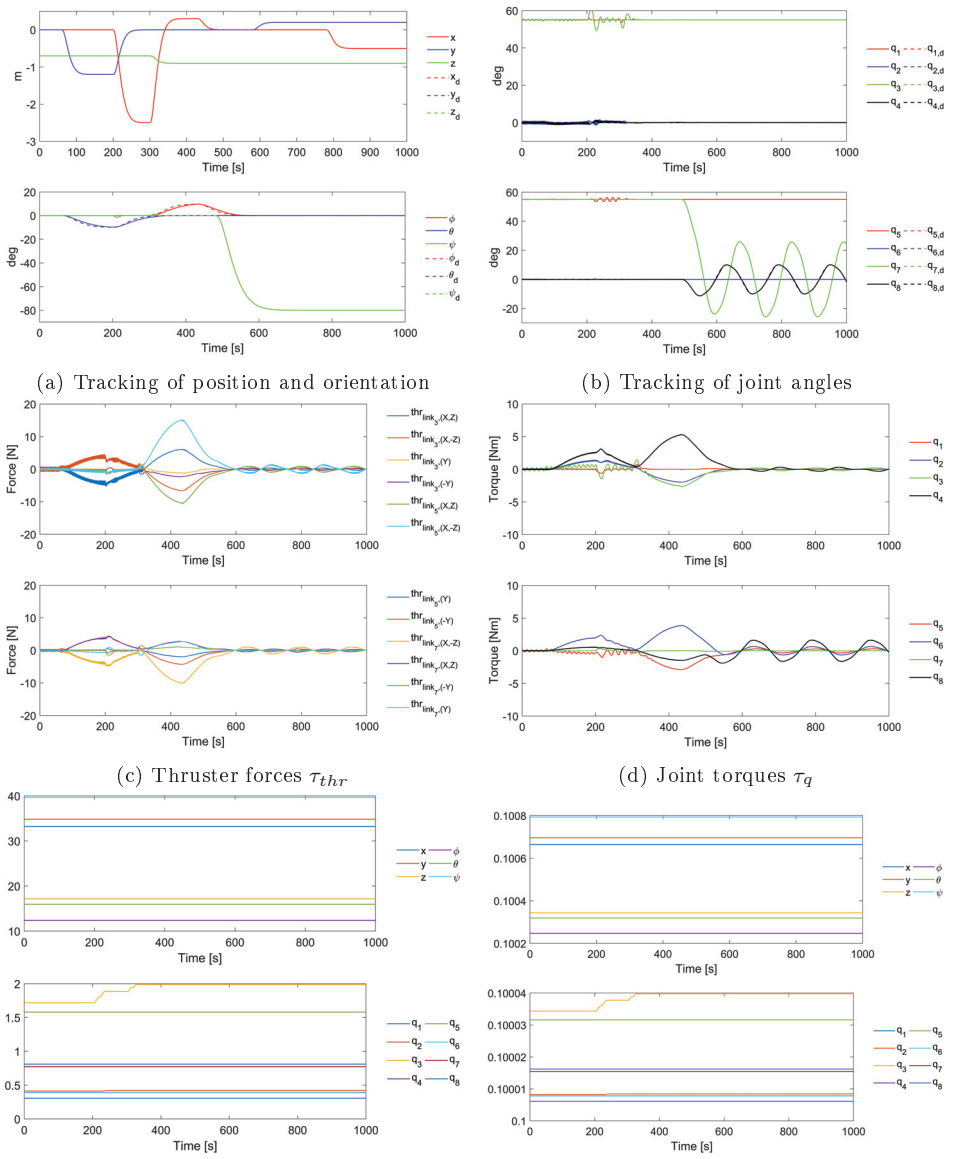
Table 6.1: Control gains for the adaptive GSTA for the inspection case

Gain	Values
ε	$1 \cdot 10^{-5} e_{14}^T$
λ	$0.1 e_{14}^T$
γ_1	e_{14}^T
ω_1	$[e_6 \quad 0.01 e_8]^T$
β	$[5e_6 \quad e_8]^T$
α_m	$0.05 e_{14}^T$

results obtained when using the control law proposed in Section 6.2.1 are presented in Figure 6.5. In Fig. 6.5d label q_i corresponds to the torque used for joint i, and in Fig. 6.5e and Fig. 6.5f label x corresponds to the evolution of the adaptive gains $k_1(t)$ and $k_2(t)$, respectively, over time for the x-dimension, label y corresponds to the evolution of the adaptive gains $k_1(t)$ and $k_2(t)$ over time for the y-dimension, etc.

Results comparison

For comparison, we also obtained results using the previously mentioned algorithms, i.e. the STA with adaptive gains (3.1) and the GSTA (4.1). For both the STA with adaptive gains and the GSTA, we used the sliding surface defined in (6.43). We also compare the results with those of a standard PID controller (5.1). The reason we chose to include a PID controller as a representative standard control method is because it is one of the most widely used types of controllers and is known to yield good results if tuned correctly. However, it does not provide any stability guarantee. We used the gains found for the STA with adaptive gains, the GSTA and the PID controller during the experiments during the summer of 2020, which are presented in Table 6.2, Table 6.3 and Table 6.4, respectively. The tracking errors for all algorithms are presented in Figure 6.6. In Figure 6.7, we present the thruster forces and joint torques for all algorithms except the adaptive GSTA, for which the forces and torques have been previously presented in Figure 6.5. For the STA with adaptive gains, the evolution of the adaptive gains over time can be



(a) Tracking of position and orientation

(b) Tracking of joint angles

(c) Thruster forces τ_{thr}

(d) Joint torques τ_q

(e) Evolution of $k_1(t)$ over time

(f) Evolution of $k_2(t)$ over time

Figure 6.5: Simulation results using the adaptive GSTA for the inspection case

Table 6.2: Control gains for the STA with adaptive gains for the inspection case

Gain	Values
ε	$1 \cdot 10^{-5} e_{14}^T$
λ	$0.1 e_{14}^T$
γ_1	e_{14}^T
ω_1	$[e_6 \quad 0.01 e_8]^T$
α_m	$0.05 e_{14}^T$

Table 6.3: Control gains for the GSTA for the inspection case

Gain	Values
k_1	$[2e_6 \quad e_8]^T$
k_2	$[0.004e_6 \quad 0.002e_8]^T$
β	$[80e_6 \quad 15e_8]^T$

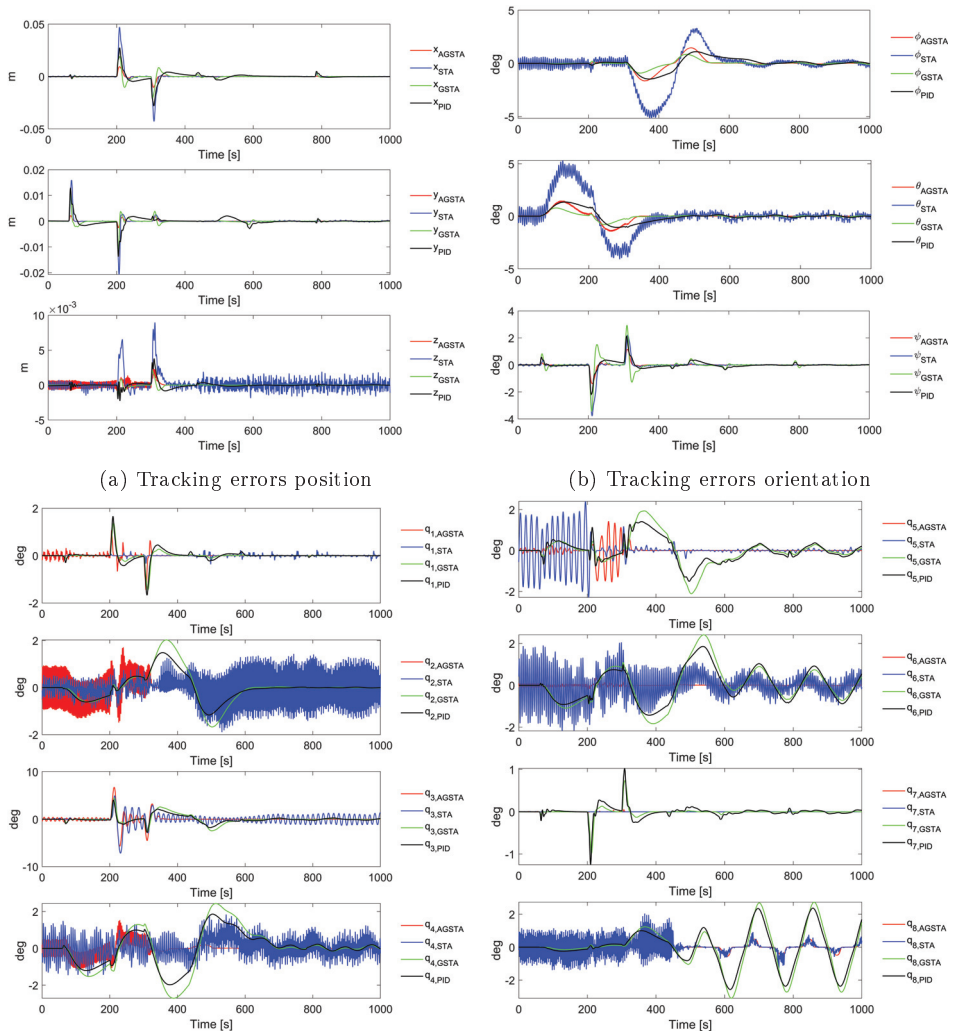
Table 6.4: Control gains for the PID controller for the inspection case

Gain	Values
k_p	$[250e_6 \quad 20e_8]^T$
k_d	$[80e_6 \quad 5e_8]^T$
k_i	$[5e_6 \quad e_8]^T$

seen in Figure 6.8. In Table 6.5, the RMSEs and maximum errors of the position, orientation and joint angles are given for all the algorithms. In Table 6.6, the RMS values of the thruster forces, the derivative of the thruster forces, the joint torques, and the derivative of the joint torques are given for all the algorithms. The RMS of the thruster forces and the joint torques provides a measure of how much control effort that is used, and the RMS of the derivative of the thruster forces and the derivative of the joint torques provides a measure of how much chattering that is present in the thruster forces and the joint torques.

Discussion

From the adaptive GSTA simulation results in Figure 6.5 we can see that the desired trajectories for position, orientation and the joint angles are tracked very well. We can see from Figure 6.5b that we get some small oscillations on joints q_3 and q_5 after approximately 200 s. These oscillations may occur because of the position and orientation movement of the AIAUV. From Figure 6.5c, we can see that the forces used are well below 60 N, which is approximately the limit of the thrusters in the 2020 version of the robot. We do see some chattering in the beginning of the simulation, however, this stops after approximately 300 s. From Figure 6.5d, we can see that the joint torques are smooth and below 16 Nm, which is the limit on the joint torques. The control inputs are therefore applicable. From Figure 6.5e and Figure 6.5f, we can also see that the gains $k_1(t)$ and $k_2(t)$ converge to suitable values, which agrees with the theoretical results we found in Section 6.1.



(a) Tracking errors position

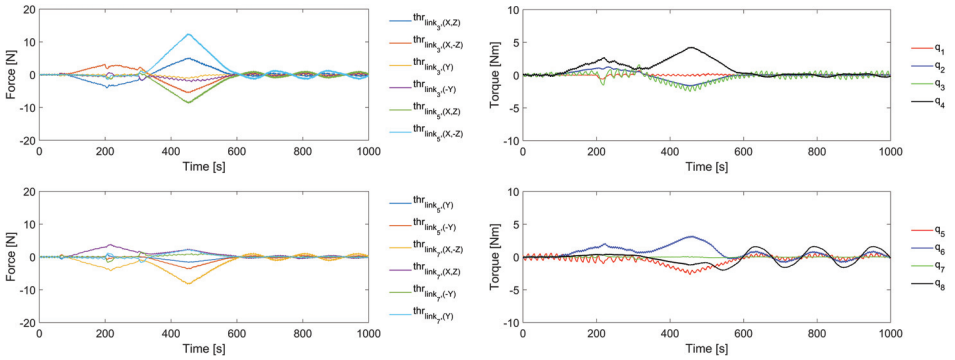
(b) Tracking errors orientation

(c) Tracking errors joints q_1-q_4

(d) Tracking errors joints q_5-q_8

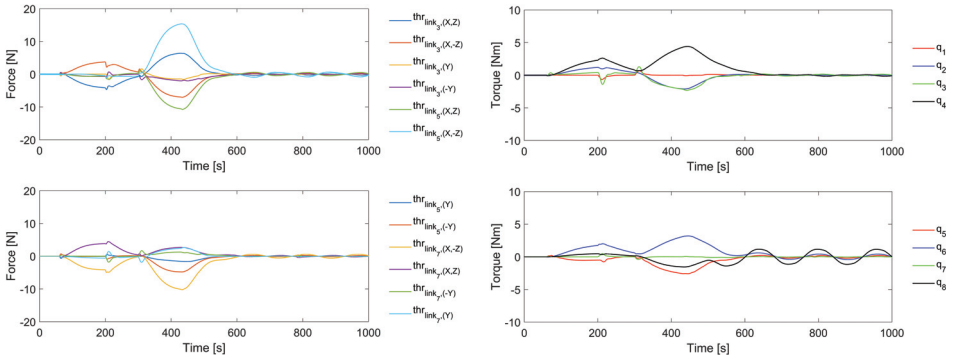
Figure 6.6: Simulation: Tracking errors for the inspection case

From Figure 6.6a, we can see that the adaptive GSTA clearly results in the smallest position errors. In orientation however, we can see from Figure 6.6b that the results are quite similar for the adaptive GSTA, the GSTA and the PID controller, while the STA with adaptive gains have larger overshoots. From Figure 6.6c and Figure 6.6d we can see that the two adaptive algorithms have more chattering and oscillations in the states, while the GSTA and the PID controller have larger errors in most of the states. These findings are further supported by Table 6.5, which shows that the adaptive GSTA gives the best RMSEs and maximum error values for position and for the joints. For orientation however, we see that the



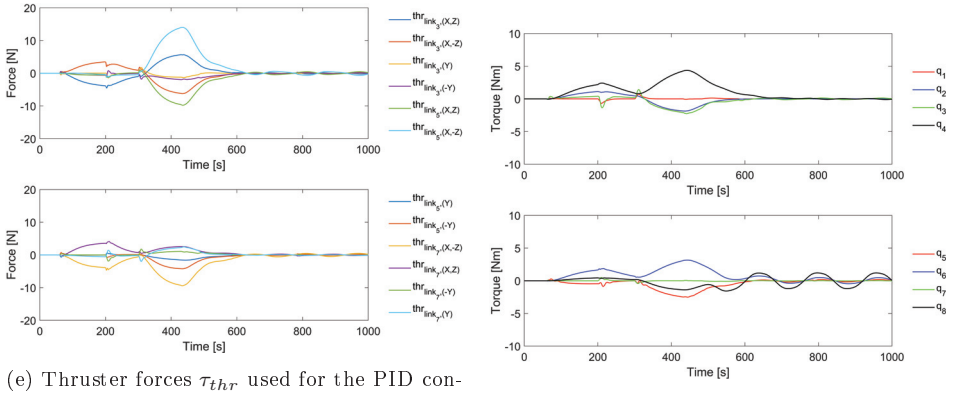
(a) Thruster forces τ_{thr} used for STA with adaptive gains

(b) Joint torques τ_q used for STA with adaptive gains



(c) Thruster forces τ_{thr} used for the GSTA

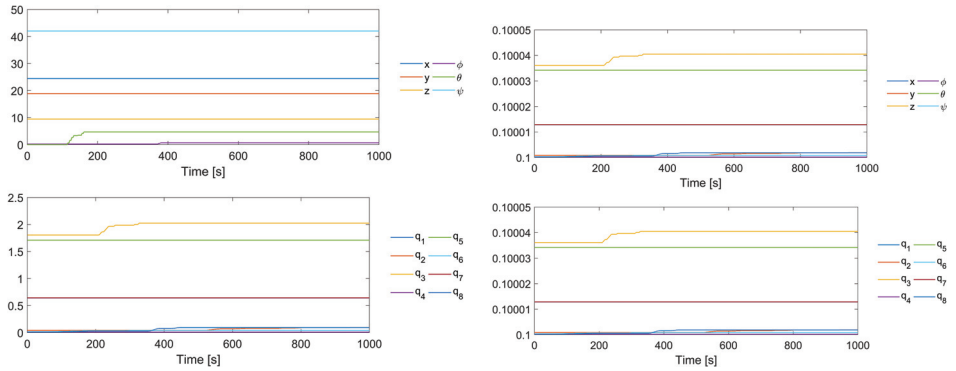
(d) Joint torques τ_q used for the GSTA



(e) Thruster forces τ_{thr} used for the PID controller

(f) Joint torques τ_q used for the PID controller

Figure 6.7: Simulation: Thruster forces τ_{thr} and joint torques τ_q used for the inspection case



(a) Evolution of $\alpha(t)$ over time

(b) Evolution of $\beta(t)$ over time

Figure 6.8: Simulation: Evolution of the adaptive gains over time for the STA with adaptive gains for the inspection case

Table 6.5: Simulation: Comparison of the tracking results for the inspection case

Algorithm		**RMSE**	**Maximum error**
AGSTA	Position	0.0007	0.0052
	Orientation	0.0066	0.0262
	Joints	0.0050	0.0305
STA	Position	0.0029	0.0256
	Orientation	0.0209	0.0829
	Joints	0.0091	0.0391
GSTA	Position	0.0014	0.0114
	Orientation	0.0056	0.0300
	Joints	0.0130	0.0421
PID	Position	0.0020	0.0152
	Orientation	0.0078	0.0292
	Joints	0.0108	0.0357

GSTA actually gives the best RMSE, while the adaptive GSTA gives the best maximum error, however the difference is not large. We can therefore conclude that the adaptive GSTA exhibits the best tracking performance overall, i.e., the smallest errors. Regarding the thruster and torque use of the different algorithms, we can see from Figure 6.5c, Figure 6.5d and Figure 6.7 that the force use is quite similar for all of the considered algorithms. However, from Table 6.6, we can see that there are some small differences between the algorithms. From the RMS of the thruster forces we see that it is actually the GSTA that uses the most thruster force, while the adaptive GSTA is not far behind, which may be the reason why the GSTA performs so well in orientation. We can also see that the STA with adaptive gains is the algorithm that uses the least amount of thruster force, which may be the reason why it performs the overall worst on tracking in position and orientation.

Table 6.6: Simulation: Comparison of the control inputs for the inspection case

Algorithm	RMS			
	τ_{thr}	$\dot{\tau}_{thr}$	τ_q	$\dot{\tau}_q$
AGSTA	1.8466	7.2346	0.8684	0.6239
STA	1.5135	0.2469	0.7505	0.1109
GSTA	1.8969	6.6455	0.7811	2.4089
PID	1.7548	0.0423	0.7665	0.0182

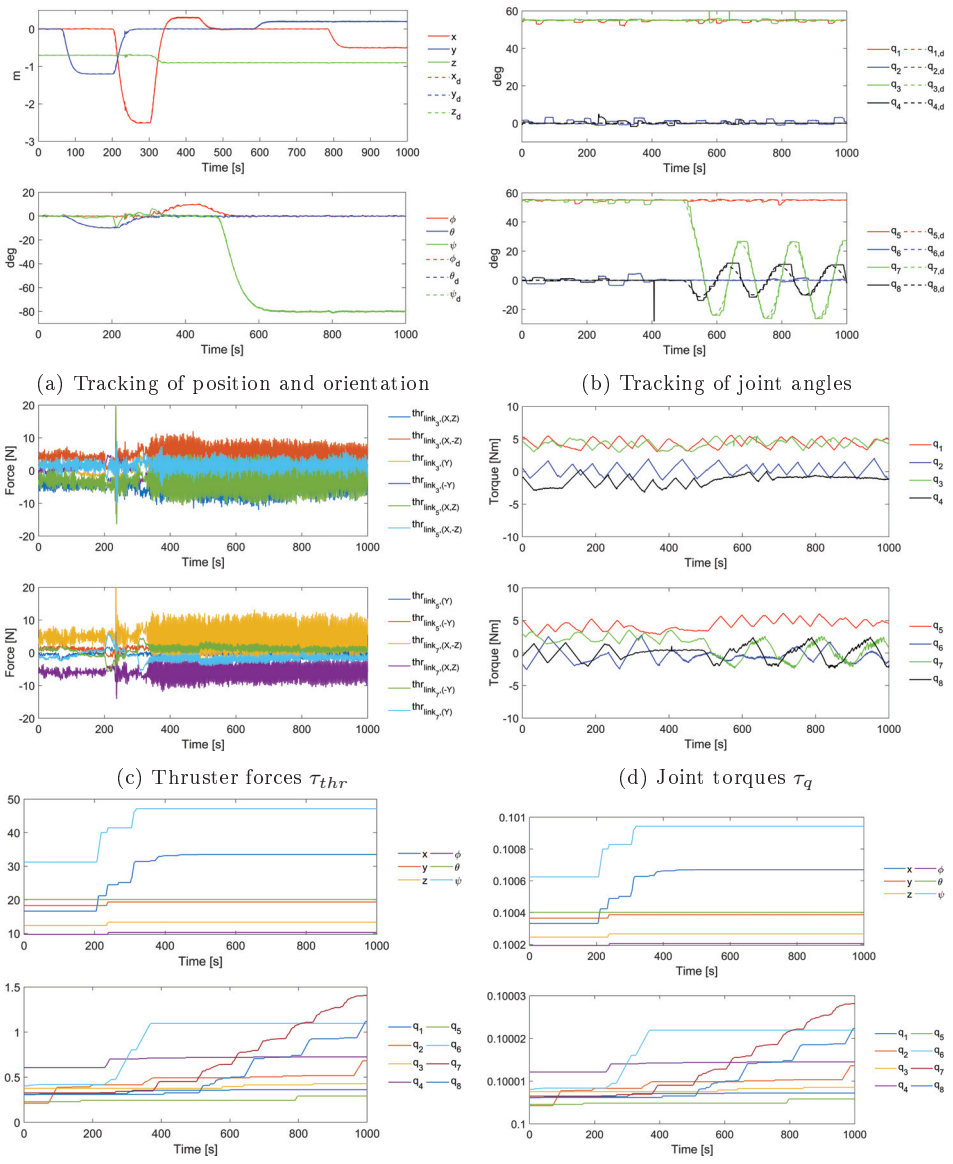
If we look at the RMS value for the joint torques, however, we can see that it is the adaptive GSTA which uses the most torque, which may be one of the reasons that it gives such good results on tracking for the joints. We can also see that the STA with adaptive gains uses the least amount of torque, but it still gives better tracking for the joints, than the GSTA and the PID controller, however, the difference in torque use is quite small. From the RMS value of $\dot{\tau}_{thr}$ we can see that the adaptive GSTA and the GSTA are the ones that introduces the most chattering, this may be connected with that they are the algorithms that also gave the best tracking performance. From the RMS value of $\dot{\tau}_q$, we can, however, see that the GSTA have the highest value, even though the adaptive GSTA and the STA with adaptive gains give better tracking for the joints than the GSTA. However, here the RMS values are so small that the chattering that is introduced by the GSTA is not visible in Figure 6.7d, and therefore negligible.

6.3.2 Experimental investigation

In this section, the experimental results obtained during the summer of 2020 are presented. The purpose of the experiments was to validate the underlying theory and the robustness of the control approach by showing that the proposed approach also works in practice and not only in the ideal case presented in simulations. The robot used was the Eelume 2020 version of the robot presented in Section 2.4.1 and the test case used was the same as for the simulation, and was explained in Section 2.4.3.

Results AGSTA

The gains used for the GSTA with adaptive gains are presented in Table 6.1. For the adaptive GSTA the choice of initial gains is not very important since the gains will autonomously adapt to a suitable value by themselves, however, the starting point of the adaptive gains does determine the convergence rate and convergence time of the adaptive gains, which means that some effort should be used to select the starting point of the adaptive gains. During the experiments we noticed that it was easy to find a starting point for the adaptive gains that yielded good tracking results. The results obtained using the control law proposed in Section 6.2.1 are presented in Figure 6.9.



(a) Tracking of position and orientation

(b) Tracking of joint angles

(c) Thruster forces τ_{thr}

(d) Joint torques τ_q

(e) Evolution of $k_1(t)$ over time

(f) Evolution of $k_2(t)$ over time

Figure 6.9: Experimental results using the adaptive GSTA for the inspection case

Results comparison

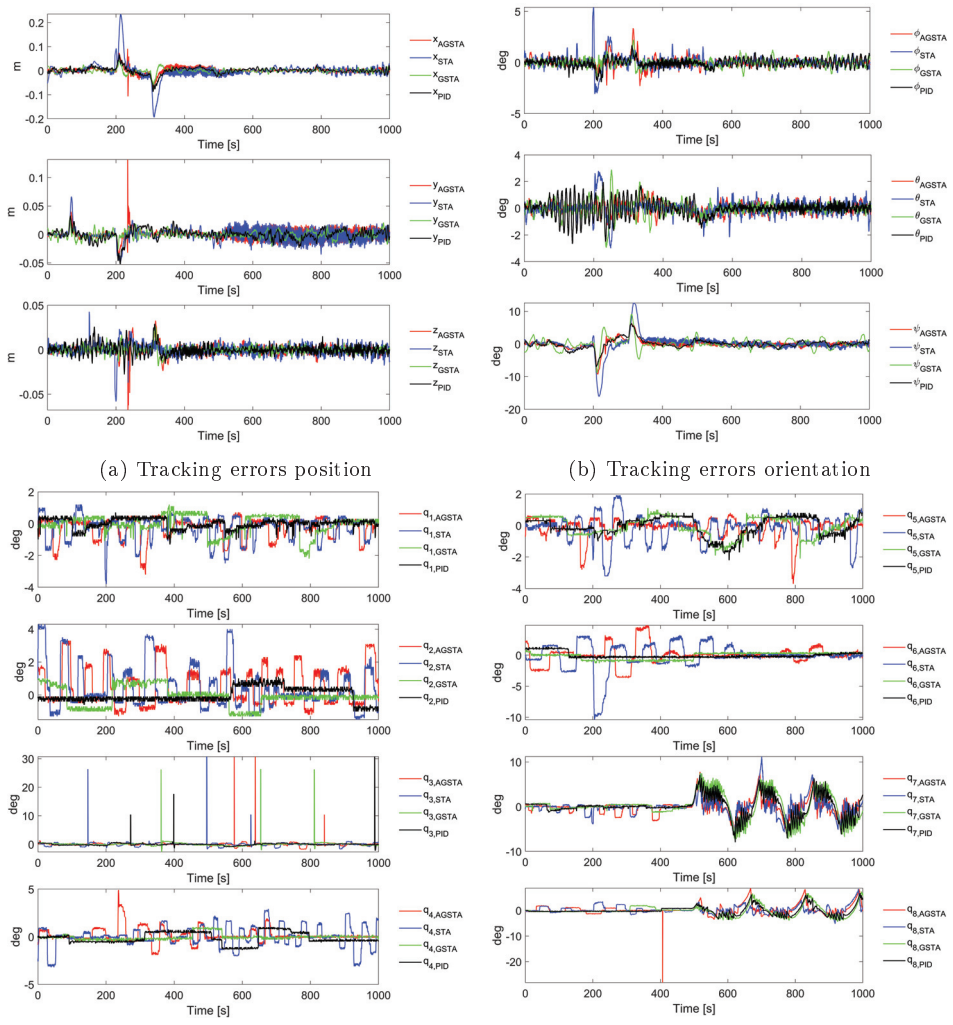
For comparison, we also obtained results using the previously mentioned algorithms, i.e. the STA with adaptive gains (3.1), the GSTA (4.1) and a PID controller (5.1). For both the STA with adaptive gains and the GSTA, we used the sliding surface defined in (6.43). The gains found for the STA with adaptive gains, the GSTA and the PID controller during the experiments during the summer of 2020 are presented in Table 6.2, Table 6.3 and Table 6.4, respectively. The tracking errors for all algorithms are presented in Figure 6.10. In Figure 6.11, we present the thruster forces and joint torques for all algorithms except the adaptive GSTA, for which the forces and torques have been previously presented in Figure 6.9. For the STA with adaptive gains, the evolution of the adaptive gains over time can be seen in Figure 6.12. In Table 6.7, the RMSEs and maximum errors of the position, orientation and joint angles are given for all the algorithms. In Table 6.8, the RMS values of the thruster forces, the derivative of the thruster forces, the joint torques, and the derivative of the joint torques are given for all the algorithms.

Table 6.7: Experiments: Comparison of the tracking results for the inspection case

Algorithm		**RMSE**	**Maximum error**
AGSTA	Position	0.0096	0.1021
	Orientation	0.0134	0.0824
	Joints	0.0212	0.1890
STA	Position	0.0180	0.1202
	Orientation	0.0232	0.1424
	Joints	0.0240	0.1639
GSTA	Position	0.0069	0.0390
	Orientation	0.0160	0.0834
	Joints	0.0187	0.1073
PID	Position	0.0096	0.0566
	Orientation	0.0140	0.0675
	Joints	0.0158	0.1150

Table 6.8: Experiments: Comparison of the control inputs for the inspection case

Algorithm	**RMS**			
	τ_{thr}	$\dot{\tau}_{thr}$	τ_q	$\dot{\tau}_q$
AGSTA	3.3061	4.2939	2.4505	0.1900
STA	3.6476	6.2583	2.4103	0.1964
GSTA	3.2591	1.3542	2.6514	0.8800
PID	3.3327	2.6029	2.6013	0.8044



(a) Tracking errors position

(b) Tracking errors orientation

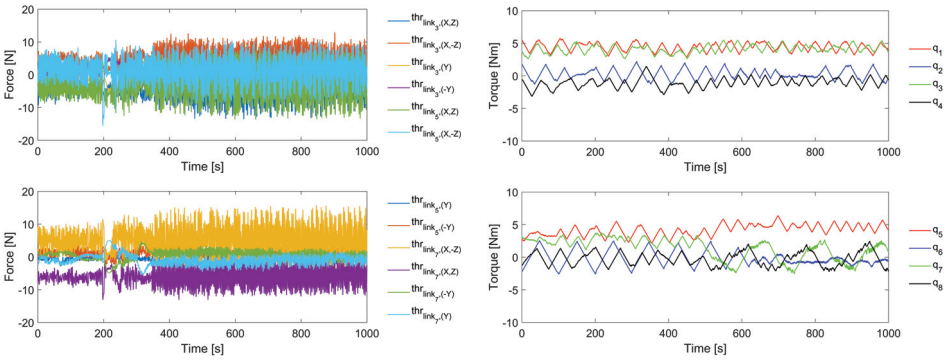
(c) Tracking errors joints q_1-q_4

(d) Tracking errors joints q_5-q_8

Figure 6.10: Experiments: Tracking errors for the inspection case

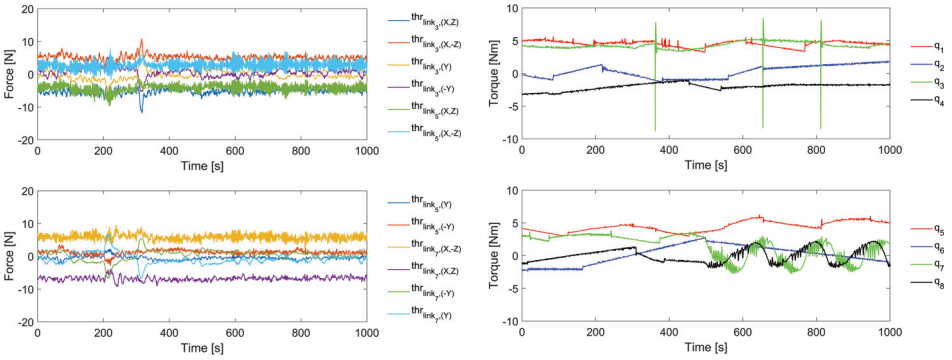
Discussion

From the adaptive GSTA experimental results in Figure 6.9 we can see that the desired trajectories for the position, orientation and the joint angles are tracked very well. From Figure 6.9a we see that the desired position trajectories are almost perfectly tracked, and that the desired orientation trajectories are also perfectly tracked except for yaw, where there are some small oscillations around 200 s. For the joint angles we can see from Figure 6.9b that the joints do deviate from the desired joint angles trajectories in a square oscillation pattern. These oscillations



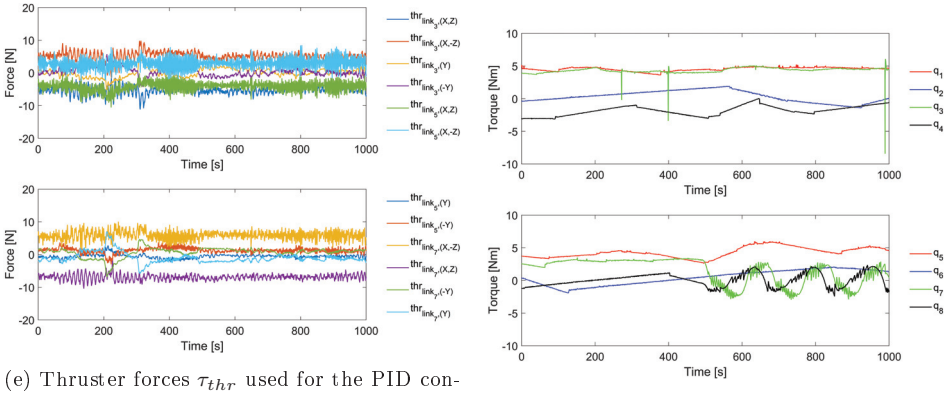
(a) Thruster forces τ_{thr} used for STA with adaptive gains

(b) Joint torques τ_q used for the STA with adaptive gains



(c) Thruster forces τ_{thr} used for the GSTA

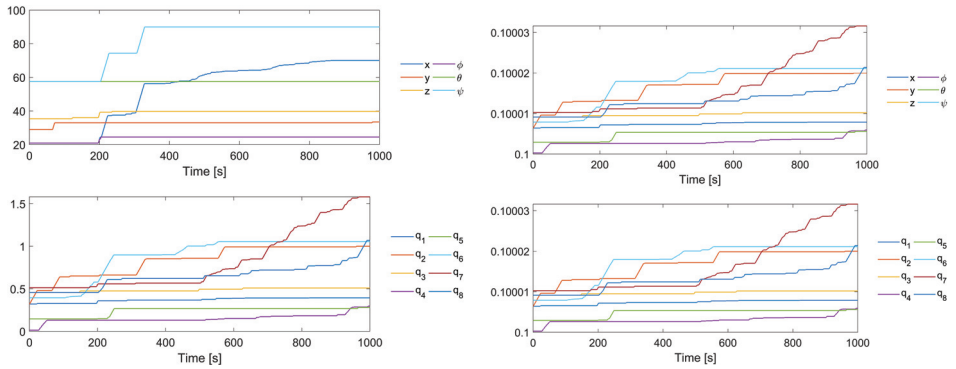
(d) Joint torques τ_q used for the GSTA



(e) Thruster forces τ_{thr} used for the PID controller

(f) Joint torques τ_q used for the PID controller

Figure 6.11: Experiments: Thruster forces τ_{thr} and joint torques τ_q used for the inspection case



(a) Evolution of $\alpha(t)$ over time

(b) Evolution of $\beta(t)$ over time

Figure 6.12: Experiments: Evolution of the adaptive gains over time for the STA with adaptive gains for the inspection case

may occur because of a dead-zone in the joint control or a time-delay between the set and the actual joint torques. Specifically, the adaptive GSTA will try to make the joint angle errors smaller, but with dead-zones and time delays the joints will not be able give the torque the adaptive GSTA calculates momentarily, and the joints therefore overshoot as the resulting torque applied becomes too high. From Figure 6.9c, we can see that the forces used are well below 60 N, which is approximately the limit of the thrusters in the 2020 version of the robot. We do see some tendencies of chattering, but the chattering is not as bad as it looks in the figure, since the figure has a small width compared to the time scale. From Figure 6.9d, we can see that the joint torques are smooth and are below 16 Nm, which is the limit on the joint torques. From Figure 6.9e and Figure 6.9f, we can also see that the gains $k_1(t)$ and $k_2(t)$ increase linearly and converge to suitable values, which agrees with the theoretical results we found in Section 6.1.

By comparing the simulation results shown in Figure 6.5 with the experimental results shown in Figure 6.9, we can see that we have slightly worse tracking in yaw because of the small oscillations around 200 s, that appear in the experiments. We can also see that the tracking of the joints is significantly worse. This is probably due to the joint dynamics that is not modelled in the simulator. In the simulator we get the torque that we calculate momentarily, and we can apply also small torque values, i.e. we do not have a dead-zone. If we compare Table 6.5 with Table 6.7, we see that both the RMSEs and the maximum errors are better for the simulation, but that is to be expected because of measurement noise, thruster dynamics, joint dynamics and other unmodelled dynamics that are inherent in experiments. If we compare the force used by comparing Figure 6.5c with Figure 6.9c and Figure 6.5d with Figure 6.9d, we see that significantly less force was used in the simulations, and this is also confirmed by comparing Table 6.6 with Table 6.8. Nevertheless, the experimental results and the simulation results are quite similar, indicating that the adaptive GSTA is indeed applicable for controlling the AIAUV.

From Figure 6.10a and Figure 6.10b, it is very difficult to see which of the proposed algorithms gives the smallest error, as there are some chattering tendencies. From Figure 6.10c and Figure 6.10d, we can see that the square oscillating tendencies are mostly present for the two adaptive algorithms. This may be because the two adaptive algorithms try to get closer to the desired joint angles trajectories, but because of a probable dead-zone in the joints or time-delay for the joint torques, end up using too much force and therefore overshoots. It can therefore be seen that the GSTA and the PID controller give smaller tracking errors for the joint angles. From Table 6.7 we can see that it is actually the GSTA that gives the best RMSE value for position, while the adaptive GSTA and the PID controller give the same value. For orientation it is the adaptive GSTA that gives the best RMSE value. If we then add the RMSE value together for the adaptive GSTA, the GSTA and the PID controller, we do actually see that the adaptive GSTA and the GSTA get the same value, and the PID controller gets a larger value. We could therefore say that the adaptive GSTA and the GSTA give equally good tracking. For the joint angles however, the GSTA and the PID controller give better RMSE values than the adaptive algorithms. The GSTA is therefore actually the algorithm that gives the overall best tracking. If we look at the maximum error values, we see that the GSTA gives the best (lowest) value for position, and that the PID controller gives the best value for orientation, while the adaptive GSTA actually is third and second, respectively. For the joint angles the GSTA and the PID controller give the best values also for the maximum error. However, if we look at the error plots in Figure 6.10 we do see that there are some outliers for especially the adaptive GSTA in the position errors, and for all the algorithms for the joint angles. The maximum error does therefore not give a fair comparison. These outliers may also have affected the RMSE values, which can be a reason for the higher RMSE for the adaptive GSTA in position. If we take the thruster use of the different algorithms into consideration, i.e. Figure 6.9c, Figure 6.9d and Figure 6.11, we see that there is not such a large difference in the amount of force used. If we look at Table 6.8 we can confirm by looking at the RMS value of the thruster forces and joint torques that the difference in the amount of force used is not that large. From the RMS value of $\dot{\tau}_{thr}$ and $\dot{\tau}_q$, we can see that the adaptive algorithms introduce some chattering into the thruster forces. However, here the RMS values are so small that the chattering that is introduced are negligible.

By comparing Figure 6.6 with Figure 6.10, we can see that overall, the tracking errors are smaller in the simulations, which can also be confirmed by comparing Table 6.5 with Table 6.7. This is to be expected because of measurement noise, thruster dynamics, joint dynamics and other unmodelled dynamics that unavoidably affect the experiments. When we compare the forces used, by comparing Figure 6.5c, Figure 6.5d and Figure 6.7 with Figure 6.9c, Figure 6.9d and Figure 6.11, we can see that during simulations we use much less force than during experiments, which can also be confirmed by comparing Table 6.6 with Table 6.8. This may be because our simulation model is completely neutrally buoyant, while the 2020 version of the robot is slightly negatively buoyant.

One thing worth noticing is that the results on which algorithms that perform the best out of the STA with adaptive gains, the GSTA and the PID controller are quite different from the results previously shown in Chapter 5. If we compare

Table 6.5 with Table 6.7 and Table 6.6 with Table 6.8 it is, however, quite obvious that this is connected with the amount of force used, as in Chapter 5 it was the STA with adaptive gains which used most force and got the best tracking results. This may mean that we were able to tune the GSTA and the PID controller better this time, and that therefore the adaptive gains did not have such a large impact on the results. We could possibly also have been able to get better results with the STA with adaptive gains if we had chosen a better starting point for the adaptive gains. It is, however, easier to tune the adaptive controllers, and get good results without extensive tuning efforts.

6.4 Chapter summary

A novel adaptive GSTA is proposed. The proposed approach consists of using dynamically adapted control gains to ensure GFT convergence in the presence of time- and state-dependent perturbations and uncertain control coefficients, such that no conservative upper bound on the uncertainties is considered. It is also shown that the proposed adaptive GSTA causes the tracking errors of the AIAUV to converge asymptotically to zero, and a simulation and experimental study is performed which shows the effectiveness of the proposed adaptive GSTA. The simulation and experimental results validate and verify that the proposed approach is well suited for controlling an AIAUV. The results are almost equally good between the simulations and experiments. The tracking errors are larger in the experiments than in the simulations; however, this is to be expected because of measurement noise, the thruster dynamics, the joint dynamics and other unmodelled dynamics that inevitably affect the experiment.

We also present a comparison with the STA with adaptive gains, the GSTA and a PID controller. In simulations we saw that the adaptive GSTA clearly gave the best results. The experimental results were, however, not so clear, because the GSTA actually ended up being the algorithm that gave the overall best tracking performance, mostly because of the poor performance in tracking of the desired joint angle trajectories when using the adaptive GSTA. This means that the choice of starting point for the adaptive gains on the joints may have been poor. We also saw that the results were different than in Chapter 5, which may suggest that we were able to tune the GSTA and the PID controller very well in the latest experiments. It is, however, much easier to tune the adaptive controllers and get good results without extensive tuning efforts.

Chapter 7

Combined Kinematic and Dynamic Control

In this chapter, we propose a combined kinematic and dynamic control approach for VMSs. The main idea is to combine the SRMTP framework [20] with a robust SMC while simultaneously ensuring that the task errors remain bounded. The kinematic stability analysis of the SRMTP method is based on the results in [3]. *Any* SMC can be used as long as it is able to make the velocity vector converge to the velocity reference vector in finite time. The reference vector is chosen as the output from the SRMTP inverse kinematic controller. This novel approach allows us to analyse the stability properties of the kinematic and dynamic subsystems together in the presence of model uncertainty while retaining the possibility of solving multiple tasks simultaneously. The finite-time convergence property of the SMC allows us to show that the multiple set-point regulation tasks will converge asymptotically to zero without the strict requirement that the velocities are perfectly controlled. The proposed control approach and the corresponding stability analysis, thus avoids the assumption of perfect dynamic control that is common in kinematic stability analyses for vehicle manipulators.

Contributions of this chapter

The contributions of this chapter can be summarized as follows. A novel combined kinematic and dynamic control approach is proposed for VMSs. The main idea is to combine the SRMTP framework with a robust SMC while simultaneously ensuring that the task errors remain bounded. This novel approach allows us to analyse the stability properties of the kinematic and dynamic subsystems together in the presence of model uncertainty while retaining the possibility of solving multiple tasks simultaneously. We present two examples of robust SMC that satisfy the required condition of making the velocity vector converge to the velocity reference vector in finite time. A simulation and experimental study is performed to demonstrate the effectiveness of the proposed method.

Organization of this chapter

The remainder of this chapter is structured as follows. The mathematical model of the VMS is presented in Section 7.1, and a brief introduction to inverse kinematic control and the SRMTP method is provided in Section 7.2. The main result with the corresponding stability analysis is described in Section 7.3. The simulation study is presented in Section 7.4 and the experimental study is presented in Section 7.5. In Section 7.6 a chapter summary is presented.

Publications

This chapter is based on [11], [13] and [14].

7.1 Vehicle-manipulator model

The model considered in this chapter is presented in this section. The dynamics of a large class of systems can be described by the vehicle-manipulator model [32], given by (2.1) and (2.11), where $D(q,\zeta)$ is only for floating and submerged vehicles. The dynamic model in (2.1) and (2.11), can be formulated with respect to a coordinate frame having its origin at an arbitrary position on the vehicle manipulator. In this chapter, we assume that the model is formulated with respect to an arbitrary link of the vehicle manipulator such that the velocity state vector, ζ, is defined as

$$\zeta = \begin{bmatrix} v_{Ia}^a \\ \omega_{Ia}^a \\ \dot{q} \end{bmatrix} \quad \in \mathbb{R}^{6+(n-1)} \tag{7.1}$$

where v_{Ia}^a and ω_{Ia}^a are the body-fixed linear and angular velocities of the chosen link of the vehicle manipulator, respectively, and $\dot{q}$ is the vector of joint velocities.

Remark 7.1. The formulation in (2.1) and (2.11), preserves the following important properties [32]:

1. $M = M^T > 0$
2. $x^T(\dot{M} - 2C)x = 0 \quad \forall \quad x \in \mathbb{R}^{6+(n-1)}$
3. $x^T D x > 0 \quad \forall \quad x \in \mathbb{R}^{6+(n-1)}$

7.2 Inverse kinematic control

Robot manipulators are designed to perform specific tasks associated with either the internal configuration of the robot (joint/configuration space) or the external configuration with respect to the environment (operational/task space). A task that specifies a desired position and/or orientation of the end effector of the robot is an example of a common operational space task. A comprehensive collection of possible tasks for underwater robotic vehicles is presented in [4]. An m-dimensional task can be described by the task variable, $\chi_i(t) \in \mathbb{R}^m$, defined as

$$\chi_i(t) = f(\xi_e(t)), \tag{7.2}$$

where $\xi_e(t)$ is the vector of generalized coordinates describing the configuration of the vehicle manipulator defined in Section 7.1. The function $f(\cdot)$ maps the configuration into the task space coordinates. The task variable $\chi_i(t)$, the generalized coordinates $\xi_e(t)$ and the system velocities $\zeta(t)$ have the following important differential relationship:

$$\dot{\chi}_i(t) = \frac{\partial f(\xi_e(t))}{\partial \xi_e}\dot{\xi}_e(t) = \frac{\partial f(\xi_e(t))}{\partial \xi_e}J(\Theta)\zeta(t) = J_i(\xi_e(t))\zeta(t) \tag{7.3}$$

where $J_i(\xi_e(t)) \in \mathbb{R}^{m\times(6+(n-1))}$ is the configuration-dependent task Jacobian matrix. Let $\chi_{i,d}(t) \in \mathbb{R}^m$ be the desired trajectory for the task variable $\chi_i(t)$ and define the task error $\tilde{\chi}_i \in \mathbb{R}^m$ as

$$\tilde{\chi}_i = \chi_{i,d} - \chi_i. \tag{7.4}$$

To determine the motion required to achieve convergence of the task error $\tilde{\chi}_i$ to zero, it is common to use the closed-loop inverse kinematic (CLIK) routine expressed as

$$\zeta_r = J_i^+(\dot{\chi}_{i,d} + \Lambda_i\tilde{\chi}_i), \tag{7.5}$$

where $J_i^+ = J_i^T(J_iJ_i^T)^{-1}$ is the right Moore-Penrose pseudo-inverse of the task Jacobian, ζ_r is the reference velocity vector, and $\Lambda_i > 0$ is a positive-definite gain matrix. The single-task approach can be extended to multiple simultaneous tasks using the SRMTP method [20]:

$$\begin{aligned}\zeta_r = {} & J_1^+\left(\dot{\chi}_{1,d} + \Lambda_1\tilde{\chi}_1\right) + N_1J_2^+\left(\dot{\chi}_{2,d} + \Lambda_2\tilde{\chi}_2\right) + \ldots \\ & + N_{12..(k-1)}J_k^+\left(\dot{\chi}_{k,d} + \Lambda_k\tilde{\chi}_k\right),\end{aligned} \tag{7.6}$$

where the null space projectors of the task Jacobians are given by $N_i = (I - J_i^+J_i)$, and $N_{12..(k-1)} = \text{Null}(\left[J_1^T, J_2^T, \ldots, J_{k-1}^T\right]^T)$ represents the combined null space projector of tasks 1 through $k-1$. The null space projector matrices ensure that conflicting velocity components generated by the lower-priority tasks are filtered out such that these do not affect the satisfaction of the higher-priority tasks. In [3], it is shown that all the task errors will converge to zero, provided that $\zeta(t) = \zeta_r(t)\,\forall\, t \geq 0$ (i.e., neglecting the dynamics), that the tasks are compatible and specified as time-independent regulation tasks (i.e., $\dot{\chi}_{i,d} = 0$) and that the task gains Λ_i are chosen appropriately.

In the next section, we extend the kinematic stability analysis in [3] by combining it with a dynamic control law for the model described in Section 7.1, and we show that all the regulation task errors will converge asymptotically to zero without the strict requirement that $\zeta(t) = \zeta_r(t)\,\forall\, t \geq 0$.

7.3 Combined kinematic and dynamic stability analysis

In this section, we propose a combined kinematic and dynamic control law for the model presented in Section 7.1 performing an arbitrary number of regulation tasks. By combining the SRMTP method with an SMC that is FTS, we show, by extending the Lyapunov analysis for regulation tasks presented in [3] to also include the

system dynamics, that all the regulation task errors will converge asymptotically to zero without the strict requirement that $\zeta(t) = \zeta_r(t) \forall t \geq 0$. This novel approach then avoids the assumption of perfect dynamic control that is common in kinematic stability analyses for vehicle manipulators. We then provide two examples of SMC algorithms that satisfy the requirement of being FTS.

To be able to use SMC, we must first define a sliding surface. Define the sliding surface as

$$\sigma \triangleq \zeta - \zeta_r. \tag{7.7}$$

where $\zeta_r = \begin{bmatrix} (v^a_{Ia,r})^T & (\omega^a_{Ia,r})^T & \dot{q}_r^T \end{bmatrix}^T \in \mathbb{R}^{6+(n-1)}$ is the reference velocity vector given as the output of the SRMTP inverse kinematics routine in (7.6) for regulation tasks (i.e., time-independent tasks). By differentiating (7.7) and inserting (2.1), we obtain

$$\dot{\sigma} = \dot{\zeta} - \dot{\zeta}_r = M^{-1}(-C\zeta - D\zeta - g + \tau) - \dot{\zeta}_r \tag{7.8}$$

and by using that $\zeta = \zeta_r + \sigma$ from (7.7), we obtain the following equation describing the dynamics of σ:

$$\dot{\sigma} = M^{-1}(-C(\zeta_r + \sigma) - D(\zeta_r + \sigma) - g + \tau) - \dot{\zeta}_r. \tag{7.9}$$

Now, define $\tilde{\chi}$ as a vector of all the regulation task errors, i.e.,

$$\tilde{\chi} = \begin{bmatrix} \tilde{\chi}_1^T, \tilde{\chi}_2^T, \ldots, \tilde{\chi}_k^T \end{bmatrix}^T. \tag{7.10}$$

By taking the time derivative of (7.10) and using (7.3), $\dot{\chi}_{i,d} = 0$ and $\zeta = \zeta_r + \sigma$ from (7.7), we have

$$\dot{\tilde{\chi}} = -\dot{\chi} = -J(\xi_e)\zeta = -J(\xi_e)(\zeta_r + \sigma), \tag{7.11}$$

where $J(\xi_e) = \begin{bmatrix} J_1^T, J_2^T, \ldots, J_k^T \end{bmatrix}^T$ is a matrix that contains the corresponding Jacobian matrices for the tasks. We make the following assumption about $J(\xi_e)$:

Assumption 7.1. $J(\xi_e)$ in (7.11) is bounded.

Remark 7.2. Note that the matrix $J(\xi_e)$ will be bounded as long as singularities in the task representations are avoided. This is also necessary for the complete control method to be singularity-free.

The total error dynamics can then be obtained by combining (7.8) and (7.11):

$$\dot{\tilde{\chi}} = -J(\xi_e)(\zeta_r + \sigma) \tag{7.12a}$$

$$\dot{\sigma} = M^{-1}(-C(\zeta_r + \sigma) - D(\zeta_r + \sigma) - g + \tau) - \dot{\zeta}_r \tag{7.12b}$$

Theorem 7.1. *Assume that Assumption 7.1 is satisfied. Let the control input of (2.1) be given by the SMC law*

$$\tau = u_{\text{SMC}}(\sigma), \tag{7.13}$$

*where σ is the sliding surface in (7.7) and u_{SMC} is **any** SMC that stabilizes the sliding surface $\sigma = 0$ in finite time. The sliding surface $\sigma = 0$ is then an FTS equilibrium of (7.12b), which ensures asymptotic convergence of the regulation task errors, i.e.,*

$$\lim_{t \to \infty} \tilde{\chi}(t) = 0.$$

Proof. In [3], it is shown that if the generalized velocities of the vehicle-manipulator follow the reference velocities, i.e., $\zeta = \zeta_r$, then the regulation task errors asymptotically converge to zero, i.e., $\lim_{t\to\infty} \tilde{\chi}(t) = 0$. Here, we will extend this analysis to include the dynamic control part of the problem. The closed-loop dynamics (7.12b), (7.13) is FTS by assumption, which means that $\zeta = \zeta_r$ after some finite time T. This also implies that $||\sigma|| \leq \delta_1 \forall t \geq 0$, where δ_1 is a positive constant. Now, rather than assuming that $\zeta = \zeta_r$ as done in [3], we take into account that $\zeta = \zeta_r + \sigma$.

We will now analyse the stability properties of the origin of (7.12a) using the Lyapunov function candidate (LFC) $V(\tilde{\chi}) = \frac{1}{2}\tilde{\chi}^T\tilde{\chi}$. The derivative of the LFC is, by (7.12a), as follows:

$$\dot{V} = \tilde{\chi}^T\dot{\tilde{\chi}} = -\tilde{\chi}^T J(\xi_e)(\zeta_r + \sigma) \tag{7.14}$$

From [3], we have that

$$\tilde{\chi}^T J(\xi_e)\zeta_r = \tilde{\chi}^T P(\xi_e)\tilde{\chi}, \tag{7.15}$$

where $P(\xi)$ is defined as

$$P(\xi_e) = \begin{bmatrix} \Lambda_1 & O_{m_1,m_2} & \dots & O_{m_1,m_k} \\ J_2 J_1^+ \Lambda_1 & J_2 N_1 J_2^+ \Lambda_2 & \dots & J_2 \bar{N} J_k^+ \Lambda_k \\ \dots & \dots & \dots & \dots \\ J_{k-1} J_1^+ \Lambda_1 & J_{k-1} N_1 J_2^+ \Lambda_2 & \dots & J_{k-1} \bar{N} J_k^+ \Lambda_k \\ J_k J_1^+ \Lambda_1 & J_k N_1 J_2^+ \Lambda_2 & \dots & J_k \bar{N} J_k^+ \Lambda_k \end{bmatrix} \tag{7.16}$$

where $\bar{N} = N_{12..(k-1)}$. Note that P is positive definite [3]. By using (7.15), we can write the LFC derivative as

$$\dot{V} = -\tilde{\chi}^T P(\xi_e)\tilde{\chi} - \tilde{\chi}^T J(\xi_e)\sigma \tag{7.17}$$

Since, as stated above, we have that $||\sigma(t)|| \leq \delta_1 \ \forall t \geq 0$, we can rewrite (7.17) as follows:

$$\begin{aligned} \dot{V} &\leq -\lambda_{\min}(P)||\tilde{\chi}||^2 + \delta_1||J(\xi_e)||||\tilde{\chi}|| \\ &= -\lambda_{\min}(P)||\tilde{\chi}||^2 + \theta||\tilde{\chi}||^2 - \theta||\tilde{\chi}||^2 + \delta_1||J(\xi_e)||||\tilde{\chi}|| \\ &\leq -(\lambda_{\min}(P) - \theta)||\tilde{\chi}||^2 \quad \forall \quad ||\tilde{\chi}|| \geq \frac{\delta_1||J(\xi_e)||}{\theta} \end{aligned} \tag{7.18}$$

where $0 < \theta < \lambda_{\min}(P)$, since P is positive definite. By Assumption 7.1, $J(\xi_e)$ is bounded; thus, the conditions of Theorem A.5 are satisfied. We can then conclude that the solutions are globally uniformly ultimately bounded (Definition A.9), which means that $||\tilde{\chi}(t)|| \leq \delta_2 \forall t \geq 0$, where δ_2 is a positive constant. Consequently, the regulation task errors will not escape to infinity when $\sigma \neq 0$. Now, after a finite time T, the sliding surface $\sigma = 0$ will be reached. Once the system trajectories are confined to $\sigma = 0$, the error dynamics are given by

$$\dot{\tilde{\chi}} = -J(\xi_e)\zeta \tag{7.19}$$

and the origin of (7.19) is asymptotically stable and the regulation task errors will thus asymptotically converge to zero, i.e., $\lim_{t\to\infty} \tilde{\chi}(t) = 0$ [3]. □

Remark 7.3. This proof can be extended to include trajectory tracking tasks for link i by including $J_{g,i}^{+}(\zeta_{Ii,d}^{i}(t) - \Lambda(\xi_{e,i}(t) - \xi_{e,i,d}(t)))$, where $J_{g,i}^{+}$ is the geometric Jacobian, which maps the velocity state vector ζ to the linear and angular velocities of link i, $\zeta_{Ii,d}^{i}$ is the desired body-fixed velocity of the link frame of link i, Λ is a gain matrix, and $\xi_{e,1}$ and $\xi_{e,i,d}$ is the position and the desired position, respectively, of link i, as the primary task in (7). When the sliding surface and the tasks errors have gone to zero, i.e., $\zeta - \zeta_r = 0$, $J_{g,i}\zeta(t) = \zeta_{Ii,d}^{i}(t) - \Lambda(\xi_{e,i}(t) - \xi_{e,i,d}(t))$ remain, which will make $\xi_{e,i}(t) - \xi_{e,i,d}(t)$ converge to zero if Λ is chosen correctly and the tasks are orthogonal. Orthogonal tasks are a strict requirement, which in practice means that the tasks utilise separate degrees of freedom of the vehicle manipulator.

7.3.1 First-order SMC

In this section, we will show that a first-order SMC stabilizes the sliding surface $\sigma = 0$, where σ is given by (7.7), in finite time. Part of the analysis of the chosen first-order SMC is based on [5], which considers a set-point regulation problem for a UVMS without taking the inverse kinematic problem into account. In this chapter, we modify the analysis in [5] to show the finite-time convergence of the velocity controller.

Theorem 7.2. *Let the control input u_{SMC} in (7.13) be given by*

$$\tau = \hat{g}(q, R_{Ib}) - K_d\sigma - K\,\text{sgn}(\sigma), \tag{7.20}$$

where $K_d > 0$, $\hat{g}(q, R_{Ib})$ represents the estimate of the gravity/buoyancy forces and moments, and

$$\text{sgn}(\sigma_i) = \begin{cases} 1, & \text{for } \sigma_i \geq 0 \\ -1, & \text{for } \sigma_i < 0 \end{cases}$$

The gain K is chosen as

$$K \geq \kappa + K_0, \tag{7.21}$$

where $K_0 > 0$ and κ is an upper bound chosen to satisfy

$$\kappa \geq ||C(q,\zeta)\zeta_r + D(q,\zeta)\zeta_r + \tilde{g}(q, R_{Ib}) + M(q)\dot{\zeta}_r||. \tag{7.22}$$

The sliding surface σ in (7.7) of the system (7.12b), (7.20) will then converge exponentially and in finite time to the sliding surface $\sigma = 0$.

Proof. Consider the LFC for the system (7.12b), (7.20)

$$V_\sigma = \frac{1}{2}\sigma^T M\sigma > 0, \quad \forall\sigma \neq 0, \tag{7.23}$$

which is positive definite as $M(q) > 0$. Differentiating (7.23) and inserting (7.7) and (7.12b) yields

$$\begin{aligned} \dot{V}_\sigma &= \sigma^T M\dot{\sigma} + \frac{1}{2}\sigma^T\dot{M}\sigma \\ &= \sigma^T\left[\tau - C\zeta - D\zeta - g - M\dot{\zeta}_r\right] + \frac{1}{2}\sigma^T\dot{M}\sigma \\ &= -\sigma^T D\sigma + \sigma^T\left[\tau - C\zeta_r - D\zeta_r - g - M\dot{\zeta}_r\right] + \frac{1}{2}\sigma^T\left[\dot{M} - 2C\right]\sigma. \end{aligned} \tag{7.24}$$

The last term in (7.24) is equal to zero because $\dot{M} - 2C$ is skew symmetric, cf., Remark 7.1. Inserting (7.20) yields

$$\begin{aligned}\dot{V}_\sigma =& -\sigma^T(K_d + D)\sigma - \sigma^T\left[K\,\mathrm{sgn}(\sigma)\right] + \sigma^T\left[-C\zeta_r - D\zeta_r - \tilde{g} - M\dot{\zeta}_r\right] \\ \leq& -\sigma^T(K_d + D)\sigma - K||\sigma|| + ||C\zeta_r + D\zeta_r + \tilde{g} + M\dot{\zeta}_r||||\sigma||,\end{aligned} \tag{7.25}$$

where we have used the Cauchy-Schwarz inequality. Using (7.21), (7.22), we have

$$\dot{V}_\sigma \leq -\sigma^T(K_d + D)\sigma - K_0||\sigma|| < 0, \quad \forall\sigma \neq 0. \tag{7.26}$$

The first term in (7.26) ensures exponential convergence towards the sliding surface $\sigma = 0$, while the second term ensures, by use of the comparison lemma (Lemma A.8), that σ reaches the sliding surface in finite time. Thus, we have exponential and finite-time convergence to the sliding surface. □

Remark 7.4. To determine κ, we need to find an upper bound on the expression in (7.22). This upper bound depends on the size of the task errors. To avoid κ, and thus K, becoming undesirably large, a reference model [29] can be used to obtain sufficiently smooth desired trajectories and avoid large jumps in the task errors when changing set-points. Such a reference model can also include saturating elements to limit the desired velocities.

Remark 7.5. The above analysis assumes ideal actuators. In practice, time delays and imperfections in the actuators will cause high-frequency chattering when SMC is applied. To eliminate the chattering problem, the discontinuous signum function in (7.20) is typically replaced by a high-slope saturation function. In this case, we achieve the ultimate boundedness of the task errors. For set-point regulation problems, integral action can be introduced in the control law to achieve zero steady-state error [40].

7.3.2 Generalized super-twisting with adaptive gains

In this section, we will show that the GSTA with adaptive gains stabilizes the sliding surface $\sigma = 0$, where σ is given by (7.7), in finite time. In Chapter 6, the GSTA with adaptive gains was proposed and employed for trajectory tracking for an AIAUV, and the tracking error for the AIAUV was shown to asymptotically converge to zero. In this chapter, a different sliding surface is employed. We show that the adaptive GSTA can also be applied to stabilise the sliding surface σ in (7.7) in finite time.

Theorem 7.3. *Let the control input u_{SMC} in (7.13) be given by the GSTA with adaptive gains defined in (6.7)-(6.9). The sliding surface σ in (7.7) of the system (7.12b), (7.13), (6.7) will then converge in finite time to the sliding surface $\sigma = 0$.*

Proof. The dynamics of σ are described by (7.12b). Now, by introducing $\Phi(\sigma, t) = \Phi_1(\sigma, t) + \Phi_2(\sigma, t)$, where $\Phi_1(0, t) = 0$, $\Gamma(\cdot) = M^{-1}$, and by setting τ as in (6.7), we obtain

$$\dot{\sigma} = -k_1\Gamma(\cdot)\phi_1(\sigma) + \Phi_1(\sigma, t) + \Gamma(\cdot)(z + \Gamma^{-1}(\cdot)\Phi_2(\sigma, t)) \tag{7.27}$$

where $\Phi_1(\sigma,t) = \Gamma(\cdot)(-C\sigma - D\sigma)$ and $\Phi_2(\sigma,t) = \Gamma(\cdot)(-C\zeta_r - D\zeta_r - g - M\dot{\zeta}_r)$. Now, by setting $\sigma_1 = \sigma$ and $\sigma_2 = z + \Gamma^{-1}(\cdot)\Phi_2(\sigma,t)$, we can write the dynamics as

$$\begin{aligned}\dot{\sigma}_1 &= -k_1\Gamma(\cdot)\phi_1(\sigma_1) + \Phi_1(\sigma_1,t) + \Gamma(\cdot)\sigma_2 \\ \dot{\sigma}_2 &= -k_2\phi_2(\sigma_1) + \frac{d}{dt}\big(\Gamma^{-1}(\cdot)\Phi_2(\sigma_1,t)\big)\end{aligned} \tag{7.28}$$

From Section 6.2.1, under Assumptions 6.5-6.9 the dynamics in (7.28) are globally FTS. The sliding surface $\sigma = 0$ is therefore a global FTS surface, which means that σ converges to zero in finite time. □

7.4 Simulation case study

In this section, a case study using an AIAUV is presented to show the effectiveness of the proposed combined kinematic and dynamic controller using the two different SMC algorithms from Section 7.3. To further highlight the advantages of using the SRMTP method combined with an SMC algorithm, we will also show the results in which the SRMTP method is combined with two standard control methods. We will also highlight the robustness the SMC algorithms provide by presenting the results for two different AIAUVs. The AIAUV is a lightweight underwater vehicle that has multiple joints and multiple thrusters [79]. The AIAUV is subject to hydrodynamic and hydrostatic parameter uncertainties, uncertain thruster characteristics, unknown disturbances, unmodelled dynamics and large coupling forces caused by joint motion. It is therefore an example of a vehicle manipulator for which we cannot assume that the reference output is tracked perfectly by the dynamic controller since it has a low mass and the coupling forces caused by the joint motion are large. We use the simulation model presented in Section 2.2 and the Eelume 2016 version of the robot presented in Section 2.2.1.

7.4.1 Simulation tasks

The combined kinematic and dynamic control of the AIAUV is demonstrated using the following three set-point regulation tasks:

1. Control the position and orientation of the centre link (main task)
2. Control the pitch and the yaw angle of the front end of the AIAUV
3. Control the pitch and yaw angle of the back end of the AIAUV

This combination of tasks illustrates the ability of the AIAUV to move to a position of interest and then perform a double observation task by simultaneously adjusting the pitch and the yaw angles of the back end and the front end of the AIAUV.

The expressions for the task error and the task Jacobian for task 1 are as follows:

Task 1 - Position and orientation of the centre link

$$\begin{aligned}\tilde{\chi}_1 &= \left[(\tilde{\eta}^c_{1,Ic})^T, \tilde{\eta}^T_{2,c}\right]^T \\ J_1 &= \begin{bmatrix} I_{6\times 6} & 0_{6\times 8}\end{bmatrix},\end{aligned}$$

where $\tilde{\eta}^c_{1,Ic}$ is the position deviation of the centre link and $\tilde{\eta}_{2,c}$ is the orientation deviation of the centre link. The task Jacobian J_1 for task 1 is the identity matrix

since the task is completely described by the position and orientation of the centre link.

The second task is fulfilled using the two double-joint modules in front of the centre link, which means that the expressions for the task error and the task Jacobian for task 2 are as follows:

Task 2 - Orientation of the front end

$$\tilde{\chi}_2 = \left[\begin{smallmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right] \tilde{\eta}_{2,f}$$
$$J_2 = J_{cf}(\text{row 5-6})$$

where $\tilde{\eta}_{2,f}$ is the orientation deviation of the front end, and J_{cf} is the front end Jacobian. The front end Jacobian relates the body-fixed velocities of the front end to the body-fixed velocity of the centre link and the joint velocities.

The third task utilizes the two double-joint modules behind the centre link, which means that the expressions for the task error and the task Jacobian for task 3 are the following:

Task 3 - Orientation of the back end

$$\tilde{\chi}_3 = \left[\begin{smallmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right] \tilde{\eta}_{2,b}$$
$$J_3 = J_{cb}(\text{row 5-6})$$

where $\tilde{\eta}_{2,b}$ is the orientation deviation of the back end, and J_{cb} is the back end Jacobian. The back end Jacobian relates the body-fixed velocities of the back end to the body-fixed velocity of the centre link and the joint velocities. By using the two double-joint modules in front of the centre link to fulfil the second task and the two double-joint modules behind the centre link to fulfil the third task, the available degrees of freedom are divided between the tasks. Although unnecessary, this ensures that the tasks are compatible such that all tasks can be fulfilled simultaneously.

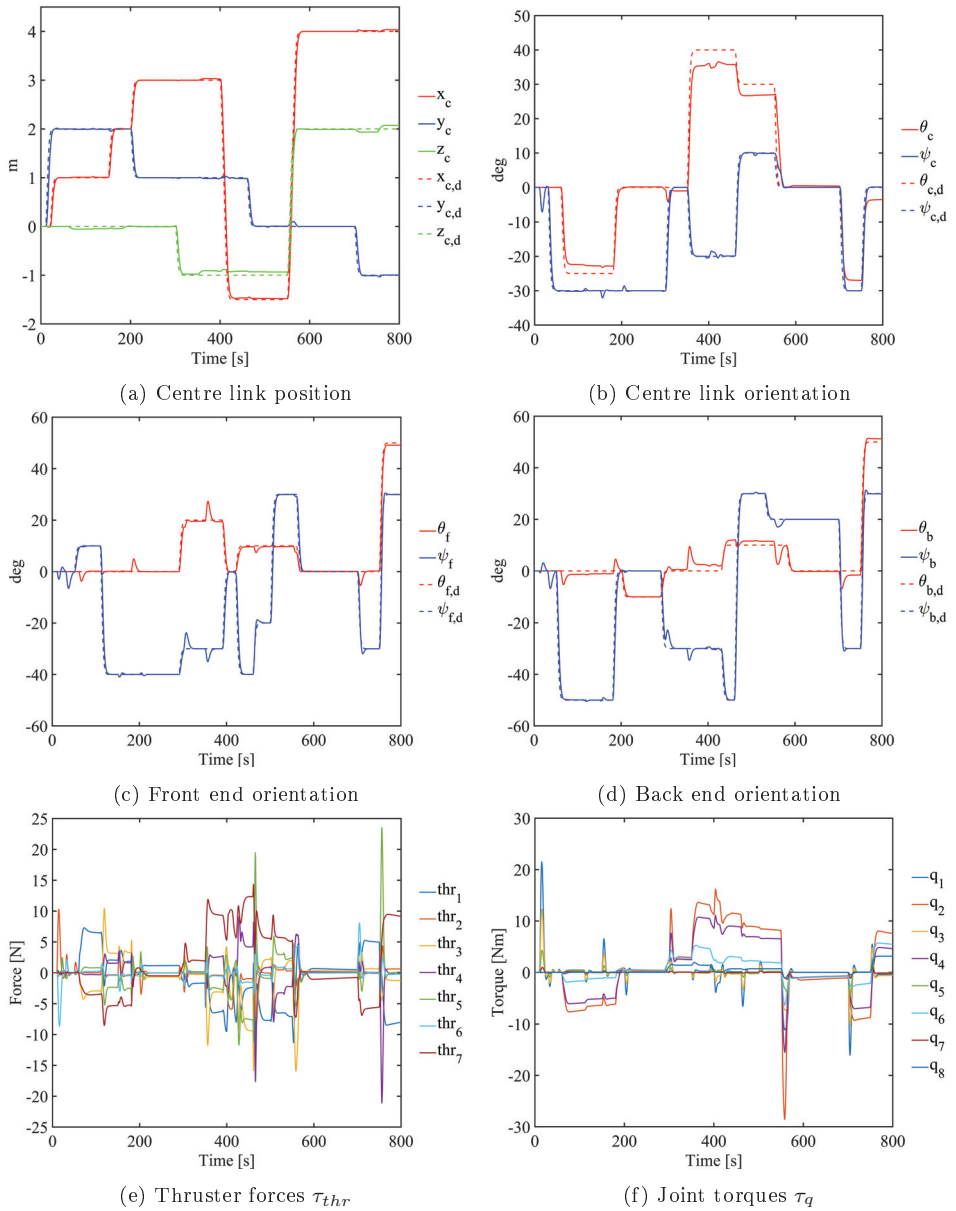
The reference velocities, ζ_r, are calculated according to

$$\zeta_r = J_1^+ \Lambda_1 \tilde{\chi}_1 + N_1 J_2^+ \Lambda_2 \tilde{\chi}_2 + N_1 J_3^+ \Lambda_3 \tilde{\chi}_3 \tag{7.29}$$

Task 2 and task 3 are always compatible, which allows us to multiply task 3 only with the null space projector of task 1 N_1. The set-points $\chi_{i,d}$ are manually controlled and filtered through a third-order reference filter to avoid discontinuities and large jumps in the calculated reference velocities. The gain parameters in (7.29) are set to $\Lambda_1 = \text{diag}(0.5, 0.5, 0.5, 0.1, 0.5, 0.5)$, $\Lambda_2 = 0.5 I_{2\times 2}$, and $\Lambda_3 = 0.5 I_{2\times 2}$.

7.4.2 Results using the AIAUV described by Table 2.2

In this section, the results using the AIAUV described by Table 2.2 will be used. Figure 7.1 shows the results when the first-order SMC (7.20) is used, with the control gains chosen as $K = \text{diag}(50e_6, 75e_8)$ and $K_d = \text{diag}(30e_6, 45e_8)$, where e_i is a $1 \times i$ vector of ones. Note that the sgn function in (7.20) is replaced by a high-slope saturation function with a boundary layer to avoid chattering in the control input, with a width of $[0.07e_{14}]$. Figure 7.2 shows the results when the GSTA with adaptive gains (6.7) is used, with the control gains chosen as $\beta = [80e_{14}]^T$, $\varepsilon = [10^{-5}e_{14}]^T$, $\lambda = [0.1e_6, 10e_8]^T$, $\gamma_1 = [e_{14}]^T$ and $\omega_1 = [e_{14}]^T$. For the GSTA



(a) Centre link position

(b) Centre link orientation

(c) Front end orientation

(d) Back end orientation

(e) Thruster forces τ_{thr}

(f) Joint torques τ_q

Figure 7.1: Results when the first-order SMC is used for the AIAUV described by Table 2.2

with adaptive gains, a small boundary is placed on σ for implementation purposes. The adaptive gains are then represented by (6.51) where the design parameter $\alpha_m = [0.005e_6, 5 \cdot 10^{-5}e_8]^T$.

To further highlight the advantages of using the SRMTP method combined with an SMC we will also show results where the SRMTP method is combined with standard control methods. We will show results with a PID controller and a standard feedback linearisation controller. The PID controller used is defined as

$$\tau = -k_p\tilde{\zeta} + k_d\dot{\zeta}_r - k_i \int \tilde{\zeta}dt \tag{7.30}$$

where $\tilde{\zeta} = \zeta - \zeta_r$ and k_p, k_d and k_i are controller gains. The reason we chose to include a PID controller as a standard control method is because it is one of the most widely used controllers and is known for obtaining good results if tuned correctly. However, it does not provide any stability guaranties and is difficult to tune for optimal performance. We therefore also include results using a feedback linearisation controller. The feedback linearisation controller gives stability guaranties and is easy to tune as long as the model parameters are known. The feedback linearisation controller used is defined as

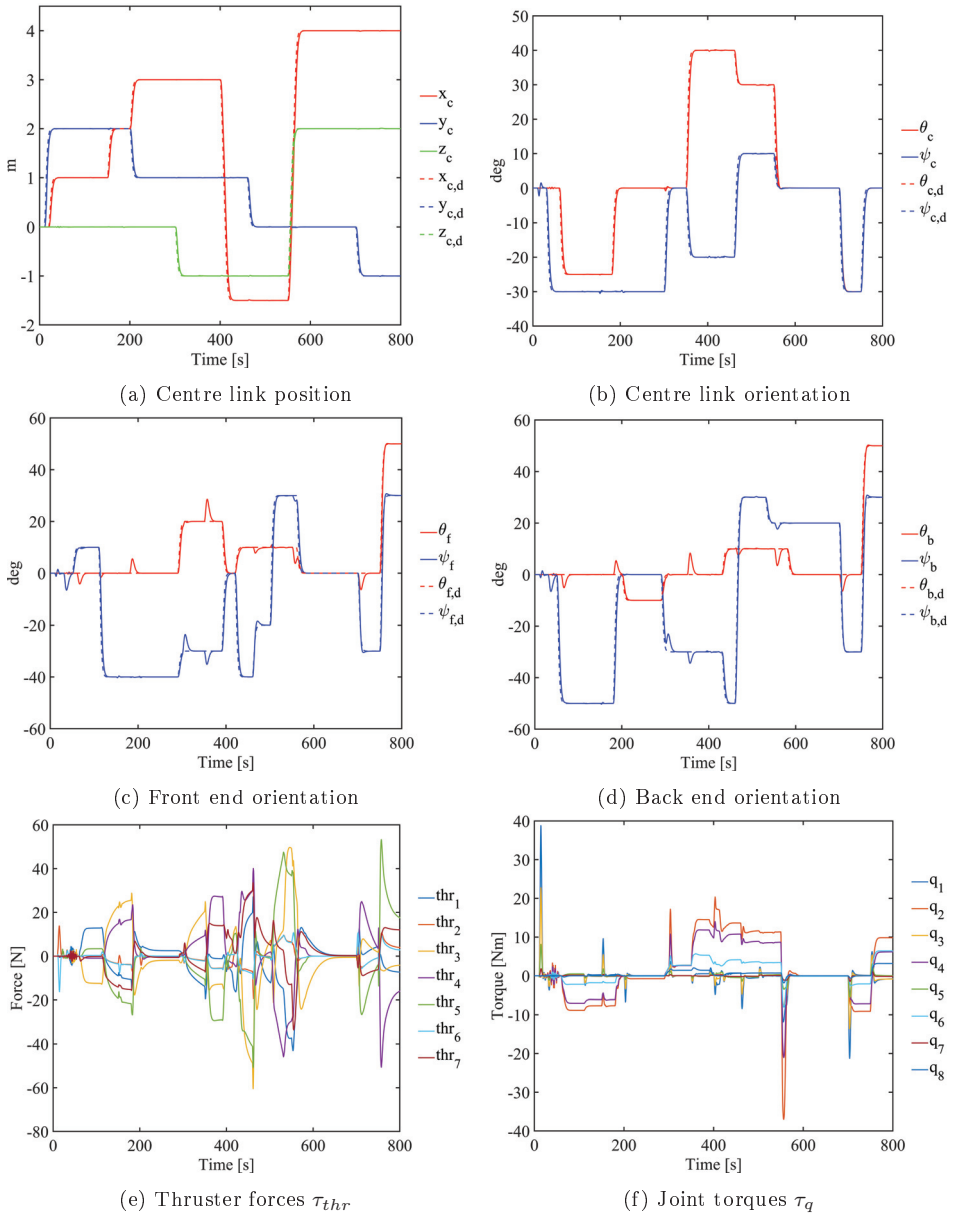
$$\tau = M(-k_p\tilde{\zeta} + \dot{\zeta}_r) + C\zeta + D\zeta + g \tag{7.31}$$

where $\tilde{\zeta} = \zeta - \zeta_r$ and k_p is a control gain.

Figure 7.3 shows the results when the PID controller (7.30) is used, with the control gains chosen as $k_p = 250$, $k_d = 80$ and $k_i = 10$. Figures 7.4 and 7.5 shows the results when the feedback linearisation controller (7.31) is used, with the control gain chosen as $k_p = 10$. The reason two figures are included for the feedback linearisation controller is because the feedback linearisation controller had problems when the thruster allocation scheme used was included; we therefore included one figure where the thruster allocation was used, i.e., Figure 7.4, and one figure where the thruster allocation was not used, i.e., Figure 7.5. The reasons why the feedback linearisation controller had problems with the thruster allocation scheme will be discussed in the discussion section below.

Discussion

Figures 7.1a and 7.1b show the commanded and the actual position and orientation of the centre link corresponding to task 1 when the first-order SMC (7.20) is used. There is a small deviation from the set-point for the centre link pitch angle due to the boundary layer of the saturation function, which is in agreement with ultimate boundedness described in Remark 7.5. The results for tasks 2 and 3 are shown in Figure 7.1c and Figure 7.1d. The combined kinematic and dynamic control law is able to fulfil all the tasks simultaneously, as stated in Theorem 7.1 in combination with Theorem 7.2, but some transient deviations can be observed for the front and the back orientations when changing the pitch and yaw set-points for the centre link. These deviations occur because task 1 is the primary task and tasks 2 and 3 are secondary tasks. The first term in (7.29) does not consider the task errors for tasks 2 and 3. Satisfying the desired centre link orientation will therefore introduce errors



(a) Centre link position

(b) Centre link orientation

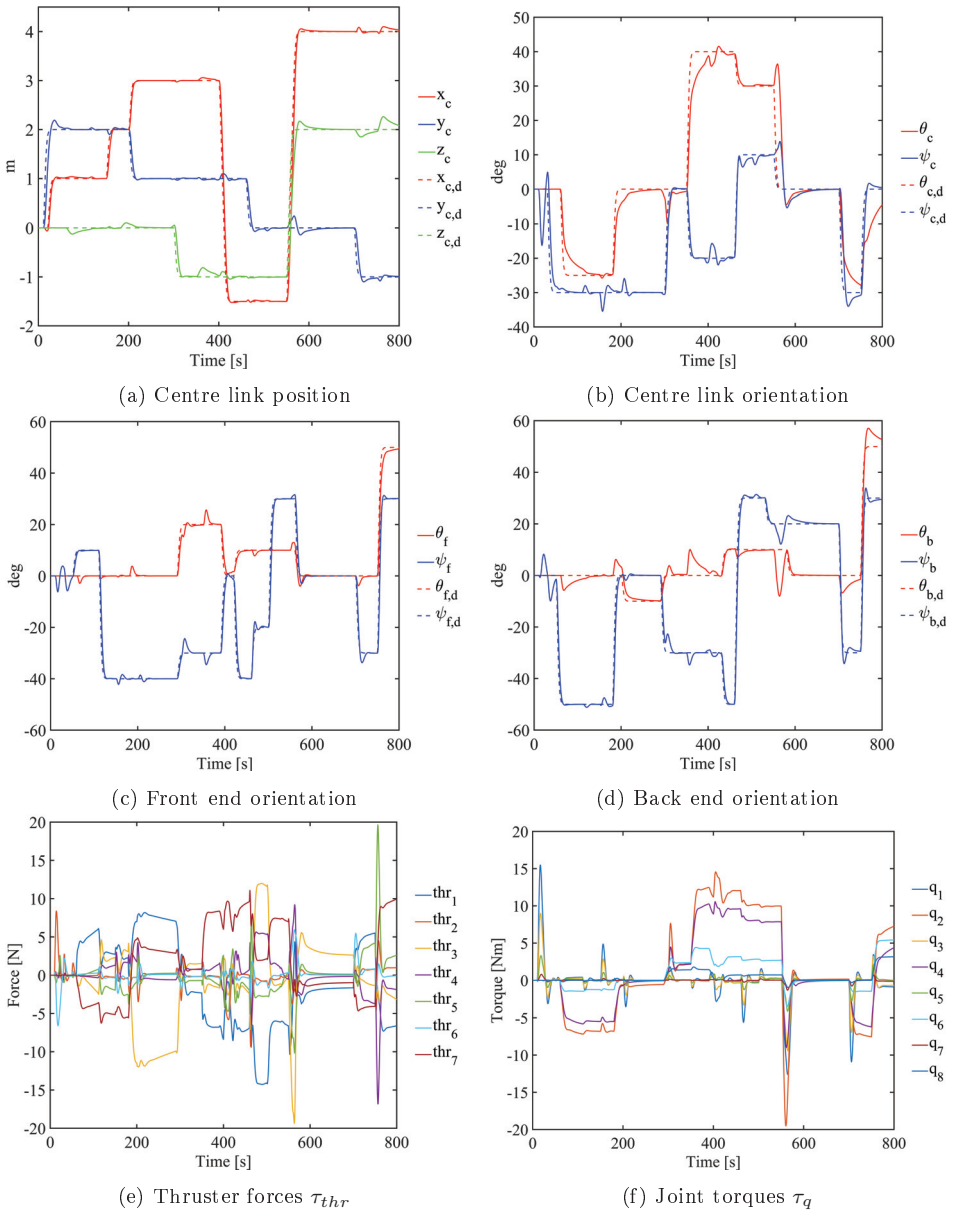
(c) Front end orientation

(d) Back end orientation

(e) Thruster forces τ_{thr}

(f) Joint torques τ_q

Figure 7.2: Results when the GSTA with adaptive gains is used for the AIAUV described by Table 2.2



(a) Centre link position

(b) Centre link orientation

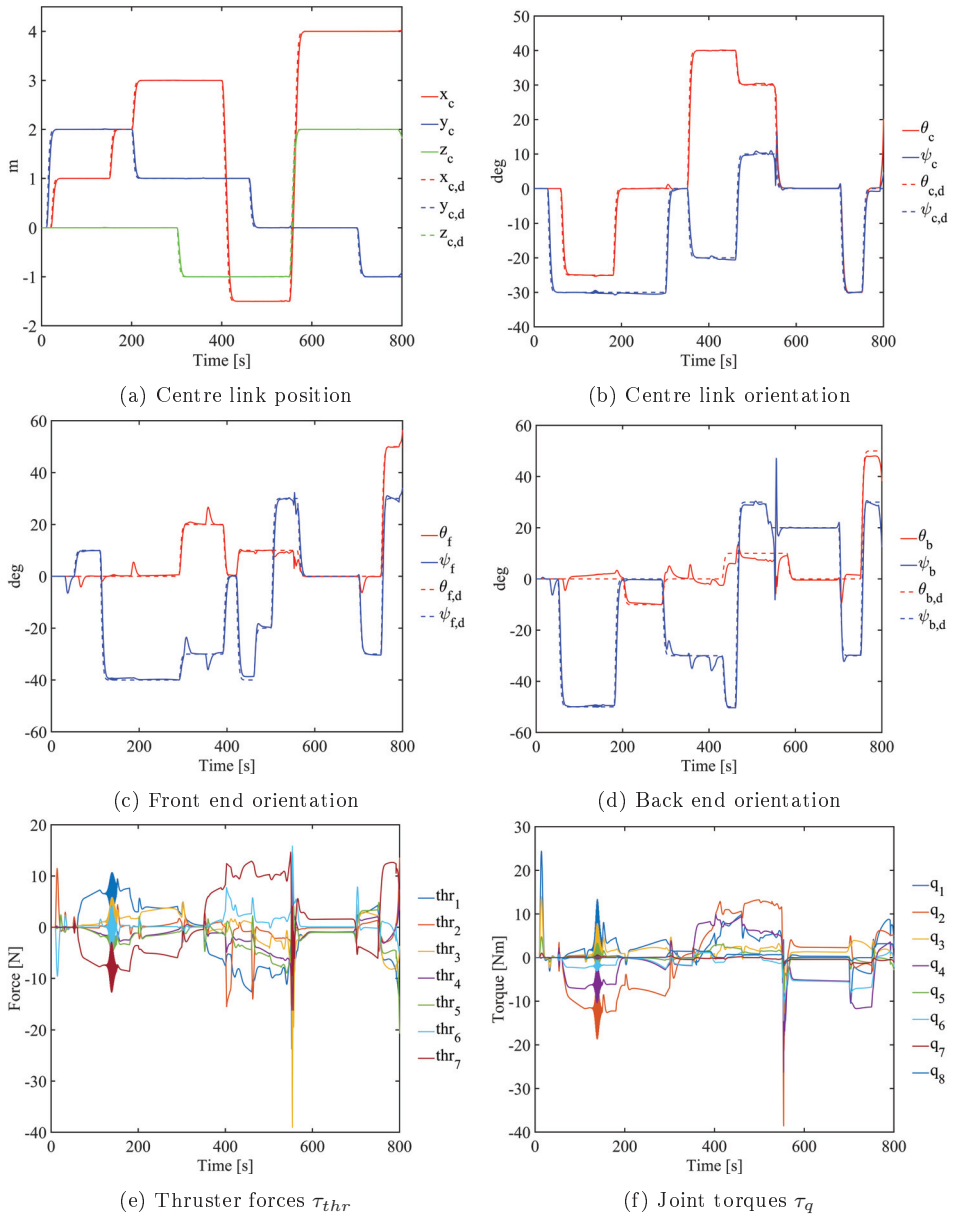
(c) Front end orientation

(d) Back end orientation

(e) Thruster forces τ_{thr}

(f) Joint torques τ_q

Figure 7.3: Results when the PID controller is used for the AIAUV described by Table 2.2



(a) Centre link position

(b) Centre link orientation

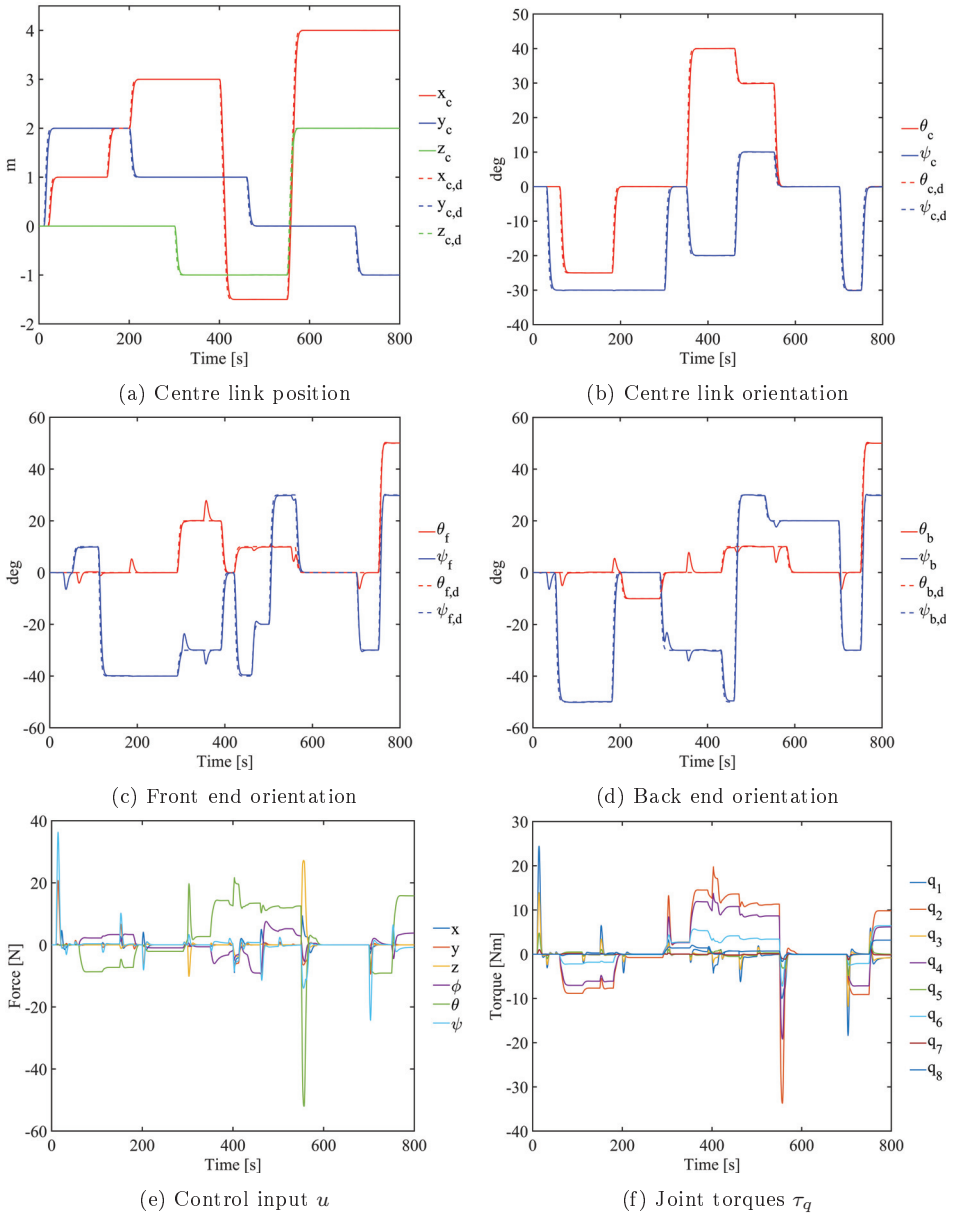
(c) Front end orientation

(d) Back end orientation

(e) Thruster forces τ_{thr}

(f) Joint torques τ_q

Figure 7.4: Results when the feedback linearisation controller with thruster allocation is used for the AIAUV described by Table 2.2



(a) Centre link position

(b) Centre link orientation

(c) Front end orientation

(d) Back end orientation

(e) Control input u

(f) Joint torques τ_q

Figure 7.5: Results when the feedback linearisation controller is used for the AIAUV described by Table 2.2

in the orientations of the front end and the back end. These deviations disappear as soon as the second term in (7.29) compensates for these task errors. Figures 7.1e and 7.1f show the thruster forces and joint torques used. The control inputs are smooth and below 100 N, which is the limit for the thrusters. The control inputs are therefore feasible.

Figures 7.2a and 7.2b show the commanded and the actual position and orientation of the centre link corresponding to task 1 when the GSTA with adaptive gains (6.7) is used. We now find that the small deviation from the set-point for the centre link pitch angle is eliminated. This result occurs because when the GSTA with adaptive gains is used, we do not have to introduce a boundary layer. The results for tasks 2 and 3 are shown in Figure 7.2c and Figure 7.2d. As shown, the combined kinematic and dynamic control law is able to fulfil all the tasks simultaneously, as stated in Theorem 7.1 in combination with Theorem 7.3, but some transient deviations can be observed for the front and the back orientations when changing the pitch and yaw set-points for the centre link for the GSTA with adaptive gains. The reason why we are unable to remove these deviations is because they are introduced by the SRMTP method, as described when the transient errors for the first-order SMC was presented. The simulation results support the theoretical results, and we find that all the set-point tasks are fulfilled. Figures 7.2e and 7.2f show the thruster forces and joint torques used. The control inputs are smooth and below 100 N, which is the limit for the thrusters. The control inputs are therefore feasible.

Figures 7.3a and 7.3b show the commanded and the actual position and orientation of the centre link corresponding to task 1 when the PID controller (7.30) is used. We can see that when the PID controller is used, the reference is tracked; small deviations from the position and orientation are observed when the other set-points are changed. We also see that the transient error for pitch is much larger than that of the SMC algorithms. The PID controller also introduces some overshoot in all states. The results for tasks 2 and 3 are shown in Figure 7.3c and Figure 7.3d. The PID controller is able to fulfil all the tasks simultaneously; however, here, we not only have transient deviations for the front and the back orientations when changing the pitch and yaw set-points for the centre link but also note tendencies of overshoot and also some other transient errors. The transient errors caused by the changing of set-points are also larger than that of the SMC algorithms. We tried to increase the gains to determine if that led to better performance, but we then obtained a non-feasible control input. The only way to obtain a better performance would therefore be to find another combination of gains, with $k_p \leq 250$, that yielded better results. After extensive tuning efforts, the results shown here represent the best performance we could achieve. It is therefore clear that the SMC algorithms perform better than the PID controller, and they are also much easier to tune to obtain optimal performance. Figures 7.3e and 7.3f show the thruster forces and joint torques used. The control inputs are smooth and below 100 N, which is the limit for the thrusters. The control inputs are therefore feasible.

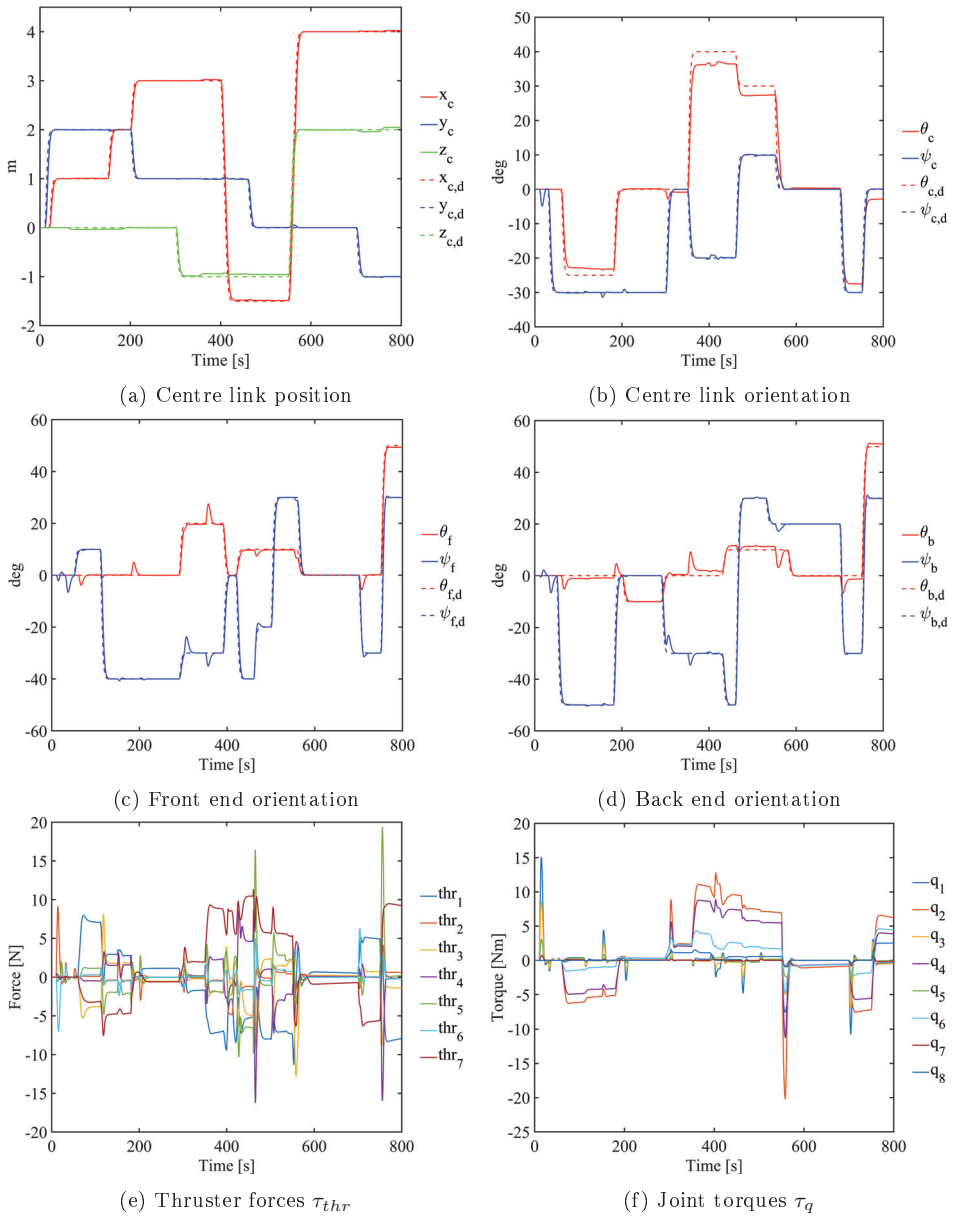
Figures 7.4a and 7.4b show the commanded and the actual position and orientation of the centre link corresponding to task 1 when the feedback linearisation controller (7.31) is used. We expect perfect tracking since there is no model uncertainty, and from the figures, we see that the position is perfectly tracked. For

the orientation however, there are some small deviation in yaw. From Figure 7.4c we can see that task 2 is fulfilled, however, from Figure 7.4d, we see that task 3 is not perfectly fulfilled. It cannot follow the pitch reference perfectly, and there is a large jump in yaw. The reason task 3 is not perfectly fulfilled is because of the thruster allocation scheme. This can be seen from Figure 7.5, when we sent the calculated desired force in six degrees of freedom (6DOF) directly to the model, i.e. the thruster allocation scheme was not used. From Figures 7.5a, 7.5b, 7.5c and 7.5d we can see that task 1, task 2 and task 3 are fulfilled when the thruster allocation scheme is not used, and that the trajectories are perfectly followed, except for the small transient deviation for the front and the back orientation when the pitch and yaw set-points for the centre link are changed, which are introduced by the SRMTP method. The reason the thruster allocation scheme is creating problems for the feedback linearisation controller is because we have to use a damped inverse to calculate the force that will be distributed to the thrusters, since the robot is un-actuated in roll when the joints angles are equal to zero. The un-actuated roll axis causes the inverse of the thruster configuration matrix to become singular, and a damped inverse therefore has to be used. However, the damped inverse does not obtain the commanded force in 6DOF that we sought, which created problems for the feedback linearisation controller. The reason why only the base is affected may be because task 1 is fulfilled given that it is the primary task, while task 2 is fulfilled because the joints are not affected. The reason the base is affected is therefore because the reference generated by the SRMTP method is for the base. We can therefore see that the feedback linearisation controller is much more sensitive to unmodelled dynamics, since the effects of the thruster allocation scheme are not visible for any of the SMC algorithms. Note that we chose $k_p = 10$ because that was when we achieved the best performance when using the thruster allocation scheme. Figures 7.4e, 7.5e, 7.4f and 7.5f show the thruster forces and joint torques used. The control inputs are smooth and below 100 N, which is the limit for the thrusters. The control inputs are therefore feasible.

7.4.3 Simulations with model parameter errors

In this section, we will show results where the AIAUV link lengths are reduced by 20%. The AIAUV simulation model is then given by the parameters in Table 7.1. However, the model used to find the control inputs is not changed; this is to highlight the robustness of the SMC algorithms. This means that we keep the controller gains and the estimates of the model parameters as in Section 7.4.2.

Figure 7.6 shows the results when the first-order SMC (7.20) is used, and Figure 7.7 shows the results when the GSTA with adaptive gains (6.7) is used. Since the PID controller already showed that its performance is worse than the SMC algorithms, we did not include results from the PID controller in this section. For the feedback linearisation controller, we include only the results for the case where the thruster allocation scheme is not used, as shown in Figure 7.8.



(a) Centre link position

(b) Centre link orientation

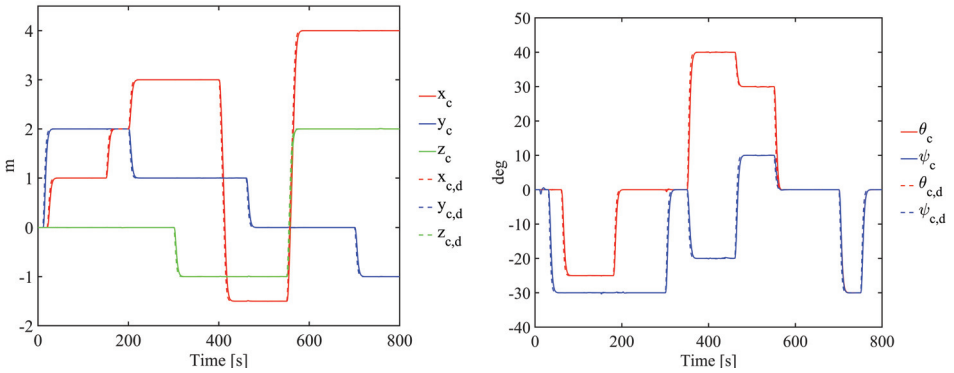
(c) Front end orientation

(d) Back end orientation

(e) Thruster forces τ_{thr}

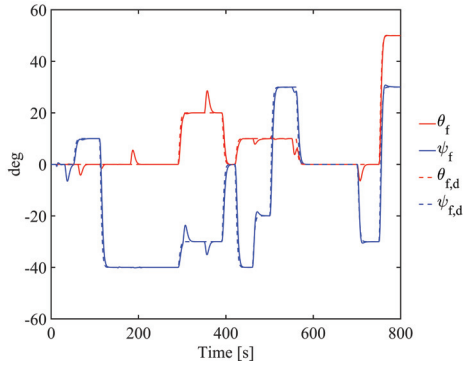
(f) Joint torques τ_q

Figure 7.6: Results when the first-order SMC is used for the AIAUV described by Table 7.1

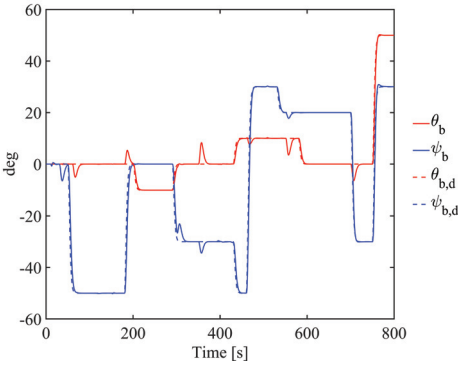


(a) Centre link position

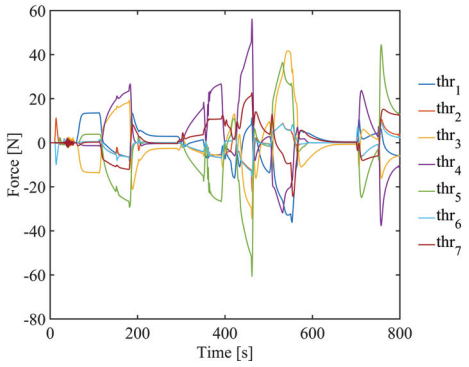
(b) Centre link orientation



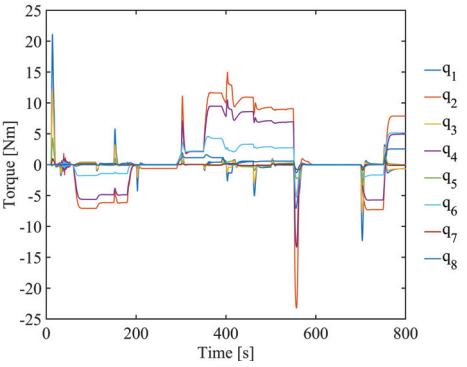
(c) Front end orientation



(d) Back end orientation

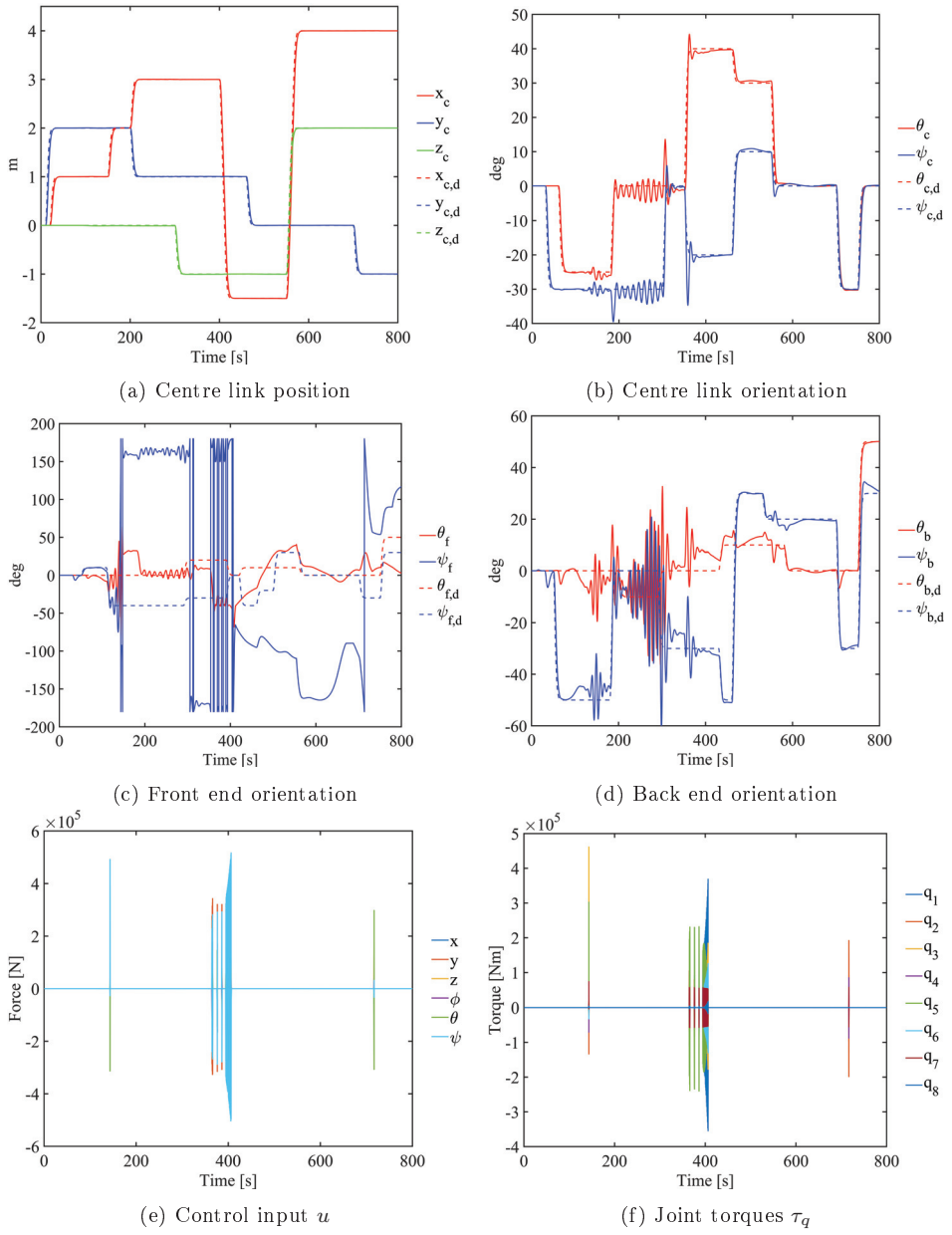


(e) Thruster forces τ_{thr}



(f) Joint torques τ_q

Figure 7.7: Results when the GSTA with adaptive gains is used for the AIAUV described by Table 7.1



(a) Centre link position

(b) Centre link orientation

(c) Front end orientation

(d) Back end orientation

(e) Control input u

(f) Joint torques τ_q

Figure 7.8: Results when the feedback linearisation controller is used for the AIAUV described by Table 7.1

Table 7.1: Eelume 2016 link properties (link lengths reduced by 20%)

Link nr.	Length [m]	Mass [kg]	Thrusters
1	0.50	12.6	None
2, 4, 6, 8	0.08	2.0	None
3	0.47	12.0	2: Z, Y
5	0.64	16.3	3: X, X, Z
7	0.47	12.0	2: Y, Z
9	0.30	7.5	None

Discussion

From Figures 7.6 and 7.7, we can see that the tracking results are almost identical to those obtained in Section 7.4.2. We can therefore conclude that the SMC algorithms are robust to modelling uncertainties, as their performance is not affected by making the AIAUV smaller. From Figure 7.8, we can see that the feedback linearisation controller is greatly affected by the change of model. To obtain these results, we had to change the control gain to $k_p = 200$; if we used the gains chosen in Section 7.4.2, we could not obtain any stable results. With $k_p = 200$, we can see from Figures 7.8a and 7.8b that task 1 is almost fulfilled, while in Figures 7.8c and 7.8d, we can see that task 2 and 3 are not fulfilled. Figures 7.8e and 7.8f reveal that the control input in this case is not feasible. We found that we had to set $k_p = 500$ for all the tasks to be fulfilled with the feedback linearisation controller. Thus, for the feedback linearisation controller to work with modelling uncertainties, we have to tune the controller correctly, which can be difficult. The SMC algorithms are therefore much more applicable for the AIAUV which is prone to modelling errors and uncertainties.

7.5 Experimental case study

In this section, a case study using the AIAUV based on experimental results obtained during the summer of 2020 is presented. The purpose of the experiments was to validate the underlying theory and the robustness of the control approach by showing that the proposed approach also works in practice and not only in the ideal case presented in simulations. The robot used is the Eelume 2020 version of the robot presented in Section 2.4.1 and the test case used is the task-priority case described in Section 2.4.3. We will also present simulation results for the test case presented in Section 2.4.3, such that we can compare the experimental results with simulation results. We show results with the adaptive GSTA from Section 7.3.2. We were not able to get experimental results by using the control law from Section 7.3.1. The first-order SMC algorithm was to difficult to tune during the experiments, and even with extensive tuning efforts, we were not able to get it to work. So we will not show results with the first-order SMC. The tuning was simpler for the higher-order SMCs, and we were able to obtain results with both the STA with adaptive gains (3.1) and the GSTA (4.1). We can therefore present a comparison with the STA with adaptive gains and the GSTA to evaluate whether

adding adaptive gains to the GSTA actually improves the resulting tracking capabilities, as we then get both the theoretical advantages afforded by the GSTA and the practical advantages afforded by adaptive gains. Additionally, we compare the results with those of a standard PID controller.

7.5.1 Realistic simulations

In this section, simulation results with control gains obtained during the experiments are presented. We used the simulation model presented in Section 2.2 and the Eelume 2020 version of the robot presented in Section 2.2.2. The test case used is the SRMTP case from Section 2.4.3. For the simulations, the ode3 fixed-step solver with a step size of 0.01 was used.

Results AGSTA

We were not able to directly use the gains obtained during the experiments; it was necessary to make some small changes to the experimental gains to allow them to work properly in the simulations. However, we used the experimental gains as the starting point and attempted to modify them as little as possible. The gains used for the GSTA with adaptive gains are presented in Table 7.2. For the adaptive GSTA, it was necessary only to change the λ gain for the joints from 0.1 to 1. The results obtained when using the control law proposed in Section 7.3.2 are

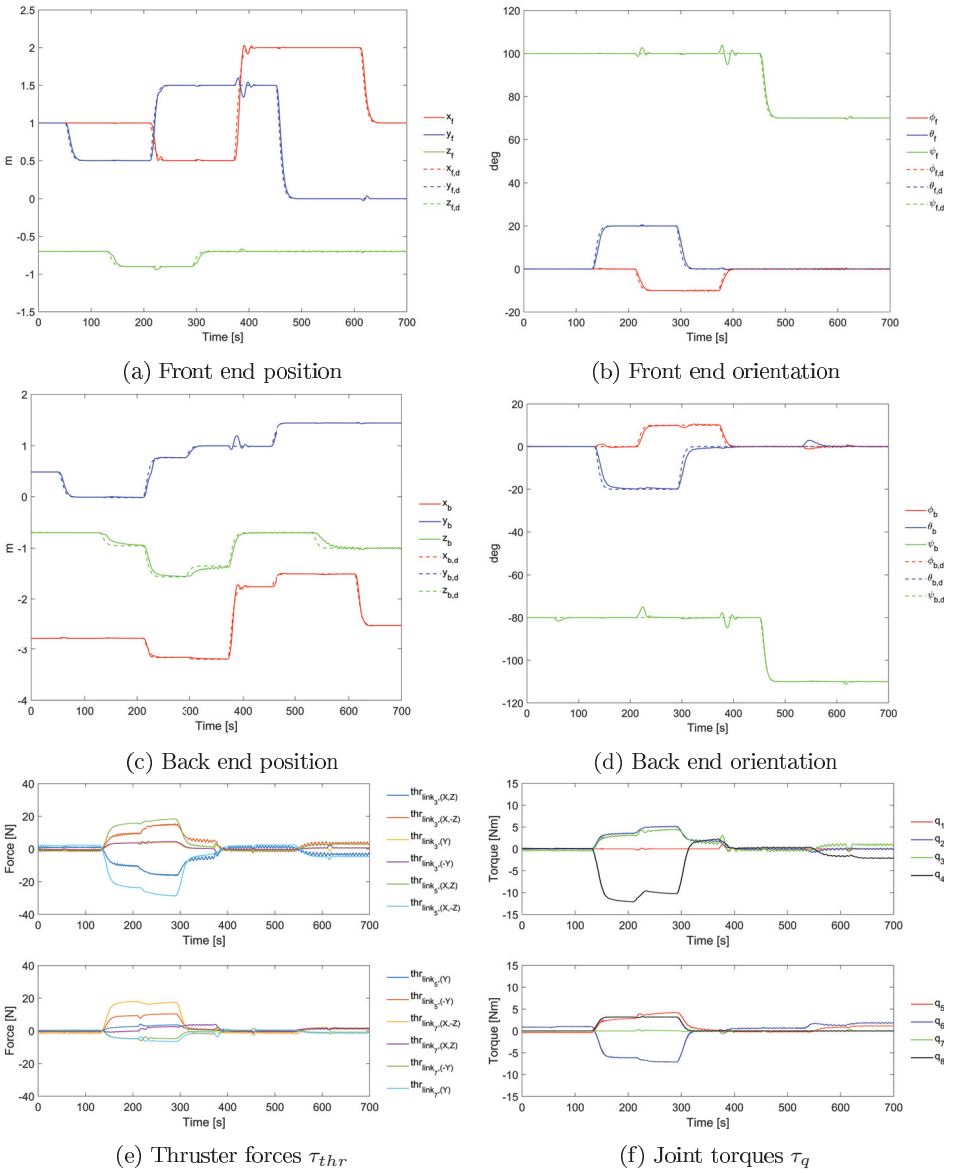
Table 7.2: Simulation: Control gains for the adaptive GSTA for the task-priority case

Gain	Values
ε	$1 \cdot 10^{-5} e_{14}^T$
λ	e_{14}^T
γ_1	e_{14}^T
ω_1	$[e_6 \quad 0.1e_8]^T$
β	$[3e_6 \quad e_8]^T$
α_m	$0.05e_{14}^T$

presented in Figure 7.9, and the evolution of the adaptive gains over time for the adaptive GSTA can be seen in Figure 7.10. In Fig. 7.9f label q_i corresponds to the torque used for joint i, and in Fig. 7.10 label x corresponds to the evolution of the adaptive gains $k_1(t)$ and $k_2(t)$ over time for the x-dimension, label y corresponds to the evolution of the adaptive gains $k_1(t)$ and $k_2(t)$ over time for the y-dimension, etc.

Results comparison

For comparison, we also obtained results using the previously mentioned algorithms, i.e., the STA with adaptive gains from (3.1) and the GSTA from (4.1). For both the STA with adaptive gains and the GSTA, we used the sliding surface defined in (7.7). We also compare the results with those of a standard PID



(a) Front end position

(b) Front end orientation

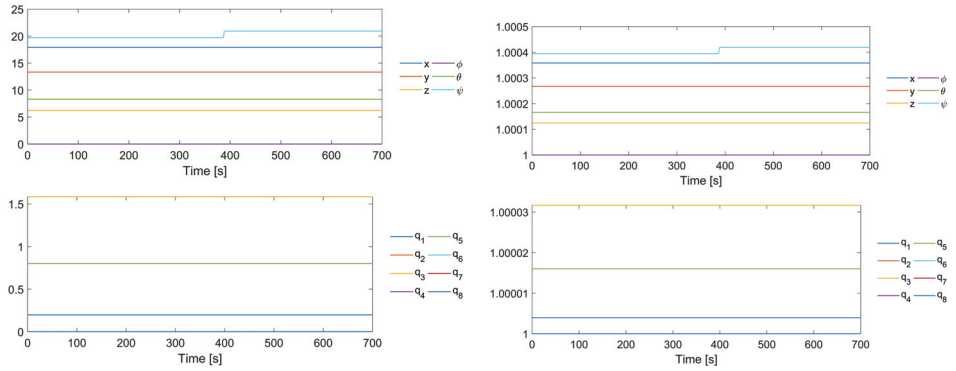
(c) Back end position

(d) Back end orientation

(e) Thruster forces τ_{thr}

(f) Joint torques τ_q

Figure 7.9: Simulation results using the adaptive GSTA for the task-priority case



(a) Evolution of $k_1(t)$ over time

(b) Evolution of $k_2(t)$ over time

Figure 7.10: Simulation: Evolution of the adaptive gains over time for the adaptive GSTA for the task-priority case

controller (7.30). The reason we chose to include a PID controller as a representative standard control method is because it is one of the most widely used types of controllers and is known to yield good results if tuned correctly. However, it does not provide any stability guarantee. As mentioned previously, we were not able to use the control gains found during the experiments directly. The gains used in the simulations are presented in Table 7.3 for the STA with adaptive gains, in Table 7.4 for the GSTA and in Table 7.5 for the PID controller. For the STA with adaptive gains, we needed only to change the λ gain for the joints from 0.1 to 1. For the GSTA, it was necessary to change the β gains from $[80e_6 \quad 15e_8]$ to $[25e_6 \quad 45e_8]$, and for the PID controller, we needed to change the k_p gain for the joints from 20 to 40 and the k_i gain for the joints from 10 to 15. The tracking errors for all

Table 7.3: Simulation: Control gains for the STA with adaptive gains for the task-priority case

Gain	Values
ε	$1 \cdot 10^{-5} e_{14}^T$
λ	e_{14}^T
γ_1	e_{14}^T
ω_1	$[e_6 \quad 0.1e_8]^T$
α_m	$0.05e_{14}^T$

the algorithms are presented in Figure 7.11. We have in this section chosen to use error plots, instead of trajectory plots as in Section 7.4, to make the comparison easier and more visible. In Figure 7.12, we present the thruster forces and joint torques for all algorithms except the adaptive GSTA, for which the forces and torques have been previously presented in Figure 7.9. For the STA with adaptive gains, the evolution of the adaptive gains over time can be seen in Figure 7.13. In Table 7.6, the RMSEs and maximum errors of the position and orientation for

Table 7.4: Simulation: Control gains for the GSTA for the task-priority case

Gain	**Values**
k_1	$[3e_6 \quad e_8]^T$
k_2	$[0.004e_6 \quad 0.002e_8]^T$
β	$[25e_6 \quad 45e_8]^T$

Table 7.5: Simulation: Control gains for the PID controller for the task-priority case

Gain	**Values**
k_p	$[250e_6 \quad 40e_8]^T$
k_d	$[80e_6 \quad 10e_8]^T$
k_i	$[15e_6 \quad 15e_8]^T$

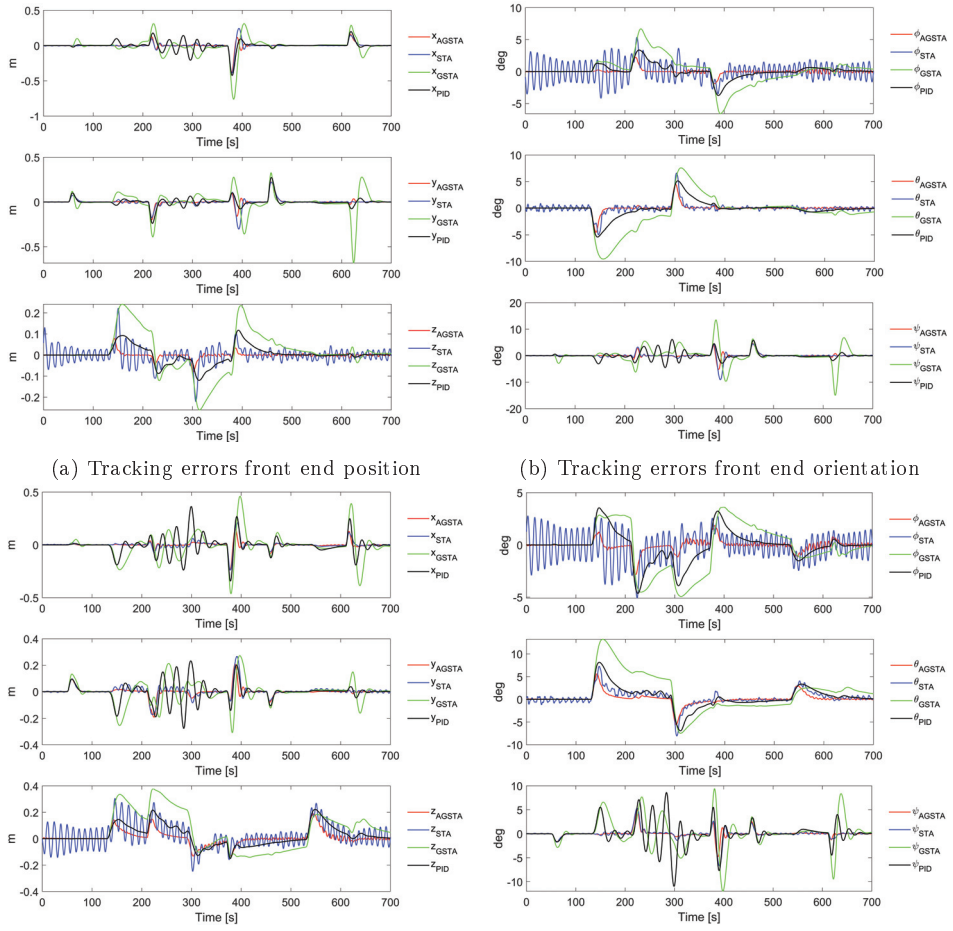
each task are given for all algorithms. In Table 7.7, the RMS values of the thruster forces, the derivative of the thruster forces, the joint torques, and the derivative of the joint torques are given for all algorithms. The RMS of the thruster forces and the joint torques provides a measure of how much control effort that is used, and the RMS of the derivative of the thruster forces and the derivative of the joint torques provides a measure of how much chattering that is present in the thruster forces and the joint torques.

Table 7.6: Simulation: Comparison of the tracking results for the task-priority case

Algorithm		**RMSE**		**Maximum error**	
		Task 1	Task 2	Task 1	Task 2
AGSTA	Position	0.0313	0.0382	0.2295	0.2124
	Orientation	0.0113	0.0146	0.0698	0.0787
STA	Position	0.0467	0.0581	0.3057	0.2713
	Orientation	0.0216	0.0239	0.1227	0.1158
GSTA	Position	0.1106	0.1177	0.5683	0.3832
	Orientation	0.0483	0.0550	0.1816	0.1755
PID	Position	0.0512	0.0763	0.2732	0.2882
	Orientation	0.0227	0.0356	0.0891	0.1385

Table 7.7: Simulation: Comparison of the control inputs for the task-priority case

Algorithm	**RMS**			
	τ_{thr}	$\dot{\tau}_{thr}$	τ_q	$\dot{\tau}_q$
AGSTA	4.7933	0.4191	1.9777	0.4487
STA	5.8351	0.7371	2.4834	0.8704
GSTA	4.3251	107.6121	1.8879	83.8284
PID	5.4231	0.1324	2.3276	0.0778



(a) Tracking errors front end position

(b) Tracking errors front end orientation

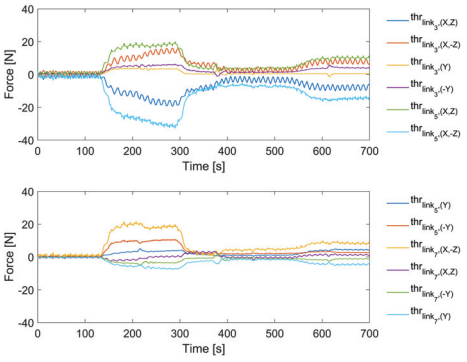
(c) Tracking errors back end position

(d) Tracking errors back end orientation

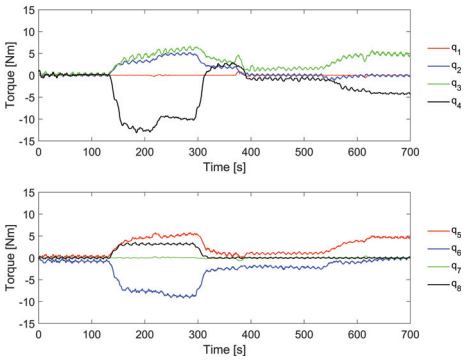
Figure 7.11: Simulation: Tracking errors for the task-priority case

Discussion

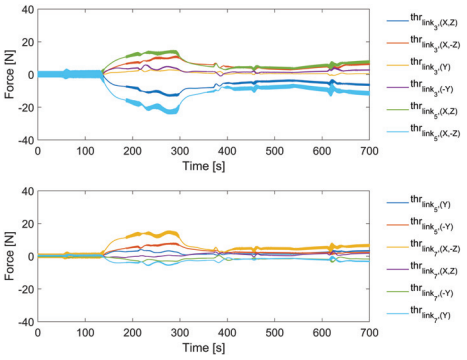
Figure 7.9a and Figure 7.9b show the commanded and actual position and orientation of the front end corresponding to task 1 when the GSTA with adaptive gains (6.7) is used. For task 1, we can see some small transient deviations at approximately 200 s and 400 s in the position and in the yaw orientation. These transient errors correspond to a roll movement of the front end, which may indicate that movement in the roll direction interferes with movement in the other DOFs. The results for task 2 are shown in Figure 7.9c and Figure 7.9d. As shown, the combination of the SRMTP method and the dynamic control law enables all tasks to be performed simultaneously, as indicated by Theorem 7.1 in combination with Theorem 7.3; however, some transient deviations can also be observed for task 2.



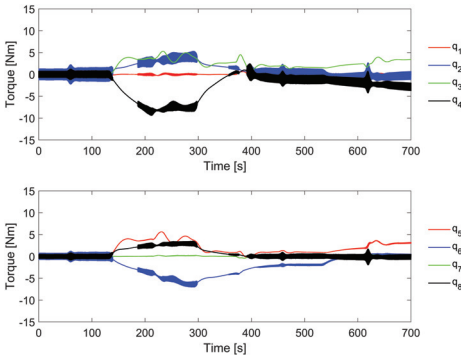
(a) Thruster forces τ_{thr} used for STA with adaptive gains



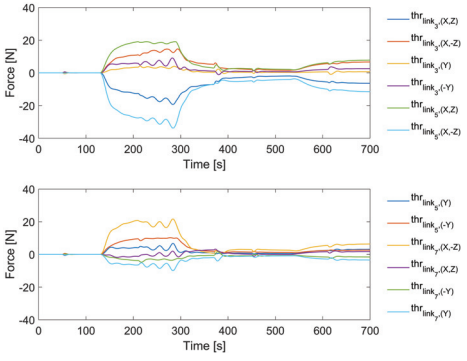
(b) Joint torques τ_q used for STA with adaptive gains



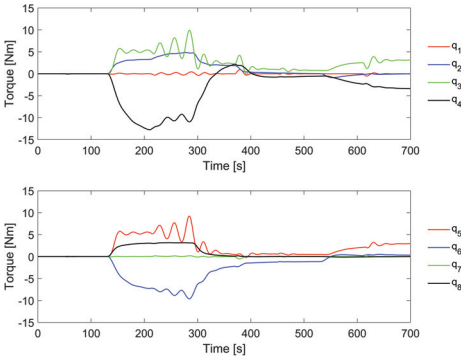
(c) Thruster forces τ_{thr} used for the GSTA



(d) Joint torques τ_q used for the GSTA

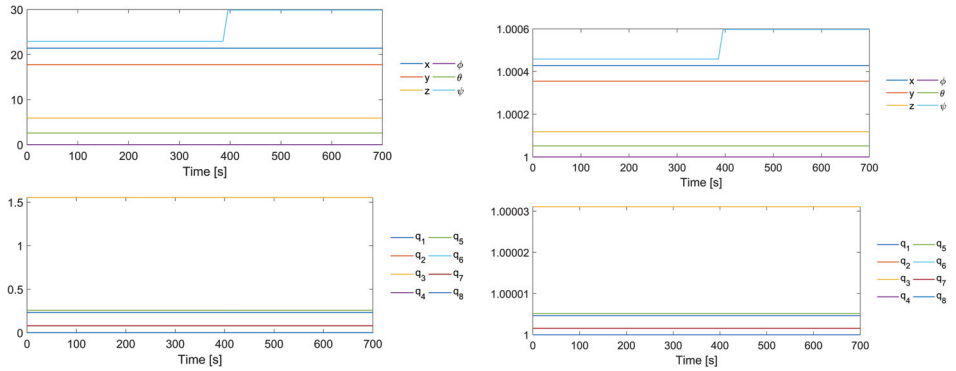


(e) Thruster forces τ_{thr} used for the PID controller



(f) Joint torques τ_q used for the PID controller

Figure 7.12: Simulation: Thruster forces τ_{thr} and joint torques τ_q used for the task-priority case



(a) Evolution of $\alpha(t)$ over time

(b) Evolution of $\beta(t)$ over time

Figure 7.13: Simulation: Evolution of the adaptive gains over time for the STA with adaptive gains for the task-priority case

We see some small transient errors around 200 s and 400 s in position and yaw that correspond to roll movements for task 1 and task 2. We also see small transient errors in the roll and pitch directions at approximately 550 s, which correspond to movement in the x-direction for task 1. As previously discussed in Section 7.4, these deviations occur because task 1 is the primary task, whereas task 2 is the secondary task. The first term in (2.13) does not consider the task errors for task 2. The attempt to achieve the desired front end position and orientation will consequently introduce errors in the back end position and orientation. These deviations disappear as soon as the second term in (2.13) compensates for these task errors. The simulation results therefore support the theoretical results, and we find that all the set-point tasks are fulfilled. From Figure 7.9e, we can see that the forces used are smooth and well below 60 N, which is approximately the limit of the thrusters in the 2020 version of the robot. From Figure 7.9f, we can see that the joint torques are smooth and below 16 Nm, which is the limit on the joint torques. The control inputs are therefore feasible. From Figure 7.10a and Figure 7.10b, we can also see that the gains $k_1(t)$ and $k_2(t)$ increase linearly and converge to suitable values.

From Figure 7.11, we can clearly see that there are some points that exhibit transient errors introduced by the SRMTP method. However, we also see that the magnitudes of these transient errors are different for the different control algorithms. From Figure 7.11a, we can see that the adaptive GSTA clearly results in the smallest position errors for task 1, while the STA with adaptive gains yields the second smallest errors. However, the STA with adaptive gains also introduces some oscillation in z. The PID controller achieves the third smallest errors, while the GSTA results in the largest errors. Figure 7.11b shows that this is also the case for the orientation errors for task 1. However, it appears that the PID controller actually yields better results than the STA with adaptive gains. The STA with adaptive gains also introduces some oscillations into the states. When we consider the position errors for task 2 in Figure 7.11c, we see that the STA with adaptive

gains again yields better results than the PID controller, but the STA with adaptive gains also introduces oscillation in z, while the adaptive GSTA still results in the lowest errors and best results. When we additionally consider the orientation errors for task 2 in Figure 7.11d, we see that here also, the adaptive GSTA achieves the lowest errors. The STA with adaptive gains again introduces some oscillations into the states; however, it also yields the second smallest errors. These findings are further supported by Table 7.6, which shows that the adaptive GSTA results in the smallest RMSE values and maximum error values in both position and orientation for both tasks. We can therefore conclude that the adaptive GSTA exhibits the best tracking performance overall, i.e., the smallest errors. The STA with adaptive gains comes in second overall, even though the PID controller has smaller maximum error values for task 1. The PID controller thus takes third place, while the GSTA comes fourth.

Regarding the thruster and torque use of the different algorithms, we can see from Figure 7.9e, Figure 7.9f and Figure 7.12 that the force use is quite similar for all of the considered algorithms. However, the GSTA uses less force than the other algorithms and introduces some chattering, which may be the reason why it results in the highest tracking errors. From Table 7.7, we can see that there are some small differences among the other algorithms, as well. However, the RMS values of the thruster forces and the joint torques indicate that the adaptive GSTA uses the least force after the GSTA, even though it produces the best tracking performance. Nevertheless, the STA with adaptive gains and the PID controller also use very similar amounts of force. Furthermore, from the RMS values of $\dot{\tau}_{thr}$ and $\dot{\tau}_q$, we can see that the only algorithm that introduces chattering is the GSTA.

7.5.2 Experimental investigation

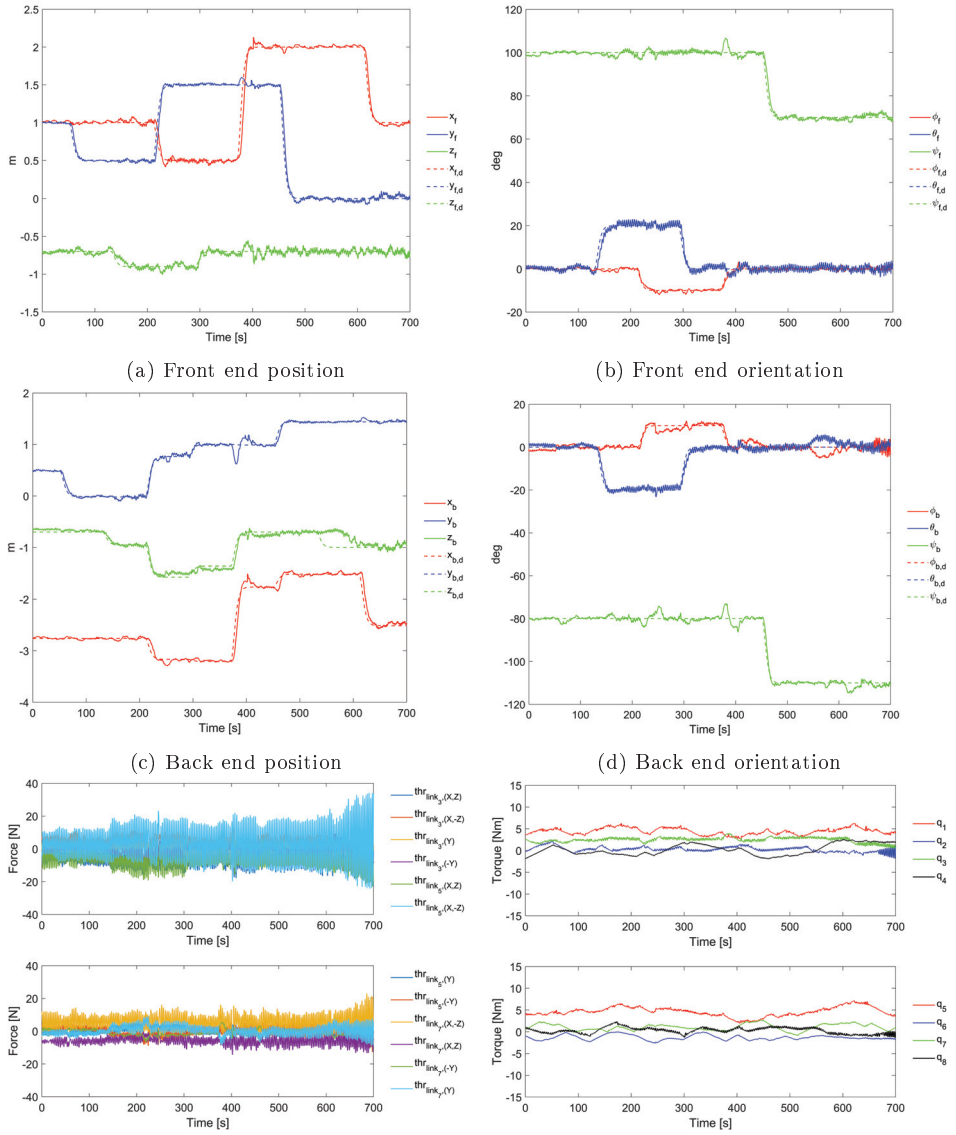
In this section, the experimental results obtained during the summer of 2020 are presented. The robot used was the Eelume 2020 version of the robot presented in Section 2.4.1 and the test case used was the same as for the simulation, and was explained in Section 2.4.3.

Results AGSTA

The gains for the GSTA with adaptive gains are presented in Table 7.8. For the adaptive GSTA, the choice of the gains should not be very important since the gains will autonomously adapt to suitable values, and indeed, it was observed during the experiments that it was easy to find a starting point for the adaptive gains that yielded good tracking results. The results obtained using the control law proposed in Section 7.3.2 are presented in Figure 7.14, and the evolution of the adaptive gains over time for the adaptive GSTA can be seen in Figure 7.15.

Results comparison

For comparison, we also obtained results using the previously mentioned algorithms, i.e., the STA with adaptive gains (3.1), the GSTA (4.1) and a PID controller (7.30). For the SMC algorithms, we used the sliding surface defined in (7.7).



(a) Front end position

(b) Front end orientation

(c) Back end position

(d) Back end orientation

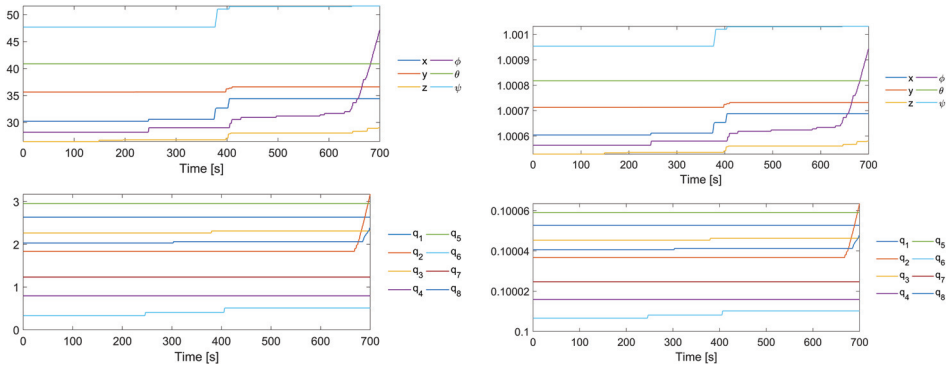
(e) Thruster forces τ_{thr}

(f) Joint torques τ_q

Figure 7.14: Experimental results using the adaptive GSTA for the task-priority case

Table 7.8: Experiments: Control gains for the adaptive GSTA for the task-priority case

Gain	Values
ε	$1 \cdot 10^{-5} e_{14}^T$
λ	$[e_6 \quad 0.1e_8]^T$
γ_1	e_{14}^T
ω_1	$[e_6 \quad 0.1e_8]^T$
β	$[3e_6 \quad e_8]^T$
α_m	$0.05e_{14}^T$



(a) Evolution of $k_1(t)$ over time

(b) Evolution of $k_2(t)$ over time

Figure 7.15: Experiments: Evolution of the adaptive gains over time for the adaptive GSTA for the task-priority case

The gains found for the STA with adaptive gains, the GSTA and the PID controller during the experiments are presented in Table 7.9, Table 7.10 and Table 7.11, respectively. The tracking errors for all algorithms are presented in Figure 7.16. Figure 7.17 shows the thruster forces and joint torques for all algorithms except the adaptive GSTA, for which the forces and torques have been previously presented in Figure 7.14. For the STA with adaptive gains, the evolution of the adaptive gains over time can be seen in Figure 7.18. In Table 7.12, the RMSEs and maximum

Table 7.9: Experiments: Control gains for the STA with adaptive gains for the task-priority case

Gain	Values
ε	$1 \cdot 10^{-5} e_{14}^T$
λ	$[e_6 \quad 0.1e_8^T]$
γ_1	e_{14}
ω_1	$[e_6 \quad 0.1e_8]^T$
α_m	$0.05e_{14}^T$

Table 7.10: Experiments: Control gains for the GSTA for the task-priority case

Gain	Values
k_1	$[3e_6 \quad e_8]^T$
k_2	$[0.004e_6 \quad 0.002e_8]^T$
β	$[80e_6 \quad 15e_8]^T$

Table 7.11: Experiments: Control gains for the PID controller for the task-priority case

Gain	Values
k_p	$[250e_6 \quad 20e_8]^T$
k_d	$[80e_6 \quad 10e_8]^T$
k_i	$[15e_6 \quad 10e_8]^T$

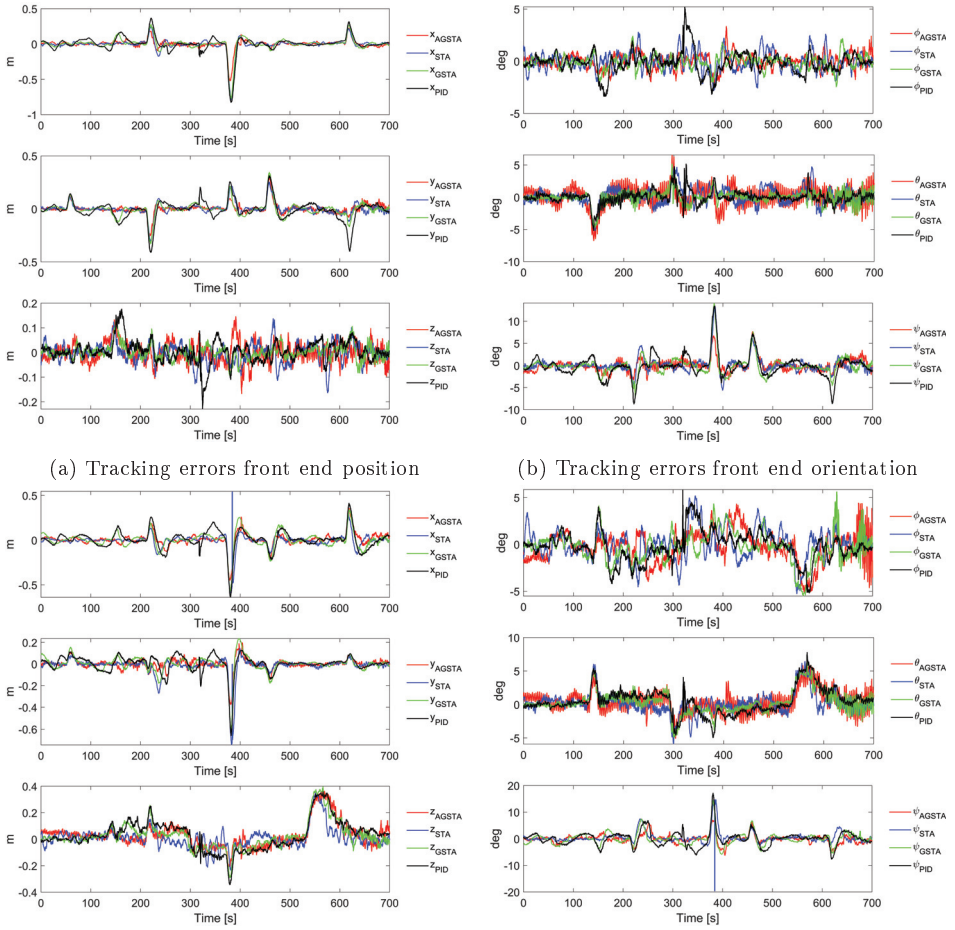
errors of the position and orientation for each task are given for all algorithms. In Table 7.13, the RMS values of the thruster forces, the derivatives of the thruster forces, the joint torques, and the derivatives of the joint torques are given for all algorithms.

Table 7.12: Experiments: Comparison of the tracking results for the task-priority case

Algorithm		RMSE		Maximum error	
		Task 1	Task 2	Task 1	Task 2
AGSTA	Position	0.0521	0.0760	0.3323	0.4011
	Orientation	0.0220	0.0296	0.1040	0.1054
STA	Position	0.0651	0.0839	0.4213	0.5638
	Orientation	0.0235	0.0321	0.1253	0.1831
GSTA	Position	0.0649	0.0915	0.4154	0.5304
	Orientation	0.0215	0.0328	0.1267	0.1560
PID	Position	0.0851	0.1020	0.4900	0.5545
	Orientation	0.0263	0.0366	0.1394	0.1797

Discussion

Figure 7.14a and Figure 7.14b show the commanded and actual position and orientation of the front end corresponding to task 1 when the GSTA with adaptive gains (6.7) is used. For task 1, we can see a small transient deviation at approximately 400 s in the yaw orientation. This transient error corresponds to a roll movement of the front end, which may indicate that movement in the roll direction interferes with movement in the yaw direction. Some other small oscillations are also evident, but these can simply be attributed to experimental uncertainties. The results for task 2 are shown in Figure 7.14c and Figure 7.14d. It is seen that the



(a) Tracking errors front end position

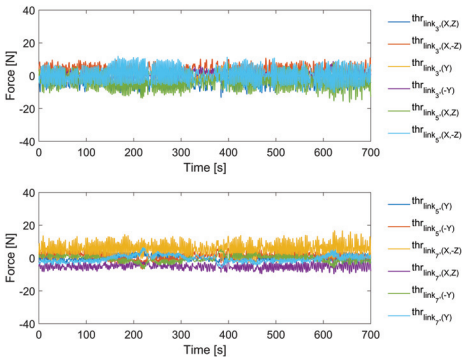
(b) Tracking errors front end orientation

(c) Tracking errors back end position

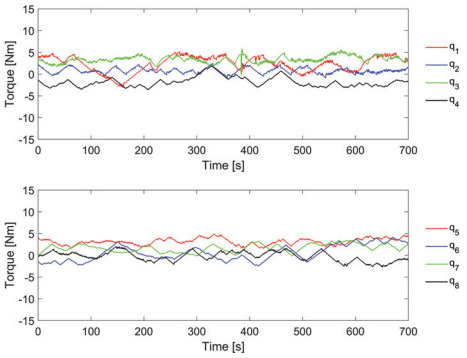
(d) Tracking errors back end orientation

Figure 7.16: Experiments: Tracking errors for the task-priority case

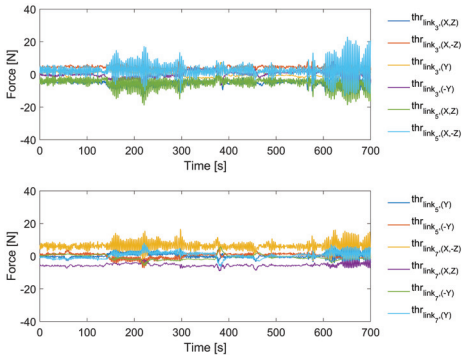
combination of the SRMTP method and the dynamic control law allows all tasks to be performed simultaneously, as indicated by Theorem 7.1 in combination with Theorem 7.3. Some transient deviations can also be observed for task 2. We see some small transient errors at approximately 200 s and 400 s in position and yaw that correspond to roll movements for task 1 and task 2. We also see small transient errors in the roll and pitch directions at approximately 550 s, which correspond to movement in the x-direction for task 1. The reason why we are unable to eliminate these deviations is because they are introduced by the SRMTP method, as discussed in Section 7.5.1. The experimental results therefore support the theoretical results, and we find that all set-point tasks are achieved. From Figure 7.14e, we can see that the forces used are well below 60 N, which is approximately the limit of the thrusters in the 2020 version of the robot. From Figure 7.14f, we can see that the



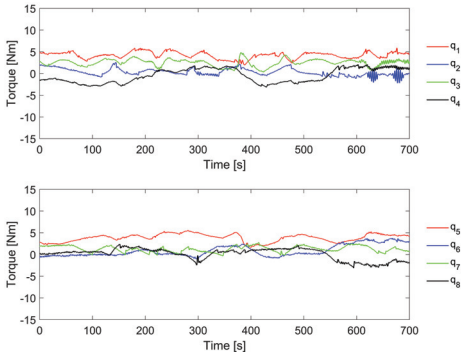
(a) Thruster forces τ_{thr} used for STA with adaptive gains



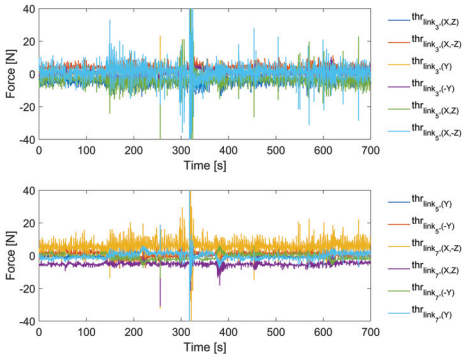
(b) Joint torques τ_q used for STA with adaptive gains



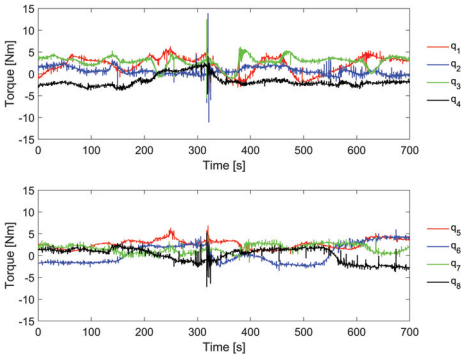
(c) Thruster forces τ_{thr} used for the GSTA



(d) Joint torques τ_q used for the GSTA

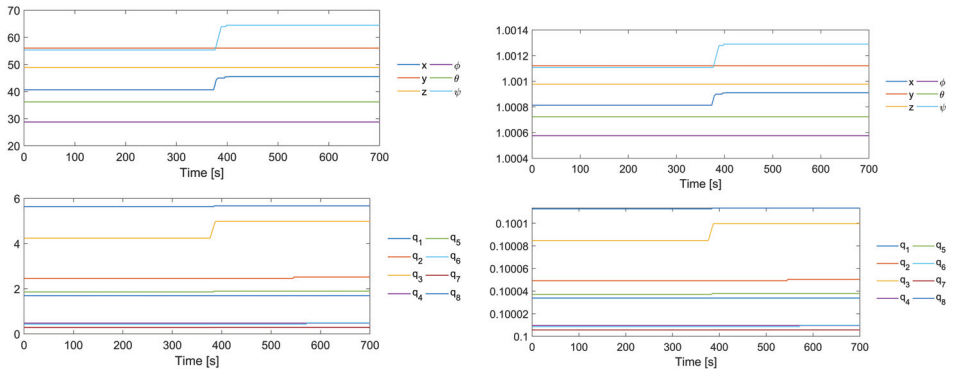


(e) Thruster forces τ_{thr} used for the PID controller



(f) Joint torques τ_q used for the PID controller

Figure 7.17: Experiments: Thruster forces τ_{thr} and joint torques τ_q used for the task-priority case



(a) Evolution of $\alpha(t)$ over time

(b) Evolution of $\beta(t)$ over time

Figure 7.18: Experiments: Evolution of the adaptive gains over time for the STA with adaptive gains for the task-priority case

Table 7.13: Experiments: Comparison of the control inputs for the task-priority case

Algorithm	**RMS**			
	τ_{thr}	$\dot{\tau}_{thr}$	τ_q	$\dot{\tau}_q$
AGSTA	5.0325	7.8897	2.1488	0.3898
STA	3.6394	5.7950	2.2093	0.3645
GSTA	3.6527	3.3307	2.2126	0.3398
PID	3.2469	42.1036	2.1960	5.3478

joint torques are smooth and below 16 Nm, which is the limit on the joint torques. The control inputs are therefore feasible. From Figure 7.15a and Figure 7.15b, we can see that the gains $k_1(t)$ and $k_2(t)$ converge to suitable values.

By comparing the simulation results in Figure 7.9 with the experimental results in Figure 7.14, we can see that the same transient errors introduced by the SRMTP method occur in both the simulations and experiments. In the experiments, we also see some more oscillations and larger deviations; however, this is to be expected because of measurement noise, the thruster dynamics, the joint dynamics and other unmodelled dynamics that inevitably affect the experiment. A comparison of the thruster forces and joint torques reveals that the greatest difference between the simulations and experiments is that there is a stronger tendency towards chattering in the control inputs in the experiments; in addition, less force is used in the simulations. Nevertheless, the experimental results and the simulation results are quite similar, indicating that the adaptive GSTA is indeed applicable for controlling the AIAUV.

From the position error plots in Figure 7.16, we can clearly see that there are some points that exhibit transient errors introduced by the SRMTP method. This behaviour is less clear in the orientation error plots, although some tendency

towards such a pattern is also seen here. It is also evident that the magnitudes of the transient errors are different for the different control algorithms, although it is not as easy to distinguish the different algorithms from each other as it is in the simulation results. We can see from Figure 7.16a that the adaptive GSTA yields the smallest position errors in x and y for task 1, although it is difficult to determine which algorithm performs the best in z. The STA with adaptive gains performs second best, while the performances of the GSTA and the PID controller are not easy to distinguish. In Figure 7.16b, it is also difficult to distinguish among the different algorithms based on the results in the roll and pitch directions for task 1, as the errors are quite small for all algorithms. For the yaw direction, the adaptive GSTA is seen to result in the smallest error for task 1, and the STA with adaptive gains yields the second best results; however, it is again difficult to distinguish between the GSTA and the PID controller. When we consider the position errors for task 2 in Figure 7.16c, we can see that for the x and y directions, the adaptive GSTA yields the smallest errors, while the results for z show little distinction. When we additionally consider the orientation errors for task 2 in Figure 7.16d, we see that here also, it is difficult to distinguish among the different algorithms based on the results for the roll and pitch directions; however, the adaptive GSTA is seen to result in the lowest error in yaw, although the other algorithms are still difficult to distinguish from each other in the yaw direction. These findings are also supported by Table 7.12, from which we can see that the adaptive GSTA results in the smallest RMSE values and maximum error values in both position and orientation for both tasks, with the exception of the RMSE orientation results for task 1, for which it is actually the GSTA that produces the lowest value. We can therefore conclude that the adaptive GSTA achieves the best tracking performance overall, i.e., the smallest errors. In terms of the RMSE values and the maximum errors, the STA with adaptive gains and the GSTA actually yield very similar results. The GSTA usually achieves the best results for task 1, while the STA with adaptive gains achieves the best results for task 2. By contrast, the PID controller exhibits the highest RMSE values for both tasks and the highest maximum error values for task 1. Meanwhile, the highest maximum error values for task 2 correspond to the STA with adaptive gains. In Figure 7.16c and Figure 7.16d, an outlier is evident in the results for the STA with adaptive gains; this outlier is probably the reason why the maximum error for task 2 is so high for the STA with adaptive gains. Regarding the thruster and torque use of the different algorithms, we can see from Figure 7.14e, Figure 7.14f and Figure 7.17 that there are no large differences in the amounts of force used. Nevertheless, Table 7.13 reveals some small differences among the algorithms. The RMS values of the thruster forces indicate that the adaptive GSTA uses the greatest amount of force, which may be the reason why the adaptive GSTA achieves the best overall performance. Nevertheless, the other algorithms also use very similar amounts of thruster force. The RMS values of the joint torques also are all very similar. From the RMS values of $\dot{\tau}_{thr}$, we can see that there is some chattering in the thruster inputs for all algorithms; however, the most rapid changes are observed with the PID controller. From the $\dot{\tau}_q$ results, we can see that there is no chattering in the torque control inputs of the SMC controllers, while the PID controller introduces some rapid changes.

By comparing Figure 7.11 with Figure 7.16, we can see that overall, the tracking

errors are smaller in the simulations, which can also be confirmed by comparing Table 7.6 with Table 7.12. This is to be expected because of measurement noise, the thruster dynamics, the joint dynamics and other unmodelled dynamics that inevitably affect the experiment. However, the GSTA actually shows better tracking performance in the experiments. This probably means that we could have found gains in the simulations that would have yielded better results. When we compare the forces used, by comparing Figure 7.9e, Figure 7.9f and Figure 7.12 with Figure 7.14e, Figure 7.14f and Figure 7.17, we can see evidence of more chattering in the control inputs during the experiments, which can also be confirmed by comparing Table 7.7 with Table 7.13. One other interesting observation is that while the adaptive GSTA uses less force in the simulations than in the experiments, the other algorithms actually use more force in the simulations than in the experiments. This may be because we tuned the gains of the controllers up in the simulations to obtain results similar to those obtained in the experiments.

One thing worth noting is that the results of this performance comparison between the STA with adaptive gains, the original GSTA and the PID controller are similar to those previously obtained in Chapter 5. This strengthens the notion from Chapter 6 that we in the results presented in Chapter 6 either chose very good gains for the GSTA and the PID controller, or did not choose an optimal starting point for the adaptive gains. Either way, we here see the advantage of the adaptive gains, as the results we obtain do not depend on our tuning capabilities. We also see that adding adaptive gains to the GSTA, thereby combining the theoretical advantages afforded by the GSTA and the practical advantages afforded by adaptive gains, improves the consequent tracking results and capabilities.

7.6 Chapter summary

In this chapter, we have proposed a combined kinematic and dynamic control approach for VMSs and presented an extended stability analysis for multiple set-point regulation tasks. The proposed method extends existing inverse kinematics control approaches to include dynamic control, and the analysis extends previous stability analyses and shows that the multiple set-point regulation tasks will converge asymptotically to zero without the strict requirement that the velocities are perfectly controlled. This novel approach thus avoids the assumption of perfect dynamic control that is common in kinematic stability analyses for vehicle manipulators. The applicability of the method is demonstrated through a simulation and experimental study of an AIAUV, which is a VMS with deep couplings between kinematics and dynamics. The simulation study includes performing three simultaneous tasks, and the results show that all the regulation tasks converge to their respective set-points. The simulation study also includes results from two standard control methods, namely, a PID controller and a feedback linearisation controller, and the results from two different AIAUVs to highlight the advantages and robustness we achieve when using SMC in the combined controller.

In the experimental study we performed two simultaneous tasks, where we controlled both the front end and the back end of the AIAUV in 6DOF, and the results showed that all the regulation tasks converged to their respective set-points. We

showed both simulation and experimental results, such that we could compare the experimental results with the ideal case of the simulations. We showed results for the adaptive GSTA, the STA with adaptive gains, the GSTA and a PID controller. It was the adaptive GSTA that gave the overall best tracking results, and we can therefore conclude that adding adaptive gains to the GSTA does indeed improve the consequent tracking results and capabilities. The simulation and experimental results were almost equally good. Of course, the tracking errors are larger in the experiments than in the simulations, but that is to be expected because of measurement noise, thruster dynamics, joint dynamics and other unmodelled dynamics that inevitably affect the experiments.

Chapter 8

Conclusions and Future Work

In this thesis, we developed trajectory tracking controllers for an articulated intervention autonomous underwater vehicle (AUV) in 6DOF based on sliding mode control. We used as little model knowledge as possible since an articulated intervention-AUV is subject to hydrodynamic and hydrostatic parameter uncertainties, uncertain thruster characteristics, unknown disturbances, and unmodelled dynamics. The stability of the proposed control approaches were analysed theoretically, and their performance was tested in simulations and experiments.

The first sliding mode control approach we proposed for trajectory tracking control for an articulated intervention-AUV was the super-twisting algorithm with adaptive gains. This algorithm is the most powerful second-order continuous sliding mode control algorithm, as it attenuates chattering and does not require the consideration of any conservative upper bound on the disturbance to maintain sliding because of the adaptive gains. We solved the trajectory tracking control problem for an articulated intervention-AUV in 6DOF using the super-twisting algorithm with adaptive gains. We considered both the case where velocity measurements were available and the case where they were not. When velocity measurements were not available, we used a higher-order sliding mode observer to estimate the linear and angular velocities. Furthermore, we proved the asymptotic convergence of the tracking errors when the control scheme including the higher-order sliding mode observer was used. Finally, we demonstrate the applicability of the presented control schemes with comprehensive simulation and experimental results.

The second sliding mode control approach proposed for trajectory tracking control for an articulated intervention-AUV was the generalized super-twisting algorithm. The generalized super-twisting algorithm is an extension of the super-twisting algorithm that provides finite-time convergence in the case when both the perturbations and control coefficients are state- and time-dependent and the control coefficients are uncertain, which is essential for robust control of an articulated intervention-AUV. We solved the trajectory tracking control problem for an articulated intervention-AUV in 6DOF using the generalized super-twisting algorithm. Furthermore, we proved the asymptotic convergence of the tracking errors. We also considered the case where velocity measurements were unavailable by solving the tracking control problem using the generalized super-twisting algorithm in com-

bination with a higher-order sliding mode observer. Furthermore, we proved the asymptotic convergence of the tracking errors when the higher-order sliding mode observer was included. Moreover, we presented comprehensive simulation and experimental results that validated and demonstrated the applicability of both control schemes.

In the subsequent chapter, we compared the previously proposed tracking control laws. Specifically, we compared the control law using the super-twisting algorithm with adaptive gains and the control law using the generalized super-twisting algorithm. We also compared the control laws in combination with a higher-order sliding mode observer. Additionally, we solved the tracking problem using a PID controller to evaluate how the sliding mode control algorithms perform compared to a linear controller. In both the simulations and experiments, the super-twisting algorithm with adaptive gains provided the best overall tracking performance, but the generalized super-twisting algorithm produced similar tracking results. Thus, the fact that we had to tune the generalized super-twisting algorithm manually was probably the reason we achieved lower tracking performance with this algorithm. Therefore, in practice, the super-twisting algorithm with adaptive gains is better than the generalized super-twisting algorithm, but the generalized super-twisting algorithm does have some theoretical advantages as it is proven to provide global finite-time stability for a larger class of systems. These additional theoretical properties made it possible to prove that the closed-loop system was uniformly globally asymptotically stable without the higher-order sliding mode observer when the generalized super-twisting algorithm was used. This proof was not possible for the super-twisting algorithm with adaptive gains.

Based on the results of the comparison, we developed an adaptive generalized super-twisting algorithm for a class of systems whose perturbations and uncertain control coefficients are time- and state-dependent, i.e., we combined the best properties of the super-twisting algorithm with adaptive gains and the generalized super-twisting algorithm. The proposed approach uses dynamically adapted control gains to ensure global finite-time convergence. The advantage of adaptive gains is that no conservative upper bound must be considered on the perturbations and control coefficients to maintain sliding. We proved that the resulting closed-loop system was globally finite-time stable. Furthermore, we demonstrated the effectiveness of the proposed adaptive generalized super-twisting algorithm by solving the trajectory tracking control problem for an articulated intervention-AUV in 6DOF. Additionally, we showed that the adaptive generalized super-twisting algorithm makes the tracking errors of the articulated intervention-AUV converge asymptotically to zero. We also presented simulation and experimental results that validated and demonstrated the applicability of the proposed control law.

Finally, to utilize the redundancy of this highly flexible underwater vehicle, we proposed a combined kinematic and dynamic control approach for vehicle-manipulator systems. The main idea is to combine the singularity-robust multiple task-priority framework with a robust sliding mode controller while simultaneously ensuring that the task errors remain bounded. *Any* controller can be used, as long as it is able to make the velocity vector converge to the velocity reference vector in finite time. The output from the singularity-robust multiple task-priority inverse kinematic controller was chosen as the reference vector. This novel approach

allowed us to analyse the stability properties of the kinematic and dynamic subsystems together in the presence of model uncertainty while retaining the possibility of solving multiple tasks simultaneously. The finite-time convergence of the sliding mode controller allowed us to show that multiple set-point regulation tasks will converge asymptotically to zero without the strict requirement that the velocities be perfectly controlled. This novel approach thus avoids the assumption of perfect dynamic control that is common in kinematic stability analyses for robot manipulators and vehicle manipulator systems. We provided two examples of sliding mode controllers that are able to make the velocity vector converge to the velocity reference vector in finite time: a first-order sliding mode controller and the adaptive generalized super-twisting algorithm. The applicability of the proposed method was illustrated via a simulation study, where the primary task was to control the position and orientation of the centre link of an articulated intervention-AUV and the secondary tasks were to control the orientations of the front end and back end of the articulated intervention-AUV, and an experimental study, where the primary task was to control the position and orientation of the front end of the articulated intervention-AUV and the secondary task was to control the position and orientation of the back end of the articulated intervention-AUV.

Future work will include extending the adaptive generalized super-twisting algorithm to multiple-input multiple-output systems. Methods where sliding mode control can be integrated or combined with an optimisation method should also be investigated. The reason for this is that sliding mode control introduces considerable energy into the system and, especially for an articulated intervention-AUV, it is often necessary to minimise or limit the control input since the vehicle has a limited amount of thrust and power. Energy-efficient robust control should therefore be investigated, as it is an important future step for conducting autonomous underwater operations.

Appendix A

Stability Definitions and Theorems

A.1 Notation

- $\mathbb{R}_+ = \{x \in \mathbb{R} : x > 0\}$
- $\mathcal{B}(r) = \{x \in \mathbb{R}^n : ||x|| < r\}$ is an open ball of the radius $r \in \mathbb{R}_+$ with the center at the origin.
- $\{y\} + \mathcal{B}(\varepsilon)$ is an open ball of the radius $\varepsilon > 0$ with center at $y \in \mathbb{R}^n$.
- $\text{int}(\Omega)$ is the interior of a set $\Omega \subseteq \mathbb{R}^n$.
- For a given number $r \in \mathbb{R}$ and a given positive definite function $W : \mathbb{R}^n \to \mathbb{R}$ defined on Ω let us introduce the set $\Pi(W, r) = \{x \in \Omega : W(x) < r\}$ which is called the level set of the function W.

A.2 Definitions and propositions

Definition A.1 (Cross-product operator, [30, Definition 2.2])**.** The vector cross-product $\times$ is defined by

$$\lambda \times a := S(\lambda)a \tag{A.1}$$

where $S \in SS(3)$ is defined as

$$S(\lambda) = -S^T(\lambda) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix}, \qquad \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \tag{A.2}$$

A.2.1 Stability

Consider the non-autonomous system

$$\dot{x} = f(t, x) \tag{A.3}$$

where $f : [0, \infty) \times D \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in x on $[0, \infty) \times D$ and $D \subset \mathbb{R}^n$ is a domain that contains the origin $x = 0$. Suppose that the origin $x = 0$ is an equilibrium of the system (A.3). We define the following.

Definition A.2 (Stability, [40, Definition 4.4]). The equilibrium point $x = 0$ of (A.3) is

- stable if, for each $\varepsilon > 0$, there is $\delta = \delta(\varepsilon, t_0) > 0$ such that

$$||x(t_0)|| < \delta \Rightarrow ||x(t)|| < \varepsilon, \qquad \forall\, t \geq t_0 \geq 0 \tag{A.4}$$

- uniformly stable if, for each $\varepsilon > 0$, there is $\delta = \delta(\varepsilon) > 0$, independent of t_0, such that (A.4) is satisfied.
- unstable if it not stable.
- asymptotically stable if it is stable and there is a positive constant $c = c(t_0)$ such that $x(t) \to 0$ as $t \to \infty$, for all $||x(t_0)|| < c$.
- uniformly asymptotically stable if it is uniformly stable and there is a positive constant c, independent of t_0, such that for all $||x(t_0)|| < c$, $x(t) \to 0$ as $t \to \infty$, uniformly in t_0; that is, for each $\eta > 0$, there is $T = T(\eta) > 0$ such that

$$||x(t)|| < \eta, \qquad \forall\, t \geq t_0 + T(\eta), \quad \forall\, ||x(t_0)|| < c \tag{A.5}$$

- uniformly globally asymptomatic stable if it is uniformly stable, $\delta(\varepsilon)$ can be chosen to satisfy $\lim_{\varepsilon \to \infty} \delta(\varepsilon) = \infty$, and, for each pair of positive numbers η and c, there is $T = T(\eta, c) > 0$ such that

$$||x(t)|| < \eta, \qquad \forall\, t \geq t_0 + T(\eta, c), \quad \forall\, ||x(t_0)|| < c \tag{A.6}$$

Definition A.3 (Exponential stability, [40, Definition 4.5]). The equilibrium point $x = 0$ of (A.3) is exponentially stable if there exist a positive constants c, k and λ such that

$$||x(t)|| \leq k||x(t_0)||e^{-\lambda(t-t_0)}, \qquad \forall\, ||x(t_0)|| < c \tag{A.7}$$

and globally exponentially stable if (A.7) is satisfied for any initial state $x(t_0)$.

Proposition A.1 ([60, Proposition 2]). *Let a set-valued function $F : \mathbb{R}^n \to \mathbb{R}^n$ be defined and upper-semi-continuous in $\mathbb{R}^n$. Let $F(x)$ be non-empty, compact and convex for any $x \in \mathbb{R}^n$. If the origin of the system*

$$\dot{x} \in F(x) \tag{A.8}$$

is asymptotically stable then it is uniformly asymptotically stable.

A.2.2 Finite-time stability

Consider the extended differential inclusion

$$\dot{x} \in F(t, x), \qquad t \in \mathbb{R} \tag{A.9}$$

for $t > t_0$ with an initial condition

$$x(t_0) = x_0, \tag{A.10}$$

where $x_0 \in \mathbb{R}^n$ is given.

Definition A.4 (Lyapunov stability [60, Definition 5])**.** The origin of the system (A.9) is said to be Lyapunov stable if $\forall \varepsilon \in \mathbb{R}_+$ and $\forall t_0 \in \mathbb{R}$ there exists $\delta = \delta(\varepsilon, t_0) \in \mathbb{R}_+$ such that $\forall x_0 \in \mathcal{B}(\delta)$

- any solution $x(t, t_0, x_0)$ of Cauchy problem (A.9), (A.10) exists for $t > t_0$;
- $x(t, t_0, x_0) \in \mathcal{B}(\varepsilon)$ for $t > t_0$.

If the function δ does not depend on t_0 then the origin is called uniformly Lyapunov stable.

Definition A.5 (Finite-time attractivity, [60, Definition 11])**.** The origin of the system (A.9) is said to be finite-time attractive if $\forall t_0 \in \mathbb{R}$ there exists a set $\mathcal{V}(t_0) \subseteq \mathbb{R}^n : 0 \in \text{int}(\mathcal{V}(t_0))$ such that $\forall x_0 \in \mathcal{V}(t_0)$

- any solution $x(t, t_0, x_0)$ of Cauchy problem (A.9), (A.10) exists for $t > t_0$;
- $T(t_0, x_0) < +\infty$ for $x_0 \in \mathcal{V}(t_0)$ and for $t_0 \in \mathbb{R}$.

Definition A.6 (Uniform finite-time attractivity, [60, Definition 13])**.** The origin of the system (A.9) is said to be uniformly finite-time attractive if it is finite-time attractive with a time-invariant attraction domain $\mathcal{V} \subseteq \mathbb{R}^n$ such that the settling time function $T(t_0, x_0)$ is locally bounded on $\mathbb{R} \times \mathcal{V}$ uniformly on $t_0 \in \mathbb{R}$, i.e. for any $y \in \mathcal{V}$ there exist $\varepsilon \in \mathbb{R}_+$ such that $\{y\} + \mathcal{B}(\varepsilon) \subseteq \mathcal{V}$ and $\sup_{t_0 \in \mathbb{R}, x_0 \in \{y\}+\mathcal{B}(\varepsilon)} T(t_0, x_0) < +\infty$.

Definition A.7 (Finite-time stability, [60, Definition 12])**.** The origin of the system (A.9) is said to be finite-time stable if it is Lyapunov stable and finite-time attractive. If $\mathcal{V}(t_0) = \mathbb{R}^n$ then the origin of (A.9) is called globally finite-time stable.

Definition A.8 (Uniform finite-time stability, [60, Definition 14])**.** The origin of the system (A.9) is said to be uniformly finite-time stable if it is uniformly Lyapunov stable and uniformly finite-time attractive. The origin of (A.9) is called globally uniformly finite-time stable if $\mathcal{V} = \mathbb{R}^n$.

Proposition A.2 ([60, Proposition 3])**.** *If the origin of the system (A.9) is finite-time stable then it is asymptotically stable and $x(t, t_0, x_0) = 0$ for $t > t_0 + T_0(t_0, x_0)$.*

A.2.3 Boundedness

Definition A.9 (Boundedness, [40, Definition 4.6])**.** The solutions of (A.3) is

- uniformly bounded if there exsist a positive constant c, independent of $t_0 \geq 0$, and for any $a \in (0, c)$, there is $\beta = \beta(a) > 0$, independent of t_0, such that
$$||x(t_0)|| \leq a \Rightarrow ||x(t)|| \leq \beta, \qquad \forall\, t \geq t_0 \tag{A.11}$$
- uniformly globally bounded if (A.11) holds for arbitrarily large a.
- uniformly ultimately bounded with ultimate bound b if there exist positive constants b and c, independent of $t_0 \geq 0$, and for every $a \in (0, c)$, there is $T = T(a, b) \geq 0$, independent of t_0, such that
$$||x(t_0)|| \leq a \Rightarrow ||x(t)|| \leq \beta, \qquad \forall\, t \geq t_0 + T \tag{A.12}$$
- uniformly globally ultimately bounded if (A.12) holds for arbitrarily large a.

A.2.4 Adaptive gains

Let the adaptive gains be defined as

$$\dot{\alpha} = \begin{cases} \omega_1 \sqrt{\frac{\gamma_1}{2}} & \text{if } \sigma \neq 0 \\ 0, & \text{if } \sigma = 0 \end{cases} \tag{A.13a}$$

$$\beta = 2\varepsilon\alpha + \lambda + 4\varepsilon^2 \tag{A.13b}$$

where ε, λ, γ_1 and ω_1 are positive constants and σ is the sliding surface.

Proposition A.3 ([69, Proposition 1]). *The adaptive gains $\alpha(t)$ and $\beta(t)$ defined by (A.13) are bounded.*

A.3 Theorems and lemmas

A.3.1 Exponential stability

Theorem A.4 (Exponential stability, [40, Theorem 4.10]). *Let $x = 0$ be an equilibrium point for (A.3) and $D \subset \mathbb{R}^n$ be a domain containing $x = 0$. Let $V : [0, \infty) \times D \to \mathbb{R}$ be a continuously differentiable function such that*

$$\begin{aligned} k_1||x||^a \leq V(t,x) \leq k_2||x||^a \\ \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t,x) \leq -k_3||x||^a \end{aligned} \tag{A.14}$$

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

A.3.2 Boundedness

Theorem A.5 (Boundedness, [40, Theorem 4.18]). *Let $D \subset \mathbb{R}^n$ be a domain that contains the origin and $V : [0, \infty) \times D \to \mathbb{R}$ be a continuously differentiable function such that*

$$\begin{aligned} \alpha_1(||x||) \leq V(t,x) \leq \alpha_2(||x||) \\ \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t,x) \leq -W_3(x), \quad \forall\, ||x|| \geq \mu > 0 \end{aligned} \tag{A.15}$$

$\forall\, t \geq 0$ and $\forall\, x \in D$, where α_1 and α_2 are class $\mathcal{K}$ functions and $W_3(x)$ is continuous positive definite function. Take $r > 0$ such that $B_r \subset D$ and suppose that

$$\mu < \alpha_2^{-1}(\alpha_1(r)) \tag{A.16}$$

Then there exists a class $\mathcal{KL}$ function β and for every initial state $x(t_0)$, satisfying $||x(t_0)|| \leq \alpha_2^{-1}(\alpha_1(r))$, there is $T \geq 0$ (dependent on $x(t_0)$ and μ) such that the solution of (A.3) satisfies

$$\begin{aligned} ||x(t)|| \leq \beta(||x(t_0)||, t - t_0), \quad \forall \quad t_0 \leq t \leq t_0 + T \\ ||x(t)|| \leq \alpha_1^{-1}(\alpha_2(\mu)), \quad \forall \quad t \geq t_0 + T \end{aligned} \tag{A.17}$$

Moreover, if $D = \mathbb{R}^n$ and α_1 belong to class $\mathcal{K}_\infty$, then (A.17) hold for any initial state $x(t_0)$, with no restrictions on how large μ is.

A.3.3 Finite-time stability

Theorem A.6 (Uniform finite-time stability, [60, Theorem 12]). *Let a continuous function $V : \mathbb{R}^n \to \mathbb{R}$ be proper on an open non-empty set $\Omega \subseteq \mathbb{R}^n : 0 \in int(\Omega)$ and*

$$D_{F(t,x)}V(x) \leq -rV^\rho(x), \qquad t > t_0, \quad x \in \Omega, \tag{A.18}$$

where $r \in \mathbb{R}_+$ and $0 < \rho < 1$. Then the origin of the system (A.9) is uniformly finite-time stable with an attraction domain $\mathcal{U}$ defined by

$$\mathcal{U} = \Pi(V, \lambda(h)) \cap \mathcal{B}(h) \tag{A.19}$$

where $\lambda(h) = inf_{x\in\mathbb{R}^n:||x||=h}\underline{V}(x)$ (since $V(x)$ is proper there exist a continuous positive definite function $\underline{V}(x)$ such that $\underline{V}(x) \leq V(x)\,\forall\, x \in \Omega$) and $h \leq sup_{r\in\mathbb{R}_+:\mathcal{B}(r)\subseteq\Omega} r$ and the settling time function $T(\cdot)$ is estimated as follows:

$$T(x_0) \leq \frac{V^{1-\rho}(x_0)}{r(1-\rho)} \quad \text{for } x_0 \in \mathcal{U}. \tag{A.20}$$

A.3.4 Cascaded systems

Consider the cascaded system

$$\sum\nolimits_1 \Big\{ \dot{x}_1 = f_1(t, x_1) + g(t, x)x_2 \tag{A.21a}$$

$$\sum\nolimits_2 \Big\{ \dot{x}_2 = f_2(t, x_2) \tag{A.21b}$$

where $x_1 \in \mathbb{R}^n$, $x_2 \in \mathbb{R}^m$, $x := [x_1^T, x_2^T]^T$. We assume that the functions $f_1(t, x_1)$, $f_2(t, x_2)$ and $g(t, x)$ are continuous in their arguments, locally Lipschitz in x, uniform in t, and $f_1(t, x_1)$ is continuously differentiable in both arguments. We also assume that there exists a non-decreasing functions $G(\cdot)$ such that

$$|g(t, x)| \leq G(|x|). \tag{A.22}$$

Lemma A.7 (Cascaded systems, [53, Lemma 2.1]). *The cascade (A.21) is uniformly globally asymptomatic stable if and only if the systems*

$$\dot{x}_1 = f_1(t, x_1) \tag{A.23}$$

and (A.21b) are uniformly globally asymptomatic stable and the solutions of (A.21) are uniformly globally bounded.

A.3.5 Comparison lemma

Lemma A.8 (Comparison lemma, [40, Lemma 3.4]). *Consider the scalar differential equation*

$$\dot{u} = f(t, u), \qquad u(t_0) = u_0 \tag{A.24}$$

where $f(t, u)$ is continuous in t and locally Lipschitz in u, for all $t \geq 0$ and all $u \in J \subset R$. Let $[t_0, T)$ (T could be infinity) be the maximal interval of existence of the solution $u(t)$, and suppose $u(t) \in J$ for all $t \in [t_0, T)$. Let $v(t)$ be a continuous function whose upper right-hand derivative $D^+v(t)$ satisfies the differential inequality

$$D^+v(t) \leq f(t, v(t)), \qquad v(t_0) \leq u_0 \tag{A.25}$$

with $v(t) \in J$ for all $t \in [t_0, T)$. Then, $v(t) \leq u(t)$ for all $t \in [t_0, T)$.

References

[1] Marine cybernetics laboratory (mc-lab). 2018. https://www.ntnu.edu/imt/lab/cybernetics.

[2] Qualisys-motion capture systems. 2018. https://www.qualisys.com.

[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, Cham, Switzerland, 3rd edition, 2014.

[5] G. Antonelli and S. Chiaverini. Singularity-free regulation of underwater vehicle-manipulator systems. In *Proc. American Control Conference*, pages 399–403, Philadelphia, Pennsylvania, June 24-26 1998.

[6] G. Antonelli, P. Di Lillo, and C. Natale. Modeling errors analysis in inverse dynamics approaches within a task-priority framework. In *Proc. IEEE Conference on Control Technology and Applications*, pages 553–558, Copenhagen, Denmark, Aug. 21-24 2018.

[7] E. A. Basso and K. Y. Pettersen. Task-priority control of redundant robotic systems using control lyapunov and control barrier function based quadratic programs. In *Proc. 21th IFAC World Congress*, Berlin, Germany, Jul. 11-17 2020.

[8] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Trajectory tracking for an articulated intervention AUV using a super-twisting algorithm in 6DOF. *IFAC PapersOnLine*, 51(29):311–316, Sep. 10-12, 2018. In: Proc. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, Opatija, Croatia.

[9] I.-L. G. Borlaug, J. T. Gravdahl, J. Sverdrup-Thygeson, K. Y. Pettersen, and A. Loría. Trajectory tracking for underwater swimming manipulator using a super twisting algorithm. *Asian Journal of Control*, 21(1):208–223, 2019.

[10] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Tracking control of an articulated intervention AUV in 6DOF using the generalized super-twisting

algorithm. In *Proc. American Control Conference*, pages 5705–5712, Philadelphia, USA, Jul. 10-12, 2019.

[11] I.-L. G. Borlaug, J. Sverdrup-Thygeson, K. Y. Pettersen, and J. T. Gravdahl. Combined kinematic and dynamic control of an underwater swimming manipulator. *IFAC PapersOnLine*, 52(21):8–13, Sep. 18-20, 2019. In: Proc. 12th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, Daejeon, Korea.

[12] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. The generalized super-twisting algorithm with adaptive gains. In *Proc. European Control Conference*, pages 1624–1631, Saint Petersburg, Russia, Mai 12-15, 2020. Best Student Paper Award Finalist.

[13] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. The generalized super-twisting algorithm with adaptive gains. *Submitted to International Journal of Robust and Nonlinear Control*, 2020. https://www.dropbox.com/s/8wwcixxpop3skeb/IJRNC_2020_fv.pdf?dl=0.

[14] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Combined kinematic and dynamic control of vehicle-manipulator systems. *Mechatronics*, 69:102380, Aug. 2020.

[15] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Comparison of two second-order sliding mode control algorithms for an articulated intervention AUV: Theory and experimental results. *Submitted to Ocean Engineering*, 2020. https://www.dropbox.com/s/ytm0o2x3toqk9ip/OE_2020_fv.pdf?dl=0.

[16] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl. Tracking control of an articulated intervention autonomous underwater vehicle in 6DOF using generalized super-twisting: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 2020.

[17] J. A. Burton and A. S. I. Zinober. Continuous approximation of variable structure control. *International Journal of Systems Science*, 17(6):875–885, 1986.

[18] I. Castillo, L. M. Fridman, and J. A. Moreno. Super-Twisting Algorithm in presence of time and state dependent perturbations. *International Journal of Control*, 91(11):2535–2548, 2018.

[19] A. Chalanga, S. Kamal, L. M. Fridman, B. Bandyopadhyay, and J. A. Moreno. Implementation of super-twisting control: Super-twisting and higher order sliding-mode observer-based approaches. *IEEE Transactions on Industrial Electronics*, 63(6):3677–3685, 2016.

[20] 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.

[21] R. Cui, X. Zhang, and D. Cui. Adaptive sliding-mode attitude control for autonomous underwater vehicles with input nonlinearities. *Ocean Engineering*, 123:4554, 2016.

[22] M. W. Dannigan and G. T. Russell. Evaluation and reduction of the dynamic coupling between a manipulator and an underwater vehicle. *IEEE Journal of Oceanic Engineering*, 23(3):260–273, 1998.

[23] A. Dietrich, C. Ott, and A. Albu-Schäffer. An overview of null space projections for redundant, torque-controlled robots. *The International Journal of Robotics Research*, 34(11):1385–1400, 2015.

[24] S. Dubowsky and E. Papadopoulos. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *IEEE Transactions on Robotics and Automation*, 9(5):531–543, 1993.

[25] A. Ferrara, G. Incremona, and E. Regolin. Optimization-based adaptive sliding mode control with application to vehicle dynamics control. *International Journal of Robust and Nonlinear Control*, 29(3):550–564, 2019.

[26] A. Ferrara, G. P. Incremona, and B. Sangiovanni. Tracking control via switched integral sliding mode with application to robot manipulators. *Control Engineering Practice*, 90:257–266, 2019.

[27] O. E. Fjellstad and T. I. Fossen. Singularity-free tracking of unmanned underwater vehicles in 6 DOF. In *Proc. 33rd IEEE Conference on Decision and Control*, pages 1128–1133, Lake Buena Vista, Florida, Dec. 14-16 1994.

[28] T. I. Fossen. Adaptive macro-micro control of nonlinear underwater robotic systems. In *Proc. 5th International Conference on Advanced Robotics*, pages 1569–1572, Pisa, Italy, June 19-22 1991.

[29] T. I. Fossen. *Marine Control Systems - Guidance, Navigation, and Control of Ships, Rigs, and Underwater Vehicles*. Marine Cybernetics, Trondheim, 2002.

[30] T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester, UK: John Wiley & Sons, Ltd, 2011.

[31] T. I. Fossen and S. Sagatun. Adaptive control of nonlinear underwater robotic systems. *Modeling, Identification and Control*, 12(2):95–105, 1991.

[32] P. J. From, J. T. Gravdahl, and K. Y. Pettersen. *Vehicle-Manipulator Systems: Modeling for Simulation, Analysis, and Control*. Advances in Industrial Control. Springer London, London, 2014.

[33] T. Gonzalez, J. A. Moreno, and L. Fridman. Variable gain super-twisting sliding mode control. *IEEE Transactions on Automatic Control*, 57(8):2100–2105, 2012.

[34] J. Guerrero, J. Torres, V. Creuze, and A. Chemori. Trajectory tracking for autonomous underwater vehicle: An adaptive approach. *Ocean Engineering*, 172:511–522, 2019.

[35] E. Guzmàn and J. A. Moreno. Super-twisting observer for second-order systems with time-varying coefficient. *IET Control Theory and Applications*, 9 (4):553–562, 2015.

[36] E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl. A waypoint guidance strategy for underwater snake robots. In *Proc. IEEE 22nd Mediterranean Conference on Control and Automation*, pages 1512–1519, Palermo, Italy, 2014.

[37] E. Kelasidi, K. Y. Pettersen, P. Liljebäck, and J. T. Gravdahl. Locomotion efficiency of underwater snake robots with thrusters. In *Proc. International Symposium on Safety, Security and Rescue Robotics*, pages 174–181, Lausanne, Switzerland, Oct. 23-27 2016.

[38] E. Kelasidi, P. Liljeback, K. Y. Pettersen, and J. T. Gravdahl. Integral line-of-sight guidance for path following control of underwater snake robots: Theory and experiments. *IEEE Transactions on Robotics*, 33(3):610–628, 2017.

[39] E. Kelasidi, K. Y. Pettersen, J. T. Gravdahl, S. Strømsøyen, and A. J. Sørensen. Modeling and Propulsion Methods of Underwater Snake Robots. In *Proc. 1st IEEE Conference on Control Technology and Applications*, pages 819–826, Kohala Coast, Hawaii, Aug. 27-30 2017.

[40] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, N.J, 3rd ed. edition, 2002.

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

[42] A. M. Kohl, E. Kelasidi, A. Mohammadi, M. Maggiore, and K. Y. Pettersen. Planar maneuvering control of underwater snake robots using virtual holonomic constraints. *Bioinspiration & Biomimetics*, 11(6):065005, 2016.

[43] A. M. Kohl, K. Y. Pettersen, E. Kelasidi, and J. T. Gravdahl. Planar path following of underwater snake robots in the presence of ocean currents. *IEEE Robotics and Automation Letters*, 1(1):383–390, 2016.

[44] A. M. Kohl, S. Moe, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl. Set-based path following and obstacle avoidance for underwater snake robots. In *Proc. 2017 IEEE International Conference on Robotics and Biomimetics*, pages 1206–1213, Parisian Macao, China, Dec. 5-8 2017.

[45] A. M. Kohl, K. Y. Pettersen, and J. T. Gravdahl. Velocity and orientation control of underwater snake robots using absolute velocity feedback. In *Proc. 1st IEEE Conference on Control Technology and Applications*, pages 752–759, Kohala Coast, Hawaii, Aug. 27-30 2017.

[46] K. Kumari, A. Chalanga, and B. Bandyopadhyay. Implementation of Super-Twisting Control on Higher Order Perturbed Integrator System using Higher Order Sliding Mode Observer. In *Proc. 10th IFAC Symposium on Nonlinear Control Systems*, pages 873–878, Monterey, CA, USA, Aug. 23-25 2016.

[47] A. Levant. Sliding order and sliding accuracy in sliding mode control. *International Journal of Control*, 58(6):1247–1263, 1993.

[48] A. Levant. Robust exact differentiation via sliding mode technique. *Automatica*, 34(3):379–384, 1998.

[49] A. Levant. Higher-order sliding modes, differentiation and output-feedback control. *International Journal of Control*, 76(9-10):924–941, 2003.

[50] A. Levant. Homogeneity approach to high-order sliding mode design. *Automatica*, 41(5):823–830, 2005.

[51] P. Liljebäck and R. Mills. Eelume: A flexible and subsea resident IMR vehicle. In *OCEANS 2017*, Aberdeen, UK, Jun. 19-22 2017.

[52] S. Liu, Y. Liu, and N. Wang. Nonlinear disturbance observer-based backstepping finite-time sliding mode tracking control of underwater vehicles with system uncertainties and external disturbances. *Nonlinear Dynamics*, 88(1): 465–476, 2017.

[53] A. Loría and E. Panteley. *Cascaded nonlinear time-varying systems: Analysis and design*, volume 311 of *Lecture Notes in Control and Information Sciences*. Springer International Publishing, Cham, Switzerland, 2005.

[54] 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, 2016.

[55] S. Moe, J. T. Gravdahl, and K. Y. Pettersen. Set-based control for autonomous spray painting. *IEEE Transactions on Automation Science and Engineering*, 15(4):1785–1796, 2018.

[56] J. A. Moreno. A linear framework for the robust stability analysis of a generalized super-twisting algorithm. In *Proc. 6th International Conference on Electrical Engineering, Computing Science and Automatic Control*, pages 1–6, 2009.

[57] J. A. Moreno. Lyapunov function for Levant's Second Order Differentiator. In *Proc. IEEE 51st IEEE Conference on Decision and Control*, pages 6448–6453, Maui, Hawaii, USA, Dec 10-13 2012.

[58] J. A. Moreno. Lyapunov approach for analysis and design of second order sliding mode algorithms. In *Sliding Modes after the First Decade of the 21st Century: State of the Art*, volume 412 of *Lecture Notes in Control and Information Sciences*, pages 113–149. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[59] K. Y. Pettersen. Snake robots. *Annual Reviews in Control*, 44:19–44, 2017.

[60] A. Polyakov and L. M. Fridman. Stability notions and lyapunov functions for sliding mode control systems. *Journal of the Franklin Institute*, 351(4): 1831–1865, 2014.

[61] S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, 2007.

[62] E. Rezapour, K. Y. Pettersen, P. Liljebäck, and J. T. Gravdahl. Differential geometric modelling and robust path following control of snake robots using sliding mode techniques. In *Proc. 2014 IEEE International Conference on Robotics and Automation*, pages 4532–4539, Hong Kong, China, May 31 - June 7, 2014.

[63] M. Rubagotti, A. Estrada, F. Castanos, A. Ferrara, and L. Fridman. Integral sliding mode control for nonlinear systems with matched and unmatched perturbations. *IEEE Transactions on Automatic Control*, 56(11):2699–2704, 2011.

[64] S. Salazar and R. Lozano. Adaptive non-singular terminal sliding mode control for an unmanned underwater vehicle: Real-time experiments. *International Journal of Control, Automation, and Systems*, 18(3):615–628, 2020.

[65] H. M. Schmidt-Didlaukies, A. J. Sørensen, and K. Y. Pettersen. Modeling of Articulated Underwater Robots for Simulation and Control. In *Proc. 2018 IEEE/OES Autonomous Underwater Vehicles*, pages 1–7, Porto, Portugal, Nov. 6-9, 2018.

[66] Y. Shtessel, M. Taleb, and F. Plestan. A novel adaptive-gain super-twisting sliding mode controller: Methodology and application. *Automatica*, 48(5):759–769, 2012.

[67] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant. *Sliding Mode Control and Observation*. Control Engineering. Birkhäuser, New York, NY, 2014.

[68] Y. B. Shtessel, I. A. Shkolnikov, and A. Levant. Smooth second-order sliding modes: Missile guidance application. *Automatica*, 43(8):1470–1476, 2007.

[69] Y. B. Shtessel, J. A. Moreno, F. Plestan, L. M. Fridman, and A. S. Poznyak. Super-twisting adaptive sliding mode control: A Lyapunov design. In *Proc. 49th IEEE Conference on Decision and Control*, pages 5109–5113, Atlanta, GA, USA, Dec. 15-17, 2010.

[70] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer, Berlin, 2008.

[71] B. Siciliano and J. J. E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *Proc. 5th International Conference on Advanced Robotics Robots in Unstructured Environments*, pages 1211–1216, 1991.

[72] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, M. J. Grimble, and M. A. Johnson. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. London: Springer London, London, 2009.

[73] E. Simetti and G. Casalino. A novel practical technique to integrate inequality control objectives and task transitions in priority based control. *Journal of Intelligent & Robotic Systems*, 84(1):877–902, 2016.

[74] J.-J. Slotine and W. Li. *Applied Nonlinear Control,*. Prentice-Hall, Englewood Cliffs, N.J, 1st ed. edition, 1991.

[75] S. Soylu, B. J. Buckham, and R. P. Podhorodeski. A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation. *Ocean Engineering*, 35(16):1647–1659, 2008.

[76] M. W. Spong. Robot modeling and control, 2006.

[77] J. Sverdrup-Thygeson, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl. A control framework for biologically inspired underwater swimming manipulators equipped with thrusters. *IFAC PapersOnLine*, 49(23):89–96, Sep. 13-16 2016. In: Proc. 10th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, Trondheim, Norway.

[78] J. Sverdrup-Thygeson, S. Moe, K. Y. Pettersen, and J. T. Gravdahl. Kinematic singularity avoidance for robot manipulators using set-based manipulability tasks. In *Proc. 1st IEEE Conference on Control Technology and Applications*, pages 142–149, Kohala Coast, Hawaii, Aug 27-30 2017.

[79] J. Sverdrup-Thygeson, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl. The Underwater Swimming Manipulator - A Bioinspired Solution for Subsea operations. *IEEE Journal of Oceanic Engineering*, 43(2):1–16, 2018.

[80] V. I. Utkin. *Sliding Modes in Control and Optimization*. Springer-Verlag, Berlin, Heidelberg, 1st ed. edition, 1992.

[81] Z. Wang. Adaptive smooth second-order sliding mode control method with application to missile guidance. *Transactions of the Institute of Measurement and Control*, 39(6):848–860, 2017.

[82] J. Wei, J. Yuan, and Z. Wang. Adaptive multivariable generalized super-twisting algorithm based robust coordinated control for a space robot subjected to coupled uncertainties. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(9):3244–3259, 2019.

[83] M. Wrzos-Kaminska, K. Y. Pettersen, and J. T. Gravdahl. Path following control for articulated intervention-AUVs using geometric control of reduced attitude. *IFAC PapersOnLine*, 52(16):192–197, Sep. 04-06 2019. In: Proc. 11th IFAC Symposium on Nonlinear Control Systemss, Vienna, Austria.

[84] J. Xu, M. Wang, and L. Qiao. Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles. *Ocean Engineering*, 105:54–63, 2015.

[85] L. Yang and J. Yang. Nonsingular fast terminal sliding-mode control for nonlinear dynamical systems. *International Journal of Robust and Nonlinear Control*, 21(16):1865–1879, 2011.

[86] S. Yu, X. Yu, B. Shirinzadeh, and Z. Man. Continuous finite-time control for robotic manipulators with terminal sliding mode. *Automatica*, 41(11):1957–1964, 2005.

[87] D. Zhu and B. Sun. The bio-inspired model based hybrid sliding-mode tracking control for unmanned underwater vehicles. *Engineering Applications of Artificial Intelligence*, 25(10):2260–2269, 2013.