Master's thesis

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

Erlend Andreas Basso

Dynamic Task Priority Control of Articulated Intervention AUVs

Using Control Lyapunov and Control Barrier Function based Quadratic Programs

Master's thesis in Cybernetics and Robotics
Supervisor: Professor Kristin Ytterstad Pettersen
July 2019

NTNU
Norwegian University of
Science and Technology

Erlend Andreas Basso

Dynamic Task Priority Control of Articulated Intervention AUVs

Using Control Lyapunov and Control Barrier Function based Quadratic Programs

Master's thesis in Cybernetics and Robotics
Supervisor: Professor Kristin Ytterstad Pettersen
July 2019

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

NTNU
Norwegian University of
Science and Technology

Abstract

The articulated intervention autonomous underwater vehicle (AIAUV) has emerged from the underwater snake robot (USR) by equipping it with longitudinal and tunnel thrusters. The AIAUV is an overactuated and highly redundant underwater floating base manipulator, where its entire articulated body serves as a floating manipulator arm, and its slender shape allows it to access narrow and confined spaces. Motivated by the redundant and overactuated AIAUV, this thesis is concerned with prioritized control of redundant robotic systems. Operational space control is investigated in the context of multiple-input and multiple-output (MIMO) feedback linearization, where conditions for input-output linearizability and full-state linearizability by an operational space control law are given. With the goal of including set-based tasks within a dynamic task priority framework, operational space control is merged with control barrier functions (CBFs). However, strict priority between tasks is lost in the process. As a consequence, operational space control is not pursued further and a novel task priority framework based on a hierarchy of control Lyapunov function (CLF) and control barrier function based quadratic programs (QPs) is developed. This framework ensures strict priority among different groups of control tasks such as safety related, operational and optimization tasks, which entails that tasks with a lower priority have no effect on the execution of all higher priority tasks. Moreover, a soft priority measure in the form of tunable 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 AIAUV control which solves the redundancy resolution, dynamic control and control allocation problems simultaneously.

Sammendrag

Artikulerte intervensjons-autonome undervannsfarkoster (AIAUV-er) har oppstått ved å utstyre undervanns slangeroboter (USR-er) med sideveis- og langsgående propeller. AIAUV-er er overaktuerte og svært redundante undervannsmanipulatorer, hvor hele dens artikulerte kropp fungerer som en flytende flerleddet manipulatorarm. Denne oppgaven er motivert av den redundante og overaktuerte AIAUV-en, og omhandler prioritert regulering av redundante robotsystemer. Sammenhengen mellom dynamisk nullromsbasert regulering og multivariabel (MIMO) linearisering ved tilbakekobling undersøkes, hvor kriteriene for inngang-utgang linearisering og full-tilstand linearisering er gitt. Dynamisk nullromsbasert regulering fusjoneres med kontrollbarrierefunksjoner (CBF-er) i et forsøk på å designe en dynamisk regulator som støtter mengdebaserte reguleringsopppgaver. Fusjonen resulterer i et optimaliseringsbasert rammeverk hvor streng prioritet mellom reguleringsoppgaver ikke lengre kan garanteres. Dynamisk nullromsbasert regulering droppes dermed og en reguleringsmetode basert på et hierarki av kontroll Lyapunovfunksjon- (CLF) og kontrollbarrierefunksjon-baserte kvadratiske programmer (QP-er) er utviklet. Dette optimaliseringsbaserte rammeverket sikrer streng prioritet mellom ulike grupper av reguleringsoppgaver som sikkerhetsrelaterte, operasjonelle og optimaliseringsoppgaver, noe som innebærer at oppgaver med lavere prioritet ikke har noen effekt på gjennomføringen av alle høyere prioriterte oppgaver. Videre kan et mykt prioritetsmål i form av innstillbare straffeparametere benyttes for å prioritere oppgaver på samme prioritetsnivå. I motsetning til kinematiske regulatorer er det foreslåtte rammeverket en helhetlig tilnærming til styring av AIAUV-er hvor redundansoppløsning, dynamisk regulering og kontrollallokering tas hånd om samtidig.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU). The work presented in this thesis has been carried out during the spring of 2019 under the supervision of Professor Kristin Y. Pettersen at the Department of Engineering Cybernetics, NTNU.

This thesis is the continuation of a specialization project conducted during the autumn of 2018. Since the specialization project is not published, important background theory and methods from the project report will be restated or paraphrased throughout this report to provide the best reading experience. A complete list of the material included from the specialization project is listed below.

- Parts of Section 1.1.
- Section 2.1.
- Section 2.2.
- Chapter 4.
- Section 5.1.
- Section 5.2.1 - Section 5.2.2.
- Parts of Section 5.3.1.2.

The MATLAB simulator used in this work was developed by PhD candidate Henrik Schmidt-Didlaukies at the Department of Marine Technology, NTNU. MATLAB functions for computation of a symbolic actuator configuration matrix as well as its partial and cross partial derivatives has been provided by MSc student Arnt-Erik Stene at the Department of Engineering Cybernetics, NTNU.

Unless otherwise stated, all figures and illustrations have been created by the author.

Acknowledgments

I would like to thank my supervisor, Professor Kristin Y. Pettersen for all of her advice and guidance. I would also like to thank the co-supervisor on my specialization project, Dr. Anna Kohl for all her help and our fruitful discussions.

Special thanks go to Assistant Professor Ole Kristian Fossum at UiT - The Arctic University of Norway for his inspiring mathematics lectures during my yearlong preparatory course for engineering studies in Tromsø. His mathematical rigor and focus on understanding rather than memorization has helped me immensely in my studies ever since.

Erlend Andreas Basso
Trondheim, July 2019

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This introductory chapter will present the motivation behind this thesis and a description of the problems to be solved. Relevant literature is also reviewed, before the contributions of the thesis are presented.

1.1 Motivation

As the number of subsea oil and gas installations continue to grow while ageing subsea infrastructure requires more preventive maintenance, the need for subsea inspection, maintenance and repair (IMR) solutions is increasing [1]. Historically, the remotely operated vehicle (ROV) has been the go-to solution for all subsea IMR operations. ROVs are operated by a human operator via a tethered telecommunications link from a submarine or surface ship, ROV operations are therefore both costly and time consuming. Increasing the autonomy of subsea IMR operations has the potential to significantly improve the safety and cost-effectiveness of operations [2]. While AUVs and smaller inspection class ROVs have gradually taken over subsea inspection operations, manipulation tasks still require the flexibility of a robotic arm.

The articulated intervention autonomous underwater vehicle (AIAUV) has emerged from the underwater snake robot (USR) by adding longitudinal and tunnel thrusters along the body. AIAUVs are a special class of underwater vehicle manipulator systems (UVMSs), where the system is both vehicle and manipulator at the same time [3]. The small size of the AIAUV and its articulated body enable it to better access narrow and confined spaces compared to a traditional UVMS with a large floating base. Because of its flexible shape, the AIAUV can assume an energy efficient torpedo shape similar to a conventional AUV for transportation or inspection tasks, while serving as a floating base manipulator for intervention tasks. The Eelume AIAUV [4] is depicted in Figure 1.1 and is designed to permanently live underwater by connecting to a docking station on the seabed.

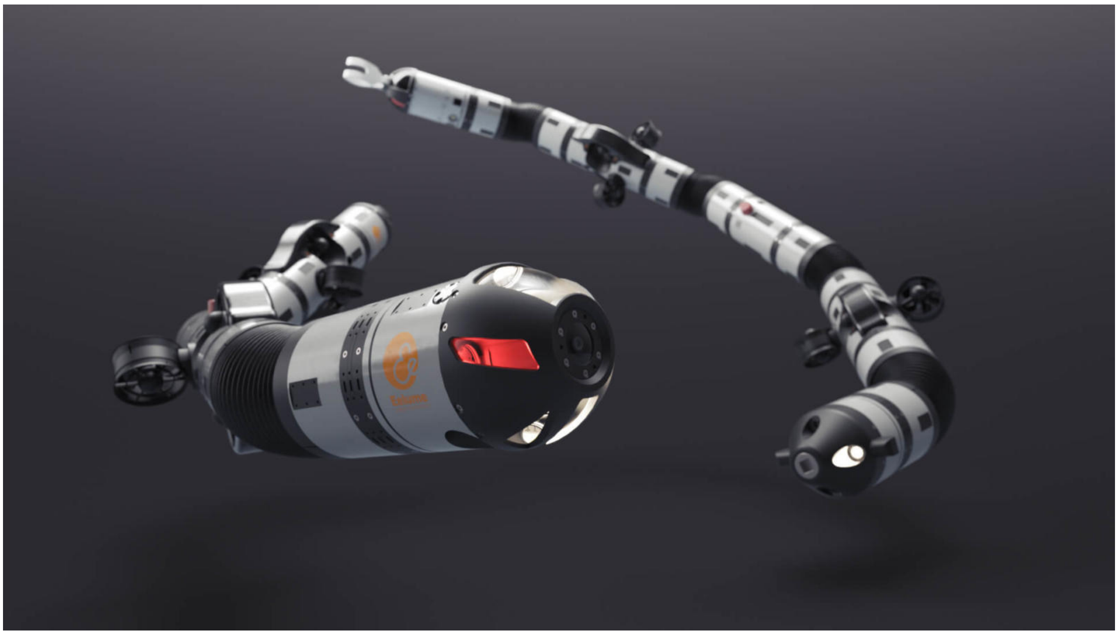


Figure 1.1: The Eelume AIAUV. Courtesy of Eelume [5].

AIAUVs are highly redundant and overactuated robotic systems, and both of these properties lead to optimization problems to be solved by the control system. Although these problems have traditionally been decoupled, this thesis develops a novel task priority control scheme that solves both problems in a unified manner.

1.2 Problem Description

AIAUVs are redundant with respect to typical equality tasks such as end-effector configuration control since they possess more degrees of freedom (DOFs) than those strictly required to execute the task. This presents the possibility of satisfying additional tasks simultaneously by utilizing the redundant DOFs of the system. To this end, task priority frameworks enable multiple tasks to be defined and prioritized with respect to their relative importance. Safety related tasks such as collision avoidance and joint limit avoidance are inherently described by inequalities that represent the sets in which they are satisfied. These types of tasks are referred to as set-based tasks, and a large research effort has been put into extending task priority frameworks to handle these tasks in recent years.

The overactuation property of AIAUVs entails that control allocation must be performed. Traditional task priority frameworks for redundancy resolution such as operational space control or kinematic control frameworks all decouple redundancy resolution and control allocation. As a result, the commanded forces and torques are computed with no regard to criteria such as actuator limits, actuator rate constraints and minimization of the actuator inputs. This is an important drawback for highly redundant robotic systems such as an AIAUV, where multiple commanded forces and torques may satisfy the control objectives. When the commanded forces and torques cannot be exactly allocated due to exceeding the actuator limits or rate constraints, tasks are no longer prioritized and the control performance degrades significantly, possibly destabilizing the system.

The main goal of the work presented in this thesis has been to develop a unified approach to redundancy resolution, dynamic control and control allocation of an AIAUV that supports set-based tasks. Thereby ensuring that the redundancy and overactuation properties can be fully exploited while strict priority among tasks can be guaranteed at all times.

1.3 Literature review

This section reviews literature relevant to this thesis in the context of the problem description in the preceding section.

1.3.1 Task priority frameworks

Redundant robotic systems are designed to accomplish multiple tasks simultaneously. Tasks are usually functions of the system configuration, and can be divided into equality- and set-based tasks. Equality-based tasks (or simply equality tasks) are control objectives that should be driven to a specific desired value, while set-based tasks should be kept within their valid set. Kinematic task priority control is a redundancy resolution method introduced in [6], developed in [7] and generalized to any number of priority levels in [8]. This control approach decouples the controller into a kinematic and dynamic controller, and has been successfully implemented on a number of robotic systems. This scheme was extended to support tasks described by sets or inequalities in [9], [10] and [11]. The aforementioned kinematic control approaches all resolve redundancy at the velocity level by generating velocity references for some dynamic controller to follow. An immediate drawback is that acceleration references cannot be included, resulting in worse tracking accuracy.

Operational space control [12] is a holistic approach that assigns joint torques directly by transforming the equations of motion from joint space into the operational space, which is also known as task space. Although it was mainly introduced for non-redundant systems, a dynamically consistent null space operator was defined in [12], that allowed two operational space tasks to be defined and controlled simultaneously. In [13], the scheme was extended to a task priority framework with an arbitrary number of tasks by generalizing the dynamically consistent null space operator from [12] to an arbitrary number of priority levels. These null space operators ensure that torques generated by lower priority tasks do not generate accelerations or forces affecting the task dynamics of higher priority tasks. The operational space framework was extended to include set-based tasks in [14], but this approach does not scale well for systems with a high number of DOFs.

Generalizing the operational space framework to set-based tasks was also attempted in [15] by utilizing the results in [9]. While this approach resulted in a set-based task priority operational space framework, the commanded force and torque vector exhibits discontinuous jumps whenever set-based tasks are activated or deactivated. Moreover, overshoots above or below the boundary of the valid set is observed whenever a set-based task is activated, which requires excessively large derivative gains in order to mitigate.

1.3.2 Control Lyapunov function based quadratic programs

Control Lyapunov functions (CLFs) extend Lyapunov theory to systems with inputs and have become an essential part of nonlinear control design after the groundbreaking work in [16–18]. The CLF concept was extended to rapidly exponentially stabilizing control Lyapunov functions (RES-CLFs) in [19], which achieve exponential convergence at a controllable rate. Through CLFs or RES-CLFs, the control designer is free to chose among an infinite number of controllers. An important example is the point-wise minimum norm controller [20, 21], which selects the control value of minimum norm from all control values that render the time derivative of the CLF negative definite. The point-wise minimum norm controller has a closed form solution since it is the solution to a quadratic program (QP) with only one inequality constraint. This QP can be augmented with control input saturation limits and other control input constraints, at the expense of a closed form solution [22]. For redundant robotic systems, two control tasks can be satisfied simultaneously by defining CLFs for each task and solving a quadratic optimization problem for the control input that ensures the negativeness of both CLF time derivatives while minimizing some quadratic objective function [23].

1.3.3 Control barrier functions

Barrier functions have been used extensively in constrained optimization [24, 25], and have motivated the concept of barrier certificates for safety-critical control. Barrier certificates were introduced as a tool for proving forward invariance of sets [26, 27]. These sets can encode safety related objectives, and proving the invariance of a safe set implies that the system will remain safe, as long as you start safe. Barrier certificates tend to infinity as the state tends to the boundary of the safe set, and in order to obtain safety guarantees beyond the boundary of the safe set, various Lyapunov-like approaches have been proposed such as in [28], where a positive definite barrier certificate is employed as a barrier Lyapunov function. However, the positive definiteness property is overly strong, since it enforces the invariance of every level set, and not just the set in question.

Barrier certificates were extended to systems with inputs by introducing the first notion of a control barrier function (CBF) in [29]. These control barrier functions were combined with control Lyapunov functions in [30], and further improved in [31] to establish conditions for so-called control Lyapunov-Barrier functions, which jointly guarantee safety and stability. These conditions were shown to be too restrictive and was hence relaxed in [32]. CBFs were generalized to exponential control barrier functions (ECBFs) in [33], which enforce forward invariance of higher order set-based tasks. The barrier function conditions were extended to the entire safe set in [34] by moving from reciprocal barrier functions to zeroing barrier functions[1], which enabled controller synthesis through optimization-based methods [35]. In particular, the CLF based QP [22, 23] could be augmented with CBFs to ensure stability and safety [32, 35].

1.3.4 Previous work on AIAUV control

The decoupled motion control framework depicted in Figure 1.2 was suggested and utilized for an AIAUV in [1]. Within this framework, the kinematic controller resolves redundancy by transforming multiple prioritized task space control objectives into system velocity references for the dynamic controller to track. The output of the dynamic controller is a vector of commanded forces and torques, which have to be transformed into thruster and joint torque control inputs by solving the control allocation problem.

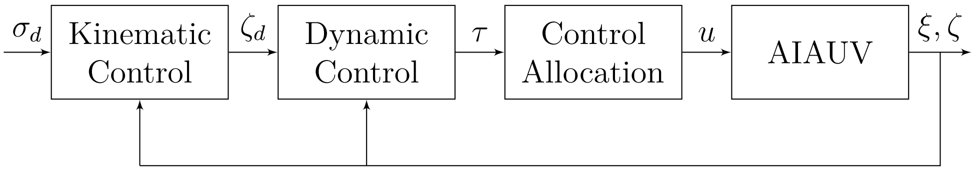


Figure 1.2: Overall control architecture for an AIAUV when employing a kinematic control scheme. The kinematic controller transforms a goal specified through an operational space task into desired system velocities that accomplish the goal.

This framework has been employed extensively within AIAUV research in recent years. Kinematic control schemes have been studied in [1, 15, 36–39], while sliding mode control has been investigated as a dynamic controller within this framework in [40]. Dynamic control of an AIAUV within this framework is also the topic of

[1] In this thesis, all barrier functions are zeroing barrier functions.

[41], which considers sliding mode control and adaptive inverse dynamics control.

The framework in Figure 1.2 decouples the redundancy resolution, dynamic control and control allocation problems into three separate parts. The main advantage is that each part can be designed under the assumption that the other parts are working perfectly, e.g. a kinematic controller can be designed under the assumption that the output velocity references are perfectly tracked. However, there are several inherent drawbacks to this scheme. Firstly, the mathematical stability proofs for velocity level kinematic control assumes that the velocity references are tracked perfectly by the dynamic controller, which entails that the kinematic controller has to be substantially slower than the dynamic controller for practical underwater applications. Secondly, task space acceleration references cannot be included when resolving redundancy at the velocity level in a kinematic controller, resulting in worse tracking performance. Finally, control allocation is performed after the reference velocities and commanded forces and torques have been computed. Hence, if exact control allocation is infeasible for the commanded forces and torques (due to e.g. violating control input limits and/or rate constraints), the control inputs are either saturated or found by minimizing the error between the commanded and allocated forces and torques in some sense. In other words, no attempt is made to utilize the inherent redundancy of the robotic system by optimizing the velocity references and/or commanded forces and torques with respect to physically realizable control inputs. As a result, priority among tasks is lost whenever exact control allocation is not possible, resulting in performance degradation of the control system, and instability in the worst case.

Alternatively, the kinematic and dynamic control blocks in the control architecture in Figure 1.2 can be merged into one by resorting to task priority operational space control [12, 13]. Within the operational space formalism, task space velocity feedback and task space acceleration references can be employed in the control laws, which improves tracking performance. In the context of AIAUV control, this approach has been the pursued in [15]. Despite improving tracking accuracy and unifying redundancy resolution and dynamic control, there are also several drawbacks to this approach for AIAUV control. Firstly, while velocity level kinematic control schemes have been successfully extended to include set-based tasks [9, 10], the same cannot be said for task priority operational space control. The method for set-based control within the operational space formalism in [15] results in discontinuous commanded forces and torques, chattering in the activation and deactivation of tasks and inevitable overshoots around the boundaries of the safe sets corresponding to set-based tasks. Secondly, control allocation is still fully decoupled from the control law, which means that the commanded forces and torques are computed with no regard to criteria such as actuator limits, actuator rate constraints and minimization of the actuator inputs. This is an important drawback for highly redundant robotic systems, where multiple commanded forces and torques may satisfy the control objectives, but where only some can be exactly allocated. Finally, since this approach relies on input-output linearization of the dynamics of every task, stabilizing terms are cancelled instead of exploited, and PD controllers are designed for the linearized dynamics of every task in order to achieve exponential convergence. The robustness of this input-output linearizing approach is questionable for practical applications, where modeling errors may lead to an unstable system.

1.4 Contributions

The main contributions of this thesis are:

- The connection between task priority operational space control as described in [13] and multiple-input and multiple-output (MIMO) feedback linearization as discussed in [42, 43]. This work includes formulating the task priority operational space control law described in [13] in matrix-vector form in a MIMO feedback linearization setting. Moreover, explicit conditions for input-output and full-state linearizability of a robotic system by a task priority operational space control law are given. (Section 3.1)

- A novel task priority framework in the form of a hierarchy of control Lyapunov function and control barrier function based quadratic programs. Strict priority levels are established by solving a quadratic program for every priority level, ensuring that tasks at lower priority levels have no effect on the execution of tasks at higher priority levels. This framework is especially useful for redundant robotic systems that are also overactuated, since the control allocation and redundancy resolution problems are solved simultaneously. (Section 3.2)

- The task priority framework is implemented and verified for an AIAUV application through MATLAB simulations. (Chapter 5)
- The main contribution on dynamic task priority control of redundant robotic systems using a hierarchy of CLF and CBF based QPs has been condensed into a conference paper draft, which is to be submitted to the 2020 American Control Conference. (Appendix B)

1.5 Outline

The rest of this thesis is organized as follows. Chapter 2 summarizes relevant background theory. In Chapter 3, the conditions for input-output and full-state linearizability of a robotic system under a task priority operational space control law are given. Chapter 3 also develops a novel task priority framework based on a hierarchy of control Lyapunov function and control barrier function based quadratic programs. In Chapter 4, kinematic and dynamic models of an AIAUV are presented for control and simulation purposes. Chapter 5 contains a case study of the proposed task priority framework for an AIAUV, while the conclusion and suggestions for future work can be found in Chapter 6. Finally, Appendix A contains various mathematical definitions and theorems employed in this thesis and Appendix B contains a draft for a conference paper based on the main result of this thesis.

Chapter 2

Background Theory

In this chapter, the necessary background material relevant to this thesis will be presented. A more thorough exposition on input-output linearization of MIMO systems can be found in [42, 43], while control barrier functions are discussed at length in [35].

2.1 Modeling of Robotic Systems

The system configuration of an n degree of freedom (DOF) robotic system can be expressed by the joint variables $q = [q_1, q_2, \dots, q_n]^{\mathrm{T}}$. The joint space dynamic equations of motion for a robotic manipulator are given by [44]

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau, \tag{2.1}$$

where $M(q)$ is the manipulator inertia matrix, $C(q, \dot{q})$ is the Coriolis and centrifugal matrix, $g(q)$ is the gravity torque vector and τ is the joint torques.

A task is defined as a generic m-dimensional control objective, specified as a function of the system configuration. The relationship between the joint space and task space variables are given by the direct kinematics equation [45]

$$\sigma(q) = f(q(t)), \tag{2.2}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a mapping from joint space to task space, which is nonlinear in general. By differentiating (2.2) with respect to time once and twice, the first- and second-order differential kinematics equation are obtained, viz.

$$\dot{\sigma} = J(q)\dot{q}, \tag{2.3}$$

$$\ddot{\sigma} = J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q}, \tag{2.4}$$

where $J(q(t)) = \frac{\partial f(q(t))}{\partial q} \in \mathbb{R}^{m \times n}$ is the configuration-dependent task Jacobian matrix, $\dot{q}(t) \in \mathbb{R}^n$ is the system velocity vector and $\ddot{q}(t) \in \mathbb{R}^n$ is the system acceleration vector. A robotic system is kinematically redundant when it has more DOFs than those strictly required to execute a given task [45], which requires the dimension of the system configuration q to be larger than the dimension of the task variable σ.

2.2 Operational Space Control

Operational space control [12] presents an alternative to controlling a robotic system in joint space by transforming the dynamic equations of motion into operational space. It is a holistic approach in the sense that forces and torques are computed directly from operational space variables, instead of relying on a kinematic controller to generate joint velocity or joint acceleration references for a dynamic controller to track in joint space.

To obtain the operational space dynamics, (2.1) is solved for $\ddot{q}$, which is inserted into (2.4)

$$\ddot{\sigma} = JM^{-1}\left(\tau - C\dot{q} - g\right) + \dot{J}\dot{q}. \tag{2.5}$$

By mapping the generalized torque into a generalized force through the relation

$$\tau = J^{\mathrm{T}}F, \tag{2.6}$$

defining the inertia matrix associated with the task variable σ as

$$\Lambda = \left(JM^{-1}J^{\mathrm{T}}\right)^{-1} \in \mathbb{R}^{m \times m}, \tag{2.7}$$

and pre-multiplying both sides of (2.5) by Λ, the operational space dynamics are obtained as

$$\Lambda\ddot{\sigma} + \Lambda\left(JM^{-1}C\dot{q} - \dot{J}\dot{q}\right) + \Lambda JM^{-1}g = f, \tag{2.8}$$

$$\Lambda\ddot{\sigma} + \mu + p = f. \tag{2.9}$$

If the system is redundant with respect to the task σ, i.e. $\dim(q) = n > m = \dim(\sigma)$, the generalized torque vector may be decomposed into a torque corresponding to the primary task and another torque acting in the null-space of the primary task as follows [12]

$$\tau = J^{\mathrm{T}}f + N\tau_0, \tag{2.10}$$

where τ_0 is an arbitrary torque and the null-space operator N is given by

$$N = I_n - J^{\mathrm{T}}\bar{J}^{\mathrm{T}}, \tag{2.11}$$

with

$$\bar{J} = M^{-1}J^{\mathrm{T}}\Lambda \tag{2.12}$$

$$= M^{-1}J^{\mathrm{T}}\left(JM^{-1}J^{\mathrm{T}}\right)^{-1} \in \mathbb{R}^{n\times m}. \tag{2.13}$$

The matrix $\bar{J}$ is known as the dynamically consistent pseudoinverse of J, which is a weighted pseudoinverse of J where the weight is the inverse of the inertia matrix [12].

2.2.1 Extension to k tasks

The null space operator in (2.11) is extended to an arbitrary number of priority levels in [13] as follows

$$N_1 = I, \tag{2.14a}$$

$$N_{k+1} = I - \sum_{i=1}^{k-1} N_i J_i^{\mathrm{T}}\left(J_i N_i^{\mathrm{T}} M^{-1} N_i J_i^{\mathrm{T}}\right)^{-1} J_i N_i^{\mathrm{T}} M^{-1}, \tag{2.14b}$$

where J_i is the Jacobian of task i. Moreover, by defining

$$\Lambda_i = \left(J_i N_i^{\mathrm{T}} M^{-1} N_i J_i^{\mathrm{T}}\right)^{-1} = \left(J_i M^{-1} N_i J_i^{\mathrm{T}}\right)^{-1}, \tag{2.15}$$

$$\bar{J}_i = M^{-1} J_i^{\mathrm{T}} \Lambda_i, \tag{2.16}$$

and employing certain properties of the null-space operator, (2.14) can be rewritten [15]

$$N_1 = I, \tag{2.17a}$$

$$N_{k+1} = N_k \left(I - J_k^{\mathrm{T}} \bar{J}_k^{\mathrm{T}} N_k\right). \tag{2.17b}$$

An operational space control torque for k tasks arranged in priority is then given by [13, 15]

$$\tau = J_1^{\mathrm{T}} f_1 + N_2 J_2^{\mathrm{T}} f_2 + \cdots + N_k J_k^{\mathrm{T}} f_k, \tag{2.18}$$

where

$$f_i = \Lambda_i a_i + \mu_i + p_i, \tag{2.19}$$

$$a_i = \ddot{\sigma}_{d,i} - K_{d,i}\dot{\tilde{\sigma}}_i - K_{p,i}\tilde{\sigma}_i - \dot{J}_i \dot{q} - J_i M^{-1} \sum_{j=1}^{i-1} N_j J_j^{\mathrm{T}} \Lambda_j a_j, \tag{2.20}$$

$$\mu_i = \Lambda_i \left(J_i M^{-1} C \dot{q} - \dot{J}_i \dot{q}\right) \tag{2.21}$$

$$p_i = \Lambda_i J_i M^{-1} g, \tag{2.22}$$

where $\tilde{\sigma} = \sigma - \sigma_d$ represents the task error and $K_{d,i}$ and $K_{p,i}$ are derivative and proportional gains, respectively.

2.3 Input-Output Linearization of MIMO Systems

Consider an input affine nonlinear control system of the form

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x) u_i \\ y_j &= h_j(x), \quad 1 \leq j \leq m \end{aligned} \tag{2.23}$$

where $x \in D \subset \mathbb{R}^n$ is the state vector, $f, g_i : D \to \mathbb{R}^n$ are smooth vector fields, and $h_i : D \to \mathbb{R}$ are smooth functions. Differentiating the ith output y_i with respect to time yields

$$\dot{y}_i = L_f h_i + \sum_{j=1}^{m} \left(L_{g_j} h_i \right) u_j. \tag{2.24}$$

Observe that if $L_{g_j} h = 0$ for all $j = 1, \ldots, m$, then the input does not appear in $\dot{y}_i$. Assume that y_i has to be differentiated with respect to time r_i times before at least one component of the control input vector u explicitly appears in a time derivative of y_i, then the r_ith derivative of y_i is given by

$$y_i^{(r_i)} = L_f^{r_i} h_i + \sum_{j=1}^{m} L_{g_j} \left(L_f^{r_i - 1} h_i \right) u_j. \tag{2.25}$$

The integer r_i is defined as the smallest integer such that

$$L_{g_j} L_f^k h_i(x) = 0, \quad 1 \leq j \leq m,\, k \leq r_i - 2 \tag{2.26}$$

$$L_{g_j} L_f^{r_i - 1} h_i(x) \neq 0, \quad \text{for at least one } 1 \leq j \leq m. \tag{2.27}$$

For single input single output (SISO) systems with $m = 1$, (2.26) and (2.27) is the definition of the relative degree of $y = h(x)$, with $h : \mathbb{R}^n \to \mathbb{R}$. The relative degree concept is extended to multiple-input and multiple-output (MIMO) systems as follows [42, 43]:

Definition 2.1 (Vector relative degree)**.** The system (2.23) has a vector relative degree $\{r_1, \ldots, r_m\}$ at a point x_0 if

(i)

$$L_{g_j} L_f^k h_i(x) = 0, \quad 0 \leq k \leq r_i - 2 \tag{2.28}$$

for all $1 \leq j \leq m$, for all $1 \leq i \leq m$, and for all x in a neighborhood of x_0.

(ii) The $m \times m$ matrix

$$A(x) = \begin{bmatrix} L_{g_1} L_f^{r_1 - 1} h_1(x) & L_{g_2} L_f^{r_1 - 1} h_1(x) & \dots & L_{g_m} L_f^{r_1 - 1} h_1(x) \\ L_{g_1} L_f^{r_2 - 1} h_2(x) & L_{g_2} L_f^{r_2 - 1} h_2(x) & \dots & L_{g_m} L_f^{r_2 - 1} h_2(x) \\ \vdots & & & \\ L_{g_1} L_f^{r_m - 1} h_m(x) & L_{g_2} L_f^{r_m - 1} h_m(x) & \dots & L_{g_m} L_f^{r_m - 1} h_m(x) \end{bmatrix} \tag{2.29}$$

is nonsingular at $x = x_0$.

Remark. For SISO systems, Definition 2.1 reduces to (2.26) and (2.27) since $m = 1$ implies that $A(x)$ is a scalar, which is invertible as long as it is non-zero. However, for MIMO systems, the characterization of r_i as the integer such that (2.26) and (2.27) holds is only implied by, but not equivalent to Definition 2.1. Importantly, MIMO systems also require that the matrix $A(x)$ is nonsingular, which implies that it is square and hence that the number of inputs must equal the number of outputs. However, as long as the number of outputs is less than or equal to the number of inputs, condition ii can be relaxed to requiring that $A(x)$ has linearly independent rows [42].

Remark. Several recent articles employ the notation $L_f h$ when $h : \mathbb{R}^n \to \mathbb{R}^m$ and $f : \mathbb{R}^n \to \mathbb{R}^n$, which is intended to represent the Lie derivative of h with respect to f [22, 32, 33, 46]. However, these papers interpret the Lie derivative of h with respect to f as $\frac{\partial h}{\partial x} f(x)$, which is only correct whenever $h : \mathbb{R}^n \to \mathbb{R}$, i.e. when h is a function. When h and f are both vector fields in $\mathbb{R}^n$, the Lie derivative of h with respect to f is known as the Lie bracket, which is given by [42, 43]

$$L_f h = [f, h] = \frac{\partial h}{\partial x} f(x) - \frac{\partial f}{\partial x} h(x). \tag{2.30}$$

Moreover, the notation $L_g h(x)$ is also used when $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$, which is supposed to be interpreted as $\frac{\partial h}{\partial x} g(x)$. However, the Lie derivative of a vector field is not defined with respect to a matrix [47].

2.3.1 Input-output dynamics

As long as r_i is well defined according to (2.26) and (2.27) for $i = 1, \ldots, m$, the input-output dynamics is given by

$$\begin{bmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \underbrace{\begin{bmatrix} L_f^{r_1} h_1 \\ L_f^{r_2} h_2 \\ \vdots \\ L_f^{r_m} h_m \end{bmatrix}}_{b(x)} + \underbrace{\begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1 & \ldots & L_{g_m} L_f^{r_1-1} h_1 \\ L_{g_1} L_f^{r_2-1} h_2 & \ldots & L_{g_m} L_f^{r_2-1} h_2 \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m & \ldots & L_{g_m} L_f^{r_m-1} h_m \end{bmatrix}}_{A(x)} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}. \tag{2.31}$$

Moreover, assuming that $r_1 = r_2 = \cdots = r_m = r$ yields

$$y^{(r)} = \begin{bmatrix} y_1^{(r)} \\ \vdots \\ y_m^{(r)} \end{bmatrix} = \underbrace{\begin{bmatrix} L_f^{r} h_1 \\ L_f^{r} h_2 \\ \vdots \\ L_f^{r} h_m \end{bmatrix}}_{b(x)} + \underbrace{\begin{bmatrix} L_{g_1} L_f^{r-1} h_1 & \ldots & L_{g_m} L_f^{r-1} h_1 \\ L_{g_1} L_f^{r-1} h_2 & \ldots & L_{g_m} L_f^{r-1} h_2 \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r-1} h_m & \ldots & L_{g_m} L_f^{r-1} h_m \end{bmatrix}}_{A(x)} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \tag{2.32}$$

and the system (2.23) can be decomposed into transverse dynamics states $\eta = \left[y^{\mathrm{T}}, \dot{y}^{\mathrm{T}}, \ldots, \left(y^{(r-1)}\right)^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{rm}$ and zero dynamics states $z \in Z \subset \mathbb{R}^{n-rm}$, viz.

$$\dot{\eta} = \bar{f}(\eta, z) + \bar{g}(\eta, z)u, \tag{2.33a}$$

$$\dot{z} = f_z(\eta, z), \tag{2.33b}$$

with $\bar{f}(\eta, z) = F\eta + Gb(x)$ and $\bar{g}(\eta, z) = GA(x)$ where

$$F = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ I \end{bmatrix}, \tag{2.34}$$

while 0 is the $m \times m$ matrix of zeros and I is the $m \times m$ identity matrix.

If the system (2.23) has a vector relative degree according to Definition 2.1, the state feedback control

$$u = A^{-1}(\mu - b), \tag{2.35}$$

will yield the linear and decoupled system

$$\dot{\eta} = F\eta + G\mu, \tag{2.36}$$

which can be exponentially stabilized by standard linear control techniques [48].

Remark. Recent articles such as [23, 32–35, 46] all characterize the scalar r in (2.32) as the relative degree of the vector-valued function $y \in \mathbb{R}^m$. However, the relative degree concept is only defined for SISO systems [42, 43], which means that the aforementioned articles should resort to the concept of vector relative degree for MIMO systems. Using the relative degree concept for MIMO systems is problematic, because it is not clear what is assumed when it is stated that some vector valued function $y \in \mathbb{R}^m$ has a relative degree of r. For instance, it could mean that at least one component of the control input u appears in at least one component of the rth derivative of the vector-valued function y, with no guarantees on the linear independence of the rows of the decoupling matrix $A(x)$. Alternatively, stating that $y \in \mathbb{R}^m$ has a relative degree of r could mean that the vector-valued function y has a vector relative degree of $\{r_1, r_2, \ldots, r_m\} = \{r, r, \ldots, r\}$, which implies that $A(x)$ is nonsingular.

2.4 Control Lyapunov Functions

The control Lyapunov function (CLF) extends Lyapunov theory to systems for which a feedback control has not already been selected [20]. For a nonlinear control system of the form

$$\dot{x} = f(x, u), \tag{2.37}$$

a CLF is a candidate Lyapunov function $V(x)$ for which there exists admissible control inputs u for every $x \neq 0$ such that $\nabla V(x) \cdot f(x, u) < 0$, which is to say that for any x, the time derivative of the candidate Lyapunov function can be made negative by appropriate selection of control input u.

Definition 2.2 (Control Lyapunov function)**.** A continuously differentiable and positive definite function $V : D \to \mathbb{R}$ is a control Lyapunov function (CLF) for the system (2.33) if there exists $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$ such that [20, 49]

$$\alpha_1(\|\eta\|) \leq V(\eta) \leq \alpha_2(\|\eta\|), \tag{2.38}$$

$$\inf_{u \in U} \Big[L_{\bar{f}} V_\epsilon(\eta, z) + L_{\bar{g}} V_\epsilon(\eta, z) u \Big] < -\alpha_3(\|\eta\|). \tag{2.39}$$

In order to explicitly control the rate of exponential convergence, a specific type of CLF is defined in [19] as follows:

Definition 2.3 (RES-CLF)**.** A continuously differentiable and positive definite function $V_\epsilon : D \to \mathbb{R}$ is said to be a rapidly exponentially stabilizing control Lyapunov function (RES-CLF) for the system (2.33) if there exists constants $c_1, c_2, c_3 > 0$ such that for all $0 < \epsilon < 1$ and for all states (η, z) it holds that

$$c_1 \|\eta\|^2 \leq V_\epsilon(\eta) \leq \frac{c_2}{\epsilon^2} \|\eta\|^2, \tag{2.40}$$

$$\inf_{u \in U} \Big[L_{\bar{f}} V_\epsilon(\eta, z) + L_{\bar{g}} V_\epsilon(\eta, z) u \Big] \leq -\frac{c_3}{\epsilon} V_\epsilon(\eta). \tag{2.41}$$

From Definition 2.3, the set of all stabilizing controllers for every $\eta \in \mathbb{R}^{rm}$ can be expressed as

$$K_\epsilon(\eta) := \Big\{ u \in U : L_{\bar{f}} V_\epsilon(\eta, z) + L_{\bar{g}} V_\epsilon(\eta, z) u \leq -\frac{c_3}{\epsilon} V_\epsilon(\eta) \Big\}. \tag{2.42}$$

Any Lipschitz continuous feedback control law $u \in K_\epsilon(\eta)$ satisfies

$$V_\epsilon(\eta) \leq e^{-\frac{c_3}{\epsilon}t} V_\epsilon(\eta(0)), \tag{2.43}$$

and

$$\|\eta(t)\| \leq \frac{1}{\epsilon}\sqrt{\frac{c_2}{c_1}} e^{-\frac{c_3}{2\epsilon}t} \|\eta(0)\|, \tag{2.44}$$

which shows that the rate of exponential convergence can be directly controlled with the constant ϵ through $\frac{c_3}{\epsilon}$ [19].

2.4.1 Constructing RES-CLFs

RES-CLFs can be constructed for the system (2.33) by solving the continuous time algebraic Riccati equation

$$F^{\mathrm{T}}P + PF - PGG^{\mathrm{T}}P + Q = 0, \tag{2.45}$$

for $P = P^{\mathrm{T}} > 0$, where Q is any positive definite matrix. In order to stabilize the transverse dynamics at a rate ϵ define

$$V_\epsilon(\eta) = \eta^{\mathrm{T}} \begin{bmatrix} \frac{1}{\epsilon}I & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} \frac{1}{\epsilon}I & 0 \\ 0 & I \end{bmatrix} \eta := \eta^{\mathrm{T}} P_\epsilon \eta, \tag{2.46}$$

where $P_\epsilon = M_\epsilon P M_\epsilon$. Inserting $P = M_\epsilon^{-1} P_\epsilon M_\epsilon^{-1}$ into (2.45) yields

$$\begin{aligned} &F^{\mathrm{T}} M_\epsilon^{-1} P_\epsilon M_\epsilon^{-1} + M_\epsilon^{-1} P_\epsilon M_\epsilon^{-1} F \\ &\quad - M_\epsilon^{-1} P_\epsilon M_\epsilon^{-1} G G^{\mathrm{T}} M_\epsilon^{-1} P_\epsilon M_\epsilon^{-1} + Q = 0, \end{aligned} \tag{2.47}$$

multiplying both sides of the equation by M_ϵ yields

$$M_\epsilon F^{\mathrm{T}} M_\epsilon^{-1} P_\epsilon + P_\epsilon M_\epsilon^{-1} F M_\epsilon - P_\epsilon M_\epsilon^{-1} G G^{\mathrm{T}} M_\epsilon P_\epsilon + M_\epsilon Q M_\epsilon = 0 \tag{2.48}$$

$$\epsilon F^{\mathrm{T}} P_\epsilon + \epsilon P_\epsilon F - P_\epsilon G G^{\mathrm{T}} P_\epsilon + M_\epsilon Q M_\epsilon = 0 \tag{2.49}$$

$$F^{\mathrm{T}} P_\epsilon + P_\epsilon F - \frac{1}{\epsilon} P_\epsilon G G^{\mathrm{T}} P_\epsilon + \frac{1}{\epsilon} M_\epsilon Q M_\epsilon = 0. \tag{2.50}$$

Now, let $V_\epsilon(\eta) = \eta^{\mathrm{T}} P_\epsilon \eta$ and differentiate with respect to time, viz.

$$\dot{V}_\epsilon(\eta, x) = (F\eta + Gb + GAu)^{\mathrm{T}} P_\epsilon \eta + \eta^{\mathrm{T}} P_\epsilon (F\eta + Gb + GAu) \tag{2.51}$$

$$= \underbrace{\eta^{\mathrm{T}} \left(F^{\mathrm{T}} P_\epsilon + P_\epsilon F\right) \eta + 2\eta^{\mathrm{T}} P_\epsilon Gb}_{L_{\bar{f}} V_\epsilon(\eta)} + \underbrace{2\eta^{\mathrm{T}} P_\epsilon GA}_{L_{\bar{g}} V_\epsilon(\eta)} u. \tag{2.52}$$

Inserting (2.50) yields

$$\dot{V}_\epsilon(\eta, x) = \eta^{\mathrm{T}} \left(\frac{1}{\epsilon} P_\epsilon G G^{\mathrm{T}} P_\epsilon - \frac{1}{\epsilon} M_\epsilon Q M_\epsilon\right) \eta + 2\eta^{\mathrm{T}} P_\epsilon Gb + 2\eta^{\mathrm{T}} P_\epsilon GAu. \tag{2.53}$$

Define $\gamma := \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} > 0$, where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are the minimum and maximum eigenvalues of the matrix, respectively. It follows that $\gamma P \leq Q$, which yields

$$\dot{V}_\epsilon(\eta, x) \leq \eta^{\mathrm{T}} \left(\frac{1}{\epsilon} P_\epsilon G G^{\mathrm{T}} P_\epsilon - \frac{\gamma}{\epsilon} P_\epsilon\right) \eta + 2\eta^{\mathrm{T}} P_\epsilon Gb + 2\eta^{\mathrm{T}} P_\epsilon GAu \tag{2.54}$$

$$= \eta^{\mathrm{T}} P_\epsilon G \left(\frac{1}{\epsilon} G^{\mathrm{T}} P_\epsilon \eta + 2b + 2Au\right) - \frac{\gamma}{\epsilon} V_\epsilon(\eta), \tag{2.55}$$

which implies that

$$\inf_{u \in \mathbb{R}^m} \left[L_{\bar{f}} V_\epsilon(\eta) + L_{\bar{g}} V_\epsilon(\eta) u\right] \leq -\frac{\gamma}{\epsilon} V_\epsilon(\eta), \tag{2.56}$$

as long as $A(x)$ has linearly independent rows. In this case, $V_\epsilon(\epsilon)$ is a RES-CLF for (2.33) with $c_1 = \lambda_{\min}(P_i)$, $c_2 = \lambda_{\max}(P_i)$ and $c_3 = \gamma$.

2.5 Control Barrier Functions

Control objectives described by inequalities or sets can be enforced by rendering the superlevel set

$$\mathcal{C} = \{x \in D \subset \mathbb{R}^n : h(x) \geq 0\}, \tag{2.57}$$

of some smooth function $h : D \to \mathbb{R}$ forward invariant. For instance, the inequality $\sigma(x) \geq \sigma_{\min}$ is satisfied by ensuring that $h(x) = \sigma(x) - \sigma_{\min} \geq 0$. These functions

h are known as control barrier functions if a control input u can be found rendering the set $\mathcal{C}$ forward invariant [34, 35]:

Definition 2.4. Let $\mathcal{C} \subset D \subset \mathbb{R}^n$ be the superlevel set of a continuously differentiable function $h : D \to \mathbb{R}$, then h is a control barrier function (CBF) for the system (2.23) if there exists an extended class K_∞ function α such that

$$\sup_{u \in U} \left[L_f h(x) + \frac{\partial h(x)}{\partial x} g(x) u \right] \geq -\alpha\left(h(x)\right), \tag{2.58}$$

for all $x \in \mathrm{Int}(\mathcal{C})$.

The existence of a CBF implies that the superlevel set of the function h is forward invariant [34], which means that if $x(t_0) = x_0 \in \mathcal{C}$, then $x = x(t) \in \mathcal{C}$ for all $t \geq t_0$. Equivalently, if $h(x_0) \geq 0$, then $h(x) \geq 0$ for all $t \geq t_0$.

2.5.1 Exponential Control Barrier Functions

Definition 2.4 assumes that $\frac{\partial h(x)}{\partial x} g(x) \neq 0$, which is to say that at least one component of the input vector u appears in the time derivative of the function h. However, safety related tasks for robotic systems are often only a function of the configuration variables, which means that they have to be differentiated twice for the input to appear. Introduced in [33] and refined in [35], exponential control barrier functions (ECBFs) generalize CBFs to functions $h(x)$ that satisfy

$$\frac{\partial L_f^k h(x)}{\partial x} g(x) = 0, \quad 0 \leq k \leq \rho - 2 \tag{2.59}$$

$$\frac{\partial L_f^{\rho-1} h(x)}{\partial x} g(x) \neq 0, \tag{2.60}$$

for an arbitrary integer $\rho \geq 1$.

Definition 2.5. Given a set $\mathcal{C} \subset D \subset \mathbb{R}^n$ defined as the superlevel set of an r-times continuously differentiable function $h : D \to \mathbb{R}$, then h is an exponential control barrier function (ECBF) for the control system (2.23) if there exists a row vector $K_\alpha \in \mathbb{R}^\rho$ such that

$$\sup_{u \in U} \left[L_f^\rho h(x) + \frac{\partial \left(L_f^{\rho-1} h(x) \right)}{\partial x} g(x) u \right] \geq -K_\alpha \xi(x), \tag{2.61}$$

where $\xi = \operatorname{col}\left(h(x), L_f h(x), L_f^2 h(x), \ldots, L_f^{\rho-1} h(x) \right)$, results in $h(x) \geq 0$ whenever $h(x_0) \geq 0$ for all $x \in \operatorname{Int}(\mathcal{C})$.

Remark. For $\rho = 1$, Definition 2.5 is a special case of Definition 2.4 where $\alpha(\cdot)$ is a linear function of $h(x)$, i.e. $\alpha(h(x)) = kh(x)$ with $k \in \mathbb{R}$.

Remark. The authors of [33, 35] define the value of ρ to be the relative degree of the ECBF h. However, the relative degree concept is only defined for SISO systems [42]. In the context of ECBFs for MIMO systems, ρ describes the number of times h must be differentiated with respect to time for *at least one* of the components of the control input to appear. In order to avoid any confusion, this thesis will refer to the scalar ρ as the order of the ECBF.

2.6 Optimization Based Control

The RES-CLF and ECBF conditions in (2.41) and (2.61) are both affine in the control input u. Consequently, these conditions can be employed as inequality constraints in a convex optimization problem for u. More specifically, by selecting a convex quadratic objective function, the convex optimization problem becomes a quadratic program (QP). The CLF-ECBF QP is given by [23], [35]:

$$\begin{aligned}
&\underset{u \in \mathbb{R}^m, \delta \in \mathbb{R}}{\text{minimize}} \quad \frac{1}{2} u^{\mathrm{T}} H(x) u + c^{\mathrm{T}}(x) u + w \delta^2 \\
&\text{subject to} \\
&L_{\bar{f}} V_\epsilon(\eta, z) + L_{\bar{g}} V_\epsilon(\eta, z) u \leq -\frac{\gamma}{\epsilon} V_\epsilon + \delta, \\
&L_f^\rho h(x) + \frac{\partial L_f^{\rho-1} h(x)}{\partial x} g(x) u \geq -K_\alpha \xi(x),
\end{aligned} \tag{CLF-ECBF QP}$$

where $H(x) \in \mathbb{R}^{m \times m}$ is any positive semi-definite matrix, $c(x) \in \mathbb{R}^m$, $\delta \in \mathbb{R}$ is a slack variable penalized by $w > 0$, ensuring feasibility of the quadratic program

(QP) in case of incompatible equality and set-based tasks, and the Lie derivatives $L_{\bar{f}}V_\epsilon(\eta, z)$ and $L_{\bar{g}}V_\epsilon(\eta, z)$ are given by (2.52). This controller ensures exponential stability of the equality task described by the RES-CLF V_ϵ, as long as it does not interfere with the satisfaction of set-based task encoded by the ρth order ECBF $h(x)$.

The CLF-ECBF QP can effortlessly be extended to incorporate control input saturation limits and rate constraints [22]:

$$\begin{aligned}
&\underset{u\in\mathbb{R}^m, \delta\in\mathbb{R}}{\text{minimize}} \quad \frac{1}{2}u^{\mathrm{T}}H(x)u + c^{\mathrm{T}}(x)u + w\delta^2 \\
&\text{subject to} \\
&L_{\bar{f}}V_\epsilon(\eta, z) + L_{\bar{g}}V_\epsilon(\eta, z)u \leq -\frac{\gamma}{\epsilon}V_\epsilon + \delta, \\
&L_f^\rho h(x) + \frac{\partial L_f^{\rho-1}h(x)}{\partial x}g(x)u \geq -K_\alpha \xi(x), \\
&u_{\min} \leq u \leq u_{\max}, \\
&\Delta u_{\min} \leq \Delta u \leq \Delta u_{\max},
\end{aligned} \tag{2.62}$$

where $u_{\min}$ and $u_{\max}$ are the lower and upper limits on the control input, Δu is the change in control input since the last time step, while $\Delta u_{\min}$ and $\Delta u_{\max}$ represents the maximum allowed change in control input from one time step to the next.

Chapter 3

Dynamic Task Priority Control of Redundant Robotic Systems

This chapter presents the main results of this thesis. In Section 3.1, the connection between operational space control of an arbitrary number of equality tasks and MIMO feedback linearization is investigated. Explicit conditions for the solvability of the input-output linearization and state space exact linearization problems are found. Section 3.1.3 continues the work in [23] on optimization based operational space controllers in an attempt to develop a dynamic controller that supports set-based tasks, which is achieved by employing CBFs. However, strict priority among tasks is lost when operational space control is combined with CLFs and ECBFs, and null space based control is therefore not pursued any further. Instead, the CLF-ECBF QP controller from [35] is extended to an arbitrary number of equality and set-based tasks in 3.2.1. Section 3.2.2 builds on this framework in order to achieve strict priority among different sets of tasks.

3.1 Operational Space Control and Feedback Linearization

The dynamic equations of motion for a robotic manipulator are given by [44]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau, \tag{3.1}$$

and a task priority operational space control law for k equality tasks is given by

$$\begin{aligned}
\tau &= J_1^{\mathrm{T}} v_1 + N_2 J_2^{\mathrm{T}} v_2 + \dots N_k J_k^{\mathrm{T}} v_k && (3.2)\\
&= \begin{bmatrix} J_1^{\mathrm{T}} & N_2 J_2^{\mathrm{T}} & \dots & N_k J_k^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix} && (3.3)\\
&= T(q)v, && (3.4)
\end{aligned}$$

where $v_i \in \mathbb{R}^{m_i}$, $v \in \mathbb{R}^m$ with $m = \sum_{i=1}^k m_i$ and where the null space operators N_i are given by (2.17). For $i = 1, \dots, k$, define the equality task variables as $\sigma_i(x) \in \mathbb{R}^{m_i}$ and $y_i(x) = \sigma_i(x) - \sigma_{i,d}(t)$ as the task error, where $\sigma_{i,d}(t)$ are the desired task values. By applying the pre-control law in (3.4), the dynamic equations of motion can be expressed in state space form as a nonlinear affine control system

$$\begin{aligned}
\dot{x} &= f(x) + g(x)v,\\
y &= h(x),
\end{aligned} \tag{3.5}$$

where $x = [x_1^{\mathrm{T}}, x_2^{\mathrm{T}}]^{\mathrm{T}} = [q^{\mathrm{T}}, \dot{q}^{\mathrm{T}}]^{\mathrm{T}} \in D \subset \mathbb{R}^{2n}$ is the state vector of joint angles and joint velocities and

$$f(x) = \begin{bmatrix} x_2 \\ -M(x_1)^{-1} \left(C(x_1, x_2) x_2 + g(x_1) \right) \end{bmatrix} \in \mathbb{R}^{2n}, \tag{3.6}$$

$$g(x) = \begin{bmatrix} 0_{n \times m} \\ M(x_1)^{-1} T(x_1) \end{bmatrix} \in \mathbb{R}^{2n \times m}, \tag{3.7}$$

$$h(x) = \begin{bmatrix} y_1(x) \\ \vdots \\ y_k(x) \end{bmatrix} \in \mathbb{R}^m. \tag{3.8}$$

3.1.1 Input-output dynamics

In the robotics literature, it is usually assumed that a task only depends on the configuration of the robotic system [45], viz.

$$\sigma(t) = f(q(t)). \tag{3.9}$$

In this case, the task error $y_i(x_1)$ has to be differentiated with respect to time twice for the input to appear. Taking the time derivative of $y_i(x_1)$ and resorting to the chain rule yields

$$\begin{aligned} \dot{y}_i(x) &= \frac{\partial \sigma_i(x_1)}{\partial x}\dot{x} - \dot{\sigma}_{i,d} && (3.10) \\ &= \frac{\partial \sigma_i(x_1)}{\partial x}\left(f(x) + g(x)v\right) - \dot{\sigma}_{i,d}, && (3.11) \end{aligned}$$

where

$$\begin{aligned} \frac{\partial \sigma_i(x_1)}{\partial x}f(x) &= \begin{bmatrix} \frac{\partial \sigma_i(x_1)}{\partial x_1} & \frac{\partial \sigma_i(x_1)}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_2 \\ -M(x_1)^{-1}\left(C(x_1,x_2)x_2 + g(x_1)\right) \end{bmatrix} && (3.12) \\ &= \frac{\partial \sigma_i(x_1)}{\partial x_1}x_2 && (3.13) \\ &= J_i(x_1)x_2, && (3.14) \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \sigma_i(x_1)}{\partial x}g(x) &= \begin{bmatrix} \frac{\partial \sigma_i(x_1)}{\partial x_1} & \frac{\partial \sigma_i(x_1)}{\partial x_2} \end{bmatrix} \begin{bmatrix} 0 \\ M(x_1)^{-1}T(x_1) \end{bmatrix} && (3.15) \\ &= 0. && (3.16) \end{aligned}$$

As expected, the control input does not appear in $\dot{y}_i$. Differentiating the task error with respect to time once more yields

$$\ddot{y}_i(x) = \frac{\partial}{\partial x}\left(\frac{\partial \sigma_i(x_1)}{\partial x}f(x)\right)f(x) + \frac{\partial}{\partial x}\left(\frac{\partial \sigma_i(x_1)}{\partial x}f(x)\right)g(x)v - \ddot{\sigma}_{i,d} \tag{3.17}$$

$$= \underbrace{\begin{bmatrix} L_f^2\sigma_{i_1}(x) - \ddot{\sigma}_{i_1,d} \\ L_f^2\sigma_{i_2}(x) - \ddot{\sigma}_{i_2,d} \\ \vdots \\ L_f^2\sigma_{i_{m_i}}(x) - \ddot{\sigma}_{i_{m_i},d} \end{bmatrix}}_{b_i(x)} + \underbrace{\begin{bmatrix} L_{g_1}L_f\sigma_{i_1}(x) & \dots & L_{g_m}L_f\sigma_{i_1}(x) \\ L_{g_1}L_f\sigma_{i_2}(x) & \dots & L_{g_m}L_f\sigma_{i_2}(x) \\ \vdots & & \vdots \\ L_{g_1}L_f\sigma_{i_{m_i}}(x) & \dots & L_{g_m}L_f\sigma_{i_{m_i}}(x) \end{bmatrix}}_{A_i(x)} v, \tag{3.18}$$

where

$$A_i(x) = \frac{\partial}{\partial x}\left(\frac{\partial \sigma_i(x_1)}{\partial x} f(x)\right) g(x) \tag{3.19}$$

$$= \frac{\partial}{\partial x}\left(J_i(x_1)x_2\right) g(x) \tag{3.20}$$

$$= \begin{bmatrix} \frac{\partial J_i(x_1)x_2}{\partial x_1} & \frac{\partial J_i(x_1)x_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} 0 \\ M(x_1)^{-1}T(x_1) \end{bmatrix} \tag{3.21}$$

$$= J_i(x_1)M(x_1)^{-1}T(x_1), \tag{3.22}$$

and

$$\begin{aligned} b_i(x) &= \frac{\partial}{\partial x}\left(\frac{\partial \sigma_i(x_1)}{\partial x} f(x)\right) f(x) - \ddot{\sigma}_{i,d}, \\ &= \frac{\partial}{\partial x}\left(J_i(x_1)x_2\right) f(x) - \ddot{\sigma}_{i,d}, \\ &= \begin{bmatrix} \frac{\partial J_i(x_1)x_2}{\partial x_1} & \frac{\partial J_i(x_1)x_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_2 \\ M(x_1)^{-1}\left(C(x_1,x_2)x_2 + g(x_1)\right) \end{bmatrix} - \ddot{\sigma}_{i,d}, \\ &= \frac{\partial J_i(x_1)x_2}{\partial x_1} x_2 - \frac{\partial J_i(x_1)x_2}{\partial x_2} M(x_1)^{-1}\left(C(x_1,x_2)x_2 + g(x_1)\right) - \ddot{\sigma}_{i,d}, \\ &= \frac{\partial J_i(x_1)}{\partial x_1}\left(x_2 \otimes I_n\right) x_2 + J_i(x_1)\frac{\partial x_2}{\partial x_1} x_2 \\ &\quad - \left(\frac{\partial J_i(x_1)}{\partial x_2}\left(x_2 \otimes I_n\right) + J_i(x_1)\frac{\partial x_2}{\partial x_2}\right) M(x_1)^{-1}\left(C(x_1,x_2)x_2 + g(x_1)\right) - \ddot{\sigma}_{i,d}, \\ &= \frac{\partial J_i(x_1)}{\partial x_1}\left(x_2 \otimes I_n\right) x_2 - J_i(x_1)M(x_1)^{-1}\left(C(x_1,x_2)x_2 + g(x_1)\right) - \ddot{\sigma}_{i,d}, \end{aligned} \tag{3.23}$$

where the product rule for the partial derivative of the product of a matrix and a vector with respect to a vector has been utilized [50], and $\otimes$ denotes the Kronecker product. Since $\dot{J}_i(x_1) = \frac{\partial J_i(x_1)}{\partial x_1}\left(\dot{x}_1 \otimes I_n\right)$ and $\dot{x}_1 = x_2$ it holds that

$$b_i(x) = \dot{J}_i(x_1)x_2 - J_i(x_1)M(x_1)^{-1}\left(C(x_1,x_2)x_2 + g(x_1)\right) - \ddot{\sigma}_{i,d}. \tag{3.24}$$

The input-output dynamics of all tasks can therefore be expressed as

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \vdots \\ \ddot{y}_k \end{bmatrix} = \underbrace{\begin{bmatrix} b_1(x) \\ b_2(x) \\ \vdots \\ b_k(x) \end{bmatrix}}_{b(x)} + \underbrace{\begin{bmatrix} A_1(x) \\ A_2(x) \\ \vdots \\ A_k(x) \end{bmatrix}}_{A(x)} v, \tag{3.25}$$

where

$$b(x) = \begin{bmatrix} \dot{J}_1 x_2 - J_1 M^{-1}\left(Cx_2 + g\right) - \ddot{\sigma}_{1,d} \\ \dot{J}_2 x_2 - J_2 M^{-1}\left(Cx_2 + g\right) - \ddot{\sigma}_{2,d} \\ \vdots \\ \dot{J}_k x_2 - J_k M^{-1}\left(Cx_2 + g\right) - \ddot{\sigma}_{k,d} \end{bmatrix}, \tag{3.26}$$

and

$$A(x) = \begin{bmatrix} J_1M^{-1}J_1^{\mathrm{T}} & 0_{m_1\times m_2} & 0_{m_1\times m_3} & \cdots & 0_{m_1\times m_k} \\ J_2M^{-1}J_1^{\mathrm{T}} & J_2M^{-1}N_2J_2^{\mathrm{T}} & 0_{m_2\times m_3} & \cdots & 0_{m_2\times m_k} \\ J_3M^{-1}J_1^{\mathrm{T}} & J_3M^{-1}N_2J_2^{\mathrm{T}} & J_3M^{-1}N_3J_3^{\mathrm{T}} & \ddots & 0_{m_3\times m_k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ J_kM^{-1}J_1^{\mathrm{T}} & J_kM^{-1}N_2J_2^{\mathrm{T}} & J_kM^{-1}N_3J_3^{\mathrm{T}} & \cdots & J_kM^{-1}N_kJ_k^{\mathrm{T}} \end{bmatrix}, \tag{3.27}$$

since $A_i = J_iM^{-1}T$ and $J_iM^{-1}N_k = 0$ for $i < k$.

3.1.2 MIMO feedback linearization

A standard assumption in kinematic or operational space control is that all tasks are compatible. For operational space control, assuming that all tasks are compatible around a point x_0 is equivalent to assuming

$$\mathrm{Null}\left(N_i J_i^{\mathrm{T}}\right) = \varnothing, \quad i = 1, \ldots, k, \tag{3.28}$$

at the point x_0. Equation (3.28) implies that the matrix $N_i J_i^{\mathrm{T}}$ has full rank at x_0 and thus that (2.15) is well-defined without resorting to a pseudoinverse for all $i = 1, \ldots, k$. Hence, every submatrix in (3.27) has full rank, which implies that $A(x)$ has full rank and the system in (3.5) has a vector relative degree

$$\{r_1, r_1, \ldots, r_m\} = \{2, 2, \ldots, 2\}, \tag{3.29}$$

at x_0.

When the system (3.5) has a well-defined vector relative degree, the system is input-output linearizable and the control input

$$v = A^{-1}(x)\left(\mu - b(x)\right), \tag{3.30}$$

will fully linearize the input-output dynamics

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \vdots \\ \ddot{y}_k \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_k \end{bmatrix}, \tag{3.31}$$

where the inverse of $A(x)$ is given by

$$A^{-1}(x) = \begin{bmatrix} M_1 & 0_{m_1 \times m_2} & 0_{m_1 \times m_3} & \dots & 0_{m_1 \times m_k} \\ -M_2 J_2 M^{-1} \Gamma_{21} & M_2 & 0_{m_2 \times m_3} & \dots & 0_{m_2 \times m_k} \\ -M_3 J_3 M^{-1} \Gamma_{31} & -M_3 J_3 M^{-1} \Gamma_{32} & M_3 & \ddots & 0_{m_3 \times m_k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -M_k J_k M^{-1} \Gamma_{k1} & -M_k J_k M^{-1} \Gamma_{k2} & -M_k J_k M^{-1} \Gamma_{k3} & \dots & M_k \end{bmatrix}, \tag{3.32}$$

with

$$\Gamma_{ij} = N_j J_j^{\mathrm{T}} \Lambda_j - \sum_{k=j+1}^{i-1} N_k J_k^{\mathrm{T}} \Lambda_k J_k M^{-1} N_j J_j^{\mathrm{T}} \Lambda_j, \tag{3.33}$$

where Λ_j is given by (2.15). Since (3.31) is a linear system, μ can easily be designed such that the input-output dynamics of every task is stable. However, the zero dynamics must be asymptotically stable in order to guarantee that the MIMO system is minimum phase, and hence stable [42].

In order to avoid analyzing complicated zero dynamics, the tasks can be designed such that

$$r_1 + r_2 + \dots + r_m = 2n, \tag{3.34}$$

and by assuming that $g(x_0)$ has rank m in a neighborhood around x_0, the state space exact linearization problem is solvable by Lemma 5.2.1 in [42]. In other words, if the tasks are defined in such a way that the system has a vector relative degree in a neighborhood around x_0, the matrix $g(x_0)$ has rank m around x_0 and $r_1 + r_2 + \dots + r_k = 2n$, then the system has no zero dynamics and the system is full-state feedback linearizable. Consequently, stability of the entire system can be guaranteed by designing μ such that the linear system described by (3.31) is stable.

The task priority operational space control law in [13] can be formulated as input-output linearization by employing PD control laws μ_i for every task. By defining the vectors $\eta_i := [y_i^{\mathrm{T}}, \dot{y}_i^{\mathrm{T}}]^{\mathrm{T}}$, eq. (3.31) yields the following differential equations

$$\dot{\eta}_i = \tilde{f}_i + \tilde{g}_i \mu_i, \tag{3.35}$$

$$= F_i \eta_i + G_i \mu_i, \tag{3.36}$$

where

$$F_i = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad G_i = \begin{bmatrix} 0 \\ I \end{bmatrix}, \tag{3.37}$$

for $i = 1, \ldots, k$. By employing the PD control law

$$\mu_i = \begin{bmatrix} -K_{p,i} & -K_{d,i} \end{bmatrix} \eta_i, \tag{3.38}$$

the closed loop system for each task becomes

$$\ddot{y}_i + K_{d,i} \dot{y}_i + K_p y_i = 0, \tag{3.39}$$

under the assumption that all tasks are compatible, i.e. that the system exhibits a vector relative degree.

Remark. The control law given by (3.4), (3.30) and (3.38) is exactly equal to (2.18), which was introduced in [13].

3.1.2.1 Incompatible tasks

If the ith task becomes incompatible with one or more higher priority tasks at the point x_0, then $A(x_0)$ is singular and the system does not have a vector relative degree. However, as long as all the higher priority tasks are still compatible with each other, the submatrix of $A(x)$ containing the first $\sum_{k=1}^{i-1} m_k$ rows and columns will still be nonsingular. Consequently, the first $i - 1$ tasks are still input-output linearizable. As for the ith task, the singular value decomposition of Λ_i^{-1} can be used to characterize the controllable directions and compute a pseudoinverse of Λ_i, which can be employed in (3.32). It can be shown that the error coordinates representing the controllable directions are still GAS under the control law given by (3.4), (3.30) and (3.38) [51].

3.1.3 Implementation through quadratic programs

Instead of choosing a predetermined control law for each task, rapidly exponentially stabilizing control Lyapunov functions can be utilized in order to guarantee rapid exponential stability of the task variables η_i [52]. By constructing a RES-CLF, $V_{\epsilon,i} = \eta_i^{\mathrm{T}} P_{\epsilon,i} \eta_i$ for (3.35), a control law $\mu_i \in K_{\epsilon,i}(x)$ can be found by solving the following quadratic program (QP)

$$\underset{\mu_i \in \mathbb{R}^{m_i}}{\text{minimize}} \quad \mu_i^{\mathrm{T}} \mu_i \tag{3.40a}$$

$$\text{subject to}$$

$$L_{\tilde{f}_i} V_{\epsilon_i} + L_{\tilde{g}_i} V_{\epsilon_i} \mu_i \leq -\frac{\gamma_i}{\epsilon_i} V_{\epsilon_i}. \tag{3.40b}$$

This QP has a single affine inequality constraint, and it therefore has an explicit solution given by [53]

$$\mu_i(\eta_i) = \begin{cases} -\frac{\psi_{0,i}(\eta_i)\psi_{1,i}(\eta_i)}{\psi_{1,i}^{\mathrm{T}}(\eta_i)\psi_{1,i}(\eta_i)}, & \text{if } \psi_{0,i}(\eta_i) > 0, \\ 0, & \text{if } \psi_{0,i}(\eta_i) \leq 0, \end{cases} , \tag{3.41}$$

where

$$\psi_{0,i} = L_{\tilde{f}_i} V_{\epsilon_i} + \frac{\gamma_i}{\epsilon_i} V_{\epsilon,i}, \tag{3.42}$$

$$\psi_{1,i} = \left(L_{\tilde{g}_i} V_{\epsilon,i}\right)^{\mathrm{T}}. \tag{3.43}$$

In order to obtain control inputs μ_i for every task according to (3.30), one possibility is to employ (3.41) for $i = 1, \dots, k$. When $A(x)$ has linearly independent rows, this approach yields controllers that achieve exponential stability of every task at a predetermined rate, while minimizing the control effort μ. However, set-based tasks described by ECBFs cannot be accounted for without augmenting every QP in (3.40) with additional inequality constraints, which entails that the QPs no longer have closed form solutions.

Alternatively, the QP in (3.40) can be extended from one task to k tasks by solving a single QP with k affine inequality constraints as shown for the two task case in [23], viz.

$$\begin{aligned}
&\underset{\mu\in\mathbb{R}^m,\delta\in\mathbb{R}^k}{\text{minimize}} \quad \mu^{\mathrm{T}}\mu+\delta^{\mathrm{T}}W\delta && (3.44\text{a})\\
&\text{subject to} &&\\
&L_{\tilde{f}_i}V_{\epsilon_i}+L_{\tilde{g}_i}V_{\epsilon_i}\mu_i \leq -\frac{\gamma_i}{\epsilon_i}V_{\epsilon_i}+\delta_i, \quad i=1,\dots,k, && (3.44\text{b})
\end{aligned}$$

where slack variables δ, penalized by $W\in\mathbb{R}^{k\times k}$ are added to each CLF constraint in order to ensure feasibility of the QP, in case of conflicting tasks. Since this formulation solves a single QP with one inequality constraint for each task, it is a simple matter of extending (3.44) to account for l set-based tasks described by ECBFs $h_j\in\mathbb{R}$ as follows [32]

$$\begin{aligned}
&\underset{\mu\in\mathbb{R}^m,\delta\in\mathbb{R}^k}{\text{minimize}} \quad \mu^{\mathrm{T}}\mu+\delta^{\mathrm{T}}W\delta && (3.45\text{a})\\
&\text{subject to} &&\\
&L_{\tilde{f}_i}V_{\epsilon_i}+L_{\tilde{g}_i}V_{\epsilon_i}\mu_i \leq -\frac{\gamma_i}{\epsilon_i}V_{\epsilon_i}+\delta_i, \qquad i=1,\dots,k, && (3.45\text{b})\\
&L_f^{\rho_j}h_j+\frac{\partial L_f^{\rho_j-1}h_j}{\partial x}gA^{-1}(\mu-b)\geq -K_{\alpha,j}\xi_j, \quad j=1,\dots,l, && (3.45\text{c})
\end{aligned}$$

where $\xi_j=\mathrm{col}\left(h(x),L_fh(x),\dots,L_f^{\rho_j-1}h(x)\right)$ and $K_{\alpha,j}=\mathrm{row}\left(\alpha_{1,j},\alpha_{2,j},\dots,\alpha_{\rho_j,j}\right)$.

A well known drawback associated with feedback linearizing control is the reliance on exact cancellation of all nonlinearities. Nonlinearities are not inherently bad, and stabilizing nonlinearities should be exploited instead of canceled by feedback. In an attempt to mitigate this, the QP in (3.45) can also be formulated in terms of v by solving (3.30) for μ [23]

$$\mu=Av+b. \tag{3.46}$$

By rewriting the objective function term $\mu^{\mathrm{T}}\mu$ as

$$\mu^{\mathrm{T}}\mu=v^{\mathrm{T}}A^{\mathrm{T}}Av+2b^{\mathrm{T}}Av+b^{\mathrm{T}}b, \tag{3.47}$$

the QP in (3.45) be expressed as

$$\underset{v\in\mathbb{R}^m,\delta\in\mathbb{R}^k}{\text{minimize}} \quad v^{\mathrm{T}}A^{\mathrm{T}}Av + 2b^{\mathrm{T}}Av + \delta^{\mathrm{T}}W\delta \tag{3.48a}$$

subject to

$$L_{\tilde{f}_i}V_{\epsilon_i} + L_{\tilde{g}_i}V_{\epsilon_i}\left(A_iv + b_i\right) \leq -\frac{\gamma_i}{\epsilon_i}V_{\epsilon_i} + \delta_i, \quad i = 1, \ldots, k, \tag{3.48b}$$

$$L_f^{\rho_j}h_j + \frac{\partial L_f^{\rho_j-1}h_j}{\partial x}gv \geq -K_{\alpha,j}\xi_j, \qquad j = 1, \ldots, l. \tag{3.48c}$$

In addition to including set-based tasks, the schemes in (3.45) and (3.48) can be extended to incorporate control input saturation limits and rate constraints as shown in Section 2.6.

3.1.3.1 Discussion

Both of the QPs in (3.45) and (3.48) extend operational space control to set-based tasks, where the former includes the nonlinear task dynamics within the CLF constraints. As a result, ‘good’ nonlinearities can be exploited instead of canceled, possibly reducing the overall control effort and robustifying the system with respect to modeling errors. However, the price to be paid is that whenever two or more tasks are incompatible, both QPs yield control inputs μ and v that do not respect the strict priority between tasks. This is due to the fact that a single objective function encodes the satisfaction/stability of all tasks through the value of the slack variables δ and penalty parameters in W. Therefore, when two or more tasks cannot be achieved simultaneously, the solution μ or v will invariably lead to trade-off configurations that do not satisfy any of the tasks perfectly. The problem can be somewhat mitigated by choosing excessively large penalty values for higher priority tasks.

There is another issue in (3.48) that leads to coupled tasks. Consider the following term in the second CLF constraint

$$A_2v = J_2M^{-1}J_1^{\mathrm{T}}v_1 + J_2M^{-1}N_2J_2^{\mathrm{T}}v_2, \tag{3.49}$$

which entails that the control inputs for higher priority tasks (such as v_1) are optimized not only with respect to their own task dynamics, but also with respect to how they influence the execution of lower priority tasks.

The aforementioned problems do not occur when applying the min-norm controller from (3.40) for every task, since these QPs are fully independent of each other. However, the closed form solutions come at the cost of a single inequality constraint, which means that set-based tasks are not supported. These issues were not addressed in [23], wherein the authors claimed that the QP in (3.48) with the control law (3.4) does not allow for dynamic weighting of the tasks, because lower priority tasks only act in the null space of higher priority tasks. While it is true that the control input v_k of some lower priority task does not directly influence the task dynamics of some task $p < k$, the control inputs v_k and v_p are found through (3.48), where the effect of v_k on the pth task is explicitly taken into account, as demonstrated for $p = 1$ and $k = 2$ in (3.49). Because slack variables are added to each CLF constraint, the tasks can in fact be dynamically weighted by tuning the penalty parameters.

The main idea behind null space based control such as operational space control is to ensure strict priority between tasks by utilizing null space operators. As a result, trade-off configurations do not occur when tasks are incompatible. In this way, lower priority optimization tasks can safely be included without affecting the mission related control tasks. By losing this defining feature, the author argues that null space based control is not suitable in combination with the optimization-based controllers from (3.45) and (3.48).

3.2 Prioritizing Tasks through Quadratic Programs

The conclusion of the previous section was that tasks at different priority levels become coupled when combining control Lyapunov function and control barrier function based optimization problems with an operational space pre-control law. Importantly, lower priority tasks affect the execution of higher priority tasks. In order to achieve strict priority between tasks, this section abandons the operational space pre-control law in (3.4), and extends the CLF-ECBF QP controller in [35] to N equality- and M set-based tasks distributed to an arbitrary number of priority levels. Strict priority between tasks at different priority levels is achieved, while a soft priority measure can be employed to prioritize tasks at the same priority level.

3.2.1 CLF penalty parameters as a priority measure

Consider the nonlinear affine control system

$$\dot{x} = f(x) + g(x)u, \tag{3.50}$$

$$y = h(x), \tag{3.51}$$

where f and g are locally Lipschitz, $x \in D \subset \mathbb{R}^n$ and $u \in U \subset \mathbb{R}^p$ is the set admissible control inputs. The N equality tasks are described by locally Lipschitz vector-valued error functions $y_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$, which are stacked in the output vector

$$h(x) = \begin{bmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_N(x) \end{bmatrix}. \tag{3.52}$$

The following is assumed on the scalar output component y_{i_j} of every task:

Assumption 3.1. For every equality task $y_i \in \mathbb{R}^{m_i}$, there exists an integer r_i such that

$$L_{g_j} L_f^k y_{i_l} = 0, \quad 1 \leq j \leq m,\ 0 \leq k \leq r_i - 2 \tag{3.53}$$

$$L_{g_j} L_f^{r_i - 1} y_{i_l} \neq 0, \quad \text{for at least one } 1 \leq j \leq m, \tag{3.54}$$

for every $l = 1, \ldots, m_i$ and $i = 1, \ldots, N$.

Assumption 3.1 states that the control input appears in each component of $y_i^{(r_i)}$, and in no components of $y_i^{(k)}$ for $k < r_i$. Under this assumption, the input-output dynamics of each task $y_i^{(r_i)}$ is given by

$$\begin{bmatrix} y_{i_1}^{(r_i)}(x) \\ \vdots \\ y_{i_{m_i}}^{(r_i)}(x) \end{bmatrix} = \underbrace{\begin{bmatrix} L_f^{r_i} y_{i_1}(x) \\ \vdots \\ L_f^{r_i} y_{i_{m_i}}(x) \end{bmatrix}}_{b_i(x)} + \underbrace{\begin{bmatrix} L_{g_1} L_f^{r_i-1} y_{i_1}(x) & \vdots & L_{g_p} L_f^{r_i-1} y_{i_1}(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_i-1} y_{i_{m_i}}(x) & \vdots & L_{g_m} L_f^{r_i-1} y_{i_{m_i}}(x) \end{bmatrix}}_{A_i(x)} u. \tag{3.55}$$

Transverse dynamics states $\eta_i = \left[y_i^{\mathrm{T}}, \dot{y}_i^{\mathrm{T}}, \ldots, \left(\dot{y}_i^{(r_i-1)} \right)^{\mathrm{T}} \right]^{\mathrm{T}} \in \mathbb{R}^{r_i m_i}$ and RES-CLFs $V_{\epsilon,i}$ can then be defined analogously to (2.33a), (2.34) and (2.46) with the goal of driving $\eta_i \to 0$ for all $i = 1, \ldots, N$.

Set-based tasks can be included if there exists control barrier functions or exponential control barrier functions $h(x) : \mathbb{R}^n \to \mathbb{R}$ encoding the set-based tasks. A straightforward extension of the QP defined for two control tasks in [23] to an arbitrary number of tasks N as well as M set-based tasks is given by

$$\underset{(u,\delta,s)\in\mathbb{R}^{m+N+K}}{\text{minimize}} \quad u^{\mathrm{T}} H(x) u + c^{\mathrm{T}}(x) u + \delta^{\mathrm{T}} W \delta + s^{\mathrm{T}} K s \tag{3.56a}$$

subject to

$$L_{\bar{f}_i} V_{\epsilon,i} + L_{\bar{g}_i} V_{\epsilon,i} u \leq -\frac{\gamma_i}{\epsilon} V_{\epsilon,i} + \delta_i, \quad i = 1, \ldots, N, \tag{3.56b}$$

$$L_f^{\rho_j} h_j + \frac{\partial L_f^{\rho_j - 1} h_j}{\partial x} g u \geq -K_{\alpha,j} \xi_j, \quad j = 1, \ldots, M, \tag{3.56c}$$

where $H(x) \in \mathbb{R}^{m \times m}$ is any positive semi-definite weighting matrix, $c(x) \in \mathbb{R}^m$ is an arbitrary vector of weights, $\delta \in \mathbb{R}^N$ is a vector of slack variables, $W \in \mathbb{R}^{N \times N}$ is a diagonal matrix of penalty parameters and

$$L_{\bar{f}_i} V_{\epsilon,i} = \eta_i^{\mathrm{T}} \left(F_i^{\mathrm{T}} P_{\epsilon,i} + P_{\epsilon,i} F_i \right) \eta_i + 2 \eta_i^{\mathrm{T}} P_{\epsilon,i} G_i b_i, \tag{3.57}$$

$$L_{\bar{g}_i} V_{\epsilon,i} = 2 \eta_i^{\mathrm{T}} P_{\epsilon,i} G_i A_i, \tag{3.58}$$

for $i = 1, \ldots, N$, where A_i and b_i are defined in (3.55). Moreover, note that the omission of slack variables in (3.56c) defines M hard constraints in the form of safety related set-based tasks, and implicitly defines two distinct priority levels, the safety related set-based tasks, and the relaxed equality tasks.

The equality tasks encoded by RES-CLFs are prioritized by adjusting the elements of the diagonal penalty matrix W. The satisfaction of all equality tasks are therefore described by a single objective function through the value of the slack variables δ and the penalty parameters in W. Whenever equality tasks are incompatible, this fact invariably leads to trade-off configurations that do not satisfy any of the tasks. Hence, strict priority between tasks cannot be achieved in the sense that lower priority tasks have no effect on the execution of higher priority tasks. As a result, it is challenging to include lower priority optimization based tasks since they will interfere with more critical higher priority tasks such as end-effector control whenever the tasks are incompatible.

3.2.2 Enforcing strict priority between a selection of tasks

In order to establish more than two strict priority levels, we propose to solve a quadratic program for every priority level as suggested for kinematic control in [11]. The idea is to begin by computing a control input according to (3.56) that only accounts for equality tasks at the highest priority level and potential safety related set-based tasks. Subsequently, a new quadratic program is solved for each priority level, which refines the previously computed solution in an attempt to satisfy lower priority tasks without affecting the execution of higher priority tasks.

3.2.2.1 Motivating example: Two priority levels

Consider $N = 2$ equality-based tasks and $M = 2$ set-based tasks distributed to $k = 3$ priority levels. Three priority levels implies that two QPs are solved when high priority set-based tasks are included. The first quadratic program finds a control input that satisfies the highest priority set-based task, which is safety-related, as well as attempting to satisfy the highest priority equality-based task as well as possible. The first control input is obtained by solving

$$\underset{(u,\delta)\in\mathbb{R}^{m+1}}{\text{minimize}} \quad \frac{1}{2}u^{\mathrm{T}}H(x)u + c^{\mathrm{T}}(x)u + w\delta^2 \tag{3.59a}$$

subject to

$$L_{\bar{f}_1}V_{\epsilon,1} + L_{\bar{g}_1}V_{\epsilon,1}u \leq -\frac{\gamma_1}{\epsilon_1}V_{\epsilon,1} + \delta, \tag{3.59b}$$

$$L_f^{\rho_1}h_1(x) + \frac{\partial L_f^{\rho_1-1}h_1(x)}{\partial x}g(x)u \geq -K_{\alpha,1}\xi_1(x). \tag{3.59c}$$

Note that the absence of a slack variable in (3.59c) implies that this quadratic program enforces strict priority between the safety-related task described by an ECBF and the equality task in (3.59b) described by a RES-CLF.

Now, if the system is redundant with respect to these two tasks, the control input u_1^* can be changed to accommodate additional tasks, without affecting how the two higher priority tasks are executed. This is achieved by solving another quadratic program, viz.

$$\underset{(u,\delta,s)\in\mathbb{R}^{m+2}}{\text{minimize}} \quad \frac{1}{2}u^{\mathrm{T}}H(x)u + c^{\mathrm{T}}(x)u + w\delta^2 + ks^2 \tag{3.60a}$$

subject to

$$L_{\bar{g}_1}V_{\epsilon,1}u \leq L_{\bar{g}_1}V_{\epsilon,1}u_1^*, \tag{3.60b}$$

$$L_{\bar{f}_2}V_{\epsilon,2} + L_{\bar{g}_2}V_{\epsilon,2}u \leq -\frac{\gamma_2}{\epsilon_2}V_{\epsilon,2} + \delta, \tag{3.60c}$$

$$\frac{\partial L_f^{\rho_1-1}h_1(x)}{\partial x}g(x)u \geq \frac{\partial L_f^{\rho_1-1}h_1(x)}{\partial x}g(x)u_1^*, \tag{3.60d}$$

$$L_f^{\rho_2}h_2(x) + \frac{\partial L_f^{\rho_2-1}h_2(x)}{\partial x}g(x)u \geq -K_{\alpha,2}\xi_2(x) - s, \tag{3.60e}$$

where s is a slack variable penalized by the penalty parameter $k > 0$. Equation (3.60b) is obtained by enforcing the condition

$$L_{\bar{f}_1}V_{\epsilon,1} + L_{\bar{g}_1}V_{\epsilon,1}u \leq L_{\bar{f}_1}V_{\epsilon,1} + L_{\bar{g}_1}V_{\epsilon,1}u_1^* \leq -\frac{\gamma_1}{\epsilon_1}V_{\epsilon,1} + \delta. \tag{3.61}$$

Similarly, (3.60d) is obtained from

$$L_f^{\rho_1} h_1(x) + \frac{\partial L_f^{\rho_1 - 1} h_1(x)}{\partial x} g(x) u \geq L_f^{\rho_1} h_1(x) + \frac{\partial L_f^{\rho_1 - 1} h_1(x)}{\partial x} g(x) u_1^* \geq -K_{\alpha,1} \xi_1(x). \tag{3.62}$$

3.2.2.2 Extension to k priority levels

Consider N equality-based tasks and M set-based tasks as in Section 3.2.1. In contrast to (3.56), ECBFs are no longer restricted to the highest priority level. This enables defining ECBFs for non-safety related tasks such as set-based optimization tasks. Moreover, let the tasks be distributed to k priority levels, and let $N_1 + \ldots + N_k = N$ and $M_1 + \ldots + M_k = M$, where N_i and M_i denotes the number of equality and set-based tasks at priority level i, respectively. A control input u_1^* that disregards all lower priority tasks is obtained by solving (3.56) with $i = 1, \ldots, N_1$ and $j = 1, \ldots, M_1$.

If the system is redundant with respect to these $N_1 + M_1$ tasks, the control input u_1^* can be refined without affecting how the N_1 higher priority equality tasks are executed by enforcing

$$L_{\bar{f}_i} V_{\epsilon,i} + L_{\bar{g}_i} V_{\epsilon,i} u \leq L_{\bar{f}_i} V_{\epsilon,i} + L_{\bar{g}_i} V_{\epsilon,i} u_1^* \tag{3.63}$$

which implies $L_{\bar{g}_i} V_{\epsilon,i} u \leq L_{\bar{g}_i} V_{\epsilon,i} u_1^*$ for all $i = 1, \ldots, N_1$. Similarly, the higher priority set-based tasks are unaffected by enforcing

$$\frac{\partial L_f^{\rho_k - 1} h_k}{\partial x} g u \geq \frac{\partial L_f^{\rho_k - 1} h_k}{\partial x} g u_1^*, \tag{3.64}$$

for all $k = 1, \ldots, M_1$. Consider N_2 additional equality-based tasks and M_2 additional set-based tasks. The control input u_2^* is obtained by solving the following

QP

$$
\begin{aligned}
&\underset{u\in\mathbb{R}^m,\delta\in\mathbb{R}^{N_2},s\in\mathbb{R}^{M_2}}{\text{minimize}} \quad u^{\mathrm{T}}H(x)u + c^{\mathrm{T}}(x)u + \delta^{\mathrm{T}}W\delta + s^{\mathrm{T}}Ks\\
&\quad\text{subject to}\\
&L_{\bar{g}_i}V_{\epsilon,i}u \le L_{\bar{g}_i}V_{\epsilon,i}u_1^*, && {\scriptstyle i=1,\dots,N_1,}\\
&L_{\bar{f}_j}V_{\epsilon,j} + L_{\bar{g}_j}V_{\epsilon,j}u \le -\frac{\gamma_j}{\epsilon_j}V_{\epsilon,j} + \delta_j, && {\scriptstyle j=N_1+1,\dots,N_1+N_2,}\\
&\frac{\partial L_f^{\rho_k-1}h_k}{\partial x}gu \ge \frac{\partial L_f^{\rho_k-1}h_k}{\partial x}gu_1^*, && {\scriptstyle k=1,\dots,M_1,}\\
&L_f^{\rho_l}h_l + \frac{\partial L_f^{\rho_l-1}h_l}{\partial x}gu \ge -K_{\alpha,l}\xi_l - s_l, && {\scriptstyle l=M_1+1,\dots,M_1+M_2,}
\end{aligned}
\tag{3.65}
$$

where slack variables s penalized by the elements in the diagonal matrix $K > 0$ have been added to the lower priority set-based tasks enforced through ECBFs to ensure feasibility of the optimization problem.

By observing that the solution u_2^* to (3.65) enforces the constraints

$$L_{\bar{g}_i}V_{\epsilon,i}u_2^* \le L_{\bar{g}_i}V_{\epsilon,i}u_1^*, \tag{3.66}$$

$$\frac{\partial L_f^{\rho_k-1}h_k}{\partial x}gu_2^* \ge \frac{\partial L_f^{\rho_k-1}h_k}{\partial x}gu_1^*, \tag{3.67}$$

for all i and k, it is straightforward to generalize (3.65) to an arbitrary priority level n, viz.

$$
\begin{aligned}
&\underset{u\in\mathbb{R}^m,\delta\in\mathbb{R}^{N_n},s\in\mathbb{R}^{M_n}}{\text{minimize}} \quad u^{\mathrm{T}}H(x)u + c^{\mathrm{T}}(x)u + \delta^{\mathrm{T}}W\delta + s^{\mathrm{T}}Ks\\
&\quad\text{subject to}\\
&L_{\bar{g}_i}V_{\epsilon,i}u \le L_{\bar{g}_i}V_{\epsilon,i}u_{n-1}^*, && {\scriptstyle i=1,\dots,\bar{N}_{n-1},}\\
&L_{\bar{f}_j}V_{\epsilon,j} + L_{\bar{g}_j}V_{\epsilon,j}u \le -\frac{\gamma_j}{\epsilon_j}V_{\epsilon,j} + \delta_j, && {\scriptstyle j=\bar{N}_{n-1}+1,\dots,\bar{N}_n,}\\
&\frac{\partial L_f^{\rho_k-1}h_k}{\partial x}gu \ge \frac{\partial L_f^{\rho_k-1}h_k}{\partial x}gu_{n-1}^*, && {\scriptstyle k=1,\dots,\bar{M}_{n-1},}\\
&L_f^{\rho_l}h_l + \frac{\partial L_f^{\rho_l-1}h_l}{\partial x}gu \ge -K_{\alpha,l}\xi_l - s_l, && {\scriptstyle l=\bar{M}_{n-1}+1,\dots,\bar{M}_n,}
\end{aligned}
\tag{3.68}
$$

where $\bar{N}_n = N_1 + N_2 + \ldots + N_n$ and $\bar{M}_n = M_1 + M_2 + \ldots + M_n$.

Note that the objective function is slightly different at every priority level, since the slack variables δ and s always correspond to tasks at the current priority level. This prevents the occurrence of an optimal compromise between the satisfaction of incompatible tasks, where a solution is found that minimizes the objective function, and is hence optimal, but in which none of the tasks are fully satisfied.

3.2.2.3 Summary

Within this framework, equality tasks are satisfied by enforcing the negative definiteness of the time derivative of rapidly exponentially stabilizing control Lyapunov functions. RES-CLFs are employed to directly control the rate of exponential convergence. Moreover, set-based tasks are satisfied by rendering their valid sets forward invariant, which is achieved by formulating the valid sets as the superlevel sets of exponential control barrier functions.

The procedure is summarized in Algorithm 1, where it is assumed that the N equality and M set-based tasks are distributed to $k+1$ priority levels[1] where N_i and M_i represents the number of equality and set-based tasks at priority level i, respectively. In Algorithm 1, $H(x) \in \mathbb{R}^{m \times m}$ and $c(x) \in \mathbb{R}^m$ are design parameters that weight the control inputs.

Algorithm 1 Task priority CLF-ECBF QP controller

```
Input: $H(x)$, $c(x)$, $V_{\epsilon,i}(\eta_i)$, $i = 1, \dots, N$, $h_j(x)$, $j = 1, \dots, M$.
Output: $u$
Solve (3.56) to obtain $u_1^*$ with $i = 1, \dots, N_1$ and $k = 1, \dots, M_1$.
for $p = 2$ to $k$ do
    Solve (3.68) to obtain $u_p^*$.
end for
return $u = u_k^*$
```

[1] If there are set-based tasks without slack variables at the highest priority level. If not, there are k priority levels.

Chapter 4

Modeling of AIAUVs

This chapter describes the dynamic and kinematic modeling of AIAUVs presented in [54, 55]. The chapter was originally written for [15] and is restated here with minor modifications for completeness.

4.1 Reference Frames

Because AIAUVs in work mode operate in local areas, approximately constant longitude and latitude can be assumed. Moreover, the Earth's rotation can be neglected and a local Earth-fixed North-East-Down tangent frame denoted $\{n\}$ is used for navigation. This frame is assumed to be inertial such that Newton's laws still apply. The origin of the NED frame is fixed, the x-axis points North, the y-axis points East while the z-axis points downwards normal to the Earth's surface. Note that in the navigation literature this frame is often referred to as NED assumed to be inertial or a non-rotating tangent frame [56].

The body-fixed reference frame $\{b\}$ is rigidly attached to the AIAUV and located at the base, the x-axis points forward, z-axis points upwards and the y-axis points sideways to complete the right-handed coordinate system. The reference frame of the end-effector $\{e\}$ is rigidly attached to the front of the AIAUV, where the x-axis points forward, z-axis points upwards and the y-axis points sideways to complete the right-handed coordinate system.

4.2 Kinematic Modeling of AIAUVs

The system configuration of an AIAUV is defined as $\xi = [\eta^{\mathrm{T}}, \theta^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{6+n}$, where $\theta \in \mathbb{R}^n$ represents the joint angles and

$$\eta = \begin{bmatrix} p^n_{nb} \\ \Theta_{nb} \end{bmatrix} \in \mathbb{R}^6, \tag{4.1}$$

where $p^n_{nb} = [N, E, D]^{\mathrm{T}} \in \mathbb{R}^3$ represents the position of the base frame in the aforementioned NED frame, while the orientation between $\{n\}$ and $\{b\}$ is represented by the Euler angles $\Theta^b_{nb} = [\phi_b, \theta_b, \psi_b]^{\mathrm{T}} \in \mathbb{R}^3$. The components of Θ_{nb} are denoted roll, pitch and yaw, respectively. Alternatively, the orientation can be represented by unit quaternions $q = [\eta, \epsilon^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^4$, $\|q\| = 1$, where $\eta \in \mathbb{R}$ is the real part of the quaternion, while $\epsilon \in \mathbb{R}^3$ corresponds to the vector part. By using a unit quaternion representation, the system configuration is defined as $\xi = \left[\eta_q^{\mathrm{T}}, \theta^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{7+n}$ where

$$\eta_q = \begin{bmatrix} p^n_{nb} \\ \eta \\ \epsilon \end{bmatrix} \in \mathbb{R}^7. \tag{4.2}$$

4.2.1 Differential kinematics

The body-fixed velocity of the AIAUV is given by

$$\zeta = \begin{bmatrix} V^b_{nb} \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^{6+n}, \quad V^b_{nb} = \begin{bmatrix} v^b_{nb} \\ \omega^b_{nb} \end{bmatrix} \in \mathbb{R}^6, \tag{4.3}$$

where $v^b_{nb} = [u, v, w]^{\mathrm{T}} \in \mathbb{R}^3$ and $\omega^b_{nb} = [p, q, r]^{\mathrm{T}} \in \mathbb{R}^3$ are the linear and angular velocities of the body-fixed frame, respectively. $\dot{\theta} \in \mathbb{R}^n$ represents the joint velocities.

The relation between the body-fixed velocities and the NED-frame velocities can be expressed by

$$\dot{\xi} = J_{\xi}(\Theta_{nb})\zeta, \tag{4.4}$$

where

$$J_{\xi}(\Theta_{nb}) = \begin{bmatrix} R_b^n(\Theta_{nb}) & 0_{3\times3} & 0_{3\times n} \\ 0_{3\times3} & T_{nb}(\Theta_{nb}) & 0_{3\times n} \\ 0_{n\times3} & 0_{n\times3} & I_n \end{bmatrix}. \tag{4.5}$$

The rotation matrix $R_b^n(\Theta_{nb}) = R_z(\psi_b)R_y(\theta_b)R_x(\phi_b)$ transforms the linear velocity from the body-frame to the NED-frame and is computed according to the zyx convention, which is common practice in guidance, navigation and control applications [57]. The angular velocity transformation matrix that maps the body-fixed angular velocities to the Euler angle derivatives is given by

$$T(\Theta) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix}. \tag{4.6}$$

4.2.1.1 Quaternions and Euler angles

While the Euler angle attitude parametrization is intuitive and only requires three parameters, it suffers from a singularity corresponding to $\theta = \pm 90^\circ$ as observed from (4.6). This singularity is not a problem for surface vehicles, but for an AIAUV it can become problematic if the vehicle is operating close to the singularities. This is why a four-parameter unit quaternion representation is used in all of the simulations conducted as part of this thesis.

By employing unit quaternions, the relation between the body-fixed velocities and the NED-frame velocities can be expressed by

$$\dot{\xi} = J_{\xi}(q)\zeta, \tag{4.7}$$

where

$$J_\xi(q) = \begin{bmatrix} R_b^n(q) & 0_{3\times 3} & 0_{3\times n} \\ 0_{4\times 3} & T_q(q) & 0_{4\times n} \\ 0_{n\times 3} & 0_{n\times 3} & I_n \end{bmatrix}. \tag{4.8}$$

The rotation matrix is given by

$$R_b^n(q) = I_{3\times 3} + 2\eta\epsilon_\times + 2\left(\epsilon_\times\right)^2, \tag{4.9}$$

where $\epsilon_\times \in \mathfrak{so}(3)$ represents the skew-symmetric form of ϵ. The transformation matrix mapping the angular velocity decomposed in the base frame to the quaternion time derivative is given by

$$T_q(q) = \frac{1}{2}\begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix}. \tag{4.10}$$

4.2.2 Differential task kinematics

For an AIAUV, operational space tasks are nonlinear mappings $\sigma_i(\xi) = f_i(\xi)$. The first order differential kinematics are given by

$$\dot{\sigma}_i(\xi,\zeta) = \frac{\partial f_i(\xi)}{\partial \xi} J_\xi(q)\zeta \tag{4.11}$$

$$= J_i\zeta. \tag{4.12}$$

4.2.3 Forward kinematics

An AIAUV consists of $n+1$ links interconnected by n joints, the links are labeled $1,\ldots,(n+1)$, where link 1 is the tail, or base link and link $n+1$ is the head. All joints are single DOF joints, and the AIAUV is assumed to only consist of revolute joints. Joints with multiple DOFs are therefore modeled as two consecutive joints with an additional link between them.

For each link a homogeneous transformation matrix is defined

$$H_i = \begin{bmatrix} R_{ni} & p_{ni} \\ 0_{1\times 3} & 1 \end{bmatrix} \in \mathrm{SE}(3), \tag{4.13}$$

for $i = 1, \ldots, n+1$ which uniquely specifies the pose of link i in the NED frame defined in Section 4.1. Furthermore, let $A_i(\theta_i) \in \mathrm{SE}(3)$ represent the mapping from the coordinate frame defined by H_i to the coordinate frame defined by H_{i+1}, where θ_i is the joint variable of joint i. Given the transformation matrix H, describing the position p_{nb}^n and orientation R_b^n of the base frame in the NED frame, the position and orientation of link $i+1$ is then given by the recursive equations

$$\begin{aligned} H_1 &= H && (4.14)\\ H_{i+1} &= H_i A_i(\theta_i) && (4.15)\\ &= H A_1(\theta_1) A_2(\theta_2) \cdots A_i(\theta_i). && (4.16) \end{aligned}$$

The position and orientation of link $n+1$, or the head frame can then be written

$$H_{n+1} = H A_{1,n}(\theta), \tag{4.17}$$

where $\theta = [\theta_1, \theta_2, \ldots, \theta_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the vector of joint parameters and

$$A_{i,j}(\theta) = \begin{cases} A_i(\theta_i) A_{i+1}(\theta_{i+1}) \cdots A_j(\theta_j), & \text{if } i \le j \\ 0, & \text{if } i > j \end{cases}. \tag{4.18}$$

The instantaneous velocity of a rigid body in terms of its linear and angular components will from now on be described by twists, which are infinitesimal versions of a screw motion. Background material can be found in [58]. Let $a_i = [\beta_i^{\mathrm{T}}, \lambda_i^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$ represent the twist coordinates of joint i, the corresponding twist is given by

$$a^\wedge = \begin{bmatrix} \beta \\ \lambda \end{bmatrix}^\wedge = \begin{bmatrix} \lambda_\times & \beta \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{4.19}$$

where the wedge operator represents the mapping $\wedge : \mathbb{R}^6 \to \mathfrak{se}(3)$ with $\mathfrak{se}(3)$ defined as

$$\mathfrak{se}(3) = \left\{ (\beta, \lambda_\times) : \beta \in \mathbb{R}^3, \lambda_\times \in \mathfrak{so}(3) \right\}. \tag{4.20}$$

The 4×4 matrix $a^\wedge \in \mathfrak{se}(3)$ can be interpreted as a generalization of the skew-symmetric matrix $\omega_\times \in \mathfrak{so}(3)$. The rigid motion associated with rotating and translating along the axis of the twist can be represented by [58]

$$A_i(\theta_i) = A_i(0) e^{a_i^\wedge \theta_i}, \tag{4.21}$$

where the exponential map $e^{a^\wedge \theta} : \mathfrak{se}(3) \to \mathrm{SE}(3)$ is given by

$$e^{a^\wedge \theta_i} = \begin{bmatrix} e^{\lambda_\times \theta} & \left(I - e^{\lambda_\times \theta}\right)(\lambda_\times \beta) + \lambda \lambda^{\mathrm{T}} \beta \theta \\ 0_{1\times 3} & 1 \end{bmatrix}, \quad \omega \neq 0, \tag{4.22}$$

$$e^{\lambda_\times \theta} = I + \lambda_\times \sin\theta + (\lambda_\times)^2 (1 - \cos\theta), \quad \|\lambda\| = 1, \tag{4.23}$$

where $e^{\lambda_\times \theta} : \mathfrak{so}(3) \to \mathrm{SO}(3)$. Assuming that the AIAUV only consists of revolute joints, the twist of each joint is given by

$$a_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \lambda_i \end{bmatrix}, \tag{4.24}$$

where $\lambda_i \in \mathbb{R}^3$ is a unit vector defining the axis of rotation of joint i. Moreover, under the additional assumption that the coordinate frame of link $i+1$ is attached to joint i with its x-axis parallel to the link direction, (4.21) can be written

$$A_i(\theta_i) = A_i(0) \begin{bmatrix} e^{(\lambda_i)_\times \theta_i} & 0 \\ 0_{1\times 3} & 1 \end{bmatrix}, \tag{4.25}$$

with

$$A_i(0) = \begin{bmatrix} I_3 & l_i e_1 \\ 0_{1\times 3} & 1 \end{bmatrix}, \tag{4.26}$$

where l_i is the length of link i and $e_1 = [1, 0, 0]^{\mathrm{T}}$.

The head frame defined by H_{n+1} has its origin at head joint, which is at the back of the head link. The transformation A_e from the head frame to the end-effector frame is given by a pure translation in the x-direction

$$A_e = \begin{bmatrix} I_3 & l_{n+1} e_1 \\ 0_{1\times 3} & 1 \end{bmatrix}, \tag{4.27}$$

such that the position p_e^n and orientation R_{ne} of the end-effector frame relative to the NED frame is found by the transformation

$$H_e = \begin{bmatrix} R_{ne} & p_e^n \\ 0_{1\times 3} & 1 \end{bmatrix} = H_{n+1} A_e. \tag{4.28}$$

4.2.4 Jacobians

The body twist of the base is defined as

$$\left(V_{nb}^b\right)^\wedge = H^{-1}\dot{H}, \tag{4.29}$$

where H is the homogeneous transformation matrix from the base frame to the NED frame. Hence, the body twist of link i is given by

$$\left(V_{ni}^i\right)^\wedge = H_i^{-1}\dot{H}_i, \tag{4.30}$$

where H_i is given by (4.16), (4.18) and (4.21). Since the joint twist given by (4.21) is constant, taking the time derivative of (4.21) yields

$$\dot{A}_i(\theta_i) = A_i(0)e^{a_i^\wedge \theta_i} a_i^\wedge \dot{\theta}_i, \tag{4.31}$$

which yields

$$\begin{align} A_i^{-1}\dot{A}_i &= e^{-a_i^\wedge \theta_i} A_i^{-1}(0) A_i(0) e^{a_i^\wedge \theta_i} a_i^\wedge \theta_i \tag{4.32} \\ &= a_i^\wedge \theta_i. \tag{4.33} \end{align}$$

The body twist of link i is then given by

$$\begin{aligned} \left(V_{ni}^i\right)^\wedge &= H_i^{-1}\dot{H}_i \\ &= A_{1,i-1}^{-1} H^{-1}\dot{H} A_{1,i-1} \\ &+ A_{2,i-1}^{-1} a_1^\wedge A_{2,i-1}\dot{\theta}_i + A_{3,i-1}^{-1} a_2^\wedge A_{3,i-1}\dot{\theta}_2 + \dots \\ &+ A_{i-1,i-1}^{-1} a_{i-2}^\wedge A_{i-1,i-1}\dot{\theta}_{i-2} + a_{i-1}^\wedge \dot{\theta}_{i-1}. \end{aligned} \tag{4.34}$$

In order to proceed the adjoint operator and its inverse has to be defined. The adjoint operator, $\mathrm{Ad}(A_i) : \mathbb{R}^6 \to \mathbb{R}^6$ maps a velocity twist in frame $i+1$ to frame i, representing the adjoint transformation associated with the homogeneous transformation matrix A_i and is defined by

$$(\mathrm{Ad}(H)V)^\wedge = HV^\wedge H^{-1}, \tag{4.35}$$

and its inverse, $\mathrm{Ad}^{-1}(H) : \mathbb{R}^6 \to \mathbb{R}^6$ is defined by

$$\left(\mathrm{Ad}^{-1}(H)V\right)^\wedge = H^{-1}V^\wedge H. \tag{4.36}$$

Both operators have matrix representations given by

$$\mathrm{Ad}(H) = \begin{bmatrix} R & p_\times R \\ 0_{1\times 3} & R \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{4.37}$$

$$\mathrm{Ad}^{-1}(H) = \begin{bmatrix} R^{\mathrm{T}} & -R^{\mathrm{T}} p_{\times} \\ 0_{1\times 3} & R^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{6\times 6}. \tag{4.38}$$

Therefore, (4.34) can be rewritten in body velocity twist coordinates as

$$V_{ni}^{i} = \mathrm{Ad}^{-1}(A_{1,i-1}) V_{nb}^{b} + a_{i-1}\dot{\theta}_{i-1} + \sum_{j=1}^{i-2} \mathrm{Ad}^{-1}(A_{j+1,i}) a_j \dot{\theta}_j. \tag{4.39}$$

The body velocity twist coordinates of the base frame and the joint angle velocities are collected in the vector $\zeta = \left[V_{nb}^{b},\, \dot{\theta}\right]^{\mathrm{T}} \in \mathbb{R}^{6+n}$. Define the Jacobian matrices $J_i \in \mathbb{R}^{6\times(6+n)}$ by

$$V_{ni}^{i} = J_i \zeta, \tag{4.40}$$

which maps the velocity twist coordinates V_{nb}^{b} and joint velocities $\dot{\theta}$ to link velocity twist coordinates V_{ni}^{i} decomposed in their own frame. From inspection of (4.39) the Jacobians are given by

$$J_1 = \begin{bmatrix} I_6 & 0_{6\times n} \end{bmatrix}, \tag{4.41}$$

$$J_{i+1} = \begin{bmatrix} \mathrm{Ad}^{-1}(A_{1,i}) & \mathrm{Ad}^{-1}(A_{2,i}) a_1 & \dots & a_i & 0_{6\times(n-i)} \end{bmatrix} \tag{4.42}$$

$$= \mathrm{Ad}^{-1}(A_i) J_i + \begin{bmatrix} 0_{6\times(5+i)} & a_i & 0_{6\times(n-i)} \end{bmatrix}. \tag{4.43}$$

The time derivatives of the Jacobians are found recursively by differentiation of (4.41) and (4.43) as [54]

$$\dot{J}_1 = 0_{6\times(6+n)}, \tag{4.44}$$

$$\dot{J}_{i+1} = -\,\mathrm{ad}(a_i) J_{i+1} \dot{\theta}_i + \mathrm{Ad}^{-1}(A_i) \dot{J}_i. \tag{4.45}$$

Since the transformation matrix A_e from the head frame to the end-effector frame is constant, the body manipulator Jacobian of the end-effector is therefore given by

$$J_e = \mathrm{Ad}^{-1}\left(A_e\right) J_{n+1}, \tag{4.46}$$

where J_{n+1} is the Jacobian of the head frame. The time derivative of the end-effector Jacobian is

$$\dot{J}_e = \mathrm{Ad}^{-1}\left(A_e\right)\dot{J}_{n+1}. \tag{4.47}$$

4.3 Equations of Motion

The equations of motion in the base frame are given by [54]

$$\dot{\xi} = J_\xi(q)\zeta, \tag{4.48a}$$

$$M(\theta)\dot{\zeta} + C(\theta,\zeta)\zeta + D(\theta,\zeta)\zeta + g(\xi) = \tau, \tag{4.48b}$$

where the control inputs u are mapped to commanded forces and moments τ through the actuator configuration matrix $B(\theta)$, viz.

$$\tau = B(\theta)u, \tag{4.49}$$

where $u = \begin{bmatrix} u_t, u_j \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^m$ consists of the thruster inputs $u_t \in \mathbb{R}^p$ and joint torque inputs $u_j \in \mathbb{R}^n$. The equations of motion can be rewritten in state space form as

$$\dot{x} = f(x) + g(x)u, \tag{4.50}$$

where $x = [x_1^{\mathrm{T}}, x_2^{\mathrm{T}}]^{\mathrm{T}} = [\xi^{\mathrm{T}}, \zeta^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{13+2n}$, $u \in \mathbb{R}^m$ and

$$f(x) = \begin{bmatrix} J_\xi(x_1)x_2 \\ -M(x)^{-1}\left(C(x)x_2 + D(x)x_2 + g(x_1)\right) \end{bmatrix}, \tag{4.51}$$

$$g(x) = \begin{bmatrix} 0 \\ M(x)^{-1}B(x_1) \end{bmatrix}. \tag{4.52}$$

The actuator configuration matrix is given by

$$B(\theta) = \begin{bmatrix} J_1(\theta)^{\mathrm{T}}B_1 & J_2(\theta)^{\mathrm{T}}B_2 & \cdots & J_n(\theta)^{\mathrm{T}}B_n & B_{\mathrm{joint}} \end{bmatrix}, \tag{4.53}$$

where the link thrust configuration matrices B_i are constant and expressed as

$$B_i = \begin{bmatrix} \beta_{t,i,1} & \beta_{t,i,2} & \cdots & \beta_{t,i,p} \\ r_{t,i,1} \times \beta_{t,i,1} & r_{t,i,2} \times \beta_{t,i,2} & \cdots & r_{t,i,p} \times \beta_{t,i,p} \end{bmatrix}, \tag{4.54}$$

where $\beta_{t,i,j}$ and $r_{t,i,j}$ are the thrust direction and point of attack of the jth thruster of link i expressed in the frame of link i. The matrix B_{joint} is given by

$$B_{\text{joint}} = \begin{bmatrix} 0_{6\times p} \\ I_n \end{bmatrix}. \tag{4.55}$$

Note that the thruster inputs also generate torques affecting the joints, while joint torque inputs only affect the joints directly.

The inertia matrix $M(\theta)$ is given by

$$M(\theta) = \sum_{i=1}^{n} J_i^{\mathrm{T}}(\theta) M_i J_i(\theta), \tag{4.56}$$

where M_i is the inertia matrix of link i containing the rigid body mass and inertia matrix and the added mass matrix

$$M_i = M_{R,i} + M_{A,i}, \tag{4.57}$$

where

$$M_{R,i} = \begin{bmatrix} m_i I_3 & m_i \left(r_{g,i}\right)_\times^{\mathrm{T}} \\ m_i \left(r_{g,i}\right)_\times & I_{R,i} \end{bmatrix}, \tag{4.58}$$

and $I_{R,i}$ is the rigid body inertia matrix of link i. By assuming cylindrical links with a common radius r and link lengths l_i the added mass matrix for link i is

$$M_{A,i} = \rho \pi r^2 l_i C_a \begin{bmatrix} \alpha_i & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \frac{1}{2} l_i \\ 0 & 0 & 1 & 0 & -\frac{1}{2} l_i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} l_i & 0 & \frac{1}{3} l_i^2 & 0 \\ 0 & \frac{1}{2} l_i & 0 & 0 & 0 & \frac{1}{3} l_i^2 \end{bmatrix}, \tag{4.59}$$

where α_i is a parameter that permits added mass in surge, ρ is the density of water and C_a is the added mass coefficient.

The Coriolis and centripetal matrix $C(\theta, \zeta)$ is given by

$$C(\theta, \zeta) = \sum_{i=1}^{n} \left(J_i(\theta)^{\mathrm{T}} M_i \dot{J}_i(\theta, \dot{\theta}) - J_i(\theta)^{\mathrm{T}} W_i(\theta, \zeta) J_i(\theta) \right), \tag{4.60}$$

$$W_i(\theta, \zeta) = \begin{bmatrix} 0_{3\times 3} & \left(\left\{ M_i V_{ni}^i \right\}_v \right)_\times \\ \left(\left\{ M_i V_{ni}^i \right\}_v \right)_\times & \left(\left\{ M_i V_{ni}^i \right\}_\omega \right)_\times \end{bmatrix}, \tag{4.61}$$

where $\left\{ M_i V_{ni}^i \right\}_v \in \mathbb{R}^3$ and $\left\{ M_i V_{ni}^i \right\}_\omega \in \mathbb{R}^3$ are the first three and final three entries of $M_i V_{ni}^i$, respectively. Hydrodynamic damping is modeled by $D(\theta, \zeta)$, which is given by

$$D(\theta, \zeta) = \sum_{i=1}^{n} J_i(\theta)^{\mathrm{T}} D_i(\theta, \zeta) J_i(\theta), \tag{4.62}$$

where $D_i(\theta, \zeta)$ is the hydrodynamic damping matrix of link i, such that $D_i V_{ni}^i$ yields the hydrodynamic forces and moments on link i.

The generalized hydrostatic force $g(\xi)$ is given as

$$g(\xi) = \sum_{i=1}^{n} J_i(\theta)^{\mathrm{T}} g_i(\xi), \tag{4.63}$$

where $g_i(\xi)$ are the hydrostatic forces and moments on link i

$$g_i(\xi) = G_i R_{ni}^{\mathrm{T}} \gamma_0, \tag{4.64}$$

where γ_0 is the constant direction of gravity in the NED frame, and R_{ni} is the rotation matrix from the NED frame to the frame of link i. The matrices $G_i \in \mathbb{R}^{6\times 6}$ are constant and given as

$$G_i = \begin{bmatrix} (\rho v_i g - m_i g)\, I_3 \\ \rho v_i g \left[r_{b,i}\right]_\times - m_i g \left[r_{g,i}\right]_\times \end{bmatrix}, \tag{4.65}$$

where ρ is the density of water, v_i is the effective volume of link i, g is the gravitational acceleration constant, m_i is the mass of link i and $r_{b,i}$ and $r_{g,i}$ is the location of the center of buoyancy and center of gravity of link i expressed in the coordinate frame of link i, respectively.

Chapter 5

AIAUV Control

In this chapter, an extensive case study of the task priority framework developed in Section 3.2 is conducted for an AIAUV application. Section 5.1 presents the AIAUV simulation model, before the set-based and equality-based tasks to be controlled are defined in Section 5.2 and Section 5.3, respectively. Two control laws are developed in Section 5.4, followed by simulation results and a discussion.

5.1 AIAUV Model

The AIAUV simulation model is identical to the one in [59]. The AIAUV consists of 5 links with 4 cardan joints connecting them, which are joints that can rotate about the y and z-axis that they are attached to. The cardan joints are modeled as consecutive 1-DOF joints, thereby introducing short links separating two joints actuated about different axes, creating four new links. The total number of joints and links are therefore 8 and 9, respectively. Odd-numbered joints rotate about the z-axis, and even-numbered joints rotate about the y-axis.

The AIAUV has 7 thrusters in total, two of which are positioned on the third link, pointing in the z and y directions of the coordinate frame of link 3. Three thrusters are positioned on the fifth link, pointing in the x, y and z directions in the coordinate frame of link 5. The last two thrusters are located at the seventh link, pointing in the y and z directions in the coordinate frame of link 7.

5.2 Set-Based Tasks for AIAUV Control

This section introduces the safety related set-based tasks that should be satisfied at all times. Their satisfaction is ensured through exponential control barrier functions (ECBFs).

5.2.1 End-effector collision avoidance

To avoid a collision between the end-effector and some obstacle, the scalar distance measure between them is employed as a set-based task

$$\sigma_a = \sqrt{\left(p^n_{\text{obs}} - p^n_{ne}\right)^{\mathrm{T}} \left(p^n_{\text{obs}} - p^n_{ne}\right)}, \tag{5.1}$$

which must be differentiated twice for the control input to appear, viz.

$$\dot{\sigma}_a = J_a \zeta, \tag{5.2}$$

$$\ddot{\sigma}_a = J_a M^{-1} \left(Bu - C\zeta - D\zeta - g\right) + \dot{J}_a \zeta, \tag{5.3}$$

where the task Jacobian $J_a \in \mathbb{R}^{1\times(6+n)}$ is given by [60]

$$J_a = \frac{-\left(p^n_{\text{obs}} - p^n_{ne}\right)^{\mathrm{T}}}{\sigma_a} \begin{bmatrix} R^n_e(q) & 0_{3\times3} \end{bmatrix} J_e \zeta, \tag{5.4}$$

where the end-effector Jacobian $J_e \in \mathbb{R}^{6\times(6+n)}$ is given by the recursive equations (4.41), (4.43) and (4.46). In order to obtain an expression for the time derivative of the task Jacobian, consider the term

$$x = \frac{-(p_o - p)}{\sigma_a}, \tag{5.5}$$

the time derivative is given by

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\partial x}{\partial p}\dot{p} \tag{5.6}$$

$$= \frac{I\sigma_a + (p_o - p)\frac{\partial \sigma_a}{\partial p}}{\sigma_a^2} v \tag{5.7}$$

$$= \left(\frac{I}{\sigma_a} - \frac{\left(p_o - p\right)\left(p_o - p\right)^{\mathrm{T}}}{\sigma_a^3} \right) v, \tag{5.8}$$

transposing both sides yields

$$\begin{aligned} \frac{\mathrm{d}x^{\mathrm{T}}}{\mathrm{d}t} &= v^{\mathrm{T}} \left(\frac{I}{\sigma_a} - \frac{\left(p_o - p\right)\left(p_o - p\right)^{\mathrm{T}}}{\sigma_a^3} \right)^{\mathrm{T}} \qquad (5.9) \\ &= v^{\mathrm{T}} \left(\frac{I}{\sigma_a} - \frac{\left(p_o - p\right)\left(p_o - p\right)^{\mathrm{T}}}{\sigma_a^3} \right). \qquad (5.10) \end{aligned}$$

Hence, the time derivative of the task Jacobian is given by

$$\begin{aligned} \dot{J}_a = \left(v_{ne}^n\right)^{\mathrm{T}} \left(\frac{I}{\sigma_a} - \frac{\left(p_{\mathrm{obs}}^n - p_{ne}^n\right)\left(p_{\mathrm{obs}}^n - p_{ne}^n\right)^{\mathrm{T}}}{\sigma_a^3} \right) \begin{bmatrix} R_e^n & 0_{3\times3} \end{bmatrix} J_e \qquad (5.11) \\ + \frac{-\left(p_{\mathrm{obs}}^n - p_{ne}^n\right)^{\mathrm{T}}}{\sigma_a} \left(\begin{bmatrix} R_e^n \left[\omega_{ne}^e\right]_\times & 0_{3\times3} \end{bmatrix} J_e + \begin{bmatrix} R_e^n & 0_{3\times3} \end{bmatrix} \dot{J}_e \right), \qquad (5.12) \end{aligned}$$

where the time derivative of the end-effector Jacobian $\dot{J}_e\left(\theta\right) \in \mathbb{R}^{6\times6+n}$ is given by the recursive equations (4.44), (4.45) and (4.47).

5.2.2 Actuator singularity avoidance

The rank deficiency of the actuator configuration matrix $B(\theta)$ was pointed out in [1], and implies that no force or moment can be generated in certain directions in the vector space $\mathbb{R}^{6+n}$ belonging to τ. Inspired by the manipulability index [44], the actuation index task is introduced to prevent singular configurations and is defined by

$$\sigma_b := \det\left(B(\theta)B(\theta)^{\mathrm{T}}\right), \tag{5.13}$$

where $B(\theta)$ is the actuator configuration matrix. The actuator singularity avoidance task must be differentiated twice with respect to time for the control input to show up, as seen from

$$\dot{\sigma}_b = J_b\zeta, \tag{5.14}$$

$$\ddot{\sigma}_b = J_b M^{-1} \left(Bu - C\zeta - D\zeta - g\right) + \dot{J}_b \zeta, \tag{5.15}$$

where the task Jacobian is given by

$$J_b = \begin{bmatrix} 0_{1\times 6} & \frac{\partial \sigma_b}{\partial \theta_1} & \frac{\partial \sigma_b}{\partial \theta_2} & \frac{\partial \sigma_b}{\partial \theta_3} & \cdots & \frac{\partial \sigma_b}{\partial \theta_n} \end{bmatrix}, \tag{5.16}$$

and the actuation index derivative is given by [61]

$$\frac{\partial \sigma_b}{\partial \theta_i} = 2\sigma_b \operatorname{Tr}\left(\frac{\partial^2 B}{\partial \theta_i \partial B}^{\dagger} \right), \tag{5.17}$$

where $B^{\dagger}$ is the right Moore-Penrose pseudoinverse of B. Furthermore, the task Jacobian derivative is given by [61]

$$\dot{J}_b = \begin{bmatrix} \frac{\partial}{\partial \theta}\left(0_{1\times 6}\right)\dot{\theta} & \frac{\partial}{\partial \theta}\left(\frac{\partial \sigma_b}{\partial \theta_1}\right)\dot{\theta} & \frac{\partial}{\partial \theta}\left(\frac{\partial \sigma_b}{\partial \theta_2}\right)\dot{\theta} & \cdots & \frac{\partial}{\partial \theta}\left(\frac{\partial \sigma_b}{\partial \theta_n}\right)\dot{\theta} \end{bmatrix} \tag{5.18a}$$

$$= \begin{bmatrix} 0_{6\times 1} \\ \frac{\partial^2 \sigma_b}{\partial \theta_1^2}\dot{\theta}_1 + \frac{\partial^2 \sigma_b}{\partial \theta_1 \partial \theta_2}\dot{\theta}_2 + \frac{\partial^2 \sigma_b}{\partial \theta_1 \partial \theta_3}\dot{\theta}_3 + \cdots + \frac{\partial^2 \sigma_b}{\partial \theta_1 \partial \theta_n}\dot{\theta}_n \\ \frac{\partial^2 \sigma_b}{\partial \theta_2 \partial \theta_1}\dot{\theta}_1 + \frac{\partial^2 \sigma_b}{\partial \theta_2^2}\dot{\theta}_2 + \frac{\partial^2 \sigma_b}{\partial \theta_2 \partial \theta_3}\dot{\theta}_3 + \cdots + \frac{\partial^2 \sigma_b}{\partial \theta_2 \partial \theta_n}\dot{\theta}_n \\ \frac{\partial^2 \sigma_b}{\partial \theta_3 \partial \theta_1}\dot{\theta}_1 + \frac{\partial^2 \sigma_b}{\partial \theta_3 \partial \theta_2}\dot{\theta}_2 + \frac{\partial^2 \sigma_b}{\partial \theta_3^2}\dot{\theta}_3 + \cdots + \frac{\partial^2 \sigma_b}{\partial \theta_3 \partial \theta_n}\dot{\theta}_n \\ \vdots \\ \frac{\partial^2 \sigma_b}{\partial \theta_n \partial \theta_1}\dot{\theta}_1 + \frac{\partial^2 \sigma_b}{\partial \theta_n \partial \theta_2}\dot{\theta}_2 + \frac{\partial^2 \sigma_b}{\partial \theta_n \partial \theta_3}\dot{\theta}_3 + \cdots + \frac{\partial^2 \sigma_b}{\partial \theta_n^2}\dot{\theta}_n \end{bmatrix}^{\mathrm{T}} \tag{5.18b}$$

$$= \begin{bmatrix} 0_{6\times 1} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} & & 0_{6\times n} & & \\ \frac{\partial^2 \sigma_b}{\partial \theta_1^2} & \frac{\partial^2 \sigma_b}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 \sigma_b}{\partial \theta_3 \partial \theta_1} & \cdots & \frac{\partial^2 \sigma_b}{\partial \theta_n \partial \theta_1} \\ \frac{\partial^2 \sigma_b}{\partial \theta_1 \partial \theta_2} & \frac{\partial^2 \sigma_b}{\partial \theta_2^2} & \frac{\partial^2 \sigma_b}{\partial \theta_3 \partial \theta_2} & \cdots & \frac{\partial^2 \sigma_b}{\partial \theta_n \partial \theta_2} \\ \frac{\partial^2 \sigma_b}{\partial \theta_1 \partial \theta_3} & \frac{\partial^2 \sigma_b}{\partial \theta_2 \partial \theta_3} & \frac{\partial^2 \sigma_b}{\partial \theta_3^2} & \cdots & \frac{\partial^2 \sigma_b}{\partial \theta_n \partial \theta_3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \sigma_b}{\partial \theta_1 \partial \theta_n} & \frac{\partial^2 \sigma_b}{\partial \theta_2 \partial \theta_n} & \frac{\partial^2 \sigma_b}{\partial \theta_3 \partial \theta_n} & \cdots & \frac{\partial^2 \sigma_b}{\partial \theta_n^2} \end{bmatrix}, \tag{5.18c}$$

where the cross partials are given by

$$\frac{\partial^2 \sigma_b}{\partial\theta_i \partial\theta_j} = 2\frac{\partial\sigma_b}{\partial\theta_j} \operatorname{Tr}\left(\frac{\partial B}{\partial\theta_i} B^+\right) + 2\sigma_b \operatorname{Tr}\left(\frac{\partial^2 B}{\partial\theta_j \partial\theta_i} B^+ + \frac{\partial B}{\partial\theta_i}\left\{\left(\frac{\partial B}{\partial\theta_j}\right)^{\mathrm{T}}\right.\right. \\ \left.\left. - B^+\left[\frac{\partial B}{\partial\theta_j} B^{\mathrm{T}} + B\left(\frac{\partial B}{\partial\theta_j}\right)^{\mathrm{T}}\right]\right\}\left(BB^{\mathrm{T}}\right)^{-1}\right). \tag{5.19}$$

When implemented, the actuator configuration matrix $B(\theta)$ is computed symbolically in order to find the partial and cross partial derivatives $\frac{\partial B}{\partial\theta_i}$ and $\frac{\partial}{\partial\theta_j}\frac{\partial B}{\partial\theta_i}$. The computations are performed offline and stored in a look-up table depending on the joint angle due to the computational complexity.

5.2.3 Joint limit avoidance

To avoid exceeding mechanical joint limits a set-based joint limit avoidance task is defined by

$$\sigma_c := \theta, \tag{5.20}$$

which when differentiated twice with respect to time yields

$$\dot{\sigma}_c = J_c \zeta, \tag{5.21}$$

$$\ddot{\sigma}_c = J_c M^{-1}\left(Bu - C\zeta - D\zeta - g\right), \tag{5.22}$$

where the task Jacobian is constant and given by

$$J_c = \begin{bmatrix} 0_{n\times 6} & I_{n\times n} \end{bmatrix}. \tag{5.23}$$

5.2.4 Exponential control barrier functions

In order to ensure that the aforementioned set-based tasks are satisfied, exponential control barrier functions are employed. Because the collision avoidance and actuator singularity avoidance tasks are only bounded from below, the ECBFs of these tasks are given by

$$h_1 = \sigma_a - \sigma_{a,\min}, \tag{5.24}$$

$$h_2 = \sigma_b - \sigma_{b,\min}. \tag{5.25}$$

The joint limit avoidance task is bounded from below and above, i.e. $\sigma_{c,\min} \leq \sigma_c \leq \sigma_{c,\max}$. Sixteen ECBFs are therefore required, which are given by

$$h_{2+i} = \sigma_{c,i} - \theta_{i,\min}, \tag{5.26}$$

$$h_{10+i} = \theta_{i,\max} - \sigma_{c,i}, \tag{5.27}$$

for $i = 1, \dots, 8$. All of the set-based tasks only depend on the system configuration, ξ. Hence, they have to be differentiated twice for the input to appear, as seen from the second time derivative of the set-based tasks in (5.3), (5.15) and (5.22). Differentiating the ECBFs twice with respect to time and inserting the second time derivatives of the set-based tasks together with the equations of motion (4.48) results in

$$\ddot{h}_1 = \dot{J}_a\zeta - J_a M^{-1}\left(C\zeta + D\zeta + g\right) + J_a M^{-1}Bu, \tag{5.28}$$

$$\ddot{h}_2 = \dot{J}_b\zeta - J_b M^{-1}\left(C\zeta + D\zeta + g\right) + J_b M^{-1}Bu, \tag{5.29}$$

$$\ddot{h}_{2+i} = -J_{c,i} M^{-1}\left(C\zeta + D\zeta + g\right) + J_{c,i} M^{-1}Bu, \tag{5.30}$$

$$\ddot{h}_{10+i} = J_{c,i} M^{-1}\left(C\zeta + D\zeta + g\right) - J_{c,i} M^{-1}Bu, \tag{5.31}$$

where $J_{c,i}$ is the ith row of J_c. These equations can be rewritten in terms of the state space model (4.50) as

$$\ddot{h}_i(x) = L_f^2 h_i(x) + \frac{\partial L_f h_i(x)}{\partial x} g(x)u, \tag{5.32}$$

for $i = 1, \dots, 18$.

5.2.5 Valid domains

The obstacle to be avoided by the collision avoidance task is a sphere with radius $r_{\text{obs}} = 0.3\,\text{m}$ centered at $p_{\text{obs}} = [2.5\,\text{m}, 0.5\,\text{m}, -10\,\text{m}]^{\text{T}}$, where the coordinates are given in a North-West-Up coordinate frame. In order to visualize how the AIAUV avoids the obstacle in a 3D plot, a threshold is defined such that the minimum distance to the obstacle becomes slightly larger. The valid domains for all set-based tasks are shown in Table 5.1.

Table 5.1: The valid domains for the set-based tasks.

	σ_a	σ_b	$\sigma_{c,i}$
$\sigma_{\min}$	$r_{obs} + 0.02\,\mathrm{m}$	0.1	$-60°$
$\sigma_{\max}$	∞	∞	$60°$

5.3 Equality-Based Tasks for AIAUV Control

This section introduces the equality-based tasks and derives their input-output dynamics. Moreover, rapidly exponentially stabilizing control Lyapunov functions (RES-CLFs) are defined for each task, and conditions on the RES-CLF time derivatives that ensure task stability are derived. Finally, the desired equality task values are discussed, where a reference model that is presented to smooth out discontinuous steps in the reference signal.

5.3.1 End-effector configuration control

The end-effector configuration task is divided into two tasks at the same priority level, namely, an end-effector position task and an attitude task. In this way, individual RES-CLFs can be defined for position and attitude control, which means that the rates of exponential convergence can differ.

5.3.1.1 End-effector position control

The pose of the end-effector relative to the NED frame is given by the forward kinematics as described in Section 4.2.3. Hence, the position of the end-effector can be found from

$$p^n_{ne} = \begin{bmatrix} I_3 & 0_{3\times1} \end{bmatrix} H_{n+1}(\theta) \begin{bmatrix} l_{n+1}e_1 \\ 1 \end{bmatrix}, \tag{5.33}$$

where $e_1 \in \mathbb{R}^3$ is the unit vector in the x-direction, and the base position p^n_{nb} is found by solving (4.48). The end-effector position error is defined by

$$y_1 := p^n_{ne} - p^n_{ne,d}, \tag{5.34}$$

which must be differentiated twice with respect to time before the input shows up, viz.

$$\dot{y}_1 = J_1\zeta - \dot{p}_d, \tag{5.35}$$

$$\ddot{y}_1 = J_1 M^{-1} \left(Bu - C\zeta - D\zeta - g\right) + \dot{J}_1\zeta - \ddot{p}_d, \tag{5.36}$$

where the Jacobian and its time derivative are given by

$$J_1 = \begin{bmatrix} R^n_e(q) & 0_{3\times3} \end{bmatrix} J_e(\theta) \in \mathbb{R}^{3\times(6+n)}, \tag{5.37}$$

$$\dot{J}_1 = \begin{bmatrix} R^n_e(q)\left[\omega^e_{ne}\right]_\times & 0_{3\times3} \end{bmatrix} J_e(\theta) + \begin{bmatrix} R^n_e(q) & 0_{3\times3} \end{bmatrix} \dot{J}_e(\theta). \tag{5.38}$$

5.3.1.2 End-effector attitude control

For attitude or orientation control, a unit quaternion representation is employed, where the end-effector quaternion $q = [\eta, \epsilon^{\mathrm{T}}]^{\mathrm{T}}$ can be obtained from the rotation matrix R^n_e given by the forward kinematics. From a quaternion $q_d = [\eta_d, \epsilon_d^{\mathrm{T}}]^{\mathrm{T}}$ describing the desired attitude, a corresponding desired rotation matrix R^n_d can be calculated. The rotation matrix representing the attitude error between the desired end-effector attitude and the end-effector attitude given by the forward kinematics is given by [60]

$$\tilde{R} = R^n_d \left(R^n_e\right)^{\mathrm{T}}. \tag{5.39}$$

The quaternion associated with $\tilde{R}$ can be computed from the quaternion product $\tilde{q} = q_d \otimes q^*$ given by

$$\tilde{\eta} = \eta_d \eta + \epsilon_d^{\mathrm{T}} \epsilon, \tag{5.40}$$

$$\tilde{\epsilon} = \eta \epsilon_d - \eta_d \epsilon + \epsilon_\times \epsilon_d, \tag{5.41}$$

where $q^* = [\eta, -\epsilon^{\mathrm{T}}]^{\mathrm{T}}$ is the conjugate of the end-effector quaternion. Because the rotation matrix representing two aligned frames is given by $\tilde{R} = I$, which corresponds to the quaternion $\tilde{q} = [1, 0^{\mathrm{T}}]^{\mathrm{T}}$, it is sufficient to represent the attitude error as the three-dimensional imaginary part $\tilde{\epsilon}$ of the quaternion error vector $\tilde{q}$. Define the end-effector attitude error by

$$y_2 := \tilde{\epsilon} = \eta \epsilon_d - \eta_d \epsilon + \epsilon_\times \epsilon_d. \tag{5.42}$$

Note that the attitude error is not a simple subtraction between a measured and desired value. Differentiating the attitude error with respect to time yields [45]

$$\dot{y}_2 = \frac{1}{2} \left(\tilde{\eta} I_3 - \tilde{\epsilon}_\times \right) \tilde{\omega}_{ne}^n, \tag{5.43}$$

where $\tilde{\omega}_{ne}^n = \omega_d - \omega_{ne}^n$ is the angular velocity error, expressed in the NED frame. Differentiating with respect to time once more yields

$$\ddot{y}_2 = \frac{1}{2} \left[\left(\dot{\tilde{\eta}} I_3 - \dot{\tilde{\epsilon}}_\times \right) \tilde{\omega}_{ne}^n + \left(\tilde{\eta} I_3 - \tilde{\epsilon}_\times \right) \left(\dot{\omega}_d - \frac{\mathrm{d}}{\mathrm{d}t} \left(R_e^n \omega_{ne}^e \right) \right) \right], \tag{5.44}$$

where

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \left(R_e^n \omega_{ne}^e \right) &= \frac{\mathrm{d}}{\mathrm{d}t} \left(\begin{bmatrix} 0_{3\times3} & R_e^n \end{bmatrix} J_e \zeta \right) &(5.45)\\ &= R_e^n \underbrace{\left[\omega_{ne}^e \right]_\times \omega_{ne}^e}_{=0} + \begin{bmatrix} 0_{3\times3} & R_e^n \end{bmatrix} \dot{J}_e \zeta + \begin{bmatrix} 0_{3\times3} & R_e^n \end{bmatrix} J_e \dot{\zeta} &(5.46)\\ &= \begin{bmatrix} 0_{3\times3} & R_e^n \end{bmatrix} \left(\dot{J}_e \zeta + J_e M^{-1} \left(Bu - C\zeta - D\zeta - g \right) \right), &(5.47) \end{aligned}$$

where $J_e \in \mathbb{R}^{6\times(6+n)}$ is the end-effector Jacobian, which is given by the recursive equations (4.41), (4.43) and (4.46), while $\dot{J}_e \in \mathbb{R}^{6\times(6+n)}$ is the time derivative of the end-effector Jacobian, which is given by (4.44), (4.45) and (4.47). Equation (5.44) is affine in the control input u and can be rewritten

$$\ddot{y}_2(x) = b_2(x) + A_2(x)u, \tag{5.48}$$

where

$$A_2(x) = -\frac{1}{2}\left(\tilde{\eta} I_3 - \tilde{\epsilon}_\times\right)\begin{bmatrix}0_{3\times 3} & R_e^n\end{bmatrix} J_e M^{-1} B, \tag{5.49}$$

and

$$\begin{aligned} b_2(x) = \frac{1}{2}\Bigg[&\left(\dot{\tilde{\eta}} I_3 - \dot{\tilde{\epsilon}}_\times\right)\tilde{\omega}_{ne}^n + \\ &\left(\tilde{\eta} I_3 - \tilde{\epsilon}_\times\right)\left(\dot{\omega}_d - R_e^n \dot{J}_{e,\omega}\zeta - \begin{bmatrix}0_{3\times 3} & R_e^n\end{bmatrix} J_e M^{-1}\left(C\zeta + D\zeta + g\right)\right)\Bigg], \end{aligned} \tag{5.50}$$

with

$$\dot{\tilde{\eta}} = -\frac{1}{2}\epsilon^{\mathrm{T}}\tilde{\omega}_{ne}^n. \tag{5.51}$$

5.3.2 Base position control

The AIAUV essentially has two end-effectors, since there is no well-defined base from a physical perspective. Nonetheless, the base is defined as the tail link of the AIAUV in this thesis, while the end-effector is at the front of the head link. The base positioning task will serve as a lower priority equality task, in this way, the effect of incompatible tasks on the control performance can be tested.

The position of the base in the NED frame is given by $p_{nb}^n \in \mathbb{R}^3$, the base position position error is defined by

$$y_3 := p_{nb}^n - p_{b,d}^n, \tag{5.52}$$

which must be differentiated with respect to time twice for the input to appear, viz.

$$\dot{y}_3 = J_3\zeta - \dot{p}^n_{b,d}, \tag{5.53}$$

$$\ddot{y}_3 = J_3 M^{-1}\left(Bu - C\zeta - D\zeta - g\right) + \dot{J}_3\zeta - \ddot{p}^n_{b,d}, \tag{5.54}$$

where the Jacobian $J_3 \in \mathbb{R}^{3\times(6+n)}$ and its time derivative are given by

$$J_3 = \begin{bmatrix} R^n_b(q) & 0_{3\times(3+n)} \end{bmatrix}, \tag{5.55}$$

$$\dot{J}_3 = \begin{bmatrix} R^n_b(q)\left(\omega^b_{nb}\right)_\times & 0_{3\times(3+n)} \end{bmatrix}. \tag{5.56}$$

$$\tag{5.57}$$

5.3.3 Joint velocity regulation

The end-effector configuration and base positioning tasks only consume 9 DOFs, which means that internal instability will occur if the zero dynamics is not asymptotically stable. In order to ensure system stability, a task at the lowest priority level is defined which attempts to regulate the first $n-1$ joint velocities to zero, thereby consuming any residual DOFs in the system. The task is defined by the error

$$y_4 = J_4\zeta, \tag{5.58}$$

which depends on the generalized system velocities, and hence only has to be differentiated once for the control input to appear, viz.

$$\dot{y}_4 = J_4 M^{-1}\left(Bu - C\zeta - D\zeta - g\right), \tag{5.59}$$

where the Jacobian $J_4 \in \mathbb{R}^{(n-1)\times(6+n)}$ is constant and given by

$$J_4 = \begin{bmatrix} 0_{(n-1)\times 6} & I_{n-1} & 0_{(n-1)\times 1} \end{bmatrix}. \tag{5.60}$$

Note that end-effector attitude control implicitly determines the angle of the nth joint, which ensures stability of the nth joint angle. Finally, remark that in the robotics literature, it is common to define a null space tasks similar to this one at the configuration level, and thereby also attempting to obtain a desired pose while regulating the joint velocities to zero [62].

5.3.4 RES-CLFs

As shown in the preceding sections, the end-effector configuration and base positioning tasks have to be differentiated twice in order for the input to appear, which yields transverse dynamics on the form

$$\ddot{y}_i = b_i(x) + A_i(x)u. \tag{5.61}$$

The joint velocity regulation task is a function of the generalized system velocity, which yields transverse dynamics of the form

$$\dot{y}_4 = b_4(x) + A_4(x)u. \tag{5.62}$$

By defining $\eta_i = [y_i^\mathrm{T}, \dot{y}_i^\mathrm{T}]^\mathrm{T}$ for $i = 1, 2, 3$ and $\eta_4 = y_4$, the transverse dynamics (5.61) and (5.62) can be rewritten as

$$\dot{\eta}_i = \bar{f}_i(x) + \bar{g}_i(x)u, \tag{5.63}$$

where $\bar{f}_i(x) = F_i\eta + G_i b_i(x)$ and $\bar{g}_i(x) = G_i A_i(x)$ with

$$F_i = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad G_i = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad i = 1, 2, 3 \tag{5.64}$$

$$F_4 = 0_{(n-1)\times(n-1)}, \quad G_4 = I_{n-1}, \tag{5.65}$$

for the end-effector configuration and base positioning tasks, where 0 is a 3 × 3 matrix of zeros and I is the 3 × 3 identity matrix.

RES-CLFs are then defined for each task as follows

$$V_{\epsilon,i}(\eta_i) := \eta_i^\mathrm{T} \begin{bmatrix} \frac{1}{\epsilon_i} I & 0 \\ 0 & I \end{bmatrix} P_i \begin{bmatrix} \frac{1}{\epsilon_i} I & 0 \\ 0 & I \end{bmatrix} \eta_i = \eta_i^\mathrm{T} P_{\epsilon,i} \eta_i, \quad i = 1, 2, 3 \tag{5.66}$$

$$V_{\epsilon,4}(\eta_4) := \left(\frac{1}{\epsilon_4}\right)^2 \eta_4^{\mathrm{T}} P_4 \eta_4 = \eta_4^{\mathrm{T}} P_{\epsilon,4} \eta_4, \tag{5.67}$$

where $P_i = P_i^{\mathrm{T}} > 0$ is the solution to the continuous time algebraic Riccati equation

$$F_i^{\mathrm{T}} P_i + P_i F_i - P_i G_i G_i^{\mathrm{T}} P_i + Q_i = 0, \tag{5.68}$$

where the positive definite matrix Q_i is chosen to be the identity matrix. From the analysis in Section 2.4.1, $V_{\epsilon,i}$ is a RES-CLF for (5.63) as long as $A_i(x)$ has linearly independent rows. Rapid exponential stability of the task variables can be ensured by selecting the control input $u \in \mathbb{R}^m$ such that

$$L_{\bar{f}_i} V_{\epsilon,i}(\eta_i) + L_{\bar{g}_i} V_{\epsilon,i}(\eta) u \leq -\frac{\gamma_i}{\epsilon_i} V_{\epsilon,i}(\eta_i), \tag{5.69}$$

where $\gamma_i := \frac{\lambda_{\min}(Q_i)}{\lambda_{\max}(P_i)}$ and the Lie derivatives are given by

$$L_{\bar{f}_i} V_{\epsilon,i}(\eta_i) = \eta_i^{\mathrm{T}} \left(F^{\mathrm{T}} P_{\epsilon,i} + P_{\epsilon,i} F \right) \eta + 2\eta_i^{\mathrm{T}} P_{\epsilon,i} G b_i, \tag{5.70}$$

$$L_{\bar{g}_i} V_{\epsilon,i}(\eta_i) = 2\eta_i^{\mathrm{T}} P_{\epsilon,i} G A_i. \tag{5.71}$$

5.3.5 Desired equality task values

The end-effector configuration task consists of a series of constants steps at 50 second intervals from $t = 0$ to $t = 350$. The base positioning task is constant and does not change, with the goal of minimizing base movement while reconfiguring the end-effector. From $t \geq 350\,\mathrm{s}$, the end-effector position is commanded outside of the manipulator workspace (when the base is kept at its current position) which implies that the base positioning task is no longer compatible with the end-effector positioning task. This is done to verify that tasks at lower priority levels have no effect on how higher priority tasks are executed in the novel hierarchical task priority framework from Section 3.2.2, ensuring that the end-effector position is reached perfectly at the expense of larger steady state errors in the lower priority base positioning task.

The discontinuities in the end-effector configuration set-points are smoothened by employing a third order reference model. A third order model was chosen to obtain continuous acceleration references. In this way, large and discontinuous jumps in the end-effector configuration set-point are avoided, mitigating excessive rates of change of the control inputs when the set-points change, which manifests itself as overshoots and slow convergence when control input rate constraints are employed. The reference model is given by [57]

$$\sigma_d^{(3)} + (2\Lambda + I)\,\Omega\ddot{\sigma}_d + (2\Lambda + I)\,\Omega^2\dot{\sigma}_d + \Omega^3\sigma_d = \Omega^3\sigma_r, \tag{5.72}$$

where $\Lambda > 0$ and $\Omega > 0$ are diagonal design matrices of relative damping ratios and natural frequencies, $\sigma_r = \left[p_{n,d}^{\mathrm{T}}, q_d^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^7$ is the constant (and discontinuous) reference signal and $\sigma_{1,d}, \dot{\sigma}_{1,d}$ and $\ddot{\sigma}_d$ are the smoothened desired end-effector configuration, linear and angular velocity and linear and angular acceleration, respectively. The first and second time derivative of the desired quaternion q_d is mapped into desired angular velocities ω_d and accelerations $\dot{\omega}_d$ through

$$\omega_d = T_q^{\dagger}(q_d)\dot{q}_d, \tag{5.73}$$

$$\dot{\omega}_d = T_q^{\dagger}(q_d)\left(\ddot{q}_d - \dot{T}_q(q_d)q_d\right), \tag{5.74}$$

where $T_q^{\dagger} \in \mathbb{R}^{3\times 4}$ is the Moore-Penrose pseudoinverse of T_q and

$$\dot{T}_q(q_d) = \frac{\partial T_q}{\partial q}\left(\dot{q} \otimes I_3\right), \tag{5.75}$$

with

$$\frac{\partial T_q}{\partial q} = \begin{bmatrix} \frac{\partial T}{\partial \eta} & \frac{\partial T}{\partial \epsilon_1} & \frac{\partial T}{\partial \epsilon_2} & \frac{\partial T}{\partial \epsilon_3} \end{bmatrix} \tag{5.76}$$

$$= \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 12}. \tag{5.77}$$

$$\tag{5.78}$$

5.4 Control Laws

In this section, the two control schemes from Section 3.2 are developed for an AIAUV performing the tasks described in the preceding sections. The objective is to control the configuration of the end-effector while minimizing base movement, avoiding a collision with a spherical obstacle, respecting the joint limits and avoiding configurations for which the actuator configuration matrix is singular.

The design matrix $H(x)$ and design vector $c(x)$ in the objective functions of the quadratic programs are selected by following the approach in Section 3.1.3, where the virtual control input $\mu = Au + b$ was minimized. In terms of u, this yields

$$\mu^{\mathrm{T}}\mu = u^{\mathrm{T}}A^{\mathrm{T}}Au + 2b^{\mathrm{T}}Au + b^{\mathrm{T}}b, \tag{5.79}$$

which implies that $H(x) = A^{\mathrm{T}}A$ and $c^{\mathrm{T}}(x) = 2b^{\mathrm{T}}A$.

5.4.1 *N* task CLF-ECBF QP controller

The control input is found from (3.56) with $H(x) = A^{\mathrm{T}}A$, $c^{\mathrm{T}}(x) = 2b^{\mathrm{T}}A$ and the ECBFs and RES-CLFs from Section 5.2.4 and Section 5.3.4, respectively. The modified quadratic program is

$$\underset{u\in\mathbb{R}^{15},\delta\in\mathbb{R}^{4}}{\text{minimize}} \quad u^{\mathrm{T}}A^{\mathrm{T}}Au + 2b^{\mathrm{T}}Au + \delta^{\mathrm{T}}W\delta \tag{5.80a}$$

subject to

$$L_{\bar{f}_j}V_{\epsilon,j} + L_{\bar{g}_j}V_{\epsilon,j}u \leq -\frac{\gamma_j}{\epsilon_j}V_{\epsilon,j} + \delta_j, \quad j = 1, \ldots, 4, \tag{5.80b}$$

$$L_f^2 h_k + \frac{\partial L_f h_k}{\partial x} gu \geq -K_{\alpha,k}\xi_k, \qquad k = 1, \ldots, 18, \tag{5.80c}$$

$$-50\,\mathrm{N} \leq u \leq 50\,\mathrm{N}, \tag{5.80d}$$

$$-0.1\,\mathrm{N/s} \leq \Delta u \leq 0.1\,\mathrm{N/s}, \tag{5.80e}$$

where $W = \operatorname{diag}\{w_1, w_2, w_3, w_4\}$ is a diagonal matrix of penalty parameters and the Lie derivatives $L_{\bar{f}_j}V_{\epsilon,j}$ and $L_{\bar{g}_j}V_{\epsilon,j}$ are given by (5.70) and (5.71), respectively.

5.4.2 Hierarchical task priority CLF-ECBF QP controller

By employing Algorithm 1 with $H(x) = A^{\mathrm{T}}A$, $c^{\mathrm{T}}(x) = 2b^{\mathrm{T}}A$ and the ECBFs and RES-CLFs from Section 5.2.4 and Section 5.3.4, respectively, the following QP is solved

$$
\begin{aligned}
&\underset{u\in\mathbb{R}^{15},\delta_1\in\mathbb{R},\delta_2\in\mathbb{R}}{\text{minimize}} \quad u^{\mathrm{T}}A^{\mathrm{T}}Au + 2b^{\mathrm{T}}Au + w_1\delta_1^2 + w_2\delta_2^2 && \text{(5.81a)}\\
&\quad\text{subject to}\\
&L_{\bar{f}_j}V_{\epsilon,j} + L_{\bar{g}_j}V_{\epsilon,j}u \leq -\frac{\gamma_j}{\epsilon_j}V_{\epsilon,j} + \delta_j, \quad j = 1,2, && \text{(5.81b)}\\
&L_f^2 h_k + \frac{\partial L_f h_k}{\partial x}gu \geq -K_{\alpha,k}\xi_k, \qquad k = 1,\dots,18, && \text{(5.81c)}\\
&-50\,\mathrm{N} \leq u \leq 50\,\mathrm{N}, && \text{(5.81d)}\\
&-0.1\,\mathrm{N/s} \leq \Delta u \leq 0.1\,\mathrm{N/s}, && \text{(5.81e)}
\end{aligned}
$$

which yields a control input $u = u_1^*$ that only accounts for the safety-related tasks and the end-effector configuration task. The end-effector configuration task only consumes 6 DOFs, which means that at most 8 DOFs may still be uncontrolled. Hence, the solution u_1^* can be refined by attempting to keep the base stationary and minimizing the joint velocities through the following QP

$$
\begin{aligned}
&\underset{u\in\mathbb{R}^{15},\delta_3\in\mathbb{R},\delta_4\in\mathbb{R}}{\text{minimize}} \quad u^{\mathrm{T}}A^{\mathrm{T}}Au + 2b^{\mathrm{T}}Au + w_3\delta_3^2 + w_4\delta_4^2 && \text{(5.82a)}\\
&\quad\text{subject to}\\
&L_{\bar{g}_i}V_{\epsilon,i}u \leq L_{\bar{g}_i}V_{\epsilon,i}u_1^*, \qquad i = 1,2, && \text{(5.82b)}\\
&L_{\bar{f}_j}V_{\epsilon,j} + L_{\bar{g}_j}V_{\epsilon,j}u \leq -\frac{\gamma_j}{\epsilon_j}V_{\epsilon,j} + \delta_j, \quad j = 3,4, && \text{(5.82c)}\\
&\frac{\partial L_f h_k}{\partial x}gu \geq \frac{\partial L_f h_k}{\partial x}gu_1^*, \qquad k = 1,\dots,18, && \text{(5.82d)}\\
&-50\,\mathrm{N} \leq u \leq 50\,\mathrm{N}, && \text{(5.82e)}\\
&-0.1\,\mathrm{N/s} \leq \Delta u \leq 0.1\,\mathrm{N/s}, && \text{(5.82f)}
\end{aligned}
$$

which yields the final control input that is applied to the AIAUV.

5.5 Simulation Results

The QPs are solved in MATLAB using `quadprog` and the closed loop system is simulated in Simulink using `ode3`.

5.5.1 Control parameters and initial conditions

The control parameters for the equality-based tasks consists of the RES-CLF convergence rates ϵ_i and the penalty parameter values w_i, these are all shown in Table 5.2. As seen from (2.44), the rate of exponential convergence is inversely proportional to ϵ_i, which entails that lower values of ϵ_i results in faster convergence.

Table 5.2: Equality task convergence rates and penalty parameters.

	σ_1	σ_2	σ_3	σ_4
ϵ	1.2	0.2	1.2	0.5
w	60	60	10	10

The set-based gains $K_{\alpha,k}$ are chosen by placing the eigenvalues of the closed loop systems

$$\dot{\xi}_k = \left(F - GK_{\alpha,k}\right)\xi_k, \tag{5.83}$$

where

$$F = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{5.84}$$

and $K_{\alpha,k} = \begin{bmatrix} \alpha_{k,1} & \alpha_{k,2} \end{bmatrix}$ at $\lambda_1 = -1$ and $\lambda_2 = -3$ for $k = 1, \dots, 18$. Solving

$$\det\left(\lambda I_2 - \left(F - GK_{\alpha,k}\right)\right) = (\lambda + 1)(\lambda + 3), \tag{5.85}$$

for $\alpha_{k,1}$ and $\alpha_{k,2}$ yields $K_{\alpha,k} = \begin{bmatrix} 3 & 4 \end{bmatrix}$.

The relative damping ratios and natural frequencies of the reference model are chosen as

$$\Lambda = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_7\} = I, \tag{5.86}$$

$$\Omega = \operatorname{diag}\{\omega_1, \omega_2, \ldots, \omega_7\} = 0.12I, \tag{5.87}$$

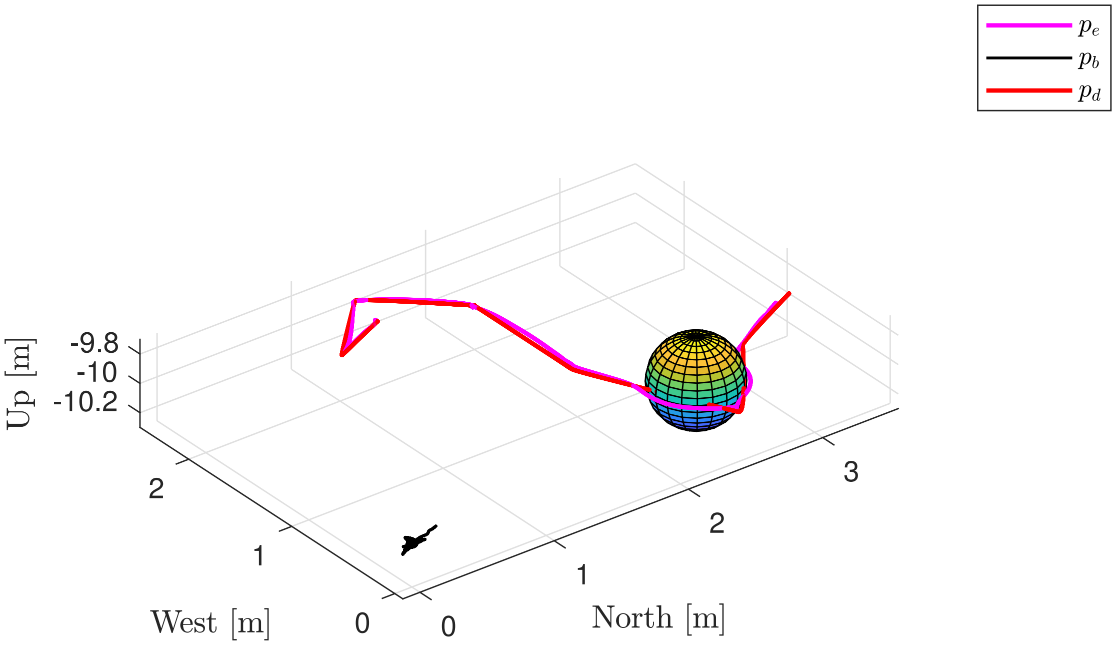
where I is the 7×7 identity matrix.

The system is initialized from the following initial configuration

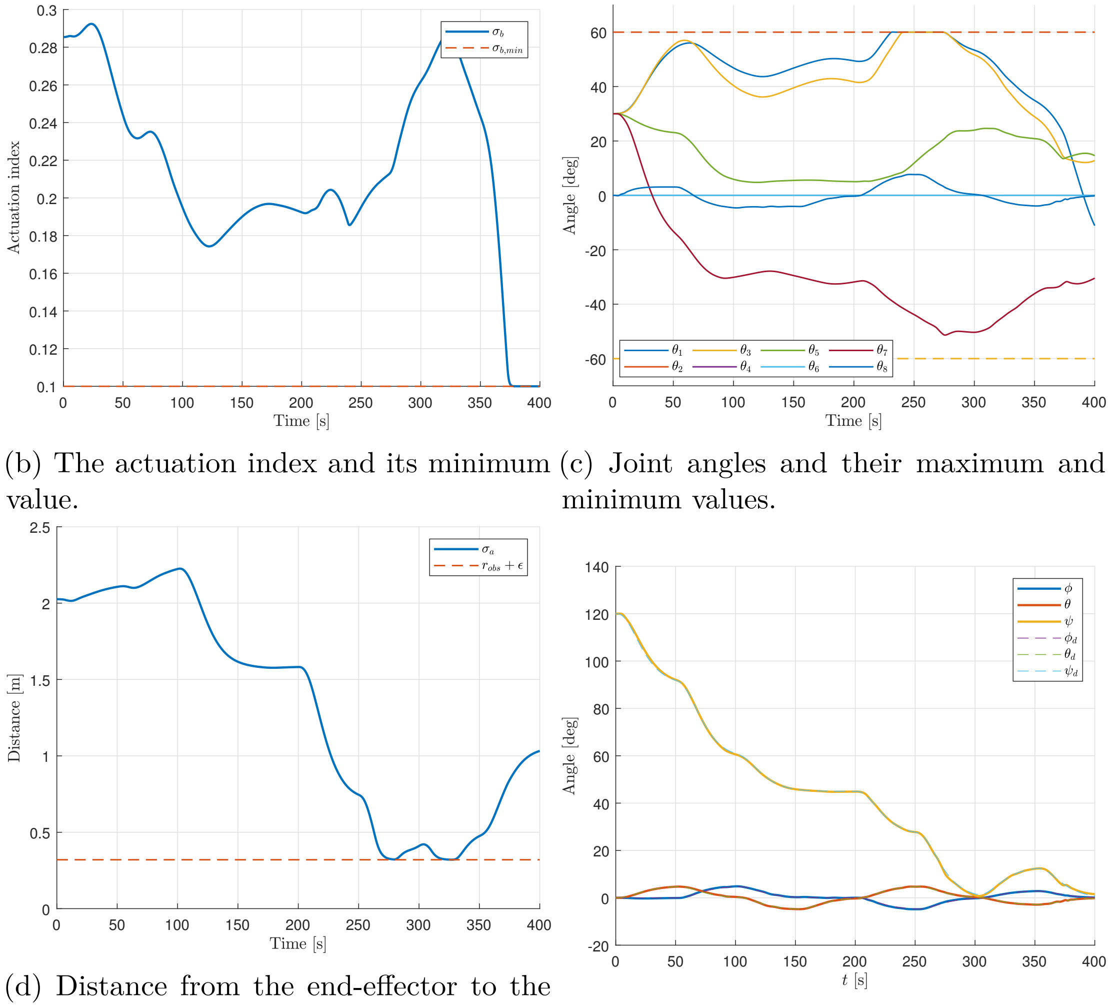
$$\xi_0 = \left[1.43\,\mathrm{m}, 2.2\,\mathrm{m}, -10\,\mathrm{m}, 1, 0, 0, 0, \frac{\pi}{6}, 0, \frac{\pi}{6}, 0, \frac{\pi}{6}, 0, \frac{\pi}{6}, 0\right]^{\mathrm{T}}. \tag{5.88}$$

5.5.2 N task CLF-ECBF QP

Simulation results for the N task CLF-ECBF QP controller in (5.80) are presented in Figure 5.1.



(a) North-West-Up plot. p_e, p_b and p_d represents the end-effector, base and the desired end-effector position, respectively.

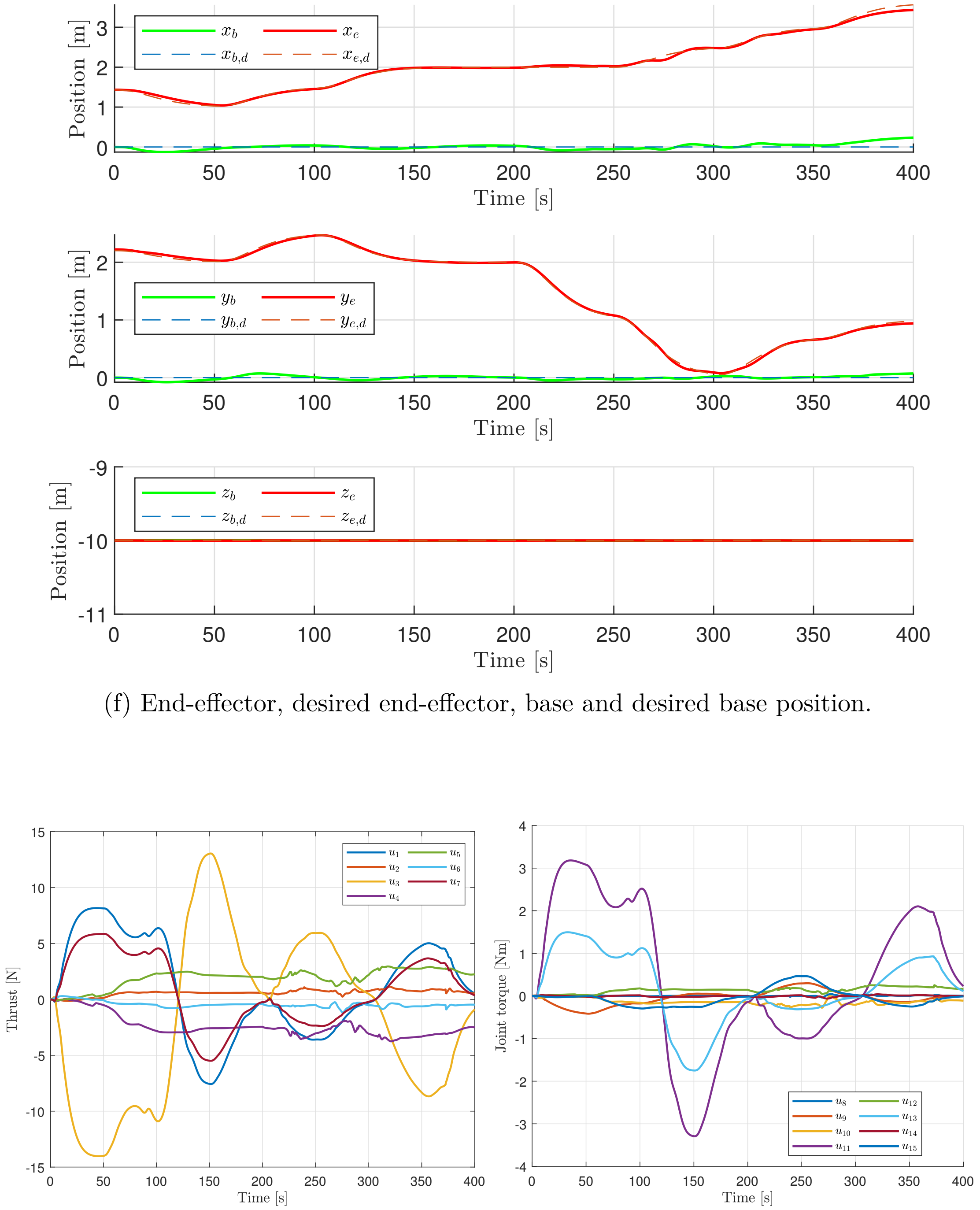


(b) The actuation index and its minimum value.

(c) Joint angles and their maximum and minimum values.

(d) Distance from the end-effector to the spherical obstacle.

(e) End-effector attitude.



(f) End-effector, desired end-effector, base and desired base position.

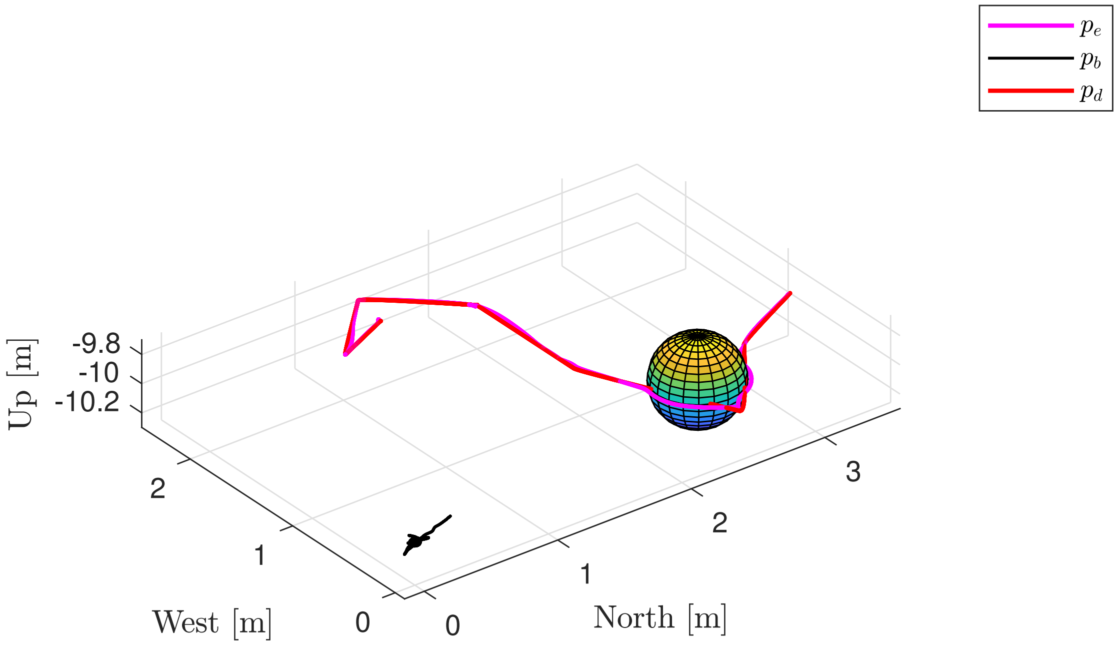
(g) Thruster input.

(h) Joint torque inputs.

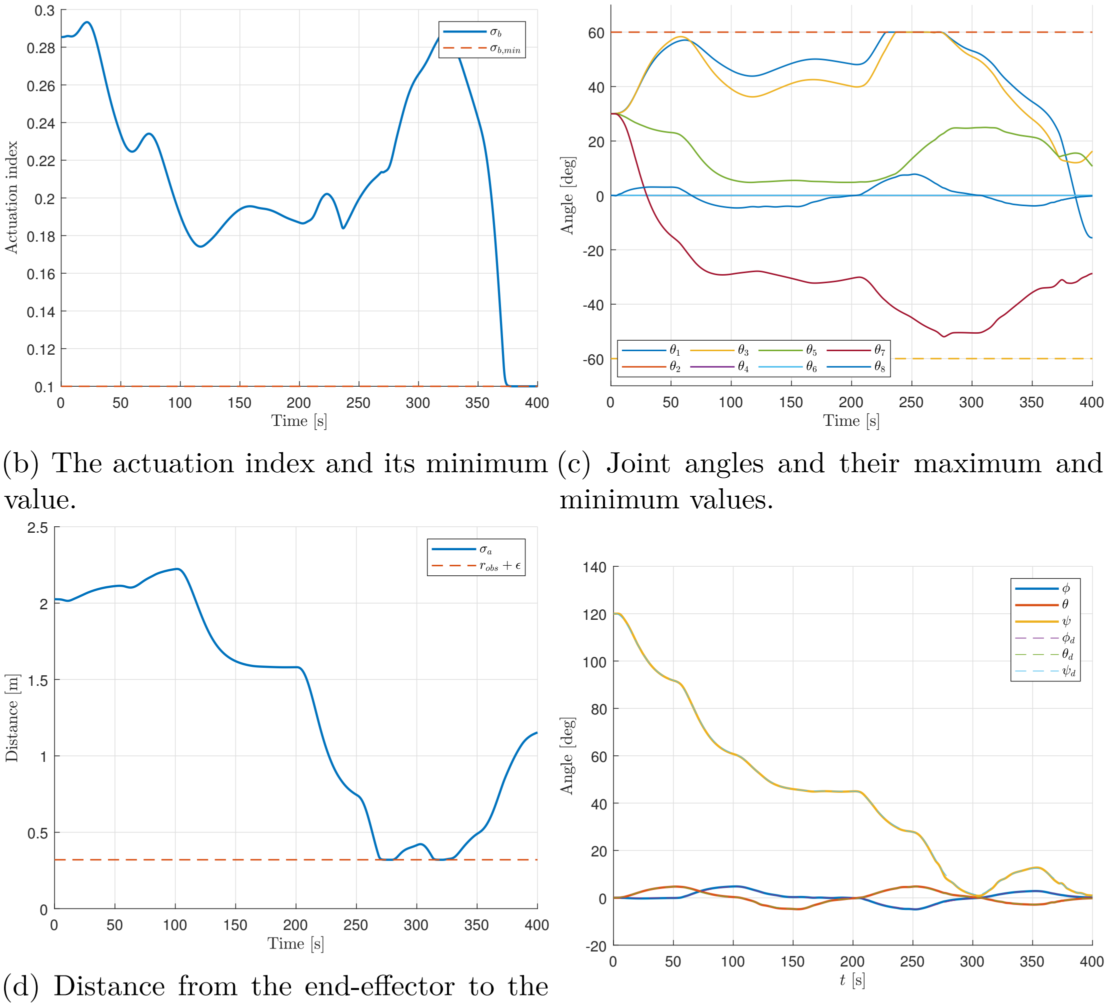
Figure 5.1: Simulation results for the N-task CLF and CBF based QP controller.

5.5.3 Task priority CLF-ECBF QP

Simulation results for the hierarchical task priority CLF-ECBF QP controller given by (5.81) and (5.82) are presented in Figure 5.1.



(a) North-West-Up plot. p_e, p_b and p_d represents the end-effector, base and the desired end-effector position, respectively.

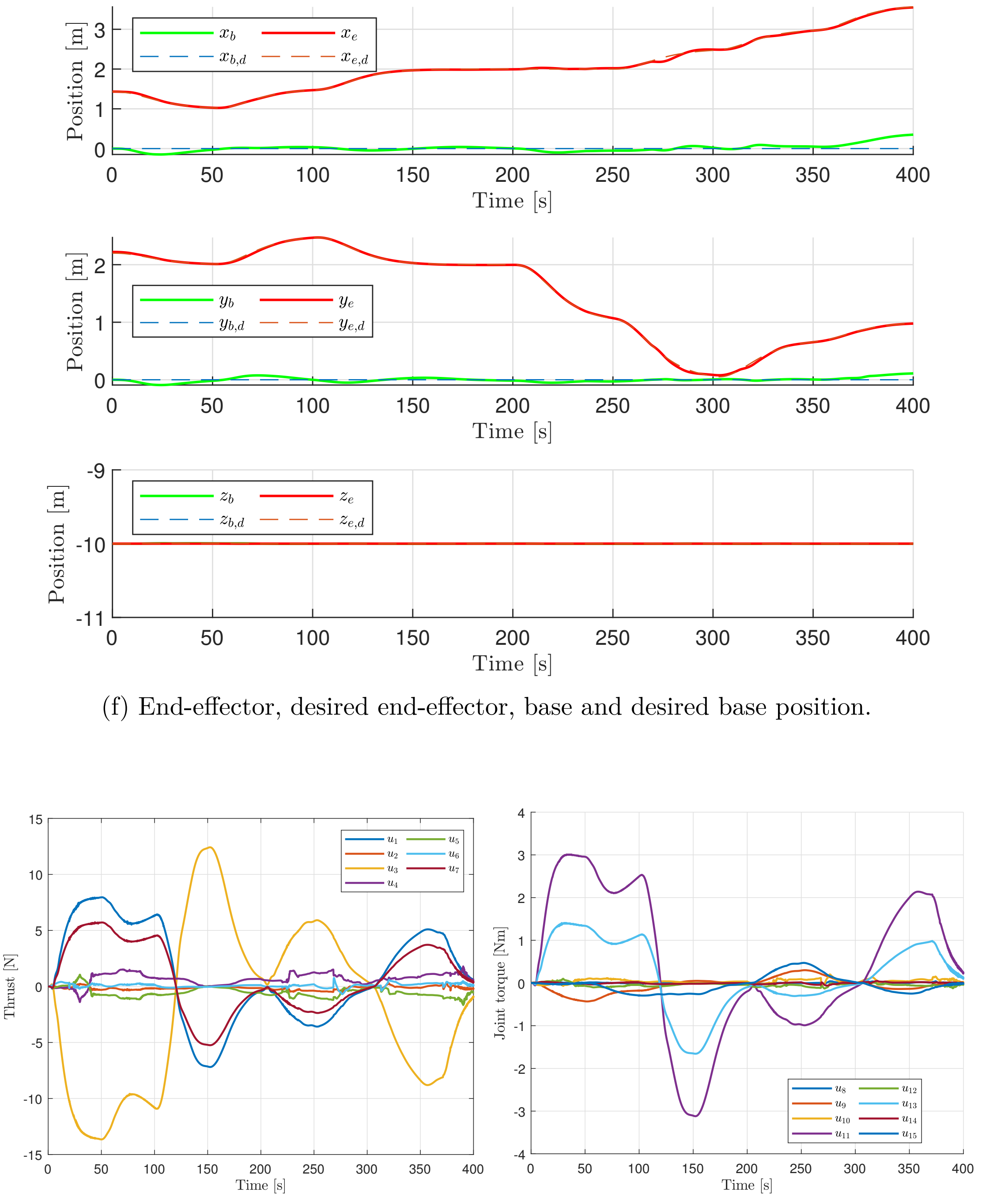


(b) The actuation index and its minimum value.

(c) Joint angles and their maximum and minimum values.

(d) Distance from the end-effector to the spherical obstacle.

(e) End-effector attitude.



(f) End-effector, desired end-effector, base and desired base position.

(g) Thruster input.

(h) Joint torque inputs.

Figure 5.2: Simulation results for a task priority framework based on a hierarchy of CLF and CBF based QPs.

5.6 Discussion

Observe that the optimization problems in (5.80), (5.81) and (5.82) are all formulated in terms of the thruster and joint torque control inputs u, and not the generalized forces and torques $\tau = Bu$. Consequently, the hierarchical CLF-ECBF QP controller solves the control allocation problem in addition to resolving redundancy and acting as a dynamic controller. By unifying the redundancy resolution and control allocation problems, strict priority among tasks can always be ensured under the assumption of a perfect dynamic model. The same cannot be said for decoupled AIAUV control approaches that resort to e.g. kinematic or operational space control. Kinematic or operational space control frameworks provide no a priori guarantees that the commanded force and torque vector computed by the dynamic controller can be exactly allocated, since the commanded forces and torques are typically computed with no regard to physical actuator limits, rate constraints, or actuator configuration matrix singularities. If the commanded forces and torques cannot be exactly allocated, the forces and torques are usually allocated to thruster inputs and joint torques by minimizing the allocation error, which is done independently of the redundancy resolution algorithm. Hence, tasks are no longer prioritized and the control system performance may dramatically decrease even though the allocation error is somewhat small.

Intervention and inspection tasks are performed at low-speed, where thruster dynamics are essential to the control problem [63]. Instead of modeling the entire thruster dynamics, the rate of change of the thruster input can be limited as seen in (5.80e), (5.81e) and (5.82f). As a result, more realistic behavior is obtained without affecting the priority between tasks. This is an important advantage of the proposed scheme with regard to a future experimental implementation.

By comparing Figure 5.1f and Figure 5.2f, it is evident that the non-hierarchical approach in (5.80) does not achieve strict priority between the end-effector and base positioning tasks for $t \geq 350\,\mathrm{s}$. At $t = 350\,\mathrm{s}$, the end-effector position is commanded outside of the manipulator workspace (when the base is kept at $p_{nb}^{n} = [0, 0, -10\,\mathrm{m}]^{\mathrm{T}}$). Instead of executing the higher priority end-effector task perfectly, a compromise between two incompatible tasks, weighted by the penalty parameters in (5.80a) is found. Even though this compromise is optimal with respect to the objective function, it is not what is desired. The end-effector task is classified as mission related, and should be satisfied whenever it does not conflict with safety related tasks. However, the base positioning task (or any other optimization task) is only included because of the redundancy of the system, and should never affect how the end-effector (or any mission related task) is executed. This is achieved by resorting to the hierarchical task priority CLF-ECBF QP scheme as seen in Figure 5.2f, where perfect end-effector position tracking is achieved at the expense of a greater error in the base position.

Observe from Figures 5.1c and 5.2c that four joints have non-zero angles at $t = 400\,\mathrm{s}$. This is somewhat unexpected since a configuration for which all joint angles are equal zero will yield the largest distance between the end-effector and the base. However, the actuator configuration matrix of the particular AIAUV used for simulation exhibits a singularity at the configuration for which all joint angles are zero, i.e. $\sigma_b = \det\left(B(0)B(0)^{\mathrm{T}}\right) = 0$. By inspecting Figures 5.1b and 5.2b it is clear that the actuation index σ_b is at its lower limit from $t > 375\,\mathrm{s}$. Consequently, multiple joint angles are kept non-zero by the actuation index task, resulting in a larger deviation in the base position than strictly necessary.

In general, the control performance is excellent for both approaches, all set-based tasks are satisfied at all times, while the end-effector configuration tracking is fast and accurate. There is a noticeable deviation in the x-coordinate of the end-effector position from $t \simeq 268\,\mathrm{s}$ to $t \simeq 281\,\mathrm{s}$, since the desired x-position of the end-effector is inside of the spherical obstacle.

The simulation results are very similar whenever tasks are compatible. Consequently, the non-hierarchical approach could be employed whenever tasks at different priority levels are compatible, and the hierarchical approach whenever tasks are incompatible, thereby ensuring strict priority between the end-effector and base task at all times, while reducing the computational burden. One possible check for compatible tasks is given by the condition (3.28).

Chapter 6

Conclusion

This thesis has presented task priority operational space control in a MIMO feedback linearization setting, where explicit conditions for input-output and full-state feedback linearizability of a task priority operational space control law has been given. This work led to the merging of operational space control with control Lyapunov function and control barrier function based quadratic programs, with the goal of obtaining a dynamic set-based task priority framework. The unification resulted in a framework that supports set-based tasks; however, strict priority among tasks was lost in the process. As a result, null space based control was not pursued any further. Instead, a novel task priority framework based on a hierarchy of control Lyapunov- and control barrier function based quadratic programs was developed, generalizing the work in [23, 32] to any number of priority levels. There is considerable design freedom within this framework, soft priority measures can be employed between tasks at the same priority level, while a strict priority measure in the form of solving another quadratic program can be used to ensure that lower priority tasks have no effect on higher priority tasks. This allows optimization based control objectives to be safely included, without affecting the execution of mission related tasks.

The proposed framework has been verified in simulation for an AIAUV, which is an overactuated and redundant robotic system. For these types of systems, the proposed task priority framework also solves the control allocation problem, which is highly advantageous since control input bounds and rate constraints can be accounted for when resolving redundancy, effectively avoiding a situation in which commanded generalized forces and torques cannot be allocated explicitly, leading to a loss of priority among tasks.

6.1 Future Work

The effect of modeling uncertainties on the CLF-ECBF QP controllers should be investigated. This is especially relevant for an underwater vehicle application such as an AIAUV, since accurate identification of the dynamic model parameters is difficult [60]. In order to mitigate the negative effects of modeling uncertainties, robust and adaptive control methods that are applicable to the task priority CLF-ECBF QP framework presented in this thesis can be investigated. Robust control of CLF based QPs has already been investigated to some extent in [64]. A robust experimental implementation of the proposed control system will help investigate the effect of modeling uncertainties.

Another topic for future work is to include force-based multi-contact tasks directly through the QP based formulations as done in [23]. In this way, the proposed framework could be employed for intervention tasks. Different objective functions could also be utilized in order to weight the usage of thruster inputs versus joint torque inputs.

Appendix A

Definitions and Theorems

Definition A.1. The Lie derivative of a scalar function $h : \mathbb{R}^n \to \mathbb{R}$ along a vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ is given by [42]

$$L_f h = \sum_{i=1}^{n} \frac{\partial h}{\partial x_i} f_i(x) = \frac{\partial h}{\partial x} f(x). \tag{A.1}$$

Definition A.2. An extended class K_∞ function is a function $\alpha : \mathbb{R} \to \mathbb{R}$ that is strictly increasing and with $\alpha(0) = 0$ [65].

Definition A.3. For $a, b \in \mathbb{R}^3$, the cross product by a is a linear operator. Hence, $b \to a \times b$ can be represented by a skew-symmetric matrix [58]

$$a \times b = a_\times b, \tag{A.2}$$

where

$$a_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \in \mathfrak{so}(3), \tag{A.3}$$

and

$$\mathfrak{so}(3) = \left\{ S \in \mathbb{R}^{3\times3} : S^{\mathrm{T}} = -S \right\}. \tag{A.4}$$

Definition A.4. If $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$, then the Kronecker product $A \otimes B \in \mathbb{R}^{mp \times nq}$ is given by [66]

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B, \end{bmatrix}. \tag{A.5}$$

Definition A.5. The partial derivative of a matrix $A(x)$ with respect to x, where $A : \mathbb{R}^n \to \mathbb{R}^{m \times p}$ is given by [55]

$$\frac{\partial A}{\partial x} = \begin{bmatrix} \frac{\partial A}{\partial x_1} & \frac{\partial A}{\partial x_2} & \dots & \frac{\partial A}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times pn}. \tag{A.6}$$

Theorem A.1. *The time derivative of a matrix $A(x) \in \mathbb{R}^{m \times p}$, with $x(t) \in \mathbb{R}^n$ is given by [55]*

$$\frac{\mathrm{d}}{\mathrm{d}t} A(x) = \sum_{i=1}^{n} \frac{\partial A}{\partial x_i} \dot{x}_i = \frac{\partial A}{\partial x} \left(\dot{x} \otimes I_p \right), \tag{A.7}$$

where $\otimes$ denotes the Kronecker product.

Theorem A.2. *The partial derivative of the product of two matrices $A(x) \in \mathbb{R}^{m \times p}$ and $B(x) \in \mathbb{R}^{p \times s}$ with respect to a vector $x \in \mathbb{R}^n$ is defined by the following rule [50]*

$$\frac{\partial}{\partial x} \left(A(x) B(x) \right) = \frac{\partial A(x)}{\partial x} \left(B(x) \otimes I_n \right) + A \frac{\partial B}{\partial x}, \tag{A.8}$$

where I_n is the $n \times n$ identity matrix.

Definition A.6. Given a dynamical system

$$\dot{x} = f(x), \tag{A.9}$$

and a trajectory $x(t, x_0)$ where x_0 is the initial point. A set M is said to be forward invariant if [65]

$$x_0 \in M \implies x(t) \in M, \tag{A.10}$$

for all $t \geq 0$.

Appendix B

Conference paper

This appendix contains a draft of a conference paper based on the main result on dynamic task priority control of redundant robotic systems. The article will be submitted to the 2020 American Control Conference.

Task Priority Control of Redundant Robotic Systems using Control Lyapunov and Control Barrier Function based Quadratic Programs

Erlend A. Basso[1] and Kristin Y. Pettersen[1]

***Abstract*—Redundant robotic systems are designed to accomplish multiple tasks simultaneously. Tasks are functions of the system configuration, and can be divided into groups by their priority. System redundancy can be exploited by including lower priority optimization tasks within the control framework. However, it is important that the inclusion of such lower priority tasks does not have an effect on higher priority safety related and operational tasks. This paper presents a novel task priority framework based on a hierarchy of control Lyapunov function (CLF) and control barrier function (CBF) based quadratic programs (QPs). 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. As opposed to kinematic control schemes, the proposed framework is a holistic approach to robot control that solves the redundancy resolution, dynamic control and control allocation problems simultaneously. Simulation results of a hyper-redundant articulated intervention underwater autonomous vehicle (AIAUV) is presented to validate the proposed framework.**

I. INTRODUCTION

A robotic system is kinematically redundant when it has more degrees of freedom (DOFs) than those strictly required to execute a given task. This enables additional tasks to be executed simultaneously by utilizing the redundant DOFs of the system. It is useful to divide control tasks into three groups, safety related tasks, operational tasks and optimization tasks, arranged by decreasing priority [1]. Redundancy should be resolved such that lower priority tasks do not affect the execution of higher priority tasks.

Kinematic task priority control is a redundancy resolution method introduced in [2], developed in [3] and generalized to any number of priority levels in [4]. This control approach decouples the controller into a kinematic and dynamic controller, and has been successfully implemented on a number of robotic systems. The framework was extended to support tasks described by sets or inequalities in [5], [6] and [7]. These kinematic control approaches all resolve redundancy at the velocity level by generating velocity references for some dynamic controller to follow. An immediate drawback is that acceleration references cannot be included, resulting in worse tracking accuracy.

Operational space control [8] is a holistic approach that assigns joint torques directly by transforming the equations of motion from joint space into the operational space (also known as task space). Although it was mainly introduced for non-redundant systems, a dynamically consistent null space operator was defined in [8], that allowed two operational space tasks to be defined and controlled simultaneously. In [9], the scheme was extended to a task priority framework with an arbitrary number of tasks by generalizing the dynamically consistent null space operator from [8] to an arbitrary number of priority levels. These null space operators ensure that torques generated by lower priority tasks do not generate accelerations that affect the task dynamics of higher priority tasks. The operational space framework was extended to include set-based tasks in [10], but this approach does not scale well for systems with a high number of DOFs.

Control Lyapunov functions (CLFs) extend Lyapunov theory to systems with inputs and have become an essential part of nonlinear control design after the groundbreaking work in [11]–[13]. The CLF concept was extended to rapidly exponentially stabilizing control Lyapunov functions (RES-CLFs) in [14], which achieve exponential convergence at a controllable rate. Through CLFs or RES-CLFs, the control designer is free to chose among an infinite number of controllers. An important example is the point-wise minimum norm controller [15], [16], which selects the control value of minimum norm from all control values rendering the time derivative of the CLF negative definite. The point-wise minimum norm controller has a closed form solution since it is the solution to a quadratic program (QP) with only one inequality constraint. This QP can be augmented with control input saturation limits and other control input constraints, at the expense of a closed form solution [17]. For redundant robotic systems, two control tasks can be satisfied simultaneously by defining CLFs for each task and finding a control input that minimizes some quadratic objective function while ensuring that the time derivatives of the CLFs are negative definite [18]. However, strict priority between tasks cannot be ensured.

Barrier functions have been used extensively in constrained optimization [19], [20], and they have motivated the concept of barrier certificates for safety-critical control. Barrier certificates were introduced as a tool for proving forward invariance of sets [21], [22]. Since these sets often encoded safety related objectives, proving invariance of a safe set implies that the system will remain safe, as long as you start safe. These barrier certificates tend to infinity as the state tends to the boundary of the safe set, and in order to obtain safety guarantees beyond the boundary of the safe set, various Lyapunov-like approaches have been proposed such as [23], where a positive definite barrier certificate is employed as a barrier Lyapunov function. However, these conditions are overly strong since the positive definiteness property enforces the invariance of every level set, and not just the safe set of the set-based task in question.

[1] Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, `erlenab@stud.ntnu.no`

Barrier certificates were extended to systems with inputs by introducing the first notion of a control barrier function (CBF) in [24]. These control barrier functions were combined with control Lyapunov functions in [25], and further improved in [26] to establish conditions for so-called control Lyapunov-barrier functions, which jointly guarantee safety and stability. These conditions were shown to be too restrictive and was hence relaxed in [27]. CBFs were generalized to exponential control barrier functions (ECBFs) in [28], which enforce forward invariance of set-based tasks with a higher relative degree. The barrier function conditions were refined and extended to the entire safe set in in [29], which enabled controller synthesis through optimization-based methods [30]. In particular, the CLF based QP [17], [18] could be augmented with CBFs to ensure stability and safety [27], [30].

The main contribution of this paper is a novel task priority framework in the form of a hierarchy of control Lyapunov function and control barrier function based quadratic programs. In contrast to the CLF based QP in [18], the proposed framework establishes strict priority levels by resorting to a hierarchy of QPs, where tasks at a lower priority level have no effect on the execution of tasks at higher priority levels. This framework is especially useful for redundant robotic systems that are also overactuated, since the control allocation and redundancy resolution problems are solved simultaneously.

II. Background Material

In this section, the necessary background material will be presented. For compactness, we will abuse notation and denote

$$L_g h(x) = \frac{\partial h}{\partial x} g(x), \tag{1}$$

whenever $h(x)$ is a scalar or vector-valued function, and $g(x)$ is a vector field or a matrix. Note that (1) is only equal to the Lie derivative of $h(x)$ along $g(x)$ when $h(x)$ is a multivariable scalar function and $g(x)$ a vector field.

A. Model

Consider the nonlinear affine control system

$$\dot{x} = f(x) + g(x)u, \tag{2}$$

where f and g are locally Lipschitz, $x \in D \subset \mathbb{R}^n$ and $u \in U \subset \mathbb{R}^p$ is the set admissible control inputs. Let the locally Lipschitz vector-valued function $y = \sigma(x) - \sigma_d(t)$ describe the error coordinates of the equality task $\sigma : \mathbb{R}^n \to \mathbb{R}^m$. Under the following assumption

$$L_g L_f^k y = 0, \quad 0 \leq k \leq \rho - 2 \tag{3}$$

$$L_g L_f^{\rho-1} y \neq 0, \tag{4}$$

the input-output dynamics becomes

$$y^{(\rho)} = \underbrace{L_f^{\rho} y(x)}_{b(x)} + \underbrace{L_g L_f^{\rho-1} y(x)}_{A(x)} u. \tag{5}$$

The system (2) can be decomposed into transverse dynamics states $\eta = \operatorname{col}\left(y, \dot{y}, \ldots, y^{(\rho-1)}\right) \in \mathbb{R}^{\rho m}$ and zero dynamics states $z \in Z \subset \mathbb{R}^{n-\rho m}$, viz.

$$\dot{\eta} = \bar{f}(\eta, z) + \bar{g}(\eta, z)u, \tag{6a}$$

$$\dot{z} = f_z(\eta, z), \tag{6b}$$

with $\bar{f}(\eta, z) = F\eta + Gb(x)$ and $\bar{g}(\eta, z) = GA(x)$ where

$$F = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ I \end{bmatrix}, \tag{7}$$

where 0 is the $m \times m$ matrix of zeros and I is the $m \times m$ identity matrix.

B. Control Lyapunov Functions

A control Lyapunov function is a candidate Lyapunov function $V(x)$ for which $\dot{V}(x)$ can be made negative by appropriate selection of the control input u. In order to explicitly control the rate of exponential convergence, a specific type of CLF is defined in [14] as follows:

Definition 1. *A continuously differentiable and positive definite function $V_\epsilon(\eta) : D \to \mathbb{R}$ is said to be a rapidly exponentially stabilizing control Lyapunov function (RES-CLF) for the system* (6) *if there exists constants $c_1, c_2, c_3 > 0$ such that for all $0 < \epsilon < 1$ and for all states (η, z) it holds that*

$$c_1 \|\eta\|^2 \leq V_\epsilon(\eta) \leq \frac{c_2}{\epsilon^2} \|\eta\|^2, \tag{8}$$

$$\inf_{u \in U} \left[L_{\bar{f}} V_\epsilon(\eta, z) + L_{\bar{g}} V_\epsilon(\eta, z) u \right] \leq -\frac{c_3}{\epsilon} V_\epsilon(\eta, z). \tag{9}$$

Such a function can be constructed by solving the continuous time algebraic Riccati equation

$$F^T P + PF - PGG^T P + Q = 0, \tag{10}$$

for $P = P^T > 0$, where Q is any positive definite matrix. In order to stabilize the transverse dynamics at a rate ϵ define

$$V_\epsilon(\eta) = \eta^T \begin{bmatrix} \frac{1}{\epsilon} I & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} \frac{1}{\epsilon} I & 0 \\ 0 & I \end{bmatrix} \eta := \eta^T P_\epsilon \eta. \tag{11}$$

When $A(x)$ has linearly independent rows, it can be shown that the time derivative of (11) satisfies [14]

$$\inf_{u \in U} \left[L_{\bar{f}} V_\epsilon(\eta, z) + L_{\bar{g}} V_\epsilon(\eta, z) u \right] \leq -\frac{\gamma}{\epsilon} V_\epsilon(\eta), \tag{12}$$

where $\gamma := \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} > 0$ and

$$L_{\bar{f}} V_\epsilon(\eta, z) = \eta^T \left(F^T P_\epsilon + P_\epsilon F \right) \eta + 2\eta^T PGb, \tag{13}$$

$$L_{\bar{g}} V_\epsilon(\eta, z) = 2\eta^T P_\epsilon GA. \tag{14}$$

C. Control Barrier Functions

Control objectives described by inequalities or sets can be enforced by rendering the superlevel set

$$\mathcal{C} = \{x \in D \subset \mathbb{R}^n : h(x) \geq 0\}, \tag{15}$$

of some smooth function $h : D \to \mathbb{R}$ forward invariant [30].

Definition 2. *Let $\mathcal{C} \subset D \subset \mathbb{R}^n$ be the superlevel set of a continuously differentiable function $h : D \to \mathbb{R}$, then h is a control barrier function (CBF) for the system* (2) *if there exists an extended class K_∞ function α such that*

$$\sup_{u \in U} \left[L_f h(x) + L_g h(x) u\right] \geq -\alpha\left(h(x)\right), \tag{16}$$

for all $x \in \operatorname{Int}(\mathcal{C})$.

The existence of a CBF implies that the superlevel set of the function h is forward invariant [29], which means that if $x(t_0) = x_0 \in \mathcal{C}$, then $x = x(t) \in \mathcal{C}$ for all $t \geq t_0$. Equivalently, if $h(x_0) \geq 0$, then $h(x) \geq 0$ for all $t \geq t_0$.

D. Exponential Control Barrier Functions

Definition 2 assumes that the relative degree of h is equal to one. However, safety related tasks for robotic systems are often a function of the configuration variables only, meaning that they have a higher relative degree. Introduced in [28] and refined in [30], exponential control barrier functions generalizes CBFs to functions $h(x)$ with arbitrary relative degree $r \geq 1$.

Definition 3. *Given a set $\mathcal{C} \subset D \subset \mathbb{R}^n$ defined as the superlevel set of an r-times continuously differentiable function $h : D \to \mathbb{R}$, then h is an exponential control barrier function (ECBF) for the control system* (2) *if there exists a row vector $K_\alpha \in \mathbb{R}^r$ such that*

$$\sup_{u \in U} \left[L_f^r h(x) + L_g L_f^{r-1} h(x) u\right] \geq -K_\alpha \xi(x), \tag{17}$$

where $\xi = \operatorname{col}\left(h(x), L_f h(x), L_f^2 h(x), \ldots, L_f^{r-1} h(x)\right)$, results in $h(x) \geq 0$ whenever $h(x_0) \geq 0$ for all $x \in \operatorname{Int}(\mathcal{C})$

E. Combining CLFs and ECBFs

The RES-CLF and ECBF conditions in (9) and (17) are both affine in the control input u, which means that the control problem can be formulated as a convex optimization problem, enabling the incorporation of control input saturation limits and rate constraints [17]. The CLF-ECBF based QP is [18], [30]:

$$\begin{aligned} &\underset{u \in \mathbb{R}^m, \delta \in \mathbb{R}}{\text{minimize}} \quad \frac{1}{2} u^T H(x) u + c^T(x) u + w\delta^2 \\ &\text{subject to} \\ &L_{\bar{f}} V_\epsilon(\eta, z) + L_{\bar{g}} V_\epsilon(\eta, z) u \leq -\frac{\gamma}{\epsilon} V_\epsilon + \delta, \\ &L_f^r h(x) + L_g L_f^{r-1} h(x) u \geq -K_\alpha \xi(x), \end{aligned} \tag{18}$$

where $H(x) \in \mathbb{R}^{m \times m}$ is any positive semi-definite matrix, $c(x) \in \mathbb{R}^m$ and $\delta \in \mathbb{R}$ is a slack variable penalized by $w > 0$, ensuring the feasibility of the QP in case of conflicting set-based and equality-based control objectives.

III. Quadratic Programs for N Equality- and M Set-Based Control Tasks

This section extends the CLF-ECBF QP controller in [30] to an arbitrary number of equality- and set-based control tasks distributed to an arbitrary number of priority levels.

A. CLF Penalty Parameters as a Priority Measure

Inspired by [18], the QP in (18) can be extended to N equality-based control objectives by deriving the input-output dynamics for each control objective as in Section II-A and defining

$$y = \begin{bmatrix} y_1^{(\rho_1)} \\ y_2^{(\rho_2)} \\ \vdots \\ y_N^{(\rho_N)} \end{bmatrix} = \underbrace{\begin{bmatrix} L_f^{\rho_1} y_1(x) \\ L_f^{\rho_2} y_2(x) \\ \vdots \\ L_f^{\rho_N} y_N(x) \end{bmatrix}}_{b(x)} + \underbrace{\begin{bmatrix} L_g L_f^{\rho_1 - 1} y_1(x) \\ L_g L_f^{\rho_2 - 1} y_2(x) \\ \vdots \\ L_g L_f^{\rho_N - 1} y_N(x) \end{bmatrix}}_{A(x)} u. \tag{19}$$

Transverse dynamics states $\eta_i = \operatorname{col}\left(y_i, \dot{y}_i, \ldots, \dot{y}_i^{(\rho_i - 1)}\right)$ and RES-CLFs $V_{\epsilon,i}$ can then be defined analogously to (6a), (7) and (11). Moreover, M set-based tasks described by the superlevel sets $\mathcal{C}_j$ of some r_j times continuously differentiable function $h_j(x)$ can be included at the highest priority level (which is implied by no slack variables). The control input can then be obtained from the QP:

$$\begin{aligned} &\underset{u \in \mathbb{R}^m, \delta \in \mathbb{R}^N}{\text{minimize}} \quad u^T H(x) u + c^T(x) u + \delta^T W \delta \\ &\text{subject to} \\ &L_{\bar{f}_i} V_{\epsilon,i} + L_{\bar{g}_i} V_{\epsilon,i} u \leq -\frac{\gamma_i}{\epsilon} V_{\epsilon,i} + \delta_i, \quad i = 1, \ldots, N, \\ &L_f^{r_k} h_k + L_g L_f^{r_k - 1} h_k u \geq -K_{\alpha,k} \xi_k, \quad k = 1, \ldots, M, \end{aligned} \tag{20}$$

where $W \in \mathbb{R}^{N \times N}$ is a diagonal matrix of penalty parameters and

$$L_{\bar{f}_i} V_{\epsilon,i} = \eta_i^T \left(F_i^T P_{\epsilon,i} + P_{\epsilon,i} F_i\right) \eta_i + 2\eta_i^T P_i G_i b_i, \tag{21}$$

$$L_{\bar{g}_i} V_{\epsilon,i} = 2\eta_i^T P_{\epsilon,i} G_i A_i, \tag{22}$$

for $i = 1, \ldots, N$, where A_i and b_i are given by

$$y_i^{(\rho_i)}(x) = \underbrace{L_f^{\rho_i} y_i(x)}_{b_i(x)} + \underbrace{L_g L_f^{\rho_i - 1} y_i(x)}_{A_i(x)} u. \tag{23}$$

The equality tasks encoded by RES-CLFs are prioritized by adjusting the elements of the diagonal penalty matrix W. The satisfaction of all equality tasks are therefore described by a single objective function through the value of the slack variables δ and the penalty parameters in W. Whenever equality tasks are incompatible, this fact invariably leads to trade-off configurations that do not satisfy any of the tasks. Hence, strict priority between tasks cannot be achieved in the sense that lower priority tasks have no effect on the execution of higher priority tasks. As a result, it is challenging to include lower priority optimization based tasks since they will interfere with more critical higher priority tasks such as end-effector control whenever the tasks are incompatible.

B. Main Result: Enforcing Strict Priority Between a Selection of Tasks

In order to establish more than two strict priority levels, we propose to solve a quadratic program for every priority level as suggested for kinematic control in [7]. The idea is to begin by computing a control input according to (20) that only accounts for safety related set-based tasks and equality tasks at the highest priority level. Subsequently, a new quadratic program is solved for each priority level, refining the previous solution in an attempt to satisfy lower priority tasks without affecting the execution of higher priority tasks.

Consider N equality tasks and M set-based tasks distributed to k priority levels, with $N = N_1 + \ldots + N_k$ and $M = M_1 + \ldots + M_k$, where N_i and M_i denotes the number of equality and set-based tasks at priority level i, respectively. A control input u_1^* that disregards all lower priority tasks is obtained by solving (20) with $i = 1, \ldots, N_1$ and $k = 1, \ldots, M_1$. If the system is redundant with respect to these $N_1 + M_1$ tasks, the control input u_1^* can be refined without affecting how the N_1 higher priority equality tasks are executed by enforcing

$$L_{\bar{f}_i} V_{\epsilon,i} + L_{\bar{g}_i} V_{\epsilon,i} u \leq L_{\bar{f}_i} V_{\epsilon,i} + L_{\bar{g}_i} V_{\epsilon,i} u_1^* \tag{24}$$

which implies that $L_{\bar{g}_i} V_{\epsilon,i} u \leq L_{\bar{g}_i} V_{\epsilon,i} u_1^*$ for all $i = 1, \ldots, N_1$. Similarly, the higher priority set-based tasks are unaffected by enforcing $L_g L_f^{r_k - 1} h_k u \geq L_g L_f^{r_k - 1} h_k u_1^*$ for all $k = 1, \ldots, M_1$. Consider N_2 additional equality-based tasks and M_2 additional set-based tasks. The control input u_2^* is obtained by solving:

$$\begin{aligned}
&\underset{(u,\delta,s)\in\mathbb{R}^{m+N_2+M_2}}{\text{minimize}} \quad u^T H(x) u + c^T(x) u + \delta^T W \delta + s^T K s \\
&\quad \text{subject to} \\
&L_{\bar{g}_i} V_{\epsilon,i} u \leq L_{\bar{g}_i} V_{\epsilon,i} u_1^*, && {\scriptstyle i=1,\ldots,N_1,} \\
&L_{\bar{f}_j} V_{\epsilon,j} + L_{\bar{g}_j} V_{\epsilon,j} u \leq -\frac{\gamma_j}{\epsilon} V_{\epsilon,j} + \delta_j, && {\scriptstyle j=N_1+1,\ldots,N_1+N_2,} \\
&L_g L_f^{r_k-1} h_k u \geq L_g L_f^{r_k-1} h_k u_1^*, && {\scriptstyle k=1,\ldots,M_1,} \\
&L_f^{r_l} h_l + L_g L_f^{r_l-1} h_l u \geq -K_{\alpha,l} \xi_l - s_l, && {\scriptstyle l=M_1+1,\ldots,M_1+M_2,}
\end{aligned} \tag{25}$$

where slack variables s penalized by the elements in the diagonal matrix $K > 0$ have been added to the lower priority set-based tasks enforced through ECBFs to ensure feasibility of the optimization problem.

By observing that the solution u_2^* to (25) enforces the constraints $L_{\bar{g}_i} V_{\epsilon,i} u_2^* \leq L_{\bar{g}_i} V_{\epsilon,i} u_1^*$ and $L_g L_f^{r_k-1} h_k u_2^* \geq L_g L_f^{r_k-1} h_k u_1^*$ for all i and k, it is straightforward to generalize (25) to an arbitrary priority level n, viz.

$$\begin{aligned}
&\underset{u\in\mathbb{R}^m,\delta\in\mathbb{R}^{N_n},s\in\mathbb{R}^{M_n}}{\text{minimize}} \quad u^T H u + c^T u + \delta^T W_n \delta + s^T K_n s \\
&\quad \text{subject to} \\
&L_{\bar{g}_i} V_{\epsilon,i} u \leq L_{\bar{g}_i} V_{\epsilon,i} u_{n-1}^*, && {\scriptstyle i=1,\ldots,\bar{N}_{n-1},} \\
&L_{\bar{f}_j} V_{\epsilon,j} + L_{\bar{g}_j} V_{\epsilon,j} u \leq -\frac{\gamma_j}{\epsilon} V_{\epsilon,j} + \delta_j, && {\scriptstyle j=\bar{N}_{n-1}+1,\ldots,\bar{N}_n,} \\
&L_g L_f^{r_k-1} h_k u \geq L_g L_f^{r_k-1} h_k u_{n-1}^*, && {\scriptstyle k=1,\ldots,\bar{M}_{n-1},} \\
&L_f^{r_l} h_l + L_g L_f^{r_l-1} h_l u \geq -K_{\alpha,l} \xi_l - s_l, && {\scriptstyle l=\bar{M}_{n-1}+1,\ldots,\bar{M}_n,}
\end{aligned} \tag{26}$$

where $\bar{N}_n = N_1 + N_2 + \ldots + N_n$ and $\bar{M}_n = M_1 + M_2 + \ldots + M_n$. Note that the objective function is slightly different at every priority level, since the slack variables δ and s always correspond to tasks at the current priority level. This prevents trade-off configurations where none of the tasks are satisfied from occurring. The procedure is summarized in Algorithm 1.

Algorithm 1 Task priority CLF-ECBF QP controller

```
Input: $H(x)$, $c(x)$, $V_{\epsilon,i}(\eta_i)$, $i = 1, \ldots, N$, $h_j(x)$, $j = 1, \ldots, M$.
Output: $u$
Solve (20) to obtain $u_1^*$ with $i = 1, \ldots, N_1$, $k = 1, \ldots, M_1$.
for $p = 2$ to $k$ do
    Solve (26) to obtain $u_p^*$.
end for
return $u = u_k^*$.
```

IV. Simulations

In this section, the proposed hierarchical control scheme is validated in simulation on an articulated intervention autonomous underwater vehicle (AIAUV) based on the Eelume robot [31], [32], see Figure 1. The AIAUV is a floating base manipulator, with $n+1$ links interconnected by n joints, where link 1 is the tail, or base link and link $n + 1$ is the head. The simulation model has $n = 8$ joints and $p = 7$ thrusters, where all joints are single DOF and revolute. The system configuration can be described by $\xi = \operatorname{col}\left(p_{ib}^i, q, \theta\right) \in \mathbb{R}^{7+n}$, where $p_{ib}^i \in \mathbb{R}^3$ is the position of the base of the AIAUV in an inertial frame, $q \in \mathbb{R}^4$ is a unit quaternion describing the orientation of the base and $\theta \in \mathbb{R}^8$ are the joint angles. The linear and angular velocities of the base frame with respect to an inertial frame are denoted v_{ib}^i and ω_{ib}^i, respectively. These quantities are collected in a velocity vector $\zeta = \operatorname{col}\left(v_{i,b}^i, \omega_{i,b}^i\right)$. The equations of motion are given by [31]

$$\dot{\xi} = J_\xi(q)\zeta, \tag{27}$$

$$M(\theta)\dot{\zeta} + C(\theta,\zeta)\zeta + D(\theta,\zeta)\zeta + g(\xi) = B(\theta)u, \tag{28}$$

which can be rearranged in state space form

$$\dot{x} = f(x) + g(x)u, \tag{29}$$

where $x = \operatorname{col}(x_1, x_2) = \operatorname{col}(\xi, \zeta)$. With $6 + n = 14$ DOFs and $n + p = 15$ control inputs, the system is overactuated as well as redundant with respect to typical control tasks such as end-effector configuration control.

For an AIAUV, operational space tasks are defined by $\sigma = f(\xi)$, and the Jacobians by

$$\begin{aligned}
\dot{\sigma}(\xi,\zeta) &= \frac{\partial f(\xi)}{\partial \xi} J_\xi(q)\zeta && (30) \\
&= J\zeta. && (31)
\end{aligned}$$

We consider three equality tasks and three set-based tasks, at three different priority levels for simulation. The set-based tasks are safety related and are thus placed at the highest priority level. The safety related tasks consist of end-effector collision avoidance, joint limit avoidance, and actuator matrix singularity avoidance and are all encoded by ECBFs with a

Fig. 1. The Eelume AIAUV [33].

relative degree of two, since they are only functions of the configuration variables. The end-effector collision avoidance and actuator matrix singularity avoidance ECBFs are given by

$$h_1 = \underbrace{\sqrt{\left(p^i_{\text{obs}} - p^i_{ie}\right)^T \left(p^i_{\text{obs}} - p^i_{ie}\right)}}_{\sigma_a} - (r_{\text{obs}} + \epsilon), \tag{32}$$

$$h_2 = \underbrace{\det\left(B(\theta)B^T(\theta)\right)}_{\sigma_b} - \sigma_{b,\min}, \tag{33}$$

where p^i_{ie} and p^i_{obs} is the position of the end-effector, which is placed at the front of the head link and the obstacle, respectively. Eight joints with upper and lower limits yield a total of 16 ECBFs of the form

$$h_{i+2} = \theta_i - \sigma_{c_i,\min}, \tag{34}$$

$$h_{i+10} = \sigma_{c_i,\max} - \theta_i. \tag{35}$$

for $i = 1, \dots, 8$. The second priority level contains the end-effector configuration task

$$y_1 = \begin{bmatrix} p^i_{ie} - p^i_{d,e} \\ \tilde{\epsilon} \end{bmatrix}, \tag{36}$$

where $p^i_{ie} \in \mathbb{R}^3$ is the end-effector position in the inertial frame, and $\tilde{\epsilon}$ is the imaginary part of the quaternion error vector $\tilde{q} = q_d \otimes q^*$, which is given by

$$\tilde{\epsilon} = \eta\epsilon_d - \eta_d\epsilon + [\epsilon]_\times \epsilon_d, \tag{37}$$

where $[\cdot]_\times : \mathbb{R}^3 \to \mathfrak{so}(3) \subset \mathbb{R}^{3\times 3}$ is the skew symmetric map.

The third and final priority level consists of a base positioning task, intended to keep the base of the AIAUV stationary while reconfiguring the end-effector. The end-effector configuration and base positioning tasks only consume 9 DOFs, which entails that there are still 5 uncontrolled DOFs if all set-based tasks are inactive. Stability of the entire system is ensured by eliminating the residual DOFs by a joint velocity regulation task. The task errors are given by

$$y_2 = p^i_{ib} - p^i_{d,b}, \tag{38}$$

$$y_3 = \dot{\theta}. \tag{39}$$

Because y_1 and y_2 only depend on the configuration variables ξ, they have to be differentiated twice with respect to time for the input u to appear. The joint velocity regulation task y_3 is a function of the generalized velocity ζ, and thus needs to be differentiated once for the input to show up. Hence, $\rho_1 = \rho_2 = 2$ and $\rho_3 = 1$. Similarly, (32)-(35) all have to be differentiated twice with respect to time for the input to show up, hence $r_1 = \dots = r_{18} = 2$. According to Algorithm 1, we solve the following QP:

$$\begin{aligned} &\underset{u\in\mathbb{R}^{15},\delta_1\in\mathbb{R}}{\text{minimize}} \quad u^TA^TAu + 2b^TAu + w_1\delta_1^2 \\ &\text{subject to} \\ &L_{\bar{f}_1}V_{\epsilon,1} + L_{\bar{g}_1}V_{\epsilon,1}u \le -\frac{\gamma_1}{\epsilon_1}V_{\epsilon,1} + \delta_1, \\ &L_f^2h_k + L_gL_fh_ku \ge -K_{\alpha,k}\xi_k, \qquad k = 1,\dots,18, \\ &-50\,\text{N} \le u \le 50\,\text{N}, \\ &-0.1\,\text{N/s} \le \Delta u \le 0.1\,\text{N/s}, \end{aligned} \tag{40}$$

which yields a control input $u = u_1^*$ that only accounts for the safety-related tasks and the end-effector configuration task. However, since the end-effector configuration task only consume 6 DOFs, we may potentially have 8 remaining uncontrolled DOFs. The solution u_1^* can be refined by attempting to keep the base stationary and minimizing the joint velocities as follows

$$\begin{aligned} &\underset{u\in\mathbb{R}^{15},(\delta_2,\delta_3)\in\mathbb{R}^2}{\text{minimize}} \quad u^TA^TAu + 2b^TAu + w_2\delta_2^2 + w_3\delta_3^2 \\ &\text{subject to} \\ &L_{\bar{g}_1}V_{\epsilon,1}u \le L_{\bar{g}_1}V_{\epsilon,1}u_1^*, \\ &L_{\bar{f}_j}V_{\epsilon,j} + L_{\bar{g}_j}V_{\epsilon,j}u \le -\frac{\gamma_j}{\epsilon_j}V_{\epsilon,j} + \delta_j, \quad j = 2,3, \\ &L_gL_fh_ku \ge L_gL_fh_ku_1^*, \qquad k = 1,\dots,18, \\ &-50\,\text{N} \le u \le 50\,\text{N}, \\ &-0.1\,\text{N/s} \le \Delta u \le 0.1\,\text{N/s}, \end{aligned} \tag{41}$$

which yields the final control input $u = u_2^*$ that is applied to the AIAUV.

The optimization problems are formulated in terms of the thruster and joint torque control inputs u, and not the commanded forces and torques $\tau = Bu$. Consequently, the proposed framework also solves the control allocation problem. By unifying the redundancy resolution and control allocation problems, strict priority among tasks can always be ensured. The same cannot be said for other redundancy resolution schemes which decouple dynamic control and control allocation, since the commanded forces and torques may not be exactly allocable, leading to a loss of priority among tasks.

Simulation results are presented in Figures 2 to 5. From Figures 4 and 5 we observe that the set-based tasks are satisfied at all times, where the collision avoidance task results in a small deviation in the x-coordinate of the end-effector position from $t \simeq 268\,\text{s}$ to $t \simeq 281\,\text{s}$. Moreover, from $t \geq 350\,\text{s}$, the end-effector position is commanded outside of the manipulator workspace (when the base is kept at its current position), which implies that the base positioning task is no longer compatible with the end-effector positioning task. As expected, the end-effector position converges to its desired value at the expense of a greater error in the base position.

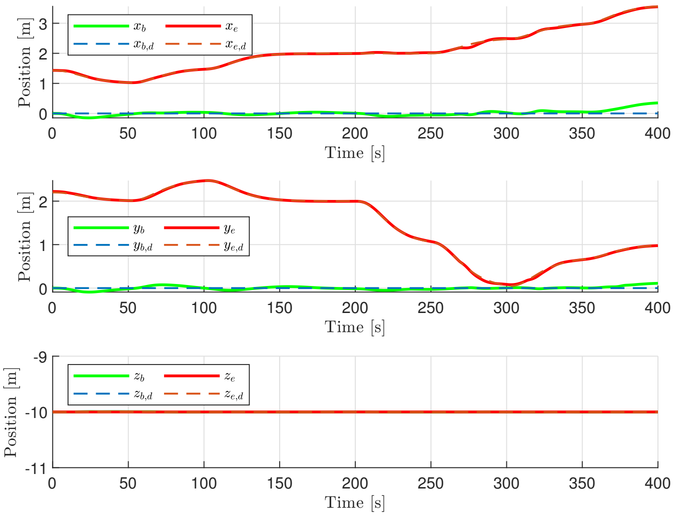


Fig. 2. The x, y, z positions of the end-effector p_e and base p_b.

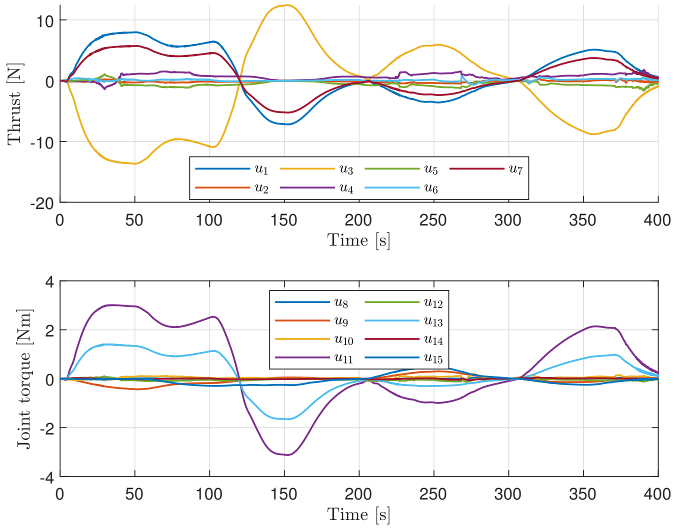


Fig. 3. The thruster and joint torque control inputs.

V. Conclusion

This paper has presented a novel framework for dynamic control of redundant robotic systems based on a hierarchy of CLF-ECBF QPs. The framework generalizes the work in [18], [27] to any number of priority levels. There is considerable design freedom within this framework, soft priority measures can be employed between tasks at the same priority level, while a strict priority measure in the form of solving another QP can be used to ensure that lower priority tasks have no effect on higher priority tasks. This allows optimization based control objectives to be safely included, without affecting the execution of mission related tasks.

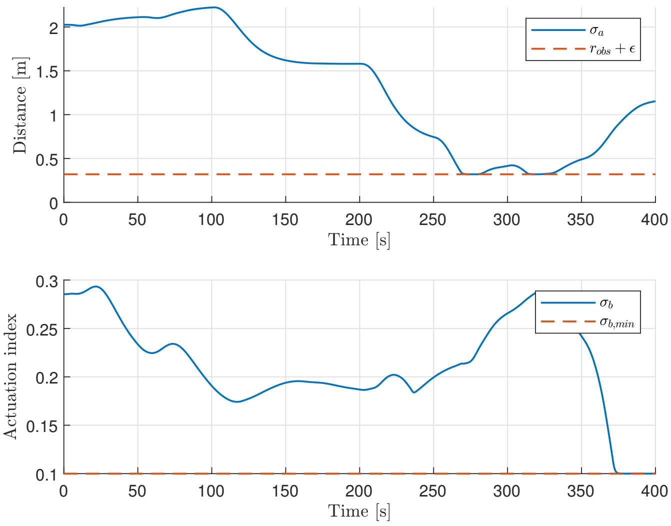


Fig. 4. The distance to the center of the spherical obstacle σ_a and its minimum value, and the actuation index σ_b and its minimum value.

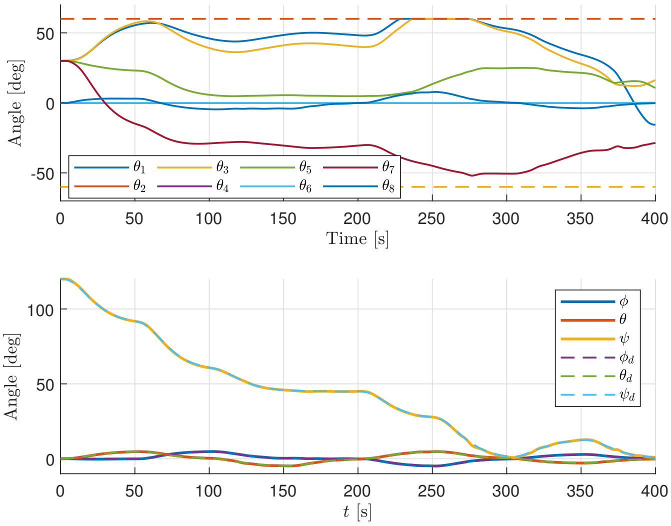


Fig. 5. The joint angles θ, their maximum and minimum limits and the attitude response.

The proposed framework has been verified in simulation for an AIAUV, which is an overactuated and redundant robotic system. For these types of systems, the proposed task priority framework also solves the control allocation problem, which is highly advantageous since control input bounds and rate constraints can be accounted for when resolving redundancy, effectively avoiding a situation in which commanded generalized forces and torques cannot be allocated explicitly, leading to a loss of priority among tasks.

References

[1] P. D. Lillo, F. Arrichiello, G. Antonelli, and S. Chiaverini, "Safety-related tasks within the set-based task-priority inverse kinematics framework," pp. 6130–6135, Oct 2018.

[2] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and control of articulated robot arms with redundancy," in *Proc. 8th IFAC World Congress on Control Science and Technology for the Progress of Society*, aug 1981.

[3] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Rob. Res.*, vol. 6, no. 2, pp. 3–15, Jul. 1987.

[4] B. Siciliano and J.-. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. Fifth Int. Conf. Advanced Robotics 'Robots in Unstructured Environments*, Jun. 1991.

[5] 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*, vol. 3, p. 16, 2016.

[6] 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*, vol. 84, no. 1, pp. 877–902, Dec 2016.

[7] O. Kanoun, F. Lamiraux, and P. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, Aug 2011.

[8] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, Feb. 1987.

[9] L. Sentis and O. Khatib, "Prioritized multi-objective dynamics and control of robots in human environments," in *Proc. 4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, Nov. 2004.

[10] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 670–685, June 2009.

[11] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 7, no. 11, pp. 1163–1173, 1983.

[12] E. D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," *SIAM Journal on Control and Optimization*, vol. 21, no. 3, pp. 462–471, 1983.

[13] ——, "A universal construction of artstein's theorem on nonlinear stabilization," *Systems & Control Letters*, vol. 13, no. 2, pp. 117–123, 1989.

[14] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 876–891, April 2014.

[25] M. Z. Romdlony and B. Jayawardhana, "Uniting control Lyapunov and control barrier functions," in *Proc. 53rd IEEE Conference on Decision and Control*, dec 2014.

[15] R. A. Freeman and P. Kokotović, *Robust Nonlinear Control Design*. Birkhäuser Boston, 1996.

[16] I. R. Petersen and B. R. Barmish, "Control effort considerations in the stabilization of uncertain dynamical systems," *Systems & Control Letters*, vol. 9, no. 5, pp. 417–422, nov 1987.

[17] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, "Torque saturation in bipedal robotic walking through control Lyapunov function-based quadratic programs," *IEEE Access*, vol. 3, pp. 323–332, 2015.

[18] A. Ames and M. Powell, "Towards the unification of locomotion and manipulation through control Lyapunov functions and quadratic programs," *Control of Cyber-Physical Systems*, pp. 219–240, 2013.

[19] A. Forsgren, P. E. Gill, and M. H. Wright, "Interior methods for nonlinear optimization," *SIAM Review*, vol. 44, no. 4, pp. 525–597, 2002.

[20] L. V. Stephen Boyd, *Convex Optimization*. Cambridge University Press, 2019.

[21] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Hybrid Systems: Computation and Control*. Springer, 2004, pp. 477–492.

[22] S. Prajna, "Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 1, pp. 117–126, jan 2006.

[23] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, apr 2009.

[24] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[26] ——, "Stabilization with guaranteed safety using control Lyapunov –barrier function," *Automatica*, vol. 66, pp. 39–47, apr 2016.

[27] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. 53rd IEEE Conf. Decision and Control*, Dec. 2014.

[28] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *Proc. American Control Conf. (ACC)*, Jul. 2016.

[29] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[30] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," mar 2019.

[31] H. M. Schmidt-Didlaukies, A. J. Sorensen, and K. Y. Pettersen, "Modeling of articulated underwater robots for simulation and control," in *Proc. 2018 IEEE OES Autonomous Underwater Vehicle Symposium (AUV)*. IEEE, nov 2018.

[32] P. Liljebäck and R. Mills, "Eelume: A flexible and subsea resident IMR vehicle," in *OCEANS 2017 - Aberdeen*, June 2017.

[33] Eelume. [Online]. Available: https://www.eelume.com

References

[1] 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*, vol. 43, no. 2, pp. 402–417, Apr. 2018.

[2] E. I. Grøtli, J. Tjønnås, J. Azpiazu, A. A. Transeth, and M. Ludvigsen, “Towards more autonomous ROV operations: Scalable and modular localization with experiment data,” *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 173–180, 2016.

[3] A. Kohl, “Redundancy resolution methods for articulated intervention-AUVs,” Norwegian University of Science and Technology, Tech. Rep., 2018.

[4] P. Liljebäck and R. Mills, “Eelume: A flexible and subsea resident IMR vehicle,” in *OCEANS 2017 - Aberdeen*, Jun. 2017.

[5] Eelume. [Online]. Available: `https://www.eelume.com` (visited on 07/08/2019).

[6] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, “Analysis and control of articulated robot arms with redundancy,” in *Proc. 8th IFAC World Congress on Control Science and Technology for the Progress of Society*, Kyoto, Japan, Aug. 1981.

[7] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, “Task-priority based redundancy control of robot manipulators,” *Int. J. Rob. Res.*, vol. 6, no. 2, pp. 3–15, Jul. 1987.

[8] B. Siciliano and J.-. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. Fifth Int. Conf. Advanced Robotics 'Robots in Unstructured Environments*, Pisa, Italy, Jun. 1991.

[9] 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*, 2016.

[10] 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*, vol. 84, no. 1, pp. 877–902, Dec. 2016.

[11] O. Kanoun, F. Lamiraux, and P. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, Aug. 2011.

[12] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, Feb. 1987.

[13] L. Sentis and O. Khatib, "Prioritized multi-objective dynamics and control of robots in human environments," in *Proc. 4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, Santa Monica, CA, USA, Nov. 2004.

[14] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, Jun. 2009.

[15] E. A. Basso, "Set-based task priority control for articulated intervention AUVs," Norwegian University of Science and Technology, Tech. Rep., Feb. 2019.

[16] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 7, no. 11, pp. 1163–1173, 1983.

[17] E. D. Sontag, “A Lyapunov-like characterization of asymptotic controllability,” *SIAM Journal on Control and Optimization*, vol. 21, no. 3, pp. 462–471, 1983.

[18] ——, “A universal construction of artstein’s theorem on nonlinear stabilization,” *Systems & Control Letters*, vol. 13, no. 2, pp. 117–123, 1989.

[19] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, “Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, Apr. 2014.

[20] R. A. Freeman and P. Kokotović, *Robust Nonlinear Control Design*. Birkhäuser Boston, 1996.

[21] I. R. Petersen and B. R. Barmish, “Control effort considerations in the stabilization of uncertain dynamical systems,” *Systems & Control Letters*, vol. 9, no. 5, pp. 417–422, Nov. 1987.

[22] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, “Torque saturation in bipedal robotic walking through control Lyapunov function-based quadratic programs,” *IEEE Access*, vol. 3, pp. 323–332, 2015.

[23] A. Ames and M. Powell, “Towards the unification of locomotion and manipulation through control Lyapunov functions and quadratic programs,” *Control of Cyber-Physical Systems*, pp. 219–240, 2013.

[24] A. Forsgren, P. E. Gill, and M. H. Wright, “Interior methods for nonlinear optimization,” *SIAM Review*, vol. 44, no. 4, pp. 525–597, 2002.

[25] L. V. Stephen Boyd, *Convex Optimization*. Cambridge University Press, Mar. 8, 2019, 732 pp.

[26] S. Prajna and A. Jadbabaie, “Safety verification of hybrid systems using barrier certificates,” in *Hybrid Systems: Computation and Control*, Springer, 2004, pp. 477–492.

[27] S. Prajna, “Barrier certificates for nonlinear model validation,” *Automatica*, vol. 42, no. 1, pp. 117–126, Jan. 2006.

[28] K. P. Tee, S. S. Ge, and E. H. Tay, “Barrier Lyapunov functions for the control of output-constrained nonlinear systems,” *Automatica*, vol. 45, no. 4, pp. 918–927, Apr. 2009.

[29] P. Wieland and F. Allgöwer, “Constructive safety using control barrier functions,” *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[30] M. Z. Romdlony and B. Jayawardhana, “Uniting control Lyapunov and control barrier functions,” in *Proc. 53rd IEEE Conference on Decision and Control*, Los Angeles, CA, USA, Dec. 2014.

[31] ——, “Stabilization with guaranteed safety using control Lyapunov –barrier function,” *Automatica*, vol. 66, pp. 39–47, Apr. 2016.

[32] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *Proc. 53rd IEEE Conf. Decision and Control*, Los Angeles, CA, USA, Dec. 2014.

[33] Q. Nguyen and K. Sreenath, “Exponential control barrier functions for enforcing high relative-degree safety-critical constraints,” in *Proc. American Control Conf. (ACC)*, Boston, MA, USA, Jul. 2016.

[34] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[35] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, *Control barrier functions: Theory and applications*, Mar. 2019. arXiv: `1903.11199 [cs.SY]`.

[36] 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. IEEE Conf. Control Technology and Applications (CCTA)*, Kohala Coast, HI, USA, Aug. 2017, pp. 142–149.

[37] M. F. Amundsen, “Control of an underwater swimming manipulator, with com- pensation for reaction forces and hydrostatic forces,” Master’s thesis, Norwegian University of Science and Technology, Jun. 2017.

[38] M. F. Amundsen, J. Sverdrup-Thygeson, E. Kelasidi, and K. Y. Pettersen, “Inverse kinematic control of a free-floating underwater manipulator using the generalized Jacobian matrix,” in *Proc. 2018 European Control Conference (ECC)*, Limassol, Cyprus, Jun. 2018.

[39] B. H. Hoffmann, “Path following and collision avoidance for an underwater swimming manipulator,” Master’s thesis, Norwegian University of Science and Technology, Jun. 2018.

[40] I.-L. G. Borlaug, K. Pettersen, and J. Gravdahl, “Trajectory tracking for an articulated intervention AUV using a super-twisting algorithm in 6 DOF,” in *Proc. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles (CAMS 2018)*, Opatija, Croatia, Sep. 2018.

[41] S. V. Aalbu, “Subsea inspection and intervention with underwater swimming manipulators,” Master’s thesis, Norwegian University of Science and Technology, Jun. 2018.

[42] A. Isidori, *Nonlinear Control Systems*, 3rd ed. Springer, 1995.

[43] S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control.* Springer, 1999.

[44] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control.* Wiley, 2005.

[45] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics.* Springer, 2016.

[46] Q. Nguyen and K. Sreenath, “Optimal robust control for constrained nonlinear hybrid systems with application to bipedal locomotion,” in *Proc. American Control Conf. (ACC)*, Boston, MA, USA, Jul. 2016.

[47] J. M. Lee, *Introduction to Smooth Manifolds*, 2nd ed. Springer, 2012.

[48] C.-T. Chen, *Linear System Theory and Design*, 3rd. Oxford University Press, 1998.

[49] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, 2nd ed. Springer, 1998.

[50] V. K. Nguyen, "Partial derivative of matrix functions with respect to a vector variable," *Vietnam Journal of Mechanics*, vol. 30, Jul. 2012.

[51] L. Sentis, J. Petersen, and R. Philippsen, "Implementation and stability analysis of prioritized whole-body compliant controllers on a wheeled humanoid robot in uneven terrains," *Autonomous Robots*, vol. 35, no. 4, p. 301, Nov. 1, 2013.

[52] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 876–891, Apr. 2014.

[53] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and Adaptive Control Design*. Wiley, 1995.

[54] H. M. Schmidt-Didlaukies, "Modeling of articulated underwater robots for simulation and control," Norwegian University of Science and Technology, Tech. Rep., 2018.

[55] H. M. Schmidt-Didlaukies, A. J. Sorensen, and K. Y. Pettersen, "Modeling of articulated underwater robots for simulation and control," in *Proc. 2018 IEEE OES Autonomous Underwater Vehicle Symposium (AUV)*, Porto, Portugal: IEEE, Nov. 2018.

[56] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, 2008.

[57] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[58] R. Murray, Z. Li, S. Sastry, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* Taylor & Francis, 1994.

[59] M. Wrzos-Kaminska, “Path following control for underwater swimming manipulators moving in 3D,” Master’s thesis, Norwegian University of Science and Technology, Jun. 2018.

[60] G. Antonelli, *Underwater Robots*, 4th ed. Springer, 2018.

[61] A. E. Stene, “Operational space singularity avoidance for articulated intervention AUV,” Norwegian University of Science and Technology, Tech. Rep., 2018.

[62] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, “Operational space control: A theoretical and empirical comparison,” *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[63] D. R. Yoerger, J. G. Cooke, and J. .-.-. E. Slotine, “The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design,” *IEEE Journal of Oceanic Engineering*, vol. 15, no. 3, pp. 167–178, Jul. 1990.

[64] Q. Nguyen and K. Sreenath, “Optimal robust control for bipedal robots through control Lyapunov function based quadratic programs.,” in *Robotics: Science and Systems*, 2015.

[65] H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[66] P. Renteln, *Manifolds, Tensors, and Forms: An Introduction for Mathematicians and Physicists.* Cambridge University Press, 2013.

Erlend Andreas Basso

Dynamic Task Priority Control of Articulated Intervention AUVs

NTNU
Norwegian University of
Science and Technology