Towards Operating Underactuated Robotic Systems by Going With the Flow

by

Marius Wiggert

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Claire Tomlin, Chair
Professor Pierre Lermusiaux
Professor Shankar Sastry
Professor Pieter Abbeel

Summer 2023

Towards Operating Underactuated Robotic Systems by Going With the Flow

Abstract

Towards Operating Underactuated Robotic Systems by Going With the Flow

by

Marius Wiggert

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Claire Tomlin, Chair

Over the centuries, humanity has created ever more ingenious systems to traverse the oceans and skies of our planet. Modern ships and planes operate with powerful engines that require substantial amounts of fuel, leading to high operating costs. However, this approach becomes impractical for applications that require extended periods of autonomous operation without the possibility of refueling. This dissertation starts with the idea of operating systems by *going with the flow*: harnessing the wind and ocean currents by letting the system drift in favorable directions and strategically using a low-power engine to change flows when this is beneficial. As the power to counteract drag forces scales cubically with the relative velocity of the system, this new paradigm reduces the power required for operation by 2-3 orders of magnitude, thereby significantly reducing the system and operating costs. This could enable a host of novel applications that require low-cost and long-term operations, such as active environmental monitoring of the oceans and atmosphere or floating solar platforms. The primary case study used throughout this work is autonomous seaweed farms that roam the oceans while rapidly growing biomass for biofuel, bioplastic, or to sink it for carbon removal.

In this dissertation, we systematically develop control techniques to tackle the four key challenges of operating by *going with the flow*: First, the system is severely underactuated with its own propulsion often being less than $\frac{1}{10}th$ of the magnitude of the surrounding flows. Second, to make strategic control decisions when to change flows, only coarse, deterministic forecasts are available. Third, the forecasts have a limited time horizon of 5-10 days, but realistic control objectives extend over weeks to months. Lastly, the forecast error defined as the difference between the forecasted and the true flows often exceeds the propulsion capabilities of the system, hence robust control is infeasible.

We start by introducing techniques for continuous-time optimal control when the complex flows are known. We use dynamic programming for the objectives of navigation and maximizing seaweed growth. Next, we turn towards the challenge of operating with imperfect and short-term forecasts. Our insight is that the value functions obtained by the previously developed optimal control methods can be used as closed-loop control policies, which are equivalent to replanning on the forecast at every step. Through extensive simulation studies in realistic ocean conditions, we demonstrate that such frequent replanning allows for reliable operation despite significant forecast errors. To enable reasoning beyond the forecast horizon, we derive a discounted optimal control formulation and demonstrate how the value function can be extended by estimating the cost-to-go using historical flow averages. In the last part of this dissertation, we focus on how to handle constraints in these challenging environments. For that, we integrate time-varying obstacles into our value function and show empirically that this almost eliminates the risk of stranding. Moreover, we develop a hierarchical control approach to operate a fleet of underactuated autonomous systems while avoiding collisions and ensuring connectivity across the fleet.

At the end of this dissertation, we summarize our techniques for operating by *going with the flow* which could enable a host of new applications of low-power autonomous systems in the oceans and skies. We also discuss promising ongoing and future research directions towards further improving the performance of underactuated robotic systems operating in flows.

To all life on our wonderful planet, may this research help protect it.
To my parents, Monika and Henning, and my twin sister Antonia.

Contents

List of Figures

List of Tables

Acknowledgments

First of all, I am deeply grateful to my wonderful PhD advisor Claire Tomlin, for her unwavering support, her trust in me that allowed me the space to develop this research project, and her genuine kindness in guiding me and helping me flourish. I aspire to mentor and help others grow the way Claire helped me. I also owe much gratitude to the other members of my dissertation committee, Pierre Lermusiaux, Shankar Sastry, and Pieter Abbeel. Pierres' extensive knowledge about the ocean currents, his insightful questions, and his humor were a tremendous support in my PhD journey and made this research possible. It has been a pleasure to be around and learn from Shankar and I am grateful for his constant encouragement to share my work at workshops and in his classes. There were many moments that I got inspired by his insightful questions at the end of a seminar pointing towards the larger context of the research and challenging all of us to take a broader perspective, especially around societal-scale cyber physical systems. Lastly, Pieter was not only one of the sharpest teachers I had the honor to learn from but also incredibly humble and approachable to students. His path of combining cutting edge research and bringing it into application through Start-Ups is an inspiration for my next steps after this PhD.

It was Nico Julian and Jeff Zerger from Phykos who had the initial hunch that maybe they could operate their floating seaweed farms by hitchhiking on ocean currents. I am grateful for their trust in working with me to develop the control methods to enable this paradigm and discussing the real challenges that the system will face. Thanks to them my PhD allowed me to work on hard intellectual challenges that enable a novel solution for scalable climate change mitigation.

The results and insights reached in this dissertation would not have been possible with the many close collaborations and stimulating discussions with my peers. The whiteboard sessions with Manan Doshi from MIT were critical to hatch many ideas for the techniques in this thesis. I also had the honor to mentor my own little group of Master students from TUM and ETH: Jerome Jeannin, Jonas Dieker, Killian Kempf, Matthias Killer, Nicolas Hoischen, and Andreas Doering. Thank you for allowing me to mentor you, learn with you, and just being in the lab together. Thank you also to Andreea Bobu, with whom I first experienced the joys of really vibing with my collaborator. I also want to express my gratitude towards my collages in the Hybrid Systems Lab for making conference travel fun, and discussing many ideas over lunches and lab meetings and sharing the PhD journey, Vicenc, Jaime, Sylvia, David, Patricia, Varun, Somil, Michael, Alonso, Andrea, Gabriel, Donggun, Margareth, Keith, Katie, Jason, Jingqi, Chams, Sara, and Ebony. Thank you also to Ashwin Balakrishna for many fun hangouts and discussions about Reinforcement Learning and life and my mentor Sergey Yaroshenko at Google X for his great support.

I had the great pleasure to live in International House and meet many friends from around the world who inspired me and expanded my worldview in many ways. First of all I am grateful for my girlfriend Daniela, for her encouraging love that helped me grow. Thank you to my dear friends Linnan, Benjamin, Ando, Micah, Raghav, William, Fatima, Mikey, and many more who I shared trips, meals, and conversations with.

Part I

Introduction and Background

Chapter 1

Introduction

Humanity's journey through the ages has been marked by our endeavors to explore our world, crafting ever more ingenious systems to traverse the oceans and skies. As this journey has unfolded, we have developed an array of vessels and systems, each designed to operate in different environments and serve various needs. In the oceans we transitioned from simple sailboats to modern container ships and submarines that are increasingly operating autonomously. Similarly, in the realm of aviation, we have progressed from lighter-than-air balloons and gliders to modern commercial airplanes and drones.

The vast majority of our contemporary systems traversing the skies and oceans are powered by powerful engines. The operational costs of these airplanes and ships are driven by the fuel required for these engines to continually counteract the drag force F_D of pushing them through air or water at high velocities v. The engine power to counteract the drag is:

$$P = F_D \cdot v = \frac{1}{2}\rho A_{cross} C_D v^3. \tag{1.1}$$

Where A_{cross} is the cross-sectional area of the system, ρ the density of the medium, and C_D the drag coefficient. This has led research toward optimizing the shape of these structures for smaller A_{cross} and improved aerodynamic and hydrodynamic coefficients C_D.

However, certain applications require long-term independent operation without the possibility of refueling. In such cases, operating a high-powered engine is not feasible, as it would require a large source of energy, such as fuel tanks or energy generation with large solar arrays. But the oceans and skies are permeated by natural flows such as winds and ocean currents, so an alternative is to passively let the system drift on these flows. This method has been adopted in drifter buoys for ocean monitoring [88] and balloons both for weather monitoring [63] and in WWII when the Japanese launched balloons to transport bombs to the continental US [4]. Although they are cost-effective, the trajectories of these drifting systems are uncontrollable and dependent on natural flows.

However, there are numerous potential applications that require control of the system, but utilizing powerful engines is infeasible. The primary example that inspired this thesis is the vision proposed by the Start-Up Phykos, to build autonomous seaweed farms that roam the oceans while growing seaweed [40, 122] (Figure 1.1). These robotic farms could scale seaweed farming from labor intensive practices near the coast to the vast expanse of the open oceans [137, 12], thus enabling various applications of seaweed biomass for climate mitigation. Seaweed can be used as human food, as cattle feed that reduces methane emissions from burping [115], as bio fuel and plastic to replace oil [147], and to capture up to 12.3 $GtCO_2$/year from the atmosphere when the biomass is sunk to the ocean floor [143, 120]. To operate such floating farms, we need to be able to control their position to prevent stranding and colliding with ships, as well as maximize seaweed growth by staying in nutrient-rich waters and minimize negative environmental externalities by avoiding protected marine zones and ensuring precise biomass sinking in suitable locations (Fig. 1.1). The drag forces for operating these farms at appropriate speeds v are enormous due to the inherently high cross-sectional area, A_{cross}, and the drag coefficient C_D. These drag forces make steering farms with powerful ship engines prohibitively expensive [116].

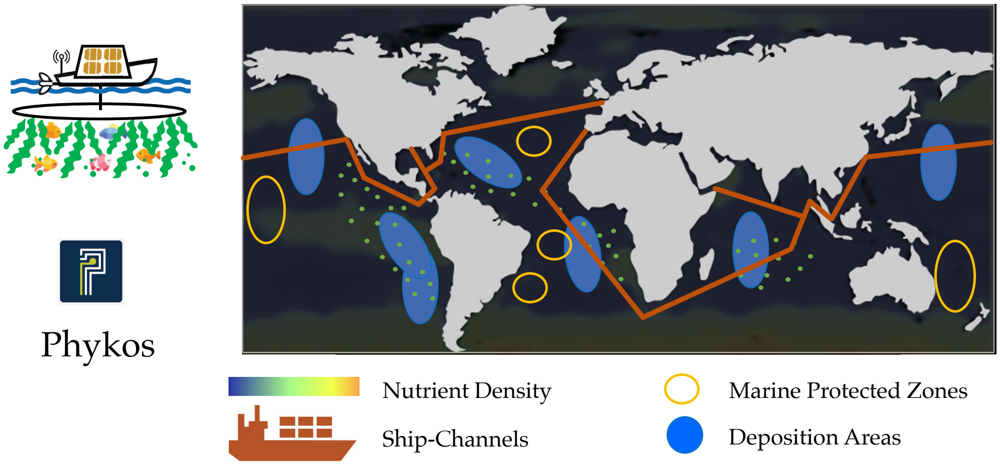


Figure 1.1: The Start-Up Phykos is building autonomous seaweed farms that roam the ocean. For efficient operations they need to stay in high nutrient areas and avoid stranding, shipping channels, and marine protected zones. At the end of a growth cycle they deposite the biomass in deep ocean trenches for carbon removal.

This dissertation starts with the idea of operating systems by ***going with the flow***: letting the system drift in a favorable direction and strategically using a small engine to change flows when this is beneficial.

Consider, as an example, the floating seaweed farm in Figure 1.2: By strategically applying its small propulsion of $u_{max} = 0.1\frac{m}{s}$, which is significantly less than surrounding flows, it can successfully navigate from the start position to the target area in 40 days. The diversity and complexity of the currents with numerous bifurcations and eddies makes this possible, as small propulsion can be enough to change from one eddy to the next.

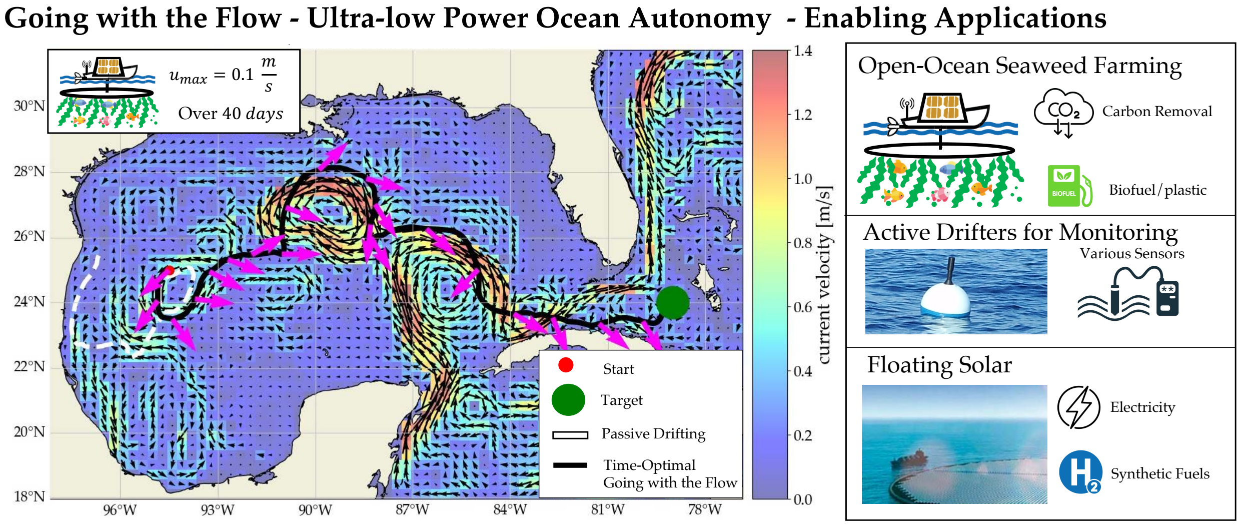


Figure 1.2: By using its small propulsion u_{max} strategically, the autonomous seaweed farm can leverage the ocean currents to reach the target within 40 days whereas passive drifting would lead it along the white trajectory. This paradigm of *going with the flow* requires only small motors which enables a host of new applications that would otherwise be prohibitively expensive.

This paradigm is particularly promising, as the required operating power scales cubically with the relative velocity v (Eq. 1.1) and this paradigm requires only minimal relative velocity. Let us compare the power required by *going with the flow* with the use of powerful engines: operating with 1 $\frac{m}{s}$ or 10 $\frac{m}{s}$ (the cruise speed of a container ship), that is, 10 times and 100 times the speed of $u_{max} = 0.1\frac{m}{s}$, implies a power requirement of 1000 times and 1.000.000 times as much as *going with the flow*. The power requirements determine the required motor, battery, and solar panel sizes and thereby the cost of the system. To compare the total energy required for the journey in Fig. 1.2 we need to multiply the power by the duration of the journey. Autonomous farms using powerful engines would arrive at the target faster; Instead of 40 days, it would take 13 days with 1 $\frac{m}{s}$ and 48 hours with 10 $\frac{m}{s}$, which means that by going with the flow, we need 325 times less energy than operating with 1 $\frac{m}{s}$ and 50.000 times less energy than operating at 10$\frac{m}{s}$, the speed of a cruise ship.

This paradigm could also enable other applications such as floating solar farms that could store their energy in synthetic fuels to be picked up by ships [105] or active drifters for ocean monitoring [99]. The principle of *going with the flow* extends beyond the ocean surface to the skies and underwater domains, where systems can leverage the three-dimensional nature of air or water flows at varying altitudes or depths to achieve their operational objectives. An example in the skies are the balloons of Project Loon [52]: they are designed to affordably connect remote rural areas with the Internet by passing messages along a network of balloons to ground stations. These balloons require control over their position but cannot afford the weight of fuel or the large solar panels required to power an engine. The balloons can change their altitude to take advantage of the different stratospheric winds that transport them in various directions [52].

By leveraging complex flows for the vast majority of movement, the power requirements and thus the costs of the systems are reduced by a large magnitude, thus unlocking the possibility for ultra-low-power autonomy in the oceans and skies.

However, these flows are inherently complex, non-linear, and highly varying over time. Reliably navigating hundreds of kilometers on those flows or operating in them when only uncertain short-term forecasts are available is challenging for existing control methods, as we will explain in Section 1.1. Hence, the focus of this dissertation is to develop control methods that enable the operation of underactuated robotic systems by leveraging complex flows in realistic conditions where only deterministic flow forecasts are available.

1.1 Central Challenges for Underactuated Robotic Systems in Flows

To enable the operation paradigm of *going with the flow* we need to develop control methods that address the four central challenges of this paradigm.

Severe Underactuation

Firstly, the system is underactuated with respect to the flow. There are two notions of underactuation that might apply. First, underactuation in the sense of limited degrees of freedom e.g. Loon balloons can only go up and down but operate in a three-dimensional space. Secondly, underactuation in the sense that the maximum propulsion is much smaller than the magnitude of the surrounding flow. For example, our seaweed farm in Fig. 1.2 has a maximum propulsion of $u_{\max} = 0.1\frac{m}{s}$ but the vast majority of flows are stronger with magnitudes up to $2\frac{m}{s}$. In settings with severe underactuation, long-term planning is critical to anticipate favorable flows. Although our methods are generally applicable to operating any kind of system in flows, we focus on the underactuated setting.

Only Coarse, Deterministic Forecasts available

Secondly, the flows are non-linear and highly dynamic, and the predictions available from operational forecast systems of the stratosphere [45] or the oceans [20] are coarse, multi-kilometer resolution and deterministic. The uncertainty in the flows is not modeled, and hence it is hard to hedge against it during operation.

Large Forecast Error

Thirdly, the forecast errors defined as the difference between the forecasted and the true flows are large with respect to the propulsion capabilities of the system. In the oceans, forecast errors are of the order of magnitude of $0.2\frac{m}{s}$ whereas the actuation of our seaweed farms is only $0.1\frac{m}{s}$. This rules out robust control methods as elaborated further in Chapter 2.2.

Limited Forecast Horizon

Thirdly, ocean and atmospheric forecasts are only available for limited horizons. For example, the Ocean System HYCOM provides forecasts over the next 5 day [20], while the time required for navigating hundreds of kilometers leveraging ocean currents is in the range of weeks to months. Therefore, planning will inevitably need to consider what happens after the forecast horizon and adapt as new ocean forecasts and observations are obtained.

1.2 Outline and Contributions

The central goal of this Ph.D. dissertation is to enable underactuated robotic systems to operate with very low power by *going with the flow*. To achieve reliable operations that tackle the four central challenges outlined in Sec. 1.1 we will use ideas and techniques from control theory and machine learning. To ensure applicability of the methods in realistic conditions, the theoretical contributions are complemented with extensive simulation studies that are designed to be as realistic as possible. The organization and key contributions are summarized below.

Chapter 2 defines the general problem of operating in flows and the system dynamics. This chapter will introduce a consistent notation that will be used throughout this document. Moreover, it provides details on flow forecasts and how we ensure a realistic simulation of real-world conditions in our empirical studies.

Part II: Optimal Control in Known Flows. This part of the thesis presents the theory and methods of optimal control when the flows are known based on the principles of dynamic programming. For navigation we introduce our formulation of Multi-Time Reachability in Chapter 3 which is particularly suited for analysis and control of systems in time-varying dynamics such as flow field. In Chapter 4 we introduce how we can maximize seaweed growth or similarly any exponential time-varying running cost while operating in flows. The techniques were first introduced in the papers [31, 60].

Part III: Handling Forecast Uncertainty. The third part focuses on how to operate when the true flows are not known and only deterministic forecasts are available. In Chapter 5 we show how we can use the techniques developed in Part I to ensure reliable operation despite forecast error. This can be done by using the value functions developed in Part II calculated on the forecasts as a feedback control policy that is equivalent to replanning at every step. Furthermore, in Chapter 6 we show how we can reason beyond the short time-horizon of the forecast to optimize long-term objectives by estimating the expected cost to go and discounting the future costs. To validate the techniques, we present extensive simulation studies using realistic conditions. The results in this part were originally introduced in [141, 60].

Part IV: Operating with Constraints. This part investigates how to satisfy constraints for underactuated systems while operating in uncertain flows. Chapter 7 elaborates on the safety concerns when operating in the oceans, focusing and evaluating the risk of stranding and showing how this can be significantly reduced by incorporating obstacles into the value function. Going beyond single systems, Chapter 8 develops a hierarchical control framework for fleets of underactuated robotic systems that want to stay connected in a local mesh network while avoiding collisions and achieving their performance objectives. The results in this part were originally introduced in [48, 30].

Part V: Conclusion and Future Directions. The last part of the thesis will discuss the results in the context of the goal of achieving ultra-low power autonomy by *going with the flow*. We elaborate on future directions to improve performance by leveraging additional information and using techniques from data assimilation and reinforcement learning.

Chapter 2

Background and Preliminaries

2.1 System Dynamics and Problem Statement

Spatial System Dynamics

The robotic systems that we want to operate by leveraging flows can be modeled by differential equations that capture the underlying physical evolution. Mathematical models of moving objects are typically derived from first principles using Newton's second law and using the notion of a state, a set of variables that are sufficient to predict the evolution of the system over time.

For our purpose of operating in n dimensional flows, let the spatial state be denoted by $\boldsymbol{x} \in \mathbb{R}^n$. Our robotics system is not purely passively drifting but its state evolution over time is at least partially influenced by a control value $\boldsymbol{u} \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ that we chose. For example, the direction and magnitude of propulsion for surface ocean vessels or the vertical movement for a balloon. We can then write the evolution of the state of the spatial system through the dynamics Ordinary Differential Equation (ODE)

$$\dot{x}(\boldsymbol{x}, t) = f(\boldsymbol{u}, \boldsymbol{x}, t) = g(\boldsymbol{u}, \boldsymbol{x}, t) + v(\boldsymbol{x}, t; \mu) \tag{2.1}$$

where f denotes the general dynamics. For our case of operating in time-varying flows the spatial movement of the system depends on two terms: the control term $g(\boldsymbol{x}, \boldsymbol{u}, t)$; and the drift due to the time-varying, non-linear flow field $v(\boldsymbol{x}, t; \mu) \to \mathbb{R}^n$ where μ is a random variable that encapsulates the uncertainty of the flow. Although more complex second-order dynamics can be used to model operating in flows, we chose this simplified dynamics for ease of exposition. This makes the common assumption that the drift of the system directly affects its state and neglects any inertial effects. The control can be holonomic when the system can directly actuate in each dimension, e.g. $g(\boldsymbol{u}, \boldsymbol{x}, t) = \boldsymbol{u}$ or nonholonomic, e.g. a balloon can only actuate up and down along the vertical axis. Note that here we only define the system dynamics of the spatial state $\boldsymbol{x}$, but the system might have other states such as the pressure of the balloon, the battery charge, or the seaweed mass. These will be defined in the respective chapters as applicable.

We denote the spatial trajectory induced by this ODE by $\boldsymbol{\xi}$. For a vessel starting in the initial state $\boldsymbol{x}_0$ at time t_0 with control signal $\boldsymbol{u}(\cdot)$, we denote the state at time t by $\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t_0,\boldsymbol{x}_0}(t) \in \mathbb{R}^n$.

The control signal $\boldsymbol{u}(\cdot)$ can be freely chosen to achieve the desired operational objective of the system. Committing to a control signal at t_0 and executing it is called open-loop control and it is rarely good ideas, as the mathematical model often does not fully represent reality. The key idea behind feedback control is to mitigate the inaccuracy of our system dynamics model by making the control input $\boldsymbol{u}$ dependent on the perceived state preceding it, thus closing the perception-action loop. This leads to the notion of a feedback policy, a function $\pi : \mathbb{R}^n \times \mathbb{R} \to \mathcal{U}$ that maps the current state $\boldsymbol{x}$ and time t to a control input $\boldsymbol{u}(t) = \pi(\boldsymbol{x}, t)$.

General Problem Statement

In realistic operational settings only coarse, deterministic forecasts $\hat{v}_{FC}$ of the flows are available which differ significantly from the true currents v (Sect. 1.1). The overarching goal of this dissertation is then to develop methods that use deterministic forecasts $\hat{v}_{FC}$ to determine a feedback control policy π that optimizes an objective in expectation over realistic flow conditions. Although the principles behind our methods apply to arbitrary objectives, throughout this thesis, we focus on the two objectives of reliable and time-optimal navigation (Chapter 3, 5) and maximization of the seaweed mass at a final time (Chapter 4, 6). In Part IV we introduce constraints that the policy π needs to satisfy during operation.

2.2 Evaluation and Realistic Simulation

Evaluating novel control methods can be achieved through two primary avenues: theoretical proofs and empirical simulations. Theoretical proofs involve making assumptions about the system and its operational context, such as the discrepancy between the forecasted flows and the actual flows. Through this mathematical approach, one can determine that a control method will satisfy certain properties such as optimality or constraint satisfaction under the assumptions. As such, the proofs are useful to understand under what conditions and with what performance a control method will work. In Chapter 3, 4, and 8 we prove various properties of our methods such as optimality of the value function when the flows are known and safe multi-agent interactions under conditions on the surrounding flow.

With the vastness of possible operational scenarios, such as the space of possible flows our systems can experience, how can we prove performance? One approach is to assume a bound on the forecast error $\boldsymbol{\delta}$, the difference between the forecasted $\hat{v}_{FC}$ and the true flows v. Using techniques from robust control we can then design a control policy π and guarantee a certain performance despite worst-case forecast errors $\boldsymbol{\delta}$. However, such a robust control policy π does not exist in a setting where the system is significantly underactuated and the average forecast error exceeds the actuation. For example, in our experiments we have $||g(\boldsymbol{u}, \boldsymbol{x}, t)||_2 \leq 0.1\frac{m}{s} \ll \text{RMSE}(\boldsymbol{\delta}) \approx 0.2\frac{m}{s}$ with Root Mean Squared Error as metric for the magnitude of the forecast error. Another approach for stochastic systems is to prove probabilistic bounds, where the system performs at a certain level with high probability. Probabilistic bounds could be established by making strong assumptions on the distribution of the forecast error field $\boldsymbol{\delta}$. But this approach would require an oversimplified flow field for tractability, thereby limiting the applicability of the results in realistic settings.

To overcome these limitations and bridge the gap between theory and practice, we turn to empirical simulations. These provide a more flexible means of evaluation, allowing for performance comparison of control methods under a wide range of realistic operational conditions. Simulations thus become invaluable for evaluating the efficacy of our methods amidst the complexities of the real world. For this reason, we conduct extensive empirical evaluations under realistic conditions in Chapters 5, 6, 7, and 8.

But what constitutes realistic operating conditions for underactuated robotic systems operating in the oceans and skies? The operation of these systems is strongly influenced by natural flows, such as wind and ocean currents. These flows are forecasted and provided at regular intervals by major institutions, which provides the basis for our simulations. In the following we elaborate on the sources and conditions for ocean currents, but the principles equally apply to winds. The two major providers of ocean data are HYCOM [20] and Copernicus [35]. They provide 5-10 day *forecasts* for the global oceans on a daily basis. Additionally, they provide *hindcasts* - retrospective analyses using additional measurement data, such as satellite measured sea surface height which is assimilated with the model to more accurately infer past ocean conditions.

To mirror the practical operation of such systems, we simulate its closed-loop operation as depicted in Figure 2.1. The controller receives daily forecasts $\hat{v}_{FC}$ of the flows and determines the control input $\boldsymbol{u}$ based on these forecasts, the spatial state $\boldsymbol{x}$, and the time t of the system. The dynamics of the system, however, are driven by the true currents v which deviate from the forecasts by the forecast error $\boldsymbol{\delta}$. Therefore, it is critical to ensure a realistic forecast error $\boldsymbol{\delta}$ to provide a faithful representation of real-world operational conditions. Our code base for simulating closed-loop operations is available on GitHub: https://github.com/MariusWiggert/OceanPlatformControl. It is fast and modular so that different flows and control architectures can be simulated.

In our experiments, we use forecasts from two major providers, HYCOM [20] and Copernicus [35]. They provide 5-10 day forecasts for the global oceans on a daily basis. Additionally, they provide hindcasts - retrospective analyses that generate historical data sets by using the current forecast model to simulate past conditions. The challenge is how to model the true currents v such that the forecast error $\boldsymbol{\delta}$ is realistic.

However, defining what constitutes a *realistic* forecast error is not straightforward. In fact, the forecast error is itself a complex entity, a vector field that varies across both space and time $\boldsymbol{\delta}(\boldsymbol{x}, t) : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$. In reality, this error is a function of numerous factors including the dynamism of the flow, the accuracy of measurement tools, and precision of

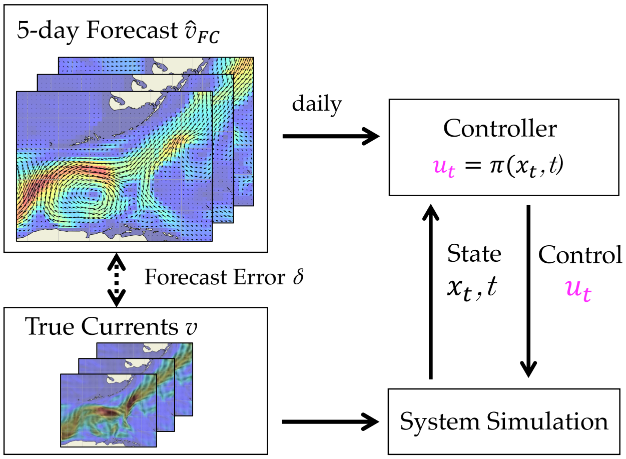


Figure 2.1: The closed-loop simulator provides the controller with daily forecasts $\hat{v}_{FC}$ while simulating the system using the true currents v.

computational models. In certain regions, forecasts are more accurate due to higher availability of observational data or better understood dynamics; conversely, other regions are harder to predict accurately. This results in a diverse error distribution, spatially and temporally, for any given forecast. There are a few sources of measurements of true currents v that allow us to estimate $\boldsymbol{\delta}$. There are high-frequency radar systems which provide coarse observations of surface currents near shore [114], but for the vast majority of the oceans, there are only sparse measurements from surface drifter buoys [88, 89] and vertical drifting ARGO floats [142]. Oceanographers use these sparse measurements to evaluate their forecasting systems and compute summary metrics of the forecast error, such as Root Mean Squared Error and vector correlation [27]. We use these metrics to evaluate the realism of our simulated forecast error.

We developed two approaches to simulate the v while ensuring a realistic forecast error $\boldsymbol{\delta}$. The first approach is to simulate daily forecasts with the forecasts $\hat{v}_{FC}$ from ocean data providers and use the hindcasts they also provide as our 'ground truth' currents v. To estimate how realistic this setup is, we empirically compare the simulated forecast error $\boldsymbol{\delta}$ over a large set of spatial and temporal regions in the Gulf of Mexico with the HYCOM forecast error as estimated by Metzger et. al. using extensive drifter buoy data [94]. In Fig.2.2 we visualize two metrics, the velocity RMSE and the vector correlation, where 2 represents perfect correlation and 0 no correlation [27]. We find that using forecasts and hindcasts from the same provider i.e. HYCOM-HYCOM underestimates the forecast error, especially in the first 24h where the forecast is identical to the hindcast. But using forecasts from one provider and the hindcasts of the other provider e.g. HYCOM-Copernicus yields simulated forecast error of similar magnitude as the actual HYCOM forecast error.

We found this approaches to be fast and straightforward, which is why we use it in

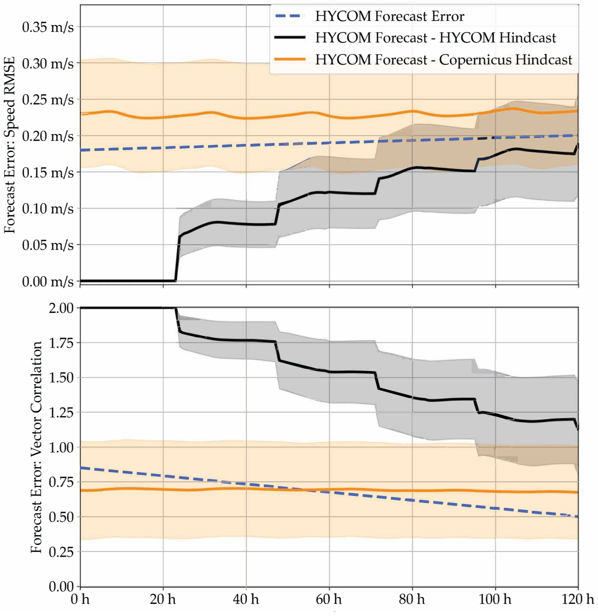


Figure 2.2: The simulated forecast errors in our experiments are of similar magnitude as the actual HYCOM forecast errors [94]. The graphs show the mean and standard deviation of two forecast error metrics, velocity RMSE and vector correlation, over a large set of spatial domains in the Gulf of Mexico.

all experiments in this dissertation. However, in parallel, together with Jonas Dieker we conducted research on how to improve on this approach and make the simulated forecast error even more realistic [28]. The key motivation is that while the difference between the numerical ocean models of the two main providers matches some summary metrics of the true forecast error, it is not ideal because both numerical ocean models neglect small-scale physical effects and make systematic errors. Therefore, as a second approach we developed, trained and evaluated various generative models producing realistic spatio-temporal error profiles $\bar{\delta}$ which can then be added to the hindcasts to serve as the simulation ground truth $v_{sim} = \hat{v}_{HC} + \bar{\delta}$. While these generative models are trained using the extremely sparse observational data of drifter buyos (0.1% of coverage) the forecast errors they produce are diverse and they match higher-order statistics of the empirical forecast error. We refer to [28] for details on these models.

Part II

Optimal Control in Known Flows

Chapter 3

Navigation With Known Flows

This chapter is based on the paper "Hamilton-Jacobi Multi-Time Reachability" [31], written in collaboration with Manan Doshi, Manmeet Bhabra, Claire Tomlin, and Pierre F.J. Lermusiaux.

3.1 Introduction

The navigation of underactuated robotic systems in known flows presents a considerable challenge given the inherent complexity of the dynamics involved. This complexity arises from two interconnected sources: the underactuated nature of the system and the intricate and time-varying character of the flows that we want to utilize to achieve the objective.

There is a rich literature on navigation in flows with both time- and energy-optimal paths in the oceans and air. When flows are known, researchers have proposed Hamilton-Jacobi (HJ) reachability [85, 128, 72], non-linear programming [152, 56], evolutionary algorithms [136], and graph-based techniques such as A* [102, 51, 64], RRT [112, 19], and time-varying Dijkstra [90]. Non-linear programming, evolutionary algorithms, and graph-based methods suffer from discretization error and the non-convexity of the optimization problem which can cause solvers to get stuck in local minima and infeasible solutions. In contrast, HJ reachability is guaranteed to obtain time-optimal paths when the target is reachable, as it solves the exact continuous-time control problem via dynamic programming, which is why we build on it.

Reachability analysis quantifies the set of all states that can be reached by an actuated dynamical system over a specified time horizon. Hamilton-Jacobi reachability formalizes how to calculate the reachable set with differential equations, hence allowing to compute reachable sets for arbitrarily complex dynamics [5, 23]. Classical HJ reachability is mainly concerned with the computation of reachable sets, forward and backward in time, often for robotics and autonomy applications. It provides analyses of the performance and safety of dynamical systems [5], including formal safety guarantees by determining regions of a system's state space that results in catastrophic failure (for example, hitting an obstacle) [22]. With some modifications, it handles moving obstacles and targets in both steady and time-varying systems [66, 13, 14, 39]. HJ reachability is a versatile method for path planning, even in complex environments such as strong and dynamic ocean currents [85, 84, 82]. It can provide the exact solution and is more efficient than other schemes, e.g., graph methods [91]. Reachability planning has been extended to uncertain environments [127] and risk optimality [130]. Finally, HJ reachability has been used in aircraft auto-landing, Model Predictive Control (MPC) of unmanned aerial vehicles (UAVs) and underwater vehicles (AUVs), and multiplayer reach-avoid games [5, 132, 50, 6, 29, 3].

Although classic HJ reachability already allows us to compute a time-optimal path for navigation, replanning is expensive as the Partial Differential Equation (PDE) has to be solved again. In other words, the value function of classic HJ reachability is not suitable for frequent replanning which is necessary when the flows are now perfectly known. Therefore, in this chapter, we introduce a new variant of reachability that expands the type of questions that can be answered and, importantly for this thesis, allows us to use the value function for closed-loop control as detailed in Chapter 5.

The key questions that motivate this work include: i) How can we extend classical reachability theory to multiple start and terminal times?, ii) What is the corresponding value function that provides all level sets at once and what is its governing HJ reachability equation?, iii) Could we compute reachable tubes for all possible times without having to resort to repeated solves of classic reachability PDEs?, iv) What are other quantities that such multi-time forward and backward analysis could compute?, and v) What are the corresponding computational costs?

To address the above questions, we derive a new approach for analyzing the reachability of time-varying dynamical systems that we refer to as *multi-time reachability*. We formulate the optimal control problem using a new running cost term and obtain new governing equations for reachability analysis over multiple start and terminal times all at once, and for systems operating in time-varying environments with dynamic obstacles and any other dynamic fields relevant to their control. Unlike prior results on moving targets and obstacles, we present a theory that incorporates these effects into the dynamics of the system without introducing additional dimensions [66] and fields [39]. We apply our results to three applications and demonstrate that the new governing equations not only more efficiently compute backward and forward reachable tubes, but also generate new secondary quantities that encode valuable reachability information such as duration maps.

In what follows, Section 3.2 outlines the problem statement and introduces key notation. Section 3.3 develops the theory and equations for multi-time reachability. Numerical methods are briefly discussed in 3.4. In Section 3.5, applications and numerical results for new multi-reachability problems are presented, followed by the conclusions in Section 3.6.

3.2 Problem Formulation

Our goal in this chapter is to accurately and efficiently compute reachable tubes for general time-varying dynamical systems. There are essentially two main cases. In the first, we consider a time-varying target set and predict what is known as the backward reachable tube. In the second, we consider a time-varying start set and compute the forward reachable tube. For each case, we derive the governing equations and also solve the added complexity of time-varying obstacles that must be avoided in the dynamic environment.

Next, we formalize the above concepts. We define the properties of the dynamical systems. Then we describe the start, target and obstacle sets, as well as the backward and forward reachable tubes. Finally, we combine all these components and define the types of problems we solve.

System Dynamics

In this work, we consider a general dynamical systems defined by an ordinary differential equation (ODE) of the form

$$\dot{\boldsymbol{\xi}}(s) = \boldsymbol{f}(\boldsymbol{\xi}(s), \boldsymbol{u}(s), s), \quad s \in [0, T], \tag{3.1}$$

with given initial or terminal conditions, where $\boldsymbol{\xi} \in \mathbb{R}^n$ is the system state governed by the ODE, s the temporal variable in an interval $[0, T]$, and $\boldsymbol{u}(\cdot)$ the control from a set $\mathbb{U}$ of measurable functions of $s \in [0, T]$ with values in $\mathcal{U}$:

$$\mathbb{U} = \{\boldsymbol{\phi} : [0, T] \to \mathcal{U} \mid \boldsymbol{\phi}(\cdot) \text{ is measurable}\}. \tag{3.2}$$

The dynamical systems (3.1) govern state variables explicitly affected by the controls (e.g. autonomy variables) but the dynamics $\boldsymbol{f}$ in general includes all other relevant forcing such as the dynamic environment with dynamic obstacles and other dynamic fields that affect the autonomy, e.g. [71, 12].

The control space $\mathcal{U}$ is often a closed bounded set in $\mathbb{R}^{n_u}$, where n_u is the number of control inputs. The system dynamics $\boldsymbol{f} : \mathbb{R}^n \times \mathcal{U} \times \mathbb{R} \to \mathbb{R}^n$ are further assumed to be continuous, bounded, and Lipschitz continuous in $\boldsymbol{\xi}$ uniformly in $\boldsymbol{u}$ [5]. Then, there exists a unique solution to Equation (3.1) for any control sequence $\boldsymbol{u}(\cdot)$ [67, 5]. For the initial value problem, this solution is the trajectory of the system from an initial state $\boldsymbol{x}_0$ at time $t \in [0, T]$ forced by the control sequence $\boldsymbol{u}(\cdot)$ and denoted here by $\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s)$.

Start, Target, and Obstacle Sets

We focus on systems that are either launched from some dynamic start set or are required to reach a dynamic target set, with the constraint of avoiding the dynamic obstacle set, i.e. any dynamic disjoint obstacles that may be present. We now formalize some of the properties of these sets. For each $t \in [0, T]$, we denote the time-varying start, target, and obstacle sets as $\mathcal{S}_t$, $\mathcal{T}_t$ and $\mathcal{O}_t$ respectively, where all sets are closed subsets of $\mathbb{R}^n$. Following closely [39], these sets, in turn, yield corresponding space-time sets $\mathbb{S}$, $\mathbb{O}$, and $\mathbb{T}$, which are all closed subsets of $\mathbb{R}^n \times [0, T]$:

$$\mathbb{S} \coloneqq \bigcup_{t\in[0,T]} \mathcal{S}_t \times \{t\}, \quad \mathbb{T} \coloneqq \bigcup_{t\in[0,T]} \mathcal{T}_t \times \{t\}, \\ \mathbb{O} \coloneqq \bigcup_{t\in[0,T]} \mathcal{O}_t \times \{t\}. \tag{3.3}$$

The start, target, and obstacle sets are further assumed to evolve "smoothly" in time. Specifically, a set $\mathcal{M}_t$ is said to evolve smoothly in time if there exists a Lipschitz continuous function $\boldsymbol{g}_{\mathcal{M}}(\boldsymbol{\xi}) : \mathbb{R}^n \to \mathbb{R}^n$ such that for the system

$$\dot{\boldsymbol{\xi}} = \boldsymbol{g}_{\mathcal{M}}(\boldsymbol{\xi}), \tag{3.4}$$

all trajectories that start in $\mathcal{M}_t$ at some time t stay in the set $\mathcal{M}_{\bar{t}}$ at all subsequent times $\bar{t} \in [t, T]$. That is,

$$\boldsymbol{\xi}_{t,\boldsymbol{x}_0}(\bar{t}) \in \mathcal{M}_{\bar{t}} \qquad\qquad \forall t \in [0, T],\ \forall \boldsymbol{x}_0 \in \mathcal{M}_t,\ \forall \bar{t} \in [t, T]. \tag{3.5}$$

Practically, this implies that the dynamic start, target, and obstacle sets cannot teleport or disappear in the state space.

Reachability Sets and Tubes

In reachability analysis, backward and forward reachable sets or tubes are commonly needed [5]. The maximal sets and tubes encompass all the states to which the system can be driven to when going forward or backward through time, avoiding dynamic obstacles. These maximal sets and tubes can be defined as follows [22, 23, 5]:

Backward Reachable Set (BRS)

Given a specified final time $t_f \in [0, T]$, the BRS at time $t \leq t_f$ is the set of all states at time t that can reach a target set $\mathcal{T}_{t_f}$ *exactly* at the final time t_f:

$$\mathcal{R}(t,t_f, \mathcal{T}_{t_f}, \mathbb{O}) = \{\hat{\boldsymbol{x}} \mid \exists \boldsymbol{u} \in \mathbb{U},\ \boldsymbol{x}_0 = \hat{\boldsymbol{x}}, \\ \boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(t_f) \in \mathcal{T}_{t_f} \wedge \forall s \in [t, t_f],\ \boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(s) \notin \mathcal{O}_s\}. \tag{3.6}$$

Backward Reachable Tube (BRT)

BRTs extend BRSs. Given a specified final time $t_f \in [0, T]$, the BRT at time $t < t_f$ is the set of all states that can reach a time-varying target $\mathbb{T} \subset \mathbb{R}^n \times [0, T]$ at any time $\bar{t} \in [t, t_f]$:

$$\bar{\mathcal{R}}(t,t_f, \mathbb{T}, \mathbb{O}) = \{\hat{\boldsymbol{x}} \mid \exists \boldsymbol{u} \in \mathbb{U},\ \exists \bar{t} \in [t, t_f],\ \boldsymbol{x}_0 = \hat{\boldsymbol{x}}, \\ \boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(\bar{t}) \in \mathcal{T}_{\bar{t}} \wedge \forall s \in [t, t_f],\ \boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(s) \notin \mathcal{O}_s\}. \tag{3.7}$$

Forward Reachable Set (FRS)

Given a specified start time $t_s \in [0, T]$, the FRS at time $t \geq t_s$ is the set of all states that can be reached at time t when starting from a state within the set $\mathcal{S}_{t_s}$ at time t_s:

$$\begin{aligned}\mathcal{F}(t,&t_s, \mathcal{S}_{t_s}, \mathbb{O}) = \{\hat{\boldsymbol{x}} \mid \exists \boldsymbol{u} \in \mathbb{U},\ \exists \boldsymbol{x}_0 \in \mathcal{S}_{t_s},\\ &\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t_s,\boldsymbol{x}_0}(t) = \hat{\boldsymbol{x}} \wedge \forall s \in [t_s, t],\ \boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t_s,\boldsymbol{x}_0}(s) \notin \mathcal{O}_s\}\end{aligned} \tag{3.8}$$

Forward Reachable Tube (FRT)

FRTs extend FRSs. Given a specified start time $t_s \in [0, T]$, the FRT at time $t > t_s$ is the set of all states that can be reached when launched from a state from a time-varying start set $\mathbb{S} \subset \mathbb{R}^n \times [0, T]$ at any time $\bar{t} \in [t_s, t]$:

$$\begin{aligned}\bar{\mathcal{F}}(t,&t_s, \mathbb{S}, \mathbb{O}) = \{\hat{\boldsymbol{x}} \mid \exists \boldsymbol{u} \in \mathbb{U},\ \exists \bar{t} \in [t_s, t],\ \exists \boldsymbol{x}_0 \in \mathcal{S}_{\bar{t}},\\ &\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{\bar{t},\boldsymbol{x}_0}(t) = \hat{\boldsymbol{x}} \wedge \forall s \in [\bar{t}, t],\ \boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{\bar{t},\boldsymbol{x}_0}(s) \notin \mathcal{O}_s\}\end{aligned} \tag{3.9}$$

Problem Statement

Given a dynamical system of the form (3.1) operating in a dynamic environment with obstacles and possibly affected by other dynamic fields, our goal is to obtain equations that govern the backward and forward reachability tubes, as well as schemes that solve these equations efficiently:

Backward Reachability

Given a final time $t_f \in [0, T]$, a time $t < t_f$, and time-varying target and obstacle sets that define space-time sets $\mathbb{T}$ and $\mathbb{O}$, Eq. (3.3), we seek to derive and solve the equations for the BRTs, $\bar{\mathcal{R}}(t, t_f, \mathbb{T}, \mathbb{O})$, Eq. (3.7).

Forward Reachability

Given a start time $t_s \in [0, T]$, a time $t > t_s$, and time-varying start and obstacle sets that define space-time sets $\mathbb{S}$ and $\mathbb{O}$, Eq. (3.3), we seek to derive and solve the equations for the FRTs, $\bar{\mathcal{F}}(t, t_s, \mathbb{S}, \mathbb{O})$, Eq. (3.9).

Once the BRTs and/or FRTs are computed $\forall(t, t_f)$ and/or $\forall(t_s, t)$, they can be used to compute various other quantities of interest. Such quantities include: time-to-reach maps, i.e. maps of the minimum travel time to the target given the present state and time; start time vs. duration plots, i.e. function that maps the travel time to the target to the time at which the trajectory starts given a start state; and time-to-leave maps, i.e. maps of the latest time at which one can depart from the start state and reach the target point at a given time. The computation of these secondary quantities is discussed in Section 3.3

3.3 Multi-time Hamilton Jacobi Reachability

We now develop the theory and obtain the governing equations for HJ multi-time reachability for systems operating in dynamic environments. We start with multi-time reachability in the backward context, i.e. predict backward reachable tubes for time-varying target sets. We use continuous-time, optimal control and show how the resulting value function elegantly captures not only the backward reachable tubes but also time-to-reach maps. We then extend the results to the forward counterpart, i.e. predict forward reachable tubes for time-varying start sets. Finally, we present remarks and discuss secondary quantities such as time-to-reach and time-to-leave maps as well as start time vs. duration plots.

Backward Multi-Time Reachability

Augmented Dynamics

To account for the dynamic target and obstacle sets, we define a new augmented dynamical system as follows:

$$\dot{\boldsymbol{\xi}} = f_a(\boldsymbol{\xi}, \boldsymbol{u}, s) = \begin{cases} g_{\mathcal{O}}(\boldsymbol{\xi}, s), & \boldsymbol{\xi} \in \mathcal{O}_s \\ g_{\mathcal{T}}(\boldsymbol{\xi}, s), & \boldsymbol{\xi} \in \mathcal{T}_s \text{ and } \boldsymbol{\xi} \notin \mathcal{O}_s \\ f(\boldsymbol{\xi}, \boldsymbol{u}, s), & \text{otherwise} \end{cases} . \tag{3.10}$$

where $g_{\mathcal{O}}(\boldsymbol{\xi}, s)$ and $g_{\mathcal{T}}(\boldsymbol{\xi}, s)$ are functions that keep trajectories within their respective sets as defined by Eq. (3.4). With this augmentation, we provide valuable properties to the system. First, once a state enters an obstacle set, it will remain in that set for all subsequent times irrespective of the controls applied. Second, the same holds for states that enter the target set. These properties will be shown to be key in our optimal control setting.

Optimal Control

For our optimal control problem, we first define the terminal cost at time T,

$$l_{term}(\boldsymbol{\xi}) = \begin{cases} \infty, & \boldsymbol{\xi} \in \mathcal{O}_T \\ d(\boldsymbol{\xi}, \mathcal{T}_T), & \text{otherwise} \end{cases} . \tag{3.11}$$

Eq. (3.11) defines the terminal cost of a state in the obstacle set to be infinitely high (in Section 3.4 we address how this property can be numerically handled). For all other states, the cost is defined as the distance of the state ($\boldsymbol{\xi}$) from the target set at the terminal time ($\mathcal{T}_T$) under some distance metric d which depends on the system at hand. Being a distance metric, we require $d(\boldsymbol{\xi}, \mathcal{T}_T) \geq 0 \ \forall(\boldsymbol{\xi}, \mathcal{T}_T)$ and $d(\boldsymbol{\xi}, \mathcal{T}_T) = 0$ if and only if $\boldsymbol{\xi} \in \mathcal{T}_T$.

The running cost is defined as a constant negative value at the target set and zero everywhere else

$$l(\boldsymbol{\xi}, s) = \begin{cases} -\alpha, & \boldsymbol{\xi} \in \mathcal{T}_s \text{ and } \boldsymbol{\xi} \notin \mathcal{O}_s \\ 0, & \text{otherwise} \end{cases}, \tag{3.12}$$

where α is an arbitrary positive constant which we set to 1. The solution is exact irrespective of the value of α. Values of α can however be used to minimize numerical errors due to discontinuities that arise out of this loss function (not shown).

Using the augmented dynamics (3.10), terminal cost (3.11), and running cost (3.12), the total cost function incurred when using controls $\boldsymbol{u}(\cdot)$ and initial state $\boldsymbol{x}_0$ at initial time t is

$$J(\boldsymbol{x}_0, \boldsymbol{u}(\cdot), t) = l_{term}(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T)) + \int_t^T l(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s), s) ds \tag{3.13}$$

To obtain an intuition for the meaning of the total cost, we substitute the functions for the case when the trajectory $\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s)$ never enters the obstacle set,

$$J(\boldsymbol{x}_0, \boldsymbol{u}(\cdot), t) = \underbrace{d(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T), \mathcal{T}_T)}_{\substack{\text{Terminal distance} \\ \text{from target set}}} - \alpha \underbrace{\int_t^T \mathbb{I}_{\mathcal{T}_s} \left(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s) \right) ds}_{\text{Time spent in target set}} \tag{3.14}$$

where the identity function $\mathbb{I}_{\mathcal{T}_s}(\boldsymbol{\xi}) = 1$ when $\boldsymbol{\xi} \in \mathcal{T}_s$ and is 0 otherwise. We note that if the trajectory ever enters the obstacle set, it will stay in the obstacle set at the terminal time under the augmented dynamics and will incur an infinitely high total cost. Additionally, the value function under the optimal control for trajectories that avoids the obstacles is,

$$J^*(\boldsymbol{x}_0, t) = \min_{\boldsymbol{u}(\cdot) \in \mathbb{U}} \left[d(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T), \mathcal{T}_T) - \alpha \int_t^T \mathbb{I}_{\mathcal{T}_s} \left(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s) \right) ds \right] \tag{3.15}$$

To explain this minimization physically, we consider two cases. In the first, we assume there exists some control that drives the system from initial state $\boldsymbol{x}_0$ (or set) at time t into the target set at a time $\bar{t} \in [t, T]$ while avoiding the obstacles. In the opposive second, we assume no control can drive the system to the target set while avoiding the obstacles.

Case 1 Let $\boldsymbol{u}^*(\cdot)$ be the set of controls that drives the system from state $\boldsymbol{x}_0$ at time t (under the augmented dynamics) to the target set at the earliest possible time $t^* = \min(\bar{t})$ while avoiding the dynamic obstacle set $\mathbb{O}$. We reiterate that for the augmented system (3.10), if the target set is reached at some $t^* < T$, the system will stay in the set at all future times and hence $\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T) \in \mathcal{T}_T$. It follows then that the cost for such a set of controls is:

$$\begin{aligned} J(\boldsymbol{x}_0, \boldsymbol{u}^*(\cdot), t) &= \underbrace{d(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T), \mathcal{T}_T)}_{0} - \alpha \underbrace{\int_t^T \mathbb{I}_{\mathcal{T}_s} \left(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s) \right) ds}_{T - t^*} \\ &= -\alpha(T - t^*) \,. \end{aligned}$$

We now provide the lower bound of the value function (3.15) under the optimal control

$$\begin{aligned} J^*(\boldsymbol{x}_0, t) &= \min_{\boldsymbol{u}(\cdot)\in\mathbb{U}} \Big[d(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T), \mathcal{T}_T) - \\ &\qquad\qquad \alpha \int_t^T \mathbb{I}\left\{\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s) \in \mathcal{T}_s\right\} ds \Big] \qquad (3.16) \\ &\geq \min_{\boldsymbol{u}(\cdot)\in\mathbb{U}} \left[d(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T), \mathcal{T}_T) \right] \\ &\qquad + \min_{\boldsymbol{u}(\cdot)\in\mathbb{U}} \left[-\alpha \int_t^T \mathbb{I}\left\{\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s) \in \mathcal{T}_s\right\} ds \right] \qquad (3.17) \\ J^*(\boldsymbol{x}_0, t) &\geq -\alpha(T - t^*)\,. \qquad (3.18) \end{aligned}$$

This lower bound is achieved under the control $\boldsymbol{u}^*$. Therefore, when the vehicle can reach the destination, the optimal control under the given loss function generates a time optimal trajectory to the target state. The value function is given by $J^*(\boldsymbol{x}_0, t) = -\alpha(T - t^*)$ where t^* is the minimum time at which a trajectory starting at $(\boldsymbol{x}_0, t)$ can reach the target state.

Case 2 When there exists no control $\boldsymbol{u}(\cdot)$ that can drive the system from $(\boldsymbol{x}_0, t)$ to the target set while avoiding the obstacles, the term $\mathbb{I}_{\mathcal{T}_s}\left(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s)\right)$ in the value function (3.15) is always 0 by construction. It follows then that:

$$J^*(\boldsymbol{x}_0, t) = \min_{\boldsymbol{u}(\cdot)\in\mathbb{U}} \left[l_{term}(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T)) \right] \qquad (3.19)$$

That is, when a trajectory with initial conditions $(\boldsymbol{x}_0, t)$ cannot reach the target set, the minimization of the cost function will lead the system as close to the target set as possible while avoiding the obstacle (since hitting the obstacle will drive the terminal cost infinitely high).

To summarize, the value function corresponding to the optimal control problem is given as

$$J^*(\boldsymbol{x}_0, t) = \begin{cases} -\alpha(T - \min(\bar{t})), & \text{if } \exists \boldsymbol{u}(\cdot) \text{ s.t.} \\ & \quad \boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(\bar{t}) \in \mathcal{T}_{\bar{t}} \\ \infty, & \text{if } \nexists \boldsymbol{u}(\cdot) \text{ s.t.} \\ & \quad \boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(\bar{t}) \notin \mathcal{O}_{\bar{t}} \\ & \quad \forall \bar{t} \in [t, T] \\ \min_{\boldsymbol{u}} d(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T), \mathcal{T}_T), & \text{Otherwise.} \end{cases} \qquad (3.20)$$

In other words, at any state $\boldsymbol{x}_0$ at a time $t \in [0, T]$, the value of $J^*(\boldsymbol{x}_0, t)$, if negative, physically implies that a state starting at $\boldsymbol{x}_0$ at time t can reach the target set before the terminal time T. Moreover, the earliest time that it can reach the destination is given by $T + \frac{J^*(\boldsymbol{x}_0,t)}{\alpha}$. If the value of $J^*(\boldsymbol{x}_0, t)$ is positive, it implies that a state starting at $\boldsymbol{x}_0$ at time

t cannot reach the target set in the time interval $[t, T]$, and the value physically corresponds to how close such a state could possibly get to the target set at the terminal time T. Finally, for states for which $J^*(\boldsymbol{x}_0, t)$ is infinite, we have that for a system starting at $\boldsymbol{x}_0$ at time t, the obstacle will inevitably be hit.

The Hamilton-Jacobi-Bellman Equation

The value function can be efficiently computed using dynamic programming. For continuous-time optimal control, it is the viscosity solution of the Hamilton-Jacobi-Bellman (HJB) partial differential equation (PDE) [36, 10, 61]

$$\begin{aligned} \frac{\partial J^*(\boldsymbol{x}_0, t)}{\partial t} + \min_{\boldsymbol{u}} \left[l(\boldsymbol{x}_0, t) + \nabla_{\boldsymbol{x}} J^* \cdot f_a(\boldsymbol{x}_0, \boldsymbol{u}, t) \right] = 0 \\ J^*(\boldsymbol{x}_0, T) = l_{term}(\boldsymbol{x}_0), \end{aligned} \tag{3.21}$$

where l_{term} and l are the terminal and running costs, respectively. For our problem, these costs are defined in Eqs. (3.11) and (3.12). Inserting them and the augmented dynamical system (3.10) in Eq. (3.21), we obtain the final HJB PDE:

$$\begin{aligned} \frac{\partial J^*(\boldsymbol{x}_0, t)}{\partial t} &= \begin{cases} -\left[-\alpha + \nabla_{\boldsymbol{x}} J^* \cdot g_{\mathcal{T}}(\boldsymbol{x}_0, t) \right], & \boldsymbol{x}_0(t) \in \mathcal{T}_t \cap (\mathcal{O}_t)^c \\ -\left[\nabla_{\boldsymbol{x}} J^* \cdot g_{\mathcal{O}}(\boldsymbol{x}_0, t) \right], & \boldsymbol{x}_0(t) \in \mathcal{O}_t \\ -\min_{\boldsymbol{u}} \left[\nabla_{\boldsymbol{x}} J^* \cdot f(\boldsymbol{x}_0, \boldsymbol{u}, t) \right], & \text{otherwise} \end{cases} \\ J^*(\boldsymbol{x}_0, T) &= \begin{cases} \infty, & \boldsymbol{x}_0 \in \mathcal{O}_T \\ d(\boldsymbol{x}_0, \mathcal{T}_T), & \text{otherwise} \end{cases}. \end{aligned} \tag{3.22}$$

Eq. (3.22) is a terminal-value problem which is solved backward in time to obtain the value of $J^*(\boldsymbol{x}_0, t)$ for all states in the state space and all times $t \in [0, T]$.

Equation for Backward Reachable Tubes

Using the value function J^* governed by the HJB PDE (3.22), we now obtain the equation for the BRTs defined by Eq. (3.7) as well as an efficient scheme for their computation. Consider a fixed time $t_f \in [0, T]$ and a time $t < t_f$. From Eq. (3.20), it follows that for any state $\boldsymbol{x}_0$ satisfying $J^*(\boldsymbol{x}_0, t) \leq -\alpha(T - t_f)$, a control function $\boldsymbol{u}(\cdot)$ exists that will drive the system from state $\boldsymbol{x}_0$ at time t to the target set at some time $\bar{t} \in [t, t_f]$. This results in an efficient scheme to compute the BRT:

$$\bar{\mathcal{R}}(t,t_f, \mathbb{T}, \mathbb{O}) = \{ \boldsymbol{x}_0 \mid J^*(\boldsymbol{x}_0, t) \leq -\alpha(T - t_f) \}. \tag{3.23}$$

For a specified final time t_f, the BRT at any time $t < t_f$ can simply be extracted by considering the appropriate sub-level set of the value function at that time. This is because an agent in this set would reach the target while avoiding the obstacle at $t < t_f$ under the optimal control and stay in the target because of the augmented dynamics 3.10 accumulating the negative cost at the destination.

Equation for Time-to-Reach Maps

The value function stores important information regarding the optimal time a system can reach the target set. This can be used to compute time-to-reach or duration maps $\mathcal{D}$ from Eq. (3.21) or (3.22):

$$\mathcal{D}(\boldsymbol{x}_0, t) = T + \frac{J^*(\boldsymbol{x}_0, t)}{\alpha} - t, \qquad \forall (\boldsymbol{x}_0, t) \text{ s.t., } J^*(\boldsymbol{x}_0, t) < 0.$$

For a state $\boldsymbol{x}_0$ at time t satisfying $J^*(\boldsymbol{x}_0, t) \leq 0$, Eq. (3.20) implies that such a state can reach the target set and the earliest possible time this will happen will be at $T + \frac{J^*(\boldsymbol{x}_0,t)}{\alpha}$. It follows then that for all $(\boldsymbol{x}_0, t)$ with $J^*(\boldsymbol{x}_0, t) \leq 0$, $\mathcal{D}(\boldsymbol{x}_0, t)$ corresponds to the minimum duration for a trajectory starting at state $\boldsymbol{x}_0$ at time t to reach the target set.

Closed-loop optimal controller

As discussed in Section 3.3, an optimal controller that minimizes the cost function will: (a) avoid dynamic obstacles; (b) reach the target in minimum time if it can; and (c), reach as close to the target as possible if it cannot reach it. The optimal controller can also compute the minimum duration to the target set from the current state. Thus, solving for $J^*(\boldsymbol{x}_0, t)$ using Eq. (3.22) provides a powerful closed-loop control policy:

$$\pi(\boldsymbol{x}_0, t) = \arg\min_{\boldsymbol{u}} \left[\nabla_{\boldsymbol{x}} J^*(\boldsymbol{x}_0, t) \cdot f(\boldsymbol{x}_0, \boldsymbol{u}, t)\right] \ \forall \boldsymbol{x}_0 \notin (\mathcal{O}_t \cup \mathcal{T}_t) \ ,$$

as demonstrated with reliable navigation in complex time-varying ocean currents with forecast errors [141].

Forward Multi-Time Reachability

Section 3.3 addressed backward multi-time reachability. That is, we considered how to efficiently compute backward reachable tubes (and time-to-reach maps) in a dynamic environment with time-varying target and obstacle sets. In this section, we examine the forward counterpart, and derive how to compute forward reachable tubes in dynamic domains containing now time-varying start sets.

The forward problem can be addressed analogously to the derivation in Section 3.3, but now analyzing the system evolution *backwards* in time. First, an augmented dynamic system akin to Eq. (3.10) can be defined as follows:

$$\dot{\boldsymbol{\xi}} = \tilde{f}_a(\boldsymbol{\xi}, \boldsymbol{u}, s) = \begin{cases} g_{\mathcal{O}}(\boldsymbol{x}, s), & \boldsymbol{\xi} \in \mathcal{O}_s \\ g_{\mathcal{S}}(\boldsymbol{x}, s), & \boldsymbol{x} \in \mathcal{S}_s \text{ and } \boldsymbol{\xi} \notin \mathcal{O}_s \ . \\ f(\boldsymbol{\xi}, \boldsymbol{u}, s), & \text{otherwise} \end{cases} \tag{3.24}$$

where now the start set is used instead of the target set. This system's evolution backwards in time can be studied by mapping time to a new "reverse-time" variable, $\tau(t) = T - t$, resulting in a mapped augmented dynamical system:

$$\frac{d\boldsymbol{\xi}}{d\tau} = -\tilde{f}_a(\boldsymbol{\xi}, \boldsymbol{u}, T - \tau)\,. \tag{3.25}$$

Analogous to backward multi-time reachability, we can formulate an optimal control for the system (3.25) while using the start set in place of the target set. Specifically, we define a terminal cost, now at time $\tau = T$, of the form:

$$\tilde{l}_{term}(\boldsymbol{\xi}) = \begin{cases} \infty, & \boldsymbol{\xi} \in \mathcal{O}_0 \\ d(\boldsymbol{\xi}, \mathcal{S}_0), & \text{otherwise} \end{cases}\,. \tag{3.26}$$

To remain consistent with the set indexing convention introduced in Section 3.2, the sets in Eq. (3.26) are evaluated at time $t = 0$ (corresponds to the "terminal" reverse-time $\tau = T$). Moreover, a running cost can be similarly defined:

$$\tilde{l}(\boldsymbol{\xi}, \tau) = \begin{cases} -\alpha, & \boldsymbol{\xi} \in \mathcal{S}_{(T-\tau)} \text{ and } \boldsymbol{\xi} \notin \mathcal{O}_{(T-\tau)} \\ 0, & \text{otherwise} \end{cases}\,, \tag{3.27}$$

The value function for this optimal control problem can again be computed by forming a HJB PDE using now the dynamical system (3.24), terminal cost (3.26), and running cost (3.27). Mapping the resulting HJB PDE back to the original time variable t so as to not have to explicitly work in reverse-time τ, the HJB PDE can be shown to be given as:

$$\begin{aligned} \frac{\partial \tilde{J}^*(\boldsymbol{x}_0, t)}{\partial t} + \max_{\boldsymbol{u}} \left[-\tilde{l}(\boldsymbol{x}_0, t) + \nabla_{\boldsymbol{x}} \tilde{J}^* \cdot \tilde{f}_a(\boldsymbol{x}_0, \boldsymbol{u}, t) \right] &= 0 \\ \tilde{J}^*(\boldsymbol{x}_0, 0) = \tilde{l}_{term}(\boldsymbol{x}_0), & \end{aligned} \tag{3.28}$$

which, upon inserting the augmented dynamics (3.24), and the costs (3.26) and (3.27), yields:

$$\begin{aligned} \frac{\partial \tilde{J}^*(\boldsymbol{x}_0, t)}{\partial t} &= \begin{cases} -\left[\alpha + \nabla_{\boldsymbol{x}} J^* \cdot g_{\mathcal{S}}(\boldsymbol{x}_0, t)\right], & \boldsymbol{x}_0(t) \in \mathcal{S}_t \cap (\mathcal{O}_t)^c \\ -\left[\nabla_{\boldsymbol{x}} J^* \cdot g_{\mathcal{O}}(\boldsymbol{x}_0, t)\right], & \boldsymbol{x}_0(t) \in \mathcal{O}_t \\ -\max_{\boldsymbol{u}} \left[\nabla_{\boldsymbol{x}} J^* \cdot f(\boldsymbol{x}_0, \boldsymbol{u}, t)\right], & \text{Otherwise} \end{cases} \\ \tilde{J}^*(\boldsymbol{x}_0, 0) &= \begin{cases} \infty, & \boldsymbol{x}_0 \in \mathcal{O}_0 \\ d(\boldsymbol{x}_0, \mathcal{S}_0), & \text{otherwise} \end{cases}\,. \end{aligned} \tag{3.29}$$

In duality to the backward multi-reach setting where a terminal-value problem was obtained, in this case the value function $\tilde{J}^*(\boldsymbol{x}_0, t)$ is given as the solution to an initial value problem. Furthermore, analogous to the backward setting, the value function can be used to extract the FRT,

$$\bar{\mathcal{F}}(t, t_s, \mathbb{S}, \mathbb{O}) = \{\boldsymbol{x}_0 \mid \tilde{J}^*(\boldsymbol{x}_0, t) \leq -\alpha \cdot t_s\}, \tag{3.30}$$

and the time-to-leave maps,

$$\tilde{\mathcal{D}}(\boldsymbol{x}_0, t) = -\frac{\tilde{J}^*(\boldsymbol{x}_0, t)}{\alpha}, \qquad \forall(\boldsymbol{x}_0, t) \text{ s.t., } \tilde{J}^*(\boldsymbol{x}_0, t) < 0. \tag{3.31}$$

In comparison to backward reachability, the optimal controller obtained by working in reverse-time allows computing the start set as quickly as possible in a reverse-time setting. This is not as commonly useful since the universe runs forward in time, thus usually requiring an open-loop controller to execute the corresponding trajectories.

Remarks and Discussion

With the evolution equations derived for multi-time reachability, we now present several properties and differences when compared to classic reachability.

Ability to compute BRTs / FRTs with arbitrary start and end times

We note that we require a single solve of the PDE Eq. 3.22 / 3.29 to compute all possible BRTs / FRTs for a given dynamical system, target/start set, and obstacle set using Eq. (3.23) / (3.30). With classic reachability, one would instead a solve of a HJB PDE for every terminal time t_f, or start time t_s. While this benefit is inconsequential when dealing with a time-invariant system (since the backward and forward reachability tubes depend only on the time duration, $t_f - t$ and $t - t_s$, respectively), this property is very useful for analyzing dynamic systems. Many multi-time autonomy problems today indeed involve dynamic environments governed by PDEs (e.g., UAVs or AUVs affected by currents or winds), dynamic target / start sets, and / or dynamic obstacle sets [81]. Multi-time reachability thus has strong appeal.

While finishing this work, [78] posted a related framework that also adds a running cost to the HJB equation. Presently however, we derive and apply the exact governing equations for BRTs / FRTs and associated quantities, for the first time for systems operating in time-varying environments with dynamic obstacles and affected by other dynamic fields.

Field of Level Sets

In classical reachability, only the data on the zero level set of the value function is typically used, as this decomposes the space into the reachable and non-reachable regions which is usually what is of interest In multi-time reachability, a PDE of essentially identical complexity is solved, yet every value on the field provides useful physical reachability information. Specifically, in the backward reachability setting, the physical meaning of other level sets of the value function is given by Eq. (3.20). In this case, we reiterate, level sets with a negative value provide minimum times at which a target set can be reached from a given state, level sets with a positive value correspond to the closest distance to the target that can be reached for states that cannot reach the target, and finally an infinite value for J^* are states where it is unavoidable that an obstacle will be hit. The case of forward reachability has a similar physical interpretation for its different level sets.

Secondary quantities

While the value function obtained by classical backward reachability determines if one can reach the destination by time t_f given a starting time and position, the value function obtained using multi-time backward reachability determines *when* one can reach the target set ($\mathcal{D}(\boldsymbol{x}_0, t)$). When evaluated at a given position $\boldsymbol{x}_0$, one can infer the map between the starting time of the trajectory to the duration it takes to reach the target. Similarly, the forward value function determines *when* to start a journey to be able to reach an arbitrary point from the starting set ($\tilde{\mathcal{D}}(\boldsymbol{x}_0, t)$). This information can be extremely valuable for time varying systems where the time to reach the destination can vary drastically with the time at which the trajectory starts. We refer to the resulting plots as duration vs. arrival time, and start time vs. duration.

Optimal Controller

The closed loop controller under the value function for multi-time reachability (Sect. 3.3) has varied desirable properties including obstacle avoidance, time optimality if the target is reachable, and distance to target minimized if not. This drastically augments classical reachability that only minimizes the terminal signed distance from the target set at time T and does not provide time optimality when the state is not on the zero level set. We will make use of this property for operating under forecasts in Chapter 5.

3.4 Computation and Numerical Schemes

In Section 3.3, the PDEs (3.21)-(3.22) for the value function of backward reachability and PDEs (3.28)-(3.29) for the value function of forward reachability are HJB PDEs. These PDEs, including the existence, uniqueness, and properties of their viscosity solutions, have been extensively studied in recent years due to their broad applicability [101, 123, 36].

Several options exist for numerically computing a viscosity solution to a given HJB PDE, ranging from Finite Volume methods to high-order discontinuous Galerkin methods [86, 46, 96, 84, 81]. Presently, we used the method of lines, with high-order finite difference methods for the temporal and spatial discretization, on structured, uniform, rectangular meshes [32, 11]. Our software was built on top of an open-source HJ equation solver, hj_reachability [119], built on JAX [16]. To numerically compute the viscosity solution, the Local Lax-Friedrichs scheme was used [123, 32, 11].

Since the above scheme is fully explicit, the computational cost is $\mathcal{O}(N_x N_t)$ where N_x and N_t are the total number of spatial grid points and of timesteps, respectively. The explicit scheme allows direct parallelization of the computation across grid points. The cost of our method is thus exactly of the same order as that of classical reachability while providing much richer information about the system.

Finally, we discuss two numerical implementation details: (1) Handling the infinite condition in the terminal cost for when a state terminates in an obstacle set, i.e. Eqs. (3.11) and

(3.26), is straightforward. Since the numerical solve is inherently restricted to a closed and bounded state space, we simply set the terminal cost value when in an obstacle set to an arbitrary constant greater than the largest possible signed distance value in the domain at terminal time. When processing the final numerically computed value function, we simply mask off all values that equal this arbitrary constant as there are no controls that allow the system to avoid the obstacle. (2) The constant α in Eq. (3.12) can be chosen to decrease round off errors associated to the numerical solve. By setting α to the order of the characteristic time of the problem, the contours of interest will be $\mathcal{O}(1)$.

3.5 Numerical Results

We illustrate our theory and schemes in three numerical cases. In the first, we verify our method by applying it to a system with analytical reachability tubes. In the second and third cases, we consider more complex systems and demonstrate the various capabilities of our approach.

Analytical Moving Target and Obstacle

Problem Setup

This case uses the example of [39, Sec 5.1]. The 2D dynamical system consists of a vehicle moving with a constant speed in any heading, $\dot{\boldsymbol{x}} = u_{veh}\hat{\boldsymbol{h}}$, where the state $\boldsymbol{x} = [x, y]$ is the vehicle position, $\hat{\boldsymbol{h}}$ the unit heading, and $u_{veh} = 0.5$ the constant speed of the vehicle. The dynamic target set is a square of side length 0.4 units centered at $[0, 0.75]$ at $t = 0$ traveling with a velocity of $[0, -1.5]$ units. The obstacle is a square of side length 0.2 initially centered at $[0, 0]$ and traveling at a velocity of $[0, -1]$ units. Our goal is to compute backward reachability tubes and compare them to the analytical solution we obtained geometrically using [39, Sec 5.1.1]. The augmented dynamics is straightforward here: we modify the system dynamics such that when the agent is in the target (/obstacle), it moves exactly with the known velocity of the target (/obstacle).

Results

Fig. 3.1 shows the analytical, overlaid on the numerical, backward reachability tubes using our multi-time reachability. The analytical and numerical tubes (left half of the domain) are effectively identical. When compared to the results in [39] that compute the BRTs for $t_f = 0.5$, our approach accurately computes tubes for all terminal times t_f and in a single PDE simulation.

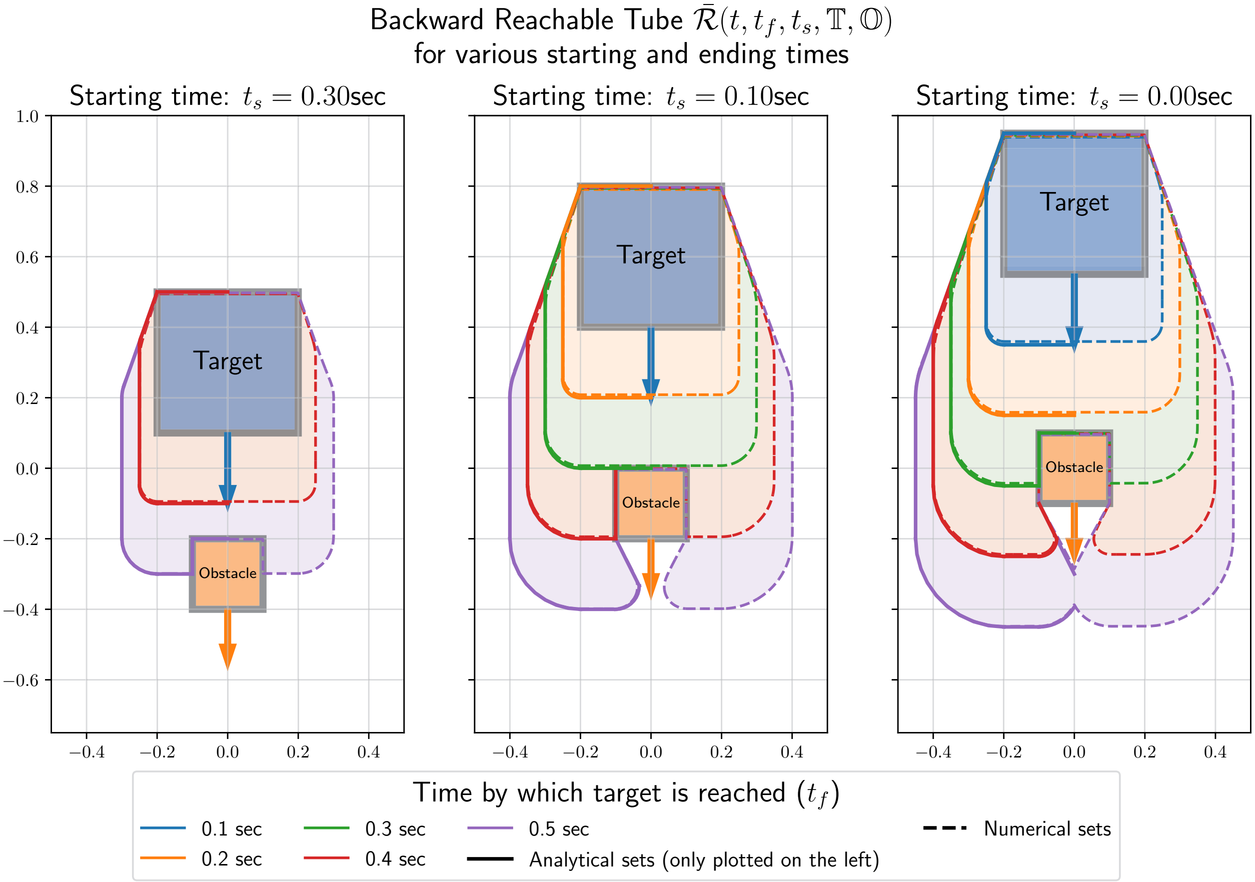


Figure 3.1: Backward reachability tubes for various start and end times. Analytically computed tubes are superimposed in the left half on top of numerically computed tubes (dashed line).

Dynamic Dubin's car

Problem Setup

We now consider a more complex $3D$ dynamical system often referred to as the Dubin's car. The state space of the car is given by its position and orientation: $\boldsymbol{x} = [x, y, \theta]$. The only control is that of the steering rate $\dot{\theta} = u_\alpha$. The dynamics is given by

$$\begin{bmatrix}\dot{x} & \dot{y} & \dot{\theta}\end{bmatrix}^T = \begin{bmatrix}v\cos\theta & v\sin\theta & u_\alpha\end{bmatrix}^T,$$

where $v = 1$ is the velocity of the car. We additionally constrain the steering rate to satisfy $|u_\alpha| < \frac{\pi}{3}$ units.

We add a moving target and moving obstacle with velocities $[0.4, 0]$ and $[0.2, 0]$, respectively. The positions of the target and the obstacle at various times are shown in Fig. 3.2

(the target is blue and the obstacle orange). Our goal is to compute backward reachable tubes as well as time-to-reach maps.

Results

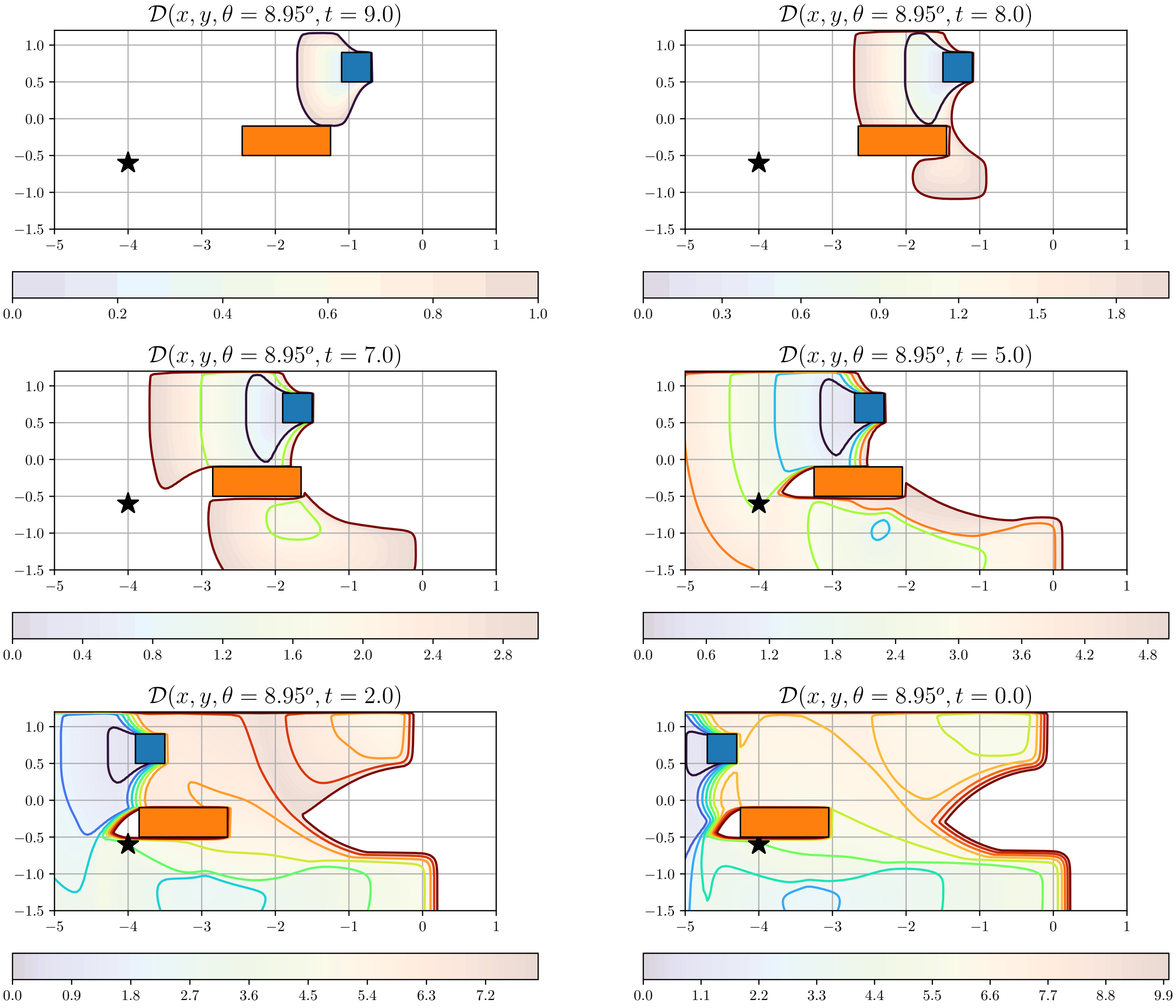
Since the state space is $3D$, the value function now lives in a $3D$ space. To get an intuition of what kind of information can be gained from the value function, we consider the duration map at a slice $\mathcal{D}(x, y, \theta = 8.95^o, t)$ (Fig. 3.2). At a given (x, y) and time t, the duration map returns the amount of time needed to reach the target when starting at that position and time, and being initially aligned at the given angle (i.e. $\theta = 8.95^o$). As expected, states to the left of the target can reach the target set more easily given their initial orientation (states to the right need to turn around to reach the target set). In addition, note the triangular region that forms on the left of the obstacle, corresponding to regions where the car cannot avoid the obstacle.

Fig. 3.2 highlights a key feature of the power of multi-time reachability. Consider the following question: given a start point (x_s, y_s, θ_s) (marked with a star), what is the minimum duration to the target as a function of the start time (t_s)? This information is readily available using the duration map, $\mathcal{D}(x_s, y_s, \theta_s, t_s)$, as seen in Fig. 3.3. As different contours of the duration map reach the start point, we see vastly different gradients of the value function – implying changing optimal control strategies based on when the car starts.

Suppose we pose another question: Given a start position (x_s, y_s) what is the optimal starting angle at different starting times to minimize the duration to the target set? We can use the following expression to compute the duration under optimal θ as a function of the starting time t_s: $d(t_s) = \min_\theta \mathcal{D}(x_s, y_s, \theta, t_s)$. The corresponding optimal θ is given by $\arg\min_\theta \mathcal{D}(x_s, y_s, \theta, t_s)$. Fig. 3.4 shows, for a start position $(x_s, y_s) = (-3, -0.7)$, this minimum duration and optimal θ as a function of start time. These plots, trivially generated using the multi-time reachability value function, contain information that is invaluable in deciding when and how to start the journey from a given start point.

Different regimes of the optimal solution can be noted, as the car decides to start with drastically different angles based on the starting time. To see why this is occurs, we plot the trajectories (Fig. 3.5) of all the points of interest marked with black dots in Figs. 3.3 and 3.4. The plot on the left traces out optimal trajectories when starting at the star (for various start times marked in Fig. 3.3) while constraining the initial heading the vehicle must start with. We find that the duration initially falls with start time as cutting across the front of the obstacle is difficult for the agent given this constrained initial angle. The plot on the right corresponds to the trajectories starting at optimal θ at various times (Fig. 3.4). We see that early on, the optimal agent cuts across the front of the obstacle and heads straight for the target. If it starts later, it has to go out of its way to drive around the front of the obstacle. However, if it waits long enough, the agent can reach the target by traversing behind the obstacle.

This case shows how we compute optimal trajectories for a variety of starting positions, angles, and start/terminal times. We further note that these can all be efficiently computed



$\mathcal{D}(x, y, \theta = 8.95^o, t = 9.0)$

$\mathcal{D}(x, y, \theta = 8.95^o, t = 8.0)$

$\mathcal{D}(x, y, \theta = 8.95^o, t = 7.0)$

$\mathcal{D}(x, y, \theta = 8.95^o, t = 5.0)$

$\mathcal{D}(x, y, \theta = 8.95^o, t = 2.0)$

$\mathcal{D}(x, y, \theta = 8.95^o, t = 0.0)$

Figure 3.2: Slice of the duration map at $\theta = 8.95^o$. This map physically represents the time needed to reach the target set based on the initial position of the car. The point marked with the star denotes an arbitrary start point that is used in subsequent analysis (Fig. 3.5).

using information from a *single* multi-time reachability PDE solve – something not possible with other approaches.

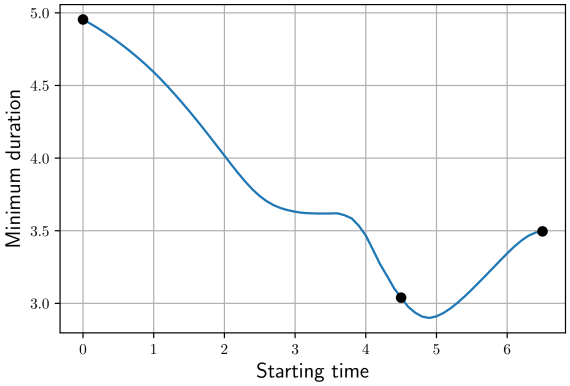


Figure 3.3: Duration of the journey to the target set as a function of the starting time. Since this is a time varying system, the duration is not constant. Due to the moving obstacle, the duration first decreases and increases. Times of interest are marked with a black dot.

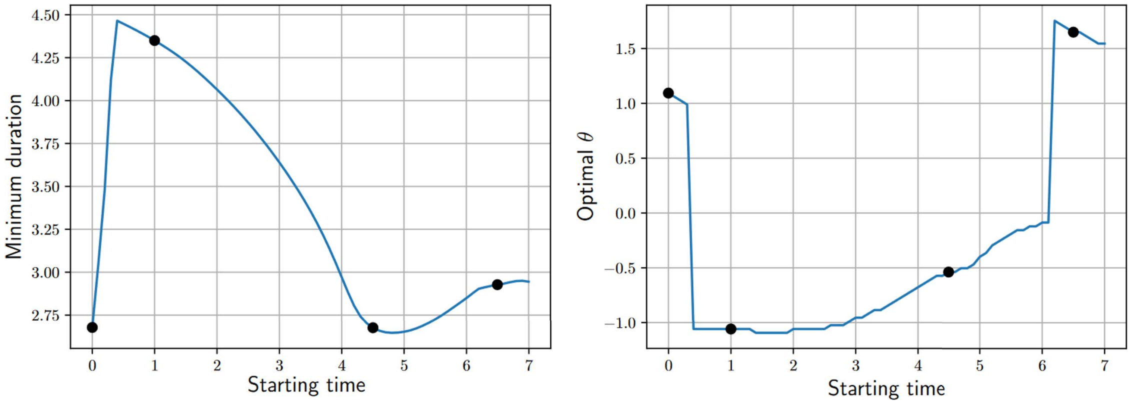


Figure 3.4: (Left) Minimum duration to reach the target set, given the starting position $(x_s, y_s) = (-3, -0.7)$, as a function of starting time (t_s). (Right) Initial angle θ_s under which this minimum duration can be achieved for each corresponding starting time. Points of interest are marked with black dots.

AUV in a Bottom Gravity Current Flow Field

Problem Setup

Finally, we showcase results when a time-varying dynamic environment affects the system and optimal control. We consider an AUV in $2D$ with state variables $\boldsymbol{x} = (x, z)$ where x and z are position and depth of the vehicle. We denote the dynamics by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} F_x \cos(u_\theta) + V_x(\boldsymbol{x}, t) & F_y \sin(u_\theta) + V_y(\boldsymbol{x}, t) \end{bmatrix}^T,$$

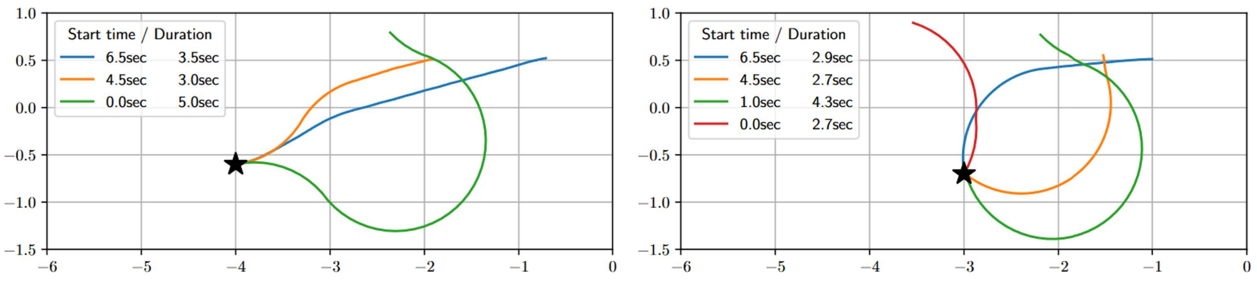


Figure 3.5: Trajectories for various start points, start times and start angles. (Left) Trajectories starting at fixed start point but with the optimal angle to reach the destination as early as possible. (Right) Trajectories start at a constrained position and angle. Both set of trajectories start at a user defined time and end at different times before the final time horizon T

where u_θ is the sole control and $\boldsymbol{V} = [V_x, V_y]$ is the dynamic background ocean flow field that advects the AUV around. The background flow is that of a non-hydrostatic bottom gravity current simulated using our Finite Volume ocean modeling software [138]. This flow involves heavy salt water flowing down an incline and creating eddies due to Kevin-Helmholtz instabilities, as visualized in Fig. 3.6 [79].

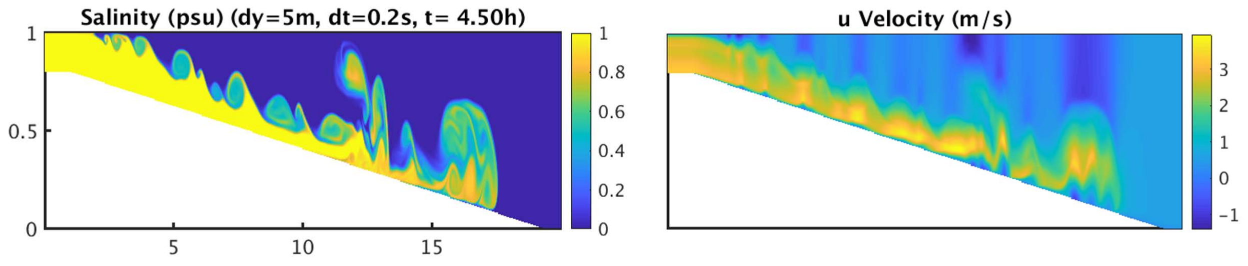


Figure 3.6: Salinity field (left) and x-component of the velocity field in m/s (right) for the the bottom gravity flow. The units of x and y are in km.

Results

We consider a domain of interest at the bottom of the incline. As expected, we see that the duration map is uniform at the time before the current reaches the bottom, whereas it is non-uniform and time-varying at the time frame when the current and its billows and waves arrive (Fig. 3.7). Applications of multi-time reachability to dynamic ocean environments can be found in [141].

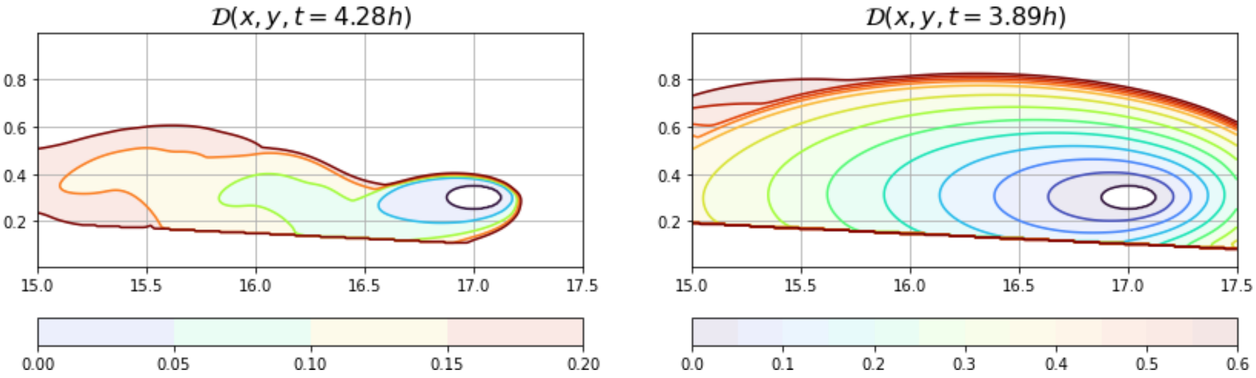


Figure 3.7: Duration maps under the bottom gravity flow after (left) and before (right) the current enters the domain. The units of x and y are in km, and the unit for time is hours.

3.6 Conclusions

In this chapter, we obtained the governing equations for reachability over multiple start and terminal times all at once for systems operating in time-varying environments with dynamic obstacles and any other relevant dynamic fields. We analytically verified the results for a moving target and obstacle problem, then applied multi-time reachability to an extended dynamic Dubin's car, and finally showcased the method in a bottom gravity current system, an example of an underactuated system operating in dynamic flows. The results highlight the novel capabilities of exact multi-time reachability in dynamic environments. In Chapter 5 these will allow us to handle forecast errors by replanning frequently at a low computational cost.

Chapter 4

Optimal Control for Seaweed Growth

This chapter is based on the paper "Maximizing Seaweed Growth on Autonomous Farms: A Dynamic Programming Approach for Underactuated Systems Navigating on Uncertain Ocean Currents" [60], written in collaboration with Matthias Killer, Hanna Krasowski, Manan Doshi, Pierre F.J. Lermusiaux, and Claire Tomlin.

4.1 Introduction

In the preceding chapter, we explored the use of dynamic programming to compute a value function for reachability and time-optimal navigation from a starting point to a target. The principles applied in Chapter 3, where we utilized dynamic programming to solve an optimal control problem, aren't limited to navigation. In fact, these principles apply to arbitrary cost functions specified as a combination of a running cost and a terminal cost. In this chapter, we apply the same principles to maximize seaweed growth for underactuated robotic systems operating in dynamic flows.

Only a few research papers in the control field focus on maximizing seaweed growth. For example, Bhabra et al. [11] maximize seaweed harvesting using autonomous vessels in varied settings. A key limitation is that the seaweed is grown not on the system itself, but in discrete seaweed farms, with the objective of optimizing the trajectory and sequence of harvesting rather than continuously maximizing growth. Bhabra et al. [11] use a 3D HJ reachability framework in which the harvesting state is augmented into the third dimension. To find the path with maximum growth, they run forward reachability in the state space for seaweed in 3D. However, a challenge with dynamic programming is that the required scales exponentially with the dimension of the state space n [5], hence this method of state-augmentation is computationally demanding due to the added dimension in the HJ calculation.

In this chapter, we present our method of maximizing seaweed growth using a running cost problem solved with Dynamic Programming (DP) in the system's 2D spatial state (Sec.

4.2). This approach considerably reduces computational complexity compared to the 3D HJ reachability method used by Bhabra et al. [11].

Additionally, the resulting value function can be used as a feedback policy to obtain the growth-optimal control for all states and times. Therefore, it can provide control inputs to multiple farms and it can be used to mitigate forecast errors by using it as a closed-loop control policy [141] which is equivalent to replanning at every time step. We will elaborate more on this in chapters 5 and 6.

4.2 Method

Problem Setting

We consider a floating seaweed farm with the system dynamics of the spatial state $\boldsymbol{x} \in \mathbb{R}^n$ influenced both by the control input $\boldsymbol{u}$ and the surrounding ocean currents $v(\boldsymbol{x}, t)$ as defined in Sec. 2.1. In our example, we use $n = 2$ for a surface vessel on the ocean. In addition to the spatial state, the farm has a seaweed mass m which evolves according an exponential growth ODE:

$$\dot{m} = m \cdot \Psi(\boldsymbol{x}, t), \quad t \in [0, T] \tag{4.1}$$

where Ψ is the growth factor per time unit, e.g. $20\frac{\%}{day}$, which depends on nutrients, incoming solar radiation, and water temperature at the spatial state $\boldsymbol{x}$ and time t.

The objective of the seaweed farm starting from $\boldsymbol{x}_0$ at t_0 with a seaweed mass $m(t_0)$ is then to maximize the mass of the seaweed at the final time T. This implies optimizing the growth over its spatial trajectory $\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t_0,\boldsymbol{x}_0}$.

$$\max_{\boldsymbol{u}(\cdot)} m(T) = m(t_0) + \max_{\boldsymbol{u}(\cdot)} \int_{t_0}^{T} m(s) \cdot \underbrace{\Psi(\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t_0,\boldsymbol{x}_0}(s), s)}_{\text{growth factor}} \, ds \tag{4.2}$$

We maximize the final seaweed mass when the ocean currents are known by formulating a running cost objective that can be efficiently solved with Dynamic Programming.

Maximizing Seaweed Mass With Known Dynamics

We use continuous-time optimal control where the value function $J(\boldsymbol{x}, \boldsymbol{u}(\cdot), t)$ of a trajectory $\boldsymbol{\xi}$ is based on a state and time-dependent reward l and a terminal reward l_{term}:

$$J(\boldsymbol{x}, \boldsymbol{u}(\cdot), t) = \int_{t}^{T} l(\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(s), s) ds + l_{term}(\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(T), T).$$

Instead of maximizing this objective with graph-search methods or non-linear programming which require discretization in space and time, we use DP. Let $J^*(\boldsymbol{x}, t) = \max_{\boldsymbol{u}(\cdot)} J(\boldsymbol{x}, \boldsymbol{u}(\cdot), t)$ be the optimal value function; we can then derive the corresponding Hamilton-Jacobi PDE [5]:

$$-\frac{\partial J^*(\boldsymbol{x},t)}{\partial t} = \max_{\boldsymbol{u}} \left[\nabla_{\boldsymbol{x}} J^*(\boldsymbol{x},t) \cdot f(\boldsymbol{x},\boldsymbol{u},t) + l(\boldsymbol{x},t)\right] \tag{4.3}$$

$$J^*(\boldsymbol{x},T) = l_{term}(\boldsymbol{x},T). \tag{4.4}$$

We can then numerically compute $J^*(\boldsymbol{x},t)$ on a spatial mesh by integrating the PDE backwards in time [97]. Next, we need to define the reward l and the terminal reward l_{term} to maximize $m(T)$. One approach is to model the seaweed farm with an augmented state $x_{aug} = (\boldsymbol{x}, m)^\top \in \mathbb{R}^3$. If we set $l_{term} = 0$ and define the reward as $l = m \cdot \Psi(\boldsymbol{x},t)$, the value function is our objective (Eq. (4.2)).

However, the computational complexity of solving for J^* scales exponentially with the state dimension. Therefore, we want a reward l that does not depend on m as augmented state. For that, we introduce the variable $\eta = \ln(m)$ with the new dynamics $\dot{\eta} = \frac{\dot{m}}{m} = \Psi(\boldsymbol{x},t)$. As $\eta(m)$ is strictly increasing in m, the control $\boldsymbol{u}^*(\cdot)$ that maximizes $\eta(T)$ is equivalent to $\boldsymbol{u}^*(\cdot)$ maximizing $m(T)$. We can then reformulate Eq. (4.2) to $\eta(T)$:

$$\max_{\boldsymbol{u}(\cdot)} \eta(T) = \eta(t_0) + \max_{\boldsymbol{u}(\cdot)} \int_{t_0}^{T} \Psi(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s), s)\, ds. \tag{4.5}$$

By setting the reward to $l = \Psi(\boldsymbol{x},t)$ the optimal value function captures this optimization without requiring m:

$$J^*(\boldsymbol{x},t) = \max_{\boldsymbol{u}(\cdot)} \int_{t}^{T} \Psi(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s), s)\, ds. \tag{4.6}$$

We can then solve the HJ PDE for $J^*(\boldsymbol{x},t)$ in $\boldsymbol{x}$ and obtain the optimal control $\boldsymbol{u}^*(\cdot)$ and trajectory $\boldsymbol{\xi}_{t_0,\boldsymbol{x}_0}^{\boldsymbol{u}^*(\cdot)}$ that maximize $m(T)$ at lower computational cost. This formulation can be applied more generally to optimize the state of exponential growth or decay ODEs. Note that we can convert the value of $J^*(\boldsymbol{x}_0,t_0)$ to the final seaweed mass of the optimal trajectory starting at $\boldsymbol{x}_0, t_0$ with $m(t_0)$:

$$m(T) = m(t_0) \cdot e^{\int_{t_0}^{T} \Psi(\boldsymbol{\xi}_{t_0,\boldsymbol{x}_0}^{\boldsymbol{u}^*(\cdot)}(s),s)\, ds} = m(t_0) \cdot e^{J^*(\boldsymbol{x}_0,t_0)}.$$

4.3 Experiments

To offer further intuition and facilitate understanding, we present a simple toy example [1] of a 2D system maximizing seaweed growth in a simple flow field and leave the realistic examples for Chapter 6.

In Fig. 4.1, a *current highway* is depicted [59]. From the y-values of [4m, 6m], a current flows from left to right with a strength of $0.2\frac{m}{s}$ (red colored), everywhere else the current velocity is zero. Within this flow, a rectangular seaweed growth region is modeled, while outside of this area, no growth occurs.

[1] This section is based on the work with Matthias Killer [59].

The 2D system is initiated at the position [1m, 3m] at t_0, with the objective of maximizing seaweed growth over a given time period $[t_0, T]$. Two settings are considered: 1) an overactuated setting, and 2) an underactuated setting. We can then use our method to calculate the trajectory that maximizes the seaweed mass at the final time.

In the overactuated scenario, the system is equipped with a propulsion strength of $0.3\frac{m}{s}$. This allows it to counteract the currents, thus being overactuated. The optimal growth trajectory (depicted in green) in this case quickly brings the system into the growth region, where it remains to maximize seaweed growth. The farm attempts to enter the highway, and while on the highway but outside the growth region, the actuation points in the direction of the flow to quickly reach the growth area. Upon arrival, the actuation opposes the current flow, as the goal is to remain within the growth field.

In the underactuated scenario, the trajectory (depicted in black) also brings the system rapidly toward the growth region. However, due to insufficient actuation to stay within the region, it is unable to simply counteract the currents for maximum growth. Instead, the system periodically circles in and out of the growth region. So it periodically exits the current flow and retreats slightly before re-entering the current flow and growth region.

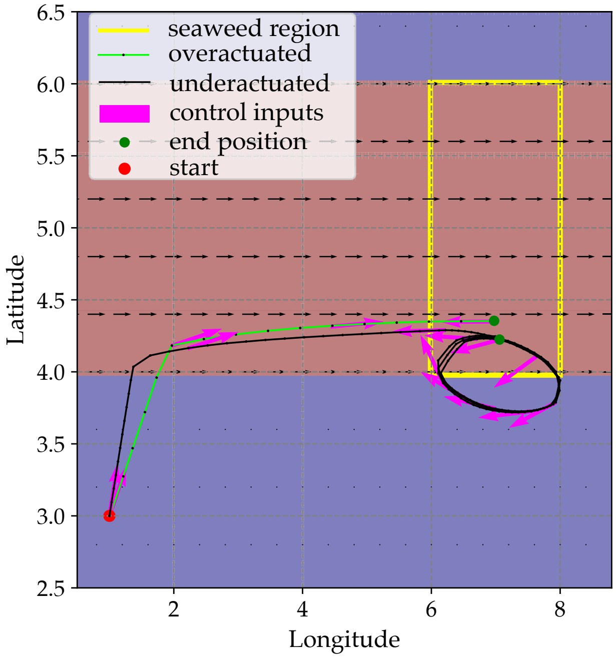


Figure 4.1: Toy example for maximizing seaweed growth highlighting the difference between the underactuation vs. overactuation setting.

4.4 Conclusion

In this chapter, we addressed the challenge of maximizing seaweed growth in underactuated autonomous ocean farms by proposing a growth-maximizing DP approach. Instead of solving this problem in 3D using state augmentation, our method solves a running-cost problem in the 2D spatial state of the system. This reduces computational complexity and also generates a value function, which can be used as a feedback policy for growth-optimal control across all states and times, as we will see in Chapter 6.

Part III

Handling Forecast Uncertainty

Chapter 5

Navigating With Forecasted Flows

This chapter is based on the paper "Navigating underactuated agents by hitchhiking forecast flows" [141], written in collaboration with Manan Doshi, Pierre F.J. Lermusiaux, and Claire Tomlin.

5.1 Introduction

In Part II we have developed the theory and methods to optimally operate underactuated robotic systems in flows. In this chapter, we discuss how to navigate when the true flows are not known and only forecasts are available. We present how we can achieve reliable navigation by using the value function obtained from Multi-Time HJ Reachability (Chapter 3) as a closed-loop control policy that is equivalent to fully replanning at every step.

There is a rich literature on planning time- and energy-optimal paths in flows both in the oceans [85, 128, 72, 152, 56, 136, 51, 64, 112, 90, 65, 130, 107, 49, 62, 17, 43, 25, 133, 117] and the air [118, 19, 102, 8]. We have already discussed various methods when the flows are known in Chapters 3 and 4. As the wind and ocean forecasts are never perfect, different paradigms and their combinations have been explored for navigation when the true flow is not known. When we have access to a *realistic distribution of the flows* we can use probabilistic methods to optimize the expected energy or travel time of a policy using stochastic reachability [129] or Markov Decision Processes (MDP) [65, 118]. Previous work also explored risk-optimal path planning in this stochastic setting [130, 107, 49] including estimating the confidence in the forecast using interpolation variance [49]. Unfortunately, for ocean surface currents, only daily deterministic forecasts are available from the leading ocean forecasting providers HYCOM [20] and EU Copernicus Marine [35]. There are ocean dynamics models with stochastic forcing to compute accurate probabilistic forecasts [75, 73, 74], however, these are too computationally expensive for real-time operations. A heuristic approach is then to assume Gaussian noise [65] or to increase the prediction uncertainty around high-velocity currents [49], which does not capture the complexity of the forecast errors.

Another paradigm is to plan on the deterministic forecast flows and follow the planned path with tracking algorithms that compensate for drifts and can guarantee a tracking error bound [117]. While this is important in tight spaces around obstacles, minimizing tracking error does not necessarily lead to time and energy optimal paths. Another family of approaches follows the Model Predictive Control (MPC) paradigm that uses frequent replanning from the positions of the agent to handle the dynamics error [51, 111]. Our approach will build on this paradigm without the need for expensive replanning operations at every step. Lastly, there are robust control methods in which we assume a bound on the forecast error to derive a path and controls that reach the target despite bounded adversarial disturbances. This can be done using reach-avoid HJ reachability formulations [39] or approximations [51]. However, for underactuated agents, often there exists no robust control for realistic forecast error bounds (see Sec. 2.2).

Recent work has explored the application of deep Reinforcement Learning (RL) to navigate in flows [43, 25, 17]. The controller trained by Gunnarson et. al. on vortical flows using only local current information navigates successfully in these flows but, unsurprisingly, fails in other flow structures [43]. The Loon team trained an RL agent for station-keeping of balloons with forecasts as inputs and it performed well in long-duration real-world experiments after training on an immense distribution of flows [8].

In this chapter, we focus on the problem of *reliable* navigation of underactuated agents leveraging flows in the realistic setting when regular deterministic forecasts are available. We define reliability empirically as *the success rate of a controller in navigating from a start point to a target region over a set of start-target missions*, as developed in Section 5.1. In this chapter, we make three core contributions: (1) We propose a control approach enabling full time-horizon replanning at every time step with a single computation. For that, we build on the recent Multi-Time HJ Reachability formulation [31] to compute a time-optimal value function (Time-to-reach) on the latest forecast and use it as a policy π for closed-loop control. This can be thought of as full-time horizon Model Predictive Control at every step; (2) We are the first to evaluate and compare the reliability of closed-loop control schemes for underactuated agents in ocean flows in the setting of daily forecasts with realistic forecast error. We evaluate performance across a large set of multi-day start-to-target missions distributed spatially across the Gulf of Mexico and temporally over four months using HYCOM and Copernicus Ocean Forecasts [35, 20] We compare several methods on this dataset across multiple metrics and find that our control architecture significantly outperforms other methods in terms of *reliability*; (3) We quantify how the reliability of various control methods is affected by the forecast error.

Problem Setting

The objective of the agent is to navigate *reliably* from a start state $\boldsymbol{x}_0$ to a target region $\mathcal{T} \in \mathbb{R}^n$ while being underactuated $max_u||g(\boldsymbol{x}(s), \boldsymbol{u}(s), s)||_2 \ll ||v(\boldsymbol{x}(s), s)||_2$ most of the time. During operation, the agent is given a forecast of the flow $\hat{v}(\boldsymbol{x}, s)$ which differs from the true flow $v(\boldsymbol{x}, s)$ by the stochastic forecast error $\delta(\boldsymbol{x}, s; \omega)$ where ω is a random variable. The error field δ can be characterized based on different metrics such as RMSE of the velocities or vector correlation [94]. The agent receives a new forecast at regular intervals (typically daily) that can be used to improve performance.

Our goal is reliable navigation of underactuated agents in realistic complex flows occurring in nature. The strongest notion of reliability is robustness to a bounded disturbance, which guarantees reaching the target despite a worst-case forecast error δ. However, proving robustness is not possible in our setting where the agent is significantly underactuated and the average forecast error is larger than the actuation. Nevertheless, we compare against a robust control baseline in Sec. 5.3. A weaker notion of reliability is a probabilistic bound, i.e. the agent reaches the target with high probability. Probabilistic bounds could be established by making strong assumptions on the distribution of the forecast error fields δ and using a simple flow field; this would render the results less meaningful for the realistic settings we consider.

For these reason we define *reliability* empirically as the success rate of a controller navigating from a start point to a target $\mathcal{T}$ over a set of start-target missions $\mathbb{M}$ in realistic flows. If the agent reaches the target $\mathcal{T}$ within a maximum allowed time T_{max} the mission is successful, otherwise it failed. In our experiments we sample missions $\mathbb{M}$ over a large spatial region and over a period of four months.

5.2 Using the Value Function for Frequent Replanning

In this section, we first outline the motivation behind our method and then detail our closed-loop control strategy.

Motivation

In our challenging setting, when only deterministic forecasts are available and the forecast error is larger than the actuation, we can neither use probabilistic methods without making unrealistic assumptions about the error distribution δ, nor can we apply robust control as the average error δ is larger than the actuation of the agent. How then can we achieve reliable navigation in this setting?

Our approach builds on the Model Predictive Control paradigm of regular replanning with deterministic dynamics to compensate for imperfect knowledge of the dynamics and achieve reliable navigation. Intuitively, the higher the frequency of replanning, the more we can adapt the control to the dynamics experienced. While we could use any planning algorithm for non-linear time-varying dynamics, as mentioned in the related work, HJ reachability is state-of-the-art, as it guarantees finding the optimal solution and can even handle time-varying obstacles and targets [85, 39, 5]. While we can obtain time-optimal trajectories from both classic forward and backward HJ reachability, the value function of backwards reachability is more useful. The backwards reachability objective is to minimize the distance of the agent to the target set at a terminal time. A key insight is that the value function of this objective can be used for closed-loop control as for every state and time in the domain we can extract the optimal control that minimizes this objective. This provides a notion of replanning at every step even when the deterministic dynamics are not accurate. However, there is a problem with directly applying classic backwards HJ reachability: the value function minimized is the distance to the target at a *fixed terminal time*. This poses the problem of which terminal time to choose to calculate the value function? If we choose it too distant in the future, the system will "loiter", not making progress towards the goal. If we choose it too close to the current time, it might be impossible to reach the target, and the agent will minimize the distance to the target in the short term, potentially at the cost of long-term progress. We can compute the earliest possible arrival time, using forward reachability, and then compute backwards reachability from that time to get our value function for closed-loop control (a baseline in 5.3). A more elegant approach is to use the Multi-Time Reachability formulation (Chapter 3), which requires only *one* backwards computation and produces a value function that yields the time-optimal control everywhere, not just at the zero level-set, as the classic reachability value function. We found that using time-optimal control is a good proxy for reliability in our setting.

Multi-Time Reachability for Closed-Loop Control

In the following we will first reiterate the relevant aspects of the multi-time reachability technique for completeness and for simplicity we leave out time-varying obstacles and targets as in the full derivation in [31].

The Multi-time reachability formulation (Chapter 3) uses dynamic programming to derive a controller that (a) if possible, will get the system to the target in the minimum time and (b) if not, will get as close to the target as possible. To achieve this behavior, we define the following cost function J:

$$J(\boldsymbol{x}_0, \boldsymbol{u}(\cdot), t) = \underbrace{d(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(T), \mathcal{T})}_{\substack{\text{Terminal distance} \\ \text{from target set}}} - \underbrace{\int_t^T \mathbb{I}_{\mathcal{T}} \left(\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s) \right) ds}_{\text{Time spent in target set}} \tag{5.1}$$

where $d(x, \mathcal{T})$ is a distance metric of a point x to the target set $\mathcal{T}$, $\boldsymbol{\xi}_{t,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(s)$ is the position of the agent at time s, when starting at $\boldsymbol{x}$ at time t with the spatial dynamics given in Eq. (2.1). $\mathbb{I}_{\mathcal{T}}(x)$ is the identity function that is 1 when the state is in the target $x \in \mathcal{T}$ and 0 otherwise. The consequence of this cost function is that if the agent can reach the destination, minimizing the cost implies reaching as quickly as possible. If the agent cannot reach the destination, the optimal control will attempt to reduce the terminal distance to the target.

Note that for underactuated systems we want to consider the dynamics of the agent only until it reaches the target. Within the target, we switch off the dynamics to reward the agent for staying in the target.

Given this cost function, we use the principle of dynamic programming to derive an HJ PDE whose viscosity solution is [31]:

$$\begin{aligned}\frac{\partial J^*(\boldsymbol{x}_0,t)}{\partial t} &= \begin{cases}1 & \boldsymbol{x}_0(t) \in \mathcal{T} \\ -\min_{\boldsymbol{u}} \left[\nabla_{\boldsymbol{x}} J^* \cdot f(\boldsymbol{x}_0,\boldsymbol{u},t)\right] & \text{otherwise}\end{cases} \\ J^*(\boldsymbol{x}_0,T) &= d(\boldsymbol{x}_0,\mathcal{T}) \end{aligned} \tag{5.2}$$

Since the value of J^* contains information about the minimum time it takes to reach the destination, we can extract an informative time-to-reach map $\mathcal{D}^*$ from it:

$$\mathcal{D}^*(\boldsymbol{x}_0,t) = T + J^*(\boldsymbol{x}_0,t) - t, \qquad \forall(\boldsymbol{x}_0,t) \text{ s.t., } J^*(\boldsymbol{x}_0,t) \leq 0 \tag{5.3}$$

If the target can be reached from $\boldsymbol{x}$ starting at t (implied by $J^*(\boldsymbol{x}_0,t) \leq 0$), then $\mathcal{D}^*(\boldsymbol{x}_0,t)$ is the minimum duration required to reach $\mathcal{T}$. This means that inside the target $\mathcal{D}^*(\boldsymbol{x}_0,t) = 0 \; \forall t, \boldsymbol{x} \in \mathcal{T}$. The optimal control $\boldsymbol{u}^*(\boldsymbol{x},t)$ is then the value that minimizes the Hamiltonian.

$$\begin{aligned}\boldsymbol{u}^*(\boldsymbol{x},t) &= \arg\min_{\boldsymbol{u}\in\mathbb{U}} f(\boldsymbol{x},\boldsymbol{u},t) \cdot \nabla_x J^*(\boldsymbol{x}_0,t) && (5.4)\\ &= \arg\min_{\boldsymbol{u}\in\mathbb{U}} g(\boldsymbol{x},\boldsymbol{u},t) \cdot \nabla_x J^*(\boldsymbol{x}_0,t) && (5.5)\end{aligned}$$

Where 5.5 follows from 5.4 because $v(\boldsymbol{x},t)$ does not depend on $\boldsymbol{u}$. Note that $\nabla_x \mathcal{D}^*(\boldsymbol{x}_0,t) = \nabla_x J^*(\boldsymbol{x}_0,t)$ which means that we can use $\mathcal{D}^*(\boldsymbol{x}_0,t)$ and $J^*(\boldsymbol{x}_0,t)$ interchangeably to obtain the optimal control. This formulation solves the two key issues with classic backwards reachability as described above: a) we do not need to fix an arrival time *a priori*, instead we can run backwards Multi-Time Reachability from a large maximum time backwards until the current time t; and b) the control obtained from this value function provides the time-optimal control at all states, making it more useful for reliable navigation.

The key insight here is that the value function allows us to extract the time-optimal control for each state $\boldsymbol{x}$ at each time t thus we can use it as a closed-loop control policy π which enables frequent replanning at low computational cost.

$$\begin{aligned}\pi(\boldsymbol{x},t) &= \arg\max_{\boldsymbol{u}\in\mathbb{U}} f(\boldsymbol{x},\boldsymbol{u},t) \cdot \nabla_x J^*(\boldsymbol{x},t) && (5.6)\\ &= \arg\max_{\boldsymbol{u}\in\mathbb{U}} g(\boldsymbol{x},\boldsymbol{u},t) \cdot \nabla_x J^*(\boldsymbol{x},t) && (5.7)\end{aligned}$$

This use of the reachability value function differs significantly from the ways in which it has been used in the literature. Much of the previous work in reachability-based control

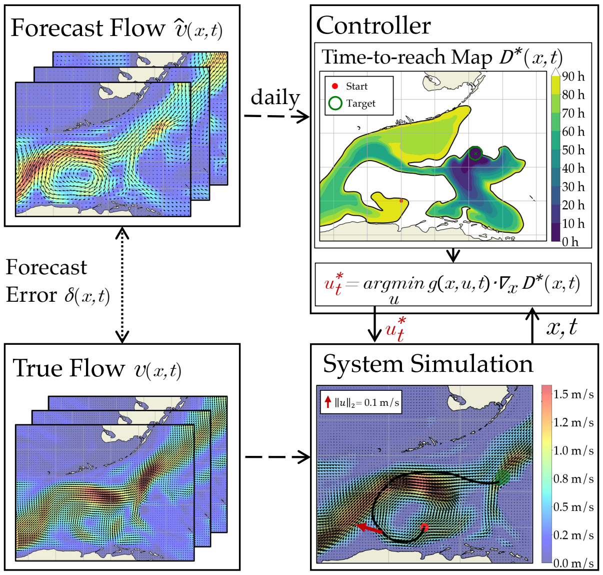


Figure 5.1: Our method for reliable navigation leveraging flows is based on frequent replanning. For that the Time-to-reach map $\mathcal{D}^*$ is computed daily as new flow forecasts $\hat{\boldsymbol{v}}(\boldsymbol{x}, t)$ become available. Then for closed-loop operation, the time-optimal control $\boldsymbol{u}_t^*$ is calculated from $\mathcal{D}^*$ which is equivalent to replanning at every step. The system is simulated using the true flow $\boldsymbol{v}(\boldsymbol{x}, t)$ which differs from the forecast by the forecast error $\delta(\boldsymbol{x}, t)$.

has focused on least restrictive control for safety specifications [5], where the terminal cost encodes constraints that must be satisfied for safety, and the solution of the corresponding HJ PDE provides both a reachable set of states that satisfy the constraints, as well as the control $\boldsymbol{u}^*$ to apply at the boundary of the reachable set in order to stay within this safe set. This least restrictive framework has been applied both to safe trajectory planning [44, 76, 113] and to learning-based control [38, 121].

For the full framework visualized in Fig.5.1 we propose to solve the Multi-time reachability problem once per forecast, which are received at regular (daily) intervals (see Alg.1). The most recent time-to-reach map is then used as closed-loop control policy π which by construction provides reliability through what can be thought of as time-optimal replanning at every time step. For an intuitive example, take the setting in Fig. 5.2: at $\boldsymbol{x}_0$ the agent applies $\boldsymbol{u}_0^*$ based on the planned time-to-reach map. As the true currents v are different than the forecast $\hat{v}$, the agent finds itself at $\boldsymbol{x}_1$, a different state than expected. Based on our forecast, the time-optimal control from this state onward can again directly be computed from the time-to-reach map.

There are two advantages of this method over classic MPC replanning based on fast non-linear programming or graph-based methods: (a) much higher replanning frequencies are possible because deriving the optimal control from the value function is computationally

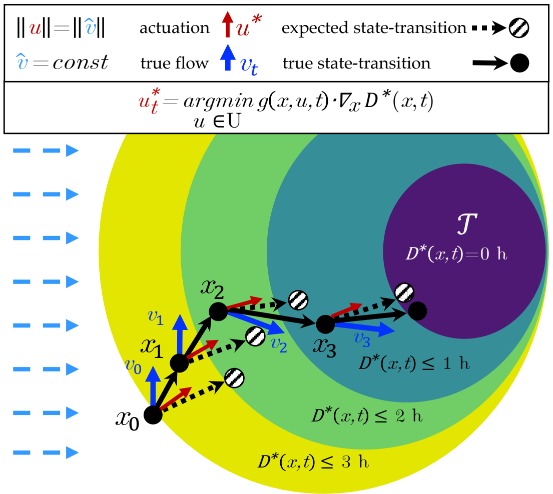


Figure 5.2: In time-invariant forecast flows $\hat{v}$ the time-to-reach map $\mathcal{D}^*(\boldsymbol{x}, t)$ is static. The level-sets of $\mathcal{D}^*(\boldsymbol{x}, t)$ indicate how fast an agent can reach the target from a specific state $\boldsymbol{x}$. With holonomic actuation the optimal control is the spatial gradient $\boldsymbol{u}_t^* = \nabla_x \mathcal{D}^*(\boldsymbol{x}, t)$. Applying $\boldsymbol{u}_t^*$ closed loop leads to reaching the target region $\mathcal{T}$ even if the true currents are different $v_t \neq \hat{v}_t$. This schema is equivalent to time-optimal replanning at every step which leads to its reliability.

cheap compared to solving the optimal control problem from a new state at every time step, and (b) less discretization error because the HJ PDE solves the continuous-time problem, whereas the classic MPC methods rely on discretization in time and space to enable fast planning in the loop.

Algorithm 1: Multi-Time HJ Closed-loop Schema

```
Input: Forecast Flow(s) $\hat{v}(\boldsymbol{x}(s), s)$, $t = 0$, $\boldsymbol{x}_t = \boldsymbol{x}_0$
while $t \leq T_{max}$ and $\boldsymbol{x}_t \notin \mathcal{T}$ do
    if new forecast available then
        compute time-to-reach map $\mathcal{D}^*$
    Use latest $\mathcal{D}^*$ for control $\boldsymbol{u}_t^* = \arg\min_{\boldsymbol{u} \in \mathbb{U}} g(\boldsymbol{x}_t, \boldsymbol{u}, t) \cdot \nabla_x \mathcal{D}^*$
    $x_{t+1} = x_t + \int_t^{t+1} f(\boldsymbol{u}, x(s), s)\, ds$
```

5.3 Experiments

In this section we evaluate our control schema of using multi-time HJ reachability as closed-loop control policy π and compare it to baseline methods on realistic ocean currents.

Experimental Set-Up

We investigate the reliability of various controllers in navigating a two dimensional Autonomous Surface Vehicle (ASV) with holonomic actuation of fixed magnitude $||g(\boldsymbol{u}, \boldsymbol{x}, t)||_2 = ||\boldsymbol{u}||_2 = 0.1\frac{m}{s}$. The control is the thrust angle θ and the ASV is navigating in strong ocean currents $v(\boldsymbol{x}, t) \in [0.3\frac{m}{s}, 2\frac{m}{s}]$ which it wants to hitchhike to reach the targets. In the following, we describe how we ensure realistic ocean forecast simulation and obtain a large set of missions. Then we explain the baselines and evaluation metrics.

Realistic Ocean Forecast Simulation We use the simulation setup as outlined in Chapter 2.2. Ocean forecast data are the HYCOM forecast [20] and Copernicus hindcasts [35] for the Gulf of Mexico region. To simulate realistic conditions, we provide the control methods daily with the HYCOM forecast as it becomes available while simulating the system dynamics with the hindcast as the true flow $v(\boldsymbol{x}, x)$(Fig. 5.1). We investigate two settings (a) planning on HYCOM forecasts and simulating on HYCOM hindcasts (HYCOM-HYCOM) and (b) planning on HYCOM forecasts and simulating on Copernicus hindcasts (HYCOM-Copernicus) (5.3).

To estimate how realistic our simulations are, we compare the simulated forecast error δ across our start-target mission set $\mathbb{M}$ with the HYCOM forecast error as estimated by Metzger et. al. using extensive drifter buoy data [94]. In Fig.5.3 we visualize two metrics, the velocity RMSE and the vector correlation, where 2 represents perfect correlation and 0 no correlation [27]. We find that the HYCOM-HYCOM setting underestimates the forecast error, especially in the first 24h where the forecast is perfect. The HYCOM-Copernicus setting is realistic as the simulated forecast error is of similar magnitude as the actual HYCOM forecast error.

Large Representative Set of Missions To obtain a set of start-target missions $\mathbb{M}$ we first fix 18 regularly spaced starting times t_i between November 2021 and February 2022. For each starting time t_i we uniformly sample 16 start points $x_{\text{Start}}^{t_i,j}$ spatially over the Gulf of Mexico. In our underactuated setting, many start-target missions are impossible even if the true currents are known. Therefore, for the test set $\mathbb{M}$ we need to ensure that each mission is fundamentally feasible given the true currents. To generate only feasible missions from each starting point $x_{\text{Start}}^{t_i,j}$, we calculate the forward reachable set (FRS) starting at t_i for a maximal time horizon of $T_{\max} = 120$h using HJ reachability. The FRS is the set of all states $\boldsymbol{x}_s$ at time s for which there exists a control signal $\boldsymbol{u}(\cdot)$ such that $\boldsymbol{\xi}_{t_i, x_{\text{Start}}^{t_i,j}}^{\boldsymbol{u}(\cdot)} = \boldsymbol{x}_s$. To get a variety of mission durations, we sample 4 relative times $\Delta t_k \in [20, 120]$h and sample a target

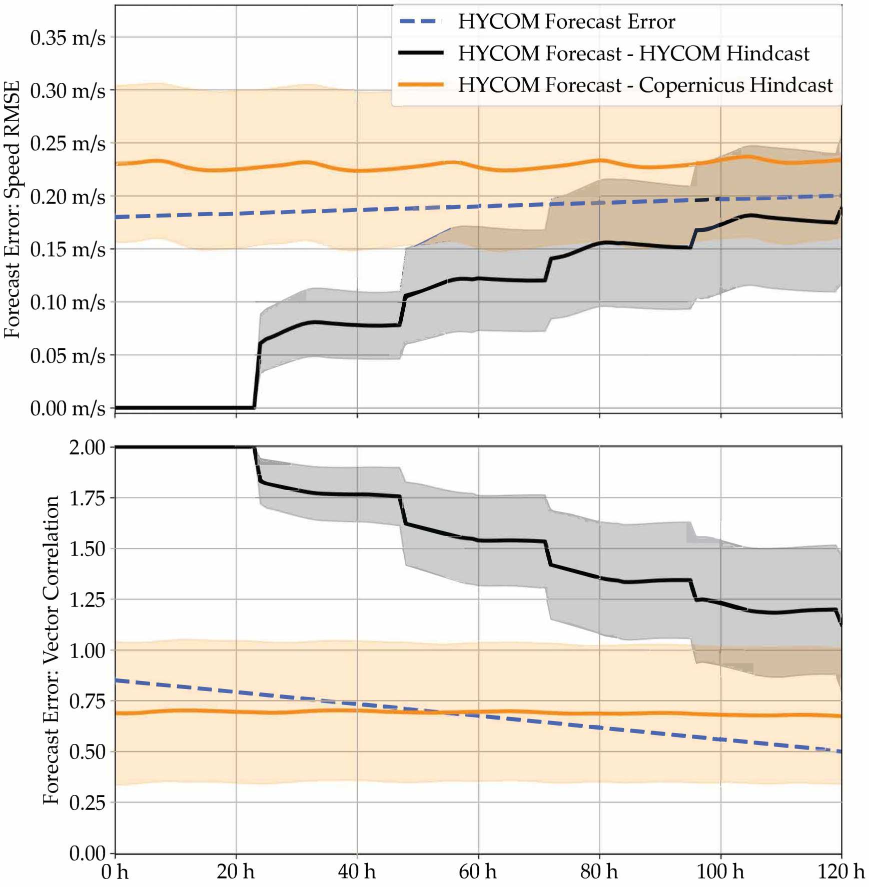


Figure 5.3: The simulated forecast errors in our experiments are of similar magnitude as the actual HYCOM forecast errors [94]. The graphs show the mean and standard deviation of two forecast error metrics, velocity RMSE and vector correlation, over our mission set $\mathbb{M}$ for the two simulation settings.

point $x_{\mathcal{T}}^{i,j,k}$ from within the forward reachable set at $s = t_i + \Delta t_k$. Then $x_{\mathcal{T}}^{i,j,k}$ is guaranteed to be reachable from $x_{\text{Start}}^{t_i,j}$ and the target $\mathcal{T}$ is the circular region of radius $r_{\mathcal{T}} = 0.1°$ around it. This gives us a large and diverse start-target set $\mathbb{M}$ of 1152 missions ranging in duration from 20 to 120 hours as visualized in Fig. 5.4, together with their respective time-optimal trajectories based on the true currents.

Baseline Controllers We compare the performance of our multi-time HJ reachability closed-loop strategy with four baselines. The first baseline is the HJ Closed-Loop method which is the same as our method except that it employs the classic backwards reachability value function. The backwards reachability value function is calculated based on the earliest possible arrival time and used for closed-loop control, as explained in Sec. 5.2. Our second

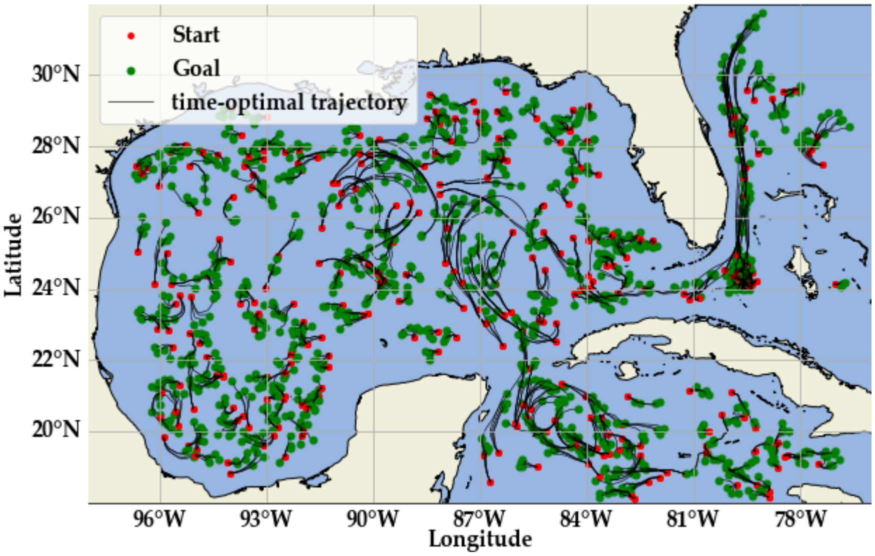


Figure 5.4: We evaluate our method over a large set of 1152 start-target missions spatially distributed over the Gulf of Mexico and temporally from November 2021 until February 2022.

baseline (Robust Hamilton-Jacobi-Isaacs (HJI) Closed-Loop) is a robust controller which is equivalent to HJ Closed-Loop but instead of solving for the classic backward reachability value function, it solves for the reach-avoid value function with a small bounded disturbance of $d = 0.05\frac{m}{s}$ [39]. This means that the control extracted from it in closed-loop is a conservative control that can arrive at $\mathcal{T}$ or get as close as possible to $\mathcal{T}$ at the earliest arrival time despite the worst-case disturbance d. The third method (HJ + Waypoint Tracking) we compare with is based on the idea of tracking waypoints to compensate for the forecast error. It plans the time-optimal path on each new forecast using HJ and at each time actuates towards the next waypoint in this plan. Lastly, we compare against a Naive-to-Target approach which ignores the forecasts altogether and always actuates towards the center of the target region. Note that we do not compare with any graph-based methods such as RRT and A* because they only approximate the time-optimal control problem that HJ reachability solves exactly in continuous time and space. Hence, we expect them to be strictly worse.

Evaluation Metrics The key performance metric for our controllers is *reliability*, defined as the success rate of a controller over the set of missions $\mathbb{M}$ (Sec. 5.1). To further elucidate the controller performances we look at two additional metrics. Finally, we investigate how the reliability of a controller changes with the forecast error.

To evaluate how fast a controller reaches a target, we compute the time-optimality ratio ρ. We define it as the time it took the controller to reach $\mathcal{T}$ divided by the fastest it could have reached $\mathcal{T}$ under perfect knowledge of the currents. It is calculated over the set of all missions in which the controller succeeds ($\tilde{\mathbb{M}}_{ctrl}$) as

$$\rho = \frac{1}{||\tilde{\mathbb{M}}_{ctrl}||} \sum_{\tilde{\mathbb{M}}_{ctrl}} \frac{\Delta t_{\text{arrival ctrl}}}{\Delta t_{\text{arrival best-in-hindsight}}} \tag{5.8}$$

A value of 1 means that the controller was as fast as possible, and 1.1 means that it took 10% longer than the fastest possible.

The minimum distance to the target measures the closest the controller got to $\mathcal{T}$ during the entire simulation horizon T_{max} in degrees of latitude and longitude. The minimum distance to the target is positive, except for when the controller succeeds in a mission, then it is 0.

Lastly, our diverse set of missions $\mathbb{M}$ allows us to investigate how the reliability of various controllers changes with increasing forecast error. For that we calculate the average forecast error RMSE(δ) for each mission spatially over a regional box containing the start and target region and temporally over the 5 day horizon of each forecast. We then group the missions by their RMSE(δ) into 20 bins and calculate for each controller the success rate in each bin. To make the trends more visible, we fit a weighted linear regression over these 20 success rate - RMSE(δ) points and weight each bin by its number of missions.

Experimental Results

Simulation Setting HYCOM-HYCOM

In this setting we evaluate the controller performance over $||\mathbb{M}|| = 1152$ start-target missions and run the simulation for $T_{max} = 150h$. If the controller reaches the target region within that time, it is successful; otherwise, it failed. Our control approach achieves a success rate of 99% and outperforms the baselines (Table 5.1). The time to reach the target is on average only 3% higher than the fastest possible, here HJ Closed-Loop performs slightly better with 2%. The Naive-to-Target controller succeeds only in 84.9% of the missions. These results highlight that highly reliable navigation is possible with a low (short-term) forecast error. However, as of now the actual forecast error of current ocean models is significantly higher (Fig.5.3) which makes this an optimistic performance estimate. To evaluate the statistical significance of our results, we perform a one-sided two sample z proportion test for each of the controllers against the Naive-to-Target baseline. With Γ denoting the success rate of a controller, our null hypothesis is $H_0 : \Gamma_{\text{Naive-to-Target}} = \Gamma_{\text{controller}}$ and the alternative hypothesis is $H_A : \Gamma_{\text{Naive-to-Target}} < \Gamma_{\text{controller}}$. We obtain that the success rate of all controllers is higher than Naive-to-Target in a statistically significant way (p-values Multi-Time HJ CL $p = 3.9e^{-36}$, HJ CL $p = 2.8e^{-27}$, Robust HJI CL $p = 1.36e^{-22}$, HJ + Waypoint Tracking $p = 3.65e^{-15}$).

Simulation Setting HYCOM-Copernicus

To ensure comparability across settings we take the same set of missions $\mathbb{M}$, however, with the Copernicus hindcasts only 837 of the missions are fundamentally feasible, which makes the set $\mathbb{M}$ smaller. Again, our control approach achieves a success rate of 82.3% and outperforms the baselines, but with a lower margin than in HYCOM-HYCOM (Table 5.1). We perform the same statistical significance test and find that only our approach has a statistically significant higher success rate than Naive-to-Target ($p = 0.012$). We want to emphasize that a 4.4% increase over the baseline in this challenging setting with large forecast errors is a sizeable improvement. Moreover, the results from HYCOM-HYCOM indicate that performance improves with better forecasts. Our closed-loop control schema is easily extendable to include learning about the currents while operating in them and using this information to improve its estimate of the future currents and thereby improve performance.

Figures 5.5 show how the success rate of the controllers changes for missions with varying forecast errors. As expected, we see that the success rate decreases with increasing forecast error with different slopes for different controllers. The performance of our Multi-Time HJ Closed-Loop approach decreases slower than the baselines. In the HYCOM-HYCOM setting it is almost unaffected by a higher forecast error. Note that we would expect Naive-to-Target to be indifferent to the forecast error as it does not consider the forecast. However, for Naive-to-Target, we observe a significant drop in performance for missions with high forecast error. We hypothesize that the forecast error is higher in regions with complex and strong currents, conditions which are inherently more challenging to navigate. This could explain why Naive-to-Target fails more frequently with a higher forecast error.

5.4 Conclusion

In this work we have demonstrated that planning with Multi-Time HJ Reachability on daily forecasts and using the value function for as closed-loop control policy π enables reliable navigation of underactuated agents leveraging complex flows. The reliability of our method stems from the fact that the optimal control extracted from the value function at every time step is equivalent to full-horizon time-optimal replanning. There are two key advantages over classic MPC with non-linear programming or graph-based methods. First, our method has lower computational cost as it only requires computing the value function once per forecast instead of having to solve an optimal control problem at every time step. Second, our method solves the continuous-time optimal control problem and does not require spatial or temporal discretization that is typically employed by MPC for fast computation. This leads to high reliability at low computational cost, enabling reliable autonomy for resource-constrained systems navigating in flows in the atmosphere and the oceans.

We evaluated the performance of our method in realistic ocean currents over a large set of multi-day start-to-target missions distributed spatially across the Gulf of Mexico and temporally across four months. In the setting of using forecasts from HYCOM and simulating the true currents with HYCOM hindcasts, our method achieves a 99% success rate and outperforms the baselines. However, this setting underestimates the actual HYCOM forecast error. Hence, we also evaluated our method in a setting with forecast errors that reflect more realistic operations: planning on HYCOM forecasts and simulating the true currents with

	Success Rate	Time-Opt.	Min. Dist.
Plan on HYCOM Forecasts – true flows HYCOM Hindcasts			
Multi-Time HJ CL	**99.0%***	1.0332	**0.060°**
HJ Closed-Loop	97.6%*	**1.0207**	0.104°
Robust HJI CL	96.6%*	1.0346	0.154°
HJ + Waypt. Tracking	94.7%*	1.1412	**0.060°**
Naive-to-Target	84.9%	1.0752	0.068°
Plan on HYCOM Forecasts – true flows Copernicus Hindcasts			
Multi-Time HJ CL	**82.3%***	1.129	0.107°
HJ Closed-Loop	79.1%	1.106	0.110°
Robust HJI CL	78.6%	1.111	0.106°
HJ + Waypt. Tracking	67.1%	1.373	0.091°
Naive-to-Target	77.9%	**1.052**	**0.078°**

Table 5.1: We compare the performance of multiple controllers in two forecast settings. Our Multi-Time HJ Closed-Loop (CL) controller outperforms the baselines in reliability, the key focus of this work. The * marks statistically significant higher success rate compared to Naive-to-Target. The average time-optimality ratio indicates how fast the controller reached the target relative to the best possible in-hindsight. Minimum distance measures the closest the controller got to the target region in degree lat, lon.

Copernicus hindcasts. In this more challenging setting, our method again outperforms the baselines achieving a 82.3% success rate. While we showcased our method on 2D ocean currents, we want to emphasize that it is directly applicable to other flows, e.g. 3D flows in the air and underwater. Lastly, we demonstrated that our method is less affected by increasing forecast errors than the baselines.

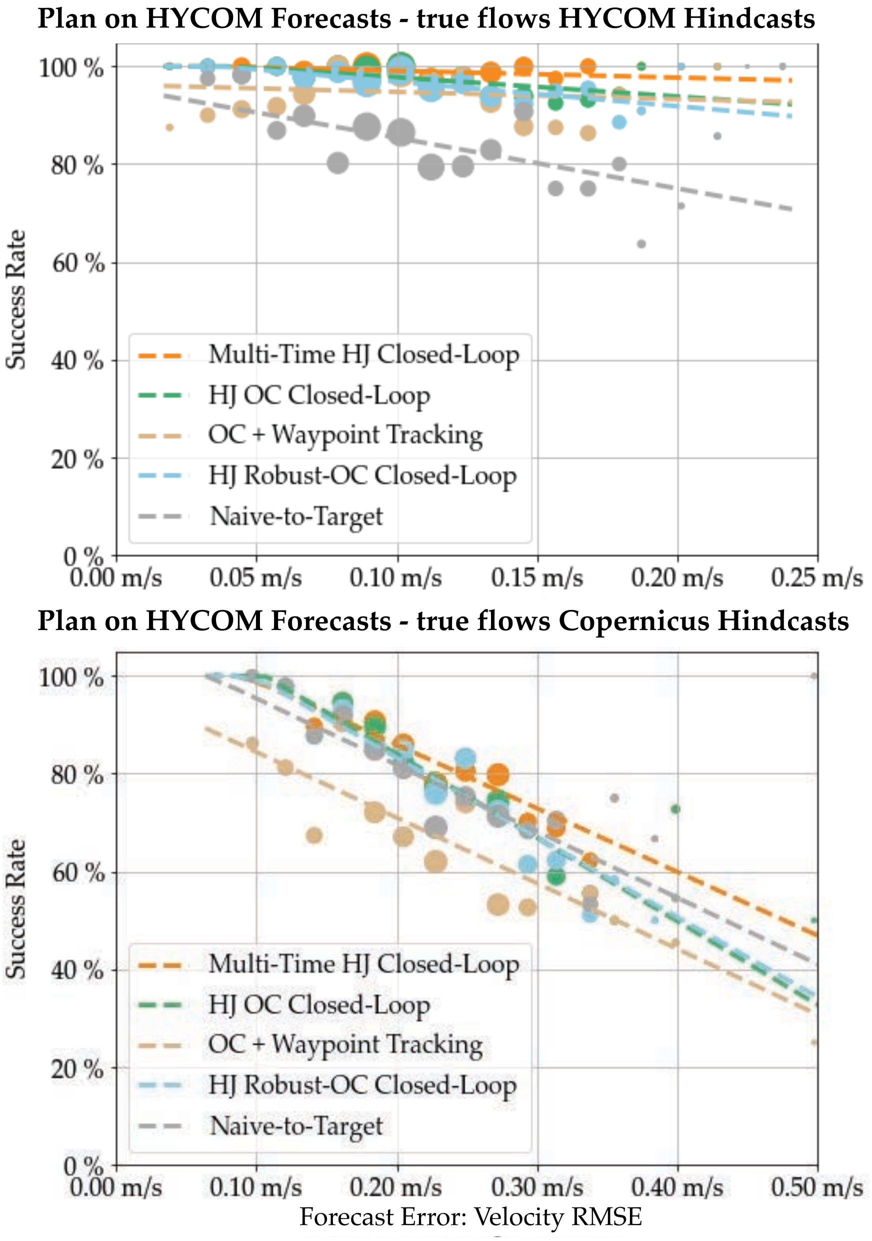


Figure 5.5: The reliability of different controllers decreases for missions with higher forecast error. The dots represent the average success rate over a subset of missions with mean forecast error plotted on the x axis. The size indicates the number of missions in the respective subset. The lines indicate the trends and represent the linear regression of the success-rate to forecast error points weighted by their number of missions.

Chapter 6

Maximizing Seaweed Growth Over Months

This chapter is based on the paper “Maximizing Seaweed Growth on Autonomous Farms: A Dynamic Programming Approach for Underactuated Systems Navigating on Uncertain Ocean Currents” [60], written in collaboration with Matthias Killer, Hanna Krasowski, Manan Doshi, Pierre F.J. Lermusiaux, and Claire Tomlin.

6.1 Introduction

In Part II we developed techniques to operate in known flows both for navigation and for maximizing seaweed growth. In Chapter 5 we saw how the value function for navigation can be used as a closed-loop control policy π which results in reliable navigation despite forecast errors for missions up to 5 days long. However, often we want our systems to optimize objectives over longer time horizons such as maximizing seaweed growth over a multi-month growth cycle or navigating longer distances. This is challenging, as the forecasts from the leading providers are only 5-10 days long [35, 20] and, similar to weather forecasts, the uncertainty for long-term predictions is very high due to the chaotic nature of the system [110, 109]. In control terms, we need to tackle long-term horizon optimization with uncertainty that increases over time. The long-term dependency of the objective, such as navigation or seaweed growth, means that it often cannot be easily decomposed into multiple short-term objectives, as can be seen in Figure 6.5.

In this chapter, we develop techniques that extend the value function beyond the forecast horizon and hence yield a control policy that optimizes for long-term performance. We develop and experimentally validate these techniques with seaweed growth, but they are equally applicable for navigation or other objectives.

Related Work

In order to address the increasing complexity associated with long-time horizons, problems are frequently divided into multiple sub-problems using graph-based methods or hierarchical RL [37, 103]. These approaches are more appropriate for combinatorial optimization problems, where dividing and conquering in subtasks is effective. However, this is not suitable for our problem involving continuous space and long-time dependencies. A potential solution to handle growing uncertainty and distribution shifts over time, as well as to balance short-term and future events, is the use of a discount factor. This technique is commonly applied in discrete RL settings [134] and continuous-time systems [34]. Another approach used to overcome the well-known shortsightedness of MPC is including a terminal value function in the objective, which propagates global information. Recent work in the controls and RL community uses machine learning to learn a terminal value function that approximates the cost to go after the short-term planning horizon [7, 87, 135].

Overview of method & contributions

In this Chapter, we make three main contributions towards controllers that can maximize the growth of floating seaweed farms over long time periods beyond the forecast horizon.

First, we propose a method to obtain a feedback policy for long-term optimal growth beyond the 5-day forecast horizon over which ocean currents are available. For that we estimate the expected growth using historical average currents over a coarse grid and then initialize the DP problem that we solve in Chapter 4 over the forecast horizon with these values (Fig. 6.1, (Sec. 6.2)).

Second, to account for the growing uncertainty of the ocean current estimates, we introduce a finite-time discounting into the DP PDE (Sec. 6.2).

Lastly, we conducted extensive empirical simulations of floating seaweed farms in realistic current settings in the Pacific Ocean over 30 days. We first investigate how different propulsion of the farms would affect the best achievable seaweed growth based on optimal control with known currents. We then evaluate how close different variations of our feedback policy can get to the best possible growth when daily 5-day forecasts are available (Sec. 6.3).

We solve the same running-cost problem as outlined in Chapter 4 and build on it. Sec. 6.2 then details our method to optimize long-term seaweed growth when only short-term, deterministic forecasts are available. Sec. 6.3 contains the closed-loop performance evaluation of our methods and baselines, and we conclude with Sec. 6.5.

6.2 Reasoning Beyond the Forecast Horizon

Feedback Policy Based on Regular Forecasts

The value function $J^*(\boldsymbol{x}, t)$ we developed in Chapter 4, allows us to compute the optimal control $\boldsymbol{u}^*(\boldsymbol{x}, t)$ for all $\boldsymbol{x}, t$ and hence a feedback policy $\pi(\boldsymbol{x}, t)$ for the vessel or multiple vessels in the same region [141]. This policy is the optimizer of the Hamiltonian:

$$\begin{aligned}\pi(\boldsymbol{x},t) &= \arg\max_{\boldsymbol{u}\in\mathbb{U}} f(\boldsymbol{x},\boldsymbol{u},t)\cdot\nabla_x J^*(\boldsymbol{x},t) &(6.1)\\ &= \arg\max_{\boldsymbol{u}\in\mathbb{U}} g(\boldsymbol{x},\boldsymbol{u},t)\cdot\nabla_x J^*(\boldsymbol{x},t) &(6.2)\end{aligned}$$

While π is optimal if J^* was calculated based on true currents v, it can also be applied for closed-loop control when imperfect forecasts $\hat{v}_{FC}$ were used to compute the value function $J^*_{\hat{v}_{FC}}(\boldsymbol{x},t)$. In that case, an agent in state $\boldsymbol{x}$ that executes $\pi_{\hat{v}_{FC}}(\boldsymbol{x},t)$ will likely find itself at a different state than anticipated, as v differs from $\hat{v}_{FC}$. But the control that would be growth optimal under $\hat{v}_{FC}$ can again be calculated directly with $\pi_{\hat{v}_{FC}}(\boldsymbol{x}(t+\Delta t),t+\Delta t)$. Applying $\pi_{\hat{v}_{FC}}$ closed-loop is hence equivalent to full-time horizon re-planning with $\hat{v}_{FC}$ at each time step [141]. This notion of re-planning at every time step at a low computational cost ensures good performance despite forecast errors. Additionally, $J^*_{\hat{v}_{FC}}(\boldsymbol{x},t)$ can be updated periodically. In our experiments, we compute $J^*_{\hat{v}_{FC}}(\boldsymbol{x},t)$ once per day when receiving a new forecast.

Reasoning Beyond the Forecast Horizon

As the growth cycles of seaweed typically span weeks to months, our objective is to maximize the seaweed mass at an *extended* future time T_{ext} after the final time of the available 5-day forecast T_{FC}. A principled way to reason beyond the planning horizon is to estimate the expected growth our seaweed farm will experience from the state $\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(T_{FC})$ onward and add this as a terminal reward l_{term} to Eq. (4.6) (Chapter 4)

$$J^*_{\hat{v}_{FC},\text{ext}}(\boldsymbol{x},t) = J^*_{\hat{v}_{FC},T_{FC}}(\boldsymbol{x},t) + \mathbb{E}\left[J^*_{T_{\text{ext}}}(\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(T_{FC}),T_{FC})\right] \tag{6.3}$$

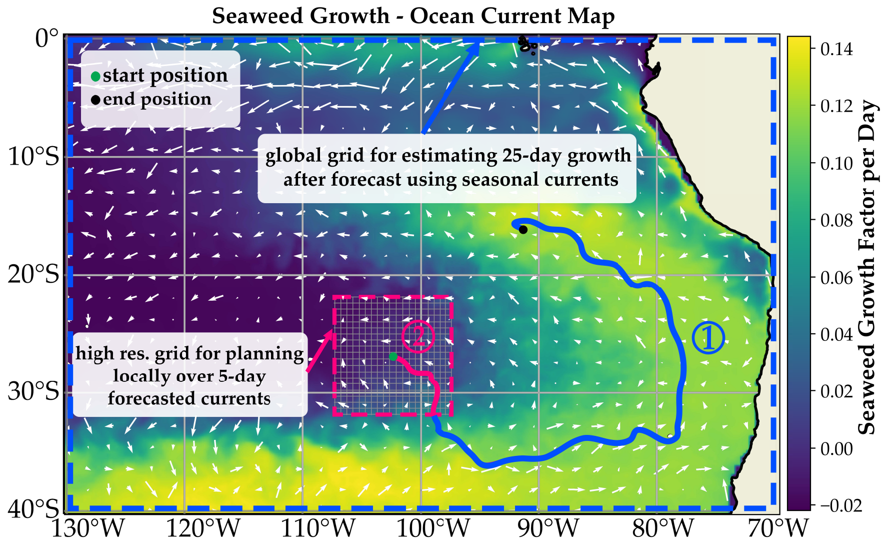
$$J^*_{\hat{v}_{FC},T_{FC}}(\boldsymbol{x},t) = \max_{\boldsymbol{u}(\cdot)} \int_t^{T_{FC}} \Psi(\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(s),s)\,ds \tag{6.4}$$

Where $J^*_{\hat{v}_{FC},T_{FC}}(\boldsymbol{x},t)$ is the growth a vessel starting from $\boldsymbol{x}$ at t will achieve at T_{FC} and $\mathbb{E}\left[J^*_{T_{\text{ext}}}(\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(T_{FC}),T_{FC})\right]$ estimates the additional growth from T_{FC} to T_{ext}. The expectation is taken over the uncertain future ocean currents.

We propose estimating $\mathbb{E}\left[J^*_{T_{\text{ext}}}\right]$ by computing a new value function $J^*_{\bar{v},T_{\text{ext}}}$ based on monthly average currents $\bar{v}$ for the region. To compute $J^*_{\hat{v}_{FC},\text{ext}}$, we again solve the HJ PDE Eq. (4.3) (Chapter 4) backwards with $l_{term}(\boldsymbol{x},T_{FC}) = J^*_{\bar{v},T_{\text{ext}}}(\boldsymbol{x},T_{FC})$. This approach is visualized in Figure 6.1.

Finite-time Discounting to Mitigate Uncertainty

As the oceans are a chaotic system, the uncertainty of the predicted ocean currents increases over time. We can incorporate this increasing uncertainty in the value function by using the


Seaweed Growth - Ocean Current Map
● start position
● end position
global grid for estimating 25-day growth
after forecast using seasonal currents
high res. grid for planning
locally over 5-day
forecasted currents
①
②
0°
10°S
20°S
30°S
40°S
130°W
120°W
110°W
100°W
90°W
80°W
70°W
Seaweed Growth Factor per Day
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−0.02

$$\pi_{Forecast}(\boldsymbol{x}, t) = \arg\max_{\boldsymbol{u}\in\mathbb{U}} f(\boldsymbol{x}, \boldsymbol{u}, t) \cdot \nabla_x J^*_{Forecast}(\boldsymbol{x}, t)$$

$$J^*_{Forecast}(\boldsymbol{x}, t) = \int^{Forecast} growth(s)ds + \mathbb{E}\left[growth_{afterForecast}\right]$$

Figure 6.1: Long-term planning for autonomous farms to maximize seaweed growth with two time-horizons: **1.** compute the terminal reward $\mathbb{E}[growth_{afterForecast}]$ by computing the value function backwards. We cover a large spatial domain with coarse grid over long-term temporal domain $T > T_{FC}$ after the end of forecast horizon (initialized with 0kg). **2.** Compute value function from $t_0 \rightarrow T_{FC}$ (end of forecast horizon) by initializing the backward value function computation over a smaller spatial domain (with a finer grid) with terminal cost computed in 1).

finite-time discounted optimal control formulation:

$$J^\tau(\boldsymbol{x}, \boldsymbol{u}(\cdot), t) = \int_t^T e^{\frac{-(s-t)}{\tau}} l(\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(s), s)\; ds + l_{term}(\boldsymbol{\xi}^{\boldsymbol{u}(\cdot)}_{t,\boldsymbol{x}_0}(T), T),$$

where τ is the discount factor. The smaller τ, the more future rewards are discounted. We derive the corresponding HJ PDE by following the steps in [34] and instead of Eq. (4.3) we obtain:

$$\frac{\partial J^{*,\tau}(\boldsymbol{x}, t)}{\partial t} = -\max_{\boldsymbol{u}} \left[\nabla_{\boldsymbol{x}} J^{*,\tau} \cdot f(\boldsymbol{x}, \boldsymbol{u}, t) + l(\boldsymbol{x}, t)\right] + \frac{J^{*,\tau}(\boldsymbol{x}, t)}{\tau}$$

Summary Control Algorithm Variations

All variations of our method are closed-loop control policies π derived from a value function (Sec. 6.2). There are four variants which differ only in how the value function is computed. First, when the true currents v are known, we compute J^* (Eq. (4.6)) for optimal control (Chapter 4). Second, when only forecasts $\hat{v}_{FC}$ are available, we can calculate $J^*_{\hat{v}_{FC}}$ for planning horizons up to the end of the forecast T_{FC} and update it as a new forecast becomes available (Sec. 6.2). Third, to optimize growth until $T_\text{ext} > T_{FC}$ we can calculate an extended value function $J^*_{\hat{v}_{FC},\text{ext}}$ (Sec. 6.2) using average currents $\bar{v}$. Lastly, we can discount future rewards with $J^{*,\tau}$ (Sec. 6.2) in any of the above value functions. In algorithm 2 we detail the discounted long-term version as it contains all components.

Algorithm 2: Discounted HJ Closed-loop Control

```
Input: Forecast Flow(s) $\hat{v}_{FC}$, $t = 0$, $\boldsymbol{x}_t = \boldsymbol{x}_0$, average Flows $\bar{v}$, discount $\tau$, plan
       until $T_\text{ext}$
Compute $J^{*,\tau}_{\bar{v},T_\text{ext}}$ using $\bar{v}$ (Sec. 6.2);
while $t \leq T$ do
    if new forecast $\hat{v}_{FC}$ available then
        Compute $J^{*,\tau}_{\hat{v}_{FC},\text{ext}}$ (Sec. 6.2);
    Apply policy induced by $J^{*,\tau}_{\hat{v}_{FC},\text{ext}}$ (Sec. 6.2);
    $\boldsymbol{u}_t = \pi^{*,\tau}_{\hat{v}_{FC},\text{ext}}(x_t, t)$;
    $x_{t+1} = x_t + \int_t^{t+1} f(\boldsymbol{u}_t, x(s), s)\, ds$;
    $t \leftarrow t + 1$;
```

6.3 Experiments

In this section, we empirically evaluate and compare various settings of our control scheme for maneuvering the two-dimensional ASV with holonomic actuation in realistic ocean currents and seaweed growth scenarios.

Experimental Set-Up

Seaweed Growth Model Macroalgae growth depends on the algae species, the water temperature, solar irradiance, and dissolved nutrient concentrations, specifically nitrate (NO_3) and phosphate (PO_4) [12]. We use the model of the Net Growth Rate (NGR) of Wu et al. [144] because it models the key growth dependencies without maintaining additional state variables to model the plant-internal nutrient state [92]. We used the parameters of a temperate species from the work of Martins and Marques [92] and Zhang et al.[150]. In this model, the NGR is determined by the gross growth rate r_{growth} and the respiration rate r_{resp} caused by metabolism. Resulting in the change of biomass:

$$\dot{m}(t) = m(t) \cdot \text{NGR} = m(t) \cdot (r_{growth} - r_{resp}). \tag{6.5}$$

Fig. 6.1 shows the NGR for our region in January 2022. We assume that the seaweed growth model is known by the planner.

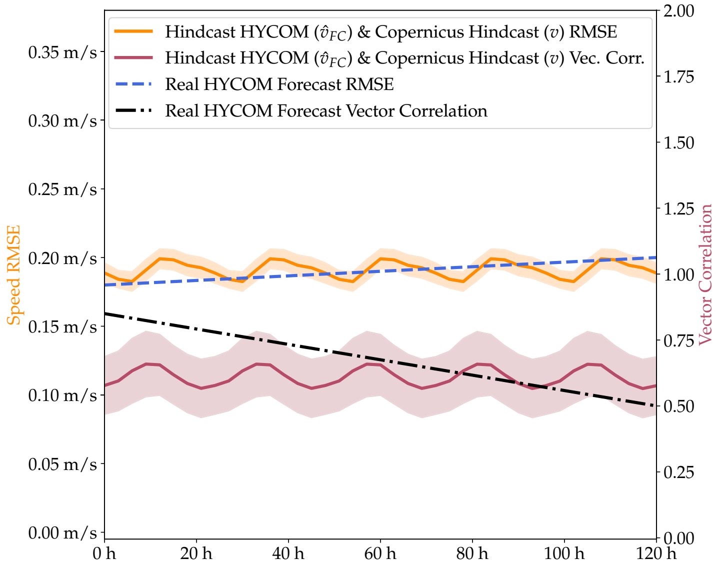


Figure 6.2: To ensure a realistic simulation we compare the forecast errors of our experiments to real HYCOM forecast errors [94]. We can see that our simulation settings closely resemble the real setting. The mean is enveloped by the standard deviation of two forecast error metrics, velocity RMSE and vector correlation over our considered area.

Realistic Ocean Forecast Simulation We simulate ocean currents and forecasts based on the oceanographic systems of HYCOM [20] and Copernicus [35] as outlined in Chapter 2.2. Each system offers 1) a 5-10 day forecast model with daily updates and 2) a hindcast model published a few days later with improved accuracy based on additional data assimilation. To simulate realistic operations, the forecast currents $\hat{v}_{FC}$ received by the planner must differ from the true currents v used for simulation by a forecast error δ that is comparable to the empirical forecast error of oceanographic systems. For our experiments, we use Copernicus hindcasts as v and mimic daily 5-day forecasts $\hat{v}_{FC}$ by giving the planner access to a 5-day sliding time window of HYCOM hindcasts. With this setting, the forecast error δ in our simulation is comparable to the true forecast error as shown in Fig. 6.2. Therefore, we use the HYCOM hindcast data as our forecast source $\hat{v}_{FC}$ and the Copernicus hindcast data as our hindcast source v aka our simulated true currents. v and $\hat{v}_{FC}$ are available with $\frac{1}{12}$th deg resolution.

To estimate the expected future growth beyond the 5-day forecast horizon of $\hat{v}_{FC}$ (Sec. 6.2) we use coarse seasonal averages $\bar{v}$ of the ocean currents in our region. In particular, we use the monthly average currents of 2021 from Copernicus with $\frac{1}{6}$th deg resolution.

Large Scale Mission Generation We conduct our experiments in the southeast Pacific due to high nutrient concentrations. For a large representative set of missions, we initially generated $|\mathbb{M}| = 1325$ starting tuples $(\boldsymbol{x}_0, t)$, uniformly distributed in time between January and October 2022 and across the specified region of longitude range -130°W to -70°W and latitude range -40°S to 0°S. This allows for varying current distributions. The samples were generated maintaining a distance of 0.5 degrees from land to avoid any instant collisions. It should be noted that issues such as stranding were not considered in this work, as an appropriate safety control scheme is proposed in the parallel work [30]. We took the intersection of admissible missions between all controllers to evaluate the results, resulting in 1035 missions for analysis. Admissible missions are defined as those that do not strand and remain within the predefined area, as data is unavailable beyond these bounds. Each mission starts with a seaweed mass of 100kg. A normal seaweed growth cycle is about 60-90 days, but we limit our large-scale experiments to a time period of 30 days per mission to keep the required compute tractable at scale.

Evaluated Controllers We evaluate a variety of controllers under different configurations, which can be classified according to the following criteria: 1) the data utilized as input for the controller, such as the true currents v for planning or forecast data $\hat{v}_{FC}$ and average data $\bar{v}$, and 2) the controller's planning horizon, which may span the entire 30-day period or a shorter *greedy* 5-day interval with periodic replanning. Additionally, we examine controllers employing a discounted value function, as outlined in Sec. 6.2. Moreover, we compare those controllers with the scenario where seaweed farms float freely in the water without any actuation. A comprehensive overview of the configurations for each evaluated controller is provided in Tab. 6.1. For all long-term (30-day horizon) controllers, we compute the terminal reward term of $J^*_{\hat{v}_{FC},\text{ext}}(\boldsymbol{x}, t)$ over the complete predefined area (60x40°) on a coarse grid ($\frac{1}{6}^\circ$ resolution). The running cost term of $J^*_{\hat{v}_{FC},\text{ext}}(\boldsymbol{x}, t)$ is computed only on a smaller grid around the current farm's position (10° window around) but with a higher resolution ($\frac{1}{12}^\circ$).

Evaluation Metrics Our primary objective is to maximize seaweed mass, so we evaluate the results based on the absolute seaweed mass at the end of each mission and the relative improvement in accumulated seaweed mass across different controllers. For relative improvement, we normalize the values within each mission, allowing us to gauge the extent to which a specific controller exceeds a baseline for that mission. This is especially important as the starting position of a mission is a major indicator of achievable growth as illustrated in Fig. 6.3. Finally, we present the average relative improvements across all missions. We consider either the floating system without actuation or our control method planning on the true currents v.

Experimental Results

We investigate our controller's performance under various maximum thrust limitations (u_{max}) in two scenarios: 1) The controller receives the true currents v as planning input, representing

controller	**planning horizon**	**discount factor τ**
w/o discount (v)	30 days	-
floating	-	-
greedy 1 hour ($\hat{v}_{FC}$)	1 hour	-
greedy 5 days ($\hat{v}_{FC}$)	5 days	-
w/o discount ($\hat{v}_{FC} + \bar{v}$)	30 days	-
w/ discount I ($\hat{v}_{FC} + \bar{v}$)	30 days	1.296.000
w/ discount II ($\hat{v}_{FC} + \bar{v}$)	30 days	1.728.000

Table 6.1: We compare various controller settings.

the best-case performance given the selected maximum propulsion; 2) the controller receives daily 5-day $\hat{v}_{FC}$ and the seasonal averages $\bar{v}$, approximating real-world conditions. Moreover, we assess the performance for a floating farm where $u_{max} = 0\frac{m}{s}$.

Fig. 6.4 and Tab. 6.2 compare seaweed mass distributions at the end of each mission under different settings. The average seaweed growth scales linearly with the farm's maximum thrust. Controllers planning on forecasts exhibit slightly inferior performance due to prediction errors compared to those using true currents. The performance shift in the upper quartile is more pronounced for lower u_{max} values, possibly because higher u_{max} values better compensate for forecast errors. Nonetheless, with minimum propulsion of $u_{max} = 0.1\frac{m}{s}$, our controller performs 9.6% better on average than a freely floating farm.

The starting location of a mission significantly influences 30-day growth, as shown in Fig. 6.3. High-growth missions are situated in the east and south of our region, aligning with nutrient-rich areas also visible in Fig. 6.1.

Higher propulsion in real-world applications may be economically infeasible due to the cubic increase in energy consumption with u_{max}. Consequently, we set u_{max} to $0.1\frac{m}{s}$ for subsequent investigations. We evaluate the performance of two greedy controllers under this setting and various configurations of our long-term controllers. We aim to increase the performance of the long-term controller operating on forecasts $\hat{v}_{FC}$ and $\bar{v}$ to match the best growth achievable with true currents v. To this end, we employ the discount formulation proposed in Sec. 6.2 for two settings (6.1) to account for increasing uncertainty over time.

As illustrated in Tab. 6.3, both the greedy and long-term controllers outperform the floating scenario. The performance of our greedy controller, planning over a 5-day horizon, closely matches that of the long-term controllers. Using the discounted control scheme slightly improves the long-term controller, yielding the best overall performance.

Fig. 6.5 evaluates the floating system, the 5-day greedy, and the long-term controller without discount (two settings: with v and $\hat{v}_{FC} + \bar{v}$) in a 60-day scenario. The greedy controller aims for the nearest growth region, while the long-term controller balances short-term losses against long-term gains, as demonstrated in the sub-figure of Fig. 6.5 depicting seaweed mass over time. The zig-zag shape of the lines is due to nocturnal respiration of

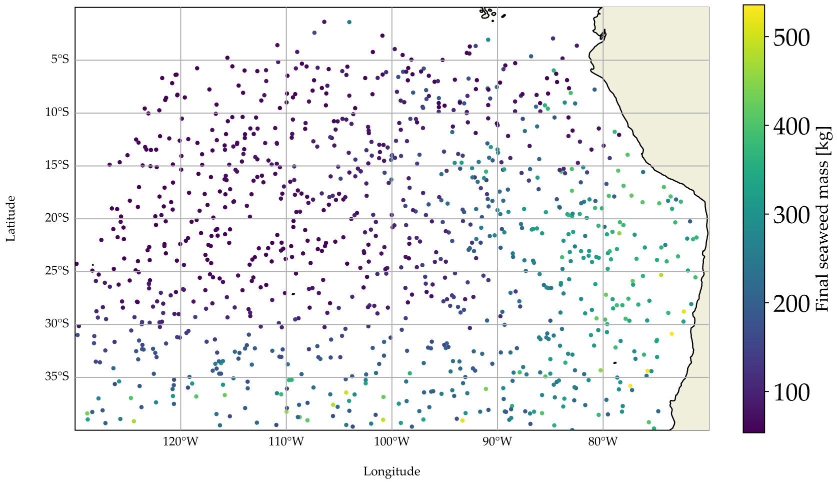


Figure 6.3: We sample a large representative set of starting points for seaweed farms in the Pacific to evaluate how much growth our controllers can achieve. The colorized points show the best achievable seaweed mass after 30 days using $u_{max} = 0.1\frac{m}{s}$.

u_{max}	**planning input**	**rel. growth**	**abs. final seaweed mass**
$0.0\frac{m}{s}$	(floating)	100%	145.29kg±100.30kg
$0.1\frac{m}{s}$	v	115.38%	166.45kg±109.67kg
	$\hat{v}_{FC} + \bar{v}$	109.62%	159.29kg±107.46kg
$0.2\frac{m}{s}$	v	128.69%	182.04kg±115.11kg
	$\hat{v}_{FC} + \bar{v}$	121.29%	173.72kg±112.94
$0.3\frac{m}{s}$	v	141.27%	194.98kg±117.39kg
	$\hat{v}_{FC} + \bar{v}$	133.28%	187.01kg±116.60kg
$0.4\frac{m}{s}$	v	153.71%	206.96kg±118.34kg
	$\hat{v}_{FC} + \bar{v}$	145.79%	199.50kg±118.09kg
$0.5\frac{m}{s}$	v	165.79%	218.10kg±118.59kg
	$\hat{v}_{FC} + \bar{v}$	158.14%	210.78kg±117.72kg

Table 6.2: Normalized seaweed growth performance per mission (averaged) for different maximum propulsions relative to floating.

seaweed. The greedy controller is driven out of the region, leading to mission termination and an earlier trajectory end.

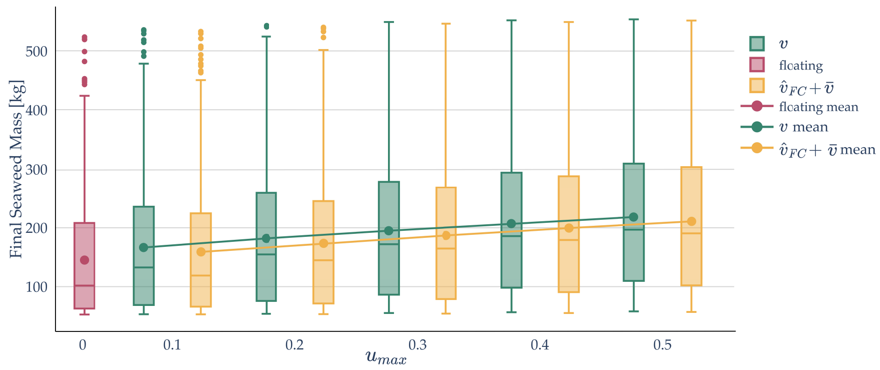


Figure 6.4: Final seaweed mass for different levels of maximum actuation u_{max}. We furthermore compare the settings where different currents are available to the controller: true currents v to forecasted currents plus averages $\hat{v}_{FC} + \bar{v}$.

controller	rel. growth	abs. final seaweed mass
w/o discount (v)	100%	168.45kg±109.67kg
floating (-)	88.20%	145.29kg±99.54kg
greedy 1 hour ($\hat{v}_{FC} + \bar{v}$)	92.24%	152.48kg±102.89kg
greedy 5 days ($\hat{v}_{FC}$)	95.19%	157.78kg±106.04kg
w/o discount ($\hat{v}_{FC} + \bar{v}$)	95.61%	158.84kg±106.71kg
w/ discount I ($\hat{v}_{FC} + \bar{v}$)	95.77%	159.16kg±106.62kg
w/ discount II ($\hat{v}_{FC} + \bar{v}$)	**95.77%**	**159.17kg**±106.66kg

Table 6.3: Normalized seaweed growth performance per mission (averaged) of different controllers with $u_{max} = 0.1\frac{m}{s}$ (normalized to the controller without discount factor given v).

6.4 Discussion

We conducted experiments over a 30-day time horizon to facilitate large-scale testing. In this context, all our controllers substantially outperform the non-actuated floating system. We observed that the performance of our short-term optimizing controller is nearly on par with our long-term controllers. We attribute this to several factors: 1) the nutrient map, and consequently, the growth map, exhibits a smooth gradient that simplifies convergence toward a global optimum, 2) a 30-day time horizon may not adequately capture the short-term and long-term trade-offs, and 3) for higher maximum thrust we would expect to see a higher performance divergence since more distance can be captured but we only evaluated greedy controllers with $u_{max} = 0.1\frac{m}{s}$. Since growth cycles typically span 60-90 days, long-term planning is crucial. In such scenarios, the myopic behavior of greedy policies not only leads them to navigate toward low-growth regions in the vicinity but also fails to account

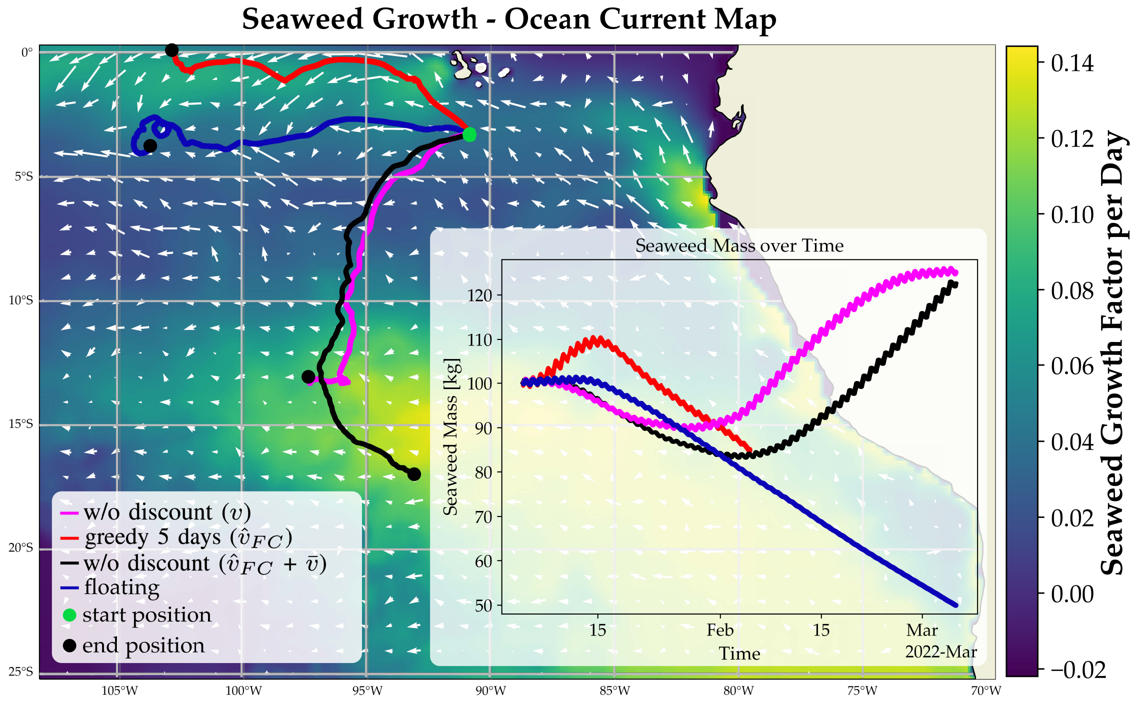


Figure 6.5: Long-term vs. short-term trade-off of a 60-day mission: 5-day greedy controller tries to get to the closest growth region but drifts away, whereas long-term controllers reach a more distant growth-richer area while compromising short-term losses. The 5-day greedy controller leaves the region, hence the termination of the trajectory.

for being pushed out of optimal growth regions, as demonstrated in Example 6.5 over 60 days. For our large-scale evaluation, we only considered missions that remained within the predefined region. As shown in the 60-day example 6.5, this often occurs with greedy controllers or in the floating case. Consequently, we would likely observe a more favorable relative performance of the long-term controllers if we accounted for the filtered missions. We noticed a high variance in seaweed mass across all controllers, which can be attributed to our vessels' inability to reach optimal growth regions within 30 days for many missions. We anticipate that for longer time horizons and increasing maximum thrust, the long-term controller would converge toward the global maximum in the southeast of our region, leading to reduced variance in seaweed mass. Therefore, in future work, we want to investigate performance for longer time horizons.

6.5 Conclusion

In this chapter, we picked off from the Dynamic Programming approach for maximizing seaweed growth in underactuated autonomous ocean farms outlined in Chapter 4. The method solves a running cost problem in the 2D spatial state of the system and generates a value function. In this chapter, we first detailed how this value function can serve as a feedback policy for growth-optimal control across all states and times. The policy can control multiple farms and mitigate forecast errors through closed-loop operation, which is equivalent to replanning at every time step. Furthermore, we extended our method for long-term optimal growth beyond the 5-day forecast horizon by estimating expected growth using seasonal average currents and initializing the DP with these values. To account for increasing uncertainty in ocean currents, we introduced finite-time discounting into the DP PDE. Our extensive empirical evaluation, based on realistic Pacific Ocean current scenarios over 30 days, demonstrated that our approach using only 5-day forecasts and limited propulsion ($u_{max} = 0.1\frac{m}{s}$) achieved 95.8% of the best possible growth and 9.6% more growth than letting the farms passively float. This confirms the feasibility of low-power propulsion and optimal control to enhance seaweed growth on floating farms in real world conditions. We further argued that long-term planning becomes even more important for time horizons over 30 days.

Future work offers multiple avenues for improvement. One possibility is to learn the terminal reward by employing approximate value iteration [7] or a value network, as proposed by Silver et al. [124]. This approach could implicitly learn the distribution shift between $\hat{v}_{FC}$ and v; however, it may require intensive computation for training due to the necessity of i.i.d. samples, which could limit the number of samples taken per mission to just one [124]. Another direction is to make the discount factor state-dependent based on the uncertainty of current predictions, which could be estimated historically or using ensemble models [127, 74]. Lastly, we plan to conduct field tests with multiple autonomous surface vehicles to further validate our method's practicality in real-world ocean environments.

Part IV

Operating with Constraints

Chapter 7

Safety for Underactuated Ocean Vessels

This chapter is based on the paper "Stranding Risk for Underactuated Vessels in Complex Ocean Currents: Analysis and Controllers" [30], written in collaboration with Andreas Doering, Hanna Krasowski, Manan Doshi, Pierre F.J. Lermusiaux, and Claire Tomlin.

7.1 Introduction

In Part III, we have demonstrated how to reliably operate an underactuated robotic system in flows under significant forecast error. However, the approaches we have discussed so far do not factor in safety aspects, although the use of ASV in unmanned and long-term operations may pose crucial safety risks. In the event of significant damage, the ASV may become inoperable and may be abandoned or sunk, resulting in financial losses and potential environmental impacts. One important safety hazard is shallow waters, especially near strong currents, as the ASV can easily strand. Another significant safety hazard is entering a garbage patch that has a high concentration of marine debris, which can cover an area of up to 1.6 million km^2 [70] as in the case of the Great Pacific Garbage Patch (GPGP). Garbage can get entangled in rotors of the ASV motor or damage other components, resulting in loss of control. Furthermore, collisions with other vessels may cause damage to the ASV and potentially endanger the crew of the other vessel. Therefore, shipping lanes are another area of increased risk to the ASV as they are used by large fast-moving vessels.

This chapter focuses on incorporating safety constraints into the closed-loop control policy π building on the Multi-time Reachability framework introduced in Chapter 3. While we focus here on the risk of stranding for navigation objectives, the same technique can be used for arbitrary objectives, such as maximizing seaweed growth and time-varying obstacles such as ships.

Related Work Let us start by looking at related work on safe motion planning for autonomous vessels in maritime environments. Most of the existing research focuses on collision avoidance [42, 57] and compliance with the Convention on the International Regulations for Preventing Collisions at Sea (COLREGS) [26, 95, 153, 154, 55, 69] as safety aspects for motion planning of autonomous vessels. For example, Zhao et al. [153] use reinforcement learning to achieve COLREGS-compliant motion planning for encounters with multiple vessels. The studies [42, 57] only look at collision avoidance and achieve this by employing velocity obstacles. In general, this research on safe motion planning for autonomous vessels considers fully actuated vessels. Since our vessel has restricted maneuverability due to its underactuation, it is unnecessary to comply with the rules for power-driven vessels from the COLREGS. Therefore, we focus on the safety specification of collision avoidance with largely static obstacles such as shallow areas, shipping lanes, or garbage patches. The avoidance of dynamic obstacles and forbidden regions including the coordinations of multiple vehicles has been treated using Hamilton-Jacobi Reachability [81] and applied in real-time with underwater vehicles to avoid too shallow areas [128]. Agents operating in three-dimensional flows can evade obstacles or strong currents by utilizing the third dimension which has been demonstrated for stratospheric balloons by [52]. They ensure safe paths by formulating the problem as a discretized Markov Decision Process and a heuristic cost function. This only ensures heuristic safety and relies on a realistic uncertainty distribution of stratospheric winds, which are not available for ocean currents.

Although many articles on maritime safety consider underactuated vessels, most consider underactuation due to non-holonomic actuation of vessels such as [98, 145, 80, 15]. In this thesis, we focus on underactuated systems that have a maximum propulsion that is less than the magnitude of forecasted flow and often also the forecast error, posing severe challenges for the safety of ASVs.

Robust MPC approaches can guarantee safety under disturbances by ensuring that the system is always in a state from which it can reach a robust control invariant set within a finite time horizon [9, 68]. In this robust control-invariant set, there always exists a control input that ensures that the system can stay in this set indefinitely. However, in our problem setting with underactuated vessels and imperfect, deterministic ocean current forecasts, no such control invariant set exists; hence robust control with realistic bounds is infeasible.

We make two main contributions: First, we perform an empirical evaluation of stranding risk for free-floating vessels in the Northeast Pacific. Second, we present our methods of Hamilton-Jacobi Multi-Time Reachability (HJ-MTR) with re-planning on two timescales for safe motion planning of underactuated ASV in a setting with realistic ocean currents and daily forecasts. Furthermore, we evaluate our controller with several baseline controllers over a large set of simulated missions.

We structure the chapter as follows: we define our problem statement in Sec. 7.2 and motivate the need for a safety controller with a stranding study in Sec. 7.3. In Sec. 7.4 we introduce our method and summarize HJ-MTR. We present our extensive simuation experiments in Sec. 7.5 and discuss them in Sec. 7.6. We conclude and present future work in 7.6.

7.2 Problem Statement

We now define the problem of collision avoidance for underactuated vessels by introducing the flow model, the vessel model, and the representation of obstacles. We then introduce the notion of stranding as our key performance measure.

System Dynamics, Obstacles and Target

We consider moving in a general time-varying non-linear flow field $v(\boldsymbol{x}, t) \to \mathbb{R}^n$, with $\boldsymbol{x} \in \mathbf{R}^n$ representing the spatial state, $t \in [0, T]$ the time and n the dimension of the spatial domain. In our case, $n = 2$ as we regard an ASV operating on the ocean surface. We denote the actuation signal by $\boldsymbol{u}$ from a bounded set $\mathbb{U} \in R^{n_u}$ with n_u the dimension of the control. Let $\boldsymbol{\xi}(s) \in \mathbf{R}^n$ denote the position of our ASV at time s. To recap from Chapter 2.1, our model for the dynamics of the ASV is given by

$$\begin{aligned} \dot{\boldsymbol{\xi}}(s) &= \boldsymbol{f}(\boldsymbol{\xi}(s), \boldsymbol{u}(s), s) && (7.1) \\ &= v(\boldsymbol{\xi}, s) + g(\boldsymbol{u}, \boldsymbol{x}, t) \quad \forall s \in [0, T], && (7.2) \end{aligned}$$

Here, the actuation $\boldsymbol{u}$ corresponds to the relative velocity $g(\boldsymbol{u}, \boldsymbol{x}, t)$ of the ASV with respect to the ocean. Therefore, the absolute velocity of the vehicle is given by the vector sum of the ocean currents at the location of the vehicle and the relative velocity of the vehicle with respect to the currents. The maximum actuation of the vessel is limited to $||g(\boldsymbol{u}, \boldsymbol{x}, t)||_2 \leq \boldsymbol{u}_{\max}$. We define the target and obstacle as sets $\mathcal{T} \in \mathbb{R}^n$ and $\mathcal{O} \in \mathbb{R}^n$ respectively. We assume that these sets are not time dependent. However, note that the methods presented can be extended to time-dependent cases by the algorithm described in [31].

Problem Setting

The agents goal is to navigate *safely* and *reliably* from a start state $\boldsymbol{x}_0$ at the start time t_0 to a target region $\mathcal{T} \in \mathbb{R}^n$. We employ the same empirical definition of *reliability* as [141] defining it as the success rate of a controller navigating from $\boldsymbol{x}_0$ at t_0 to $\mathcal{T}$ within the maximum allowed time T_{max}, over a representative set of start-target missions $\{\boldsymbol{x}_0, t_0, \mathcal{T}\} \in \mathbb{M}$. We define stranding as an agent entering the obstacle set $\mathcal{O}$ before T_{max}. We then quantify *safety* as the stranding rate of a controller over the same set of missions. We define stranding as entering waters with depth less than −150 m. This is application-specific and can be extended to include other entering obstacles such as garbage patches or areas with high traffic density.

We use the oceanographic systems of HYCOM [20] and Copernicus [35], similar to prior research [141]. Each of the systems offers a 5- to 10-day ocean current forecast based on their models with daily updates. They also offer a so-called hindcast with higher accuracy, which is assimilated from further data and published several days later. For a realistic simulation of real-world operations, the forecasted currents $\hat{v}_{FC}$ received by the vessel need to differ from the true currents v by the forecast error δ, which needs to be comparable to empirical forecast errors of the oceanographic system. If the true currents are known *a* priori and there exists a trajectory that prevents stranding, our method guarantees safety. However, we are interested in realistic settings with a complex empirical distribution of forecast errors $\delta(\boldsymbol{x}, t)$ and severe underactuation, e.g. in our experiments $||g(\boldsymbol{u}, \boldsymbol{x}, t)||_2 = 0.1\,\mathrm{m\,s^{-1}} \ll \mathrm{RMSE}(\delta) \approx 0.2\,\mathrm{m\,s^{-1}}$ and currents $||max(v)||_2 \approx 1.4\,\mathrm{m\,s^{-1}}$ where safety despite worst-case forecast errors is impossible. Hence, in Sec. 7.5 we evaluate the performance of our method empirically over a large set of missions $\mathbb{M}$ in realistic settings with currents and forecasted currents similar to realistic currents and forecasts $\mathbb{V}$. We evaluate the performance in terms of reliability and safety based on the identity functions $\mathbb{I}_{Suc}, \mathbb{I}_{Obs}$ that evaluate to 1 if any state along the trajectory $\boldsymbol{\xi}_{t_0,\boldsymbol{x}_0}^{\boldsymbol{u}(\cdot)}(t)$ is in the sets $\mathcal{T}$ and $\mathcal{O}$, respectively, and to 0 otherwise:

$$\mathbb{E}_{\underbrace{\boldsymbol{x}_0, t \sim \mathbb{M}}_{\text{initial condition}}\;;\;\underbrace{v, \hat{v}_{FC} \sim \mathbb{V}}_{\substack{\text{real and forecasted}\\ \text{ocean currents}}}} \underbrace{\{\mathbb{I}_{Suc}, \mathbb{I}_{Obs}\}}_{\text{Success and Stranding}} \tag{7.3}$$

7.3 Stranding Study

To illustrate the need for our safety controller, we analyze the rate of stranding for free-floating vessels off the coast of California and Mexico between N15° and N40° and W160° and W105°. We define entering an area with a depth of less than 150 m as stranding, due to operational needs of some ASV.

The stranding study can be conducted analytically or experimentally. Here, we perform it empirically by sampling 10 000 missions. Each mission consists of a uniformly sampled starting location in the region investigated, outside the stranding area, and a uniformly sampled starting time. We simulate the trajectories over a time horizon of 10 and 90 days, using Copernicus data for the year 2022. We observe that 1.67% missions strand within 10 days and 5.04% strand within 90 days. In Figure 7.1 we show the spatial distribution of the stranding rate over 90 days. We can only count the strandings that occur in the defined region, as we stop the simulation for platforms leaving the regarded region, which occurred for 1.78% missions within 10 days and 18.75% within 90 days.

7.4 Safe Hamilton Jacobi Controller

The forecasts provided by the available ocean forecasting systems are deterministic. This prohibits us from applying probabilistic methods that would require a realistic distribution of currents [73]. Ocean current forecasts exhibit a distribution shift to real currents; e.g., for HYCOM data [20] the global forecast error for speed is $\mathrm{RMSE}(\delta) = 0.2\,\mathrm{m\,s^{-1}}$ with a vector correlation [27] decreasing over the 5-day forecast horizon [94]. For an underactuated vessel with $||\max(g(\boldsymbol{u}, \boldsymbol{x}, t))||_2 = 0.1\,\mathrm{m\,s^{-1}}$ there exists no control that is robust against a disturbance of $d = 0.2\,\mathrm{m\,s^{-1}}$. Hence we choose to not use robust control but to ensure safety

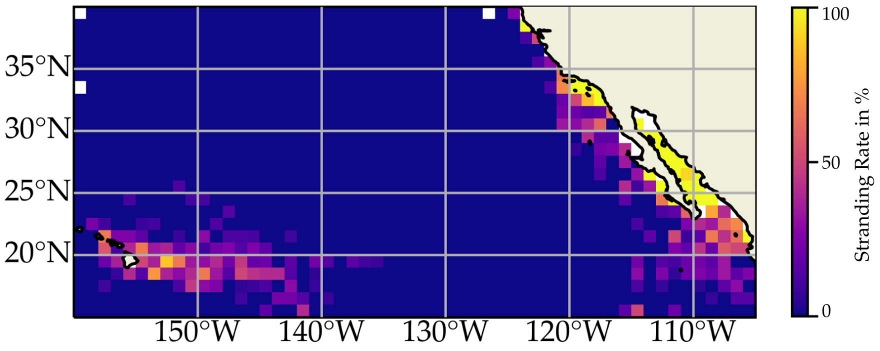


Figure 7.1: Rate of passively-floating vessels stranding, which we define as entering waters with depth less than −150 m. 5.04% of 10 000 simulated vessels that start in the region strand within 90 days. Some vessels float over 20° before stranding.

despite forecast errors by replanning on two timescales. First, we compute the value function daily as we receive new forecasts using HJ-MTR [31].

Second, we use the value function as closed-loop control policy as explained in chapters 5 and 6. This means that for every time step, e.g. 10 min, we re-plan by taking the spatial gradient of the value function at $\boldsymbol{x}$ to obtain the time-optimal control from the state $\boldsymbol{x}$ onward. This is necessary because the true currents v differ from the forecasted currents $\hat{v}_{FC}$, thus at each step we will be in a spatial state $\boldsymbol{x}$ different from that anticipated.

Multi-Time HJ Reachability for Closed-Loop Control

For the sake of completeness of the chapter and focusing on the relevant aspects we summarize the relevant aspects of the HJ-MTR which has been introduced in detail in chapter 3.

We first define a modified dynamical system f_a such that the state $\boldsymbol{x}$ of the vessel becomes frozen when it hits either the target or an obstacle.

$$\dot{\boldsymbol{\xi}} = f_a(\boldsymbol{\xi}, \boldsymbol{u}, s) = \begin{cases} 0, & \boldsymbol{\xi} \in \mathcal{O} \cup \mathcal{T} \\ v(\boldsymbol{\xi}, s) + \boldsymbol{u}(s), & \text{otherwise.} \end{cases} . \tag{7.4}$$

We define the modified loss function $l(\boldsymbol{\xi}, s)$ such that we earn a reward based on how early we reach the target:

$$l(\boldsymbol{\xi}, s) = \begin{cases} -1, & \boldsymbol{\xi} \in \mathcal{T} \text{ and } \boldsymbol{\xi} \notin \mathcal{O}_s \\ 0, & \text{otherwise.} \end{cases} \tag{7.5}$$

Finally, we define the terminal cost function $l_{term}(\boldsymbol{\xi})$ to be infinitely high if the ASV terminates in an obstacle and is equal to the distance from the target set otherwise.

$$l_{term}(\boldsymbol{\xi}) = \begin{cases} \infty, & \boldsymbol{\xi} \in \mathcal{O}_T \\ d(\boldsymbol{\xi}, \mathcal{T}), & \text{otherwise.} \end{cases} . \tag{7.6}$$

Finally, we obtain the Hamilton-Jacobi PDE which lets us solve for the value function:

$$\frac{\partial J^*(\boldsymbol{x}_0, t)}{\partial t} = \begin{cases} 1, & \boldsymbol{x}_0(t) \in \mathcal{T} \cap (\mathcal{O}_t)^c \\ 0, & \boldsymbol{x}_0(t) \in \mathcal{O}_t \\ -v(\boldsymbol{x}, t) - \boldsymbol{u}_{\max} \left\| \nabla_{\boldsymbol{x}} J^* \right\|_2, & \text{otherwise.} \end{cases}$$

$$J^*(\boldsymbol{x}_0, T) = \begin{cases} \infty, & \boldsymbol{x}_0 \in \mathcal{O}_T \\ d(\boldsymbol{x}_0, \mathcal{T}), & \text{otherwise.} \end{cases} . \tag{7.7}$$

This value function J^* subsequently allows us to compute a feedback policy for this system given by

$$\pi_{\hat{v}_{FC}}(\boldsymbol{x}_0, t) = \arg\min_{\boldsymbol{u}} \left[\nabla_{\boldsymbol{x}} J^*(\boldsymbol{x}_0, t) \cdot f(\boldsymbol{x}_0, \boldsymbol{u}, t) \right] \; \forall \boldsymbol{x}_0 \notin (\mathcal{O}_t \cup) \ , \tag{7.8}$$

$$= -\frac{\nabla_{\boldsymbol{x}} J^*(\boldsymbol{x}_0, t)}{\left\| \nabla_{\boldsymbol{x}} J^*(\boldsymbol{x}_0, t) \right\|_2} \boldsymbol{u}_{\max} \tag{7.9}$$

This policy guarantees safety when the value function was computed with the true currents v. However, in realistic settings with only forecasts $\hat{v}_{FC}$ available, we apply $\pi_{\hat{v}_{FC}}$ closed-loop which is equivalent to replanning at every time step. Applying this policy in closed-loop (see Fig. 7.3) means we take the time-optimal control at each state, which is equivalent to full time horizon replanning at each time step.

We introduce the safe Time-to-Reach (TTR) map D^*, which is easier to interpret and can be easily computed from the value function.

$$D^*(\boldsymbol{x}, t) = T + J^*(\boldsymbol{x}, t) - t, \, \forall (\boldsymbol{x}, t) \text{ s.t., } J^*(\boldsymbol{x}, t) \leq 0 \tag{7.10}$$

We illustrate the interpretability of the safe TTR in Fig. 7.2. If $D^*(\boldsymbol{x}, t)$ is e.g. 3, it means that a vessel starting at $\boldsymbol{x}$ at time t can reach the target in 3 time units when following the optimal control (Eq. (7.8)). We solve the HJ-MTR in periodic intervals to update the safe TTR value function D^*. In our work, we solve it once per day upon receiving new forecasts similar to [141].

In summary, there are three core advantages of our method compared to classical MPC with non-linear programming. We can guarantee time-optimality in non-linear dynamics over the full time horizon when the currents are accurate. We require very low online computation to extract the gradient at each step. In case the vessel cannot reach the destination, the optimal control will attempt to minimize the terminal distance to the target, while non-linear programming would not provide us with a trajectory in such a case.

7.5 Experimental Results

We conduct experiments to ascertain that our method of using HJ-MTR with obstacles is capable of reaching the target without colliding with obstacles despite forecast errors. We simulate a large number of missions on realistic ocean currents and compare the performance of our control schema to baseline methods.

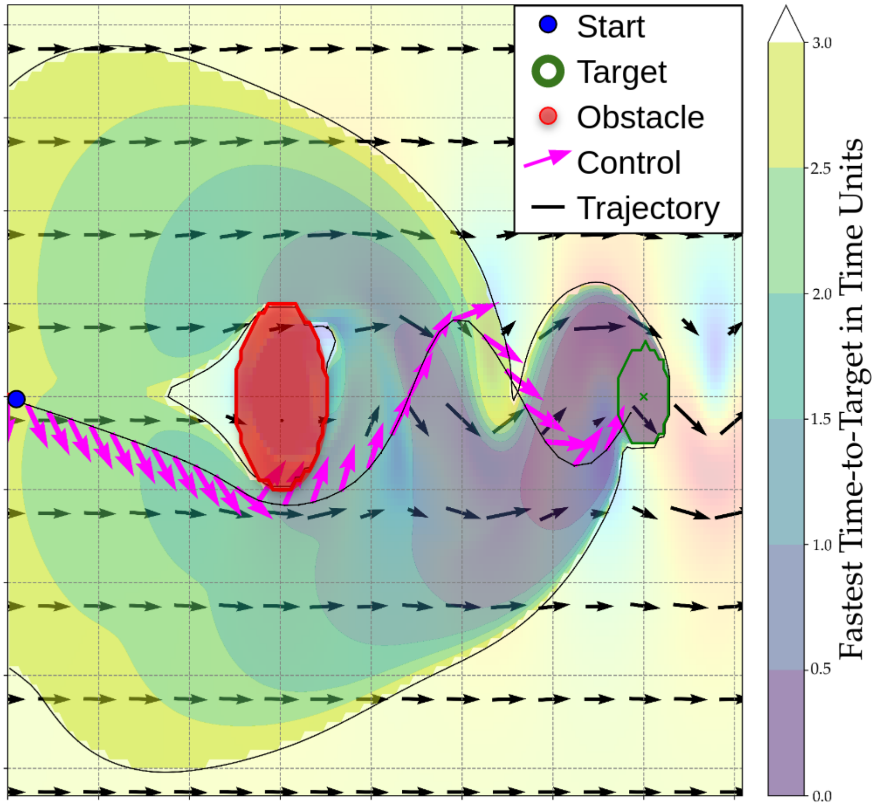


Figure 7.2: Trajectory of our method in analytical currents evading obstacle. The safe Time-to-Reach map spares out obstacle and areas where an underactuated agent is inevitably pushed into the obstacle.

Experimental Set-Up

Our experiments investigate the stranding rate and reliability of several controllers for navigating a two-dimensional ASV with fixed magnitude, holonomic actuation of $||g(\boldsymbol{u}, \boldsymbol{x}, t)||_2 = ||\boldsymbol{u}||_2 = 0.1\,\mathrm{m\,s^{-1}}$. The control input is the angle θ for steering the ASV in ocean currents $v(\boldsymbol{x}, t) \in [0\,\mathrm{m\,s^{-1}}, 1.4\,\mathrm{m\,s^{-1}}]$, which the vessel uses to reach its target region. Additionally, we describe how we ascertain the realism of our ocean forecast simulation and the creation of our obstacle sets, and the generation of a representative set of missions. Subsequently, we explain our baseline methods and evaluation metrics.

Realistic Ocean Forecast Simulation In a real-world setting, a vessel can receive the most recent forecast in regular intervals, e.g. daily, and provide it to the control methods to perform replanning. We use ocean current hindcast data from Copernicus [35] and HYCOM [20] for the region off the coast of California and Mexico between N15° and N40° and W160° and W105°.

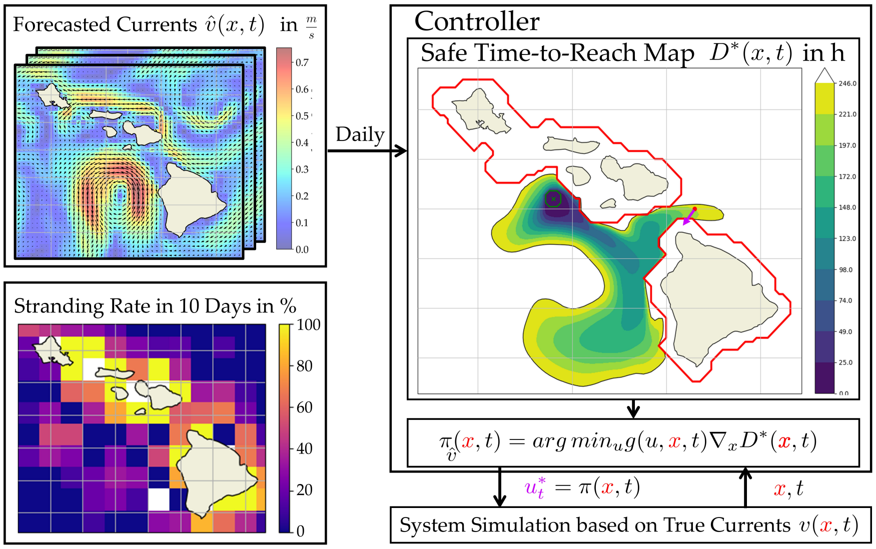

Forecasted Currents $\hat{v}(x,t)$ in $\frac{m}{s}$
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Daily
Controller
Safe Time-to-Reach Map $D^*(x,t)$ in h
246.0
221.0
197.0
172.0
148.0
123.0
98.0
74.0
49.0
25.0
0.0
$\pi_{\hat{v}}(x,t) = arg\,min_u g(u,x,t)\nabla_x D^*(x,t)$
$u_t^* = \pi(x,t)$
x,t
System Simulation based on True Currents $v(x,t)$
Stranding Rate in 10 Days in %
100
80
60
40
20
0

Figure 7.3: Our method for safe navigation is based on re-planning on two timescales: 1) compute safe Time-to-Reach map D^* daily as new forecasts become available; 2) For every timestep, e.g., 10 min, we extract $\boldsymbol{u}^*$ from $\pi(\boldsymbol{x},t)$, which is a policy equivalent to re-planning. This is necessary, as the real currents $v(\boldsymbol{x},t)$ differ from the forecasted currents $v \neq \hat{v}$, thus, we will be in a different spatial state $\boldsymbol{x}$ than anticipated. Further motivation is provided in the lower left image which displays the stranding rate of free-floating vessels over 10 days.

We simulate the system dynamics based on the hindcast as the true flow $v(\boldsymbol{x},t)$ using Copernicus hindcasts and use a series of 5 days of HYCOM hindcasts as forecasts for planning. It should be noted that, unlike HYCOM, Copernicus incorporates tidal currents into its forecasts. We want to ensure a realistic simulation of the forecast error δ. This forecast error can be measured with various metrics such as RMSE, vector correlation, and separation distance [35, 20]. In our simulations, these are on average $0.18\,\mathrm{m\,s^{-1}}$ RMSE, which is close to the validation RMSE of $0.19\,\mathrm{m\,s^{-1}}$ of the HYCOM forecast error [94]. We measured 0.63 vector correlation compared to 0.64 for HYCOM [94] and 0.62 for Copernicus [35], each measured at $t = 71\,\mathrm{h}$, with a value of 2 representing perfect correlation and 0 no correlation. Thus, our simulation setup represents realistic situations well.

Obstacles derived from Bathymetry The bathymetry data we use is the GEBCO 2022 grid [41]. It is a continuous global terrain grid with a resolution of 15″. We coarsen it to the same resolution as the current data 5′ by using the maximum in each grid cell to overestimate the elevation in each grid cell. Furthermore, we precompute a distance map corresponding to the minimum distance to obstacle areas of depth under 150 m for each cell of the grid. This is done by employing a Breadth-First-Search starting in the obstacle area with a distance of zero and exploring outwards. We use this distance as a switching condition for our baseline controller switching controller (Switch-MTR-no-Obs).

Large Representative Set of Missions We generate a representative set of 1146 start-target missions $\mathbb{M}$ with the following procedure. We uniformly sample target points $x_{\mathcal{T},center}$ spatially in the region introduced in Sec. 7.3. We reject points with a minimum distance below 0.5° to the boundary of the region or with a distance from obstacles below 0.025°. We want to generate missions with a higher risk of stranding, hence we limited the maximum distance from an obstacle to 3°. We then uniformly sample the final times t_T for a time horizon of 10 days using data from 2022. To validate that each mission is feasible for MTR-no-Obs, introduced in [141], given the true currents we calculate the backward reachable tube (BRT) using HJ-MTR from $x_{\mathcal{T}_{\text{center}}}$ at $t_{\mathcal{T}}$. We then sample a start position from the BRT so we can be sure that it is fundamentally possible for the ASV to reach the target within 5-9 days. Finally, we define the target region $\mathcal{T}$ as a circular region with radius $r_{\mathcal{T}} = 0.1°$ around $x_{\mathcal{T},center}$. Note that the introduction of obstacles increases the difficulty of the mission and renders some of these missions infeasible, as they can block a desired path.

Controllers It has been shown that navigating from start to target with an underactuated ASV can be successfully done by hitchhiking ocean currents [128, 141]. However, their controller MTR-no-Obs does not incorporate safety aspects into its value function. Our work seeks to extend its capabilities by incorporating collision avoidance into planning [81]. Therefore, we compare the performance of several controllers to the baseline presented in [141]. The first controller is the Passive Floating Controller (Floating), with an actuation of $\boldsymbol{u} = 0$. It is the same as we used in Sec. 7.3 and serves as a lower bound on performance to show how complex it is to safely complete the missions. The second controller is a reactive safety controller, meaning that it does not reason about currents for safety. Instead, it is a switching controller with switching condition being the distance to the closest obstacle. If the distance is below a threshold, the safety controller takes over and actuates with full actuation away from the obstacle, i.e. in the direction of the largest distance to the obstacles. We set the threshold to 20 km. Once the distance from the obstacles is greater than the threshold, the navigation controller MTR-no-Obs takes over again. This is a simple method that adds safety functionality to the baseline controller MTR-no-Obs. Our MTR is the controller we presented in detail in Sec. 7.4. There are two ablations of the MTR. The switching controller with obstacles (Switch-MTR) is a switching controller that uses MTR for navigation. The switching condition is the same as for the MTR-no-Obs. The last controller that we examine is the MTR with small disturbance (SmallDist-MTR), which is the MTR controller with an unrealistically low disturbance of $d = 0.05\,\mathrm{m\,s^{-1}}$. As explained in Sec. 7.4 in a realistic setting with $\delta = 0.2\,\mathrm{m\,s^{-1}}$ we cannot be robust with an actuation of only $\boldsymbol{u}_{max} = 0.1\,\mathrm{m\,s^{-1}}$.

Evaluation Metrics We define our key metric *stranding rate* as the rate of a controller entering the obstacle set $\mathcal{O}$ over the set of missions $\mathbb{M}$. We further evaluate the *reliability*, defined as the success rate of a controller over the set of missions $\mathbb{M}$ [141].

Experimental Results

We evaluate the controllers performances over $||\mathbb{M}|| = 1146$ start-target missions and run the simulation for $T_{max} = 240\,\text{h}$. If the ASV collides with an obstacle, we terminate the mission and count it as stranded, if it reaches the target region within T_{max} we count it as success, if it does neither we count it as timeout. In complex flows with forecast errors and in close proximity to obstacles, our controller MTR has a stranding rate of only 0.96%, compared to 4.71% of the baseline MTR-no-Obs (Table 7.1).

Next, we evaluate whether the stranding rate of our controllers is lower than the baseline of MTR-no-Obs in a statistically significant manner by performing a one-sided two-sample z proportion test for the other controllers. Let Γ be the stranding rate of a controller and our null hypothesis be:

$$H_0 : \Gamma_{\text{MTR-no-Obs}} = \Gamma_{\text{controller}}. \tag{7.11}$$

With the alternate hypothesis:

$$H_A : \Gamma_{\text{MTR-no-Obs}} > \Gamma_{\text{controller}}. \tag{7.12}$$

The stranding rate of both controllers is higher than MTR-no-Obs in a statistically significant way with p-values Switch-MTR-no-Obs $p = 2.6e^{-3}$, MTR $p = 3.1e^{-8}$, Switch-MTR $p = 9.3e^{-7}$, SmallDist-MTR $p = 9.5e^{-5}$). Furthermore, the success rate of MTR is not reduced by safety and even shows the highest success rate.

Table 7.1: We compare the performance of multiple controllers, the arrows indicate if high, or low is preferred. Our MTR significantly outperforms the other controllers with respect to the stranding rate. The * marks statistically significant lower stranding rates compared to the MTR-no-Obs.

Controller	Stranding Rate ↓	Success Rate ↑
MTR	**0.96%***	**37.26%**
Switch-MTR	1.31%*	37.17%
SmallDist-MTR	1.92%*	33.16%
Switch-MTR-no-Obs	2.53%*	36.82%
MTR-no-Obs	4.71%	36.91%
Floating	4.89%	2.53%

7.6 Conclusion

We note that our controllers exhibit a lower success rate than in [141]. We believe this is due to three differences in the setup. First, the time-to-target for each mission in [141] is between 20-120h with $T_{max} = 150\,\text{h}$, while our sampled time-to-target is 120-216h with

$T_{max} = 240\,\text{h}$. Hence, our missions are longer and have smaller time buffers to reach the target. In extreme cases, their missions are expected to finish in 20 h with 130 additional hours to reach the target before T_{max}, while in the worst case, our missions can have a 216 h mission with a buffer of 24 h. Second, the sampling is feasible for MTR-no-Obs without considering obstacles (Sec. 7.5). Hence missions may be unfeasible for MTR-no-Obs due to stranding on obstacles and unfeasible for the other controllers, as it may take them time to circumvent obstacles in the path. Third, we sample missions with a maximum distance to shore of 3°, exposing the vessels more to tidal currents near shore.

In this chapter, we have shown that HJ-MTR with obstacles can be used to reduce the rate of stranding even in complex flows using daily forecasts with large errors. We evaluated our method over a large set of 5-9 day start-to-target missions distributed spatially near the Coast of California, Hawaii, and the Baja California area and temporally across the year 2022 using realistic ocean currents. In our experiments, our method has achieved a stranding rate of 0.96% which is significantly lower than the baseline controllers and also has a slightly higher success rate. While we have demonstrated the ability of our method with two-dimensional ocean currents, we emphasize that it is also applicable in a three-dimensional setting such as underwater or in the air and objectives other than navigation. Furthermore, HJ-MTR is able to handle dynamic constraints [31]. However, including dynamic obstacles such as ships that move fast and change their course during the day could require a higher frequency of replanning to account for those course changes, resulting in higher computational costs.

In the future, we plan to model zones of a potential hazard, e.g. shipping lanes and garbage patches, as soft constraints, where instead of preventing entering altogether the agent should minimize the time spent therein. By reducing time in shipping lanes, an ASV could avoid many vessels. As of now, it is also unknown how underactuated ASVs would be classified under the COLREGS and if evasion is necessary or if they should stop their propulsion to be floated along other vessels [131, 132]. Getting the rotors of an ASV entangled in the garbage can render it inoperable, hence it is beneficial to avoid areas with a larger density of garbage such as the center of the GPGP, while not making the whole 1.6 million km^2 [70] of the GPGP an obstacle to avoid. We can investigate using a risk-based extension of a soft-edge and dynamic forbidden region [81, 130].

Chapter 8

Operating Multiple Robot Vessel in Flows

This chapter is based on the paper “Safe Connectivity Maintenance in Underactuated Multi-Agent Networks for Dynamic Oceanic Environments” [48], written in collaboration with Nicolas Hoischen and Claire Tomlin.

8.1 Introduction

So far we only considered single agent robotics systems in this thesis. However, in many applications, it is beneficial to operate multiple agents that communicate among each other for various purposes, such as coordination to achieve a joint objective, to ensure internet coverage [52], or to reduce the amount of communication needed to an external centralized controller. Local communication often relies on limited-range systems, for example sonar or radar, requiring agents to stay close to each other for network connectivity (see Fig. 8.1).

In Parts II and III we demonstrated that an agent can achieve its objective, be it station-keeping of balloons or navigating in the oceans, with very little energy by *going with the flow*: drifting on flows and using its small actuation strategically to nudge itself into beneficial flows.

In this chapter, we want to bring these developed controllers to multi-agent networks operating in complex flows. The DP approaches we have developed provide a value function that yields optimal individual agent controls for an arbitrarily high number of agents at the cost of a cheap gradient computation. This is especially powerful for multi-agent problems where the objective can be decomposed into the sum of independent single agent objectives; for example, in floating seaweed farms, the seaweed growth of each agent does not depend on the growth of other agents [60]. Given such individual agent performance controllers, our goal is to develop a method that ensures the safe interaction of multiple underactuated

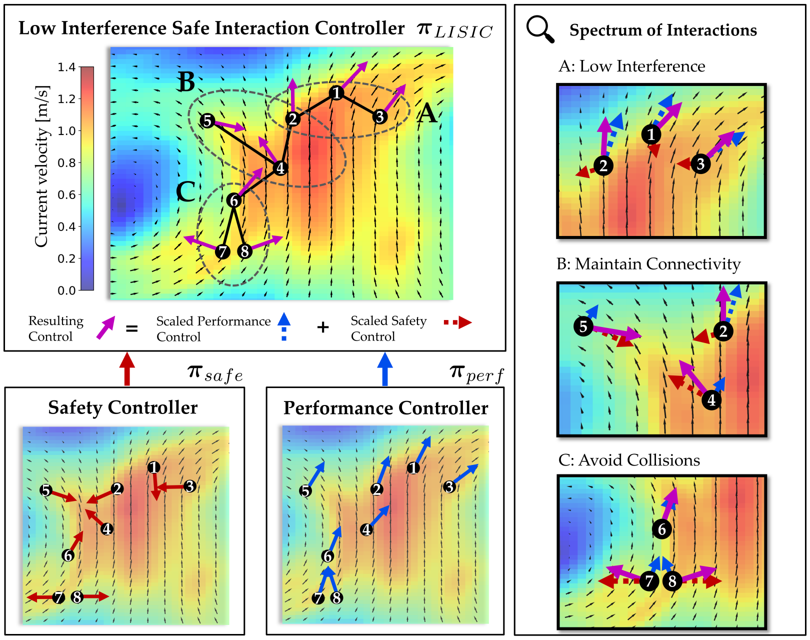


Figure 8.1: Our LISIC policy blends a single agent performance control input with a flocking-based safety control input to avoid connectivity losses and collisions in a multi-agent network, while minimally interfering with the performance objective of each agent. This ensures safe performance in ocean environments with strong ocean currents affecting the low-powered agents.

agents operating in complex flows. Specifically, safe interaction in terms of maintaining network connectivity and avoiding collisions.

From the control perspective, this is challenging because of two key reasons. First, in the underactuated setting disconnections are sometimes unavoidable as the non-linear, time-varying flows can push agents in opposing directions. The safe interaction controller needs to be resilient and recover connectivity after it was lost. Second, constraint satisfaction needs to be traded off intelligently with the performance objective of each agent, as they can often be in conflict. For example, a time-optimal controller for an agent would prefer to stay in strong flows, whereas the safe interaction controller needs to trade this off with maintaining connected to other agents (Fig. 8.1).

Our insight is that the structure of this problem allows us to tackle the multi-agent problem with three different controllers in a Hierarchical Control of Multi-Agent-Systems (H-MAS) approach (Fig. 8.1).

The literature contains various approaches that are relevant for this work. H-MAS can be viewed from the perspective of hierarchical reinforcement learning [103], which shares substantial similarities with our approach of breaking down complex tasks into two more manageable subtasks. In H-MAS, agents are organized into multiple levels of hierarchy, with higher-level agents having more authority and control over lower-level agents, designated as followers [33]. For instance, [83] solves path planning and ocean vehicles coordination separately with a hierarchical leader-follower structure. Graph methods provide a solid theoretical foundation for analyzing connectivity properties in distance-based control applications [21]. Our interest is drawn to flocking techniques incorporating a single agent navigation task via a dynamic virtual leader, the γ-agent [100], which maintain connectivity by influencing the agents' behavior to follow the movement of their neighbors, while avoiding collisions. Extensions to multiple virtual leaders are proposed in [126, 125, 77] which hold promise for the generalization of virtual leaders as performance controllers. While most of the flocking schemes generally assume simple double integrator dynamics, adaptive flocking also emerged to handle non-linear uncertain dynamics [148, 140, 151, 155]. More recently, Model Predictive Control (MPC) has been successful in ensuring connectivity and collision-free operation in Multi-Agent-Systems (MAS) [18].

MPC approaches to connectivity or safety [18, 139], in addition to being computationally intensive, often assume position-invariant dynamics, which do not hold in dynamic ocean environments. This motivates a cheaper reactive strategy approach in this context. Determining the follower position with respect to the leader in [83] requires solving an additional optimization problem to achieve formation control, adding complexity and feasibility issues for underactuated agents. On the other hand, adaptive and robust flocking schemes rely on overactuated agents to overcome disturbances. Adaptive flocking in underactuated systems has been considered using a Radial Basis Function Neural Network (RBF-NN) to approximate uncertain non-linear dynamics [106], but can easily overfit the training data, which is a significant concern given the sparse and stochastic nature of ocean data. Existing literature on flocking has primarily focused on tracking a reference trajectory of a virtual leader or multiple virtual leaders. However, in complex flows, an optimal feedback control policy leads to significantly better results than tracking a reference trajectory [141].

To address the above shortcomings, we propose a safe interaction control policy Low Interference Safe Interaction Controller (LISIC), blending a single agent performance controller with a flocking-inspired safe controller. In addition to ensuring the safe operation of the networked agents in terms of avoiding collisions and maintaining connectivity, our approach also enables recoveries in case of connectivity failures. The dynamics of communication and information sharing could then later be handled on another level with a Plug-and-Play control scheme [1], but is not within the scope of this work. In Sec. 8.2, we introduce important background and relevant metrics to measure connectivity in terms of time and network topology, as well as a single agent performance trade-off. We show our results theoretically in Sec. 8.3 and Sec. 8.4. Finally, we assess the performance of our approach by conducting a large-scale empirical evaluation with agents that are underactuated in the sense that their propulsion is smaller than the surrounding flows.

8.2 Problem Formulation

In this section we first describe the systems dynamics and give a brief summary of connectedness in communication graphs. Then we define our problem statement and the metrics we use to measure constraint violation.

System Dynamics

We consider a swarm of N agents and use $\mathcal{V}$ to describe the set of all agents. The dynamics for each agent $i \in \mathcal{V}$ are given by:

$$\dot{\boldsymbol{q}}_i = v(\boldsymbol{q}_i, t) + g(\boldsymbol{q}_i, \boldsymbol{u}_i, t),\ t \in [0, T] \tag{8.1}$$

$\boldsymbol{q}_i \in \mathbb{R}^n$ denotes the position of agent i in the n dimensional state space, where $n = 2$ for a surface vessel on the ocean. The movement of the agent i depends on the drift of the surrounding flow $v(\boldsymbol{q}_i, t)$ and the bounded control $\boldsymbol{u}_i \in \mathbb{U} \in \mathbb{R}^{n_u}$ where n_u is the dimensionality of the control. Let the agent trajectory resulting from Eq. (8.1) be described by $\boldsymbol{\xi}_i$ with $\boldsymbol{\xi}_i(t)$ the state $\boldsymbol{q}_i$ at t. For the global system of all N agents, we use $\boldsymbol{q} = [\boldsymbol{q}_1^\top, \boldsymbol{q}_2^\top, \dots, \boldsymbol{q}_N^\top]^\top$, $\boldsymbol{u} = [\boldsymbol{u}_1^\top, \boldsymbol{u}_2^\top, \dots, \boldsymbol{u}_N^\top]^\top$, and $\boldsymbol{\xi}$ respectively to describe the state, control, and trajectory.

Communication Graph Preliminaries

The network topology of our Multi-Agent-Systems with state $\boldsymbol{q}$ can be represented with a graph abstraction in order to model interactions among agents. The communication graph $G(t)$ can be built from a set of finite vertices $\mathcal{V} = \{1, 2 \dots N\}$ representing individual agents and a time-varying set of edges $\mathcal{E}(t) \subseteq \{(i, j) \in (\mathcal{V} \times \mathcal{V}), j \neq i\}$ representing direct communication between agents. We focus on *undirected* graphs implying that information can flow between agents in both directions. We further assume that only neighbors that are spatially close with respect to a distance measure $d(\boldsymbol{q}_i, \boldsymbol{q}_j)$ can communicate directly with each other. Given an upper communication threshold R_{com}, the pair of vertices i, j is connected by an edge $d(\boldsymbol{q}_i, \boldsymbol{q}_j) < R_{com} \iff (i, j) \in \mathcal{E}(t)$. The graph $G(t)$ is said to be *connected* if there exists an undirected path between every pair of distinct vertices. We can analyze the connectivity of the graph with its Laplacian matrix L which is a symmetric and positive semidefinite matrix based on the adjacency and the degree matrix that we define next. The adjacency matrix $A(t)$ is a binary matrix $n \times n$ that encodes which vertices are connected to each other $A(G(t)) = [a_{ij}(t)] \in \{0, 1\}$ with $a_{ij}(t) = 1 \iff (i, j) \in \mathcal{E}(t)$.

The valency or degree of a vertex i is denoted by $deg(i, t)$ and represents the number of its incident edges, which is the row sum of the adjacency matrix $deg(i, t) = \sum_{j=1}^{N} A_{ij}(t)$. The degree matrix D is then defined as the diagonal matrix $D(G(t)) = \text{diag}\,(deg(i, t))$. The Laplacian matrix L can then be inferred as $L(G(t)) = D(G(t)) - A(G(t)$, which is a symmetric and positive semidefinite matrix. The eigenvalues of L let us measure the connectivity of the graph. In particular, the second smallest eigenvalue $\lambda_2(L(G(t))$ is commonly called the algebraic connectivity or Fiedler value of the network and captures the robustness of the network to link failures. The graph $G(t)$ is connected only if and only if it is strictly positive, that is, $\lambda_2(L(G(t))) > 0$ [93, 18, 149].

Problem Statement

We focus on multi-agent problems where the joint objective is the sum of independent objectives $\mathbb{P}_i$ which can be sketched as:

$$\begin{aligned}
\min_{\pi} \quad & \sum_{i=1}^{N} \mathbb{P}_i(\boldsymbol{\xi}_i, \boldsymbol{u}_i(\cdot)) && (8.2)\\
\text{s.t.} \quad & \forall\, t \in [t_0, T] \\
& \dot{\boldsymbol{\xi}}(t) = v(\boldsymbol{\xi}(t), t) + \pi(\boldsymbol{\xi}(t)) \quad \text{global dynamics Eq. (8.1)} \\
& d(\boldsymbol{\xi}_i(t), \boldsymbol{\xi}_j(t)) > R_{coll} \quad (i,j) \in V \times V, i \neq j && (8.3)\\
& \lambda_2(L(G(\boldsymbol{\xi}(t), R_{com}))) > 0 && (8.4)
\end{aligned}$$

The goal is to find the control policy π. The agents are coupled in only two constraints: the collision constraint (Eq. (8.3)) where $d(\boldsymbol{q}_i, \boldsymbol{q}_j)$ represents the distance between agent i and j and R_{coll} the minimum safe distance, and second, in maintaining a graph where all agents are connected to each other based on the communication range R_{com} (Eq. (8.4)). Our insight is that in these settings we can decompose the problem and handle the objectives and constraints on different levels with (1) a performance controller π_{perf} for each agent, (2) a safety controller π_{safe}, and (3) a low-interference safe interaction controller π_{LISIC} trading off the two (Fig. 8.1).

The performance controller of an agent i minimizes its $(\pi_{perf})_i = \arg\min_{\pi_i} \mathbb{P}_i(\boldsymbol{\xi}_i, \boldsymbol{u}_i(\cdot))$ only considering its own dynamics $\boldsymbol{q}_i$ (Eq. (8.1)). π_{perf} can be an arbitrary control policy, from a fixed control signal to a feedback controller based on learning or dynamic programming (Sec. 8.5). In challenging settings like ours with nonlinear, time-varying dynamics, it is easier to design single agent feedback controllers than solving the coupled multi-agent problem, e.g. for time-optimal navigation, reference tracking, or optimizing seaweed growth [60]. The safety controller π_{safe}, determines the control for all agents to ensure that the interaction constraints are satisfied (Eq. (8.3), (8.4)). Lastly, based on the control inputs $\boldsymbol{u}_{perf}$ and $\boldsymbol{u}_{safe}$ from the respective policies, the safe interaction controller decides the agents final control inputs $\boldsymbol{u} = \pi_{LISIC}(\boldsymbol{u}_{perf}, \boldsymbol{u}_{safe})$. To achieve good performance, the safe interaction controller should not interfere too much with $\boldsymbol{u}_{perf}$ while still ensuring connectivity and avoiding collisions.

In this work, we focus on designing π_{safe} and π_{LISIC} for an arbitrary π_{perf}. In Sec. 8.4 we prove that our method guarantees constraint satisfaction under certain conditions on the maximum magnitude of the control $\mathbb{U}$ and the flow field velocities v across the agents. Additionally, we test how our method performs in realistic ocean currents where these conditions are not always met and quantify constraint violations with the metrics we define below.

Constraint Violation Metrics

As in some settings it is impossible to guarantee constraint satisfaction, we now define the metrics we use to evaluate how much the constraints are violated in our experiments in Sec. 8.5. A collision occurs between any of the agents in the swarm if $\exists t \in [0, T]$ at which Eq. (8.3) is violated. We denote this with the collision indicator $\mathbb{I}_{coll} \in \{0, 1\}$.

To measure various aspects of losing connectivity, we use three metrics. First, for a binary measure, if disconnections occur, we define the disconnection indicator $\mathbb{I}_{disconn} \in \{0, 1\}$ which is 1 if $\exists t \in [0, T]$ at which Eq. (8.4) is violated and zero otherwise.

As we are also interested in the networks' robustness against connectivity losses or link failures, we additionally measure the minimum Fiedler value over time, the higher the more robust the communication network (Sec. 8.2):

$$\lambda_2^{min} = \min_{t \in [0,T]} \lambda_2(L(G(\boldsymbol{\xi}(t), R_{com})) \tag{8.5}$$

Lastly, it often matters for how long an agent is isolated from all other agents. Therefore, we introducing a new measure that we call IPM

$$\text{IPM} = \frac{1}{T} \int_{t=0}^{T} M(deg(i, t) = 0)\, dt \tag{8.6}$$

where $M(deg(i, t) = 0)$ counts the number of disconnected vertices, which corresponds to the number of zeros in the diagonal of the graph degree matrix $D(G(\boldsymbol{\xi}(t), R_{com})$ (Sec. 8.2).

In our simulations, we empirically compare the constraint violation of different controllers over a large, representative set of missions $\mathbb{M}$ by evaluating the collision rate $\mathbb{E}_{\boldsymbol{q}(t_0), t_0 \sim \mathbb{M}}[\mathbb{I}_{coll}]$, the disconnection rate $\mathbb{E}_{\boldsymbol{q}(t_0), t_0 \sim \mathbb{M}}[\mathbb{I}_{disconn}]$, as well as the distributions of IPM and λ_2^{min}. In our setting where the performance objectives $\mathbb{P}_i$ are the minimum time-to-target for each agent i, the connectivity constraint often leads to a trade-off with the performance objective. Hence, we also quantify the degradation of the performance controller by quantifying the minimum distance the swarm center got to the target area $\mathcal{T}$ during the mission time $t \in [0, T]$ as $d_{min}(\mathcal{T})$.

8.3 Method

Our method tackles the multi-agent problem with a hierarchical control approach. The low interference safe interaction controller π_{LISIC} ensures performance and safe control based on an arbitrary performance controller π_{perf} and a safety controller π_{safe} (see Fig. 8.1). We first introduce our flocking-inspired safety controller based on potential functions and then detail our design for π_{LISIC}. For ease of understanding, we assume holonomic actuation $g(x, u, t) = u$, but note that the method can be generalized.

Flocking-Inspired Safety Controller

The sole objective of the safety controller is to ensure adequate distances between agents such that their communication graph is connected and that they do not collide with each other. We are inspired by the reactive flocking approaches for achieving ideal inter-agent distances without prescribing a formation. Hence, we design our safety controller π_{safe} based on the gradients of a potential function ψ.

To explain the principle, let us first focus on just two agents i and j that are connected and are at an inter-agent distance $\left\|\boldsymbol{q}_{ij}\right\|_2 = \left\|\boldsymbol{q}_i - \boldsymbol{q}_j\right\|_2$. Consider the following bowl shaped potential function

$$\psi_{\text{connected}}(\left\|\boldsymbol{q}_{ij}\right\|_2) = \frac{\kappa R_{com}}{\left\|\boldsymbol{q}_{ij}\right\|_2 \left(R_{com} - \left\|\boldsymbol{q}_{ij}\right\|_2\right)}, \tag{8.7}$$

where $\kappa > 0$ is a tuning factor to adjust the bell shape (see left of R_{com} in Fig. 8.2). Let the safety controllers for i be $\nabla_{\boldsymbol{q}_i}\psi_{\text{connected}}(\left\|\boldsymbol{q}_{ij}\right\|_2)$ and for j $\nabla_{\boldsymbol{q}_j}\psi_{\text{connected}}(\left\|\boldsymbol{q}_{ji})\right\|_2 = -\nabla_{\boldsymbol{q}_i}\psi_{\text{connected}}(\left\|\boldsymbol{q}_{ij}\right\|_2)$. When those two agents are getting too close $\left\|\boldsymbol{q}_{ij}\right\|_2 \to 0$ the potential $\psi(\left\|\boldsymbol{q}_{ij}\right\|_2)$ goes to infinite, so the gradient-controllers are a strong repulsive force that pushes them away from each other. Conversely, when the two connected agents are at risk of losing their communication link $\left\|\boldsymbol{q}_{ij}\right\|_2 \to R_{com}$ then $\psi(\left\|\boldsymbol{q}_{ij}\right\|_2) \to \inf$ which means the gradient-controllers result in a strong attractive force that brings them closer again. For multiple agents, the control becomes the sum of gradient potential terms of the other agents, and the magnitude of the gradients helps prioritize the critical inter-agent distances $\boldsymbol{q}_{ij}$.

When the agents are disconnected, which is sometimes unavoidable in underactuated settings where strong flows push them apart, we want them to reconnect. Therefore, our final potential function is augmented with a second term that accounts for agents outside their communication range R_{com} to encourage achieving connectivity between disconnected agents [146]. This results in our final potential function $\psi(z) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ that is also visualized in Fig. 8.2:

$$\begin{aligned}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right) = \sigma_{ij} & \underbrace{\frac{\kappa R_{com}}{\left\|\boldsymbol{q}_{ij}\right\|_2 \left(R_{com} - \left\|\boldsymbol{q}_{ij}\right\|_2\right)}}_{\text{for connected agents}} \\ + \left(1 - \sigma_{ij}\right) & \underbrace{\sqrt{\left(\left\|\boldsymbol{q}_{ij}\right\|_2 - R_{com} + \epsilon\right)}}_{\text{for disconnected agents}}.\end{aligned} \tag{8.8}$$

where σ_{ij} is an edge indicator similar to a_{ij} in Sec. 8.2 but with an hysteresis parameter ϵ defined below in (8.9). Hence, $\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$ switches between two terms to determine whether the pair of agents (i, j) is within the communication range $(\sigma_{ij} = 1)$ or disconnected $(\sigma_{ij} = 0$ $)$[146]. The hysteresis mechanism avoids constant switching of the dynamical network with multiple agents for edges close to R_{com} and helps preserve connectivity in reactive control schemes [54].

$$\sigma_{ij}[t] = \begin{cases} 0, & \text{if } \left(\left(\sigma_{ij}\left[t^{-}\right] = 0\right) \cap \left(\left\| \boldsymbol{q}_{ij} \right\| \geq R_{com} - \varepsilon \right) \right) \\ & \cup \left(\left(\sigma_{ij}\left[t^{-}\right] = 1\right) \cap \left(\left\| \boldsymbol{q}_{ij} \right\| \geq R_{com} \right) \right), \\ 1, & \text{if } \left(\left(\sigma_{ij}\left[t^{-}\right] = 1\right) \cap \left(\left\| \boldsymbol{q}_{ij} \right\| < R_{com} \right) \right) \\ & \cup \left(\left(\sigma_{ij}\left[t^{-}\right] = 0\right) \cap \left(\left\| \boldsymbol{q}_{ij} \right\| < R_{com} - \varepsilon \right) \right), \end{cases} \tag{8.9}$$

where $\epsilon > 0$ is the switching threshold inducing a hysteresis in the process of adding new links to the flock.

Our approach yields a relatively low attraction force for agents far outside of their communication range. This is a design choice in the context of *underactuated* agents in a dynamic oceanic environment, where remote flock members can experience strong divergent flows and direct connectivity may be infeasible or undesirable to achieve.

The final safe interaction controller for each agent i with maximum propulsion $U_{max,i}$ is then defined as

$$(\pi_{safe})_i(\boldsymbol{q}) = -\frac{\sum_{j=1}^{N} \nabla_{\boldsymbol{q}_i} \psi\left(\left\| \boldsymbol{q}_{ij} \right\|_2 \right)}{\left\| \sum_{j=1}^{N} \nabla_{\boldsymbol{q}_i} \psi\left(\left\| \boldsymbol{q}_{ij} \right\|_2 \right) \right\|_2} U_{max,i} \tag{8.10}$$

Low Interference Safe Interaction Controller

For our π_{LISIC} that trades off the performance inputs $\boldsymbol{u}_{perf}$ with the safety inputs $\boldsymbol{u}_{safe}$ we propose an approach that weights these control vector inputs for each agent i depending on the risk of losing connectivity or colliding.[1]

$$\boldsymbol{u}_i = (\pi_{LISIC})_i(\boldsymbol{u}_{perf}, \boldsymbol{u}_{safe}) = c_i^{(1)} \boldsymbol{u}_i^{safe} + c_i^{(2)} \boldsymbol{u}_i^{perf}, \; \forall i \in \mathcal{V} \tag{8.11}$$

where $c_i^{(1)}$ and $c_i^{(2)}$ are weighting factors. Note that $\boldsymbol{u}_i^{safe} = (\pi_{safe})_i(\boldsymbol{q})$ depends on the other agents positions to guarantee safe interactions.

When collisions or connectivity losses are imminent, $\boldsymbol{u}_i$ should be able to rapidly tend to $\boldsymbol{u}_i^{safe}$ to prioritize safety over performance, i.e. $c_i^{(1)} \to 1$ and $c_i^{(2)} \to 0$ (Fig. 8.1 B, C). Conversely, when the network is well connected and there is low danger of collisions, $\boldsymbol{u}_i$ should align with $\boldsymbol{u}_i^{perf}$ to have low interference with the agent's performance control, i.e. $c_i^{(1)} \to 0$ and $c_i^{(2)} \to 1$ (Fig. 8.1 A).

Hence, we defined a weighting function $\alpha(\boldsymbol{q}) : \mathbb{R}^N \to [0, 1]$ such that $c_i^{(1)} = \alpha(\boldsymbol{q})$ and $c_i^{(2)} = 1 - \alpha(\boldsymbol{q})$. 8.5. This function $\alpha(\boldsymbol{q})$ measures the urgency of $\boldsymbol{u}_i$ to converge to $\boldsymbol{u}_i^{safe}$ and we define it as

$$c_i^{(1)} = \alpha \left(\left\| \sum_{j=1}^{N} \nabla_{\boldsymbol{q}_i} \psi\left(\left\| \boldsymbol{q}_{ij} \right\|_2 \right) \right\|_2 \right) \tag{8.12}$$

[1] For ease of understanding we derive the controller assuming holonomic actuation g(x,u,t) = u but note that it can be extended to more general actuations.

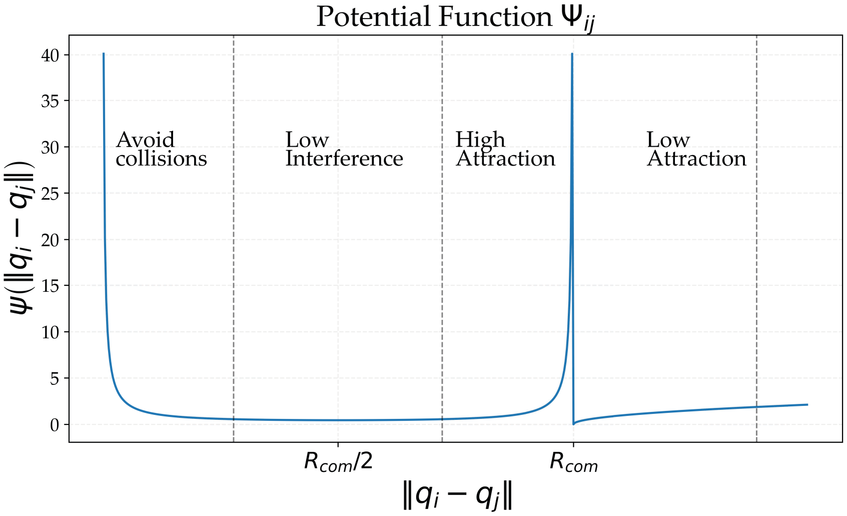


Figure 8.2: Augmented potential function, with two terms to account for agents within the communication range and agents outside the communication range R_{com}. A high κ parameter is shown to increase the steepness of the slope around $\frac{R_{com}}{2}$, depending how achieving the exact ideal distance is valued

The function α can be thought as a monotonically increasing safety activation function taking values between $[0, 1]$ depending on the (unbounded) magnitude of it's argument. From the definition of $\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$ in (8.8), $\lim_{\left\|\boldsymbol{q}_{ij}\right\|_2 \to 0} \psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right) = \infty$ and $\lim_{\left\|\boldsymbol{q}_{ij}\right\|_2 \to R_{com}} \psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right) = \infty$. Hence, in critical situations $\left\|\sum_{j=1}^{N} \nabla_{\boldsymbol{q}_i} \psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)\right\|_2$ becomes very large so that $c_i^{(1)}$ saturates to 1 and $c_i^{(2)}$ to 0, thus prioritizing the network safety for the concerned agents $i \in \mathcal{V}$ i.e. $\boldsymbol{u}_i \to \boldsymbol{u}_i^{safe}$, over the individual objectives of each agent $\boldsymbol{u}_i^{perf}$.

In other words, $\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$ has a contractivity property for agent inter-distances at the boundary of the safe set, defined by 0 and R_{com}, similarly to Control Barrier Functions (CBFs) [2]. With this design, we ensure that agents moving from a disconnected state $\sigma_{ij} = 0$ to a connected state $\sigma_{ij} = 1$, experience a strong attracting gradient $\boldsymbol{u}_i^{safe}$ to avoid escaping the communication range again. From Fig. 8.2 it is also clear that when the network is close to being ideally connected, the gradient norm of the potential function $\left\|\sum_{j=1}^{N} \nabla_{\boldsymbol{q}_i} \psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)\right\|_2$ is low, so that the agent is' control input is dominated by the performance controller since $c_i^{(1)} \to 0$ and $c_i^{(2)} \to 1$.

8.4 Theoretical Analysis

In this section, we analyze the conditions under which our safe interaction controller is able to maintain connectivity and avoid collisions. Here we sketch out the proof, which can be found in full length in the Appendix A and the Master Thesis of Nicolas Hoischen [47].

Energy-based analysis and LaSalle's invariance principle are commonly utilized to obtain analytical guarantees for flocking. These methods establish the stability of the system and demonstrate that the flock converges to a lattice structure while preventing inter-agent collisions, as demonstrated in [100]. The collective structural dynamics can be derived using a moving referential [100] with respect to the flock centroid $\boldsymbol{q}_c$. The relative coordinates are given by $\tilde{\boldsymbol{q}}_i = \boldsymbol{q}_i - \boldsymbol{q}_c$ and $\tilde{\boldsymbol{q}}_{ij} = \tilde{\boldsymbol{q}}_i - \tilde{\boldsymbol{q}}_j = \boldsymbol{q}_{ij}$. Therefore, $\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right) = \psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|_2\right)$, and the *total tension energy* or potential energy for the structural dynamics in the relative coordinates yields

$$H(\tilde{\boldsymbol{q}}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|_2\right) \tag{8.13}$$

A possible approach, although conservative, is to show that a global tension energy decrease of the system $\dot{H} = \sum_{i=1}^{n}\dot{H}_i \leq 0$ can be achieved by guaranteeing local tension energy decrease $\forall i \in \mathcal{V}$. Assume that $G(t)$ switches at time t_l for $l = 0, 1, 2\ldots$ and $\dot{H} \leq 0$ on each $[t_l, t_{l+1})$. Then, at the switching time k, $H(t_k) = H(t_k^-) + \psi\left(\|R_{com} - \epsilon\|\right)$ [146]. As the graph topology becomes fixed after a certain time and only a finite number of maximum edges can be added, the energy can be shown to be bounded for any subsequent time. The time-derivative of H_i along the trajectory of agent i yields

$$\dot{H}_i = \dot{\tilde{\boldsymbol{q}}}_i^{\top}\sum_{\substack{j=1\\ j\neq i}}^{N}\nabla_{\tilde{\boldsymbol{q}}_i}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|_2\right) \tag{8.14}$$

where we exploited the relation $\nabla_{\boldsymbol{q}_i}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right) = -\nabla_{\boldsymbol{q}_j}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$. Substituting $\dot{\tilde{\boldsymbol{q}}}_i = \dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_c$ into Eq. (8.14) and by applying the Cauchy–Schwarz inequality after some iterations, we obtain that $\dot{H}_i \leq 0$ holds if:

$$\left\|c_i^{(2)}\boldsymbol{u}_i^{perf}(\boldsymbol{q}_i) + v\left(\boldsymbol{q}_i\right) - \operatorname{Ave}\left(\dot{\boldsymbol{q}}_{Ni}\right)\right\|_2 \leq c_i^{(1)}U_{max,i} \tag{8.15}$$

where $\operatorname{Ave}(\cdot)$ denotes the average and the set $N_i = \mathcal{V}\setminus\{i\}$ the neighboring agents of i. The dynamics of the other agents are also defined by their surrounding flow and their individual control inputs such that $\operatorname{Ave}\left(\dot{\boldsymbol{q}}_{Ni}\right) = \operatorname{Ave}\left(v(\boldsymbol{q}_{Ni})\right) + \operatorname{Ave}\left(\boldsymbol{u}_{Ni}\right)$.

The agents do not necessarily need to be overactuated despite strong flows to achieve a local energy decrease ($\dot{H}_i \leq 0$). If the currents experienced by agent i are of magnitude and direction similar to the average current experienced by neighboring agents, then $v\left(\boldsymbol{q}_i\right)$ compensates $\operatorname{Ave}\left(v(\boldsymbol{q}_{Ni})\right)$ and Eq. (8.15) can be fulfilled even if $\left\|v\left(\boldsymbol{q}_i\right)\right\|_2 > U_{max,i}$. The neighboring flocking control inputs $\operatorname{Ave}\left(\boldsymbol{u}_{Ni}\right)$ also help to account for the current difference term $v\left(\boldsymbol{q}_i\right) - \text{Ave}\left(v(\boldsymbol{q}_{Ni})\right)$. If $\|v\left(\boldsymbol{q}_i\right) - \text{Ave}\, v(\boldsymbol{q}_{Ni})\|_2 \gg U_{max,i}$, satisfying Eq. (8.15) becomes challenging, which can occur if agents experience strong divergent currents. Under these assumptions, we can show that $\dot{H} \leq 0$, which allows us to bound the maximum energy and apply LaSalle's Invariance Principle [146], [140], thus ensuring that no collisions or disconnections occur, since $\psi(\left\|\boldsymbol{q}_{ij}\right\|_2) \to \infty$ when $\left\|\boldsymbol{q}_{ij}\right\|_2 \to 0$ or $\left\|\boldsymbol{q}_{ij}\right\|_2 \to R_{com}$. Condition Eq. (8.15) is sufficient but not necessary to guarantee $\dot{H} < 0$, as negative local energies can compensate for positive ones.

8.5 Simulation Study

In the following section, the proposed flocking control scheme is evaluated on realistic ocean currents. We use Multi-Time HJR as a single agent performance controller, since it generates one value function yielding the time-optimal control for all agents at once [141].

Experimental Set-Up

We study the effectiveness of different controllers in maneuvering a two-dimensional ASV with holonomic actuation of fixed thrust magnitude $\|u\|_2 = 0.1$ m/s. The control input in this context is the thrust angle θ. We consider a group of identical $n = 30$ ASVs with omnidirectional communication capabilities, navigating in strong ocean currents $v(\boldsymbol{q}, t) \in [0.3m/s, 2m/s]$, where each agent aims to reach a common predefined target, so that the group objective can be considered as the center of the flock reaching the target. In the following, we describe the creation of simulation experiments in a realistic ocean environment and how we obtain a large set of missions to best illustrate trade-offs between single-agent performance and flock connectivity maintenance.

Realistic Simulation of Ocean Conditions We focus on the Gulf of Mexico region (Fig. 8.3), as it presents interesting and challenging currents. As explained in Chapter 2.2, we employ two sources of ocean current data that we refer to as HYCOM *hindcast* [20] and Copernicus hindcast [35] which we use as *forecast* for realistic scenarios. The ocean current data and the forecast error introduced are particularly relevant to the multi-time HJ reachability controller, as it uses the forecasted currents to plan on, which impacts the performance of the time-optimal nominal controller. We propose two settings to investigate our approach, namely (a) performance HJR planning on *hindcast* and multi-agent simulation on *hindcasts* (HC-HC) and (b) performance HJR planning on *forecast* and multi-agent simulation on *hindcasts* (FC-HC). The first allows us to assess performance in an idealized setting where true flows are known, whilst the second reflects a realistic application in dynamic ocean environments.

Large Representative Set of Missions We assume that all agents start a navigation mission to a target region T at the same time t_0. The navigation objective is to drive the

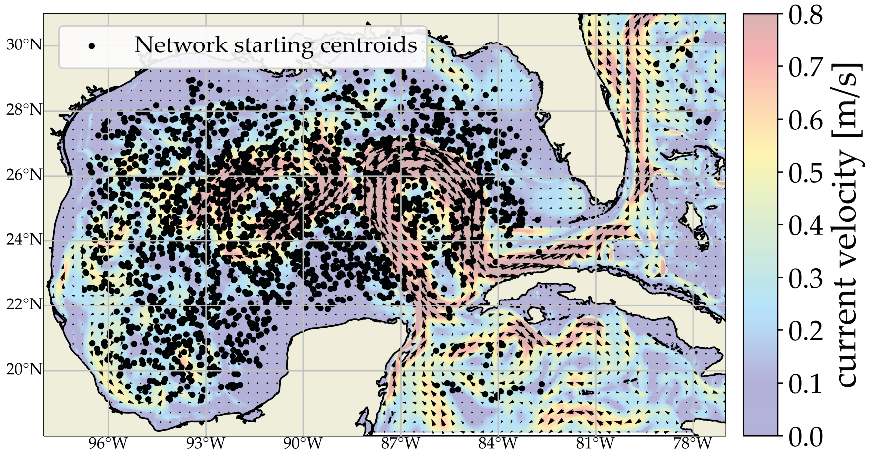


Figure 8.3: We sample a large set of missions $|\mathbb{M}| = 1000$ in the Gulf of Mexico that are spatially and temporally representative of realistic scenarios.

ASVs from their start states $(\boldsymbol{q}_1(t_0)\ldots\boldsymbol{q}_n(t_0))$ to T in the maximum allowed time $T_{timeout}$. The target T is defined as a circular region with center coordinates $\boldsymbol{q}_T$ and fixed radius $r_T = 0.1°$ around it. To obtain a diverse set of missions $\mathbb{M}$, the starting times t_0 are uniformly sampled between April 2022 and December 2022. $T_{timeout}$ is set to 144 h, and the starting points are sampled so that agents can reach the target in $[72, 144]h$. This ensures that missions are feasible with true flow data and temporally representative of realistic scenarios. To prevent stranding, we impose a minimum distance of 111km between the target area and the land and a minimum distance of 40km between each ASV's initial position and the land. We generated a total of $|\mathbb{M}| = 1000$ missions of initially connected and collision-free networks (see Fig. 8.3).

Baseline controllers

We build on recent work that proposed a reliable multi-time HJR controller for underactuated agents utilizing complex flows [141]. This approach directly extends to multiple agents with little extra computation, and the feedback controller for agent i can be obtained from an optimal value function $\mathcal{J}^*$ at time t as $\boldsymbol{u}_i(t)^* = \arg\min_{u_i \in \mathbb{U}} g(\boldsymbol{q}_i, \boldsymbol{u}_i, t) \cdot \nabla_{\boldsymbol{q}_i} \mathcal{J}^*(\boldsymbol{q}_i, t)$. All evaluated controllers use the multi-time HJR formulation for single-agent performance control. Our baseline scheme, called HJR-Baseline, involves each agent only utilizing its time-optimal performance control HJR without considering multi-agent interactions. Thus our baseline also provides a good likelihood estimation of collisions and communication losses if each agent were to rely solely on its performance control. In addition, we define a second baseline controller from [108], denoted as HJR-Reactive. This controller operates in three modes: achieveConnectivity, maintainConnectivity, and GoToGoal, which are selected based on the ASVs' relative positions. The maintainConnectivity and GoToGoal modes employ a general navigation function for each agent, which we instantiate to our HJR performance controller. This approach is easily integrated with the time-optimal control HJR, and the reactive control term can be implemented in a decentralized manner.

Finally, we define our LISIC as HJR-Flocking with the safe interaction controller Eq. (8.10), where we also implement HJR as the single agent performance controller $\boldsymbol{u}_i^{perf}$. The trade-off between each agent's navigational objective and the safe network interaction can be tuned with two parameters. First, the shape of the potential function (Fig. 8.2) can be more or less flat around the ideal distance $R_{com}/2$. In this application, we set $\kappa = 2$. Furthermore, we now detail our weighting scheme for $c_i^{(}1)$ and $c_i^{(}2)$ via the definition of α Eq. (8.12) as a softmax-like function

$$c_1^{(i)} = \frac{e^{\left\|\sum_{j=1}^{n} \nabla_{\boldsymbol{q}_i}\psi\left(\|\boldsymbol{q}_{ij}\|_2\right)\right\|_2}}{e^{\left\|\sum_{j=1}^{n} \nabla_{\boldsymbol{q}_i}\psi\left(\|\boldsymbol{q}_{ij}\|_2\right)\right\|_2} + e^{\rho}}, \quad \forall i \in \mathcal{V}. \tag{8.16}$$

where the parameter $\rho \geq 0$ can be adjusted to achieve faster saturation of the gradient term of the potential function $\boldsymbol{u}_i^{safe}(\boldsymbol{q})$.

Additional Evaluation Metrics

The upper connectivity bound R_{com} in Eq. (8.4) and Eq. (8.9) is set to 9km, which corresponds approximately to radio communication capabilities for ASV and we set the lower collision threshold from Eq. (8.3) to $R_{coll} = 100$m, such that the ASVs would still have some margin in real conditions. Moreover, we also set the edge hysteresis parameter from Eq. (8.9) to $\epsilon = 300$m. We use the Euclidean norm to measure the distances between agents $d(\boldsymbol{q}_i, \boldsymbol{q}_j)$ and the minimum flock center distance to the target $d_{min}(T)$.

Numerical results

The results when the true currents are known (HC-HC) and in the realistic scenario where only forecasts are available (FC-HC) are presented in Table 8.1. Both HJR-Flocking and HJR-Reactive exhibit superior performance in terms of connectivity and collision metrics compared to the baseline HJR. Thus, we conduct statistical testing to compare HJR-Reactive and HJR-Flocking. Regarding disconnection and collision rate, we perform a one-sided two-sample z proportion test for HJR-Flocking against HJR-Reactive.

Let Γ be the rate of collision or disconnection over $\mathbb{M}$ with the null hypothesis being $H_0 : \Gamma_{\text{HJR-Flocking}} = \Gamma_{\text{HJR-Reactive}}$ to reject in favor of the alternative hypothesis $H_A : \Gamma_{\text{HJR-Flocking}} < \Gamma_{\text{HJR-Reactive}}$. HJR-Flocking is statistically significantly better than HJR-Reactive at avoiding disconnections in both (HC-HC) and (FC-HC) scenarios, with p-values of $p = 6.3e^{-69}$ and $p = 1.7e^{-114}$, respectively. However, it is not significantly better than HJR-Reactive

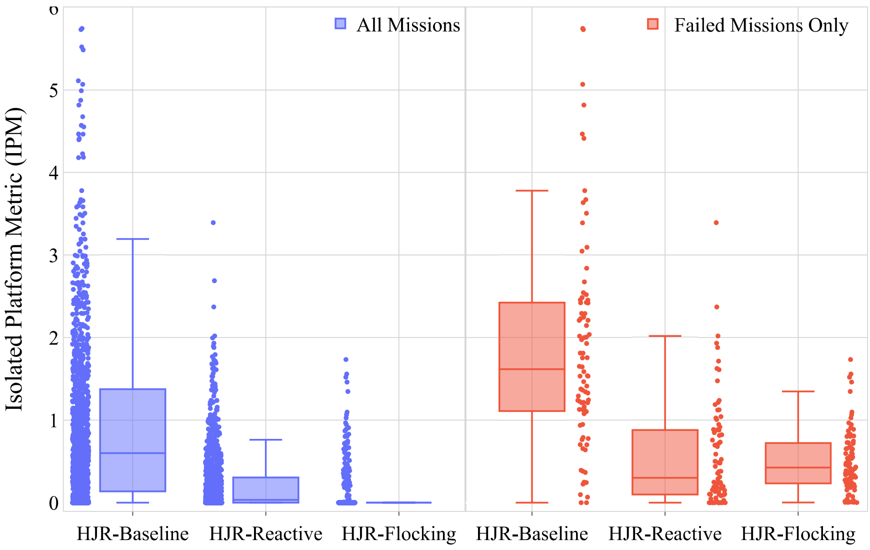


Figure 8.4: IPM. Left evaluated on the set of all missions $\mathbb{M}$, right on the missions where HJR-Flocking failed only. Due to its low IPM, HJR-Flocking typically has both a low disconnection time and a low number of disconnected agents.

at avoiding collisions. To compare the means over $|\mathbb{M}|$ of $\mu(\text{IPM})$ and $\mu(\lambda_2^{min})$ for connectivity and $d_{min}(\mathcal{T})$ for the performance trade-off, we performe a Welch's t-test due to the unequal variances of HJR-Reactive and HJR-Flocking. HJR-Flocking leads to statistically significantly better results for network connectivity with $p < 1e^{-30}$ for $\mu(\text{IPM})$ and $\mu(\lambda_2^{min})$ for both (HC-HC) and (FC-HC) scenarios. Due to its higher value of $\mu(\lambda_2^{min})$, HJR-Flocking is more robust against disconnections; see Fig. 8.5 and should be the preferred control choice for communication maintenance.

Moreover, we plot the IPM for the three controllers in Fig. 8.4 for two cases, (1) the IPM evaluated on the full set of missions $|\mathbb{M}|$ (2) on a subset of missions where flocking failed to maintain connectivity. Among the three controllers evaluated, HJR-Flocking has the lowest IPM. Taking into account the missions where HJR-Flocking failed to maintain connectivity, it still achieves a shorter disconnection time or fewer disconnected agents than the HJR-Baseline, but it is not as distinguishable from the HJR-Reactive controller. Interestingly, HJR-Reactive yields a statistically significantly better outcome for the objective trade-off $\mu(d_{min}(\mathcal{T}))$ with p-values $p < 1e^{-40}$ in both (HC-HC) and (FC-HC). Finally, Fig. 8.6 illustrates a navigation mission, comparing a naive multi-agent approach (HJR-Baseline) to our safe interaction controller, HJR-Flocking.

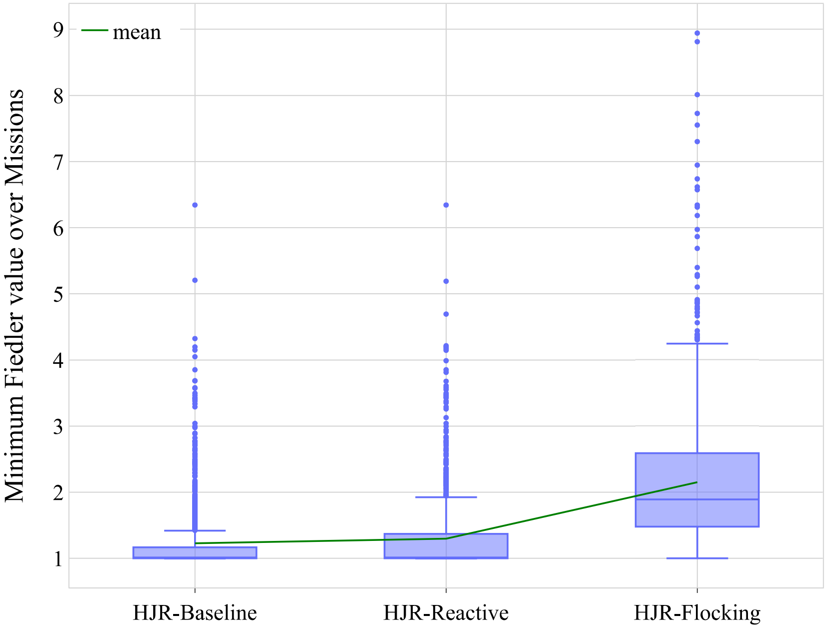


Figure 8.5: The minimum Fiedler value λ_2^{min} can be used as a graph connectivity measure. HJR-Flocking has the highest minimum Fiedler value, which ensures better robustness against connectivity failures.

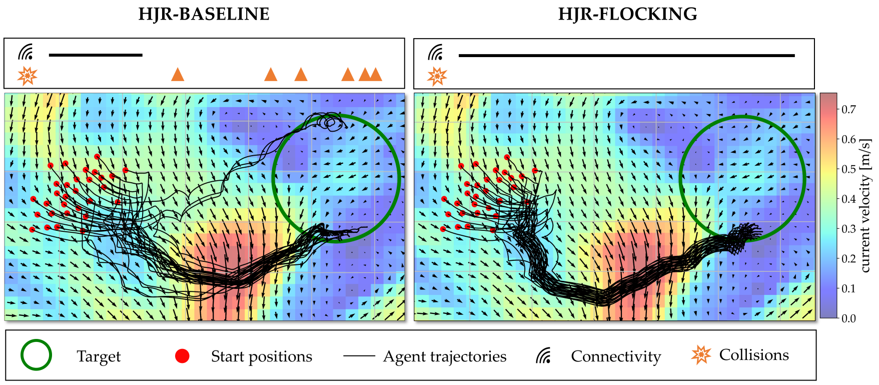


Figure 8.6: Comparison of the HJR-Baseline with the low interference safe interaction controller HJR-Flocking. HJR-Flocking (right) guarantees communication through the full length of the mission, avoids collisions and ensures that all agent reach the target.

	Coll.	Disconn.	$\mu(\text{IPM}) \downarrow$	$\mu(\lambda_2^{min}) \uparrow$	$\mu\left(d_{min}\left(\mathcal{T}\right)\right) \downarrow$
π_{perf} plans on true flows					
HJR-Baseline	68.5%	50.1%	0.37	0.39	**0** km
HJR-Reactive	**0**%	44.8%	0.19	0.42	0.14 km*
HJR-Flocking	0.7%	**9.9**%*	**0.05***	**1.15***	5.90 km
π_{perf} plans on forecast					
HJR-Baseline	39.1 %	70.5%	0.92	0.23	**10.55** km
HJR-Reactive	**0**%	58%	0.23	0.30	10.84 km*
HJR-Flocking	0.7%	**9.9**%*	**0.043***	**1.15***	13.96 km

Table 8.1: We compare the performance of multiple controllers in two forecast settings. The * marks a statistically significant better performance of either HJR-Reactive or HJR-Flocking for the connectivity metrics $\mu(\text{IPM})$, $\mu(\lambda_2^{min})$ and the performance metric $d_{min}\left(\mathcal{T}\right)$. Regarding the collision rate (Coll.) and disconnection rate (Disconn.), * denotes a statistically significant conclusion that HJR-Flocking leads to a lower collision or disconnection rate respectively.

Discussion

It is clear that HJR-Flocking outperforms HJR-Reactive and the HJR-Baseline in terms of connectivity metrics. Interestingly, HJR-Flocking leads to a slightly higher collision rate in Table 8.1 than HJR-Reactive. We believe that it is mainly due to two reasons: (1) In HJR-Reactive the expected risk of collisions is inherently lower as each agent can achieve connectivity with a maximum amount of two other agents while HJR-Flocking achieves a similar structure to a lattice configuration [100] (2) In our example, all agents navigate to the same target, which also increases the risk of collisions, as it is a common implicit regularizer. We expect improvement in terms of collision rate for application to autonomous ASVs, where each agent maximizes an objective along its trajectory [60]. The discrepancy between the performance trade-off with each agent target reaching objective $d_{min}\left(\mathcal{T}\right)$ in Table 8.1 is less noticeable in the (FC-HC) setting, since the HJR performance is also degraded due to stochastic error when planning on forecasts [141].

8.6 Conclusion

In this chapter, we proposed a H-MAS approach to maintain network connectivity in complex dynamical flows while satisfying the single agent objectives when feasible. Our method blends a network safety controller for collisions and connectivity maintenance with a performance control policy, which allows us to decompose a complex multi-agent problem effectively. Our Low Interference Safe Interaction Controller prioritizes a safe control input from a flocking-inspired potential function in critical scenarios. We showed that connectivity can be maintained and collision avoided in underactuated agents, as long as the flow dynamics divergence between neighboring agents can be compensated for. Our empirical results in realistic ocean dynamics showed that our method efficiently maintains connectivity and avoids collisions in most scenarios, while reasonably trading off with each agent's performance objective. Future work includes leveraging the agent's dynamics with forecast flows to predict future disconnections or collisions using predictive methods [18]. We anticipate that these methods will perform well on the true flow scenario (HC-HC) but may exhibit a performance drop when stochastic error is present as in the (FC-HC) scenario. It will be interesting to evaluate whether the additional computational cost of predictive methods pays off.

Part V

Conclusion and Future Directions

Chapter 9

Conclusion

The primary objective of this dissertation was to explore the potential of operating underactuated robotic systems by ***going with the flow***: letting the system drift in natural flows in the oceans and skies and strategically using a small engine to change flows when beneficial. Compared to using powerful engines for operation, this paradigm requires 2-3 orders of magnitude less power and energy and hence enables applications that were prohibitively expensive before, e.g., floating seaweed farms for Carbon Dioxide Removal (CDR), floating solar farms, long duration and active monitoring of the oceans and atmosphere. Throughout this thesis we have developed techniques that tackle the four key challenges of this paradigm: (a) the severe underactuation of the system with respect to the surrounding flows, (b) only coarse, deterministic forecasts are available, (c) the large forecast error that is often larger than the propulsion of the system, and (d) the limited forecast horizon of 5-10 days when the control objective can be over months.

In part II, we focused on establishing optimal control strategies when the flows are known. We developed these strategies based on dynamic programming principles and introduced two formulations. First, Multi-Time Reachability, which is particularly suited for analysis and navigation of systems in time-varying dynamics. Second, we introduce how we can maximize seaweed growth without state augmentation by formulating the objective as a running cost in the 2D spatial state.

In part III, we addressed the challenge of operating with short-term deterministic forecasts to achieve long-term objectives. We saw that the dynamic programming formulations from Part II can be used as closed-loop control policy that is equivalent to replanning at every step. This frequent replanning makes the methods reliable despite forecast errors and allowed us to navigate with a 82.3% success rate over 5 day missions. Furthermore, we proposed a method to account for long-term objectives beyond the forecast horizon by estimating the expected cost-to-go using historical averages and discounting future costs. We validated these extensions in experiments to maximize seaweed growth over 30 days, where our method achieved 95.77% of the best possible growth.

In part IV, we turned to the question how we can satisfy constraints while operating underactuated systems in strong flows. For that we first evaluated the risk of stranding and then demonstrated that by incorporating land as obstacles in the Multi-Time Reachability formulation, we can reduce the 10-day risk of stranding from 4% to below 1%. Lastly, we investigated operating fleets of underacted systems with the constraint to stay connected in a local mesh network while avoiding collisions and achieving their performance objectives. Our hierarchical control framework can guarantee safety under mild conditions on the flow and reduced collisions to below 1% and disconnections to below 10% for 30 agents operating in realistic flows in the Gulf of Mexico.

In conclusion, this dissertation represents a significant step towards enabling underactuated robotic systems to operate in the oceans and skies with ultra-low power by *going with the flow*. Our research has shown that by taking advantage of natural flows rather than disregarding them or treating them as disturbances, it is feasible to achieve efficient and reliable operation while drastically reducing energy consumption. We are convinced that this will enable a host of new applications that would otherwise be prohibitively expensive to operate with powerful engines.

Chapter 10

Ongoing and Future Directions

Several ongoing and future directions are promising to further improve the performance of underactuated systems that operate by *going with the flow*.

Real World Validation Tests A crucial next step will be testing these methods in real applications. For that, we are currently conducting ocean trials with floating seaweed farms of the company Phykos [40]. These trials will further validate the realism of the simulation studies and potentially surface issues that were not considered yet. Figure 10.1 shows a picture of a recent trial at Ocean Beach, San Francisco.

Figure 10.1: Launching an autonomous seaweed vessel using the methods developed in this thesis at Ocean Beach for real world validation tests.

Learning About the Currents Online The methods in this thesis were developed using only short-term deterministic flow forecasts and using frequent replanning to mitigate the forecast error. Another possible approach to handle flow uncertainty is to utilize the information available by operating in the flows. When drifting with the flow, it is possible to infer the true flows at the system location. Using the history of these flow measurements, it is possible to use techniques from data assimilation to improve the forecast and thereby improve operating performance. There are two different approaches for this depending on the forecasts available. If only deterministic forecasts are available, data assimilation methods are a form of interpolation. Together with the Master Student Killian Kempf, we found that using a combination of a 3D Gaussian Process and a Unet Neural Network it is possible to reduce the forecast error around the platform by up to 35% RMSE and achieve an R2 value of 0.413 [58]. For details on this method, we refer to the thesis [58]. If a distribution of forecasts is available, for example in the form of probabilistic ensembles from classical numerical prediction systems or neural surrogate models [104], methods can be developed to compute a posterior distribution that assimilates the measurements while still providing a measure of uncertainty. These improved forecasts, be they in the form of a deterministic forecast or a probabilistic ensemble, can then be used to regularly replan with. An open research question is how to best plan with probabilistic ensembles.

Deep Reinforcement Learning The promise of Deep Reinforcement Learning in this application is that it can learn to hedge against flow uncertainty from experience without needing an explicit distribution of flows. Deep RL can build on the techniques we introduced in this dissertation. For example in the Master Thesis of Jerome Jeannin [53], we used the value function of Multi-time Reachability (Chapter 3) and the improved forecasts [58] as inputs for an RL algorithm. The value function naturally lends itself as a dense reward signal as progress on it signals progress towards the target. We found a small but significant improvement compared to directly using the value function as closed-loop policy (Chapter 5). Another recent result showed that in simple, analytical current flows using a transformer architecture the RL policy network is able to directly use the measurements to hedge against uncertainty [24]. Further exploration of this approach of combining learning with classical techniques seems promising for improving performance.

Partially Observable Markov Decision Process The most principled way for sequential decision making under uncertainty with incoming measurements is formulating the problem as Partially Observable Markov Decision Process (POMDP). This formulation naturally blends information gathering with reward-maximizing actions. Assuming a probabilistic ensemble for the flow distribution and an observation model, one can formulate operating in flows as POMDP thereby anticipating future measurements compared to control approach based on the separation principle where data assimilation and replanning are treated independent.

Bibliography

[1] Ahmed Aboudonia et al. "Reconfigurable Plug-and-play Distributed Model Predictive Control for Reference Tracking". In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. 2022, pp. 1130–1135. DOI: 10.1109/CDC51059.2022.9993380.

[2] Aaron D. Ames et al. "Control Barrier Functions: Theory and Applications". In: *2019 18th European Control Conference (ECC)*. 2019 18th European Control Conference (ECC). June 2019, pp. 3420–3431. DOI: 10.23919/ECC.2019.8796030.

[3] Anil Aswani et al. "Provably safe and robust learning-based model predictive control". In: *Automatica* 49.5 (2013), pp. 1216–1226.

[4] Melissa August. *History of deadly japanese balloon bomb in World War II*. May 2023. URL: https://time.com/6276685/japanese-balloon-bomb-history-world-war-ii/.

[5] Somil Bansal et al. "Hamilton-jacobi reachability: A brief overview and recent advances". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 2242–2253.

[6] Alexandre M Bayen et al. "Aircraft autolander safety analysis through optimal control-based reach set computation". In: *Journal of Guidance, Control, and Dynamics* 30.1 (2007), pp. 68–77.

[7] Lukas Beckenbach and Stefan Streif. "Approximate infinite-horizon predictive control". In: *IEEE International Conference on Decision and Control*. IEEE. 2022, pp. 3711–3717.

[8] Marc G Bellemare et al. "Autonomous navigation of stratospheric balloons using reinforcement learning". In: *Nature* 588.7836 (2020), pp. 77–82.

[9] Alberto Bemporad and Manfred Morari. "Robust model predictive control: A survey". In: *Robustness in identification and control*. Springer London, 1999, pp. 207–226. ISBN: 978-1-84628-538-7.

[10] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*. Vol. 1. Athena scientific, 2012.

[11] M.S. Bhabra et al. "Optimal Harvesting with Autonomous Tow Vessels for Offshore Macroalgae Farming". In: *Global Oceans 2020: Singapore – U.S. Gulf Coast* (2020), pp. 1–10.

[12] Manmeet S. Bhabra et al. "Optimal Harvesting with Autonomous Tow Vessels for Offshore Macroalgae Farming". In: *OCEANS 2020 IEEE/MTS*. IEEE. Oct. 2020, pp. 1–10. DOI: 10.1109/IEEECONF38699.2020.9389474.

[13] Olivier Bokanowski, Nicolas Forcadel, and Hasnaa Zidani. "Reachability and Minimal Times for State Constrained Nonlinear Problems without Any Controllability Assumption". In: *SIAM J. on Control and Optim.* 48.7 (2010), pp. 4292–4316. DOI: 10.1137/090762075.

[14] Olivier Bokanowski and Hasnaa Zidani. "Minimal Time Problems With Moving Targets and Obstacles". In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, pp. 2589–2593. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20110828-6-IT-1002.02261.

[15] Even Borhaug, Alexey Pavlov, and Kristin Y Pettersen. "Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents". In: *2008 47th IEEE conference on decision and control*. IEEE. 2008, pp. 4984–4991.

[16] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.2.5. 2018.

[17] Michele Buzzicotti et al. "Optimal control of point-to-point navigation in turbulent time dependent flows using Reinforcement Learning". In: *International Conference of the Italian Association for Artificial Intelligence*. Springer. 2020, pp. 223–234.

[18] Andrea Carron et al. "Multi-agent Distributed Model Predictive Control with Connectivity Constraint". In: *arXiv preprint arXiv:2303.06957* (2023).

[19] Anjan Chakrabarty and Jack Langelaan. "UAV flight path planning in time varying complex wind-fields". In: *2013 American control conference*. IEEE. 2013, pp. 2568–2574.

[20] Eric P Chassignet et al. "US GODAE: global ocean prediction with the HYbrid Coordinate Ocean Model (HYCOM)". In: *Oceanography* 22.2 (2009), pp. 64–75.

[21] Fei Chen and Wei Ren. "On the Control of Multi-Agent Systems: A Survey". In: *Foundations and Trends® in Systems and Control* 6.4 (2019), pp. 339–499. ISSN: 2325-6818, 2325-6826. DOI: 10.1561/2600000019. URL: http://www.nowpublishers.com/article/Details/SYS-019 (visited on 12/07/2022).

[22] Mo Chen and Claire J Tomlin. "Hamilton–Jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1 (2018), pp. 333–358.

[23] Mo Chen et al. "Decomposition of reachable sets and tubes for a class of nonlinear systems". In: *IEEE Transactions on Automatic Control* 63.11 (2018), pp. 3675–3688.

[24] Rohit Chowdhury, Raswanth Murugan, and Deepak Subramani. “Intelligent Onboard Routing in Stochastic Dynamic Environments using Transformers”. In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems.* 2023, pp. 1688–1696.

[25] Rohit Chowdhury and Deepak N Subramani. “Physics-driven machine learning for time-optimal path planning in stochastic dynamic flows”. In: *International Conference on Dynamic Data Driven Application Systems.* Springer. 2020, pp. 293–301.

[26] *COLREGs: Convention on the International Regulations for Preventing Collisions at Sea.* International Maritime Organization (IMO), 1972.

[27] DS Crosby, LC Breaker, and WH Gemmill. “A proposed definition for vector correlation in geophysics: Theory and application”. In: *Journal of Atmospheric and Oceanic Technology* 10.3 (1993), pp. 355–367.

[28] Jonas Dieker. “Generative Models for Simulation of Realistic Ocean Currents”. In: *Master Thesis at the School of Computation, Information and Technology* (2023).

[29] Jerry Ding et al. “Reachability calculations for automated aerial refueling”. In: *47th IEEE International Conference on Decision and Control (CDC).* IEEE. 2008, pp. 3706–3712.

[30] Andreas Doering et al. “Stranding Risk for Underactuated Vessels in Complex Ocean Currents: Analysis and Controllers”. In: *2023 IEEE 62th Annual Conference on Decision and Control (CDC).* IEEE. 2023.

[31] Manan Doshi et al. “Hamilton-Jacobi Multi-Time Reachability”. In: *2022 IEEE 61th Annual Conference on Decision and Control (CDC).* IEEE. 2022, pp. 2443–2450.

[32] Manan M. Doshi, Manmeet S. Bhabra, and Pierre F. J. Lermusiaux. “Energy-Time Optimal Path Planning in Dynamic Flows: Theory and Schemes”. In: *Computer Methods in Applied Mechanics and Engineering* (2022). Sub-judice.

[33] Chun-Xia Dou and Bin Liu. “Multi-Agent Based Hierarchical Hybrid Control for Smart Microgrid”. In: *IEEE Transactions on Smart Grid* 4.2 (June 2013), pp. 771–778. ISSN: 1949-3061. DOI: 10.1109/TSG.2012.2230197.

[34] Kenji Doya. “Reinforcement learning in continuous time and space”. In: *Neural computation* 12.1 (2000), pp. 219–245.

[35] EU Copernicus Marine Service Information. *Global Ocean 1/12 Physics Analysis and Forecast updated Daily Product [Data set].* Mar. 21, 2022. URL: https://doi.org/10.48670/moi-00016.

[36] L.C. Evans. *Partial Differential Equations.* Graduate studies in mathematics. American Mathematical Society, 2010. ISBN: 9780821849743.

[37] Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. “Search on the Replay Buffer: Bridging Planning and Reinforcement Learning”. In: *Advances in Neural Information Processing Systems.* Vol. 32. 2019.

[38] Jaime F Fisac et al. “A general safety framework for learning-based control in uncertain robotic systems”. In: *IEEE Transactions on Automatic Control* 64.7 (2018), pp. 2737–2752.

[39] Jaime F Fisac et al. “Reach-avoid problems with time-varying dynamics, targets and constraints”. In: *Proceedings of the 18th international conference on hybrid systems: computation and control*. HSCC ’15. Seattle, Washington: Association for Computing Machinery, 2015, pp. 11–20.

[40] *Floating Seaweed Farms*. Phykos PBC. URL: https://www.phykos.co.

[41] GEBCO Bathymetric Compilation Group 2022. *The GEBCO_2022 Grid - a continuous terrain model of the global oceans and land*. June 22, 2022. DOI: 10.5285/e0f0bb80-ab44-2739-e053-6c86abc0289c.

[42] Xiongfei Geng et al. “Motion plan of maritime autonomous surface ships by dynamic programming for collision avoidance and speed optimization”. In: *Sensors* 19.2 (2019). ISSN: 14248220.

[43] Peter Gunnarson et al. “Learning efficient navigation in vortical flow fields”. In: *Nature communications* 12.1 (2021), pp. 1–7.

[44] Sylvia L Herbert et al. “FaSTrack: A modular framework for fast and guaranteed safe motion planning”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 1517–1522.

[45] Hans Hersbach et al. “The ERA5 global reanalysis”. In: *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020), pp. 1999–2049.

[46] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[47] Nicolas Hoischen. “Multi-Agent Control for a fleet of floating seaweed farms”. Master’s thesis, ETH Zurich and University of California, Berkeley. 2023.

[48] Nicolas Hoischen, Marius Wiggert, and Claire J Tomlin. “Safe Connectivity Maintenance in Underactuated Multi-Agent Networks for Dynamic Oceanic Environments”. In: *arXiv preprint arXiv:2307.01927* (2023).

[49] Geoffrey A Hollinger et al. “Learning uncertainty in ocean current predictions for safe and reliable navigation of underwater vehicles”. In: *Journal of Field Robotics* 33.1 (2016), pp. 47–66.

[50] Haomiao Huang et al. “A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag”. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 1451–1456.

[51] Van T Huynh, Matthew Dunbabin, and Ryan N Smith. “Predictive motion planning for AUVs subject to strong time-varying currents and forecasting uncertainties”. In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 1144–1151.

[52] X Development Inc. *Project Loon*. 2021. URL: https://x.company/projects/loon/ (visited on 08/20/2021).

[53] Jerome Jeannin. “Reinforcement Learning for Under-Actuated Current-Based Navigation”. In: *Master Thesis at ETH Zurich* (2022).

[54] Meng Ji and Magnus Egerstedt. “Distributed Coordination Control of Multiagent Systems While Preserving Connectedness”. In: *IEEE Transactions on Robotics* 23.4 (Aug. 2007), pp. 693–703. ISSN: 1941-0468. DOI: 10.1109/TRO.2007.900638.

[55] Tor Arne Johansen, Tristan Perez, and Andrea Cristofaro. “Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.12 (2016), pp. 3407–3422. ISSN: 15249050.

[56] Dylan Jones and Geoffrey A Hollinger. “Planning energy-efficient trajectories in strong disturbances”. In: *IEEE Robotics and Automation Letters* 2.4 (2017), pp. 2080–2087.

[57] Mou Junmin et al. “Mechanism of dynamic automatic collision avoidance and the optimal route in multi-ship encounter situations”. In: *Journal of Marine Science and Technology* 26 (2021), pp. 141–158. ISSN: 09484280.

[58] Killian Kempf. “Improving Local Ocean Currents Forecasts with Data Assimilation and Deep Learning Techniques”. In: *Master Thesis at ETH Zurich* (2022).

[59] Matthias Killer. “Long-term Horizon Planning for Underactuated Autonomous Vessels”. In: *Master Thesis at the School of Computation, Information and Technology* (2023).

[60] Matthias Killer et al. “Maximizing seaweed growth on autonomous farms: A dynamic programming approach for underactuated systems navigating on uncertain ocean currents”. In: *arXiv preprint arXiv:2307.01916* (2023).

[61] Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2004.

[62] Gedaliah Knizhnik et al. “Flow-Based Control of Marine Robots in Gyre-Like Environments”. In: *IEEE International Conference on Robotics and Automation*. 2022, pp. 3047–3053.

[63] Andreas Kräuchi et al. “Controlled weather balloon ascents and descents for atmospheric research and climate monitoring”. In: *Atmospheric measurement techniques* 9.3 (2016), pp. 929–938.

[64] Dhanushka Kularatne, Subhrajit Bhattacharya, and M Ani Hsieh. “Going with the flow: a graph based approach to optimal path planning in general flows”. In: *Autonomous Robots* 42.7 (2018), pp. 1369–1387.

[65] Dhanushka Kularatne, Hadi Hajieghrary, and M Ani Hsieh. “Optimal path planning in time-varying flows with forecasting uncertainties”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 4857–4864.

[66] A. B. Kurzhanski and P. Varaiya. “Ellipsoidal Techniques for Reachability Under State Constraints”. In: *SIAM Journal on Control and Optimization* 45.4 (2006), pp. 1369–1394. DOI: 10.1137/S0363012903437605.

[67] Alexander B Kurzhanski and Pravin Varaiya. “Dynamic optimization for reachability problems”. In: *J. of Optimization Theory and Applications* 108.2 (2001), pp. 227–251.

[68] M. Lazar et al. “On input-to-state stability of min–max nonlinear model predictive control”. In: *Systems & Control Letters* 57.1 (2008), pp. 39–48. ISSN: 0167-6911. DOI: https://doi.org/10.1016/j.sysconle.2007.06.013.

[69] Agnieszka Lazarowska. “A Trajectory Base Method for Ship’s Safe Path Planning”. In: *Procedia Computer Science* 96 (2016), pp. 1022–1031. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2016.08.118.

[70] L. Lebreton et al. “Evidence that the Great Pacific Garbage Patch is rapidly accumulating plastic”. In: *Scientific Reports* 8.1 (Mar. 22, 2018). Number: 1 Publisher: Nature Publishing Group, p. 4666. ISSN: 2045-2322. DOI: 10.1038/s41598-018-22939-w.

[71] P. F. J. Lermusiaux et al. “A Future for Intelligent Autonomous Ocean Observing Systems”. In: *Journal of Marine Research* 75.6 (Nov. 2017). The Sea. Volume 17, The Science of Ocean Prediction, Part 2., pp. 765–813. DOI: 10.1357/002224017823524035.

[72] P. F. J. Lermusiaux et al. “Optimal Planning and Sampling Predictions for Autonomous and Lagrangian Platforms and Sensors in the Northern Arabian Sea”. In: *Oceanography* 30.2 (June 2017). Special issue on Autonomous and Lagrangian Platforms and Sensors (ALPS), pp. 172–185. DOI: 10.5670/oceanog.2017.242.

[73] P. F. J. Lermusiaux et al. “Quantifying Uncertainties in Ocean Predictions”. In: *Oceanography* 19.1 (2006), pp. 92–105. DOI: 10.5670/oceanog.2006.93.

[74] Pierre Lermusiaux et al. “Real-time Probabilistic Coupled Ocean Physics-Acoustics Forecasting and Data Assimilation for Underwater GPS”. In: *OCEANS 2020 IEEE*. IEEE. Oct. 2020, pp. 1–9. DOI: 10.1109/IEEECONF38699.2020.9389003.

[75] Pierre F. J. Lermusiaux et al. “Progress and prospects of U.S. data assimilation in ocean research”. In: *Oceanography* 19.1 (2006), pp. 172–183. DOI: 10.5670/oceanog.2006.102.

[76] K. Leung et al. “On Infusing Reachability-Based Safety Assurance within Probabilistic Planning Frameworks for Human-Robot Vehicle Interactions”. In: *Int. Symp. on Experimental Robotics*. 2018.

[77] Zonggang Li, Chen Na, and Guangming Xie. “Flocking of Multi-Agent Systems with Multiple Virtual Leaders Based on Connectivity Preservation Approach”. In: *Proceedings of the 2015 Chinese Intelligent Systems Conference*. Ed. by Yingmin Jia et al. Lecture Notes in Electrical Engineering. Berlin, Heidelberg: Springer, 2016, pp. 465–473. ISBN: 978-3-662-48386-2. DOI: 10.1007/978-3-662-48386-2_48.

[78] Weiwei Liao and Tao Liang. *Computation of Reachable Sets Based on Hamilton-Jacobi-Bellman Equation with Running Cost Function*. 2021. DOI: 10.48550/ARXIV.2107.11941.

[79] Jing Lin and P. F. J. Lermusiaux. "Bayesian Learning of Turbulent Bottom Gravity Currents Models". In: (2022). In preparation.

[80] Junpo Liu, Baigang Zhao, and Lianbo Li. "Collision Avoidance for Underactuated Ocean-Going Vessels Considering COLREGs Constraints". In: *IEEE Access* 9 (2021). Conference Name: IEEE Access, pp. 145943–145954. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3123449.

[81] T. Lolla, P. J. Haley Jr., and P. F. J. Lermusiaux. "Path planning in multiscale ocean flows: Coordination and dynamic obstacles". In: *Ocean Modelling* 94 (2015), pp. 46–66. DOI: 10.1016/j.ocemod.2015.07.013.

[82] T. Lolla, P. J. Haley Jr., and P. F. J. Lermusiaux. "Time-Optimal Path Planning in Dynamic Flows using Level Set Equations: Realistic Applications". In: *Ocean Dynamics* 64.10 (2014), pp. 1399–1417. DOI: 10.1007/s10236-014-0760-3.

[83] T. Lolla, P.J. Haley Jr., and P.F.J. Lermusiaux. "Path Planning in Multi-Scale Ocean Flows: Coordination and Dynamic Obstacles". In: *Ocean Modelling* 94 (Oct. 2015), pp. 46–66. ISSN: 14635003. DOI: 10.1016/j.ocemod.2015.07.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S1463500315001225 (visited on 12/07/2022).

[84] T. Lolla et al. "Time-Optimal Path Planning in Dynamic Flows using Level Set Equations: Theory and Schemes". In: *Ocean Dynamics* 64.10 (2014), pp. 1373–1397. DOI: 10.1007/s10236-014-0757-y.

[85] Tapovan Lolla et al. "Path planning in time dependent flow fields using level set methods". In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*. 2012, pp. 166–173. DOI: 10.1109/ICRA.2012.6225364.

[86] Harvard Lomax, Thomas H Pulliam, and David W Zingg. *Fundamentals of computational fluid dynamics*. Springer Science & Business Media, 2013.

[87] Kendall Lowrey et al. "Plan online, learn offline: Efficient learning and exploration via model-based control". In: *arXiv preprint arXiv:1811.01848* (2018).

[88] R Lumpkin and L Centurioni. *NOAA global drifter program quality-controlled 6-hour interpolated data from ocean surface drifting buoys. NOAA National Centers for Environmental Information*. 2019.

[89] Rick Lumpkin, Mayra Pazos, et al. "Measuring surface currents with Surface Velocity Program drifters: the instrument, its data, and some recent results". In: *Lagrangian analysis and prediction of coastal and ocean dynamics* 39 (2007), p. 67.

[90] Gianandrea Mannarini and Lorenzo Carelli. "VISIR-1. b: Ocean surface gravity waves and currents for energy-efficient navigation". In: *Geoscientific Model Development* 12.8 (2019), pp. 3449–3480.

[91] Gianandrea Mannarini et al. "Graph-Search and Differential Equations for Time-Optimal Vessel Route Planning in Dynamic Ocean Waves". In: *IEEE Transactions on Intelligent Transportation Systems* 21.6 (June 2020), pp. 1–13. DOI: 10.1109/TITS.2019.2935614.

[92] I. Martins and J.C. Marques. "A Model for the Growth of Opportunistic Macroalgae (Enteromorpha sp.) in Tidal Estuaries". In: *Estuarine, Coastal and Shelf Science* 55.2 (2002), pp. 247–257. ISSN: 0272-7714. DOI: https://doi.org/10.1006/ecss.2001.0900.

[93] Mehran Mesbahi and Magnus Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton Series in Applied Mathematics. Princeton: Princeton University Press, 2010. 403 pp. ISBN: 978-0-691-14061-2.

[94] EJ Metzger et al. *Validation Test Report for the Global Ocean Forecast System 3.5-1/25 degree HYCOM/CICE with Tides*. Tech. rep. Vaval Research Lab Washington DC, US, 2020.

[95] Eivind Meyer et al. "COLREG-Compliant Collision Avoidance for Unmanned Surface Vehicle Using Deep Reinforcement Learning". In: *IEEE Access* 8 (2020), pp. 165344–165364. DOI: 10.1109/ACCESS.2020.3022600.

[96] Ian M Mitchell et al. "A toolbox of level set methods". In: *UBC Department of Computer Science Technical Report TR-2007-11* (2007), p. 31.

[97] Ian M Mitchell and Jeremy A Templeton. "A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems". In: *Hybrid Systems: Computation and Control: 8th International Workshop, HSCC*. 2005, pp. 480–494.

[98] Signe Moe and Kristin Y. Pettersen. "Set-Based line-of-sight (LOS) path following with collision avoidance for underactuated unmanned surface vessels under the influence of ocean currents". In: *2017 IEEE Conf. on Control Tech. and Applications (CCTA)*. 2017 IEEE Conference on Control Technology and Applications (CCTA). Aug. 2017, pp. 241–248. DOI: 10.1109/CCTA.2017.8062470.

[99] Artem Molchanov, Andreas Breitenmoser, and Gaurav S Sukhatme. "Active drifters: Towards a practical multi-robot system for ocean monitoring". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 545–552.

[100] R. Olfati-Saber. "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory". In: *IEEE Transactions on Automatic Control* 51.3 (Mar. 2006), pp. 401–420. ISSN: 0018-9286. DOI: 10.1109/TAC.2005.864190. URL: http://ieeexplore.ieee.org/document/1605401/ (visited on 12/07/2022).

[101] Stanley Osher and Ronald P Fedkiw. *Level set methods and dynamic implicit surfaces*. Vol. 1. Springer New York, 2005.

[102] Michael Otte, William Silva, and Eric Frew. "Any-time path-planning: Time-varying wind field+ moving obstacles". In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 2575–2582.

[103] Shubham Pateria et al. "Hierarchical Reinforcement Learning: A Comprehensive Survey". In: *ACM Computing Surveys* 54.5 (June 5, 2021), 109:1–109:35. ISSN: 0360-0300. DOI: 10.1145/3453160. URL: https://dl.acm.org/doi/10.1145/3453160 (visited on 03/18/2023).

[104] Jaideep Pathak et al. "Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators". In: *arXiv preprint arXiv:2202.11214* (2022).

[105] Bruce D Patterson et al. "Renewable CO2 recycling and synthetic fuel production in a marine environment". In: *Proceedings of the National Academy of Sciences* 116.25 (2019), pp. 12212–12219.

[106] Zhouhua Peng et al. "Neural Adaptive Flocking Control of Networked Underactuated Autonomous Surface Vehicles in the Presence of Uncertain Dynamics". In: *Proceedings of the 31st Chinese Control Conference*. Proceedings of the 31st Chinese Control Conference. July 2012, pp. 2865–2870.

[107] Arvind A Pereira et al. "Risk-aware path planning for autonomous underwater vehicles using predictive ocean models". In: *Journal of Field Robotics* 30.5 (2013), pp. 741–762.

[108] Guilherme AS Pereira et al. "Decentralized motion planning for multiple robots subject to sensing and communication constraints". In: *Departmental Papers (MEAM)* (2003), p. 45.

[109] Nadia Pinardi et al. "From weather to ocean predictions: an historical viewpoint". In: *Journal of Marine Research* 75.3 (May 2017). Special issue: The Science of Ocean Prediction, vol. 17 of The Sea, pp. 103–159. DOI: 10.1357/002224017821836789.

[110] Nadia Pinardi et al. "The Sea: The Science of Ocean Prediction". In: *Journal of Marine Research* 75.3 (May 2017). Special issue: The Science of Ocean Prediction, vol. 17 of The Sea, pp. 101–102.

[111] Saša V Raković and William S Levine. *Handbook of model predictive control*. Springer, 2018.

[112] Dushyant Rao and Stefan B Williams. "Large-scale path planning for underwater gliders in ocean currents". In: *Australasian conference on robotics and automation (ACRA)*. Citeseer. 2009, pp. 2–4.

[113] P. Rivera-Ortiz and Y. Diaz-Mercado. "On Guaranteed Capture in Multi-Player Reach-Avoid Games via Coverage Control". In: *IEEE Control Systems Letters* (2018).

[114] Hugh Roarty et al. "The global high frequency radar network". In: *Frontiers in marine science* 6 (2019), p. 164.

[115] Breanna Michell Roque et al. "Effect of the macroalgae Asparagopsis taxiformis on methane production and rumen microbiome assemblage". In: *Animal Microbiome* 1 (2019), pp. 1–14.

[116] Finnley Ross, Patrick Tarbuck, and Peter I Macreadie. "Seaweed afforestation at large-scales exclusively for carbon sequestration: Critical assessment of risks, viability and the state of knowledge". In: *Frontiers in Marine Science* 9 (2022), p. 2269.

[117] Raja Rout, Rongxin Cui, and Zhengqing Han. "Modified line-of-sight guidance law with adaptive neural network control of underactuated marine vehicles with state and input constraints". In: *IEEE transactions on control systems technology* 28.5 (2020), pp. 1902–1914.

[118] Wesam H Al-Sabban, Luis F Gonzalez, and Ryan N Smith. "Wind-energy based path planning for unmanned aerial vehicles using markov decision processes". In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 784–789.

[119] Ed Schmerling. *hj_reachability - Online*. 2022. (Visited on 03/28/2022).

[120] *Seaweed Farming as Climate Solution*. Project Drawdown. URL: https://drawdown.org/solutions/seaweed-farming.

[121] Y. S. Shao et al. "Reachability-based trajectory safeguard (rts): A safe and fast reinforcement learning safety layer for continuous control". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3663–3670.

[122] Martin T Sherman et al. "SeaweedPaddock: Initial Modeling and Design for a Sargassum Ranch". In: *OCEANS 2018 MTS/IEEE*. IEEE. 2018, pp. 1–6.

[123] Chi-Wang Shu. "High order numerical methods for time dependent Hamilton-Jacobi equations". In: *Mathematics and computation in imaging science and information processing*. World Sc., 2007, pp. 47–91.

[124] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529 (2016), pp. 484–503.

[125] Hou-Sheng Su. "Flocking in Multi-Agent Systems with Multiple Virtual Leaders Based Only on Position Measurements". In: *Communications in Theoretical Physics* 57.5 (May 2012), pp. 801–807. ISSN: 0253-6102. DOI: 10.1088/0253-6102/57/5/10. URL: https://iopscience.iop.org/article/10.1088/0253-6102/57/5/10 (visited on 01/08/2023).

[126] Housheng Su, Xiaofan Wang, and Wen Yang. "Flocking in Multi-Agent Systems with Multiple Virtual Leaders". In: *Asian Journal of Control* 10.2 (2008), pp. 238–245. ISSN: 1934-6093. DOI: 10.1002/asjc.22. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/asjc.22 (visited on 01/08/2023).

[127] D. N. Subramani, Q. J. Wei, and P. F. J. Lermusiaux. "Stochastic Time-Optimal Path-Planning in Uncertain, Strong, and Dynamic Flows". In: *Computer Methods in Applied Mechanics and Engineering* 333 (2018), pp. 218–237. DOI: 10.1016/j.cma.2018.01.004.

[128] D. N. Subramani et al. "Time-Optimal Path Planning: Real-Time Sea Exercises". In: *Oceans '17 MTS/IEEE Conference*. Aberdeen, June 2017. DOI: 10.1109/OCEANSE.2017.8084776.

[129] Deepak N Subramani, Quantum J Wei, and Pierre FJ Lermusiaux. "Stochastic time-optimal path-planning in uncertain, strong, and dynamic flows". In: *Computer Methods in Applied Mechanics and Engineering* 333 (2018), pp. 218–237.

[130] Deepak N. Subramani and Pierre F. J. Lermusiaux. "Risk-Optimal Path Planning in Stochastic Dynamic Environments". In: *Computer Methods in Applied Mechanics and Engineering* 353 (Aug. 2019), pp. 391–415. DOI: 10.1016/j.cma.2019.04.033.

[131] W. Sun et al. "Pursuit-Evasion Games in Dynamic Flow Fields via Reachability Set Analysis". In: *2017 American Control Conference (ACC)*. Seattle, 2017, pp. 4595–4600. DOI: 10.23919/ACC.2017.7963664.

[132] Wei Sun et al. "Multiple-Pursuer-One-Evader Pursuit Evasion Game in Dynamic Flow Fields". In: *Journal of Guidance, Control and Dynamics* 40.7 (Apr. 2017). DOI: 10.2514/1.G002125.

[133] Wei Sun et al. "Pursuit-evasion games in dynamic flow fields via reachability set analysis". In: *American Control Conference*. 2017, pp. 4595–4600.

[134] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018. ISBN: 0262039249.

[135] Brijen Thananjeyan et al. "Safety augmented value estimation from demonstrations (SAVED): Safe deep model-based RL for sparse cost robotic tasks". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3612–3619.

[136] M-C Tsou and H-C Cheng. "An Ant Colony Algorithm for efficient ship routing". In: *Polish Maritime Research* (2013).

[137] Robert Maxwell Tullberg, Huu Phu Nguyen, and Chien Ming Wang. "Review of the Status and Developments in Seaweed Farming Infrastructure". In: *Journal of Marine Science and Engineering* 10.10 (2022).

[138] M. P. Ueckermann and P. F. J. Lermusiaux. *2.29 Finite Volume MATLAB Framework Documentation*. MSEAS Report 14. Cambridge, MA: Department of Mechanical Engineering, Massachusetts Institute of Technology, 2012. URL: http://mseas.mit.edu/?p=2567.

[139] Kim Peter Wabersich and Melanie N. Zeilinger. "A Predictive Safety Filter for Learning-Based Control of Constrained Nonlinear Dynamical Systems". In: *Automatica* 129 (July 1, 2021), p. 109597. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2021.109597.

[140] Miaomiao Wang et al. "Flocking of Multiple Autonomous Agents with Preserved Network Connectivity and Heterogeneous Nonlinear Dynamics". In: *Neurocomputing* 115 (Sept. 2013), pp. 169–177. ISSN: 09252312. DOI: 10.1016/j.neucom.2012.12.033. URL: https://linkinghub.elsevier.com/retrieve/pii/S092523121300101X (visited on 12/07/2022).

[141] Marius Wiggert et al. "Navigating Underactuated Agents by Hitchhiking Forecast Flows". In: *arXiv preprint, in review for CDC 2022* (2022), pp. 2417–2424. DOI: 10.1109/CDC51059.2022.9992375.

[142] Annie PS Wong et al. "Argo data 1999–2019: Two million temperature-salinity profiles and subsurface velocity observations from a global array of profiling floats". In: *Frontiers in Marine Science* 7 (2020), p. 700.

[143] Jiajun Wu, David P Keller, and Andreas Oschlies. "Carbon dioxide removal via macroalgae open-ocean mariculture and sinking: an Earth system modeling study". In: *Earth System Dynamics* 14.1 (2023), pp. 185–221.

[144] Jiajun Wu, David P. Keller, and Andreas Oschlies. "Carbon Dioxide Removal via Macroalgae Open-ocean Mariculture and Sinking: An Earth System Modeling Study". In: *Earth System Dynamics* (2022).

[145] Minling Wu, Lijun Liu, and Zhen Yu. "Augmented safety guarantee-based area keeping control for an underactuated USV with environmental disturbances". In: *ISA Transactions* 127 (Aug. 1, 2022), pp. 415–422. ISSN: 0019-0578. DOI: 10.1016/j.isatra.2021.08.024. (Visited on 03/26/2023).

[146] Tingruo Yan et al. "Flocking of Multi-agent Systems with Unknown Nonlinear Dynamics and Heterogeneous Virtual Leader". In: *International Journal of Control, Automation and Systems* 19.9 (Sept. 2021), pp. 2931–2939. ISSN: 1598-6446, 2005-4092. DOI: 10.1007/s12555-020-0578-3. URL: https://link.springer.com/10.1007/s12555-020-0578-3 (visited on 12/07/2022).

[147] Wilson Thau Lym Yong et al. "Seaweed: a potential climate change solution". In: *Renewable and Sustainable Energy Reviews* 159.112222 (2022).

[148] Wenwu Yu and Guanrong Chen. "Robust Adaptive Flocking Control of Nonlinear Multi-Agent Systems". In: *2010 IEEE International Symposium on Computer-Aided Control System Design*. 2010 IEEE International Symposium on Computer-Aided Control System Design. Sept. 2010, pp. 363–367. DOI: 10.1109/CACSD.2010.5612787.

[149] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas. “Graph-Theoretic Connectivity Control of Mobile Robot Networks”. In: *Proceedings of the IEEE* 99.9 (Sept. 2011), pp. 1525–1540. ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2011.2157884. URL: http://ieeexplore.ieee.org/document/5948318/ (visited on 03/04/2023).

[150] Jihong Zhang et al. “A model for the growth of mariculture kelp Saccharina japonica in Sanggou Bay, China”. In: *Aquaculture Environment Interactions* 8 (Apr. 2016). DOI: 10.3354/aei00171.

[151] Qing Zhang et al. “Adaptive Flocking of Non-Linear Multi-Agents Systems with Uncertain Parameters”. In: *IET Control Theory & Applications* 9.3 (2015), pp. 351–357. ISSN: 1751-8652. DOI: 10.1049/iet-cta.2014.0471. URL: https://onlinelibrary.wiley.com/doi/abs/10.1049/iet-cta.2014.0471 (visited on 03/04/2023).

[152] Weizhong Zhang et al. “Optimal trajectory generation for a glider in time-varying 2D ocean flows B-spline model”. In: *IEEE International Conference on Robotics and Automation*. IEEE. 2008, pp. 1083–1088.

[153] Luman Zhao and Myung Il Roh. “COLREGs-compliant multiship collision avoidance based on deep reinforcement learning”. In: *Ocean Engineering* 191 (2019), pp. 106436–106450. ISSN: 00298018.

[154] Luman Zhao, Myung-Il Roh, and Sung-Jun Lee. “Control Method for Path Following and Collision Avoidance of Autonomous Ship Based on Deep Reinforcement Learning”. In: *Journal of Marine Science and Technology* 27.4 (2019), pp. 293–310.

[155] Yong Zou et al. “Flocking of Uncertain Nonlinear Multi-Agent Systems via Distributed Adaptive Event-Triggered Control”. In: *Neurocomputing* 465 (Nov. 20, 2021), pp. 503–513. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2021.09.005. URL: https://www.sciencedirect.com/science/article/pii/S092523122101359X (visited on 03/04/2023).

Appendix A

Proof of Stability for Multi-Robot Fleets

Here we give the full proof of stability for our multi-agent controller developed in Chapter 8 under mild conditions on the flows. By stability in this setting we means collision avoidance and connectivity maintenance. This proof is taken from the Master Thesis of Nicolas Hoischen [48].

Potential Function Gradients

As preperation for our proof we first derive the gradients of the potential function Eq. (8.8). Given $\left\|\boldsymbol{q}_{ij}\right\|_2$ the Euclidean norm where $\boldsymbol{q}_{ij} = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top$, the gradient w.r.t to $\boldsymbol{q}_i$ can be computed as $\nabla_{\boldsymbol{q}_i} \left\|\boldsymbol{q}_{ij}\right\|_2 = \frac{\boldsymbol{q}_{ij}}{\left\|\boldsymbol{q}_{ij}\right\|_2}$

Let us derive the gradient of the potential function $\nabla_{\boldsymbol{q}_i} \psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$. For the first term, when $\sigma_{ij} = 1$:

$$\begin{aligned}
&\nabla_{\boldsymbol{q}_i} \frac{\kappa R_{com}}{\left\|\boldsymbol{q}_{ij}\right\|_2 \left(R_{com} - \left\|\boldsymbol{q}_{ij}\right\|_2\right)} \\
&= -\frac{\kappa R_{com}}{\left\|\boldsymbol{q}_{ij}\right\|_2^2 \left(R_{com} - \left\|\boldsymbol{q}_{ij}\right\|_2\right)^2} \nabla_{q_i} \left(\left\|\boldsymbol{q}_{ij}\right\|_2 \left(R_{com} - \left\|\boldsymbol{q}_{ij}\right\|_2\right)\right) \\
&= -\frac{\kappa R_{com}}{\left\|\boldsymbol{q}_{ij}\right\|_2^2 \left(R_{com} - \left\|\boldsymbol{q}_{ij}\right\|_2\right)^2} \left[\nabla_{\boldsymbol{q}_i} \left\|\boldsymbol{q}_{ij}\right\|_2 \left(R_{com} - \left\|\boldsymbol{q}_{ij}\right\|_2\right) - \left\|\boldsymbol{q}_{ij}\right\|_2 \nabla_{\boldsymbol{q}_i} \left\|\boldsymbol{q}_{ij}\right\|_2\right] \qquad \text{(A.1)} \\
&= -\frac{\kappa R_{com}}{\left\|\boldsymbol{q}_{ij}\right\|_2^2 \left(R_{com} - \left\|\boldsymbol{q}_{ij}\right\|_2\right)^2} \left(R_{com} - 2\left\|\boldsymbol{q}_{ij}\right\|_2\right) \nabla_{\boldsymbol{q}_i} \left\|\boldsymbol{q}_{ij}\right\|_2 \\
&= -\frac{\kappa R_{com} \left(R_{com} - 2\left\|\boldsymbol{q}_{ij}\right\|_2\right)}{\left\|\boldsymbol{q}_{ij}\right\|_2^2 \left(R_{com} - \left\|\boldsymbol{q}_{ij}\right\|_2\right)^2} \frac{\boldsymbol{q}_{ij}}{\left\|\boldsymbol{q}_{ij}\right\|_2}
\end{aligned}$$

From there, it can also be inferred that:

$$\begin{aligned}\nabla_{\boldsymbol{q}_j}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|\right) &= \nabla_{\boldsymbol{q}_j}\frac{\kappa R_{com}}{\left\|\boldsymbol{q}_{ij}\right\|_2\left(R_{com}-\left\|\boldsymbol{q}_{ij}\right\|_2\right)} \\ &= -\frac{\kappa R_{com}\left(R_{com}-2\left\|\boldsymbol{q}_{ij}\right\|_2\right)}{\left\|\boldsymbol{q}_{ij}\right\|_2^2\left(R_{com}-\left\|\boldsymbol{q}_{ij}\right\|_2\right)^2}\nabla_{\boldsymbol{q}_j}\left\|\boldsymbol{q}_{ij}\right\|_2 \\ &= \frac{\kappa R_{com}\left(R_{com}-2\left\|\boldsymbol{q}_{ij}\right\|_2\right)}{\left\|\boldsymbol{q}_{ij}\right\|_2^2\left(R_{com}-\left\|\boldsymbol{q}_{ij}\right\|_2\right)^2}\frac{\boldsymbol{q}_{ij}}{\left\|\boldsymbol{q}_{ij}\right\|_2}\end{aligned} \tag{A.2}$$

since $\nabla_{\boldsymbol{q}_j}\left\|\boldsymbol{q}_{ij}\right\|_2 = -\nabla_{\boldsymbol{q}_i}\left\|\boldsymbol{q}_{ij}\right\|_2 = -\frac{\boldsymbol{q}_{ij}}{\left\|\boldsymbol{q}_{ij}\right\|_2}$

When $\sigma_{ij} = 0$:

$$\begin{aligned}\nabla_{\boldsymbol{q}_i}\sqrt{\left(\left\|\boldsymbol{q}_{ij}\right\|_2 - R_{com}+\epsilon\right)} &= \frac{1}{2\sqrt{\left(\left\|\boldsymbol{q}_{ij}\right\|_2 - R_{com}+\epsilon\right)}}\nabla_{\boldsymbol{q}_i}\left(\left\|\boldsymbol{q}_{ij}\right\|_2 - R_{com}+\epsilon\right) \\ &= \frac{1}{2\sqrt{\left(\left\|\boldsymbol{q}_{ij}\right\|_2 - R_{com}+\epsilon\right)}}\frac{\boldsymbol{q}_{ij}}{\left\|\boldsymbol{q}_{ij}\right\|_2}\end{aligned} \tag{A.3}$$

Following a similar reasoning than for the first term, observe that:

$$\nabla_{\boldsymbol{q}_i}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|\right) = -\nabla_{\boldsymbol{q}_j}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|\right) = \nabla_{\boldsymbol{q}_{ij}}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|\right) \tag{A.4}$$

Stability Analysis

Within this section, we examine under which conditions our safe interaction controller enables to sustain connectivity and avert collisions. Especially, we prove under certain assumptions on the flows that:

Theorem 1. *Consider a system of N agents governed by the dynamics defined in Chapter 8 Eq. (8.1) with holonomic actuation $g(\boldsymbol{q}, \boldsymbol{u}, t) = \boldsymbol{u}$, each steered by the LISIC policy Eq. (8.11). Under the assumption regarding the flow dynamics and maximum actuation in Eq. (A.16c), then:*

i. Each agent approaches a configuration that is the local minimum of its potential function $\sum_{\substack{j=1 \\ j\neq i}}^{N}\nabla_{\boldsymbol{q}_i}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$ i.e. the safe interaction input contribution of each agent in Eq. (8.10) approaches a local minimum.

ii. Inter-agents collisions are avoided

Proof. Energy-based analysis and LaSalle's invariance principle are commonly utilized to obtain analytical guarantees for flocking. These methods establish the stability of the system and demonstrate that the flock converges to a lattice structure while preventing inter-agent collisions, as demonstrated in [100]. Given the potential function in Eq. (8.8), we can infer the the *collective potential function* or *total tension energy* as:

$$H(\boldsymbol{q}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \psi \left(\left\| \boldsymbol{q}_{ij} \right\|_2 \right) \tag{A.5}$$

With the assumption of linear navigational feedback, the structural and translational dynamics can be decoupled [100]. However, if the navigational feedback is nonlinear, it is not possible to simply decompose the nominal controller into decoupled components without linearizing $\boldsymbol{u}_{perf}$ around a local neighborhood. As the agents have complex nonlinear dynamics, this approach is not feasible, and instead, a coupled form of structural dynamics is expressed. Specifically, we use a moving referential with respect to the flock centroid $\boldsymbol{q}_c$. The relative coordinates are given by $\tilde{\boldsymbol{q}}_i = \boldsymbol{q}_i - \boldsymbol{q}_c$ and $\tilde{\boldsymbol{q}}_{ij} = \tilde{\boldsymbol{q}}_i - \tilde{\boldsymbol{q}}_j = \boldsymbol{q}_{ij}$.

It follows that $\psi \left(\left\| \boldsymbol{q}_{ij} \right\|_2 \right) = \psi \left(\left\| \tilde{\boldsymbol{q}}_{ij} \right\|_2 \right)$. Thus, the potential energy for the structural dynamics in the relative coordinates yields:

$$H(\tilde{\boldsymbol{q}}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \psi \left(\left\| \tilde{\boldsymbol{q}}_{ij} \right\|_2 \right) \tag{A.6}$$

Finally, using the system dynamics in Eq. (8.1) and assuming holonomic actuation ($\boldsymbol{u}_i = g(\boldsymbol{q}_i, \boldsymbol{u}_i, t)$), the dynamics in the relative coordinates can be expressed as:

$$\begin{aligned}
\dot{\tilde{\boldsymbol{q}}}_i &= \dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_c && \text{(A.7a)} \\
&= \dot{\boldsymbol{q}}_i - \frac{1}{N} \sum_{l=1}^{N} \dot{\boldsymbol{q}}_l && \text{(A.7b)} \\
&\overset{(8.1)}{=} \boldsymbol{u}_i + v\left(\boldsymbol{q}_i\right) - \frac{1}{N} \sum_{l=1}^{N} \boldsymbol{u}_l + v\left(\boldsymbol{q}_l\right) && \text{(A.7c)} \\
&= \frac{N-1}{N} \left(\boldsymbol{u}_i + v\left(\boldsymbol{q}_i\right)\right) - \frac{1}{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \boldsymbol{u}_l + v\left(\boldsymbol{q}_l\right) && \text{(A.7d)} \\
&= \frac{N-1}{N} \left(\boldsymbol{u}_i + v\left(\boldsymbol{q}_i\right)\right) - \frac{1}{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \dot{\boldsymbol{q}}_l && \text{(A.7e)}
\end{aligned}$$

We can now compute the time-derivative of H along the trajectories of the agents as:

$$\dot{H} = \frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \dot{\psi} \left(\left\| \tilde{\boldsymbol{q}}_{ij} \right\| \right) \tag{A.8}$$

We can exploit the gradient relation result $\nabla_{\boldsymbol{q}_i}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right) = -\nabla_{\boldsymbol{q}_j}\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$ of Eq. (A.4) to derive the expression of $\dot{\psi}\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|\right)$

$$\sum_{i=1}^{N}\sum_{\substack{j=1\\j\neq i}}^{N}\dot{\psi}\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|\right) \tag{A.9a}$$

$$= \sum_{i=1}^{N}\sum_{\substack{j=1\\j\neq i}}^{N}\dot{\tilde{\boldsymbol{q}}}_i^\top\nabla_{\tilde{\boldsymbol{q}}_{ij}}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|\right) - \sum_{j=1}^{N}\sum_{\substack{i=1\\i\neq j}}^{N}\dot{\tilde{\boldsymbol{q}}}_j^\top\nabla_{\tilde{\boldsymbol{q}}_{ij}}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|\right) \tag{A.9b}$$

$$\overset{(A.4)}{=} \sum_{i=1}^{N}\sum_{\substack{j=1\\j\neq i}}^{N}\dot{\tilde{\boldsymbol{q}}}_i^\top\nabla_{\tilde{\boldsymbol{q}}_{i}}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|\right) + \sum_{j=1}^{N}\sum_{\substack{i=1\\i\neq j}}^{N}\dot{\tilde{\boldsymbol{q}}}_j^\top\nabla_{\tilde{\boldsymbol{q}}_{j}}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|\right) \tag{A.9c}$$

$$= 2\sum_{i=1}^{N}\sum_{\substack{j=1\\j\neq i}}^{N}\dot{\tilde{\boldsymbol{q}}}_i^\top\nabla_{\tilde{\boldsymbol{q}}_{i}}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|\right) \tag{A.9d}$$

where we also used the symmetry of the potential function $\psi(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|_2) = \psi(\left\|\tilde{\boldsymbol{q}}_{ji}\right\|_2)$. Then, substituting Eq. (A.9d) back into Eq. (A.8):

$$\dot{H} = \sum_{i=1}^{N}\sum_{\substack{j=1\\j\neq i}}^{N}\dot{\tilde{\boldsymbol{q}}}_i^\top\nabla_{\tilde{\boldsymbol{q}}_{i}}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|\right) \tag{A.10}$$

A possible approach, although conservative, is to show that a global tension energy decrease of the system $\dot{H} = \sum_{i=1}^{N}\dot{H}_i \leq 0$ can be achieved by guaranteeing local tension energy decrease $\forall i \in \mathcal{V}$. Assume that $G(t)$ switches at time t_l for $l = 0, 1, 2\ldots$ and $\dot{H} \leq 0$ on each $[t_l, t_{l+1})$. Then, at switching time k, $H(t_k) = H(t_k^-) + \psi\left(\|R_{com} - \epsilon\|\right)$ [146]. As the graph topology becomes fixed after a certain time and only a finite number of maximum edges can be added, the energy can be shown to be bounded for any subsequent time. The time-derivative of H_i along the trajectory of agent i yields:

$$\dot{H}_i = \dot{\tilde{\boldsymbol{q}}}_i^\top\underbrace{\sum_{\substack{j=1\\j\neq i}}^{N}\nabla_{\tilde{\boldsymbol{q}}_{i}}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|_2\right)}_{\boldsymbol{\phi}_i} \tag{A.11}$$

where to simplify notations for the rest of our development we introduce $\boldsymbol{\phi}_i = \sum_{\substack{j=1\\j\neq i}}^{N}\nabla_{\tilde{\boldsymbol{q}}_{i}}\psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|_2\right)$

Energy Analysis Substituting Eq. (A.7e) in the expression for $\dot{H}_i$ in Eq. (A.11):

$$\dot{H}_i = \dot{\boldsymbol{q}}_i^\top \boldsymbol{\phi}_i \tag{A.12a}$$

$$\overset{(A.7e)}{=} \left[\frac{N-1}{N} \left(\boldsymbol{u}_i + v\left(\boldsymbol{q}_i\right)\right) - \frac{1}{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \dot{\boldsymbol{q}}_l \right]^\top \boldsymbol{\phi}_i \tag{A.12b}$$

$$\overset{(8.11)}{=} \left[\frac{N-1}{N} \left(c_i^{(1)} \boldsymbol{u}_i^{safe} + c_i^{(2)} \boldsymbol{u}_i^{perf} + v\left(\boldsymbol{q}_i\right)\right) - \frac{1}{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \dot{\boldsymbol{q}}_l \right]^\top \boldsymbol{\phi}_i \tag{A.12c}$$

$$= \left[\frac{N-1}{N} \left(c_i^{(2)} \boldsymbol{u}_i^{perf} + v\left(\boldsymbol{q}_i\right)\right) - \frac{1}{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \dot{\boldsymbol{q}}_l \right]^\top \boldsymbol{\phi}_i + \frac{N-1}{N} c_i^{(1)} \left(\boldsymbol{u}_i^{safe}\right)^\top \boldsymbol{\phi}_i \tag{A.12d}$$

Recall our safe interaction policy $\boldsymbol{u}_i^{safe} = (\pi_{safe})_i(\boldsymbol{q})$ where replacing $\boldsymbol{\phi}_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} \nabla_{\boldsymbol{q}_i} \psi \left(\left\| \boldsymbol{q}_{ij} \right\|_2 \right)$ in Eq. (8.10) yields:

$$\boldsymbol{u}_i^{safe} = -\frac{\boldsymbol{\phi}_i}{\left\|\boldsymbol{\phi}_i\right\|_2} U_{max,i}$$

Therefore:

$$\left(\boldsymbol{u}_i^{safe}\right)^\top \boldsymbol{\phi}_i = -\left\|\boldsymbol{\phi}_i\right\|_2 U_{max,i} \tag{A.13}$$

To bound $\dot{H}_i$, A.12d can be reformulated using the inner product

$$\dot{H}_i = -\left\langle \frac{N-1}{N} \left(c_i^{(2)} \boldsymbol{u}_i^{perf} + v\left(\boldsymbol{q}_i\right)\right) - \frac{1}{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \dot{\boldsymbol{q}}_l \ , \ \boldsymbol{\phi}_i \right\rangle - \frac{N-1}{N} c_i^{(1)} \left\|\boldsymbol{\phi}_i\right\|_2 U_{max,i} \tag{A.14a}$$

$$\leq \left| \left\langle \frac{N-1}{N} \left(c_i^{(2)} \boldsymbol{u}_i^{perf} + v\left(\boldsymbol{q}_i\right)\right) - \frac{1}{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \dot{\boldsymbol{q}}_l \ , \ \boldsymbol{\phi}_i \right\rangle \right| - \frac{N-1}{N} c_i^{(1)} \left\|\boldsymbol{\phi}_i\right\|_2 U_{max,i} \tag{A.14b}$$

$$\overset{Cauchy-Schwarz}{\leq} \left\| \frac{N-1}{N} \left(c_i^{(2)} \boldsymbol{u}_i^{perf} + v\left(\boldsymbol{q}_i\right)\right) - \frac{1}{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \dot{\boldsymbol{q}}_l \right\|_2 \left\|\boldsymbol{\phi}_i\right\|_2 - \frac{N-1}{N} c_i^{(1)} \left\|\boldsymbol{\phi}_i\right\|_2 U_{max,i} \tag{A.14c}$$

So $\dot{H}_i$ can be expressed as

$$\dot{H}_i \leq \left(\left\| \frac{N-1}{N} \left(c_i^{(2)} \boldsymbol{u}_i^{perf} + v\left(\boldsymbol{q}_i\right)\right) - \frac{1}{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \dot{\boldsymbol{q}}_l \right\|_2 - \frac{N-1}{N} c_i^{(1)} U_{max,i} \right) \underbrace{\left\|\boldsymbol{\phi}_i\right\|_2}_{\geq 0} \tag{A.15}$$

Then, a local tension energy decrease $\dot{H}_i \leq 0$ can be guaranteed from Eq. (A.15) if

$$\left\|\frac{N-1}{N}\left(c_i^{(2)}\boldsymbol{u}_i^{perf}(\boldsymbol{q}_i)+v\left(\boldsymbol{q}_i\right)\right)-\frac{1}{N}\sum_{\substack{l=1\\l\neq i}}^{N}\dot{\boldsymbol{q}}_l\right\|_2 \leq \frac{N-1}{N}c_i^{(1)}U_{max,i} \tag{A.16a}$$

$$\Leftrightarrow \left\|c_i^{(2)}\boldsymbol{u}_i^{perf}(\boldsymbol{q}_i)+v\left(\boldsymbol{q}_i\right)-\frac{1}{N-1}\sum_{\substack{l=1\\l\neq i}}^{N}\dot{\boldsymbol{q}}_l\right\|_2 \leq c_i^{(1)}U_{max,i} \tag{A.16b}$$

$$\Leftrightarrow \left\|c_i^{(2)}\boldsymbol{u}_i^{perf}(\boldsymbol{q}_i)+v\left(\boldsymbol{q}_i\right)-\underbrace{\text{Ave}\left(\dot{\boldsymbol{q}}_{\mathcal{N}_i}\right)}_{\text{Ave}\left(v(\boldsymbol{q}_{Ni})\right)+\text{Ave}\left(\boldsymbol{u}_{Ni}\right)}\right\|_2 \leq c_i^{(1)}U_{max,i} \tag{A.16c}$$

where Ave$(\cdot)$ denotes the average and the set $\mathcal{N}_i = \mathcal{V}\setminus\{i\}$ the neighboring agents of i. The dynamics of the other agents are also defined by their surrounding flow and their individual control inputs such that $\text{Ave}\left(\dot{\boldsymbol{q}}_{\mathcal{N}_i}\right) = \text{Ave}\left(v(\boldsymbol{q}_{\mathcal{N}_i})\right) + \text{Ave}\left(\boldsymbol{u}_{\mathcal{N}_i}\right)$. As the other agents also apply the LISIC policy Eq. (8.11):

$$\text{Ave}\left(\dot{\boldsymbol{q}}_{\mathcal{N}_i}\right) = \frac{1}{N-1}\sum_{\substack{j=1\\j\neq i}}^{N}\left(v(\boldsymbol{q}_j)+c_j^{(1)}\boldsymbol{u}_j^{safe}+c_j^{(2)}\boldsymbol{u}_j^{perf}(\boldsymbol{q}_j)\right) \tag{A.17}$$

Since $c_i^{(1)} \in [0,1]$, $c_i^{(1)}U_{max} \leq U_{max}$ then the maximum control input that can be applied is bounded by $U_{max} > 0$. In the *overactuated* setting, the condition Eq. (A.16c) could be satisfied independently of the possible scenarios for currents $v(\boldsymbol{q}_i)$ and neighbor dynamics $\text{Ave}\left(\dot{\boldsymbol{q}}_{\mathcal{N}_i}\right)$.

Nevertheless, the agents do not necessarily need to be overactuated despite strong flows to achieve a local energy decrease ($\dot{H}_i \leq 0$). If the currents experienced by agent i are of similar magnitude and direction to the average current experienced by the neighboring agents, then $v\left(\boldsymbol{q}_i\right)$ compensates $\text{Ave}\left(v(\boldsymbol{q}_{Ni})\right)$ and Eq. (A.16c) can be fulfilled even if $\left\|v\left(\boldsymbol{q}_i\right)\right\|_2 > U_{max,i}$. The neighboring flocking control inputs $\text{Ave}\left(\boldsymbol{u}_{Ni}\right)$ also helps accounting for the current difference term $v\left(\boldsymbol{q}_i\right) - \text{Ave}\left(v(\boldsymbol{q}_{Ni})\right)$. If $\left\|v\left(\boldsymbol{q}_i\right) - \text{Ave}\, v(\boldsymbol{q}_{Ni})\right\|_2 \gg U_{max,i}$, satisfying Eq. (A.16c) becomes challenging, which can happen if agents experience strong divergent currents.

To illustrate this behavior, consider the case where an agent i is about to lose communication with the rest of the flock or to collide with one of the other members. Then, the potential function $\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$ tends to very large values (see Fig. 8.2). Therefore, by definition of our weighting function $\alpha(\boldsymbol{q})$ in Eq. (8.12), the weighted coefficients $c_i^{(1)} \rightarrow 1$ and $c_i^{(2)} \rightarrow 0$ and Eq. (A.16c) asymptotically yields:

$$\left\|v\left(\boldsymbol{q}_i\right)-\text{Ave}\left(\dot{\boldsymbol{q}}_{\mathcal{N}_i}\right)\right\|_2 \leq U_{max} \tag{A.18a}$$

$$\Leftrightarrow \left\|v\left(\boldsymbol{q}_i\right)-\text{Ave}\left(v(\boldsymbol{q}_{\mathcal{N}_i})\right)-\text{Ave}\left(\boldsymbol{u}_{\mathcal{N}_i}\right)\right\|_2 \leq U_{max} \tag{A.18b}$$

where we assumed that all agents are identical in terms of hardware design so that they also share the same maximum thrust U_{max}. Therefore if the currents experienced by agent i are of similar magnitude and direction than the average currents experienced by the neighboring agents $v\left(\boldsymbol{q}_i\right) - \text{Ave}\left(v(\boldsymbol{q}_{\mathcal{N}_i})\right)$ would tend to cancel out, while $(\boldsymbol{u}_{\mathcal{N}_i})$ cannot exceed U_{max}.

When the ideal spacing is achieved in the opposite scenario, the safe interaction controller's weighting function $\alpha(\boldsymbol{q})$ gives priority to the performance controller, resulting in $c_i^{(1)} \to 0$ and $c_i^{(2)} \to 1$. If the performance control input is applied alone and does not disturb the ideal flock configuration, the energy of H_i should not increase. In such cases, $\boldsymbol{\phi}_i$ will asymptotically approach zero, and $\dot{H}_i = 0$, as shown in equation (A.15).

Discussion 1. *The analysis presented applies to any platform and is supported by the finding that if every agent undergoes a decrease in local energy $\dot{H}_i \leq 0$, then it can be ensured that there will be a decrease in global energy as well. Note that condition Eq.* (A.16c) *is sufficient but not necessary to guarantee $\dot{H} < 0$, as negative local energies can compensate for positive ones.*

LISIC Stability Now that the proposed flocking law has been proven to generate an energy decrease under certain assumptions, we can derive the implication for the *graph connectivity*. Following the reasoning of [146], assume that the Graph $G(t)$ switches at time t_l for $l = 0, 1, 2\ldots$, such that the graph topology is invariant within each of the non-empty bounded and non-overlapping time intervals $[t_l, t_{l+1})$. Under the assumption that the previously highlighted conditions for an energy decrease can be satisfied, then

$$H(t) \leq H(t_0) < \infty, \quad \forall t \in [t_0, t_1) \tag{A.19}$$

since $\dot{H} \leq 0$ and assuming the initial energy $H(t_0)$ is finite. As the potential function $\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$ tends to ∞ whenever $\left\|\boldsymbol{q}_{ij}\right\|_2 \to R_{com}$ no existing edges will be lost at time t_1. Nevertheless, new edges might be added at switching time t_1, t_2 etc. Without loss of generality, assume that N_1 new links are added at t_1. The maximum of edges that can be added is $N_{max} = \frac{N(N-1)}{2}$ where N is the number of agents and clearly $0 \leq N_1 \leq N_{max}$. Then

$$H(t_1) = H(t_0) + N_1\psi\left(\|R_{com} - \epsilon\|\right) < \infty \tag{A.20}$$

where ϵ appears from the hysteresis process for adding edges in Eq. (8.9). A similar discussion can be led for t_l, $l = 2, 3, \ldots$ and it can be concluded that under the assumption of an initial finite energy $H(t_0)$ and $\|\boldsymbol{u}_i\| \geq L$, $\forall i$ $\forall t$ that $H(t_l)$ is bounded. This implies that

$$H(t) < H(t_{l-1}) < \infty \quad t \in [t_{l-1}, t_l], l = 1, 2, \ldots \tag{A.21}$$

Therefore, no edges will be lost at switching times and if $G(t_0)$ is connected then $G(t)$ will stay connected $\forall t > 0$. As time evolves, the interaction network $G(t)$ eventually becomes fixed for $t \geq t_l$, i.e. when there are no more edges to add and the graph is complete. As the energy is still minimized, the graph should converge to a lattice configuration, achieving the optimal distance between agents, defined as the minimum of the potential function $\psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right)$ i.e. $\frac{R_{com}}{2}$. Precisely, the energy can be bounded by $H_{max} = H(t_0) + N_{max}\psi\left(\|R_{com} - \epsilon\|\right)$, assuming no edges are lost and Eq. (A.21) holds. Therefore, for $t \geq 0$ there must be a set such that

$$\Omega = \left\{\tilde{\boldsymbol{q}} \in \mathbb{R}^{Nn} \mid H(\tilde{\boldsymbol{q}}) \leq H_{max}\right\} \tag{A.22}$$

with N the number of agents and n the number of coordinates considered to describe the agent positions, $H(\tilde{\boldsymbol{q}})$ the total potential energy and assuming Eq. (A.16c) can be satisfied at all time and for all agents. Then LaSalle Invariance Principle can be applied [140], to show that the trajectories of the agent converge to the largest invariant set inside the region defined by

$$\Gamma = \left\{\tilde{\boldsymbol{q}} \in \mathbb{R}^{Nn} \mid \dot{H}(\tilde{\boldsymbol{q}}) = 0\right\} \tag{A.23}$$

In summary, we have shown that the total tension energy of the system decreases under assumption Eq. (A.16c). This is evident from the fact that $\dot{H} = \sum_{i=1}^{N} \dot{H}_i = \sum_{i=1}^{N} \dot{\tilde{\boldsymbol{q}}}_i^\top \boldsymbol{\phi}_i$ as stated in Eq. (A.11). In the steady state of the network configuration, where $\dot{H} = 0$, this can be achieved if $\boldsymbol{\phi}_i = 0$ for all agents $i \in \mathcal{V}$, i.e. $\sum_{\substack{j=1 \\ j \neq i}}^{N} \nabla_{\tilde{\boldsymbol{q}}_i} \psi\left(\left\|\tilde{\boldsymbol{q}}_{ij}\right\|_2\right) = 0$.

In other words, this proves that the potential energy of the system, guided by the control law in Eq. (8.11), decreases until reaching a fixed network configuration (steady-state) in the moving frame, which corresponds to a minimum of each agent's local potential function. This establishes part **i.** of Theorem 1. As an implication for the LISIC policy, this means that low interference is achieved with each agent's performance controller, as long as the flock structure is not disturbed.

Moreover, collisions can also be avoided under the previous assumptions. Indeed, $\lim_{\boldsymbol{q}_{ij} \to 0} \psi\left(\left\|\boldsymbol{q}_{ij}\right\|_2\right) = \infty$ would contradict Eq. (A.22) with $H(\tilde{\boldsymbol{q}}) \leq H_{max}$. This proves part **ii.** of Theorem 1.

□