Autonomous flow-based navigation in unsteady underwater environments

Thesis by
Peter Gunnarson

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
Defended 31 May 2024

Peter Gunnarson
ORCID: https://orcid.org/0000-0002-4437-5379

ACKNOWLEDGMENTS

While the work in this dissertation represents the cumulative efforts of my time as a graduate student, it would not have been possible without a large supporting network of colleagues, friends, and mentors, to whom I am forever indebted for their exceptional generosity and kindness.

I am thankful to my advisor, John Dabiri, for welcoming me into his research group and providing invaluable mentorship, support, and encouragement. My skills as a scientist have been sharpened by his high standards in scholarship, writing, and the scientific process, which I hope to embody in my future endeavors. I will always be inspired by his never-ending scientific curiosity, boundless energy and enthusiasm, and most importantly, relentless optimism.

I would also like to thank the members of my dissertation committee, Jane Bae, Joel Burdick, and Mory Gharib for their generous discussions and support throughout my time as graduate student at Caltech. I was also fortunate to collaborate with Petros Koumoutsakos, Guido Novati, and Ioannis Mandralis, and am thankful for their partnership and for graciously introducing me to the techniques of machine learning. I am also grateful to my undergraduate research advisor, Dan Quinn, and graduate student mentor, Qiang Zhong, for their mentorship and generosity while starting my journey as a scientist.

Throughout my time as a graduate student, the members of the Dabiri lab have made it joy to come into work every day: Simon Anuszczyk, Jen Cardona, Malaika Cordeiro, Matt Fu, Roni Goldshmid, Mike Howland, Nina Mohebbi, Kelsi Rutledge, Nathan Wei, Nicole Xu, and Noa Yoder. I am grateful for their enduring friendship and constant collaboration, especially with solving the problems inherent to working in a lab filled with more water tanks than graduate students.

The entire GALCIT community has been incredibly welcoming and supporting. In particular, my first year class, Brayden Aller, John Pederson, Jimmy Ragan, Miles Chan, Nikhil Ranganathan, Niyati Desai, Sahangi Dassanayake, Sorina Lupu, Tanner Harms, and Ying Lou, have been a bedrock of camaraderie throughout my time at Caltech. Without their support, I would not have made it through the first year of GALCIT, and I am fortunate to have forged lifelong connections with them.

Additionally, it must be stated that none of the work in this thesis was possible without the Herculean efforts of the administrative staff in GALCIT, including Sarah

Pontes, Martha Salcedo, Jamie Meighen-Sei, and Christine Ramirez. GALCIT is fortunate to have such exceptional and skilled staff who keep the department running smoothly.

I would like to thank my family for their constant support, love, and encouragement throughout my entire life. I am grateful to my dad for instilling within me a curiosity of nature and the mechanical world. He taught me that the best way to get to the bottom of a question is to build something and test it in real life, which explains why I've always felt home as an experimentalist. I greatly admire and strive to replicate his commitment to being an engineer with integrity, and dedication to protecting the environment and natural world around us. I am thankful for my mom, who inspired me to pursue graduate school and a PhD. My mom has always represented the values of research and scientific methods, while being incredibly loving, empathetic, and supportive. I am grateful that my parents always believed in learning as a family activity, providing me with endless opportunities to explore science and engineering at my own pace. I am also thankful for my brother, Jacob, who I look up to as role model of a scholar, scientist, and a writer. Throughout my life, he has imparted his infectious love for the natural world, and especially, the ocean. My entire family, including my grandfather, John Gunnarson, has been a constant source inspiration as I have pursued engineering and graduate school.

Lastly, I would like to thank my wife and the love of my life, Susan. I met her before I even knew that I wanted to go to graduate school, and she has supported me every step of the way. In addition to graciously enduring my practice presentations and managing a difficult commute to UCLA during her time as a graduate student, Susan has provided me with guidance, love, and support every day that we have known each other. Her creativity as an artist inspires my creativity as a scientist, and our commitment to each other motivates me to work hard and strive for excellence. I owe Susan deeply for her immense contributions to this body of work, and to my life and well-being. To Susan, as I finish my time as a graduate student, the thought foremost in my mind is how truly lucky I am to have met you. I could not have asked for a better person to be with as we start this next chapter in our lives.

ABSTRACT

Autonomous ocean-exploring robots promise to significantly enhance the rate at which we can explore ocean environments. However, the limited range and speed of existing autonomous underwater vehicles (AUVs) are barriers to comprehensive ocean exploration. To address these limitations, the work in this thesis investigates strategies for improving the capabilities of existing AUVs, such as targeted sampling and efficient navigation through background flows. Inspired by the ability of aquatic animals to navigate via flow sensing, hydrodynamic cues are investigated as a sensory input for accomplishing these feats of autonomous navigation using only onboard sensors. First, reinforcement learning (RL) is investigated as an algorithm for accomplishing efficient point-to-point navigation in simulated cylinder flow. The algorithm entails inputting point measurements of flow quantities such as velocity and vorticity into a deep neural network, which then determines a swimmer's actions. Using point velocity as the sensory input, the RL algorithm achieved a near 100% success rate in reaching the target locations while approaching the time-efficiency of optimal navigation trajectories. To test RL and flow-based navigation in a physical setting, we next developed the Caltech autonomous reinforcement learning robot (CARL), a palm-sized underwater robotic platform. As proof-of-concept analogy for tracking hydrothermal vent plumes in the ocean, the robot was tasked with locating the center of turbulent jet flows in a 13,000-liter water tank using data from onboard pressure sensors. Using a navigation policy trained with RL in a simulated flow environment, CARL successfully located the turbulent plumes at more than twice the rate of random searching by detecting mean flow gradients with the onboard pressure sensors. Lastly, combing both flow sensing and efficient navigation, the accelerometer onboard CARL was used to sense and exploit the flow from a passing vortex ring for energy-efficient propulsion. Body acceleration and rotation were shown to be effective methods of indirect flow sensing, which enabled the energy-efficient vortex ring surfing strategy. Throughout this work, efforts are made to understand the governing physics behind the discovered navigation strategies to generalize the results beyond a specific navigation problem, sensor type, or robotic implementation.

PUBLISHED CONTENT AND CONTRIBUTIONS

1. Gunnarson, P., and J.O. Dabiri (2024). "Fish-inspired tracking of underwater turbulent plumes". *Submitted*. doi: 10.48550/arXiv.2403.06091.
   P.G. and J.O.D. conceived of project, P.G. conducted experiments, P.G. and J.O.D. analyzed results and wrote paper.

2. Gunnarson, P., I. Mandralis, G. Novati, P. Koumoutsakos, and J.O. Dabiri (2021). "Learning efficient navigation in vortical flow fields". In: *Nature Communications* 12.1. doi: 10.1038/s41467-021-27015-y.
   P.G., I.M., G.N., P.K., and J.O.D. designed research and were involved in discussions to interpret the results; P.G. performed research and analyzed results; G.N. and P.K. developed the V-RACER algorithm; G.N. wrote the software implementation of V-RACER; I.M. simulated the cylinder flow field; P.G. drafted the paper, and all authors helped edit and review.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Number *Page*

Chapter 1

INTRODUCTION

1.1 Challenges in ocean exploration

There is a critical need for comprehensive sampling and exploration of the ocean. In addition to representing 99% of habitable volume on Earth and a chief source of food, transportation, and trade in the global economy (OECD, 2020), the ocean plays a primary role in climate change through carbon sequestration (T. DeVries et al., 2019). Detailed observations of the ocean and the biological processes within are required at both high spatial and temporal resolution in order to understand the ocean as it changes through the accelerating pace of climate change (Benway et al., 2019). Yet, after centuries of effort, 80% of the seafloor remains unexplored (Wölfl et al., 2019) and an estimated 90% of species biodiversity is unstudied (Reaka-Kudla, 2001). While satellites and remote sensing can provide a wealth of data for understanding oceanic properties, measurements of critical quantities such as vertical profiles of salinity and temperature are only accessible via direct, in situ measurements in the bulk of the ocean. For the same reason, maps of the seafloor are poorer in resolution than those of the surface of Mars and Venus (Copley, 2014).

Autonomous underwater vehicles (AUVs) have emerged as a powerful tool at the forefront of tackling this grand challenge of comprehensive ocean exploration. Fleets of undersea gliders (Rudnick, 2016) and autonomous floats (Wong et al., 2020) can sample large regions of the ocean without requiring expensive ship time and constant communication to the surface. The spread of small autonomous robots is fueled by the miniaturization of electronics, sensors, and energy-dense batteries, which have dramatically reduced the cost and difficulty of developing such systems.

However, even with advances in low-cost AUVs, the challenge of comprehensive ocean coverage appears insurmountable in the face of the overwhelmingly large volume of the ocean, measuring 1.3 billion cubic kilometers (Charette and Smith, 2010). Even with a million AUVs distributed evenly through the ocean, each AUV would be required to cover over 1,000 square kilometers of area and a kilometer in depth (Xu and John Dabiri, 2022). In addition to spatial resolution, the movement of water due to ocean currents and the changing pace of climate change necessitates that the entire volume is sampled repeatedly in time (Benway et al., 2019). With

limited range and speed, it may not be feasible to accomplish widespread coverage of the ocean with current AUV technology (Zereik et al., 2018). New approaches are needed to enable a better understanding of the ocean. At the same time as researchers investigate new types of underwater robots (e.g., biohybrid AUVs (Xu and John Dabiri, 2022)), this thesis investigates opportunities for intelligent maneuvering and sensing to enhance the capabilities of existing and future AUVs.

1.2 Efficient navigation for enhancing AUV capabilities

The range and speed of ocean-exploring AUVs could be significantly improved through more intelligent navigation algorithms. For example, one such approach is the idea of targeted sampling, in which robots use information from onboard sensors to actively seek out areas of scientific interest such as undersea thermal vents, coastal upwelling fronts, or phytoplankton patches (Zhang, Ryan, et al., 2019). For example, AUVs have used machine-vision algorithms to track animals (Katija et al., 2021), chemical sensors to locate undersea thermal vents (Preston et al., 2022), and salinity sensors to measure Gulf Stream salinity fronts (Zhang, Yoder, et al., 2022). By actively steering towards areas of high scientific value, AUVs can sample information-rich locations more quickly and better allocate their limited energy supplies.

Another potential area for optimization is to take advantage of existing ocean currents and background flows for efficient propulsion. Rather than fighting against these currents, robots have the potential to save energy and gain speed by riding the background flows naturally occurring throughout the ocean. In combination with targeted sampling, better navigation algorithms have the potential to improve both the speed and energy-efficiency of AUVs.

However, existing algorithms for accomplishing these techniques often require information that is not available to AUVs. For example, path planning algorithms from control theory typically require full knowledge of the background flow field to compute energy or time-efficient trajectories (Zermelo, 1931; Techy, 2011; Petres et al., 2007; Lolla et al., 2014). Not only is a full map of the background flow often difficult or impossible to obtain in real time, but communication with AUVs under the ocean surface is effectively blocked by the salt water. Without a direct tether or acoustic link to the surface, meaningful amounts of data transfer through the bulk of the ocean volume is virtually impossible with current technologies. Additionally, scientifically valuable targets in the ocean (e.g., plankton

patches, salt-intrusion fronts) are constantly changing and being transported by the ocean currents, which are themselves also changing in time. Somehow, autonomous underwater robots must be equipped with a navigation strategy that can accomplish target sampling and efficient navigation using only the information of the robot's immediate surroundings.

1.3 Looking to nature for inspiration: opportunities enabled by flow sensing

Nature offers a potential solution to help solve the problem of autonomous underwater navigation: flow sensing. The ability to sense fluid flow is ubiquitous among aquatic organisms and is used to accomplish a wide range of tasks to great effect, including tracking the wakes of prey in complete darkness (Pohlmann, Grasso, and Breithaupt, 2001; Dehnhardt, Mauck, Hanke, et al., 2001), obstacle avoidance (Oteiza et al., 2017), and following walls (Patton, Windsor, and Coombs, 2010). These biological tasks could inspire similar functionalities in AUVs for targeted sampling, such as detecting the wake signatures from physical obstacles or hydrothermal vents. Tracking these oceanic features with flow sensing also has the benefit of functioning even when poor water visibility would otherwise obscure visual tracking. Likewise, sensing local flow information could be used for efficient navigation and path planning. For example, fish use flow sensing to station keep in the wake behind obstacles (Liao, 2007) and for energy-efficient schooling behavior (Weber et al., 2020).

Inspired by the potential of flow sensing in underwater robots, scientists and engineers have developed bio-inspired flow sensors that mimic the function of their biological counterparts (Ko, Lauder, and Nagpal, 2023). Examples include hair-like sensors that mimic superficial neuromasts in fish lateral lines (Bora et al., 2018), pressure sensors that mimic canal neuromasts (Zhai, Zheng, and Xie, 2021), and sensors inspired by the geometry of sea lion whiskers (Zheng, Kamat, et al., 2021).

While these sensors have achieved high accuracy, interpreting flow measurements for autonomous navigation remains an open challenge. Some studies have used potential flow models to model the mapping between background flow and sensor inputs (Dagamseh et al., 2013; Zheng, Wang, Xiong, et al., 2020) or heuristic strategies for turning in response to flow information (Monthiller et al., 2022; Salumäe et al., 2012). As will be discussed in the subsequent sections, reinforcement learning (RL) is a new and promising technique for flow-based navigation (e.g., Reddy et al., 2018). In general, however, unifying design principles for swimming strategy,

sensor type, and sensor placement remain unclear. There is significant unrealized potential for implementing onboard flow sensing for both targeted sampling and efficient navigation.

1.4 Aims and objectives of the dissertation

Given the success of flow sensing in aquatic animals and the potential for robotic applications, the aim of this thesis is as follows: how can autonomous underwater robots use flow sensing to detect and exploit background fluid flows? This question is addressed in this dissertation though a combination of computational and experimental methods.

A main computational technique used in Chapters 2 and 3 is deep RL. Through repeated interactions with an environment, an RL algorithm trains an agent to take actions that maximize a reward function (Sutton et al., 2000). Deep RL utilizes deep neural networks to represent the policy, which is the function that maps sensor inputs to the actions of the agent. RL promises to discover navigation strategies from data without the need for human fine-tuning. For example, by using a reward function tailored to an ocean-exploration task, such as minimizing the time required to reach a target, Chapter 2 demonstrates that deep RL can learn how to incorporate point measurements of fluid properties such as velocity and vorticity for time-efficient navigation through simulated cylinder flow. Appendix A.2 includes details of the RL algorithm and its implementation.

While RL has the potential to help solve flow-based navigation problems, the applicability of a trained neural network is limited if it does not generalize to real-world systems. And in practice, policies learned in simulation often do not transfer to physical robots due to sensor noise and unmodeled dynamics (Dulac-Arnold et al., 2021), which often necessitates testing and additional training in physical robots (Hasselmann et al., 2021). To address the need for experimental verification, a significant contribution of this thesis is the development of the Caltech autonomous reinforcement learning robot (CARL), a scratch-built underwater robotic platform built to test RL and flow-based navigation in an experimental setting. In Chapter 3, CARL is used to demonstrate the transfer of a learned policy for targeted sampling of turbulent plumes into a physical robot and flow environment. Additionally, pressure sensors are investigated as a low-cost, miniature flow sensor for applications in flow-based navigation.

Chapter 2 focuses on efficient navigation, and Chapter 3 focuses on onboard flow sensing. As a synthesis of both sensing and navigation problems, Chapter 4 investigates using acceleration-based flow sensing to detect and exploit the flow from a passing vortex ring for energy-efficient propulsion. Using an inexpensive (~$5) accelerometer, body acceleration is shown to be an effective means of sensing the background flow acceleration, which provides sufficient information to maneuver onto and "surf" vortex rings for forward propulsion.

More broadly, we seek to understand the fundamental physics principles that generalize beyond a specific navigation problem, sensor type, or robotic implementation. For example, in Chapter 2, variations of sensor noise, sensor type, and policy transfer between flows are investigated to probe questions of robustness and generalizability. In Chapter 3, the gradient-based mechanism of the trained policy is interrogated and made interpretable using Shapley values (Lundberg and Lee, 2017), which enables successful transfer to the physical robot while accounting for sensor noise. Additionally, limitations due to sensor noise are quantified by modeling sources of noise and turbulence from the background flow. Finally, in Chapter 4, Lagrangian coherent structures are used to explain the fluid mechanical principles behind exploiting vortex rings for propulsion. Additionally, the background flow acceleration is used to analytically model the flow signals sensed by CARL, and rotation is suggested as an additional signal for detecting background vorticity.

Chapter 2

LEARNING EFFICIENT NAVIGATION IN VORTICAL FLOW FIELDS

The contents of this chapter have been adapted from Gunnarson et al. (2021), published in Nature Communications under the title "Learning efficient navigation in vortical flow fields". As the first author, Peter Gunnarson made the primary contributions towards performing the research and data analysis. All authors designed the research and were involved in discussions to interpret the results. Petros Koumoutsakos developed the V-RACER algorithm. Guido Novati wrote the software implementation of V-RACER. Ioannis Mandralis simulated the cylinder flow field. Peter Gunnarson drafted the paper, and all authors helped edit and review.

Abstract

Efficient point-to-point navigation in the presence of a background flow field is important for robotic applications such as ocean surveying. In such applications, robots may only have knowledge of their immediate surroundings or be faced with time-varying currents, which limits the use of optimal control techniques. Here, we apply a recently introduced RL algorithm to discover time-efficient navigation policies to steer a fixed-speed swimmer through unsteady two-dimensional flow fields. The algorithm entails inputting environmental cues into a deep neural network that determines the swimmer's actions, and deploying Remember and Forget Experience Replay. We find that the resulting swimmers successfully exploit the background flow to reach the target, but that this success depends on the sensed environmental cue. Surprisingly, a velocity sensing approach significantly outperformed a bio-mimetic vorticity sensing approach, and achieved a near 100% success rate in reaching the target locations while approaching the time-efficiency of optimal navigation trajectories.

2.1 Introduction

Navigation in the presence of a background unsteady flow field is an important task in a wide range of robotic applications, including ocean surveying (Weizhong Zhang et al., 2008), monitoring of deep-sea animal communities (Kuhnz et al., 2020), drone-based inspection and delivery in windy conditions (Guerrero and Bestaoui, 2013), and weather balloon station keeping (Bellemare et al., 2020). In such applications, robots must contend with unsteady fluid flows such as wind gusts or ocean currents in order to survey specific locations and return useful measurements, often autonomously. Ideally, robots would exploit these background currents to propel themselves to their destinations more quickly or with lower energy expenditure

If the entire background flow field is known in advance, numerous algorithms exist to accomplish optimal path planning, ranging from the classical Zermelo's equation from optimal control theory (Zermelo, 1931; Techy, 2011) to modern optimization approaches (Panda et al., 2020; Kularatne, Bhattacharya, and Hsieh, 2018; Guerrero and Bestaoui, 2013; Weizhong Zhang et al., 2008; Petres et al., 2007; Lolla et al., 2014). However, measuring the entire flow field is often impractical, as ocean and air currents can be difficult to measure and can change unpredictably. Robots themselves can also significantly alter the surrounding flow field, for example, when multi-rotors fly near obstacles (Shi et al., 2019) or during fish-like swimming (Verma, Novati, and Koumoutsakos, 2018). Additionally, oceanic and flying robots are increasingly operated autonomously and therefore do not have access to real-time external information about incoming currents and gusts (e.g., (Fiorelli et al., 2006; Caron et al., 2008)).

Instead, robots may need to rely on data from on-board sensors to react to the surrounding flow field and navigate effectively. A bio-inspired approach is to navigate using local flow information, for example, by sensing the local flow velocity or pressure. Zebrafish appear to use their lateral line to sense the local flow velocity and avoid obstacles by recognizing changes in the local vorticity due to boundary layers (Oteiza et al., 2017). Some seal species can orient themselves and hunt in total darkness by detecting currents with their whiskers (Dehnhardt, Mauck, and Bleckmann, 1998). Additionally, a numerical study of fish schooling demonstrated how surface pressure gradient and shear stress sensors on a downstream fish can determine the locations of upstream fish, thus enabling energy-efficient schooling behavior (Weber et al., 2020).

Reinforcement Learning (RL) offers a promising approach for replicating this feat

of navigation from local flow information. In simulated environments, RL has successfully discovered energy-efficient fish swimming (Gazzola, Hejazialhosseini, and Koumoutsakos, 2014; Jiao et al., 2020) and schooling behavior (Verma, Novati, and Koumoutsakos, 2018), as well as a time-efficient navigation policy for a repeated, deterministic snapshot of turbulent flow using position information (Biferale et al., 2019). In application, RL using local wind velocity estimates outperformed existing methods for energy-efficient weather balloon station keeping (Bellemare et al., 2020) and for replicating bird soaring (Reddy et al., 2018). Other methods exist for navigating uncertainty in a partially known flow field such as fuzzy logic or adaptive control methods (Panda et al., 2020). Finite-horizon model predictive control has been also used to plan energy-efficient trajectories using partial knowledge of the surrounding flow field (Krishna, Song, and Brunton, 2022). However, RL can be applied generally to an unknown flow field without requiring human tuning for specific scenarios.

The question remains, however, as to which environmental cues are most useful for navigating through flow fields using RL. A biomimetic approach suggests that sensing the vorticity could be beneficial (Oteiza et al., 2017); however, flow velocity, pressure, or quantities derived thereof are also viable candidates for sensing.

In this work, we find that deep RL can indeed discover time-efficient, robust paths through an unsteady, two-dimensional (2D) flow field using only local flow information, where simpler strategies such as swimming towards the target largely fail at the task. We find, however, that the success of the RL approach depends on the type of flow information provided. Surprisingly, a RL swimmer equipped with local velocity measurements dramatically outperforms the bio-mimetic local vorticity approach. These results show that combining RL-based navigation with local flow measurements can be a highly effective method for navigating through unsteady flow, provided the appropriate flow quantities are used as inputs to the algorithm.

2.2 Results

2.2.1 Simulated navigation problem

As a testing environment for RL-based navigation, we pose the problem of navigating across an unsteady von Kármán vortex street obtained by simulating 2D, incompressible flow past a cylinder at a Reynolds number of 400. Other studies have investigated optimal navigation through real ocean flows (Weizhong Zhang et al., 2008), simulated turbulence (Biferale et al., 2019), and simple flows for which

there exist exact optimal navigation solutions (Kularatne, Bhattacharya, and Hsieh, 2018). Here, we investigate the flow past a cylinder to retain greater interpretability of learned navigation strategies while remaining a challenging, unsteady navigation problem.

The swimmer is tasked with navigating from a starting point on one side of the cylinder wake to within a small radius of a target point on the opposite side of the wake region. For each episode, or attempt to swim to the target, a pair of start and target positions are chosen randomly within disk regions as shown in Figure 2.1.

Additionally, the swimmer is assigned a random starting time in the vortex shedding cycle. The spatial and temporal randomness prevent the RL algorithm from speciously forming a one-to-one correspondence between the swimmer's relative position and the background flow, which would not reflect real-world navigation scenarios (see Appendix A.1). All swimmers have access to their position relative to the target $(\Delta x, \Delta y)$ rather than their absolute position to further prevent the swimmer from relying on memorized locations of flow features during training. For this reason, the start and target regions were chosen to be large relative to the width of the cylinder wake.

For simplicity and training speed, we consider the swimmer to be a massless point with a position $\mathbf{X}_n = [x, y]$ which advects with the time-dependent background flow $\mathbf{U}_{\text{flow}} = [u(x, y, t), v(x, y, t)]$. The swimmer can swim with a constant speed U_{swim} and can directly control its swimming direction θ. These dynamics are discretized with a time step $\Delta t = 0.3D/U_\infty$ using a forward Euler scheme, where D is the cylinder diameter and U_∞ is the freestream flow velocity:

$$\mathbf{X}_0 = \mathbf{X}_{\text{start}}, \tag{2.1}$$

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \Delta t \left(U_{\text{swim}} \left[\cos(\theta), \sin(\theta) \right] + \mathbf{U}_{\text{flow}} \right). \tag{2.2}$$

It is also possible to apply RL-based navigation with more complex dynamics, including when the swimmer's actions alter the background flow (Verma, Novati, and Koumoutsakos, 2018).

We chose a swimming speed of 80% of the freestream speed U_∞ to make the navigation problem challenging, as the swimmer cannot overcome the local flow in some regions of the domain. A slower speed ($U_{\text{swim}} < 0.6U_\infty$) makes navigating this flow largely intractable, while a swimming speed greater than the freestream

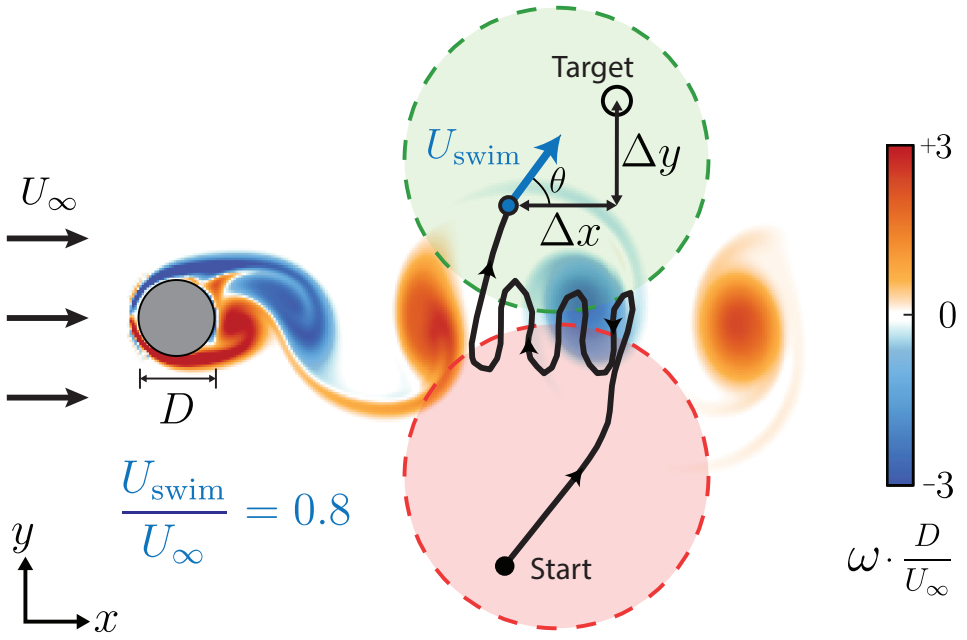


Figure 2.1: Test navigation problem of navigating through unsteady cylinder flow. Swimmers are initialized randomly inside the red disk and are assigned a random target location inside the green disk. These regions of start and target points are $4D$ in diameter, and are located $5D$ downstream and centered $2.05D$ above and below the cylinder. Additionally, each swimmer is initialized at a random time step in the vortex shedding cycle. An episode is successful when a swimmer reaches within a radius of $D/6$ around the target location.

($U_{\text{swim}} > U_\infty$) would allow the swimmer to overcome the background flow and easily reach the target.

2.2.2 Navigation using deep reinforcement learning

In RL, an agent acts according to a policy, which takes in the agent's state s as an input and outputs an action a. Through repeated experiences with the surrounding environment, the policy is trained so that the agent's behavior maximizes a cumulative reward. Here, the agent is a swimmer, the action is the swimming direction θ, and we seek to determine how the performance of a learned navigation policy is impacted by the type of flow information contained in the state.

To this end, we first consider a flow-blind swimmer as a baseline, which cannot sense the surrounding flow and only has access to its position relative to the target ($s = \{\Delta x, \Delta y\}$). Next, inspired by the vorticity-based navigation strategy of the zebrafish (Oteiza et al., 2017), we consider a vorticity swimmer with access to the local vorticity at the current and previous time step in order to sense changes in the

local vorticity ($s = \{\Delta x, \Delta y, \omega_n, \omega_{n-1}\}$). We also consider a velocity swimmer, which has access to both components of the local background velocity ($s = \{\Delta x, \Delta y, u, v\}$). Results for additional swimmers with different states are shown in Appendix A.3. In a real robot, velocity sensing could be implemented via a variety of methods including pitot tubes and hot wire or hot film anemometry. Local vorticity could be computed from several velocity sensors. Not considered here are distributed sensing schemes, such as distributed pressure or shear sensors, which can be effective for flow sensing and identification (Weber et al., 2020). Coupling optimal flow sensor distribution (e.g., Verma, Papadimitriou, et al., 2020) with the present RL navigation method may be a fruitful, but computationally challenging, extension of this point-swimmer proof of concept.

We employ deep RL for this navigation problem, in which the navigation policy is expressed using a deep neural network. Previously, Biferale et al. (Biferale et al., 2019) employed an actor-critic approach for RL-based navigation of a repeated, deterministic snapshot of turbulent flow, which is similar to navigating a steady flow field (see Appendix A.1). The policy was expressed using a basis function architecture, requiring a coarse discretization of both the swimmer's position and swimming direction for computational feasibility. In contrast, V-RACER (Novati and Koumoutsakos, 2019) is well suited for this navigation problem, as it is designed for continuous problems and can accept additional sensory inputs with negligible impact in computational complexity. A single 128×128 deep neural network is used for the navigation policy, which accepts the swimmers state (i.e., flow information and relative position) and outputs the swimming direction as continuous variables. The network also outputs a Gaussian variance in the swimming direction to allow for exploration during training. The policy network is randomly initialized and then iteratively updated through repeated attempts to reach the target following the policy gradient theorem (Sutton et al., 2000). V-RACER employs Remember and Forget Experience Replay to reuse past experiences over multiple iterations to update the swimmer's policy in a stable and data-efficient manner. Additional details of the V-RACER algorithm are shown in Appendix A.2. Results such as the success rate and cumulative reward curves were averaged after training each swimmer five times. This step helped ensure that differences in performance did not arise spuriously from the random initialization of the policy network, as described in Henderson et al. (2019).

At each time step, the swimmer receives a reward according to the reward function

r_n, which is designed to produce the desired behavior of navigating to the target. We employ a similar reward function as Biferale et al. (2019):

$$r_n = -\Delta t + 10\left[\frac{||\mathbf{X}_{n-1} - \mathbf{X}_{\text{target}}||}{U_{\text{swim}}} - \frac{||\mathbf{X}_n - \mathbf{X}_{\text{target}}||}{U_{\text{swim}}}\right] + \text{bonus}. \tag{2.3}$$

The first term penalizes duration of an episode to encourage fast navigation to the target. The second two terms give a reward when the swimmer is closer to the target than it was in the previous time step. The final term is a bonus equal to 200 time units, or approximately 30 times the duration of a typical trajectory. The bonus is awarded if the swimmer successfully reaches the target. Swimmers that exit the simulation area or collide with the cylinder are treated as unsuccessful. The second two terms are scaled by 10 to be on the same order of magnitude as the first term, which we found significantly improved training speed and navigation success rates. We also investigated a non-linear reward function, in which the second two terms are the reciprocal of the distance to the target, however, it exhibited lower performance. The RL algorithm seeks to maximize the total reward, which is the sum of the reward function across all N time steps in an episode:

$$r_{\text{total}} = \sum_{n=1}^{N} r_n = -T_{\text{f}} + 10\frac{||\mathbf{X}_{\text{start}} - \mathbf{X}_{\text{target}}||}{U_{\text{swim}}} + \text{bonus}. \tag{2.4}$$

The evolution of r_{total} during training for each swimmer is shown in Figure 2.2. All RL swimmers were trained for 20,000 episodes.

The reward function can be tuned to optimize for specific objectives such as minimum fuel consumption by including additional terms (e.g., (Buzzicotti et al., 2021)). Here, the reward function acts to optimize for two objectives: minimal arrival time to the target ($-T_{\text{f}}$) and maximum success rate of reaching the target (second two terms). The ability of RL to achieve these two objectives is explored in the following sections.

2.2.3 Success of RL navigation

After training, Deep RL discovered effective policies for navigating through this unsteady flow. An example of a path discovered by the velocity RL swimmer is shown in Figure 2.3. Because the swimming speed is less than the free-stream velocity, the swimmer must utilize the wake region where it can exploit slower background flow to swim upstream. Once sufficiently far upstream, the swimmer

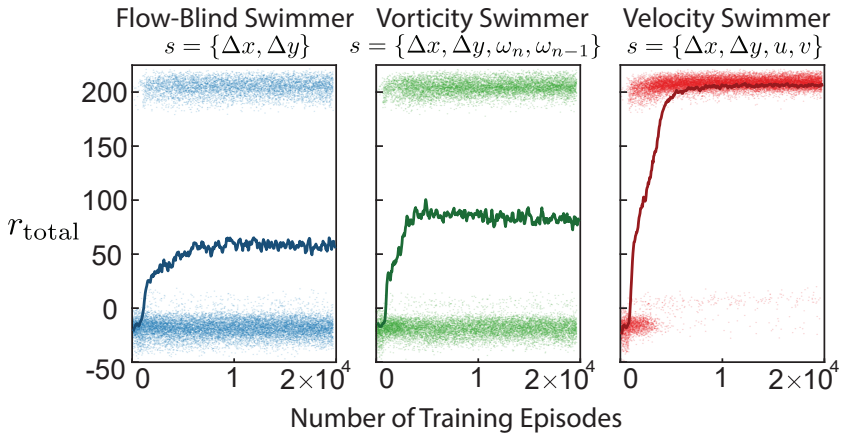


Figure 2.2: Evolution of the cumulative reward during training for the three RL swimmers. The cumulative rewards for each episode are plotted as points, and a moving average with a window of 201 episodes is plotted with a solid line. Because the swimmer gains a bonus of 200 for reaching the target, successful episodes are clustered around a reward of 200 while unsuccessful episodes are clustered below zero.

can then steer towards the target. The plot of the swimming direction inside the wake (Figure 2.3b) shows how the swimmer changes its swimming direction in response to the background flow, enabling it to maintain its position inside the wake region and target low-velocity regions.

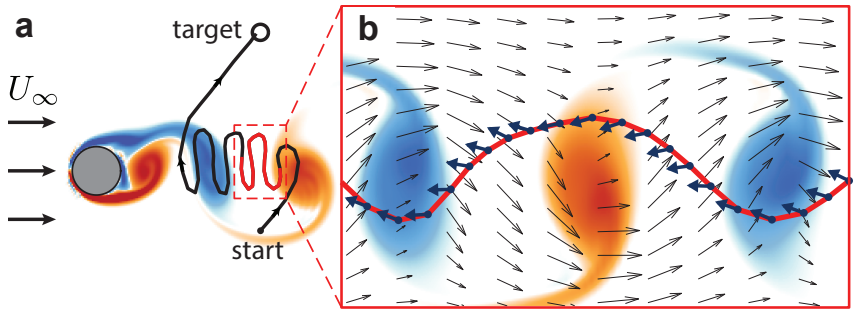


Figure 2.3: Example trajectory of the velocity RL swimmer. **a** Trajectory plotted in a cylinder-fixed frame, showing the swimmer successfully navigate from its starting location to the target.**b** Segment of this trajectory plotted in a wake-stationary frame of reference on top of the background flow field, which highlights the swimmer exploiting low-velocity regions in the cylinder wake to swim upstream. The swimming direction is plotted at each time step along the trajectory, revealing that this RL swimmer adjusts it swimming direction in response to the changing background flow, enabling time-efficient navigation.

However, the ability of Deep RL to discover these effective navigation strategies de-

pends on the type of local flow information included in the swimmer state. To illustrate this point, example trajectories and the average success rates of the flow-blind, vorticity, and velocity RL swimmers are plotted in Figure 2.4, and are compared with a naïve policy of simply swimming towards the target ($\theta_{\text{naïve}} = \tan^{-1}(\Delta y/\Delta x)$).

A naïve policy of swimming towards the target is highly ineffective. Swimmers employing this policy are swept away by the background flow, and reached the target only 1.2% of the time on average. A RL approach, even without access to flow information, is much more successful: the flow-blind swimmer reached the target locations nearly 40% of the time.

Giving the RL swimmers access to local flow information increases the success further: the vorticity RL swimmer averaged a 47.2% success rate. Surprisingly, however, the velocity swimmer has a near 100% success rate, greatly outperforming the zebrafish-inspired vorticity approach. With the right local flow information, it appears that an RL approach can navigate nearly without fail through a complex, unsteady flow field. However, the question remains as to why some flow properties are more informative than others.

To better understand the difference between RL swimmers with access to different flow properties, the swimming direction computed by each RL policy is plotted over a grid of locations in Figure 2.5. The flow-blind swimmer does not react to changes in the background flow field, although it does appear to learn the effect of the mean background flow, possibly through correlation between the mean flow and the relative position of the swimmer in the domain. This provides it an advantage over the naïve swimmer. The vorticity swimmer adjusts its swimming direction modestly in response to changes in the background flow, for example, by swimming slightly upwards in counter-clockwise vortices and slightly downwards in clockwise vortices. The velocity swimmer appears most sensitive to the background flow, which may help it respond more effectively to changes in the background flow.

Station-keeping inside the wake region may be important for navigating through this flow. In the upper right of the domain, the velocity swimmer learns to orient downwards and back to the wake region, while the other swimmers swim futilely towards the target. Because the vorticity depends on gradients in the background flow, that property cannot be used to respond to flow fields that are spatially uniform. These differences appear to explain many of the failed trajectories in Figure 2.4, in which the flow-blind and vorticity swimmers are swept up and to the right by the background flow. Other swimmers with partial access to the background flow fared

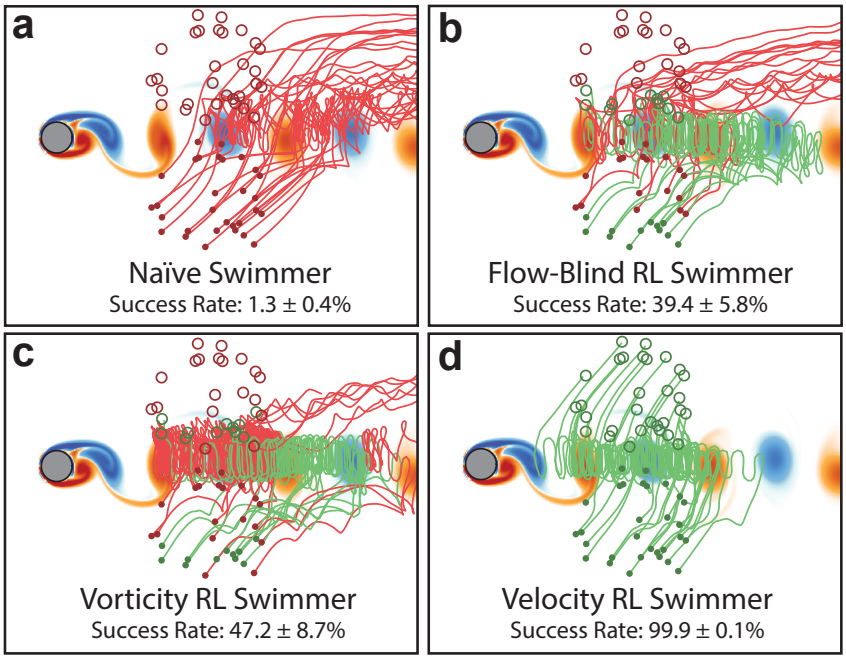


Figure 2.4: Average success rate with 30 example trajectories for each swimmer type. Successful attempts to reach the target are green, while unsuccessful attempts are red. (**a**) Naïve policy of swimming towards the target is rarely successful. (**b**) The flow-blind RL swimmer navigates more effectively than the naïve swimmer. (**c**) The vorticity RL swimmer is more successful than the flow-blind swimmer, showing that sensing the local flow can improve RL-based navigation. (**d**) Surprisingly, the velocity RL swimmer nearly always reaches the target using only the local flow velocity. The stated success rates are averaged over 12,500 episodes and are shown with one standard deviation arising from the five times each swimmer was trained.

similarly to the vorticity swimmer, further suggesting that sensing both velocity components are required for best performance (see Appendix A.3).

While sensing of point vorticity is insufficient to detect spatially uniform flow fields, it can be useful for distinguishing the vortical wake from the freestream flow. This can explain why the vorticity swimmer performs better than the flow-blind swimmer. A similar reasoning could apply to swimmers that sense other flow quantities such as pressure or shear. Indeed, Alsalman et al. found that velocity sensors outperformed vorticity sensors for neural network-based flow classification (Alsalman, Colvert, and Kanso, 2018).

In addition to providing environmental cues, however, the background flow velocity may be particularly important for navigation, as it affects the future state of the swimmer. Because the flow advects the swimmers according to linear dynamics

(Equation 2.2), the local velocity can exactly determine the swimmer's position at the next time step. This may explain the high navigation success of the velocity swimmer, as it has the potential to accurately predict its next location. To be sure, the Deep RL algorithm must still learn where the most advantageous next location ought to be, as the flow velocity at the next time step is still unknown.

For real swimmers, vorticity may also affect the future state of the swimmer, for example, by causing a swimmer to rotate in the flow (Colabrese et al., 2017) or by altering boundary layers and skin friction drag (Verma, Novati, and Koumoutsakos, 2018). Real robots would also be subject to additional sources of complexity not considered in this simplified simulation, which would make it more difficult to determine a swimmer's next position from local velocity measurements alone.

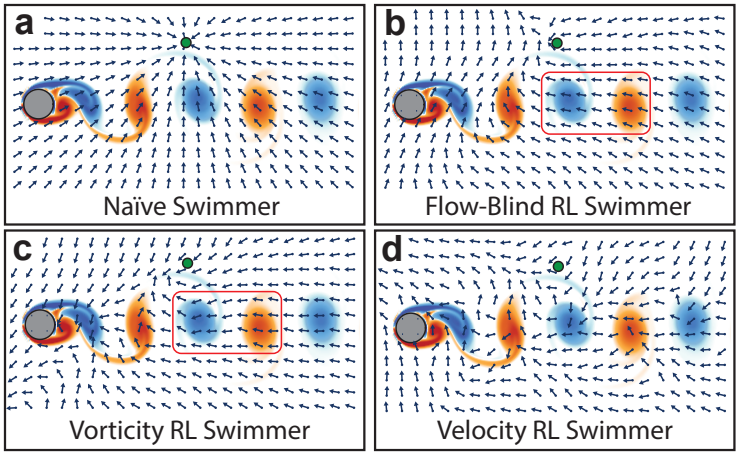


Figure 2.5: Swimming direction policy plotted across the domain for a fixed target (green circle) at a given time instant. (**a**) The naïve swimmer swims towards the target. (**b**) The red outline highlights how the flow-blind swimmer navigates irrespective of the background flow, while the vorticity swimmer (**c**) adjusts its swimming direction modestly. (**d**) The velocity swimmer appears even more sensitive to the unsteady background flow.

2.2.4 Comparison with optimal control

In addition to reaching the destination successfully, it is desirable to navigate to the target while minimizing energy consumption or time spent traveling. Biferale et al. (2019) demonstrated that RL can approach the performance of time-optimal trajectories in steady flow for fixed start and target positions. Here, we find that this result also holds for the more challenging problem of navigating unsteady flow with variable start and target points.

Assuming the swimmer reaches the target location, the only term in the cumulative reward r_{total} that depends on the swimmer's trajectory is $-T_{\text{f}}$ (Equation 2.4). Therefore, maximizing the cumulative reward of a successful episode is equivalent to finding the minimum time path to the target. Because the velocity RL swimmer always reaches the target successfully, we compare the velocity RL swimmer to the time-optimal swimmer derived from optimal control.

To find time-optimal paths through the flow, given knowledge of the full velocity field at all times, we constructed a path planner that finds locally optimal paths in two steps. First, a rapidly-exploring random tree algorithm (RRT) finds a set of control inputs that drive the swimmer from the starting location to the target location, typically non-optimally (LaValle and Kuffner, 2001). Then we apply constrained gradient-descent optimization (i.e., the fmincon function in MATLAB) to minimize the time step (and therefore overall time T_{f}) of the trajectory while enforcing that the swimmer starts at the starting point (Equation 2.1), obeys the dynamics at every time step in the trajectory (Equation 2.2), and reaches the target ($||\mathbf{X}_N - \mathbf{X}_{\text{target}}|| \leq D/6$). The trajectories produced by this method are local minima, so the fastest trajectory was chosen out of 100 runs and validated to be globally optimal by comparing it with the output of the level set method described in Lolla et al. (Lolla et al., 2014) computed using a MATLAB level set toolbox (Mitchell, 2008). Other algorithms could also be used to find optimal trajectories for unsteady flow given knowledge of the entire flow field (Kularatne, Bhattacharya, and Hsieh, 2018).

A comparison between RL and time-optimal navigation for three sets of start and target points is shown in Figure 2.6. These points were chosen to represent a range of short and long duration trajectories. Despite only having access to local information, the RL trajectories are nearly as fast and qualitatively similar to the optimal trajectories, which were generated with the advantage of having full global knowledge of the flow field.

The surprisingly high performance of the RL approach compared to a global path planner suggests that deep neural networks can, to some extent, approximate how local flow at a particular time impacts navigation in the future. In other words, a successful RL swimmer must simultaneously navigate and identify the approximate current state of the environment using only a single flow measurement at one instant in time at an unknown absolute location in the flow field. In comparison, the optimal control approach relies on knowledge of the environment in advance. There are limitations to the RL approach, however. For example, the optimal swimmer

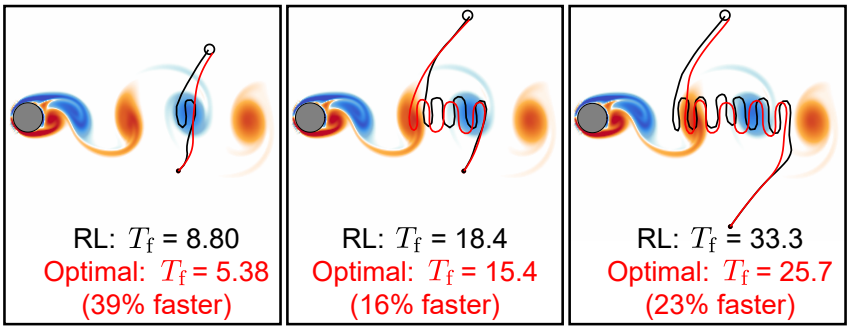


Figure 2.6: Comparison between time-optimal and RL trajectories. Time-optimal trajectories are shown in red and RL trajectories are shown in black. The RL swimmer used the state $s = \{\Delta x, \Delta y, u, v\}$. Time to reach the target T_f is made non-dimensional using the timescale D/U_∞.

on the right of Figure 2.6 enters the wake region at a different location than the RL swimmer to avoid a high velocity region, which the RL swimmer may not have been able to sense initially.

In addition to approaching the optimality of a global planner, RL navigation offers a robustness advantage. As noted by Biferale et al. (2019), RL can be robust to small changes in initial conditions. Here, we show that RL navigation can generalize to a large area of initial and target conditions as well as random starting times in the unsteady flow. Additionally, we found that the velocity RL swimmer is robust to realistic amounts of sensor noise from turbulent fluctuations (see Appendix A.4).

In contrast, the optimal trajectories here are open loop: any disturbance or flow measurement inaccuracy would prevent the swimmer from successfully navigating the target. While robustness can be included with optimal control in other ways (Panda et al., 2020), responding to changes in the surrounding environment is the driving principle of this RL navigation policy. Indeed, the related algorithm of imitation learning has been used for drone control by employing a neural network to mimic an optimal flight path while reacting to local disturbances (Rivière et al., 2020).

2.2.5 Policy Transfer to Double Gyre Flow

The RL swimmer showed robustness to large changes in the start and target positions, and to realistic amounts of sensor noise (Appendix A.4). However, it is worth considering if a learned navigation policy can transfer between different flow fields, which would reduce the amount of training required for navigating a new flow field

and increase the robustness of a swimmer to sudden changes in its environment.

Colabrese et al. (2017) demonstrated that an RL swimmer trained on a vortical flow field can navigate successfully in a new, but topologically similar, flow field without additional training. However, they noted that learned navigation strategies may not transfer between dissimilar flows, thus requiring additional training to form a new navigation strategy. Here, we consider if the learned policy for navigating the cylinder flow can transfer to a double gyre flow, which is topologically dissimilar.

The double gyre flow is a 2D, unsteady, periodic flow field that is a simplified representation of circulation patterns found frequently in the ocean (S. C. Shadden, Lekien, and J. E. Marsden, 2005; Solomon and Gollub, 1988; Krishna, Song, and Brunton, 2022). The velocity field is defined analytically in S. C. Shadden, Lekien, and J. E. Marsden (2005), where all length units are non-dimensional (i.e., $L = 1$). Here, we used $A = 2/3U_{\text{swim}}$, $\epsilon = 0.3$, and $\omega = 20\pi U_{\text{swim}}/3L$, which presents a challenging navigation problem that is unsteady on a similar time scale as the cylinder flow. Swimmers were started at a random time step in the right gyre and are tasked with navigating to a randomly chosen target in the left gyre. The problem setup is shown in Figure 2.7a.

To see if the learned RL policy transfers to the double gyre flow, two versions of the velocity RL swimmer were tested: one trained on the unsteady cylinder flow and one trained for the double gyre flow. Additionally, the naïve swimmer was included for comparison. The success rates of these swimmers are shown in Figure 2.7b-d.

The learned policy for navigating the cylinder wake did not transfer effectively to the double gyre flow, resulting in only a 4.1% average success rate (Figure 2.7C) compared to the naïve swimmer's 40.9% average success rate (Figure 2.7B). Poor performance was also observed when the problem coordinates were rotated and scaled to match the start and target regions of the cylinder flow navigation problem.

With training, however, new and effective navigation strategies can be learned. The velocity RL swimmer trained on the double gyre flow achieved a high average success rate of 87.4%, leveraging the background flow to escape the right gyre and navigate to its target locations in the left gyre. These results suggest that learned policies may indeed only transfer between similar flows, and that effective navigation in new flow fields requires additional training. Additionally, while all investigations here are in simulated flow environments, future studies may benefit from investigating the transfer of learned behaviors between simulated and real environments, which

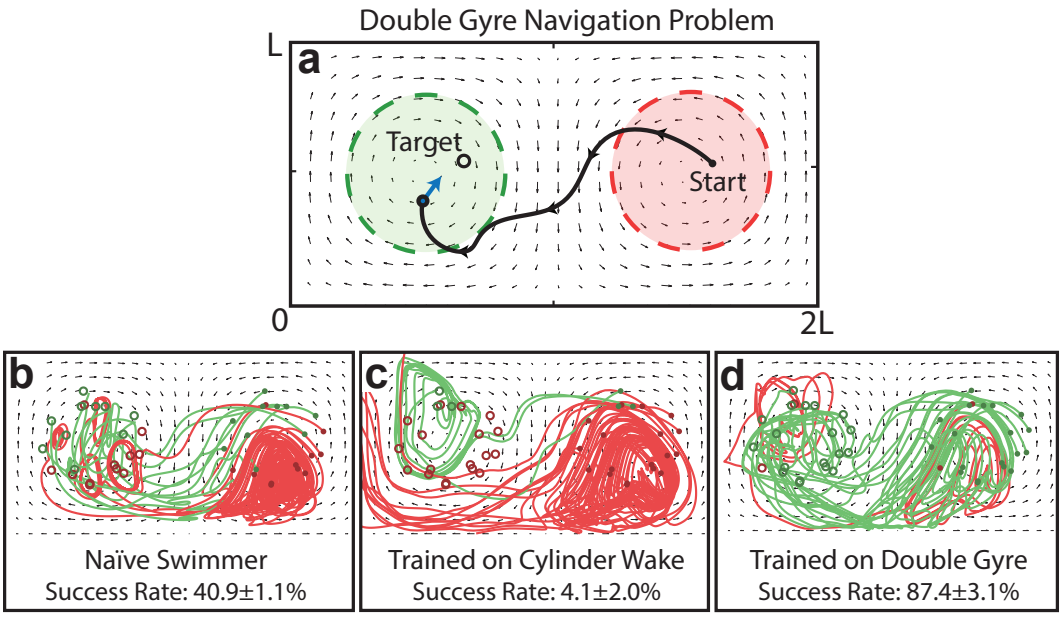


Figure 2.7: RL navigation in the double gyre flow field. (**a**) Navigation problem setup. The start and target regions are $L/2$ in diameter and located at $(3L/2, L/2)$ and $(L/2, L/2)$, respectively. (**b**) A naïve policy achieves 40.9% success rate on average. (**c**) The velocity RL swimmer trained on the cylinder wake navigates the double gyre flow poorly, indicating its navigation policy did not generalize. (**d**) After receiving training for the double gyre flow, the velocity RL swimmer is able to adapt and navigate more effectively than either swimmer. As with the cylinder flow, successful attempts to reach the target are green, while unsuccessful attempts are red. An episode is successful when a swimmer reaches within a radius of $L/50$ around the target location. The stated success rates are averaged over 12,500 episodes and are shown with one standard deviation arising from the five times each swimmer was trained.

can reduce in situ training time for physical robots.

2.3 Discussion

We have shown in this study how deep RL can discover robust and time-efficient navigation policies which are improved by sensing local flow information. A bio-inspired approach of sensing the local vorticity provided a modest increase in navigation success over a position-only approach, but surprisingly the key to success was discovered to lie in sensing the velocity field, which more directly determined the future position of the swimmer. This suggests that RL coupled with an on-board velocity sensor may be an effective tool for robot navigation. While the learned policy for navigating an unsteady cylinder wake did not transfer to a dissimilar double gyre flow, additional training enabled the RL swimmer adapt to the new flow field.

Future investigation is warranted to examine the extent to which the success of the velocity approach extends to real-world scenarios, in which robots may face more complex, 3D fluid flows, and be subject to non-linear dynamics and sensor errors.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1745301. P.G. was supported by this fellowship.

Data Availability

All data generated and discussed in this study are available within the article and its supplementary files, or are available from the authors upon request.

Code Availability

The deep RL algorithm V-RACER is available at github.com/cselab/smarties.

Chapter 3

FISH-INSPIRED TRACKING OF UNDERWATER TURBULENT PLUMES

The contents of this chapter have been adapted from Gunnarson and Dabiri (P. Gunnarson and J.O. Dabiri, 2024), submitted to Bioinspiration and Biomimetics under the title "Fish-inspired tracking of underwater turbulent plumes." Peter Gunnarson and John Dabiri conceived of the project. Peter Gunnarson conducted experiments. Both authors analyzed results and wrote the paper.

Abstract

Autonomous ocean-exploring vehicles have begun to take advantage of onboard sensor measurements of water properties such as salinity and temperature to locate oceanic features in real time. Such targeted sampling strategies enable more rapid study of ocean environments by actively steering towards areas of high scientific value. Inspired by the ability of aquatic animals to navigate via flow sensing, this work investigates hydrodynamic cues for accomplishing targeted sampling using a palm-sized robotic swimmer. As proof-of-concept analogy for tracking hydrothermal vent plumes in the ocean, the robot is tasked with locating the center of turbulent jet flows in a 13,000-liter water tank using data from onboard pressure sensors. To learn a navigation strategy, we first implemented RL on a simulated version of the robot navigating in proximity to turbulent jets. After training, the RL algorithm discovered an effective strategy for locating the jets by following transverse velocity gradients sensed by pressure sensors located on opposite sides of the robot. When implemented on the physical robot, this gradient following strategy enabled the robot to successfully locate the turbulent plumes at more than twice the rate of random searching. Additionally, we found that navigation performance improved as the distance between the pressure sensors increased, which can inform the design of distributed flow sensors in ocean robots. Our results demonstrate the effectiveness and limits of flow-based navigation for autonomously locating hydrodynamic features of interest.

3.1 Introduction

The ocean is critically under-explored and under-sampled. In response to the need for increased ocean sampling, AUVs such as undersea gliders (Rudnick, 2016) and autonomous floats (Wong et al., 2020) have become vital tools for in-situ sampling of ocean environments. Autonomy lowers the cost of deploying an underwater vehicle and enables exploration of larger swaths of the ocean volume by removing the need for constant communication with the surface. However, the limited range and speed of current AUVs are barriers to accomplishing widespread coverage of the ocean (Zereik et al., 2018).

A promising technique for increasing the effectiveness of AUVs is targeted sampling, in which robots actively seek out areas of scientific interest such as undersea thermal vents, coastal upwelling fronts, or phytoplankton patches (Zhang, Ryan, et al., 2019). Vehicles may use a variety of sensing methodologies for tracking areas of high importance. For example, an AUV used cameras and a machine vision algorithm to autonomously track animals in the midwater for hours at a time (Katija et al., 2021). Another vehicle used chemical and turbidity measurements to autonomously locate an undersea thermal vent (Preston et al., 2022). Salinity-sensing was used for autonomously locating and mapping the boundary of a salinity-intrusion front from the Gulf Stream (Zhang, Yoder, et al., 2022). By using onboard sensors to actively steer towards areas of high scientific value, AUVs can sample information-rich locations more quickly and better allocate their limited energy supplies.

A biology-inspired approach to seeking out targets underwater is to take advantage of hydrodynamic cues present in ocean environments. For example, many aquatic animals including sea lions and catfish can hunt by sensing the wakes left behind by their prey (Pohlmann, Grasso, and Breithaupt, 2001; Dehnhardt, Mauck, Hanke, et al., 2001). This ability to track animals without any visual information is advantageous for predators. If implemented in underwater vehicles, flow sensing could be a means to track oceanic features of interest when poor water and lighting conditions would otherwise obscure visual tracking. Aquatic animals use flow sensing for a variety of other tasks, including following walls (Patton, Windsor, and Coombs, 2010) and station keeping in the wake behind obstacles (Liao, 2007). Each of these biological tasks could inspire similar functionalities in autonomous vehicles, e.g., autonomous navigation along the seafloor and detecting wake signatures from physical obstacles or hydrothermal vents (Germanovich et al., 2015).

Given the potential applications of flow sensing in underwater robots, significant

effort has been dedicated to the development of bio-inspired flow sensors (Ko, Lauder, and Nagpal, 2023). For example, engineers have developed various hair-like sensors that mimic the flow-sensing mechanism of superficial neuromasts in fish lateral lines (Bora et al., 2018). By measuring the deflection of micro-pillars, these sensors can detect fluid shear and infer flow velocity. The undulatory geometry of sea lion whiskers has also inspired the design of flow velocity sensors (Zheng, Kamat, et al., 2021). In addition to velocity sensing, the pressure sensing function of canal neuromasts in fish lateral lines has been mimicked using distributed pressure sensors (Zhai, Zheng, and Xie, 2021).

In general, these flow sensors have achieved high accuracy and sensitivity, but developing strategies to interpret these flow measurements for autonomous exploration remains an active area of research. Idealized potential flow models, often supplemented by empirical measurements or regression models, have been used for tasks such as characterizing the free stream flow (L. DeVries et al., 2015), locating dipole sources (Dagamseh et al., 2013), swimming along walls (Yen and Guo, 2016), and vehicle state estimation (Zheng, Wang, Xiong, et al., 2020). In the context of navigation for reaching a destination or finding flow features, a variety of heuristic strategies have been studied, such as navigating using the local velocity gradient (Monthiller et al., 2022); turning in the direction of maximum flow information (Colvert, Liu, et al., 2020); balancing the signal of two sensors for rheotaxis (Salumäe et al., 2012); and synchronizing swimming motion with the frequency of a Kármán Wake (Jez̲ov et al., 2012). Many of these studies indicate that gradient detection with distributed flow sensors may be important for flow-based navigation (see also Colvert, Chen, and Kanso, 2017; Oteiza et al., 2017; Reddy et al., 2018; Weber et al., 2020), but unifying design principles remain unclear.

Recently, machine learning algorithms such as reinforcement learning (RL) have become powerful tools for developing more complex flow-based navigation strategies. RL has been used in simulated environments for locating the source of turbulent odor plumes (Singh et al., 2023), following hydrodynamic trails behind simulated fish (Hang et al., 2023), and for point-to-point navigation in vortical flows (Peter Gunnarson et al., 2021). In a few cases, RL has been successfully applied to physical robots for navigation in background flow fields, such as a glider that learned to ride atmospheric thermals (Reddy et al., 2018) and an AUV that learned autonomous underwater target tracking (Masmitja et al., 2023). However, the majority of RL studies are performed in silico because of the high data gathering requirements, poor

interpretability, and computational complexity of neural networks (Dulac-Arnold et al., 2021; Masmitja et al., 2023). Indeed, a well-documented gap exists between successfully applying RL in simulated environments and deploying it on real systems, perhaps due to noise and factors that are difficult to model in simulations (Dulac-Arnold et al., 2021). Trials using physical robots are needed to validate RL approaches before deployment in real-world scenarios (Hasselmann et al., 2021).

In this work, we introduce the Caltech autonomous reinforcement learning robot (CARL), a palm-sized hardware and software platform for testing flow-based navigation and RL in large, controlled underwater environments. The robot is made from low-cost, off-the-shelf components and 3D-printed parts, and can swim autonomously underwater. We mounted pressure sensors at four locations around the robot to serve as flow sensors. As a proof-of-concept analogy for tracking hydrothermal vent plumes in the ocean, the robot was tasked with locating the center of turbulent jet plumes in a large water tank. A schematic overview of CARL and the tank environment is shown in Fig. 3.1.

To discover an effective navigation strategy, we implemented RL Hasselt, Guez, and Silver, 2015 in a simulated version of the robot and tank environment (see Methods for details). Using simulated pressure measurements as inputs to a neural network, the RL algorithm successfully learned to locate the turbulent plumes. Calculating a feature importance metric Lundberg and Lee, 2017 revealed that this navigation strategy depended primarily on the lateral sensors, which provide information about the transverse velocity gradient. Using this interpretation, we transferred a simplified version of the navigation policy to the physical robot, taking into account physical sensor noise. After transferring this policy onto the physical robot, CARL located the turbulent plumes in the tank at more than double the rate of random searching, demonstrating the effectiveness of flow sensing for autonomous navigation in a physical setting. Additionally, we found that the success rate and gradient sensing ability depended greatly on the spacing of the physical sensors. An analysis of the signal-to-noise ratio suggests that the sensor spacing may limit navigation performance for physical swimmers. Our results demonstrate the effectiveness of using simulation and physical experiments in tandem to design simple but effective navigation strategies for deployment in real-world environments.

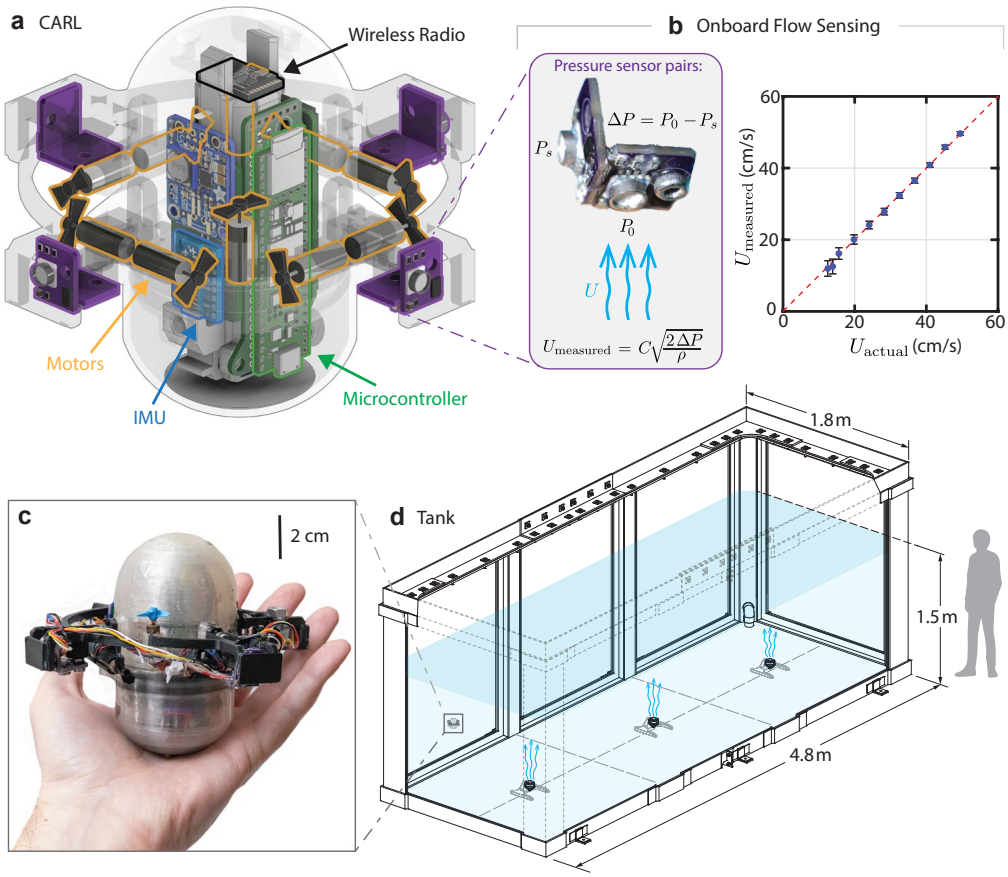


Figure 3.1: Schematic of CARL and tank facility. (**a**) Schematic of CARL, showing the arrangement of four pressure sensor pairs (see also panel (**b**)) an inertial measurement unit (IMU), ten motors for propulsion, and a Teensy 4.1 microcontroller for onboard processing. (**b**) Pressure sensors are arranged in pairs: a side-facing pressure sensor measures the static pressure (P_s), and a downward facing sensor measures the stagnation pressure (P_0) from upwards flow. The difference in pressure between these sensors ($\Delta P = P_0 - P_s$) can accurately measure flow velocity, as shown in the right plot (see Methods for details). (**c**) CARL with a 2 cm scale bar. (**d**) Tank facility with a human figure for scale. Three thrusters are arranged on the bottom of the tank, which generate three vertical turbulent plumes. CARL swims throughout the tank at a fixed depth of 30 cm below the water surface.

3.2 Results

3.2.1 Flow sensing with pressure sensors

To test flow-based navigation in a physical robot, we developed CARL, an autonomous underwater robotic platform. For onboard flow sensing, we mounted eight pressure sensors (MS5803-02BA, TE Connectivity) at four locations around CARL (Fig. 3.1a). We chose this piezo-resistive micro-electromechanical system (MEMS) pressure sensor for its small size (6 mm diameter), low cost ($16), and high precision (2.4 Pa resolution at 100 Hz). Additionally, the MS5803 sensors are manufactured with a waterproof gel coating and are already deployed in the ocean environments as depth sensors (Beddows and Mallon, 2018). Pressure sensors mimicking the canal neuromasts of fish were selected due to their great mechanical robustness and commercial availability compared to micro-pillar velocity sensors that mimic superficial neuromasts. The convenience and performance of pressure sensors makes them an attractive option for experimental studies: these sensors and other piezo sensor arrays have been used in several previous works for flow characterization and robot state estimation in quiescent flow (Zheng, Wang, Xiong, et al., 2020; Venturelli et al., 2012; Asadnia et al., 2013).

We arranged the pressure sensors in pairs to form downward-facing Pitot tubes, in which one sensor is exposed to impinging vertical flow and the other is shielded by a 3D-printed cover (see Fig. 3.1b). In this arrangement, the difference in pressure between the exposed and shielded sensors can be used to measure the upwards flow velocity component at these four locations. We verified the accuracy of these sensors for detecting steady flow in water channel test (see Methods).

Because the center of mass of CARL was located below the center of buoyancy, the exposed pressure sensors maintained a downward-facing orientation while swimming. The pressure sensors also functioned as depth sensors; while swimming, CARL attempted to maintain a constant depth using a proportional integral derivative (PID) control loop running at 50 Hz using this depth measurement and the vertically oriented motors (see Appendix B).

3.2.2 Navigation task and underwater testing environment

As a proof-of-concept analogy for tracking underwater thermal vents, which create large, turbulent jet plumes with flow velocities on the order of 1 m s^{-1} (Germanovich et al., 2015), we tasked CARL with locating the core of the turbulent jet plumes in a 1.8 m deep, 1.8 m wide, and 4.8 m long water tank. We mounted three thrusters (Blue

Robotics T200) on the bottom of the tank, as shown in Fig. 3.1d. The three thrusters were equally spaced 1.6 m apart along the centerline of the tank length, which created three distinct turbulent plumes. The thrusters have a diameter of approximately D = 10 cm, which was used as a reference length scale in the subsequent analysis. To reduce the complexity of this navigation problem, CARL swam at a fixed depth of 30 cm, which is approximately $12D$ above the thrusters on the bottom of the tank. At this depth, the turbulent plumes have a spread to a diameter of approximately $5D$, which is significantly larger than the size of CARL. By swimming at a fixed depth, the navigation problem becomes effectively two-dimensional, which simplifies the possible action space for CARL. Additionally, onboard flow measurements are minimally impacted by the motion of CARL because the flow due to horizontal robot motion is perpendicular to the vertical flow-sensing orientation of the pressure sensors. The downward facing orientation of the sensors enabled CARL to detect flow from the upward-facing thrusters at the bottom of the tank and measure the mean velocity profile of the plume (see Methods for details).

3.2.3 Learning a navigation policy in a simulated environment

To develop a navigation strategy for autonomously locating the turbulent jets, we first trained a navigation policy using RL in a simulated environment (see Methods). By training in a virtual environment, hyperparameters such as the reward function, network size and action space could be rapidly tested and fine-tuned. For example, our simulated environment trained using 600 episodes generated over the course of several minutes, which would take several hours to accomplish with CARL in the physical tank. We used the policy learned in simulation as a starting point to design an interpretable and robust navigation policy that can function on the physical version of CARL.

In the virtual environment, we modeled CARL as a massless point swimmer that swam at a constant speed in a 2D plane to emulate swimming at a constant depth in the physical tank. Because the mean flow of the jet was normal to the swimming direction of CARL, we made the simplifying assumption that the trajectory of CARL was unaffected by the surrounding flow field and vice-versa, which eliminated the need to solve for the background flow field at each time step. The tank dimensions, sensor spacing, swimming speed, and radius of the simulated swimmer were all matched with their physical counterparts.

To train a navigation policy, we implemented the Double Deep Q-network (DDQN)

RL algorithm (Hasselt, Guez, and Silver, 2015), which seeks to predict the Q-values, i.e., the value of an action in a particular state, and selects the actions with the highest predicted Q-values (see Methods for implementation details). For the state, we used one time step of simulated pressure measurements (ΔP_{front}, ΔP_{left}, ΔP_{back}, and ΔP_{right} as shown in Fig. 3.2a). In the physical CARL robot, pressure measurements were time-averaged over 0.3 seconds to reduce sensor noise, and the navigation policy also updated at this interval. This duration of averaging was chosen to maximally reduce sensor noise without introducing an excessive delay in navigation. For example, with a swimming speed of 20 $\text{cm}\,\text{s}^{-1}$, it typically took CARL 2.5 seconds to cross the width of the turbulent plume.

To simulate these pressure measurements, we included four virtual sensors that measured the square of the vertical velocity component of a simulated turbulent jet flow field with Gaussian sensor noise scaled to approximate the noise of the physical sensors. Details of the simulated flow field are shown in Methods. To avoid simple memorization of the turbulent flow field by the neural network, the simulation starting time and the starting location for CARL was randomized at the start of each episode.

Both the virtual swimmer and CARL could swim in five possible directions as shown in Fig. 3.2a. All actions included a component in the forward swimming direction to ensure exploration of the tank environment. After eventually running into the side walls of the tank, CARL turned around by a random angle and continued swimming. Each collision with the walls of the tank constituted the start and end of an episode.

Initially, the swimmer selected between the five possible actions at each time step with equal probability. After training on 600 episodes of exploration with random actions, the swimmer navigated by choosing the action with the highest Q-value as predicted by the neural network. For reproducibility, we trained the policy using ten different initial random seeds. Details of the reward function and training procedure are shown in Methods.

3.2.4 Navigation strategy learned in simulation

After training, we recorded the navigation success rate of the swimmer over 2,000 test episodes. As a baseline navigation policy for comparison, we used random exploration, in which the swimmer swam in straight lines and turned around by a random angle after colliding with the side walls of the tank.

Given the noise of the sensors, a flow may only be detectable if the swimmer is

within approximately $2.5D$ of a jet center, which comprises only 7% of the total area of the tank (see Methods for details). To reject episodes in which the swimmer never encountered a turbulent plume, we defined the navigation success rate as the probability of successfully finding a jet center given that a turbulent plume was encountered. For consistency with subsequent sections, we defined a plume encounter as occurring if the difference in flow measured by any two sensors was greater than twice the root-mean-square (RMS) of the sensor noise.

The results are plotted in Fig. 3.2e, and an example trajectory that shows the simulated swimmer steering towards the center of a turbulent plume is plotted in Fig. 3.2d. Whereas random exploration resulted in a successful navigation rate of 14%, the RL policy located the center of a jet in 34% of plume encounters. Using flow sensing, the learned navigation policy significantly outperformed random searching.

To investigate how the learned policy uses sensor measurements to locate the plumes, we computed SHAP (SHapley Additive exPlanations) values for each sensor. SHAP values are based on Shapley values from game theory and quantify the contribution of an input to a model to its output (Lundberg and Lee, 2017). In Fig. 3.2c, we plot the mean absolute SHAP value for each sensor, which represents the importance of each sensor averaged across all time steps of the test episodes. According to the SHAP values, the left and right sensors contributed significantly more to the learned policy than the front and back sensors.

To understand the relative importance of the left and right pressure sensor pairs, we plotted the swimming direction versus the difference of the left and right sensors ($\Delta P_{\text{right}} - \Delta P_{\text{left}}$) for every timestep in the 2,000 test episodes (Fig. 3.2b). From this plot, the learned policy appears to involve turning in the direction of the sensor experiencing higher flow. Given that the flow velocity is higher in the center of a turbulent plume, navigation towards faster flow leads the robot to the center of the turbulent plume. In situations where there is not a significant left-to-right velocity gradient (i.e., $\Delta P_{\text{right}} - \Delta P_{\text{left}}$ is close to zero), the learned policy does not appear to strongly correlate with measurements from any of the four sensors. Without a large gradient signal, the sensor inputs were dominated by turbulent fluctuations and simulated sensor noise, and therefore no action had a detectable advantage over any other. Because ΔP_{right} and ΔP_{left} measure vertical flow velocity at two spatially separated locations, the difference between these two sensors can be interpreted as representing a transverse velocity gradient, or a gradient in the direction perpendicular to forward swimming. Since all actions were biased towards

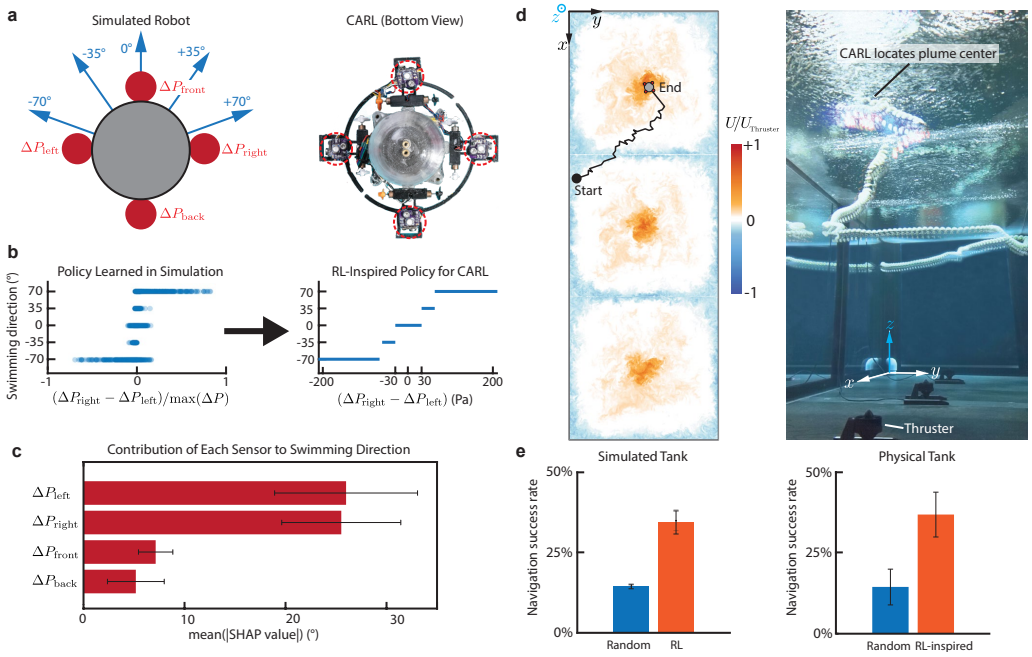


Figure 3.2: Navigation policy learned in simulation transfers to physical robot. (**a**) A simulated version of CARL (Left) was modeled as a point swimmer with distributed sensors that emulate those on CARL (Right). (**b**) Left: navigation policy learned in simulated is plotted at each timestep, showing a clear dependence on the transverse gradient ($\Delta P_{\text{right}} - \Delta P_{\text{left}}$). Right: a simplified version of the learned policy captured the same behavior but was scaled to account for the physical sensor noise ($\sigma_{\Delta P}$ = 15 Pa). (**c**) The left and right sensors have the largest impact on the learned swimming direction, as quantified by the SHAP values. Error bars are the standard deviation from training with 10 random seeds. (**d**) Example successful episodes in the simulated flow field (Left) and in the physical tank (Right). The background wall behind the tank is blurred for readability. (**e**) Left: after training, the simulated swimmer more than doubles the probability of locating a jet compared to random exploration. Error bars represent the standard deviation of the success rate after training with 10 random seeds. Right: using the RL-inspired policy, CARL achieved a similar improvement in jet-finding performance. Error bars are 95% confidence intervals using the Wilson score interval (N = 340 and 365).

forward swimming, a velocity gradient in the front-back direction may be less important: the swimmer will tend to explore in the forwards direction with random swimming.

In summary, the virtual swimmer learned an effective navigation strategy for locating the turbulent jet plumes, which depends primarily on the transverse velocity gradient. However, it is not guaranteed that this policy generalizes to a physical robot with noisy sensors and encountering a real-world turbulent flow. Therefore, we next tested the learned policy in the physical tank using CARL.

3.2.5 Plume localization with the physical CARL

To test the gradient-based navigation strategy on the physical CARL robot, we designed an "RL-inspired" navigation policy, which is a simplified version of the policy learned in simulation that takes into account the sensor noise of the physical pressure sensors. A plot of the RL-inspired policy is shown in Fig. 3.2b. In summary, if the difference between the left and right sensors is less than two standard deviations of the sensor noise, i.e., the signal-to-noise ratio (SNR) is less than two, CARL swims straight forward. Otherwise, CARL swims in the direction of larger transverse velocity gradient, as in the virtually-learned policy. The RL-inspired policy was programmed with simple if-then statements, which is computationally simple to evaluate onboard the microcontroller, particularly when compared to evaluating the output of neural networks (see Section 3.4). Hang et al. (2023) also used RL to handcraft a simple navigation policy for following hydrodynamic trails. Here, we additionally take into account limitations of a physical robot such as sensor noise and limited computation.

We conducted navigation tests in the physical tank to compare the RL-inspired policy with random navigation. Because the physical CARL lacks knowledge of its absolute position in the tank, we used depth as a proxy to determine successful location of the turbulent plume. Specifically, when CARL entered the center of the turbulent plume, the two diving motors on CARL were unable to overcome the mean flow, and CARL was pushed upwards by several centimeters. This effect only occurred in the center of the plume; in the edges of the turbulent plume the diving motors were strong enough to maintain a constant depth. The change in depth also served as a measurement of success that was independent of the flow sensors. The start and end of each episode were marked by CARL colliding with the walls of the tank, which was detected with the onboard IMU.

An example successful trajectory is plotted in Fig. 3.2d. Initially, while the sensed gradient was below the SNR threshold, CARL swam straight and explored the tank. When CARL detected a gradient, CARL turned in the direction of the transverse gradient in order to locate the plume above the labeled thruster.

The navigation results averaged over two hours of swimming in the tank for each policy (approximately 350 episodes), are plotted in Fig. 3.2e. Random exploration successfully located the jet center in 14% of plume encounters, while the RL-inspired policy achieved a significantly higher success rate of 37%. By sensing a transverse velocity gradient, CARL was able to locate turbulent jets autonomously.

3.2.6 Effect of sensor spacing on navigation performance

The learned navigation policy relied on detecting flow gradients using physically separated flow sensors in the presence of sensor noise and turbulent fluctuations. Therefore, the success of gradient-based navigation may be limited by the minimum detectable gradient over the background noise floor. For example, the distance between flow sensors (L) may be an important design consideration, since a robot with sensors spaced father apart may have a greater sensitivity to spatial gradients in the background flow but may be unable to detect flow structures smaller than L.

To vary the minimum detectable gradient on CARL, we created two additional sensor mounts with reduced distance between the pressure sensors (see Fig. 3.3a). In general, flow structures of size L or smaller may be undetected or spatially aliased when sampled by two sensors. However, for this experimental setup, the mean flow profile of the turbulent plume was significantly larger than CARL for all sensor configurations (see Fig. 3.3b). Reducing the sensor separation therefore reduced the difference in mean flow measured by the left and right sensors, negatively impacting sensitivity to flow gradients. Because the navigation policy depends on the inherent noise of the sensors which is independent of the sensor spacing or background turbulence, we tested navigation performance using the same policy for all three sensor mounts and compared the results with random exploration. For each case, we recorded the navigation performance over two hours of swimming, or approximately 350 episodes.

In Fig. 3.3c, we plot the plume detection rate, which we define as the chance of detecting any gradient signal above two times the noise floor of the sensors in a given episode. Since the swimming direction only changed if this threshold was exceeded, the random and RL-inspired swimming behaviors are expected to

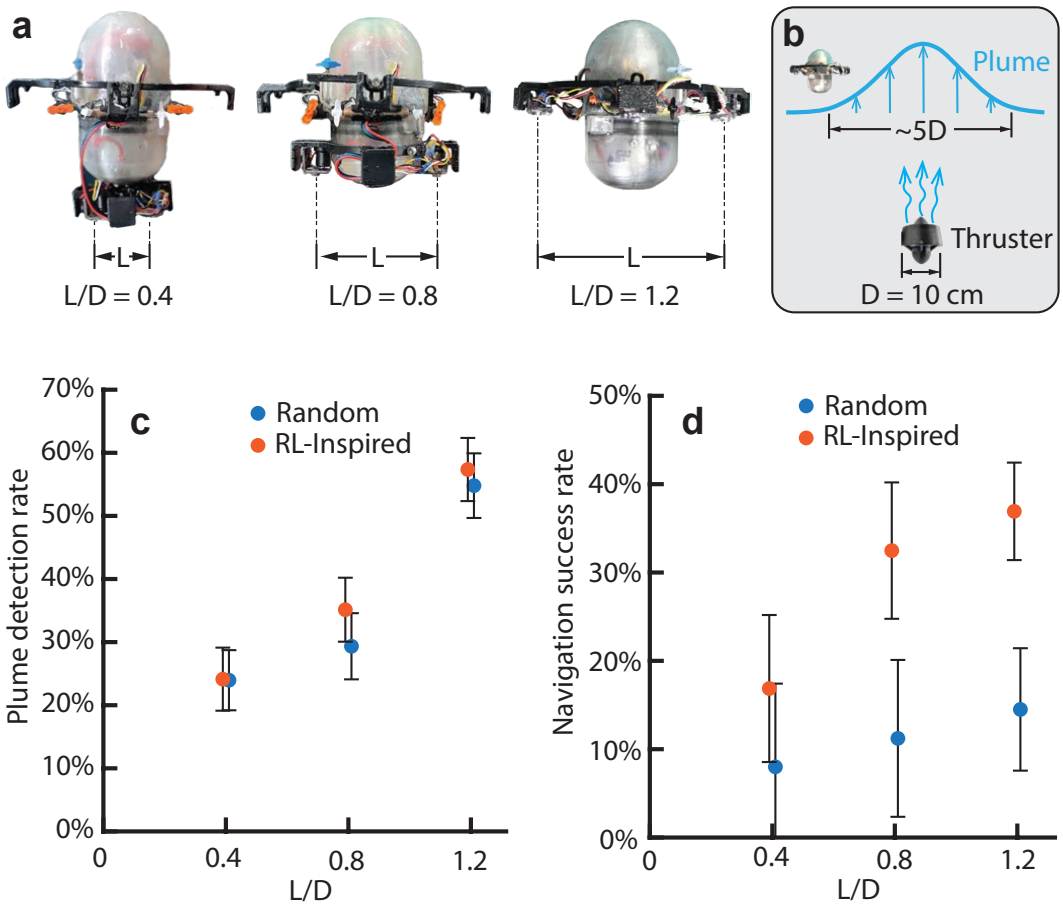


Figure 3.3: Navigation performance depends on sensor spacing. (**a**) We created three versions of CARL with sensors of varying spacing L. (**b**) At the swimming depth of CARL, the mean profile of the turbulent plume is approximately $5D$, according to flow measurements (see Methods). (**c**) The chance of CARL detecting a transverse velocity gradient above the noise floor increased with sensor separation. (**d**) The ability to locate a turbulent plume also increased with larger sensor separation. Error bars indicate 95% confidence intervals using the Wilson score interval (N for each data point ranges from 313 to 365).

be indistinguishable. The plume detection rate increased with a greater distance between the sensors.

Additionally, we plot the navigation success rate in Fig. 3.3d. The success rate significantly increased as the sensor spacing increases. In the case of the greatest sensor spacing ($L/D = 1.2$), the RL-inspired policy successfully located the jet cores 37% of the time, compared with only 17% for the smallest sensor spacing ($L/D = 0.4$). As the sensors are placed more closely together, the navigation ability became increasing similar to that of random exploration.

The dependence of navigation success rate on sensor spacing may be explained by the signal-to-noise ratio (SNR) of the measured flow gradient. Since CARL is significantly smaller than the mean flow profile, we use a linear mean background flow (i.e., $\mathrm{d}\bar{U}/\mathrm{d}x$ is constant) to model the SNR. Because the pressure sensor pairs measured the dynamic pressure of incoming flow (i.e., Fig. 3.1b), the gradient signal used for navigation is equal to:

$$\Delta P_{\text{right}} - \Delta P_{\text{left}} = \frac{1}{2}\rho\left(\bar{U}_{\text{left}}^2 - \bar{U}_{\text{right}}^2\right) = \rho\bar{U}_{\text{avg}}\frac{\mathrm{d}\bar{U}}{\mathrm{d}x}L\,,$$

where $\bar{U}_{\text{avg}}$ is the average of $\bar{U}_{\text{Left}}$ and $\bar{U}_{\text{Right}}$. According to this model, the signal scales with L: farther apart sensors experience a greater velocity differential, and therefore $\Delta P_{\text{right}} - \Delta P_{\text{left}}$ scales with L. The inherent noise of the sensors was fixed, depending only on the time-averaging window used. Noise from turbulent fluctuations may vary spatially and temporally, but for simplicity, we assumed the magnitude of turbulent fluctuations was similar for both sensors because CARL is small relative to the size of the mean flow profile. Under these assumptions, the signal is proportional to L while the noise sources remain constant with sensor spacing. This may explain the increase in navigation ability as the sensor spacing increases and may also contribute to the plume detection rate.

Another potential effect is that farther apart sensors are more likely to encounter a plume during straight-line swimming. During straight swimming, the area swept out between the two sensors is proportional to L. Therefore, the chance of any sensor encountering a plume will scale with L, provided that any relevant flow features are not smaller than the distance between the sensors. Given that the turbulent plumes only occupy 7% of the area of the tank, an increase in swept area may partially explain the increased plume detection rate as the sensor spacing increases. However, the navigation success rate should not depend on this effect, since it only counts episodes in which a jet is already detected.

3.3 Discussion

The effectiveness of the learned navigation policy in both simulations and physical experiments demonstrates the applicability of RL for solving flow-based navigation problems and identifying useful hydrodynamic cues. In this study, the robot learned to follow transverse flow gradients to localize the turbulent jet plumes. Applying interpretability metrics to the policy learned in simulation enabled us to simplify and adapt the learned policy for deployment in a physical robot, maintaining overall

effectiveness while taking into account computational constraints and the noise of the physical sensors. The success of CARL at locating turbulent plumes demonstrates the potential for targeted sampling of real-world flow features with onboard flow sensing.

The importance of the transverse velocity gradient suggests that onboard flow sensors may provide higher utility when arranged perpendicular to the direction of swimming. Such sensing arrangements are not uncommon in animals. For example, zebrafish were shown to require flow sensing on both sides of their body in order to detect flow gradients for avoiding walls (Oteiza et al., 2017). Swimming in the direction of greater flow is similar to the turning strategy employed by Braitenberg vehicles (Braitenberg, 1986; Salumäe et al., 2012) and a virtual robot that tracked the wake behind simulated fish (Hang et al., 2023). While a turbulent plume is qualitatively different than the vortex shedding wake produced by an animal, both types of flows involve spreading wake-like structures and intermittent eddies. Future work could investigate tracking vortex shedding dominated wakes in a physical tank.

The dependence of navigation performance on sensor spacing suggests that the SNR is limiting for navigating via flow gradients. Therefore, the navigation strategy employed by CARL may be most effective when mean flow gradients are significant, such as in close proximity to a hydrothermal plume or the turbulent wake behind an obstacle. In addition to using more accurate flow sensors, a larger separation between sensors can improve the SNR, provided that the gradients of interest are larger than the gap between sensors. Fish lateral lines often extend over the entire body, which may be advantageous for increasing sensitivity to flow gradients. Additionally, using an array of sensors to measure flow at many locations could provide additional information for navigation. In fish, distributed flow sensing can indicate flow direction and location of oscillating sources, e.g., other animals (Bleckmann and Zelick, 2009).

Onboard distributed pressure sensing offers a convenient, low-cost, and low-power method for measuring flow gradients. If deployed in an ocean environment, calibration procedures such as those outlined in (Strokina et al., 2016) may be needed to compensate for water temperature and atmospheric pressure variations to achieve same flow-sensing accuracy as in lab studies. Additionally, CARL navigated using the transverse gradient, which was perpendicular to the motion of the robot. Given the three-dimensional nature of underwater navigation in ocean environments, swimming may not be limited to directions normal to all onboard sensors,

which may induce flow signals during swimming. Bio-inspired robots which swim using undulatory motion also generate confounding flow signals from body motion. If sensors readings are coupled with the swimmer's motion, pre-calculated models such as those used in (Akanyeti et al., 2013) could be implemented to disentangle pressure signals from self-motion and external stimuli.

CARL located the turbulent plumes using only a single time step of pressure measurements. However, several studies have shown that neural network architectures with memory, such as Long Short-Term Memory (LSTM) networks, demonstrate performance improvements for tasks such as locating the source of odor plumes (Singh et al., 2023) and controlling the lift of a wing in turbulent conditions (Renn and Gharib, 2022). Memory may be particularly useful in turbulent flows, which are inherently intermittent and time-varying. Turbulent fluctuations could themselves be a useful signal for navigation. For example, turbulent fluctuations have been used as a signal for distinguishing between flows (Venturelli et al., 2012), and there is evidence that aquatic animals sense intermittency at the edge of turbulent odor plumes for locating the source of the plume (Michaelis et al., 2020). Other flow signals, such as the static pressure or vorticity, are coupled with the flow velocity and could be useful hydrodynamic cues for navigation (e.g., Hang et al., 2023). Additionally, memory itself could be used for gradient sensing, since directionality can be encoded in a time series (Kadakia et al., 2022). Finally, training using physically collected data or directly onboard an underwater robot may improve performance and allow for real-time adaptation to changing flow conditions.

3.4 Methods

3.4.2 Flow sensing with pressure sensors

We mounted pressure sensors at four locations on CARL, as shown in Fig. 3.1a-b. The pressure sensors record an absolute pressure measurement, resulting in a signal largely dominated by the hydrostatic pressure, which varies with depth. However, taking the difference between the exposed and covered pressure sensors effectively cancels out the hydrostatic pressure. Because CARL maintains an upright orientation, the sensors maintain a constant depth relative to each other, requiring only that an initial offset is subtracted at the beginning of each episode. The exposed and covered pressure sensors at each location on CARL are mounted close together, which reduces any change relative depth due to small wobbling motions during swimming from the turbulent jets.

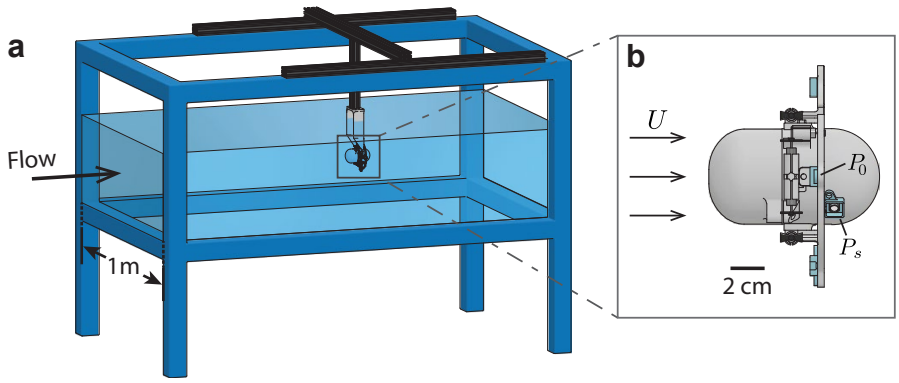


Figure 3.4: Validating flow sensing in a water channel. (**a**) Schematic setup showing CARL placed in the water channel to validate the flow sensors. Because flow is horizontal, CARL is oriented left-to-right. (**b**) Illustration of the incoming flow and location of the flow-facing and side-facing pressure sensors, which measure P_0 and P_s, respectively.

To validate the flow sensing capabilities of these sensors, we mounted CARL horizontally in a water channel and recorded the pressure from two sensors in steady flow conditions with speeds ranging from approximately 11 cm s^{-1} to 49 cm s^{-1} (Fig. 3.4a). For this water channel test, one sensor was mounted on the side of CARL and another sensor was mounted to point into the free stream flow as shown in Fig. 3.4b. In all other experiments, the sensors were arranged according to the Pitot-tube arrangement previously described in Fig. 3.1. The initial offset was recorded for each sensor in zero flow conditions and subtracted from subsequent measurements.

Because the exposed sensor is pointed into the freestream flow, it is expected to experience stagnation flow and a pressure increase equal to the dynamic pressure, $1/2\rho U^2$. According to an idealized potential flow model with no body wake such as the one used in (Zheng, Wang, Li, et al., 2020), a side-facing sensor should experience a pressure decrease of approximately twice the dynamic pressure as the flow accelerates over the body of the robot. Therefore, the difference in pressure between the sensors, or ΔP, is expected to equal $3/2\rho U^2$, or equivalently:

$$U = C\sqrt{\frac{2\Delta P}{\rho}},$$

where C is a Pitot tube calibration constant equal to $\sqrt{1/3}$. Using this equation, we computed the measured velocity and compared it with the flow velocity of the water channel in Fig. 3.1b. Using a fitted constant of $C = \sqrt{0.301}$, which differs from

the predicted constant by only 5%, the sensors were able to accurately measure the freestream flow velocity, demonstrating the efficacy of these pressure sensors for quantifying flow. In all free-swimming navigation experiments, the signals from the sensors were left as pressure measurements to streamline onboard signal processing and simplify the SNR analysis.

During free-swimming tests, the pressure sensors also functioned as depth sensors. The depth was estimated by taking an average of the pressure measured by the four side-facing sensors and applying the equation for the hydrostatic pressure of an incompressible fluid:

$$h = P_s/\rho g$$

While the flow in the tank was not static, we estimate that the hydrostatic pressure at the typical swimming depth of 30 cm was more than an order of magnitude greater than dynamic pressure created by the highest measured flow impinging on CARL.

3.4.3 Flow measurement of the turbulent plumes

For comparison with the simulated jets, we measured the physical turbulent plumes using particle image velocimetry (PIV). We seeded the tank with 100-micron silver-coated hollow glass particles (AGSL150-30TRD, Potters Industries) and illuminated a cross section of the jet with a 532 nm continuous-wave laser (6 watt, Laserglow Technologies) and sheet-forming optical assembly (see Fig. 3.5a for the experimental setup). A high speed camera (Edgertronic SC2, Sanstreak Corp) with a fixed lens (Nikon 50 mm f/1.8 D) recorded the flow at 200 frames per second for 15 seconds. We processed the images in MATLAB using PIVlab (Thielicke and Sonntag, 2021). We measured the flow at a range of throttle values from 15% to 100%, and plotted a snapshot of the turbulent jet flow field at 75% throttle in Fig. 3.5c. Flow speeds of the physical jet are on the order of 1 $\mathrm{m\,s^{-1}}$, and the flow is turbulent. The thrusters were powered with a 12 V DC power adapter, and the speed was controlled via pulse width modulation (PWM) outputted by a Teensy 4.1 microcontroller. During navigation tests, the thruster operated at 35% throttle to generate a strong enough flow to be detectable by CARL without being too strong as to greatly disrupt swimming.

To verify that the pressure sensors on CARL can detect the jet profile, we statically mounted CARL in the tank at a depth of 30 cm and recorded the time-averaged velocity at locations along the plume profile using the Pitot tube equation. The time-averaged vertical flow speeds measured by the pressure sensors are compared

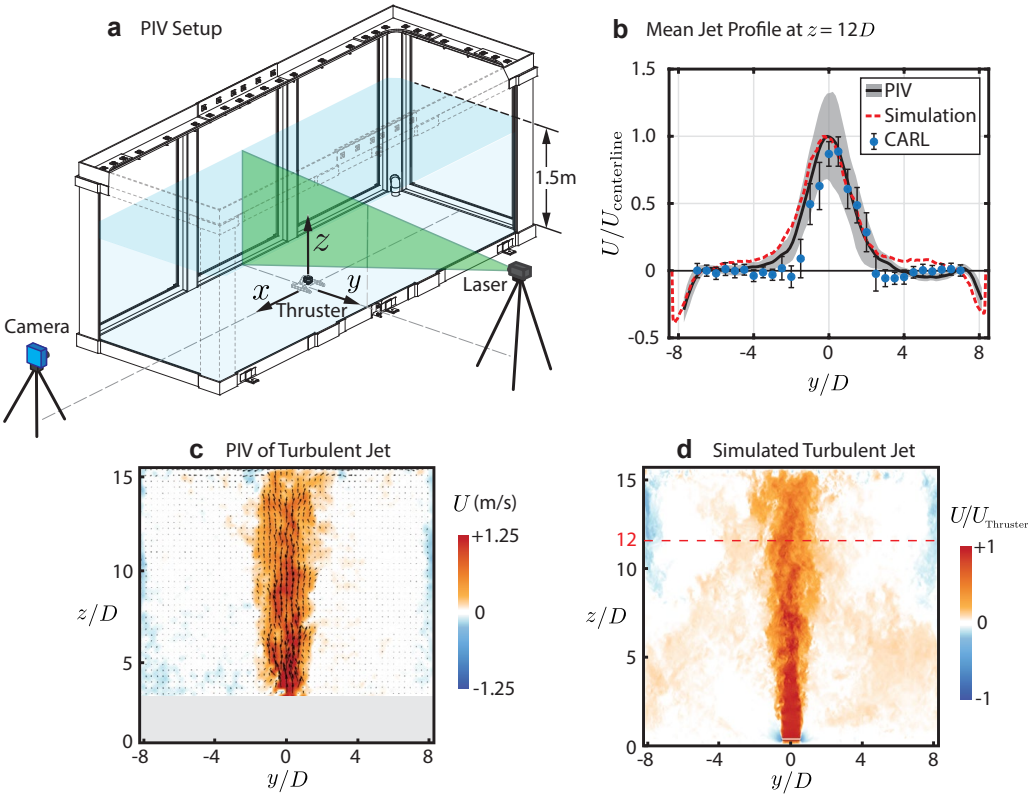


Figure 3.5: Measurement and simulation of the turbulent plume. (**a**) PIV setup. (**b**) Mean flow profile at a depth of 30 cm below the water surface ($z \approx 12D$) as computed from PIV, a static traverse of CARL in the tank, and the simulated jet flow. Velocity measurements from CARL are normalized by the centerline velocity from PIV measurements ($U_{\text{centerline}} \approx 0.58$ m s^{-1}). The error band and error bars indicate one standard deviation of the measured flow, which arises from the combination of turbulent fluctuations and measurement noise. (**c**) Snapshot of the turbulent jet PIV. Arrows are plotted to indicate flow direction and magnitude. Colors indicate the magnitude of the vertical velocity component U. (**d**) Snapshot of the simulated turbulent jet flow. Colors indicate the magnitude of the vertical velocity component normalized by the velocity imposed at the jet outlet (U_{Thruster}). A dotted red line at $z = 12D$ indicates the depth of the mean profiles plotted in (**b**).

with the mean flow profile measured with PIV in Fig. 3.5b and show good agreement. This demonstrates that in a time-averaged sense, the flow sensors onboard CARL can detect and measure the mean profile of the turbulent jet flow. Error bars on the measurements from CARL indicate error due to the combination of turbulence and inherent sensor noise, which highlights a limitation to the sensing abilities of CARL for this flow field: if CARL is greater than $\sim 2.5D$ away from the center of the plume, the noise is greater in magnitude than the mean flow.

3.4.4 Simulated turbulent jet flow

To generate a flow field for the simulated robot that models the turbulent jet flow in the physical tank, we simulated the jet flow using a lattice Boltzmann solver with a Smagorinsky-Lilly subgrid turbulence model (FluidX3D software Lehmann, 2023). Taking advantage of the fact that the three thrusters are equally spaced in the tank, we simulated one-third of the tank volume with one thruster. For simplicity, we applied a no-slip condition to all boundaries including the free surface. The thruster was modeled by enforcing a uniform velocity condition on a disk of magnitude U_{thruster} with the same dimensions as the thruster outlet and applying a jet Reynolds number of 100,000 to approximately match the flow speeds found by PIV measurements of the physical jet. The flow was simulated on a uniform 652 by 622 by 612 grid on an NVIDIA RTX 3090 GPU, resulting in a grid cell size of $\Delta x \approx D/36$. We generated 150 seconds of the turbulent jet flow, which is significantly longer than it takes CARL to swim across the tank, and used a 2D slice of the vertical velocity component at the swimming depth of CARL for training (see Fig. 3.2d). A snapshot of the turbulent jet flow is shown in Fig. 3.5d and the mean profile is plotted in Fig. 3.5b, and show good agreement with PIV measurements of the physical jets.

3.4.5 Reinforcement learning algorithm

To train a navigation policy for the simulated swimmer, we implemented the Double DQN algorithm (Hasselt, Guez, and Silver, 2015), which seeks to optimize the actions of an agent to maximize a cumulative reward function. DDQN trains two sets of weights in a neural network to predict the Q-values, i.e., the value of an action in a particular state, and selects actions with higher predicted Q-values. DDQN as implemented in this study is limited to discrete outputs, however, this proved sufficient for the plume-finding task. For the Q-network, we used a two-layer multi-layer perceptron network with 64 softsign neurons per layer, which has been sufficiently expressive to solve flow-based navigation problems in previous

work (Peter Gunnarson et al., 2021). Other RL algorithms may certainly provide stability or data efficiency benefits for solving tasks involving fluids (Berger et al., 2024), but because fluid-robot interactions are ignored in this simplified simulation, exploring the environment is computationally inexpensive and thus data efficiency is not critical.

The virtual CARL received a reward if the center of a jet was successfully reached, which occurred if CARL swam within a diameter of $2D$ from the center of a turbulent plume. In previous work for navigating in flow fields (Biferale et al., 2019; Peter Gunnarson et al., 2021), intermediate rewards such as the change in distance to the target were necessary to supply a consistent reward signal during training. Untrained swimmers could not overcome the strong background flow and reach the target, and would therefore not receive a reward signal without the inclusion of an intermediate reward. Here, in the absence of strong currents that restrict swimming, random exploration occasionally resulted in successfully finding the center of a plume, therefore intermediate rewards were not required.

3.4.6 RL-inspired navigation policy

The RL-inspired navigation policy is described below in Algorithm 1. The RL-inspired policy involves turning in the direction of the transverse velocity gradient, as sensed by $(\Delta P_{\text{right}} - \Delta P_{\text{left}})$. If the transverse velocity gradient is less than twice the RMS of the sensor noise ($\sigma_{\Delta P}$), then CARL swims straight forward. This is changed from the random swimming of the simulated policy, since straight swimming still achieves random exploration of the tank, but induces fewer vibrations and jittery motion in the physical robot compared with taking random actions. The sensor noise was measured beforehand by recording the RMS value of $(\Delta P_{\text{right}} - \Delta P_{\text{left}})$ while swimming in the tank with no flow from the thrusters.

Algorithm 1 RL-inspired navigation protocol

while battery voltage > 3.5 V **do**
    1) Measure transverse gradient ($\Delta P_{\text{right}} - \Delta P_{\text{left}}$)
    2) Pick swimming direction θ:
    if $\Delta P_{\text{right}} - \Delta P_{\text{left}} \geq 3\sigma_{\Delta P}$ **then**
        $\theta \leftarrow 75°$
    else if $2\sigma_{\Delta P} \leq \Delta P_{\text{right}} - \Delta P_{\text{left}} < 3\sigma_{\Delta P}$ **then**
        $\theta \leftarrow 35°$
    else if $-2\sigma_{\Delta P} < \Delta P_{\text{right}} - \Delta P_{\text{left}} < 2\sigma_{\Delta P}$ **then**
        $\theta \leftarrow 0°$
    else if $-3\sigma_{\Delta P} \leq \Delta P_{\text{right}} - \Delta P_{\text{left}} < -2\sigma_{\Delta P}$ **then**
        $\theta \leftarrow -35°$
    else if $\Delta P_{\text{right}} - \Delta P_{\text{left}} \leq -3\sigma_{\Delta P}$ **then**
        $\theta \leftarrow -75°$
    end if
    3) Swim for one timestep with direction θ
    if CARL impacts tank wall **then**
        Turn around by random angle between 90° and 270°
        Episode ← Episode + 1
    end if
end while

Acknowledgments

Funding: This work was supported by the National Science Foundation Alan T. Waterman Award and NSF Graduate Research Fellowship Grant No. DGE 1745301. **Author contributions:** P.G. and J.O.D. conceived of project, P.G. conducted experiments, P.G. and J.O.D. analyzed results and wrote paper. **Competing interests:** The authors declare that they have no competing financial interests. **Data and materials availability:** The data collected by CARL, code for simulating the turbulent jets, and RL algorithm used in this study are available at https://doi.org/10.22002/da2jr-yx734.

Chapter 4

SURFING VORTEX RINGS FOR ENERGY-EFFICIENT PROPULSION

Abstract

Exploiting background fluid flows for propulsion has the potential to enhance both the range and speed of AUVs. In this work, we demonstrate a novel strategy involving surfing a vortex ring for energy-efficient propulsion. First, an autonomous underwater robot used an onboard Inertial Measurement Unit (IMU) to sense the motion induced by the passage of a vortex ring generated by a thruster in a 13,000-liter water tank. In response to the sensed acceleration, an impulsive maneuver entrained the robot into the material boundary of the vortex ring. After entrainment, the robot was propelled across the tank without expending any additional energy or control effort. Modeling the energy and momentum transfer between the robot and the vortex ring reveals that the surfing strategy requires only one-fifth of the energy that would otherwise be expended under self-propulsion without the vortex ring. We then analyzed the controlled finite-time Lyapunov exponent field and corresponding Lagrangian Coherent Structures to understand the initial entrainment process and the sensitivity to the starting time and position of the surfing maneuver. Additionally, body acceleration as sensed by the onboard IMU was found to correspond with the acceleration of the background flow, and body rotation is suggested as a method for measuring the vorticity of the vortex ring. This study serves as a proof-of-concept demonstration of the potential for onboard inertial measurements to enable efficient interaction with background fluid flows.

4.1 Introduction

The ability to leverage background fluid flows for efficient propulsion has significant implications for ocean-exploring vehicles. Rather than expending energy to swim against ocean currents, AUVs may be able to significantly improve their range and speed by maneuvering into unsteady background currents that push the robot towards the desired destination (Inanc, S. Shadden, and J. Marsden, 2005; Rhoads, Mezić, and Poje, 2013).

While underwater, AUVs typically lack communication with the surface or real time knowledge of the full background flow field. Therefore, the challenge of efficient navigation must be accomplished with onboard sensing, computation, and actuation. Recent approaches for determining efficient navigation strategies using onboard sensors have involved data-driven techniques such as Reinforcement Learning in a variety of canonical and oceanic flow environments (e.g., Bellemare et al., 2020; Hang et al., 2023; Peter Gunnarson et al., 2021; Masmitja et al., 2023; Reddy et al., 2018; Krishna, Brunton, and Song, 2023). While powerful, these data-driven navigation algorithms often lack verification in physical robotic systems, particularly in ocean applications (Masmitja et al., 2023). To accomplish robotic sensing of background flows, researchers have invented various onboard flow sensors, including bio-inspired sensors (Ko, Lauder, and Nagpal, 2023; Bora et al., 2018; Zhai, Zheng, and Xie, 2021). However, solving the combined problem of sensing the background fluid flow with physical onboard sensors and then navigating intelligently in response to those measurements, remains an active area of research.

A promising approach for sensing background flow is to infer it from inertial measurements. For example, acceleration experienced by a robot may indicate the presence and direction of a background current, which could be used for efficient navigation. In nature, it has been observed that aquatic animals such as fish use their vestibular system to detect body acceleration induced by background flows (Coombs et al., 2014). In robotic applications, inertial measurements have the advantage of being inexpensive to implement due to the ubiquity of micro-electromechanical systems (MEMS) accelerometers. For example, IMUs are often already present on many existing robots such as quadcopters for stabilization and inertial guidance. Several studies have investigated using inertial data to infer background wind flows in aerial vehicles. For example, the tilt of a robot can aid with the identification of wind for more accurate maneuvering (O'Connell et al., 2022). The combination of vertical acceleration and roll in a remote-controlled airplane was used to

autonomously detect and exploit atmospheric thermal currents (Reddy et al., 2018). Additionally, micro aerial vehicles have been suggested as Lagrangian flow tracers for atmospheric flow measurement (Simon et al., 2023). However, to the best of our knowledge, combining onboard inertial data for efficient underwater navigation is a relatively unexplored technique.

In this study, we demonstrate experimentally a strategy for exploiting passing vortex rings for propulsion using inertial data as a sensory input. We tested the strategy using CARL, a palm-size autonomous underwater robot (P. Gunnarson and J.O. Dabiri, 2024). A thruster in a 13,000-liter water tank generated individual vortex rings, which served as a repeatable background flow unit that CARL exploited for forward propulsion. To detect both the presence and location of the vortex ring relative to CARL, an inexpensive accelerometer (~$5, MPU-6050) measured the motion induced by a passing vortex ring. Equipped with the ability to determine the relative location of the vortex ring, CARL executed a short burst maneuver in the direction of sensed acceleration, which typically resulted in entrainment into the vortex ring. After entrainment, CARL remained caught inside the vortex ring, surfing the flow structure across the tank without the need for additional control effort or propulsion.

By analyzing the transfer of energy and momentum from the vortex ring to CARL, we find that the surfing strategy requires one-fifth of the energy consumption compared with self-propulsion across the same distance at the same speed. Additionally, the controlled finite-time Lyapunov exponent field provides an explanation of the dynamics responsible for converting the small surfing maneuver into a long-distance, energy-efficient trajectory. Lastly, body rotation is found to serve as an additional indirect flow signal that can indicate background vorticity to further increase the available knowledge of the background flow. This work demonstrates a flow-based navigation strategy that closes the loop between sensing a background flow and exploiting that flow with an efficient and targeted maneuver.

4.2 Experimental setup

As a testing environment, we used a portion of a 1.5 m deep, 1.8 m wide, and 4.8 m long water tank (Figure 4.1). Vortex rings were generated by pulsing a thruster (Blue Robotics T200) mounted on a wall of the tank pointing horizontally in the x-direction (see Figure 4.1). Details about the vortex generation and properties of the resulting vortex rings are discussed in Appendix C.1. The thruster has a diameter

of D = 10 cm, which is used as the reference length scale for this study.

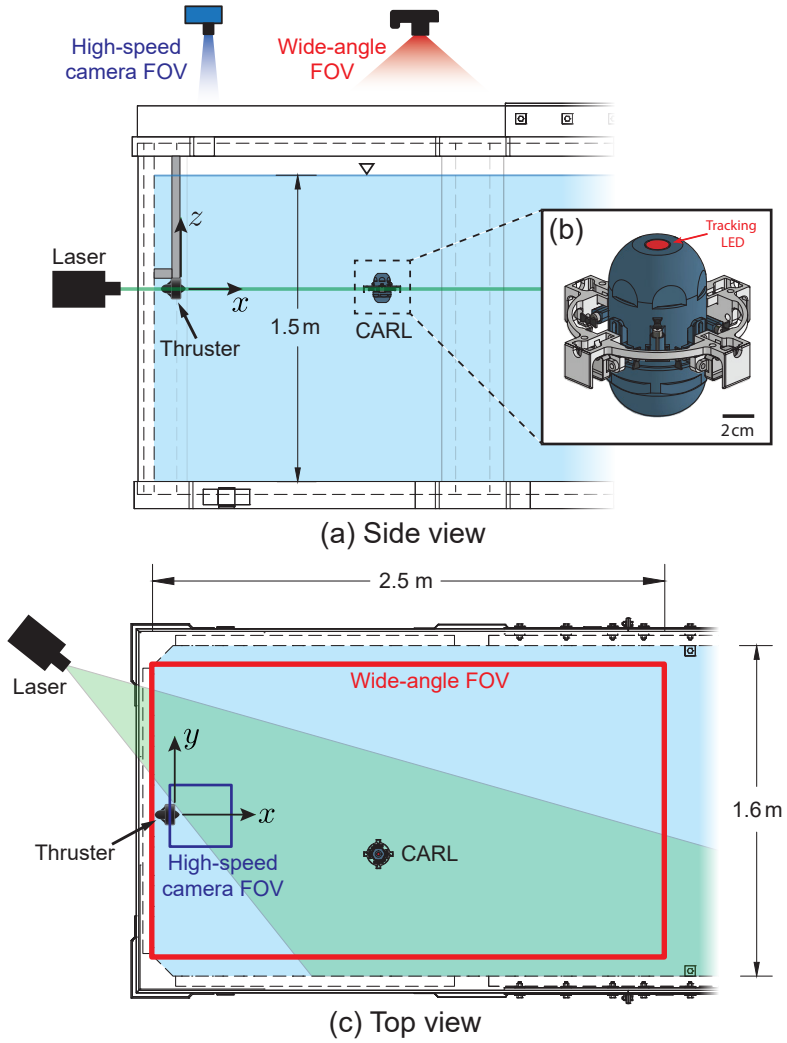


Figure 4.1: Experimental setup showing a side view (**a**) and top view (**c**) of the water tank. (**b**) CARL with a 2 cm scale bar and the LED used to for position tracking. A thruster mounted on a side wall of the tank (D = 10 cm) generates vortex rings ($\Gamma/\nu \approx$ 200,000; $L/D \approx 3.6$), which are detected and surfed by CARL. A wide-angle and a high-speed camera are mounted above the tank and track the position of CARL while simultaneously recording PIV measurements. A laser sheet illuminates the horizontal x-y plane for PIV measurements. For some PIV recordings, the laser was re-positioned to fully illuminate the entire FOV of the high-speed camera.

To test flow-based navigation in the water tank, we used CARL, a palm-sized, autonomous underwater robotic platform (see Appendix B for details). CARL was equipped with an onboard IMU (MPU-6050), which measured the linear acceleration and angular velocity of CARL. For each trial, CARL began at the surface, and was commanded to dive and maintain the same depth as the thruster. This

effectively constrained the navigation problem to the x-y plane. A Proportional-Integral-Derivative (PID) control loop ran onboard CARL at 50 Hz to control the depth using two vertically oriented motors. The depth was estimated using the hydrostatic pressure averaged from four onboard pressure sensors (MS5803-02BA, TE Connectivity). Another PID loop fixed the orientation of CARL in the vertical axis (z-axis) using the angular velocity data from the onboard IMU.

We mounted two cameras above the tank, which both recorded the position of CARL in the x-y plane and simultaneous Particle Image Velocimetry (PIV). The wide-angle camera (Canon RP with an RF 16 mm F2.8 STM lens) recorded approximately half of the area of the tank (2.5 by 1.6 meters) at a resolution of 1920 by 1080 pixels at 59.94 frames per second. The high-speed camera (Edgertronic SC2, Sanstreak Corp, with a Nikon 50 mm f/1.8 D lens) recorded a narrower field of view (FOV) near the jet (0.4 by 0.5 meters; 1280 by 1024 pixels at 300 frames per second) to better resolve the initial vortex formation and entrainment of CARL. The camera positions and fields of view are shown in Figure 4.1.

To track the position of CARL in the tank, we added an LED to the top of CARL (WS2812B RGB LED, ALITOVE) that changed from green to red at the start of onboard data recording. The change in color served to synchronize the camera position tracking with onboard measurements recorded by CARL. The position of CARL was computed by thresholding the red color channel of each frame of video and computing the "center-of-mass" of the result. Distortion corrections due to refraction at the air-water interface are described in Appendix C.2. The thruster was mounted approximately 0.56 meters below the free surface such that the generation of vortex rings had little effect on the free surface of the tank. For PIV measurements, we seeded the tank with 100-micron silver-coated hollow glass particles (AGSL150-30TRD, Potters Industries) and illuminated the horizontal x-y plane in the tank with a 532 nm continuous-wave laser (6 watt, Laserglow Technologies) and sheet-forming optical assembly (see Figure 4.1).

At the start of each episode, CARL was manually piloted to a position approximately $3D$ downstream and $2D$ in the negative y direction from the thruster (see Figure 4.5 for measured starting positions). CARL was then commanded to dive to the depth of the thruster and began recording onboard data. At this starting position, CARL was outside of the direct path of the vortex ring, but still close enough to sense the effects of a passing vortex ring. After CARL reached the depth of the thruster, the surfing policy described in the following section was activated, and the thruster was

commanded to generate a vortex ring. After 12 seconds, CARL stopped recording data and returned to the surface, which marked the end of an episode. The 12-second duration was chosen such that CARL always remained within the field of view of the wide-angle camera during an episode.

Throughout this study, time is normalized by the vortex formation time ($\overline{U}_p/D$, Gharib, Rambod, and Karim Shariff, 1998), which is measured to be approximately 9 seconds. The Reynolds number (Γ/ν) and formation number (L/D) of the generated vortex rings are approximately 200,000 and 3.6, respectively, which are discussed in Appendix C.1. The time $t = 0$ is defined by the origination of a vortex ring, which is estimated by fitting a model shown in Figure 4.3.

4.3 Vortex ring surfing strategy

To use the vortex ring for propulsion, CARL was programmed with a simple but effective policy: if the acceleration in the y-direction exceeded a threshold, CARL would swim impulsively in the same direction as that sensed acceleration. After this initial swimming maneuver, CARL entered the vortex ring, and was propelled across the tank without requiring any additional control effort or energy expenditure. It is important to highlight that this impulsive maneuver was perpendicular to the direction of surfing motion; zero thrust contributed directly towards the forward propulsion of CARL. An example trajectory and time-history of sensed acceleration is plotted in Figure 4.2. By accomplishing both tasks of sensing and navigating onto the vortex ring, CARL can autonomously exploit the background flow for energy-efficient propulsion.

Next, we describe each step of the surfing strategy for the example episode shown in Figure 4.2. As the vortex ring passed near CARL, CARL was pushed by the flow surrounding the vortex ring. The resulting body acceleration was detected by the onboard IMU (Figure 4.2c). For these experiments, the orientation of CARL was fixed so that the coordinates of the reference frame onboard CARL matched the coordinate system of the tank. Therefore, the sign of the acceleration in the y-direction indicated whether the vortex ring was to the left or right side of CARL. If the y-acceleration exceeded a threshold of $0.5\,\mathrm{m\,s^{-2}}$, CARL swam at maximum thrust for 0.3 seconds in the same direction as the sensed y-acceleration. This impulsive maneuver was highly successful at entraining CARL into the vortex ring. Additionally, the surfing strategy is fully autonomous, i.e., CARL had no prior knowledge of where and when the vortex ring would propagate.

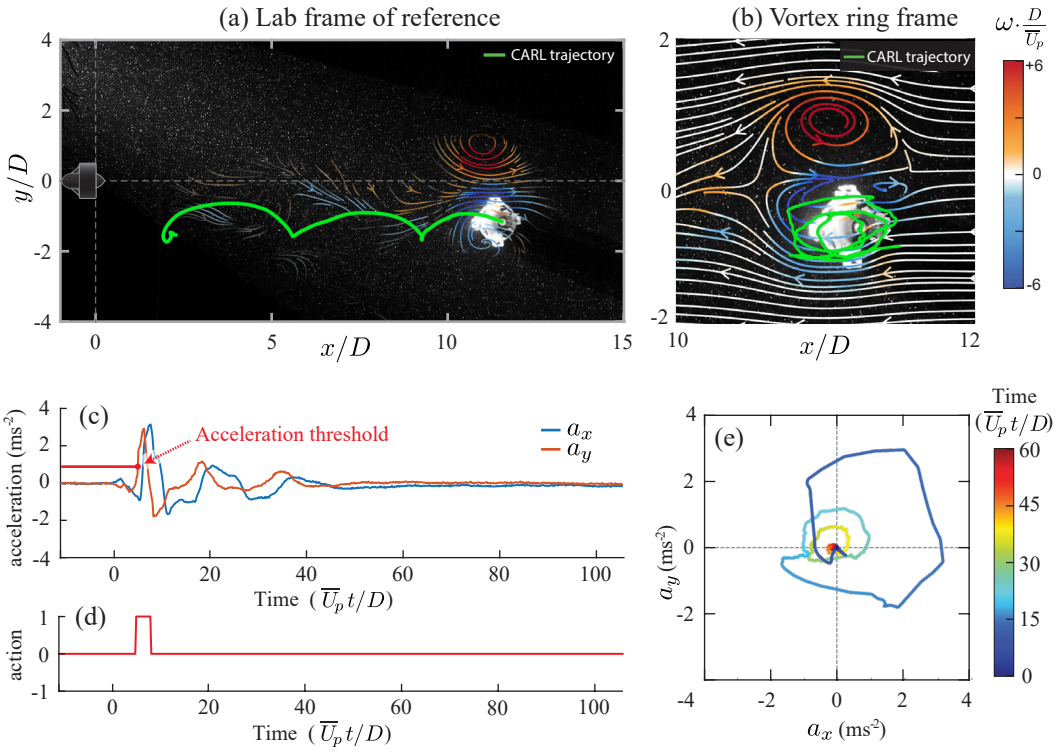


Figure 4.2: Detecting and exploiting a vortex ring for propulsion. (**a**) Example trajectory of CARL (green line) while surfing a vortex ring. Using a snapshot of simultaneous PIV measurements, instantaneous streamlines are plotted in the lab frame of reference and shaded by the vorticity to visualize the vortex ring. (**b**) Flow streamlines are plotted in a reference frame translating with the vortex ring. The trajectory of CARL is contained inside the lower half of the vortex ring and orbits in the same direction as the local vorticity. (**c**) Linear acceleration signal sensed by the IMU onboard CARL. The red line indicates the threshold used to detect the presence of a vortex ring, which autonomously triggered an impulsive maneuver (**d**) with the same sign as the sensed y-acceleration. After the impulsive maneuver, CARL remained inside the vortex ring with no additional control effort or energy expenditure. (**e**) The x and y components of acceleration orbit the origin, further highlighting the circular motion of CARL induced by the vortex ring.

After successful entrainment, CARL traveled with the vortex ring across the tank, which is visualized in Figure 4.2a. In Figure 4.2b, the trajectory of CARL and the flow measured by PIV from the wide-angle camera are plotted in a frame of reference traveling at the same speed as the vortex ring. While entrained inside the vortex ring, CARL orbits the lower half of the vortex ring in a clockwise direction, i.e., in the same direction as the local vorticity in that part of the vortex ring. In this sense, the translation of CARL matches that of the surrounding fluid. Additionally, the vorticity of the surrounding fluid may be expected to induce body rotation, but

CARL maintains a constant heading using the onboard PID control loop.

While the navigation policy used in these experiments is simple, it highlights the potential of combining onboard sensing and navigation for exploiting unsteady background flows. Additionally, it is straightforward to extend this strategy beyond sensing and swimming in only the y-direction. For example, both components of the onboard acceleration clearly detect the motion induced by the vortex ring. Figure 2e plots the x-component of acceleration versus the y-component. Before the vortex ring is generated, the acceleration vector starts at the origin. When the vortex ring passes, the acceleration increases to its maximum magnitude, and begins orbiting the origin in a clockwise direction, which matches the sign of the nearby vorticity in the vortex ring. After CARL is entrained into the vortex ring, the acceleration vector continues to orbit the origin in the same clockwise direction, gradually decreasing in magnitude as the vortex ring slows and diffuses. This circular motion can also be seen in Figure 4.2c, in which the phase of the x and y acceleration appear offset by roughly 90 degrees. In addition to linear acceleration, there are other types of sensor measurements that could be used to detect the vortex ring. For example, the potential for rotation and vorticity sensing is discussed in Section 4.9.

4.4 Repeatability and success rate

The autonomous surfing strategy is highly repeatable and successful. We recorded the results of the surfing strategy over 37 episodes, and compared the resulting trajectories with 26 episodes in which CARL took no action, and 46 episodes in which CARL attempted to avoid the vortex ring (i.e., swam in the opposite direction of sensed y-acceleration). These trajectories are plotted below in Figure 4.3.

For the surfing policy, in which CARL attempted to surf the vortex rings (Figure 4.3a), the location of the vortex ring was correctly identified in 81% percent of the episodes. Successful identifications are visualized by the color of the trajectories: blue trajectories indicate that CARL swam in the direction of the vortex ring; black trajectories indicate that no vortex ring was detected; red trajectories indicate that CARL swam away from the vortex ring (i.e., the sign of detected acceleration was in the negative y-direction). Of those episodes in which the vortex ring was correctly identified, CARL successfully swam into the vortex ring at a rate of 77%. In summary, both tasks of sensing the vortex ring and navigating into the vortex ring have high success rates with this acceleration-based surfing strategy.

For comparison, we also plot trajectories in which CARL was programmed to not

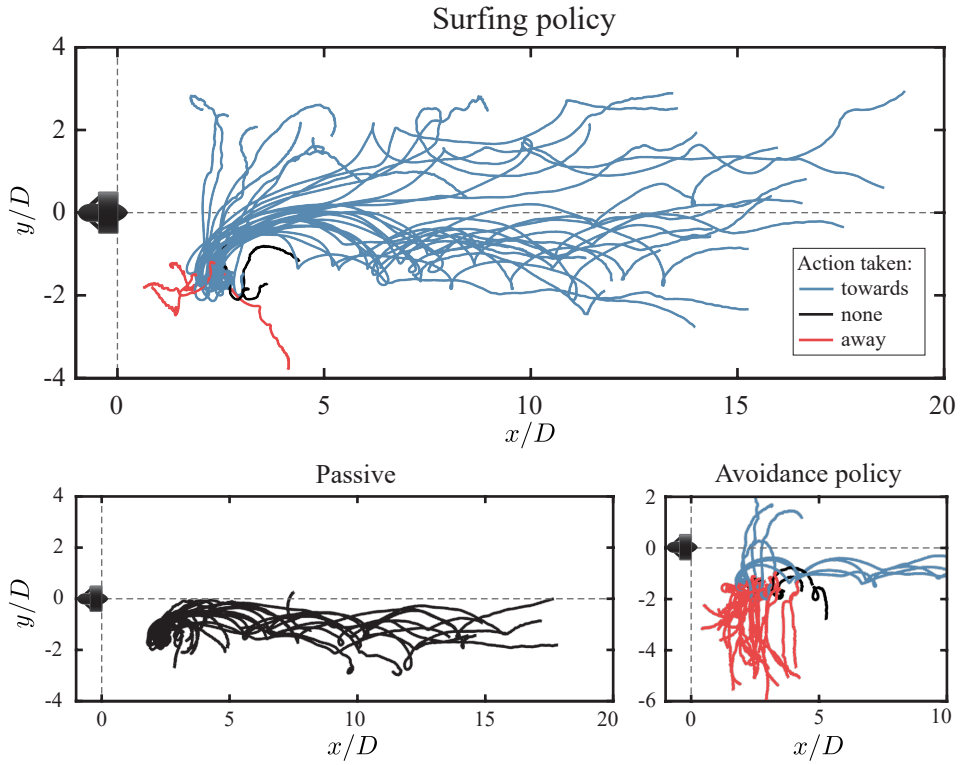


Figure 4.3: Comparison of control strategies for exploiting or avoiding a passing vortex ring. (**a**) Surfing policy (N=37). Using the acceleration in the y-direction, CARL correctly identified the presence and direction of the vortex rings in 81% of trials, which triggered a swim maneuver towards the vortex rings (blue lines). Of those correct identifications, CARL successfully surfed the vortex ring at a rate of 77%. (**b**) CARL was commanded to take no action (N=26), which results in entrainment in 50% of episodes. (**c**) CARL was commanded to avoid the vortex rings (N=46). CARL correctly identified the location of the vortex ring with a rate of 72%, resulting in a swim maneuver away from the vortex rings (red lines). The avoidance maneuver successfully avoids entrainment in 100% of episodes.

take any action (Figure 4.3b) and to avoid the vortex ring (Figure 4.3c). As with the surfing policy, the avoidance policy is highly effective: CARL correctly detects the vortex ring in 72% of episodes and avoids the vortex ring in 100% of those episodes.

The inclusion of the avoidance policy is motivated for two reasons. For a navigation task in the real world, flow structures may not necessarily propagate in the direction of desired motion. In that case, it would be advantageous to avoid entrainment to prevent motion in a counterproductive direction. Additionally, many animals such as jellyfish employ vortex rings for prey capture (Peng and J. O. Dabiri, 2009). In turn, these preys often exhibit escape maneuvers in response to deformation of the surrounding flow, and thus it may be insightful to use these robotic experiments to

probe questions of predator–prey interactions. The effectiveness of the avoidance policy demonstrates that the surfing policy can be simply adjusted to test these different behaviors of surfing or avoiding flow structures.

4.5 Energy savings and momentum transfer

Surfing on background flow structures has the potential to greatly reduce the energy required for propulsion. In this section, we quantify the energy savings gained by surfing vortex rings in our experiment and analyze the transfer of momentum between the vortex ring and CARL.

To quantify the energy savings gained by vortex ring surfing, we compare the energy required to surf a vortex ring to the energy required to travel the same distance under self-propulsion. To surf the vortex ring, CARL swims at maximum thrust for 0.3 seconds, which accelerates CARL from rest to a speed of $u_{\text{impulse}} \approx 24\,\text{cm}\,\text{s}^{-1}$. Afterwards, no energy is expended to maintain position inside the vortex ring. The primary energy expenditure during this maneuver is the change in kinetic energy of CARL required to accelerate CARL from rest:

$$E_{\text{surf}} = \frac{1}{2} m_{\text{CARL}} (1 + \alpha_{xx}) u_{\text{impulse}}^2 \approx 16.7\,\text{mJ}, \tag{4.1}$$

where m_{CARL} is the mass of CARL (~355 g) and α_{xx} is the added mass coefficient of CARL in the direction of propulsion (~0.63). To estimate the added mass coefficient, we fit a prolate ellipsoid to the shape of CARL with a volume of equal to m_{CARL}/ρ and used the analytical formula for added mass coefficients from Lamb (1932). The speed u_{impulse} was estimated by measuring the position of CARL while executing an impulsive maneuver in quiescent flow.

To estimate the energy required to swim the same distance without the aid of the vortex ring, we considered steady swimming at constant velocity. Under this condition, the primary energy expenditure is overcoming steady drag while swimming at the speed u_{swim}. To be a fair comparison, we assume that u_{swim} is the same as the average speed of the vortex ring, $\overline{u}_{\text{vortex}}$, over the duration of an episode. In other words, this analysis considers the energy expended to swim the same distance in the same amount of time as surfing the vortex ring:

$$E_{\text{self-propulsion}} = \frac{1}{2} \rho \overline{u}_{\text{vortex}}^2 C_d A L \approx 82.7\,\text{mJ}, \tag{4.2}$$

where C_d is the drag coefficient ($\sim$1.1), A is the frontal area of CARL ($\sim$78.3 cm^2), and L is the distance traveled during a surfing episode ($\sim$ 1.32 m). The drag coefficient was estimated by tracking the position of CARL while coasting after an impulsive maneuver. Using these values, the energy saved by surfing the vortex ring can be estimated:

$$E_{\text{self-propulsion}}/E_{\text{surf}} \approx 4.9. \tag{4.3}$$

By surfing the vortex ring, CARL requires approximately 4.9 times less energy than would be expended to traverse the same distance under self-propulsion, clearly demonstrating the potential for energy savings using background flow features.

To be sure, there is room for optimization for both cases of vortex surfing and self-propulsion. For example, it may be possible to jump onto the vortex ring with a significantly smaller impulsive maneuver depending on the initial position of CARL relative to the vortex ring (see Section 4.6 for a discussion). Likewise, CARL has a relatively high C_d of 1.1, so the energy required for self-propulsion could be significantly reduced through streamlining. However, our results demonstrate that, for a given vehicle design, energy can be harvested from the background flow by appropriately maneuvering in response to onboard detection of the background flow. In the case of our experiments, the energy savings is significant. Additionally, the thrust used for the impulsive surfing maneuver was directed in the y-direction, perpendicular to the forward motion of CARL after entrainment into the vortex ring. In this sense, none of the thrust generated by CARL during the impulsive maneuver directly contributed towards forward propulsion; energy for forward propulsion was derived from the surrounding flow.

To this point, it is important to note that the energy gained by exploiting vortex rings for propulsion does not appear from thin air; energy must be extracted from the vortex ring according to the conservation of energy. Therefore, we next consider how the entrainment of CARL affects the velocity of the vortex ring. In Figure 4.4, the streamwise position of vortex rings with and without CARL are plotted as a function of time. The position of the vortex rings was computed by measuring the "center of mass" of the velocity magnitude from simultaneous PIV data taken with the wide-angle camera. It is expected that the energy and therefore speed of the vortex should be reduced by the entrainment of CARL, but within the error of our measurements, the entrainment of CARL does not significantly alter the average

trajectory of the vortex rings.

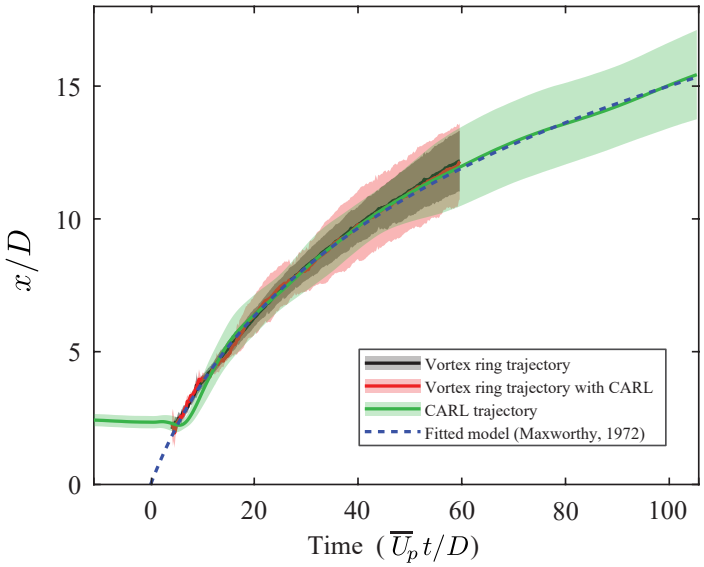


Figure 4.4: Average trajectories of the vortex rings and CARL. The average trajectory of the vortex ring propagating freely in the tank (black line, $N = 5$) does not significantly change when CARL is entrained (red line, $N = 3$). This indicates that the vortex rings do not significantly change speed if CARL is entrained, which is consistent with an inelastic collision model ($m_{\text{vortex}}/m_{\text{CARL}} \approx 22$). For comparison, the average trajectory of CARL is plotted in green ($N = 35$). Whereas the duration of the PIV data is limited by the extent of the laser sheet, the tracking LED on CARL is visible for the entire duration of each trial. CARL is initially stationary, and then is entrained by a vortex ring. After entrainment, the average trajectory of CARL closely tracks the average vortex position (circular motion is averaged out between trials). The Maxworthy model for vortex propagation (blue) is fitted to the average CARL trajectory, and the intersection with $x/D = 0$ is used to define the time $t = 0$.

To model the transfer of momentum between the vortex rings and CARL, we consider an inelastic collision model, in which CARL and the vortex ring are treated as solid bodies, and CARL travels with the vortex ring after "colliding" with the vortex ring. To estimate the mass and added mass coefficient of the vortex ring ($m_{\text{vortex}} \approx 9.0\,\text{kg}$; $\alpha_{xx}^{\text{vortex}} \approx 0.44$), we fitted an oblate ellipsoid to the dividing streamline of the vortex ring as measured by PIV from the high-speed camera. The change in speed of the vortex is then simply due to conservation of momentum and energy through the collision:

$$\frac{u_{\text{vortex,CARL}}}{u_{\text{vortex}}} = \left(1 + \frac{m_{\text{CARL}}(1+\alpha_{xx}^{\text{CARL}})}{m_{\text{vortex}}(1+\alpha_{xx}^{\text{vortex}})}\right)^{-1} \approx 0.96. \tag{4.4}$$

Because the vortex ring is approximately 22 times as massive as CARL, the predicted change in velocity of the vortex ring is only four percent, which explains the lack of significant change in the trajectory of the vortex. While this model ignores the motion of CARL inside the vortex ring, and it is a simplification to treat the vortex ring and CARL as solid masses, the model underscores the point that the vortex ring is much more massive than CARL and therefore is not significantly affected by the entrainment of CARL. As another point of comparison, the total kinetic energy of the vortex ring was estimated to be approximately 1.5 J (see Appendix C.1 for details), which is roughly 23 times the energy saved by CARL when surfing the vortex ring. If the vortex rings were significantly smaller, or the robot significantly larger, one might expect the entrainment of the robot to alter the trajectory and possibly formation of the vortex ring.

A related question is the maximum possible distance that CARL could surf until the ring loses coherence, for example, through viscous diffusion or the growth of instabilities. In Figure 4.4, a model for the trajectory of the vortex ring ($x(t) = 1/c \ln(cu_0 t + 1)$, Maxworthy, 1972) was fitted to the CARL trajectory data, and shows good agreement with both the trajectory of CARL and the trajectories of the vortex rings for all of the available data. Additionally, simultaneous PIV measurements of CARL surfing the vortex ring show the presence of the vortex ring for the entire FOV of the wide-angle camera. Therefore, it may be possible for CARL to continue surfing the vortex rings for a longer duration than was recorded with our experimental setup. After a sufficiently long duration, the vortex ring may begin to break up or dissipate, although additional experiments are needed to determine whether the entrainment of CARL would accelerate or impede this process.

4.6 Connections with Lagrangian coherent structures

Tools from dynamical systems such as Lagrangian Coherent Structures (LCS) have become popular for analyzing transport in fluid flows. LCS define material barriers in fluid flow and have been used to analyze coastal flows (S. C. Shadden, Lekien, and J. E. Marsden, 2005), the spread of pollution in the ocean (Lekien et al., 2005), and prey capture during jellyfish feeding (Peng and J. O. Dabiri, 2009). Vortex rings can be viewed a Lagrangian Coherent Structure in the sense that a vortex ring forms a distinct region of transport of the fluid parcels inside the boundary of the vortex ring.

Additionally, researchers have drawn connections between LCS and optimal paths through flow fields in the context of underwater robots. For example, Inanc, S. Shadden, and J. Marsden (2005) observed that energy-efficient trajectories through ocean currents in Monterey Bay coincided with the LCS of the background flow (Inanc, S. Shadden, and J. Marsden, 2005). Taking this a step further, the LCS can be used as a tool for generating optimal paths (Senatore and Ross, 2008). More recently, Krishna et al. have shown that LCS can explain key characteristics of optimal trajectory generation in fluid flows, such as the sensitivity of the optimal path to initial conditions and cost functions (Krishna, Song, and Brunton, 2022; Krishna, Brunton, and Song, 2023). In this section, we use LCS to better understand the mechanisms underlying the vortex ring surfing strategy from a dynamical systems point of view.

A common method of quantifying LCS is to compute the finite-time Lyapunov exponent (FTLE), which measures the degree to which the paths of neighboring fluid parcels diverge over a fixed time horizon. The FTLE is a scalar field whose ridges correspond to transport barriers, and therefore also the boundaries of Lagrangian coherent structures (S. C. Shadden, Lekien, and J. E. Marsden, 2005).

The first step to computing the FTLE field is to cast the fluid flow as a dynamical system, in which a given fluid tracer particle with position $\mathbf{x}(t)$ moves according to the background flow:

$$\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}(t), t). \tag{4.5}$$

For a given point in the fluid domain, we can define the flow map Φ_0^T, which maps the starting position of a particle to the position after being advected by the background flow for a time T:

$$\Phi_0^T : \mathbf{x}(0) \mapsto \mathbf{x}(0) + \int_0^T \mathbf{u}(\mathbf{x}(t), t)\mathrm{d}t. \tag{4.6}$$

To compute the flow map using PIV data, a grid of particles is initialized over the measurement domain and advected according to the PIV flow field. Next, the Jacobian of the flow map $\mathbf{D}\Phi_0^T$ is computed by applying finite differences to the flow map of the grid of particles. In 2D, the computation is as follows:

$$
\begin{aligned}
(\mathbf{D}\Phi_0^T)_{i,j} &\approx \begin{bmatrix} \dfrac{\Delta x_i(T)}{\Delta x_i(0)} & \dfrac{\Delta x_j(T)}{\Delta y_j(0)} \\ \dfrac{\Delta y_i(T)}{\Delta x_i(0)} & \dfrac{\Delta y_j(T)}{\Delta y_j(0)} \end{bmatrix} \\
&= \begin{bmatrix} \dfrac{x_{i+1,j}(T) - x_{i-1,j}(T)}{x_{i+1,j}(0) - x_{i-1,j}(0)} & \dfrac{x_{i,j+1}(T) - x_{i,j-1}(T)}{y_{i,j+1}(0) - y_{i,j-1}(0)} \\ \dfrac{y_{i+1,j}(T) - y_{i-1,j}(T)}{x_{i+1,j}(0) - x_{i-1,j}(0)} & \dfrac{y_{i,j+1}(T) - y_{i,j-1}(T)}{y_{i,j+1}(0) - y_{i,j-1}(0)} \end{bmatrix},
\end{aligned} \tag{4.7}
$$

where i and j are the indices of a particle with position $\mathbf{x}_{i,j} = [x_{i,j}, y_{i,j}]$. The flow map Jacobian is then used to compute the Cauchy-Green deformation tensor:

$$
\Delta_{i,j} = (\mathbf{D}\Phi_0^T)^*(\mathbf{D}\Phi_0^T), \tag{4.8}
$$

where * denotes a matrix transpose. Finally, taking the logarithm of the maximum eigenvalue of $\Delta_{i,j}$ at every grid point generates the FTLE field:

$$
\sigma_{i,j} = \frac{1}{|T|}\log\sqrt{\lambda_{\max}(\Delta_{i,j})}. \tag{4.9}
$$

LCS can be extracted by computing ridges of the FTLE field. In this study, we simply threshold the value of the FTLE field to visualize coherent structures.

The concept of the FTLE field can be extended to incorporate the active motion of swimmers or the inertia of particles in the flow. For example, this has been referred to as the control FTLE (cFTLE, e.g., Krishna, Song, and Brunton, 2022) or particle LCS (pLCS, e.g., Peng and J. O. Dabiri, 2009), where each fluid tracer is modeled as moving according to the background flow and the swimmer's self-propulsion. In our case:

$$
\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}(t), t) + \mathbf{u}_{\mathrm{CARL}}(t), \tag{4.10}
$$

where $\mathbf{u}_{\mathrm{CARL}}(t)$ is the motion due to CARL executing the impulsive surfing maneuver. This model ignores coupling between CARL and the surrounding flow, and also does not include the inertia of CARL, but offers qualitative insight into dynamics

of this vortex surfing strategy. The cFTLE effectively shows transport boundaries for the swimmer rather than just the surrounding fluid, which gives insight into the swimmer trajectories and their sensitivity to initial conditions.

As an additional note, in the above formulae, particles are integrated forward in time, which results in repelling LCS, or boundaries over which particle paths diverge. Integrating particles backwards in time computes structures that are attracting, which is of particular interest for the particle entrainment into vortex rings.

In Figure 4.5a, we plot the attracting and repelling LCS for a single vortex ring on top of flow streamlines. As expected, the LCS delineates the boundary of the vortex ring. Barring a small amount of viscous diffusion, particles inside the vortex ring stay inside the vortex ring. Below in Figure 4.5d, we plot the starting points of CARL in which CARL takes no action and drifts passively in the flow (i.e., $\dot{\mathbf{x}} = \mathbf{u}$). The starting positions are shaded red if CARL was entrained into the vortex ring. For unsteady flows such as the vortex rings in this study, the FTLE field is time-dependent. Therefore, all FTLE computations are performed with an initial time chosen to approximately match the location of the vortex ring with the CARL starting positions plotted in the bottom row of Figure 4.5. The time interval of integration ($T = 0.5$ s) was chosen to be of sufficient duration to reveal the LCS of the vortex ring, while also not being so long that too many particles exited the PIV domain.

To investigate the effects of the vortex surfing maneuver, we plot the cFTLE for CARL swimming impulsively into the vortex ring (Figure 4.5b) and away from the ring (Figure 4.5c). In these cases, the velocity of the swimmer $\mathbf{u}_{\mathrm{CARL}}$ is directly copied from position data of CARL executing the impulsive surfing maneuver in the appropriate direction.

Figures 4.5a and 4.5b show the impact of the surfing maneuver on the LCS. When swimming into the vortex ring, the attracting region extends downwards in the negative y direction. Correspondingly, the percentage of starting points that are captured by the vortex ring increases from 27% with passive swimming to 80% with the surfing maneuver (Figure 4.5e). Conversely, when CARL executes a maneuver to avoid the vortex ring (Figures 4.5b and 4.5c), the attracting cLCS shifts upwards, while the repelling LCS moves downwards, and none of the starting locations are entrained.

The starting positions for the vortex surfing case appear divided into two clusters,

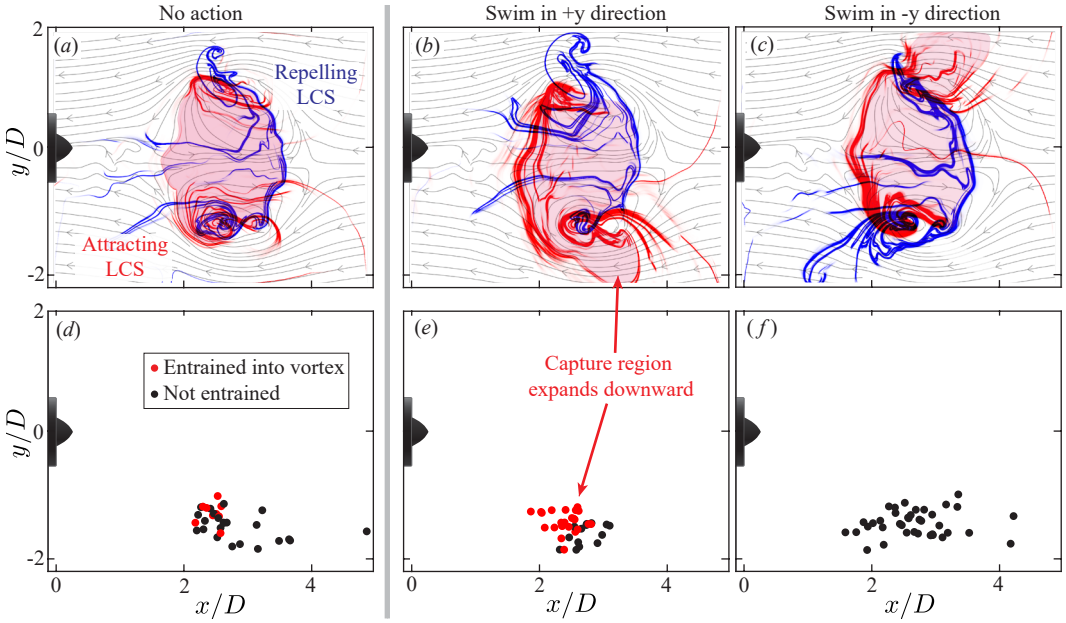


Figure 4.5: LCS may explain the positional dependence of entrainment. Top row: computed from PIV measurements of a vortex ring, the forward time (red) and backward time (blue) FTLE field is plotted for the undisturbed flow (**a**) and with an impulsive maneuver towards (**b**) and away (**c**) the vortex ring that matches the impulsive maneuvers performed by CARL. Streamlines of the undisturbed flow in the vortex ring reference frame are plotted in the background for comparison. Bottom row: starting locations of CARL, shaded red if CARL is entrained in the vortex ring. If CARL moves passively, the LCS (**a**) matches the boundary of the vortex ring, and the entrainment rate (**b**) is 26%. When swimming in the positive y direction (**b**,**e**), the attracting LCS extends downward, and 63% of the starting locations result in entrainment. When swimming in the negative y direction (**c**,**f**), the repelling LCS extends downward, and there are no starting locations that result in entrainment.

with the entrained (red) points positioned closer to the jet. The pLCS appears to explain this division: points lying within the attracting pLCS are entrained, while points outside are not entrained. In this way, the LCS may both predict and explain the effect of a given control policy on the entrainment and future trajectory of a swimmer. Improved modeling of the inertia of the robot may enable further quantitative comparisons (e.g., Peng and J. O. Dabiri, 2009).

The FTLE field also explains why a small impulsive maneuver from CARL can have a large impact on final position and energy savings. At regions of high stretching, one can achieve a large change in position for a small input. This idea has been formalized throughout the FTLE literature by considering the effect of a small

perturbation in position, ϵ, on the flow map $\Phi(x)$. Taking a Taylor expansion about $\epsilon = 0$:

$$\Phi(x + \epsilon) = \Phi(x) + \mathbf{D}\Phi(x) \cdot \epsilon + O(\epsilon^2). \tag{4.11}$$

If the flow map Jacobian $\mathbf{D}\Phi(x)$ is large, which is the case at the boundary of an LCS or the ridge of an FTLE field, then to leading order, a small jump in position (ϵ) can become amplified into a large change in the flow map. Surfing vortex rings takes this idea to an extreme: a small jump across the boundary of the vortex ring results in CARL being propelled across the tank, whereas without the perturbation, CARL remains effectively stationary. Furthermore, because of the dot product in the $\mathbf{D}\Phi(x) \cdot \epsilon$ term, the direction of ϵ matters: CARL must maneuver in the right direction to either avoid or jump onto the vortex ring, depending on direction of stretching in the background flow.

Additionally, Krishna, Song, and Brunton (2022) observed that for energy-efficient trajectories, there are spikes in control effort and cost function that correlate with LCS boundaries because of this amplification of small perturbations at LCS boundaries. Our experiments demonstrate this principle in practice: a small spike in thrust near the LCS boundary of the vortex ring results in large energy savings for forward propulsion.

4.7 Dependence on initial time and position

Lagrangian coherent structures help explain the sensitivity of trajectories to their initial conditions, such as the starting position of CARL. However, in unsteady flow fields, trajectories near boundaries of the LCS are also highly sensitive to the starting time (Lekien et al., 2005). From the perspective of CARL in the lab frame (i.e., while stationary in quiescent fluid), a passing vortex ring is an unsteady flow field, which implies that the final trajectory of CARL may be sensitive to both starting position and the starting time of the surfing maneuver. Phrased more colloquially, surfing a vortex ring involves being in the right place at the right time.

To illustrate the combined spatial and temporal dependence, several example CARL trajectories are plotted in Figure 4.6. In panel (a), we plot an example trajectory in which CARL successfully maneuvers into the vortex ring. On the left, the trajectory of CARL is plotted in the vortex ring reference frame, which illustrates CARL becoming entrained into the vortex ring after executing the surfing maneuver. On

the right, it is apparent that CARL becomes entrained into the lower half of the vortex ring, and remains below the y axis.

A similarly successful surfing trajectory is plotted in Figure 4.6b. However, in this example, CARL is entrained into the upper half of the vortex ring, and stays above the y axis after entrainment. This can be explained by the starting position of CARL when the surfing maneuver is executed. Compared with example (a), CARL begins closer to the centerline in example (b), and so the surfing maneuver carries CARL farther upwards and into the top half of the vortex ring.

These two examples illustrate that the trajectory of CARL is sensitive to the initial starting location near this region of high stretching at the vortex ring boundary. Additionally, the top and bottom halves of the vortex ring can each be considered distinct regions of material transport: CARL is either entrained into the top or bottom half of the vortex ring, and remains there after entrainment. The division between the top and bottom halves is faintly shown by the repelling LCS curves in Figure 4.5a-c.

To illustrate the temporal sensitivity of the surfing maneuver, we plot an unsuccessful trajectory in Figure 4.6c. Compared with the previous two examples, CARL begins the surfing maneuver at a later time. Or equivalently, in the vortex ring reference frame (left column of Figure 4.6), CARL begins the surfing maneuver farther to the left, or farther downstream, compared with examples (a) and (b). As a result, CARL executes the surfing maneuver too late and is not entrained into the vortex. In other words, CARL must detect the vortex ring early enough to successfully become entrained into the vortex ring. This idea is reflected in the pLCS in Figure 4.5b, because the capture region expands both down and also slightly upstream of the vortex ring.

The difference in maneuver starting time between the successful surfing examples (a,b) and the ”too late“ example (c) is approximately half of the duration of the maneuver, or about 0.15 seconds. The literal split-second time scale of the surfing maneuver highlights the power of using an autonomous robotic system to exploit background flow at these rapid time scales.

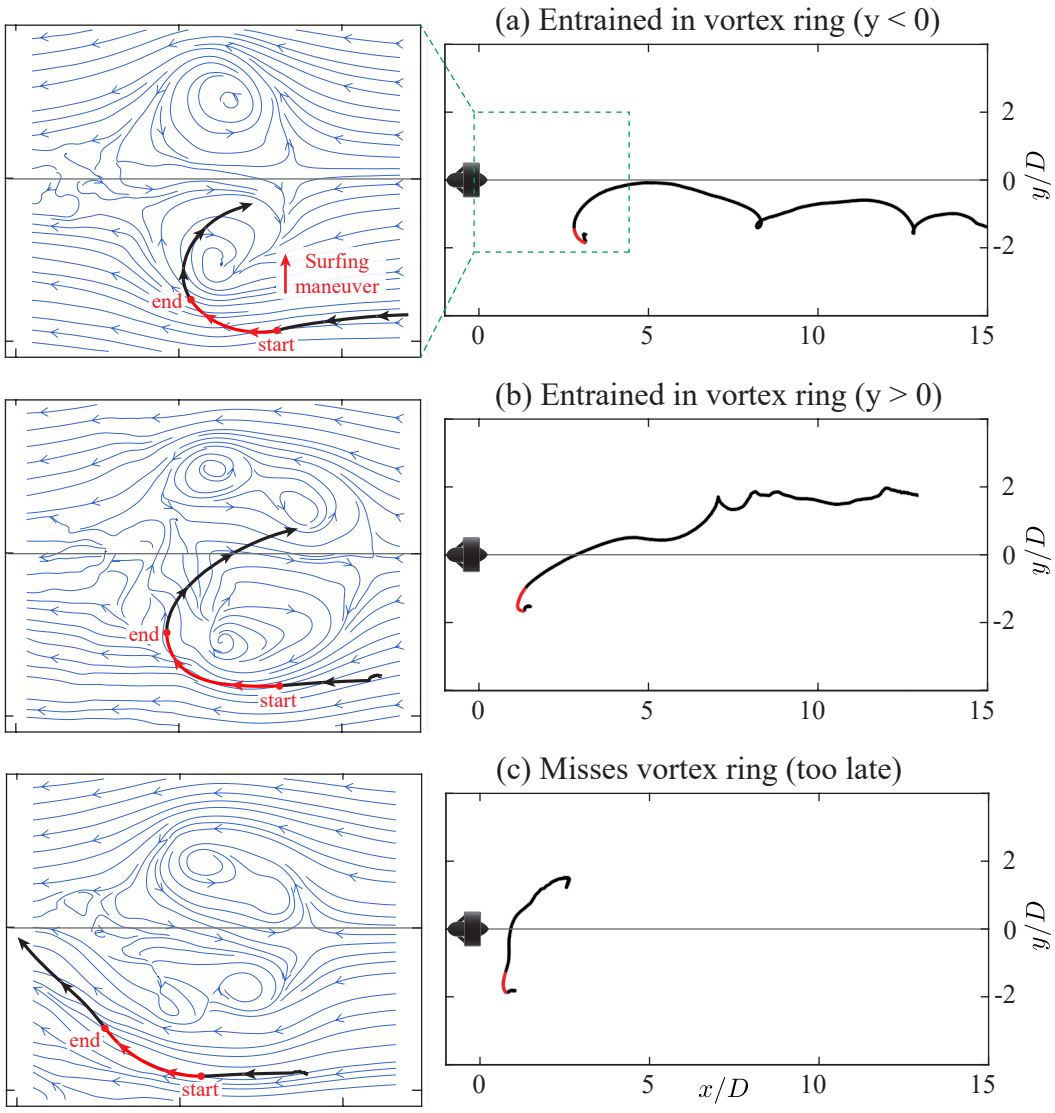


Figure 4.6: Example trajectories illustrating the dependence of initial entrainment on the starting position and time of the surfing maneuver. (**a**) Example trajectory in which CARL successfully surfs the vortex ring. Left: the trajectory of CARL is plotted against flow streamlines in the vortex ring reference frame, which shows the initial entrainment. The portion of the trajectory in which CARL swims towards the vortex ring is highlighted in red. Right: the full trajectory is plotted in the lab frame, showing that CARL is entrained into the bottom half of the vortex ring. (**b**) Example trajectory in which CARL is entrained into the top half of the vortex ring due to the initial starting position being closer to the centerline of the vortex ring. (**c**) Example trajectory in which CARL misses the vortex ring due to initiating the surfing maneuver too late.

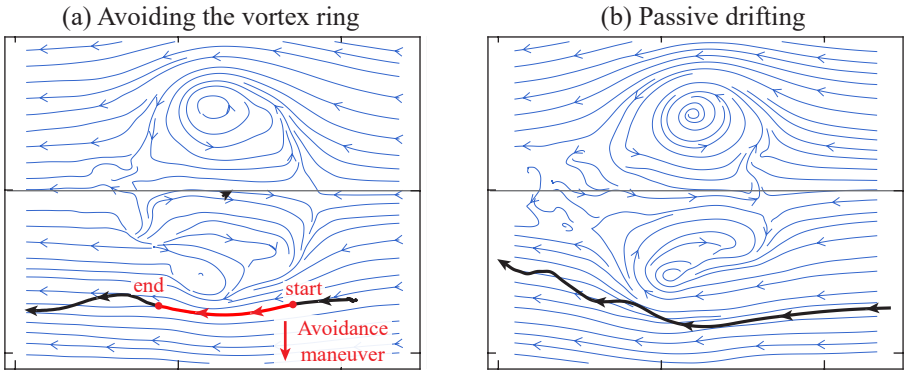


Figure 4.7: Example trajectories of CARL avoiding entrainment into the vortex ring either through an active avoidance maneuver (**a**) or passively drifting around the vortex ring (**b**). Trajectories are plotted in the vortex ring reference frame against streamlines in the vortex ring reference frame. Small oscillation in the trajectory of CARL is apparent in panel (**b**) due to CARL wobbling after the vortex ring passes by.

For completeness, we also plot two example trajectories in which CARL was commanded to avoid the vortex ring (Figure 4.7a) and one in which CARL drifted passively (Figure 4.3b). In both cases, CARL begins and ends outside of the boundary of the vortex ring.

4.8 Analysis of acceleration as a sensory input

As demonstrated in the previous section, the final trajectory of CARL is sensitive to the timing and location of the surfing maneuver. For the experiments in this study, the starting time of the maneuver is triggered by the y-component of the acceleration as measured by the IMU onboard CARL (see Figure 4.2). Therefore, in this section, we seek to model and understand how the acceleration is used as a signal to detect the presence and location of a passing vortex ring.

To understand the acceleration of CARL as a vortex ring passes nearby, we first examine the acceleration of idealized passive tracer particles using PIV measurements of the vortex rings. While CARL is by no means an ideal tracer particle due to having finite size and inertia, the acceleration of the background flow provides a simple model of the acceleration induced by the passing vortex ring.

In Figure 4.8a, we plot $\mathrm{D}v/\mathrm{D}t$ computed from PIV measurements, which represents the y-component of acceleration experienced by ideal tracer particles in the flow. An ellipse is overlaid to indicate the approximate boundary of the vortex ring, since this section considers only the signals initially sensed by CARL before deciding to

maneuver, i.e., before entrainment. Streamlines in the vortex ring reference frame are also plotted to show the direction of flow. Additionally, contours of +0.5 $\mathrm{m\,s^{-2}}$ and -0.5 $\mathrm{m\,s^{-2}}$ of $\mathrm{D}v/\mathrm{D}t$ are included for direct comparison with the acceleration threshold used to trigger the surfing maneuver.

Upstream of the vortex ring, the streamlines bend away from the front edge of the vortex ring boundary. As a result, upstream tracer particles initially experience an acceleration away from the centerline as the vortex ring approaches. For example, in Figure 4.8a, particles near region (1) experience a negative y-acceleration. Around the top and bottom edges of the vortex ring, the streamlines curve the opposite direction, resulting in a positive y-acceleration in region (2) that points towards the vortex ring. Finally, near the rear of the vortex ring, the streamlines once again curve away from the vortex ring, and the y-acceleration is negative in region (3).

For comparison, the y-acceleration recorded by CARL is plotted below in Figure 4.8c during an episode in which CARL passively drifted as a vortex ring passed nearby (e.g., Figure 4.3b). The recorded acceleration exhibits features that correspond to regions of the PIV data. As the vortex ring initially approaches, CARL records a small, negative y-acceleration. This corresponds to region (1) in the PIV data, in which streamlines bend away from the front of the vortex ring. Next, CARL experiences a large positive y-acceleration, which corresponds with region (2) in the flow, as the trajectory of CARL curves around the vortex ring boundary. Finally, the acceleration becomes negative again, which corresponds with region (3). It may be that the acceleration sensed by CARL corresponds the curvature of the streamlines bending around the vortex ring boundary.

To model the acceleration of CARL analytically, we consider Hill's spherical vortex model (Hill, 1894) and plot the resulting flow acceleration in Figure 4.8b. The radius of the model vortex ring was set equal to the average of the estimated semi-major and semi-minor axes of the physical vortex ring.

The regions of positive and negative acceleration plotted for the model vortex ring match qualitatively the PIV data. For example, the contour of 0.5 $\mathrm{ms^{-2}}$ acceleration encompassing region (2) in panel (a) also appears in the modeled vortex ring. Even though the shape of the physical vortex ring is not circular, the overall trend of the streamlines curving around the bubble of the vortex ring is the same.

In panels (d) and (f), the x and y-acceleration are plotted for an ideal tracer particle moving through the modeled flow. The trend of the particle acceleration in (d)

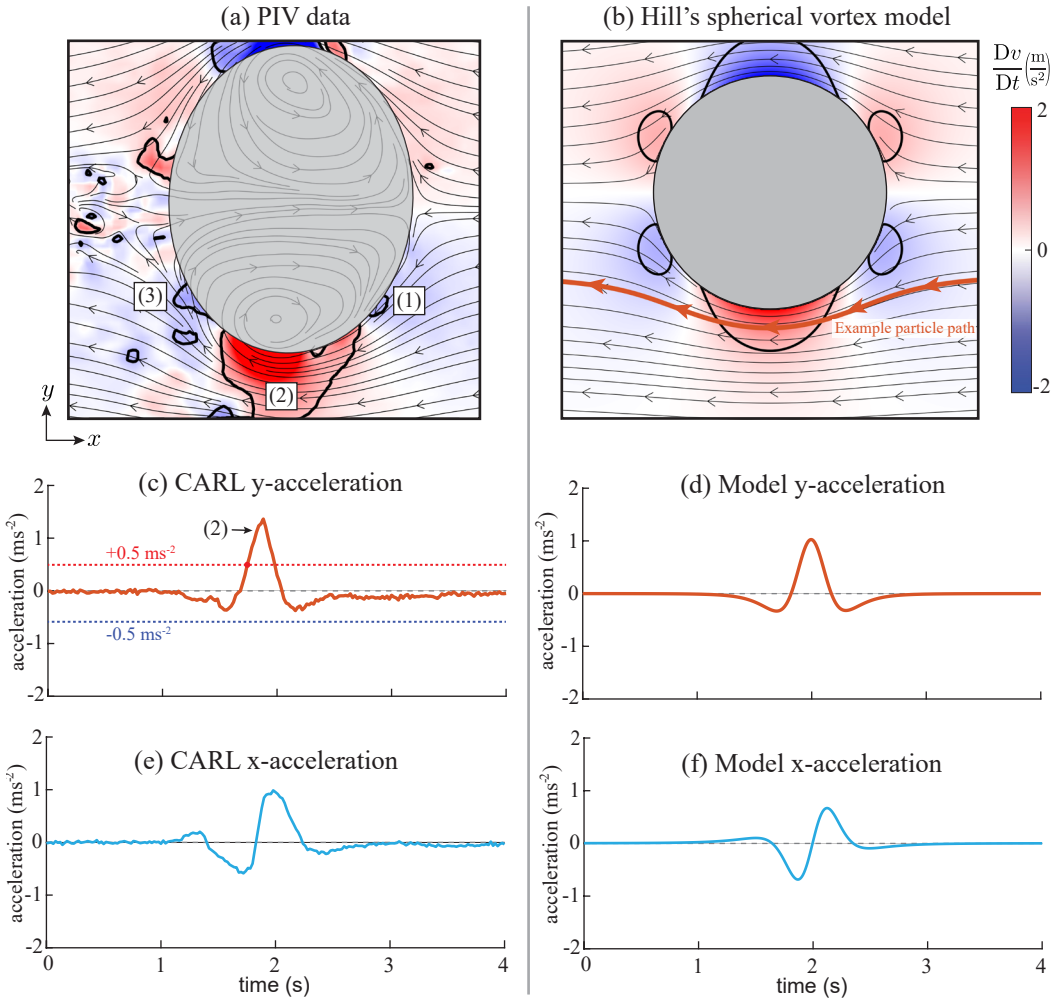


Figure 4.8: Comparison between the measured and modeled acceleration signal from a passing vortex ring. (**a**) The y-component of the material derivative is plotted using PIV data of a vortex ring, indicating the acceleration experienced by ideal fluid tracers. For example, particles in regions (1) and (3) experience negative y-acceleration, while particles in region (2) experience larger and positive y-acceleration, potentially explaining the signals sensed by CARL. Contours of +0.5 and -0.5 ms^{-2} are also plotted, which correspond to threshold used by CARL to execute the surfing maneuver. An ellipse is overlaid to indicate the approximate boundary of the vortex ring. (**c**) Acceleration in the y-direction sensed by CARL as a vortex ring passes nearby. The spike in positive y-acceleration may correspond to region (2) in the panel above, and is used to locate the vortex ring during a surfing maneuver. (**b**) Hill's spherical vortex model. The regions of positive and negative acceleration agree with the PIV data qualitatively. Both components of the acceleration of a ideal tracer particle traveling through the potential flow model (**d**,**f**) qualitatively agree with the signal sensed by the IMU onboard CARL (**c**,**e**).

and (f) matches the acceleration recorded by CARL in (c) and (e) qualitatively. In summary, it appears that the acceleration experienced by CARL can be modeled by considering the flow acceleration around the boundary of the vortex ring.

As a result, the vortex ring detection rates shown in Figure 4.3 can now be understood. When CARL was commanded to execute the surfing policy, CARL correctly identified the location of the vortex rings in 81% of episodes, which involved swimming in the direction of sensed acceleration above the 0.5 $\mathrm{m\,s^{-2}}$ threshold. This corresponds to sensing the large acceleration in region (2) of Figure 4.8a. In this region, the y acceleration points towards the vortex ring, and therefore the surfing maneuver is also directed towards and into the vortex ring.

In 14% of episodes, CARL swam in the wrong direction (i.e., away from the vortex ring), due to sensing the negative y acceleration in region (1). The smaller size of this negative y-acceleration region explains why the correct detections are more likely: with random variations in the starting position, CARL will encounter the larger region (2) more often than the smaller region (1). Additionally, in 5% of episodes, the y-acceleration did not exceed the threshold value, and CARL did not execute a surfing maneuver. These cases may correspond to being outside of either region (1) or (2). In this way, the contours of acceleration may explain and potentially predict the dependence of the sensed acceleration on the position of CARL relative to the vortex ring.

Since the detection of the vortex ring depends on the flow outside of the vortex ring boundary, the particular vortex ring model may not be of great importance. For a vortex ring propagating in quiescent flow, the flow outside of the vortex ring is free of vorticity, with the exception of a small trailing wake. Therefore, the flow upstream and outside of the vortex ring can be modeled as potential flow, and indeed, the flow outside Hill's spherical vortex ring is identical to the potential flow past a sphere. The trends in acceleration and the curvature of streamlines would be analogous for various vortex ring models, or for that matter, the potential flow around a round object.

Consequently, acceleration sensing in this context can considered a form of pressure sensing. In the potential flow outside of the vortex ring boundary, the flow acceleration is balanced by the pressure gradient:

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\frac{1}{\rho}\nabla P. \tag{4.12}$$

Therefore, the acceleration experienced by an idealized tracer particle is directly proportional to the pressure gradient in the background flow. For an object with finite size such as CARL and in the absence of viscous forces, body acceleration can be attributed to the pressure field integrated around the body of the object. In this sense, acceleration sensing has a direct correspondence with pressure sensing.

An advantage of modeling the flow exterior to the vortex ring with potential flow is that the pressure gradient, and therefore acceleration field, can be computed analytically. Scaling laws can then be developed in order to better generalize the surfing strategy, for example, by determining the required sensitivity of the onboard accelerometer for a vortex ring of arbitrary size, speed, and relative distance. To demonstrate, we compute the pressure field exterior to Hill's spherical vortex analytically using the velocity field and the Bernoulli equation:

$$\frac{u_r}{U_\infty} = \left(1 - \left(\frac{R}{r}\right)^3\right)\cos(\theta), \quad \frac{u_\theta}{U_\infty} = -\left(1 + \frac{1}{2}\left(\frac{R}{r}\right)^3\right)\sin(\theta), \quad r \geq R, \tag{4.13}$$

$$\frac{P - P_\infty}{\frac{1}{2}\rho U_\infty^2} = 1 - \frac{|\mathbf{u}|^2}{U_\infty^2} = -\left(\frac{R}{r}\right)^3\left(1 - 3\cos^2(\theta)\right) - \frac{1}{4}\left(\frac{R}{r}\right)^6\left(1 + 3\cos^2(\theta)\right), \tag{4.14}$$

where P is the static pressure, P_∞ is free-stream pressure, U_∞ is the propagation speed of the spherical vortex, r is the distance from the center of the vortex ring, R is the radius of the spherical vortex, θ is the polar angle in spherical coordinates, and u_r and u_θ are the radial and polar components of velocity, respectively. From this pressure field, two quantities of interested can be computed.

First, we compute the maximum flow acceleration, which predicts the largest acceleration signal that could be detected by the robot within this idealized model. The maximum flow acceleration occurs on the sides of the vortex ring ($r = R, \theta = \pm 90°$), which corresponds to the region of high acceleration in region (2) in Figure 4.8. Taking the gradient of Equation 4.14:

$$\frac{1}{\rho}|\nabla P|_{\max} = \left|\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\right|_{\max} = \frac{9}{4}\frac{U_\infty^2}{R}. \tag{4.15}$$

For a vortex ring with a given radius R and propagation speed U_∞, this equation provides a prediction of the maximum flow acceleration. Intuitively, this equation

takes the form of a centripetal acceleration, since the flow acceleration around the vortex ring is related to the streamline curvature. The maximum acceleration can also be computed in terms of the circulation of the vortex ring. For Hill's spherical vortex model, the circulation is related to the propagation speed of the vortex ring by $\Gamma = 5U_\infty R$. Therefore, the max acceleration can be expressed as:

$$\left|\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\right|_{\max} = \frac{9}{100}\frac{\Gamma^2}{R^3}. \tag{4.16}$$

These expressions for the maximum acceleration could be useful for designing the sensing capabilities of robots. For example, a vortex ring that too large or too slow may not be detectable for a given sensitivity to acceleration.

Another useful scaling to consider is the dependence of flow acceleration on distance to the vortex ring. For example, if a robot is too far from a vortex ring, the acceleration signal may be too weak to detect with onboard accelerometers. To investigate this scaling, we consider the pressure gradient when the robot is far from the vortex ring ($r \gg R$):

$$\frac{1}{\rho}\left|\nabla P\right| = \left|\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\right| \approx \frac{3}{2}\frac{U_\infty^2}{R}\left(\frac{R}{r}\right)^4 \sqrt{1 - 2\cos^2(\theta) + 5\cos^4(\theta)}. \tag{4.17}$$

Therefore, the magnitude of the flow acceleration scales with distance to the vortex ring according to:

$$\left|\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\right| \Big/ \left|\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\right|_{\max} \sim \left(\frac{R}{r}\right)^4, \quad (r \gg R). \tag{4.18}$$

In other words, the acceleration signal decays proportional to r^{-4}. The required sensitivity of an onboard accelerometer therefore increases correspondingly with distance from the vortex ring.

To be sure, interpreting body acceleration as a form of pressure sensing requires several caveats. First, the inertia of CARL results in a lag between the motion of CARL and the background flow, effectively low pass filtering temporal variations in the pressure. To address this limitation, the robot's acceleration could be estimated using models of small particles with inertia (e.g., Maxey and Riley, 1983) or by empirically modeling the inertia and drag forces on CARL. Second, the acceleration of CARL represents the pressure integrated over the body of CARL, effectively

averaging out any pressure fluctuations of a scale significantly less than the size of CARL. To account for the finite size of CARL, the pressure field could be integrated over a circular or spherical body that approximates the shape of CARL. However, even with these caveats, the results in Figure 4.8 suggest that a potential flow model and ideal tracer particles capture the dominant mechanisms behind the acceleration signal sensed by CARL.

As an additional note, the use of body acceleration as a sensory input is complicated by changes in the orientation of the robot. For example, any roll or pitch of CARL results in a component of the gravity vector, which points nominally in the z-direction, contributing to the acceleration in the x or y-directions in the body frame of reference. In Appendix C.3, we discuss corrections due to rotation of the body frame of CARL, and verify the onboard acceleration measurements using position data from the overhead cameras.

4.9 Body rotation and vorticity sensing

In addition to acceleration, another sensor input that could potentially aid in recognizing the vortex ring is body rotation due to vorticity in the background flow, sensed via gyroscopes in the onboard IMU. For example, Reddy et al. (2018) used the flow-induced rolling moment of a glider to sense shear caused by thermal plumes in the atmosphere. In our experiments, CARL prevented body rotation about the vertical (z) axis using a PID feedback loop, which took the angular velocity from the IMU as an input and outputted a rotational control signal to the thrusters. The magnitude and direction of this rotational control signal, which we denote as τ_{control}, was therefore correlated to the torque applied to CARL by the surrounding fluid. In this section, we investigate the extent to which τ_{control} could be used to indirectly sense vorticity in the background flow.

In Figure 4.9a, we plot the time-averaged vorticity ($\overline{\omega} = 1/T \int_0^T \omega \mathrm{d}t$) computed from PIV data of eight consecutive vortex rings ($T = 96\,\mathrm{s} \approx 937 D/\overline{U}_p$). Because the vortex rings propagate across the tank in the x direction, the time-averaged vorticity is positive (counter-clockwise) above the y-axis, and negative (clockwise) below the y-axis. Since CARL blocks a portion of the laser sheet while entrained in the vortex ring, this PIV data was taken without the presence of CARL. However, the analysis in Section 4.5 suggests that it is reasonable to assume that the flow of the vortex ring is largely unaffected by the entrainment of CARL. For comparison, trajectories from 109 episodes of CARL surfing vortex rings are plotted in Figure 4.9b, with

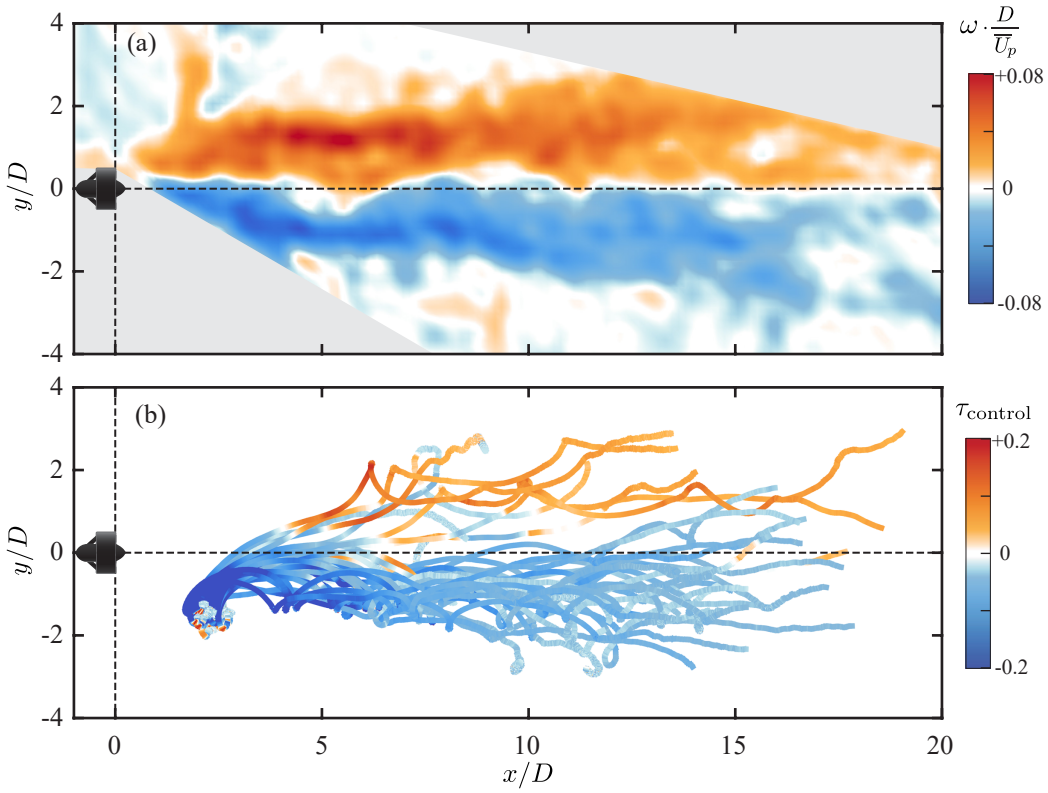


Figure 4.9: Correspondence between the time-averaged vorticity and the rotational control applied to prevent CARL from rotating. (**a**) Time-averaged vorticity field computed from eight consecutive vortex rings without the presence of CARL. (**b**) Trajectories in which CARL surfed the vortex ring, shaded by the rotational control signal (τ_{control}) with the sign reversed for comparison with the vorticity in (**a**). The sign of the rotational control signal corresponds with the direction of background vorticity, suggesting that body rotation could be used as an additional signal for detecting background flow structures.

the trajectories shaded by the rotational control signal commanded by CARL. The sign of the control signal is flipped for more direct comparison with the background vorticity, since the control signal opposes the rotation induced by the background flow.

The sign of the rotational control signal appears to match the background vorticity, typically being negative below the y-axis, and positive above the y-axis. This suggests that the body rotation of CARL could be used to detect the presence of the vortex ring, and the direction of the background vorticity. The rotational control signal may therefore be useful as an additional flow measurement technique.

To see if τ_{control} coincides with the background vorticity quantitatively, the control

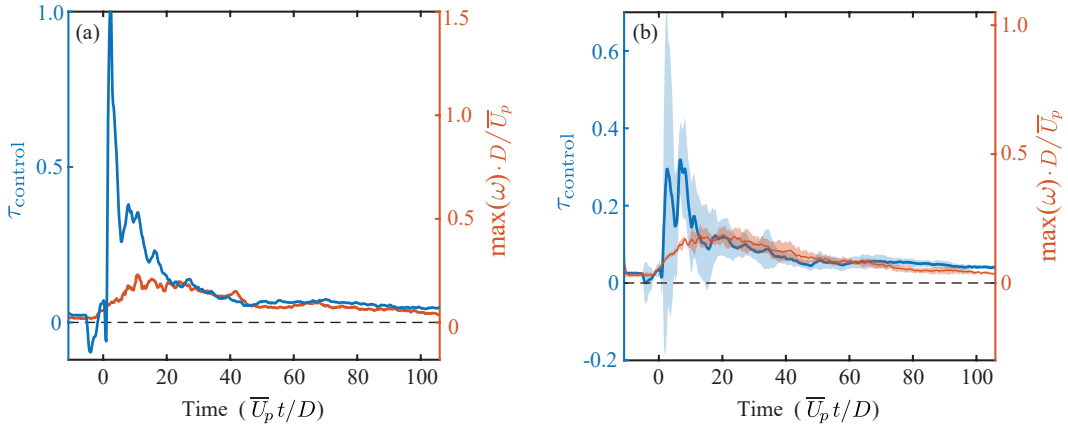


Figure 4.10: Time history of τ_{control} and the maximum vorticity measured by simultaneous PIV for one example trajectory (**a**) and the averaged from an ensemble of five trajectories (**b**). Both plots have the same relative scaling of 1.5 between the left and right axes. Initially, there is a large discrepancy, but after the initial entrainment into the vortex ring, the magnitude of the control signal and maximum vorticity are in agreement and both decrease over time as the vortex ring reduces speed.

signal and maximum background vorticity are plotted as functions of time in Figure 4.10. Here, the maximum vorticity is computed using trials in which simultaneous PIV is available. The maximum vorticity is taken from the unobstructed side of the vortex (i.e., the half of the vortex ring not containing CARL).

Initially, the rotational control signal does not correspond well with the maximum vorticity. However, after a short period of time corresponding with the initial entrainment into the vortex ring, both τ_{control} and $\max(\omega_z)$ are in agreement, slowly decreasing with time as the vortex ring propagates and slows down. Therefore, in addition to showing the sign of the background vorticity, it may be possible to use τ_{control} to reconstruct the background vorticity quantitatively.

The discrepancy at early times could be caused by several factors. First, CARL likely exerts a large rotational control effort during the impulsive maneuver, since the thrusters on opposite sides of CARL typically produce imbalanced forward thrust. Second, the measured vorticity has sources of error at this early time because the thruster casts a shadow in the region of CARL's entrainment, and the resolution of the wide-angle camera struggles to resolve the vortex when it is small in size. Third, there may be vorticity generated as CARL interacts with the vortex ring during entrainment. Future work could incorporate the effects of torque induced by unequal thrust in the motors on CARL, as well as more detailed fluid-robot interactions.

4.10 Conclusions and future work

In this study, we implemented energy-efficient flow-based navigation in a physical tank and robot. Using an onboard accelerometer, our robot successfully identified and maneuvered onto vortex rings, exploiting the background fluid flow for a near fivefold reduction in energy required for forward propulsion. These results demonstrate the potential of accomplishing flow sensing using an inexpensive accelerometer (~$5), which can be found in countless drones, phones, and other consumer devices. Additionally, simultaneous PIV measurements of the background flow and tools such as the FTLE field give insight into the fluid mechanic principles underlying the surfing strategy and entrainment of CARL. In summary, this study serves as a proof-of-concept demonstration of the potential of onboard inertial measurements to enable intelligent interaction with background fluid flows.

Both tasks of sensing and maneuvering could be explored further in several directions. First, an immediate extension of the current experimental setup would be to utilize both components of acceleration as well as rotation to detect the vortex ring with greater reliability. Additionally, the models of sensed acceleration in Section 4.8 could be tested systematically by more greatly varying the starting position of CARL relative to the generated vortex rings, or by varying the initial depth of CARL to test sensing in three dimensions. It may also be possible to complement onboard inertial sensing with direct flow measurements from flow sensors (e.g. pressure sensors) to resolve the higher-frequency temporal and spatial scales of the background flow that are filtered by the inertia and finite size of the robot. Connecting body acceleration to the pressure gradient in the background flow could also facilitate comparisons with existing flow sensing literature. For example, studies have investigated the ability of pressure sensors and aquatic animals to locate oscillating spheres (e.g., Dehnhardt, Mauck, and Bleckmann, 1998; Dagamseh et al., 2013), which is an unsteady version of the external flow in the spherical vortex model in Section 4.8.

In addition to exploring the sensing problem, optimal maneuvering could be investigated. For swimming into the vortex ring, there may be trajectories more efficient than the fixed-speed impulsive surfing maneuver used in this study, which would further increase the energy advantage of surfing the vortex rings. Efficient trajectories could be estimated in advance using path planning algorithms applied to the vortex ring flow (Lolla et al., 2014) or learned using data-driven methods such as RL (e.g., Chapters 2 and 3). Combining these tools of optimal maneuvering with

the analytical model of the sensed acceleration could fully close the loop between flow-based sensing and navigation.

In addition to optimizing the initial stage of jumping onto the vortex ring, there is an opportunity to investigate path planning at the scale of the entire tank or fluid domain. For example, if CARL were tasked with navigating to a particular location rather than just downstream, it may be optimal to surf the vortex ring for a short period of time, or even avoid it entirely, depending on the direction of the incoming vortex ring relative to the location of the target. Such optimizations could be performed over the Pareto front that defines tradeoffs between energy consumption and minimal travel time for point-to-point navigation (Krishna, Song, and Brunton, 2022).

It is also important to generalize these results to other types of flows. A simple extension to this work would be to investigate the coupling that may occur when a generated vortex ring is equal or smaller in size than the robot. Additionally, it would be useful to test navigation strategies in other flows that involve different arrangements of vortical structures, such as a von Kármán vortex street, a turbulent wake, or double gyre flow. Of particular practical application are flows found in ocean environments. While vortex rings are found throughout nature in biological propulsion and other processes, exploitable currents in the ocean may be more turbulent and chaotic. However, vortical structures are often present within those chaotic flows. By recognizing individual vortices via inertial measurements or flow sensing, it may be possible to optimize over short time horizons and plan energy-efficient paths.

In addition to underwater environments, this work could be extended to aerial applications. Uncrewed aerial vehicles such as quadcopters and drones are increasingly tasked with flying in chaotic flow environments, in which gusts and vortical structures can be larger in scale than the robot (Watkins et al., 2020). In such cases, optimizing the interaction between vortical structures may be useful for reducing energy consumption or maintaining stable flight (Renn and Gharib, 2022). It is also worth noting that many quadcopters use the identical accelerometer that is mounted onboard CARL (MPU-6050), and so quadcopters may already be equipped with the necessary sensor data in order to exploit the benefits of flow-based navigation.

Chapter 5

CONCLUSIONS

In this thesis, onboard flow sensing was investigated as a means to enhance the capabilities of autonomous underwater robots through efficient navigation and targeted sampling. The results are enabled through a multi-disciplinary approach combining fluid mechanics, machine learning algorithms, and CARL, a scratch-built autonomous robotic platform. By demonstrating methods for improved autonomy in both simulated and physical environments, it is my hope that this work can contribute to tools that will improve our collective knowledge of the ocean and enable responsible stewardship of the future of our planet.

5.1 Summary of contributions

The studies in this thesis aimed to address the dual problem of flow-based sensing and navigation using new tools of machine learning and robotics combined with the fundamental fluid mechanics of the background flow. Chapter 2 focused on the problem of efficient navigation using background flows, and demonstrated the potential of deep RL to assimilate flow measurements for navigation in a simulated sensing. Chapter 3 demonstrate how strategies learned in silico can be transferred to a physical setting in generalizable manner that takes into account the sensor noise and turbulence experienced by a real robotic swimmer. Lastly, Chapter 4 synthesized the dual problem of sensing and navigation, demonstrating experimentally how onboard sensors and autonomous maneuvers can effectively exploit a background flow for energy efficient propulsion.

Specifically, Chapter 2 demonstrated that point measurements of flow quantities such as velocity are sufficient to accomplish efficient point-to-point navigation in simulated cylinder flow. RL was shown to be an effective algorithm for discovering such flow-based navigation strategies without human fine-tuning. Additionally, multiple sensor inputs were investigated, and velocity sensing outperformed the bio-inspired vorticity sensing approach. Using only an instantaneous velocity measurement, the virtual swimmer achieved a near 100% success rate in reaching the target locations while approaching the time-efficiency of optimal navigation trajectories, even with a swimming speed less than the free-stream flow speed.

In Chapter 3, RL and flow-based navigation were investigated using a physical robot and flow environment. Pressure sensors were tested as a low-cost, miniature flow sensing methodology that was effective for tracking turbulent plumes in a physical tank. By varying the spacing of flow sensors, the limits of the navigation approach were modeled in terms of the noise of the sensors, turbulence in the background flow, and strength of the mean flow gradient. Another contribution of Chapter 3 is the use of interpretability metrics to understand the gradient-seeking principle of the policy learned in simulation, which enabled transfer to the physical robot.

In Chapter 4, onboard inertial measurements were demonstrated as an effective technique for sensing the presence and location of passing vortex rings, which enabled a novel and energy-efficient surfing strategy. Many existing robots in the air and underwater are equipped with onboard accelerometers for stabilization and inertial guidance, suggesting the potential for sensing background flows without the need for additional flow-sensing hardware. Using vortex rings as a repeatable flow unit enabled the analysis of energy and momentum transfer between the vortex ring and the robot, which could provide insight into the energetics of efficient navigation in other flow environments. Additionally, body rotation was shown to be a potential candidate for sensing background flow vorticity. Lastly, an analysis of LCS provided a connection between the small surfing maneuver and the resulting large propulsive benefit, which could aid with the discovery of navigation strategies across various types of fluid flows.

Another important contribution of this thesis is CARL, a low-cost autonomous underwater platform designed for investigating sensing and navigation strategies in an experimental setting. In order to ensure reproducibility, and to disseminate the lessons and techniques learned over years of designing CARL, the full design files, code, and bill of materials are available at https://doi.org/10.22002/b291n-nx970. Details about the construction of CARL are shown in Appendix B. There have been seven major revisions to the design of CARL, and certainly there are many opportunities for further improvement or alternative approaches to accomplish the same goals. However, it is my hope that this design could aid other researchers in investigating flow-based navigation in an experimental setting.

5.2 Opportunities for further exploration

The results in this thesis have demonstrated the potential for onboard flow sensing to enhance robotic exploration of underwater environments. However, there are a number of opportunities for further exploration, which are detailed in this section.

Incorporating a time history of sensor measurements

In all experiments in this thesis, instantaneous flow measurements were sufficient to accomplish proof-of-concept demonstrations, for example, one time step of pressure measurements in Chapter 3 and the instantaneous acceleration in Chapter 4. However, to realize the full potential of flow-based navigation, a time-history of sensor measurements may provide a significant improvement. For example, the inclusion of a time-history has consistently resulted in increased performance for tasks involving machine learning and fluid mechanics, such as mitigating the effects of wind gusts on the lift of a wing (Renn and Gharib, 2022), locating the source of turbulent odor plumes (Singh et al., 2023), and learning fish schooling strategies (Verma, Novati, and Koumoutsakos, 2018). A potential explanation for the improvement in performance may be Takens' embedding theorem (Takens, 1981), which states that an under-observed chaotic dynamical system can be reconstructed with a sufficiently long time-history of observations. For example, in the case of onboard flow-sensing, the surrounding fluid system has an extremely large number of effective degrees of freedom, but may only be observed by a handful of sensors. To implement a time-history of measurements in an RL algorithm, researchers often use specific network architectures, such as transformers or long short-term memory (LSTM) networks (e.g. Verma, Novati, and Koumoutsakos, 2018), which explicitly account for the sequential order of measurements. Another approach could be to compute a real time spectrum of the sensor measurements, which could provide a rich set of information in turbulent flows such as the turbulent plumes encountered in Chapter 3. Additionally, system identification techniques such as Eigenmode Realization or Koopman operator theory could potentially be used to identify models of the system dynamics (Brunton et al., 2022), for example, enabling reduced order models of the trajectory of CARL during encounters with the vortex rings in Chapter 4.

Distributed flow sensing

In addition to sampling a time-history of measurements, the accuracy and spatial resolution of sensors could be increased. In Chapter 3, CARL used pressure mea-

surements at four locations. Using more sensors may enable measurement of a wider range of spatial scales and could be relevant to the fish-sensing literature, as fish often have flow sensing arrayed across the entire lateral line (Verma, Papadimitriou, et al., 2020). Additionally, novel flow sensors could increase the sensitivity of the strategies used in this thesis, for example, using micro-pillar arrays instead of pressure sensors to detect turbulent flow in Chapter 3. Another approach is to fully embrace the promise of machine learning algorithms to automatically assimilate information from a variety of sources without human fine-tuning. Perhaps that could be beneficial for an underwater robot equipped with an array of many types of flow sensors. However, on the other extreme, it is imperative that AUVs minimize cost and complexity to enable cost-scalability for comprehensive ocean exploration (Xu and John Dabiri, 2022). Therefore, determining the minimum set of sensor measurements needed for effective navigation is a potential future direction. For all experiments in this dissertation, low-cost sensors such as pressure sensors (~$16) and accelerometers (~$5) were sufficient for proof-of-concept demonstrations, and may be similarly effective in field tests in oceanic environments.

Extensions to other flow environments

In this thesis, four types of fluid flows were investigated: simulated cylinder flow, simulated double gyre flow, turbulent jet flow, and isolated vortex rings. For increased generalization and deployment in oceanic applications, it may be useful to consider other flow environments. An immediate extension using the same experimental facilities is to use multiple thrusters to generate flows in the tank, which could include launching multiple vortex rings from several directions, or combining vortex rings with turbulent plumes, and tasking a robot with distinguishing between the two. As another example, the flow generated by a flapping fin could provide a biologically relevant navigation problem for fish schooling or predator–prey interactions (e.g., Hang et al., 2023). Additionally, it may be useful to consider flows based directly on measurements of the ocean, such as large-scale eddies measured from satellite data.

It is also worth noting that researchers have begun developing an open-source library of fluid flows and problems for collaboration on data-driven modeling and control (github.com/dynamicslab/hydrogym). Investigating flow-based navigation within a standard library of fluid flows could enhance future collaboration and generalization of results. Additionally, techniques such as LCS could be used as a tool to link together results from different flows (for example, linking navigation in a vortex ring

with double gyre flow in Chapter 4.)

Another important point of consideration is that oceanic environments are typically stratified, which was not directly addressed by the experiments in this thesis. Stratification can, for example, affect the spreading behavior of turbulent wakes and plumes, as well as the dynamics of vortex rings and other vortical structures (e.g., Su, Wilhelmus, and Zenit, 2023).

Testing in other flows has the potential to encourage discovery of new navigation strategies. For example, the discovery of the vortex ring surfing strategy in Chapter 4 was accidental. While testing whether CARL could sense vortex rings, it was observed that under the right conditions, CARL would be carried across the tank by the vortex ring. This inspired the idea to use onboard measurements to actively surf on the vortex rings for propulsion. Exposing a robotic platform such as CARL to a variety of other flow fields may similarly reveal unexpected navigation strategies. Lastly, in order to map an unknown flow environment, it may prove useful to optimize for the combined objective of efficient navigation and maximum observability of the background fluid flow (Mei, Kutz, and Brunton, 2023).

Extensions to aerial vehicles

The primary motivation of this thesis is addressing the challenges of autonomous underwater navigation. In underwater environments, robots are often under-actuated while swimming against surrounding ocean currents, and must contend with flow disturbances that are significantly larger than the vehicle. Aerial vehicles are increasingly facing similar challenges, being tasked to navigate through highly turbulent flow environments such as urban environments (e.g., O'Connell et al., 2022; Renn and Gharib, 2022). For example, the flow from a single building may produce structures larger in size than the vehicle (Scott Watkins, Ravi, and Loxton, 2009). Experimental campaigns could be conducted in air that mirror the experiments in this thesis. For example, a small drone could be exposed to incoming vortex rings, which could directly connect with the results in Chapter 4.

Onboard training of machine learning algorithms

In the experiments in this thesis, the RL algorithms were all trained using simulated data on a desktop computer. A natural extension, which was not implemented in Chapter 3 due to time constraints, is to still train on a desktop computer, but use data collected by the physical robot. Other robotic systems have been trained using physically recorded data, which can better optimize for the specific noise and

dynamics of the intended robot and environment (e.g., Renn and Gharib, 2022; Masmitja et al., 2023; Dulac-Arnold et al., 2021).

However, a further extension, is to additionally *train* onboard the robot. To be sure, adaptive control schemes have been studied and verified on aerial robots (O'Connell et al., 2022). But for deep neural networks, training onboard a robot is typically avoided, since a desktop computer can perform the task significantly more quickly than the low-power microprocessors found onboard typical robots.

Autonomous navigation in the ocean presents a unique opportunity for this kind of onboard learning. First, aerial vehicles demand fast timescales for decision making: a millisecond-scale control error could result in a crash for an aerial drone, whereas underwater vehicles can drift passively for hours, days, and even years without catastrophic failure. Therefore, it is feasible to perform the requisite forward and backward passes of neural networks onboard ocean vehicles at speeds significantly slower than would be required for aerial drones. Second, while networks trained on powerful computers can be easily transferred to terrestrial and aerial robots, communication is extremely limited in the ocean, and any neural networks would need to be pre-trained and communicated before deployment. Third, microprocessors are now powerful enough to accomplish onboard training at small enough size and power consumption to be relevant for ocean robots. For example, the Teensy 4.1 microcontroller used in CARL was tested to be able to perform several hundred forward and back-propagations of a standard 128 by 128 MLP network every second, using only one-half of a watt of power. Additionally, ocean robots often record data for months or years at a time, which could satisfy the large data requirements of RL algorithms.

Of course, a primary goal of this thesis is to learn strategies that generalize, and can be transferred to different situations by taking into account the relevant physics. However, by training directly onboard a robot, it may be possible to optimize for the specific scenario encountered by a robot, such as the particular currents in a region of the ocean, or the exact stratification and interaction between the robot and the surrounding flow. Additionally, new algorithms such as SINDy-RL (Zolman et al., 2024) may enable onboard learning with policies that are also interpretable.

BIBLIOGRAPHY

Akanyeti, Otar et al. (Mar. 2013). "Self-motion effects on hydrodynamic pressure sensing: part I. Forward–backward motion". In: *Bioinspiration & Biomimetics* 8.2. Publisher: IOP Publishing, p. 026001. DOI: 10.1088/1748-3182/8/2/026001.

Alsalman, Mohamad, Brendan Colvert, and Eva Kanso (Nov. 2018). "Training bioinspired sensors to classify flows". en. In: *Bioinspiration & Biomimetics* 14.1. Publisher: IOP Publishing, p. 016009. DOI: 10.1088/1748-3190/aaef1d.

Asadnia, Mohsen et al. (Oct. 2013). "Flexible and Surface-Mountable Piezoelectric Sensor Arrays for Underwater Sensing in Marine Vehicles". In: *IEEE Sensors Journal* 13.10, pp. 3918–3925. DOI: 10.1109/JSEN.2013.2259227.

Beddows, Patricia A. and Edward K. Mallon (Feb. 2018). "Cave Pearl Data Logger: A Flexible Arduino-Based Logging Platform for Long-Term Monitoring in Harsh Environments". In: *Sensors* 18.2. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 530. DOI: 10.3390/s18020530.

Bellemare, Marc G. et al. (Dec. 2020). "Autonomous navigation of stratospheric balloons using reinforcement learning". In: *Nature* 588.7836. Number: 7836 Publisher: Nature Publishing Group, pp. 77–82. DOI: 10.1038/s41586-020-2939-8.

Benway, Heather M. et al. (2019). "Ocean Time Series Observations of Changing Marine Ecosystems: An Era of Integration, Synthesis, and Societal Applications". In: *Frontiers in Marine Science* 6.

Berger, Sandrine et al. (2024). "Reliability assessment of off-policy deep reinforcement learning: A benchmark for aerodynamics". In: *Data-Centric Engineering* 5, e2. DOI: 10.1017/dce.2023.28.

Biferale, L. et al. (Oct. 2019). "Zermelo's problem: Optimal point-to-point navigation in 2D turbulent flows using reinforcement learning". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.10. Publisher: American Institute of Physics, p. 103138. DOI: 10.1063/1.5120370.

Bleckmann, Horst and Randy Zelick (2009). "Lateral line system of fish". In: *Integrative Zoology* 4.1. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1749-4877.2008.00131.x, pp. 13–25. DOI: 10.1111/j.1749-4877.2008.00131.x.

Bora, Meghali et al. (Jan. 2018). "Sensing the flow beneath the fins". In: *Bioinspiration & Biomimetics* 13.2. Publisher: IOP Publishing, p. 025002. DOI: 10.1088/1748-3190/aaa1c2.

Braitenberg, Valentino (Feb. 7, 1986). *Vehicles: Experiments in Synthetic Psychology*. Google-Books-ID: 7KkUAT_q_sQC. MIT Press. 174 pp.

Brunton, Steven L. et al. (May 2022). “Modern Koopman Theory for Dynamical Systems”. In: *SIAM Review* 64.2. Publisher: Society for Industrial & Applied Mathematics (SIAM), pp. 229–340. DOI: 10.1137/21m1401243.

Buzzicotti, Michele et al. (Feb. 2021). “Optimal control of point-to-point navigation in turbulent time-dependent flows using Reinforcement Learning”. In: *arXiv:2103.00329 [physics]*. arXiv: 2103.00329.

Caron, David A. et al. (2008). “Macro- to fine-scale spatial and temporal distributions and dynamics of phytoplankton and their environmental driving forces in a small montane lake in southern California, USA”. In: *Limnology and Oceanography* 53.5part2, pp. 2333–2349. DOI: https://doi.org/10.4319/lo.2008.53.5_part_2.2333.

Charette, Matthew and Walter Smith (June 2010). “The Volume of Earth’s Ocean”. In: *Oceanography* 23.2. Publisher: The Oceanography Society, pp. 112–114. DOI: 10.5670/oceanog.2010.51.

Colabrese, Simona et al. (Apr. 2017). “Flow Navigation by Smart Microswimmers via Reinforcement Learning”. In: *Physical Review Letters* 118.15. Publisher: American Physical Society, p. 158004. DOI: 10.1103/PhysRevLett.118.158004.

Colvert, Brendan, Kevin Chen, and Eva Kanso (May 10, 2017). “Local flow characterization using bioinspired sensory information”. In: *Journal of Fluid Mechanics* 818, pp. 366–381. DOI: 10.1017/jfm.2017.137.

Colvert, Brendan, Geng Liu, et al. (Aug. 1, 2020). “Flowtaxis in the wakes of oscillating airfoils”. In: *Theoretical and Computational Fluid Dynamics* 34.4, pp. 545–556. DOI: 10.1007/s00162-020-00546-8.

Coombs, Sheryl et al., eds. (2014). *The Lateral Line System*. Vol. 48. Springer Handbook of Auditory Research. New York, NY: Springer New York. DOI: 10.1007/978-1-4614-8851-4.

Copley, Jon (Oct. 9, 2014). *Just how little do we know about the ocean floor?* The Conversation.

Dagamseh, Ahmad et al. (June 6, 2013). “Imaging dipole flow sources using an artificial lateral-line system made of biomimetic hair flow sensors”. In: *Journal of The Royal Society Interface* 10.83. Publisher: Royal Society, p. 20130162. DOI: 10.1098/rsif.2013.0162.

Dehnhardt, Guido, Björn Mauck, and Horst Bleckmann (July 1998). “Seal whiskers detect water movements”. In: *Nature* 394.6690. Number: 6690 Publisher: Nature Publishing Group, pp. 235–236. DOI: 10.1038/28303.

Dehnhardt, Guido, Björn Mauck, Wolf Hanke, et al. (July 6, 2001). “Hydrodynamic Trail-Following in Harbor Seals (Phoca vitulina)”. In: *Science* 293.5527. Publisher: American Association for the Advancement of Science, pp. 102–104. DOI: 10.1126/science.1060514.

DeVries, Levi et al. (Mar. 2015). "Distributed flow estimation and closed-loop control of an underwater vehicle with a multi-modal artificial lateral line". In: *Bioinspiration & Biomimetics* 10.2. Publisher: IOP Publishing, p. 025002. DOI: 10.1088/1748-3190/10/2/025002.

DeVries, Tim et al. (June 11, 2019). "Decadal trends in the ocean carbon sink". In: *Proceedings of the National Academy of Sciences* 116.24. Publisher: Proceedings of the National Academy of Sciences, pp. 11646–11651. DOI: 10.1073/pnas.1900371116.

Dulac-Arnold, Gabriel et al. (Sept. 1, 2021). "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis". In: *Machine Learning* 110.9, pp. 2419–2468. DOI: 10.1007/s10994-021-05961-4.

Fiorelli, E. et al. (Oct. 2006). "Multi-AUV Control and Adaptive Sampling in Monterey Bay". In: *IEEE Journal of Oceanic Engineering* 31.4. Conference Name: IEEE Journal of Oceanic Engineering, pp. 935–948. DOI: 10.1109/JOE.2006.880429.

Gazzola, Mattia, Babak Hejazialhosseini, and Petros Koumoutsakos (Jan. 2014). "Reinforcement Learning and Wavelet Adapted Vortex Methods for Simulations of Self-propelled Swimmers". In: *SIAM Journal on Scientific Computing* 36.3. Publisher: Society for Industrial and Applied Mathematics, B622–B639. DOI: 10.1137/130943078.

Germanovich, Leonid N. et al. (2015). "Measuring fluid flow and heat output in seafloor hydrothermal environments". In: *Journal of Geophysical Research: Solid Earth* 120.12. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/2015JB012245, pp. 8031–8055. DOI: 10.1002/2015JB012245.

Gharib, Morteza, Edmond Rambod, and Karim Shariff (Apr. 1998). "A universal time scale for vortex ring formation". In: *Journal of Fluid Mechanics* 360, pp. 121–140. DOI: 10.1017/S0022112097008410.

Glezer, Ari (Dec. 1988). "The formation of vortex rings". In: *The Physics of Fluids* 31.12. Publisher: AIP Publishing, pp. 3532–3542. DOI: 10.1063/1.866920.

Guerrero, Jose Alfredo and Yasmina Bestaoui (Jan. 2013). "UAV Path Planning for Structure Inspection in Windy Environments". In: *Journal of Intelligent & Robotic Systems* 69.1, pp. 297–311. DOI: 10.1007/s10846-012-9778-2.

Gunnarson, P, I Mandralis, G Novati, P Koumoutsakos, and J.O. Dabiri (2021). "Learning efficient navigation in vortical flow fields". In: *Nature Communications* 12.1. DOI: 10.1038/s41467-021-27015-y.

Gunnarson, P. and J.O. Dabiri (2024). "Fish-inspired tracking of underwater turbulent plumes". In: *Submitted*. DOI: 10.48550/arXiv.2403.06091.

Gunnarson, Peter et al. (Dec. 8, 2021). "Learning efficient navigation in vortical flow fields". In: *Nature Communications* 12.1. Number: 1 Publisher: Nature Publishing Group, p. 7143. DOI: 10.1038/s41467-021-27015-y.

Hang, Haotian et al. (Dec. 16, 2023). *Interpretable and Generalizable Strategies for Stably Following Hydrodynamic Trails*. Pages: 2023.12.15.571932 Section: New Results. DOI: 10.1101/2023.12.15.571932.

Hasselmann, Ken et al. (July 16, 2021). "Empirical assessment and comparison of neuro-evolutionary methods for the automatic off-line design of robot swarms". In: *Nature Communications* 12.1, p. 4345. DOI: 10.1038/s41467-021-24642-3.

Hasselt, Hado van, Arthur Guez, and David Silver (Dec. 8, 2015). *Deep Reinforcement Learning with Double Q-learning*. version: 3. DOI: 10.48550/arXiv.1509.06461. arXiv: 1509.06461[cs].

Henderson, Peter et al. (Jan. 2019). "Deep Reinforcement Learning that Matters". In: *arXiv:1709.06560 [cs, stat]*. arXiv: 1709.06560.

Hill, M. J. M. (1894). "On a Spherical Vortex". In: *Philosophical Transactions of the Royal Society of London. A* 185. Publisher: The Royal Society, pp. 213–245.

Inanc, T., S.C. Shadden, and J.E. Marsden (June 2005). "Optimal trajectory generation in ocean flows". In: *Proceedings of the 2005, American Control Conference, 2005.* Proceedings of the 2005, American Control Conference, 2005. 674–679 vol. 1. DOI: 10.1109/ACC.2005.1470035.

Jezov, Jaas et al. (Oct. 2012). "Sensing oscillations in unsteady flow for better robotic swimming efficiency". In: *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 91–96. DOI: 10.1109/ICSMC.2012.6377682.

Jiao, Yusheng et al. (Dec. 2020). "Learning to swim in potential flow". In: *arXiv:2009.14280 [physics, q-bio]*. arXiv: 2009.14280.

Jiménez, J. (2006). "Intermittency in Turbulence". In: DOI: 10.1016/B0-12-512666-2/00368-0.

Kadakia, Nirag et al. (Nov. 2022). "Odour motion sensing enhances navigation of complex plumes". In: *Nature* 611.7937. Number: 7937 Publisher: Nature Publishing Group, pp. 754–761. DOI: 10.1038/s41586-022-05423-4.

Katija, Kakani et al. (Jan. 2021). "Visual tracking of deepwater animals using machine learning-controlled robotic underwater vehicles". In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 859–868. DOI: 10.1109/WACV48630.2021.00090.

Ko, Hungtang, George Lauder, and Radhika Nagpal (Oct. 25, 2023). "The role of hydrodynamics in collective motions of fish schools and bioinspired underwater robots". In: *Journal of The Royal Society Interface* 20.207. Publisher: Royal Society, p. 20230357. DOI: 10.1098/rsif.2023.0357.

Krishna, Kartik, Steven L. Brunton, and Zhuoyuan Song (2023). "Finite Time Lyapunov Exponent Analysis of Model Predictive Control and Reinforcement Learning". In: *IEEE Access* 11. Conference Name: IEEE Access, pp. 118916–118930. DOI: 10.1109/ACCESS.2023.3326424.

Krishna, Kartik, Zhuoyuan Song, and Steven L. Brunton (Feb. 2, 2022). "Finite-horizon, energy-efficient trajectories in unsteady flows". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 478.2258. Publisher: Royal Society, p. 20210255. DOI: 10.1098/rspa.2021.0255.

Kuhnz, Linda A. et al. (Mar. 2020). "Benthic megafauna assemblage change over three decades in the abyss: Variations from species to functional groups". In: *Deep Sea Research Part II: Topical Studies in Oceanography*. Thirty-year time-series study in the abyssal NE Pacific 173, p. 104761. DOI: 10.1016/j.dsr2.2020.104761.

Kularatne, Dhanushka, Subhrajit Bhattacharya, and M. Ani Hsieh (Oct. 2018). "Going with the flow: a graph based approach to optimal path planning in general flows". In: *Autonomous Robots* 42.7, pp. 1369–1387. DOI: 10.1007/s10514-018-9741-6.

Lamb, Horace (1932). *Hydrodynamics*. Google-Books-ID: 2grlAQAACAAJ. Cambridge University Press. 770 pp.

LaValle, Steven M. and James J. Kuffner (May 2001). "Randomized Kinodynamic Planning". In: *The International Journal of Robotics Research* 20.5. Publisher: SAGE Publications Ltd STM, pp. 378–400. DOI: 10.1177/02783640122067453.

Lehmann, Moritz (Nov. 29, 2023). "Computational study of microplastic transport at the water-air interface with a memory-optimized lattice Boltzmann method". Doctoral thesis. Bayreuth. 230 pp.

Lekien, Francois et al. (Oct. 1, 2005). "Pollution release tied to invariant manifolds: A case study for the coast of Florida". In: *Physica D: Nonlinear Phenomena* 210.1, pp. 1–20. DOI: 10.1016/j.physd.2005.06.023.

Liao, James C (May 2007). "A review of fish swimming mechanics and behaviour in altered flows". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 362.1487. Publisher: Royal Society, pp. 1973–1993. DOI: 10.1098/rstb.2007.2082.

Lidtke, Artur K., Stephen R. Turnock, and Jon Downes (Nov. 2016). "Assessment of underwater glider performance through viscous computational fluid dynamics". In: *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. Tokyo, Japan: IEEE, pp. 364–373. DOI: 10.1109/AUV.2016.7778698.

Lolla, Tapovan et al. (Oct. 2014). "Time-optimal path planning in dynamic flows using level set equations: theory and schemes". In: *Ocean Dynamics* 64.10, pp. 1373–1397. DOI: 10.1007/s10236-014-0757-y.

Lundberg, Scott M. and Su-In Lee (Dec. 4, 2017). "A unified approach to interpreting model predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., pp. 4768–4777.

Masmitja, I. et al. (July 26, 2023). "Dynamic robotic tracking of underwater targets using reinforcement learning". In: *Science Robotics* 8.80. Publisher: American Association for the Advancement of Science, eade7811. DOI: 10.1126/scirobotics.ade7811.

Maxey, Martin R. and James J. Riley (Apr. 1983). "Equation of motion for a small rigid sphere in a nonuniform flow". In: *The Physics of Fluids* 26.4. Publisher: AIP Publishing, pp. 883–889. DOI: 10.1063/1.864230.

Maxworthy, T. (Jan. 1972). "The structure and stability of vortex rings". In: *Journal of Fluid Mechanics* 51.1. Publisher: Cambridge University Press (CUP), pp. 15–32. DOI: 10.1017/s0022112072001041.

Mei, Jiazhong, J. Nathan Kutz, and Steven L. Brunton (Dec. 2023). "Observability-Based Energy Efficient Path Planning with Background Flow via Deep Reinforcement Learning". In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. 2023 62nd IEEE Conference on Decision and Control (CDC), pp. 4364–4371. DOI: 10.1109/CDC49753.2023.10383428.

Michaelis, Brenden T. et al. (May 14, 2020). "Odor tracking in aquatic organisms: the importance of temporal and spatial intermittency of the turbulent plume". In: *Scientific Reports* 10.1. Number: 1 Publisher: Nature Publishing Group, p. 7961. DOI: 10.1038/s41598-020-64766-y.

Milne, I. A. et al. (Feb. 2013). "Characteristics of the turbulence in the flow at a tidal stream power site". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1985. Publisher: Royal Society, p. 20120196. DOI: 10.1098/rsta.2012.0196.

Mitchell, Ian M. (June 2008). "The Flexible, Extensible and Efficient Toolbox of Level Set Methods". en. In: *Journal of Scientific Computing* 35.2, pp. 300–329. DOI: 10.1007/s10915-007-9174-4.

Monthiller, Rémi et al. (Aug. 5, 2022). "Surfing on Turbulence: A Strategy for Planktonic Navigation". In: *Physical Review Letters* 129.6. Publisher: American Physical Society, p. 064502. DOI: 10.1103/PhysRevLett.129.064502.

Novati, Guido and Petros Koumoutsakos (May 2019). "Remember and Forget for Experience Replay". In: *arXiv:1807.05827 [cs, stat]*. arXiv: 1807.05827.

O'Connell, Michael et al. (May 2022). "Neural-Fly enables rapid learning for agile flight in strong winds". In: *Science Robotics* 7.66. Publisher: American Association for the Advancement of Science (AAAS). DOI: 10.1126/scirobotics.abm6597.

OECD (Sept. 2020). *Sustainable Ocean for All: Harnessing the Benefits for Developing Countries*. OECD. DOI: 10.1787/bede6513-en.

Oteiza, Pablo et al. (July 2017). "A novel mechanism for mechanosensory-based rheotaxis in larval zebrafish". In: *Nature* 547.7664. Number: 7664 Publisher: Nature Publishing Group, pp. 445–448. DOI: 10.1038/nature23014.

Panda, Madhusmita et al. (June 2020). "A Comprehensive Review of Path Planning Algorithms for Autonomous Underwater Vehicles". In: *International Journal of Automation and Computing* 17.3, pp. 321–352. DOI: 10.1007/s11633-019-1204-9.

Patton, Paul, Shane Windsor, and Sheryl Coombs (Nov. 1, 2010). "Active wall following by Mexican blind cavefish (Astyanax mexicanus)". In: *Journal of Comparative Physiology A* 196.11, pp. 853–867. DOI: 10.1007/s00359-010-0567-8.

Peng, J. and J. O. Dabiri (Mar. 25, 2009). "Transport of inertial particles by Lagrangian coherent structures: application to predator–prey interaction in jellyfish feeding". In: *Journal of Fluid Mechanics* 623, pp. 75–84. DOI: 10.1017/S0022112008005089.

Petres, C. et al. (Apr. 2007). "Path Planning for Autonomous Underwater Vehicles". In: *IEEE Transactions on Robotics* 23.2. Conference Name: IEEE Transactions on Robotics, pp. 331–341. DOI: 10.1109/TRO.2007.895057.

Pohlmann, Kirsten, Frank W. Grasso, and Thomas Breithaupt (June 19, 2001). "Tracking wakes: The nocturnal predatory strategy of piscivorous catfish". In: *Proceedings of the National Academy of Sciences* 98.13. Publisher: Proceedings of the National Academy of Sciences, pp. 7371–7374. DOI: 10.1073/pnas.121026298.

Preston, Victoria et al. (June 2, 2022). *Robotic Planning under Uncertainty in Spatiotemporal Environments in Expeditionary Science*. arXiv: 2206.01364[cs].

Reaka-Kudla, Marjorie L. (2001). "Known and Unknown Biodiversity, Risk of Extinction and Conservation Strategy in the Sea". In: *Waters in Peril*. Ed. by Leah Bendell-Young and Patricia Gallaugher. Boston, MA: Springer US, pp. 19–33. DOI: 10.1007/978-1-4615-1493-0_2.

Reddy, Gautam et al. (Oct. 2018). "Glider soaring via reinforcement learning in the field". In: *Nature* 562.7726. Number: 7726 Publisher: Nature Publishing Group, pp. 236–239. DOI: 10.1038/s41586-018-0533-0.

Renn, Peter I. and Morteza Gharib (Dec. 16, 2022). "Machine learning for flow-informed aerodynamic control in turbulent wind conditions". In: *Communications Engineering* 1.1. Number: 1 Publisher: Nature Publishing Group, pp. 1–9. DOI: 10.1038/s44172-022-00046-z.

Rhoads, Blane, Igor Mezić, and Andrew C. Poje (July 2013). "Minimum time heading control of underpowered vehicles in time-varying ocean currents". In: *Ocean Engineering* 66. Publisher: Elsevier BV, pp. 12–31. DOI: 10.1016/j.oceaneng.2013.03.012.

Rivière, B. et al. (July 2020). "GLAS: Global-to-Local Safe Autonomy Synthesis for Multi-Robot Motion Planning With End-to-End Learning". In: *IEEE Robotics and Automation Letters* 5.3. Conference Name: IEEE Robotics and Automation Letters, pp. 4249–4256. DOI: 10.1109/LRA.2020.2994035.

Rudnick, Daniel L. (2016). "Ocean Research Enabled by Underwater Gliders". In: *Annual Review of Marine Science* 8.1. _eprint: https://doi.org/10.1146/annurev-marine-122414-033913, pp. 519–541. DOI: 10.1146/annurev-marine-122414-033913.

Salumäe, Taavi et al. (May 2012). "Against the flow: A Braitenberg controller for a fish robot". In: *2012 IEEE International Conference on Robotics and Automation*. 2012 IEEE International Conference on Robotics and Automation, pp. 4210–4215. DOI: 10.1109/ICRA.2012.6225023.

Senatore, Carmine and Shane D. Ross (June 2008). "Fuel-efficient navigation in complex flows". In: *2008 American Control Conference*. 2008 American Control Conference, pp. 1244–1248. DOI: 10.1109/ACC.2008.4586663.

Shadden, Shawn C., Francois Lekien, and Jerrold E. Marsden (Dec. 2005). "Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows". en. In: *Physica D: Nonlinear Phenomena* 212.3, pp. 271–304. DOI: 10.1016/j.physd.2005.10.007.

Shariff, K. and A. Leonard (Jan. 1, 1992). "Vortex Rings". In: *Annual Review of Fluid Mechanics* 24 (Volume 24, 1992). Publisher: Annual Reviews, pp. 235–279. DOI: 10.1146/annurev.fl.24.010192.001315.

Shi, G. et al. (May 2019). "Neural Lander: Stable Drone Landing Control Using Learned Dynamics". In: *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9784–9790. DOI: 10.1109/ICRA.2019.8794351.

Simon, Nathaniel et al. (2023). "FlowDrone: Wind Estimation and Gust Rejection on UAVs Using Fast-Response Hot-Wire Flow Sensors". In: *Proceedings - ICRA 2023: IEEE International Conference on Robotics and Automation*. 2023 IEEE International Conference on Robotics and Automation, ICRA 2023. Institute of Electrical and Electronics Engineers Inc., pp. 5393–5399. DOI: 10.1109/ICRA48891.2023.10160454.

Singh, Satpreet H. et al. (Jan. 2023). "Emergent behaviour and neural dynamics in artificial agents tracking odour plumes". In: *Nature Machine Intelligence* 5.1, pp. 58–70. DOI: 10.1038/s42256-022-00599-w.

Solomon, T. H. and J. P. Gollub (Dec. 1988). "Chaotic particle transport in time-dependent Rayleigh-Bénard convection". en. In: *Physical Review A* 38.12, pp. 6280–6286. DOI: 10.1103/PhysRevA.38.6280.

Strokina, Nataliya et al. (Mar. 2016). "Joint Estimation of Bulk Flow Velocity and Angle Using a Lateral Line Probe". In: *IEEE Transactions on Instrumentation and Measurement* 65.3, pp. 601–613. DOI: 10.1109/TIM.2015.2499019.

Su, Yunxing, Monica M. Wilhelmus, and Roberto Zenit (Mar. 2023). "Asymmetry of motion: vortex rings crossing a density gradient". In: *Journal of Fluid Mechanics* 960. Publisher: Cambridge University Press (CUP). DOI: 10.1017/jfm.2023.165.

Sutton, Richard S et al. (2000). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press.

Takens, Floris (1981). "Detecting strange attractors in turbulence". In: *Dynamical Systems and Turbulence, Warwick 1980*. Springer Berlin Heidelberg, pp. 366–381. DOI: 10.1007/bfb0091924.

Techy, Laszlo (Oct. 2011). "Optimal navigation in planar time-varying flow: Zermelo's problem revisited". In: *Intelligent Service Robotics* 4.4, pp. 271–283. DOI: 10.1007/s11370-011-0092-9.

Thielicke, William and René Sonntag (May 31, 2021). "Particle Image Velocimetry for MATLAB: Accuracy and enhanced algorithms in PIVlab". In: 9.1. Number: 1 Publisher: Ubiquity Press, p. 12. DOI: 10.5334/jors.334.

Tian, Wenlong et al. (2017). "Layout Optimization of Two Autonomous Underwater Vehicles for Drag Reduction with a Combined CFD and Neural Network Method". en. In: *Complexity* 2017, pp. 1–15. DOI: 10.1155/2017/5769794.

Venturelli, Roberto et al. (Sept. 1, 2012). "Hydrodynamic pressure sensing with an artificial lateral line in steady and unsteady flows". In: *Bioinspiration & Biomimetics* 7.3, p. 036004. DOI: 10.1088/1748-3182/7/3/036004.

Verma, Siddhartha, Guido Novati, and Petros Koumoutsakos (June 2018). "Efficient collective swimming by harnessing vortices through deep reinforcement learning". In: *Proceedings of the National Academy of Sciences* 115.23. Publisher: National Academy of Sciences Section: Physical Sciences, pp. 5849–5854. DOI: 10.1073/pnas.1800923115.

Verma, Siddhartha, Costas Papadimitriou, et al. (Feb. 2020). "Optimal sensor placement for artificial swimmers". en. In: *Journal of Fluid Mechanics* 884. Publisher: Cambridge University Press. DOI: 10.1017/jfm.2019.940.

Watkins, S et al. (Jan. 2020). "Ten questions concerning the use of drones in urban environments". In: *Building and Environment* 167. Publisher: Elsevier BV, p. 106458. DOI: 10.1016/j.buildenv.2019.106458.

Watkins, Scott, Sridhar Ravi, and Benjamin Loxton (2009). "The effect of turbulence on the aerodynamics of low Reynolds number wings". eng. In: *Engineering Letters* 18.3. Place: Hong Kong Publisher: International Association of Engineers, p. 6.

Weber, Pascal et al. (Mar. 2020). “Optimal Flow Sensing for Schooling Swimmers”. In: *Biomimetics* 5.1, p. 10. DOI: 10.3390/biomimetics5010010.

Weizhong Zhang et al. (May 2008). “Optimal trajectory generation for a glider in time-varying 2D ocean flows B-spline model”. In: *2008 IEEE International Conference on Robotics and Automation*, pp. 1083–1088. DOI: 10.1109/ROBOT.2008.4543348.

Wölfl, Anne-Cathrin et al. (2019). “Seafloor Mapping – The Challenge of a Truly Global Ocean Bathymetry”. In: *Frontiers in Marine Science* 6.

Wong, Annie P. S. et al. (2020). “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.

Xu, Nicole and John Dabiri (2022). “Bio-Inspired Ocean Exploration”. In: *Oceanography*. Publisher: The Oceanography Society, pp. 35–48. DOI: 10.5670/oceanog.2022.214.

Yen, Wei-Kuo and Jenhwa Guo (Apr. 2016). “Wall following control of a robotic fish using dynamic pressure”. In: OCEANS 2016 - Shanghai, pp. 1–7. DOI: 10.1109/OCEANSAP.2016.7485700.

Zereik, Enrica et al. (Jan. 1, 2018). “Challenges and future trends in marine robotics”. In: *Annual Reviews in Control* 46, pp. 350–368. DOI: 10.1016/j.arcontrol.2018.10.002.

Zermelo, E. (1931). “Über das Navigationsproblem bei ruhender oder veränderlicher Windverteilung”. In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 11.2, pp. 114–124. DOI: https://doi.org/10.1002/zamm.19310110205.

Zhai, Yufan, Xingwen Zheng, and Guangming Xie (Mar. 1, 2021). “Fish Lateral Line Inspired Flow Sensors and Flow-aided Control: A Review”. In: *Journal of Bionic Engineering* 18.2, pp. 264–291. DOI: 10.1007/s42235-021-0034-y.

Zhang, Yanwu, John P. Ryan, et al. (2019). “Targeted Sampling by Autonomous Underwater Vehicles”. In: *Frontiers in Marine Science* 6.

Zhang, Yanwu, Noa Yoder, et al. (Oct. 2022). “Autonomous Tracking of Salinity-Intrusion Fronts by a Long-Range Autonomous Underwater Vehicle”. In: *IEEE Journal of Oceanic Engineering* 47.4. Conference Name: IEEE Journal of Oceanic Engineering, pp. 950–958. DOI: 10.1109/JOE.2022.3146584.

Zheng, Xingwen, Amar M. Kamat, et al. (Oct. 27, 2021). “Creating underwater vision through wavy whiskers: a review of the flow-sensing mechanisms and biomimetic potential of seal whiskers”. In: *Journal of The Royal Society Interface* 18.183. Publisher: Royal Society, p. 20210629. DOI: 10.1098/rsif.2021.0629.

Zheng, Xingwen, Wei Wang, Liang Li, et al. (Nov. 2020). "Artificial lateral line based relative state estimation between an upstream oscillating fin and a downstream robotic fish". In: *Bioinspiration & Biomimetics* 16.1. Publisher: IOP Publishing, p. 016012. DOI: 10.1088/1748-3190/abb86c.

Zheng, Xingwen, Wei Wang, Minglei Xiong, et al. (Apr. 2020). "Online State Estimation of a Fin-Actuated Underwater Robot Using Artificial Lateral Line System". In: *IEEE Transactions on Robotics* 36.2. Conference Name: IEEE Transactions on Robotics, pp. 472–487. DOI: 10.1109/TRO.2019.2956343.

Zolman, Nicholas et al. (2024). *SINDy-RL: Interpretable and Efficient Model-Based Reinforcement Learning*. DOI: 10.48550/ARXIV.2403.09110.

Appendix A

SUPPLEMENTARY MATERIAL FOR CHAPTER 2

A.1 Navigation in steady flow

For navigating through an unsteady von Kármán vortex street, flow sensing appeared critical for learning effective swimming strategies with RL. For navigating steady flows with a fixed target, however, an RL-swimmer may be able to navigate simply by forming a one-to-one correspondence between its position and the fixed background flow field.

Consider the steady 2D flow past a cylinder at a Reynolds number of 40. Here, the target position is fixed to be just downstream of the cylinder, and the swimmers are started randomly throughout the entire domain with a swimming speed of $U_{\text{swim}} = 0.5U_\infty$ (see Figure A.2). Successful swimmers generally start close enough to the cylinder to use the wake to navigate to the target, while unsuccessful swimmers generally start too far away from the cylinder to reach the target before being swept downstream.

We trained a flow-blind swimmer ($s = \{\Delta x, \Delta y\}$) and a vorticity swimmer ($s = \{\Delta x, \Delta y, \omega_n, \omega_{n-1}\}$). Even though the flow-blind swimmer had no knowledge of the background flow, it trained as quickly as the vorticity swimmer, which can be seen in the evolution of the cumulative reward over training (Figure A.1).

Additionally, both swimmers were equally successful at reaching the target after training, which is visualized by plotting the region of starting points from which the swimmer can reach the target (Figure A.2). This region is computed by extracting the ridges of the FTLE field (S. C. Shadden, Lekien, and J. E. Marsden, 2005) of the flow field formed by the background flow plus the swimmer's learned policy.

Because an RL swimmer can navigate steady flow with position alone, the unsteady cylinder wake was chosen for testing flow sensing-based navigation. Additionally, the starting time was randomized, as presenting a swimmer with a repeated, deterministic snapshot of a flow field could also be navigable by memorizing the flow based on position alone.

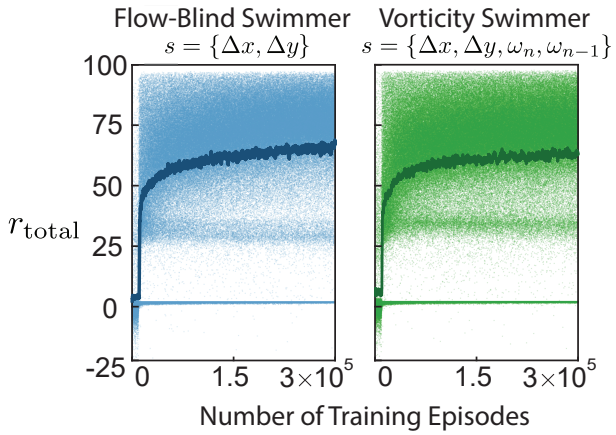


Figure A.1: Evolution of the cumulative reward for a flow-blind swimmer and a vorticity swimmer in steady flow past a cylinder. Because the background flow is steady and the target location is fixed, both the flow-blind and vorticity swimmers train at a similar rate.

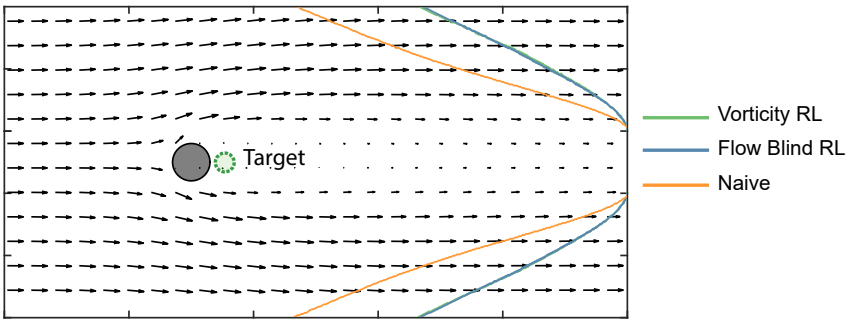


Figure A.2: Success of a naïve swimmer, flow-blind RL swimmer, and vorticity RL swimmer at navigating the steady flow past a cylinder. The plotted curves display the right-hand (i.e., downstream) boundary of the region in which the swimmer can reach the target. This region of attraction to the target is identical for both RL swimmers, and larger than the region for the naïve swimmer. This implies that in this steady flow field, both RL swimmers are equally successful at reaching the target despite one swimmer lacking flow sensing abilities.

A.2 Reinforcement learning algorithm

We employed the V-RACER algorithm for training the deep RL swimmers using the smarties framework. Some details of the algorithm are presented here, but a complete description can be found in Novati and Koumoutsakos (2019).

The goal of V-RACER is to train the weights w of a neural network using experiences with the environment. At each time-step t, the neural network takes in the agent's state s_t as an input and outputs the mean action μ^{w}, standard deviation σ^{w}, and value estimate v^{w}. During training, the swimmer takes an action a_t sampled from

the normal distribution $\mathcal{N}(\mu^{\mathrm{w}}, (\sigma^{\mathrm{w}})^2)$. The architecture of this deep neural network is shown below in Figure A.3.

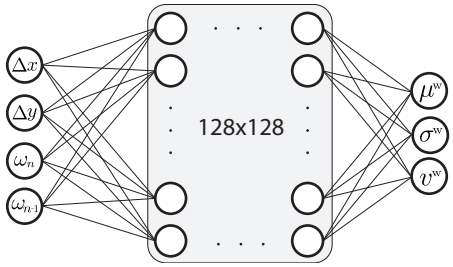


Figure A.3: Policy network used for deep RL with V-RACER. The state (e.g., $s = \{\Delta x, \Delta y, \omega_n, \omega_{n-1}\}$) is inputted into a deep neural network and the output is a mean action μ^{w}, standard deviation σ^{w}, and value v^{w}. During training, the swimmer chooses a swimming angle by random sampling from a normal distribution: $\theta \sim \mathcal{N}(\mu^{\mathrm{w}}, (\sigma^{\mathrm{w}})^2)$. After training, the mean action is selected ($\theta = \mu^{\mathrm{w}}$).

At every time step in an episode, the agent takes the action a_t and stores information such as the state (s_t), reward information ($r_t, \hat{V}_t^{\mathrm{tbc}}$), the current policy ($\mu_t, \sigma_t$), and action it took ($a_t$) in the Replay Memory (RM). The number of recorded experiences is kept to a fixed size: as new experiences are added, the oldest ones are forgotten.

To update the network's weights and biases, experiences are sampled from the RM, and a gradient is computed for each past experience according to two loss functions. The value output v of the neural network is trained using the following loss function, which seeks to improve the neural network's value estimate $v^{\mathrm{w}}(s_t)$ to better match the estimated value from experiences $\hat{V}_t^{tbc}$:

$$L^{\mathrm{ret}}(w) = \frac{1}{2}\left(v^{\mathrm{w}}(s_t) - \hat{V}_t^{tbc}\right)^2. \tag{A.1}$$

The mean and standard deviation outputs (μ^{w} and σ^{w}) are trained using the following loss function, which seeks to change the policy to improve on-policy returns:

$$L^{\mathrm{off-PG}}(\mathrm{w}) = -\rho_t^{\mathrm{w}}\left(r_{t+1} + \gamma\hat{V}_{t+1}^{tbc} - v^{\mathrm{w}}(s_t)\right), \tag{A.2}$$

where the importance weight ρ_t^{w} is the ratio of the probability of selecting a_t given s_t using the current policy $\pi(\cdot \mid s_t) \sim \mathcal{N}(\mu^{\mathrm{w}}, (\sigma_t^{\mathrm{w}})^2)$ and the old policy $\beta(\cdot \mid s_t) \sim \mathcal{N}(\mu_t, \sigma_t^2)$:

$$\rho_t^{\mathrm{w}} = \frac{\pi^{\mathrm{w}}(a_t \mid s_t)}{\beta(a_t \mid s_t)}. \tag{A.3}$$

The estimate of the return $\hat{V}_t^{tbc}$ is calculated using a recursive formula starting at the terminal time step N of an episode in the RM and stepping backwards:

$$\hat{V}_N^{tbc} = 0, \tag{A.4}$$

$$\hat{V}_t^{tbc} = v^{\mathrm{w}}(s_t) + \min\{1, \rho_t^{\mathrm{w}}\}\left[r_{t+1} + \gamma \hat{V}_{t+1}^{tbc} - v^{\mathrm{w}}(s_t)\right]. \tag{A.5}$$

The gradient estimate is then:

$$\hat{g}_{t_i} = \frac{\partial L^{\mathrm{ret}}}{\partial v^{\mathrm{w}}} \nabla v^{\mathrm{w}} + \frac{\partial L^{\mathrm{off-PG}}}{\partial \mu^{\mathrm{w}}} \nabla \mu^{\mathrm{w}} + \frac{\partial L^{\mathrm{off-PG}}}{\partial \sigma^{\mathrm{w}}} \nabla \sigma^{\mathrm{w}}. \tag{A.6}$$

If the old policy is too dissimilar from the current policy ($1/c_{max} < \rho_{t_i}^{\mathrm{w}} < c_{max}$), $\hat{g}_{t_i}$ is set to zero. Additionally, this gradient estimate is mixed with a gradient that points in the direction of the current policy to prevent the policy from changing too greatly in an unstable manner:

$$\hat{g}_{t_i}^{ReF-ER} = \lambda \hat{g}_{t_i} + (1-\lambda) \nabla D_{\mathrm{KL}}(\beta(\cdot \mid s_t) \mid\mid \pi(\cdot \mid, s_t)), \tag{A.7}$$

where D_{KL} is the Kullback–Leibler divergence, and λ is a chosen parameter. For the RL swimmers, the neural network was updated using $\hat{g}_{t_i}^{ReF-ER}$ at each time step. Additional details, such as pseudocode, hyperparameters, and the scaling of the neural network inputs are shown in Novati and Koumoutsakos (2019).

A network size of 128×128 was selected to ensure the network would be sufficiently expressive to learn effective navigation strategies. To confirm that a 128×128 is sufficiently expressive, we found that a network size of 64×64 was able to match the performance of a 128×128 network, which can be see below in Figure A.4.

A.3 Additional RL swimmers

In addition to flow-blind, vorticity, and velocity RL swimmers, several other swimmers with different states were investigated. The u-velocity swimmer has access to the only the x component of the flow velocity ($s = \{\Delta x, \Delta y, u\}$). The transverse-velocity swimmer has access to the component of the fluid velocity perpendicular to the swimmer's previous direction of travel ($s = \{\Delta x, \Delta y, u_\perp, v_\perp\}$). Finally, the vorticity-velocity swimmer has access to the both the vorticity and velocity of the fluid ($s = \{\Delta x, \Delta y, \omega_n, \omega_{n-1}, u, v\}$).

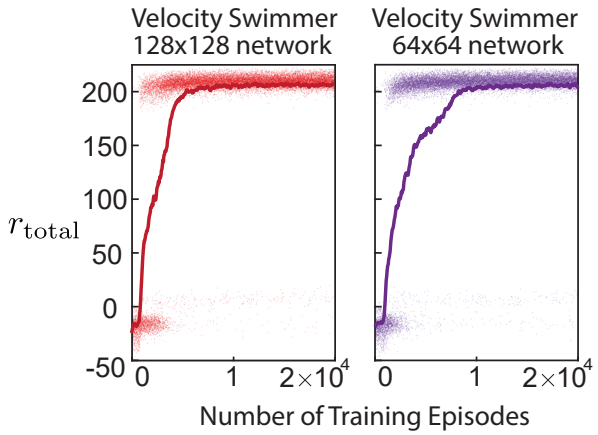


Figure A.4: Evolution of the cumulative reward for a velocity swimmer with a 128×128 neural network and a 64×64 neural network, which shows that a 128×128 neural network is sufficiently large to express an effective swimming policy.

The success rate of all swimmers is plotted in Figure A.5. All swimmers with access to partial-flow information (e.g., vorticity, one velocity component) have a slightly higher success rate than the flow-blind swimmer. The two swimmers with access to both components of the fluid velocity reach a nearly 100% success rate, and the inclusion of vorticity in addition to the flow velocity did not appear to impact training time.

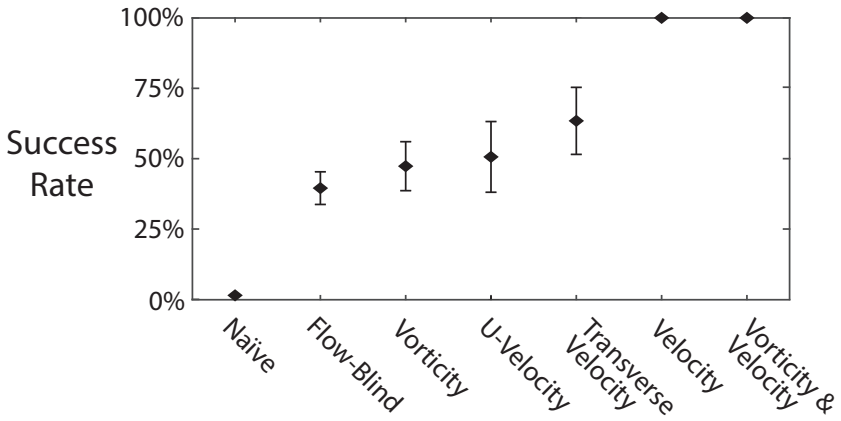


Figure A.5: Success rates of all investigated swimmers with various local sensing schemes. The stated success rates are averaged over 12,500 episodes and are shown with one standard deviation arising from the five times each swimmer was trained.

A.4 Sensor noise

All swimmers have thus far been presented noiseless measurements of the background flow and the swimmer's position. In a biological context, such low-noise

measurements may not be unreasonable: seals were reported to detect flow velocities as low as 245 microns per second using specially adapted whiskers (Dehnhardt, Mauck, and Bleckmann, 1998). In robotic systems, however, measurement noise could arise from a variety of sources, such as a measurement device itself, or from small scale turbulence when navigating with background flow. Given that turbulence is ubiquitous in real-world flows and the RL swimmers in the present study rely on background flow measurements to navigate, we investigated how noise in the flow measurements affects the success rate of RL swimmers. We found that the velocity swimmer can be robust to realistic amounts of flow sensing noise.

To simulate flow measurement noise, an already trained velocity RL swimmer was tasked with navigating across the unsteady cylinder wake with zero-mean Gaussian noise added to both components of its velocity measurement. The standard deviation of the velocity sensor noise, σ_{sensor}, was varied between 0 and 0.5 times U_∞. The success rates of the velocity swimmer for various amounts of noise are shown in Figure A.6.

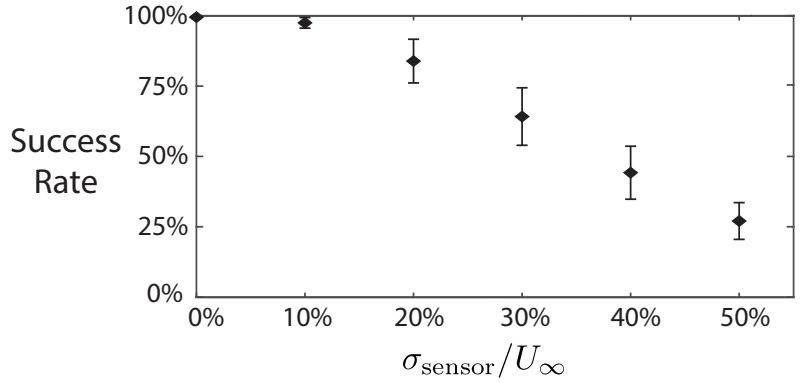


Figure A.6: Success rates of a velocity RL swimmer with various amounts of zero-mean Gaussian noise added to its local velocity measurement (position measurements were left noiseless). The stated success rates are averaged over 12,500 episodes and are shown with one standard deviation arising from the five times each swimmer was trained.

The velocity RL swimmer demonstrated robustness to noise in its local velocity measurement, showing little decrease in the success rate with a σ_{sensor} of up to 10% of the freestream flow velocity. With higher amounts of noise, the success rate decreased, although the velocity swimmer can still navigate more successfully than flow-blind swimmer even when σ_{sensor} reaches 40% of U_∞. It is not surprising that velocity measurements are less useful at this noise level, because the noise is comparable to the measured signal.

A convenient, albeit crude, metric for comparison with real fluid flows is the turbulence intensity, defined as the ratio of the root-mean squared velocity fluctuations over the mean velocity ($u_{\text{rms}}/\bar{U}$). While turbulence does not exactly follow a Gaussian distribution (Jiménez, 2006) and not all scales of turbulence would be small enough to appear as random fluctuations to a swimmer (Scott Watkins, Ravi, and Loxton, 2009), we can compare the turbulence intensity of real flows to the random noise added to the swimmer's sensor readings.

Typical turbulence intensities used for the design of underwater autonomous vehicles can range from 0.2% to 9% (Lidtke, Turnock, and Downes, 2016; Tian et al., 2017). For tidal flows in the ocean, the turbulent intensity has been measured to be approximately 12% to 13% (Milne et al., 2013) but could be higher near surface waves or lower in the mid-ocean. For aerial vehicles, typical turbulence intensities could range from 1.2% to 12.6% (Scott Watkins, Ravi, and Loxton, 2009). By comparison, the RL swimmer has a minimal decrease in its success rate with a similar magnitude of random noise in its velocity measurement. To be sure, the turbulence intensity can range from zero to infinity depending on the presence of a mean background current, and can vary in magnitude for different velocity components (Scott Watkins, Ravi, and Loxton, 2009), so this comparison is only a rough approximation of in situ noise levels. A physical implementation of RL navigation with real-world noise would provide a more conclusive result, but it is nevertheless promising that perfectly noiseless flow measurements are not required for high navigation success in the simulated case.

Appendix B

COMPONENTS AND CONSTRUCTION OF CARL

In this appendix, the components and construction of CARL are described. This documentation will focus specifically on the most recent and robust version of CARL, which was used for the experiments in Section 4. In addition to the information contained here, the full design files, code, and bill of materials can be found at https://doi.org/10.22002/b291n-nx970.

General overview of CARL

An overview of the main components of CARL is shown below in Figure B.1. The hull was 3D printed using photopolymer resin, and other structural components were 3D printed using Acrylonitrile Butadiene Styrene (ABS). The hull was approximately 6 cm in diameter and 10 cm tall, and was weighted on the bottom to passively stabilize the robot in its vertical orientation. The majority of electronic components were housed inside the hull and accessible via a removable cap that formed a watertight seal with two O-rings. A Teensy 4.1 microcontroller provided onboard control and computation, and data were stored onboard using a micro-SD card. A wireless communication module (Songhe NRF24l01+ mini) was used to send and receive commands and data from CARL while on the surface, using a corresponding Teensy and wireless module that communicate with a computer over serial. Because the water in the tank blocked wireless communication, CARL operated autonomously while underwater. Additionally, an IMU (MPU-6050, TDK InvenSense) measured acceleration and angular rotation rate for wall impact detection and active rotational stabilization. A 12-watt-hour lithium-ion battery (Samsung 35E 18650) provided power to the robot and was replaced after an hour of swimming in typical experimental conditions. The electronic components were mounted on custom printed circuit boards (PCBs).

For propulsion, CARL was equipped with ten brushed DC motors (Crazepony 615 17500KV) with corresponding propellers. The motors receive power from the battery through H-bridge motor drivers (Texas Instruments DRV8833) and are controlled using PWM signals from the Teensy microcontroller. The arrangement of the motors is shown in Figure B.1. This arrangement of ten motors was chosen to provide full control over translation in all three axes. Wires from all exterior

electronic components were passed through the hull and sealed with epoxy.

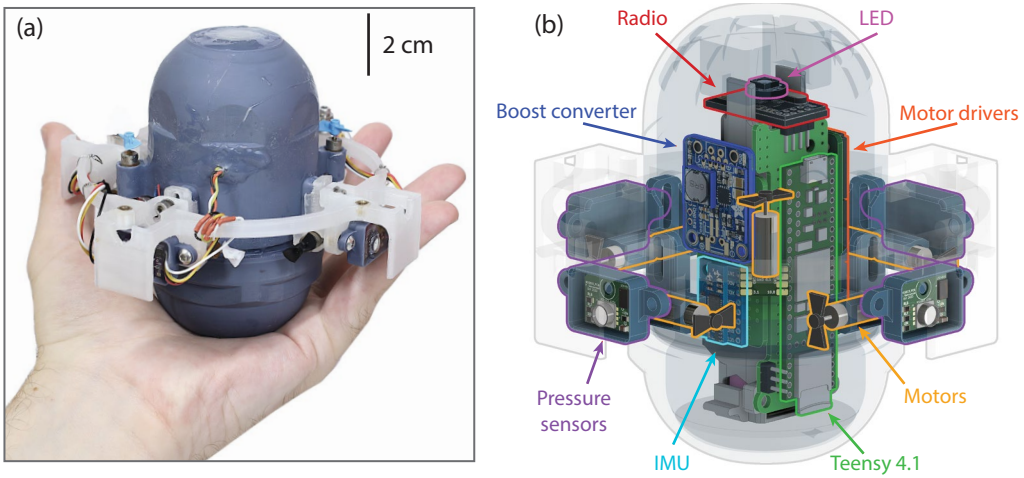


Figure B.1: General overview of the design of CARL. (**a**) Photograph of CARL with a 2 cm scale bar. (**b**) Overview of the primary electronic systems inside CARL.

Hull design and waterproofing

The main hull of CARL was fully 3D-printed using photo-polymer resin (Siraya Tech Fast ABS-Like 3D Printer Resin, Navy Grey) on a masked stereo-lithography apparatus (MSLA) 3D-printer (Elegoo Mars Pro 2). Based on the experience of designing CARL, it is *strongly* recommended to use resin 3D printing as opposed to fused-deposition modeling (FDM) printing for waterproof enclosures.

The previous version of CARL used in Section 3 was printed using polylactic acid (PLA) on a fused-deposition modeling printer (Voron 2.4). Small gaps between layers in the print are inherent to the FDM printing process, and necessitated that the entire hull be waterproofed with a coating of two-part epoxy. Additionally, the relatively rough surface finish often resulted in small leaks through the double O-ring shaft seal of the hull, in one case resulting in CARL sinking to the bottom of the tank.

In contrast, the resin-printing processed used for the current version of CARL is superior for two reasons. First, the printing process results in a fully watertight surface without post-processing, since no gaps form between curing the layers of resin in the printing process. Second, resin prints can easily achieve a highly smooth surface finish, particularly when a flat face is aligned with the xy plane of the build platform or the xz and yz planes of the pixels of the masking display of the printer. With a smooth surface finish, O-real seals are highly effective. As a result, zero

leaks occurred throughout all experiments and preliminary tests in Section 4. The other 3D printed parts used on the exterior of CARL, such as the mounts for the pressure sensors and motors, were typically printed in ABS plastic since they did not need to be waterproof.

To enable easy access to the electronics inside the hull of CARL, the hull was manufactured in two parts: the primary hull, and the hull cap. The two pieces are threaded such that they can be twisted together, which engages an O-ring face seal. To make it easier to open and close the two sections, particularly when slippery with water after being in the tank, two 3D-printed wrenches are used to easily twist apart or tighten the main hull and hull cap. This O-ring design and waterproofing method was the most successful and robust by far, and resulted in zero leaks during operation. The O-ring design is shown below in Figure B.2.

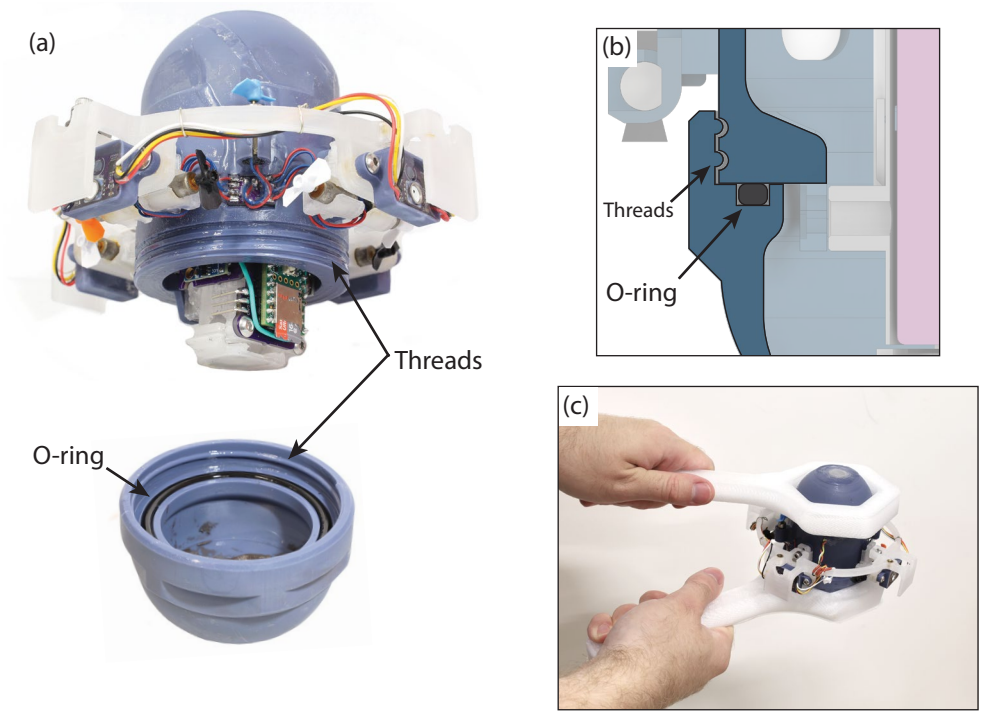


Figure B.2: O-ring system used to seal the hull of CARL. (**a**) Photograph of the main hull of CARL (top) and the hull cap (bottom), which screw together to form a watertight seal. (**b**) Design of the O-ring face seal and threads. (**c**) 3D-printed wrenches used to more easily open and close the hull of CARL.

Additionally, it was necessary to pass wires from the exterior of the hull to the interior, such as wires for the motors and wires for the pressure sensors. To accomplish this, the wires passed through small holes in the hull, and the holes were

sealed using two-part epoxy (see Figure B.3 for examples). Many ocean robots use specially designed cable penetrators to pass wires through pressure vessels (e.g., AUVs designed by Blue Robotics). While highly robust for ocean environments and high pressures, it was simpler, more cost effective, and more space-efficient to simply seal the wire holes on CARL with epoxy. Because the resin used for the hull of CARL was opaque, a small clear window was added to the top of CARL, printed using ABS plastic. Epoxy was also used to seal this component.

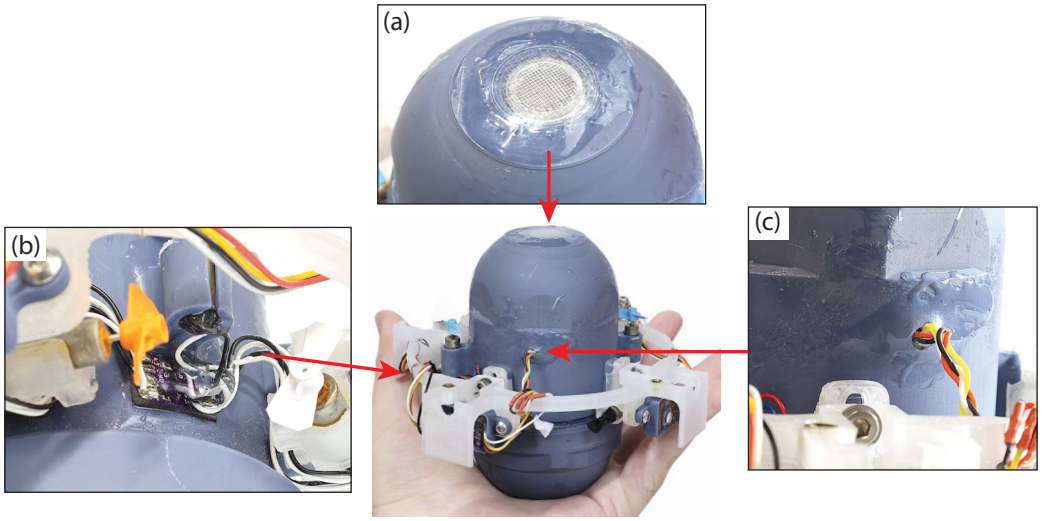


Figure B.3: Clear epoxy was used to seal the holes left by wires penetrating the hull of CARL. (**a**) A clear window was added to the top of CARL for the LED. (**b**) Motor wires were sealed and passed through the hull of CARL using epoxy. (**c**) The wires to the pressure sensors were sealed with epoxy.

Weight distribution and passive stability

The hull cap was weighted to shift the center of mass below the center of buoyancy, which ensured passive roll and pitch stability. Any dense material, including loose bolts and pieces of metal, could be used to add weight. However, tungsten putty, often used in model cars and airplanes, was preferred due to its relatively high density, the ease to which it can be shaped and pieces can be added and removed, and that it attaches securely to the inside of the hull cap without adhesives. Additionally, for fine-tuning the side-to-side position of the center of mass, a small amount of weight was added to the outside pressure sensor mounting ring. Screws and washers were used for this weight, and added manually until CARL floated without listing to one side.

CARL was made positively buoyant such that in the event of a hardware failure or

software bug, CARL would float to the surface for rescue. To dive to a certain depth, two diving motors described in subsequent sections provided downward thrust. The weight of CARL was adjusted so that the diving thrusters only needed to operate at approximately 25% thrust to maintain a constant depth.

Central electronics package

The majority of electronics components, including the battery, microcontroller, and wireless radio, were contained in the central electronics package inside the hull of CARL. The battery was centrally located in this assembly for weight distribution. Electronic components were mounted around a 3D printed structure and connected together with custom printed circuit boards (PCBs, see the repository for design files). The wireless radio was soldered on top of the electronics package to maximize reception while floating at the surface. An RGB LED was also mounted at the top, and used for tracking with an overhead camera. The Teensy 4.1 microcontroller provided all onboard computation, and the relevant code is uploaded to the repository. A PSRAM module was soldered to the Teensy, which increased the onboard memory capacity by 8 MB. After each episode in the tank, data were transferred from memory to an onboard SD card in a binary format for quick transfer to a computer. The Teensy was oriented such that the SD card could be easily accessed after opening the hull cap of CARL. To make the USB port on the Teensy accessible in this orientation, the ground, D+, and D- pins of the teensy were wired to pins on the bottom of the electronics package. Communication with the pressure sensors and accelerometer were accomplished over the I2C ports on the Teensy.

Power was provided to the Teensy through a 5 V boost converter, which steps the 3.7 V of the battery up to 5 V for the Vin pin on the Teensy. The voltage regulator on the Teensy then steps this voltage down to 3.3 V, which was used as the power source for the pressure sensors and accelerometer. The motors were controlled using H-bridge drivers, which sourced power directly from the battery. It is recommended that future versions of CARL use a driver with a higher maximum current capacity, as the current version can sometimes be overloaded during swimming. The battery voltage was recorded using an analog input pin on the Teensy, and communicated wirelessly in the acknowledgment packets sent by the wireless radio so that the battery voltage could be monitored during experiments.

Motors and active stability

CARL was equipped with ten brushed DC motors (Crazepony 615 17500KV) with corresponding propellers. Originally intended for propulsion in air, the propellers were cut by hand to a diameter of 15 mm from the original diameter of 42 mm to account for the higher torque required for operation in water. While not designed for use in water, the motors operated without failure in the freshwater environment of the tanks used in this dissertation.

Eight of the motors were arranged horizontally to enable translation in both horizontal axes (front-back, left-right) and to control the rotation of the robot in the vertical axis. A 50 Hz PID control loop running on the microcontroller read the angular rate from the onboard IMU and controlled the rotation of the robot to ensure straight swimming and accurate turning. Two propellers were mounted vertically for diving. CARL was slightly positively buoyant and rises to the surface when these two propellers are turned off. Another 50 Hz PID control loop ran in parallel which controlled the depth of CARL. Using the measured pressure at the surface as the reference pressure, the depth was estimated using the hydrostatic equation (h = P/rho g).

Pressure sensors

Up to eight pressure sensors (MS5803-02BA, TE Connectivity) were mounted on CARL for pressure sensing. Each pressure sensor was soldered to custom PCBs with I2C translator chips, which allowed all pressure sensors to communicate over the same I2C data lines. The pressure sensors were daisy-chained together and wired through the hull to the I2C port in the central electronics package. Inside the hull of CARL, the I2C wires were made using shielded cable, which eliminated noise in the pressure measurements caused by electromagnetic interference from the boost converter. For future versions of CARL, it is recommended to also shield the wires external to CARL, as the surrounding water adds capacitance to the I2C wires, which can interrupt communication. The pressure sensors come with a waterproof gel coating for the sensor. To waterproof the surrounding PCB, the pressure sensor was potted in two-part epoxy, which is shown below in Figure B.4. As an additional note, for the differential pressure measurements performed in Section 3, it is imperative that the side-facing and downward-facing sensors are as close together as possible to reduce errors caused by body rotation, which changes the relative hydrostatic pressure of the sensors.

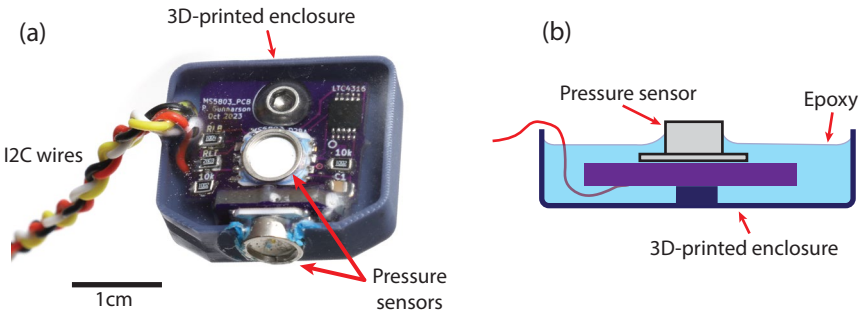


Figure B.4: Method for waterproofing the pressure sensors. (**a**) Photograph of a pressure sensor pair with a 1 cm scale bar. (**b**) Schematic of the waterproofing method, in which the printed circuit board was potted with clear epoxy.

Afterword

The design and development of CARL over nearly four years of my life has been a challenging experience. There have been no shortage of frustrating days and weeks with sensor failures, electromagnetic interference between components, mistakes in PCB design, broken 3D printers, misbehaving neural networks, the C++ programming language, and the near constant war against water ingress and leaks. Nevertheless, the rewards of the end result has outweighed the difficulties of development. What initially began as a toy submarine tested in my bathtub during the height of the COVID-19 pandemic is now a robust experimental platform with onboard flow sensing, autonomous navigation, reinforcement learning, and vortex-ring-surfing capabilities. Since it may not have been appropriate to thank and anthropomorphize a robot in the acknowledgments, I'll will say it here: thank you for everything, CARL.

"Another Hafu original!"

— Hafu, Mata Nui Online Game

Appendix C

SUPPLEMENTARY MATERIAL FOR CHAPTER 4

C.1 Measurement of vortex ring properties

To measure the circulation of the vortex ring, we used PIV measurements of an isolated vortex ring (i.e., without the presence of CARL) recorded by the high-speed camera. The circulation was computed by integrating the vorticity field from those PIV measurements:

$$\Gamma(t) = \int \omega \, \mathrm{d}x\mathrm{d}r, \tag{C.1}$$

where x is the streamwise direction as defined in Figure 4.1, and r is the distance from the x-axis. Because the left edge of the FOV of the high-speed camera is centered on the jet, the integration in Equation C.1 and the following equations are averaged over the top and bottom halves of the field of view. Additionally, the raw velocity vectors were filtered in time with a notch filter to eliminate noise due to oscillations in laser intensity ($f_{\text{laser}} \approx 68.15 \pm 0.15$ Hz) and a moving average filter with a window of 0.033 seconds to reduce high-frequency noise.

After the vortex ring was fully formed and separated from the thruster, we measured the circulation to be approximately 0.21 $\mathrm{m^2 s^{-1}}$. The circulation normalized by the kinematic viscosity ν can be considered a Reynolds number of the vortex ring (Glezer, 1988), which here is measured to be $\mathrm{Re} = \Gamma/\nu \approx 200,000$.

To define an appropriate time scale for the evolution of the vortex ring and the surfing experiments, we use the non-dimensional formation time, defined as $\overline{U}_p t/D$ (Gharib et al. 1998), where D is the diameter of the orifice producing the vortex ring, and $\overline{U}_p$ is the running mean of the fluid velocity exiting the thruster ($\overline{U}_p = 1/t \int_0^t U_p \mathrm{d}t$). While U_p can be directly measured in experiments that produce vortex rings with a moving piston (e.g. Gharib, Rambod, and Karim Shariff, 1998), it is difficult to accurately measure the flow velocities near the thruster exit. Therefore, we use a slug model to estimate U_p, which assumes that the vorticity flux arises from a thin boundary layer with freestream velocity U_p (K. Shariff and Leonard, 1992):

$$\frac{\mathrm{d}\Gamma}{\mathrm{d}t}(t) \approx \frac{1}{2} U_p^2(t). \tag{C.2}$$

Using Equation C.2 and taking the finite difference of the measured circulation, We measure a mean piston velocity of 0.98 m s^{-1}, and a plot of the estimated piston velocity over time is shown in Figure C.1. To generate a vortex, the thruster was commanded to operate at maximum speed for 0.5 seconds, but due to the time required to ramp up to that maximum speed, the jet velocity U_p was non-zero for a duration of approximately 0.36 seconds. We used that time interval to compute the mean piston velocity. Using $\overline{U}_p$, we estimate the formation number to be $\overline{U}_p t/D \approx 3.61$, which is within the typical range at which the vortex ring accumulates the maximum possible circulation (Gharib, Rambod, and Karim Shariff, 1998).

To facilitate further comparison with existing vortex ring literature, we also computed the non-dimensional energy of the vortex ring (Gharib, Rambod, and Karim Shariff, 1998), which is defined as:

$$\alpha(t) \equiv \frac{E}{I^{1/2}\Gamma^{3/2}}, \tag{C.3}$$

where E is the kinetic energy, I is the impulse, and Γ is the circulation. These quantities are computed from PIV measurements at each time step as follows:

$$E = \pi \int \frac{1}{2}|\mathbf{u}|^2 r \mathrm{d}x\mathrm{d}r, \quad I = \pi \int \omega r^2 \mathrm{d}x\mathrm{d}r \tag{C.4}$$

with the density set to unity. The non-dimensional energy α and piston velocity U_p are plotted below in Figure C.1. In dimensional units and using $\rho = 1000$ kg m^{-3} for water, the energy and impulse were measured to be 1.48 J and 4.48 kg m s^{-1}, respectively.

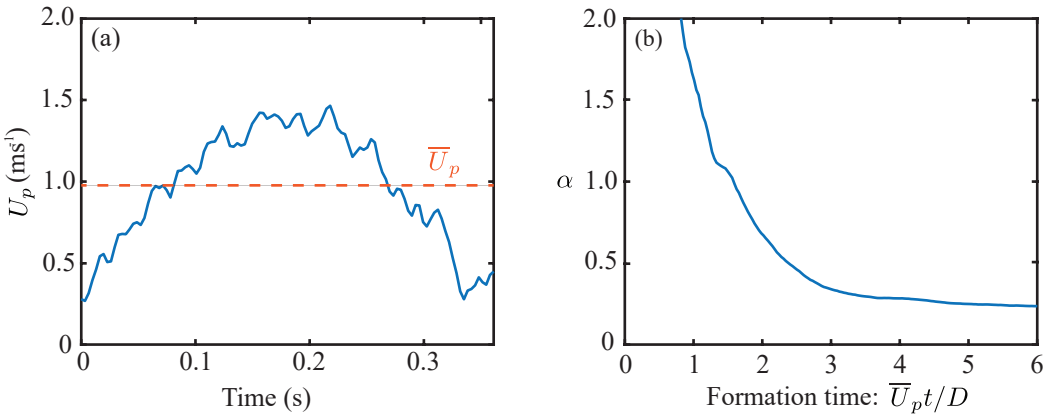


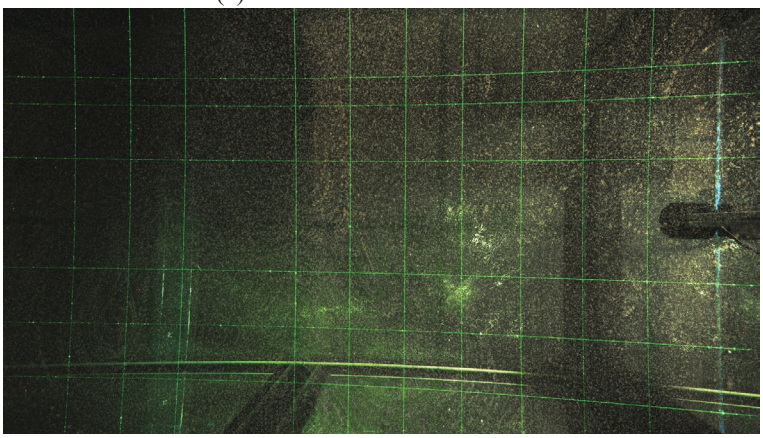
Figure C.1: Properties of a vortex ring generated by the thruster, measured using PIV from the high-speed camera. (**a**) Piston velocity estimated using a slug model. (**b**) Non-dimensional energy.

C.2 Free surface distortion correction

Due to constrains with the size of the room and tank used for experiments, a camera with a wide angle of view was positioned above the tank to track the positions of CARL and record wide-field PIV measurements (see Figure 4.1). However, the extreme FOV (107 degrees) resulted in significant "pincushion" image distortion due to refraction at the air-water interface. In order to record accurate PIV measurements and position data of CARL, we performed distortion correction on all images and video captured by the wide angle camera.

Image distortion was measured by shining a laser pointer through the side walls of the tank at the height of the thruster and at regular eight-inch intervals, using the reflection from the opposite side of the tank to ensure the laser was perpendicular and level. At each position, an image was taken, and then all images were combined together to produce a grid of distorted lines (Figure C.2a). Using this combined image, the optics compensation effect in Adobe After Effects was applied to produce an undistorted image in which all laser lines were perpendicular and equally spaced (Figure C.2b). The distortion correction was then applied to all recorded images and video files from the wide-angle camera. The average distance between laser lines was also used as a distance scale bar. The high-speed camera used a narrower FOV (23 degrees diagonally), and so distortion corrections were not required.

(a) Before distortion correction



(b) After distortion correction

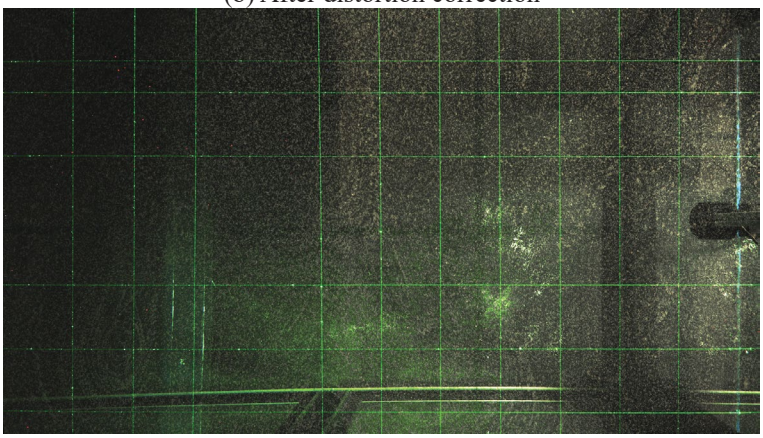


Figure C.2: Field of view of the wide-angle camera before correcting for refraction at the air water interface (**a**) and after correction (**b**).

C.3 Effects of body tilt on accelerometer measurements

An important consideration for implementing onboard acceleration sensing is to account for body rotation, such as roll and tilt of the robot. For example, when CARL is entrained into the vortex ring, the combined effects of the surfing maneuver and the vortex ring result in wobbling motion, which introduces a component of the gravitational acceleration into the x and y-acceleration in the body frame of the robot.

To rotate the acceleration vector back into a fixed, lab frame of reference, the coordinate system of a robot can be estimated at every instant in time using rotational acceleration data. Accelerometers often are paired with three-axis gyroscope measurements, which can be integrated to recover the orientation of the robot over time. To compensate for drift due to offsets in the integrated angular acceleration, some

IMUs also include a three-axis magnetometer, which uses Earth's magnetic field as an absolute orientation measurement. For the purposes of the experiments in Chapter 4, which only lasted 12 seconds, it was sufficient to integrate the gyroscope data directly with an offset measured at the start of each episode.

The differences between the acceleration in the body frame and lab fixed frame are shown below in Figure C.3a and C.3b, respectively. To verify that the corrected acceleration in (b) is accurate, the acceleration of CARL was computed from overhead camera position measurements in panel (c), and shows good agreement. A small amount of oscillation is present throughout the acceleration computed with the overhead cameras due to position errors from small waves in the free surface of the tank.

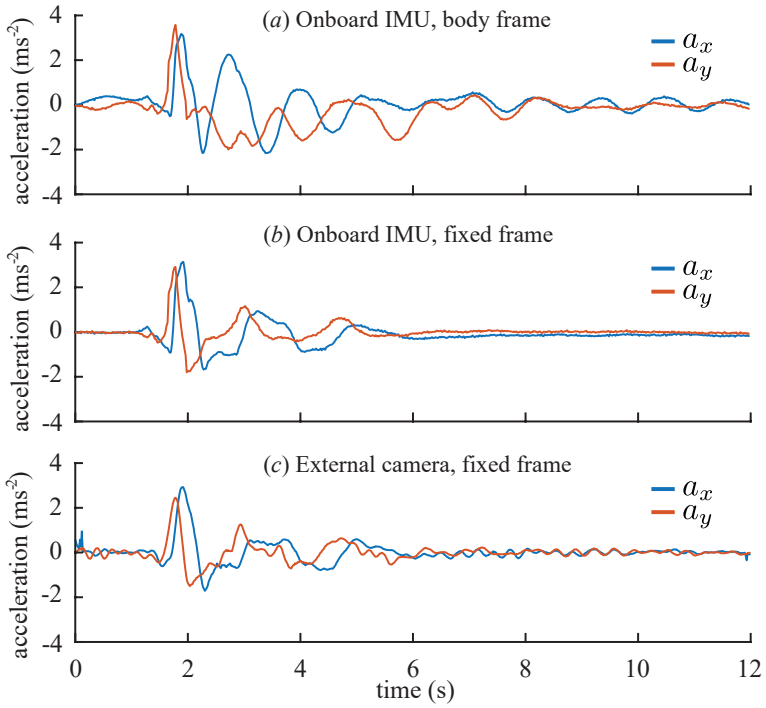


Figure C.3: Validation of the acceleration measurements recorded by the IMU onboard CARL. (**a**) Raw acceleration measured by the IMU onboard CARL in the CARL body frame. (**b**) Onboard acceleration in a fixed frame, using integrated gyroscope data to remove the tilt of the body frame. The onboard acceleration in a fixed frame matches the acceleration calculated from trajectory data measured recorded by the wide-angle camera (**c**). Because the center of mass of CARL is located below the center of buoyancy, CARL often wobbles during forward swimming or while near a vortex ring. This wobble is present in the raw acceleration data (**a**), but is effectively removed by computing the acceleration in a fixed frame of reference (**b**).