

“© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”  
DOI: 10.1109/ICRA.2019.8793567

# Streamlines for Motion Planning in Underwater Currents

K. Y. Cadmus To<sup>1</sup>, Ki Myung Brian Lee<sup>1</sup>, Chanyeol Yoo<sup>1</sup>, Stuart Anstee<sup>2</sup> and Robert Fitch<sup>1</sup>

**Abstract**—Motion planning for underwater vehicles must consider the effect of ocean currents. We present an efficient method to compute reachability and cost between sample points in sampling-based motion planning that supports long-range planning over hundreds of kilometres in complicated flows. The idea is to search a reduced space of control inputs that consists of stream functions whose level sets, or streamlines, optimally connect two given points. Such stream functions are generated by superimposing a control input onto the underlying current flow. A streamline represents the resulting path that a vehicle would follow as it is carried along by the current given that control input. We provide rigorous analysis that shows how our method avoids exhaustive search of the control space, and demonstrate simulated examples in complicated flows including a traversal along the east coast of Australia, using actual current predictions, between Sydney and Brisbane.

## I. INTRODUCTION

Underwater vehicles are important in many applications including environmental monitoring [1], oil and gas exploration [2], and defence [3]. While the majority of robotic ocean sensors drift freely with the current [4], the requirement to concentrate sensing in areas of high priority has led to increased interest in buoyancy-driven autonomous underwater gliders [5,6], propeller-driven autonomous underwater vehicles (AUVs) [7], and hybrids of the two [8]. We are interested in improving the autonomous operation of underwater vehicles by introducing a new planning tool that gains computational efficiency by exploiting a concept in fluid dynamics called a *stream function*.

The motion of underwater vehicles is heavily influenced by prevailing ocean currents. Optimal motion planning for underwater vehicles can be viewed as an instance of the long-standing *Zermelo's Problem* [9], for which there is no known efficient analytical solution in general. A numerical approach is to apply the well-known *shooting method*, which can be used within a sampling-based planning framework to find edge connections and costs. We use such a method, for example, within FMT\* to produce an asymptotically optimal minimum-energy planner for gliders in our recent work [10]. However, an open challenge is how to efficiently find edge connections in this approach. The shooting method involves forward integration along the edge, which must be repeated

for each of a set of control values to solve the underlying two-point boundary value problem. This high computational cost limits the number of edges that can be evaluated in practice, which in turn limits the solution quality or geographical scale of problem instances that can be feasibly solved.

In this paper, we examine the use of stream functions within a sampling-based planner to find edge connections. Level sets of the stream function, known as *streamlines*, represent paths that a vehicle would follow as it is carried along by the ocean current in the absence of any control input. We propose control inputs that induce a new stream function that, when superimposed on the underlying current, acts as a kind of local roadmap. We show how to efficiently search the space of control actions so that a streamline path between two points is found, if one exists. Forward integration of control inputs is still required, but our method avoids exhaustive search in control space and hence gains a major computational advantage by performing far fewer integrations than in a typical shooting method implementation.

We integrate our streamline method with PRM\* and provide rigorous computational analysis. The planning problem is formulated in the horizontal plane, and we demonstrate paths through flow fields with multiple gyres; some over hundreds of kilometres. The main significance of our result is that the streamline method can be used with a variety of asymptotically optimal planning algorithms, such as RRT\*, FMT\*, and other variants, to efficiently find high quality paths with sparse control inputs. We focus here on time-optimal paths, but other objectives such as minimum energy can be accommodated by modifying the objective function of our streamline method.

## II. RELATED WORK

Mission plans for underwater vehicles must allow for the oceanic flows through which they travel, and predictions of three-dimensional (3D) oceanic flow fields are freely available from multiple sources [11–13]. A method that estimates flows from observations is given by [14].

Although optimal planning in flow fields is generally well-studied, existing methods exhibit critical limitations. Work most closely related to ours elegantly uses level-set methods to find time-optimal [15] or energy-optimal [16] paths by explicitly solving for the reachable set as a level set of a scalar function. A numerical solution of the partial differential equation involved, however, can be computationally prohibitive in robotics applications. A variety of graph-based methods have been proposed, where the workspace is uniformly [17] or adaptively [18] discretised. This class of algorithms is resolution-complete, and is subject to well-known

This research is supported by an Australian Government Research Training Program (RTP) Scholarship, Australia's Defence Science and Technology Group, Bureau of Meteorology, and the University of Technology Sydney.

<sup>1</sup>Authors are with the University of Technology Sydney, Ultimo, NSW 2006, Australia {Cadmus.To, Brian.Lee}@student.uts.edu.au and {Chanyeol.Yoo, Robert.Fitch}@uts.edu.au

<sup>2</sup>Author is with the Defence Science and Technology Group, Department of Defence, Australia stuart.anstee@dst.defence.gov.au

performance trade-offs and computational drawbacks [19]. Related work that uses sampling-based planning includes application of the RRT algorithm [20], where expansions are biased to follow current flow. Our previous work presents an FMT\*-based energy-optimal algorithm in 3D where edges represent glider trim states computed using the shooting method [10].

Here, we propose a principled method to reduce the size of the control space when computing reachability between sample points. The use of numerical computation is drastically reduced compared to fully computing reachable sets, and reachability computation is performed relatively infrequently compared to graph-based methods with dense resolution. Our method thus supports long-range navigation as we demonstrate in this paper.

### III. BACKGROUND

#### A. Vehicle model

Suppose we represent the motion of a vehicle  $G$  by a continuous-time transition model

$$\dot{\mathbf{x}} = F_v(\mathbf{x}) + \mathbf{v}_g, \quad (1)$$

where  $\mathbf{x} = [x, y]^\top$  is the position of the glider in two-dimensional (2D) space,  $F_v(\mathbf{x}) = [u_f, v_f]^\top$  is the time-independent flow vector and  $\mathbf{v}_g = [u_g, v_g]^\top$  is the glider's velocity *relative* to the flow, which is bounded by maximum speed  $V_{max}$  such that

$$|\mathbf{v}_g| \leq V_{max}. \quad (2)$$

The continuous system (1) can be discretised with a small time step  $\Delta t$  as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (F_v(\mathbf{x}_k) + \mathbf{v}_{g,k}) \Delta t. \quad (3)$$

The discrete system is controlled by action  $a_k = \mathbf{v}_{g,k}$  at  $k$ -th time step. In a short form, we have  $f_x(\mathbf{x}_k, a_k)$ . The sequence of vehicle controls  $\sigma$  is denoted as

$$\sigma = a_0 a_1 \cdots a_{K-1}, \quad (4)$$

where  $K$  is the discrete time horizon. We denote  $f_x(\mathbf{x}, \sigma)$  as the resultant state after executing the control sequence  $\sigma$  from  $\mathbf{x}$ . Note that a control action  $a_k \in \mathcal{A}$  is constrained by the upper limit on the reference velocity  $\mathbf{v}_g$

$$\mathcal{A} = \{\mathbf{v} \in \mathbb{R}^2 \mid |\mathbf{v}| \leq V_{max}\}. \quad (5)$$

The cost of transiting from state  $\mathbf{x}$  with control  $a$  is denoted as  $f_c(\mathbf{x}, a)$ . The cost for control sequence  $\sigma$  from state  $\mathbf{x}$  is denoted as  $f_c(\mathbf{x}, \sigma)$ .

#### B. Stream functions

We consider a time-invariant, incompressible flow field [21]

$$\nabla \cdot F_v = 0, \quad (6)$$

where  $\nabla \cdot$  is the divergence operator. With this condition, which requires that density does not change with flow, a 2D flow field can be represented using the *stream function*  $\psi: \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ , which describes the flux that passes

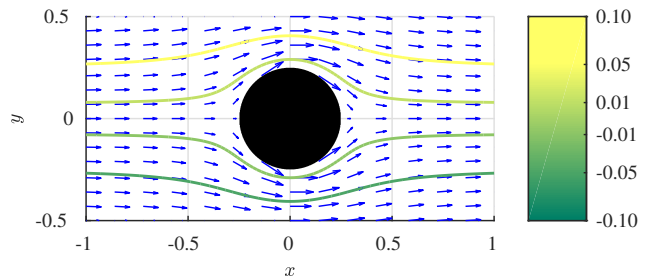


Fig. 1. Incompressible flow visualised by streamlines. The stream values of the streamlines refer to the flux (rate of flow) of the fluid flowing between each streamline and the arbitrary reference point  $(-1, 0)$ .

through a curve connecting two points in 2D space [22]. The value of the stream function  $\psi_{PQ}$ , or *stream value* between two points  $P$  and  $Q$  is

$$\psi(\mathbf{x}_P, \mathbf{x}_Q) = \int_{\mathbf{x}_P}^{\mathbf{x}_Q} (u_f(\mathbf{x}) dy - v_f(\mathbf{x}) dx). \quad (7)$$

If the flow field is incompressible (i.e. satisfies (6)), then the line integral (7) is well-defined in the sense that the value is the same for any curve connecting  $\mathbf{x}_P$  and  $\mathbf{x}_Q$ . A continuous set of points that have the same stream value relative to some reference point  $P$  is known as a *streamline*. The term is apt, because the stream value of an idle vehicle (i.e.  $\mathbf{v}_g = \mathbf{0}$ ) advected by the flow field remains constant. In other words, we have:

$$\frac{d}{dt} \psi(\mathbf{x}_P, \mathbf{x}_t) = -v_f u_f + u_f v_f = 0. \quad (8)$$

In Fig. 1, four arbitrary streamlines with different stream values illustrate the flow of fluid around a circular obstacle. Disjointed streamlines with the same stream value are referred to as being *distinct*.

**Remark 1** (Additive property of stream function). A useful property of incompressible flow fields is that they can be represented by the sum of their stream functions: given two incompressible flow fields  $F_A$  and  $F_B$  with corresponding stream functions  $\psi_A$  and  $\psi_B$ , the superimposed flow field  $F_{A+B}$  is

$$F_{A+B} = F_A + F_B, \quad (9)$$

and the corresponding stream value  $\psi_{A+B}$  is:

$$\psi_{A+B} = \psi_A + \psi_B. \quad (10)$$

This property becomes vital in Sec. V-A to gain deeper insight to the vehicle's net trajectory by considering the net stream function after treating the vehicle's relative velocity as a flow field.

### IV. PROBLEM STATEMENT

We consider a path planning problem for vehicle  $G$ . We seek an optimal control sequence  $\sigma^* \in \mathcal{A}^K$  over a cost function  $f_c$  from initial position  $\mathbf{x}_{init}$  to goal position  $\mathbf{x}_{goal}$  in the presence of an incompressible flow field  $F_v$ .

Motivated by the need to control underwater vehicles with limited energy capacity and connectivity [23], we are

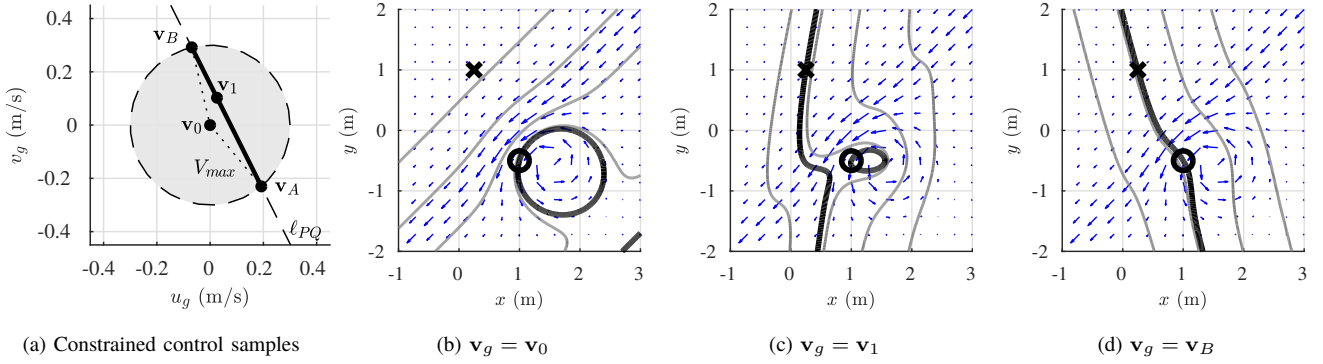


Fig. 2. Streamlines resulting from superimposing stream functions of the environmental flow and the vehicle's relative velocity.

interested in a scenario which minimises changes in control inputs to reduce energy expenditure that would otherwise be spent on computation. With this constraint, the vehicle executes a sequences of *persistent* controls  $\omega$

$$\omega = (a_0, \tau_0), \dots, (a_{K-1}, \tau_{K-1}), \quad (11)$$

where a *persistent control*  $\omega_i$  consists of a control action  $a_i \in \mathcal{A}$  with a duration  $\tau_i$ . The key difference between  $\sigma$  and  $\omega$  is that the actions resulting from  $\sigma$  are separated by constant sampling time  $\Delta t$ , whereas those from  $\omega$  coincide with decision points; for example, when a vehicle surfaces for a position fix. We slightly abuse the notation for cost such that  $f_c(\mathbf{x}_k, \omega_k)$  denotes the cost of executing persistent control  $\omega_k$  starting from  $\mathbf{x}_k$  and  $f_c(\mathbf{x}_{init}, \omega)$  denotes the cost of executing the entire sequence  $\omega$ . Similarly, we denote by  $f_x(\mathbf{x}_{init}, \omega)$  the resultant state after executing  $\omega$ . We assume that any control duration  $\tau_i$  is of sufficient length so that the control transition time from  $a_i$  and  $a_{i+1}$  is negligible.

The path planning problem with a persistent control sequence is formally defined as follows:

**Problem 1** (Energy-aware optimal path planning in a flow field). *Given a vehicle  $G$ , a set of controls  $\mathcal{A}$ , initial position  $\mathbf{x}_{init}$ , goal position  $\mathbf{x}_{goal}$ , and time-invariant incompressible flow field  $F_v$ , find an energy-aware disjoint control sequence  $\omega^*$  that minimises the overall cost from  $\mathbf{x}_{init}$  to  $\mathbf{x}_{goal}$  such that*

$$\omega^* = \arg \min_{\omega} \sum_{k=0}^{K-1} f_c(\mathbf{x}_k, \omega_k), \quad (12)$$

where  $\mathbf{x}_{k+1} = f_x(\mathbf{x}_k, \omega_k)$ ,  $\mathbf{x}_0 = \mathbf{x}_{init}$ , and  $\mathbf{x}_K = \mathbf{x}_{goal}$ .

This problem consists of two sub-problems: 1) find a persistent control connecting two arbitrary points in the flow field (if it exists), and 2) find a sequence that minimises the overall cost. Without loss of generality, we consider time cost in this paper.

The key aspect of this formulation is that it fits naturally with sampling-based planning methods such as PRM\* where samples are separated in space and edge connections are made between pairs of sample points. We show that our proposed method reduces a complexity bottleneck inherent in

making the edge connections in a flow field by significantly reducing the control space.

## V. STREAMLINE-BASED CONTROL SEARCH

In this section, we present a novel method for efficiently finding a persistent control between two arbitrary points in the presence of an incompressible time-invariant flow field. We significantly reduce the control space by exploiting the idea of stream functions.

### A. Finding control bound using stream function

The vehicle  $G$  is controlled by adapting its relative velocity  $\mathbf{v}_g$ . The relative velocity can be viewed as an additional flow acting on an idle vehicle. We denote such flow  $F_g = \mathbf{v}_g$  and we call  $F_g$  the *flow due to control*. Formally, the vehicle is in an equivalent reference frame with a flow field  $(F_v + F_g)$  and control  $|\bar{\mathbf{v}}_g| = 0$ . From (7), for *constant* relative velocity  $\mathbf{v}_g$ , the stream value for  $F_g$  between two points  $P$  and  $Q$  is simply

$$\psi_g(\mathbf{x}_P, \mathbf{x}_Q) = u_g \Delta y - v_g \Delta x, \quad (13)$$

where  $\Delta x = x_Q - x_P$  and  $\Delta y = y_Q - y_P$ .

By Remark 1, the superimposed stream value for external flow  $F_v$  and the flow due to vehicle motion  $F_g$  is

$$\psi_{PQ} \equiv \psi(\mathbf{x}_P, \mathbf{x}_Q) = \psi_v(\mathbf{x}_P, \mathbf{x}_Q) + u_g \Delta y - v_g \Delta x. \quad (14)$$

The aim is to find a control  $\mathbf{v}_g$  such that two points  $P$  and  $Q$  are on the same streamline in the superimposed flows, that is,  $\psi_{PQ} = 0$  or

$$\psi_v(\mathbf{x}_P, \mathbf{x}_Q) + u_g \Delta y - v_g \Delta x = 0. \quad (15)$$

Note that all points on the same streamline have the same stream value, but the converse need not apply, since points with the same stream value may reside on distinct streamlines as shown in Fig. 2c.

In (15), we defined a line in the control space over  $u_g$  and  $v_g$  that connects points  $P$  and  $Q$ . We denote it by  $\ell_{PQ}$ . The streamline-based set of *feasible* controls for manoeuvring from  $P$  to  $Q$  is given as

$$\mathcal{A}_{PQ} = \{\mathbf{v} \in \ell_{PQ} \mid |\mathbf{v}| \leq V_{max}\}. \quad (16)$$

Importantly, with streamline-based control, we need only search a one-dimensional (1D) control space; in our previous

approach [10], which was based on (5), the space was 2D. We discuss the reduction in complexity in Sec. VI.

Since the set of controls satisfying  $|\mathbf{v}_g| \leq V_{max}$  is convex and  $\ell_{PQ}$  is linear in control space, three sets of controls are possible depending on the number of intersections between the two conditions: 1) the empty set if there is no intersection, 2) a set with one element if there is only one intersection, and 3) a set of infinite solutions between two intersections. Defining the intersection points in control space as *endpoints*  $\mathbf{v}_A$  and  $\mathbf{v}_B$ , we find

$$\begin{aligned} \mathbf{v}_A &= V_{max} [\cos \theta_A, \sin \theta_A]^\top \\ \mathbf{v}_B &= V_{max} [\cos \theta_B, \sin \theta_B]^\top, \end{aligned} \quad (17)$$

where

$$\begin{aligned} \theta_A &= \delta + \frac{\pi}{2} + \arccos(\kappa) \\ \theta_B &= \delta + \frac{\pi}{2} - \arccos(\kappa) \\ \delta &= \text{atan2}(\Delta y, \Delta x) \\ \kappa &= \frac{\psi_v(\mathbf{x}_P, \mathbf{x}_Q)}{V_{max} \|\mathbf{x}_Q - \mathbf{x}_P\|_2} \end{aligned} \quad (18)$$

Intuitively, the set of feasible controls lies along the straight line between the endpoints.

In Fig. 2, we illustrate the streamlines for different controls  $\mathbf{v}_g \in \mathcal{A}_{PQ}$ . The control space shown in Fig. 2a includes the maximum speed constraint  $|\mathbf{v}_g| \leq V_{max}$ , the control line  $\ell_{PQ}$ , and the endpoints  $\mathbf{v}_A$  and  $\mathbf{v}_B$  of its feasible subset. In Fig. 2b, the control  $\mathbf{v}_g = \mathbf{v}_0$  is not on the control line  $\ell_{PQ}$ , so the points  $P$  and  $Q$  have different stream values and one is not reachable from the other. In Fig. 2c, the control is on the control line and points  $P$  and  $Q$  have the same stream value but are on distinct streamlines. In Fig. 2d, the control is on the control line and point  $Q$  is downstream from  $P$ , and is therefore reachable.

### B. Finding critical points in a control set

We have shown that endpoints  $\mathbf{v}_A$  and  $\mathbf{v}_B$  define a set of feasible controls  $\mathcal{A}_{PQ}$  that guarantees points  $P$  and  $Q$  have the same stream values. However, as Fig. 2c shows, the endpoint-based method does not guarantee that they are on the same streamline. Note that the points are guaranteed to be unreachable if they have different stream values.

It turns out that distinct streamlines originate at saddle points in the flow. This can be shown through *Morse theory* [24], the study of critical points of a smooth function. Given two points  $P$  and  $Q$  in state space and flow field  $F$ , a point  $\mathbf{c}$  is a saddle point iff [24]

$$F(\mathbf{c}) = 0 \text{ and } \lambda_1 \lambda_2 < 0, \quad (19)$$

where  $\lambda_1$  and  $\lambda_2$  are the eigenvalues of the Hessian matrix  $\Delta\psi(\mathbf{c})$ . Intuitively, a point is a saddle point iff the flow at the point is idle but is at neither a maximum nor a minimum. A result from Morse theory is that level sets, such as the streamlines we consider in this paper, are *connected* if there are no saddle points [24]; that is, the sampling strategy (18) then guarantees that the vehicle will travel from  $P$  to  $Q$ . A simple way to check for the existence of a saddle point is to

check the magnitude of the net velocity. If it is close to zero and the determinant of the Hessian of the stream function is negative, then a saddle point exists near the trajectory. In physical terms, the saddle points imply that the vehicle comes to a ‘stall’. This is clearly undesirable in either time- or energy- optimal case.

The condition (19) thus implies that two points  $P$  and  $Q$  lie on the same streamline if the net velocity of the vehicle with respect to absolute reference frame (i.e.,  $F_v + F_g$ ) is sufficiently large over the streamline (i.e., trajectory) from  $P$  to  $Q$ . Importantly, this serves as a stopping condition that further reduces the time complexity.

### C. Control space sampling

The reduced control space and the saddle point-based stopping condition (19) allow us to find the control  $a^* \in \mathcal{A}$  that minimises the objective function. Two steps are required to find the optimal control between two states: control sampling and forward integration.

Given a control line  $\ell_{PQ}$ , we linearly sample  $C$  controls between and inclusive of endpoints  $\mathbf{v}_A$  and  $\mathbf{v}_B$ . For each control, we find a continuous trajectory over state space by forward integrating the state of the vehicle from point  $P$  to  $Q$  based on (3). We continue the integration until any of the following conditions holds true: 1) the vehicle is near  $Q$  (i.e., destination reached), 2) the integration has exceeded a specified maximum time horizon  $H$ , or 3) the vehicle has reached a saddle point (i.e., an infeasible control). Once we complete the enumeration over a set of controls for a given state sample pair, we find the control that minimises the travel time among the set of controls that reached the destination.

### D. Streamline-based motion planning

We present a high-level analysis of a *probabilistic roadmap\** (PRM\*) approach to path planning in a flow field, in which the evaluation of edge costs for pairs of points in state space is a computational bottleneck. Other sampling-based motion planners such as RRT\* and FMT\* would show similar reductions in edge cost complexity.

The PRM\* algorithm randomly selects pairs of samples in state space. For each directed pair of state samples  $P$  and  $Q$ , we find the edge time cost  $\text{Cost}(\mathbf{x}_P, \mathbf{x}_Q)$  using the streamline-based method. If there exists no solution for a pair, the samples are not connected. Once we find all the necessary edge costs, we use Dijkstra’s algorithm over the graph to find an optimal overall path from  $\mathbf{x}_{init}$  to  $\mathbf{x}_{goal}$ .

## VI. ANALYSIS

Given the number of state samples  $N$ , the number of control samples for a state sample pair  $C$  and the time horizon for forward integration  $H$ , the worst-case time complexity of the framework is  $\mathcal{O}(N^2 \cdot C \cdot H)$ . Although the complexity for the standard shooting method and the proposed method are similar, there are fundamental differences that make our framework significantly more efficient.

In our framework, controls sampled from a 1D line segment are guaranteed to have the same stream values. In

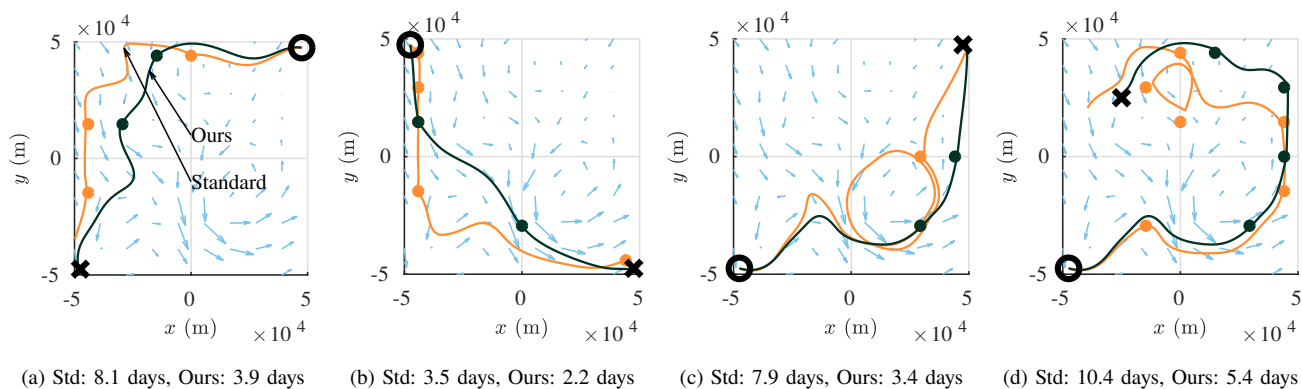


Fig. 3. Minimum time trajectories using the PRM\* with streamline-based control search (dark green) and standard control search (orange) for a vehicle (circle) to reach its goal (cross). Intermediate waypoints (dots) are used to achieve this despite the limited speed of vehicle.  $N = 49$ ,  $C = 19$

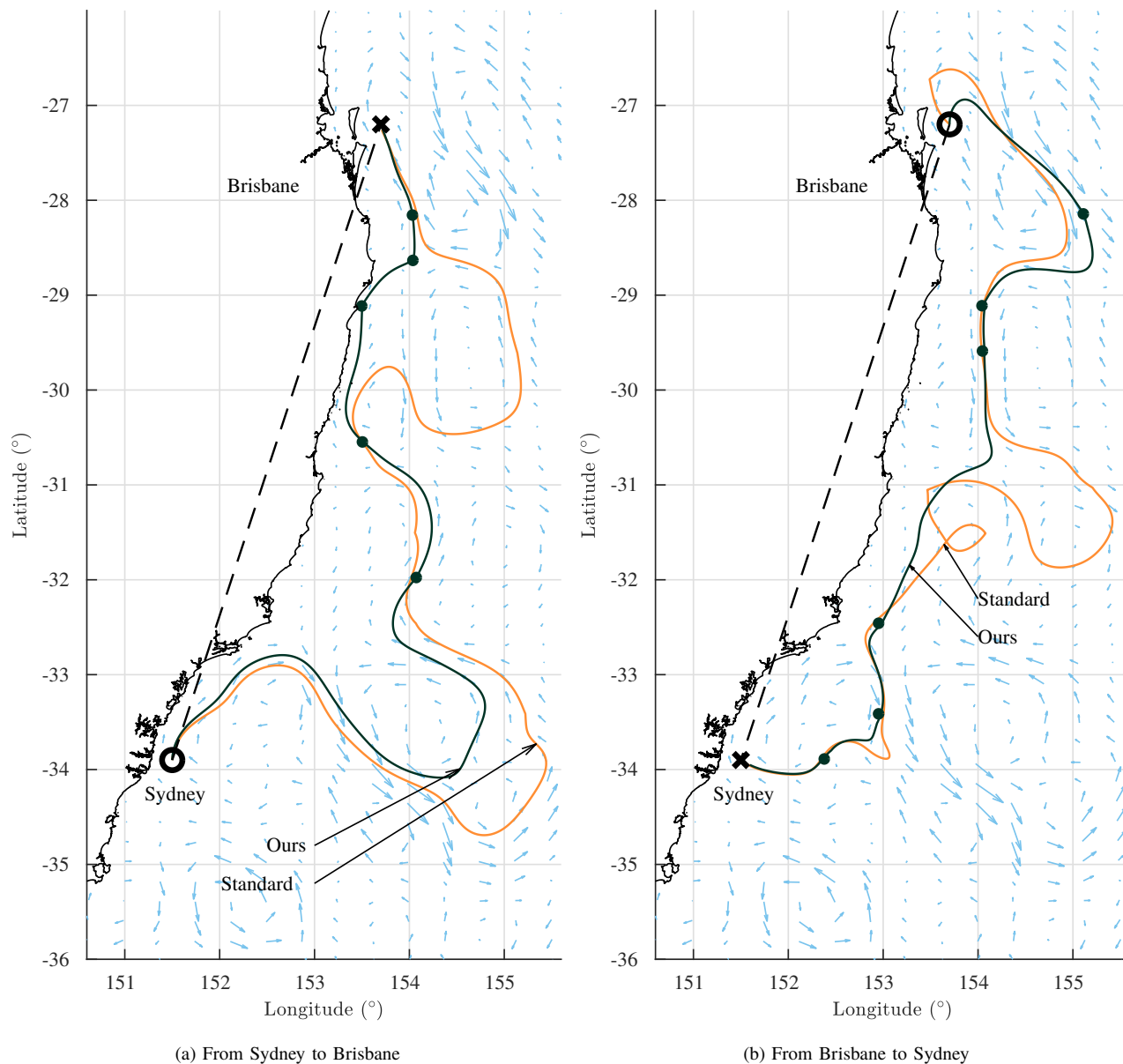


Fig. 4. PRM\* on ocean currents based on real dataset, starting at the circle and finishing at the cross. Waypoints for the streamline-based method (in dark green) are marked with dots, the waypoints for the standard shooting method (in orange) are similarly spaced.

contrast, the shooting method samples controls from a 2D control space bounded by the maximum speed. Because a state is guaranteed unreachable from another if the states do not have the same stream value, most of the control samples evaluated using the standard shooting are invalid. As a result, significantly fewer control samples fall on the control line  $\ell_{PQ}$  leading to numerous unnecessary forward integrations, substantially increasing the computation time.

An interesting empirical observation is that the time-optimal control for a given state pair seems to be the control with the maximum speed. In other words, the optimal control is at one of the endpoints  $\mathbf{v}_A$  or  $\mathbf{v}_B$ . If the special case is correct, then the overall time complexity becomes  $\mathcal{O}(N^2 \cdot H)$ .

## VII. CASE STUDIES

In this section, we demonstrate our implementation of the streamline-based motion planning method with cases. The first employs a simulated environment where we compare our framework against the standard to significant difference in performance. We then use our framework to find time-optimal solution using a real ocean dataset provided by the Australian Bureau of Meteorology (BoM). We show that our method works efficiently over a large scale with challenging ocean currents. For both cases, we consider a vehicle with maximum speed of  $0.3 \text{ ms}^{-1}$ , time horizon  $H$  of 2000 steps, and step size of 750 s.

### A. Simulated environment

In Fig. 3, we have four pairs of starting and goal positions (i.e. from circle to cross) in a simulated environment where the maximum flow magnitude is  $1 \text{ ms}^{-1}$ , which is well beyond the vehicle’s maximum speed. We compare our streamline-based method (in orange) against the standard shooting method (in dark green) with sampling parameters  $N = 49$  and  $C = 19$ .

For all scenarios, our method found significantly better solutions than the standard shooting method. This is because we consider a more focussed set of samples in control space that increases the chance of finding an optimal solution. In general, the streamline-based solutions exploit the currents to travel faster, whereas the standard shooting-based solutions are less effective in doing so. As discussed in Sec. VI, we found that the time-optimal controls lie at the endpoints. If this endpoint hypothesis is true, the computation time could be reduced by eight times in this case.

### B. Eastern Australian Currents

We demonstrate the use of the streamline-based framework to find time-optimal paths between Sydney and Brisbane. It is important to note that the region is very challenging to operate a vehicle with relatively slow speed; there exists strong southward currents and a number of eddies where the vehicle simply cannot pass or would otherwise get trapped.

The dataset provided by BoM is generated by an ocean model using a series of satellite images measuring the ocean heights. In this case study, we use a dataset that estimates the currents on 5th September 2018. From the dataset,

we numerically computed the stream values using (7) by assuming that the ocean flow is incompressible. Note that the dataset includes ocean flows more than 7 times faster than the vehicle’s maximum speed.

The resulting paths between Sydney and Brisbane are shown in Fig. 4. We also compare our streamline-based method (in orange) against the standard shooting-based method (in dark green). For both methods, we use the sampling parameters  $N = 210$  and  $C = 19$ .

For the path from Sydney to Brisbane in Fig. 4b, both methods seem to take advantage of the currents (bottom right) before moving towards Brisbane. Although the travelled distance is similar, the streamline-based method took 17 days to reach Brisbane whereas the standard shooting-based method took 22.8 days. The vehicle took almost the same time (17.6 and 29.4 days, respectively) to travel back from Brisbane to Sydney shown in Fig. 4b. For the purpose of comparing the results, the benchmark time it would take to travel along a straight line between Sydney and Brisbane in still water is 29.8 days. Our method clearly took a more efficient path, whereas the standard shooting-based method performed as poorly as this benchmark.

An important aspect of the proposed framework is that the control stays the same between two adjacent state samples (i.e. waypoints). Intuitively, once we set the control from the starting state, we let the vehicle go until it reaches the destination in the absence of active control. This aspect is important, especially for underwater platforms where the energy capacity is limited and the energy consumption is related to changes in control [10]. From Sydney to Brisbane, the vehicle changes its control only 5 times over 17 days. Graph-based methods, for example, would potentially apply a new control input at every time step.

## VIII. CONCLUSION AND FUTURE WORK

We have presented an algorithm to efficiently find a stream function induced by a control input superimposed on a flow field, such that two given points are connected by a streamline. We showed how this method can be integrated with a sampling-based algorithm to plan long-range paths for underwater vehicles in real-world ocean currents.

One limitation of our approach so far is that it is restricted to 2D planning. It is useful for gliders, for example, because typical onboard software accepts 2D waypoints as input and generates a depth profile automatically. However, stream functions are also defined in three dimensions, and it would be interesting to extend our approach in this way.

We have designed our method to be easily integrated with other sampling-based algorithms. Implementing FMT\* or BIT\*, for example, would be interesting avenues to pursue, in addition to performing field experiments with gliders and other types of AUVs. Beyond motion planning, the proposed method can be used in task planning for vehicles in flow fields [25, 26], which would benefit from the reduced complexity.

## REFERENCES

- [1] D. L. Rudnick, R. E. Davis, C. C. Eriksen, D. M. Fratantoni, and M. J. Perry, "Underwater gliders for ocean research," *Marine Technology Society Journal*, vol. 38, no. 2, pp. 73–84, 2004.
- [2] L. M. Russell-Cargill, B. S. Craddock, R. B. Dinsdale, J. G. Doran, B. N. Hunt, and B. Hollings, "Using Autonomous Underwater Gliders for Geochemical Exploration Surveys," *The APPEA Journal*, vol. 58, pp. 367–380, 2018.
- [3] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard, "Imaging sonar-aided navigation for autonomous underwater harbor surveillance," in *Proc. of IEEE/RSJ IROS*, 2010, pp. 4396–4403.
- [4] Argo, "Argo float data and metadata from Global Data Assembly Centre (Argo GDAC)," 2000. [Online]. Available: <http://doi.org/10.17882/42182>
- [5] D. C. Webb, P. J. Simonetti, and C. P. Jones, "SLOCUM: An underwater glider propelled by environmental energy," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 447–452, 2001.
- [6] R. Bachmayer, N. E. Leonard, J. Graver, E. Fiorelli, P. Bhatta, and D. Paley, "Underwater gliders: Recent developments and future applications," in *Proc. of Underwater Technology*, 2004, pp. 195–200.
- [7] R. P. Stokey, A. Roup, C. von Alt, B. Allen, N. Forrester, T. Austin, R. Goldsborough, M. Purcell, F. Jaffre, G. Packard, and A. Kukulya, "Development of the REMUS 600 autonomous underwater vehicle," in *Proc. of IEEE OCEANS*, 2005, pp. 1301–1304.
- [8] B. Claus, R. Bachmayer, and C. D. Williams, "Development of an auxiliary propulsion module for an autonomous underwater glider," *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 224, no. 4, pp. 255–266, 2010.
- [9] E. Zermelo, *Über das Navigationsproblem bei ruhender oder veränderlicher Windverteilung*. WILEY-VCH Verlag, 1931, vol. 11, no. 2.
- [10] J. J. H. Lee, C. Yoo, R. Hall, S. Anstee, and R. Fitch, "Energy-optimal kinodynamic planning for underwater gliders in flow fields," in *Proc. of ARAA ACRA*, 2017.
- [11] P. R. Oke, A. Schiller, D. A. Griffin, and G. B. Brassington, "Ensemble data assimilation for an eddy-resolving ocean model of the Australian region," *Quarterly Journal of the Royal Meteorological Society*, vol. 131, no. 613, pp. 3301–3311, 2005.
- [12] P. R. Oke, D. A. Griffin, A. Schiller, R. J. Matear, R. Fiedler, J. Mansbridge, A. Lenton, M. Cahill, M. A. Chamberlain, and K. Ridgway, "Evaluation of a near-global eddy-resolving ocean model," *Geoscientific Model Development*, vol. 6, no. 3, p. 591, 2013.
- [13] A. F. Shchepetkin and J. C. McWilliams, "The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean Modelling*, vol. 9, no. 4, pp. 347–404, 2005.
- [14] D. Chang, W. Wu, C. R. Edwards, and F. Zhang, "Motion tomography: Mapping flow fields using autonomous underwater vehicles," *International Journal of Robotics Research*, vol. 36, no. 3, pp. 320–336, 2017.
- [15] T. Lolla, P. F. J. Lermusiaux, M. P. Ueckermann, and P. J. Haley, "Time-optimal path planning in dynamic flows using level set equations : theory and schemes," *Ocean Dynamics*, pp. 1373–1397, 2014.
- [16] D. N. Subramani and P. F. J. Lermusiaux, "Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization," *Ocean Modelling*, vol. 100, pp. 57–77, 2016.
- [17] D. Kularatne, S. Bhattacharya, and M. Ani Hsieh, "Time and energy optimal path planning in general flows," in *Proc. of RSS*, 2016.
- [18] D. Kularatne, S. Bhattacharya, and M. A. Hsieh, "Going with the flow: a graph based approach to optimal path planning in general flows," *Autonomous Robots*, vol. 42, no. 7, pp. 1369–1387, 2018.
- [19] S. M. LaValle, *Planning Algorithms*, 2006.
- [20] I. Ko, B. Kim, and F. C. Park, "Randomized path planning on vector fields," *International Journal of Robotics Research*, vol. 33, no. 13, pp. 1664–1682, 2014.
- [21] P. J. Pritchard and J. W. Mitchell, *Fox and McDonald's Introduction to Fluid Mechanics*, 8th ed. Wiley, 2011.
- [22] G. K. Batchelor, *An introduction to fluid dynamics*. Cambridge University Press, 1967.
- [23] M. R. Dhanak and N. I. Xiros, *Springer Handbook of Ocean Engineering*. Springer, 2016.
- [24] K. Smith, "Connectivity and convexity properties of the momentum map for group actions on Hilbert Manifolds," Ph.D. dissertation, University of Toronto, 2013.
- [25] C. Yoo, R. Fitch, and S. Sukkarieh, "Online task planning and control for fuel-constrained aerial robots in wind fields," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 438–453, 2016.
- [26] ———, "Probabilistic temporal logic for motion planning with resource threshold constraints," in *Proc. of RSS*, 2012.