Time and Energy Optimal Path Planning in General Flows

Dhanushka Kularatne*, Subhrajit Bhattacharya† and M. Ani Hsieh*
*Drexel Universiy, Philadelphia, PA 19104
†Universiy of Pennsylvania, Philadelphia, PA 19104

Abstract—**Autonomous surface and underwater vehicles (ASVs and AUVs) are increasingly being used for persistent monitoring of ocean phenomena. Typically, these vehicles are deployed for long periods of time and must operate with limited energy budgets. As a result, there is increased interest in recent years on developing energy efficient motion plans for these vehicles that leverage the dynamics of the surrounding flow field. In this paper, we present a graph search based method to plan time and energy optimal paths in a flow field where the kinematic actuation constraints on the vehicles are captured in our cost functions. We also use tools from topological path planning to generate optimal paths in different homotopy classes, which facilitates simultaneous exploration of the environment. The proposed strategy is validated using analytical flow models for large scale ocean circulation and in experiments using an indoor laboratory testbed capable of creating flows with ocean-like features. We also present a Riemannian metric based approximation for these cost functions which provides an alternative method for computing time and energy optimal paths. The Riemannian approximation results in smoother trajectories in contrast to the graph based approach while requiring less computational time.**

I. INTRODUCTION

To better understand oceanic processes, researchers are employing autonomous underwater and surface vehicles (AUVs and ASVs) for long-term surveillance of the dynamics of plankton assemblages [5], temperature and salinity profiles [19, 29, 27], and the onset of harmful algae blooms [6, 9]. In these and similar environmental monitoring applications in the ocean, AUVs and ASVs are often deployed over long periods while operating with limited energy budgets. As such, there is increased interest in recent years on developing more energy efficient motion and trajectory plans for AUVs and ASVs.

In recent years, researchers have demonstrated how AUV/ASV motion planning and adaptive sampling strategies can be improved by incorporating either historical or current ocean flow data [24, 26, 25]. In [24, 26], Smith *et al.* rely on regional ocean model systems (ROMS) to predict the dynamics of an evolving ocean front and uses the resulting predictions to generate waypoints for the AUV to enable it to track the feature of interest. The approach was then integrated with an unscented Kalman filter to better estimate the vehicle's dead-reckoning error along a given path in [25]. Path planning for an AUV tasked with an adaptive sampling task was achieved in [8] by employing Lagrangian drifters to drift along a "patch" of water of interest. Rather than rely on ROMS data, Das *et al.* effectively achieves real-time sampling of the ocean currents in the region of interest using the drifters. Wu *et al.* solves the inverse problem in [30] where differences in actual and predicted AUV surfacing positions are leveraged to estimate the at-depth flow field. In all these works, the idea is to leverage available ocean data to improve the selection of future vehicle sampling locations, thus minimizing the vehicle's energy expenditure by avoiding navigation to erroneous or low information value regions.

While the high inertia environment of the ocean couples the environmental dynamics to the dynamics of the vehicles that operate in them, it presents a unique opportunity for vehicles to exploit these surrounding flow dynamics for more efficient navigation. However, the exploitation of the surrounding flow field in the planning of optimal trajectories is an extremely challenging endeavor since it requires full knowledge of the flow field in both space and time. Our limited understanding of the complexities of ocean dynamics means that in general accessibility to and the overall quality of the flow data and/or numerical models is highly dependent on how well a given region of interest is instrumented. While it is possible to obtain ocean current hindcasts, nowcasts, and forecasts from Naval Coastal Ocean Model (NCOM) databases [1] and regional ocean model systems (ROMS) [26], the data is assimilated from satellite and field observations in conjunction with predictions from numerical models [22, 23]. And since numerical partial differential equation models are derived through a combination of theoretical and field observations, existing data sets are mostly finite-time, of low spatio-temporal resolution, and limited to specific regions.

Recently, a level-set approach for AUV/ASV path planning where the external flow was explicitly accounted for was described in [16, 17]. In the level-set approach, time-optimal trajectories for single and multiple vehicles are determined by employing a level set expansion of the flow field given the desired start and goal positions for the vehicle(s). The underlying motion model behind the level-set method assumes that the net velocity of the vehicle is composed of the flow velocity component and a velocity component orthogonal to it (due to actuation). This velocity model is, very often, restrictive. Furthermore, such a model does not take into account kinematic constraints of the vehicle, and provides limited flexibility in the choice of the optimization objective. Since the strategy requires full knowledge of the flow field and requires significant computational resources when performing the various level set expansions, the strategy is mostly applicable for pre-deployment planning purposes and not amenable for real-time planning purposes. Graph-based strategies for computing
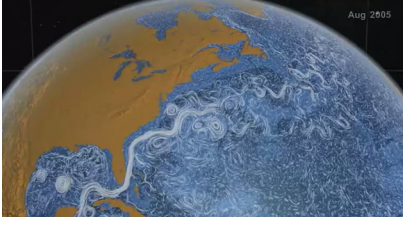
Fig. 1. Snapshot (August 2005) of visualization of ocean surface currents for June 2005 through December 2007 generated using NASA/JPLs Estimating the Circulation and Climate of the Ocean, Phase II (ECCO2) ocean model.

Fig. 2. (a) The flow vector, $\mathbf{V}$, at a point, p, and the flow-parallel coordinate system. (b) The net velocity of the vehicle is the vector sum of the field velocity and the vehicle's still-water velocity.

minimum energy paths for marine vehicles subject to external flow conditions include [10, 11, 15, 14]. These works mostly employ a variant of A^* with a suitably selected energy or time based heuristic. However, these approaches are generally restrictive in their optimization objective, are computationally expensive, and/or too simplistic in their assumptions about the dynamics of the fluid environment.

In this work, we present a graph-based path planning approach for computing time and energy optimal paths for vehicles operating in a general flow field. Similar to [16, 17, 10, 15, 14, 11], the strategy leverages the surrounding flow field in the synthesis of optimal trajectories. Different from these existing strategies, we employ graph search-based methods coupled with more accurate cost functions in computing the optimal trajectories. In particular, the primary merits of our method are the following:

(a) We are able to design cost functions for obtaining both time and energy optimal trajectories. More complex optimization objectives can also be constructed.
(b) In our method we are able to impose kinematic constraints such as maximum still-water speed of the vehicle.
(c) We provide a Riemannian approximation of our cost function so as to be able to model them as Riemannian metrics. This allows us to develop insights into the problem as well as pose the optimization problem as one of solving the *geodesic equation* on a Riemannian manifold.
(d) Lastly, large scale ocean circulation often exhibit significant eddy and jet structures (see Fig. 1), which gives rise to the presence of fixed points and invariant manifolds in the flow field. Coupled with the presence of obstacles, *e.g.*, islands and archipelagos, the result is multiple topological classes of trajectories. Recent work showed the importance of reasoning about such topological classes in marine operations in the presence of obstacles [4]. As such, a *fundamental advantage* of our proposed graph-based strategy over existing ones [10, 15, 14, 11] is the ability to use computational tools from *topological path planning* in the computation of time/energy optimal paths in different homotopy classes of the environment.

The rest of the paper is organized as follows: in Section II we present the preliminaries of the problem and its formulation, in Section III we describe the methods used to solve the optimization problem, Sections IV and V describe the simulation and experimental results respectively and we conclude with a discussion of our findings and directions for
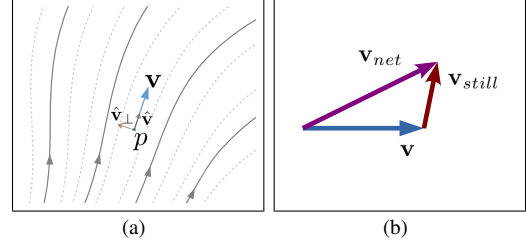
future work in Section VI.

II. PROBLEM FORMULATION

We begin with a brief description of our assumptions and then describe the development of our time and energy cost functions. We conclude this section with a formal statement of our problem.

A. Flow Model

In this work, we consider the planning of time and/or energy optimal paths in a two-dimensional (2D) planar flow field of the form

$$\dot{\mathbf{p}}(t) = \mathbf{v}(t) = \mathbf{F}(\mathbf{p}) \qquad (1)$$

where $\mathbf{p} \in W$ denotes the position in the 2D workspace $W \subseteq \mathbb{R}^2$. Then $\mathbf{v}$ denotes the velocity of the flow field at $\mathbf{p}$ at a given time t. In the case when the flow field is time-varying, (1) becomes $\mathbf{v} = \mathbf{F}(\mathbf{p}, t)$ with $(\mathbf{p}, t) \in W_T \subseteq \mathbb{R}^3$ where the third dimension denotes time. We employ a *flow-parallel* coordinate representation of the vector field given by (1), such that for every point $p \in W$, its axes are aligned along the unit vectors $\hat{\mathbf{v}}_{\mathbf{p}} = \frac{\mathbf{F}(p,t)}{\|\mathbf{F}(p,t)\|}$ (the "x" axis) and $\hat{\mathbf{v}}_{\mathbf{p}\perp}$ (the "y" axis, orthogonal to the x axis – see Fig. 2(a)). We note that this coordinate system is not "co-moving" with the flow and thus for simplicity we will drop the $\mathbf{p}$ subscript wherever it is possible to do so without ambiguity.

Thus, in the flow-parallel coordinate system, the flow vector at $\mathbf{p}$ can simply be written as $\mathbf{v} = v\hat{\mathbf{v}}$, where $v = \|\mathbf{v}\|$ is the speed of the flow at $\mathbf{p}$. With a little abuse of notation, we will also write the coordinate representation of $\mathbf{v}$ in the flow-parallel coordinate system as $\mathbf{v} = [v, 0]^T$.

B. Vehicle Kinematics

We assume the following 2D kinematic model for the planar autonomous underwater/surface vehicle:

$$\dot{X} = v_{still} \cos\theta + v_X, \quad \dot{Y} = v_{still} \sin\theta + v_Y \qquad (2)$$

where X and Y denote the global Euclidean coordinates of the vehicle (in contrast to x and y denoting the local flow-parallel coordinates), $v_{still} = \|\mathbf{v}_{still}\|$ denotes the speed/velocity of the vehicle in still water (relative to the flow), and θ denotes the angle of $\mathbf{v}_{still}$ with respect to the global X axis.

In general, a vehicle which travels infinitesimal distances, dx and dy, along $\hat{\mathbf{v}}_{\mathbf{p}}$ and $\hat{\mathbf{v}}_{\mathbf{p}\perp}$ respectively in time dt, has a net velocity $\mathbf{v}_{net} = [dx, dy]^T/dt$ (expressed in the components of the flow-parallel coordinates). If $\mathbf{v}_{still}$ is the velocity of the vehicle in still water, its flow-parallel coordinate representation

is $\mathbf{v}_{still} = \mathbf{v}_{net} - \mathbf{v} = [\frac{dx}{dt} - v, \frac{dy}{dt}]^T$ (see Fig. 2(b)). Thus the speed of the vehicle in still water is given by

$$v_{still} = \|\mathbf{v}_{still}\| = \sqrt{\left(\frac{dx}{dt} - v\right)^2 + \left(\frac{dy}{dt}\right)^2}. \qquad (3)$$

As such, $\dot{X}$ and $\dot{Y}$ in (2) are the components of $\mathbf{v}_{net}$ in the global coordinate frame with v_X and v_Y denoting the X and Y components of the background flow velocity respectively. In this work, rather than focusing on the non-holonomic constraints enforced by the vehicle kinematics given by (2), we instead focus on the kinematic actuation constraints of the vehicle at the trajectory planning stage. This is because for energy constrained vehicles deployed in geophysical fluid environments like the ocean, the environmental dynamics tend to dominate. As such, we assume that the actuation capability of the vehicle is limited and slower than the surrounding flow ($v_{still} < V_{max} < \max_{\mathbf{p} \in W} v_{\mathbf{p}}$) and that the vehicle's lower level controllers can achieve the desired still water speed dictated by the trajectory planner.

C. Cost Functions

We will assign (infinitesimal) costs to infinitesimal displacements of the vehicle. Thus, in general, it is a function of differentials, dx and dy (where x and y are the flow-parallel coordinates), and is of the form $dc_2 = f(dx, dy)$. However, if necessary, our framework lets us incorporate the differential time element (time required to produce the differential displacement) in the function, in which case the form will be $dc_3 = f(dx, dy, dt)$. This also lets us define cost functions and thus find optimal trajectories in time-varying flow fields, although we will primarily focus on time-invariant flows in this work.

1) Time Minimizing Cost: If we assume $v_{still} = V_{max}$, a fixed maximum still-water speed for the vehicle, we can solve for dt using (3) as follows,

$$\left(\left(\frac{dx}{dt} - v\right)^2 + \left(\frac{dy}{dt}\right)^2\right) = V_{max}^2$$

$$\Rightarrow dx^2 - 2vdxdt + (v^2 - V_{max}^2)dt^2 + dy^2 = 0$$

$$\Rightarrow dt = \frac{v}{v^2 - V_{max}^2}dx - \frac{\sqrt{V_{max}^2(dx^2 + dy^2) - v^2dy^2}}{v^2 - V_{max}^2}. \qquad (4)$$

Note that we discard the solution with the positive root since when $dy = 0$, we should have $dt = dx/(v + V_{max})$.

Thus, the time-minimizing cost for a differential elements in W is

$$dc_{2,time} = \frac{v}{v^2 - V_{max}^2}dx - \frac{\sqrt{V_{max}^2(dx^2 + dy^2) - v^2dy^2}}{v^2 - V_{max}^2}. \qquad (5)$$

The corresponding still-water velocity of the vehicle will be $\mathbf{v}_{still,time} = [\frac{dx}{dt} - v, \frac{dy}{dt}]^T$, where dt is given by (4).

Without imposing any velocity restrictions when computing time optimal trajectories, the differential cost of a differential element in W_T would simply be the time taken to traverse the differential element: $dc_{3,time} = dt$.

2) Energy Minimizing Cost: The differential cost in this case is the energy expended in achieving a differential displacement, $[dx, dy]^T$ in time dt. If $\mathbf{v}_{still}$ is the still-water velocity of the vehicle required for achieving this displacement, the drag force generated is assumed to be $\mathbf{F} = \kappa\mathbf{v}_{still}$ (where κ is the drag coefficient) [20], and the displacement in still water is $d\mathbf{x}_{still} = \mathbf{v}_{still}dt$. Thus, using (3), the cost of a differential element in W_T is

$$dc_{3,energy} = \mathbf{F} \cdot d\mathbf{x}_{still} = \kappa\|\mathbf{v}_{still}\|^2 dt$$

$$= \kappa\left(\left(\frac{dx}{dt} - v\right)^2 + \left(\frac{dy}{dt}\right)^2\right)dt$$

$$= \kappa\left(\frac{dx^2 + dy^2}{dt} + v^2dt - 2vdx\right) \qquad (6)$$

If we fix dx and dy, the differential displacement, and let the vehicle choose any velocity so as to minimize the differential cost, it is easy to check that the cost given in (6) is minimized when $dt = \frac{\sqrt{dx^2 + dy^2}}{v}$. Now since we can effectively remove the time dimension from the cost function, the cost of a differential element in W is

$$dc_{2,energy} = 2\kappa v\left(\sqrt{dx^2 + dy^2} - dx\right) \qquad (7)$$

and the corresponding still-water velocity of the vehicle is,

$$\mathbf{v}_{still,energy} = v\left[\left(\frac{dx}{\sqrt{dx^2 + dy^2}} - 1\right), \frac{dy}{\sqrt{dx^2 + dy^2}}\right]^T. \qquad (8)$$

D. Problem Statement

Given the above time and energy cost functions, the objective is to find paths that minimizes the respective total costs. As such, the objective of this work is to find a path $\gamma\star$, such that it is the solution to the following optimization problem:

$$\min_{\gamma} \int_{\gamma} dc_{\alpha,\beta} \qquad (9)$$

$$\text{subject to} \quad \gamma(\tau_0) = \mathbf{p}_s, \quad \gamma(\tau_g) = \mathbf{p}_g, \quad v_{still} \leq V_{max}.$$

Here, $\mathbf{p}_s$ and $\mathbf{p}_g$ are the desired start and goal positions, α denotes the dimension of the workspace, and β denotes either *time* or *energy.*. When $\alpha = 2$, the infinitesimal element belongs to W and when $\alpha = 3$ it belongs to W_T. We present our graph based search strategies for computing $\gamma\star$ in the following section.

III. ALGORITHM

A. Graph Search-based Optimal Trajectory Planning

We use a discrete graph, $\mathcal{G} = (V, E)$, to represent the workspace W (or W_T). Vertices in this graph are centroids of cells in a uniform square/cubic discretization (in the global coordinates), and edges are established between immediate neighbors as well as some 1-hop neighbors.
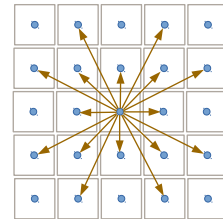

Fig. 3. The connectivity of a vertex with other neighboring vertices in the graph $\mathcal{G}$.

In particular, for the planar case, each vertex is connected to 16 neighboring vertices as illustrated in Figure 3. Each edge in the graph can now be approximately considered as an infinitesimal segment with projections dx and dy along $\hat{\mathbf{v}}(p)$ and $\hat{\mathbf{v}}_\perp(p)$ respectively, where p is the position of the vertex at the base of the edge. This lets us assign a cost to the edge using formula (5) (time optimal search) or (7) (energy optimal search).

Given a start and a goal vertex in the graph, we can thus use Dijkstra's optimal search algorithm [7] to find the shortest path in the graph connecting the vertices. Whenever possible, we can also design admissible heuristic functions, $h : V \to \mathbb{R}_{\geq 0}$, in order to run more efficient search algorithms such as A* [7]. In particular, if (X_s, Y_s) is the coordinate of the start vertex in a global Euclidean coordinate system, and (X_g, Y_g) is the global coordinate of vertex $g \in V$ then in case of the *time minimizing search*, a lower bound for the minimum time required to reach g from s is $h_{time}(g) = \frac{\sqrt{(X_g - X_s)^2 + (Y_g - Y_s)^2}}{V_{max} + v_{max}}$, where v_{max} is the maximum speed of the flow over all points in W. However for the *energy minimizing search* one cannot write a reasonable heuristic function other than the trivial $h(q) = 0$, in which case the A* search becomes equivalent to Dijkstra's.

B. Planning Optimal Trajectories in different Topological Classes

In multi-robot applications such as exploration or data collection in the ocean, it is vital that the team of surface vehicles can be distributed effectively across the region of interest. This task is made particularly challenging in the presence of obstacles or fluidic structures. An effective approach to this problem is topological exploration [12], where robots are assigned different topological classes of trajectories to disperse into and explore (see Figure 4(a)). Reasoning about topological classes of trajectories is also vital to automated surface cleaning operations using surface vehicles [4]. Thus we demonstrate, using the proposed approach, how such topological path planning can be performed for multi-robot systems.

Presence of obstacles in W give rise to multiple topological classes of trajectories. Since we plan trajectories in a graph, our method lends itself quite naturally to *homotopy-aware path planning*, where we can compute shortest trajectories restricted to different homotopy classes. The basic algorithm for doing this is outlined in [3, 13]. The fundamental idea is to use certain *homotopy invariants* (called h-signature) [2] to construct a homotopy-augmented graph, $\mathcal{G}_h = (V_h, E_h)$, in which every vertex is a pair of the form $(m, \mathfrak{w}) \in V_h$, with $m \in V$ and $\mathfrak{w}$ is a "word" made up of letters associated with non-intersecting rays emanating from connected components of obstacles (see Figure 4(b)). The edge set, E_h is described incrementally as follows: If $(m, \mathfrak{w}) \in V_h$, then for every $[m, n] \in E$ (where $m, n \in V$), there exists an edge $[(m, \mathfrak{w}), (n, \mathfrak{w} \circ h(\overline{mn}))] \in E_h$ (where $\mathfrak{w} \circ h(\overline{mn})$ indicate concatenation of the words $\mathfrak{w}$ and the h-signature of the directed segment $\overline{mn}$). The cost of an edge in G_h is taken to be same as the cost of the corresponding projected edges in G. Executing a graph search
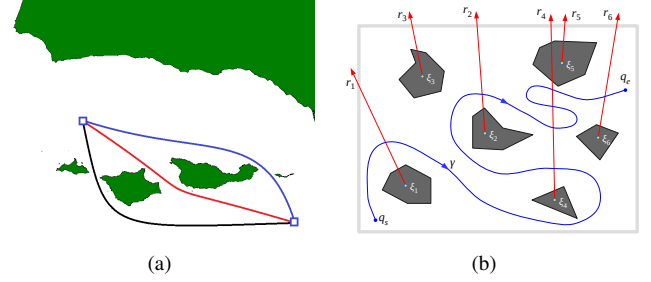

(a)   (b)

Fig. 4. *Homotopy-aware path planning*: (a) Simultaneous exploration/sampling around a set of islands by a set of ASVs. Homotopy aware planning allows the generation of optimal paths in different classes of trajectories simultaneously (b) Non-intersecting rays emanate from every connected component of obstacles. The homotopy invariant (h-signature) of a path, γ, is constructed by tracing the curve and constructing a word, in which we insert a letter or its inverse everytime we cross a ray from right-to-left or left-to-right respectively. Thus, in the figure, $h(\gamma) = $ "$r_1^{-1} r_4\ r_2^{-1} r_4^{-1} r_6^{-1}$".

in the h-augmented graph, G_h, can give us optimal trajectories constrained to different homotopy classes connecting given vertices $s, e \in V$.

C. An Alternative Approximate Model for Fast Computation of Smooth Trajectories

The optimal trajectories obtained using the graph search-based approach, being constrained to the graph, are piece-wise linear, and in general not smooth. In order to be able to solve the problem efficiently, we need to employ an approximation at some level, which in case of the graph search-based method is the discrete representation of the free space. In this section we present an alternative approximate model, derived from the cost functions described in Section II-C, to represent the optimal trajectory in the form of a solution to an ordinary differential equation. Thus the solutions obtained using this method can be made arbitrarily smooth. Moreover integration of a differential equation is in general faster to compute than running a graph search algorithm. In particular, we construct a Riemannian metric [21] that approximates the cost functions. This method is particularly useful for two reasons: (i) trajectories obtained as a solution to the geodesic differential equation are smooth, unlike paths in a graph that are restricted to the discrete graph, and (ii) computing solutions to the geodesic equation is often less computationally intensive than a full-blown graph search.

The Riemannian metric, $\mathbf{g}(p)$, for every $p \in W$ is defined as a (symmetric, positive-definite) bilinear form over differentials dx and dy (and dt if we use $p \in W_T$ as the workspace) such that the cost (equivalently the "length" induced by the metric) of the differential element can be written as

$$ dc = \sqrt{[dx, dy]\ g(p) \begin{bmatrix} dx \\ dy \end{bmatrix}} \quad (10) $$

where $g(p)$ is a 2×2 matrix, and is called the *matrix representation of the Riemannian metric* in the flow parallel coordinate system. Using this framework, we propose the following approximate time and energy minimizing Riemannian metric modes.

Riemannian Metric Model for Time Minimization: From equation (5) one has,

$$dc^2_{2,time} = \frac{(V^2_{max} + v^2)dx^2 + (V^2_{max} - v^2)dy^2}{(v^2 - V^2_{max})^2}$$
$$-\frac{2vdx\sqrt{V^2_{max}dx^2 + (V^2_{max} - v^2)dy^2}}{(v^2 - V^2_{max})^2}.$$

In order to make the above quadratic in dx and dy, we use the approximation that the velocity of the vehicle is almost parallel to that of the flow, i.e., $\left|\frac{dy}{dx}\right| \ll 1$. With this approximation, $\sqrt{V^2_{max}dx^2 + (V^2_{max} - v^2)dy^2} \cong V_{max}dx + \frac{1}{2}\frac{V^2_{max} - v^2}{V_{max}}\frac{dy}{dx}dy$. Thus we get,

$$dt^2 \cong \frac{1}{(v + V_{max})^2}dx^2 + \frac{1}{V_{max}(V_{max} + v)}dy^2$$

Thus, the matrix representation of the metric tensor in the 2-dimensional flow-parallel coordinates can be written as,

$$g = \begin{bmatrix} \frac{1}{(v+V_{max})^2} & 0 \\ 0 & \frac{1}{V_{max}(V_{max}+v)} \end{bmatrix}. \tag{11}$$

Riemannian Metric Model for Energy Minimization: Similar to above, from (7), one can write,

$$dc^2_{2,energy} = 4\kappa^2 v^2 \left(2dx^2 + dy^2 - 2dx^2\sqrt{1 + \left(\frac{dy}{dx}\right)^2}\right).$$

Using the approximation that the velocity of the vehicle is almost parallel to that of the flow, i.e., $\left|\frac{dy}{dx}\right| \ll 1$, we could rewrite above as,

$$dc^2_{2,energy} = 4\kappa^2 v^2 dy^2. \tag{12}$$

However, the matrix representation of the metric tensor that arises from (12) is singular and cannot be used with the geodesic equation. As such, we use the following matrix representation of the metric tensor in flow-parallel coordinates:

$$g = \begin{bmatrix} \frac{\epsilon}{v^2} & 0 \\ 0 & 4\kappa^2 v^2 \end{bmatrix} \tag{13}$$

where $\epsilon \ll 1$.

From g, one can use a coordinate transformation, $G = R^T g R$, to compute the matrix representations of the metric tensors in the global Euclidean coordinates, where $R = \begin{bmatrix} \hat{v}_X & -\hat{v}_Y \\ \hat{v}_Y & \hat{v}_X \end{bmatrix}$ is a suitable rotation matrix for local-to-global coordinate transformation. We can then solve the geodesic equation [21], $\frac{d^2\gamma^i}{d\tau^2} + \Gamma^i_{jk}\frac{d\gamma^j}{d\tau}\frac{d\gamma^k}{d\tau} = 0$, which is a second order ODE describing the shortest path, γ, parameterized by τ, the cumulative cost of the trajectory. Note that summation over repeated indices is assumed by Einstein notation. The quantities Γ^i_{jk} are called the Christoffel symbols, and are given by $\Gamma^i_{jk} = \frac{1}{2}\overline{G}^{im}\left(\frac{dG_{km}}{dX^j} + \frac{dG_{jm}}{dX^i} - \frac{dG_{jk}}{dX^m}\right)$ with $\overline{G} = G^{-1}$.

One consequence of the Riemannian approximation is the fact that the metric is indifferent to the direction of the flow, and the vector field in effect gets converted to a line field. Geodesics computed using the approximate Riemannian metric can thus be both in the direction of the flow and against it. However, the Riemannian model does not involve any discretization of the free space and can perform better than the graph search-based approach with a coarse discretization. This is illustrated in the example of Figure 8, where a coarse discretization gives a path in the graph that is significantly

different and of higher cost than the geodesic computed using the Riemannian metric. However, making the discretization finer (increasing connectivity of the graph) gives a path in the finer graph that matches the geodesic more closely. Other than this particular example, however, all the simulation and experimental results presented in this paper use the graph search-based method and the exact cost functions described in Section II-C.

IV. SIMULATIONS

In this section we present simulation results for planning time and energy optimal paths in a flow field using the exact cost functions described in Section II-C and the graph search-based approach described in Section III-A. We begin with the wind-driven double gyre model which is an analytical model often used because its flow patterns are similar to those seen in large scale recirculation regions in the ocean [28]. The flow velocity components in the global coordinate system are given in this model by,

$$\dot{X} = -\pi A \sin(\pi\frac{X}{s})\cos(\pi\frac{Y}{s}), \tag{14a}$$
$$\dot{Y} = \pi A \cos(\pi\frac{X}{s})\sin(\pi\frac{Y}{s}), \tag{14b}$$

where A determines the amplitude of the flow velocity vector and s scales the dimensions of the gyres. In our simulations, we set $A = 0.02$ and $s = 1$ so that the flow field approximately matches the flow field generated in the experimental setup with an average flow velocity of 0.025 m/s. In each of the simulations, we used the Dijkstra's search algorithm to find optimal paths in a 3m×3m workspace discretized as described in Section III-A.

Additionally, we present simulations with ocean current data obtained from the Naval Coastal Ocean Model (NCOM) database hosted by the Coastal Observing Research and Development Center (CORDC) [1] for simulations involving homotopy aware path planning. In particular, we used ocean current data from January 2016 for the Santa Barbara Channel along the California coast. This region is instrumented with several high frequency radar stations which provide hourly surface current measurements on a 2 km grid.

A. Time Optimal Paths

For this case, the differential cost given in (5) was used in the graph search. The flow field given by (14) was used for the simulations. Two cases, $V_{max} = 0.01$m/s ($< \|\mathbf{v}(p)\|$) and $V_{max} = 0.05$m/s ($> \|\mathbf{v}(p)\|$) were simulated. Figure 5 shows the results for these two cases for two start-goal combinations. As expected, in the case with the larger maximum velocity, the planned path takes a more direct route to the destination, and it demonstrates how the proposed method is able to explicitly account for actuation constraints of the vehicle.

B. Energy Optimal Paths

For this case, the differential cost given in (7) was used for the graph search. Similar to the time optimal case, two cases, $V_{max} = 0.01$m/s ($< \|\mathbf{v}(p)\|$) and $V_{max} = 0.05$m/s ($> \|\mathbf{v}(p)\|$) were considered. Figure 6 shows the energy optimal paths for several start-goal combinations. For the energy
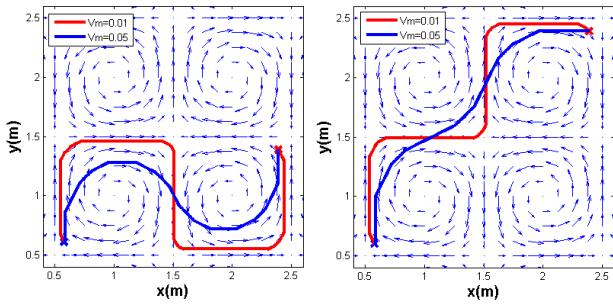
Fig. 5. Planned optimal time paths for two start-goal combinations. In red: $V_{max} = 0.01$m/s, in blue: $V_{max} = 0.05$m/s. Path costs, (a) red: 87.3s, blue: 33.5s, (b) red: 76.2s, blue: 30.5s.
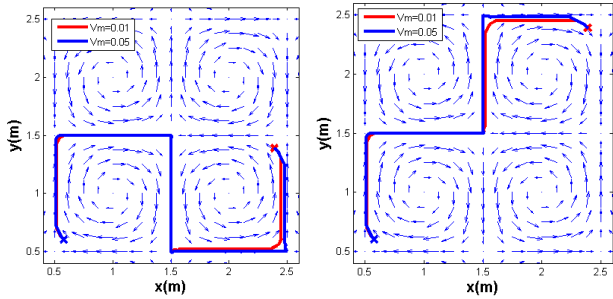

Fig. 6. Planned optimal Energy paths for two start-goal combinations. In red: $V_{max} = 0.01$m/s, in blue: $V_{max} = 0.05$m/s. Path costs, (a) red: 1.48×10^{-3} Nm, blue: 1.05×10^{-3} Nm, (b) red: 1.48×10^{-3} Nm, blue: 1.05×10^{-3} Nm.

optimal case, V_{max} does not play as an important role as in the time-optimality case, since the cost function selects the minimum possible $v_{still} < V_{max}$ to minimize the expended energy.

C. Planning in Multiple Homotopy Classes

A described in Section III-B, the proposed path planning framework can be used to plan optimal trajectories in different homotopy classes. This approach can be used for simultaneous navigation in regions riddled with obstacles, *e.g.*, the Philippine archipelago. Furthermore, as mentioned in Section III-B, this approach is particularly attractive for energy efficient exploration/sampling of ocean phenomena around a set of islands using a team of ASVs. This approach generates optimal paths along each coastal segment off an island. Figure 7(a) shows optimal energy paths generated around the Santa Cruz island and Figure 7(b) shows optimal energy paths generated around the Anacapa island, both off the coast of California. In both cases, the blue path shows the globally cost optimal path between the start and goal locations, and the red path shows a cost optimal path that is in a different homotopy class (i.e., on the opposite side of the island).

D. Using the Approximate Riemannian Metric Model for Path Planning

The time and energy minimizing Riemannian metric tensors developed in Section III-C were also used to generate time and energy optimal paths. Finding a path from the start position to the goal position translates to solving the two point boundary value problem posed by the geodesic equation. This was achieved by the *shooting method*: integrating the geodesic equation from the start position for a succession of initial
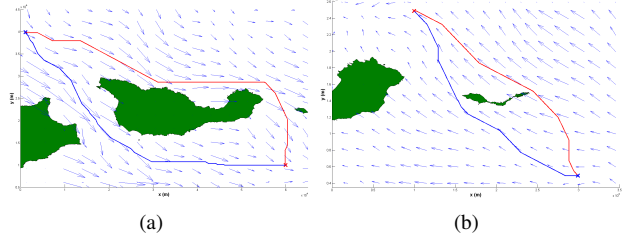

(a) (b)

Fig. 7. (a) Optimal Energy paths around Santa Cruz island. Path costs, blue: 1607 Nm, red: 2871 Nm. (b) Optimal Energy paths around Anacapa island. Path costs, blue: 396 Nm, red: 1390 Nm. In both cases, the blue path is the globally optimal path while the red path is optimal on the opposing side of the island. The blue cross gives the start location and the red cross gives the goal location.

directions until the goal position was reached, *i.e.*, the problem was solved as a series of initial value problems.

Figure 8 shows the comparison between the graph search method and the geodesic integration method, for $V_{max} = 0.01$m/s. Figure 8(a) shows an instance where the graph search method considers 16 neighbors for each node in the graph. Clearly the geodesic and the graph search generated paths do not agree. The cost of the path generated by the graph search is 1.96. In Figure 8(b), where the results of the two methods are in agreement, the graph search method considers 48 neighbors for each node in the graph, and the resulting path cost is only 1.53 (which is less than the previous case). However, in this instance, the graph search method takes around 50s to compute the path while the geodesic method takes less than 2s. This example clearly highlights three important advantages of the geodesic integration method: (i) it produces smooth trajectories (see Figure 8(a)), (ii) in contrast to graph search methods, the accuracy of the generated path is not limited by the resolution of the underlying workspace, and (iii) the geodesic integration method is much faster.

However, it was observed that for some instances, the geodesic integration method gave undesirable results. Figure 9 shows the optimal time path obtained from geodesic integration for $V_{max} = 0.05$m/s. The generated path is not necessarily a time-optimal path since parts of it fall along flow opposing directions. This occurs due to the symmetry of the Riemannian metric, where the cost heading against the flow is equal to the cost heading along the flow (see discussion in Section III-C). However, a method can be developed to construct a piecewise geodesic path from the start to the goal which ignores flow opposing directions. This is a direction for future work.

E. Performance comparison with existing methods

We compared the proposed method with *(i)* a level set method for planning optimal time paths [18], *(ii)* a graph search based method for planning optimal energy paths [14].

Level set method for optimal time paths: In this method introduced by Lolla *et al.*, a virtual front (a zero level set of a distance function), that is initialiazed around the start point, is advected by the flow as well as by a velocity component perpendicular to the front, until it reaches the goal position. The point on the front that reached the goal position is then integrated backwards in time to find the optimal time path. We
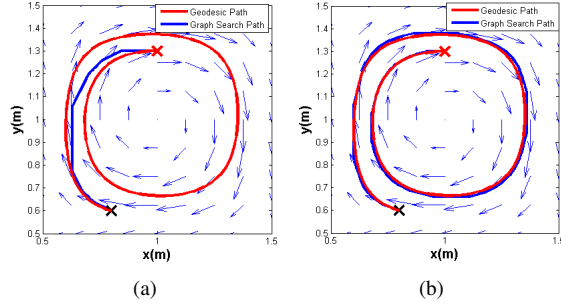
Fig. 8. Comparison of optimal energy paths generated by the graph search method (blue) with the path generated by integrating the geodesic equation (red). (a) Coarse graph constructed by connecting each vertex with 16 neighbors: Paths not in agreement and the cost of the path in graph is significantly worse, (b) Finer graph constructed by connecting each vertex with 48 neighbor: Paths in agreement and produces a lower cost path.
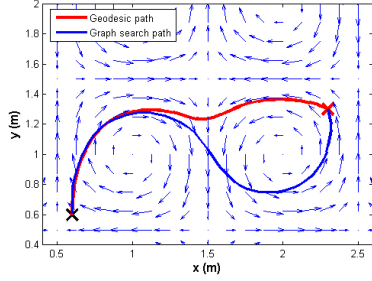

Fig. 9. Comparison of exact optimal cost paths (blue) with the paths generated by integrating the geodesic equations obtained from the approximate Riemannian metric model (red). In this instance the Riemannian metric model fails to generate a feasible path.

implemented a first order integration to compute time optimal path using the level-set method. To compute the path costs, we use the exact minimum-time cost function in (5), on the discretized level set path. As seen in Fig. 10 and Table I, the paths and the corresponding path costs obtained from the two methods are almost identical.

It can be seen that the proposed method produces comparable results to those obtained from the level set method. However, for a workspace containing n grid points, the computational cost of the level set method is $\mathcal{O}(n^3)$[18] while for graph search method with E connected neighbors, the computational cost is only $\mathcal{O}(nE\log(n))$. Furthermore, the level set method periodically performs a re-initialization step, which has a computational cost of $\mathcal{O}(n^3)$. In addition to lower computational cost, a further advantage of our method is the ability to incorporate different cost functions, whereas the level set method can only generate time optimal paths.

Comparison with a graph search method for optimal energy paths: We compared the optimal energy paths generated from our method with another graph search based method presented by Koay *et al.*[14] (henceforth referred to as the Koay method). Even though the approach is similar, the cost function proposed in the Koay method is approximate. In their method, the authors minimize $\mathbf{v}_{still}$ for each path segment in the graph and use this minimized value to compute the energy cost for that path segment. In effect, the cost of each path segment is computed to be $\int \kappa \|min(\mathbf{v}_{still})\|^2 dt$ whereas the actual energy cost should be $min \int \kappa \|\mathbf{v}_{still}\|^2 dt$. Fig. 11 shows the "optimal" energy paths computed using the two methods near
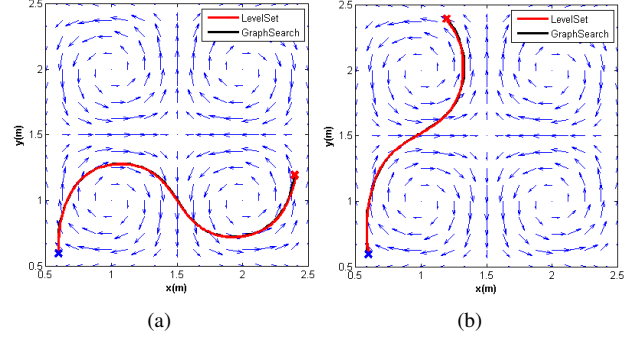

Fig. 10. Comparison of time optimal paths obtained from the level set method (red) with paths from the graph search method with 48-neighbors (black). The costs for (a) path 1 and (b) path 2 are given in Table I, and are comparable.

TABLE I
COMPARISON OF PATH COSTS AGAINST THE LEVEL SET METHOD FOR TIME OPTIMAL PATHS. THE PATH COSTS OBTAINED FROM THE TWO METHODS ARE COMPARABLE.

	Level Set	Graph Search(48-con.)	Graph Search(16-con.)
Path 1	30.39	30.53	31.07
Path 2	22.47	22.76	23.07

the Santa Cruz island. The cost of the path generated from the Koay method, when evaluated using the cost function given in (7), is 3% higher than the actual optimal path, and it lies on the opposite side of the island. Furthermore, the cost of the actual optimal energy path (generated using the proposed method), is 16% higher when evaluated using the cost metric used in the Koay method.

V. EXPERIMENTS

We further validate the paths computed by our proposed graph search method (using the exact cost functions described in Section II-C) through experiments. The experiments were conducted using the multirobot Coherent Structure Testbed (mCoSTe) which consists of a micro Autonomous Surface Vehicle (mASV) and the $3 \times 3 \times 1$ m^3 Multi Robot flow tank (MR tank). The mASV is a differential drive vehicle with a maximum forward speed of 0.2m/s. Localization for the vehicle was provided by an external motion capture system. Multi-gyre flows were created in the MR tank using four flow driving cylinders rotating at approximately 100 rpm. Figure 12 shows the components of the experimental setup.

In Section IV, the parameters of (14) were selected such that the simulated flow field closely approximates the flow field generated in the tank. Therefore, we use the paths generated in the simulations as reference trajectories for the mASV to follow. In particular, we used the path generated for the case where $V_{max} = 0.05$m/s from start location $[0.6, 0.6]^T$ to goal location $[2.3, 1.6]^T$. For the experiments, we considered three paths between the start and goal locations: the time-optimal path, the energy-optimal path and the minimum-distance path. The minimum-distance path was considered in order to provide a baseline to compare the results of the time-optimal and energy-optimal paths. Seven runs were carried out for each case and results were recorded. The energy expended by the mASV was computed as,
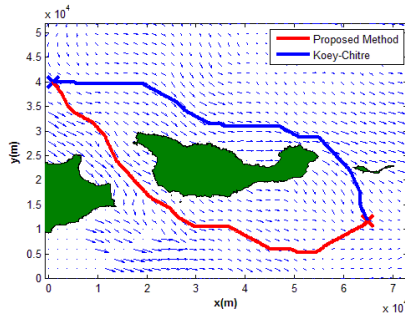
Fig. 11. Comparison of optimal energy paths generated using the proposed method and the Koay method. Path costs: our method (red path): 1802 (2097) Nm, Koay method (blue path): 1864 (1942) Nm. The values outside parenthesis are the energy values computed using the exact energy cost (7), while in parenthesis are the path costs calculated using the approximate cost metric used in the Koay method.
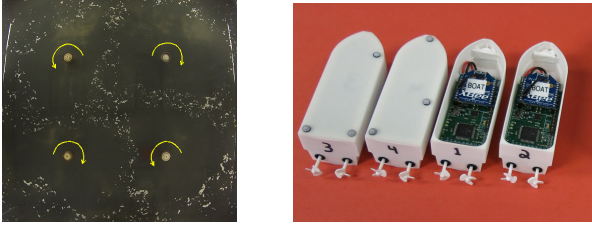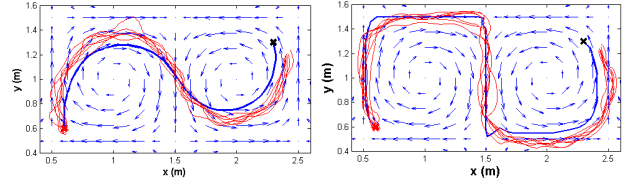

Fig. 12. Experimental setup, right: four flow driving cylinders creating a double gyre like flow in the MR tank, left: mASVs used for the experiments.

$$E = \int \left(V_{Lmotor}^2 + V_{Rmotor}^2 \right) dt \approx \frac{V_{in}^2}{255^2} \Delta t \sum_i \left(L_{cmd_i}^2 + R_{cmd_i}^2 \right)$$

where V_{Lmotor} and V_{Rmotor} are the voltages applied to the left and right motors respectively, V_{in} is the battery voltage, L_{cmd_i} and R_{cmd_i} are the PWM signal commands sent out to the mASV during the ith command cycle, and Δt is the duration of each command cycle.

Table II shows the average time taken and the average energy expended by the mASV to complete each path. As expected, the mASV completes the time-optimal path the quickest and expends the least amount of energy while completing the energy-optimal path. On average the time-optimal path is 19% faster than the minimum-distance path and the energy-optimal path expends 51% less energy than the minimum-distance path. Fig.13 shows the paths taken by the mASV against the path planned by the graph search method and Fig. 14 shows the path taken by the mASV overlaid on an image of the tank.

VI. CONCLUSIONS AND DISCUSSIONS

In this work we presented graph-search based methods to plan optimal trajectories in flow fields. We designed exact cost functions which explicitly consider the kinematic constraints of the vehicle for both time and energy optimal path planning. In addition, we used tools from topological path planning to generate optimal paths in different homotopy classes of the environment. We were able to verify the efficacy of the

TABLE II
PATH COSTS FOR THE THREE PATHS AVERAGED OVER SEVEN TRIALS

	time optimal	energy optimal	min distance
Time (s)	44.8	92.7	55.4
Energy (W)	250.3	222.3	449.3

(a) time-optimal path   (b) energy-optimal path

Fig. 13. Paths taken by the mASV in the MR tank (in red) and the paths planned by the graph search algorithm (in blue).

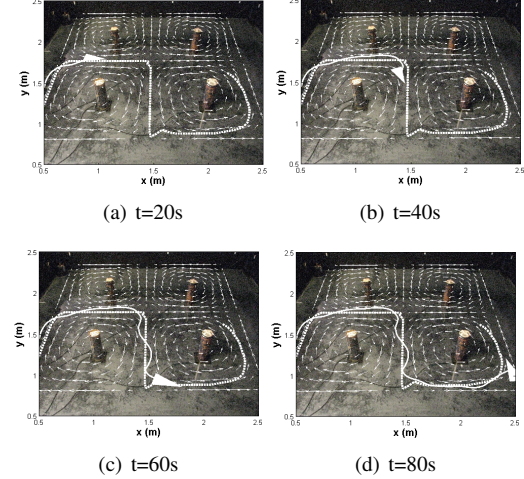(a) t=20s   (b) t=40s

(c) t=60s   (d) t=80s

Fig. 14. Path taken by the ASV in the tank while following the optimal energy path. The dotted white line is the reference path, the solid white line shows the path taken by the ASV and the solid triangle represents the ASV.

planned paths through experiments conducted in an indoor laboratory testbed, where, despite having partial/noisy flow information, our method was successful in planning optimal trajectories. Even though in this paper we only considered time-invariant flow fields, the presented methods can easily be extended for path planning in time-varying flows. However, the utility of such an extension is limited by the accuracy of flow model used to predict the future time variations of the flow. We compared our results with other methods in literature: The level-set method, although being computationally expensive and be able to compute time-optimal paths only, give comparable time-optimal path costs; and the Koay method for energy optimal paths give more expensive paths than ours.

We also presented a Riemannian approximations to the exact cost functions used for the graph search method. We showed how these Riemannian metrics could be used with the geodesic equation to generate approximations to time and energy optimal paths. Even though some drawbacks were observed in this method, we believe that this approach to path planning in a flow not only provides better insights into the problem but also opens up new possibilities for path planning in a flow field, and as a results warrants further investigation. For example, a possible application for this approach would be the generation of real-time energy efficient paths for a team of autonomous vehicles using in-situ flow velocity measurements.

VII. ACKNOWLEDGEMENTS

REFERENCES

[1] Naitonal HF RADAR network - surface currents. URL http://cordc.ucsd.edu/projects/mapping/maps/.

[2] Subhrajit Bhattacharya and Robert Ghrist. Path homotopy invariants and their application to optimal trajectory planning. In Proceedings of IMA Conference on Mathematics of Robotics (IMAMR), St Anne's College, University of Oxford, September 9-11 2015.

[3] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. Topological constraints in search-based robot path planning. Autonomous Robots, 33(3):273–290, October 2012. ISSN 0929-5593. DOI: 10.1007/s10514-012-9304-1.

[4] Subhrajit Bhattacharya, Soonkyum Kim, Hordur Heidarsson, Gaurav Sukhatme, and Vijay Kumar. A topological approach to using cables to separate and manipulate sets of objects. International Journal of Robotics Research, 34(6):799–815, April 2015. DOI: 10.1177/0278364914562236.

[5] D. Caron, B. Stauffer, S. Moorthi, A. Singh, M. Batalin, E. Graham, M. Hansen, W. Kaiser, J. Das, A. de Menezes Pereira, B. Zhang A. Dhariwal, C. Oberg, and G. Sukhatme. Macro- to fine-scale spatial and temporal distributions and dynamics of phytoplankton and their environmental driving forces in a small subalpine lake in southern california, usa. Journal of Limnology and Oceanography, 53(5):2333–2349, 2008.

[6] V. Chen, M. Batalin, W. Kaiser, and G. Sukhatme. Towards spatial and semantic mapping in aquatic environments. In IEEE International Conference on Robotics and Automation, pages 629–636, Pasadena, CA, 2008.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms. MIT Press, 2nd edition, 2001.

[8] Jnaneshwar Das, Frdric Py, Thom Maughan, Tom OReilly, Monique Messi, John Ryan, Kanna Rajan, and GauravS. Sukhatme. Simultaneous tracking and sampling of dynamic oceanographic features with autonomous underwater vehicles and lagrangian drifters. In Oussama Khatib, Vijay Kumar, and Gaurav Sukhatme, editors, Experimental Robotics, volume 79 of Springer Tracts in Advanced Robotics, pages 541–555. Springer Berlin Heidelberg, 2014. ISBN 978-3-642-28571-4. doi: 10.1007/978-3-642-28572-1_37.

[9] Tathagata Das, Prashanth Mohan, Venkata N. Padmanabhan, Ramachandran Ramjee, and Asankhaya Sharma. Prism: Platform for remote sensing using smartphones. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys '10, pages 63–76, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-985-5. doi: 10.1145/1814433.1814442.

[10] B. Garau, A. Alvarez, and G. Oliver. Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an a* approach. In Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pages 194–198, April 2005. doi: 10.1109/ROBOT.2005.1570118.

[11] V.T. Huynh, M. Dunbabin, and R.N. Smith. Predictive motion planning for auvs subject to strong time-varying currents and forecasting uncertainties. In Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 1144–1151, 2015. doi: 10.1109/ICRA.2015.7139335.

[12] Soonkyum Kim, Subhrajit Bhattacharya, Robert Ghrist, and Vijay Kumar. Topological exploration of unknown and partially known environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, November 3-7 2013.

[13] Soonkyum Kim, Subhrajit Bhattacharya, and Vijay Kumar. Path planning for a tethered mobile robot. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, May 31 - June 7 2014.

[14] Teong-Beng Koay and M. Chitre. Energy-efficient path planning for fully propelled auvs in congested coastal waters. In OCEANS - Bergen, 2013 MTS/IEEE, pages 1–9, 2013. doi: 10.1109/OCEANS-Bergen.2013.6608168.

[15] D. Kruger, R. Stolkin, A. Blum, and J. Briganti. Optimal auv path planning for extended missions in complex, fast-flowing estuarine environments. In Robotics and Automation, 2007 IEEE International Conference on, pages 4265–4270, April 2007. doi: 10.1109/ROBOT.2007.364135.

[16] T. Lolla, M. P. Ueckermann, P. Haley, and P. F. J. Lermusiaux. Path planning in time dependent flow fields using level set methods. In in the Proc. IEEE International Conference on Robotics and Automation, Minneapolis, MN USA, May 2012.

[17] T. Lolla, P.J. Haley Jr., and P.F.J. Lermusiaux. Path planning in multi-scale ocean flows: Coordination and dynamic obstacles. Ocean Modelling, 94:46 – 66, 2015. ISSN 1463-5003. doi: http://dx.doi.org/10.1016/j.ocemod.2015.07.013.

[18] Tapovan Lolla, Pierre F. J. Lermusiaux, Mattheus P. Ueckermann, and Patrick J. Haley. Time-optimal path planning in dynamic flows using level set equations: theory and schemes. Ocean Dynamics, 64(10):1373–1397, 2014. ISSN 1616-7228.

[19] K. M. Lynch, P. Schwartz, I. B. Yang, and R. A. Freeman. Decentralized environmental modeling by mobile sensor networks. IEEE Trans. Robotics, 24(3):710–724, 2008.

[20] B. Munson, D. Yound, and T. Okiishi. Fundamentals of Fluid Mechanics. John Wiley & Sons, 2006. ISBN 9780471675822.

[21] P. Petersen. Riemannian Geometry. Graduate Texts in Mathematics. Springer, 2006. ISBN 9780387294032.

[22] A. Shchepetkin and J. McWilliams. Quasi-monotone advection schemes based on explicit locally adaptive dissipation. Monthly Weather Review, 126:1541–1580,

1998.

[23] A. F. Shchepetkin and J. C. McWilliams. The regional oceanic modeling system (roms): a split-explicit, free-surface, topography-following-coordinate oceanic model. Ocean Modeling, 9:347–404, 2005.

[24] R. Smith, A. M. Pereira, Y. Chao, P.P. Li, D. A. Caron, B. H. Jones, and G. Sukhatme. Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study. In Proc. of the 2010 IEEE International Conference on Robotics and Automation, pages 4770–4777, Anchorage, AK, 2010.

[25] R. Smith, J. Kelly, and G. Sukhatme. Towards improving mission execution for autonomous gliders with an ocean model and kalman filter. In Proc. of the 2012 IEEE International Conference on Robotics and Automation, Minneapolis, MN, 2012.

[26] Ryan N. Smith, Yi Chao, Peggy P. Li, David A. Caron, Burton H. Jones, and Gaurav S. Sukhatme. Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model. International Journal of Robotics Research, 29(12):1475–1497, 2010. doi: http://dx.doi.org/10.1177/0278364910377243.

[27] N. Sydney and D. A. Paley. Multi-vehicle control and optimization for spatiotemporal sampling. In IEEE Conf. Decision and Control, pages 5607–5612, Orlando, FL, 2011.

[28] G. Veronis. Wind-driven ocean circulation, part i and part ii. Deep-Sea Res., 13(31), 1966.

[29] W. Wu and F. Zhang. Cooperative exploration of level surfaces of three dimensional scalar fields. Automatica, the IFAC Journall, 47(9):2044–2051, 2011.

[30] W. Wu, D. Chang, and F. Zhang. Glider ct: Reconstructing flow fields from predicted motion of underwater gliders. In Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems (WuWNet'13), page Article no. 47, Kaohsiung, Taiwan., 2013.