# Distance and Steering Heuristics for Streamline-Based Flow Field Planning

K. Y. Cadmus To[1], Chanyeol Yoo[1], Stuart Anstee[2], and Robert Fitch[1]

*Abstract*— Motion planning for vehicles under the influence of flow fields can benefit from the idea of streamline-based planning, which exploits ideas from fluid dynamics to achieve computational efficiency. Important to such planners is an efficient means of computing the travel distance and direction between two points in free space, but this is difficult to achieve in strong incompressible flows such as ocean currents. We propose two useful distance functions in analytical form that combine Euclidean distance with values of the stream function associated with a flow field, and with an estimation of the strength of the opposing flow between two points. Further, we propose steering heuristics that are useful for steering towards a sampled point. We evaluate these ideas by integrating them with RRT* and comparing the algorithm's performance with state-of-the-art methods in an artificial flow field and in actual ocean prediction data in the region of the dominant East Australian Current between Sydney and Brisbane. Results demonstrate the method's computational efficiency and ability to find high-quality paths outperforming state-of-the-art methods, and show promise for practical use with autonomous marine robots.

## I. INTRODUCTION

Streamline-based planning [1, 2] uses the concepts of *stream functions* and *streamlines* from fluid dynamics to efficiently plan paths through flow fields for vehicles, such as underwater gliders [3–5] and autonomous surface vessels [6]. In this approach, the motion of water in the upper levels of the ocean is modelled as a 2-dimensional, incompressible flow. Efficiency arises from an elegant dimensionality reduction of the control space made possible by the additive property of stream functions [7]. Many planning frameworks rely on the ability to quickly estimate travel distance between two locations, however, this is not straightforward for the nonlinear spaces that arise from incompressible flows. We examine distance heuristics that are useful for streamline-based planning and that complement existing work, enabling the development of exciting new planning algorithms that aim to improve the autonomy of marine robots.

The notion of distance is fundamental for graph-based planning algorithms, which has been applied in both static and time-varying flow fields [8]. Sampling-based motion planners in particular uses this notion for nearest neighbour search and other algorithmic components seen in probabilistic roadmaps (PRMs) [9] and rapidly-exploring random trees (RRTs) [10]. For example, $L^\infty$ distance is useful in planning the motion of manipulator arms [11]. $L^2$ distance (Euclidean distance) has been used for planning in flow fields [12, 13], but in strong flows, where the flow speeds are comparable to or exceed the vehicle speed, this metric can be wildly misaligned with true traversal time and the reachability of some locations. Figure 1 illustrates how a goal position that is a short $L^2$ distance away from its position nevertheless falls outside the reachable space.

Finding useful distance heuristics is challenging in a flow field because the effective distance depends on the interaction of vehicle dynamics with the direction and magnitude of the flow. Flow fields induced by ocean currents can be highly non-uniform; locations that are geographically close together may effectively be far apart or unconnected. Conversely, large geographical distances can be traversed quickly given high-magnitude flow in an advantageous direction. Forward integration is possible [14] but computation can be prohibitively expensive, especially in sampling-based algorithms that require frequent connection of samples within a given distance threshold [12, 13].

In this paper, we consider distance heuristics derived from stream function equations and derive effective steering functions for efficiently connecting new samples in the RRT* family of algorithms. We construct heuristics that combine $L^2$ distance with stream function values and a measure of the opposing flow between two points. These are used to associate a generated sample with a node in the existing tree. We then propose heuristics for choosing control actions that can steer the system towards long-distance sample points. For the rewiring step in RRT*, edge connection costs can already be computed efficiently; our previous work showed that in 2D incompressible flow fields, suitable control actions
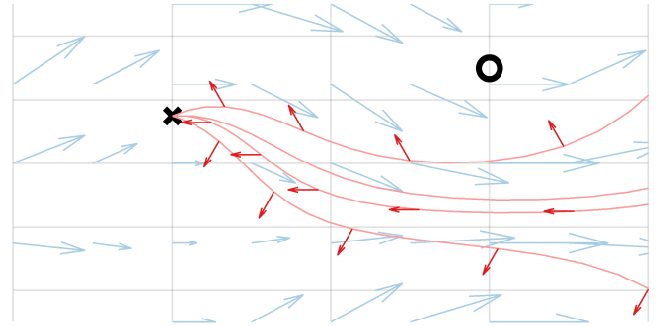


Fig. 1. The limited reachable space of a vehicle (black cross) in a strong flow field (blue arrows) attempting to reach a goal (black circle). Four trajectories (pink lines) are shown: three with maximum velocity in the direction of the red arrows, and one with no control applied (no arrows).

lie surprisingly on a line that is convenient to search [1]. We integrate these ideas within the RRT* framework and show that it generates longer edge connections than existing methods, resulting in earlier feasible solutions and higher-quality final solutions. This approach is also applicable to planners that address time-varying flows [8] and to many RRT variants, such as goal-biasing RRT [15] and birectional RRT [16], and will complement informed RRT [17] by finding feasible solutions quickly in a bounded search space.

We present a detailed experimental evaluation of our methods in comparison to existing variants of RRT* in two example scenarios. The first scenario is a quad-vortex example that has been used previously in the literature [6, 18]. The second scenario is a traversal from Sydney to Brisbane, a direction that opposes the mean flow of the East Australian Current, using a numerical estimate of the actual flow. Results show that our methods outperform existing RRT* variants in all evaluation criteria, and find paths from Sydney to Brisbane with traversal times that are two times (5 days) faster. This is notable because it is also faster than the great circle (direct) path in still water, even though the dominant current flows in the opposite direction. Further, our method's first feasible traversal time, found after 31 s of computation, is faster than the comparison method's final solution.

The main contribution of this paper is a novel set of distance and steering heuristics to aid streamline-based planning, and their evaluation in a sampling-based planning framework. The significance of this contribution is an improved capability of autonomous marine robots by enabling on-board planning with embedded computation, by allowing for fast replanning to respond to unexpected situations, and by finding high-quality paths over large distances.

## II. Related work

The problem setting we consider requires a flow field prediction. Oceanic current predictions are freely available [19–21], but may have uncertainties that are significant when planning for slow-moving vehicles. In previous work we successfully used drift errors to estimate a local stream function online [22].

The general problem of finding an optimal path in a flow field is known as *Zermelo's problem* [23]. Level-set methods have been proposed for finding time-optimal [24] and energy-optimal [25] paths. Numerical solutions to the underlying partial differential equation have also been proposed [26]. Both assume a discretised workspace, as do graph-based methods with uniform [27] or adaptive [28] sampling. These methods can be computationally impractical for large problems or applications that require onboard planning.

Our previous work introduced the idea of using sampling-based methods for asymptotically optimal planning in flow fields with underwater gliders that have complex dynamics [18], and introduced efficient methods with analytical guarantees in the time-varying case [8]. Algorithms such as RRT* [29, 30] can use pseudo-metrics such as cost-to-go, which unfortunately is computationally expensive to evaluate
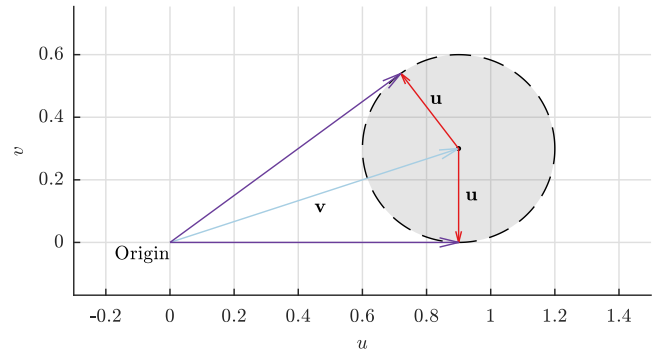


Fig. 2. Diagram in velocity space showing limited total velocity angles when a vehicle is in strong flows. The flow velocity is represented by a blue arrow. The critical vehicle velocities (red arrows) are shown that corresponds to total velocities (purple arrows) at their limits. The speed of the vehicle is limited by its maximum speed, which is shown as a grey circle.

in flow fields [31]. Non-Euclidean distance functions for strong incompressible flows have not appeared previously.

An approach that is close to ours is the vector field RRT (VF-RRT) [13], which introduces an upstream criterion to steer depending on a global measurement of exploration inefficiency and a user-specified reference exploration inefficiency factor $E_s$. This global quantity can overgeneralise the behaviour of the flow field, which leads to poor coverage of the configuration space. We provide performance comparisons with VF-RRT in this paper.

## III. Problem formulation and approach

We assume a vehicle in a flow field has dynamics represented by the continuous-time transition model

$$\dot{\mathbf{x}} = \mathbf{v}(\mathbf{x}) + \mathbf{u}, \tag{1}$$

where $\mathbf{x} = [x, y]^\top$ is vehicle position, $\mathbf{v}(\mathbf{x}) = [u_c, v_c](\mathbf{x})^\top$ is the time-independent flow field, and $\mathbf{u} = [u_s, v_s]^\top$ is the relative velocity of the vehicle with respect to the local flow. The vehicle's control efforts are constrained only by its speed:

$$\|\mathbf{u}\|_2 \leq V_{\max}, \tag{2}$$

where $\|\mathbf{u}\|_2$ is the $L^2$-norm of vehicle velocity and $V_{\max} \in \mathbb{R}^+$ is its maximum speed.

The dynamics of the vehicle in (1) can be discretised with a small time step $\Delta t$. Thus, we have

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (\mathbf{v}(\mathbf{x}_k) + \mathbf{u}_k)\,\Delta t, \tag{3}$$

for time step $k$. The vehicle can apply a *persistent* (that is, piecewise-constant) control $a = (\mathbf{u}, \tau)$, where $\tau$ is the time duration for which the control is applied. The position of the vehicle after applying control $a_k$ is expressed as

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, a_k), \tag{4}$$

where $\mathbf{x}_k$ is the prior position of the vehicle.

We consider a path planning problem in a 2D incompressible flow field where the vehicle is to reach a given goal position. The vehicle is guided by a sequence of $K$ controls,

$$\omega = (a_0, \ldots, a_{K-1}), \tag{5}$$

that are applied in sequence starting from the vehicle's initial location. An optimal sequence of controls $\omega^*$ minimises a cost function $J(\mathbf{x}, a)$ summed over the corresponding path to the destination. This problem is formally defined as follows:

**Problem 1** (Path planning in 2D incompressible flow fields). *Given vehicle dynamics* (1)*, a start and destination pair* $(\mathbf{x}_{init}, \mathbf{x}_{goal})$*, an incompressible flow field* $\mathbf{v}$*, and the cost function* $J(\mathbf{x}, a)$*, find the optimal sequence of controls*

$$\omega^* = \arg\min_{\omega} \sum_{k=0}^{K-1} J(\mathbf{x}_k, a_k) \quad for \ K \in \mathbb{N}, \quad (6)$$

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

Sampling-based motion planners is a class of algorithms that addresses this problem by building a graph by probabilistically sampling from the free configuration space and establishing edge connections to existing nodes with a weight computed by a given cost function. RRT* in particular, builds a tree from which a path can be queried. Such algorithms rely on coverage of the configuration space (through successful node placement) in order to increase the probability of including configurations that lie on an optimal path.

A core issue is that the strength of the flow field affects the reachable space from any point, as illustrated in Fig. 2. Unlike physical obstacles, reachability depends on the reference position. These circumstances impede successful coverage, which can prevent the algorithm from finding low-cost paths.

Coverage can be quantified in terms of its *dispersion* [30, 32, 33], which measures the radius of the largest empty ball in the configuration space. Given a set of vertices $\mathcal{V}$ that represent points in a 2D configuration space, the $L^2$ dispersion is the radius $\delta$ of the largest empty circle in bounded search space $X \subset \mathbb{R}^2$, i.e.

$$\delta(\mathcal{V}) = \max_{\mathbf{x}_P \in X} \left( \min_{\mathbf{x}_Q \in \mathcal{V}} \|\mathbf{x}_Q - \mathbf{x}_P\|_2 \right). \quad (7)$$

To increase the chances of connecting to samples from low-dispersion sampling sequences (highlighted by [30]), we focus in particular on: a) formulating new heuristics that better align with the effective travel distance between points, and b) steering towards the sample point to reduce the number of intermediate samples generated.

## IV. DISTANCE HEURISTICS FOR INCOMPRESSIBLE FLOW FIELDS

In this section we present two streamline-based distance heuristics that increase the chances of successful edge connection in an RRT* framework. We begin with a brief overview of stream functions and streamline-based control. Further details can be found in our previous work [1].
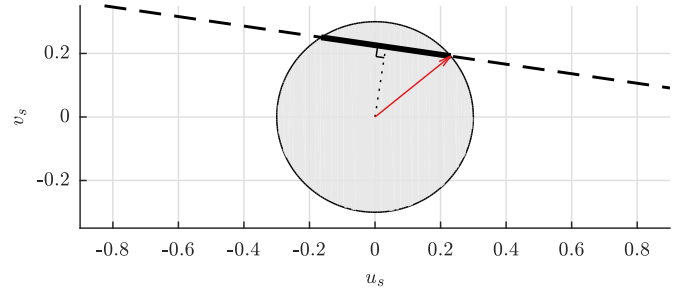
### A. Stream functions

A stream function is defined for a 2D vector field $\mathbf{v}(\mathbf{x})$ that is incompressible (the net flow through any closed boundary is zero), i.e.,

$$\nabla \cdot \mathbf{v}(\mathbf{x}) = 0, \quad (8)$$
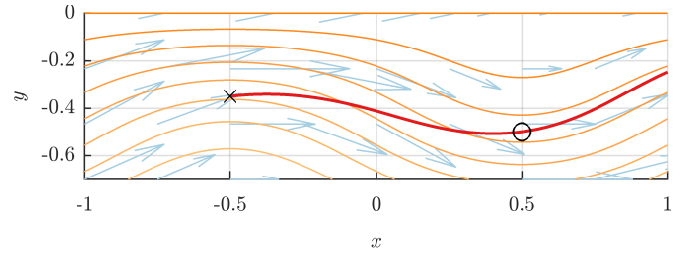


(a) Configuration space



(b) Velocity space

Fig. 3. Example of an optimistic control action that steers the vehicle to a goal position. (a) In configuration space, the trajectory (red line) of the vehicle (starting at position marked by a cross) is heavily influenced by the underlying flow field (blue arrows) that guides it to the goal (circle). The streamlines of the flow field are shown as orange lines. (b) In velocity space, the optimistic control (red arrow) is selected from the streamline-based control line (dashed line) constrained by the vehicle's maximum speed (grey circle) in the direction of the goal. The lower speed bound of this case corresponds to the length of the dotted line.

where $\nabla \cdot$ is the divergence operator. The value of the stream function or *stream value* from $\mathbf{x}_P$ to $\mathbf{x}_Q$ is [7]

$$\psi(\mathbf{x}_P, \mathbf{x}_Q) = \int_{\mathbf{x}_P}^{\mathbf{x}_Q} \left( u_c(\mathbf{x}) dy - v_c(\mathbf{x}) dx \right). \quad (9)$$

Continuous level sets of the stream function are referred to as *streamlines* (orange lines in Fig. 3a). The stream value measures the degree to which a path crosses streamlines; an implicit consequence is that if $\mathbf{x}_Q$ is downstream (or upstream) of $\mathbf{x}_P$, then the stream value between them must be zero. Like the flow fields themselves, stream functions also possess the additive property. Given two incompressible flow fields $\mathbf{v}_A(\mathbf{x})$ and $\mathbf{v}_B(\mathbf{x})$, the superimposed flow field is

$$\mathbf{v}_{A+B} = \mathbf{v}_A + \mathbf{v}_B. \quad (10)$$

The corresponding stream function is

$$\psi_{A+B} = \psi_A + \psi_B. \quad (11)$$

In flows that are irrotational, streamlines are perpendicular to the velocity potential contours [7]. Distance could be accurately captured by the pair of functions since, intuitively, the shape of the functions conforms to the shape of the flow field. We expect that stream functions maintain some of these properties in mixed flows with rotational components.

### B. Streamline-based 2D control search

To steer the vehicle towards a goal, we need to search the space of possible control velocities to find one that is feasible.

(a) $L^2$-stream distance     (b) $L^2$-LSB distance     (c) Time cost-to-go (one persistent control)
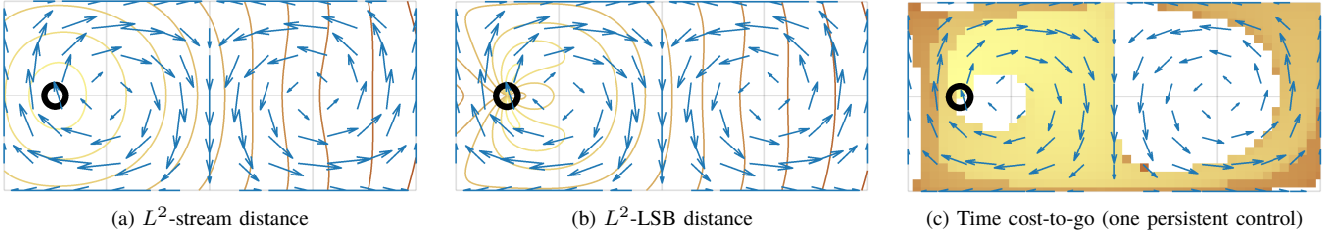
Fig. 4. Contours of distance function. The maximum flow strength is $4\times$ the vehicle's speed.

Our previous work [1] showed that properties of the stream function can be used to limit the size of this search space. Travelling from $\mathbf{x}_P$ to $\mathbf{x}_Q$, the vehicle velocity $\mathbf{u}$ must lie on the *streamline-based control line* described by

$$\psi(\mathbf{x}_P, \mathbf{x}_Q) + u_s \Delta y - v_s \Delta x = 0, \qquad (12)$$

where $\Delta x = x_Q - x_P$ and $\Delta y = y_Q - y_P$. This is illustrated as a dashed line in Fig. 3b.

The distance between the origin and the control line is a *lower speed bound (LSB)* for the vehicle's speed, i.e., it must have a speed of at least

$$V_{\mathrm{LSB}}(\mathbf{x}_P, \mathbf{x}_Q) = \frac{|\psi(\mathbf{x}_P, \mathbf{x}_Q)|}{\|\mathbf{x}_Q - \mathbf{x}_P\|_2} \qquad (13)$$

to be able to travel from $\mathbf{x}_P$ to $\mathbf{x}_Q$. The length of the dotted line in Fig. 3b corresponds to this value. The LSB therefore acts as a measure of the difficulty of reaching $\mathbf{x}_Q$. A larger LSB means a stronger opposing flow, which we interpret as a relative increase in the still-water distance (or time) required for traversal.

### C. Streamline-based distance functions

The shortcoming of Euclidean distance in strong flow fields is that it ignores their potentially powerful effect on vehicle motion (see Fig. 2). We propose to augment Euclidean distance with stream function-dependent terms to construct distance heuristics in three dimensions rather than two.

The first heuristic we term the $L^2$-*stream distance* is defined as

$$d(\mathbf{x}_P, \mathbf{x}_Q) = \sqrt{\|\mathbf{x}_P - \mathbf{x}_Q\|_2^2 + [\psi(\mathbf{x}_P, \mathbf{x}_Q)/\alpha]^2}, \quad (14)$$

where $\alpha$ is a scaling term that may be thought of as a characteristic velocity. The space $(x, y, \psi)$ can be verified to be a metric space with this distance definition by considering the four necessary conditions [33] (nonnegativity, reflexivity, symmetry, and triangle inequality). The $L^2$-stream distance penalises paths that cross streamlines; its level sets are therefore elongated along the flow direction. Contours of $L^2$-stream distance are illustrated in Fig. 4a.

We also propose a second distance heuristic, which we term $L^2$-*LSB distance*, formulated using (13) as:

$$d(\mathbf{x}_P, \mathbf{x}_Q) = \sqrt{\|\mathbf{x}_P - \mathbf{x}_Q\|_2^2 + [V_{\mathrm{LSB}}(\mathbf{x}_P, \mathbf{x}_Q)\beta]^2}, \quad (15)$$

where $\beta$ is a scaling term equivalent to a characteristic time. The $L^2$-LSB distance does not satisfy the triangle inequality, so $(x, y, V_{\mathrm{LSB}})$ is not a metric space with this distance

definition, but it behaves in a similar fashion. Contours of $L^2$-LSB distance are illustrated in Fig. 4b.

These streamline-based distance functions penalise paths that cross streamlines, which we have observed from previous work are unlikely to be part of optimal connections. The penalty in the $L^2$-LSB distance is explicitly local, since it approaches Euclidean distance for points with increasing separation.

In strong flow fields, the ideal distance function is asymmetrical and should favour paths that follow the direction of local flow. This is confirmed by Fig. 4c, which illustrates the numerically-computed cost-to-go function resulting from a given persistent control. Despite its symmetry, we see that the $L^2$-LSB distance has a similar shape to cost-to-go in the region local to the vehicle position.

We can impose a natural ordering on the $(x, y, \psi)$ space with an external reference point, since

$$\psi(\mathbf{x}_P, \mathbf{x}_Q) = \psi(\mathbf{x}_Q, \mathbf{x}_0) - \psi(\mathbf{x}_P, \mathbf{x}_0), \qquad (16)$$

where $\mathbf{x}_0 \in \mathbb{R}^2$ is an arbitrary reference point. There is no equivalent equation for $V_{\mathrm{LSB}}$.

Lacking a global reference, the space $(x, y, V_{\mathrm{LSB}})$ cannot be used with tree data structures such as $k$-d trees [34], $R^*$ trees [35], and balanced-box decomposition (BBD) trees [36] for fast neighbourhood query methods. These methods have $\mathcal{O}(\log |\mathcal{V}|)$ time complexity, which is necessary to match the performance demonstrated in [29].

To overcome this limitation and achieve fast nearest neighbour queries, $k$ nearest neighbours can be first obtained using the $L^2$-stream distance in $\mathcal{O}(\log |\mathcal{V}|)$ time. The $L^2$-LSB nearest neighbour can then be approximated by finding the minimum distance from among the $k$ candidates. The value of $k$ is chosen from [29] as $k_{\mathrm{RRG}} \log(|\mathcal{V}|)$.

## V. ADAPTIVE ARC-LENGTH

To improve coverage performance, we leverage the low dispersion properties of Halton sequences [37] to select sample positions in 2D. We connect to widely separated samples with forward integration with adaptive arc-lengths in an effort to maintain the properties discussed in [30].

Approaches that use forward integration in time with fixed horizons, such as [1], prioritise search depth and are useful for establishing long connections and finding initial feasible solutions. However, other approaches that use closely separated points for which the connection cost can be calculated analytically by approximating the flow between them as uniform (short horizons) prioritise breadth and can lead to higher path quality.

(a) Number of connections over iterations



(b) $L^2$ dispersion over iterations
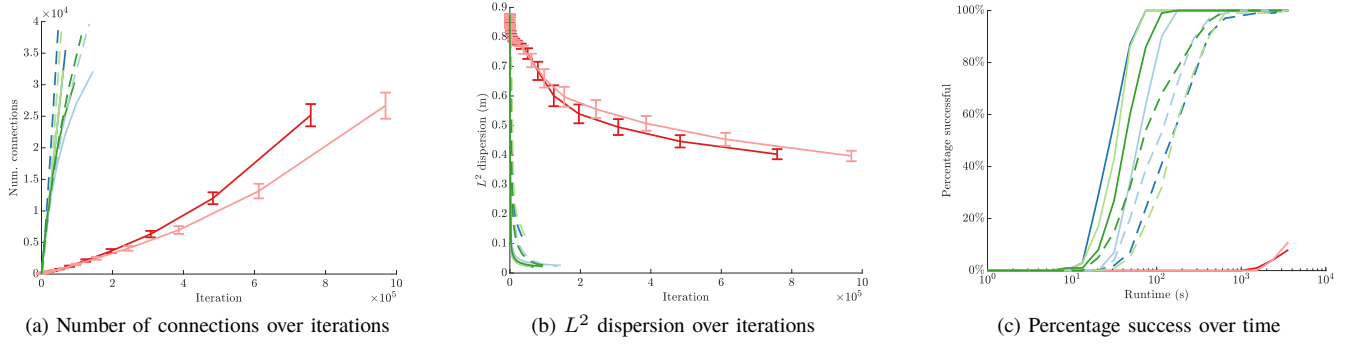


(c) Percentage success over time

Fig. 5. Comparison of results between different methods using RRT* after one hour of computation. The flow moves up to four times faster than the vehicle's maximum speed. Seven velocity samples were used for the adaptive arc-length approaches. Mean values are shown from using 100 randomly rotated and offset Halton sequences. Euclidean distance is shown in light blue, $L^2$-stream distance in dark green, $L^2$-LSB distance in dark blue, and approximated $L^2$-LSB distance in light green. The solid lines correspond to dynamic arc-length implementations, whilst dashed lines correspond to analytic steps. Bars showing 99.7% confidence intervals are restricted to VF-RRT* (red) [13] and RRT* (pink) as implemented in [29], for clarity.
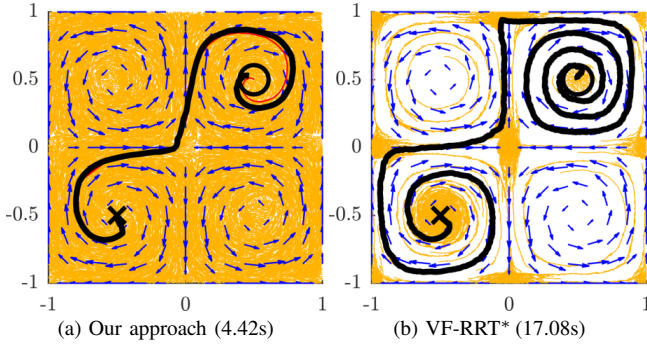


(a) Our approach (4.42s)  (b) VF-RRT* (17.08s)

Fig. 6. Examples of trees grown using $L^2$-LSB distance approximation with adaptive arc-length (left) and VF-RRT* [13] (right) in a strong vortex field after one hour of computation. In each case, the path from the start (cross) to the goal (circle) is shown as a black line. The underlying flow field is shown in blue and the final graph connections are shown in orange.

We consider a different mode of forward integration to enable adaptive horizon length. The discrete dynamics (1) can be expressed with the time step reformulated in terms of an equivalent Euclidean distance step $\|\Delta \mathbf{x}\|_2$, as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (\mathbf{v}(\mathbf{x}_k) + \mathbf{u}_k) \frac{\|\Delta \mathbf{x}\|_2}{\|\mathbf{v}(\mathbf{x}_k) + \mathbf{u}_k\|_2}. \quad (17)$$

This formulation allows us to heuristically upper bound the path length by the arc length of a semicircle with diameter equal to the Euclidean distance between the sample points. Given the current position $\mathbf{x}_P$ and next position $\mathbf{x}_Q$, the dynamic number of integration steps is thus

$$n_{\text{step}} = \text{ceil}\left(\frac{\pi \|\mathbf{x}_Q - \mathbf{x}_P\|_2}{2 \|\Delta \mathbf{x}\|_2}\right). \quad (18)$$

This variable stepping approach allows RRT* to inherit accurate integrated early weights for long connections that evolve towards fast analytic calculations for closely separated samples, to maintain path quality.

We use this adaptive arc-length approach to implement the primitive functions `Steer`, `CollisionFree`, and `Cost` in the RRT* framework [29]. In `Steer`, a velocity is selected optimistically from the intersection of (12) and (2) such that the vehicle moves towards the target. This selection corresponds to the red arrow in Fig. 3b. This is reasonable

since the gradient of (12) in velocity space corresponds to the displacement of $\mathbf{x}_Q$ from $\mathbf{x}_P$. If the maximum vehicle speed were much faster than the current ($V_{\max} \gg \|\mathbf{u}\|_2$), this control would asymptotically steer the vehicle directly towards the goal. In the case that the trajectory does not pass through $\mathbf{x}_Q$, the closest point on the trajectory (using `Nearest`'s distance metric) is returned instead.

Finally, we linearly sample velocities from the intersection of (12) and (2) to implement `CollisionFree` and `Cost`, as in [1]. In practice, only a few samples are needed due to the robustness of the approach.

## VI. Experiments

We compare our approach with the VF-RRT* algorithm [13] in strong flow fields to evaluate its performance. First, the quad-vortex flow example from [6, 18] is used to establish relative performance. Then, we consider an example with an actual oceanic flow prediction to demonstrate the implications of performance differences for a marine robotics application. The `Nearest` and `Near` functions of RRT* [29] are implemented without fast neighbourhood querying, which is left for future work. Furthermore we use a value of 1 for $\alpha$ and $\beta$. Selection of these values are left for future work. Each algorithm is allocated one hour of computation time on a single core of a 2.1 GHz Intel Xeon Platinum 8160 processor.

### A. Quad-vortex

The environment consists of four vortices that rotate in opposite directions with a maximum speed that is four times the vehicle's maximum speed. The flow field is illustrated by the blue arrows in Fig. 6. The vehicle starts at the centre of one vortex and its goal lies at the centre of the diagonally opposite vortex. The maximum tolerable integration distance $\|\Delta \mathbf{x}\|_2$ is $0.01\,\text{m}$, relative to a workspace size of $2\,\text{m}$.

We consider eight different variations of our method compared with RRT* (as implemented in [29]) and VF-RRT*. The variations of our method are pairwise combinations of different implementations of the `Nearest` function and edge connections. The four `Nearest` implementations are: Euclidean, $L^2$-stream, $L^2$-LSB, and its approximation. The

two edge connection implementations are: the adaptive arc-length forward integration, and analytical forward propagation [12, 13] bounded by $\|\Delta\mathbf{x}\|_2$. The $E_s$ parameter for VF-RRT* is tuned to 0.05 for this experiment.

Figure 5a shows the number of connections made during RRT* iterations. VF-RRT* (red) makes more connections than standard RRT* (pink), as expected, but all variations of our approach find connections at a higher rate. Euclidean distance (light blue) performs worse than the distance heuristics that account for the flow field; in particular, $L^2$-LSB distance (dark blue) has the highest connection rate with its approximated version (dark green) closely following. It should be noted that we expect the analytical steps approach (dashed) to have higher connection rates since it analytically determines locally reachable regions.

We see the advantages of the adaptive arc-length approach in Fig. 5b, where we compare how each approach covers the configuration space with (7). VF-RRT* again performs better than standard RRT*, and adaptive-arc length performs better than analytical steps. $L^2$-LSB has the best coverage, followed by its approximated version.

On average, our methods finds solutions orders of magnitude faster, as shown in Fig. 5c. This is a result of maintaining a tree that evenly covers the work space (Fig. a) in contrast to VF-RRT* (Fig. b) which becomes sparse in strong flows.

### B. East Australian Current

We now demonstrate the performance of using the approximated $L^2$-LSB distance with adaptive arc-length and VF-RRT* tuned to $E_s = 0.5$ in an application example. The Gaussian process (GP) method proposed in [22] is used to compute stream values from an ocean current prediction dataset provided by the Australian Bureau of Meteorology. The vehicle's maximum velocity is $1.1\,\mathrm{m/s}$, and the maximum flow velocity is $4.45\,\mathrm{m/s}$. The maximum tolerable integration distance $\|\Delta\mathbf{x}\|_2$ is $1\,\mathrm{km}$. Our approach was able to find a solution using a maximum vehicle velocity of $0.3\,\mathrm{m/s}$, but VF-RRT* failed in this case; we increased the value of this parameter to allow for comparison.

The first (dashed) and final (solid) paths from Sydney to Brisbane are illustrated in Fig. 7. Our approach found the first solution after 31 seconds of computation time with a path duration of 6 days and 2 hours. The best path found had a duration of 5 days and 4 hours. VF-RRT* found its first solution in 44 seconds with a path duration of 11 days and 5 hours. Its best path had a duration of 10 days and 22 hours. The great circle path with this vehicle in still water would take 7 days and 9 hours, implying that our approach utilises the flow field effectively.

The solution quality of VF-RRT* is poor due to its inability to connect new samples to an appropriate node in the existing tree. This hinders its ability to explore, leading to sparse trees similar to the one shown in 6b. Our approach covers the space more evenly and finds better solutions. The adaptive arc-length method also allowed it to find a feasible solution quickly for a problem of this spatial scale. Another consequence is that our method was able to improve its initial
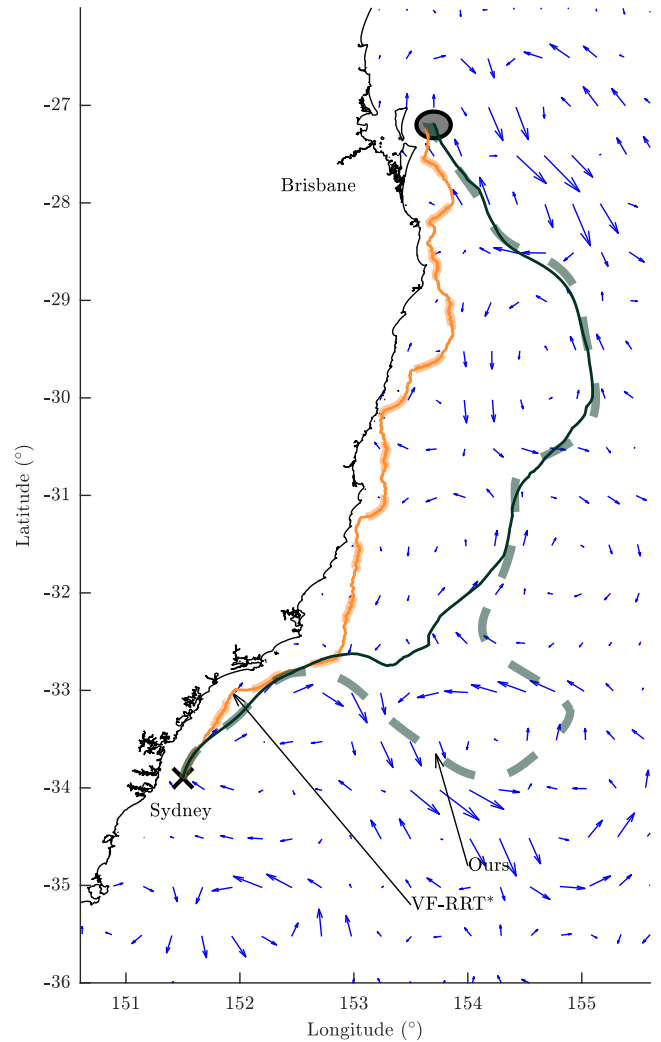


Fig. 7. Planning from Sydney to Brisbane against the mean north to south trend of the East Australian Current. The best path found by approximated $L^2$-LSB distance with adaptive arc-length is shown in dark green, and the path found by VF-RRT* [13] is shown in orange. The first feasible solutions found are shown as dashed lines.

solution within the given computation time, whereas VF-RRT* made only minor adjustments.

### VII. CONCLUSION AND FUTURE WORK

We have presented new distance functions and steering heuristics for planning in incompressible flows and evaluated them in the RRT* framework. Results in an artificial flow field showed that our method produces high-quality solutions and is computationally efficient. In a real-world example with actual ocean current prediction data, our method was able to exploit favourable flows to find a path that was not overwhelmed by the mean strong opposing current.

These results are promising and show potential for practical use in marine robotics. Important future work is to explore scaling terms $\alpha$ and $\beta$, incorporate recent work in efficient planning for time-varying flows [8, 38], extend to 3D spaces for plume localisation [22], and validate in real world scenarios where these complex flow exist.

## References

[1] K. Y. C. To, K. M. B. Lee, C. Yoo, S. Anstee, and R. Fitch, "Streamlines for motion planning in underwater currents," in *Proc. of IEEE ICRA*, 2019, pp. 4619–4625.

[2] K. Y. C. To, J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, "Streamline-based control of underwater gliders in 3D environments," in *Proc. of CDC*, 2019, pp. 1–8, accepted.

[3] 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.

[4] R. Bachmayer, N. E. Leonard, J. Graver, E. Fiorelli, P. Bhatta, and D. Paley, "Underwater gliders: Recent developments and future applications," in *Proc. of IEEE UT*, 2004, pp. 195–200.

[5] C. Yoo, S. Anstee, and R. Fitch, "Stochastic path planning for autonomous underwater gliders with safety constraints," in *Proc. of IEEE/RSJ IROS*, 2019.

[6] 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.

[7] G. K. Batchelor, *An Introduction to Fluid Dynamics*. Cambridge: Cambridge University Press, 2000.

[8] J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, "Hierarchical planning in time-dependent flow fields for marine robots," in *Proc. of IEEE ICRA*, 2020.

[9] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 166–171, 1998.

[10] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[11] F. Sukkar, "Fast, reliable and efficient database search motion planner (FREDS-MP) for repetitive manipulator tasks," Master's thesis, University of Technology Sydney, 2018.

[12] D. Rao and S. B. Williams, "Large-scale path planning for underwater gliders in ocean currents," in *Proc. of ARAA ACRA*, 2009, pp. 1–8.

[13] 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.

[14] J. H. Jeon, S. Karaman, and E. Frazzoli, "Anytime computation of time-optimal off-road vehicle maneuvers using the RRT*," in *Proc. of IEEE CDC-ECC*, 2011, pp. 3276–3282.

[15] B. Akgun and M. Stilman, "Sampling heuristics for optimal motion planning in high dimensions," in *Proc. of IEEE/RSJ IROS*, 2011, pp. 2640–2645.

[16] J. Barry, L. P. Kaelbling, and T. Lozano-Pérez, "A hierarchical approach to manipulation with diverse actions," in *Proc. of IEEE ICRA*, 2013, pp. 1799–1806.

[17] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. of IEEE/RSJ IROS*, 2014, pp. 2997–3004.

[18] 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, pp. 42–51.

[19] 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, 2006.

[20] 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, pp. 591–615, 2013.

[21] 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.

[22] K. M. B. Lee, C. Yoo, B. Hollings, S. Anstee, and R. Fitch, "Online estimation of ocean current from sparse GPS data for underwater vehicles," in *Proc. of IEEE ICRA*, 2019, pp. 3443–3449.

[23] E. Zermelo, *Über das Navigationsproblem bei ruhender oder Veränderlicher windverteilung*. WILEY-VCH Verlag, 1931, vol. 11, no. 2.

[24] 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*, vol. 64, no. 10, pp. 1373–1397, 2014.

[25] 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.

[26] B. Rhoads, I. Mezić, and A. Poje, "Minimum time feedback control of autonomous underwater vehicles," in *Proc. of IEEE CDC*, 2010, pp. 5828–5834.

[27] D. Kularatne, S. Bhattacharya, and M. A. Hsieh, "Time and energy optimal path planning in general flows," in *Proc. of RSS*, 2016, pp. 1–10.

[28] ——, "Optimal path planning in time-varying flows using adaptive discretization," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 458–465, 2018.

[29] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[30] L. Janson, B. Ichter, and M. Pavone, "Deterministic sampling-based motion planning: Optimality, complexity, and performance," *International Journal of Robotics Research*, vol. 37, no. 1, pp. 46–61, 2018.

[31] J. D. Hernández, E. Vidal, M. Moll, N. Palomeras, M. Carreras, and L. E. Kavraki, "Online motion planning for unexplored underwater environments using autonomous underwater vehicles," *Journal of Field Robotics*, vol. 36, no. 2, pp. 370–396, 2019.

[32] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, 1992.

[33] S. M. LaValle, *Planning Algorithms*. Cambridge: Cambridge University Press, 2006.

[34] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[35] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles," in *Proc. of ACM SIGMOD*, 1990, pp. 322–331.

[36] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching in fixed dimensions," *Journal of the ACM*, vol. 45, no. 6, pp. 891–923, 1998.

[37] J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numerische Mathematik*, vol. 2, pp. 84–90, 1960.

[38] J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, "Efficient optimal planning in non-FIFO time-dependent flow fields," in *arXiv:1909.02198 [cs.RO]*, 2019, pp. 1–10.