2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Hierarchical Planning in Time-Dependent Flow Fields for Marine Robots

James Ju Heon Lee[1], Chanyeol Yoo[1], Stuart Anstee[2] and Robert Fitch[1]

***Abstract*— We present an efficient approach for finding shortest paths in flow fields that vary as a sequence of flow predictions over time. This approach is applicable to motion planning for slow marine robots that are subject to dynamic ocean currents. Although the problem is NP-hard in general form, we incorporate recent results from the theory of finding shortest paths in time-dependent graphs to construct a polynomial-time algorithm that finds continuous trajectories in time-dependent flow fields. The algorithm has a hierarchical structure where a graph is constructed with time-varying edge costs that are derived from sets of continuous trajectories in the underlying flow field. We show that the continuous algorithm retains the time complexity and path quality properties of the discrete graph solution, and demonstrate its application to surface and underwater vehicles including a traversal along the East Australian Current with an autonomous marine vehicle. Results show that the algorithm performs efficiently in practice and can find paths that adapt to changing ocean currents. These results are significant to marine robotics because they allow for efficient use of time-varying ocean predictions for motion planning.**

I. INTRODUCTION

Autonomous operation is an important capability for a variety of marine robots, including autonomous surface vessels (ASVs) [12], underwater gliders [1, 19, 29], autonomous underwater vehicles (AUVs) [26], and hydrids of the these [2]. Motion planning problems for marine robots can represent the prevailing ocean currents as a flow field which influences the robot's motion [28]. Commonly available sources of ocean current predictions provide data at a sequence of discrete time points [14], but unfortunately these dynamic predictions are difficult to use in motion planning. Problems that involve time-varying costs are notoriously difficult to solve; the simplest general form is the *time-dependent shortest path* (TDSP) problem, which is defined over graphs and is known to be NP-hard [10]. In recent work we have introduced a new polynomial-time special case of TDSP where edge costs are a piecewise-constant function of time. [17]. We are interested in exploiting this result to find approximations to the shortest continuous paths in time-dependent flows.

Interest in planning that considers time-varying phenomena extends back over 50 years to the earliest introduction of shortest path problems on graphs [3]. A number of problem variants have been defined [4], but the most relevant case for marine robots is that in which delaying departure (waiting) can be beneficial. This is known as the non-FIFO (first-in-first-out) property. Efficient algorithms for non-FIFO TDSP have only recently been discovered [10, 17], and this presents an important opportunity for improving motion planning in time-dependent flow fields, such as ocean currents.

There are two main challenges in applying a TDSP solution to motion planning for marine robots. One challenge is how to represent the dynamic flow field as a time-dependent graph, where edge costs are represented as *edge time* functions. The other challenge is how to generate a continuous trajectory based on the optimal path in this graph while retaining the analytical guarantees of the discrete solution.

In this paper, we present a hierarchical algorithm that addresses these challenges. We assume that flow field predictions are provided as a series of snapshots over time. This aligns with the necessary piecewise-constant assumption required by our TDSP solution and with available ocean current prediction data. The top layer of the hierarchy is a graph representation of the flow field in the form of a TDSP problem, illustrated in Fig. 1. Nodes are chosen according to a uniform sampling of the workspace. Edge connections and edge time functions are constructed by the bottom layer of the hierarchy, which uses forward integration of a set of control values from each node to connect samples. This process is repeated at each flow field snapshot to construct edge time functions. The shortest path in this graph is then used to construct a corresponding continuous path.

Our main analytical result is that the algorithm retains the resolution completeness and optimality properties of the discrete layer overall, and also retains the polynomial time complexity of the TDSP solution. This is an important theoretical contribution because it avoids the need to construct a time-expanded graphs, where nodes are replicated at discrete steps. Although it may appear that our graph construction resembles the equivalent time-expanded graph representation, the important difference is that paths through this graph can arrive at nodes at any time; they are not restricted by the choice of time discretisation, which would be necessary when searching a time-expanded graph.

This algorithm also makes a meaningful contribution to motion planning for marine robots. We demonstrate the behaviour of our method in application examples with an ASV and an underwater glider in simulated dynamic flow fields and with 14 days of actual ocean current prediction data. These examples assume 2D flows, but there are no

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

[1]Authors are with the University of Technology Sydney, Ultimo, NSW 2006, Australia JuHeon.Lee@student.uts.edu.au and {Chanyeol.Yoo,Robert.Fitch}@uts.edu.au

[2]Author is with the Defence Science and Technology Group, Department of Defence, Australia Stuart.Anstee@dst.defence.gov.au

restrictions in the algorithm that would prevent its use with 3D flows. Results show how dynamic flows can have a negative effect on the performance of marine robot navigation. Paths found using static snapshots of the dynamic flow fail to reach the goal when executed in the time varying dataset, but our method finds a path that is able to exploit the prevailing currents as they change over time. In a traversal from Brisbane to Sydney along the East Australian Current, our method found a path of length 13.8 days, whereas the same vehicle would take over 27 days to traverse the great circle path in still water.

II. RELATED WORK

Common approaches to the problem we consider use level set methods [23, 24]. A more recent approach is to formulate the problem as a time-varying Markov decision process [22]. Both methods assume a general edge function, making the worst-case computation time for these methods non-polynomial in the non-FIFO case.

There has been other work in finding discrete time solutions using a time-expanded graph. Dijkstra [7] and A* [8, 9, 13] approaches to this problem have appeared. A recent improvement to the A* method has also been presented using adaptive sampling [16]. A common issue with searching in time-expanded graphs is that the solution is resolution complete in time; paths arrival times at nodes must align with the given time discretisation. This property encourages fine discretisation, which consequently increases the size of the problem representation and implies a corresponding increase in required computation time.

The problem can also be approached as a TDSP problem, where a solution is found by searching in graphs with time-dependent edge costs, known as time-dependent graphs [4, 10]. A*-based approaches for searching time-dependent graphs include those with precomputed heuristics [15] and bidirectional searches [6]. Time aggregate graphs [11], which represent the edge costs as time series, are another approach to the TDSP problem. A challenge to these approaches is that the environment must be sufficiently discretised for a high-quality path to be found.

Recently, we have proposed an efficient approach to optimal planning in time-dependent flow fields expressed as time-dependent graphs, which reaches a solution in polynomial time [17]. The algorithm can correctly find cyclic time-optimal paths in pathological flow regimes. We also build on our previous work that synthesises a continuous path given a sequence of discrete path states for time-invariant wind fields [33].

III. BACKGROUND

A. General dynamical model in a flow field

Given an n-dimensional vehicle state $\mathbf{x}$, time-dependent 3D flow field at the vehicle position $\mathbf{v}_c(\mathbf{x}, t) = [u_c(t), v_c(t), w_c(t)]^T$, and an m-dimensional control input $\mathbf{u}(t) \in \mathbf{U}$ where $\mathbf{U}$ is the set of all possible

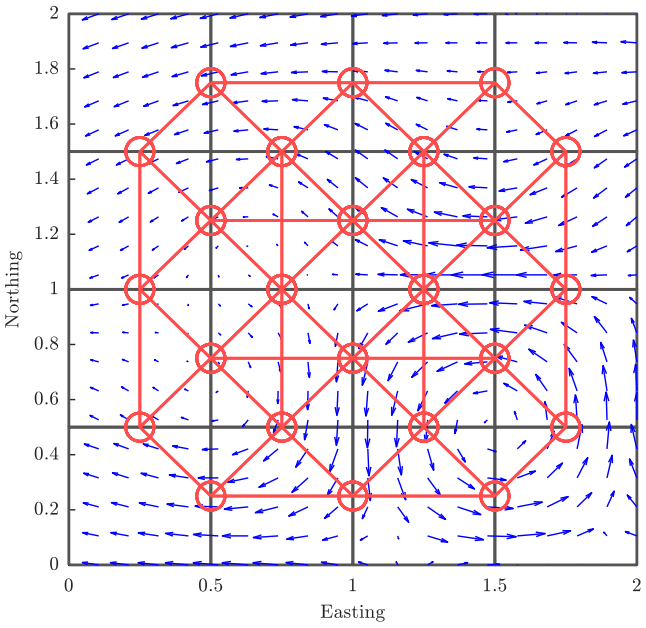


Fig. 1. Example time-dependent directed graph. The environment is evenly distributed into rectangular grids (shown in black). The red circles are the graph states, defined as the mid point of each line on the rectangle. The red line is the graph edge defined by the neighbouring states.

controls, the motion of a vehicle travelling in a 3D flow field at time t is represented as:

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

where $f(\mathbf{x}(t), \mathbf{u}(t))$ is the vehicle velocity in still water. The continuous trajectory the vehicle follows by executing the sequence of control inputs in the flow field is denoted as σ.

The vehicle is controlled using a sparse velocity control scheme [18, 21]; controls are only varied at specific times, rather than attempting to continuously adjust for the spatial variation of the flow field. Intuitively, this is a *"let it go"* approach where we apply a control input and then make no further adjustment for a set period of time.

B. Time-dependent directed graph

A time-dependent directed graph $G = (S, E)$ consists of finite sets of states S and edges $(s, s') \in E$ where $s, s' \in S$. For each edge (s, s'), we associate an *edge time* $C_{ss'}(t)$ that represents the time to traverse from state s to s'. For all pairs of edges, $C_{ss'}(t)$ is ∞ for all $t < 0$. We denote the set of states that are immediately reachable from s as $S_s \subseteq S$ where $s' \in S_s$, and the set of goal states as $S_g \subset S$ where $|S_g| \geq 1$. We assume waiting is not allowed in the graph (i.e. an edge must be traversed as soon as it is encountered) as it is difficult for an underwater glider to stay in the same position.

We define an $(n+1)$-state discrete path $\psi = s_0 s_1 \cdots s_n$ within S as a sequence of states, where $s_k \in S \backslash S_g$, $s_n \in S_g$, and $(s_k, s_{k+1}) \in E$ for $k = 0, 1, \cdots, n-1$. We denote ψ_k as the prefix of ψ up to the k-th state in the path (i.e., $\psi_k = s_0 s_1 \cdots s_k$), noting that it can start no earlier than $t = 0$.

C. Arrival and travel times

Arrival time, $a_{ss'}(t)$, is defined as the time that a vehicle would arrive at s' when departing from s at time t, i.e. $a_{ss'}(t) = C_{ss'}(t) + t$ [4]. As waiting is not allowed, the arrival time at the end of the $(n+1)$-length discrete path, $a_\psi(t)$, can be expressed as the arrival time at s_n departing from s_{n-1} at time $a_{s_{n-2}s_{n-1}}(t)$. The arrival time at the end of ψ can be recursively expressed as: $a_\psi(t) = a_{s_{n-1}s_n}(a_{\psi_{n-1}}(t))$. The total *travel time*, T_ψ, is simply

$$T_\psi(t_0) = a_\psi(t_0) - t_0, \tag{2}$$

which is the time taken to traverse the discrete path ψ.

IV. Problem Statement

The problem is to find a continuous path that minimises the time to travel from initial to goal position. Let a_σ denote the arrival time and $T_\sigma = a_\sigma(t) - t$ denote the travel time for the continuous path σ. The problem can be viewed from two perspectives:

Problem 1 (Minimising travel time)**.** *Given vehicle dynamics, time-dependent flow field $\mathbf{v}_c$, starting state $\mathbf{x}_{init}$, and a goal region $\mathbf{X}_g$, find the continuous path σ^* and starting time t_0^* that minimises the travel time:*

$$\begin{aligned} &(\sigma^*, t_0^*) = \arg\min_{\sigma,t} T_\sigma(t) \\ &\textit{s.t. } \mathbf{x}(t_0^*) = \mathbf{x}_{init} \textit{ and } \mathbf{x}(a_{\sigma^*}(t_0^*)) \in \mathbf{X}_g. \end{aligned} \tag{3}$$

Problem 2 (Minimising travel time for given start time)**.** *Given vehicle dynamics, time-dependent flow field $\mathbf{v}_c$, starting state $\mathbf{x}_{init}$, starting time t_0 and a goal region $\mathbf{X}_g$, find the continuous path σ^* that minimises the travel time:*

$$\begin{aligned} &\sigma^* = \arg\min_{\sigma} T_\sigma(t_0) \\ &\textit{s.t. } \mathbf{x}(t_0) = \mathbf{x}_{init} \textit{ and } \mathbf{x}(a_{\sigma^*}(t_0)) \in \mathbf{X}_g. \end{aligned} \tag{4}$$

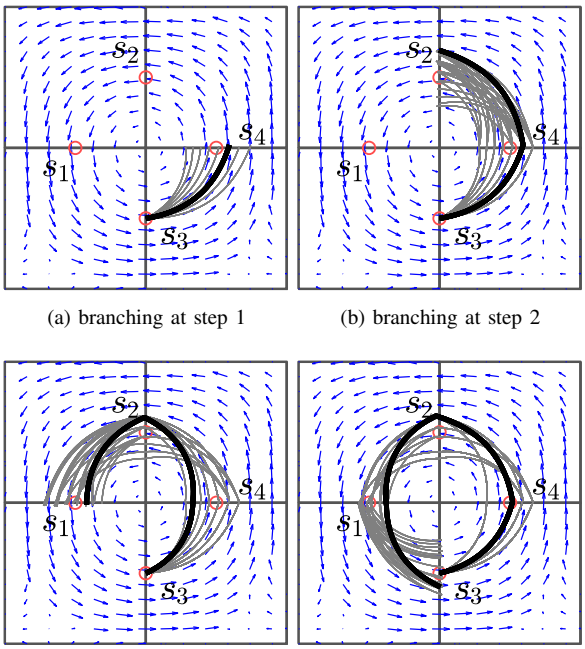
The two problems are similar to the TDSP problem on a time-dependent directed graph, as shown in our previous work [17]. We extend the problem by solving for a continuous path, where the time dependency arises from a dynamic flow field that is one way to represent an ocean current.

V. Time-dependent shortest path in flow fields

In this section, we present our hierarchical path planning framework. We first build a time-dependent directed graph over a continuous time-dependent flow field. We then find the time-optimal path in this graph in the form of a policy. Finally, we generate a continuous path that follows the time schedule set by our discrete solution.

A. Building a time-dependent graph in a dynamic flow field

In this section, we present how to express the continuous flow field as a time-dependent directed graph. Given a continuous environment space $\mathbf{X} \subset \mathbf{R}^2$, we discretise it into uniform regions. We define each line segment of the discretisation as a *state line*, ℓ_s, and situate the set of graph states $s \in S$ at the mid-points of those ℓ_s that are not part of the external boundary, as illustrated by example in Fig. 1.



(a) branching at step 1 (b) branching at step 2

(c) branching at step 3 (d) branching at step 4

Fig. 2. Step-by-step visualisation of an example of a continuous path branching that follows a discrete path. N closest solutions to the discrete travel time are kept for each step. The trajectory with minimal travel time is highlighted in black.

The set of neighbouring states S_s for all $s \in S$ are defined as the states that lie on the region edges that adjoin ℓ_s.

The associated edge time $C_{ss'}(t)$ for each edge $(s, s') \in E$ is evaluated through forward integration using the average vehicle speed $\mathbf{v}_v$ over the set of controls $\mathbf{U}$; derived from the glider dynamics and over the time-dependent flow field $\mathbf{v}_c$. We express $C_{ss'}(t)$ as a piecewise function with uniform time partitioning of the flow field data up to the time horizon t_H. The time partitioning is defined as $t_\Delta = (t_H - t_0)/Q$ where $Q \in \mathbb{Z}$. For each state $s' \in S_s$ at time $t_k = q(t_H - t_0)/Q + t_0$, $q = 0, \cdots, Q$, we enumerate a set of possible heading controls $\Phi = \{\phi_0, \cdots\}$ and set the time for the fastest control to cross $\ell_{s'}$ as the edge time for that input time. We show later that this approach has minimal impact on the time complexity.

B. Solving for TDSP across time-dependent graph

We solve the TDSP problem for a time-dependent graph as demonstrated in our previous work [17]. In this section, we summarise the implementation for completeness.

The travel time function (2) can be re-expressed as $T_s^k(t)$, the travel time after k edge transitions from s. The travel time function can then be written recursively as:

$$T_s^{k+1}(t) = C_{ss'}(t) + T_{s'}^k(t + C_{ss'}(t)). \tag{5}$$

Denoting $T_s^*(t)$ as the converged travel time, where $T_s^{k+1}(t) = T_s^k(t)$ for all $t \in \mathbb{R}$ and some finite $k \in \mathbb{Z}$, we can choose the next neighbouring state s_i that minimises

total travel time of the discrete path:

$$s_i = \arg\min_{s' \in S_s} C_{ss'}(t) + T^*_{s'}(t + C_{ss'}(t)). \tag{6}$$

Now we can derive the optimal time-dependent policy for each $s \in S$:

$$\pi^*_s(t) = \begin{cases} \vdots & \vdots \\ s_{ii} \in S_s & \text{else if } t > t_1 \\ s_i \in S_s & \text{if } t > t_0 \end{cases}. \tag{7}$$

Based on the policy, we can find the time-optimal discrete path $\psi^* = s_0 s_1 \cdots s_n$.

C. Hierarchical planning for continuous TDSP

In this section, we present how to realise a continuous path that follows the time schedule of the time-optimal discrete path ψ^*. We denote $\mathbf{U} = \{\theta_0, \cdots\}$ as the finite set of demand heading values the vehicle controller can select.

Starting from the initial state s_0 at time t_0, we forward integrate for all controls $u \in U$ to find a set of trajectories that reaches the state line, ℓ_{s1}. To limit the number of branching trajectories we keep only the N best, i.e., the N trajectories with closest total travel time to $T_{\psi^*_1}$. We branch each of the N trajectories to the next state by forward integrating to the next state line, pruning again for the N best trajectories. Once we reach the end of the discrete path, we pick the trajectory with the fastest total travel time.

A visualisation of an example case is shown in Fig. 2. Suppose we were to generate a continuous path from a discrete path $\psi = s_3 s_4 s_2 s_1 s_3$. We first propagate the path across the control input from s_3, pick N trajectories that intersect ℓ_{s_4} and have the closest total travel time T_{ψ_1}, and prune away the rest. We extend each trajectory by propagating new paths from the trajectory's end point and continue the process for ℓ_{s_2}. After each extension, more trajectories are pruned away to keep only the N best solutions.

VI. ANALYSIS

It is important to note that solving for general non-FIFO TDSP over discrete space (i.e., graph) is NP-hard [5, 25]. The problem becomes even harder with continuous state space.

Our hierarchical approach divided the problem into two parts, discrete path and continuous path. In our previous work [17], we have shown that the discrete path ψ gives the time optimal solution in polynomial computation time. Since our continuous path is generated by finding a uniform control within each region in the discrete sequence, the final continuous path is optimal with respect to discrete regions. The result implies that as we reduce the size of each region, our continuous path would also approach the optimal in continuous state space.

In continuous path generation, our aim is to find a control that triggers a transition to the designated next region within a time window. Under the assumption that the flow within a region is uniform, there always exists a continuous control from one edge to another. Therefore our framework is complete in both discrete and continuous state space.

The time complexity for finding an optimal discrete path is $\mathcal{O}(|S||C|^k + |\Phi||S|)$, where $|C|$ is the worst case number of variations in flow, k is number of edge transitions, $|\Phi|$ is the number of discrete controls and $|S|$ is the number of discrete sequences. Given a discrete sequence ψ, we find a continuous path for each region for $|\psi|$ times. Therefore, the time complexity for finding a continuous path is $\mathcal{O}(|\psi||U|)$. The overall complexity is $\mathcal{O}(|S||C|^k + |\Phi||S| + |\psi||U|)$. Since edge costs are found by enumerating all controls U and Ψ for forward integration, the part can be parallelised. The overall complexity with parallel processing is $\mathcal{O}(|S||C|^k + |S| + |\psi|) = \mathcal{O}(|S||C|^k + |\psi|)$. In either case, the time complexity is linear in the number of discrete regions and polynomial in the number of variations in flow.

VII. CASE STUDIES

In this section, we present 3 case studies: an ASV travelling in a 2D time-dependent flow field, an underwater glider travelling in a 3D time-dependent flow field, and a real-world example where an ASV traverses the East Australian Current (EAC) to find the shortest path from Brisbane to Sydney. The case studies demonstrate our algorithm's ability to handle various dimensions, dynamics, and problem sizes. All distance units are in meters and time units are in seconds unless otherwise stated.

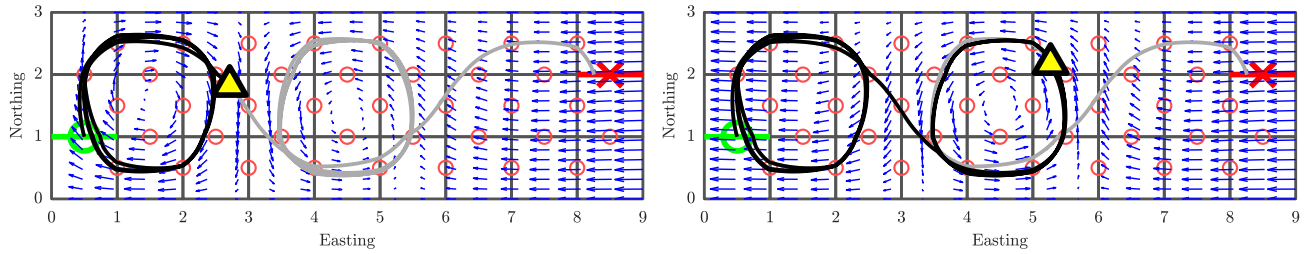
A. Autonomous surface vehicle (ASV) case

In this case study, we generate a continuous path that follows the time-optimal discrete path starting from $t_0 = 0$. We also evaluate the optimal start time t^*_0 using a standard optimisation technique, and generate a continuous path for that start time.

We consider an ASV with a constant forward velocity of 0.5 m/s travelling from $[0.5, 1]^T$ to $[8.5, 2]^T$ through a 2D time-dependent flow field. The planning environment is dominated by a westward current of up to 1.2 m/s around the goal state that makes it impossible for the ASV to travel directly east towards the goal. Vortices with radius 1.5 m appear at $[1.5, 1.5]^T$ for $t \in [0, 15]$, at $[4.5, 1.5]^T$ for $t \in [10, 30]$, and at $[7.5, 1.5]^T$ for $t \in [25, \inf)$. To solve the for the time-optimal discrete path, the environment was discretised into a 9x3 grid, yielding a graph with 42 states.

Figures 3a to 3c show time instants along the continuous path starting at $t_0 = 0$. The travel time for discrete and continuous path were $T_\psi = 31.95$ and $T_\sigma = 32.53$. The ASV loops around the first vortex 3 times to wait for the time when the transition from the first to the second vortex is possible. After transitioning to the second vortex, the paths cycles another 3 times until it is feasible to transition into the third vortex and reaches the goal line. This shows how cyclic paths are used to loiter in anticipation of advantageous transitions in the flow field.

Figure 3d shows the trajectory for the optimal starting time $t^*_0 = 9$ with travel times for discrete and continuous path $T_\psi = 19.75$ and $T_\sigma = 20.1$. The path shows its best to start near the end of the first vortex period.



(a) ASV path at 13.5 second after departing at $t_0 = 0$ departure

(b) ASV path at 23.5 second after departing at $t_0 = 0$

(c) ASV path at 32.53 second after departing at $t_0 = 0$

(d) ASV path at 19.75 second after departing at $t_0 = 9$

Fig. 3. 2D continuous path of an ASV from $[0.5, 1]^T$ to $[8.5, 2]^T$ with a constant velocity of 0.5 m/s. A strong current flowing west opposes the glider's path to the goal. Figures (a)-(c) show paths for starting times in steps starting at $t = 0$. Figure (d) is the final path when starting at the optimal start time $t_0^* = 9$

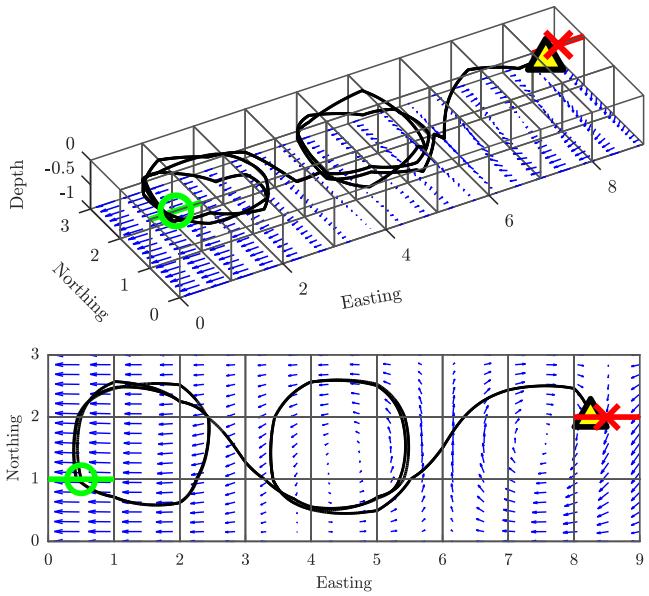


Fig. 4. TDSP across a time-varying 3D flow field with underwater glider dynamics. The travel times for the discrete and continuous paths were both 15.4 sec. The sample points are omitted for clarity. The trajectory and the flow field at the final time step are shown.

B. Underwater glider in 3D ocean current case

In this case, we used underwater glider dynamics to generate the continuous path that follows the time-optimal discrete path commencing at $t_0 = 0$. We consider a hypothetical underwater glider with a top horizontal speed of 0.868 m/s. The edge time was evaluated using an average glider horizontal speed of 0.7435 m/s.

The glider typically operates in a sawtooth motion, controlling its glide angle to control its net velocity. The sparse control scheme for the glider model can be implemented as a *trim-state control*. A *trim-state* is the state of dynamic equilibrium that is maintained under no disturbance or control variation. A description of 3D glider trim control can be found in our previous work [18].

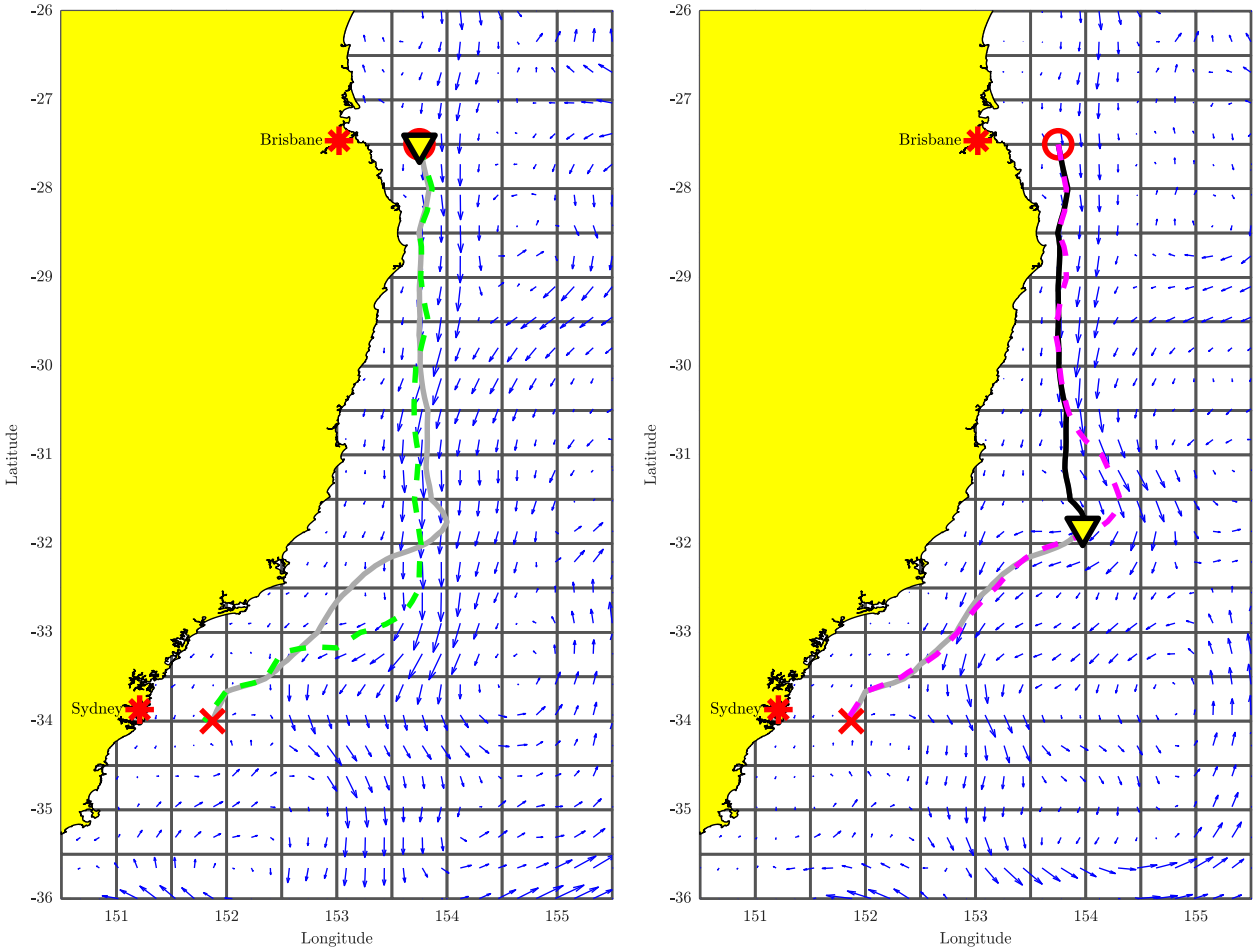
Both the flow field environment and the time-dependent graph are similar to those described in Sec. VII-A, extended in depth to allow for manoeuvre in 3D. Consequently, the state line ℓ_s is extended into a *state plane*, $\mathcal{P}_s$. For simplicity, we set the flow velocity to be the same for all depths. For the edge time evaluation $C_{ss'}(t)$, a set of trajectories is propagated in 3D from s using forward integration, and we pick the time that the fastest trajectory intersects with $\mathcal{P}_{s'}$. The start and goal positions are the same as in the previous.

Figure 4 shows the continuous 3D path with $t_0 = 0$, and travel times for discrete and continuous path $T_\psi = T_\sigma = 15.4$. The continuous trajectory illustrates properties of the glider dynamics, as the manoeuvres exhibit a sawtooth motion. Otherwise, we see the same decision making we saw with the ASV example: the glider loops around the first vortex twice before entering the second vortex, where it loops twice again until the final transition to the goal is feasible. Furthermore, as the glider velocity can be controlled by the sawtooth motion, we have greater control on T_σ

C. Real world ocean current case

In this case study, we generated a continuous path along the East Australian Current (EAC) that follows the time-optimal discrete path at $t_0 = 0$. We consider an ASV with a constant forward velocity of 0.3 m/s travelling from Brisbane $(-27.5°, 154°)$ to Sydney $(-34°, 151.5°)$.

The flow field data is taken from a numerical hindcast [14]. The data set was obtained from the School of Mathematics and Statistics at the UNSW. To solve the for the time-optimal



(a) 0 days

(b) 7 days

Fig. 5. 2D continuous path from Brisbane to Sydney in a time-dependent East Australian Current representation generated by the Australian Bureau of Meteorology. The vehicle velocity is set to a constant value of 0.3 m/s. The travel time for discrete and continuous path were $T_\psi = 13.375$ days and $T_\sigma = 14.7$ days. The red circle is the vehicle's starting point and the red cross is its end point. The green and magenta dashed lines are the paths the vehicle would have followed if the flow field were fixed at its states at $t = 0$ days and $t = 7$ days, respectively. The environment was discretised on a 20x10 grid, shown in black.

discrete path, the environment was discretised into a 20x10 grid, yielding a graph with 370 states.

Figure 5 shows the initial and final time steps of the resulting continuous ASV path. The travel time was $T_\psi = 13.375$ days and $T_\sigma = 14.7$ days. The ASV path follows the movement of the strongest part of the EAC to minimise the time to the goal. We also compared our result with paths generated in time-varying flow field using control sequence appropriate for time-invariant flow fields snapshot at $t = 0$ days and $t = 7$ days. Applying the control for snapshot at $t = 0$ results in the vehicle colliding with the Australian coast after 11.35 days, and applying the controls using snapshot at $t = 7$ results in the vehicle completing its trajectory 130.2464 km away from the goal. In comparison, our result delivers the ASV to a point 11.2504 km away from the goal. This relatively minor position error is due to approximations made when generating the continuous path.

The computation time was also favourable; we were able to solve for the discrete path in 440.77 sec and the continuous path in 642.26 sec. The computation time can be tuned by reducing the N value and the time step when propagating the continuous path.

VIII. CONCLUSION AND FUTURE WORK

In this paper we have presented an efficient algorithm for finding shortest continuous paths in flow fields that is based on a polynomial-time solution to the TDSP. It has analytical guarantees on path quality and time complexity. We have showed that the algorithm exhibits favourable computation time performance in practice, is effective in finding paths that exploit advantageous ocean currents, and is applicable to several types of marine robots.

Important avenues for future work include improving the algorithm through smarter flow field heuristics for edge cost computation [27] and adaptive discretisation and minimising energy use as opposed to the time length of the path [31]. We would also like to extend our work on planning with forecast data. Such planning would need to consider uncertainty [20, 30, 32] in the flow field predictions. We also intend to conduct field trials to evaluate our methods in practice.

REFERENCES

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

[2] B. Claus, R. Bachmayer, and C. D. Williams, "Development of an auxiliary propulsion module for an autonomous underwater glider," *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 224, no. 4, pp. 255–266, 2010.

[3] K. L. Cooke and E. Halsey, "The shortest route through a network with time-dependent internodal transit times," *J. Math. Anal. Appl.*, vol. 14, no. 3, pp. 493–498, 1966.

[4] B. C. Dean, "Shortest paths in FIFO time-dependent networks: Theory and algorithms," MIT, Tech. Rep., 2004.

[5] D. Delling and D. Wagner, "Time-dependent route planning," in *Robust and Online Large-Scale Optimization*, R. K. Ahuja, R. H. Möhring, and C. D. Zaroliagis, Eds., vol. 5868 of Lecture Notes in Computer Science. Springer, 2009, pp. 207–230.

[6] U. Demiryurek, F. Banaei-Kashani, C. Shahabi, and A. Ranganathan, "Online computation of fastest path in time-dependent spatial networks," in *Proc. of SSTD*, 2011, pp. 92–111.

[7] S. E. Dreyfus, "An Appraisal of Some Shortest-Path Algorithms," *Oper. Res.*, vol. 17, no. 3, pp. 395–412, 1969.

[8] E. Fernndez-Perdomo, J. Cabrera-Gmez, D. Hernndez-Sosa, J. Isern-Gonzlez, A. C. Domnguez-Brito, A. Redondo, J. Coca, A. G. Ramos, E. . Fanjul, and M. Garca, "Path planning for gliders using Regional Ocean Models: Application of Pinzn path planner with the ESEOAT model and the RU27 trans-Atlantic flight data," in *Proc. of IEEE OCEANS*, 2010, pp. 1–10.

[9] E. Fernndez-Perdomo, D. Hernndez-Sosa, J. Isern-Gonzlez, J. Cabrera-Gmez, A. C. Domnguez-Brito, and V. Prieto-Maran, "Single and multiple glider path planning using an optimization-based approach," in *Proc. of IEEE OCEANS*, 2011, pp. 1–10.

[10] L. Foschini, J. Hershberger, and S. Suri, "On the complexity of time-dependent shortest paths," *Algorithmica*, vol. 68, no. 4, pp. 1075–1097, 2014.

[11] B. George and S. Shekhar, "Time-aggregated graphs for modeling spatio-temporal networks," *J. Semantics Data*, vol. 9, pp. 191–212, 2008.

[12] R. Hine, S. Willcox, G. Hine, and T. Richardson, "The wave glider: A wave-powered autonomous marine vehicle," in *OCEANS 2009*. IEEE, 2009, pp. 1–6.

[13] J. Isern-Gonzalez, D. Hernandez-Sosa, E. Fernandez-Perdomo, J. Cabrera-Gamez, A. C. Domnguez-Brito, and V. Prieto-Maranon, "Path planning for underwater gliders using iterative optimization," in *Proc. of IEEE ICRA*, 2011, pp. 1538–1543.

[14] C. Kerry, B. Powell, M. Roughan, and P. Oke, "Development and evaluation of a high-resolution reanalysis of the East Australian Current region using the Regional Ocean Modelling System (ROMS 3.4) and Incremental Strong-Constraint 4-Dimensional Variational (IS4D-Var) data assimilation," *Geoscientific Model Development*, vol. 9, no. 10, pp. 3779 – 3801, 2016.

[15] S. Kontogiannis and C. Zaroliagis, "Distance Oracles for Time-Dependent Networks," *Algorithmica*, vol. 74, no. 4, pp. 1404–1434, 2016.

[16] D. Kularatne, S. Bhattacharya, and M. A. Hsieh, "Optimal Path Planning in Time-Varying Flows with Forecasting Uncertainties," in *Proc. of IEEE ICRA*, 2018, pp. 1–8.

[17] J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, "Efficient Optimal Planning in non-FIFO Time-Dependent Flow Fields," *arXiv e-prints*, p. arXiv:1909.02198, Sep 2019.

[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.

[19] K. M. B. Lee, J. J. H. Lee, C. Yoo, B. Hollings, and R. Fitch, "Active perception for plume source localisation with underwater gliders," in *Proc. of ARAA ACRA*, 2018.

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

[21] N. Leonard and J. Graver, "Model-based feedback control of autonomous underwater gliders," *IEEE J. Oceanic Eng.*, vol. 24, no. 4, pp. 633–645, 2001.

[22] L. Liu and G. S. Sukhatme, "A Solution to Time-Varying Markov Decision Processes," in *Proc. of IEEE ICRA*, 2018.

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

[24] T. Lolla, M. P. Ueckermann, K. Yigit, P. J. Haley, and P. F. J. Lermusiaux, "Path planning in time dependent flow fields using level set methods," in *Proc. of IEEE ICRA*, 2012, pp. 166–173.

[25] A. Orda and R. Rom, "Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length," *J. ACM*, vol. 37, no. 3, pp. 607–625, 1990.

[26] R. P. Stokey, A. Roup, C. von Alt, B. Allen, N. Forrester, T. Austin, R. Goldsborough, M. Purcell, F. Jaffre, G. Packard, and A. Kukulya, "Development of the REMUS 600 autonomous underwater vehicle," in *Proc. of IEEE OCEANS*, 2005, pp. 1301–1304.

[27] K. Y. C. To, C. Yoo, S. Anstee, and R. Fitch, "Distance and steering heuristics for streamline-based flow field planning," in *Proc. of IEEE ICRA*, 2020, pp. 1–7.

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

[29] 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.

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

[31] C. Yoo, R. Fitch, and S. Sukkarieh, "Probabilistic temporal logic for motion planning with resource threshold constraints," in *Proc. of RSS*, 2012.

[32] ——, "Provably-correct stochastic motion planning with safety constraints," in *Proc. of IEEE ICRA*, 2013, pp. 981–986.

[33] ——, "Online task planning and control for fuel-constrained aerial robots in wind fields," *Int. J. Robot. Res.*, vol. 35, no. 5, pp. 438–453, 2016.