frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 14 July 2021
doi: 10.3389/frobt.2021.575267

Bounded Cost Path Planning for Underwater Vehicles Assisted by a Time-Invariant Partitioned Flow Field Model

Mengxue Hou[1], *Sungjin Cho*[2], *Haomin Zhou*[3], *Catherine R. Edwards*[4] *and Fumin Zhang*[1]*

[1]Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, United States, [2]Department of Guidance and Control, Agency for Defense Development, Daejeon, South Korea, [3]School of Mathematics, Georgia Institute of Technology, Atlanta, GA, United States, [4]Skidaway Institute of Oceanography, University of Georgia, Savannah, GA, United States

A bounded cost path planning method is developed for underwater vehicles assisted by a data-driven flow modeling method. The modeled flow field is partitioned as a set of cells of piece-wise constant flow speed. A flow partition algorithm and a parameter estimation algorithm are proposed to learn the flow field structure and parameters with justified convergence. A bounded cost path planning algorithm is developed taking advantage of the partitioned flow model. An extended potential search method is proposed to determine the sequence of partitions that the optimal path crosses. The optimal path within each partition is then determined by solving a constrained optimization problem. Theoretical justification is provided for the proposed extended potential search method generating the optimal solution. The path planned has the highest probability to satisfy the bounded cost constraint. The performance of the algorithms is demonstrated with experimental and simulation results, which show that the proposed method is more computationally efficient than some of the existing methods.

Keywords: robotic path planning, graph search method, bounded cost search, parameter identification, underwater vehicle

OPEN ACCESS

Edited by:
Victor Zykov, Independent researcher, Menlo Park, CA, United States

Reviewed by:
Navid Razmjooy, Independent researcher, Ghent, Belgium
Guangjie Han, Hohai University, China

***Correspondence:**
Fumin Zhang
fumin@gatech.edu

Specialty section:
This article was submitted to Multi-Robot Systems, a section of the journal Frontiers in Robotics and AI

Received: *23 June 2020*
Accepted: *17 June 2021*
Published: *14 July 2021*

Citation:
Hou M, Cho S, Zhou H, Edwards CR and Zhang F (2021) Bounded Cost Path Planning for Underwater Vehicles Assisted by a Time-Invariant Partitioned Flow Field Model. Front. Robot. AI 8:575267. doi: 10.3389/frobt.2021.575267

1 INTRODUCTION

Over the last few decades, autonomous underwater vehicles (AUVs) have been employed for ocean sampling (Leonard et al., 2010; Smith et al., 2010), surveillance and inspection (Ozog et al., 2016; Xiang et al., 2010), and many other applications. Ocean flow is the dominant factor that affects the motion of AUVs (Zhang et al., 2016). Ocean flow dynamics vary in both space and time, and can be represented as geophysical Partial Differential Equations (PDEs) in ocean circulation models (e.g., the Regional Ocean Modeling System (ROMS, Shchepetkin and McWilliams, 2005; Haidvogel et al., 2008) and the Hybrid Coordinate Ocean Model (HYCOM, Chassignet et al., 2007)). While these models can provide flow information over a large spatial domain and forecast over several days, the available flow field forecast may still contain high uncertainty and error. The uncertainty comes from multiple sources, including the incomplete physics or boundary conditions (Haza et al., 2007; Griffa et al., 2004) and even terms in the equations themselves (Lermusiaux, 2006). In addition, the high complexity of the flow dynamics makes solving these PDEs computationally expensive. Data-driven flow models (Mokhasi et al., 2009; Chang et al., 2014) can provide short-term flow prediction in a relatively smaller area with significantly lower computational cost, and can be more suitable for

supporting real-time AUV navigation, particularly for systems with strong gradients and/or high uncertainty.

Path planning is one of the crucial and fundamental functions to achieve autonomy. Two key considerations for an AUV path planner are computational efficiency and path quality. The path planning strategy should be computationally efficient so that the time for generating a path can be kept to a minimum. When these methods are sufficiently fast, path planning can be performed in near-real time, generating a feasible solution in minutes while the AUV has surfaced to get a GPS fix and communicate with shore. Advantages of real time path planning are that more recent information, including the real time data from the vehicle, can be incorporated in path planning. Hence there will be less planning error due to outdated information (Leonard et al., 2010). At the same time, it is desired that the path planning algorithm has theoretical guarantee on the quality of the generated path.

Most path planning algorithms aim to design optimal path minimizing certain cost, for example, those associated with engineering or flight characteristics (battery life, travel time) or scientific value (e.g., distance relative to other assets or spacing of relevant processes). Algorithms that have been applied to AUV optimal path planning include: 1) graph-based methods such as the A* method (Rhoads et al., 2012; Pereira et al., 2013; Kularatne et al., 2017, 2018) and the Sliding Wavefront Expansion (SWE) (Soulignac, 2011); 2) sampling-based methods like the Rapidly exploring Random Trees (RRTs) (Kuffner and LaValle, 2000; Cui et al., 2015), RRT* (Karaman and Frazzoli, 2011) and informed RRT* (Gammell et al., 2018); 3) methods that approximate the solution of HJ (Hamilton-Jacobi) equations, such as the Level Set Method (LSM) (Subramani and Lermusiaux, 2016; Lolla et al., 2014), and 4) the evolutionary algorithms, including the particle swarm optimization methods (Roberge et al., 2012; Zeng et al., 2014), and the differential evolution methods (Zamuda and Sosa, 2014; Zamuda et al., 2016). See (Zamuda and Sosa, 2019; Zeng et al., 2015; Panda et al., 2020) for a comprehensive review on the existing AUV path planning methods. However, the computational cost of the above mentioned methods could be high, especially in cases where the AUV deployment domain is large.

Using a regular grid to discretize the flow field can result in unnecessary large number of cells, which increases the computational burden of the graph search methods. Since the flow speed in adjacent cells is usually similar, we partition the flow field into piece-wise constant subfields, within each the flow speed is a constant vector, and introduce the Method of Evolving Junctions (MEJ) (Zhai et al., 2020) to solve the optimal path planning problem. MEJ solves for the optimal path by recasting the infinite dimensional path planning problem into a finite dimensional optimization problem through introducing a number of junction points, defined as the intersection between the path and the region boundaries. Hence the computation cost of MEJ is significantly lower than other optimal path planning methods, especially when the flow field is partitioned into a small number of cells (Zhai et al., 2020). We identify Soulignac. (2011) as the work closest related to ours, in which sliders, defined as points sliding on the partitioned region boundaries, are introduced to describe the wavefront expansion of the graph search methods. In each iteration of the wavefront expansion, each slider's position on the wavefront is derived by minimizing the travel cost in a single cell, and then the planned path is computed by the backtracking of wavefronts. Both MEJ and SWE are based on a novel parameterization of the path by introducing the junctions and the sliders, that were discovered independently by the two research groups. The main difference between MEJ and SWE lies in that MEJ solves for junction positions by formulating a non-convex optimization problem, and derives the global minimizer by intermittent diffusion (Li et al., 2017), which intermittently adds white noise to the gradient flow, while SWE solves for slider positions by graph search methods. MEJ has been justified to find the global minimizer with probability 1. However, since the method does not pose any structure in the search, the computational cost of MEJ could be less favorable compared to SWE if the number of cells scales up.

To reduce the computational cost of the path planning problem, the search can be reduced to find paths with total cost less than an upper bound. Stern et al. (2014) present two algorithms to solve the bounded cost search problem: the Potential Search (PTS) and the Anytime Potential Search (APTS). The PTS method defines the Potential Ordering Function, which is an implicit evaluation of the probability that a node is on a path satisfying the bounded cost constraint, and iteratively expands the nodes in the graph with the highest Potential Ordering Function value. The wavefront expansion terminates when the goal node has been expanded, and the path is found by backtracking of the wavefronts. The APTS method runs the PTS algorithm iteratively to improve on the incumbent solution, with the upper bound on total cost lowered in each iteration of the algorithm. Later work (Thayer et al., 2012) improves on the PTS method by minimizing both the potential and an estimation of the remaining search effort, so that the bounded cost search problem will be solved faster.

In this paper, our first objective is to develop a data-driven computational flow model that approximates the true flow field in the region of interest to assist AUV path planning. The proposed data-driven flow model divides the flow field into cells, within which the flow is represented as a single flow vector. The optimal flow cell partition and initial values of the flow vectors in each cell are derived from prior flow information, from numerical ocean models or from observations. To improve model accuracy, AUV observational data can be incorporated into the data-driven model in near-real time, for example, in the form of observed or estimated velocities (Chang et al., 2015). Here, we design a learning algorithm that estimates the flow field parameters based on the AUV path data. Our second objective is to develop an algorithm that solves the AUV bounded cost path planning problem. Given that the vehicle is traveling in a flow field represented by the data-driven computational model, the goal is to design a path that connects AUV initial position with goal position with the highest probability to have travel cost less than a pre-assigned upper bound. By introducing the key function, which is an implicit evaluation function of the probability that a path satisfies the bounded cost constraint, the optimal

path is computed by searching for the nodes with lowest key function value using an informed graph search method.

The main novelty of this work is introducing the modified PTS method to solve the bounded cost search problem. Unlike the PTS method (Stern et al., 2014), which assumes that the branch cost of the graph is known exactly, our method deals with problems where the branch cost of the graph is uncertain. Given assumptions on the distribution of cost-of-arrival and cost-to-go, we prove that the proposed algorithm guarantees optimality of the planned path, that is, the planned path has the highest probability of satisfying the bounded cost constraint. To the best of our knowledge, this is the first time that the optimality of the modified PTS solution to bounded cost problems is proved. The proposed bounded cost path planning method can be viewed as an extension to MEJ. Compared to MEJ, the modified PTS algorithm is computationally more efficient, since a graph search method is adopted to search for the junction positions. At the same time, optimality of the planned path can be theoretically justified. The major benefit of the proposed bounded cost path planning algorithm lies in that it plans a path faster, while at the same time still guarantees the path quality. This paper is a significant extension of the conference proceeding (Hou et al., 2019), which proposes a flow partition method that approximates the flow field by a set of cells of uniform flow speed. The main extensions of this paper are that taking advantage of the flow model proposed in (Hou et al., 2019), we propose the modified PTS method to solve the AUV bounded cost path planning problem, and present theoretical justification on the optimality of the proposed PTS method. The proposed bounded cost search method is potentially applicable for all bounded cost path planning problems with uncertain branch cost.

We believe the proposed data-driven flow modeling and bounded cost path planning methods are well-suited for path planning of underwater glider deployment near Cape Hatteras, NC, a highly dynamic region characterized by confluent western boundary currents and convergence in the adjacent shelf and slope waters. While deployed, the gliders are subject to rich and complex current fields driven by a combination and interaction of Gulf Stream, wind, and buoyancy forcing, with significant cross-shelf exchange on small spatial and temporal scales (Savidge et al., 2013a, Savidge et al., 2013b) that would be highly difficult to sample using traditional methods. Path planning must consider spatial variability of the flow field. Because spatial gradients are significant, real-time path planning is critical to take advantage of real-time data streams. Through simulated experiments, we demonstrate the performance of applying the proposed algorithms to underwater glider deployment in this area, and show that the proposed algorithm is more computationally efficient than A* and LSM.

2 PROBLEM FORMULATION

2.1 Vehicle Dynamics

Let $F_R : \mathcal{D} \to \mathbb{R}^2$ represent a spatially distributed vector field for the ambient flow velocity, where $\mathcal{D} \in \mathbb{R}^2$ is the domain of interest. Let $[T_0, T_f]$ be the AUV deployment time interval. The AUV model is described as

$$\dot{x} = F_R(x) + V_R \Psi_C(t), \tag{1}$$

where $x \in \mathcal{D}$ denotes vehicle position. V_R is the through-water speed of the vehicle, and $\Psi_C(t) = [\cos\psi_C, \sin\psi_C]^T$ is a unit vector that represents the direction of the vehicle motion along heading angle ψ_C.

Assumption 2.1: During the operation, V_R is an unknown constant.

Remark 2.1: Actual vehicle speed may depend on a number of factors that affect an AUV's speed, including water depth, efficiency of propulsion, and bio-fouling. These effects are difficult to estimate. Hence the vehicle forward speed is assumed to be an unknown constant.

Assumption 2.2: We assume that the heading $\Psi_C(t)$ can be controlled for all time t, and the vehicle trajectory $x(t)$ can be measured or estimated for all time.

Remark 2.2: Though the actual location of a vehicle may only be known occasionally when the vehicle is underwater, the trajectory of the vehicle can be estimated through localization algorithms, which incorporates the known locations and the heading angle commands as inputs to generate the optimal state estimation.

Assumption 2.3: We assume the flow field is time-invariant throughout the deployment.

Remark 2.3: Even though there are existing work that considers the time-variant flow field in solving the AUV planning problem, such as (Eichhorn, 2013; Lolla et al., 2014; Zamuda and Sosa, 2014), we make this assumption due to the patterns of the flow field in this domain. In the domain of interest considered in this paper, which is near Cape Hatteras, NC, the current field is driven by a combination and interaction of Gulf Stream, wind, and buoyancy forcing (Savidge et al., 2013a, Savidge et al., 2013b). Because magnitude and spatial gradients of the flow field are significant relative to the temporal variation of the flow field (mostly the tidal flow component), time variation of the flow does not have a significant influence over the planned path.

2.2 Data-Driven Flow Modeling

Flow speed at neighboring grid points often exhibits similarity in both strength and direction. Hence we assume that at the time scale of an AUV deployment, the flow field can be divided into finite number of regions $\{\mathcal{R}_i\}_{i\in I_R}$, with the union of all cells being the domain, $\bigcup_{i\in I_R}\{\mathcal{R}_i\} = \mathcal{D}$. The regions are separated by continuous boundary curves. boundary curves $\{f_{i,j}\}_{i,j\in I_R}$, and $f_{ij}(x) = 0$ is the one dimensional compact boundary of the region $\mathcal{R}_i$ and $\mathcal{R}_j$. We define an indicator function $\phi^i(x)$ as follows:

$$\phi^i(x) = \mathbb{1}\{x \in \mathcal{R}_i\} = \begin{cases} 1 & \text{if } x \in \mathcal{R}_i \\ 0 & \text{otherwise.} \end{cases}$$

This function indicates whether x is in $\mathcal{R}_\alpha$. Let $\phi : \mathbb{R}^2 \to \mathbb{R}^N$ be defined as $\phi(x) = \begin{bmatrix} \phi^1(x) & \dots & \phi^N(x) \end{bmatrix}^T$. Then $\phi(x)$ are a set of spatial basis functions of $\mathcal{D}$.

In order to compute the partition, which is represented by the basis functions ϕ. We need to use prior information of the

environment obtained either from forecast data of the existing ocean models, or from historical datasets. Let $F_0 : \mathcal{D} \times [T_0, T_f] \to \mathbb{R}^2$ denote the discretized flow map forecast available on a set of grid points in $\mathcal{D}$, and let $y \in \mathbb{R}^4, y = [x^T, F_0(x,t)^T]^T$ denote the vector at position x and time t. We can define a distance function as $\text{dist} : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}$ as $\text{dist}^2(y, y') = (y - y')^T Q (y - y')$, where $y, y' \in \mathbb{R}^4$, Q is a weight matrix. For each cell $\mathcal{R}_i$, let $\bar{r}_i$ represents its center, and let ν_i be the flow vector in this cell. Our goal is to find the optimal values of $\phi(x)$ and $\{\nu_i\}_{i \in I_R}$ by solving the following optimization problem:

$$\min_{\phi(x), \{\nu_i\}_{i \in I_R}} J = \sum_{i \in I_R} \sum_{x \in \mathcal{R}_i} \sum_{t \in [T_0, T_f]} \text{dist}^2\left(y(x,t), [\bar{r}_i^T, \nu_i^T]^T\right). \tag{2}$$

After the optimal partition is computed from forecast or historical datasets, we need to compute the strength of the flow in each partition based on the path information of the AUV while moving through the flow field. Again, we assume that the true flow field is constant in each of the partitioned cell. Let $\theta \in \mathbb{R}^{2N}$ be the true flow vectors in all of the partitioned cells, $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \theta_1^1 & \dots & \theta_1^N \\ \theta_2^1 & \dots & \theta_2^N \end{bmatrix}$, with $\theta^i = [\theta_1^i, \theta_2^i]^T$ denoting the flow vector in partitioned region $\mathcal{R}_\alpha$. Then the partitioned flow field can be represented as

$$F_R(x) = \theta \phi(x). \tag{3}$$

To estimate the true flow field F_R, we use the AUV path data $x(t)$. Let

$$\xi(t) = \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix} = \begin{bmatrix} \xi_1^1(t) & \dots & \xi_1^N(t) \\ \xi_2^1(t) & \dots & \xi_2^N(t) \end{bmatrix}$$

be our estimate of the parameter θ and $V_L(t)$ be our estimate for V_R. We will design a learning algorithm to achieve c convergence of $\boldsymbol{\xi}(t)$ and $V_L(t)$ to the true values e.g., $\xi(t) \to \theta$ and $V_L(t) \to V_R$ as $t \to \infty$.

2.3 Bounded Cost Path Planning

Our goal is to find a path connecting the vehicle current position x_0 to the final position x_f that results in total travel time less than an upper bound C. In practice, the planning and replanning process of AUV happen over long intervals, in order to avoid increased computation cost. Hence we assume that the estimated parameters have converged to their true value counterparts before the planning process. There may be more than one path satisfying the bounded cost constraint. Thus we formulate the following optimization problem, in which the decision variable is the vehicle's heading angle, $\psi_C(t)$. The optimization problem is to find the decision variable that is most likely to satisfy the bounded cost constraint:

$$\begin{aligned} &\max_{\psi_C(t) \in [-\pi, \pi]} \Pr\left(T(\psi_C(t)) < C\right) \\ &s.t. \quad \dot{x} = V_R \Psi_C(t) + \theta \phi(x), \\ &x(T_0) = x_0, \\ &x\left(T_0 + T(\psi_C(t))\right) = x_f. \end{aligned} \tag{4}$$

where the total travel time to start from the initial position x_0 to reach the destination position x_f under the control signal $\psi_C(t)$ is denoted as $T(\psi_C(t))$.

3 FLOW FIELD ESTIMATION

First let us describe **Algorithm 1**. We derive the spatial basis function and the initialized flow model parameters by solving **eq. 2** using the K-means algorithm. Since this optimization depends on both spatial and temporal variance of the flow field, solving this problem can be computationally expensive. To simplify this problem, instead of optimizing the difference between the time-varying flow forecast and the partitioned flow field, as described in **eq. 2**, we optimize the difference between the time-averaged flow forecast and the partitioned flow field:

$$\min_{\phi(x), \{\nu_i\}_{i \in I_R}} J' = \sum_{i \in I_R} \sum_{x \in \mathcal{R}_i} \text{dist}^2\left(\bar{y}(x), [\bar{r}_i^T, \nu_i^T]^T\right) \tag{5}$$

where $\bar{y}(x) = \left[x^T, \frac{1}{T_f - T_0} \sum_{t \in [T_0, T_f]} F_0(x,t)^T\right]^T$ denote the time averaged flow observation at position x.

To implement the K-means algorithm, we start by randomly selecting k cell centroids, and then use Lloyd iterations to solve the optimization problem. The Lloyd iteration contains two steps, first assign the points that are closest to a centroid to that centroid, and then recompute the cell centroid. These two steps are repeated until cell membership no longer changes. The K-means algorithm requires proper selection of the number of partitioned cells, k, the choice of which affects the path planning performance and flow modeling quality. If the field is divided into too many regions, then it results in a complicated flow structure and potentially increases the computational cost of path planning. On the other hand, dividing the field into too few regions may result in a large error between the true flow field and the modeled flow field, which potentially leads to significant path planning error. Therefore, we introduce an iterative K-means algorithm that can guarantee a bounded flow field partition error, and at the same time utilize the smallest number of partition regions.

Let $\overline{F}_0 : \mathcal{D} \to \mathbb{R}^2$ denote the time-averaged flow field over the time interval $[T_0, T_f]$, and ν_i is the uniform flow velocity in $\mathcal{R}_i$. Define the flow field partition error as:

$$\delta F = \max_{i \in I_R} \max_{x \in \mathcal{R}_i} \left\| \overline{F}_0(x) - \nu_i \right\|. \tag{6}$$

Given an initialized k, we will iteratively perform the K-means algorithm (lines 8, 9), and check if the flow partition error satisfies $\delta F < \varepsilon$, where ϵ is a pre-defined upper bound on the flow partitioning error. If this condition is satisfied, then the current k is designated for partitioning; otherwise, the number of cells is increased by 1, and we recompute the solution to **eq. 5** using K-means method.

Now let us explain **Algorithm 2**. An estimate $z(t)$ for the vehicle trajectory can be computed by integrating

$$\dot{z} = \xi(t) \phi(z) + V_L(t) \Psi_C + \beta(t), \tag{7}$$

where $\beta(t) \in \mathbb{R}^2$ is introduced as a learning injection parameter. The term $e = x - z$ is the controlled Lagrangian Localization error (CLLE, Cho and Zhang, 2016; Cho et al., 2021), which describes how much the actual trajectory deviates from the estimated trajectory. A learning algorithm will then compute $\beta(t), \xi(t), V_L(t)$ so that the CLLE can be reduced.

The CLLE dynamics can be derived from **eqs 7**, **1**.

ALGORITHM 1 | Flow Field Partition Algorithm

Data: flow field observations $\{y(x_i,t)\}, i \in [1,m], t \in [T_0,T_f]$, flow partition error threshold ϵ
Output: Spatial basis function $\phi(x_i)$, piece-wise constant flow speed of cells $\nu_j, j \in [1,k]$
$\overline{F}_0(x_i) = \frac{1}{T_f - T_0} \sum_{t \in [T_0,T_f]} y(x_i,t)$, $\overline{y}_i = [x_i^T, \overline{F}_0(x_i)^T]^T$
Initialize cluster number $k = 1$, $\bar{r}_1 = \frac{1}{m}\sum_{i=1}^{m} x_i$, $\nu_1 = \frac{1}{m}\sum_{i=1}^{m} \overline{F}_0(x_i)$
Compute δF using **eq. 6**
while $\delta F > \varepsilon$ **do**
  $k = k + 1$
  Randomly initialize cluster centroids
  while not converges **do**
    For all $i \in [1,m]$, set $c_i = \arg\min \text{dist}^2(\overline{y}_i, [\bar{r}_j^T, \nu_j^T]^T)$
    For all $j \in [1,k]$, set $\bar{r}_j = \frac{\sum_i 1(c_i=j) x_i}{\sum_i 1(c_i=j)}$, $\nu_j = \frac{\sum_i 1(c_i=j) \overline{F}_0(x_i)}{\sum_i 1(c_i=j)}$
    Compute δF using **eq. 6**
  end
end
For all $i \in [1,m]$, $\phi(x_i) = [\, 1\{c_i = 1\} \quad \dots \quad 1\{c_i = k\}\,]^T$
For all $j \in [1,k]$, $\mathcal{R}_j = \cup_i 1\{c_i = j\} x_i$

ALGORITHM 2 | Flow Estimation Algorithm

Data: Vehicle trajectory $x(t)$, estimated vehicle trajectory $z(t)$, heading angle $\Psi_C(t)$, initial estimated parameter $\overline{\xi}(0)$, initial estimated speed $V_L(0)$
Output: Estimated parameter $\overline{\xi}(t+1)$, estimated flow parameter $\xi(t+1)$, estimated vehicle speed $V_L(t+1)$
while $t <$ ending time **do**
  $e(t) = x(t) - z(t)$
  Update $\beta(t)$ using **eq. 9**
  Update $\overline{\xi}$ and $V_L(t)$ using **eq. 11**
  Update $z(t)$ by integrating **eq. 7**
end

$$\dot{e} = \dot{x} - \dot{z} = \theta\phi(x) - \xi(t)\phi(z) + (V_R - V_L(t))\Psi_c - \beta(t). \tag{8}$$

We design the learning parameter injection as

$$\beta(t) = \xi(t)\phi(x) - \xi(t)\phi(z) + \text{Ke}, \tag{9}$$

and hence the CLLE dynamics becomes

$$\dot{e} = -\text{Ke} + (\theta - \xi(t))\phi(x) + (V_R - V_L(t))\Psi_c. \tag{10}$$

The learning algorithm updates parameters $\xi(t)$ and $V_L(t)$ so that the CLLE converges to zero. Let $\overline{\xi}(t) = [\xi_1^1(t), \dots, \xi_1^N(t), \xi_2^1(t), \dots, \xi_2^N(t)]^T$, $\overline{\theta} = [\theta_1^1, \dots, \theta_1^N, \theta_2^1, \dots, \theta_2^N]^T$, and $e \otimes \phi = [e_1\phi^1, \dots, e_1\phi^N, e_2\phi^1, \dots, e_2\phi^N]^T$, where $\otimes$ is the Kronecker product. We design the updating rules for parameter estimation as follows,

$$\begin{aligned} \dot{\overline{\xi}}(t) &= \rho e \otimes \phi(x) \\ \dot{V}_L(t) &= \rho e^T \Psi_c. \end{aligned} \tag{11}$$

These rules are then used in **Algorithm 2**.

4 BOUNDED COST PATH PLANNING

Given the piece-wise constant flow model described in **eq. 3**, the domain is divided into a finite number of regions $\{\mathcal{R}_i\}_{i \in I_R}$. Thus, all possible trajectories cross a sequence of cells of uniform flow, and finally reach the goal position. Since the vehicle moves in constant speed, and the flow in one cell is uniform and constant, the vehicle's optimal heading angle in each cell should be constant, and the vehicle path in each cell is a straight line. We define junction points as the position where the path intersects with cell boundaries. Below we show that in each cell, due to the time invariance of the flow field, solving for the heading angle is equivalent to solving for the junction points of a path.

Let γ_1, γ_2 denote two junction points on two different boundary curves of the same cell $\mathcal{R}_i$. Since the vehicle moves at constant speed, the total vehicle speed $V_R\Psi_C + \theta^i$ must be in the same direction as the segment of the path,

$$\frac{\gamma_2 - \gamma_1}{\|\gamma_2 - \gamma_1\|} = \frac{V_R\Psi_C + \theta^i}{\|V_R\Psi_C + \theta^i\|}. \tag{12}$$

From **eq. 12**, we can represent the vehicle's heading angle as a function of the junction points,

$$\Psi_C = \frac{1}{V_R}\left(\frac{\gamma_2 - \gamma_1}{\|\gamma_2 - \gamma_1\|}\sqrt{V_R^2 + 2V_R(\theta^i)^T\Psi_C + \|\theta^i\|^2} - \theta^i\right). \tag{13}$$

The vehicle's travel time in $\mathcal{R}_i$, denoted as τ, can be computed given the junction points and the vehicle's heading angle,

$$\tau(\gamma_1, \gamma_2, \theta^i) = \frac{\|\gamma_2 - \gamma_1\|}{\|V_R\Psi_C + \theta^i\|}. \tag{14}$$

Combining **eq. 14** and **eq. 12** we can write the travel time in cell $\mathcal{R}_i$ as a function dependent only on the junction points,

$$\begin{aligned} \tau(\gamma_1, \gamma_2, \theta^i) = \frac{1}{\|\theta^i\|^2 - V_R^2}\Bigg(&(\gamma_2 - \gamma_1)^T\theta^i \\ &- \sqrt{\left((\gamma_2 - \gamma_1)^T\theta^i\right)^2 + \|\gamma_2 - \gamma_1\|^2\left(V_R^2 - \|\theta^i\|^2\right)}\Bigg). \end{aligned} \tag{15}$$

eq. 15 describes the travel time in one single cell. Based on the discussion of the one cell case, next we talk about solving **eq. 4** across multiple cells.

Let $\gamma_1, \gamma_2, \dots, \gamma_n$ denote the chain of junctions position, and $p_1, p_2, \dots, p_{n+1}$ represent the index of the sequence of cells that the path crosses. The planning problem **eq. 4** can be transformed into the following mixed integer optimization problem, where the decision variables are the cell sequence and the junctions position,

$$\begin{aligned} &\max_{\{\gamma_i\}_{i=1}^n, \{p_i\}_{i=1}^{n+1}} \Pr\left(\tau(x_s, \gamma_1, \theta^{p_1}) + \sum_{i=1}^{n-1}\tau(\gamma_i, \gamma_{i+1}, \theta^{p_i}) + \tau(\gamma_n, x_f, \theta^{p_{n+1}}) < C\right), \\ &s.t. \quad f_{p_i,p_{i+1}}(\gamma_i) = 0, \forall i \in [1,n]. \end{aligned} \tag{16}$$

Now let us explain the proposed solution to **eq. 16**. The solution is presented in **Algorithm 3**. We propose a bounded cost path planning algorithm that solves for the path that is most likely to satisfy the bounded cost constraint. The solution contains two steps, the first step is solving for the optimal sequence of cells that is most likely to result in a bounded cost path, in the discretized flow map described by the piece-wise constant flow cells. In this step, the junction positions are unknown. An informed graph

ALGORITHM 3 | Bounded Cost Search in Piece-wise Constant Flow Field

Data: Start and goal node s, g, start and goal position x_0, x_f, travel cost upper bound C, graph $\mathcal{G}$
Output: Optimal heading angle Ψ_C
PTS $(s, g, C, \mathcal{G}) \rightarrow \mathcal{P}$ *Step 1: Find the optimal cell sequence
MEJ $(\mathcal{P}, x_0, x_f) \rightarrow \Psi_C$ *Step 2: Find the optimal heading angle
Function PTS a$(s, g, C, \mathcal{G})$
  Initialization. $g(n) = \infty, h(n) = \infty, K(n) = \infty, \forall n \in \mathcal{G}$
  $s \rightarrow$ {CLOSED}
  Set the heuristics and estimated cost-of-arrival of s
  (OPEN, CLOSED) = Expand $(s,$ OPEN, CLOSED, $\mathcal{G})$
  while OPEN is not empty **do**
    if $g \in$ CLOSED **then**
      Backtracking $(s, g) \rightarrow \alpha$
    end
    $v = \arg\min_{n \in \text{OPEN}} K(n)$
    (OPEN, CLOSED) = Expand (v, OPEN, CLOSED)
  end
  return α
Function Epand v, OPEN, CLOSED, $\mathcal{G}$
  {OPEN}\v $\rightarrow$ {OPEN}
  $v \cup$ {CLOSED} $\rightarrow$ {CLOSED}
  Find adjacent nodes $\{n_i\}_{i=1}^{m}$ to v in $\mathcal{G}$
  for i = 1 to m **do**
    Compute minimum branch cost $w^*(v, n_i)$ by solving **eq. 17**
    if $w^*(v, n_i) + g(v) < g(n_i)$ n_i **then**
      predecessor = v
      {OPEN} $\cup\, n_i \rightarrow$ {OPEN}
      Compute $h(n_i)$ using **eq. 18**
      $g(n_i) = g(v) + w^*(v, n_i)$
      Update $K(n_i)$ using **eq. 19**
    end
  end
  return OPEN, CLOSED
Function Backtrack (s, g)
  $g \rightarrow \mathcal{P}$
  while $s \notin \{\mathcal{P}\}$ **do**
    $v = \mathcal{P}$(end)
    v.predecessor $\cup\, \mathcal{P} \rightarrow \mathcal{P}$
  end
  return $\mathcal{P}$
Function MEJ $(\mathcal{P}, x_0, x_f)$
  Initialize junction set γ^0 on the boundary curves of the cell sequence $\mathcal{P}$
  Compute junction positions by solving **eq. 20**
  Compute heading angle from junction positions by **eq. 21**
  return $\Psi_C(t)$

search method is used for the first step of the proposed solution. The second step is optimizing junction positions on the boundaries of the optimal cell sequence.

The proposed solution is presented in **Algorithm 3**. First we describe the first step of our solution. Consider having a candidate junction on boundaries of all cells. Two junctions γ_i and γ_j are defined as adjacent if $f_{p,q}(\gamma_i) = 0, f_{r,s}(\gamma_j) = 0, \{p, q\} \cap \{r, s\} \neq \varnothing$, indicating that the two junctions are on different boundaries of the same cell. Two adjacent junctions are connected by an edge. A non-directed graph $\mathcal{G}$ can be formed with the vertices being all the candidate junctions in the domain, and the edges being the path segment between the adjacent candidate junctions. Let $n_{i,j}$ describe the node that corresponds to the junction on boundary curve between $\mathcal{R}_i$ and $\mathcal{R}_j$, and let s and g denote the node corresponding to the starting and final position. Then in this context, a path Γ from the starting position x_0 to the final position x_f, crossing the cells $\mathcal{R}_{p_1}, \ldots, \mathcal{R}_{p_{n+1}}$ in sequence can be represented by a sequence of nodes $s \rightarrow n_{p_1 p_2} \rightarrow n_{p_2 p_3} \rightarrow \ldots \rightarrow n_{p_n p_{n+1}} \rightarrow g$ on the graph. **Figure 1** is an example of the graph representation of the workspace, in which case the flow field is partitioned into 4 cells.

The branch cost of the graph is defined as the travel time from one junction to another adjacent junction. The travel time can be computed by **eq. 15** if the two junction positions are known. However, since the junction positions are unknown when optimizing the cell sequence, the branch cost of the graph cannot be explicitly computed. Hence we introduced the following assumption:

Assumption 4.1: We assume that the branch cost for all edges in $\mathcal{E}$ is a random variable, with a known minimum value.

Remark 4.1: Even though the branch cost is unknown, its minimum value can be computed, since the branch cost **eq. 15** is convex with respect to $\gamma_2 - \gamma_1$ (Soulignac, 2011).

We solve for the minimum cost of all edges in the graph, denoted as $w^*_{ij,jk}$ by solving the following constrained optimization problem, using the interior-point method. (Kim et al., 2007),

$$\begin{aligned} &\min_{\gamma_1, \gamma_2} \tau\left(\gamma_1, \gamma_2, \theta^j\right) \\ &s.t. \quad f_{ij}(\gamma_1) = 0, f_{jk}(\gamma_2) = 0. \end{aligned} \tag{17}$$

The informed graph search method we propose is an extension to a class of graph search algorithms called potential search (PTS) (Stern et al., 2014). The PTS algorithms can be viewed as modifications to the celebrated A* algorithm for path planning (Hart et al., 1968). To determine which nodes should be searched, the algorithms maintain an OPEN list and a CLOSED list. A graph node is labeled NEW if it has not been searched by the algorithm. The OPEN list contains all the nodes that are searched, but still have a NEW neighbor. The CLOSED list consists of all the nodes that have been accessed by the search algorithm.

To determine which cells should be searched first by the algorithm, the algorithm computes the cost-of-arrival, which is the minimal cost of going from the starting node s to an arbitrary node n, and cost-to-go, which is the minimal cost of going from n to the goal point g. Let $g^*(n)$ denote the actual cost-of-arrival, and let $h^*(n)$ denote the actual cost-to-go of a node n. Since the actual cost-to-go is unknown during the search, a heuristic cost $h(n) \leq h^*(n)$, is usually used by the search algorithm. The A* search algorithm sort the OPEN list according to the value of $g^*(n) + h(n)$. The node with lowest value is searched first. In our problem, the following estimated cost-to-go is used to guide the search:

$$h(n) = \frac{\min_{\gamma_n} \left\| x_f - \gamma_n \right\|}{V_R + \max_{i \in I_R} \left\| \theta^i \right\|}. \tag{18}$$

The heuristic function defined in **eq. 18** is the travel time of the vehicle traveling in the most favorable flow condition, reaching goal position from the junction position that is closest to the goal. Hence, $h(n) \leq h^*(n)$, which is required by A* search. However, in our problem, the branch cost is unknown. The exact value of

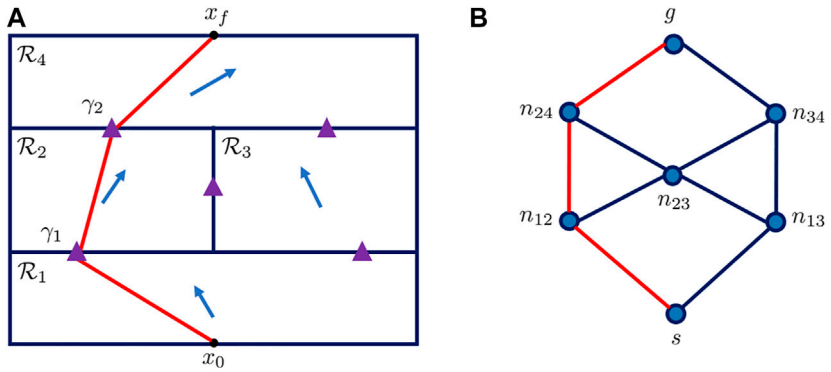

FIGURE 1 | (A) Partitioned cells in the domain. On each boundary of two adjacent cells there is a candidate junction point, represented as the purple triangle **(B)** Graph representation of the workspace. The vertices represent the candidate junctions, while the edges are the path segment between the adjacent junctions. The red line on both of the plots represent the same example path.

actual cost-of-arrival cannot be computed during the search process, which is different from a typical path planning problem that can be solved by A* or PTS method. Hence we introduce the estimated cost-of-arrival, denoted as $g(n)$. The estimated cost-of-arrival is computed by summation of the minimum branch cost along the path, thus $g(n) \leq g^*(n)$.

The goal of A* search is to find the path with minimum cost. In our problem, due to the uncertainties in the branch cost, this goal is overly ambitious. Hence our problem formulation **eq. 16** aims to find a path with bounded cost. We define a potential function described as follows:

Definition 4.1: The potential of a node n, denoted as $\mathrm{PT}(n)$, is $\Pr(h^*(n) + g^*(n) < C)$.

The potential function characterizes the probability that a node is on a path that satisfies the bounded cost constraint. Nodes with high potential have higher probability to be part of the desired path. However, the exact potential of nodes cannot be computed or compared, since both $h^*(n)$ is unknown before the optimal path is found. Therefore, PTS algorithms usually design a key function to determine the nodes that need to be searched at each step of the graph search. Nodes in the OPEN list are sorted by the key function value instead of $g^*(n) + h(n)$, which is the main difference between the PTS algorithms and the A* method. Various key functions have been proposed for different path planning problems with bounded cost (Thayer et al., 2012; Stern et al., 2014, Stern et al., 2011).

One significant contribution of this paper is in extending the PTS method to solving bounded cost problems with uncertain branch cost, by introducing a new form of key function $K(n) \in \mathbb{R}_{\geq 0}$ as

$$K(n) = \begin{cases} \dfrac{h(n)g(n)}{\left(C - h(n) - g(n)\right)^2}, & \text{if } h(n) + g(n) < C \\ \infty, & \text{otherwise} \end{cases}. \tag{19}$$

$K(n)$ is an indication of the probability of the node n being on a path that satisfies the bounded cost constraint. Nodes with lower key function value have a higher probability of being on a path satisfying the bounded cost constraint. The intuition is that, if $h(n) + g(n) < C$, the key function value increases if either $h(n)$ or $g(n)$ is larger. In this case, the estimated cost $h(n) + g(n)$ increases, and will be closer to C, then it is less likely that the true cost satisfies the bounded cost constraint, and the node n is less likely to be on a feasible path. If $h(n) + g(n) \geq C$, then n cannot be on a path satisfying the bounded cost constraint, since $h^*(n) + g^*(n) \geq h(n) + g(n) \geq C$. In this case, the key function is set as positive infinity.

The PTS with our new key function is then applied to search for the optimal cell in **Algorithm 3**. The only difference between our PTS and A* is that the total cost used by A* to sort the OPEN list is replaced by the key function, as shown in line 13. Similar to A*, The search algorithm consists of two processes: the expansion process and the backtracking process. During the iterative expansion process, the algorithm orders the nodes in the OPEN set according to the key function value, and inserts the node with the lowest key function value to the CLOSED set (lines 19, 20). Neighbors of this node and their key function values are updated if the neighboring nodes can be reached with a lower cost through the current node (lines 26, 28, 29). The propagation continues until the OPEN list is depleted, or the goal node is in the OPEN set. Starting from the goal position, the backtracking process searches for the predecessor of the last node in the path set and add it to the path, until the starting node is included in the path (lines 37, 38).

The PTS algorithm fulfills step one of the bounded cost path planning solution. We have found the vector of indices $\{p_i\}_{i=1}^{n+1}$, which indicates the optimal indices that is most likely to result in a bounded cost path. In step two, we find the optimal junction positions that leads to the minimum total cost. Given the optimal cell sequence, the problem **eq. 16** converts to an optimization problem depending on the junction positions $\{\gamma_i\}_{i=1}^{n}$ in all cells,

$$\begin{aligned} &\min_{\{\gamma_i\}_{i=1}^{n}} \tau(x_0, \gamma_1, \theta^{p_1}) + \sum_{i=1}^{n} \tau(\gamma_i, \gamma_{i+1}, \theta^{p_i}) + \tau\left(\gamma_n, x_f, \theta^{p_{n+1}}\right) \\ &s.t. \quad f_{p_i, p_{i+1}}(\gamma_i) = 0. \end{aligned} \tag{20}$$

This optimization problem is solved by the interior-point method. The optimal heading angle can be computed from the junction

positions using **eq. 12**. In each cell of the sequence $\{p_i\}_{i=1}^{n}$, given the optimal junction position γ_{i+1} and γ_i, the heading angle in cell $\mathcal{R}_{p_i}$ can be derived by

$$\Psi_C = \frac{1}{V_R}\left(\frac{\gamma_{i+1}-\gamma_i}{\left\|\gamma_{i+1}-\gamma_i\right\|}\sqrt{V_R^2 + 2V_R\left(\theta^{p_i}\right)^T\Psi_C + \left\|\theta^{p_i}\right\|^2} - \theta^{p_i}\right). \quad (21)$$

5 THEORETICAL JUSTIFICATION

In this section, we give theoretical justification to the proposed data-driven flow modeling and bounded cost path planning method.

5.1 Data-Driven Flow Modeling

Algorithm 1 can be theoretically justified by proving that the optimal solution to **eq. 5** is the optimal solution to **eq. 2**. We also prove that **Algorithm 2** achieves error convergence and parameter convergence, indicating that the estimated trajectory converges to the actual trajectory, and the estimated parameter converges to the true values.

Lemma 5.1: The optimal flow partition derived by solving **eq. 2** is the same as the optimal flow partition derived from **eq. 5**.

PROOF: Let $\delta y(x,t) = y(x,t) - \overline{y}(x)$. Since $\sum_{t=T_0}^{T_f}\delta y(x,t) = \sum_{t=T_0}^{T_f} y(x,t) - (T_f - T_0)\overline{y}(x) = 0$, the following equality holds

$$\begin{aligned} J &= \sum_{i=1}^{N}\sum_{x\in\mathcal{R}_i}\sum_{t=T_0}^{T_f}\mathrm{dist}^2\left(y(x,t),\mu_i\right) \\ &= \sum_{i=1}^{N}\sum_{x\in\mathcal{R}_i}\sum_{t=T_0}^{T_f}\mathrm{dist}^2\left(\overline{y}(x)+\delta y(x,t),\mu_i\right) \\ &= \sum_{i=1}^{N}\sum_{x\in\mathcal{R}_i}\left[\left(T_f - T_0\right)\mathrm{dist}^2\left(\overline{y}(x),\mu_i\right) + \sum_{t=T_0}^{T_f}\mathrm{dist}^2\left(y(x,t),\overline{y}(x)\right)\right]. \end{aligned}$$

The second term in J represents the temporal variation of flow speed on one grid point, which does not change with respect to the partitioning of the flow field. Hence $\arg\min_{\phi(x),v} J = \arg\min_{\phi(x),v} J'$, where J' is defined in **eq. 5**. Thus the optimal solution of **eq. 2** equals the optimal solution of **eq. 5**, which implies that the optimal flow partition of the time-varying flow field is equivalent to the optimal flow partition of the time-invariant flow field, computed by taking the time-average of the flow field observations, as described in line 1, **Algorithm 1**.

Next we will prove that under Assumption 2.3, the estimated trajectory converges to the actual trajectory, and that the estimated parameter converges to the true value using adaptive control theory. In order to prove convergence, the persistent excitation condition must be demonstrated, and is given below.

Definition 5.2: (Sastry and Bodson, 1994; Khalil, 1996) A vector signal u is persistently exciting if there exist positive constants κ_1, κ_2, and T such that $\kappa_2 I \geq \int_t^{t+T} u(\tau)u^T(\tau)d\tau \geq \kappa_1 I\ \forall t$

Let $\tilde{\phi}_1 = \begin{bmatrix} \phi^1 & \cdots & \phi^N \\ 0 & 0 & 0 \end{bmatrix}$ and $\tilde{\phi}_2 = \begin{bmatrix} 0 & 0 & 0 \\ \phi_1 & \cdots & \phi^N \end{bmatrix}$. Let $w = [\tilde{\phi}_1, \tilde{\phi}_2, \Psi_C]^T \in \mathbb{R}^{(2N+1)\times 2}$, which is the input signal to **eq. 24**. We can construct a matrix $W(t) \in \mathbb{R}^{(2N+1)\times(2N+1)}$ as follows

$$W(t) = \int_t^{t+T}\begin{bmatrix} \phi^1\phi^1 & \phi^1\phi^2 & \cdots & \phi^1\phi^N & 0 & \cdots & 0 & \phi^1\cos\psi_c \\ \phi^2\phi^1 & \phi^2\phi^2 & \cdots & \phi^2\phi^N & 0 & \cdots & 0 & \phi^2\cos\psi_c \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi^N\phi^1 & \phi^N\phi^2 & \cdots & \phi^N\phi^N & 0 & \cdots & 0 & \phi^N\cos\psi_c \\ 0 & 0 & \cdots & 0 & \phi^1\phi^1 & \cdots & \phi^1\phi^N & \phi^1\sin\psi_c \\ 0 & 0 & \cdots & 0 & \phi^2\phi^1 & \cdots & \phi^2\phi^N & \phi^2\sin\psi_c \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \phi^N\phi^1 & \cdots & \phi^N\phi^N & \phi^N\sin\psi_c \\ \phi^1\cos\psi_c & \phi^2\cos\psi_c & \cdots & \phi^N\cos\psi_c & \phi^1\sin\psi_c & \cdots & \phi^N\sin\psi_c & 1 \end{bmatrix} d\tau. \quad (22)$$

The persistent excitation condition is critical to prove the convergence of parameters (Narendra and Annaswamy, 1989). When $W(t)$ is singular the estimation errors of parameters may not converge to zero. The persistent excitation condition requires that the trajectories traveled by the robot to spread over all the partitioned cells, as stated in the following Lemma.

Lemma 5.3: The signal vector w is persistently exciting if the vehicle visits all the partitioned cells.

PROOF: Since the partitioned cells do not overlap with each other, for $\forall\tau$, $x(\tau)$ can only be in one cell. Hence for all $i, j \in I_R$,

$$\phi^i(x(\tau))\phi^j(x(\tau)) = \mathbb{1}\{i=j\} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise} \end{cases}.$$

Thus $W(t)$ can be simplified to

$$W(t) = \int_t^{t+T}\begin{bmatrix} \phi^1\phi^1 & & & & & & & \phi^1\cos\psi_c \\ & \ddots & & & & 0 & & \vdots \\ & & \phi^N\phi^N & & & & & \phi^N\cos\psi_c \\ & & & \phi^1\phi^1 & & & & \phi^1\sin\psi_c \\ & 0 & & & & \ddots & & \vdots \\ & & & & & & \phi^N\phi^N & \phi^N\sin\psi_c \\ \phi^1\cos\psi_c & \cdots & \phi^N\cos\psi_c & \phi^1\sin\psi_c & \cdots & & \phi^N\sin\psi_c & 1 \end{bmatrix} d\tau.$$

If $\forall i, \exists\tau \in [t, t+T], s.t.\ \phi^i(x(\tau)) = 1$, meaning that the vehicle visits all cell during time $[t, t+T]$, then $W(t)$ is full rank, and hence w is persistently exciting.

The persistent excitation condition must be satisfied in order to have the flow parameters of all the cells and vehicle speed estimation converge to the true value. The persistent excitation condition requires the vehicle to visit all the partitioned regions. If this condition is not satisfied, not all flow parameters in the partitioned cells can be accurately estimated. We will further address this condition in the simulation and experimental result section.The convergence of CLLE is presented as follows.

Theorem 5.4: Under the updating law **eq. 11**, CLLE converges to zero when time goes to infinity.

PROOF: Consider the following Lyapunov function,

$$V\left(e,\overline{\xi},V_L\right) = \frac{1}{2}\left(e^Te + \frac{1}{\rho}\left(\overline{\theta}-\overline{\xi}(t)\right)^T\left(\overline{\theta}-\overline{\xi}(t)\right) + \frac{1}{\rho}(V_R - V_L(t))^2\right).$$

Since $e^T(\theta-\xi(t))\phi(x) = (\overline{\theta}-\overline{\xi}(t))e\otimes\phi(x)$, the derivative of V is

$$\begin{aligned} \dot{V} &= \left(-\mathrm{Ke} + (\theta-\xi)\phi(x) + (V_R - V_L(t))\Psi_C\right)^Te + \frac{1}{\rho}\left(-\rho e\otimes\phi(x)\right)\left(\overline{\theta}-\overline{\xi}\right) \\ &\quad + \frac{1}{\rho}\left(V_R - V_L(t)\right)\left(-\rho e^T\Psi_C\right) \\ &= e^T\mathrm{Ke} \leq 0. \end{aligned}$$

$\dot{V}$ is negative semi-definite, which implies that $e, \overline{\xi}, V_L(t)$ are bounded. In addition, the second order time derivative of V is

$$\ddot{V} = -2e^T K\big((\theta - \xi(t))\phi(x) + (V_R - V_L(t))\Psi_C - \mathrm{Ke}\big).$$

Thus $\ddot{V}$ bounded, and hence $\dot{V}$ is uniformly continuous. Therefore $\lim_{t\to\infty}\dot{V}(t) = 0$. Since K is a diagonal matrix, $e(t) \to 0$ as $t \to \infty$.

Theorem 5.5: Under the updating law **eq. 11**, if the vehicle visits all the partitioned cells, $\overline{\xi}(t)$ and $V_L(t)$ converges to $\overline{\theta}$ and V_R respectively as time goes to infinity.

PROOF: Let $\eta_1 = \theta_1 - \xi_1(t), \eta_2 = \theta_2 - \xi_2(t), \eta_3 = V_R - V_L(t)$, then the CLLE dynamics can be written as

$$\dot{e} = \tilde{\phi}_1(x)\eta_1 + \tilde{\phi}_2(x)\eta_2 + \eta_3\Psi_C - \mathrm{Ke}.$$

We define a new state variable $X = [e^T, \eta_1^T, \eta_2^T, \eta_3]^T$, and an output variable $Y = e$, then the dynamics for the state variable and the output variable satisfy

$$\dot{X} = A(t)X, Y = \mathrm{CX},$$

$$\text{where } A(t) = \begin{bmatrix} -K & \tilde{\phi}_1 & \tilde{\phi}_2 & \Psi_C \\ -\rho\tilde{\phi}_1 & 0 & 0 & 0 \\ -\rho\tilde{\phi}_2 & 0 & 0 & 0 \\ -\rho\Psi_C & 0 & 0 & 0 \end{bmatrix}, C = \begin{bmatrix} I & 0 & 0 & 0 \end{bmatrix}.$$

Our goal is to show that the origin of $\dot{X} = A(t)X$ is uniformly asymptotically stable, which indicates that $\overline{\xi}$ converges to $\overline{\theta}$, and $V_L(t)$ converges to V_R. Let

$$P = \begin{bmatrix} \frac{1}{2}K^{-1} & 0 & 0 & 0 \\ 0 & \frac{1}{2\rho}K^{-1} & 0 & 0 \\ 0 & 0 & \frac{1}{2\rho}K^{-1} & 0 \\ 0 & 0 & 0 & \frac{1}{2\rho}K^{-1} \end{bmatrix}.$$

There exists some c_1, c_2 such that $c_1 I \le P \le c_2 I$, and there exists some constant $0 < \nu < 1$ such that

$$A(t)^T P + \mathrm{PA}(t) + \dot{P} + \nu \mathrm{C}^T C = (1-\nu)\begin{bmatrix} -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

which is negative semi-definite.

Then by the Lyapunov theorem (Theorem 3.8.4 in (Ioannou and Sun, 1995)), $\dot{X} = A(t)X$ is uniformly asymptotically stable if we can prove that (C, A) is uniformly completely observable. First, we will find a bounded matrix L, and show that $(C, A + \mathrm{LC})$ is uniformly completely observable. Then, this will lead to the conclusion that (C, A) is uniformly completely observable. Let $L = \begin{bmatrix} 0 \\ \rho\tilde{\phi}_1 \\ \rho\tilde{\phi}_2 \\ \rho\Psi_C^T \end{bmatrix}$. Since Ψ_C is uniformly bounded, and all elements in $\tilde{\phi}$ is either 0 or 1, L is uniformly bounded, and

$$A + \mathrm{LC} = \begin{bmatrix} -K & \tilde{\phi}_1 & \tilde{\phi}_2 & \Psi_C \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Thus, we now consider the observability of

$$\begin{aligned} \dot{X} &= (A + \mathrm{LC})X \\ Y &= \mathrm{CX}. \end{aligned} \tag{23}$$

Let $\eta = [\eta_1, \eta_2, \eta_3]^T$, then the system **eq. 23** has the following form:

$$\begin{aligned} \dot{e} &= -\mathrm{Ke} + w^T\eta \\ \dot{\eta} &= 0 \\ Y &= e \end{aligned} \quad . \tag{24}$$

Due to the assumption that the vehicle visits all cells, by Lemma 5.3, w is persistently exciting. Let $\Phi(\tau) = \int_t^\tau e^{-K(\tau-\sigma)}w(\sigma)d\sigma$ be the output of **eq. 24** given input w. Then $\Phi(\tau)$ satisfies persistently exciting conditions because $w(\sigma)$ is persistently exciting, and the transfer function of **eq. 24**, $(sI + K)^{-1}$ is stable, minimum phase, proper rational transfer function. Therefore, there exists constant $\kappa_1, \kappa_2, T_0 > 0$ such that $\kappa_2 I \ge \frac{1}{T_0}\int_t^{t+T_0}\Phi(\tau)\Phi(\tau)^T d\tau \ge \kappa_1 I, \forall t \ge 0$.

By applying Lemma 4.8.4 in (Ioannou and Sun, 1995), we can conclude that the system of **eq. 24** is uniformly completely observable. In other words, we have proved that $(C, A + \mathrm{LC})$ is uniformly completely observable. By applying Lemma 4.8.1 in (Ioannou and Sun, 1995) the system (C, A) is uniformly completely observable. Therefore, the origin of $\dot{X} = \mathrm{AX}$ is uniformly asymptotically stable, that is, $X \to 0$ as $t \to \infty$. This means that η_1, η_2 and η_3 go to zeros, individually. Thus $\overline{\xi}$ and $V_L(t)$ converges to $\overline{\theta}$ and V_R, respectively.

5.2 Bounded Cost Path Planning

In this subsection, we prove that **Algorithm 3** finds the optimal solution of **eq. 4**. Assumption 5.1 and Assumption 5.2 are required for the optimality proof.

Assumption 5.1: Consider any node not the starting node s or the goal node g, the estimated cost-of-arrival $g(n)$ and the estimated cost-to-go $h(n)$ are bounded below, $g(n) \ge g_{\min}, h(n) \ge h_{\min}$, where $g_{\min} > 0$ and $h_{\min} > 0$.

Remark 5.1: For any node that is not the goal node, $h(n)$ reaches its minimum when n is an adjacent node of g. Similarly, for any node that is not the start node, $g(n)$ reaches its minimum when n is adjacent to s. Since the flow partition algorithm is performed over discrete grid points in $\mathcal{D}$, size of the cells cannot be infinitely small. Therefore, $h(n)$ can only be zero if the junction represented by the node n is sliding on the same boundary of the goal point, and $g(n)$ can only be zero if the junction represented by the node n is sliding on the same boundary of the start point. However, by junction assignment, only one junction can be assigned on each boundary. Hence there exists $h_{\min}$ and $g_{\min}$ that bound $h(n)$ and $g(n)$ from below, and the lower bound cannot be infinitely small.

Let $H_{\max} = \max\left\{\frac{C}{h_{\min}}, \frac{C}{g_{\min}}\right\}$. Consider $\{X_n\}_{n=1}^{N}, \{Y_n\}_{n=1}^{N}$ to denote sequences of independent and identically distributed random variables uniformly distributed over $[1, H_{\max}]$. To prove optimality of the algorithm, we make assumptions on the statistical relationship between $h^*(n), h(n)$, and $g^*(n), g(n)$ as follows.

Assumption 5.2: The true cost-to-go, $h^*(n)$ and the heuristic function $h(n)$, as well as the true cost-of-arrival, $g^*(n)$ and estimated cost-of-arrival, $g(n)$ both satisfy $h^*(n) = h(n)Y_n, g^*(x) = g(n)X_n$.

Remark 5.2: Both $h^*(n)$ and $g^*(n)$ are summation of branch cost along the optimal path. Since the branch cost is the travel time between two adjacent junctions sliding on two boundaries, the branch cost of all edges must have both a lower bound and an upper bound. Hence both $h^*(n)$ and $g^*(n)$ are assumed to be a uniform distribution, with the minimum of it being $h(n)$ and $g(n)$, and the maximum being $h(n)H_{\max}$ and $g(n)H_{\max}$. In practice, the statistical model of $h^*(n)$ and $g^*(n)$ depends on the distribution of the flow field, and may not be uniform distribution in some flow cases. However, the following theoretical analysis can be adapted to other parameterization of the statistical model of $h^*(n)$ and $g^*(n)$.

We will show below that, by expanding the nodes with the lowest key function value without explicitly calculating the potential of nodes, the proposed algorithm expands the nodes with the highest potential value, and thus guarantees to find the optimal solution to **eq. 4**. Lemma 5.6 states that the key function is an equivalent evaluation of the potential value of nodes. Lemma 5.7 demonstrates that the optimal path can be equally defined by either the potential or the key function value of nodes. Finally, given the two Lemmas, we justify the optimality of the proposed algorithm, which is stated in Theorem 5.8.

Lemma 5.6: For all $n_1, n_2 \in \mathcal{G}$, $\mathrm{PT}(n_1) < \mathrm{PT}(n_2)$ if and only if $K(n_1) > K(n_2)$.

PROOF: To simplify notation, let h_1, h_1^*, g_1, g_1^* denote $h(n_1), h^*(n_1), g(n_1)$, and $g^*(n_1)$, respectively. The Lemma trivially holds in the cases where either $K(n_1)$ or $K(n_2)$ is infinity. Below we show that the Lemma holds in the case where both $K(n_1)$ and $K(n_2)$ are not infinity; equivalently, $h_1 + g_1 < C$ and $h_2 + g_2 < C$. Due to the i.i.d. distribution assumption stated in Assumption 5.2, X_1, X_2 can be written as a single random variable uniformly distributed on $[1, H_{\max}]$, and Y_1, Y_2 also can be written as a single random variable uniformly distributed on $[1, H_{\max}]$. Therefore, $\mathrm{PT}(n_1) < \mathrm{PT}(n_2)$ if and only if

$$\Pr(h_1 Y + g_1 X < C) < Pr(h_2 Y + g_2 X < C).$$

Terms in the above inequality can be computed by integration of the probability density function,

$$\iint_{S_1} \rho_{X,Y}(x,y)dxdy < \iint_{S_2} \rho_{X,Y}(x,y)dxdy,$$

where $S_1 = \{(x,y) \big| h_1 y + g_1 x < C, x \in [1, H_{\max}], y \in [1, H_{\max}]\}$, $S_2 = \{(x,y) \big| h_2 y + g_2 x < C, x \in [1, H_{\max}], y \in [1, H_{\max}]\}$. By Assumption 5.2, $X \le \frac{C}{g_{\min}}$, $Y \le \frac{C}{h_{\min}}$, and therefore S_1, S_2 are two triangles, as shown in **Figure 2**. Due to the uniform distribution of X, Y, the above integration can be easily computed by multiplying area of S_1, S_2 with the joint distribution $\rho_{X,Y}(x,y)$, which is a constant. Hence,

$$\frac{1}{2}\rho_{XY}\left(\frac{C-g_1}{h_1} - 1\right)\left(\frac{C-h_1}{g_1} - 1\right) < \frac{1}{2}\rho_{XY}\left(\frac{C-g_2}{h_2} - 1\right)\left(\frac{C-h_2}{g_2} - 1\right)$$

which implies $K(n_1) > K(n_2)$.

Let the ordered sequence Γ denote a path connecting the start node s with the goal node g in the graph $\mathcal{G}$. Define $K_{\max}(\Gamma)$ as the highest key function value of all nodes on the path Γ, that is, $K_{\max}(\Gamma) = \max_{n \in \Gamma} K(n)$.

Lemma 5.7: The optimal path minimizes $K_{\max}(\Gamma)$ over all paths in the graph.

PROOF: Let Γ^* denote the optimal path that maximizes $\Pr(h^*(n) + g^*(n) < C)$. Suppose that there is a path Γ' that is different from the optimal path Γ^* with $K_{\max}(\Gamma') < K_{\max}(\Gamma^*)$, then there exists $n' \in \Gamma'$, and $n \in \Gamma^*$ that satisfies $K(n') < K(n)$. By Lemma 5.6, $\mathrm{PT}(n') > \mathrm{PT}(n)$, indicating that $\Pr(h^*(n') + g^*(n') < C) > \Pr(h^*(n) + g^*(n) < C)$, which contradicts the assumption that Γ^* is the optimal path. Hence, a path is the optimal one if it minimizes $K_{\max}(\Gamma)$.

Conversely, let $\Gamma = \arg\min_{\Gamma'' \in \mathcal{G}} K_{\max}(\Gamma'')$, then for all Γ' that is different from Γ, for all $n \in \Gamma, \exists n' \in \Gamma'$, such that $K(n) < K(n')$. Thus by Lemma 5.6, $\mathrm{PT}(n) > \mathrm{PT}(n')$ for n' in any arbitrary path that is not Γ in the graph, and hence Γ that minimizes $K_{\max}(\Gamma)$ is the optimal path.

Theorem 5.8: When a feasible solution exists, the proposed algorithm terminates if and only if an optimal path is found.

PROOF: **Algorithm 3** can only terminate by finding the goal node, or after depleting the OPEN set. However, the OPEN set can never be empty before termination if there is a feasible path from s to goal point. Hence **Algorithm 3** must terminate by finding a goal point.

Next we show that **Algorithm 3** terminates only by finding an optimal path to the goal node. Suppose that the algorithm terminates by finding a path, Γ' other than the optimal path Γ^*, then by Lemma 5.7, $K_{\max}(\Gamma^*) < K_{\max}(\Gamma')$, that is, there exists $n' \in \Gamma'$, $n \in \Gamma^*$ such that $K(n) < K(n')$. Thus during the propagation process, **Algorithm 3** would have selected n for expansion rather than n', contradicting the assumption that the algorithm terminates by finding Γ'. Hence the algorithm must terminate by finding the optimal path to the goal node.

5.3 Complexity Analysis

In our analysis, we derive the worst case running time for **Algorithm 3**, and compare it with dynamic programming based planning methods, such as A*, to demonstrate the computational efficiency of the proposed planning algorithm. Let us suppose the flow field forecast is available on $N \times N$ grid points in the deployment domain, and suppose that the domain is partitioned into M cells by performing **Algorithm 1**.

To derive the worst case running time of the proposed algorithm, we first consider the partitioning. Since one junction

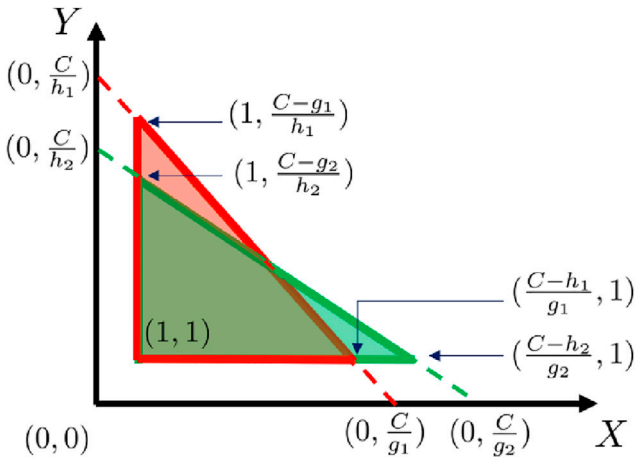

FIGURE 2 | Illustration of computing PT(n_1) and PT(n_2). The red triangle is $S_1 = \{(x,y)|h_1y + g_1x < C, x \in [1, H_{\max}, y \in [1, H_{\max}]]\}$, and the green triangle is $S_2 = \{(x,y)|h_2y + g_2x < C, x \in [[1, H_{\max}], y \in [1, H_{\max}]]\}$.

must be formed by the boundary of at least two cells, the total number of junctions cannot exceeds $M(M-1)$, and hence the total number of nodes in the graph is at most $M(M-1)$. In one iteration process, the sorting operation (line 13), and computation of the key function, the minimum branch cost, and the heuristics (line 29, line 23 and line 27 are performed for one time. Suppose that the OPEN set is implemented using a heap data structure, the worst case running time of the operation in line 13 is $\mathbb{O}(\log(M(M-1)))$. We assume that the computation of the key function, the minimum branch cost, and the heuristics can be performed in constant time. There can be at most $M(M-1)$ iterations during the entire execution, before the OPEN set is depleted. Hence, the worst case running time of **Algorithm 3** is $\mathbb{O}(M(M-1)\log(M(M-1))$.

The worst case running time of A* is $\mathbb{O}(2N^2 \log N)$ (Nilsson, 2014). Thus, the proposed algorithm is more computationally efficient than A* if $M(M-1) < N^2$, meaning that **Algorithm 1** partitions the domain into less number of cells than the number of rectangular cells in the original gridded domain.

6 EXPERIMENT AND SIMULATION RESULTS

In this section, we provide the results of the implementation of our flow field modeling and path planning methods in a simulated experiment. First, we describe the simulated experimental set-up and recent field experiments, which serve as a strong test of the methods due to the magnitude and variability of the flow. We validate the proposed flow modeling algorithm by comparing the estimated flow model parameters generated from the proposed flow estimation algorithm with the glider estimated flow collected during the experiment. Based on the estimated flow model, simulation of the bounded cost path planning algorithm is performed, and its performance is compared with other AUV path planning algorithms.

6.1 Experimental and Simulation Setup

Our study is motivated by the use of underwater gliders off the coast near Cape Hatteras, North Carolina, US as part of a 16-months experiment (Processes driving Exchange At Cape Hatteras, PEACH) to study the processes that cause exchange between the coastal and deep ocean at Cape Hatteras, a highly dynamic region characterized by confluent western boundary currents and convergence in the adjacent shelf and slope waters. Underwater gliders, AUVs that change their buoyancy and center of mass to "fly" in a sawtooth-shaped pattern, were deployed on the shelf and shelf edge to capture variability in the position of the Hatteras Front, the boundary between cool, fresh water on the shelf of the Mid Atlantic Bight and the warmer, saltier water observed in the South Atlantic Bight.

While the energy efficiency of the glider's propulsion mechanism permits endurance of weeks to months, the forward speed of the vehicles is fairly limited (0.25–0.30 m/s), which can create significant challenges for navigation in strong currents. Use of a thruster in a so-called "hybrid" glider configuration can increase forward speed to approximately 0.50 m/s, but at great energetic cost. The continental shelf near Cape Hatteras is strongly influenced by the presence of the Gulf Stream, which periodically intrudes onto the shelf, resulting in strong and spatially variable flow that can be nearly an order of magnitude greater than the forward speed of the vehicle (2+ m/s). With realistic estimates of the spatial and temporal variability of the flow, path planning can provide a significant advantage for successful sampling.

We deployed one glider off Oregon Inlet, NC May 16, 2017 and recovered it 14 days later. For its mission, the glider was initially tasked to sample along a path with offshore, inshore,

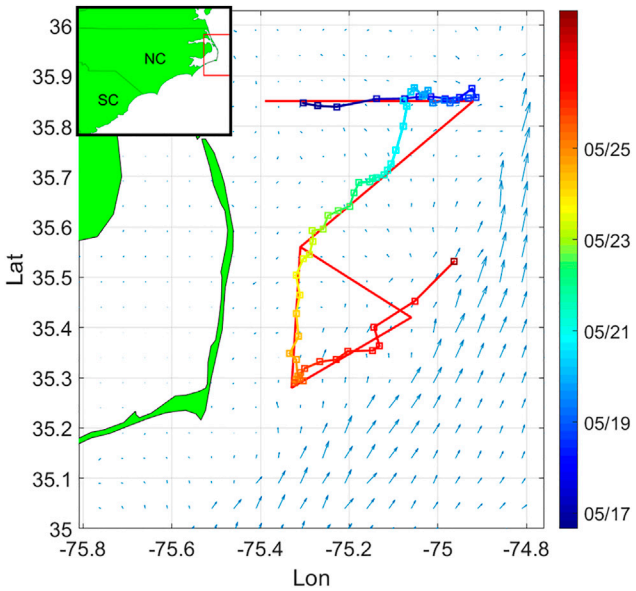

FIGURE 3 | Survey domain near Cape Hatteras. The curve represents glider trajectory during the first PEACH deployment. The red line path is the pre-assigned sampling pattern. Squares denote the glider surfacing positions along trajectory, and color of the trajectory depicts timestamps. The arrows represent the NCOM-predicted flow field at the starting time of the deployment.

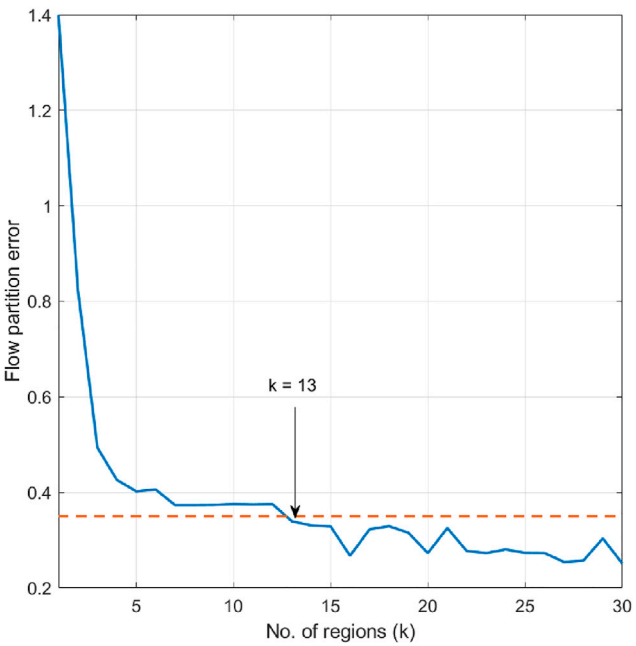

FIGURE 4 | Flow partition error when the number of cells is set as $k = 1, \ldots, 30$.

and triangular segments designed to sample critical flow features (**Figure 3**), and was not used with a thruster. The glider surfaced approximately every 4 h to update its position with a GPS fix, communicate with shore, transmit a subset of data, and most importantly, receive mission updates and commands to adapt sampling.

6.2 Flow Modeling Using Glider Experimental Data

In this example, we present flow modeling results using the proposed flow partition and parameter estimation methods.

The flow map forecast is given by a 1-km horizontal resolution version of the Navy Coastal Ocean Model

(NCOM, Martin, 2000), made available by J. Book and J. Osborne (Naval Research Laboratory, Stennis Space Center, United States). In the domain of interest, the ocean model flow forecast is given at 106 × 106 rectangular grid points. Tidal flow accounts for much of the short-term (< 24 hour) temporal variation of the flow field. Hence the partition time interval is taken over multiple periods of the largest tidal constituent, the lunar semidiurnal M_2 tide (period 12.42 h). Maximum flow speed in this area is 2.2788 m/s, approximately 7.5 times the vehicle speed, and 4.5 times the speed of a hybrid glider using a thruster. We set the upper bound for flow partition error to be 0.35 m/s, which is about 15% of the maximum flow speed in the domain. **Figure 4** describes the flow partition error in the case of different selection of cell number. Since the flow partition error goes below the upper bound when $k = 13$, the number of cells is chosen as 13. We smooth the cell boundaries into straight lines using the Least Mean Square method. Even though smoothing the cell boundaries might overlook more detailed spatial variability of the flow field, it helps to reduce the computational cost of solving the planning problem, specifically, in solving **eq. 17** and **eq. 20**. The partitioned flow field is shown in **Figure 5**. Comparing **Figures 3**, **5**, it can be seen that the proposed algorithm captures the major spatial variation of the flow field, by separating the high speed flow regions from the area where the flow is at lower speed.

At each surfacing, the vehicle position is given by the GPS location. When the glider is underwater, we use linear interpolation to estimate the heading and vehicle position. The vehicle's forward speed is zero when it is at the surface of the ocean, and the vehicle drifts freely with the surface current. This violates the constant forward speed assumption stated in Assumption 2.1. Hence, we remove the segment of data when the vehicle is drifting at surface, and then compute the estimated flow parameters by the proposed algorithm. Glider speed is initialized to be 0.3 m/s, while the flow parameters are initialized by the flow vectors found by partitioning the NCOM data. Since the vehicle trajectory crosses cell 3, 6, 7, 10, and does not enter other cells, the glider trajectory does not satisfy persistent excitation condition described in Lemma 5.3. Hence only the flow parameters in cells 3, 6, 7, 10 can be updated by the adaptive updating law, while the flow parameters in the rest cells remain to be the initial value. To justify performance of the proposed flow parameter estimation algorithm, we use the ADCIRC (Advanced Circulation) model output (Luettich et al., 1992) to model the tidal flow component, and derive the non-tidal glider estimated flow speed by subtracting ADCIRC reported flow from the flow parameter estimate. The de-tided glider estimated flow speed is considered as the ground truth of flow parameters in the corresponding cells. The root mean square error (rmse) between the estimated parameter and the ground truth in cell 3, 6, 7, 10 is shown in **Table 1**. It can be seen that in all of the four cells, the estimated flow parameters is in good agreement with the true flow parameters. The rmse in all of the four cells is within 5% of the maximum flow speed in the domain. **Figure 6** shows the comparison between the estimated flow parameters and the true flow parameter value in one of the cells that the glider trajectory visits. It is shown that in cell 7, the estimated flow parameter matches well with the true value.

6.3 Bounded Cost Path Planning

In this example, we present simulation results of AUV bounded cost path planning. Since the flow field is of high speed in the domain of interest, we assume that the glider is sampling the domain using combined propulsion of buoyancy and thrusters for the operation. Hence the AUV through-water speed is set to be 0.5 m/s. The simulations are run on a core i7 at 1.80 GHz PC with 32GB RAM.

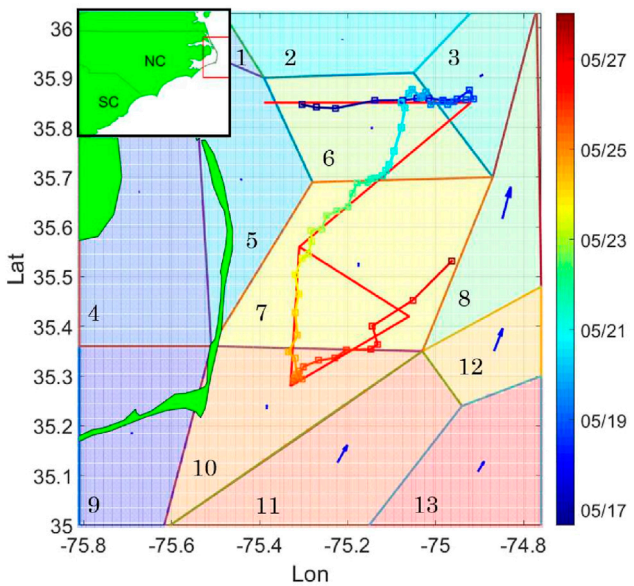

FIGURE 5 | Partitioned cells of the survey domain. The polygons are the partitioned regions. The blue arrows represent uniform flow speed in each of the cells generated from the proposed algorithm.

TABLE 1 | Root mean square error between the estimated and the true flow parameters.

Cell number	Flow parameter	Estimation error (m/s)
Cell 3	W-E flow	0.0558
	N-S flow	0.0082
Cell 6	W-E flow	0.0526
	N-S flow	0.0831
Cell 7	W-E flow	0.0415
	N-S flow	0.0388
Cell 10	W-E flow	0.0961
	N-S flow	0.0975

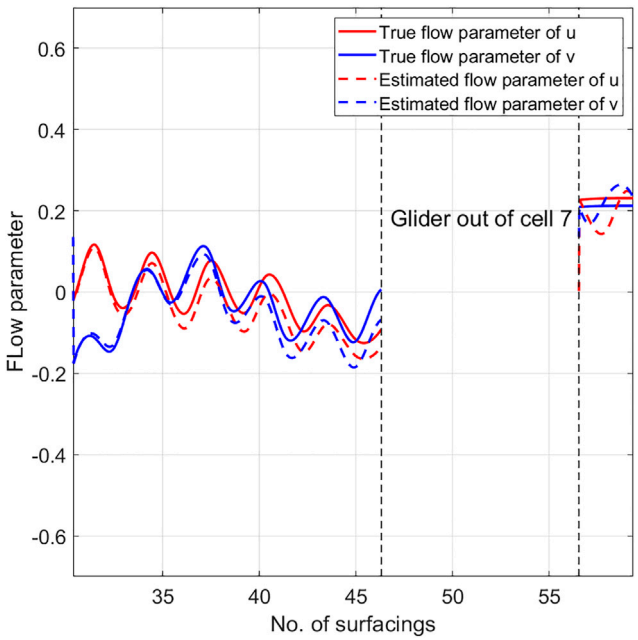

FIGURE 6 | Estimated flow parameters and the ground truth value in cell 7.

Figure 7 shows one example of the proposed bounded cost path planning method. The start position and goal position is assigned as $(-75.60, 35.06)$ and $(-74.98, 35.83)$ in longitude and latitude, respectively. Upper bound of the travel time is set as 72 h. The travel cost of the resulting path is 62.650 h, which satisfies the bounded cost constraint. As shown in the figure, the generated path makes a detour and takes advantage of the strong northward ocean flow to travel to the goal position.

We perform A* (Carroll et al., 1992) and Level Set Method (LSM) (Lolla et al., 2014) as comparison to the proposed method. 15 test cases are generated in total. Each test case $\mathbb{T} = \{\text{Start}, \text{Goal}, \text{d}\}$ is built by first assign the distance between the start and the goal node d to be 20 km, 50 km, 80, or 100 km, then randomly place the Start point in the domain, and select the Goal node so that the distance to the start node is d. The computation time column in **Table 2** shows comparison of the averaged computational time for A*, LSM, and the proposed algorithm. **Table 3** presents the post-hoc analysis results of the simulation. The post-hoc analysis rejects null hypotheses of the same performance, i.e. the proposed algorithm spends less computation time to solve the planning problem than the A* and the LSM method, for all different scenarios of d. The difference between the three algorithms is due to the number of nodes in the graph. By partitioning the domain into 13 cells, the proposed algorithm searches for the optimal path in a graph with only 13×12 nodes, while both the A* and LSM algorithm searches for the optimal path in a domain containing 106×106 nodes. Thus, the computational cost of the proposed algorithm is significantly lower than the other two methods.

Further, we compare the percentage of increase in the computation time when d increases. In **Table 2**, the % of increase column shows the increase in the computation cost when d increases from 20 to 50, 80, and 100, respectively. The percentage is calculated by considering the computation time of each algorithm when $d = 20$ km as the base time, and divide the increase of computation time when d scales up by the base time. It can be seen that when the domain of interest scales up, the

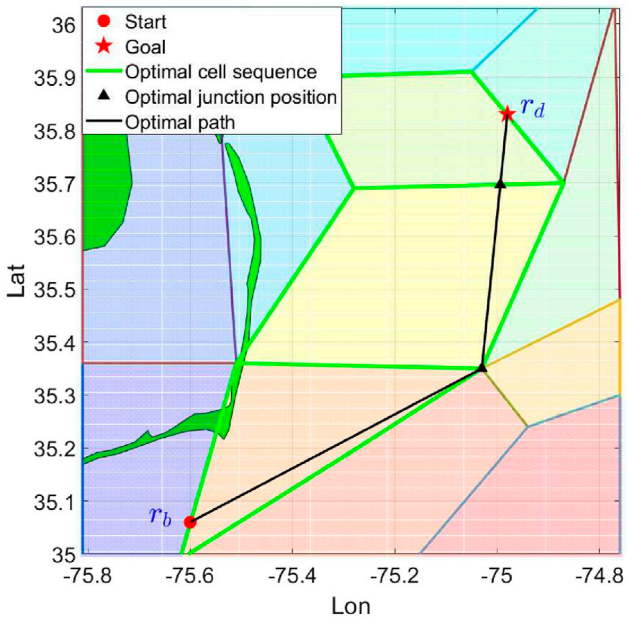

FIGURE 7 | Example of simulation case. The resulting path is computed by the proposed method.

TABLE 2 | Computation time comparison of A*, Level Set Method, and the proposed algorithm. Avg. comp. time represents the averaged computation time for each simulation scenario, and STD comp. time represents the standard deviation of the computation time. % of increase describes the percentage increase in the computation cost when d increases.

Method	d (km)	Avg. Comp. Time (s)	Std comp. Time	% Of increase
Algorithm 3	20	0.1576	0.0365	—
	50	0.1603	0.0385	1.7%
	80	0.2796	0.1127	77%
	100	0.4276	0.2318	171%
A*	20	1.6776	0.5199	—
	50	7.4376	0.7692	343%
	80	11.8376	1.3216	605%
	100	14.9040	1.1900	788%
LSM	20	86.8376	9.4397	—
	50	145.7043	30.0730	67%
	80	210.6376	23.4036	142%
	100	241.9043	25.6065	178%

TABLE 3 | Post-hoc analysis of simulation comparison between the proposed method, A*, and LSM. The mean and STD of difference describe the mean and standard deviation of the computation time difference between the proposed method and the two other methods. The significance level is set as $\alpha = 0.05$ when computing the *p*-value.

Method	d (km)	Mean of difference	Std of difference	t-score	*p*-value
A*	20	1.5200	0.5108	11.53	$2e^{-8}$
	50	7.2773	0.7744	36.40	0
	80	11.5580	1.2980	34.49	0
	100	14.4767	1.1168	50.20	0
LSM	20	86.6800	9.4511	35.52	0
	50	145.5440	30.0805	18.74	0
	80	210.3580	23.4411	34.76	0
	100	241.4767	25.6747	36.43	0

computational cost of the proposed algorithm has the least increase, compare with A* and LSM. This is because as d increases, both A* and LSM have to expand significantly more nodes before finding the optimal solution. For the proposed algorithm, the number of nodes to be expanded stays relatively constant as d increases. Hence, its computation cost does not increase as much as A* and LSM as the domain scales up.

It is worth mentioning that the proposed algorithm achieves decreased computation cost by compromising the path quality. Even though optimality of the planned path is guaranteed in the partitioned flow field, as shown in Theorem 5.8, the planned path may not be optimal in the actual flow field, since the partitioned flow field is different from the actual flow field. We identify the compromised path quality as the major constraint of the proposed algorithm.

In cases where the domain is larger, **Algorithm 1** may still result in large number of cells, leading to increased computation cost in solving the bounded cost planning problem. In such scenarios, stochastic optimization methods, such as the differential evolution method, may be helpful in further reducing the computation cost of solving the planning problem. We refer to a survey paper on the differential evolution methods (Das et al., 2016) for this matter.

7 CONCLUSION

In this paper, a bounded cost path planning method is developed for underwater vehicles assisted by a data driven flow field modeling method. The main advantage of the proposed modified PTS method is that it is more computational efficient comparing to A* and LSM in solving AUV planning problem in time-invariant 2D fields, as demonstrated by the simulation result. Major limitation of the proposed algorithm is the compromised solution quality, resulting from the model reduction error introduced by the flow partition process. The proposed method has the potential to be extended to other path planning applications where the task performance is sensitive to planner's computational efficiency. Future work will include performing the proposed method in real glider deployments, to compare the planned trajectory with the real mission trajectory, where drift and time-varying fields happen. Future work will also include comparing the proposed method with other algorithms, such as the differential evolution algorithms.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/navoceano-ncom-reg. The flow map data analyzed in this work is given by the Navy Coastal Ocean Model made available by J. Book and J. Osborne of Naval Research Laboratory, Stennis Space Center, US.

AUTHOR CONTRIBUTIONS

MH, SC, and FZ contributed to design and theoretical analysis of the methods, and wrote the first draft of the manuscript. HZ, CE and FZ wrote sections of the manuscript and provided guidance for the research work. All authors contributed to the manuscript revision, read, and approved the submitted version.

REFERENCES

Carroll, K. P., McClaran, S. R., Nelson, E. L., Barnett, D. M., Friesen, D. K., and Williams, G. N. (1992). "AUV Path Planning: An A* Approach to Path Planning with Consideration of Variable Vehicle Speeds and Multiple, Overlapping, Time-Dependent Exclusion Zones," in Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology.

Chang, D., Liang, X., Wu, W., Edwards, C. R., and Zhang, F. (2014). Real-time Modeling of Ocean Currents for Navigating Underwater Glider Sensing Networks. *Coop. Robots Sensor Networks, Stud. Comput. Intelligence* 507, 61–75. doi:10.1007/978-3-642-39301-3_4

Chang, D., Zhang, F., and Edwards, C. R. (2015). Real-Time Guidance of Underwater Gliders Assisted by Predictive Ocean Models. *J. Atmos. Oceanic Tech.* 32, 562–578. doi:10.1175/jtech-d-14-00098.1

Chassignet, E. P., Hurlburt, H. E., Smedstad, O. M., Halliwell, G. R., Hogan, P. J., Wallcraft, A. J., et al. (2007). The HYCOM (HYbrid Coordinate Ocean Model) Data Assimilative System. *J. Mar. Syst.* 65, 60–83. doi:10.1016/j.jmarsys.2005.09.016

Cho, S., and Zhang, F. (2016). "An Adaptive Control Law for Controlled Lagrangian Particle Tracking," in Proceedings of the 11th ACM International Conference on Underwater Networks & Systems, 1–5.

Cho, S., Zhang, F., and Edwards, C. R. (2021). Learning and Detecting Abnormal Speed of marine Robots. *Int. J. Adv. Robotic Syst.* 18, 1729881421999268. doi:10.1177/1729881421999268

Cui, R., Li, Y., and Yan, W. (2015). Mutual Information-Based Multi-AUV Path Planning for Scalar Field Sampling Using Multidimensional RRT. *IEEE Trans. Syst. Man, Cybernetics: Syst.* 46, 993–1004.

Das, S., Mullick, S. S., and Suganthan, P. N. (2016). Recent Advances in Differential Evolution - an Updated Survey. *Swarm Evol. Comput.* 27, 1–30. doi:10.1016/j.swevo.2016.01.004

Eichhorn, M. (2013). *Optimal Routing Strategies for Autonomous Underwater Vehicles in Time-Varying Environment*. Robotics and Autonomous Systems. 67, 3–43. doi:10.1016/j.robot.2013.08.010

Gammell, J. D., Barfoot, T. D., and Srinivasa, S. S. (2018). Informed Sampling for Asymptotically Optimal Path Planning. *IEEE Trans. Robot.* 34, 966–984. doi:10.1109/tro.2018.2830331

Griffa, A., Piterbarg, L. I., and Özgökmen, T. (2004). Predictability of Lagrangian Particle Trajectories: Effects of Smoothing of the Underlying Eulerian Flow. *J. Mar. Res.* 62, 1–35. doi:10.1357/00222400460744609

Haidvogel, D. B., Arango, H., Budgell, W. P., Cornuelle, B. D., Curchitser, E., Di Lorenzo, E., et al. (2008). Ocean Forecasting in Terrain-Following Coordinates: Formulation and Skill Assessment of the Regional Ocean Modeling System. *J. Comput. Phys.* 227, 3595–3624. doi:10.1016/j.jcp.2007.06.016

Hart, P., Nilsson, N. J., and Raphael, B. (1968). *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. IEEE Transaction of Systems Science and Cybernetics SSC, 4.

Haza, A. C., Piterbarg, L. I., Martin, P., Özgökmen, T. M., and Griffa, A. (2007). A Lagrangian Subgridscale Model for Particle Transport Improvement and Application in the Adriatic Sea Using the Navy Coastal Ocean Model. *Ocean Model.* 17, 68–91. doi:10.1016/j.ocemod.2006.10.004

Hou, M., Zhai, H., Zhou, H., and Zhang, F. (2019). "Partitioning Ocean Flow Field for Underwater Vehicle Path Planning," in *OCEANS 2019-Marseille* (IEEE), 1–8.

Ioannou, P. A., and Sun, J. (1995). *Robust Adaptive Control*. Prentice-Hall Upper Saddle River, NJ.

Karaman, S., and Frazzoli, E. (2011). Sampling-based Algorithms for Optimal Motion Planning. *Int. J. Robotics Res.* 30, 846–894. doi:10.1177/0278364911406761

Khalil, H. K. (1996). *Nonlinear Systems*. 2nd edn. Upper Saddle River, NJ: Prentice-Hall.

Kim, S.-J., Koh, K., Lustig, M., Boyd, S., and Gorinevsky, D. (2007). An Interior-Point Method for Large-Scale-Regularized Least Squares. *IEEE J. Sel. Top. Signal. Process.* 1, 606–617. doi:10.1109/JSTSP.2007.910971

Kuffner, J. J., and LaValle, S. M. (2000). "RRT-connect: An Efficient Approach to Single-Query Path Planning," in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation (Symposia Proceedings (Cat. No. 00CH37065) (IEEE)), 995–1001.

Kularatne, D., Bhattacharya, S., and Hsieh, M. A. (2018). Going with the Flow: a Graph Based Approach to Optimal Path Planning in General Flows. *Auton. Robot* 42, 1369–1387. doi:10.1007/s10514-018-9741-6

Kularatne, D., Bhattacharya, S., and Hsieh, M. A. (2017). Optimal Path Planning in Time-Varying Flows Using Adaptive Discretization. *IEEE Robotics Automation Lett.* 3, 458–465.

Leonard, N. E., Paley, D. A., Davis, R. E., Fratantoni, D. M., Lekien, F., and Zhang, F. (2010). Coordinated Control of an Underwater Glider Fleet in an Adaptive Ocean Sampling Field experiment in Monterey Bay. *J. Field Robotics* 27, 718–740. doi:10.1002/rob.20366

Lermusiaux, P. F. J. (2006). Uncertainty Estimation and Prediction for Interdisciplinary Ocean Dynamics. *J. Comput. Phys.* 217, 176–199. doi:10.1016/j.jcp.2006.02.010

Li, W., Lu, J., Zhou, H., and Chow, S.-N. (2017). Method of Evolving Junctions: A New Approach to Optimal Control with Constraints. *Automatica* 78, 72–78. doi:10.1016/j.automatica.2016.12.023

Lolla, T., Lermusiaux, P. F. J., Ueckermann, M. P., and Haley, P. J. (2014). Time-optimal Path Planning in Dynamic Flows Using Level Set Equations: Theory and Schemes. *Ocean Dyn.* 64, 1373–1397. doi:10.1007/s10236-014-0757-y

Luettich, R. A., Westerink, J. J., Scheffner, N. W., et al. (1992). ADCIRC: an advanced three-dimensional circulation model for shelves, coasts, and estuaries. Report 1, Theory and methodology of ADCIRC-2DD1 and ADCIRC-3DL. Tech. rep., Coastal Engineering Research Center, Mississippi, US.

Martin, P. J. (2000). *Description of the Navy Coastal Ocean Model Version 1.0. Tech. Rep. NRL/FR/7322–67900-9962*. Mississippi, US: Naval Research Lab Stennis Space Center.

Mokhasi, P., Rempfer, D., and Kandala, S. (2009). Predictive Flow-Field Estimation. *Physica D: Nonlinear Phenomena* 238, 290–308. doi:10.1016/j.physd.2008.10.009

Narendra, K. S., and Annaswamy, A. M. (1989). *Stable Adaptive Systems*. Prentice-Hall.

Nilsson, N. J. (2014). *Principles of Artificial Intelligence*. Massachusetts, US: Morgan Kaufmann.

Ozog, P., Carlevaris-Bianco, N., Kim, A., and Eustice, R. M. (2016). Long-term Mapping Techniques for Ship hull Inspection and Surveillance Using an Autonomous Underwater Vehicle. *J. Field Robotics* 33, 265–289. doi:10.1002/rob.21582

Panda, M., Das, B., Subudhi, B., and Pati, B. B. (2020). A Comprehensive Review of Path Planning Algorithms for Autonomous Underwater Vehicles. *Int. J. Autom. Comput.* 17, 321–352. doi:10.1007/s11633-019-1204-9

Pereira, A. A., Binney, J., Hollinger, G. A., and Sukhatme, G. S. (2013). Risk-aware Path Planning for Autonomous Underwater Vehicles Using Predictive Ocean Models. *J. Field Robotics* 30, 741–762. doi:10.1002/rob.21472

Rhoads, B., Mezic, I., and Poje, A. C. (2012). Minimum Time Heading Control of Underpowered Vehicles in Time-Varying Ocean Currents. *Ocean Eng.* 66, 12–31.

Roberge, V., Tarbouchi, M., and Labonté, G. (2012). Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. *IEEE Trans. Ind. Inform.* 9, 132–141.

Sastry, S., and Bodson, M. (1994). *Adaptive Control: Stability, Convergence and Robustness*. New Jersey, US: Prentice-Hall.

Savidge, D. K., Austin, J. A., and Blanton, B. O. (2013a). Variation in the Hatteras Front Density and Velocity Structure Part 1: High Resolution Transects from Three Seasons in 2004-2005. *Continental Shelf Res.* 54, 93–105. doi:10.1016/j.csr.2012.11.005

Savidge, D. K., Austin, J. A., and Blanton, B. O. (2013b). Variation in the Hatteras Front Density and Velocity Structure Part 2: Historical Setting. *Continental Shelf Res.* 54, 106–116. doi:10.1016/j.csr.2012.11.006

Shchepetkin, A. F., and McWilliams, J. C. (2005). The Regional Oceanic Modeling System (ROMS): a Split-Explicit, Free-Surface, Topography-Following-Coordinate Oceanic Model. *Ocean Model.* 9, 347–404. doi:10.1016/j.ocemod.2004.08.002

Smith, R. N., Chao, Y., Li, P. P., Caron, D. A., Jones, B. H., and Sukhatme, G. S. (2010). Planning and Implementing Trajectories for Autonomous Underwater Vehicles to Track Evolving Ocean Processes Based on Predictions from a Regional Ocean Model. *Int. J. Robotics Res.* 29, 1475–1497. doi:10.1177/0278364910377243

Soulignac, M. (2011). Feasible and Optimal Path Planning in strong Current fields. *IEEE Trans. Robot.* 27, 89–98. doi:10.1109/tro.2010.2085790

Stern, R., Felner, A., van den Berg, J., Puzis, R., Shah, R., and Goldberg, K. (2014). Potential-based Bounded-Cost Search and Anytime Non-parametric A*. *Artif. Intelligence* 214, 1–25. doi:10.1016/j.artint.2014.05.002

Stern, R. T., Puzis, R., and Felner, A. (2011). "Potential Search: A Bounded-Cost Search Algorithm," in *Twenty-First International Conference on Automated Planning and Scheduling*.

Subramani, D. N., and Lermusiaux, P. F. J. (2016). Energy-optimal Path Planning by Stochastic Dynamically Orthogonal Level-Set Optimization. *Ocean Model.* 100, 57–77. doi:10.1016/j.ocemod.2016.01.006

Thayer, J. T., Stern, R., Felner, A., and Ruml, W. (2012). "Faster Bounded-Cost Search Using Inadmissible Estimates," in Twenty-Second International Conference on Automated Planning and Scheduling.

Xiang, X., Jouvencel, B., and Parodi, O. (2010). Coordinated Formation Control of Multiple Autonomous Underwater Vehicles for Pipeline Inspection. *Int. J. Adv. Robotic Syst.* 7, 3. doi:10.5772/7242

Zamuda, A., Hernández Sosa, J. D., and Adler, L. (2016). Constrained Differential Evolution Optimization for Underwater Glider Path Planning in Sub-mesoscale Eddy Sampling. *Appl. Soft Comput.* 42, 93–118. doi:10.1016/j.asoc.2016.01.038

Zamuda, A., and Hernández Sosa, J. D. (2014). Differential Evolution and Underwater Glider Path Planning Applied to the Short-Term Opportunistic Sampling of Dynamic Mesoscale Ocean Structures. *Appl. Soft Comput.* 24, 95–108. doi:10.1016/j.asoc.2014.06.048

Zamuda, A., and Sosa, J. D. H. (2019). Success History Applied to Expert System for Underwater Glider Path Planning Using Differential Evolution. *Expert Syst. Appl.* 119, 155–170. doi:10.1016/j.eswa.2018.10.048

Zeng, Z., Lammas, A., Sammut, K., He, F., and Tang, Y. (2014). Shell Space Decomposition Based Path Planning for AUVs Operating in a Variable Environment. *Ocean Eng.* 91, 181–195. doi:10.1016/j.oceaneng.2014.09.001

Zeng, Z., Lian, L., Sammut, K., He, F., Tang, Y., and Lammas, A. (2015). A Survey on Path Planning for Persistent Autonomy of Autonomous Underwater Vehicles. *Ocean Eng.* 110, 303–313. doi:10.1016/j.oceaneng.2015.10.007

Zhai, H., Hou, M., Zhang, F., and Zhou, H. (2020). *Method of Evolving junction on Optimal Path Planning in Flow fields*. IEEE Transactions on Robotics.

Zhang, F. (2016). Cyber-maritime Cycle: Autonomy of marine Robots for Ocean Sensing. *FNT in Robotics* 5, 1–115. doi:10.1561/2300000037

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.