IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

State-Continuity Approximation of Markov Decision Processes via Finite Element Methods for Autonomous System Planning

Junhong Xu, Kai Yin, Lantao Liu

***Abstract*— Motion planning under uncertainty for an autonomous system can be formulated as a Markov Decision Process with a continuous state space. In this paper, we propose a novel solution to this decision-theoretic planning problem that directly obtains the continuous value function with only the first and second moments of the transition probabilities, alleviating the requirement for an explicit transition model in the literature. We achieve this by expressing the value function as a linear combination of basis functions and approximating the Bellman equation by a partial differential equation, where the value function can be naturally constructed using a finite element method. We have validated our approach via extensive simulations, and the evaluations reveal that compared to baseline methods, our solution leads to better planning results in terms of path smoothness, travel distance, and time costs.**

I. INTRODUCTION

Many autonomous vehicles that operate in flow fields, e.g. aerial and marine vehicles, can be easily influenced by environmental disturbances. For example, autonomous marine vehicles might experience ocean currents as in Fig. 1. Vehicles' uncertain behavior in an externally disturbed environment can be modeled using a decision-theoretic planning framework where the substrate is the Markov Decision Process (MDP) [1]. Since vehicles operate in a continuous domain, to obtain a highly accurate solution, it is desirable to solve the continuous-state-space MDP directly. However, this is generally difficult because the exact algorithmic solutions are known to be computationally challenging [2]. Besides, finding the solutions to MDPs typically requires knowing an accurate transition model, which is usually an unrealistic assumption. Existing works that apply the MDP for navigating aerial vehicles [3] or autonomous underwater vehicles (AUVs) [4], [5] typically use a simplified version of the continuous state space MDP. They represent the original MDP using a grid-map based representation and assume the vehicle can only transit to adjacent grid cells. This coarse simplification poorly characterize the original problem, and thus may lead to inconsistent solutions.

In this work, we propose a novel method that obtains a high-quality solution in continuous state space MDPs without requiring the exact form of the transition function. Compared to the majority of continuous MDP frameworks, we tackle the difficulties in large scale MDP problems from a different perspective: the integration of two layers of approximations. The value function is approximated by a linear combination of basis functions and the Bellman equation is approximated by a diffusion type partial differential equation (PDE), which allows us to express the Bellman equation using minimum characteristics of transition probabilities. This combination naturally leads to the applications of the Finite Element Method (FEM) for the solution.

J. Xu and L. Liu are with the Luddy School of Informatics, Computing, and Engineering at Indiana University, Bloomington, IN 47408, USA. E-mail: {xu14, lantao}@iu.edu. K. Yin is with Expedia Group. E-mail: kyin@expediagroup.com. J.X. and K.Y. equally contributed.

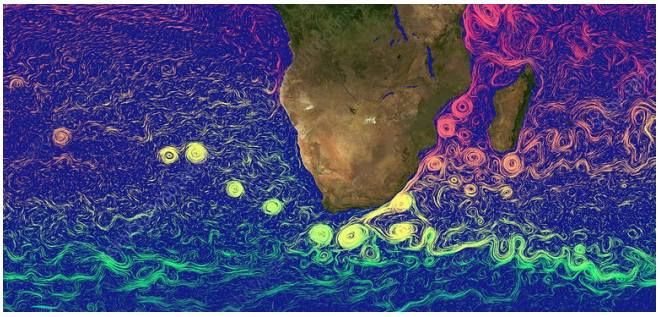


Fig. 1: Oceans currents can cause significant disturbances for autonomous marine vehicles. The Agulhas Ring gyres of southern Africa are unusually strong. (Source: NASA.)

Specifically, we first approximate the value function by a weighted linear combination of finite basis functions. Then using the Taylor expansion of the value function [6], [7], we show that it satisfies a diffusion-type PDE, which only depends on the first and second moments of the transition probability. We apply the FEM to solve the PDE with suitable boundary conditions. The method is based on discretization of the workspace into small patches in which the continuous value function may be approximated by a finite linear combination of basis functions, and the resulting approximation naturally extends over the entire continuous workspace. Combining these procedures, we propose an approximate policy iteration algorithm to obtain the final policy and a continuous value function. Our framework in principle allows us to compute the policy on the entire planning domain (space). Finally, we validate our method in a scenario involving navigating a marine vehicle in the ocean. Our simulation results show that the proposed approach produces results superior to the classic grid-map based MDP solutions.

II. RELATED WORK

Planning under the presence of action uncertainty can be viewed as a decision-theoretic planning problem which is usually modeled using MDPs [8]. Most existing methods that solve MDPs in the robotic planning literature use tabular methods, i.e., converting the continuous state space into a set of discrete indices [9], [1]. For example, in [3], the

authors model the task of minimum energy-based planning for aerial vehicles under a wind disturbance using a finite-state MDP. Similar work [10], [11], [12], [13] addresses tasks of minimum time and energy planning under uncertain ocean currents as an MDP problem. These grid-based solutions typically use a histogram-type approximation and assume the same values within each discretized state representation. To achieve good accuracy, the histogram-type approximation usually requires fine discretization resolution, which leads to significant computational time. In contrast, our method approximates the value function in a continuous form directly by the basis function.

Another well-known approach for optimal planning is based on incremental sampling-based methods, e.g., PRM* and RRT* [14]. These methods have been used in target searching [15], ship hull inspection [16], and autonomous driving [17]. The asymptotically optimal behavior of these algorithms is the result of the principle of dynamic programming [18], [19], which shares the same root as solving the MDP problems. However, these algorithms are not suitable for planning under uncertainty problems because they assume the motion of the vehicle is deterministic. Recent work [20], [21] extends the original RRT* and PRM* algorithms to solve planning tasks which have stochastic action effects. It builds a sequence of MDPs by sampling the states using an RRT-like method then performs asynchronous value iteration [22] iteratively. The above incremental sampling-based methods find trajectories and attempt to follow them. In contrast to these methods, our proposed method finds the optimal value function directly, which in turn generates optimal behavior.

It is also worth mentioning that many direct value function approximation methods have been proposed in machine learning and reinforcement learning [1]. Generally, the approximate value function is represented as a parametric functional form with weights [23], [24]. Fitted value iteration (FVI) is one popular sampling method that approximates continuous value functions [25], [26]. It requires batches of samples to approximate the true value function via regression methods. However, the behaviour of the algorithm is not well understood, and without careful choice of a function approximator, the algorithm may diverge or become very slow to converge [27], [28]. In contrast to the sampling methods, we attempt to approximate the value function using second-order Taylor expansion and calculate the resulting partial differential equation via a finite element method [29].

III. Problem Description and Formulation

This section describes decision-theoretic planning via Markov Decision Processes (MDPs). We consider planning of minimum time cost under uncertain actions caused by disturbances (e.g., ocean currents, air turbulence). This problem is usually formulated as an infinite time-horizon MDP.

We represent the infinite discrete time horizon MDP as a 4-tuple $(\mathbb{S}, \mathbb{A}, \mathcal{T}, R)$. The continuous *spatial* state space $\mathbb{S}$ denotes the entire autonomous vehicle planning region (or domain). Accordingly, a state $s \in \mathbb{S}$ is a spatial point $(x, y) \in \mathbb{R}^2$ on the plane, and it indicates the location of the vehicle. When the autonomous vehicle starts from a state, it takes an action to move toward the next state. We model the action space $\mathbb{A}$ as a finite space. An action depends on the state; that is, for each state $s \in \mathbb{S}$, we have a feasible action set $A(s) \subset \mathbb{A}$. The entire set of feasible state-action tuples is $\mathbb{F} := \{(s, a) \in \mathbb{S} \times \mathbb{A}\}$. There is a probability transition law $\mathcal{T}(s, a; \cdot)$ on $\mathbb{S}$ for all $(s, a) \in \mathbb{F}$. $\mathcal{T}(s, a; s')$ specifies the probability of transitioning to the state s' given the current state s with the chosen action a constrained by system dynamics. The final element $R : \mathbb{F} \to \mathbb{R}^1$ is a real-valued reward function that depends on state and action.

We consider the class of deterministic Markov policies [30], denoted by Π, i.e., the mapping $\pi : \mathbb{S} \to \mathbb{A}$ depends on the current state and the current time, and $\pi(s) \in A(s)$ for a given state. For a given initial state s_0, the expected discounted total reward is represented as:

$$v^{\pi}(s_0) = \mathbb{E}^{\pi}_{s_0}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k)\right], \tag{1}$$

where $\gamma \in [0, 1)$ is a discount factor that discounts the reward at a geometrically decaying rate. The aim is to find a policy π^* to maximize the expected cumulative discounted reward starting from the initial state s_0, i.e.,

$$\pi^*(s_0) = \arg\max_{\pi \in \Pi} v^{\pi}(s_0). \tag{2}$$

Accordingly, the optimal value is denoted by $v^*(s_0)$.

Under certain conditions [30], [31], it is well-known that the optimal solution satisfies the following recursive relationship

$$v(s) = \max_{a \in A(s)} \left\{R(s, a) + \gamma \cdot \mathbb{E}^a[v(s') \mid s]\right\}, \tag{3}$$

where $\mathbb{E}^a[v(s')|s] = \int \mathcal{T}(s, a; s')\, v(s') ds'$. This is the Bellman optimality equation, which serves as a basis to solve the problem expressed by Eq. (1) and Eq. (2). Popular algorithms to obtain the solutions include policy and value iteration algorithms as well as linear program based methods [30].

IV. Methodology

Our proposed methodological framework consists of four key interconnected elements for the solution to the MDP problem with a continuous or large-scale state space. First, we approximate the value function by a weighted linear combination of basis functions. Once the weights are determined, the value function for the entire state space is readily evaluated. Secondly, to alleviate the requirement for an explicit transition model, we approximate the Bellman equation by a diffusion-type partial differential equation (PDE). This is achieved through Taylor expansion of the value function with respect to the state variable [6], [7]. The obtained PDE only requires the first and second moments of the transition probabilities, potentially leading to wide applicability. Thirdly, integrating the above key steps together naturally leads to the Finite Element Method (FEM) for the solution to value function approximation given a policy. The FEM framework

allows us to transfer the approximate PDE to a linear system of equations whose solutions are exactly the weights in the linear combination of basis functions. Moreover, the resulting linear system depends on finitely many discrete states only. Finally, we propose a policy iteration algorithm based on the diffusion-type PDE and FEM to obtain the final policy and continuous approximate value function.

A. Value Function Approximation by Basis Functions

The value function plays a central role in the MDP. Because daunting computational requirements typically prevent any direct value computation on continuous or large scale states, value function approximation becomes necessary to obtain an approximate solution. We approximate the value function as the linear span of a finite set of basis functions $\{\phi_i(s)\}$,

$$v^\pi(s) = \sum_{i=1}^{n} w_i^\pi \phi_i(s), \tag{4}$$

where w_i^π are weights under the policy π. Then the methodological focus shifts to tractable and efficient algorithms for the weight computation as well as the policy improvement. Basis functions appropriate for the robotic motion planning will be specified within the following framework.

B. Diffusion-Type Approximation to Bellman Equation

Besides the difficulties in handling large scale continuous states, a critical feature of the Bellman optimality equation Eq. (3) is the requirement of being able to evaluate exact transition probabilities to all the next states for all state-action pairs. Such requirement is unrealistic in many applications. Therefore, we seek an approximation to Eq. (3) that only needs the minimum characteristics of transition probabilities.

We subtract $v(s)$ from both sides of Eq. (3) and then take Taylor expansions of the value function around s up to the second order [6]:

$$\begin{aligned} 0 = &\max_{a\in A(s)} \Big\{ R(s,a) + \gamma \left(\mathbb{E}^a[v(s') \mid s] - v(s) \right) \\ &- (1-\gamma)\, v(s) \Big\} \end{aligned} \tag{5}$$

$$\begin{aligned} \approx &\max_{a\in A(s)} \Big\{ R(s,a) + \gamma \left((\mu_s^a)^T \nabla v(s) + \frac{1}{2} \nabla \cdot \sigma_s^a \nabla v(s) \right) \\ &- (1-\gamma)\, v(s) \Big\}, \end{aligned} \tag{6}$$

where

$$\mu_s^a = \int \mathcal{T}(s,a;s')(s'-s)ds', \tag{7a}$$

$$\sigma_s^a = \int \mathcal{T}(s,a;s')(s'-s)(s'-s)^T ds'. \tag{7b}$$

The notation $\cdot$ in Eq. (6) denotes the inner product. Here we assume that $v(s)$ is continuously differentiable up to the second order. For most of motion planning problems where location may be modeled as state, we have $s = [x,y]^T$, the operator $\nabla \triangleq [\partial/\partial x, \partial/\partial y]^T$, and $\nabla \cdot \sigma_s \nabla \triangleq \sigma_{xx}^a \frac{\partial^2}{\partial x^2} + 2\sigma_{yx}^a \frac{\partial^2}{\partial y \partial x} + \sigma_{yy}^a \frac{\partial^2}{\partial y^2}$.

Eq. (6) approximates the Bellman equation of the original problem. As a result, the solution approximates the optimal solution to the MDP. The benefit of such an approximation is that it only uses the first and second moments of the transition probabilities (Eq. (7a)) rather than the full expression.

Under the optimal policy, Eq. (6) is a diffusion type partial differential equation (PDE) with the optimal value function as its solution. Similarly, given a policy, the solution to the PDE from Eq. (6) is the associated value function. Because the obtained value function from a given policy can be used to improve the policy, it inspires us to derive a policy iteration algorithm for the solution [6]:

- In the policy evaluation stage, we solve for a diffusion type PDE with proper boundary conditions to obtain the value function $v(s)$;
- In the policy improvement stage, we search for a policy that maximizes the values of the right hand side of Eq. (6) with $v(s)$ obtained from the previous policy evaluation stage.

The next section provides the PDE for the policy evaluation stage.

C. Partial Differential Equation Representation

It is well known that the suitable boundary conditions must be imposed in order to obtain the appropriate solution to a PDE. We thus find such boundary conditions for the PDE from Eq. (6) and provide the policy evaluation approach.

Since the value function does not have values outside the motion planning workspace, the directional derivative of the value function with respect to the unit normal vector at the boundary of the planning workspace must be zero. In addition, we constrain the value function at the goal state to be a constant value to ensure that there is a unique solution. Let $\mathbb{S}$ in the MDP formulation be the entire continuous spatial planning region (called the domain), denote its boundary by $\partial\mathbb{S}$, and the goal state by s_g. We also use $\hat{\boldsymbol{n}}$ to denote the unit vector normal to $\partial\mathbb{S}$ pointing outward. Under the policy π, we aim to solve the following diffusion type PDE:

$$\begin{aligned} -R(s,\pi(s)) = &\gamma \left((\mu_s^\pi)^T \nabla v(s) + \frac{1}{2} \nabla \cdot \sigma_s^\pi \nabla v(s) \right) \\ &- (1-\gamma)\, v(s), \end{aligned} \tag{8}$$

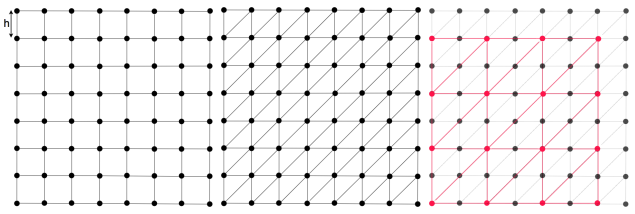
with boundary conditions

$$\sigma_s^\pi \nabla v(s) \cdot \hat{\boldsymbol{n}} = 0, \text{ on } \partial\mathbb{S} \tag{9a}$$

$$v(s_g) = v_g, \tag{9b}$$

where μ_s^π and σ_s^π indicate that μ_s and σ_s are obtained under the policy π; v_g is the value at the goal state. The condition (9a) is a type of homogeneous Neumann condition, and (9b) can be thought of as a Dirichlet condition [32]. We assume that the solution to the above Eq. (8) with boundary conditions (9) exists. Other conditions for a well-posed PDE can be found in [32].

It is generally impossible to obtain closed-form solution to Eq. (8). One must resort to certain numerical methods. Next we will introduce a finite element method to construct



(a) Grids of States (b) Meshes (c) Meshes on a subset of states

Fig. 2: Examples of triangular meshes with discrete states $\mathbb{S}'$.

the solution. The method just happens to leverage the linear approximation to the value function introduced in Section IV-A, and perfectly serves our objectives.

D. A Finite Element Method

Given a policy, we use a finite element method, in particular, a Galerkin method, to compute the weights for the linear value function approximation Eq. (4) through the PDE Eq. (8) with boundary conditions Eq. (9), and thereby obtain the value function approximation in the policy evaluation step.

We will sketch the main ideas involved in finite element methods, and refer the readers to literature [33], [34] for a full account of the theory [1]. The method consists of dividing the domain $\mathbb{S}$ into a finite number of simple subdomains, the finite elements, and then using a variational form (also called weak form) of the differential equation to construct the solution through the linear combination of basis functions over elements [34].

The variational or weak form of the problem amounts to multiplying the both sides of Eq. (8) by a test function $\omega(s)$ with suitable properties, and using integration-by-parts and the boundary condition Eq. (9) we have

$$-\int R\,\omega\,ds = \gamma\int(\mu_s^\pi)^T\nabla v\,\omega\,ds - \frac{\gamma}{2}\int\sigma_s^\pi\nabla v\cdot\nabla\omega\,ds - (1-\gamma)\int v\,\omega\,ds. \tag{10}$$

Here the test function $w(s)=0$ on s_g. It can be shown the solution to this variational form is also the solution to the original form.

Next, we partition the continuous domain $\mathbb{S}$ into suitable discrete elements. Typically the mesh is constructed from triangular elements, because they are general enough to handle all kinds of domain geometry. Suppose that the elements in our problem setting are pinned to a finite set of discrete states $\mathbb{S}'\subset\mathbb{S}$. For example, Fig. 2(a) shows a continuous, square domain approximated by an 8×8 set of discrete points $\mathbb{S}'$ separated by a constant distance h, where h stands for the resolution parameter and the dark dots are the discrete states. Fig. 2(b) shows an example of triangular finite elements, where each vertex of each element corresponds to a state in $\mathbb{S}'$.

[1]An example illustrating FEMs through a diffusion equation can also be found in the appendix of our paper at http://arxiv.org/abs/1903.00948.

Algorithm 1 Policy Iteration with State-Continuity Approximation and Finite Element Methods

Input: Transition function $\mathcal{T}$, reward function R, discount γ, continuous spatial states $\mathbb{S}$, a finite subset $\mathbb{S}'\subseteq\mathbb{S}$, and goal state s_g.

Output: Policy π and continuous value function $v(s)$.

1: Initialize $\pi, \forall s\in\mathbb{S}$, and set $i=0$.
2: **repeat**
3: Step 1. Finite Element Based Policy evaluation.
4: Compute μ_s^π and σ_s^π on the finite states $\mathbb{S}'\subseteq\mathbb{S}$;
5: Obtain continuous value function v^i by solving Eq. (8) with boundary conditions Eq. (9) using Finite Element Method described in Section IV-D;
6: Step 2. Policy improvement.
7: Update the policy π on $\mathbb{S}'$ using the value function v^i according to the following equation: $\pi(s)=\arg\max_{a\in A(s)}\left\{R(s,a)+\gamma\left(\mu_s^T\nabla v^i(s)+\frac{1}{2}\nabla\cdot\sigma_s\nabla v^i(s)\right)-(1-\gamma)\,v^i(s)\right.$
8: Set $i:=i+1$.
9: **until** $\pi(s)$ does not change for all $s\in\mathbb{S}'$

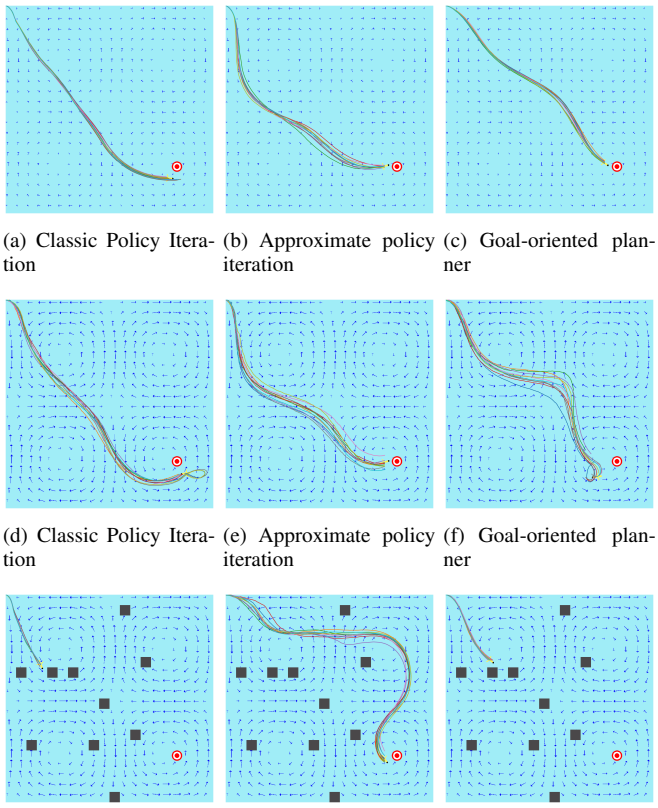
In addition to the linear approximation to the value function Eq. (4), we also represent the test function by a linear combination of basis functions, i.e., $\omega(s)=\sum_{i=1}^n c_i\phi_i(s)$. We use the Lagrange interpolation polynomials as basis functions for the test and value functions, constructed based on nodes of the triangle elements. Substituting the approximations of value and test functions into Eq. (10), we get an equation for the weights c_i and w_i. Because coefficients c_i should be arbitrary, this along with condition (9b) leads to a coupled system of linear algebraic equation of type $KW=F$. Each entry of the matrix K corresponds to integrals over the product of derivatives of basis functions on the right-hand side of Eq. (10). If we choose n number of Lagrange interpolation polynomials, the K is of $n\times n$ size. F corresponds to the remaining integrals on the both hand sides of Eq. (10). Solving this linear system gives the estimates of w_i, i.e., the approximate value function $v(s)$.

We make two notes here. First, we may choose a relatively small number of state points to form $\mathbb{S}'$ for the finite element method. This is because the finite element method approximate the solution with high precision. Fig. 2(c) provides an example of using a smaller number of larger elements, which are pinned to fewer discrete states compared to Fig. 2(b). The red dots are the selected states and red triangles are the corresponding elements. We will demonstrate this point in numerical examples in Section V. Second, different types of applications and equations may require different mesh designs.

E. Approximate Policy Iteration Algorithm

We summarize our approximate policy iteration algorithm in Algorithm 1.

In the policy evaluation step, we apply the FEM to a subset $\mathbb{S}'\subset\mathbb{S}$. If the computational cost is a concern, we can further



(a) Classic Policy Iteration (b) Approximate policy iteration (c) Goal-oriented planner

(d) Classic Policy Iteration (e) Approximate policy iteration (f) Goal-oriented planner

(g) Classic Policy Iteration (h) Approximate policy iteration (i) Goal-oriented planner

Fig. 3: Trajectory comparisons of 10 trials under gyre disturbances. The top three figures demonstrate the paths under weak gyre disturbances, where $A = 0.32$. Standard deviations of the disturbance velocity vector are set to $\sigma_x = \sigma_y = 1km/h$. The middle row shows the paths under stronger and more uncertain disturbances, where $A = 0.8$ and $\sigma_x = \sigma_y = 1.5km/h$. The last row demonstrates the paths with random obstacles (e.g., oil platforms or islets) in the environment.

reduce the size of $\mathbb{S}'$ by adjusting the resolution parameter h used to create the subset $\mathbb{S}'$ in the preceding section. Note that the obtained value function is continuous on the entire planning region. Therefore, it is possible to evaluate policy π on the whole state space $\mathbb{S}$.

V. Experiments

In this section, we present experimental results for autonomous vehicle planning with an objective of minimizing time cost to a designated goal position under uncertain ocean disturbances. Although we focus on marine vehicles in the experiment, our method is general enough to other types of robots that suffer from environmental disturbances. For example, our approach can be directly applied to plan an energy-efficient path for aerial vehicles under the wind disturbances, where the MDP is a typical modeling method [3].

Our method is implemented using the *FEniCS* package [35] a widely used FEM library. Throughout the experiments, we use first-order Lagrange polynomials as the basis functions. We compare the proposed approach with two other baseline methods. The first one is the classic MDP policy iteration (classic PI), which tessellates the continuous state space into grids, and the value function is represented by a look-up table [9]. This approach requires knowledge of the transition function of the MDP. The second one is a goal-oriented planner which maintains the maximum vehicle speed and always keeps the vehicle heading angle towards the goal direction. The latter method has been widely used in navigating underwater vehicles due to its "effective but lightweight" properties [36]. We measure the performance in terms of the time cost of the trajectories.

A. Evaluation with Gyre Model

1) Experimental Setup: We first consider the task of navigating an autonomous underwater vehicle (AUV) to a designated goal area subject to currents from a wind-driven ocean gyre. The gyre model is commonly used in analyzing flow patterns in large scale recirculation regions in the ocean [37]. In this experiment, the dimension of the ocean surface is set to $20km \times 20km$. Similar to [38], [12], we use a velocity vector field $\mathbf{v}^d(s) = [v_x^d(s), v_y^d(s)]^T$ to represent the gyre disturbance at each location of the 2-D ocean surface. Its velocity components are given by $v_x^d(s) = -\pi A \sin(\pi \frac{x}{e}) \cos(\pi \frac{y}{e})$ and $v_y^d(s) = \pi A \cos(\pi \frac{x}{e}) \sin(\pi \frac{y}{e})$ respectively, where $s = [x, y]^T$ is the location, A denotes the strength of the current, and e determines the size of the gyres.

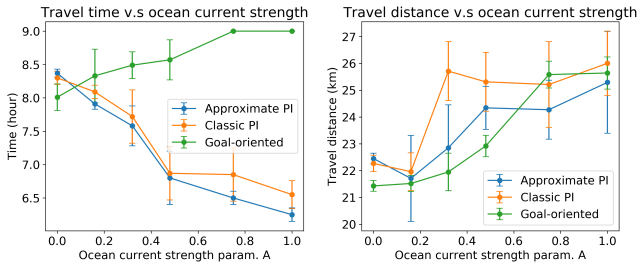
Due to estimation uncertainties [39], the resulting v_x^d and v_y^d do not accurately reflect the actual dynamics of the ocean disturbance. For effective and accurate planning, these uncertainties need to be considered, and they are modeled as additive environmental noises. To reduce the complexities on modeling and computing, we adopt the existing approximation methods [12], [10], [11] and assume the noise along two dimensions v_x^d and v_y^d follows independent Gaussian distributions $\tilde{\mathbf{v}}^d(s) = \mathbf{v}^d(s) + \mathbf{w}(s)$, where $\tilde{\mathbf{v}}^d(s)$ denotes the velocity vector at position s after introducing the uncertainty and $\mathbf{w}(s)$ is the noise. The components of the noise vector are given by

$$w_x(s) \sim \mathcal{N}(0, \sigma_x^2(s)), w_y(s) \sim \mathcal{N}(0, \sigma_y^2(s)), \tag{11}$$

where $\mathcal{N}(\cdot, \cdot)$ denotes Gaussian distribution, and σ_x^2 and σ_y^2 are the noise variance for each component, respectively.

The state is defined as the position of the AUV, i.e., $s \in \mathbb{S} \subseteq \mathbb{R}^2$. The actions are defined by the vehicle moving at its maximum speed (determined by the vehicle's capability) $v_{max} = 3km/h$ towards $Q = 8$ desired heading directions in the fixed world frame $\mathbb{A} = \{a_i | i \in \{1 \dots Q\}\}$, where $a_i = [v_{max} \cos(\frac{2\pi i}{Q}), v_{max} \sin(\frac{2\pi i}{Q})]^T$ describes the velocity vector of the vehicle. The vehicle's motion is affected by both the vehicle's action and the uncertain external disturbance. Thus, the next state s' of the vehicle starting at state s after following the desired action a for a fixed time interval dt is given by $s' = (a + \tilde{\mathbf{v}}^d(s))dt$. Since the velocity vector of the ocean current is perturbed by the additive Gaussian noise, the next state is a random variable, and the corresponding transition probability is given by

$$\mathcal{T}(s, a; s') = \mathcal{N}(\mu(s), \text{diag}[\sigma_x^2 dt^2, \sigma_y^2 dt^2]), \tag{12}$$



(a) (b)

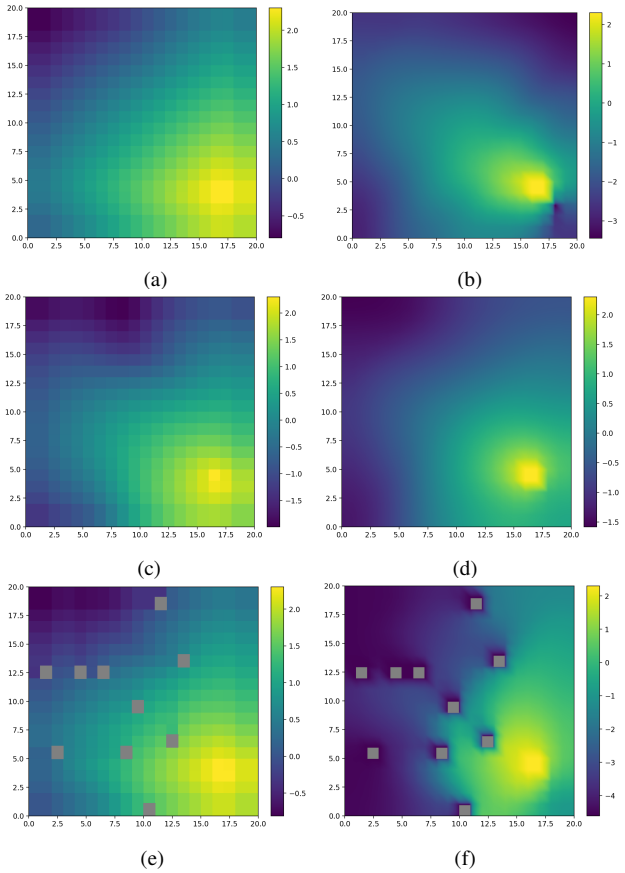
Fig. 4: The time costs and trajectory lengths yielded by the three methods with different disturbance strengths. These results are averaged over 10 trials.

A = 0.0	PI (400 states)	$h = 0.5km$	$h = 1km$	$h = 2km$
Time cost	8.3 ± 0.1	$\mathbf{8.23 \pm 0.16}$	8.37 ± 0.3	8.36 ± 0.1
Traj. len.	22.27 ± 0.3	$\mathbf{22.03 \pm 0.5}$	22.45 ± 0.2	22.46 ± 0.2
A = 0.16				
Time cost	8.09 ± 0.1	$\mathbf{7.9 \pm 0.16}$	7.91 ± 0.08	8.09 ± 0.12
Traj. len.	21.96 ± 0.7	21.8 ± 0.14	$\mathbf{21.71 \pm 1.6}$	23.58 ± 0.5
A = 0.32				
Time cost	7.72 ± 0.4	$\mathbf{7.52 \pm 0.11}$	7.58 ± 0.3	7.64 ± 0.29
Traj. len.	25.71 ± 1.1	$\mathbf{22.34 \pm 0.86}$	22.85 ± 1.6	23.85 ± 1.1
A = 0.48				
Time costs	6.87 ± 0.4	$\mathbf{6.51 \pm 0.1}$	6.8 ± 0.4	7.6 ± 0.1
Traj. len.	25.31 ± 1.1	$\mathbf{23.92 \pm 1.1}$	24.34 ± 0.8	24.43 ± 1.1
A = 0.75				
Time costs	6.85 ± 0.4	6.6 ± 0.5	$\mathbf{6.5 \pm 0.4}$	6.82 ± 0.12
Traj. len.	25.21 ± 1.6	24.5 ± 0.86	$\mathbf{24.34 \pm 1.1}$	24.53 ± 1.8
A = 1.0				
Time costs	6.55 ± 0.21	$\mathbf{6.08 \pm 0.12}$	6.25 ± 0.1	6.51 ± 0.16
Traj. len.	26.00 ± 1.2	25.74 ± 1.03	25.29 ± 1.0	$\mathbf{24.00 \pm 1.8}$

TABLE I: Time and trajectory costs averaged over 10 trials with different ocean current strengths A. The statistics of classic policy iteration (PI) and our method with different resolution parameters h are shown in each column. The best performing statistics are highlighted with a bold font.

where $\mu(s) = s + (a + \mathbf{v}^d(s))dt$. We have assumed that the ocean disturbances are constant near the current state s, and executing the action will not carry the vehicle too far away from the current state. To satisfy this assumption, we set the action execution time to a relatively small duration $dt = 0.1h$. The reward is one when the current state is the goal state, and zero otherwise. Because we are interested in minimizing the travel time, we set the reward discount factor as $\gamma = 0.9$. Also, we set goal and obstacle areas to be $1km \times 1km$ regions, and the states within these areas are absorbing states, i.e., the vehicle cannot transit to any other states if the current state is within these areas. Thus, the boundary conditions within the goal and obstacle areas have values of $\frac{1}{1-\gamma} = 10$ and $\frac{0}{1-\gamma} = 0$, respectively.

To model the transition function of the classic MDP planner, we follow the approach commonly used in AUV planning literature [40]. Specifically, each state s is represented by a regular grid, and the next state transition probabilities are only assigned to its 8-connected neighbors based on Eq. (12).

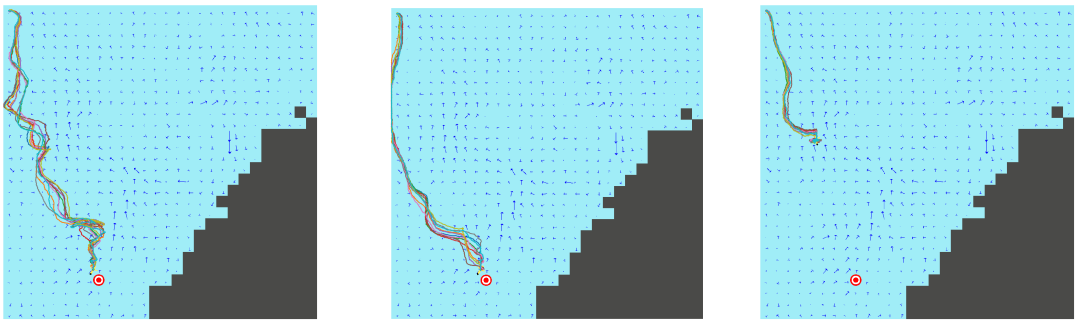


(a) (b) (c) (d) (e) (f)

Fig. 5: Value functions (values are in log scale) calculated from PI on the left and approximate policy iteration on the right. The environment parameters and settings are the same as for Fig. 3.

2) Trajectories and Time Costs: We present the performance evaluations in terms of trajectories and time costs. For the classic MDP planner (classic PI), we use 20×20 grids, and each cell has a dimension of $1km \times 1km$. To make a fair comparison, we use 20×20 state points for our method (approximate PI), and the corresponding resolution parameter is $h = 1km$. We set $s = 10$ which generates four gyre areas. To simulate the uncertainty of estimations during each experimental trial, we sample Gaussian noises from Eq. (11) and add them to the calculated velocity components of the gyre model.

We examine the trajectories of different methods under weak and strong disturbances in Fig. 3. We run each method for 10 times. The colored curves represent accumulated trajectories of multiple trials. For the weak disturbance, parameter A is set to 0.32 and the standard deviation of the noise is set to $1km/h$. The resulting maximum ocean current velocity is around $1km/h$, which is smaller than the vehicle's maximum velocity of $3km/h$. In this case, the vehicle has the capability of going forward against ocean currents. In contrast, the strong ocean current has a maximum velocity of $2.5km/h$, which is similar to the maximum vehicle velocity. By comparing the first and second rows of Fig. 3, we can easily observe that under weak ocean disturbances, the three



(a) (b) (c)

Fig. 6: Trajectories from the three methods using ocean data. (a) Classic policy iteration; (b) Approximate policy iteration; (c) Goal-oriented planner.

methods produce trajectories that converge on the target and look similar. This is because the weak disturbance results in smaller vehicle motion uncertainty. The goal-oriented planner has the shortest travel distance, but its travel time is the highest (Fig. 3(c) and Fig. 4) under the weak disturbance. On the other hand, the MDP policies can leverage the fact that moving in the same direction as the ocean current allows the vehicle to obtain a faster net speed and thus results in a shorter time to the goal. Specifically, with strong disturbances, instead of taking a more direct path towards the goal position, the MDP policies take a longer path but with a faster speed by following the direction of the ocean currents. In contrast, the goal-oriented planner is not able to reach the goal subject to the strong disturbance within the time budget (9h). Planning subject to the appearance of obstacles is shown in the last row of Fig. 3. We can observe that, among the three methods, only ours successfully navigates to the goal area with respect to the strong ocean currents. Our method attempts to minimize the travel time while avoiding possible collision. It first travels against the current to find a safe region and then follows the ocean current to advance to the goal area.

We further quantitatively evaluate the averaged time costs and trajectory lengths of the three methods subject to different disturbance strengths with fixed $\sigma_x = \sigma_y = 1km/h$, as illustrated in Fig. 4. The overall performance for the proposed approximate policy iteration is consistently better than the other two methods in terms of the time costs. The goal-oriented planner yields a slightly better result when no disturbance is present ($A = 0$). However, whenever noticeable disturbance is present, its performance degrades dramatically. In addition, it halts before arriving at the goal because it exceeds the given time budget ($9h$ with $A \geqslant 0.8$).

The superior performance of our method can also be explained in the value function plots shown in Fig. 5. The classic MDP planner produces a discrete value function, i.e., the state-values are the same within each $1km \times 1km$ cell. The resulting policy outputs the same actions regardless of the position of the vehicle within each grid cell. In contrast, the proposed method outputs a continuous value function, which varies smoothly across continuous space, directly. It can better characterize the actual optimal continuous value function. Thus, the resulting policy is more flexible and more diverse, which can lead to more intelligent behavior.

Finally, we perform detailed comparisons of approximate policy iteration using three different resolution parameters h in Table I. When the resolution is set to $h = m$, a number of $(\frac{20}{m})^2$ evenly-spaced state points are generated. In general, increasing the number of state points leads to better performance. We also observe that the gain between $h = 0.5km$ and $h = 1.0km$ is less than that between $h = 1.0km$ and $h = 2.0km$. This implies that when the solution is close to the optimal, the gain in the performance by a finer discretization becomes small. By comparing with the classic PI planner, it can be seen that the performance of our method is only marginally affected after halving the resolution and remains a good performance equivalent or superior to the classic PI planner.

B. Evaluation with Ocean Data

In addition to the gyre model, we also used Regional Ocean Model System (ROMS) [41] data to evaluate our approach. The dataset provides ocean current forecasts in the Southern California Bight region. Since ROMS only provides the ocean currents statistics at discrete locations, it cannot be directly used to evaluate the algorithms. To address this problem, we use Gaussian Process Regression (GPR) to interpolate ocean currents at every location on the 2-D ocean surface, from a single time snapshot. We used $h = 2km$ in our method when approximating the value function. For the classic MDP planner, the environment was discretized into 28×28 grids and each cell has a dimension of $2km \times 2km$. Fig. 6 shows the trajectories generated with the three methods. Note that the maximum speed of the ocean currents from this dataset is around $3km/h$, which is similar to the vehicle's maximum linear speed. We can see that trajectories from approximate policy iteration are smoother (shorter distance and time costs) than those from the classic policy iteration approach.

VI. Conclusions

In this paper, we propose a solution to solving the autonomous vehicle planning problem using a continuous

approximate value function for the infinite horizon MDP and using finite element methods as key tools. Our method leads to an accurate and continuous form of the value function even if we only use a smaller number of discrete states and if only the first and second moments of the model transition probability are available. We achieve this by leveraging a Taylor expansion of the Bellman equation to obtain a value function approximated by a diffusion-type partial differential equation, which can be naturally solved using finite element methods. Extensive simulations and evaluations have demonstrated advantages of our methods for providing continuous value functions, and for better path results in terms of path smoothness, travel distance and time costs.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[2] R. E. Bellman, *Adaptive control processes: a guided tour*. Princeton university press, 2015.

[3] W. H. Al-Sabban, L. F. Gonzalez, and R. N. Smith, "Wind-energy based path planning for unmanned aerial vehicles using markov decision processes," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 784–789.

[4] A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme, "Risk-aware path planning for autonomous underwater vehicles using predictive ocean models," *Journal of Field Robotics*, vol. 30, no. 5, pp. 741–762, 2013.

[5] S. Feyzabadi and S. Carpin, "Risk-aware path planning using hierarchical constrained markov decision processes," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2014, pp. 297–303.

[6] A. Braverman, I. Gurvich, and J. Huang, "On the taylor expansion of value functions," *Operations Research*, vol. 68, no. 2, pp. 631–654, 2020.

[7] J. Buchli, F. Farshidian, A. Winkler, T. Sandy, and M. Giftthaler, "Optimal and learning control for autonomous robots," *arXiv preprint arXiv:1708.09342*, 2017.

[8] C. Boutilier, T. Dean, and S. Hanks, "Decision-theoretic planning: Structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 11, pp. 1–94, 1999.

[9] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[10] V. T. Huynh, M. Dunbabin, and R. N. Smith, "Predictive motion planning for auvs subject to strong time-varying currents and forecasting uncertainties," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 1144–1151.

[11] L. Liu and G. S. Sukhatme, "A solution to time-varying markov decision processes," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1631–1638, 2018.

[12] D. Kularatne, H. Hajieghrary, and M. A. Hsieh, "Optimal path planning in time-varying flows with forecasting uncertainties," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[13] J. Xu, K. Yin, and L. Liu, "Reachable space characterization of markov decision processes with time variability," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[15] G. A. Hollinger, "Long-horizon robotic search and classification using sampling-based motion planning." in *Robotics: Science and Systems*, vol. 3, 2015.

[16] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1445–1464, 2012.

[17] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[18] O. Arslan and P. Tsiotras, "Dynamic programming guided exploration for sampling-based motion planning algorithms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4819–4826.

[19] ——, "Incremental sampling-based motion planners using policy iteration methods," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 5004–5009.

[20] V. A. Huynh, S. Karaman, and E. Frazzoli, "An incremental sampling-based algorithm for stochastic optimal control," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2865–2872.

[21] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "Firm: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 268–304, 2014.

[22] D. P. Bertsekas, "Dynamic programming and optimal control 3rd edition, volume ii," *Belmont, MA: Athena Scientific*, 2011.

[23] G. Konidaris, S. Osentoski, and P. Thomas, "Value function approximation in reinforcement learning using the fourier basis," in *Twenty-fifth AAAI conference on artificial intelligence*, 2011.

[24] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[25] C. Szepesvári and R. Munos, "Finite time bounds for sampling based fitted value iteration," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 880–887.

[26] A. Antos, C. Szepesvári, and R. Munos, "Fitted q-iteration in continuous action-space mdps," in *Advances in neural information processing systems*, 2008, pp. 9–16.

[27] D. J. Lizotte, "Convergent fitted value iteration with linear function approximation," in *Advances in Neural Information Processing Systems*, 2011, pp. 2537–2545.

[28] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 30–37.

[29] I. Babuška, "Error-bounds for finite element method," *Numerische Mathematik*, vol. 16, no. 4, pp. 322–333, 1971.

[30] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[31] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.

[32] L. C. Evans, *Partial Differential Equations: Second Edition (Graduate Series in Mathematics)*. American Mathematical Society, 2010.

[33] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[34] J. T. Oden and J. N. Reddy, *An introduction to the mathematical theory of finite elements*. Courier Corporation, 2012.

[35] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, "The fenics project version 1.5," *Archive of Numerical Software*, vol. 3, no. 100, 2015.

[36] K.-C. Ma, L. Liu, H. K. Heidarsson, and G. S. Sukhatme, "Data-driven learning and planning for environmental sampling," *Journal of Field Robotics*, vol. 35, no. 5, pp. 643–661, 2018.

[37] A. F. Blumberg and G. L. Mellor, "A description of a three-dimensional coastal ocean circulation model," *Three-dimensional coastal ocean models*, vol. 4, pp. 1–16, 1987.

[38] D. Kularatne, S. Bhattacharya, and M. A. Hsieh, "Time and energy optimal path planning in general flows." in *Robotics: Science and Systems*, 2016.

[39] R. N. Smith, J. Kelly, K. Nazarzadeh, and G. S. Sukhatme, "An investigation on the accuracy of regional ocean models through field trials," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3436–3442.

[40] G. A. Hollinger, A. A. Pereira, J. Binney, T. Somers, and G. S. Sukhatme, "Learning uncertainty in ocean current predictions for safe and reliable navigation of underwater vehicles," *Journal of Field Robotics*, vol. 33, no. 1, pp. 47–66, 2016.

[41] A. F. Shchepetkin and J. C. McWilliams, "The regional oceanic modeling system (roms): a split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean modelling*, vol. 9, no. 4, pp. 347–404, 2005.