Belief Space Partitioning for Symbolic Motion Planning

Mengxue Hou[1], Tony X. Lin[1], Haomin Zhou[2], Wei Zhang[3], Catherine R. Edwards[4], and Fumin Zhang[1]

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561121

***Abstract*— We propose a memory-constrained partition-based method to extract symbolic representations of the belief state and its dynamics in order to solve planning problems in a partially observable Markov decision process (POMDP). Our K-means partitioning strategy uses a fixed number of symbols to represent the partitions of the belief space and ensures the parameterization of the belief dynamics does not grow exponentially as the system dimension increases. By casting our problem as a partitioning of the POMDP, we can then solve planning problems using traditional symbolic planning solvers (such as HTN or A* solvers). Our work is motivated by an autonomous underwater vehicle navigation problem where the vehicle is affected by uncertain flow conditions and receives severely limited position observations. Simulation experiments are provided to validate the performance of the proposed algorithms.**

I. Introduction

For continous partially observable Markov decision processes (POMDPs), the introduction of a belief state, a continuous posterior distribution conditioned on historical observations [1], allows for the transformation of a POMDP problem to an MDP problem where each state is a possible belief state. However, solving planning problems in MDPs is often intractable as planning forward in time yields an exponential growth of belief-state and action pairs (i.e., the curse of dimensionality). Many methods in the literature design optimal control inputs by taking the expectation of the belief state as the true state, converting the problem into a deterministic search. Such strategies lead to an optimal policy for the Linear Quadratic Gaussian controller [2]. However, in cases where the state and observation equations are nonlinear and the cost is non-quadratic, such strategies yield sub-optimal results.

The POMDP literature contains many methods that approximate the continuous belief state so that the search for an optimal control strategy becomes tractable. The authors of [3] use Gaussian distributions to approximate the belief state and construct a deterministic approximation of the belief state dynamics. Other methods, such as the generalized cell mapping method proposed in [4], discretize the belief state into rectangular grids and represent the belief dynamics as a Markov chain. However, such approximations may result in MDPs that are either too large to solve computationally or insufficiently precise.

Once an approximation of the belief state and deterministic belief dynamics have been identified, symbolic planning strategies can then search for an optimal sequence of control actions while under constraints. For example, classic graph search methods such as breadth first search, iterative deepening, and A* have all been applied to solve symbolic search problems [5]. Compared to other continuous-state POMDP solutions, e.g., point-based and grid-based discretization strategies [6], [7] shows that symbolic search methods, due to the reduced dimension of the search space, can reduce the computation time of solving a POMDP. However, large action spaces can still dramatically hinder plan construction due to the large branching factor. As such, hierarchical task network (HTN) planners use primitive action groupings to prune the breadth of the search tree [8]. These approaches take advantage of the hierarchical nature of tasks, and search for plans by decomposing compound tasks until all tasks are decomposed into primitive actions. Thus, such approaches are well-suited to problems in which large groupings of actions are available.

Motivated by an autonomous underwater vehicle (AUV) navigation problem, we propose a discretization strategy of the belief space that allocates a fixed number of partitions according to regions where the AUV most likely visits. Unlike other approximation strategies, we use a K-means-based partition approach to allocate a fixed number of partitions according to a set of Monte Carlo simulations. As we pre-define our partition count, our approach avoids the dimensionality problem other abstraction methods must deal with while retaining abstraction accuracy.

AUVs have been employed for a variety of ocean sampling and surveillance tasks [9], [10], [11], [12] and must perform successful navigation through uncertain ocean flows while retaining an uncertain position estimate. Operational ocean models like the Regional Ocean Modeling System [13] and the Hybrid Coordinate Ocean Model [14] can provide spatio-temporal flow forecasts, but these forecasts can contain high error and uncertainty that arise from incomplete physics or uncertain boundary conditions [15]. In addition, underwater localization is a difficult problem, as many conventional terrestrial or aerial localization methods (GPS, map-based localization) are wholly unavailable. Generally, localization uncertainty can be reduced via acoustic beacons or periodic surfacings. However, acoustic positioning systems can be degraded by sever multi-pathing effects, large Doppler spread,

[1] Mengxue Hou, Tony X. Lin and Fumin Zhang are with School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA, 30309 {mhou30, tlin339, fzhang37}@gatech.edu

[2] Haomin Zhou is with School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia, USA, 30309 hz25@gatech.edu

[3] Wei Zhang is with Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, China, 518000 zhangw3@sustech.edu.cn

[4] Catherine R. Edwards is with Skidaway Institute of Oceanography, University of Georgia, Savannah, Georgia, USA, 31411 catherine.edwards@skio.uga.edu

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

and limited bandwidth [16], and frequent surfacings lead to a degradation of valuable underwater data collection.

Our major contributions are as follows: **i)** we propose a novel discretization strategy of the belief state that allows us to allocate a fixed number of partitions over regions that the AUV most likely will visit, **ii)** we construct the belief dynamics and deterministic predicate transition system from the abstracted partitions in order to solve the AUV navigation problem, and **iii)** we provide simulation studies comparing our proposed method with other approximation strategies demonstrating that the proposed algorithm achieves significantly reduced computation times.

The organization of this paper is as follows. In Section II, we formulate three problems necessary for solving planning problems in infinite dimensional POMDPs. In Section III, we discuss our solution to the first problem in which we propose a partitioning strategy for representing an infinite dimensional belief space with a finite number of cells. In Section IV, we describe our approach to computing the transition probabilities from the belief state abstraction using Monte Carlo simulations, and in Section V, we describe the mapping process of the partition distribution to a set of predicate symbols that will allow us to solve planning problems in POMDPs. Finally, in Section VI, we provide a simulated AUV example that demonstrates the efficacy of the proposed approach and conclude with some discussions in Section VII.

II. Problem Formulation

In this section, we describe a POMDP planning problem by decomposing the problem into three subsequent problems: **i)** The partitioning of the continuous belief state into a discretized distribution with a finite number of cells, **ii)** the identification of the transition probabilities between the cells, and **iii)** the transformation of the cell distribution to a symbolic representation for planning purposes.

A. Finite Partitioning of the Belief State

We introduce the full belief state of the AUV agent and the observation model. Let $x_k \in \mathbb{R}^3$ be the three-dimensional position, $y_k \in \{0,1\}^N$ be the underwater observations from N acoustic beacons, and $u_k \in \mathcal{U}$ be the controlled heading angle and depth change at timestep k. The vehicle's through-water speed is denoted as $V \in \mathbb{R}$. A general description of the system dynamics and the observation model can be described by

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, \omega_k) \\ y_k &= h(x_k, n_k) \end{aligned} \tag{1}$$

and ω_k and n_k are the process and observation noises. The specific AUV obeys the dynamics

$$x_k = F(x_k, t_k) + \begin{bmatrix} V(\cos(u_{1,k}) - \frac{\sqrt{2}}{2}|u_{2,k}|) \\ V(\sin(u_{1,k}) - \frac{\sqrt{2}}{2}|u_{2,k}|) \\ u_{2,k} \end{bmatrix} + w_k, \tag{2}$$

where the flow field, denoted as $F : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^3$ is given by a gyre model,

$$\begin{aligned} F^{(1)}(x,k) &= -\pi A \sin(\pi g(x^{(1)},k)) \cos(\pi x^{(2)}) \\ F^{(2)}(x,k) &= \pi A \cos(\pi g(x^{(1)},k)) \sin(\pi x^{(2)}) \frac{dg}{dx^{(1)}} \\ F^{(3)}(x,k) &= 0, \end{aligned} \tag{3}$$

where $g(x_1,k) = a_k (x^{(1)})^2 + b_k x^{(1)}$, $a_k = \epsilon \sin \omega k$, $b_k = 1 - 2\epsilon \sin(\omega k)$. $\omega \in \mathbb{R}$ is the frequency of gyre oscillation, and $\epsilon \in \mathbb{R}$ determines the amplitude of the oscillation. For observation model h, we assume beacons are detected with a certain probability when in range of a beacon, i.e., for a beacon j located at position r_j

$$\begin{aligned} \Pr(y_k^{(j)} = 1) &= p\mathbb{1}\{\|x_k - r_j\| \leq R_j\} \\ &\quad + (1-p)\mathbb{1}\{\|x_k - r_j\| > R_j\}, \\ \Pr(y_k^{(j)} = 0) &= (1-p)\mathbb{1}\{\|x_k - r_j\| \leq R_j\} \\ &\quad + p\mathbb{1}\{\|x_k - r_j\| > R_j\}. \end{aligned} \tag{4}$$

We assume that the AUV can apply 8 different move actions in the x-y plane and 2 actions to change between being at the surface and being at some depth, i.e., $\mathcal{U} = \mathcal{U}_{xy} \cup \mathcal{U}_z$ where $\mathcal{U}_{xy} = \{\begin{bmatrix} \frac{j\pi}{4} & 0 \end{bmatrix}^\intercal\}_{j=0}^{7}$ are the planar move actions and $\mathcal{U}_z = \{\begin{bmatrix} \frac{\pi}{4} & 1 \end{bmatrix}^\intercal, \begin{bmatrix} \frac{\pi}{4} & -1 \end{bmatrix}^\intercal\}$ are two depth changing actions, the first to enable surfacing and the second to enable diving. Notice that if any action in $\mathcal{U}_{xy}$ is taken, the AUV will move along the planar direction with no change in altitude while if any action in $\mathcal{U}_z$ is taken, then the AUV will change depth without changing its x-y position. While $\mathcal{U}_{xy}$ actions are useful for reaching target destinations, $\mathcal{U}_z$ actions enable the glider to surface in order to re-acquire an improved position estimate via GPS. In this work, we additionally assume that there is no process noise on the $\mathcal{U}_z$ actions and we can always surface or dive within one timestep.

However, since we are unable to directly observe the planar x-y components of the state x_k, we instead retain a density $b_k = \Pr(x_k | u_{k-1}, u_{k-2}, \ldots, u_1, y_k, y_{k-1}, \ldots, y_1)$. At each timestep, b_k is updated by the inputs u_{k-1} and noisy observations y_k. Note that in this case, b_k is a recursively updated infinite dimensional belief state leading to an intractable computation of an optimal input u_k.

As such, we are interested in computing an abstraction of b_k that makes computing an optimal u_k tractable. In addition, while other methods in the literature have utilized partitioning to make computing an optimal u_k feasible, such methods are subject to scaling issues as the dimensionality of the system increases. However, in many robotic applications, large portions of the belief space are completely un-utilized. Instead, we seek to find a partitioning that focuses a fixed number of K partitions in regions of the belief space most frequently visited in order to ensure the representation achieves a close approximation of the belief state without scaling as the dimensionality of the system increases. We assume that we are provided a set of zero-input trajectories $\Psi_0 = \{(\bar{x}_m, \bar{y}_m, \bar{u}_m = 0)\}_{m=1}^{M}$ drawn from a distribution of initial states x_0 and simulated forward using system (1). Note here that the zero-input trajectories reveal information about the system dynamics as the input to the system is additive as shown in (1).

Problem 2.1 (Finite Partitioning): Given the dynamics (1) and the set of trajectories Ψ, find a partitioning of the belief state b denoted $\Phi = \{\mathcal{R}_i\}_{i=1}^{K}$, where each partition $\mathcal{R}_i$ is associated with a probability $\Theta^{(i)}$ so that $\Theta = \left[\Theta^{(1)}, \ldots, \Theta^{(K)}\right]^{\intercal}$ is a distribution over the partitions Φ.

B. Identification of the Partitioned Belief Dynamics

Having found a focused partitioning Φ, we then seek to accurately represent the transitions between the partitions Φ. In particular, we are interested in identifying the evolution of Θ as a sequence of discrete actions u_k are selected, i.e.,

$$\Theta_{k+1} = T(u_k)\Theta_k. \tag{5}$$

We assume that a set of trajectories $\Psi_u = \{(\bar{x}_m, \bar{y}_m, \bar{u}_m = u)\}_{m=1}^{M}$ drawn from a distribution of initial states x_0 and simulated forward using system (1) are available where $\bar{u}$ for each trajectory is fixed to a specific input $u \in \mathcal{U}_{xy}$ throughout.

Problem 2.2 (Finite Belief Dynamics): Given a set of trajectories Ψ_u, estimate the transition matrix $T \in \mathbb{R}^{K \times K}$ that describes the evolution of the partition probabilities Θ as a function of the applied input.

C. Symbolic Predicate Transitions

The solutions of Problem 2.1 and Problem 2.2 provide us with a decomposition of the continuous belief space into a set of partitions Φ and an associated distribution Θ. However, to solve planning problems in the POMDP domain, we must construct a deterministic graph where nodes are abstracted Θ distributions and the corresponding children nodes are the resulting abstracted distributions from taking u_k. To do so, we must find a mapping from Θ to a new set of predicates that discretize how strongly we believe the AUV lies within any particular partition in Φ. In addition, we assume the AUV may choose to periodically surface to immediately observe its current position through a GPS measurement.

Problem 2.3 (Symbolic Predicate Transitions): Let $\mathcal{S} = \{\mathrm{at\_surf}, \mathrm{at\_depth}\} \cup \{\mathrm{BelH}_i, \mathrm{BelM}_i, \mathrm{BelL}_i\}_{i=1}^{K}$ be a set of boolean-valued predicates where each BelH_i, BelM_i, and BelL_i describes the discretized certainty level associated with partition $i \in \Phi$, and $\mathrm{at\_surf}, \mathrm{at\_depth}$ are associated with the current depth of the AUV. Given a set of partitions Φ and a distribution Θ over Φ, find mapping function $\Gamma : \mathcal{S} \times \Theta \to \{1, 0\}$ that maps from the distribution to the set of active predicates.

Once Problem 2.3 is solved, we can use the mapping $\Gamma(\mathrm{Bel}, \Theta_k)$ to check if the predicate Bel is true at time k as long as we retain the underlying distribution Θ_k at each time-step.

III. Focused Finite Partitioning

In this section, we discuss our approach to solving Problem 2.1. To identify the optimal cell partition, we use a finite number of particles to approximate the dynamics of the belief state. Suppose $\{r_{k-1}^{(m)}\}_{m=1}^{M}$ are drawn i.i.d. from b_{k-1}, and $r_{k|k-1}^{i}$ is drawn from $\Pr(r_k | u_{k-1}, r_{k-1}^{(i)})$ for each particle. Then $b_k(x_k)$ can be approximated by the probability mass function

$$\hat{b}_k(r_k) = \sum_{m=1}^{M} w_k^{(m)} \delta(r_k - r_{k|k-1}^{(m)}), \tag{6}$$

where δ denotes the Kronecker delta function. $w_k^{(m)}$ is the associated weight to each node, and weights of all nodes sum up to one. At each timestep, the particle position is first updated according to (2), and then the weight of each particle is updated by the Bayes's rule,

$$w_k^{(m)} = \frac{p(y_k^{(m)} | x_{k|k-1}^{(m)}, u_{k-1})}{\sum\limits_{m=1}^{M} p(y_k^{(m)} | x_{k|k-1}^{(m)}, u_{k-1})}. \tag{7}$$

Since future observations is unknown. We make the following assumption to determinize the future observation, and therefore determinize the future belief dynamics.

Assumption 3.1: We assume the future actual state will obtain the most likely values, and $y_k = h(\hat{x}_k, n_k)$, where $\hat{x} = \arg\max\limits_{x} b_k(x)$.

Remark 3.1: In order to determinize the belief dynamics to make belief space planning tractable, this is a commonly made assumption in literature, such as in [17], [18], [19], [20]. If less likely future observation is received during execution, replanning scheme can be used to re-evaluate action sequences given the updated belief state, so that the planner remains robust to future observation uncertainty.

Consider Monte Carlo simulation datapoint before resampling. Let $r_k^{(m)} \in \mathbb{R}^2$ denote the m^{th} particle position at timestep k, $m \in [1, M]$, and $w_k^{(m)} \in [0, 1]$ represent the particle weights. Let $z_k^{(m)} = [(r_k^{(m)})^{\intercal}, w_k^{(m)}]^{\intercal}$ denote the augmented particle data point. The simplified belief representation should well approximate the actual belief b_k. Hence we formulate the following constrained optimization problem to learn the optimal partition structure from the simulated particle trajectories. $\mu^{(i)}$ denotes the centroid of cell i, $\mu^{(i)} = [(\bar{x}^{(i)})^{\intercal}, \Theta^{(i)}]^{\intercal}$, where $\bar{x}^{(i)}$ is the centroid position of cell i, and $\Theta^{(i)}$ denotes the belief in cell i.

$$\begin{aligned} \min_{\mathcal{R}} J = & \sum_{k \in [1,n]} \sum_{i \in [1,K]} \sum_{r_k^m \in \mathcal{R}_i} \mathrm{dist}^2(z_k^{(m)}, \mu^{(i)}) \\ \text{s.t.} \quad & \sum_{i \in [1,K]} I\mu^{(i)} = 1, I\mu^{(i)} > 0, \forall i \in [1, K] \end{aligned} \tag{8}$$

where $I = [0, 0, 1]$. (8) is constrained by the non-negativity of the belief in each cell and the conservation of the total probability. Due to a trade-off between accuracy and speed, the squared distance function $\mathrm{dist} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ is defined as the Euclidean distance, $\mathrm{dist}^2(y_k, y_k') = (y_k - y_k')^T Q (y_k - y_k')$, where $y_k, y_k' \in \mathbb{R}^3$, and Q is a weight matrix.

We derive the belief space partition Φ by solving (8) using the K-means algorithm. 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, the first of which is to assign the points that are closest to a centroid to that centroid,

$$c_k^m = \arg\min_i \operatorname{dist}(z_k^{(m)}, \mu^{(i)}), \tag{9}$$

where $c_k^m \in [1, K]$ is the index of the partition that datapoint $z_k^{(m)}$ is assigned to. The second step is recomputing the cell centroid by taking average of all the data points assigned to this centroid. To guarantee that the constraints in (8) is satisfied, a normalization term ν is multiplied to the belief to guarantee the conservation of probability,

$$\mu^{(i)} = \nu \frac{\sum\limits_{k\in[1,n]} \sum\limits_{m\in[1,M]} \mathbb{1}(c_k^m = i) z_k^{(m)}}{\sum\limits_{k\in[1,n]} \sum\limits_{m\in[1,M]} \mathbb{1}(c_k^m = i)}. \tag{10}$$

These two steps are repeated until cell membership no longer changes.

Algorithm 1: Focused finite partition

```
Initialize particles by drawing $\{r_0\}_{m=1}^M$ i.i.d. from $b_0$;
for $k = 1$ to $n$ do
    Compute $r_{k|k-1}^{(m)}, m \in [1, M]$ by propagating the
      particles according to system dynamics (2)
      given control input being zero and randomly
      generated noise $\{\omega_i\}_{m=1}^M$;
    Compute $y_k, y_{k+1}, \dots, y_n$ from $\hat{x}_k, \hat{x}_{k+1}, \dots, \hat{x}_n$
      according to (4) using randomly generated noise
      $\{n_m\}_{m=1}^M$;
    For each $r_k^{(i)}$, update weights using (7) ;
end
Denote $M$ evaluation datapoints as
  $z_k^{(m)} = [(r_k^{(m)})^\intercal, w_k^{(m)}]^\intercal$, $m \in [1, M]$;
$Z = \cup_{k\in[1,n]} \cup_{m\in[1,M]} z_k^{(m)}$;
Randomly initialize cluster centroids $\mu^{(i)}$;
while not converges do
    For $z \in Z$, assign $z$ to the centroid that is closest
      to it using (9) ;
    Recompute all centroid using (10) ;
end
For all $i \in [1, K]$, $\mathcal{R}_i = \cup_m \cup_k \big(\mathbb{1}\{c_k^m = i\} r_k^{(m)}\big)$;
```

IV. Identification of the Partition Belief Dynamics

Leveraging the solution to Problem 2.1 designed in Section III, we now discuss our solution to Problem 2.2. Given the finite dimension partition, we project the infinite dimensional belief dynamics onto the partitioned cells to find the parameters of the Markov chain transition model in (5). Each element in column i and row j of the transition matrix represents the probability that the system state starts from cell $\mathcal{R}_j$, and transits to cell $\mathcal{R}_i$ at the next time step,

$$T^{(ij)}(u) = \Pr(x_k \in \mathcal{R}_i | x_{k-1} \in \mathcal{R}_j, u). \tag{11}$$

To evaluate (11), we assume that we are provided a set of trajectories $r_k^{(i)}$ drawn from a distribution of initial distribution of x_0 and simulated forward using (2) and (4), and the control input u for simulating the trajectories fixed throughout. The particle weights are updated by (7). We use the Ulam-Galerkin method [21] to compute the transition matrix, which estimates (11) by the number and the weights of particles moving from one cell to another,

$$T^{(ij)}(u) = \frac{1}{n} \sum_{k=1}^{n} \left[\frac{\sum\limits_{m=1}^{M} w_k^{(m)} \mathbb{1}\big(r_k^{(m)} \in \mathcal{R}_i\big) \mathbb{1}\big(r_{k-1}^{(m)} \in \mathcal{R}_j\big)}{\sum\limits_{m=1}^{M} w_{k-1}^{(m)} \mathbb{1}(r_{k-1}^{(m)} \in \mathcal{R}_j)} \right]. \tag{12}$$

The denominator represents the total mass of particles inside the cell $\mathcal{R}_j$, while the numerator is the mass transported from cell $\mathcal{R}_j$ to $\mathcal{R}_i$ at the next time step.

Algorithm 2: Identification of partitioned belief dynamics

```
for $u_i \in \mathcal{U}_{xy}$ do
    Initialize particles by drawing $\{r_0^{(m)}\}_{m=1}^M$ i.i.d.
      from $b_0$ ;
    for $k = 1$ to $n$ do
        Compute $r_{k|k-1}^{(m)}$, $m \in [1, M]$ by propagating
          the particles according to (2) given control
          input being fixed to $u_i$ throughout, and
          randomly generated noise $\{\omega_m\}_{m=1}^M$;
        Compute $y_k, y_{k+1}, \dots, y_n$ from
          $\hat{x}_k, \hat{x}_{k+1}, \dots, \hat{x}_n$ according to (4) using
          randomly generated noise $\{n_m\}_{m=1}^M$;
        Update weights for particles using (7);
    end
    for $i \in [1, K]$ do
        for $j \in [1, K]$ do
            Compute the transition probability from
              cell $j$ to $i$ using (12);
        end
    end
end
```

We find the initial condition of (5) by projecting the initial distribution of states, denoted as ρ_0 onto the partitions. Let $\phi(x) \in \mathbb{R}^K$ denote the spatial basis function we find through partitioning, $\phi(x) = \big[\mathbb{1}(x \in \mathcal{R}_1), \dots, \mathbb{1}(x \in \mathcal{R}_K)\big]^\intercal$. Since the initial belief is a continuous distribution, a set of sample points $x \in \mathbb{R}^2$ is selected, and the parameters can be found by minimizing the difference between the parameterized belief and the initial belief on these sample points,

$$\Theta_0 = \arg\min_{\Theta} \sum_{x\in\mathbb{R}^2} (\rho_0(x) - \Theta^T \phi(x))^2. \tag{13}$$

V. Identification of the Predicate Transitions

Having obtained partitions Φ and transition model T by solving Problems 2.1 and 2.2, we now discuss the solution to Problem 2.3. In this section, we define and describe the mapping function Γ to construct current set of active predicates s_k given the distribution over Φ, Θ_k which will allow us to solve the planning problems. We design Γ to evaluate each predicate according to the following rules

$$\Gamma(\text{Bel}, \Theta_k) = \begin{cases} 1, & \text{Bel} = \text{BelH}_i, \ \Theta_k^{(i)} > p_H \\ 1, & \text{Bel} = \text{BelM}_i, \ p_H \geq \Theta_k^{(i)} > p_L \\ 1, & \text{Bel} = \text{BelL}_i, \ p_L \geq \Theta_k^{(i)} \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Here, parameters p_H and p_L are threshold values to discretize the belief values. These values can be chosen to reflect the confidence precision necessary for satisfying desired planning goals. In this work, we choose $p_H = 0.7$ and $p_L = 0.2$. Applying (14) allows us to identify K active predicates that denote the confidence over each partition. We can then evaluate the predicates in $\mathcal{S}$ at each iteration by enumerating over BelH_j, BelM_j, BelL_j for $j \in \{1, \ldots, K\}$ with Γ and retaining a memory of the previous AUV depth for predicates at_surf, and at_depth as we assume depth change actions are deterministic. Let $s_{xy,k} = \{s | \Gamma(s, \Theta_k) = 1\}$ be the current set of active predicates computed by enumerating over each BelH_j, BelM_j, BelL_j for $j \in \{1, \ldots, K\}$ and let $s_{z,k} \in \{\text{at\_surf}, \text{at\_depth}\}$. Then, the total current set of active predicates is defined as $s_k = s_{xy,k} \cup s_{z,k}$.

From any s_k, we define the action set to be the finite set of inputs $\mathcal{U} = \mathcal{U}_{xy} \cup \mathcal{U}_z$, where the effects of $\mathcal{U}_{xy}$ are defined by the identified transition dynamics T and the effects of $\mathcal{U}_z$ are defined as an information gathering action for position estimation. Let $j_{\max} = \arg\max \Theta^{(j)}$ be the index of the partition with the highest probability. For planning purposes, we assume that the effects of taking surfacing action u_9 on the distribution Θ are

$$\Theta^{(i)} = \begin{cases} 1, & i = j_{\max} \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

while taking diving action u_{10} has no effect on Θ. However, we assume that move actions may only be performed at depth, yielding a precondition function

$$\text{Pre}(a_i, s_k) = \begin{cases} 1, & \text{if } s_{z,k} = \text{at\_depth}, i < 9 \\ 0, & \text{if } s_{z,k} = \text{at\_surf}, i < 9 \end{cases} . \quad (16)$$

Having defined (14), (15), and (16), we can construct a planning problem $\Sigma = (\mathcal{S}, \mathcal{U}, \gamma, s_0, g)$ where $g \in \mathcal{S}$ is a set of desired predicates to activate, and $\gamma : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ is a predicate transition function that determines the effects of taking an action u from a state s.

We can now construct γ based on a graph search using (14) to construct branches based on the action set $\mathcal{U}$ where each branch cost is a function dependent on Θ_k and u_k, and using (16) to prune branches according to whether they satisfy the desired preconditions. In this work, we solve this planning problems using a HTN planner to handle the preconditions of taking actions while using an A* planner to find the cost-optimal sequence of actions. An example of one iteration of the graph search is shown in Figure 1.

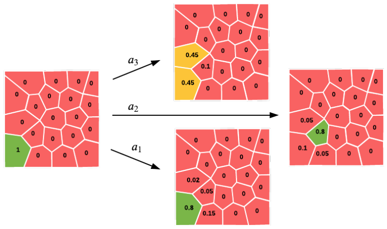

Fig. 1. Example of the belief state evolution over one iteration. By taking different control actions, three different Θ distributions are possible. The active predicate set is shown by the color: green, yellow, red for BelH, BelM, and BelL, respectively.

VI. Simulation Results

In this section, we provide the results of the implementation of the symbolic dynamics approximation and path planning methods in a simulated experiment of AUV deployment. Performance of the proposed methods is compared with the Generalized Cell Mapping algorithm.

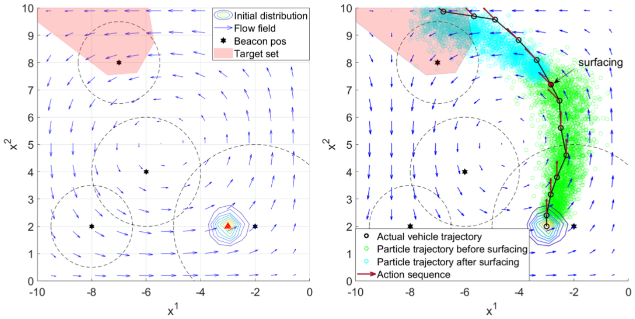

Fig. 2. **Left:** Setup of the simulation domain. The contour represents initial distribution of the state variable, and the pink region represents the terminal set. Beacons are represented as the black hexagram while the detection range is marked by the dashed curve. A snapshot of flow field is described by the blue arrows. **Right:** Actual vehicle trajectory under the influence of process noise and flow field. The belief state evolution is described by the simulated particle trajectories. Red arrows represent the sequence of actions designed by the HTN planning algorithm. At the 7^{th} surfacing (the red dot), surfacing and diving actions are injected into the control sequence to satisfy the constraint on variance of the belief state.

The beacon placement and detection range setup is shown at left on Figure 2, and the parameters in the simulation setup is shown in Table I. The initial distribution of vehicle position is set as a Gaussian distribution, and the terminal constraint of the planning problem is set as BelH_1 is true, meaning that the vehicle has high probability of being inside cell one. As the vehicle travels underwater, the belief will approach the uniform distribution, which contains no information of the actual vehicle position. Hence, we impose one constraint that the belief should concentrate in a small number of cells. If this constraint is violated, the vehicle is forced to surface to receive position estimation. We aim to find the minimum-time path. Hence the stage cost for the moving action is set as one. The surface/diving action is associated with a higher stage cost to penalize frequent surfacing.

Figure 3 shows partition of the belief space. The choice of the number of clusters is a trade-off between numerical

TABLE I

SIMULATION SETUP

Flow mag., angular vel. Vehicle speed	$A = 0.3$, $\omega = \pi/2$, $\epsilon = 0.01$ $V = 1$
Initial condition	$\rho_0 = \mathcal{N}(\begin{bmatrix} -3 \\ 2 \end{bmatrix}, \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix})$
Terminal condition	BelH_1 is true
Constraint on state var.	$\sum_{i=1}^{K} \mathrm{BelL}_i > 6$
Surf. and div. cost	5
Move action cost	1

TABLE II

COMPARISON BETWEEN GCM AND THE PROPOSED ALGORITHM

Generalized cell mapping			
Grid size	# of Pre.	Mod. time	Plan time
$10 \times 10 \times 2$	600	14.2845 s	**5.9873** s
$100 \times 100 \times 2$	6e4	15.5396	**54.3957** s
$150 \times 150 \times 2$	$1.35e5$	16.4289 s	**154.368** s
Proposed solution			
Grid size	# of Pre.	Mod. time	Plan time
$10 \times 10 \times 2$	120	18.3886	**3.9702** s
$50 \times 50 \times 2$	120	19.845 s	**4.7825** s
$100 \times 100 \times 2$	120	20.5480 s	**4.9796** s

accuracy and computational cost. We choose to partition the belief space into 20 cells, based on the "Elbow criterion" [22]. Boundaries of cells are smoothed into straight lines using the Least mean square method.

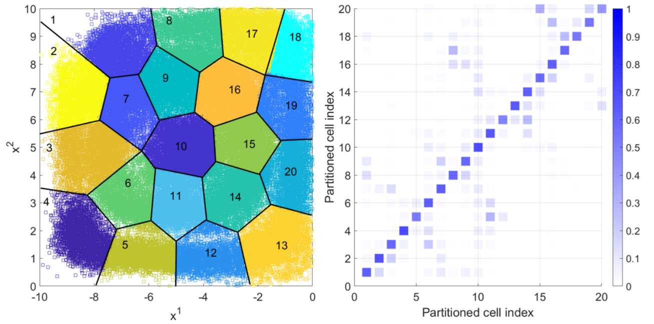

Fig. 3. **Left:** Partitioned cells in belief space. The colored squares represent datapoint position in Monte Carlo simulation. Squares with same color belong to the same partition. **Right:** Transition model of the parameterized belief state when $u = 0$.

Given the transition model, we use HTN to perform planning in symbolic space. Fig. 2, right shows the optimal control sequence computed by the proposed planning algorithm. The vehicle takes advantage of the flow speed, and goes along the streamline of the flow field, in order to move towards the target position in maximum total speed, and achieve minimum travel time. At the 7^{th} timestep, the constraint on state variance is not satisfied. Hence the HTN planner injects a surfacing action to the control sequence, so that the state variance is reset, and the constraint is satisfied.

To evaluate performance of the proposed algorithm, we compare our partition-based Markov chain approximation method to the generalized cell mapping method (GCM) [4]. The GCM method discretizes the belief space into rectangular grid cells, and uses the Ulam-Galerkin method to estimate the belief dynamics. Given the identified belief dynamics from both GCM and the proposed method, we use the HTN planning to solve for the optimal sequence of actions. The simulation comparison on belief dynamics identification, and symbolic planning is shown in Table II. The first and the second column show the number of predicates. Mod. time refers to the run time the two methods construct the Markov chain model of the belief state. Plan time describes the time to solve the symbolic planning problem. As shown in the table, for the two methods, computation time for identifying the belief dynamics does not have significant difference. GCM results in slightly less computation time, since the GCM uses the grid cells to partition the belief space, while the proposed algorithm takes an extra step to compute the optimal partition. However, the method proposed here significantly reduces the computational cost of symbolic planning. The main reason is that when grid size increases, the number of predicates will grow correspondingly. The large number of predicates introduces significant computation cost in evaluating the constraints in the symbolic planning problem. Therefore, the simulation comparison shows that by introducing the belief space partition, we significantly reduce the number of predicates, which reduces the total computation time of solving a POMDP problem. The difference of computation time for planning between the two algorithms can be explained by the worst case running time. Suppose the discretized grid cell in the x-y plane is $N \times N$, the worst case running time of running A* HTN planner with GCM is $\mathbb{O}(2 \times 3^{N^2}(N^2 \log 3 + \log 2))$ [23]. Suppose we partition the x-y plane into M cells, the worst case running time of A* HTN planner with belief abstraction is $\mathbb{O}(2 \times 3^M(M \log 3 + \log 2))$, which is less than GCM if $M < N^2$, meaning that we partition the domain into less number of cells compared with the grid based discretization.

VII. CONCLUSION

In this paper, we propose a symbolic representation and symbol space planning method for a general POMDP problem. Such symbolic representations allows for efficient abstraction of complex tasks of the infinite dimensional belief state. Given the symbol space abstraction, we leverage HTN and A* planning to find cost-optimal trajectories.

VIII. ACKNOWLEDGEMENT

The research work is supported by ONR grants N00014-19-1-2556 and N00014-19-1-2266; AFOSR grant FA9550-19-1-0283; NSF grants CNS-1828678, S&AS-1849228 and GCR-1934836; NRL grants N00173-17-1-G001 and N00173-19-P-1412 ; and NOAA grant NA16NOS0120028.

References

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

[2] A. Bensoussan and S. C. P. Yam, “Mean field approach to stochastic control with partial information,” *arXiv preprint arXiv:1909.10287*, 2019.

[3] L. P. Kaelbling and T. Lozano-Pérez, “Integrated task and motion planning in belief space,” *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1194–1227, 2013.

[4] J.-Q. Sun, F.-R. Xiong, O. Schütze, and C. Hernández, *Cell mapping methods*. Springer, 2018.

[5] Á. Torralba, V. Alcázar, P. Kissmann, and S. Edelkamp, “Efficient symbolic search for cost-optimal planning,” *Artificial Intelligence*, vol. 242, pp. 52–79, 2017.

[6] G. Norman, D. Parker, and X. Zou, “Verification and control of partially observable probabilistic systems,” *Real-Time Systems*, vol. 53, no. 3, pp. 354–402, 2017.

[7] L. Winterer, S. Junges, R. Wimmer, N. Jansen, U. Topcu, J.-P. Katoen, and B. Becker, “Strategy synthesis for POMDPs in robot planning via game-based abstractions,” *IEEE Transactions on Automatic Control*, 2020.

[8] A. Herzig, L. Perrussel, and Z. Xiao, “On hierarchical task networks,” in *European Conference on Logics in Artificial Intelligence*. Springer, 2016, pp. 551–557.

[9] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang, “Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay,” *Journal of Field Robotics*, vol. 27, no. 6, pp. 718–740, nov 2010. [Online]. Available: http://doi.wiley.com/10.1002/rob.20366

[10] R. N. Smith, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme, “Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a Regional Ocean Model,” *The International Journal of Robotics Research*, vol. 29, no. 12, pp. 1475–1497, 2010.

[11] P. Ozog, N. Carlevaris-Bianco, A. Y. Kim, and R. M. Eustice, “Long-term mapping techniques for ship hull inspection and surveillance using an autonomous underwater vehicle,” *Journal of Field Robotics*, vol. 33, pp. 265–289, 2016.

[12] X. Xiang, B. Jouvencel, and O. Parodi, “Coordinated formation control of multiple autonomous underwater vehicles for pipeline inspection,” *International Journal of Advanced Robotic Systems*, vol. 7, no. 1, p. 3, 2010.

[13] 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.

[14] E. P. Chassignet, H. E. Hurlburt, O. M. Smedstad, G. R. Halliwell, P. J. Hogan, A. J. Wallcraft, R. Baraille, and R. Bleck, “The HYCOM (HYbrid Coordinate Ocean Model) data assimilative system,” *Journal of Marine Systems*, vol. 65, no. 1, pp. 60 – 83, 2007.

[15] A. C. Haza, L. I. Piterbarg, P. Martin, T. M. Ozgokmen, and A. Griffa, “A Lagrangian subgridscale model for particle transport improvement and application in the Adriatic Sea using the Navy Coastal Ocean Model,” *Ocean Modelling*, vol. 17, no. 1, pp. 68–91, 2007.

[16] W. Wu, A. Song, P. Varnell, and F. Zhang, “Cooperatively mapping of the underwater acoustic channel by robot swarms,” in *Proceedings of the International Conference on Underwater Networks & Systems*, 2014, pp. 1–8.

[17] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, “Belief space planning assuming maximum likelihood observations,” in *Robotics: Science and Systems*, 2010.

[18] E. Zhou, M. C. Fu, and S. I. Marcus, “Solving continuous-state POMDPs via density projection,” *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1101–1116, 2010.

[19] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, “Efficient planning in non-Gaussian belief spaces and its application to robot grasping,” in *Robotics Research*. Springer, 2017, pp. 253–269.

[20] C. R. Garrett, C. Paxton, T. Lozano-Pérez, L. P. Kaelbling, and D. Fox, “Online replanning in belief space for partially observable task and motion problems,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5678–5684.

[21] E. M. Bollt and N. Santitissadeekorn, *Applied and computational measurable dynamics*. SIAM, 2013.

[22] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[23] N. J. Nilsson, *Principles of artificial intelligence*. Morgan Kaufmann, 2014.