

# ASPIRe: An Informative Trajectory Planner with Mutual Information Approximation for Target Search and Tracking

Kangjie Zhou<sup>1</sup>, Pengying Wu<sup>1</sup>, Yao Su<sup>2</sup>, Han Gao<sup>1</sup>, Ji Ma<sup>1</sup>, Hangxin Liu<sup>2</sup>, and Chang Liu<sup>1</sup>

**Abstract**—This paper proposes an informative trajectory planning approach, namely, *adaptive particle filter tree with sigma point-based mutual information reward approximation* (ASPIRe), for mobile target search and tracking (SAT) in cluttered environments with limited sensing field of view. We develop a novel sigma point-based approximation to accurately estimate mutual information (MI) for general, non-Gaussian distributions utilizing particle representation of the belief state, while simultaneously maintaining high computational efficiency. Building upon the MI approximation, we develop the Adaptive Particle Filter Tree (APFT) approach with MI as the reward, which features belief state tree nodes for informative trajectory planning in continuous state and measurement spaces. An adaptive criterion is proposed in APFT to adjust the planning horizon based on the expected information gain. Simulations and physical experiments demonstrate that ASPIRe achieves real-time computation and outperforms benchmark methods in terms of both search efficiency and estimation accuracy.

## I. INTRODUCTION

Target search and tracking (SAT) using autonomous robots play significant roles in various military and civilian applications such as surveillance [1], disaster response [2], and environment exploration [3]. In these applications, the robot first needs to explore the environment and search for the target. Once the target is detected, the robot enters the tracking stage to maintain the target inside the field of view (FOV). Mainstream strategies formulate SAT as an information-gathering problem, where the robot trajectory is optimized to obtain informative measurements to reduce the expected target state uncertainty [4, 5]. Solving this problem mainly encompasses target state estimation to handle inherent sensing uncertainty and motion planning to generate informative trajectories for more precise target localization.

State estimation is a crucial part in SAT, and the filtering approaches for state estimation in SAT primarily include the grid-based Bayesian filters [6], the Kalman filter variants [7] and the particle filter [8]. Since the particle filter can form non-parametric representations for arbitrary probability distributions, the particle filter has shown superior performance in many SAT tasks when compared to its competitors [4, 9].

Several objective functions have been utilized for robot trajectory planning to facilitate SAT. While early works used the probability of detection as the objective function [10–12], recent studies have shown that information-theoretic objectives, especially the mutual information (MI) [13, 14],

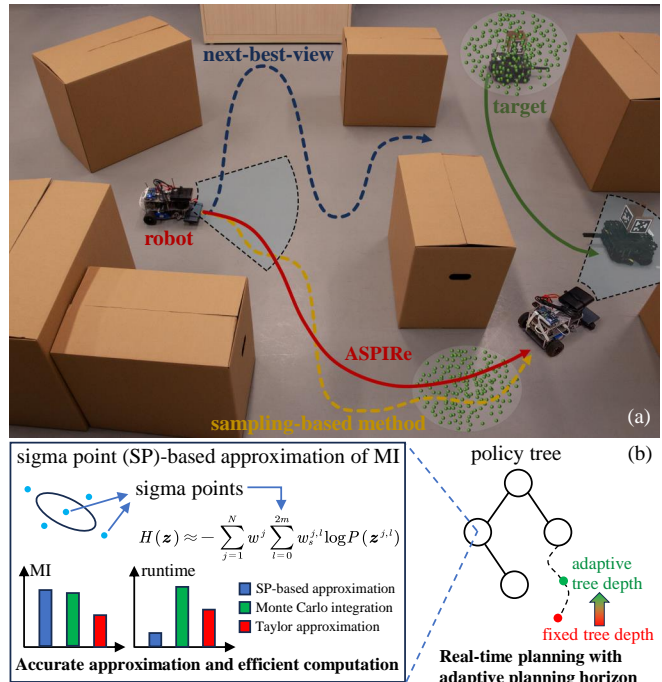


Fig. 1: (a) Comparison of different planning methods in mobile target SAT with prior uncertainty (green particles show the initial target state distribution). Suffering from the myopic horizon, the next-best-view method tends to randomly explore the environment. Although the sampling-based method can make long-term planning, ASPIRe generates a smoother and shorter trajectory. (b) ASPIRe combines SP-based MI approximation with an adaptive planning horizon instead of a fixed one in the policy tree, enabling the maintenance of abundant particles for precise distribution representation, accurate reward approximation, and efficient planning.

demonstrate superior performance in encouraging robots to proactively gather target information. However, calculating MI for non-Gaussian belief states in SAT, typically caused by nonlinear sensor models and target dynamics, involves integration over the continuous state and measurement spaces, lacking a general analytical expression. To address this computational challenge, the particle filter has been employed to sample the continuous state space, facilitating MI computation [9, 15]. However, integrating over the continuous measurement space remains a significant hurdle. To mitigate this issue, the approximation of MI based on Taylor expansion has been combined with the particle filter to approximate integration in the continuous measurement space [4]. Nevertheless, the approximation accuracy is sacrificed for the purpose of computational efficiency, leading to non-trivial approximation error.

Various planning methods have been utilized for information gathering, employing simplification techniques to

\*This work was sponsored by Beijing Nova Program (20220484056) and the National Natural Science Foundation of China (62203018).

All correspondences should be sent to Chang Liu.

<sup>1</sup>Department of Advanced Manufacturing and Robotics, College of Engineering, Peking University.

<sup>2</sup>National Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI).

make the computation of information-theoretic objectives tractable. The greedy policy that chooses the next-best-view (NBV) has been widely adopted for its low computation complexity [11, 15], but such a strategy always falls short in complex environments due to its myopic nature. In response to this challenge, Jadidi et al. [16] proposed sampling-based methods for non-myopic informative path planning, which enables online replanning by incorporating an automatic stopping criterion. Nonetheless, the discrete measurement space is used to simplify MI computation, which inevitably introduces discretization errors. Liu et al. [17] used model predictive control to generate informative trajectories for SAT in continuous spaces, yet relied on a restrictive Gaussianity assumption of belief state for computability, which severely limits the generality of the method.

In recent years, Monte Carlo Tree Search (MCTS) has gained popularity as an online approach for non-myopic planning due to its ability to allocate computation resources to more valuable subtrees to prevent exhaustive search and has been utilized in trajectory planning for SAT tasks [18–20]. In order to handle non-Gaussian belief states, MCTS combines the particle filter for belief inference, and uses the progressive widening strategy to allow for long planning horizon in continuous spaces, as exemplified by PFT-DPW [21] and IPFT [22]. However, due to computationally expensive belief inference, these methods usually employ a limited number of particles, which compromises estimation accuracy and cannot meet the needs of many SAT tasks, especially under multimodal prior uncertainty.

In this work, we propose the adaptive particle filter tree with sigma point-based mutual information reward approximation (ASPIRe) to generate non-myopic informative trajectories for mobile target SAT under continuous state and measurement spaces in cluttered environments (Fig. 1).

The main contributions can be summarized as follows:

- We propose a novel sigma point (SP)-based approximation approach to compute the predictive MI under continuous state and measurement spaces for non-parametric belief states, while taking the limited sensing FOV into account. The approximation is more accurate and computationally efficient than state of the arts.
- We develop an adaptive particle filter tree (APFT) approach to generate kinematically feasible, informative trajectories. An adaptive criterion is proposed for automatic termination in tree construction to improve the search efficiency.
- We combine APFT with SP-based approximation to obtain ASPIRe that enables online replanning and accurate target localization and tracking with abundant particles. Simulations and physical experiments demonstrate that ASPIRe achieves superior real-time computational capability, search efficiency, and estimation accuracy.

## II. PROBLEM FORMULATION

### A. System Models

Consider a discrete-time kinematic model for the robot,

$$\mathbf{x}_{k+1}^r = \mathbf{f}^r(\mathbf{x}_k^r, \mathbf{u}_k^r), \quad (1)$$

where  $\mathbf{f}^r$  is the kinematic model,  $\mathbf{x}_k^r$  and  $\mathbf{u}_k^r$  denote the robot state and control at time step  $k$ , respectively, and the

superscript  $r$  represents the robot. The target state and control are denoted as  $\mathbf{x}_k^t$  and  $\mathbf{u}_k^t$ , where the superscript  $t$  represents the target. The target motion model is defined as

$$\mathbf{x}_{k+1}^t = \mathbf{f}^t(\mathbf{x}_k^t, \mathbf{u}_k^t) + \boldsymbol{\eta}_k, \boldsymbol{\eta}_k \sim \mathcal{N}(0, \mathbf{Q}), \quad (2)$$

where  $\mathbf{f}^t$  represents the kinematic model. Here  $\boldsymbol{\eta}_k$  is a Gaussian noise with zero mean and covariance matrix  $\mathbf{Q}$ .

Due to the limited sensing domain and obstacle occlusion, when the target is outside the FOV, the robot cannot detect the target and no measurement can be obtained. To reflect the intermittency of sensor measurements, a binary parameter  $\gamma_k$  is defined to indicate if the target is inside the FOV ( $\gamma_k = 1$ ) or not ( $\gamma_k = 0$ ), and the measurement model is [23]:

$$\mathbf{z}_k = \begin{cases} \mathbf{h}(\mathbf{x}_k^r, \mathbf{x}_k^t) + \boldsymbol{\varepsilon}_k, \boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}) & \gamma_k = 1 \\ \emptyset & \gamma_k = 0 \end{cases}, \quad (3)$$

where  $\mathbf{z}_k \in \mathbb{R}^m \cup \emptyset$  is the sensor measurement,  $\mathbf{h}$  is the observation function, and  $\boldsymbol{\varepsilon}_k$  is a zero-mean Gaussian white noise with covariance matrix  $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times m}$ .

### B. Belief MDP Formulation with Particle Filter

We formulate the SAT problem as a finite-horizon belief Markov decision process (MDP)  $(h, \mathcal{B}, \mathcal{A}, \tau, \mathcal{R}, \gamma)$ , with belief state space  $\mathcal{B}$ , action space  $\mathcal{A}$ , belief transition model  $\tau$ , planning horizon  $h$ , discount factor  $\gamma$ , and reward function  $\mathcal{R}$ , which will be detailed in Section III. The robot state is assumed to be fully known, and the belief state is defined as  $\mathbf{B}_k = [\mathbf{x}_k^r, P(\mathbf{x}_k^t)] \in \mathcal{B}$ , where  $P(\mathbf{x}_k^t)$  represents the probability distribution of the target state, denoted as the target belief state. To handle potential nonlinearity in target dynamics and sensor models, especially due to the limited sensing domain, we use the particle filter to estimate the target belief because of its capability to represent arbitrary probability distributions. Specifically, the target belief can be approximated by weighted particles as  $P(\mathbf{x}_k^t) \approx \sum_{j=1}^N w_k^j \delta(\mathbf{x}_k^t - \tilde{\mathbf{x}}_k^{t,j})$ , where  $\tilde{\mathbf{x}}_k^{t,j}$  is the  $j$ th particle,  $w_k^j$  is the corresponding weight,  $N$  is the number of particles, and  $\delta(\cdot)$  is a Dirac function. The action  $\mathbf{a}_k = [\mathbf{u}_k^r, \mathbf{u}_k^t] \in \mathcal{A}$  encompasses the control inputs of the robot and the target, respectively. The belief transition model  $\tau$  is defined as  $\mathbf{B}_{k+1} = \tau(\mathbf{B}_k, \mathbf{a}_k, \mathbf{z}_k) = [\mathbf{x}_{k+1}^r, P(\mathbf{x}_{k+1}^t)]$ , where  $\mathbf{x}_{k+1}^r$  is propagated based on Eq. (1), and  $P(\mathbf{x}_{k+1}^t)$  is updated using particle filtering with the following prediction step (Eq. (4)) and update step (Eq. (5)),

$$\tilde{\mathbf{x}}_{k+1}^{t,j} \sim \mathcal{N}(\mathbf{f}^t(\tilde{\mathbf{x}}_k^{t,j}, \mathbf{u}_k^t), \mathbf{Q}), \quad j = 1, \dots, N, \quad (4)$$

$$w_{k+1}^j = \frac{P(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j})}{\sum_{j=1}^N P(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j})} w_k^j, \quad j = 1, \dots, N, \quad (5)$$

where the particles' states are first forward predicted based on the target dynamics, then the weights are updated with new measurements. To alleviate particle degeneracy, a resampling procedure is performed subsequently. We use low variance resampling strategy to mitigate sampling error [24].

We aim to obtain the optimal policy  $\boldsymbol{\pi}^* = (\pi_1^*, \dots, \pi_h^*)$  that maximizes the expected total discounted reward,

$$\boldsymbol{\pi}^* = \arg \max_{\boldsymbol{\pi}} \mathbb{E} \left[ \sum_{t=k}^{k+h-1} \gamma^{t-k} \mathcal{R}(\mathbf{B}_t, \mathbf{a}_t) \middle| \mathbf{a}_t = \pi_{t-k+1}(\mathbf{B}_t) \right],$$

where  $\mathbb{E}$  is the expectation over future beliefs. The robot then

executes the optimal action  $\mathbf{a}_k^* = \pi_1^*(\mathbf{B}_k)$  and replans at the next time step based on the new measurements.

### III. SIGMA POINT-BASED MUTUAL INFORMATION APPROXIMATION

#### A. Reward Function Definition

To gather more target information from future observations, we define the reward as the MI between the target belief and predicted measurements,

$$\mathcal{R}(\mathbf{B}_k, \mathbf{a}_k) = I(\mathbf{x}_{k+1}^t; \mathbf{z}_{k+1}) = H(\mathbf{z}_{k+1}) - H(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^t), \quad (6)$$

where  $I$  and  $H$  denote the MI and the entropy, respectively.

Using particle filter, future belief can be approximated as

$$P(\mathbf{x}_{k+1}^t) \approx \sum_{j=1}^N w_k^j \delta(\mathbf{x}_{k+1}^t - \tilde{\mathbf{x}}_{k+1}^{t,j}), \quad (7)$$

where the particle state is predicted based on the target dynamics, while the weight remains unchanged since the future measurement is unknown.

First, we calculate the conditional measurement entropy in Eq. (6) with the particle expression,

$$\begin{aligned} H(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^t) &\approx \sum_{j=1}^N P(\tilde{\mathbf{x}}_{k+1}^{t,j}) H(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^t = \tilde{\mathbf{x}}_{k+1}^{t,j}) \\ &\approx \sum_{j=1}^N w_k^j H(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j}), \end{aligned}$$

where the first equality is derived from the definition of conditional entropy and the second equality is obtained by substituting  $P(\tilde{\mathbf{x}}_{k+1}^{t,j})$  with Eq. (7). According to the measurement model, the likelihood is

$$P(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j}) = \begin{cases} \mathcal{N}(\mathbf{z}_{k+1}; \mathbf{h}(\mathbf{x}_{k+1}^r, \tilde{\mathbf{x}}_{k+1}^{t,j}), \Sigma) & \gamma_{k+1}^j = 1 \\ \mathbf{1}_{\mathbf{z}_{k+1} = \emptyset} & \gamma_{k+1}^j = 0 \end{cases},$$

where  $\gamma_{k+1}^j$  denotes whether the  $j$ th particle is inside the FOV. If the  $j$ th particle can be detected, i.e.,  $\gamma_{k+1}^j = 1$ , the likelihood is an  $m$ -dimensional Gaussian distribution and its entropy has explicit expression  $H_0 = \frac{m}{2}(\log 2\pi + 1) + \frac{1}{2} \log |\Sigma|$ . Otherwise, no observation is expected to be obtained, and the corresponding entropy is zero. So we can derive that

$$H(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j}) = \mathbf{1}_{\gamma_{k+1}^j = 1} H_0. \quad (8)$$

Next, we consider computing entropy  $H(\mathbf{z}_{k+1})$ . Utilizing the particle representation of target state distribution, the measurement distribution  $P(\mathbf{z}_{k+1})$  can be computed as

$$\begin{aligned} P(\mathbf{z}_{k+1}) &= \int P(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^t) P(\mathbf{x}_{k+1}^t) d\mathbf{x}_{k+1}^t \\ &\approx \sum_{j=1}^N w_k^j P(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j}). \end{aligned} \quad (9)$$

When the  $j$ -th particle is in the FOV,  $P(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j})$  is a Gaussian distribution by the definition of the measurement model, and  $P(\mathbf{z}_{k+1})$  follows a Gaussian Mixture Model (GMM), whose entropy has no closed form. Though the entropy of GMM can be numerically evaluated using Monte Carlo integration, the amount of samples required for accurate approximation usually results in large computational overhead and therefore limits this method in practice. The Taylor expansion has been used to approximate the entropy of GMM [25], yet this method usually leads to a non-trivial approximation error. To overcome the undesirable tradeoff

between the computational efficiency and the approximation accuracy in existing approaches, we propose to utilize the sigma points associated with each Gaussian component in GMM to approximate the GMM entropy, which will be detailed in the next subsection.

#### B. Sigma Point-Based Entropy Approximation

The entropy  $H(\mathbf{z}_{k+1})$  can be computed as

$$H(\mathbf{z}_{k+1}) = - \int P(\mathbf{z}_{k+1}) \log P(\mathbf{z}_{k+1}) d\mathbf{z}_{k+1} \approx - \sum_{j=1}^N w_k^j p_j,$$

where  $p_j = \int P(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j}) \log P(\mathbf{z}_{k+1}) d\mathbf{z}_{k+1}$ . Denote  $A$  as the set of indices that particles are inside the FOV, i.e.,  $\forall i \in A, \gamma_{k+1}^i = 1$ . If  $j \notin A$ , then  $p_j$  has an explicit expression,

$$p_j = P(\emptyset | \tilde{\mathbf{x}}_{k+1}^j) \log P(\emptyset) = \log \sum_{i \notin A} w_k^i. \quad (10)$$

If  $j \in A$ , we propose to utilize sigma points from the Unscented Transform [26] to approximate  $p_j$ . Denote the sigma points and their weights corresponding to the  $j$ th Gaussian component as  $[\tilde{\mathbf{z}}_{k+1}^{j,0}, \dots, \tilde{\mathbf{z}}_{k+1}^{j,2m}]^T$  and  $[w_s^{j,0}, \dots, w_s^{j,2m}]^T, \forall j = 1, \dots, N$ , respectively, and they are computed as follows [26],

$$\begin{aligned} \tilde{\mathbf{z}}_{k+1}^{j,0} &= \boldsymbol{\mu}_j, \\ \tilde{\mathbf{z}}_{k+1}^{j,l} &= \boldsymbol{\mu}_j + (\sqrt{(\lambda + m)\Sigma})_l, \quad l = 1, \dots, m \\ \tilde{\mathbf{z}}_{k+1}^{j,l} &= \boldsymbol{\mu}_j - (\sqrt{(\lambda + m)\Sigma})_{l-m}, \quad l = m + 1, \dots, 2m \\ w_s^{j,0} &= \frac{\lambda}{\lambda + m}, \quad w_s^{j,l} = \frac{1}{2(\lambda + m)}, \quad l = 1, \dots, 2m \end{aligned}$$

where  $\lambda$  is a parameter that determines the sigma points spread,  $\boldsymbol{\mu}_j = \mathbf{h}(\mathbf{x}_{k+1}^r, \tilde{\mathbf{x}}_{k+1}^{t,j})$  is the mean of  $P(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j})$ , and  $(\sqrt{(\lambda + m)\Sigma})_l$  is the  $l$ -th column of the matrix square root. Since the observation is  $m$ -d, there are  $2m + 1$  sigma points for each Gaussian component. Thus the  $j$ th Gaussian component can be approximated as

$$P(\mathbf{z}_{k+1} | \tilde{\mathbf{x}}_{k+1}^{t,j}) \approx \sum_{l=0}^{2m} w_s^{j,l} \delta(\mathbf{z}_{k+1} - \tilde{\mathbf{z}}_{k+1}^{j,l}), \quad (11)$$

and  $p_j$  can be approximated as follows,

$$\begin{aligned} p_j &\approx \sum_{l=0}^{2m} w_s^{j,l} \log P(\tilde{\mathbf{z}}_{k+1}^{j,l}) \\ &\approx \sum_{l=0}^{2m} w_s^{j,l} \log \sum_{i \in A} w_k^i P(\tilde{\mathbf{z}}_{k+1}^{j,l} | \tilde{\mathbf{x}}_{k+1}^{t,i}). \end{aligned} \quad (12)$$

By employing this approach, with Eq. (10) and Eq. (12), we obtain an explicit expression to approximate the entropy  $H(\mathbf{z}_{k+1})$ .

### IV. ADAPTIVE PARTICLE FILTER TREE FOR PLANNING

We propose the APFT, a MCTS-based planning algorithm to generate informative trajectories for SAT tasks. By exploring possible sequences of actions and observations in the planning horizon, the proposed method constructs an asymmetric policy tree to direct the robot's actions.

#### A. Algorithm Overview

The entire procedure of APFT is shown in Alg. 1. The algorithm proceeds with the current belief state  $\mathbf{B}_k$  as input and the optimal action  $\mathbf{a}^*$  as output. The algorithm iterates through simulating action-observation sequences to

---

**Algorithm 1 Adaptive Particle Filter Tree**

---

```
1: function APFT( $\mathcal{B}_k$ )
2:  $n_r = \langle \emptyset, \emptyset, 0, 0 \rangle$ 
3: for  $i \in 1 : n$  do
4:   SIMULATE( $\mathcal{B}_k, n_r, h$ )
5: end for
6:  $n^* = \arg \max_{n \in C(n_r)} \text{UCB}(n)$ 
7:  $\mathbf{a}^* = \mathcal{H}(n^*)$ 
8: return  $\mathbf{a}^*$ 
9: function SIMULATE( $\mathcal{B}, n, d$ )
10: if  $d = 0$  then
11:    $R = 0$ 
12: else
13:    $n_a, \mathbf{a} \leftarrow \text{SELECTACTION}(n)$ 
14:    $r \leftarrow \mathcal{R}(\mathcal{B}, \mathbf{a})$ 
15:    $\mathcal{B} \leftarrow \tau(\mathcal{B}, \mathbf{a}, \emptyset)$ 
16:   if  $|C(n_a)| \leq k_o W(n_a)^{\alpha_o}$  then
17:      $\mathbf{z} \leftarrow \text{SAMPLENEWOBSERVATION}(\mathcal{B})$ 
18:     ADDNODE( $\mathbf{z}$ )
19:      $\mathcal{B} \leftarrow \tau(\mathcal{B}, \emptyset, \mathbf{z})$ 
20:      $R \leftarrow r + \gamma \cdot \text{ROLLOUT}(\mathcal{B}, d - 1)$ 
21:   else
22:      $n_o, \mathbf{z} \leftarrow \text{SELECTOBSERVATION}(n_a)$ 
23:      $\mathcal{B} \leftarrow \tau(\mathcal{B}, \emptyset, \mathbf{z})$ 
24:      $R \leftarrow r + \gamma \cdot \text{SIMULATE}(\mathcal{B}, n_o, d - 1)$ 
25:   end if
26:    $W(n_c) \leftarrow W(n_c) + 1$ 
27:    $W(n_a) \leftarrow W(n_a) + 1$ 
28:    $Q(n_a) \leftarrow Q(n_a) + \frac{R - Q(n_a)}{W(n_a)}$ 
29: end if
30: return  $R$ 
```

---

expand action nodes  $n_a$  and observation nodes  $n_o$  in the rewarding subtree, based on upper confidence bound (UCB) criterion [27]. Each node  $n = \langle \mathcal{H}, C, W, Q \rangle$  includes action-observation history  $\mathcal{H}$ , children set  $C$ , visit count  $W$ , and estimated value  $Q$ . The planning horizon and remaining tree depth are denoted by  $h$  and  $d$ , respectively.

First, the root node  $n_r$  is initialized (Line 2), and SIMULATE is called repeatedly to construct the policy tree. Specifically, In SIMULATE, an action node is first selected by the UCB criterion (Line 13) from the robot action space, which consists of motion primitives generated by the robot kinematics to allow performing smooth trajectories while avoiding obstacles. Then the information reward of taking action  $\mathbf{a}$  in belief  $\mathcal{B}$  is calculated, and the belief state is updated (Line 14-15). Given the selected action node, an observation node is obtained by either sampling a new observation from the updated belief or selecting an existing observation node depending on the parameters  $k_o$  and  $\alpha_o$ . If a new observation is generated, it is inserted into the policy tree as a new child node, and ROLLOUT is subsequently performed to estimate the accumulated reward  $R$  of the new node (Line 17-20), which will be detailed in the next subsection. If, on the other hand, a previous observation node is chosen, SIMULATE is called recursively on the new belief and observation node (Line 22-24). At last, the information of visited nodes is updated (Line 26-28). After performing the desired number of iterations, the optimal action is chosen from the children of the root node that maximizes the UCB.

### B. Adaptive Criterion for Efficient Tree Search

The standard tree search method typically employs a fixed planning horizon, which may be inefficient for target search tasks. Concretely, if the fixed planning horizon is short, the limited search space makes it difficult to find a remote target,

---

**Algorithm 2 Adaptive Rollout**

---

```
1: function ROLLOUT( $\mathcal{B}, d$ )
2: if  $d = 0$  then
3:    $R = 0$ 
4: else
5:    $\mathbf{a} \leftarrow \text{DefaultPolicy}()$ 
6:    $r \leftarrow \mathcal{R}(\mathcal{B}, \mathbf{a})$ 
7:    $\mathcal{B} \leftarrow \tau(\mathcal{B}, \mathbf{a}, \emptyset)$ 
8:   if  $r > \delta_r$  then
9:     return  $r$ 
10:  else
11:     $R \leftarrow r + \gamma \cdot \text{ROLLOUT}(\mathcal{B}, d - 1)$ 
12:  end if
13: end if
14: return  $R$ 
```

---

while setting a lengthy horizon will result in computational redundancy when the robot is close to the target.

To alleviate this limitation, we provide an adaptive termination criterion to dynamically adjust the length of the planning horizon to reach a desirable trade-off between efficiency and effectiveness. As shown in Alg. 2, when ROLLOUT progresses recursively, if the reward of one iteration in rollout exceeds the threshold  $\delta_r$  (Line 8), ROLLOUT will terminate in advance rather than reach the defined maximum depth. This is because the increasing information gain demonstrates that the robot will find the target in future rollout steps, which is informative for target search, and thus the planning horizon can be decreased to reduce computation time. Benefitting from the computational efficiency achieved through the adaptive termination criterion and MI approximation method, we can adopt a large set of particles in the tree search to ensure accurate belief representation, while maintaining the ability to perform online operations.

## V. SIMULATION

We conduct simulations to validate the proposed method in MATLAB using a desktop with Intel Core i7 CPU@2.10GHz and 16GB RAM. The environment is set as a  $50m \times 50m$  planar space comprising multiple obstacles, and the map is known to the robot. We consider the robot state  $\mathbf{x}_k^r = [x_k^r, y_k^r, \theta_k^r]^T \in \mathbb{R}^3$  includes the  $x$ - $y$  position  $x_k^r, y_k^r$  and orientation  $\theta_k^r$  of the robot, and the control input  $\mathbf{u}_k^r \in \mathbb{R}^2$  consists of the linear velocity  $v_k^r$  and angular velocity  $w_k^r$ . The robot motion models use the following unicycle model,

$$\mathbf{f}^r(\mathbf{x}_k^r, \mathbf{u}_k^r) = \mathbf{x}_k^r + [v_k^r \cos \theta_k^r, v_k^r \sin \theta_k^r, w_k^r]^T \cdot \Delta t, \quad (13)$$

where  $\Delta t$  is the sampling interval. The target adopts the similar state, control input, and unicycle model as the robot, and its control is assumed to be known for the robot in simulations. For the sensing module, we adopt a range-bearing sensor with fan-shaped FOV, whose sensing range is from  $1m$  to  $6m$  and sensing angle is  $90^\circ$ . To demonstrate the generalization capability, we test the proposed method in 50 scenarios where target trajectories are randomly generated.

### A. Approximation Accuracy and Efficiency

To investigate the performance of SP-based approximation, we manually control the robot to follow the target, ensuring the target remains in the FOV, and compare MI approximated by different methods, including Monte Carlo integration, Taylor approximation [25] and SP-based approximation, referred as *SP* for simplicity. We also combine SP

TABLE I: Comparison of MI approximation methods

Methods	$\varepsilon_a$	$\varepsilon_r$ (%)	$\tau$ (s)
Monte Carlo integration	-	-	0.0614
0th Taylor Approximation	0.4220	38.9	0.0070
2nd Taylor Approximation	0.0993	8.66	0.3502
SP	<b>0.0395</b>	<b>3.42</b>	0.0302
SP with particle simplification	0.0533	4.69	<b>0.0017</b>

with the particle simplification method [4], referred to as *SP with particle simplification*, which partitions the state space and generates a simplified particle set by replacing particles in the same cell with their weighted average to improve computational efficiency.

The covariance matrix in Eq. (2) and Eq. (3) are  $\mathbf{Q} = \text{diag}(0.5, 0.5, 0.1)$  and  $\Sigma = \text{diag}(0.5, 0.05)$ , respectively. Define the MI computed by Monte Carlo integration as  $\mathcal{I}$ , which is treated as the ground-truth value, and the MI computed by other methods as  $\mathcal{I}_m$ . We record the absolute error  $\varepsilon_a$ , defined as  $\varepsilon_a = |\mathcal{I} - \mathcal{I}_m|$ , the relative error  $\varepsilon_r$ , defined as  $\varepsilon_r = \frac{|\mathcal{I} - \mathcal{I}_m|}{\mathcal{I}}$ , and the computational time  $\tau$ .

Table I shows the average results over 50 scenarios. It indicates that SP obtains the minimal approximation error among all methods. Besides, SP with particle simplification achieves the smallest computational time and is more accurate than 2nd Taylor approximation approach, indicating desirable advantages in both computational efficiency and approximation accuracy over the state of the arts.

### B. Benchmark Comparisons

We compare ASPIRe with other benchmark methods to assess its performance in mobile target search and tracking. Concretely, we consider the next-best-view (NBV) strategy and IIG-tree approach [16], a sampling-based information gathering algorithm, as baselines to make comparisons. All methods utilize the same motion model and control constraints, and use SP-based MI approximation with particle simplification technique [4] as the objective function. Since the IIG-tree was proposed to solve the exploration problem originally, to adapt it to the tracking task, we adjust its sample policy from sampling the whole workspace to sampling the area nearby the robot once the target has been detected.

Simulations include the unimodal and multimodal case. The unimodal case is initialized with a Gaussian prior distribution whose mean is the target true pose, while the multimodal case considers a GMM with a Gaussian component initialized as above and two additional Gaussian distributions as distraction. The mean of disruptive Gaussian components are randomized in the obstacle-free space, and the covariance matrix for all components is  $\mathbf{V} = \text{diag}(3, 3, 0.01)$ . The weight assigned to the component with the target pose as mean is 0.2, while the disruptive components each have a weight of 0.4. The covariance matrix are  $\mathbf{Q} = \text{diag}(0.5, 0.5, 0.1)$  and  $\Sigma = \text{diag}(0.1, 0.01)$ . Three metrics are evaluated for quantitative comparisons: the search time  $t_s$ , target loss rate  $r_{los}$ , and estimation error  $\varepsilon_{est}$ , defined as  $\varepsilon_{est} = \frac{1}{T_{tra}} \sum_{k=1}^{T_{tra}} \|\mathbf{x}_k^t - \hat{\mathbf{x}}_k^t\|$ , where  $T_{tra}$  is the total tracking time and  $\hat{\mathbf{x}}_k^t$  is the average of the particles' positions. For each scenario, we repeat 5 trials and average the results, which are shown in Fig. 2. Fig. 3 show the qualitative

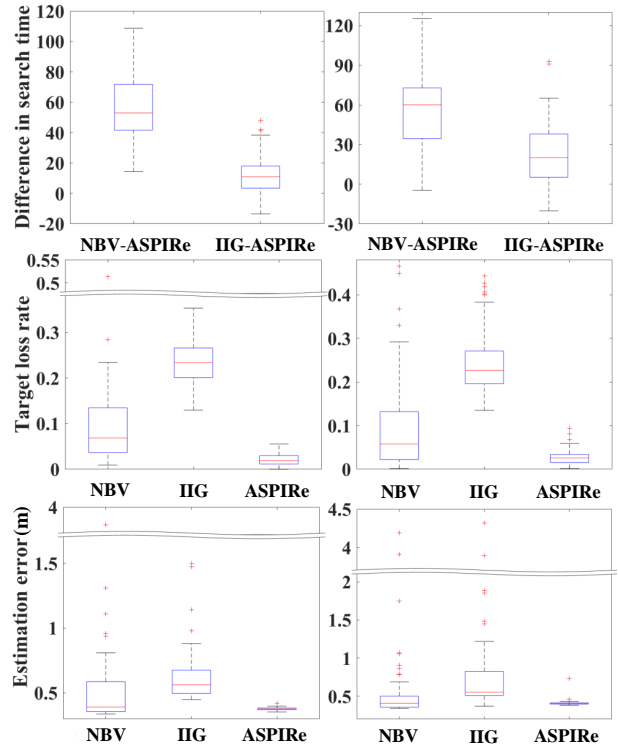


Fig. 2: Quantitative comparisons in the unimodal (left column) and multimodal case (right column). NBV-ASPIRe and IIG-ASPIRe in the first row represent the difference in search time using the NBV strategy and IIG-tree, respectively, compared to ASPIRe.

comparisons in the unimodal and multimodal case.

1) *Unimodal Case:* The left column of Fig. 2 shows the comparison results in the unimodal case. Suffering from the myopic horizon, the NBV policy initially cannot gain information reward and can only randomly act to search for the target, which results in poor performance in target search. IIG-tree can avoid obstacles and find the target faster than the NBV policy. However, due to the random nature of samples, IIG-tree usually generates sinuous trajectories and loses sight of the target, leading to the highest average loss rate and estimation error. ASPIRe significantly outperforms other methods by a large margin with less search time, considerably lower loss rate and localization error. In 41 out of 50 scenarios, ASPIRe spends the least time to find the target. Compared to the NBV policy and IIG-tree, ASPIRe achieves 3 times and 10 times improvements in the target loss rate and yields 50% and 80% improvements in the estimation error, respectively. The quantitative comparisons show the effectiveness of ASPIRe in the unimodal case.

2) *Multimodal Case:* As depicted in the right column of Fig. 2, consistent with the unimodal case, ASPIRe maintains its advantages in search efficiency and tracking stability compared to other methods. The times of ASPIRe spends fewer simulation steps to find the target than the NBV policy and IIG-tree are 50 and 40 out of 50 scenarios, and the proposed approach outperforms the benchmark methods by reducing the target loss rate and the estimation error at least 50% and 60%, respectively. The comparisons demonstrate the effectiveness and robustness of ASPIRe with unreliable prior information.

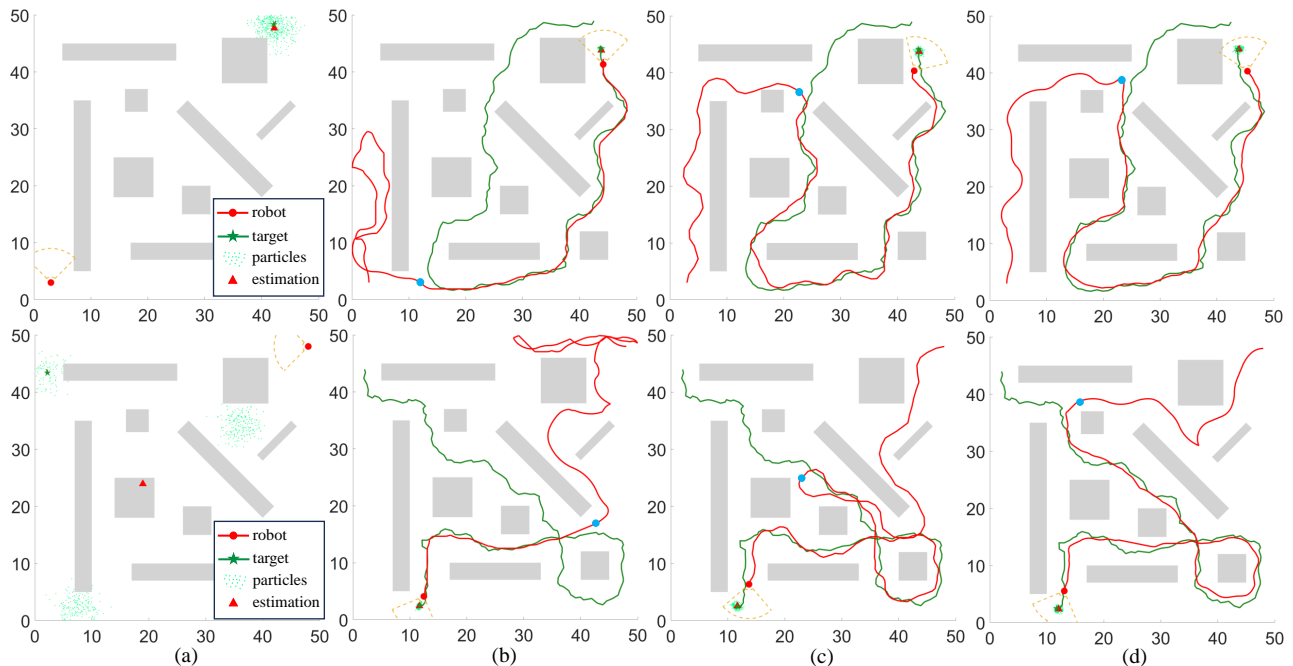


Fig. 3: **Qualitative comparisons in the unimodal case (top row) and multimodal case (bottom row).** (a) Initialization. (b) The NBV policy. (c) IIG-tree. (d) ASPIRe. The blue circle represents the moment when the target is detected. ASPIRe shows shorter search time and more stable tracking performance with smoother trajectories, even under distracting prior information.

## VI. EXPERIMENT

We investigate the performance of ASPIRe in real-world scenarios. The environment is a  $6m \times 4m$  planar space and contains several obstacles. We use a Wheeltec ground robot to search for and track the moving target, which is a Turtlebot3 that carries three Apriltags [28] for camera detection. Since the target control inputs are unavailable to the robot in real-world scenarios, we assume the target motion model follows an autonomous Markov model,

$$\mathbf{x}_{k+1}^t = \mathbf{f}^t(\mathbf{x}_k^t) + \eta_k, \eta_k \sim \mathcal{N}(0, \mathbf{Q}), \quad (14)$$

and the robot has access to the model information. The motion noise is set as  $\mathbf{Q} = \text{diag}(0.05, 0.05, 0.01)$ . We adopt a Vicon motion-capture system to measure the poses of the robot and target, which are treated as the ground-truth.

We conducted three experiments in the multimodal case in the same environment. The first two scenarios have two Gaussian components as the initial belief, and the last

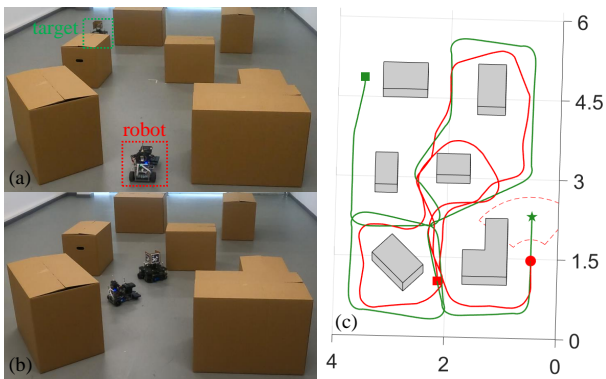


Fig. 4: **Indoor experiments with inaccurate prior information.** The red circle and green star show the current positions of the robot and the target, and the squares represent their starting positions.

TABLE II: **Performance of ASPIRe in real-world experiments**

	$r_{vis}$ (%)	$\varepsilon_{est}$ (m)	$t_{run}$ (s/step)
Scenario 1	96.44	0.0770	0.0920
Scenario 2	95.11	0.0843	0.0923
Scenario 3	90.20	0.0927	0.0932

scenario has three. For each scenario, we record the visibility rate  $r_{vis}$  that denotes the percentage of time the target is in FOV once detected, the estimation error  $\varepsilon_{est}$  as defined in Section V-B, and the computational time  $t_{run}$ . The robot successfully accomplishes the SAT task in all scenarios, and Fig. 4 illustrates the trajectory performed by the robot in one scenario. As shown in Table II, the proposed method can achieve a considerably high visibility rate and low estimation error while maintaining 10.8Hz computing speed for real-time operation, which demonstrates the effectiveness of ASPIRe.

## VII. CONCLUSION

This work presents ASPIRe, an informative trajectory planning approach for mobile target SAT in cluttered environments with limited sensing FOV. A novel sigma point-based approximation is proposed to accurately and efficiently compute mutual information in continuous measurement spaces. APFT is also developed to generate informative trajectories while simultaneously achieving accurate target state estimation and efficient planning. Simulation results demonstrate the superiority of ASPIRe compared to benchmark methods in terms of MI approximation, search efficiency and estimation accuracy. We also demonstrate the robustness and real-time performance of ASPIRe in real-world scenarios.

**Acknowledgement:** We thank Dr. Meng Wang at BI-GAI for his help with experiment photography.

## REFERENCES

- [1] E. Lozano, U. Ruiz, I. Becerra, and R. Murrieta-Cid, "Surveillance and collision-free tracking of an aggressive evader with an actuated sensor pursuer," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 3, pp. 6854–6861, 2022.
- [2] M. Aggravi, A. A. S. Elsherif, P. R. Giordano, and C. Pacchierotti, "Haptic-enabled decentralized control of a heterogeneous human-robot team for search and rescue in partially-known environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 3, pp. 4843–4850, 2021.
- [3] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 2, pp. 610–617, 2019.
- [4] B. Charrow, V. Kumar, and N. Michael, "Approximate representations for multi-robot control policies that maximize mutual information," *Autonomous Robots*, vol. 37, pp. 383–400, 2014.
- [5] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2009.
- [6] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte, "Recursive bayesian search-and-tracking using coordinated uavs for lost targets," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, IEEE, 2006.
- [7] T. H. Chung, J. W. Burdick, and R. M. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, IEEE, 2006.
- [8] J. Tisdale, A. Ryan, Z. Kim, D. Tornqvist, and J. K. Hedrick, "A multiple uav system for vision-based search and localization," in *American Control Conference (ACC)*, IEEE, 2008.
- [9] A. Ryan and J. K. Hedrick, "Particle filter based information-theoretic active sensing," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 574–584, 2010.
- [10] G. Hollinger, S. Singh, J. Djughash, and A. Kehagias, "Efficient multi-robot search for a moving target," *International Journal of Robotics Research (IJRR)*, vol. 28, no. 2, pp. 201–219, 2009.
- [11] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Optimal search for a lost target in a bayesian world," *Field and Service Robotics: Recent Advances in Reserch and Applications*, pp. 209–222, 2006.
- [12] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous uav path planning and estimation," *IEEE Robotics & Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.
- [13] A. Asgharivaskasi, S. Koga, and N. Atanasov, "Active mapping via gradient ascent optimization of shannon mutual information over continuous se (3) trajectories," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022.
- [14] P. Yang, Y. Liu, S. Koga, A. Asgharivaskasi, and N. Atanasov, "Learning continuous control policies for information-theoretic active perception," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2098–2104, IEEE, 2023.
- [15] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *International Journal of Robotics Research (IJRR)*, vol. 31, no. 10, pp. 1134–1154, 2012.
- [16] M. Ghaffari Jadidi, J. Valls Miro, and G. Dissanayake, "Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring," *International Journal of Robotics Research (IJRR)*, vol. 38, no. 6, pp. 658–685, 2019.
- [17] C. Liu and J. K. Hedrick, "Model predictive control-based target search and tracking using autonomous mobile robot with limited sensing domain," in *American Control Conference (ACC)*, IEEE, 2017.
- [18] F. Vanegas, D. Campbell, M. Eich, and F. Gonzalez, "Uav based target finding and tracking in gps-denied and cluttered environments," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016.
- [19] A. Goldhoorn, A. Garrell, R. Alquézar, and A. Sanfeliu, "Searching and tracking people with cooperative mobile robots," *Autonomous Robots*, vol. 42, no. 4, pp. 739–759, 2018.
- [20] A. Wandzel, Y. Oh, M. Fishman, N. Kumar, L. L. Wong, and S. Tellex, "Multi-object search using object-oriented pomdps," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, IEEE, 2019.
- [21] Z. Sunberg and M. Kochenderfer, "Online algorithms for pomdps with continuous state, action, and observation spaces," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 28, pp. 259–263, 2018.
- [22] J. Fischer and Ö. S. Tas, "Information particle filter tree: An online algorithm for pomdps with belief-based rewards on continuous domains," in *International Conference on Machine Learning*, PMLR, 2020.
- [23] H. Gao, P. Wu, Y. Su, K. Zhou, J. Ma, H. Liu, and C. Liu, "Probabilistic visibility-aware trajectory planning for target tracking in cluttered environments," in *American Control Conference (ACC)*, IEEE, 2024.
- [24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [25] M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. D. Hanebeck, "On entropy approximation for gaussian mixture random vectors," in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, IEEE, 2008.
- [26] R. Van Der Merwe, *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. Oregon Health & Science University, 2004.
- [27] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *European conference on machine learning*, Springer, 2006.
- [28] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, IEEE, 2011.