H-SwarmLoc: Efficient Scheduling for Localization of Heterogeneous MAV Swarm with Deep Reinforcement Learning

Haoyang Wang*
Shenzhen International Graduate School, Tsinghua University
Shenzhen, China
haoyang-22@mails.tsinghua.edu.cn

Xuecheng Chen*
Tsinghua-Berkeley Shenzhen Institute, Tsinghua University
Shenzhen, China
chenxc21@mails.tsinghua.edu.cn

Yuhan Cheng
Shenzhen International Graduate School, Tsinghua University
Shenzhen, China
cyh22@mails.tsinghua.edu.cn

Chenye Wu
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
chenyewu@yeah.net

Fan Dang
Global Innovation Exchange, Tsinghua University
Beijing, China
dangfan@tsinghua.edu.cn

Xinlei Chen†
Shenzhen International Graduate School, Tsinghua University
Peng Cheng Laboratory
Shenzhen, China
chen.xinlei@sz.tsinghua.edu.cn

ABSTRACT

Emergency rescue scenarios are considered to be high-risk scenarios. Using a micro air vehicle (MAV) swarm to explore the environment can provide valuable environmental information. However, due to the absence of localization infrastructure and the limited on-board capabilities, it's challenging for the low-cost MAV swarm to maintain precise localization. In this paper, a collaborative localization system for the low-cost heterogeneous MAV swarm is proposed. This system takes full advantage of advanced MAV to effectively achieve accurate localization of the heterogeneous MAV swarm through collaboration. Subsequently, *H-SwarmLoc*, a reinforcement learning-based planning method is proposed to plan the advanced MAV with a non-myopic objective in real-time. The experimental results show that the localization performance of our method improves 40% on average compared with baselines.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; • **Computing methodologies → Planning under uncertainty**.

KEYWORDS

Heterogeneous MAV swarm, Localization, Reinforcement Learning

ACM Reference Format:
Haoyang Wang, Xuecheng Chen, Yuhan Cheng, Chenye Wu, Fan Dang, and Xinlei Chen. 2022. H-SwarmLoc: Efficient Scheduling for Localization of Heterogeneous MAV Swarm with Deep Reinforcement Learning. In *The 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3560905.3568432

1 INTRODUCTION

Emergency response scenarios, including earthquakes, gas leaks, and nuclear radiation, are considered hazardous due to the complexity of the operating environment. Rescuers have little prior information about the environment, which may cause inefficient subsequent rescue operations [1, 2].

Using MAV swarm consisting of miniature aerial sensors to navigate and explore in such dangerous environments autonomously provides valuable fine-grained real-time environmental information. This information allows rescuers to understand the evolution of the emergency scenarios in advance, and to draw up effective strategies to minimize the loss of life and damage to property [3–7].

Accurate localization of the MAV swarm is a crucial step to complete exploration in emergency scenarios [8, 9]. Precisely, the MAV swarm can navigate to a set of goal locations only if they have accurate location information. Subsequently, they are able to collect environmental information [10–12]. The following challenges must be considered to localize the MAV swarm.

- **Limited onboard capabilities:** The above scenarios need a large-scale MAV swarm to explore simultaneously to increase efficiency. Meanwhile, MAVs are highly susceptible to damage in these hazardous scenarios. To reduce overall cost, the MAV swarm usually consists of low-cost and low-complexity nodes with limited onboard sensing, computing, and communication capabilities. Using classic methods such as dead-reckoning, the localization error of MAVs accumulates rapidly over time due to noisy measurements [13].
- **Lack of localization infrastructure:** The localization infrastructure provides external reference signals for the MAV swarm to limit the localization error. However, it is impossible to manually deploy supporting sensors such as beacons in advance in these dangerous scenarios. Even if the localization infrastructure existed, it is highly likely damaged due to the disasters. This indicates that no reference signal is available to correct the localization error [14].

Researchers have proposed methods to maintain localization accuracy for the MAV swarm. Several methods rely on external infrastructures, such as GPS and motion capture devices [15–20]. The

[*] Both authors contributed equally to this research.
[†] Xinlei Chen is the corresponding author.

SenSys '22, November 6–9, 2022, Boston, MA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9886-2/22/11.
https://doi.org/10.1145/3560905.3568432

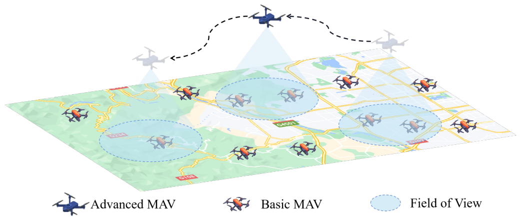

Figure 1: Illustration of the heterogeneous MAV swarm. The AMAV provides localization services to BMAVs.

reliance makes it necessary to deploy a huge amount of supporting sensors in advance [21–23]. Recently, a few methods have been proposed to accurately localize the MAV swarm without relying on any localization infrastructure [24–29]. These methods require advanced sensors such as depth cameras or LiDARs, which are too expensive to be applied to the MAV swarm on a large scale [30–32]. To avoid over-dependence on localization infrastructure and advanced sensors, a few methods are proposed to provide temporary localization infrastructure by landing some MAVs. These methods require a large number of MAVs to establish temporary localization infrastructure, especially for large-scale areas. [13, 33–35].

This paper tries to answer the research question: **How can the low-cost MAV swarm accomplish precise localization in a large space without localization infrastructure?** As Fig.1 illustrates, we propose a collaborative localization system for the heterogeneous MAV swarm consisting of two types of nodes: (1) A large number of *basic MAVs (BMAVs)* with limited sensing, computing, and communication capabilities. The BMAVs operate according to the requirements of tasks. The precise localization of BMAVs is the key to successful mission execution. However, the localization error of BMAVs increases over time when using only noisy measurements with classic methods. (2) A small number of *advanced MAVs (AMAVs)* with premium computing, sensing, and communication capabilities. *The AMAVs can autonomously localize themselves by utilizing premium capabilities. The AMAVs are equipped with downward visual sensors to generate observations to localize BMAVs.*

Since a few of AMAVs generate observations for plenty of BMAVs to limit the localization error, the major challenges in planning AMAVs are two folds. First, the AMAVs at a higher altitude can generate observations for more BMAVs, but higher altitude results in noisier observations. The AMAVs need to balance a trade-off between the number of BMAVs observed and the quality of the observations. Second, the AMAVs can move in any direction at any time, which leads to a considerable action space. It's challenging to find an optimal solution. In addition, the AMAVs need to be planned to limit the localization error of BMAVs for a long term. Therefore, the planning of AMAVs is a receding horizon path-planning problem, which usually implies the high computational cost [36–38].

In this paper, we model the planning of AMAVs as a sequential decision problem and discretize their action space. *H-SwarmLoc*, a deep reinforcement learning-based planning method is proposed then to learn the structural characteristics of BMAVs' estimations and plan the AMAVs for a non-myopic objective. This method maximizes a discounted sum of future rewards to handle infinite planning horizons. Besides, the networks are trained to improve the policy during the offline phase, and effectively plan the movement of AMAVs online at deployment in a short time.

The main contributions of this paper are listed as follows:

- Propose a system to localize the low-cost heterogeneous MAV swarm accurately and effectively by taking full advantage of AMAVs through collaboration.
- Propose *H-SwarmLoc*, a reinforcement learning-based planning method to schedule the AMAV for the non-myopic objective in real time to localize the MAV swarm.
- Evaluate the proposed system and method with physical feature-based experiments on the simulation platform.

The paper is organized as follows: §2 presents the key components and problem definition. §3 introduces our system and *H-SwarmLoc*. §4 illustrates experimental results. Finally, §5 discusses the future research directions, and §6 concludes the paper.

2 PROBLEM DEFINITION

In this session, we first give preliminary definitions of the system, including the state models of the BMAV and the AMAV, the observation model of the AMAV, and the estimation model of the BMAV. Subsequently, we will describe the goal of the *H-SwarmLoc*. Finally, we formulate the problem of planning the AMAV to limit the location estimation error of the heterogeneous MAV swarm.

2.1 Key Definitions

Environment Description: Consider a bounded and obstacles free region $\Omega \in \mathbb{R}^3$ with length L, width W, and height H. A heterogeneous MAV swarm operates in Ω. The swarm consists of several BMAVs and an AMAV. The BMAVs move independently in the Ω with a low height which is constant during the mission. The number of BMAVs is constant and known (i.e., the BMAVs do not join/leave Ω), and the BMAVs can be uniquely identified. The AMAV operates above all BMAVs and generates observations to help localize BMAVs.

State model of BMAV: The heterogeneous MAV swarm contains N BMAVs identified by $\mathbf{B} = \{B_1, B_2, \dots, B_N\}$, N >1. The state of the BMAV B_i at time t is $\boldsymbol{y_i^t} = (x_i^t, y_i^t, \dot{x}_i^t, \dot{y}_i^t)$, where (x_i^t, y_i^t) is the actual location of B_i and $(\dot{x}_i^t, \dot{y}_i^t)$ is the actual velocity of B_i. The BMAVs have limited sensing and computing capabilities and cannot obtain their states. The estimation of BMAV B_i is $\widehat{\boldsymbol{y_i^t}} = (\widehat{x_i^t}, \widehat{y_i^t}, \widehat{\dot{x}_i^t}, \widehat{\dot{y}_i^t})$. $(\widehat{x_i^t}, \widehat{y_i^t})$ is location estimation of B_i and $(\widehat{\dot{x}_i^t}, \widehat{\dot{y}_i^t})$ is velocity estimation of B_i. The covariance matrix of B_i at time t is $\Sigma_{i,t}$. The B_i follows double integrator dynamics with Gaussian noise, and the state of BMAV B_i after movement is

$$\boldsymbol{y_i^{t+1}} = C\boldsymbol{y_i^t} + \gamma_i, \quad \gamma_i \sim \mathcal{N}(0, \Gamma)$$
$$C = \begin{bmatrix} I_2 & \tau I_2 \\ 0 & I_2 \end{bmatrix}, \quad \Gamma = q \begin{bmatrix} \tau^3/3I_2 & \tau^2/2I_2 \\ \tau^2/2I_2 & \tau I_2 \end{bmatrix}, \tag{1}$$

where τ is a sampling period, q is a noise constant factor, and I_2 is an identity matrix. The B_i can acquire noisy velocity measurement $v_{i,t}$ while moving with limited sensing and computing capabilities. We define $\boldsymbol{y^t} = \{\boldsymbol{y_1^t}, \boldsymbol{y_2^t}, \dots, \boldsymbol{y_N^t}\}$, $\Sigma_t = \{\Sigma_{1,t}, \Sigma_{2,t}, \dots, \Sigma_{N,t}\}$ and $v_t = \{v_{1,t}, v_{2,t}, \dots, v_{N,t}\}$.

State model of AMAV: The heterogeneous MAV swarm contains an AMAV A, which can acquire accurate location estimation with its sensing and computing capabilities using SLAM methods.

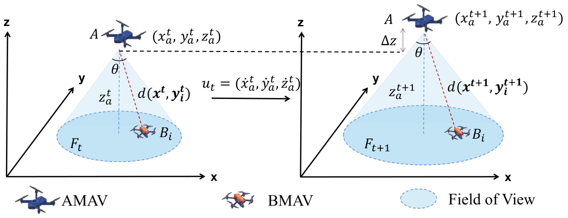
z
A
(x_a^t, y_a^t, z_a^t)
θ
y
z_a^t
$d(\boldsymbol{x}^t, \boldsymbol{y}_i^t)$
B_i
F_t
x
$u_t = (\dot{x}_a^t, \dot{y}_a^t, \dot{z}_a^t)$
z
A
$(x_a^{t+1}, y_a^{t+1}, z_a^{t+1})$
Δz
θ
y
z_a^{t+1}
$d(\boldsymbol{x}^{t+1}, \boldsymbol{y}_i^{t+1})$
B_i
F_{t+1}
x
AMAV
BMAV
Field of View

Figure 2: The observation model of the AMAV.

At time t, location of A is $\boldsymbol{x^t} = (x_a^t, y_a^t, z_a^t)$. When receiving a velocity command $u_t = (\dot{x}_a^t, \dot{y}_a^t, \dot{z}_a^t)$, A moves accordingly. The AMAV is faster than all BMAVs.

Observation model of AMAV: As Fig.2 illustrates, the AMAV A is equipped with a downward vision sensor that can generate observations for some BMAVs with respect to their actual location. The observation angle of the visual sensor is θ, and the field of view (FOV) of the visual sensor at time t is

$$F_t = \{(x, y) \mid \sqrt{(x - x_a^t)^2 + (y - y_a^t)^2} \le z_a^t \tan \frac{\theta}{2}\}. \tag{2}$$

The FOV of the visual sensor is related to the height of the AMAV A, z_a^t. The observation of BMAV B_i is noised. The degree of noise is related to the observation media and the distance between A and B_i. After pre-processing, the observed location of B_i is obtained from the observation. Specifically, when B_i appears within F_t, the observed location of B_i is

$$\boldsymbol{z_i^t} = \begin{cases} (x_i^t, y_i^t) + f_{\mu,\Sigma}^i, & (x_i^t, y_i^t) \in F_t \\ \text{none}, & \text{otherwise}, \end{cases} \tag{3}$$

where $f_{\mu,\Sigma}^i$ is noise of observation, $\mu = [0, 0]^T$ and Σ is defined as

$$\Sigma = \begin{cases} \begin{bmatrix} h^2(d(\boldsymbol{x^t}, \boldsymbol{y_i^t})) & 0 \\ 0 & h^2(d(\boldsymbol{x^t}, \boldsymbol{y_i^t})) \end{bmatrix}, & y_i^t \in F_t \\ \text{none}, & \text{otherwise}. \end{cases} \tag{4}$$

$$h(d(\boldsymbol{x^t}, \boldsymbol{y_i^t})) = n\sqrt{(x_a^t - x_i^t)^2 + (y_a^t - y_i^t)^2 + {z_a^t}^2}, \tag{5}$$

where n is a constant noise factor related to the observation media. We define $\boldsymbol{z^t} = \{\boldsymbol{z_1^t}, \boldsymbol{z_2^t}, \dots, \boldsymbol{z_N^t}\}$.

Estimation model of BMAV: In the process of moving, the AMAV A uses noisy velocity measurement from the BMAV B_i at time $t-1$, the estimation and the observed location of B_i at time $t-1$ to calculate the estimation of B_i at time t, $\widehat{\boldsymbol{y_i^t}}$. Specifically, A first calculates prior of B_i at time t, $\boldsymbol{y_i^{t^-}} = (x_i^{t^-}, y_i^{t^-}, \dot{x}_i^{t^-}, \dot{y}_i^{t^-})$ using noisy velocity measurement from B_i, where $(x_i^{t^-}, y_i^{t^-})$ is prior location of B_i and $(\dot{x}_i^{t^-}, \dot{y}_i^{t^-})$ is prior velocity of B_i. Then A calculates posterior of B_i, $\boldsymbol{y_i^{t^+}} = (x_i^{t^+}, y_i^{t^+}, \dot{x}_i^{t^+}, \dot{y}_i^{t^+})$, if B_i is observed by A at time t. $(x_i^{t^+}, y_i^{t^+})$ is posterior location of B_i and $(\dot{x}_i^{t^+}, \dot{y}_i^{t^+})$ is posterior velocity of B_i. We define $\boldsymbol{y^{t^-}} = \{\boldsymbol{y_1^{t^-}}, \boldsymbol{y_2^{t^-}}, \dots, \boldsymbol{y_N^{t^-}}\}$, $\boldsymbol{y^{t^+}} = \{\boldsymbol{y_1^{t^+}}, \boldsymbol{y_2^{t^+}}, \dots, \boldsymbol{y_N^{t^+}}\}$. Finally, A calculates the estimation of B_i at time t, which is defined as

$$\widehat{\boldsymbol{y_i^t}} = \begin{cases} \boldsymbol{y_i^{t^+}}, & \text{if } A \text{ observes } \mathrm{B_i} \\ \boldsymbol{y_i^{t^-}}, & \text{if } A \text{ doesn't observe } \mathrm{B_i} \end{cases}. \tag{6}$$

We define $\widehat{\boldsymbol{y^t}} = \{\widehat{\boldsymbol{y_1^t}}, \widehat{\boldsymbol{y_2^t}}, \dots, \widehat{\boldsymbol{y_N^t}}\}$. The specific calculation process will be presented in §3.

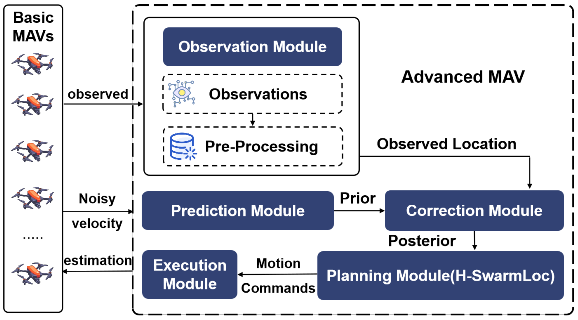

Figure 3: The collaborative localization system for the low-cost heterogeneous MAV swarm

2.2 Optimized Objective

The state of the BMAV B_i at time t is $\boldsymbol{y_i^t}$ and the observation of B_i at time t is $\boldsymbol{z_i^t}$. The AMAV generates observed locations for different BMAVs at different times, such that the AMAV can have an accurate estimation for each BMAV, at each time in the entire time horizon. As a measure of the reduction in uncertainty, mutual information is able to capture the statistical dependencies between two variables [39, 40]. The objective of the planning is to optimize the uncertainty in the estimation of BMAVs, which is to optimize the sum of mutual information between $\boldsymbol{y_i^t}$ and $z_i^{1:t}$ of each BMAV in each time t of entire time horizon T,

$$\phi = \sum_{t=1}^{T} \sum_{i=1}^{N} I(\boldsymbol{y_i^t}; \boldsymbol{z_i^{1:t}} \mid \boldsymbol{x}^{1:t}, v_{i,1:t-1}). \tag{7}$$

2.3 Problem Formulation

Given an initial state of AMAV $\boldsymbol{x^0}$, the initial estimation of BMAVs $\widehat{\boldsymbol{y^0}}$ and the time horizon T. The planning objective of the AMAV is to choose a sequence of actions $u_t = \pi(\widehat{\boldsymbol{y^t}}, \boldsymbol{x^t}, \Sigma_t)$ to generate observed locations for BMAVs to reduce the uncertainty in the estimation of BMAVs. Thereafter, the goal is to maximize the sum of mutual information between $\boldsymbol{y_i^t}$ and $z_i^{1:t}$of B in each time t,

$$\max_{\pi} \ \phi \tag{8}$$

$$\text{s.t. } 0 \le x_i^t, x_a^t \le L, t = 0, \dots, T, i = 1, \dots, N \tag{9}$$

$$0 \le y_i^t, y_a^t \le W, t = 0, \dots, T, i = 1, \dots, N \tag{10}$$

$$0 \le z_a^t \le H, t = 0, \dots, T \tag{11}$$

Constraint (9)-(10) ensures the BMAVs operate in Ω with a low height which is constant during the mission. Constraint (9)-(11) ensures the AMAV operates in Ω.

3 SYSTEM DESIGN

In this section, we will introduce the design of the collaborative localization system, which enables the heterogeneous MAV swarm to accomplish high-precision localization without external localization infrastructure. We first give an overview of the system architecture in §3.1, and then introduce *H-SwarmLoc* which plans the movement of the AMAV in §3.2.

3.1 System Architecture

To enable the heterogeneous MAV swarm to acquire high-precision localization, we design the collaborative localization system which takes full advantage of AMAV to help localize the BMAVs. This system overcomes the limitations of the low-cost MAV swarm, and efficiently achieves system-wide precise localization through collaboration without significant cost increase.

Fig. 3 illustrates the diagram of the collaborative localization system. The AMAV can obtain a precise location with its premium capabilities using SLAM methods. Utilizing the noisy velocity measurement from the BMAVs and the observation from the downward visual sensor, the AMAV helps localize BMAVs.

The **Observation module** uses the observation of the BMAVs acquired by the downward visual sensor to calculate the observed location of the BMAVs. The noise model of different observation media is different. The observation module generates the observation on B_i and calculates the observed location of B_i only when B_i appears in the FOV of AMAV at time t. Several methods can be used to calculate the observed location of BMAV B_i [41, 42].

The **Prediction module** uses the noisy velocity measurement from the BMAVs to calculate prior of the BMAVs. The estimation of BMAV B_i at time t is the prior when B_i does not appear in the FOV of the AMAV. The uncertainty in the estimation of BMAVs and the localization error of BMAVs gradually increases with time when only the noisy velocity measurement is used for state estimation.

The **Correction module** calculates the posterior of the BMAVs by fusing the observed location from the observation module with the prior from the prediction module. This module reduces the uncertainty in the estimation of the BMAVs. The localization error of BMAVs decreases in the presence of the observed location. The correction module calibrates the estimation of B_i only when B_i appears in the FOV of the AMAV at time t.

The **Planning module** uses the estimation of the BMAVs and the state of the AMAV to calculate the following command of the AMAV utilizing a reinforcement learning-based method in real-time. The Execution module takes action according to the command to generate observations for BMAVs. The details of this module will be further discussed in §3.2.

The system estimates states of BMAVs based on Bayes filter algorithm. As summarized in Algorithm 1, the input of the algorithm includes the estimation of BMAVs **B**, the covariance matrix of **B**, the noisy measurement velocity from **B**, the state of AMAV, and the command of AMAV at time $t-1$. The output of the algorithm is the estimation of **B** at time t. The algorithm first calculates F_t according to equation (2). Then the for loop calculates the prior using the noisy velocity measurement of BMAVs. If BMAV B_i is not in the FOV of A at time t, the estimation of B_i at time t is prior, else the algorithm calculates the posterior of B_i using the observed location with equations (2)-(4). In this case, the estimation of B_i at time t is the posterior. Note that, η is the normalization constant.

Algorithm 1 The AMAV assists BMAVs for localization using the noisy measurement velocity from the BMAVs and the observation.

```
Input: $\widehat{\boldsymbol{y}^{t-1}}, \Sigma_{t-1}, v_{t-1}, \boldsymbol{x}^{t-1}, u_{t-1}$
Output: $\widehat{\boldsymbol{y}^{t}}$
$\widehat{\boldsymbol{y}^{t}} \leftarrow \emptyset$;
Calculate $F_t$ according to equation (2);
for $i \leftarrow 1$ to $N$ do
    $\overline{bel(\boldsymbol{y}_i^t)} = \int p\left(\boldsymbol{y}_i^t \mid v_{i,t-1}, \boldsymbol{y}_i^{t-1}\right) bel(\boldsymbol{y}_i^{t-1}) d\boldsymbol{y}_i^{t-1}$;
    Calculate $\boldsymbol{y}_i^{t^-}$ from $\overline{bel(\boldsymbol{y}_i^t)}$;
    $\widehat{\boldsymbol{y}_i^t} = \boldsymbol{y}_i^{t^-}$;
    Calculate $\boldsymbol{z}_i^t$ according to equation (3), (4), (5);
    if $z_i^t$ is not none then
        $bel(\boldsymbol{y}_i^t) = \eta p\left(z_i^t \mid \boldsymbol{y}_i^t\right) \overline{bel(\boldsymbol{y}_i^t)}$;
        Calculate $\boldsymbol{y}_i^{t^+}$ from $bel(\boldsymbol{y}_i^t)$;
        $\widehat{\boldsymbol{y}_i^t} = \boldsymbol{y}_i^{t^+}$; % if A observes $B_i$
    end if
    $\widehat{\boldsymbol{y}^t} \leftarrow \widehat{\boldsymbol{y}_i^t}$;
end for
```

3.2 *H-SwarmLoc*: Planning of the AMAV

The AMAV needs to balance the trade-off between the number of BMAVs observed and the quality of the observation and keep the localization performance of the heterogeneous MAV swarm at a high level in the entire time horizon. Meanwhile, the planning of AMAV is a receding horizon path-planning problem, which usually implies high computational complexity.

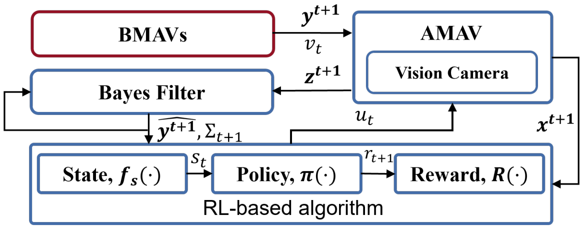

Figure 4: Illustration of the planning methods.

As Fig.4 illustrates, *H-SwarmLoc*, a reinforcement learning-based planning method is proposed to plan the motion of the AMAV for the non-myopic objective. By using this method, the discounted sum of future rewards is maximized over the entire time horizon. Additionally, the policies obtained by this method can be executed online with an extended training stage.

State Space: The state of the AMAV and the estimation of the BMAVs form the state space. More formally,

$$s_{i,t} \equiv \left[\widehat{x_i^t}, \widehat{y_i^t}, \widehat{\dot{x}_i^t}, \widehat{\dot{y}_i^t}, \log\det\Sigma_{i,t}, \boldsymbol{z}_i^t\right]^T, \tag{12}$$

$$s_t \equiv \left[s_{1,t}^T, \ldots, s_{N,t}^T, \boldsymbol{x}^t\right], \tag{13}$$

where $\log\det\Sigma_{i,t}$ indicates the estimation uncertainty of the BMAV B_i at time t. A higher value indicates a more inaccurate estimation of the BMAV. The s_t is a collection of the $s_{i,t}$.

Action Space: We define the action space with a finite number of motion options, including six possible actions: forward, backward, go left, go right, go up or go down.

Reward: We define the reward function of *H-SwarmLoc* as

$$R\left(s_t, u_t, s_{t+1}\right) = -\alpha \frac{\sum_{i=1}^{N} \log\det\Sigma_{i,t+1}}{N} - \beta\sigma\left(\log\det\Sigma_{t+1}\right), \tag{14}$$

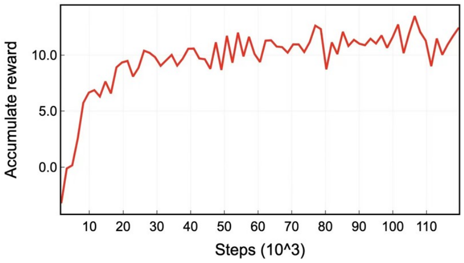

Figure 5: The training process of the control policy. As the training time increases, the network eventually converges.

The first item penalizes the mean uncertainty of BMAVs. The second item calculates standard deviation to prevent the AMAV from generating observation for only a few BMAVs when not all the BMAVs are within the FOV. α and β are constant factors to adjust the weight between two items.

Training: We use a reinforcement learning algorithm to learn an optimal policy for this planning problem. The training process is summarized in Algorithm 2. In the training process, the algorithm continues to improve the model by interacting with the environment, and the algorithm only needs state and reward after each step and does not require any knowledge of the dynamic model, estimation model, and observation model.

Algorithm 2 Learning a Policy for Planning base on reinforcement learning algorithm

```
Randomly initialize a model $\pi$
for epoch=1:$M$ do
    randomly initialize $\widehat{\boldsymbol{y}^{0}}$, $\boldsymbol{y}^{0}$, $\boldsymbol{x}^{0}$ and $\Sigma_0$;
    for $t = 0 : T - 1$ do
        AMAV executes an action $u_t = \pi(s_t)$;
        Calculate $\boldsymbol{x}^{t+1}$;
        Calculate $\boldsymbol{z}^{t+1}$; % according to equation (3)
        $(\widehat{\boldsymbol{y}^{t+1}}, \Sigma_{t+1}) \leftarrow$ Bayes Filter $(\widehat{\boldsymbol{y}^{t}}, \Sigma_t, \boldsymbol{z}^{t})$
        $r_{t+1} = R(s_t, u_t, s_{t+1})$
        Update the model $\pi$
        Update the state, $s_{t+1}$
    end for
end for
```

4 EVALUATION

In this section, we evaluate the performance of the collaborative localization system and *H-SwarmLoc* in location estimation on the simulation platform. First, we illustrate the training process of the planning policy. Then, we show the localization performance of the system. Finally, we give a demonstration of the running process.

4.1 Experiment Setup

The arena and BMAV model: The length L, width W, and height H of our simulation test scenario are $100m$, $100m$, $40m$, where the BMAVs execute different tasks. The BMAVs used in the simulation are modeled after the crazyfile platform [43]. Each node has a 3-axis accelerometer, a 3-axis gyroscope, a high-precision pressure sensor, and an optical flow sensor, which can be used to measure velocity.

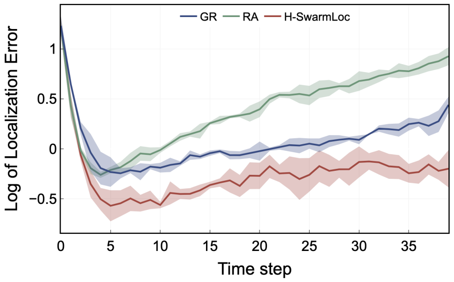

Figure 6: Comparison of localization performance with different planning methods in the presence of the AMAV. The solid line represents the mean value of the localization error of BMAVs, and the shaded part represents the variance.

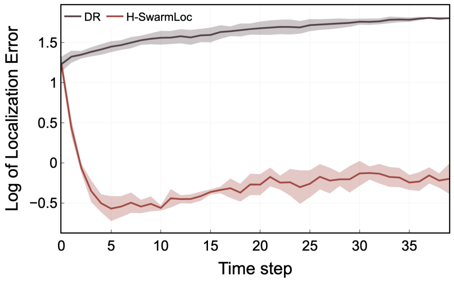

Figure 7: The localization performance improvement with the presence of the AMAV. The solid line represents the mean value of the localization error of BMAVs, and the shaded part represents the variance.

Estimation model of the BMAV: We use the Kalman filter to estimate the state of the BMAVs. The Kalman filter can be replaced by any kind of Bayes filter or other state estimation methods.

Metrics: As our goal is to minimize localization error of all BMAVs, we use δ as metric, where

$$\delta = \frac{\sum_{i=1}^{N} \left\| \boldsymbol{y}_i^t - \widehat{\boldsymbol{y}_i^t} \right\|_2}{N}. \tag{15}$$

A smaller value of δ indicates a smaller localization error of all BMAVs at the time t.

Parameters: We set $N = 3$, the maximum speed of BMAVs is $2m/s$, the initial velocity of BMAVs is randomly initialized, and the covariance of all BMAVs is initialized to $30.0I_4$. To make the experiment more realistic, the localization error of BMAVs is randomly initialized and less than $10m$. The actual location of AMAV is randomly initialized, and the speed of AMAV is $5m/s$, which means AMAV can forward, backward, go left, go right, and go up or down with a rate of $5m/s$. The FOV of AMAV is $2\pi/3$. We set $\tau = 0.5$, $n = 0.2$, and $q = 0.5$. All experimental results are obtained with twenty different random seeds.

Baselines:

- *Dead-Reckoning with Map Bias (DR):* DR is an infrastructure-free technique which uses noisy measurements from the accelerometer, gyroscope and optical flow from BMAVs to estimate the locations of BMAVs[44].
- *Random (RA):* RA randomly selects an action for AMAV to execute to limit the localization error of BMAVs.
- *Greedy (GR):* GR selects an action leads AMAV to the BMAV which has the maximum $\log\det\Sigma_{i,t}$.

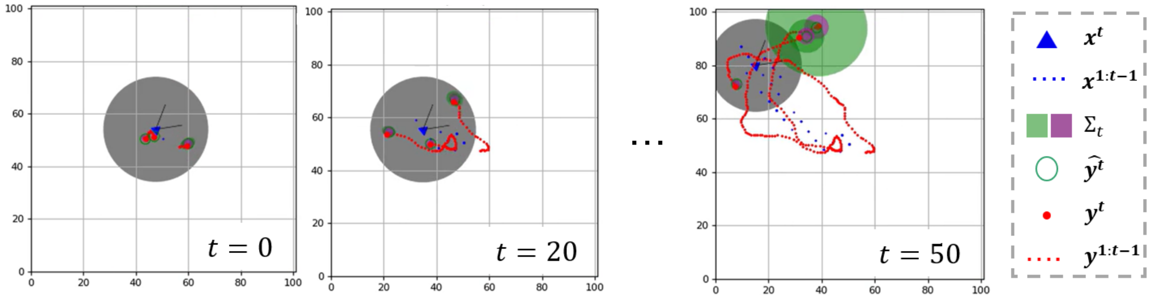

Figure 8: Demonstration of collaborative localization with three BMAVs and one AMAV. Time step increases from left to right. Blue triangle: x^t. Blue dots: $x^{1:t-1}$. Big red dots: y^t. Small red dots: $y^{1:t-1}$. Green circles: $\widehat{y^t}$. Green and purple shaded area: Σ_t.

4.2 Evaluation Results and Analysis

In general, the localization performance of the collaborative localization system and *H-SwarmLoc* outperform all baselines. The system can keep the localization error of the whole system at a relatively low level in the entire time horizon.

Policy training: We use the Proximal Policy Optimization (PPO) algorithm to train the control policy of the AMAV. During the training process of the policy, we set $\alpha = 0.1$, $\beta = 0.1$ to calculate the reward. We used a multi-layer perception (MLP) as the model with two layers of 64 units and a learning rate of 0.0003. The model is updated every 1024 training steps. The batch size is 128. As Fig.5 illustrates, our model converges after training.

Localization Error: To investigate the performance of *H-SwarmLoc*, we compare *H-SwarmLoc* with RA and GA. As Fig. 6 illustrates, the localization error of *H-SwarmLoc* decreases 38% and 12% on average compared to RA and GR. The localization error of RA, GR, and *H-SwarmLoc* gradually increases as time increases, but *H-SwarmLoc* growth slower than RA and GR. This is because *H-SwarmLoc* can plan the motions of the AMAV for the non-myopic objective. At the beginning of the experiment, the localization errors of GR, RA, and *H-SwarmLoc* are all at a high level because there is a distance between the initial actual location and the initial location estimation of the BMAV. Subsequently, the AMAV generates observation for the BMAVs, and the distance begins to decrease. However, the observations generated by the AMAV for the BMAVs are different with different planning algorithms, and the localization error of the BMAVs varies. Finally, both RA and GR cannot maintain high accuracy localization for BMAVs, while *H-SwarmLoc* can still keep the localization error at a relatively low level.

To verify the performance of the introduction of the AMAV, we compare *H-SwarmLoc* with DR. As Fig. 7 illustrates, the localization error of *H-SwarmLoc* decreases by 78% on average compared to DR. The localization error of DR keeps increasing because the BMAVs have no external observation and can only be localized by their noisy measurements. In the beginning, both our *H-SwarmLoc* and DR have a high localization error because the distance between the initial actual location and location estimation of BMAVs is not zero. Subsequently, the AMAV generates observations for the BMAVs, and the localization error of BMAVs decreases and remains low. However, the localization error of the BMAVs increases rapidly with time when using only noisy measurement velocity from BMAVs for Dead-Reckoning.

Demonstration: To verify the feasibility of the system, we design the visualization platform based on [45], and use three BMAVs and one AMAV to run the simulation. As Fig.8 illustrates, three BMAVs move as time increases. The blue triangle represents the state of AMAV at time t, the blue dots represents the state of AMAV from time 1 to time $t-1$, the big red dots represents the state of BMAVs at time t, the small red dots represents the state of BMAVs from time 1 to time $t-1$, the green circles represents the location estimation of BMAVs at time t, the green and purple shaded area represents the covariance at time t. The AMAV generates observations for the BMAVs. During operation, *H-SwarmLoc* can select the action for the AMAV to minimize the localization error of the heterogeneous MAV swarm on the entire time horizon.

5 DISCUSSION

In this section, we discuss several problems about the system and the future research directions of this paper. Firstly, the maximum number of the BMAVs that an AMAV can observe depends on the observation media. If we utilize tag family TAG36H11 of AprilTag as the observation media, the maximum number of the BMAVs is 587 because the id of tag family TAG36H11 ranges from 0 to 586. When we consider the physical size of the BMAV, the maximum number of the BMAVs is less than 587. Secondly, the minimum hardware requirement of the AMAV is a monocular camera and an IMU which are basic requirements to execute the SLAM methods [24]. This setting is cheap enough to balance the trade-off between the accuracy and the cost of the system. Meanwhile, it is useful to use a Stereo camera and an high-precision IMU to make the system more robust in a realistic setting. Finally, the future research directions of this paper include developing more robust methods to generate observation for BMAVs and planning more AMAVs to generate observation for BMAVs simultaneously.

6 CONCLUSION

In this paper, we propose a collaborative localization system , which takes full advantage of the AMAV to improve the localization performance of the heterogeneous MAV swarm. This system overcomes the limitations of the low-cost MAV swarm through collaboration. Subsequently, *H-SwarmLoc*, a reinforcement learning-based planning method is proposed to plan the movement of AMAV in real-time. Evaluations show that *H-SwarmLoc* achieves better localization performance compared to baselines.

REFERENCES

[1] Yunzhong Jiang, Rui Zhang, and Bende Wang. Scenario-based approach for emergency operational response: Implications for reservoir management decisions. *International Journal of Disaster Risk Reduction*, 80:103192, 2022.

[2] Han Fan, Victor Hernandez Bennetts, Erik Schaffernicht, and Achim J Lilienthal. Towards gas discrimination and mapping in emergency response scenarios using a mobile robot with an electronic nose. *Sensors*, 19(3):685, 2019.

[3] Yuting Wan, Yanfei Zhong, Ailong Ma, and Liangpei Zhang. An accurate uav 3-d path planning method for disaster emergency response based on an improved multiobjective swarm intelligence algorithm. *IEEE Transactions on Cybernetics*, 2022.

[4] Xinlei Chen, Susu Xu, Jun Han, Haohao Fu, Xidong Pi, Carlee Joe-Wong, Yong Li, Lin Zhang, Hae Young Noh, and Pei Zhang. Pas: Prediction-based actuation system for city-scale ridesharing vehicular mobile crowdsensing. *IEEE Internet of Things Journal*, 7(5):3719–3734, 2020.

[5] Xuecheng Chen, Haoyang Wang, Zuxin Li, Wenbo Ding, Fan Dang, Chenye Wu, and Xinlei Chen. Deliversense: Efficient delivery drone scheduling for crowdsensing with deep reinforcement learning. 2022.

[6] Xinlei Chen, Susu Xu, Xinyu Liu, Xiangxiang Xu, Hae Young Noh, Lin Zhang, and Pei Zhang. Adaptive hybrid model-enabled sensing system (hmss) for mobile fine-grained air pollution estimation. *IEEE Transactions on Mobile Computing*, 2020.

[7] Zhengwei Wu, Xiaoxi Zhang, Susu Xu, Xinlei Chen, Pei Zhang, Hae Young Noh, and Carlee Joe-Wong. A generative simulation platform for multi-agent systems with incentives. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pages 580–587, 2020.

[8] Weipeng Guan, Shihuan Chen, Shangsheng Wen, Zequn Tan, Hongzhan Song, and Wenyuan Hou. High-accuracy robot indoor localization scheme based on robot operating system using visible light positioning. *IEEE Photonics Journal*, 12(2):1–16, 2020.

[9] Mehdi Hellou, Norina Gasteiger, Jong Yoon Lim, Minsu Jang, and Ho Seok Ahn. Personalization and localization in human-robot interaction: A review of technical methods. *Robotics*, 10(4):120, 2021.

[10] Priya Roy and Chandreyee Chowdhury. A survey of machine learning techniques for indoor localization and navigation systems. *Journal of Intelligent & Robotic Systems*, 101(3):1–34, 2021.

[11] Susu Xu, Xinlei Chen, Xidong Pi, Carlee Joe-Wong, Pei Zhang, and Hae Young Noh. ilocus: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing. *IEEE Transactions on Mobile Computing*, 19(8):1831–1847, 2019.

[12] Xinlei Chen, Xiangxiang Xu, Xinyu Liu, Shijia Pan, Jiayou He, Hae Young Noh, Lin Zhang, and Pei Zhang. Pga: Physics guided and adaptive approach for mobile fine-grained air pollution estimation. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 1321–1330, 2018.

[13] Xinlei Chen, Carlos Ruiz, Sihan Zeng, Liyao Gao, Aveek Purohit, Stefano Carpin, and Pei Zhang. H-drunkwalk: Collaborative and adaptive navigation for heterogeneous mav swarm. *ACM Transactions on Sensor Networks (TOSN)*, 16(2):1–27, 2020.

[14] Nipun D Nath, Chih-Shen Cheng, and Amir H Behzadan. Drone mapping of damage information in gps-denied disaster sites. *Advanced Engineering Informatics*, 51:101450, 2022.

[15] Shangyao Yan, Zhimeng Yin, and Guang Tan. Curvelight: An accurate and practical indoor positioning system. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 152–164, 2021.

[16] Jonas Beuchert and Alex Rogers. Snappergps: Algorithms for energy-efficient low-cost location estimation using gnss signal snapshots. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 165–177, 2021.

[17] Prabin Kumar Panigrahi and Sukant Kishoro Bisoy. Localization strategies for autonomous mobile robots: A review. *Journal of King Saud University-Computer and Information Sciences*, 2021.

[18] James A Preiss, Wolfgang Honig, Gaurav S Sukhatme, and Nora Ayanian. Crazyswarm: A large nano-quadcopter swarm. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3299–3304. IEEE, 2017.

[19] Jun Liu, Jiayao Gao, Sanjay Jha, and Wen Hu. Seirios: leveraging multiple channels for lorawan indoor and outdoor localization. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 656–669, 2021.

[20] Wenqiang Chen, Maoning Guan, Lu Wang, Rukhsana Ruby, and Kaishun Wu. Floc: Device-free passive indoor localization in complex environments. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, 2017.

[21] Jiageng Liu and Ge Guo. Vehicle localization during gps outages with extended kalman filter and deep learning. *IEEE Transactions on Instrumentation and Measurement*, 70:1–10, 2021.

[22] Rui Chen, Xiyuan Huang, Yan Zhou, Yilong Hui, and Nan Cheng. Uhf-rfid-based real-time vehicle localization in gps-less environments. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[23] Phuc Nguyen, Taeho Kim, Jinpeng Miao, Daniel Hesselius, Erin Kenneally, Daniel Massey, Eric Frew, Richard Han, and Tam Vu. Towards rf-based localization of a drone and its controller. In *Proceedings of the 5th workshop on micro aerial vehicle networks, systems, and applications*, pages 21–26, 2019.

[24] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[25] Tong Qin and Shaojie Shen. Online temporal calibration for monocular visual-inertial systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3662–3669. IEEE, 2018.

[26] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[27] Patrik Schmuck and Margarita Chli. Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *Journal of Field Robotics*, 36(4):763–781, 2019.

[28] Zhengxiong Li, Baicheng Chen, Xingyu Chen, Chenhan Xu, Yuyang Chen, Feng Lin, Changzhi Li, Karthik Dantu, Kui Ren, and Wenyao Xu. Reliable digital forensics in the air: Exploring an rf-based drone identification system. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–25, 2022.

[29] Sofiya Semenova, Pranay Meshram, Timothy Chase, Steven Y Ko, Yu David Liu, Lukasz Ziarek, and Karthik Dantu. A modular, extensible framework for modern visual slam systems. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pages 579–580, 2022.

[30] Qin Zou, Qin Sun, Long Chen, Bu Nie, and Qingquan Li. A comparative analysis of lidar slam-based indoor navigation for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[31] Jingao Xu, Hao Cao, Zheng Yang, Longfei Shangguan, Jialin Zhang, Xiaowu He, and Yunhao Liu. {SwarmMap}: Scaling up real-time collaborative visual {SLAM} at the edge. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 977–993, 2022.

[32] Jingao Xu, Hao Cao, Danyang Li, Kehong Huang, Chen Qian, Longfei Shangguan, and Zheng Yang. Edge assisted mobile semantic visual slam. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1828–1837. IEEE, 2020.

[33] Aveek Purohit, Zheng Sun, and Pei Zhang. Sugarmap: Location-less coverage for micro-aerial sensing swarms. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 253–264, 2013.

[34] Xinlei Chen, Aveek Purohit, Carlos Ruiz Dominguez, Stefano Carpin, and Pei Zhang. Drunkwalk: Collaborative and adaptive planning for navigation of micro-aerial sensor swarms. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 295–308, 2015.

[35] Xinlei Chen, Aveek Purohit, Shijia Pan, Carlos Ruiz, Jun Han, Zheng Sun, Frank Mokaya, Patric Tague, and Pei Zhang. Design experiences in minimalistic flying sensor node platform through sensorfly. *ACM Transactions on Sensor Networks (TOSN)*, 13(4):1–37, 2017.

[36] Luca Varotto, Angelo Cenedese, and Andrea Cavallaro. Active sensing for search and tracking: A review. *arXiv preprint arXiv:2112.02381*, 2021.

[37] Lars Kunze, Nick Hawes, Tom Duckett, Marc Hanheide, and Tomáš Krajník. Artificial intelligence for long-term robot autonomy: A survey. *IEEE Robotics and Automation Letters*, 3(4):4023–4030, 2018.

[38] Ao Qu, Yihong Tang, and Wei Ma. Attacking deep reinforcement learning-based traffic signal control systems with colluding vehicles. *arXiv preprint arXiv:2111.02845*, 2021.

[39] Gabriel M Hoffmann and Claire J Tomlin. Mobile sensor network control using mutual information methods and particle filters. *IEEE Transactions on Automatic Control*, 55(1):32–47, 2009.

[40] Justin B Kinney and Gurinder S Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, 111(9):3354–3359, 2014.

[41] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE international conference on robotics and automation*, pages 3400–3407. IEEE, 2011.

[42] Mohammad Fattahi Sani and Ghader Karimian. Automatic navigation and landing of an indoor ar. drone quadrotor using aruco marker and inertial sensors. In *2017 international conference on computer and drone applications (IConDA)*, pages 102–107. IEEE, 2017.

[43] Wojciech Giernacki, Mateusz Skwierczyński, Wojciech Witwicki, Paweł Wroński, and Piotr Kozierski. Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 37–42. IEEE, 2017.

[44] B Everett and L Feng. Navigating mobile robots: Systems and techniques. *AK Peters, Ltd. Natick, MA, USA*, 1996.

[45] Heejin Jeong, Hamed Hassani, Manfred Morari, Daniel D. Lee, and George J. Pappas. Learning to track dynamic targets in partially known environments. *ArXiv*, abs/2006.10190, 2020.