

# Uncertainty-bounded Active Monitoring of Unknown Dynamic Targets in Road-networks with Minimum Fleet

Shuaikang Wang<sup>1</sup>, Yiannis Kantaros<sup>2</sup> and Meng Guo<sup>1</sup>

**Abstract**—Fleets of unmanned robots can be beneficial for the long-term monitoring of large areas, e.g., to monitor wild flocks, detect intruders, search and rescue. Monitoring numerous dynamic targets in a collaborative and efficient way is a challenging problem that requires online coordination and information fusion. The majority of existing works either assume a passive all-to-all observation model to minimize the summed uncertainties over all targets by all robots, or optimize over the jointed discrete actions while neglecting the dynamic constraints of the robots and unknown behaviors of the targets. This work proposes an online task and motion coordination algorithm that ensures an explicitly-bounded estimation uncertainty for the target states, while minimizing the average number of active robots. The robots have a limited-range perception to actively track a limited number of targets simultaneously, of which their future control decisions are all unknown. It includes: (i) the assignment of monitoring tasks, modeled as a flexible size multiple vehicle routing problem with time windows (m-MVRPTW), given the predicted target trajectories with uncertainty measure in the road-networks; (ii) the nonlinear model predictive control (NMPC) for optimizing the robot trajectories under uncertainty and safety constraints. It is shown that the robots can switch between active and inactive roles dynamically online as required by the unknown monitoring task. The proposed methods are validated via large-scale simulations of up to 100 robots and targets.

## I. INTRODUCTION

With the rapid development of robotic perception and motion techniques, mobile robots have been deployed to monitor areas that are otherwise too large or hostile for humans. Fleets of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) can be particularly suitable for this purpose, via wireless communication and collaborative execution, to e.g., map time-varying fields [1], [2], [3], track moving targets [4], [5] and explore unknown territory [6]. As a fundamental technique to many applications, active monitoring of unknown dynamic targets has attracted significant attention, see [7], [8], [9], [10]. It remains a challenging problem as it involves the online coordination of monitoring tasks among the robots, and the real-time adaptation of their trajectories w.r.t. the observed behaviors of the targets.

### A. Related Work

As the most relevant work, [7], [11] pioneered the problem of active information acquisition (AIA) with sensing robots.

The authors are with <sup>1</sup>the College of Engineering, Peking University, Beijing 100871, China; and <sup>2</sup>the Department of Electrical and Systems Engineering, Washington University at St. Louis, St. Louis, MO, 63130, USA. This work was supported by <sup>1</sup>National Science and Technology Major Project under grant 2022ZD0116401; <sup>1</sup>the National Natural Science Foundation of China (NSFC) under grants 62203017, U2241214, T2121002; and <sup>2</sup>the ARL grant DCIST CRA W911NF-17-2-0181. Contact: meng.guo@pku.edu.cn.

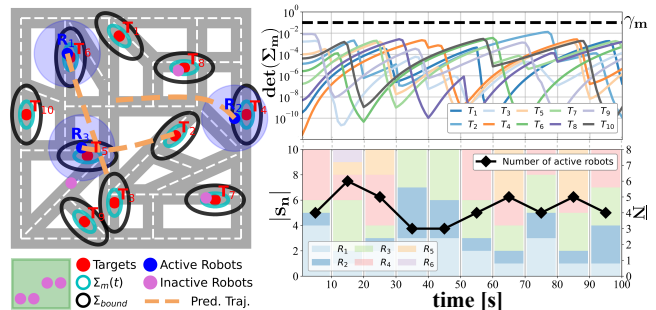


Fig. 1. Illustration of the considered scenario. **Left**: 3 UAVs (in blue) are actively monitoring 10 targets (in red) within a road network, with 7 inactive ones (in orchid) that are recruited later online; **Top-right**: Evolution of estimation uncertainty for all targets below a specified upper-bound (in black); **Bottom-right**: Number of active UAVs in the fleet and number of targets assigned to each active UAV.

The proposed solutions exploit the separation principles under the linear observation model and utilize an offline search algorithm over the joint state-and-information space, via the forward value iteration (FVI) and later the reduced forward value iteration (RVI). This line of methods has been extended to multi-robot fleets, where an optimal, non-myopic, and centralized sampling-based solution is proposed in our earlier work [5], [12]; and decentralized myopic planners can be found in [8], [9], [10], [13]. The target tracking problem is combined with robot localization in [14] for only one target. Recent work [4] utilizes graph neural networks to learn distributed and scalable action policies from the optimal planner. However, most of these works assume a *fixed* size of the robotic team with discretized control inputs and a linear observation model, and more importantly the targets move with a known model, i.e., known dynamics and control inputs. Instead, behaviors of the considered targets including velocities and paths within the road network are all unknown, thus requiring online prediction and tracking.

On the other hand, another line of research formulates the multi-target monitoring problem as a sequential high-level assignment problem, where robots are assigned to targets for reducing uncertainty. A distributed Hungarian method is proposed in [15], which however only applies to one-to-one assignments. Numerous works [14], [16], [17] reformulate it as a simultaneous action and target assignment (SATA) problem, and propose distributed approximation algorithms such as the linear programs. However, the target trajectories are often assumed to be given, and the robots move synchronously with a set of pre-defined motion primitives. The work [18] considers a complementary problem to this work, which maximizes the number of targets being tracked by a fixed number of robots via a 2-approximation greedy

solution. Recent work [19], [20] studies the robust assignment strategies against possible attacks in communication and sensing, which also assumes a synchronized team motion with a set of finite primitives. Instead, asynchronous and timed robot trajectories are computed recursively online in this work given the observed target behaviors.

### B. Our Method

This work addresses the dynamic monitoring problem from a practical perspective, i.e., *how many robots are needed to monitor a set of dynamic targets with an upper-bounded uncertainty, and how to control their trajectories.* Different from many existing works, these targets follow a constant-velocity model, but with unknown velocity on each road, and unknown path within the road network. The proposed solution consists of two-layers: (i) the first layer explicitly formulates the assignment problem of the active monitoring tasks, under the constraints that each robot can only monitor a maximum number of targets simultaneously. The assignment scheme first over-approximates the monitoring quality given the predicted trajectories and uncertainties of all targets, then formulates a flexible-size multiple vehicle routing problem with time windows (m-MVRPTW); (ii) the second layer then formulates a constrained optimal control problem given the assignment results. The robot trajectories are optimized online via the distributed nonlinear model predictive control (NMPC), where the dynamical constraints and predicted target trajectories are all incorporated. Both optimizations can be solved via off-the-shelf solvers.

Main contribution of this work is three-fold: (i) the novel formulation of active monitoring problem for unknown targets with an upper-bounded uncertainty and a minimum fleet. To the best of our knowledge, this problem has not been addressed in related work; (ii) the proposed two-level solution adapts the fleet size and the robot trajectories online, according to real-time measurements of the target behavior; (iii) the scalable solution that allows the deployment of a few UAVs to monitor a large number of dynamic targets.

## II. PROBLEM DESCRIPTION

### A. Robots and Targets

Consider a team of  $N$  autonomous robots that coexist within a complex workspace  $\mathcal{W} \subset \mathbb{R}^3$ , where the total number  $N$  might be time-varying thus  $N(t)$  for time  $t > 0$ . Without loss of generality, each robot  $n \in \mathcal{N}(t) \triangleq \{1, \dots, N\}$  follows the same generic nonlinear dynamics:

$$\mathbf{x}_n(t+1) \triangleq f(\mathbf{x}_n(t), \mathbf{u}_n(t)), \quad (1)$$

where  $f(\cdot)$  is the dynamic model;  $\mathbf{x}_n(t) \in \mathcal{W}$  and  $\mathbf{u}_n(t) \in \mathcal{U}$  are the state and control input of robot  $i$  at time step  $t \geq 0$ . It is assumed in this work that the robots can communicate freely via a distributed network. Note that the total number  $N$  is not fixed, rather a part of the objective to minimize.

On the other hand, there are  $M$  targets that move dynamically within a known road network. Denote by  $\mathcal{R} \triangleq (\mathcal{V}, \mathcal{E})$  this network, where  $\mathcal{V} \subset \mathcal{W}$  is the set of hub points and

$\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$  is a set of *straight* roads connecting these hubs. Each target follows a sequence of roads to move from its starting point to the destination points, which are all *unknown* to the robots. On different roads, each target  $m \in \mathcal{M} \triangleq \{1, \dots, M\}$  follows the same constant-velocity model [7] but with different velocities, i.e.,

$$\mathbf{y}_m(t+1) \triangleq G \mathbf{y}_m(t) + \mathbf{w}_m(t), \quad (2)$$

where  $G \triangleq [\mathbf{I}_2, h\mathbf{I}_2; \mathbf{0}, \mathbf{I}_2]$  with sampling time  $h > 0$ ;  $\mathbf{y}_m(t) \in \mathcal{W}$  is the target state at  $t \geq 0$ , and  $\mathbf{w}_m(t) \sim \mathcal{N}(0, W_m)$  is the Gaussian noise with covariance  $W_m$ . Namely, *the target may switch to a different road and move with an unknown velocity, i.e., following an unknown path.*

### B. Active Sensing Model

Moreover, each robot  $n \in \mathcal{N}$  is equipped with a perception module to *actively track* the targets via action  $\mathbf{s}_n : \mathbb{Z} \rightarrow 2^{\mathcal{M}}$ :

$$\mathbf{z}_{nm}(t) \triangleq h(\mathbf{x}_n(t), \mathbf{y}_m(t)) + \mathbf{v}(t), \quad \forall m \in \mathbf{s}_n(t); \quad (3)$$

where  $h(\cdot)$  maps the current states of robot  $n$  and target  $m$  to the measurement  $\mathbf{z}_{nm}$ , i.e., robot  $n$  w.r.t. target  $m$ ; and  $\mathbf{v}(t) \sim \mathcal{G}(0, V(\mathbf{x}_n(t), \mathbf{y}_m(t)))$  is the Gaussian noise with a covariance  $V(t)$  that varies by the robot and target states.

**Remark 1.** Note that the observation model  $h(\cdot)$  is a general nonlinear and differentiable function, not necessarily linear w.r.t.  $\mathbf{y}_m$ . Thus, the separation principle does not hold anymore [7]. Consequently, the offline approaches in [8], [10], [12] cannot be applied; instead, the target state and covariance are estimated simultaneously online. ■

Denote by  $n \in \bar{\mathcal{N}} \subset \mathcal{N}$  if robot  $n$  becomes active. There are two constraints for its choice of actions  $\mathbf{s}_n(t)$ , i.e.,

$$d(\mathbf{x}_n(t), \mathbf{y}_m(t)) < R_n, \quad |\mathbf{s}_n(t)| < C_n; \quad (4)$$

where (i)  $d(\cdot)$  is a distance measure and  $R_n > 0$ , as typical sensors such as radar and lidar, are only valid within a limited range. Outside this range, its uncertainty  $V(t)$  in (3) is set to infinity. Note that  $R_n$  might be different across the robots; (ii) there is often *maximum* number of targets each robot  $n$  can track simultaneously due to limited computation or energy resources, denoted by  $C_n \geq 1$ .

**Remark 2.** Note that the definition of action  $\mathbf{s}_n(\cdot)$  is essential, and different from most existing work. Namely, some work [4], [7], [10] and our previous work [12], [21] assume that the robot can obtain measurements of *all* targets at all time, while others [14], [16], [17] limit to only *one* target. In practice, there is often either an energy or computation cost associated with detection and tracking one target. Thus, the robot has to actively decide which targets to measure within its sensing range, under a maximum number. ■

### C. Target State Estimation and Tracking

Initially, each target  $m \in \mathcal{M}$  has a prior Gaussian distribution for its initial state  $\mathbf{y}_m(0) \sim \mathcal{G}(\mu_m^0, \Sigma_m^0)$ , which is known to all robots. For time  $t > 0$ , each robot  $n \in \mathcal{N}$  can move around and make observations of some targets via (3),

i.e.,  $\mathbf{Z}_m(t) \triangleq \{\mathbf{z}_{nm}(t), \forall n \in \mathcal{N}\}$ , which may contain time-stamped observations of all targets if measured, otherwise set to empty. Given the accumulated measurements  $\mathbf{Z}_m(0:t)$ , the posteriori distribution of target  $m$  is denoted by:

$$\hat{\mathbf{y}}_m(t) \sim \mathcal{G}(\hat{\mu}_m(t), \hat{\Sigma}_m(t)), \quad (5)$$

which is the estimated mean and covariance of target  $m$  at time  $t$ , e.g., conveniently by the EKF [7], [22] when the target moves along one fixed road as described in the sequel. However, if target  $m$  is *not* monitored when it crosses a  $K$ -way intersection, its estimation  $\hat{\mathbf{y}}_m(t)$  changes by approximating the  $K$  possible directions statistically, i.e.,

$$\mu_m(t), \Sigma_m(t) \triangleq \mathcal{G}_{\text{apx}}\left(\{\mu_m^k(t), k = 1, \dots, K\}\right), \quad (6)$$

where  $\mathcal{G}_{\text{apx}}(\cdot)$  is the Gaussian approximation of a set of points;  $\|\hat{\mu}_m(t-1) - \nu\| < \varepsilon$ , for some intersection  $\nu \in \mathcal{V}$  and a small margin  $\varepsilon > 0$ ; and  $\mu_m^k(t)$  is the predicted next state if target  $m$  follows the  $k$ -th direction at time  $t-1$ . In other words, the uncertainty increases drastically if a target crosses intersections without being monitored. Thus, the determinant of the covariance matrix  $\mathbf{det}(\Sigma_m(t))$  is used to evaluate the quality of state estimation for target  $m$  up to time  $t$ .

#### D. Problem Statement

The considered problem can be stated as a long-term constrained optimization problem, i.e.,

$$\min_{\bar{N}(t), \{(\mathbf{u}_n, \mathbf{s}_n)\}} \left\{ \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \bar{N}(t) \right\}$$

**s.t.**  $\mathbf{det}(\Sigma_m(t)) < \gamma_m, \forall m, \forall t;$  (7a)

$$(1) - (6), \forall n, \forall m, \forall t; \quad (7b)$$

where  $\bar{N}(t) > 0$  is the number of active robots in  $\bar{N}(t)$  for  $t \geq 0$ ; the objective is the average number of active robots in the long horizon;  $\gamma_m > 0$  is a user-specified bound on the estimation uncertainty for each target  $m \in \mathcal{M}$  in (7a); constraints in (7b) are the aforementioned dynamic and observation models, and how the estimation uncertainties of the targets evolve under the measurements.

Note that  $\gamma_m$  above is a key design parameter that greatly influence the system behavior. For instance, if  $\gamma_m$  is the minimum variance that can be achieved if  $N_0$  robots monitor one target at all time, then the required team size is at least  $N_0 M / (\max_m \{C_m\})$ . On the other hand, if  $\gamma_m$  is sufficiently large, one robot could suffice by observing each target in sequence. More numerical analyses are given in Sec. IV.

**Remark 3.** The objective in (7) boils down to extending the longevity of each robot, e.g., when a UAV is inactive for the monitoring task, it can hover, glide or even park to charge battery until being active next time. ■

### III. PROPOSED SOLUTION

Due to the dynamic nature of the proposed problem, the proposed solution consists of two hierarchical components, i.e., the dynamic assignment of monitoring tasks based on

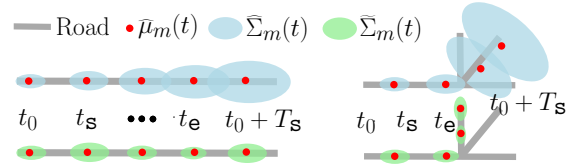


Fig. 2. Illustration of how the posteriori uncertainty  $\tilde{\Sigma}_m$  changes by (10), compared with the prior  $\hat{\Sigma}_m$ : on one road (**Left**) and an interaction (**Right**).

the estimated target behaviors with a minimum fleet size; and the optimal control of each robot given the assigned targets to ensure the estimation uncertainty is bounded. Both components are executed in a receding horizon fashion, but with different frequencies and horizons.

#### A. Dynamic Assignment of Monitoring Tasks

##### 1) Prior and Posteriori Estimation of Target Trajectory:

Let the estimated state of target  $m$  be  $(\mu_m(t_0), \Sigma_m(t_0))$  at time  $t_0 \geq 0$ . Then, assuming that *no* measurements are obtained during the future period  $t \in [t_0, t_0 + T_s]$  for  $T_s > 0$ , the predicted evolution of mean and covariance is given by:

$$\begin{aligned} \hat{\mu}_m(t+1) &= G \hat{\mu}_m(t), \\ \hat{\Sigma}_m(t+1) &= G \hat{\Sigma}_m(t) G + W_m, \\ (\hat{\mu}_m(t+1), \hat{\Sigma}_m(t+1)) &= \mathcal{G}_{\text{apx}}(\{\hat{\mu}_m^k(t)\}), \end{aligned} \quad (8)$$

where  $(\hat{\mu}_m, \hat{\Sigma}_m)$  are the predicted mean and covariance;  $W_m$  is the uncertainty from (2); the last case is when target  $m$  is predicted to cross an intersection from (6). Thus, the prior estimation of target  $m$  during  $[t_0, t_0 + T_s]$  is denoted by  $\hat{\mu}_m(t)$  and  $\hat{\Sigma}_m(t)$ ; and  $\hat{\mathbf{y}}_m(t) \triangleq (\hat{\mu}_m(t), \hat{\Sigma}_m(t))$ .

Then, assume that one robot  $n$  actively monitors target  $m$  during the period  $[t_s, t_e]$ , where  $t_0 + T_s > t_e > t_s > t_0$ . The resulting change in its covariance is given by the EKF [7], [22], [23] via the prediction and update as follows:

$$\begin{aligned} \Sigma'_m(t+1) &= G \tilde{\Sigma}_m(t) G + W_m, \quad \forall t \in [t_s, t_e]; \\ \tilde{\Sigma}_m(t+1) &= \Sigma'_m(t) - K_m(t) H_m(t) \Sigma'_m(t); \\ K_m(t) &= \Sigma'_m(t) H_m^T(t) R_m^{-1}(t) \\ R_m(t) &= H_m(t) \Sigma'_m(t) H_m^T(t) + V(\mathbf{x}_n(t), \hat{\mu}_m(t)); \\ H_m(t) &= \nabla_{\mathbf{y}} h(\mathbf{x}, \mathbf{y})|_{(\mathbf{x}_n(t), \hat{\mu}_m(t))}; \end{aligned} \quad (9)$$

where  $\mathbf{x}_n(t)$  is the robot state;  $\tilde{\Sigma}_m(t)$  is the posteriori estimation uncertainty assuming that robot  $n$  starts monitoring target  $m$  from time  $t_s$ ;  $H_m(t)$  is the linearized observation model of  $h(\cdot)$  given  $\mathbf{x}_n(t)$  and  $\hat{\mu}_m(t)$ ;  $V(\cdot)$  is the state-dependent uncertainty during observation from (3); and  $\tilde{\Sigma}_m(t_s)$  is initialized according to the prior estimation. Afterwards, for  $t > t_e$  without measurements,  $\tilde{\Sigma}_m(t)$  evolves in the same way as (8). In other words, the posteriori estimation  $\tilde{\Sigma}_m(t)$  is computed in three segments as follows:

$$\begin{aligned} \tilde{\Sigma}_m(t) &= \hat{\Sigma}_m(t), \quad \forall t \in [t_0, t_s]; \\ \tilde{\Sigma}_m(t+1) &= \rho_{\text{post}}(\tilde{\Sigma}_m(t), \mathbf{x}_n(t)), \quad \forall t \in [t_s, t_e]; \\ \tilde{\Sigma}_m(t+1) &= \rho_{\text{pri}}(\tilde{\Sigma}_m(t)), \quad \forall t \in [t_s, t_0 + T_s]; \end{aligned} \quad (10)$$

where functions  $\rho_{\text{post}}(\cdot)$  and  $\rho_{\text{pri}}(\cdot)$  encapsulate (9) and (8), respectively. An illustration is given in Fig. 2.

2) *Formulation of Task Assignment Problem:* Clearly, the computation of  $\hat{\Sigma}_m(t)$  above depends on the robot trajectory  $\mathbf{x}_n(t)$ . Due to the inherent combinatorial complexity of the assignment process, an approximation method is introduced here to bound the duration required for robot  $n$  to monitor target  $m$  without specifying the trajectory  $\mathbf{x}_n(t)$ , such that the uncertainty measure  $\mathbf{det}(\hat{\Sigma}_m(t))$  above is reduced below  $\gamma_m$  in (7a). Instead, the exact optimization of robot trajectories is performed in Sec. III-B afterwards. More specifically, consider first the following assumption:

**Assumption 1.** For arbitrary  $(\hat{\mu}_m(t), \hat{\Sigma}_m(t))$  above such that  $\hat{\Sigma}_m(t_0) < \gamma_m$  and a large enough  $T_s$ , there always exists a period  $[t_s, t_e]$  and a choice of  $\mathbf{x}_n(t)$  such that  $\mathbf{det}(\hat{\Sigma}_m(t)) < \gamma_m, \forall t \in [t_e, t_0 + T_s]$ . ■

The above assumption basically states that *one robot is sufficient to monitor one target while ensuring that its uncertainty stays below the given bound*. This assumption stems from the application that a few UAVs are deployed to monitor a large number of dynamic targets. If more than one robot is required to monitor even a single target, then there are more robots than targets, which is not the focus here.

More specifically, the starting time  $t_s$  is chosen between  $[t_0, \bar{t}_s]$ , where  $\bar{t}_s$  is the *latest* starting time such that:  $\hat{\Sigma}_m(\bar{t}_s - 1) < \gamma_m$  and  $\hat{\Sigma}_m(\bar{t}_s) \geq \gamma_m$  hold. Given a choice of  $t_s$ , the ending time  $t_e$  is minimized to monitor more targets, i.e., as the *earliest* instant such that:  $\mathbf{x}_n(t) = \min_{\mathbf{x}_n(t)} \mathbf{det}(\hat{\Sigma}_m(t + 1)), \forall t \in [t_s, t_e]$  and  $\mathbf{det}(\hat{\Sigma}_m(t_s + T_s)) < \gamma_m$  hold, where  $\hat{\Sigma}_m(t)$  is determined by (10); the first condition enforces that robot  $n$  chooses the best observation position to reduce the uncertainty the most during each step of the monitoring period  $(t_s, t_e)$ ; the second condition requires that the terminal uncertainty is below  $\gamma_m$ . If both conditions hold, the choice of schedule  $(t_s, t_e)$  is called feasible. Thus, for a robot-target pair  $(n, m)$ , the time to start monitoring  $t_s$  has a window  $[t_0, \bar{t}_s]$ . The monitoring plan for each robot  $n$  is defined as:

$$\tau_n \triangleq (t_s^{m_1}, t_e^{m_1}) \dots (t_s^{m_L}, t_e^{m_L}), \quad (11)$$

where  $(t_s^{m_\ell}, t_e^{m_\ell})$  is the schedule for monitoring target  $m_\ell \in \mathcal{M}, \forall \ell = 1, \dots, L$ . The plan length is determined by the elapsed time  $t_e^{m_L} - t_s^{m_1}$ . A complete plan is *feasible* if all contained schedules are feasible.

**Problem 1.** Given the prediction of all targets  $\{(\hat{\mu}_m(t), \hat{\Sigma}_m(t)), \forall m\}$ , determine first (I) the minimum number of active robots  $\bar{N}(t)$ ; and then (II) a feasible monitoring plan  $\tau_n$  for each  $n \in \bar{N}(t)$ , such that the maximum plan length of all robots is minimized. ■

3) *Solution Based on m-MVRPTW:* The solution to Problem 1 is based on formulating it as a flexible-size multiple vehicle routing problem with time windows (m-MVRPTW) [24], [25]. In particular, three steps are followed: (I) A service location is created for each target  $m \in \mathcal{M}$  at  $p_m \triangleq \hat{\mu}_m(\hat{t}_m)$  for  $\hat{t}_m = (t_0^m + \bar{t}_s^m)/2$ , with a *strict* time window  $w_m \triangleq [t_0^m, \bar{t}_s^m]$ . Its service time  $T_m$  is given

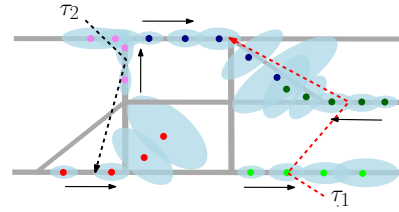


Fig. 3. Illustration of the assignment algorithm given the predicted trajectories with covariances, i.e., 5 targets monitored by 2 robots.

by:  $T_m \triangleq \max_{t'_s \in [t_0^m, \bar{t}_s^m]} |t'_e - t'_s|$ , where  $t'_s$  is a valid starting time and  $t'_e$  is the associated ending time;  $T_m$  is the longest duration for target  $m$ ; (II) The transition time from target  $m_1$  to target  $m_2$  is computed as:  $T_{m_1 m_2} \triangleq \max_{t_s^{m_1}, t_s^{m_2}} \{ \|\hat{\mu}_{m_1}(t_s^{m_1}) - \hat{\mu}_{m_2}(t_s^{m_2})\| / v_n \}$ , where  $t_s^{m_1}$  is the allowed ending time for target  $m_1$ ;  $t_s^{m_2}$  is the allowed starting time for target  $m_2$ ; and  $v_n$  is the average velocity of robot  $n$ ; (III) The m-MVRPTW problem is formulated as  $M$  service locations at  $\{p_m, \forall m\}$  with a strict time window  $\{w_m, \forall m\}$ , among which the time matrix is given by  $\{T_{m_1} + T_{m_1 m_2}, \forall m_1, m_2\}$ . Each robot can depart from its current location and end anywhere. The proposed algorithm gradually adds robots to the active fleet, ranked by their minimum distance to any target. Given a fixed fleet size, the standard MVRPTW can be solved by any solver, e.g., OR-TOOLS [26]. The outputs are the set of required robots  $\bar{N} \subseteq \mathcal{N}$  with cardinality  $\bar{N}$ ; and their schedule  $\tau_n$  from (11), with the sequence of assigned targets  $\mathbf{m}_n \triangleq m_1 \dots m_L$  and the associated service schedule  $\mathbf{t}_n, \forall n \in \bar{N}$ .

### B. Informative Trajectory Planning via Distributed NMPC

Given the schedule  $\{\tau_n, \forall n \in \bar{N}\}$ , each robot  $n$  should optimize its actual trajectory to execute this schedule, while ensuring the uncertainty constraints. The main challenges are two-fold: (i) both the robot dynamics and the uncertainty bounds are nonlinear and (ii) the target behavior is unknown and only observed online. Thus, the framework of distributed nonlinear model predictive control (NMPC) is applied [27], [28], for the trajectory optimization problem below.

**Problem 2.** Given the schedule  $\tau_n$ , compute the optimal control and action  $(\mathbf{u}_n, \mathbf{s}_n)$  for each active robot  $n \in \bar{N}$ , such that the uncertainties of all assigned targets  $\mathbf{m}_n$  are bounded and the control cost is minimized. ■

Consider another planning horizon  $T_c > 0$ , which is different and shorter than the horizon  $T_s$  of task assignment in Sec. III-A. Given the prediction of target behaviors  $\{(\hat{\mu}_m(t_0), \hat{\Sigma}_m(t_0)), \forall m \in \mathbf{m}_n\}$  at time  $t_0$ , the NMPC problem is formulated for each robot  $n \in \bar{N}$  as follows:

$$\begin{aligned} \min_{(\mathbf{u}_n, \mathbf{s}_n)} \quad & \sum_{t \in \mathcal{T}_c} \Upsilon(\mathbf{x}_n(t), \mathbf{u}_n(t)) \\ \text{s.t.} \quad & \mathbf{x}_n(t+1) \triangleq f(\mathbf{x}_n(t), \mathbf{u}_n(t)); \\ & |\mathbf{s}_n(t)| < C_n; \mathbf{det}(\hat{\Sigma}_m(t)) < \gamma_m; \\ & \hat{\Sigma}_m(t+1) = \rho(\hat{\Sigma}_m(t), \mathbf{x}_n(t), \hat{\mathbf{y}}_m(t_0)); \end{aligned} \quad (12)$$

where  $\mathcal{T}_c \triangleq \{t_0, \dots, t_0 + T_c\}$ ;  $\Upsilon: \mathcal{W} \times \mathcal{U} \rightarrow \mathbb{R}^+$  is a general function to measure the control cost given a choice of control

---

**Algorithm 1:** Uncertainty-bounded Monitoring
 

---

**Input:**  $\mathcal{M}, \mathcal{R}, \{\hat{\mathbf{y}}_m(0)\}, T_s, T_c$ .  
**Output:**  $\bar{\mathcal{N}}(t), \{\hat{\mathbf{y}}_m(t)\}, \{(\mathbf{u}_n, \mathbf{s}_n)\}$ .  
 1 **while not terminated do**  
 2     Compute  $\{\hat{\mathbf{y}}_m(t)\}$  by (8);  
 3     Formulate m-MVRPTW by Sec. III-A.3;  
 4     Derive  $\bar{\mathcal{N}}(t)$  and  $\{(\mathbf{m}_n, \mathbf{t}_n)\}$ , e.g., via [26];  
 5     Formulate NMPC (12) for each robot  $n \in \bar{\mathcal{N}}(t)$ ;  
 6     Derive solution  $\{(\mathbf{u}_n, \mathbf{s}_n)\}$ , e.g., via [30];  
 7     Each robot  $n \in \bar{\mathcal{N}}(t)$  applies  $(\mathbf{u}_n(t), \mathbf{s}_n(t))$ ;  
 8     Make observations  $\{\mathbf{z}_{nm}\}$ ;  
 9     Update  $\{\hat{\mathbf{y}}_m(t+1)\}$  by (14);  
 10     $t \leftarrow t + 1$ ;  


---

input at certain states; the uncertainty measure  $\tilde{\Sigma}_m(t)$  is the posteriori estimation from (9); function  $\rho$  summarizes the updating rules from (10) given the current prediction  $\hat{\mathbf{y}}_m(t_0)$ ; and these constraints hold for all  $t \in \mathcal{T}_c$  and  $m \in \mathbf{m}_n$ .

Note that the above constrained optimization is different from the standard NMPC in [27] due to the variable  $\mathbf{s}_n$  as the monitoring action to decide which targets to monitor, which is discrete thus yielding a combinatorial complexity over  $\mathbf{m}_n$ . Instead, a hierarchical structure is followed to decouple the optimization over  $\mathbf{s}_n$  and  $\mathbf{u}_n$ . Specifically, given the current states of robot  $n$  and all targets in  $\mathbf{m}_n$  at each time step  $t_0$ , the *first*  $C_n$  targets in  $\mathbf{m}_n$  with the largest predicted prior uncertainty  $\hat{\Sigma}_m$  within  $\mathcal{T}_c$ , i.e.,

$$\mathbf{s}_n(t) = \mathbf{max\_k} \left\{ \max_{k=C_n, m \in \mathbf{m}_n} \det(\hat{\Sigma}_m(t)) \right\}, \forall t \in \mathcal{T}_c; \quad (13)$$

where  $\mathbf{max\_k}(\cdot)$  is the operator for choosing the  $k$  largest elements in a set;  $\mathbf{s}_n(t)$  is kept constant in the optimization (12). Note that if  $|\mathbf{m}_n| \leq C_n$ , all targets are monitored. Once  $\mathbf{s}_n$  is determined, the optimization (12) is transformed into a nonlinear optimization problem, where the variables are: (i) the robot state  $\mathbf{x}_n(t)$  and control input  $\mathbf{u}_n(t)$  at each step  $t \in \mathcal{T}_c$ ; and (ii) the posteriori uncertainty  $\tilde{\Sigma}_m(t)$  at each step  $t \in \mathcal{T}_c$  and for each target  $m \in \mathbf{s}_n$ . The determinant  $\det(\tilde{\Sigma}_m(t))$  of a square matrix can be computed *analytically* [29], i.e., as a polynomial in the traces of the powers of  $\tilde{\Sigma}_m(t)$ . Thus, the formulated problem can be solved by off-the-shelf solvers, e.g., CasADi [30]. The results are the desired control  $\mathbf{u}_n(t)$  in the planning horizon.

**Remark 4.** Related work often assumes either a discretized control space [4], [7], [12], [31] or pre-defined motion primitives [16], [17], [20], which cannot be applied here for trajectory optimization under dynamic constraints. ■

### C. Overall Framework

As summarized in Alg. 1, the above two modules are executed online in a receding horizon fashion with different frequencies. More specifically, at each time step  $t > 0$ , the prior estimation  $\hat{\mathbf{y}}_m(t)$  is computed via (8) for each target. Based on this estimation, the monitoring tasks are assigned to a subset of robots by solving (8), yielding the planned schedule  $\tau_n$ . Then, the optimization for control and

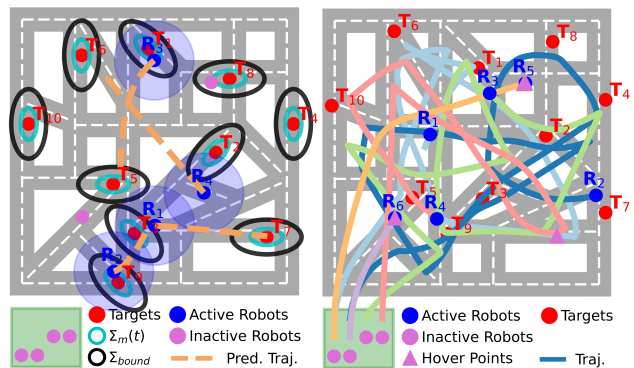


Fig. 4. **Left:** Snapshot at  $t = 52s$  where 4 robots are active. **Right:** Final trajectories of all robots with marked status.

monitoring action is formulated via (12) given  $\tau_n$  and  $\hat{\mathbf{y}}_m(t)$ , yielding the optimal control sequence  $\mathbf{u}_n$  and the monitored action  $\mathbf{s}_n$ . Afterwards, each robot moves by executing the next control input  $\mathbf{u}_n(t)$ , while obtaining possible new measurements  $\{\mathbf{z}_{nm}, \forall m \in \mathbf{s}_n\}$ . Given these measurements, the posteriori estimation  $\hat{\mathbf{y}}_m(t+1) = (\hat{\mu}_m(t+1), \hat{\Sigma}_m(t+1))$  is updated for each observed target as follows:

$$\begin{aligned} \hat{\mu}_m(t+1) &= \hat{\mu}_m(t) + K_m(\mathbf{z}_{nm} - h(\mathbf{x}_n(t), \hat{\mu}_m(t))), \\ \hat{\Sigma}_m(t+1) &= \rho_{\text{post}}(\hat{\Sigma}_m(t), \mathbf{x}_n(t)), \end{aligned} \quad (14)$$

where  $K_m$  is computed as in (9) and  $\rho_{\text{post}}(\cdot)$  defined in (10). This procedure is repeated until the system is terminated. It is worth noting that while the trajectory optimization is executed at every time step locally by each robot, the module of task assignment is executed at a much lower frequency via a central coordinator, which can be periodic or event-based, e.g., each time the set  $\mathbf{s}_n$  computed in (13) changes.

Lastly, note that even as approximated sub-problems, both Problems 1 and 2 have high computation complexity. In particular, Problem 1 remains NP-hard [24] as being a more general problem than the classic MVRP, while there is no guarantee that Problem 2 converges to the global optimal solution [32] as being nonlinear and strongly non-convex. Nonetheless, for moderate problem sizes e.g., 10 robots and 10 targets, these problems can be solved in 0.89s and 0.54s. More details are given in the next section.

## IV. NUMERICAL EXPERIMENTS

Extensive large-scale simulations are presented in this section. The proposed method is implemented in Python3 and tested on a laptop with an Intel Core i7-1280P CPU. Simulation videos can be found in the supplementary files.

### A. Description

As shown in Fig. 4, consider 10 targets moving on the road-networks with a size of  $10m \times 10m$ , and there are 10 robots in the fleet. Each robot follows the standard unicycle dynamics for UAV flying at a fixed height. The observation model is the commonly-used range-and-bearing model [10], [12], where the uncertainty  $V$  in (3) grows linearly in range and bearing. The exact mathematical model is omitted here due to limited space. The limited range in (4) is set to  $R_n = 1.5m$  and the capacity  $C_n = 5$  uniformly for all

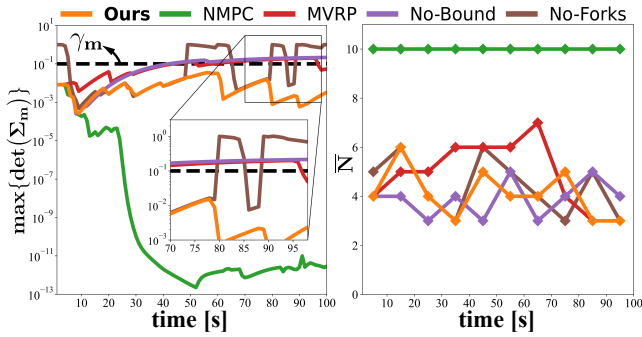


Fig. 5. Comparison against four baselines: the maximum uncertainty of all targets (**Left**) and the number of active robots within the fleet (**Right**).

robots. Initially, the target states are chosen *randomly* along the roads with a small uncertainty, while the robots start from the base. During execution, the targets choose the next road randomly at intersections, which are unknown to the robots. The threshold for uncertainty is set to  $\gamma_m = 0.1$  for all targets. Moreover, the horizon for dynamic assignment is set to  $T_s = 50$  and updated every  $10s$ , while the NMPC planning horizon is set to  $T_c = 10$  and updated every  $0.1s$ . The simulation is terminated at  $100s$ .

### B. Results

The final results are shown in Fig. 1 and 4. The assignment method via  $m$ -MVRPTW takes in average  $0.85s$ , which generates a varying number of active robots (minimum 3 and maximum 6). For instance, it is enough at  $31s$  for 3 robots to monitor all 10 targets, while the rest becomes inactive. At  $50s$ , more robots are recruited given the predicted uncertainty. Moreover, the average computation time of NMPC is  $0.57s$  and the resulting trajectory often covers multiple targets simultaneously. The estimation uncertainties of all targets are below the specified bound  $0.1$ , while the average number of active robots is  $4.3$  and the average number of targets assigned to each robot is  $2.3$ . The final trajectories for all the robots are depicted in Fig. 4, which switches between being active and inactive.

### C. Comparisons

To further validate the effectiveness of the proposed framework (as **Ours**), a quantitative comparison is conducted against four baselines (two ablation studies and two common approaches): (i) **NMPC**, where the first layer of task assignment is skipped and each robot is assumed to track one single target at each time step; (ii) **MVRP**, where the proposed assignment algorithm remains but the trajectory optimization is replaced by a simple strategy that tracks the assigned targets in order; (iii) **No-Bound**, where the objective in (12) is changed to minimize the summed uncertainties of all targets (see [7], [8], [10]), i.e., not as a hard constraint; (iii) **No-Forks**, where the target model follows (2) but without considering the unknown behavior at intersections. The compared metrics are the planning time, the success rate, the maximum uncertainty of all targets, and finally the average number of active robots.

The results are summarized in Fig. 5. It can be seen that both the proposed method and the NMPC can ensure

TABLE I  
SCALABILITY ANALYSIS RESULTS

Param.	Values	Success Rate [%]	MVRP Time [s]	NMPC Time [s]	Average $\bar{N}$
Nominal*		100.0	0.80	0.57	5.0
$M$	20	100.0	1.37	0.54	10.3
	100	100.0	14.2	0.56	56.3
$\gamma_m$	$10^{-3}$	98.3	1.24	0.53	5.6
	$10^{-5}$	83.8	4.43	0.67	7.7
$C_n$	3	100.0	0.74	0.40	6.6
	7	100.0	0.67	0.83	3.3

\*Nominal case:  $M = 10$ ,  $\gamma_m = 10^{-1}$ , and  $C_n = 5$ .

that the estimation uncertainty is below the given bound at all time, while others fail. However, the NMPC requires that all robots are active at all time, while ours requires in average 4 robots. In contrast, the MVRP method optimizes the task assignment, but the estimation uncertainty might exceed the bound as the motion strategy cannot react to all assigned targets. Similarly, despite of faster planning, the No-Bound method loses the guarantee on uncertainty as the hard constraints are relaxed. At last, the necessity of modeling intersection is clear with the No-Forks method, where the uncertainty increases drastically above the bound when the targets cross intersections. Our method however can predict this change and monitor these targets in advance.

### D. Scalability Analysis

Scalability of the proposed algorithm is analyzed w.r.t. three aspects: the number of targets  $M$ , the upper bound of uncertainty  $\gamma_m$ , and the capacity  $C_n$ . The recorded metrics are similar to the previous part, including the detailed planning time for MVRP and NMPC. The results are summarized in Table I, where the nominal setup is from Sec. IV-B. Notably, when  $M$  is increased to 20 and 100, the success rate remains 100%. Planning time of MVRP is increased significant as the number of tasks has increased to 100, while the NMPC takes roughly the same time as each robot solve it locally. It is worth noting that the ratio between the average number of active robots and the total number of targets remains constant, meaning that the *fleet efficiency* does not degrade as  $M$  increases. Furthermore, when the threshold  $\gamma_m$  is decreased to  $10^{-3}$  and even  $10^{-5}$ , the success rate decreases as the NMPC is harder to solve with a tighter bound, yielding a much higher number of active robots. Lastly, when  $C_n$  is decreased to 3, the planning time of NMPC is decreased as the number of constraints in (12) is reduced, and the average number of active robots is increased as one robot can monitor less targets simultaneously. The opposite holds for  $C_n = 7$ , where the average number of active robots is as low as 3.3 out of 10 in total.

## V. CONCLUSION

This work proposes an online task and motion coordination algorithm for large-scale monitoring tasks over road-networks. It ensures an explicitly-bounded estimation uncertainty for the targets, while employing a minimum number of heterogeneous robots. Future work involves a time-varying number of targets and complex tasks.

## REFERENCES

- [1] R. Khodayi-mehr, W. Aquino, and M. M. Zavlanos, "Model-based active source identification in complex environments," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 633–652, 2019.
- [2] R. Khodayi-mehr, Y. Kantaros, and M. M. Zavlanos, "Distributed state estimation using intermittently connected robot networks," *IEEE transactions on robotics*, vol. 35, no. 3, pp. 709–724, 2019.
- [3] M. Guo and M. M. Zavlanos, "Multirobot data gathering under buffer constraints and intermittent communication," *IEEE transactions on robotics*, vol. 34, no. 4, pp. 1082–1097, 2018.
- [4] M. Tzes, N. Bousias, E. Chatzipantazis, and G. J. Pappas, "Graph neural networks for multi-robot active information acquisition," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3497–3503.
- [5] S. Kalluraya, G. J. Pappas, and Y. Kantaros, "Multi-robot mission planning in dynamic semantic environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1630–1637.
- [6] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.
- [7] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *IEEE International conference on robotics and automation (ICRA)*, 2014, pp. 6447–6454.
- [8] P. Dames, M. Schwager, V. Kumar, and D. Rus, "A decentralized control policy for adaptive information gathering in hazardous environments," in *IEEE Conference on Decision and Control (CDC)*, 2012, pp. 2807–2813.
- [9] T. H. Chung, J. W. Burdick, and R. M. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2416–2422.
- [10] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018.
- [11] J. Le Ny and G. J. Pappas, "On trajectory optimization for active sensing in gaussian process models," in *IEEE Conference on Decision and Control (CDC)*, 2009, pp. 6286–6292.
- [12] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multi-robot information gathering," in *Robotics: Science and Systems*, 2019, pp. 22–26.
- [13] J. Chen, Z. Tang, and M. Guo, "Accelerated k-serial stable coalition for dynamic capture and resource defense," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 443–450, 2024.
- [14] Y. Sung, A. K. Budhiraja, R. K. Williams, and P. Tokekar, "Distributed assignment with limited communication for multi-robot multi-target tracking," *Autonomous Robots*, vol. 44, no. 1, pp. 57–73, 2020.
- [15] S. Chopra, G. Notarstefano, M. Rice, and M. Egerstedt, "A distributed version of the hungarian method for multirobot assignment," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 932–947, 2017.
- [16] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3067–3072.
- [17] L. Zhou and P. Tokekar, "Sensor assignment algorithms to improve observability while tracking targets," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1206–1219, 2019.
- [18] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1540–1553, 2017.
- [19] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Resilient active target tracking with multiple robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 129–136, 2018.
- [20] L. Zhou and V. Kumar, "Robust multi-robot active target tracking against sensing and communication attacks," *IEEE Transactions on Robotics*, 2023.
- [21] Y. Kantaros and G. J. Pappas, "Scalable active information acquisition for multi-robot systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 7987–7993.
- [22] M. Athans, "On the determination of optimal costly measurement strategies for linear stochastic systems," *Automatica*, vol. 8, no. 4, pp. 397–412, 1972.
- [23] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.
- [24] M. Polacek, R. F. Hartl, K. Doerner, and M. Reimann, "A variable neighborhood search for the multi depot vehicle routing problem with time windows," *Journal of heuristics*, vol. 10, pp. 613–627, 2004.
- [25] P. P. Repoussis and C. D. Tarantilis, "Solving the fleet size and mix vehicle routing problem with time windows via adaptive memory programming," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 695–712, 2010.
- [26] G. OR-Tools, <https://github.com/google/or-tools>.
- [27] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*. Birkhäuser, 2012, vol. 26.
- [28] Y. Kang and J. K. Hedrick, "Linear tracking for a fixed-wing uav using nonlinear model predictive control," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1202–1210, 2009.
- [29] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
- [30] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [31] M. Guo and D. V. Dimarogonas, "Task and motion coordination for heterogeneous multiagent systems with loosely coupled local tasks," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 797–808, 2016.
- [32] S. J. Wright, *Numerical optimization*, 2006.