DECENTRALIZED COORDINATION OF MULTI-ROBOT NETWORKS FOR ACTIVE TARGET TRACKING

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master

by

Kyung Rak Jang

August 2023

ABSTRACT

This paper introduces a decentralized framework for optimizing the coordination of robot networks to track multiple moving targets in applications like security and surveillance. The problem of network optimization is proven to be NP-hard, highlighting the need for efficient solutions. The proposed framework presents two novel decentralized coordination methods: the group-based algorithm and the bundle-based algorithm. These methods aim to achieve adaptive and conflict-free target assignments, with the bundle-based algorithm providing more effective coordination and guaranteeing a $\frac{1}{2}$-approximation in the worst-case scenario. Simulation results demonstrate that the proposed approaches outperform existing algorithms, achieving performance close to the optimal solution in significantly less time. Compared to EER control and PD control, the group-based assignment and control (GBAC) and the bundle-based assignment and control (BBAC) demonstrate superior performance due to their adaptive target assignments achieved through network coordination. Among the two methods, BBAC shows higher average target tracking rate (ATTR) and lower average robot traveling distance (ARTD), resulting in improved tracking efficiency. Physical experiments using a network of ground robots tracking human targets further validate the practicality of the proposed approach in real-world scenarios.

BIOGRAPHICAL SKETCH

Kyung-Rak Jang is a Master of Science graduate student in the Laboratory for Intelligent Systems and Controls(LISC) supervised by Professor Silvia Ferrari at Cornell University. He received B.S. in Mechanical Engineering from the University of Washington, Seattle. Upon graduating, he worked at TÜV SÜD as an Automotive engineer for three years. His research interests include autonomous systems, robotics, multi-agent systems.

This document is dedicated to all Cornell graduate students.

ACKNOWLEDGEMENTS

I am grateful to Professor Silvia Ferrari for her generous guidance and unwavering support during my study at Cornell University. Working under her supervision was a tremendous privilege. I would also like to express my gratitude to Professor Wendy Ju for serving as a committee member for my research. Additionally, I extend my thanks to Junyi Dong, Sushrut Surve, and Frank Kim for their valuable assistance in both my research and defense. Lastly, I am appreciative of the support provided by all the members of LISC, who have been instrumental in both my professional and personal endeavors.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The progress in robot sensing and mobility has made it possible to use multi-robot networks to track a greater number of moving targets in various applications such as security and surveillance. The essential objective of tracking is to maximize its quality through network coordination and control, which includes assigning specific targets to particular robots and determining the best states for tracking. While centralized strategies are optimal, they are not feasible for large-scale networks because they require excessive computation and communication. Thus, distributed methods are preferred because of their scalability, robustness, and efficiency. This chapter introduces a new distributed framework for coordinating a network of robots. The framework is designed to address the challenges of limited sensing and communication ranges and achieve conflict-free assignments through local communication.

1.1 Motivations

Multi-robot networks (MRNs) present a crucial set of problems in robotics, which have stimulated a range of research directions over the last decade due to the increasing complexity of envisioned missions [16, 22]. For example, these networks, comprising unmanned aerial vehicles (UAVs) and/or unmanned ground vehicles (UGVs), are being used to perform challenging tasks in area such as urban search and rescue [6, 1], automated warehouses [26, 9], information gathering [5, 30], and surveillance [36, 19]. Through intelligent coordination and autonomous control, MRNs can work simultaneously to reduce task completion time, share mission-critical data to enhance situational awareness, and create redundancy to improve fault tolerance. Therefore, MRNs are more promising than single robot systems.

To address the multi-target tracking problem in MRNs, several approaches have been proposed. One category of methods is based on centralized optimization, which involves solving a combinatorial optimization problem to assign targets to robots. For example, a graph-based approach has been proposed in [2], where a bipartite graph is constructed to represent the assignment problem and a minimum-cost flow algorithm is used to find the optimal assignment. Another centralized approach is based on particle swarm optimization [13], where a global optimization algorithm is used to optimize the assignment of targets to robots. On the other hand, decentralized approaches have also been developed to address the multi-target tracking problem in MRNs. One popular approach is based on auction algorithms [3], where robots bid on targets and the targets are assigned to the robots with the highest bids. A distributed auction algorithm has been proposed in [27], where the auction process is carried out locally at each robot and a consensus algorithm is used to ensure that the final assignment is globally optimal. Overall, the multi-target tracking problem in MRNs is a challenging research topic and various approaches have been proposed to address it. The choice of method depends on the specific application and requirements, as well as the computational resources available.

1.2 Challenges

This paper focuses on addressing the challenges of coordinating MRNs in situations where the number of targets exceeds the number of robots. This is a challenging scenario as it requires simultaneously determining the target assignment and controlling the robots in real-time. Some existing methods proposed in [30, 34, 33] converts the problem into an integral linear programming (ILP) formulation, which aims to assign the robots to targets and determine their control inputs. However, the limitation of the methods is that they rely on prespecified actions for the robot control, which may not be optimal or feasible in all situations. This may result in suboptimal tracking performance, especially in complex and dynamic environments where targets may move unpredictably. Another limitation is that these method use omnidirectional sensors and assume that the robot network has access to perfect information about the targets and the environment, which is often not the case in real-world scenarios. In practice, robots may have limited sensing and communication capabilities, which result in difficulty of real-world target detection, classification, and state estimation using robot onboard sensors. Furthermore, these methods are based on centralized approaches, which require a centralized controller to make decisions for the entire robot network. This may not be scalable or robust, especially for large-scale and distributed systems, where communication and computation resources are limited.

1.3 Contributions

This paper addresses the limitations of existing methods for the coordination of multi-robot networks problem by proposing a decentralized approach that maximizes network tracking quality in real-time for target assignment problem. In this paper, two novel methods, the group-based assignment and the bundle-based assignment, are developed to find adaptive target assignments through multi-hop communication. The paper also introduces an integrated pipeline for online sensing of target detection, classification, and state estimation, which relies on data from robot onboard vision and motion sensors. The numerical simulations demonstrate that the proposed approaches perform better than the pre-defined baseline methods for various communication ranges. The quantitative analysis also evaluates the effectiveness of different methods for target tracking in a decentralized network using two metrics: average target tracking rate (ATTR) and average robot traveling distance (ARTD). The results demonstrate that the proposed GBAC and BBAC methods exhibit higher average efficiency compared to EER and PD controls, leading to improved tracking efficiency. Furthermore, the computational complexity analysis shows that the proposed target assignment methods converge in a finite number of steps and have polynomial time complexity. Finally, physical experiments with a network of ground robots executing the distributed optimization in real-time provide additional evidence of the effectiveness of the proposed approach.

CHAPTER 2

PROBLEM FORMULATION AND ASSUMPTIONS

This paper considers the problem of coordinating a group of mobile robots, specifically unmanned ground vehicles (UGVs), to track multiple moving targets for security and surveillance purposes. The workspace denoted by $\mathcal{W} \subset \mathbb{R}^2$ is assumed to be a closed and bounded two-dimensional (2D) space, populated with N number of robots denoted by $\mathcal{N} = \{1, \dots, N\}, N \in \mathbb{N}^+$. As shown in Fig. 1, the inertial frame $\mathcal{F}_\mathcal{W}$ is fixed with respect to the world and used as the reference frame for the robots. Each robot i has a moving Cartesian frame $\mathcal{F}_{\mathcal{A}_i}$ that is embedded in the robot and moves with it. The state of each robot is represented by a state vector $\mathbf{s}_i = [x_i \quad y_i \quad \theta_i]^T$, where x_i and y_i are the 2D coordinates of the robot in the moving Cartesian frame $\mathcal{F}_{\mathcal{A}_i}$ and θ_i is its orientation. The unicycle motion model [24, 11] is used to describe the motion of each robot,

$$\dot{\mathbf{s}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} v_i \cos\theta_i \\ v_i \sin\theta_i \\ \omega_i \end{bmatrix} = \mathbf{f}(\mathbf{s}_i, \mathbf{u}_i), \quad \forall i \in \mathcal{N} \tag{2.1}$$

with the control vector $\mathbf{u}_i = [v_i \quad w_i]^T \in \mathbb{R}^2$ representing its linear and angular velocities, respectively. Assuming a constant sampling interval $\Delta t \in \mathbb{R}^+$, the robot state and control vector at time $k\Delta t$ can be represented by $\mathbf{s}_i(k) = \mathbf{s}_i(k\Delta t)$ and $\mathbf{u}_i(k) = \mathbf{u}_i(k\Delta t)$. Then, the state and control of the robot network can be written as $\mathbf{s}(k) = [\mathbf{s}_1^T(k) \quad \dots \quad \mathbf{s}_N^T(k)]^T \in \mathbb{R}^{3N}$ and $\mathbf{u}(k) = [\mathbf{u}_1^T(k) \quad \dots \quad \mathbf{u}_N^T(k)]^T \in \mathbb{R}^{2N}$, respectively.

Let $\mathcal{M} = \{1, \dots, M\}$, $M \in \mathbb{N}^+$ denote the index set of dynamic targets, where M is the total number of targets, and the number of targets is no less than the number of robots in a MRN, $(M > N)$. Assuming a unique tar-

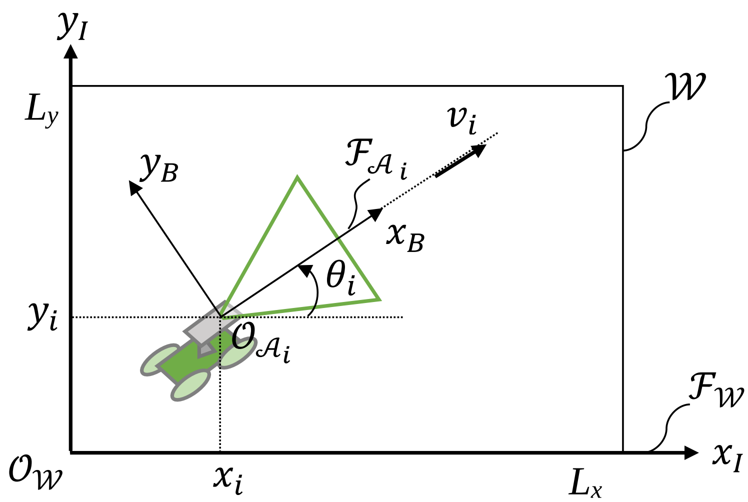

Figure 2.1: Definition of the state of a robot

get index j, $j \in \mathcal{M}$, the state of target j at time $k\Delta t$ can be represented by $\mathbf{x}_j(k) = [x_j(k) \quad y_j(k) \quad v_{x,j}(k) \quad v_{y,j}(k)]^T \in \mathbb{R}^4$, where $x_j(k)$, $y_j(k)$ are the positions and $v_{x,j}(k)$, $v_{y,j}(k)$ are the velocities with respect to $\mathcal{F}_{\mathcal{W}}$. The position and velocity of all targets at a particular time k are represented by the state vector $\mathbf{x}(k)$. The state vector is a concatenation of individual target state vectors $\mathbf{x}_1(k)$ to $\mathbf{x}_M(k)$. Each target's velocity is assumed to be constant and is affected by additive, zero-mean Gaussian process noise $\mathbf{w}(k)$. The target's motion model at any discrete time k is given by

$$\mathbf{x}_j(k) = \mathbf{F}\mathbf{x}_j(k-1) + \mathbf{w}(k), \quad \mathbf{w}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2.2}$$

where $\mathbf{F}$ is a known state transition matrix and $\mathbf{Q}$ is the covariance matrix of the process noise.

The robots in the workspace observe the targets that are within their field-of-view (FOV), which is a compact subset of $\mathcal{W}$ represented by $\mathcal{S}i \subset \mathcal{W}, \forall i \in \mathcal{N}$. For estimating the position of targets, this paper proposes an online sensing approach to measure target positions in the inertial frame using RGB image, depth image, and robot localization. This approach differs from existing methods that use vision-based sensors which measure target positions in the camera frame or virtual image plane [12, 20, 37], leading to complex nonlinear measurement models. Equation 2.3 represents the measurement model used to estimate the state of the targets. Here, $\mathbf{z}_{i,j}(k)$ is the measurement of target j by robot i at time k, $\mathbf{x}_j(k)$ is the true state of target j at time k, and $\mathbf{v}_((k)$ is the measurement noise which is assumed to be Gaussian with covariance matrix $\mathbf{R}$. The matrix $\mathbf{H}$ maps the target state into the measurement space, and in this case, it simply selects $\mathbf{H} = [\mathbf{I}_2 \quad \mathbf{0}_2]$.

$$\mathbf{z}_{i,j}(k) = \mathbf{H}\mathbf{x}_j(k) + \mathbf{v}(k) \quad \text{if} \quad \mathbf{x}_j(k) \in \mathcal{S}_i(k) \tag{2.3}$$

The measurement model in Equation 2.3 is linear, which simplifies the estimation process. It allows the use of the Kalman filter [38], a popular recursive algorithm for estimating the state of dynamic systems. In particular, given the target motion model and measurement model, $\hat{\mathbf{x}}_{i,j}(k)$ can be recursively estimated by Kalman filtering using Equation 2.4. The Kalman filter operates by maintaining a predicted state estimate based on the previous estimate and the motion model, and updates this estimate with new measurements as they become available. The innovation term $\mathbf{e}_{i,j}(k)$ is the difference between the actual measurement and the predicted measurement based on the current state estimate, and is used to update the estimate with the Kalman gain matrix $\mathbf{K}(k)$.

$$\hat{\mathbf{x}}_{i,j}(k) = \begin{cases} \mathbf{F}\hat{\mathbf{x}}_{i,j}(k-1) + \mathbf{K}(k)\mathbf{e}_{i,j}(k) & \text{if } \mathbf{x}_j(k) \in \mathcal{S}_i(k) \\ \mathbf{F}\hat{\mathbf{x}}_{i,j}(k-1) & \text{if } \mathbf{x}_j(k) \notin \mathcal{S}_i(k) \end{cases} \tag{2.4}$$

In this paper, the tracking problem is discussed where the number of targets exceeds the number of robots ($M > N$), and not all targets can be tracked simultaneously. Therefore, network coordination and control are crucial in determining a valid target assignment and obtaining the most informative measurements.

Definition 1 (Valid Target Assignment) : Given M targets represented by the index set $\mathcal{M} = \{1, \ldots, M\}$, a valid target assignment to a network of N robots at any time k is defined as a collection of N subsets, $P(k) \triangleq \{P_1(k), \ldots, P_N(k)\}$, such that every element in $\mathcal{M}$ is included in one and only one subset in $P(k)$, i.e.,

$$P_i(k) \cap P_{i'}(k) = \emptyset, \text{ if } i \neq i', \text{ and } \bigcup_{i=1}^{N} P_i(k) = \mathcal{M} \tag{2.5}$$

The target assignment space, denoted by $\mathcal{P}$, is the family of all valid target assignments that satisfy (2.5). Because by definition $P_i(k) \cap P_{i'}(k) = \emptyset$, then no conflicts may arise and, thus, a valid assignment is also called a conflict-free assignment. Moreover, target assignment in this work may vary over time as a function of the robot and target states, which forms a distinct contrast to the existing work that assumes static target assignment [5, 10].

A utility function, $U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k))$, is introduced in this paper to measure the performance of tracking target. Assuming that the targets move independently of each other, the overall tracking quality of a multi-robot network (MRN) can be evaluated by summing up the tracking utility of all targets:

$$U_g \triangleq \sum_{i \in N} \left[\sum_{j \in P_i(k)} U_{i,j}(\mathbf{s}_i(k), \hat{\mathbf{x}}_{i,j}(k)) \right] \tag{2.6}$$

The objective of this paper is to determine the optimal target assignment by solving a novel network optimization problem that maximizes U_g. Therefore, it is assumed that the optimal network control is found simultaneously.

CHAPTER 3

DECENTRALIZED OPTIMIZATION FRAMEWORK

3.1 Online Sensing

Online sensing allows robots to gather information from their environment using various sensors and modalities, and use that information to perform tasks such as localization, detection, classification, and state estimation. By integrating multiple data modalities such as RGB, depth, and odometry data, robots can obtain a more complete and accurate understanding of their surroundings, and make more informed decisions based on that information. In this paper, three primary sub-tasks are introduced for the online sensing process as shown in Fig.3.1.

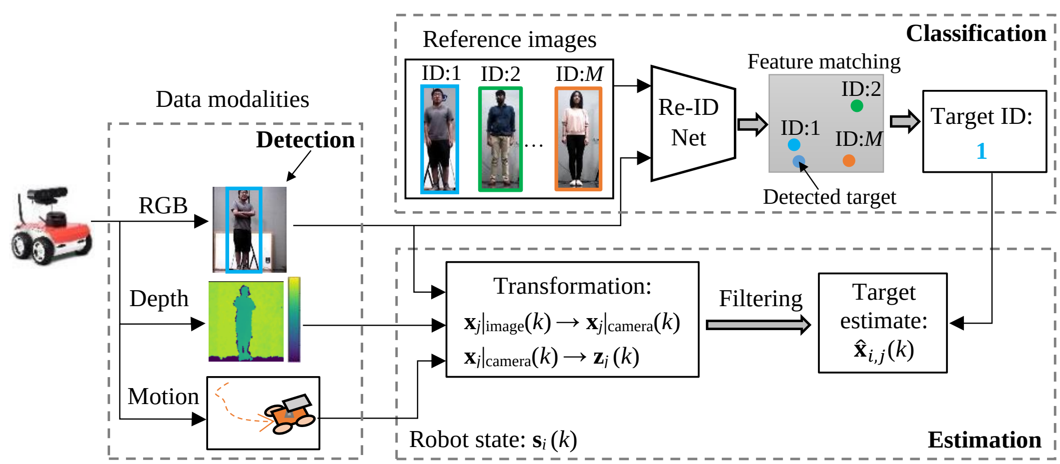

Figure 3.1: Online sensing pipeline for integrated target detection, classification, and state estimation

Firstly, Robot localization can be achieved by onboard odometry sensors or by overhead localization like motion capture systems. As robot localization can be measured through these existing techniques, this paper focuses on the integrated approach for target detection, classification, and state estimation.

Secondly, target detection and classification is performed using CNN-based architectures such as YOLO or Mask-RCNN, as mentioned in [29, 15]. These architectures are implemented on RGB images from robot cameras, and they output bounding box of human targets in the image frame, which are then used to query the target-classification pipeline. The goal of target classification is to associate the online detected targets with the pre-specified targets-of-interest they most resemble. This paper assumes that each target-of-interest is known to the MRN by a reference image with a unique ID, as shown in Fig.3.1. However, due to the dynamic characteristics of the mobile network, target classification needs to be robust to viewpoint changes to prevent frequent ID-switching as robots and targets move across the workspace. To achieve robust target classification, a deep neural network trained for person re-identification, known as the Re-ID Net, is adopted as described in [35, 23]. The Re-ID Net is invariant to the translation and rotation of image features, making it ideal for this task. During the tracking mission, the Re-ID Net implemented on each robot extracts convolutional features from the bounding box of the detected targets, which are then compared to the re-ID features of the reference images to find the closest match as the recognized target.

Lastly, the states of targets are estimated by fusing RGB data, depth data, and robot localization. This paper utilizes a ray tracing method that takes into account the camera's intrinsic and extrinsic parameters. The ray tracing method involves mapping the 2D pixel coordinate of the target in the image frame to the 3D camera frame using the camera's intrinsic parameters. Then, the 3D coordinate in the camera frame is transformed to the global inertial frame using the robot's localization information.

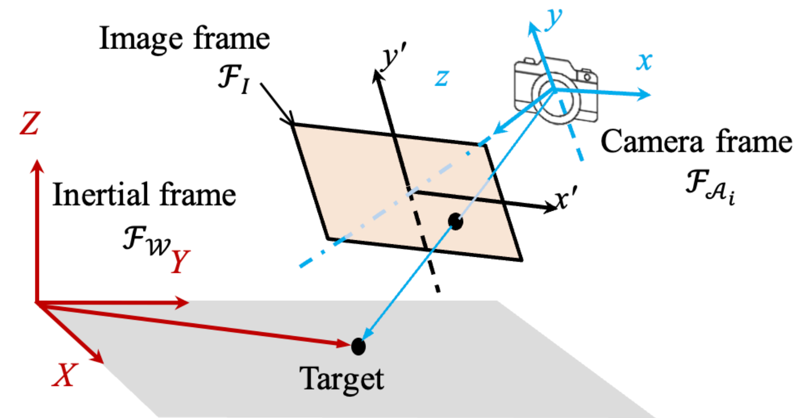

Figure 3.2: Target state estimation using ray tracing method

Let $\mathbf{x}_j|_{\text{image}}(k) \in \mathbb{R}^2$ be the $2D$ position of the j^{th} target with respect to the image reference frame, which can be approximated by the image coordinate at the center of the target's bounding box. Given $\mathbf{x}_i|_{\text{image}}$, the target depth, $d_j(k)$, can be obtained by extracting the corresponding pixel value in the depth image. Then, the target position with respect to the camera frame, $\mathcal{F}_{\mathcal{A}_i}$, is given by

$$\mathbf{x}_j|_{\text{camera}}(k) = d_j(k)\mathbf{M}^{-1}[\mathbf{x}_j|_{\text{image}}(k) \quad 1]^T \tag{3.1}$$

where $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix. The target measurement $\mathbf{z}_{i,j}(k)$ in the inertial frame $\mathcal{F}_W$ is obtained by mapping $\mathbf{x}_j|_{\text{camera}}(k)$ from $\mathcal{F}_{\mathcal{A}_i}$ to $\mathcal{F}_W$

$$\mathbf{z}_{i,j}(k) = \mathbf{R}_i(k)\mathbf{x}_j|_{\text{camera}}^T(k) + \mathbf{r}_i^T(k) \tag{3.2}$$

where $\mathbf{R}_i(k)$ and $\mathbf{r}_i(k)$ are camera extrinsic parameters estimated from the robot state vector, $\mathbf{s}_i(k) = [x_i(k) \quad y_i(k) \quad \theta_i(k)]^T$, as follows:

$$\mathbf{R}_i(k) = \begin{bmatrix} cos(\theta_i(k)) & -sin(\theta_i(k)) & 0 \\ sin(\theta_i(k)) & cos(\theta_i(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{r}_i(k) = [x_i(k) \quad y_i(k) \quad 0]^T \tag{3.3}$$

Compared to the true target state $\mathbf{x}_j(k)$, the measurement $\mathbf{z}_{i,j}(k)$ does not contain the velocity terms and is assumed to be subjected to white, additive Gaussian noise $\mathbf{v}(k)$, which yields

$$\mathbf{z}_{i,j}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_j(k) + \mathbf{v}(k) \tag{3.4}$$

as specified in (2.3). This linear measurement model is used to recursively estimate the target state $\hat{\mathbf{x}}_{i,j}(k)$ in (2.4).

3.2 Local Communication

In multi-robot systems, communication is critical for cooperation and coordination among robots. In this paper, it is assumed that the network communication is free of delays, which means that messages are transmitted and received instantaneously without any loss of information. The communication range r_c is defined as the maximum distance between robots for reliable communication [34, 18, 28, 25]. If the distance between robots is greater than r_c, they cannot communicate with each other directly. Since the robots can move around and change their positions, the network communication topology is dynamic and can change over time. At any instant of time, the communication topology can be represented by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes representing the robots and $\mathcal{A}$ is the set of arcs representing the communication links between robots. The communication links are established based on the inter-robot Euclidean distance, and only the robots within the communication range r_c are connected. Therefore, robots that are far away from each other can be disconnected, forming local networks of various sizes.

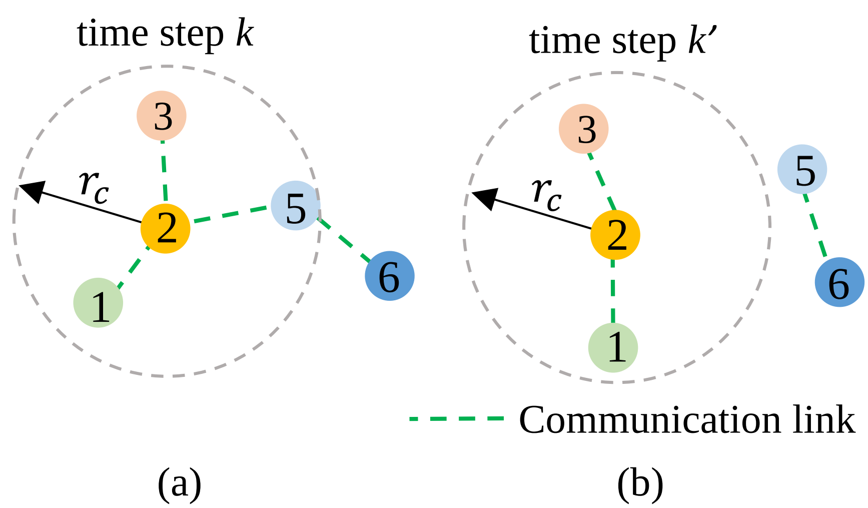

Figure 3.3: An illustrative example of a single connected communication graph (a) and two locally connected communication graphs (b) formed by five nodes (robots) of different configurations

The communication graphs representing the network in Fig. 3.3 show that, at time step k, all five nodes (robots) are connected because their distance from each other is within the communication range r_c. However, at time step k', when the distance between node 2 and node 5 exceeds the communication range, the connection between them is lost, and the communication network splits into two locally connected graphs.

Furthermore, this paper considers a communication network that uses multi-hop data transmission, which allows a robot's messages to be transmitted through one or more intermediate neighbors to reach distant, non-adjacent neighbors. As a result, multi-hop communication offers advantages over the standard single-hop communication, such as improved connectivity and extended coverage [1, 8, 31]. The subsequent sections describe decentralized coordination approaches that rely on communication-based information exchange. To simplify the descriptions, the approaches assume that N robots establish a single communication network to track M targets. Moreover, if the communication graph becomes disconnected, the methods remain applicable to any local network without any loss of generality.

CHAPTER 4

DECENTRALIZED COORDINATION

The main objective of decentralized coordination of MRNs is to find a valid target assignment through local robot estimation and communication, where every robot-target pair is associated with a tracking utility. In general, target assignment problem can be divided into two categories as shown in Fig. 4.1: single-assignment and multi-assignment problems. In single-assignment problems, each target is assigned to only one robot in the network [32]. In this case, the objective is to find a one-to-one matching between the robots and targets, such that each robot is responsible for tracking a unique target. In the multi-assignment problems, each robot may be assigned to one or more targets, and each target may be assigned to one or more robots. This flexibility allows for more efficient use of the robots for target tracking, but also makes the assignment problem more complex as the number of targets increases because the number of possible assignments grows exponentially.

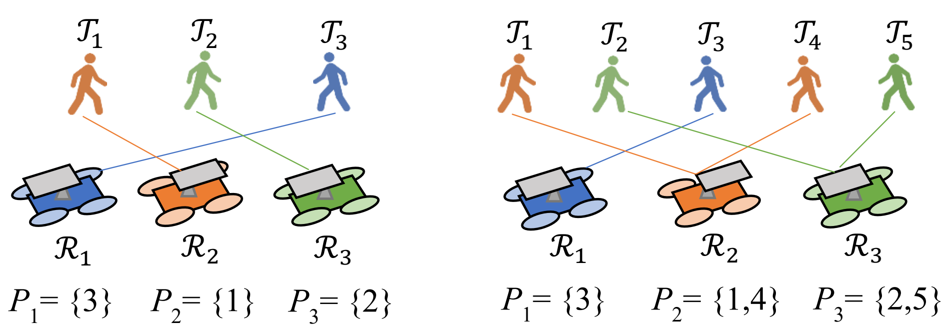

Figure 4.1: Single-assignment problem and multi-assignment problem.

In order to solve the target-assignment problems, the auction-based algorithm has been proposed [39, 3]. The algorithm consists of two phases, bidding and assignment. In the bidding phase, each robot places a bid for each target, where the bid represents the cost or utility of the robot tracking that target. The bidding process is carried out in a distributed manner where each robot communicates its bid to its neighbors in the network. The neighbors then combine the bids from all robots and broadcast the updated values to their own neighbors, leading to a network-wide distribution of bid values. In the assignment phase, each robot selects a target to track based on the bids received in the bidding phase. The robot selects the target with the highest bid that has not been assigned to any other robot. The algorithm continues until all targets have been assigned to a robot.

This paper proposes two new distributed algorithms for solving the multi-assignment problem. These algorithms, called group-based assignment and bundle-based assignment, both use an auction phase and a consensus phase to achieve a conflict-free assignment. However, they differ in the way they construct assignment combinations and the auction protocols they use. In both algorithms, the notation introduced here is used. Let $\tilde{\mathcal{R}}(k) \subseteq \mathcal{R}(k)$ represent a local network at an arbitrary time instant $k > 1$, where multi-hop communication is established. The set of robots in this network is represented by $\tilde{\mathcal{N}}(k)$. The set of targets tracked by all robots in $\tilde{\mathcal{R}}(k)$ at the previous time instant $k-1$ can be communicated across the local network and is denoted by $\mathcal{M}(k-1)$. Let $\mathcal{R}_i$ denote the i^{th} robot in the local network $\tilde{\mathcal{R}}(k)$, although it may have a different index in the entire network $\mathcal{R}(k)$ due to the subset relationship $\tilde{\mathcal{R}}(k) \subseteq \mathcal{R}(k)$. At each time step k, $\mathcal{R}_i$ maintains a winning bid list $\mathbf{y}_i$ and a winning agent list $\mathbf{a}_i$ that are iteratively updated during the auction process. The winning bid list keeps track of the highest bid for each target, while the winning agent list indicates which agents own each target. The index k is omitted for brevity in the remainder of this section. The tracking utility function $U_{i,j}(\cdot)$ is used to calculate bids, where larger values represent more favorable targets. The goal of robot coordination is to maximize tracking performance in the local network. Based on these preliminaries, the following subsections describe the design of two multi-assignment algorithms.

4.1 Group-Based Assignment

The group-based assignment works by dividing the set of targets ($\mathcal{M}$) into a number of groups equal to the robots' number (N) and having the agents bid on the group, rather than individual targets. Then, each robot submits a bid for each group of targets, and selects the group with the highest bid. Each target group is then assigned to one and only one robot in the local network.

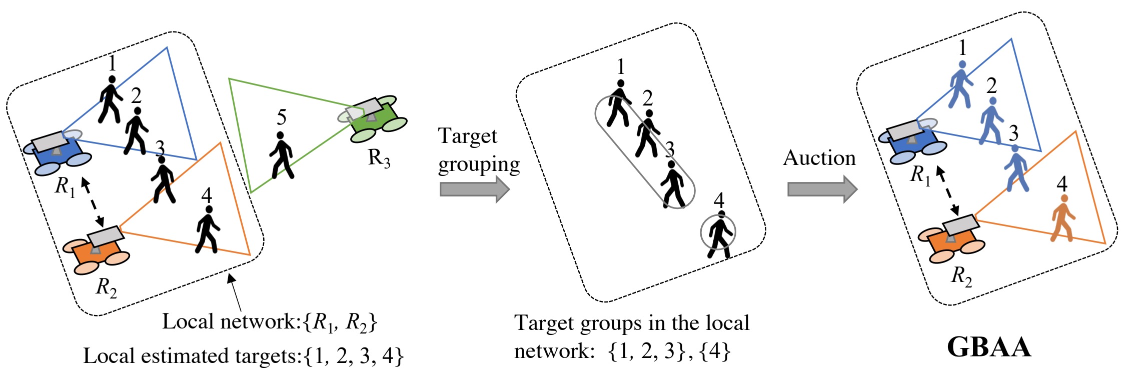

Figure 4.2: Group-based target assignment in a local network

The group-based assignment converts a multi-assignment problem to a single-assignment problem. By definition, the number of robots in a local network $\tilde{\mathcal{R}}$ is equivalent to the size or cardinality of the network, denoted by $|\tilde{\mathcal{R}}|$. Thus, the target set $\mathcal{M}$ obtained through communication is divided into $|\tilde{\mathcal{R}}|$ groups for assignment. This partitioning is based on k-means clustering [14, 21].

Assuming that target groups can be represented by their centroid position, the tracking utility gained by robot i from selecting the $\jmath^{th}$ target group is denoted by $U_{i,\jmath}(\cdot)$. This utility can be obtained by replacing the individual target estimate with the group centroid. However, to claim the $\jmath^{th}$ target group, robot i also pays a bid $y_{i,\jmath}$. Therefore, the net utility associated with a target group is defined as $U_{i,\jmath} - y_{i,\jmath}$. Each robot $i \in \mathcal{N}$ aims to maximize this net utility while avoiding conflicts.

As robot performs the same iterative process for auction and consensus at each time instance, this paragraph focuses on describing one iteration which consists of a single run of bidding and consensus. Let $\mathbf{y}_i^{(l)} = \{y_{i,\jmath}^{(l)} \mid \jmath \in \{1, \ldots, N\}\}$ and $\mathbf{a}_i^{(l)} = \{a_{i,\jmath}^{(l)} \mid \jmath \in \{1, \ldots, N\}\}$ represent the winning bids list and the winning agent list carried by robot i up to the l^{th} iteration, respectively. At the start of the $(l+1)^{th}$ iteration, robot i aims to claim the target group $\jmath_i$ that maximizes its net utility

$$\jmath_i = \arg\max_{\jmath} \; U_{i,\jmath} - y_{i,\jmath}^{(l)}, \quad \jmath \in \{1, \ldots, N\} \tag{4.1}$$

During the iterative process, if the another robot $i', i' \neq i$ won the bid for the target group $\jmath_i^{th}$ in the l^{th} iteration, robot i will need to increase its bid to compete for the target group in the $(l+1)^{th}$ iteration

$$y_{i,\jmath_i}^{(l+1)} = y_{i,\jmath_i}^{(l)} + \delta \tag{4.2}$$

where δ is the largest increment by which the bid can be increased, with the $\jmath_i^{th}$ target group still being the best option for robot i [3, 27, 4]. The δ is the net utility difference between the current best and second-best target group

$$\delta = (U_{i,\jmath_i} - y_{i,\jmath_i}^{(l)}) - \max_{\jmath' \neq \jmath_i}(U_{i,\jmath'} - y_{i,\jmath'}^{(l)}) \tag{4.3}$$

Robot i increases its bid by δ and becomes the new winning agent for the $\jmath_i^{th}$ target group. The winning bids list $\mathbf{y}_i^{(l+1)}$ is then broadcast to neighboring robots to reach a consensus. Upon receiving their neighbors' winning bids lists, robot i updates its own lists using Equations (4.4) and (4.5).

$$y_{i,J}^{(l+1)} = \max_{\imath \in \mathcal{N}} y_{\imath,J}^{(l+1)}, \quad J \in \{1, \ldots, N\} \tag{4.4}$$

$$a_{i,J}^{(l+1)} = \arg\max_{\imath \in \mathcal{N}} y_{\imath,J}^{(l+1)}, \quad J \in \{1, \ldots, N\} \tag{4.5}$$

The auction and consensus processes are repeated until the winning agent lists $\mathbf{a}_i, i \in \mathcal{N}$ converge.

The group-based assignment algorithm can be summarized as follows:

Algorithm 1: One iteration of group-based assignment

Input: $y_i^{(l)}$, $a_i^{(l)}$ in the l^{th} iteration

Output: $y_i^{(l+1)}$, $a_i^{(l+1)}$ in the $l+1^{th}$ iteration

1. The i^{th} robot tries to assume the best target group:
   $J_i = \arg\max_{J} U_{i,J} - y_{i,J}^{(l)}, \quad J \in \{1, \ldots, N\}$
2. If the j_i^{th} target group was won by another robot $i' \neq i$, Robot i will increase its bid and update the winning agent:
   $y_{i,J_i}^{(l+1)} = y_{i,J_i}^{(l)} + \delta, a_{i,J_i}^{(l+1)} = i$
3. Send $y_i^{(l+1)}$ to all neighbors $i' \in N, i' \neq i$
4. Receive $y_{i'}^{(l+1)}$ from all neighbors $i' \in N, i' \neq i$
5. Update the winning bid list $y_{i,J}^{(l+1)} = \max_{i' \in \mathcal{N}} y_{i',J}^{(l+1)}$
6. Update the winning agent list $a_{i,J}^{(l+1)} = \arg\max_{i' \in \mathcal{N}} y_{i',J}^{(l+1)}$

In the group-based approach, the multi-assignment problem is efficiently solved by dividing the targets into groups of the same size as the robot network, simplifying it into a single-assignment problem. However, this method assumes that target groups are best represented by their centroids and treats them as a single item in the auction. This abstraction results in robots considering distinct targets equally important when grouped together, thereby losing important information about the contribution of each target to the overall tracking utility. To address these limitations, the bundle-based assignment method is presented in the next subsection.

4.2 Bundle-Based Assignment

Bundle-based assignment is a different approach compared to the group-based assignment, allowing robots to bid on individual targets instead of groups. In this method, each robot constructs a bundle during the auction by adding targets sequentially based on how much they improve the tracking utility that needs to be optimized. The order in which targets are added reflects their importance, with the first target being the most desirable for the robot. This bundle is dynamic, unlike a group in the group-based assignment which is static. The significance of targets within the bundle is used to calculate bids during the auction. Therefore, this adaptability makes the bundle-based assignment more flexible and robust than the group-based assignment, which requires a fixed number of target groups to be defined before the auction. The bundle-based assignment method draws inspiration from the consensus-based bundle algorithm (CBBA) [7] which was created to allocate tasks to robots for a mission with fixed locations and time windows. However, the CBBA only allocates tasks at the beginning of the mission and cannot be modified during the mission. In contrast, the proposed bundle-based assignment solves the target assignment problem online, allowing for real-time adaptation to the dynamic nature of the robots and targets.

In the bundle-based assignment, in addition to the winning bid list $\mathbf{y}_i$ and winning agent list $\mathbf{a}_i$, robots also maintain a bundle list $\mathbf{b}_i$ to keep track of the dynamic assignment combination created during the auction. The length of $\mathbf{y}_i$ and $\mathbf{a}_i$ is the same as the number of targets in the local network, whereas the length of $\mathbf{b}_i$ represents the maximum allowable number of targets that can be assigned to each robot and can be chosen based on factors such as the average target to robot ratio. The bundle-based assignment shares the same goal with the group-based assignment, which is to find a conflict-free assignment iteratively. However, the auction protocols used by the two methods are different. The following section describes an iteration of the bundle-based assignment to highlight the primary distinctions between the two approaches.

In each iteration, after obtaining the winning bid list $\mathbf{y}_i^{(l)}$, winning agent list $\mathbf{a}_i^{(l)}$, and bundle list $\mathbf{b}_i^{(l)}$, robots in the bundle-based assignment aim to fill the first empty element in $\mathbf{b}_i^{(l)}$ in the subsequent iteration. The index of the first empty element in $\mathbf{b}_i^{(l)}$ is denoted by n_i, and the robot selects the best target that is not yet in the bundle to fill this element. The index of such a target is denoted by j and is determined using the equation:

$$j_i = \arg\max_j \beta^{n_i-1} U_{i,j}, \quad j \in \mathcal{M} \backslash \mathbf{b}_i^{(l)} \tag{4.6}$$

Here, $U_{i,j}$ is the tracking utility function with the robot states being temporarily constant, and β^{n_i-1}, $0 < \beta \leqslant 1$, is a factor that hinders a target from being selected by the robot who has already won many targets (a large n_i). The factor is a design parameter that considers the amount of utility that each target can contribute to the robot under consideration, and its value determines the significance of each target.

Unlike the bid update in the group-based assignment (4.2), the element in the winning bid list is updated after selecting the j_i^{th} target

$$y_{i,j_i}^{(l+1)} = \max_j \beta^{n_i-1} U_{i,j},\, j \in \mathcal{M} \backslash \mathbf{b}_i^{(l)} \tag{4.7}$$

Then, the n^{th} element in the bundle list $\mathbf{b}_i^{(l+1)}$ of the new iteration is constructed by comparing the updated $\mathbf{y}_i^{(l+1)}$ with $\mathbf{y}_i^{(l)}$ from the previous iteration

$$b_{i,n_i}^{(l+1)} = \begin{cases} j_i & \text{if} \quad y_{i,j_i}^{(l+1)} > y_{i,j_i}^{(l)} \\ \varnothing & \text{otherwise} \end{cases} \tag{4.8}$$

where $\varnothing$ means no target will be added to $\mathbf{b}_i^{(l+1)}$ because robot i is not able to offer a larger bid than the previous rounds.

Once $\mathbf{y}_i^{(l+1)}$ and $\mathbf{b}_i^{(l+1)}$ are obtained, they are broadcast in the local network by multi-hop communication [39] such that every robot will receive the up-to-date highest bid for each target. After the broadcast of updated bid and winning target information, the robots perform a consensus step to update their winning bid lists $\mathbf{y}_i^{(l+1)}$ with the highest bid offered by themselves or their neighbors [17]. This ensures that all robots have the most up-to-date information about the highest bids for each target. Any targets that have been outbid are also released from the bundle list $\mathbf{b}_i^{(l+1)}$. Finally, the updated bundle list $\mathbf{b}_i^{(l+1)}$ is used to determine the winning agent list $\mathbf{a}_i^{(l+1)}$ for each robot i. If a target is assigned to a robot in the updated bundle list, that robot is recorded as the winning agent for that target. Otherwise, the winning agent for that target remains the same as in the previous iteration. The updated winning agent list is given by

$$a_{i,j}^{(l+1)} = \begin{cases} i & \text{if} \quad j \in \mathbf{b}_i^{(l+1)} \\ a_{i,j}^{(l)} & \text{otherwise} \end{cases},\ \forall i \in \mathcal{N},\ \forall j \in \mathcal{M} \tag{4.9}$$

This completes one iteration of the bundle-based assignment algorithm. The process of bidding, bundle construction, and consensus is repeated until a conflict-free target assignment is achieved.

The bundle-based assignment algorithm can be summarized as follows:

Algorithm 2: One iteration of bundle-based assignment

Input: $y_i^{(l)}$, $a_i^{(l)}$, $b_i^{(l)}$ in the l^{th} iteration for Robot i

Output: $y_i^{(l+1)}$, $a_i^{(l+1)}$, $b_i^{(l+1)}$ in the $l{+}1^{th}$ iteration for Robot i

1. Robot i selects the best target that is not yet in $b_i^{(l)}$:

   $$j_i = \arg\max_j \ \beta^{n_i-1} U_{i,j}, \quad j \in \mathcal{M} \backslash \mathbf{b}_i^{(l)}$$

2. Update the winning bid list:

   $$y_{i,j_i}^{(l+1)} = \max_j \ \beta^{n_i-1} U_{i,j}, \quad j \in \mathcal{M} \backslash \mathbf{b}_i^{(l)}$$

3. Update the n^{th} element in the bundle:

   $$b_{i,n_i}^{(l+1)} = \begin{cases} j_i & \text{if} \quad y_{i,j_i}^{(l+1)} > y_{i,j_i}^{(l)} \\ \varnothing & \text{otherwise} \end{cases}$$

4. Send $y_i^{(l+1)}$ and $b_i^{(l+1)}$ to all neighbors $i' \in N, i' \neq i$
5. Receive $y_{i'}^{(l+1)}$ and $b_i^{(l+1)}$ from all neighbors $i' \in N, i' \neq i$
6. Update the winning bid list:

   $$y_{i,J}^{(l+1)} = \max_{i' \in N} \ y_{i',J}^{(l+1)}$$

7. Update the winning agent list:

   $$a_{i,j}^{(l+1)} = \begin{cases} i & \text{if} \quad j \in \mathbf{b}_i^{(l+1)} \\ a_{i,j}^{(l)} & \text{otherwise} \end{cases}, \ \forall i \in \mathcal{N}, \ \forall j \in \mathcal{M}$$

In summary, both group-based assignment and bundle-based assignment aim to iteratively update bids and assignments until a converged solution is reached. Based on the converged $\mathbf{a}_i$, the target assignment solution to the decentralized coordination is obtained as

$$P_i = \{ j \in \mathcal{M} \mid \mathbf{1}(a_{i,j} = i) \}, \quad \forall i \in \mathcal{N} \tag{4.10}$$

P_i represents the set of targets that are assigned to robot i after the auction and consensus process. For each target j in the entire set of possible targets $\mathcal{M}$, the indicator function $\mathbf{1}(a_{i,j} = i)$ equals one if robot i is the winning agent for that target j and zero otherwise. Thus, P_i is the set of all targets assigned to robot i based on the converged winning agent list $\mathbf{a}_i$.

4.3 Complexity Analysis of Group-Based Assignment

The time complexity of the group-based assignment consists of three sequential steps. In the first step, the robots establish a communication network, which takes $\mathcal{O}(NM)$ time to exchange locally estimated target states. This ensures consistency of the information used in subsequent steps. The second step involves target grouping, where the M targets are divided into N groups, one for each robot. The paper uses k-means clustering [14, 21] for target grouping, which takes $\mathcal{O}(\kappa_1 NM)$ time to converge, where κ_1 is the number of iterations required for clustering. The last step is to determine a valid assignment through iterative auction and has a worst-case time complexity of $\mathcal{O}(N^3 \max\limits_{i,j}(U_{i,j}/\epsilon))$, where ϵ is the minimum bid increment. This worst-case scenario involves a network of chain structure that takes $N-1$ communication rounds to propagate information, a conflict on the assignment of every target group for all robots, and all robots persistently placing minimum bid increments of ϵ to compete for a target group [39]. Because it takes no more than $N \max\limits_{i,j}(U_{i,j}/\epsilon)$ iterations to resolve conflicts on a single assignment among all robots [3], it follows that the worst-case scenario requires no more than $\mathcal{O}(N^3 \max\limits_{i,j}(U_{i,j}/\epsilon))$ time to terminate. As a result, the total computational complexity of the group-based assignment is

$$\mathcal{O}(NM + \kappa_1 NM + N^3 \max_{i,j}(U_{i,j}/\epsilon)) \tag{4.11}$$

4.4 Complexity Analysis of Bundle-Based Assignment

In contrast to the group-based assignment, the bundle-based assignment has a lower time complexity because it does not require target grouping before assignment. The first contribution term in the time complexity comes from constructing the communication network, which takes $O(NM)$ time, the same as the group-based assignment. The bundle-based algorithms have a different bidding scheme than the group-based assignment, which affects the running time required by the iterative auction. However, the worst-case scenario for computing the running time is constructed in a similar way as described in the group-based assignment. In bundle-based assignment, robots bid on each of the M targets separately, without dividing them into N groups as in the group-based assignment. Conflicting assignments are resolved by each robot releasing the targets that are outbid, which results in a worst-case time complexity of N^2M to reach consensus on all targets. Therefore, the total computational complexity of the bundle-based assignment is

$$O(NM + N^2M) \tag{4.12}$$

Although the worst-case scenario involves N^2M iterations, in practice, the algorithm usually converges much earlier because some robots may form smaller local networks, resulting in a more efficient communication topology than the chain graph assumed in the worst-case scenario.

CHAPTER 5

EXPERIMENTS AND RESULTS

In this chapter, the proposed decentralized coordination approach is tested and evaluated through extensive simulations under various scenarios, including randomized initial network configurations, different target trajectories, and varying communication ranges for inter-robot communication. Section 5.1 assesses the performance of the two proposed assignments. In Section 5.2, the impact of network coordination is examined. Lastly, Section 5.4 involves physical experiments with several UGVs to track human targets, demonstrating the potential real-time implementation of the proposed methods.

5.1 Performance Analysis of Group-Based Assignment and Bundle-Based Assignment

The group-based assignment simplifies the multi-assignment problem by first grouping the targets before the auction. To achieve this, robots within a local network share their estimated target states and divide the targets into groups based on their distance. As shown in Fig. 5.1(a), robots R1 and R2 form a local network and identify three targets. Then, they divide the targets into two groups, which is the same number as the robots (Fig.5.1(b)). Next, robots estimate the tracking utility of each group and conduct an auction. In Fig. 5.1(c), R1 selects the Group 1, which has the highest net utility, and places the largest bid it can afford on the Group 1. Following the same strategy, R2 also places a bid on the Group 1 (Fig.5.1(d)). At this time, a conflict arises between the two robots as they both bid on the same group. However, through the local network communication, R1 decides to select the Group 1 because it places a higher bid. In the second iteration, R1 maintains its bid as it had already won in the previous round (Fig.5.1(e)). On the other hand, R2 changes its bid to the Group 2 (Fig.5.1(f)), as the price of the Group 1 has increased due to R1's bidding. Finally, R2 chooses Group 2, resulting in the target assignments being completed without any conflicts.

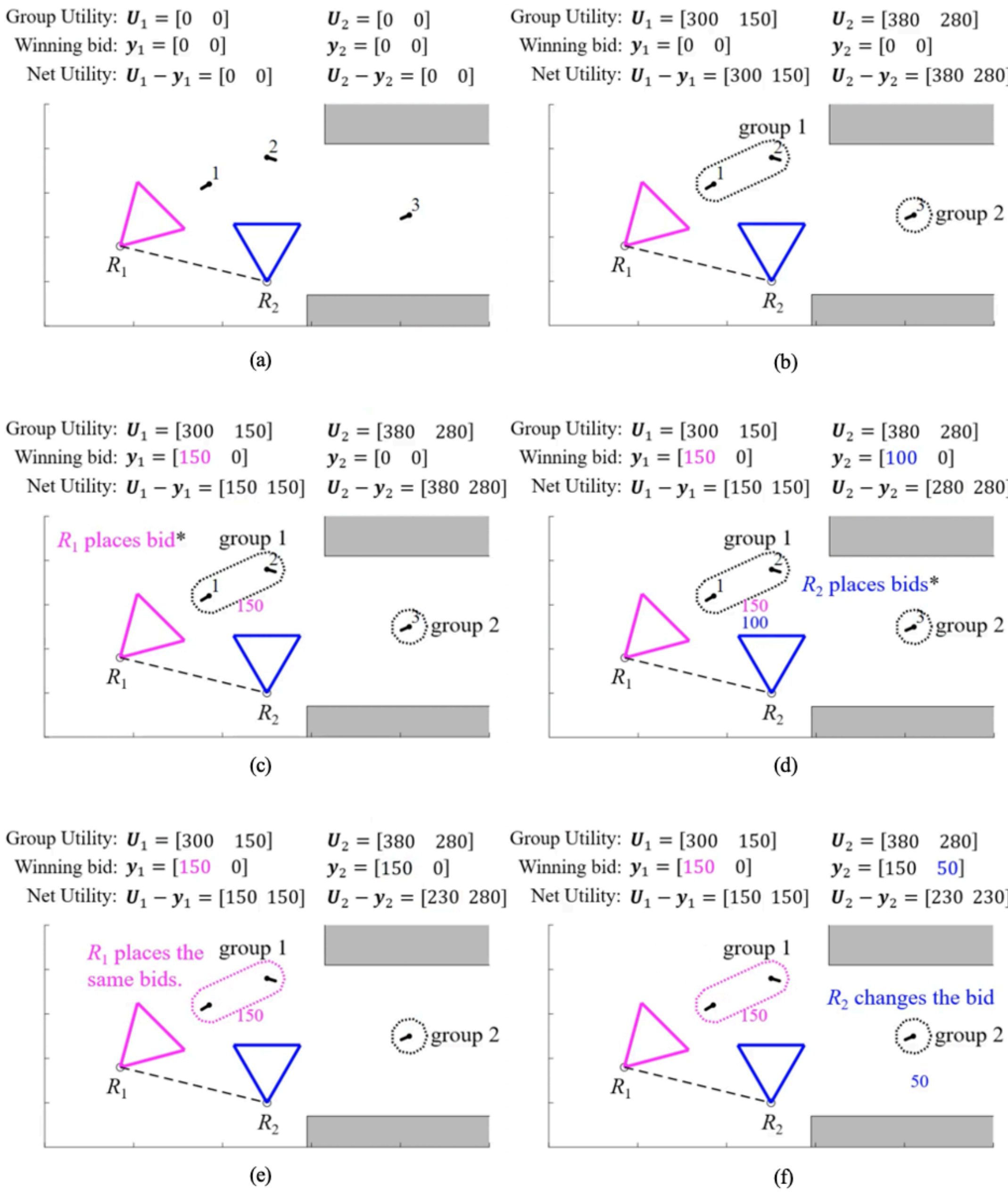

Figure 5.1: Process of the group-based assignment in a local network

In the bundle-based assignment, each robot constructs a bundle by sequentially adding targets based on the individual target's tracking utility. Therefore, a bundle refers to a dynamic combination of targets that is formed during the auction process. Overall process of the bundle-based assignment is simulated in Fig. 5.2. To begin the bundle-based assignment process, each robot starts with an empty bundle list, as illustrated in Fig. 5.2(a). Next, R1 selects the target with the highest tracking utility, which is the Target 1, and places its initial bid on it, as shown in Fig. 5.2(b). However, R2 also wants the Target 1 the most and places its initial bid on the Target 1 as well, resulting in a conflict (Fig.5.2(c)-(d)). The robots then communicate with each other to resolve the conflict, and R2 releases its bid on the Target 1 because R2 has been outbid (Fig.5.2(e)). All the robots keep bidding for targets until all targets are assigned to one of the robots. At the end of the auction, each robot has a bundle list that contains the assigned targets. The final bundle list for each robot is illustrated in Fig. 5.2(f).

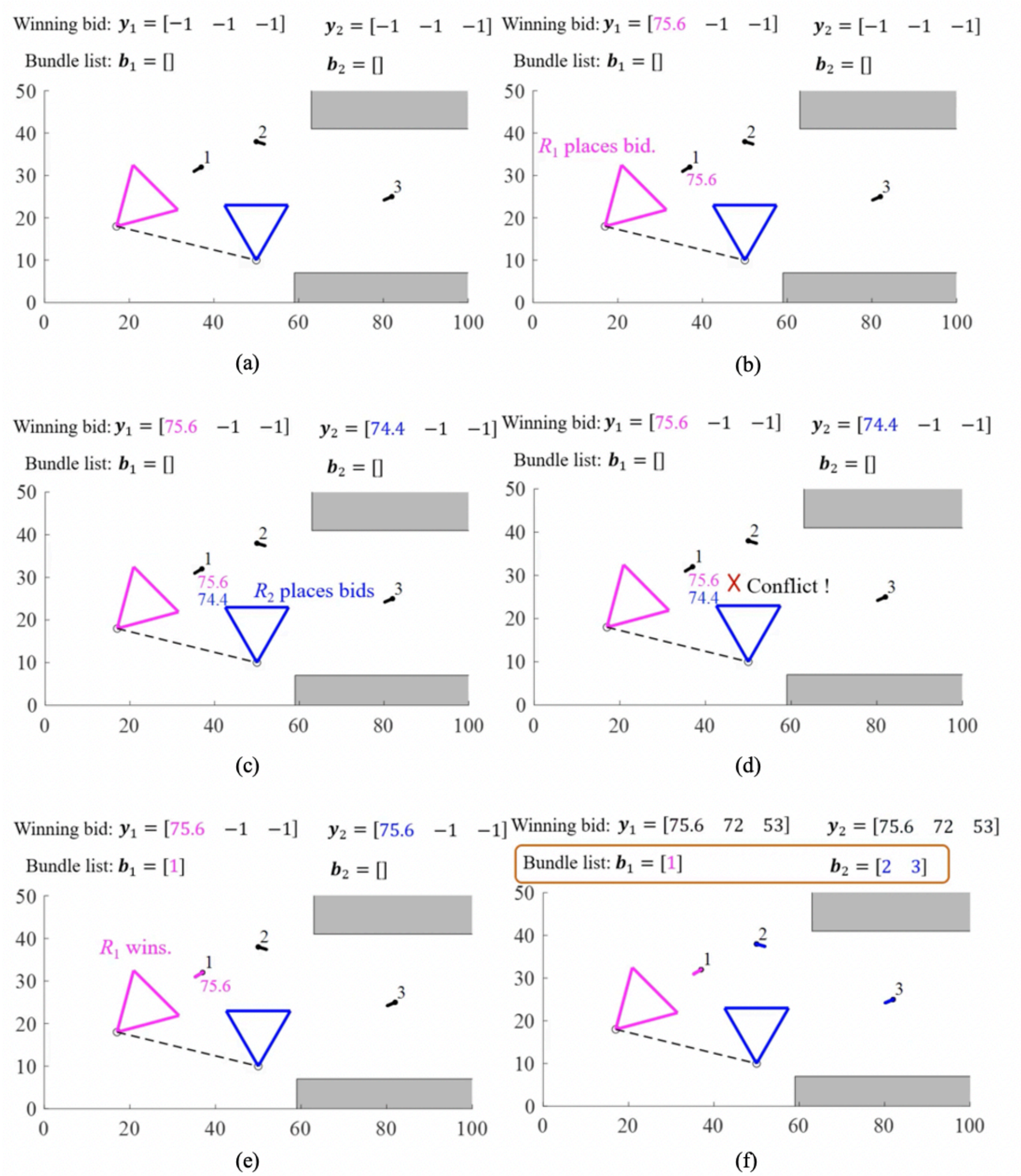

Figure 5.2: Process of the bundle-based assignment in a local network

5.2 Impact of Network Coordination

In this section, the impact of network coordination is demonstrated by comparing the two proposed assignments with the non-adaptive assignment method, which is the expected entropy reduction (EER) based tracking control. It is assumed that the non-adaptive assignment is realized by fixing the initial target assignment throughout the simulation and only optimizing the EER-based tracking utility for network control. As illustrated in Fig. 5.3, The simulation is conducted in a bounded workspace of size $100m \times 50m$ containing six moving targets and four mobile robots. The targets are randomly placed at the beginning of the simulation, and the initial target states are assumed to be known to the robot network. Also, targets are denoted by the color of the robot they are assigned to throughout the simulation.

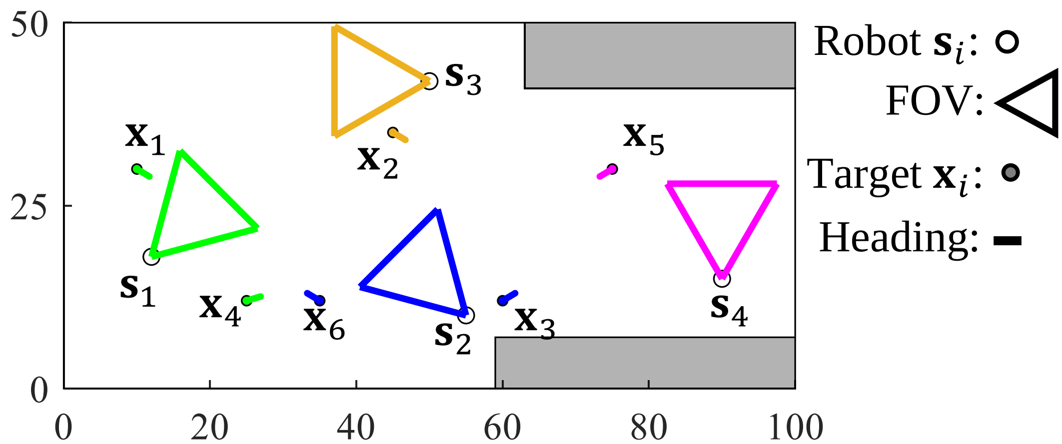

Figure 5.3: An example of the initial network configuration with robots and the assigned targets visualized in the same color

To ensure a fair comparison among the three methods, the same initial configuration and a communication range of $30m$ are used for testing. The tracking results at the same time instant for all three methods are illustrated in Fig. 5.4. In

EER control, it can be seen that the targets $\mathbf{x}_4$ and $\mathbf{x}_6$ are being tracked by robots $\mathbf{s}_1$ and $\mathbf{s}_2$ respectively, even though the two targets are in close proximity to each other and could have been efficiently tracked by a single robot. This inefficiency arises due to the lack of coordination in the EER control method. Additionally, the tracking of target $\mathbf{x}_3$ is negatively impacted as robot $\mathbf{s}_2$ has to travel a significant distance to track it after tracking target $\mathbf{x}_6$. This could potentially decrease the overall target tracking rate and increase uncertainty in the estimates of target $\mathbf{x}_3$ and ultimately lead to tracking failure. The group-based assignment shown in Fig.5.4(b) achieves coordination by having robots select target groups based on a distance criterion. In this case, robot $\mathbf{s}_1$ selects the group of targets $\mathbf{x}_1$, $\mathbf{x}_4$, and $\mathbf{x}_6$, which allows it to track targets $\mathbf{x}_4$ and $\mathbf{x}_6$ while freeing up robot $\mathbf{s}_2$ to track target $\mathbf{x}_3$. This approach improves the overall target tracking rate and reduces uncertainty in the estimates of the targets. Even more effective coordination is achieved by the bundle-based assignment in Fig.5.4(c), where the assignment of targets $\mathbf{x}_3$ and $\mathbf{x}_4$ change to adapt to the target and robot movements. This results in more effective coordination, as the targets are assigned to the robots that are closest to them. In the case of targets $\mathbf{x}_3$ and $\mathbf{x}_5$, they are assigned to robot $\mathbf{s}_4$, which is closest to their current positions. This allows for instantaneous or easy tracking in the next time steps, leading to better overall performance.

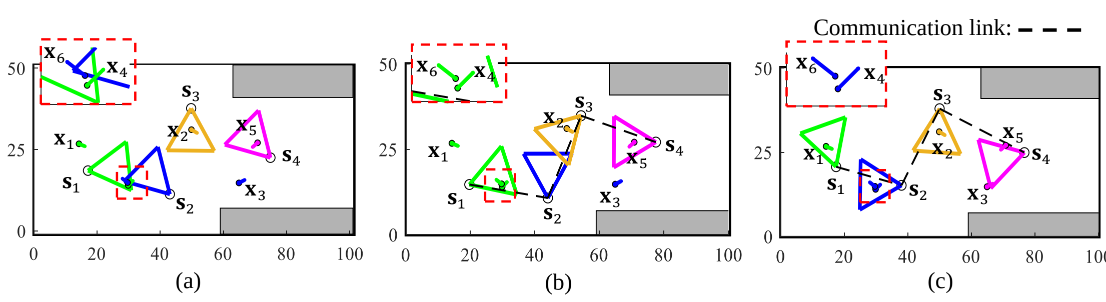

Figure 5.4: Demonstration of tracking using EER control (a) and group-based assignment (b) and bundle-based assignment (c)

5.3 Quantitative Analysis

The quantitative analysis aims to compare the proposed GBAC and BBAC methods with an offline optimal solution and two decentralized approximate approaches: EER control and PD control. EER control optimizes the decentralized network under the assumption that target assignments are known beforehand, while PD control minimizes the Euclidean distance between initially assigned targets and the center of the robot's field of view.

The effectiveness of these methods is evaluated using two metrics: average target tracking rate (ATTR), which measures the ratio of average target tracking time to the total simulation time, and average robot traveling distance (ARTD), which represents the average distance covered by all robots in the network. For all methods, 15 tests of varying initial network configurations were performed in simulation including six moving targets and four mobile robots. The ATTR and ATRD metrics recorded for each of these tests are presented in Table 5.1. While ARTD alone doesn't solely indicate tracking performance, it can be combined with ATTR to analyze the tracking efficiency in relation to energy consumption. Thus, the tracking efficiency metric, defined as the ratio of ATTR to ARTD, is also presented in Table 5.1, providing insights into the average target tracking rate per unit distance traveled by the robot.

From Table 5.1, it is evident that the proposed GBAC and BBAC methods outperform all the baselines, except for the offline optimal solution, which sets the upper bound for decentralized approaches. Compared to EER control and PD control, the superior performance of GBAC and BBAC can be attributed to adaptive target assignments achieved through network coordination. Among

Table 5.1: Comparison of Tracking Performance

Methods	**Assignment**	**Average ATTR**	**Average ARTD (m)**	**Average efficiency**
Optimal	Adaptive	77.43%	98.71	0.78
BBAC	Adaptive	71.36%	92.80	0.77
GBAC	Adaptive	66.69%	109.65	0.61
EER control	Pre-defined	53.68%	92.22	0.58
PD control	Pre-defined	45.83%	90.48	0.51

the two proposed methods, BBAC consistently demonstrates a higher ATTR on average while maintaining a lower ARTD. This observation is also reflected in the tracking efficiency metric (Table 5.1). GBAC falls short because it utilizes an abstract group representation for distinct targets when determining target assignments. In contrast, BBAC achieves more effective coordination by considering the individual targets' contributions to the network tracking utility.

5.4 Physical Experiments

The physical experiments are conducted to validate the proposed coordination methods in a decentralized manner using physical robots while achieving real-time tracking objectives. The experiments are performed in an indoor lab workspace, as shown in Fig. 5.5, using Husarion ROSbots equipped with Orbecc Astra RGB-D cameras, WiFi antennas, and odometry sensors. The robot's localization can be achieved through onboard odometry sensors or external motion capture systems installed in the lab, and the communication among robots is achieved by exchanging data over a shared WiFi network (Fig.5.6).

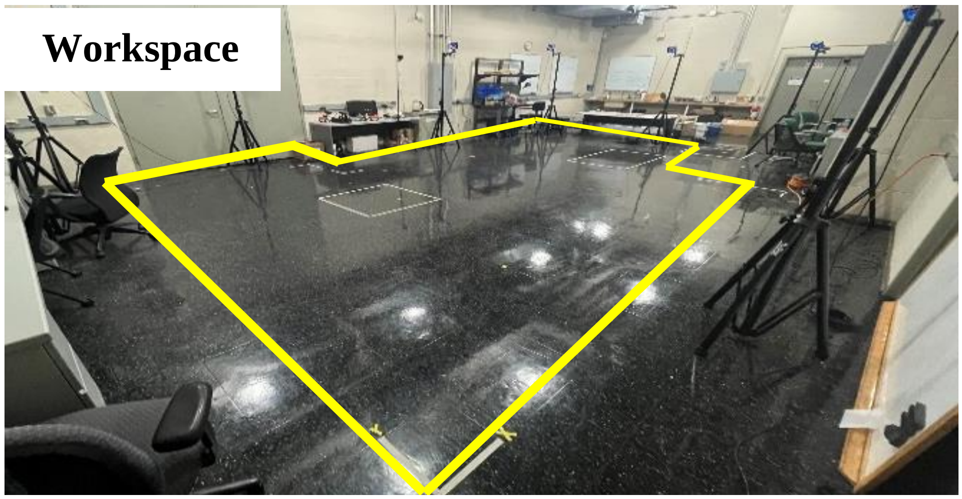

Figure 5.5: Indoor workspace

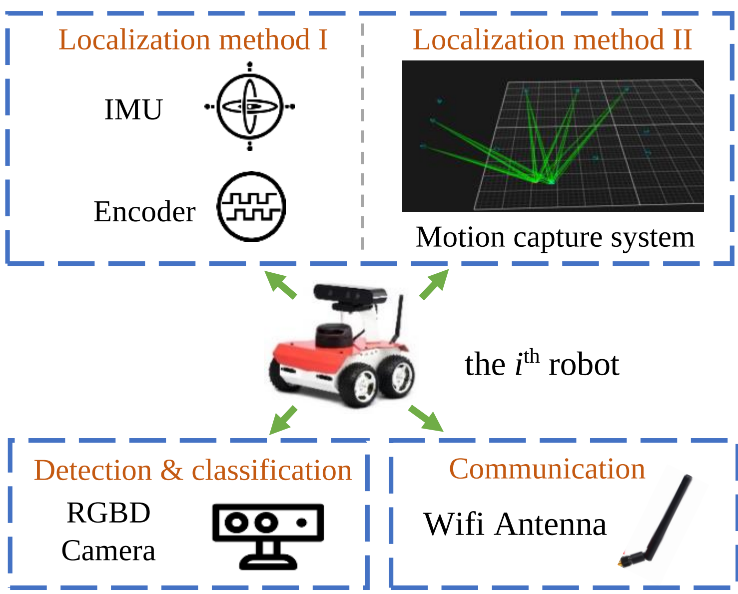

Figure 5.6: The UGV with various sensing capabilities

The initial phase of the experiments involves testing the performance of vision-based target detection, classification, and state estimation using a single target, which is presented in Appendix A. Afterwards, the decentralized optimization framework for multi-target tracking is evaluated, where the robot network is given the initial target positions and reference images of the targets-of-interest, simulating a real-life tracking scenario with some prior knowledge of the targets. The experiments were recorded by a surveillance camera placed at a predetermined location that covers the majority of the workspace.

To evaluate the effectiveness of the decentralized coordination, a scenario is designed with three targets moving in opposite directions. This scenario poses a significant challenge as it involves spatially close targets that cross each other's paths, potentially leading to confusion in target classification, assignment, and a negative impact on tracking performance. Both the EER control and the proposed methods are tested on this scenario, with the EER control demonstrating real-time tracking ability that is independent of adaptive target assignment. In contrast, the proposed methods showcase the role of network coordination in physical experiments.

Fig. 5.7 illustrates the successful tracking of assigned targets by the robots under the EER control. The assigned targets are indicated by the color of the robot they are assigned to. The figures show that the robots are able to track their initially assigned targets (Fig. 5.7(a)) consistently and accurately throughout the experiment (5.7(b) and 5.7(c)), without being distracted or confused by the other targets in the network. The trajectory of the targets and the optimized robot paths are shown in Fig.5.8. The EER control method is able to track targets consistently and accurately, but it is not an optimal solution as it cannot adapt to dynamic changes in robot and target movements. As a result, the proposed coordination methods show better performance in terms of average target tracking time and robot traveling distance, as illustrated in Fig. 5.9 and Fig. 5.10.

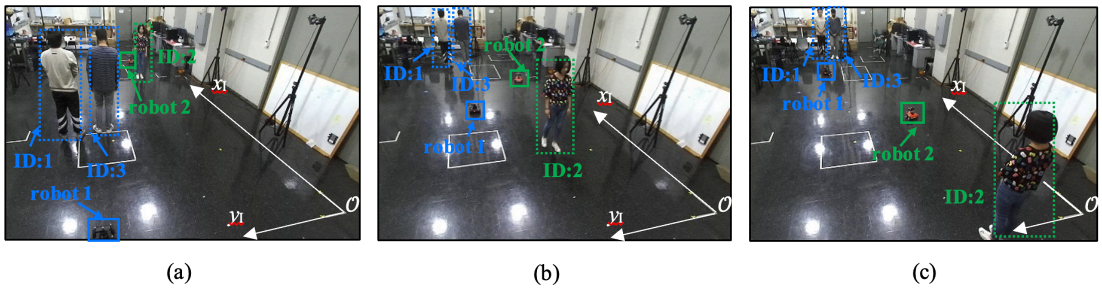

Figure 5.7: The EER control tracking three moving targets with the view of initial configuration (a), intermediate configuration (b) and final configuration (c)

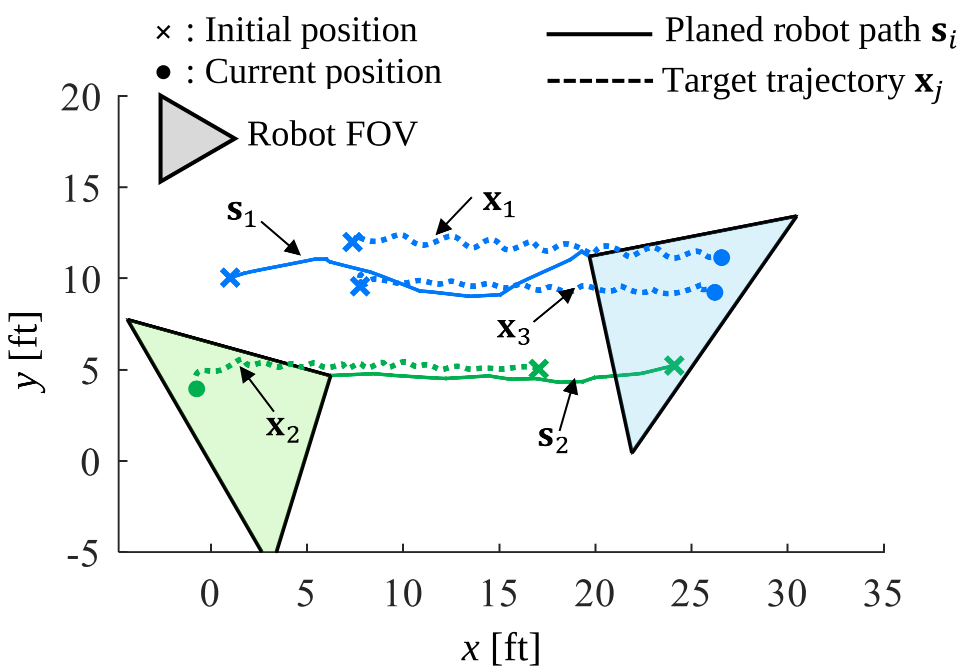

Figure 5.8: The robot path ($\mathbf{s}_i$) for tracking the initially assigned targets ($\mathbf{x}_j$) under EER control

On the other hand, the tracking results of the proposed coordination methods show the difference in the target assignment given the same initial configuration, demonstrating that the assignment is able to dynamically adapt to robot and target movements by network coordination. The results shown in Fig. 5.9 and Fig. 5.10 are obtained by implementing the group-based assignment. Compared to the results in EER control, the results in the group-based assignment show that the target assignment is changed dynamically. Initially, targets 1 and 3 are assigned to robot 1, but they move in the direction of robot 2, and hence, they are automatically reassigned to robot 2 to improve tracking efficiency. Similarly, despite being initially assigned to robot 2, target 2 is reassigned to robot 1 due to its movement towards robot 1. The color change in the target trajectories in Fig. 5.10 represents the timing of the reassignment of targets. The results demonstrate that the robot paths are optimized to track the selected targets both before and after the target assignment change. Additionally, the inter-robot communication enables the network to negotiate and resolve any conflicting assignments that arise due to robot and target movements. The bundle-based assignment is also implemented in this scenario, but the results are omitted for brevity as they show similar adaptive assignment to the group-based assignment.

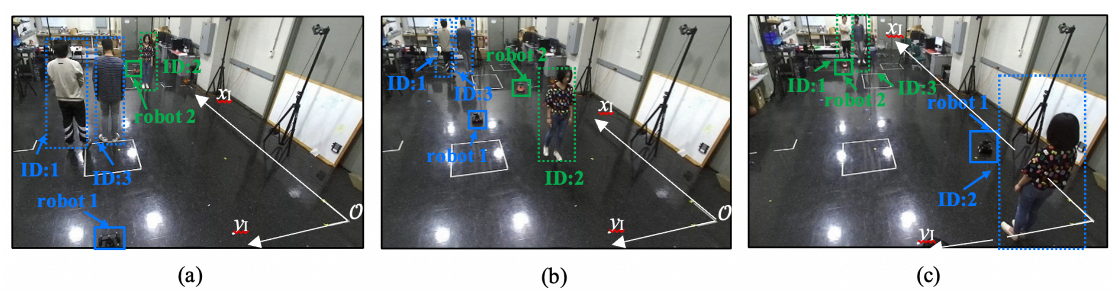

Figure 5.9: The group-based assignment tracking three moving targets with the view of initial configuration (a), intermediate configuration (b) and final configuration (c)

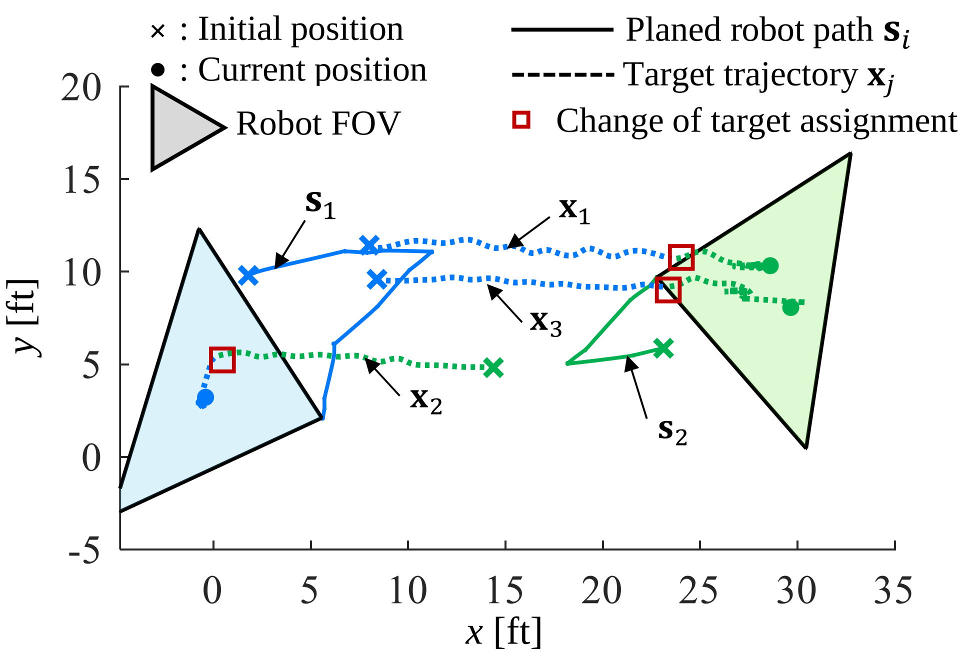

Figure 5.10: The robot path ($\mathbf{s}_i$) for tracking the initially assigned targets ($\mathbf{x}_j$) under the group-based assignment

CHAPTER 6

CONCLUSION

The presented paper introduces a decentralized coordination framework for multi-robot networks in tracking multiple dynamic targets. Two approximate approaches, namely group-based assignment and bundle-based assignment, are proposed to tackle the NP-hard optimization problem of target assignment. These approaches leverage local communication and auction mechanisms to achieve conflict-free assignments and optimize the tracking utility of each robot in real-time.

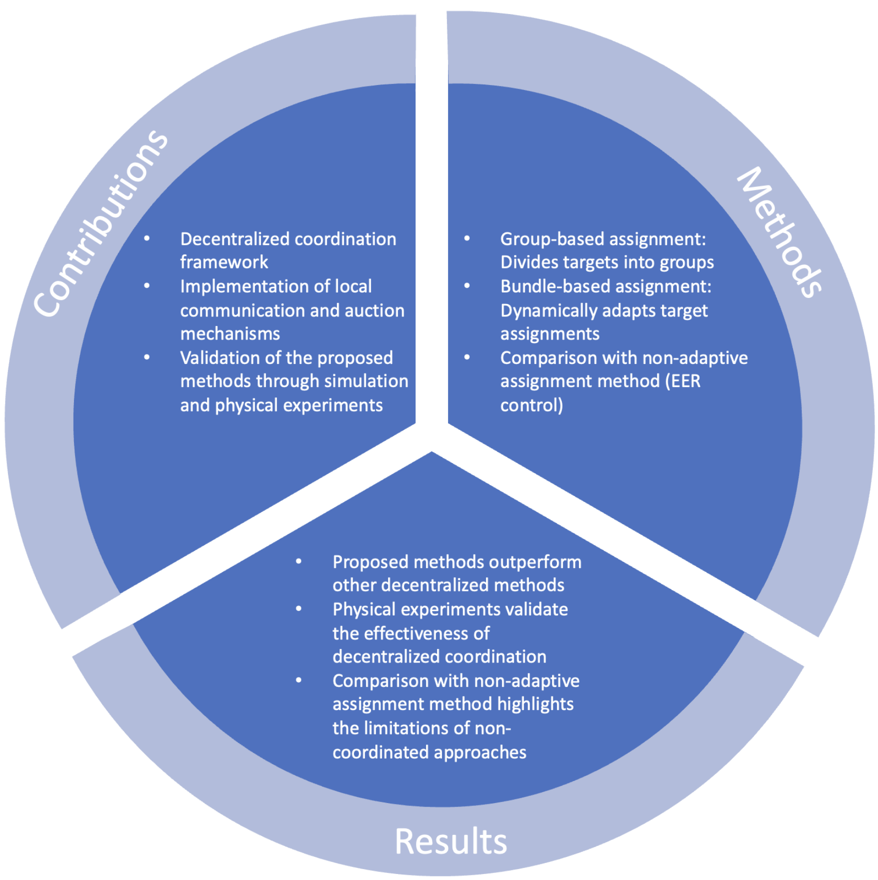

Figure 6.1: Diagram summarizing the contributions, methods implemented, and results

The simulation results demonstrate that both the group-based assignment and bundle-based assignment perform close to the optimal solution and outperform other decentralized methods. The quantitative analysis utilized two metrics, average target tracking rate (ATTR) and average robot traveling distance (ARTD), to evaluate the effectiveness of these methods. The analysis shows that the proposed GBAC and BBAC methods achieve higher average efficiency than the EER and PD controls, resulting in improved tracking efficiency. In the case study with six moving targets and four mobile robots, the impact of network coordination is compared with the non-adaptive assignment method (EER control). It is observed that the EER control lacks coordination, leading to suboptimal target assignments and potential tracking inefficiencies. In contrast, the group-based assignment successfully coordinates the robots by dividing the targets into groups based on distance and achieves more efficient tracking. The bundle-based assignment further improves coordination by dynamically adapting the target assignments to the movements of the targets and robots, resulting in even better performance.

Additionally, the physical experiments validate the effectiveness of decentralized coordination and applicability of the proposed methods in a real-world scenarios. The experiments using the non-adaptive assignment method showcased the robots' ability to consistently and accurately track their initially assigned targets without being distracted or confused by other targets in the network. The trajectory of the targets and optimized robot paths further illustrate the successful tracking under the non-adaptive assignment method. However, it is noted that the non-adaptive assignment method is not optimal as it cannot adapt to dynamic changes in robot and target movements. In contrast, the proposed coordination methods, exhibit dynamic adaptation in target assignment based on robot and target movements. The results obtained through the group-based assignment demonstrate the reassignment of targets based on their proximity to robots, resulting in improved tracking efficiency.

Overall, the paper presents a comprehensive research study on decentralized coordination for multi-robot target tracking. The proposed approaches provide valuable insights into achieving efficient and adaptive target assignments in real-world applications such as surveillance and security. The experimental results validate the effectiveness of the methods and contribute to the understanding of the benefits of network coordination in multi-robot tracking scenarios.

CHAPTER 7

FUTURE WORK

Based on the outcomes of our experiments, we can propose two main possible direction for the future work. Firstly, we can consider merging the suggested framework for robot network coordination and control with a virtual reality interface. Our investigation revealed several constraints arising from the scarcity of equipment, as well as concerns regarding privacy and security. To address these limitations, we can conduct experiments that simultaneously bridge the physical and digital world. This approach overcomes these limitations by augmenting a physical workspace and physical agents with visually realistic environments and virtual agents rendered through simulation tools.

Secondly, the proposed robot network coordination and control framework can be extended to human-robot collaboration, which can leverage complementary skills of different agents for cooperative tracking. Adverse weather conditions serve as an example where the vision of robots becomes limited due to factors such as fog or rain, leading to potential loss of targets in the environment. In such scenarios, a human operator can play a crucial role by sharing information and effectively assisting the robots. By incorporating human intelligence into the mixed team, the complexity of situations that can be tackled by robotic agents alone is significantly increased. Furthermore, human input can be transmitted to the robots asynchronously, enabling the human operator to decide when to interact with the robot team without negatively impacting robot performance.

APPENDIX A

EXPERIMENT OF THE ONLINE SENSING

The aim of this experiment was to verify the effectiveness of the vision-based target detection, classification, and state estimation pipeline (illustrated in Fig. 5.6) independently from the network coordination and communication. In one of the experimental trials, a sequence of frames was captured and presented in Fig. A.1. The robot was provided with the initial target position and reference images of the target-of-interest beforehand. Although the target was not within the robot's field-of-view (FOV) at $t = 0s$, the robot successfully tracked the target based on the predicted target dynamics at $t = 18s$ and correctly identified the target ID. Throughout the experiment, the robot continuously measured the target states by integrating RGB data, depth data, and robot localization information using equations (3.1) to (3.3). The robot path was planned by optimizing the tracking utility to ensure the target stayed within its FOV, as shown in the snapshot at $t = 46s$. The target trajectory and the planned robot path are illustrated in Fig. A.2.

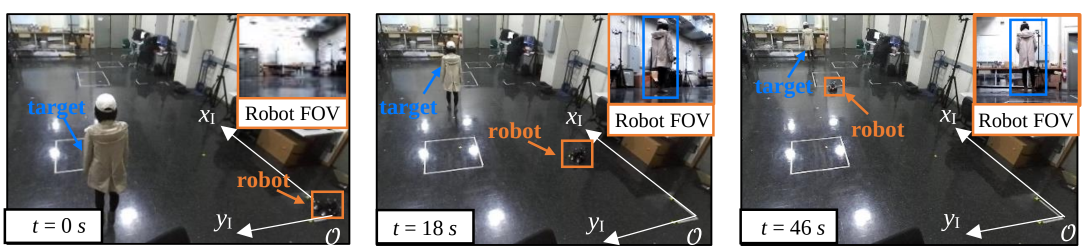

Figure A.1: Demonstration of vision-based tracking with the robot view superimposed with the recording camera view

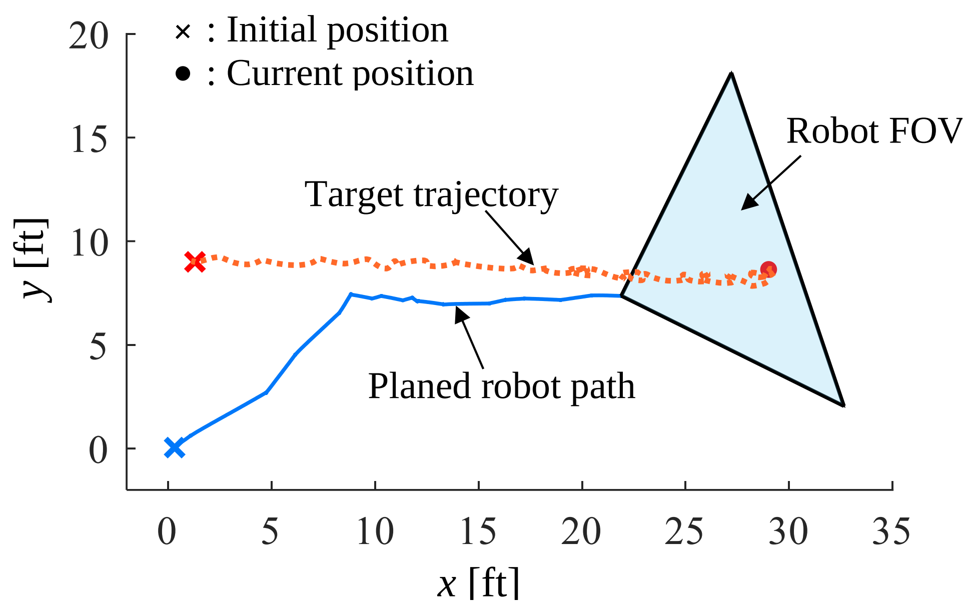

Figure A.2: Planned robot path for target tracking corresponding to Fig. A.1

BIBLIOGRAPHY

[1] Jacopo Banfi, Jérôme Guzzi, Francesco Amigoni, Eduardo Feo Flushing, Alessandro Giusti, Luca Gambardella, and Gianni A Di Caro. An integer linear programming model for fair multitarget tracking in cooperative multirobot systems. *Autonomous Robots*, 43(3):665–680, 2019.

[2] Jacopo Banfi, Jérôme Guzzi, Alessandro Giusti, Luca Gambardella, and Gianni A Di Caro. Fair multi-target tracking in cooperative multi-robot systems. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5411–5418. IEEE, 2015.

[3] Dimitri Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational optimization and applications*, 1(1):7–66, 1992.

[4] Dimitri P Bertsekas. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, 20(4):133–149, 1990.

[5] Graeme Best, Oliver M Cliff, Timothy Patten, Ramgopal R Mettu, and Robert Fitch. Dec-mcts: Decentralized planning for multi-robot active perception. *The International Journal of Robotics Research*, 38(2-3):316–337, 2019.

[6] Jesús Capitan, Luis Merino, and Aníbal Ollero. Cooperative decision-making under uncertainties for multi-target surveillance with multiples uavs. *Journal of Intelligent & Robotic Systems*, 84(1):371–386, 2016.

[7] Han-Lim Choi, Luc Brunet, and Jonathan P How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4):912–926, 2009.

[8] Jeferson Rodrigues Cotrim and João Henrique Kleinschmidt. Lorawan mesh networks: A review and classification of multihop communication. *Sensors*, 20(15):4273, 2020.

[9] Ivica Draganjac, Damjan Miklić, Zdenko Kovačić, Goran Vasiljević, and Stjepan Bogdan. Decentralized control of multi-agv systems in autonomous warehousing applications. *IEEE Transactions on Automation Science and Engineering*, 13(4):1433–1447, 2016.

[10] Abdul Qadir Faridi, Sanjeev Sharma, Anupam Shukla, Ritu Tiwari, and Joydip Dhar. Multi-robot multi-target dynamic path planning using artifi-

cial bee colony and evolutionary programming in unknown environment. *Intelligent Service Robotics*, 11(2):171–186, 2018.

[11] Silvia Ferrari and Thomas A Wettergren. *Information-Driven Planning and Control*. MIT Press, 2021.

[12] Jake Gemerek, Silvia Ferrari, Brian H Wang, and Mark E Campbell. Video-guided camera control for target tracking and following. *IFAC-PapersOnLine*, 51(34):176–183, 2019.

[13] D. O. Gorodnichy. Particle swarm optimization for tracking in video images. In *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, pages 1245–1248, 2007.

[14] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, Venice, 2017.

[16] Jonathan P How, Cameron Fraser, Karl C Kulling, Luca F Bertuccelli, Olivier Toupet, Luc Brunet, Abe Bachrach, and Nicholas Roy. Increasing autonomy of uavs. *IEEE Robotics & Automation Magazine*, 16(2):43–51, 2009.

[17] Simon Hunt, Qinggang Meng, Chris Hinde, and Tingwen Huang. A consensus-based grouping algorithm for multi-agent cooperative task allocation with complex requirements. *Cognitive computation*, 6(3):338–350, 2014.

[18] Asif Khan, Evsen Yanmaz, and Bernhard Rinner. Information exchange and decision making in micro aerial vehicle networks for cooperative search. *IEEE Transactions on Control of Network Systems*, 2(4):335–347, 2015.

[19] Su-Jin Lee, Soon-Seo Park, and Han-Lim Choi. Potential game-based non-myopic sensor network planning for multi-target tracking. *IEEE Access*, 6:79245–79257, 2018.

[20] Keith A LeGrand, Pingping Zhu, and Silvia Ferrari. A random finite set sensor control approach for vision-based multi-object search-while-

tracking. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2021.

[21] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[22] Chang Liu, Zhihao Liao, and Silvia Ferrari. Rumor-robust decentralized gaussian process learning, fusion, and planning for modeling multiple moving targets. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3066–3071. IEEE, 2020.

[23] Yu Liu, Junjie Yan, and Wanli Ouyang. Quality aware network for set to set recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5790–5799, 2017.

[24] Wenjie Lu, Guoxian Zhang, and Silvia Ferrari. An information potential approach to integrated sensor path planning and control. *IEEE Transactions on Robotics*, 30(4):919–934, 2014.

[25] Sean Luke, Keith Sullivan, Liviu Panait, and Gabriel Balan. Tunably decentralized algorithms for cooperative target observation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 911–917, 2005.

[26] Keisuke Okumura and Xavier Défago. Solving simultaneous target assignment and path planning efficiently with time-independent execution. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pages 270–278, 2022.

[27] Rahul Pal, KP Sarawadekar, and KV Srinivas. A decentralized beam selection for mmwave beamspace multi-user mimo system. *AEU-International Journal of Electronics and Communications*, 111:152884, 2019.

[28] Lynne E Parker. Decision making as optimization in multi-robot teams. In *International Conference on Distributed Computing and Internet Technology*, pages 35–49. Springer, 2012.

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[30] Brent Schlotfeldt, Dinesh Thakur, Nikolay Atanasov, Vijay Kumar, and

George J Pappas. Anytime planning for decentralized multirobot active information gathering. *IEEE Robotics and Automation Letters*, 3(2):1025–1032, 2018.

[31] Ajit Kumar Singh, Sandeep Rajoriya, Subham Nikhil, and Tapan Kumar Jain. Design constraint in single-hop and multi-hop wireless sensor network using different network model architecture. In *International Conference on Computing, Communication & Automation*, pages 436–441. IEEE, 2015.

[32] Stephen Smith and Francesco Bullo. Monotonic target assignment for robotic networks. *IEEE Transactions on Automatic Control*, 54(9):2042–2057, 2009.

[33] Yoonchang Sung, Ashish Budhiraja, Ryan K Williams, and Pratap Tokekar. Distributed assignment with limited communication for multi-robot multi-target tracking. *Autonomous Robots*, 44(1):57–73, 2020.

[34] Yoonchang Sung, Ashish Kumar Budhiraja, Ryan K Williams, and Pratap Tokekar. Distributed simultaneous action and target assignment for multi-robot multi-target tracking. In *2018 IEEE International conference on robotics and automation (ICRA)*, pages 3724–3729. IEEE, 2018.

[35] Yan Wang, Lequn Wang, Yurong You, Xu Zou, Vincent Chen, Serena Li, Gao Huang, Bharath Hariharan, and Kilian Q Weinberger. Resource aware person re-identification across multiple resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8042–8051, 2018.

[36] Yiming Wang and Andrea Cavallaro. Prioritized target tracking with active collaborative cameras. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 131–137. IEEE, 2016.

[37] Hongchuan Wei, Pingping Zhu, Miao Liu, Jonathan P How, and Silvia Ferrari. Automatic pan–tilt camera control for learning dirichlet process gaussian process (dpgp) mixture models of multiple moving targets. *IEEE Transactions on Automatic Control*, 64(1):159–173, 2018.

[38] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[39] Michael M Zavlanos, Leonid Spesivtsev, and George J Pappas. A distributed auction algorithm for the assignment problem. In *2008 47th IEEE Conference on Decision and Control*, pages 1212–1217. IEEE, 2008.