Collision-Free Velocity Tracking of a Moving Ground Target by Multiple Unmanned Aerial Vehicles*

Amir Salimi Lafmejani[1,**], Hamed Farivarnejad[2,**], Mostafa Rezayat Sorkhabadi[2], Fatemeh Zahedi[2], Azadeh Doroudchi[1], and Spring Berman[2]

[1] School of Electrical, Computer and Energy Engineering, Arizona State University
[2] School for Engineering of Matter, Transport and Energy, Arizona State University

Abstract. In this paper, we present a controller for collision-free velocity tracking of a moving ground target by multiple unmanned aerial vehicles (UAVs). The controller combines a feedforward proportional-derivative (PD) control term and a term that is based on the gradient of an artificial potential function. We use Lasalle's invariance principle to analytically prove the convergence of the UAVs to a fixed formation above the target that tracks the target's velocity and provide mathematical guarantees on the UAVs' collision avoidance. As a result, the Euclidean distance between each pair of UAVs approaches a constant value at equilibrium. In the event of UAV failure, the remaining UAVs reconfigure to a new fixed formation and maintain collision-free tracking of the target's velocity, demonstrating the robustness of our control approach to failure. We validate this control approach on different simulated scenarios in MATLAB and the Gazebo robot simulator. We also experimentally test the performance of the control approach on physical robots, using Crazyflie quadrotors as the UAVs and a Turtlebot3 Burger robot as the moving ground target. The simulation and experimental results demonstrate the effectiveness of our control approach at collision-free tracking and its robustness to UAV failure.

Keywords: Target Tracking · Collision Avoidance · Unmanned Aerial Vehicles · Feedforward Proportional-Derivative Controller

1 Introduction

Tracking of a moving target on the ground can be achieved by employing multiple unmanned aerial vehicles (UAVs). This scenario commonly occurs in different applications, including but not limited to **(1)** localization of a rover via multiple UAVs in space applications and new planet exploration [5,7], **(2)** food and medicine delivery in response to disaster situations and search-and-rescue operations [20], **(3)** capturing aerial footage for films and sporting events [14], and **(4)** police chase of a moving suspect [9].

* This work was supported by the Arizona State University Global Security Initiative.
** These two authors contributed equally to this work.

There are various studies in the literature on control approaches for tracking of a moving target by multiple UAVs [12]. These existing control approaches can be categorized as *Centralized*, *Distributed*, and *Decentralized*. In centralized control approaches, a central computational unit calculates and sends control commands to all the UAVs. Centralized approaches can guarantee collision-free and deadlock-free [19,10] navigation of the robots. However, the required computational resources of the central unit increase with the number of UAVs. One way to reduce the computational complexity of this control problem is to employ distributed control approaches [8]. In these approaches, the control problem is decomposed into multiple subproblems, which are assigned to each UAV to solve using its on-board computational resources. In this setting, neighboring UAVs communicate their solutions to each other to plan collision-free paths. However, control methods that require inter-robot communication can suffer from communication losses or delays in the transmission of information between robots [18,15]. To eliminate the requirement for a central computational unit or inter-robot communication, decentralized approaches have been developed for multi-robot systems, in which each robot computes its own control inputs using only local measurements, without communicating with other robots [6,17]. Decentralized control approaches can also be scaled up to large numbers of robots. However, these approaches require UAVs to be equipped with on-board 3D LiDAR sensors or cameras to detect other robots during tracking, and they necessitate a complicated controller design to guarantee both collision-free and deadlock-free navigation of the UAVs. Furthermore, most of the existing control methods only focus on tracking and collision avoidance, without addressing the controller's robustness to failure of UAVs during tracking.

In this paper, we propose a centralized feedforward PD controller for multiple UAVs to track a moving target on the ground while avoiding collisions with one another. Although our approach may be subject to the limitations associated with centralized control approaches, it does not depend on inter-robot communication, which is used in distributed methods, and does not require UAVs to have any on-board sensing for collision avoidance, which is needed in decentralized methods. Moreover, our control approach exhibits robustness to the failure of UAVs during target tracking. We also provide theoretical guarantees that the controller prevents collisions among the UAVs and drives them to a fixed formation above the target that tracks the target's velocity. In Section 2, we present the dynamic model of the UAVs, describe the proposed controller, and analyze its properties. The controller is validated in both MATLAB and Gazebo [13] simulations and in experiments with real robots, and the simulation and experimental results are presented in Sections 3 and 4, respectively.

2 Modeling and Control Approach

In this section, we first describe the assumed capabilities of the UAVs and define the model that is used to represent their dynamics. Then, we design a control

law to achieve collision-free tracking of a moving ground target by the UAVs and prove that the control law achieves this objective.

2.1 Dynamic Model

Figure 1 shows an illustration of our multi-robot target tracking problem, in which a group of N UAVs must track a wheeled mobile robot (WMR), the target, which is moving on the ground. Let $\mathbf{x}_t = [x_t \;\; y_t \;\; 0]^T$ denote the position of the target in the global coordinate frame, whose origin is defined on the ground. We assume that the UAVs have access to the instantaneous position $\mathbf{x}_t$, velocity $\dot{\mathbf{x}}_t$, and acceleration $\ddot{\mathbf{x}}_t$ of the target. In real-world scenarios, measurements of $\mathbf{x}_t$, $\dot{\mathbf{x}}_t$, and $\ddot{\mathbf{x}}_t$ can be obtained by cameras on the UAVs or by a GPS sensor on the moving target. We also assume that each UAV, indexed by $i \in \{1, ..., N\}$, can use measurements from its onboard sensors to accurately estimate its current position $\mathbf{x}_i \in \mathbb{R}^3$ and velocity $\dot{\mathbf{x}}_i \in \mathbb{R}^3$ in the global coordinate frame, its distance to every other UAV, and its angle with respect to every other UAV in the global coordinate frame. The UAVs can receive information from a central computational unit, but they do not communicate with one another. We also assume that the UAVs always operate close to a hovering condition and do not need to perform maneuvers with large changes in orientation, which allows us to represent the dynamics of each UAV as a point-mass double-integrator model,

$$m\ddot{\mathbf{x}}_i = \mathbf{u}_i, \tag{1}$$

where $m \in \mathbb{R}$ is the UAV's mass, assumed to be the same for all UAVs, and $\mathbf{u}_i \in \mathbb{R}^3$ is its control input vector. Using this double-integrator model facilitates the implementation of the control approach on any type of multi-rotor UAV. However, if the UAVs must perform aggressive maneuvers to track the moving target, then each UAV should instead be described by a nonlinear dynamic model [16].

2.2 Controller Design

Our proposed control law $\mathbf{u}_i$ for the ith UAV consists of two components, each designed to achieve one of the following objectives: (**1**) tracking the velocity of the moving target, and (**2**) avoiding collisions with the other UAVs. The component of the control law that achieves the tracking objective is defined as a feedforward Proportional-Derivative (PD) controller. The component that ensures collision avoidance is defined as a potential-based controller with the following potential function V_{ij}:

$$V_{ij} = r_{ij} + \frac{a}{r_{ij}}, \tag{2}$$

where $a \in \mathbb{R}_{>0}$ is a positive constant and $r_{ij} \in \mathbb{R}_{>0}$ is the Euclidean distance between the ith and j^{th} UAVs, i.e., $r_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||_2$. The full control law for the ith UAV is given by:

$$\mathbf{u}_i = -\mathbf{K}_1\mathbf{s}_i + m\ddot{\mathbf{x}}_t - \mathbf{K}_2\mathbf{h}_i, \tag{3}$$

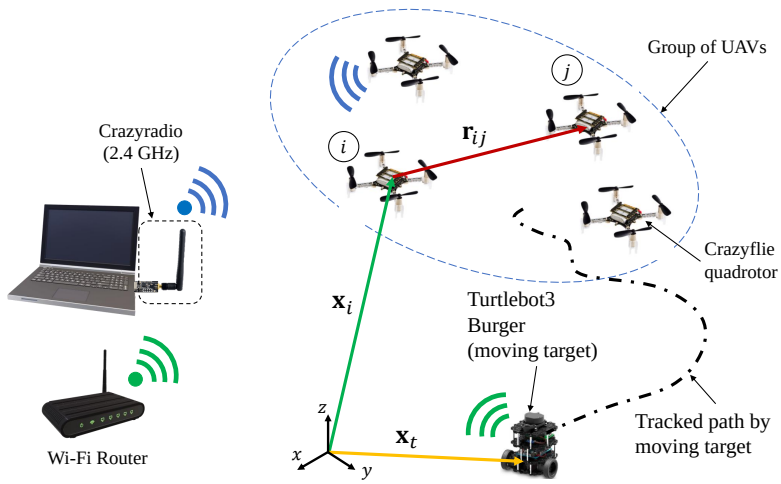


Fig. 1: Schematic of our experimental setup for the target tracking problem, in which four Crazyflie quadrotors track a Turtlebot3 Burger robot moving on the ground.

where $\mathbf{K}_1 = k_1\mathbf{I}_{3\times3}$ and $\mathbf{K}_2 = k_2\mathbf{I}_{3\times3}$ are positive definite gain matrices, with $k_1, k_2 \in \mathbb{R}_{>0}$ and $\mathbf{I}_{3\times3}$ denoting the 3×3 identity matrix; $\mathbf{s}_i \in \mathbb{R}^3$ is defined as

$$\mathbf{s}_i = \mathbf{e}_i + \mathbf{\Lambda}\dot{\mathbf{e}}_i, \quad \mathbf{\Lambda} = \lambda\mathbf{I}_{3\times3}, \quad \lambda \in \mathbb{R}_{>0}, \tag{4}$$

where $\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}_t$ is the error between the position of the ith UAV and the target; and $\mathbf{h}_i$ is the sum

$$\mathbf{h}_i = \sum_{i\neq j} \nabla V_{ij} = \sum_{i\neq j} \left(1 - \frac{a}{r_{ij}^2}\right)\mathbf{p}_{ij}, \tag{5}$$

where $\mathbf{p}_{ij}$ is the unit vector along the distance vector $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, i.e., $\mathbf{p}_{ij} = \mathbf{r}_{ij}/r_{ij}$. To execute this controller, each UAV must use its measurements of the position error $\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}_t$ and its time derivative $\dot{\mathbf{e}}_i = \dot{\mathbf{x}}_i - \dot{\mathbf{x}}_t$, the target's acceleration $\ddot{\mathbf{x}}_t$, and the magnitudes and directions of the vectors $\mathbf{r}_{ij}$.

2.3 Analysis of Closed-Loop Dynamics

We obtain the closed-loop dynamics of our target tracking control system by substituting our proposed control law in Eq. (3) into the UAV model in Eq. (1):

$$m\ddot{\mathbf{x}}_i = -\mathbf{K}_1\mathbf{s}_i + m\ddot{\mathbf{x}}_t - \mathbf{K}_2\mathbf{h}_i. \tag{6}$$

Using the expressions for $\mathbf{s}_i$ from Eq. (4) and $\mathbf{h}_i$ from Eq. (5), we obtain:

$$\begin{aligned} m\ddot{\mathbf{x}}_i &= -\mathbf{K}_1(\mathbf{e}_i + \mathbf{\Lambda}\dot{\mathbf{e}}_i) + m\ddot{\mathbf{x}}_t - \mathbf{K}_2\sum_{j\neq i}\nabla V_{ij} \\ &= -\mathbf{K}_1\mathbf{e}_i - \mathbf{K}_1\mathbf{\Lambda}\dot{\mathbf{e}}_i + m\ddot{\mathbf{x}}_t - \mathbf{K}_2\sum_{j\neq i}\left(1 - \frac{a}{r_{ij}^2}\right)\mathbf{p}_{ij}. \end{aligned} \tag{7}$$

By rearranging this equation, the closed-loop error dynamics for the ith UAV can be computed as follows:

$$m\ddot{\mathbf{e}}_i + \mathbf{K}_1\mathbf{\Lambda}\dot{\mathbf{e}}_i + \mathbf{K}_1\mathbf{e}_i + \mathbf{K}_2\sum_{j\neq i}\left(1-\frac{a}{r_{ij}^2}\right)\mathbf{p}_{ij} = \mathbf{0}. \tag{8}$$

Without loss of generality, we assume that $m = 1$. We define the following positive definite Lyapunov function:

$$H = \frac{1}{2}k_2\sum_{i=1}^{N}\sum_{j\neq i}V_{ij} + \frac{1}{2}\sum_{i=1}^{N}\left(\dot{\mathbf{e}}_i^{\mathrm{T}}\dot{\mathbf{e}}_i + \mathbf{e}_i^{\mathrm{T}}\mathbf{K}_1\mathbf{e}_i\right). \tag{9}$$

The time derivative of H can be calculated as (see Appendix):

$$\dot{H} = -\sum_{i=1}^{N}\dot{\mathbf{e}}_i^{\mathrm{T}}\mathbf{\Lambda}\mathbf{K}_1\dot{\mathbf{e}}_i, \tag{10}$$

The positive definiteness of the matrix $\mathbf{\Lambda}\mathbf{K}_1$ and the absence of $\mathbf{e}_i$ in Eq. (10) show that $\dot{H}$ is negative semidefinite. This fact, along with the positive definiteness of H, imply the boundedness of H, and consequently, the boundedness of $\mathbf{e}_i$, $\dot{\mathbf{e}}_i$, and V_{ij} for $i = 1, ..., N$, $j \neq i$. In addition, given the continuity of H, we can conclude that

$$\dot{\mathbf{e}}_i \to \mathbf{0} \quad \text{as} \quad t \to \infty, \qquad \forall\, i = 1, ..., N. \tag{11}$$

This demonstrates that the velocities of all UAVs converge to the velocity of the moving target. Moreover, Eq. (11) implies that $\mathbf{e}_i$ converges to a constant value for each UAV, and by using the fact that $\mathbf{x}_i - \mathbf{x}_j = \mathbf{e}_i - \mathbf{e}_j$, we conclude that

$$r_{ij} \to \text{constant} \quad \text{as} \quad t \to \infty, \qquad \forall\, i \neq j. \tag{12}$$

Eq. (12) shows that the UAVs converge to a fixed formation above the moving target at steady-state. Furthermore, given the definition of V_{ij} in Eq. (2), the boundedness of V_{ij} for each UAV implies that r_{ij} never equals zero for any pair of UAVs i, j, which guarantees collision avoidance for all UAVs.

Finally, using *LaSalle's invariance principle* [11], we can confirm that the trajectories of the closed-loop system converge to the largest invariant set in

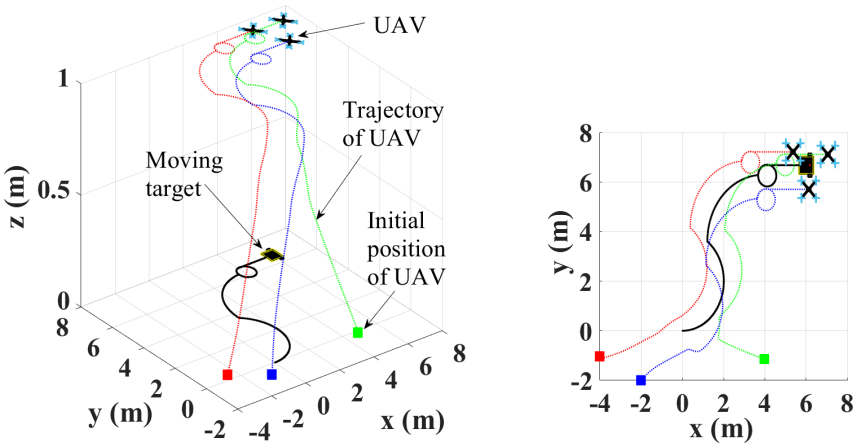
$$\mathcal{E} = \left\{\dot{\mathbf{e}}_i \in \mathbb{R}^3 \mid \dot{H} \equiv 0\right\}, \quad i = 1, \ldots, N. \tag{13}$$

Taking into account the closed-loop dynamics in Eq. (6) and the fact that $\dot{H} \equiv 0 \Rightarrow \dot{\mathbf{e}}_i, \ddot{\mathbf{e}}_i \equiv \mathbf{0}$, $\forall\, i \in \{1, \ldots, N\}$, $\mathcal{E}$ can be rewritten as

$$\mathcal{E} = \left\{\mathbf{e}_i \in \mathbb{R}^3 \mid \mathbf{K}_1\mathbf{e}_i + \mathbf{K}_2\mathbf{h}_i = \mathbf{0}\right\}, \quad i = 1, \ldots, N. \tag{14}$$

Summing the equations $\mathbf{K}_1\mathbf{e}_i + \mathbf{K}_2\mathbf{h}_i = \mathbf{0}$ over all UAVs $i = 1, ..., N$, we obtain

$$\mathbf{K}_1\sum_{i=1}^{N}\mathbf{e}_i + \mathbf{K}_2\sum_{i=1}^{N}\mathbf{h}_i = \mathbf{0}. \tag{15}$$



(a) 3D view **(b)** View in $x - y$ plane

Fig. 2: Robot trajectories during a MATLAB simulation of *Scenario 1*.

We can confirm that $\mathbf{K}_2 \sum_{i=1}^{N} \mathbf{h}_i = \mathbf{0}$, since it is the sum of the mutual repulsion forces of every pair of UAVs on each other, which cancel out. Hence, Eq. (15) yields

$$\sum_{i=1}^{N} \mathbf{e}_i = \mathbf{0}, \tag{16}$$

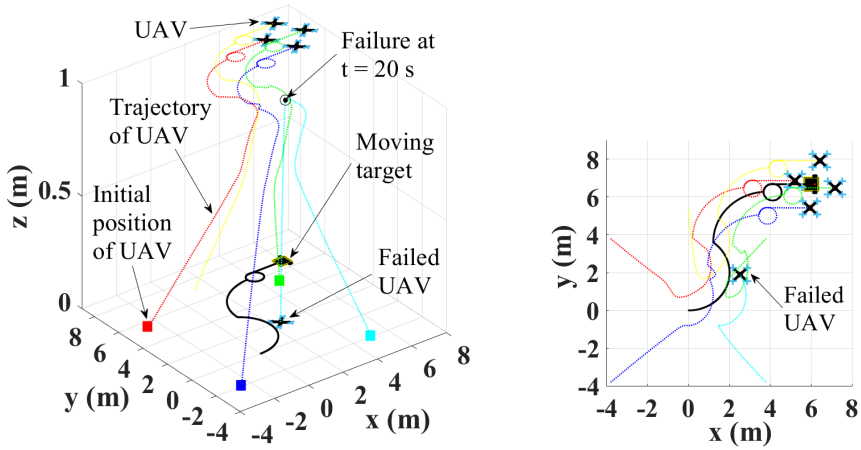
which shows that the geometric center of the polygon defined by the UAVs (the polygon vertices are the position coordinates of the UAVs) tracks the moving target.

3 Simulation Results

We evaluated our proposed controller for multi-robot target tracking in different simulated scenarios, both in MATLAB and in the Gazebo robot simulator. A video recording of all simulations described here, as well as additional simulations, is available online at [3].

3.1 MATLAB Simulations

We simulated two scenarios in which several quadrotor UAVs track a moving nonholonomic WMR while avoiding collisions with one another. The UAVs start from arbitrary initial configurations on the ground and then take off simultaneously to fly at a constant altitude above the WMR before starting the tracking task.



(a) 3D view **(b)** View in $x - y$ plane

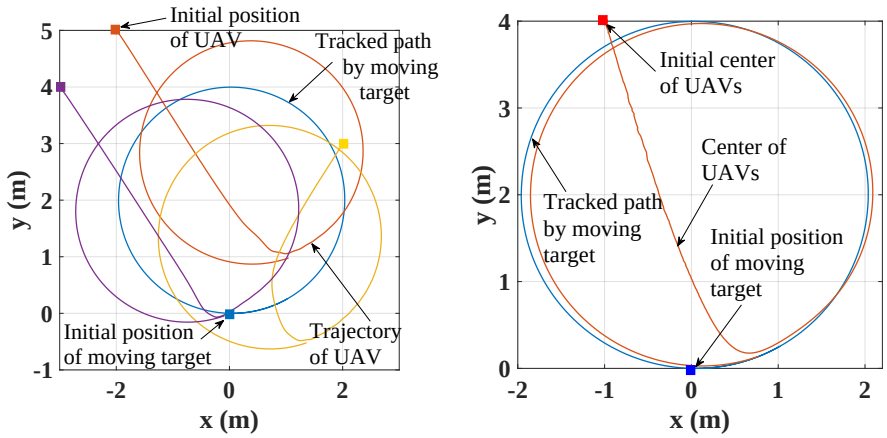
Fig. 3: Robot trajectories during a MATLAB simulation of *Scenario 2*. The position at which one UAV fails is indicated by a small black circle.

Scenario 1: Three UAVs track the target WMR as it drives along a trajectory that includes a small circle. Figure 2 plots the trajectories of the UAVs and the WMR over the duration of this simulation (60 s). As expected, the UAVs successfully converge to a fixed formation that tracks the trajectory of the WMR, enclosing its position in the $x - y$ plane, while avoiding collisions with one another.

Scenario 2: Five UAVs initially take off to track the target WMR as it drives along the same trajectory as in *Scenario 1*, as shown by the plots of the UAV and WMR trajectories in Figure 3. The UAVs converge to a fixed formation above the WMR and track its trajectory. At time $t = 20$ s, one UAV fails and drops to the ground. The remaining four UAVs reconfigure themselves into a new formation and continue to track the trajectory of the WMR for the duration of the simulation (60 s). The UAVs avoid collisions with one another throughout the simulation. This scenario demonstrates the robustness of our multi-robot target tracking approach to UAV failures.

3.2 Gazebo Simulations

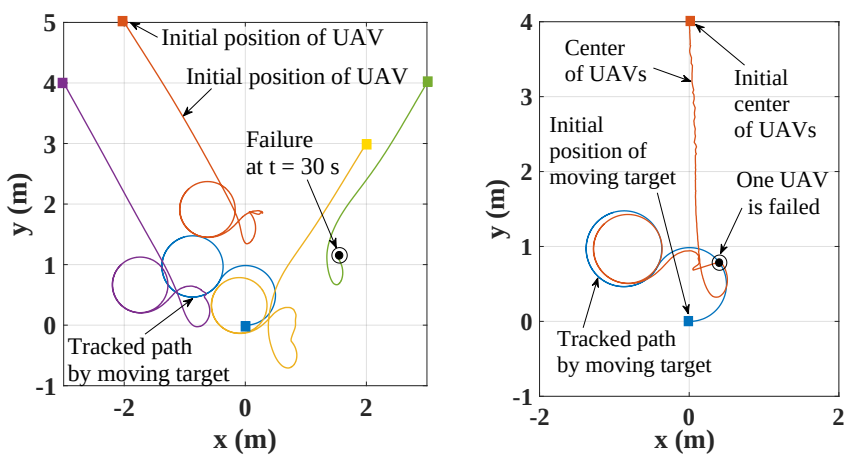
In order to evaluate our control approach in a more realistic simulation environment, we simulated two scenarios in Gazebo that were similar to *Scenarios 1* and *2* in MATLAB. There were four UAVs instead of five in *Scenario 2*, still with one UAV failure. We plot the trajectories of the moving target and the UAVs in the $x - y$ plane for both scenarios in Figs. 4a and 5a. To better visualize the tracking performance of the UAVs, we also computed the geometric center of the convex hull of the UAVs during the simulation and plotted its trajectory



(a) View in $x - y$ plane of the target and UAV trajectories

(b) View in $x - y$ plane of trajectories of the target and center of UAV formation

Fig. 4: Robot trajectories during a Gazebo simulation of *Scenario 1*.



(a) View in $x - y$ plane of the target and UAV trajectories

(b) View in $x - y$ plane of trajectories of the target and center of UAV formation

Fig. 5: Robot trajectories during a Gazebo simulation of *Scenario 2*. The position at which one UAV fails is indicated by a small black circle.

in Figs. 4b and 5b, along with the trajectory of the target. Figure 4b shows that the geometric center of the UAV formation successfully tracks the target's trajectory. Figure 5b shows that initially, the center of the four UAVs closely tracks the target, and right after a UAV fails at time $t = 30$ s, the center of the remaining three UAVs soon converges back to the target's trajectory. The UAVs avoid collisions throughout the duration of both simulations (100 s).

4 Experimental Implementation and Results

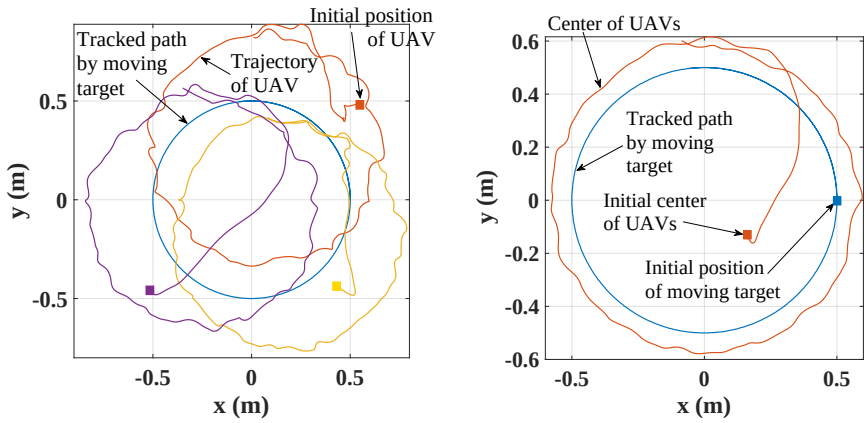
We first describe the experimental setup for all tested scenarios and provide corresponding plots of the results for each experiment. A video recording of the experimental tests described here, as well as additional tests, is also available online at [4].

4.1 Experimental Setup

In order to validate the effectiveness of our proposed control method in practice, we implement our proposed control law in Eq. (3) on physical robots. As shown in Fig. 1, Crazyflie 2.1 [1] quadrotors are used as the UAVs. Crazyflie is a small, light-weight, open-source platform commonly used for swarm and multi-robot applications. We employ multiple Crazyflies as the tracker UAVs and a Turtlebot3 Burger robot [2] as the moving target on the ground. We first plan curved trajectories for the Burger robot by setting arbitrary linear and angular velocities. This planned trajectory is not known by the UAVs. Then, we read the Burger robot's odometry information in real-time using the Robot Operating System (ROS). We then calculate the instantaneous position, velocity, and acceleration of the Burger robot and compute the control commands for the Crazyflie quadrotors accordingly. The control commands are sent to the Crazyflies using Crazyradio, a 2.4 GHz radio USB dongle. There is no explicit communication between the Crazyflies. Each Crazyflie is equipped with a Flow deck v2, which uses a VL53L1x ToF sensor to measure the distance to the ground (z-axis) and a PMW3901 optical flow sensor to measure displacement in the plane of the ground ($x - y$ plane). State estimation and PID controllers have been implemented on the Crazyflie firmware, which enable the Crazyflie to follow given position setpoints. At each time instant, the desired position setpoints are sent to the Crazyflies as control commands from a central computer through Wi-Fi communication provided by the Crazyradio between the computer and the Crazyflies.

4.2 Experimental Results

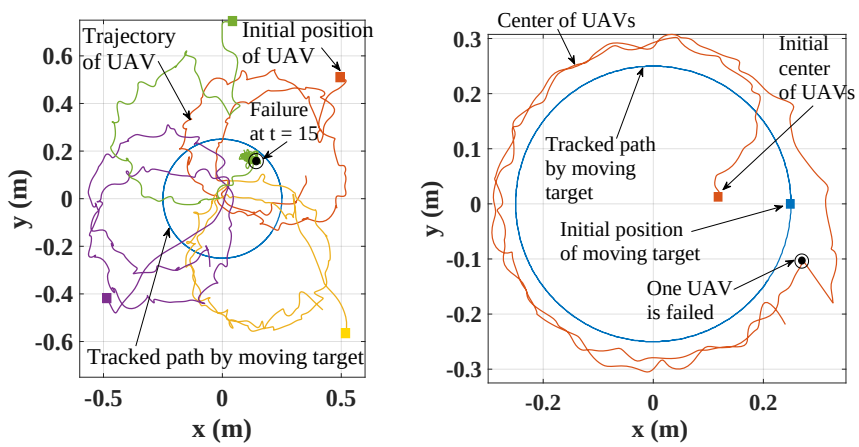
For the experimental tests, two scenarios were implemented. In *Scenario 1*, three UAVs track the moving target while avoiding collisions with one another. In *Scenario 2*, four UAVs track the target, and after 30 s, one of the UAV lands (fails), and the tracking continues with three UAVs. The second scenario is designed to demonstrate the robustness of our controller at tracking the moving target in the event of UAV failure. In all experiments, the UAVs start from arbitrary initial configurations on the ground and then take off simultaneously to fly at a constant altitude before starting to track the ground robot. The real-time positions of the UAVs and the target were collected at 20 Hz frequency, and a Butterworth low-pass filter was applied to these data with a 5 Hz cut-off frequency in MATLAB.



(a) View in $x - y$ plane of the target and UAV trajectories

(b) View in $x - y$ plane of trajectories of the target and center of UAV formation

Fig. 6: Robot trajectories during an experimental trial (duration: 30 s) of *Scenario 1*.



(a) View in $x - y$ plane of the target and UAV trajectories

(b) View in $x - y$ plane of trajectories of the target and center of UAV formation

Fig. 7: Robot trajectories during an experimental trial (duration: 40 s) of *Scenario 2*. The position at which one UAV fails is indicated by a small black circle.

The experimental results show that the UAVs successfully tracked the ground robot in each scenario while avoiding collisions. The fluctuations in the UAV trajectories can be attributed to the UAVs' low-level position control loop and near-ground altitude. For *Scenario 1*, along with the plots for trajectory of each UAV in Fig. 6a, we also plotted the trajectory of the geometric center of the group of UAVs in Fig. 6b. The group of UAVs effectively tracked the moving target without colliding with one another. *Scenario 2* was performed with four UAVs. Figure 7 shows the trajectories of the UAVs and their geometric center. At

$t = 15$ s, one UAV stops tracking and lands on the ground, emulating the failure of one member of the UAV team. This event is indicated in Fig. 7 with black arrows. After this event, the remaining three UAVs continue tracking the target and ignore the failed UAV in their collision avoidance controller. As shown in Fig. 7b, the failure event causes a sudden shift in the distance between the center of UAVs and the target. Nevertheless, this distance decreases to approximately the same distance exhibited before the failure event. Comparing the experimental results with the simulation results in Section 3, we observe similar tracking behavior by the UAVs. The main difference is the presence of fluctuations in the UAV trajectories during the experiments, which is expected in real-world implementations.

5 Conclusion

In this paper, we address the problem of tracking a moving ground target with multiple UAVs that do not have prior information about the target's motion. We analytically proved the convergence of the UAVs to a fixed formation above the target that tracks its trajectory, and we provided mathematical guarantees that the UAVs avoid collisions with one another. We validated our control approach for different scenarios in MATLAB and Gazebo simulations and in physical experiments with ground and aerial robots. In our simulations and experiments, we also demonstrated the robustness of the UAVs' target tracking performance and collision avoidance to the failure of one or more UAVs during the task. Future work includes the design of a fully decentralized tracking controller and extension of the collision avoidance strategy to enable the UAVs to perform collision-free tracking in environments with unknown and dynamic obstacles.

Appendix

We define the following Lyapunov function:

$$H = \frac{1}{2}k_2 \sum_{i=1}^{N} \sum_{j \neq i} V_{ij} + \frac{1}{2} \sum_{i=1}^{N} \left(\dot{\mathbf{e}}_i^{\mathrm{T}} \dot{\mathbf{e}}_i + \mathbf{e}_i^{\mathrm{T}} \mathbf{K}_1 \mathbf{e}_i \right). \tag{17}$$

Given the expression for V_{ij} in Eq. (2), which is a continuously differentiable function, the time derivative of H can be calculated as:

$$\dot{H} = \frac{1}{2}k_2 \sum_{i=1}^{N} \sum_{j \neq i} \dot{V}_{ij} + \sum_{i=1}^{N} \left(\dot{\mathbf{e}}_i^{\mathrm{T}} \ddot{\mathbf{e}}_i + \mathbf{e}_i^{\mathrm{T}} \mathbf{K}_1 \dot{\mathbf{e}}_i \right). \tag{18}$$

Considering the following representation for the time derivative of V_{ij},

$$\dot{V}_{ij} = \left(\nabla_{\mathbf{r}_{ij}} V_{ij} \right)^{\mathrm{T}} \dot{\mathbf{r}}_{ij}, \tag{19}$$

and given that $\nabla_{\mathbf{r}_{ij}} V_{ij} = \nabla_{\mathbf{x}_i} V_{ij} = -\nabla_{\mathbf{x}_j} V_{ij}$, we have that:

$$\dot{V}_{ij} = \left(\nabla_{\mathbf{x}_i} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{r}}_{ij} = \left(-\nabla_{\mathbf{x}_j} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{r}}_{ij}. \tag{20}$$

Furthermore, the term $\sum_{i=1}^{N} \sum_{j \neq i} \dot{V}_{ij}$ includes both $\dot{V}_{ij}$ and $\dot{V}_{ji}$ for each pair of robots i, j, where $i \neq j$. Taking into account the fact that $V_{ij} = V_{ji}$, the sum of these two terms is calculated as:

$$\begin{aligned} \dot{V}_{ij} + \dot{V}_{ji} &= \left(\nabla_{\mathbf{x}_i} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{r}}_{ij} + \left(\nabla_{\mathbf{x}_j} V_{ji}\right)^{\mathrm{T}} \dot{\mathbf{r}}_{ji} = \left(\nabla_{\mathbf{x}_i} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{r}}_{ij} + \left(-\nabla_{\mathbf{x}_i} V_{ji}\right)^{\mathrm{T}} \left(-\dot{\mathbf{r}}_{ij}\right) \\ &= \left(\nabla_{\mathbf{x}_i} (V_{ij} + V_{ji})\right)^{\mathrm{T}} \dot{\mathbf{r}}_{ij} = 2 \left(\nabla_{\mathbf{x}_i} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{r}}_{ij}. \end{aligned} \tag{21}$$

Moreover, if we define $\mathbf{e}_{ij} := \mathbf{e}_i - \mathbf{e}_j$, we can confirm that $\dot{\mathbf{e}}_{ij} = \dot{\mathbf{r}}_{ij}$. Incorporating this into Eq. (21), we can write:

$$\frac{1}{2}\left(\dot{V}_{ij} + \dot{V}_{ji}\right) = \left(\nabla_{\mathbf{x}_i} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{e}}_{ij} = \left(\nabla_{\mathbf{x}_i} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{e}}_i + \left(\nabla_{\mathbf{x}_j} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{e}}_j.$$

Therefore, the first summation in Eq. (18) can be written as:

$$\frac{1}{2} k_2 \sum_{i=1}^{N} \left(\sum_{j \neq i} \dot{V}_{ij} \right) = k_2 \sum_{i=1}^{N} \sum_{j \neq i} \left(\nabla_{\mathbf{x}_i} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{e}}_i. \tag{22}$$

Then, solving Eq. (8) for $\ddot{\mathbf{e}}_i$ and substituting this expression into Eq. (18), we obtain Eq. (10) for $\dot{H}$:

$$\begin{aligned} \dot{H} &= k_2 \sum_{i=1}^{N} \sum_{i \neq j} \left(\nabla_{\mathbf{x}_i} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{e}}_i + \sum_{i=1}^{N} \left(-\dot{\mathbf{e}}_i^{\mathrm{T}} \mathbf{\Lambda} \mathbf{K}_1 \dot{\mathbf{e}}_i - \dot{\mathbf{e}}_i^{\mathrm{T}} \left(k_2 \sum_{j \neq i} \nabla_{\mathbf{x}_i} V_{ij} \right) \right) \\ &= \underbrace{k_2 \sum_{i=1}^{N} \sum_{i \neq j} \left(\nabla_{\mathbf{x}_i} V_{ij}\right)^{\mathrm{T}} \dot{\mathbf{e}}_i - \dot{\mathbf{e}}_i^{\mathrm{T}} k_2 \sum_{i=1}^{N} \sum_{j \neq i} \nabla_{\mathbf{x}_i} V_{ij}}_{=\,0} + \sum_{i=1}^{N} \left(-\dot{\mathbf{e}}_i^{\mathrm{T}} \mathbf{\Lambda} \mathbf{K}_1 \dot{\mathbf{e}}_i \right) \\ &= -\sum_{i=1}^{N} \left(\dot{\mathbf{e}}_i^{\mathrm{T}} \mathbf{\Lambda} \mathbf{K}_1 \dot{\mathbf{e}}_i \right). \end{aligned}$$

References

1. Bitcraze. `https://www.bitcraze.io/` (2021)
2. Turtlebot 3. `https://www.turtlebot.com/` (2021), Open Source Robotics Foundation, Inc.
3. A. Doroudchi, F. Zahedi, A. Salimi Lafmejani, S. M. Rezayat Sorkhabadi: Tracking of a ground moving target by multiple unmanned aerial vehicles (simulations). `https://www.youtube.com/watch?v=GqF56jXwRIs` (2020)

4. A. Salimi Lafmejani, S. M. Rezayat Sorkhabadi, A. Doroudchi, F. Zahedi: Tracking of a ground moving target by multiple unmanned aerial vehicles (experiments). `https://www.youtube.com/watch?v=G1DW2Pq_p7M` (2020)
5. Agha-Mohammadi, A.A., Ebadi, K.: Rover localization in Mars helicopter-generated aerial maps: Experimental results in a Mars-analogue environment. In: International Symposium on Experimental Robotics. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space (2018)
6. Antonelli, G., Arrichiello, F., Caccavale, F., Marino, A.: Decentralized time-varying formation control for multi-robot systems. The International Journal of Robotics Research **33**(7), 1029–1043 (2014)
7. Cristofalo, E., Leahy, K., Vasile, C.I., Montijano, E., Schwager, M., Belta, C.: Localization of a ground robot by aerial robots for GPS-deprived control with temporal logic constraints. In: International Symposium on Experimental Robotics. pp. 525–537. Springer (2016)
8. Dames, P.M.: Distributed multi-target search and tracking using the PHD filter. Autonomous Robots pp. 1–17 (2019)
9. Dille, M.: Search and pursuit with unmanned aerial vehicles in road networks. Tech. rep., Carnegie Mellon University, The Robotics Institute, Pittsburgh, PA (2013)
10. Grover, J., Liu, C., Sycara, K.: Deadlock analysis and resolution in multi-robot systems (extended version). arXiv preprint arXiv:1911.09146 (2019)
11. Khalil, H.K.: Nonlinear systems. Prentice Hall, Upper Saddle River, N.J. (1996)
12. Khan, A., Rinner, B., Cavallaro, A.: Cooperative robots to observe moving targets. IEEE Transactions on Cybernetics **48**(1), 187–198 (2016)
13. Koenig, N., Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). vol. 3, pp. 2149–2154. IEEE (2004)
14. Mademlis, I., Nikolaidis, N., Tefas, A., Pitas, I., Wagner, T., Messina, A.: Autonomous unmanned aerial vehicles filming in dynamic unstructured outdoor environments [applications corner]. IEEE Signal Processing Magazine **36**(1), 147–153 (2018)
15. Marcotte, R.: Adaptive Communication for Mobile Multi-Robot Systems. Ph.D. thesis, University of Michigan (2019)
16. Mellinger, D.W.: Trajectory generation and control for quadrotors. Ph.D. thesis, University of Pennsylvania (2012)
17. Omidshafiei, S., Agha-Mohammadi, A.A., Amato, C., Liu, S.Y., How, J.P., Vian, J.: Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions. The International Journal of Robotics Research **36**(2), 231–258 (2017)
18. Reis, J.C., Lima, P.U., Garcia, J.: Efficient distributed communications for multi-robot systems. In: Robot Soccer World Cup. pp. 280–291. Springer (2013)
19. Wang, L., Ames, A.D., Egerstedt, M.: Safety barrier certificates for collisions-free multirobot systems. IEEE Transactions on Robotics **33**(3), 661–674 (2017)
20. Zohdi, T.I.: Multiple UAVs for mapping: A review of basic modeling, simulation, and applications. Annual Review of Environment and Resources **43**, 523–543 (2018)