



# COVER: cross-vehicle transition framework for quadrotor control in air-ground cooperation

Qiuyu Ren<sup>1,2</sup> · Miao Xu<sup>2</sup> · Mengke Zhang<sup>1,2</sup> · Nanhe Chen<sup>1,2</sup> · Mingwei Lai<sup>1,2</sup> · Chao Xu<sup>1,2</sup> · Fei Gao<sup>1,2</sup> · Yanjun Cao<sup>1,2</sup>

Received: 13 April 2025 / Revised: 3 July 2025 / Accepted: 11 August 2025

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2025

## Abstract

UAV transitions across UGVs enable diverse air-ground cooperation (AGC) applications, such as cross-vehicle landing, delivery, and rescue. However, achieving precise and efficient transitions across multiple moving UGVs without prior knowledge of their trajectories remains highly challenging. This paper proposes COVER, a cross-vehicle transition framework for quadrotor control in AGC scenarios. In COVER, the UAV is directly controlled in UGVs' body frames as non-inertial frames, thus eliminating all dependencies in the world frame. Each transition process is divided into three stages: the initial stage, transition stage, and final stage, with pre-set stage transition points and stage-varying system states. Then, an optimal reference trajectory is generated at each stage by solving a non-linear programming (NLP) problem. The effect of the target UGV's rotation on the initial relative velocity is eliminated to obtain a dynamically feasible and smooth transition reference trajectory. Finally, we design a stage-adaptive model predictive control (SAMPC) method, proposing a novel MPC position reference mode to avoid indirect routes at the transition stage. The SAMPC method effectively mitigates the flight instability caused by reference frame transition and eliminates the effect of reference frame rotation at the transition stage. And it can flexibly adapt to accurate requirements at the final stage by switching position reference mode and adjusting cost weights. Simulation benchmarks and extensive real-world experiments validate that our approach can achieve smooth, short-distance, and accurate cross-vehicle operations.

**Keywords** Motion control · Multiple non-inertial frames · Air-ground cooperation · Trajectory optimization · Model predictive control

## 1 Introduction

With the rapid development of unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV) technologies, air-ground cooperation (AGC) applications with UAVs and UGVs, such as autonomous landing (Niu et al., 2021; Ji

et al., 2022; Gao et al., 2023; Wang et al., 2022; Vlantis et al., 2015), leader-follower systems (Di Caro et al., 2021; Shi et al., 2023; Chen et al., 2024), and transportation (Pei et al., 2023; Bachetti et al., 2022; Santos Cardoso et al., 2023; Xu et al., 2024) are attracting strong attention. In some AGC scenarios, UAVs need to interact across multiple UGVs to

✉ Yanjun Cao  
yanjunhi@zju.edu.cn

Qiuyu Ren  
ren.qy@zju.edu.cn

Miao Xu  
miaoxy010522@gmail.com

Mengke Zhang  
mkzhang233@zju.edu.cn

Nanhe Chen  
nanhe\_chen@zju.edu.cn

Mingwei Lai  
laimingwei0216@gmail.com

Chao Xu  
cxu@zju.edu.cn

Fei Gao  
fgaoaa@zju.edu.cn

<sup>1</sup> The State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China

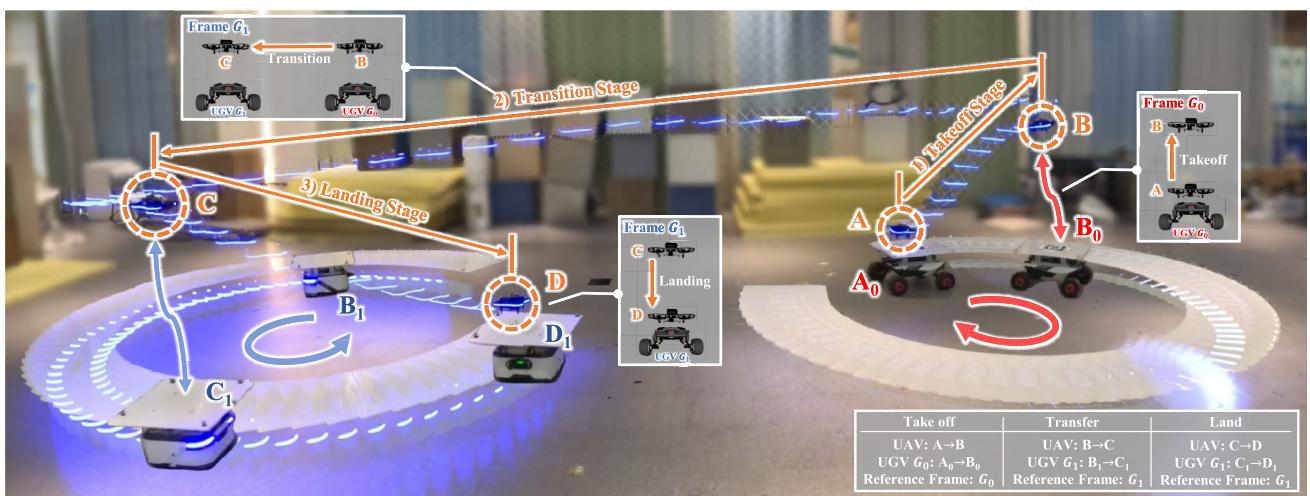
<sup>2</sup> Huzhou Institute of Zhejiang University, Huzhou 313000, China

accomplish complex tasks such as cross-vehicle landing, delivering parcels, and rescue, which we summarize as cross-vehicle transitions based on their key feature. During each transition process, the UAV will start from an initial UGV, then transit to another target UGV, and finally finish its task on the target UGV. For example, in a cross-vehicle landing task, the UAV takes off from UGV  $G_0$ , transits close to UGV  $G_1$ , and finally lands on UGV  $G_1$ , as shown in Fig. 1. Some experiments were conducted using carefully planned UGVs and controlling the UAV to follow a defined path, which can be fragile in real-world tasks. This becomes very challenging if both the initial and target UGVs move individually throughout the transition process and the UAV does not have any prior knowledge of their trajectories.

We expect an ideal transition process to meet several requirements. Firstly, the UAV's trajectory should be sufficiently smooth to conserve energy and minimize jitter. Secondly, while approaching the target UGV at the desired velocity, the UAV is supposed to minimize its traveling distance to avoid unnecessary energy loss. Last but not least, the UAV is expected to move accurately along the designated path on the target UGV to successfully complete tasks such as landing and delivery. In classical pipelines, planning and control of the UAV are performed in the world frame. To meet the accuracy requirement, precise absolute state estimations for both UAV and UGV are necessary. However, in challenging environments, such as GPS-denied or feature-sparse areas, achieving long-term robust localization is difficult. Additionally, adapting to changes in UGVs' motion demands high-frequency target motion prediction and trajectory re-planning, which is quite computationally expensive.

To overcome these challenges, we choose to directly control the UAV in UGVs' body frames (non-inertial frames), avoiding the use of the world frame as an intermediate frame for the robot control. In our recent work, CoNi-MPC (Zhang et al., 2023), we formulate a non-linear MPC controller of the UAV in the non-inertial frame utilizing the relative states and IMU data of the UAV and UGV. This framework enables accurate control of the UAV in the non-inertial frame and eliminates dependencies that are vital when controlling in world frame based solutions. However, since CoNi-MPC only handles the UAV control in a single non-inertial frame, there are still some challenges to address in order to achieve multiple non-inertial frames based UAV transitions, as reflected in the following aspects:

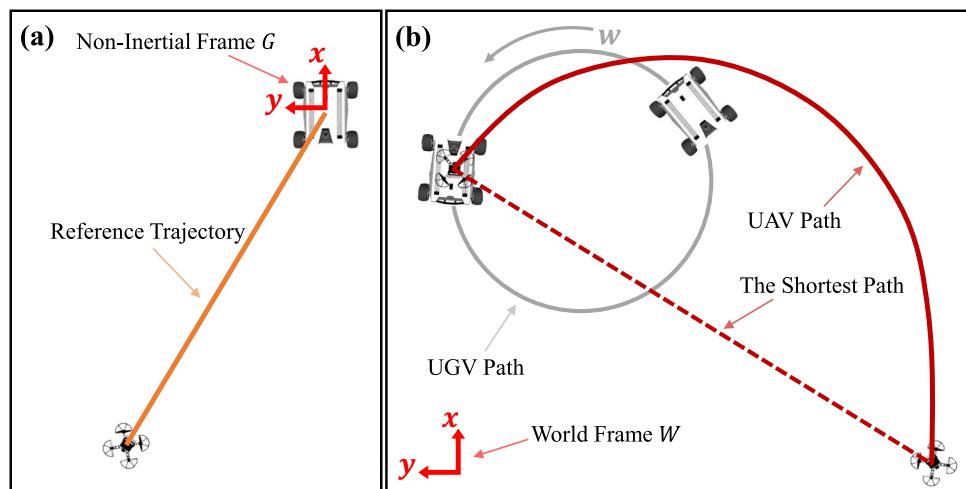
- In each transition process, the non-inertial frame in which the UAV operates (the reference frame) will transit. It is crucial to determine **the timing of the transition** and **the UAV's dynamic states (relative position, velocity, etc.)** during the transition process.
- When the reference frame transits, the rotation of the new target UGV may cause the UAV's initial relative velocity to change abruptly (explained in Sect. 4.2). And it's quite challenging to perform planning and control **under a very high initial velocity**.
- It's hard to address **the detour issue** in non-inertial frame based frameworks. As explained in Fig. 2, if the UAV is required to follow the reference trajectory precisely in the reference frame ( $x$ - $y$ - $z$ -ref mode), when it is far from the target UGV and the UGV has a large angular velocity, its travel path length in the world frame may increase considerably, which is called the detour issue.



**Fig. 1** A UAV transits across two UGVs ( $G_0$  on the right and  $G_1$  on the left, both moving in a circular path) in their non-inertial frames. It first takes off from UGV  $G_0$  in the frame  $G_0$  (from point  $A$  to  $B$ ), then transits to UGV  $G_1$  in the frame  $G_1$  (from point  $B$  to  $C$ ), and finally

lands on UGV  $G_1$  in the frame  $G_1$  (from point  $C$  to  $D$ ). The frame in which the UAV is located (the reference frame) transits at the transition stage. Points  $A_0$ - $B_0$  and  $B_1$ - $D_1$  represent the two UGVs' positions at the corresponding timestamps

**Fig. 2** Explanation of the detour issue. **a** The UAV is expected to transit to UAV  $G$  along a straight trajectory in the frame  $G$ . **b** UGV  $G$  moves in a circular path in the world frame with an angular velocity of  $w$ . If the UAV follows the reference trajectory precisely in the frame  $G$ , its transition path in the world frame will be significantly longer than the shortest possible transition path



Taking the above challenges into account, we propose COVER, a cross-vehicle transition control framework for quadrotor in AGC scenarios. In COVER, the UAV is controlled in UGVs' non-inertial frames, and each transition process is divided into three stages: the initial stage, transition stage, and final stage. In each transition task with a UAV, we define the first reference UGV as the initial UGV, then the UAV transits to the target UGV, and finally completes the task on the target UGV. The reference frame will transit at the transition stage, and we define the stage-varying system state and provide the differential model of the system. To control stage transitions, we set a target position for each stage, which serves as the stage transition point. By solving a non-linear programming (NLP) problem in the reference frame, a spatial-temporal optimal reference trajectory from the UAV's initial position to the stage transition point is generated. Notably, at the transition stage, we eliminate the effect of the target UGV's rotation on the UAV's initial velocity to achieve a dynamically feasible and smooth reference trajectory. After that, we design a stage-adaptive MPC (SAMPC) method, proposing a novel MPC position reference mode that utilizes the horizontal distance and the  $z$ -axis position as the position reference ( $xy$ - $z$ -ref mode). The controller effectively mitigates the flight instability caused by reference frame transition and eliminates detouring issues induced by reference frame rotation at the transition stage to achieve smooth and short-distance transitions. At the final stage, it flexibly adapts to accurate requirements by switching the position reference mode and adjusting cost weights accordingly. Additionally, we customize the velocity reference at the transition stage to further eliminate the effects of the target UGV's rotation. Simulation benchmarks and extensive real-world experiments demonstrate that our method can achieve smooth, short-distance, and accurate cross-vehicle transitions, highlighting its practical application potential across various scenarios. To the best of our

knowledge, this work presents the first solution for cross-vehicle transition control in AGC scenarios based purely on relative localization.

The contributions of our work are summarized as follows:

- We propose COVER, the first cross-vehicle transition control framework based on multiple non-inertial frames. This approach eliminates all the dependencies on the world frame, enabling the UAV to achieve smooth, short-distance, and accurate cross-vehicle transitions relying only on limited relative state estimations.
- We propose a non-inertial frame based reference trajectory generation method, which accounts for both the UAV's dynamic constraints and the effects of reference frame rotation to generate a dynamically feasible and smooth transition trajectory.
- We design a stage-adaptive MPC (SAMPC) method and introduce a novel position reference mode. The SAMPC method effectively mitigates the flight instability caused by reference frame transition and eliminates detouring issues induced by reference frame rotation at the transition stage, while flexibly adapting to accurate requirements at the final stage by switching position reference mode and adjusting cost weights accordingly.
- Simulation benchmarks and extensive real-world experiments are conducted to validate the performance of our method, demonstrating its potential for enabling UAV collaboration with multiple UGVs in GPS-denied or feature-sparse environments.

## 2 Related work

AGC systems have been widely studied in autonomous landing (Niu et al., 2021; Ji et al., 2022; Gao et al., 2023; Wang et al., 2022; Vlantis et al., 2015), leader-follower architecture

(Di Caro et al., 2021; Shi et al., 2023; Chen et al., 2024) and transportation (Pei et al., 2023; Bacheti et al., 2022; Santos Cardoso et al., 2023; Xu et al., 2024). Niu et al. (2021) present a vision-based autonomous landing method for UAV-UGV cooperation. The system leverages multiple QR codes placed on the UGV's landing platform to estimate the relative distance, velocity, and direction between the UAV and UGV. The UAV's state is derived from either Visual Inertial Odometry (VIO) or GPS. Utilizing these estimations, a velocity controller is developed using a combination of a control barrier function (CBF) and a control Lyapunov function (CLF) to enable precise quadrotor landings on the moving UGV. Ji et al. (2022) propose a novel trajectory planning method for aerial perching that allows for adaptive adjustment of terminal states and trajectory durations, rather than setting them in advance. The method minimizes tangential relative speed while ensuring safety and dynamic feasibility, making it particularly effective for micro aerial robots with low maneuverability or in constrained spaces. Gao et al. (2023) present an adaptive dynamic tracking and perching scheme for quadrotors, enhancing endurance and operational range in AGC tasks. The approach ensures robust perching through elastic visibility-aware planning to prevent occlusion and target loss, along with a flexible terminal adjustment method that synchronizes the quadrotor's state with the time-varying perching surface. Di Caro et al. (2021) address the problem of surveying large regions using an autonomous multi-robot team. The proposed method uses a leader-follower architecture, where the leader defines a convex containment region and assigns sampling tasks to the followers based on Bayesian optimization and Monte Carlo simulation. Each follower independently plans a collision-free path to maximize information gain. Pei et al. (2023) introduce a trajectory planning method for non-holonomic robotic teams operating in unstructured environments. They model the process of catching and transporting targets using a net in a continuous and differentiable manner, allowing for adaptive state collaboration among robots. The method ensures net safety through geometric constraints and is tested on a car-like robot team in both simulations and real-world experiments.

The above works perform planning and control of UAVs in the world frame, necessitating accurate state estimations in the world frame and continuous trajectory re-planning, which can be mitigated by using relative localization. Relative localization based AGC systems are more widely applicable in formation control (Han et al., 2018; DeVries & Dawkins, 2018), exploration (Li et al., 2023; Chen et al., 2024), and target tracking (Sun & Huo, 2015). Han et al. (2018) present an integration of relative localization with formation control for leader-follower networks. They develop a consensus-like relative localization scheme that

estimates real-time relative positions using velocity and distance measurements. This scheme converges exponentially under persistent excitation conditions. Combined with a Laplacian-based formation control strategy, the method ensures global asymptotic convergence to a stationary similar formation with error-free estimates and bounds in the presence of estimation errors. Li et al. (2023) propose ColAG, a system designed for autonomous navigation of blind UGVs using a single perceptive UAV. The UGVs use the data shared by the UAV through limited relative pose estimation to plan their trajectories and anticipate potential failures due to wheel odometry uncertainty and risky areas. The UAV dynamically schedules waypoints to prevent collisions among UGVs by solving the vehicle routing problem. Chen et al. (2024) propose a framework for navigating a multi-robot system consisting of UAVs and UGVs in unknown environments using only one depth camera. Their approach includes a sequential exploration and aiding localization (SEAL) strategy for the UAV, which aids UGVs with relative pose estimation and exploration, and a collision-adaptive trajectory (CAT) optimization for UGVs. The UAV provides environmental information and supports UGV localization, allowing the UGVs to navigate and maintain formations in obstacle-rich areas using only wheel odometry and UAV assistance.

In many studies (Nascimento et al., 2019; Ji et al., 2021; Romero et al., 2022; Wang et al., 2024; Zhang et al., 2023; Sun et al., 2024; Gao et al., 2023), the MPC controller is employed as the low-level controller for UAVs. Nascimento et al. (2019) present a stochastic non-linear model predictive control (SNMPC) algorithm for active target tracking in mobile robotics. This SNMPC method incorporates uncertainty prediction models for both target and robot pose estimation, enabling the robot to optimally track the target while respecting mobility, vision, and localization constraints. Ji et al. (2021) propose a corridor-based model predictive contouring control (CMPCC) method, which enhances system robustness by optimizing flight aggressiveness and tracking accuracy while using the flight corridor as a hard safety constraint. Similarly, Romero et al. (2022) propose a CMPCC approach that concurrently solves the time allocation and control problems and optimizes future states at runtime, which improves efficiency and enables real-time generation of near time-optimal paths. Wang et al. (2024) propose a hybrid non-linear model predictive control (HNMPMC) method for the suspended payload system. They introduce a hybrid dynamics model for air resistance and propose a hybrid mode prediction and switching algorithm to achieve efficient and precise control of load-carrying drones. Zhang et al. (2023) propose CoNi-MPC for cooperative multi-robot systems that operate directly in the target's body frame. The framework reduces dependencies on accurate

state estimations in the world frame and prior target motion models, offering robust control without continuous trajectory re-planning. The aforementioned MPC controllers have demonstrated notable performance in their respective tasks, yet they still lack flexibility in handling multi-stage transition tasks.

### 3 System model and cross-vehicle transition tasks

#### 3.1 The multiple non-inertial frames based system model

We consider an AGC system consisting of one UAV and  $N$  UGVs, which are denoted as  $A$  (agent frame) and  $G_0, G_1, \dots, G_{N-1}$  (non-inertial frames), respectively. The UAV needs to cooperatively interact with multiple UGVs to accomplish specific tasks such as cross-vehicle landing and delivery. The control of UAV relies solely on its relative information (relative position, velocity, etc.) in each related UGV's frame, without requiring knowledge of UGV's state in the world frame  $W$ . In other words, both planning and control of the UAV are conducted in the UGV's frame as a non-inertial frame. Supposing the  $k$ -th cooperated UGV  $G_k$  with a frame id of  $G_k$  ( $k \in \{0, 1, \dots, N-1\}$ ), the system state in the frame  $G_k$  is described as

$${}^{G_k} \mathbf{x} = [{}^{G_k} \mathbf{p}_A; {}^{G_k} \mathbf{v}_A; {}^{G_k} \mathbf{q}_A; {}^{G_k} \mathbf{a}_{G_k}; {}^{G_k} \boldsymbol{\Omega}_{G_k}; {}^{G_k} \boldsymbol{\beta}_{G_k}] \in \mathbb{R}^{19}, \quad (1)$$

where  ${}^{G_k} \mathbf{p}_A, {}^{G_k} \mathbf{v}_A \in \mathbb{R}^3$ ,  ${}^{G_k} \mathbf{q}_A \in \mathbb{R}^4$  are the relative position, velocity, and quaternion of the UAV in the frame  $G_k$ , respectively, which can be estimated from CREPES (Xun et al., 2023) or NOKOV motion capture system.  ${}^{G_k} \mathbf{a}_{G_k}, {}^{G_k} \boldsymbol{\Omega}_{G_k}, {}^{G_k} \boldsymbol{\beta}_{G_k} \in \mathbb{R}^3$  are the linear acceleration, angular velocity, and angular acceleration of  $G_k$  expressed in the frame  $G_k$ , respectively, which can be measured by the IMU attached to  $G_k$ .

The control input of the quadrotor is defined as

$$\mathbf{u} = [F; {}^A \boldsymbol{\Omega}_A] \in \mathbb{R}^4, \quad (2)$$

where  $F = \sum_{i=1}^4 F_i$  is the normalized thrust force of four UAV motors.  ${}^A \boldsymbol{\Omega}_A = [{}^A \Omega_A^x, {}^A \Omega_A^y, {}^A \Omega_A^z]^T \in \mathbb{R}^3$  is the angular velocity of the UAV.

The differential model of the system can be obtained by utilizing the quadrotor model and the chain rule of differentiation, which is directly presented by the following equations for brevity:

$${}^{G_k} \dot{\mathbf{p}}_A = {}^{G_k} \mathbf{R}_W \dot{\mathbf{t}}_{\overline{G_k A}}, \quad (3)$$

$${}^{G_k} \dot{\mathbf{p}}_A = {}^{G_k} \mathbf{v}_A = -[{}^{G_k} \boldsymbol{\Omega}_{G_k}] \times {}^{G_k} \mathbf{p}_A + {}^{G_k} \mathbf{R}_W \dot{\mathbf{t}}_{\overline{G_k A}}, \quad (4)$$

$$\begin{aligned} {}^{G_k} \dot{\mathbf{v}}_A &= -[{}^{G_k} \boldsymbol{\beta}_{G_k}] \times {}^{G_k} \mathbf{p}_A - 2[{}^{G_k} \boldsymbol{\Omega}_{G_k}] \times {}^{G_k} \mathbf{v}_A \\ &\quad - [{}^{G_k} \boldsymbol{\Omega}_{G_k}]^2 {}^{G_k} \mathbf{p}_A + {}^{G_k} \mathbf{R}_A {}^A \mathbf{F}_A - {}^{G_k} \mathbf{a}_{G_k}, \end{aligned} \quad (5)$$

$${}^{G_k} \dot{\mathbf{q}}_A = -\frac{1}{2} {}^{G_k} \boldsymbol{\Omega}_{G_k} \odot {}^{G_k} \mathbf{q}_A + \frac{1}{2} {}^{G_k} \boldsymbol{\Omega}_{G_k} \odot {}^A \boldsymbol{\Omega}_A, \quad (6)$$

$${}^{G_k} \dot{\boldsymbol{\Omega}}_{G_k} = {}^{G_k} \boldsymbol{\beta}_{G_k}, \quad (7)$$

where  $\dot{\mathbf{t}}_{\overline{G_k A}}$  is the translational vector pointing from the frame  $G_k$  to the frame  $A$ .  ${}^* \mathbf{R}_*$  represents the rotation matrix from the frame  $*$  to the frame  $*$ , and  $[(\cdot)]_\times$  represents the skew-symmetric matrix of  $(\cdot)$ .  ${}^A \mathbf{F}_A = [0; 0; F]$  is the normalized collective thrust of the quadrotor.  $\odot$  means quaternion Hamilton product. Notably, as shown in Eq. (4), the relative velocity of the UAV in the reference frame  $G_k$  is influenced not only by their relative translational velocity  $\dot{\mathbf{t}}_{\overline{G_k A}}$  but also by the reference frame's rotation  ${}^{G_k} \boldsymbol{\Omega}_{G_k}$ . Moreover, as the relative distance increases, the rotation effect becomes more pronounced.

As for  ${}^{G_k} \mathbf{a}_{G_k}$  and  ${}^{G_k} \boldsymbol{\beta}_{G_k}$ , their derivatives correspond to the UGV's linear jerk and angular jerk. These quantities are difficult to measure directly or reliably estimate, and they are high-order dynamics with relatively minor influence on the overall system behavior. Therefore, we assume the derivatives to be zero, i.e.,

$${}^{G_k} \dot{\mathbf{a}}_{G_k} = 0, {}^{G_k} \dot{\boldsymbol{\beta}}_{G_k} = 0. \quad (8)$$

This simplification has been validated to induce negligible degradation in control performance, as demonstrated in Sect. 5.

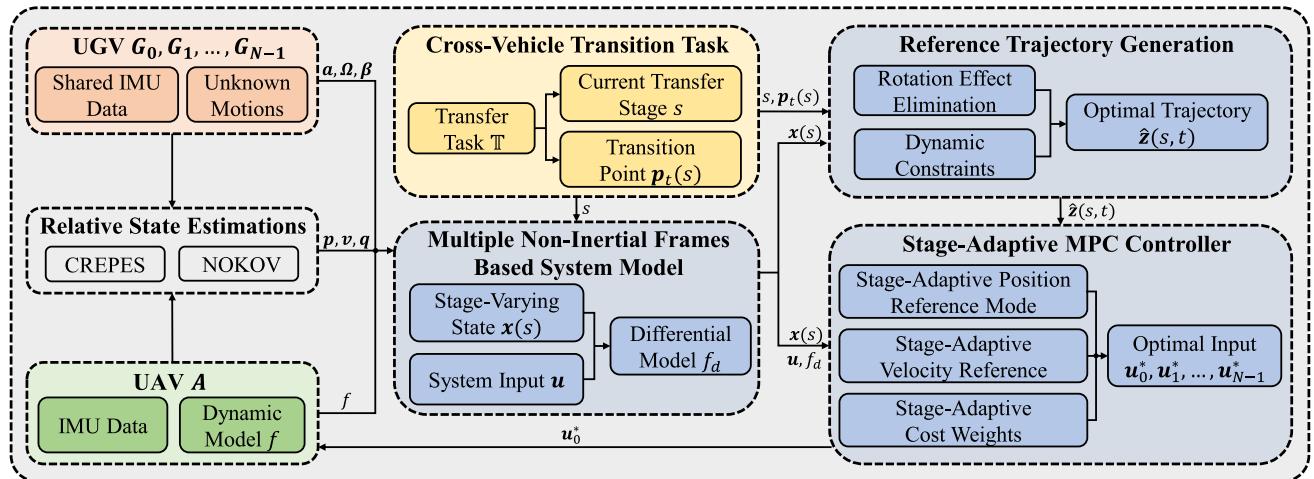
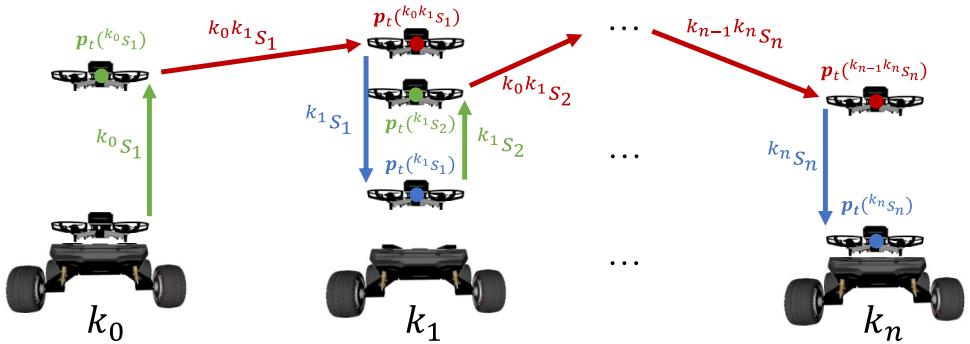
#### 3.2 Cross-vehicle transition tasks

We first define the transition task from the initial UGV  $G_i$  to the target UGV  $G_j$  ( $i, j \in \{0, 1, \dots, N-1\}$ ) as one transition process, denoted by  ${}^{G_i} \rightarrow {}^{G_j} S$ . The task is divided into three stages: (1) the initial stage, where the UAV operates in the frame  $G_i$ ; (2) the transition stage, where the UAV needs to adapt its reference frame from  $G_i$  to  $G_j$ ; and (3) the final stage, where the UAV operates in the frame  $G_j$ , which can be described as

$${}^{G_i} \rightarrow {}^{G_j} S = \{{}^{G_i} s, {}^{G_i} {}^{G_j} s, {}^{G_j} s\}, \quad (9)$$

It can be seen that the reference frame transits at the transition stage  ${}^{G_i} {}^{G_j} s$ , and we describe the stage-varying system state as

**Fig. 3** Illustration of cross-vehicle transition tasks. A transition task may consist of multiple transition processes. We set a target position for each stage of the transition process, which serves as the stage transition point. If the current transition process is complete, the UAV will execute the next transition process until the entire task is finished



**Fig. 4** Overview of the proposed framework

$$\begin{aligned} \mathbf{x}(s) &= [\mathbf{p}(s); \mathbf{v}(s); \mathbf{q}(s); \mathbf{a}(s); \boldsymbol{\Omega}(s); \boldsymbol{\beta}(s)] \\ &= \begin{cases} {}^{G_i}\mathbf{x}, & s = {}^{G_i}s, \\ {}^{G_j}\mathbf{x}, & s = {}^{G_i G_j}s \text{ or } {}^{G_j}s, \end{cases} \end{aligned} \quad (10)$$

where  $s \in {}^{G_i \rightarrow G_j}S$ . The state at the current timestamp  $t$  is denoted by  $\chi(s, t)$ , where  $t$  is reset to 0 at the beginning of each stage.

To control stage transitions, we set a stage transition point  $p_t(s)$  for each stage  $s \in {}^{G_i \rightarrow G_j}S$ , where  $p_t({}^{G_i}s)$  is located in the frame  $G_i$  while  $p_t({}^{G_i G_j}s)$  and  $p_t({}^{G_j}s)$  are located in the frame  $G_j$ . The stage transition point serves as the ending position of the current stage and the starting position of the next stage. If the UAV reaches  $p_t(s)$  at the stage  $s$  and its speed in the reference frame is sufficiently low, it will transit to the next stage and move toward the next stage transition point. This approach ensures the UAV has a stable state for the transition stage and completes its final stage tasks from a suitable initial position.

A complete user-defined transition task  $\mathbb{T}$  may consist of multiple transition processes, denoted by

$$\mathbb{T} = \{{}^{k_0 \rightarrow k_1}S_1, {}^{k_1 \rightarrow k_2}S_2, \dots, {}^{k_{n-1} \rightarrow k_n}S_n\}, \quad (11)$$

where  $k_0, \dots, k_n \in \{G_0, G_1, \dots, G_{N-1}\}$  represent UAV IDs and  $n$  is the number of transition processes. At the beginning of the task, the UAV is located in the frame  $k_0$ , and it will sequentially pass through the transition point of each stage of the first transition process from its current position. After completing the current transition process, the UAV will execute the next transition process until the entire task is completed, shown in Fig. 3. Notably, during the task, the reference frame will transit in the order of  $k_0 \rightarrow k_1 \rightarrow \dots \rightarrow k_n$ , and the task will ultimately conclude with the UAV operating in the frame  $k_n$ .

## 4 The COVER framework

### 4.1 System overview

The proposed framework overview is illustrated in Fig. 4. For a system consisting of one UAV and  $N$  UGVs, given a transition task  $\mathbb{T}$ , we first utilize CREPES (Xun et al., 2023) or NOKOV motion capture system to estimate relative states between the UAV and each UGV. Then, utilizing relative state estimations, UGVs' IMU data, and the UAV's dynamic model, we derive the multiple non-inertial frames based

system model, which includes the stage-varying system state and the system differential model. Subsequently, by solving an NLP problem, we generate a reference trajectory from the UAV's current position to the state transition point in the non-inertial frame (Sect. 4.2). Specifically, at the transition stage, we eliminate the influence of the target UGV's rotation on the UAV's initial velocity to obtain a feasible transition reference trajectory. Finally, we design a SAMPC method to control the UAV to each stage transition point (Sect. 4.3) to meet the varying requirements of different stages.

## 4.2 Reference trajectory generation

In this paper, we utilize the trajectory class  $\mathfrak{T}_{\text{MINCO}}$  (Wang et al., 2022) for reference trajectory generation, which is defined as

$$\mathfrak{T}_{\text{MINCO}} = \left\{ \mathbf{z}(t) : [0, T] \mapsto \mathbb{R}^m \mid \mathbf{c} = \mathcal{M}(\mathbf{q}, \mathbf{T}), \mathbf{q} \in \mathbb{R}^{m(M-1)}, \mathbf{T} \in \mathbb{R}_{>0}^M \right\}, \quad (12)$$

where  $\mathbf{z}(t)$  is an  $m$ -dimensional polynomial trajectory of  $N = 2n - 1$  order and  $M$  pieces.  $\mathbf{q} = [q_1, q_2, \dots, q_{M-1}]$  are the intermediate waypoints.  $\mathbf{T} = [T_1, T_2, \dots, T_M]^T$  and  $\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T, \dots, \mathbf{c}_M^T]^T \in \mathbb{R}^{2Ms \times m}$  are the durations and coefficients of the trajectory pieces, respectively. The  $i$ -th piece in  $\mathfrak{T}_{\text{MINCO}}$  is denoted by

$$\mathbf{z}_i(t) = \mathbf{c}_i^T \beta(t), \quad (13)$$

where  $\beta(t) = [1, t, \dots, t^N]^T$  is the natural basis. The function  $\mathcal{M}$  enables the computation of  $\mathbf{c}$  from  $\mathbf{q}$  and  $\mathbf{T}$  with linear complexity, allowing any second-order continuous cost function  $\mathcal{J}(\mathbf{c}, \mathbf{T})$  with available gradient applicable to  $\mathfrak{T}_{\text{MINCO}}$ .

In each transition process, at the beginning of each stage  $s$ , we obtain an initial front-end path by uniformly interpolating points between the initial position  $\mathbf{p}(s, 0)$  and the stage transition point  $\mathbf{p}_t(s)$ , and then optimize it utilizing  $\mathfrak{T}_{\text{MINCO}}$ . The trajectory in the stage  $s$  is denoted by  $\mathbf{z}(s, t)$ . To balance trajectory smoothness and transition efficiency, we define the cost function as a trade-off between minimum jerk (i.e.,  $n = 3$ ) and minimum time, which is formulated as

$$\begin{aligned} \min_{\mathbf{c}, \mathbf{T}} \mathcal{J}_m(s, \mathbf{c}, \mathbf{T}) &= \sum_{i=1}^M \left( \int_0^{T_i} \|\mathbf{z}_i^{(3)}(s, t)\| + \rho T_i \right), \\ \text{s.t. } s &\in {}^{G_i \rightarrow G_j} S, \\ \mathbf{z}(s, 0) &= \mathbf{p}_0(s), \quad \mathbf{z}(s, T) = \mathbf{p}_t(s), \\ \mathbf{z}^{(1)}(s, 0) &= \mathbf{v}_0(s), \quad \mathbf{z}^{(1)}(s, T) = \mathbf{0}, \\ \mathbf{z}^{(2)}(s, 0) &= \mathbf{0}, \quad \mathbf{z}^{(2)}(s, T) = \mathbf{0}, \\ \|\mathbf{z}^{(1)}(s, t)\| &\leq v_m, \quad \|\mathbf{z}^{(2)}(s, t)\| \leq a_m, \quad \forall t \in [0, T] \end{aligned} \quad (14)$$

where  $\rho$  is a tunable variable.  $\mathbf{p}_0(s) = \mathbf{p}(s, 0)$  is the initial position of stage  $s$ .  $\mathbf{v}_0(s)$  is the initial velocity of stage  $s$ .  $v_m$  and  $a_m$  in (14) are velocity and acceleration bounds, which are defined as

$$\begin{cases} v_m = {}^W v_{mA} - {}^W v_{mG}, \\ a_m = {}^W a_{mA} - {}^W a_{mG}, \end{cases} \quad (15)$$

where  ${}^W v_{mA}$  ( ${}^W a_{mA}$ ) and  ${}^W v_{mG}$  ( ${}^W a_{mG}$ ) are the maximum velocity (acceleration) of the UAV and UGV in the world frame, respectively. To eliminate dynamic constraints, we formulate them into the following penalty terms:

$$\begin{cases} \mathcal{G}_v = \|\mathbf{z}_i^{(1)}(s, t)\|^2 - v_m^2 \leq 0, & \forall t \in [0, T_i], \\ \mathcal{G}_a = \|\mathbf{z}_i^{(2)}(s, t)\|^2 - a_m^2 \leq 0, & \forall t \in [0, T_i], \end{cases} \quad (16)$$

which can be further transformed into the penalized sampled function  $\mathcal{J}_d$  via the time integral method (Jennings & Teo, 1990), denoted by

$$\mathcal{J}_d(s, \mathbf{c}, \mathbf{T}) = \sum_* \sum_{i=1}^M \mathcal{I}_i^*, * = \{v, a\}, \quad (17)$$

$$\mathcal{I}_i^* = \frac{T_i}{\kappa_i} \sum_{j=0}^{\kappa_i} \bar{\omega}_j \max\{\mathcal{G}_*(\mathbf{c}_i, T_i, \frac{j}{\kappa_i}), 0\}^3, \quad (18)$$

where  $\kappa_i$  is the number of samples on each piece of the trajectory and  $[\omega_0, \omega_1, \dots, \omega_{\kappa_i-1}, \omega_{\kappa_i}] = [\frac{1}{2}, 1, \dots, 1, \frac{1}{2}]$  are the quadrature coefficients resulting from the trapezoidal rule. The corresponding gradients of  $\mathcal{I}^*$  can be calculated by

$$\frac{\partial \mathcal{I}_i^*}{\partial \mathbf{c}_i} = \frac{\partial \mathcal{I}^*}{\partial \mathcal{G}_*} \frac{\partial \mathcal{G}_*}{\partial \mathbf{c}_i}, \quad \frac{\partial \mathcal{I}_i^*}{\partial T_i} = \frac{\mathcal{I}^*}{T_i} + \frac{\partial \mathcal{I}^*}{\partial \mathcal{G}_*} \frac{\partial \mathcal{G}_*}{\partial T_i}, \quad (19)$$

$$\frac{\partial \mathcal{I}^*}{\partial \mathcal{G}_*} = 3 \frac{T_i}{\kappa_i} \sum_{j=0}^{\kappa_i} \bar{\omega}_j \max\{\mathcal{G}_*(\mathbf{c}_i, T_i, \frac{j}{\kappa_i}), 0\}^2, \quad (20)$$

$$\frac{\partial \mathcal{G}_*}{\partial \mathbf{c}_i} = \frac{\partial \mathcal{G}_*}{\partial \mathbf{z}_i^{(k)}} \frac{\partial \mathbf{z}_i^{(k)}}{\partial \mathbf{c}_i} = 2 \mathbf{z}_i^{(k)} \boldsymbol{\beta}^{(k)}, \quad (21)$$

$$\frac{\partial \mathcal{G}_*}{\partial T_i} = \frac{\partial \mathcal{G}_*}{\partial \mathbf{z}_i^{(k)}} \frac{\partial \mathbf{z}_i^{(k)}}{\partial T_i} = 2 \frac{j}{\kappa_i} \mathbf{z}_i^{(k)} \mathbf{z}_i^{(k+1)}, \quad (22)$$

where  $k = 1$  in the velocity penalty term and  $k = 2$  in the acceleration penalty term. After dynamic constraints elimination, the constrained NLP problem (14) is transformed into an unconstrained NLP problem.

As explained in Sect. 3.1, the velocity of the UAV in the frame  $G_j$  is influenced by the rotation of  $G_j$  and the relative position of them. Therefore, when the UAV transits to the transition stage, if the angular velocity of the reference frame is large and the horizontal distance between them is significant, the initial relative velocity may exceed the velocity bound, i.e.,  $\|v(s, 0)\| > v_m$ . This will result in a large gradient from the velocity penalty term, leading to a non-smooth transition reference trajectory, shown in Fig. 5(b). Therefore, we utilize the relative translational velocity as the initial velocity of the reference trajectory by subtracting the component induced by the reference frame's rotation, with  $v_0(s)$  represented by

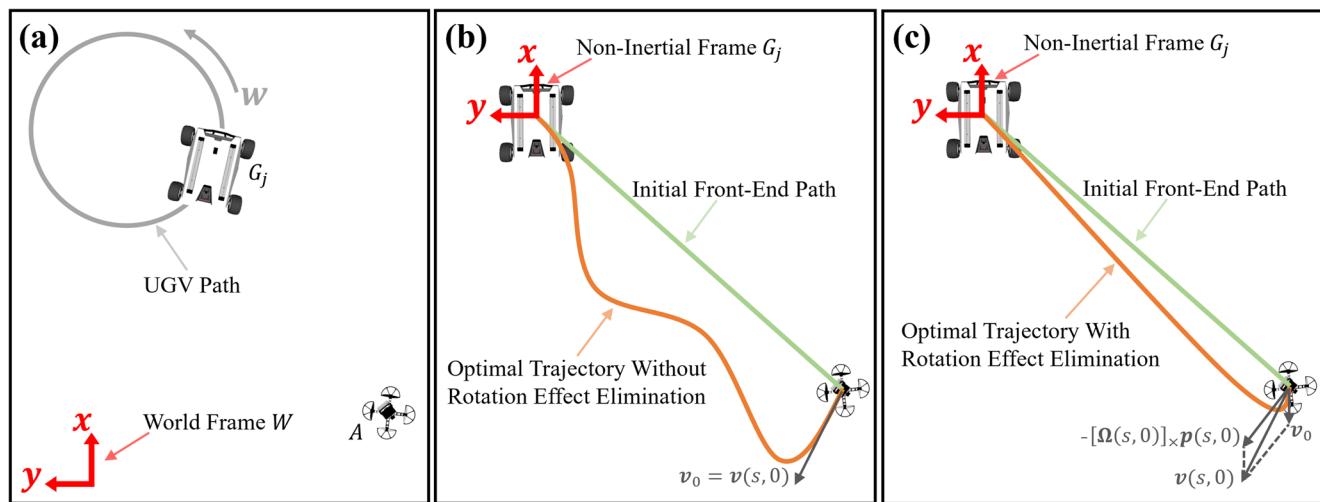
$$v_0(s) = \begin{cases} v(s, 0) + [\Omega(s, 0)]_{\times} p(s, 0), & s = {}^{G_i}G_j s, \\ v(s, 0), & s = {}^{G_i}s \text{ or } {}^{G_j}s. \end{cases} \quad (23)$$

where  $v(s, 0) + [\Omega(s, 0)]_{\times} p(s, 0) = {}^{G_j}R_W t_{\overrightarrow{G_j A}}$  is actually the relative translational velocity. This design eliminates the impact of the reference frame's rotation, enabling a more reasonable initial velocity to generate a dynamically feasible and smooth transition reference trajectory (shown in Fig. 5c).

Finally, we solve the unconstrained NLP problem utilizing the L-BFGS (Liu & Nocedal, 1989) method to obtain an optimal reference trajectory for each stage  $s$ , denoted by  $\hat{z}(s, t)$ .

### 4.3 Stage-adaptive model predictive control

After generating the reference trajectory, we design a stage-adaptive MPC (SAMPC) method to control the UAV to each pre-set target position.



**Fig. 5** Elimination of the target UGV's rotation effect at the transition stage. **a** At the beginning of the transition stage, UGV  $G_j$  moves in a circular path. **b** The optimal reference trajectory with the initial veloc-

As mentioned in Sect. 1, the detour issue may be serious at the transition stage due to the requirement for the UAV to accurately move along the reference trajectory ( $x$ - $y$ - $z$ -ref mode) in the non-inertial frame. This can be improved by reference trajectory re-planning, but it is time-consuming and may lead to instability in the UAV's flight. In fact, at the transition stage, we only require the UAV to reduce its horizontal distance from the target UGV at a desired velocity. Therefore, we propose a novel  $xy$ - $z$ -ref mode that utilizes the horizontal distance and the  $z$ -axis position as the MPC reference position. The SAMPC cost function is designed as

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} \quad & \sum_{k=0}^{N-1} (\|\mathbf{x}(s, k) - \hat{\mathbf{x}}(s, k)\|_{Q(s, k)} \\ & + \|d(s, k) - \hat{d}(s, k)\|_{Q_d(s, d_0, k)} \\ & + \|\mathbf{u}(k) - \hat{\mathbf{u}}_h\|_{R(s, k)}) \\ & + \|\mathbf{x}(s, N) - \hat{\mathbf{x}}(s, N)\|_{Q_N(s)} \\ & + \|d(s, N) - \hat{d}(s, N)\|_{Q_{Nd}(s, d_0)}, \\ \text{s.t. } \quad & s \in {}^{G_i \rightarrow G_j} S, \\ & d(s, k) = \|p^x(s, k)\|^2 + \|p^y(s, k)\|^2, \\ & \mathbf{x}(s, 0) = \chi(s, t), \\ & \mathbf{x}(s, k+1) = f_d(\mathbf{x}(s, k), \mathbf{u}(k)), \\ & F_{min} \leq F \leq F_{max}, \\ & \|{}^A\Omega_A^x\| \leq \Omega_{rp}, \|{}^A\Omega_A^y\| \leq \Omega_{rp}, \\ & \|{}^A\Omega_A^z\| \leq \Omega_{yaw}, \end{aligned} \quad (24)$$

where  $N$  is the total number of discrete time steps and  $k \in \{0, 1, \dots, N-1\}$  denotes the time step index.  $\mathbf{x}(s, k)$  and  $\mathbf{u}(k)$  are the discrete stage-varying state and input determined by Eqs. (10) and (2).  $\|\cdot\|_M = (\cdot)^T M (\cdot)$ .  $\mathbf{u}_h = [g; 0; 0; 0]$  represents the

ity of  $v(s, 0)$  ( $s = {}^{G_i}G_j s$ ) is infeasible and non-smooth. **c** After eliminating the effect of  $G_j$ 's rotation on the initial velocity of the UAV, the optimal reference trajectory becomes dynamically feasible and smooth

hover input.  $d_0 = d(s, 0) = \|p^x(s, t)\|^2 + \|p^y(s, t)\|^2$  is the square of the horizontal distance between the UAV and the reference frame in each control iteration.  $\hat{d}(s, k) = \|\hat{z}^x(s, t + k\Delta t)\|^2 + \|\hat{z}^y(s, t + k\Delta t)\|^2$  is the square of the horizontal distance reference and  $\Delta t$  is the MPC time step.  $f_d$  is the differential model determined by Eqs. (3)–(8).  $F_{min}$  and  $F_{max}$  are the minimum and maximum thrust constraints, respectively.  $\Omega_{rp}$  is the maximum roll and pitch angular speed, and  $\Omega_{yaw}$  is the maximum yaw angular speed. The state reference  $\hat{x}$  and the corresponding cost matrix  $Q$  are denoted by

$$\hat{x}(s, k) = [\hat{p}(s, k); \hat{v}(s, k); \hat{q}(s, k); \mathbf{0}; \mathbf{0}; \mathbf{0}], \quad (25)$$

$$Q(s, k) = e^{-\varepsilon_0 k} \text{diag}(Q_p(s), Q_v(s), Q_q(s), \mathbf{0}, \mathbf{0}, \mathbf{0}), \quad (26)$$

where  $\hat{p}(s, k) = \hat{z}(s, t + k\Delta t)$ .  $\varepsilon_0$  is the decay weight to prioritize the cost at the current time over future time.

We customize MPC references and the corresponding cost weights to meet the specific requirements of each stage. Firstly, we utilize the  $xy$ - $z$ -ref mode at stage  $G_i G_j s$  to address the detour issue, and utilize the  $x$ - $y$ - $z$ -ref mode at stage  $G_i s$  and  $G_j s$  to meet the accuracy requirement. Mode transitions are achieved by customizing the corresponding cost weights  $Q_p$  and  $Q_d$ , denoted by

$$Q_p(s) = \begin{cases} \text{diag}(0, 0, Q_{0p}^z), & s = G_i G_j s, \\ \text{diag}(Q_{1p}^x, Q_{1p}^y, Q_{1p}^z), & s = G_i s \text{ or } G_j s, \end{cases} \quad (27)$$

$$Q_d(s, k, d_0) = \begin{cases} e^{-\varepsilon_1 k} (q_0 e^{\varepsilon_2 d_0} + q_1), & s = G_i G_j s, \\ 0, & s = G_i s \text{ or } G_j s, \end{cases} \quad (28)$$

where  $Q_{0p}^*$ ,  $Q_{1p}^*$ ,  $q_0$  and  $q_1$  are constant cost weights.  $\varepsilon_1$  and  $\varepsilon_2$  are decay weights. We increase  $Q_d$  as  $d_0$  decreases at the transition stage because we expect the UAV to move more smoothly when it is far from  $G_j$  and to reach the stage transition point accurately as it approaches  $G_j$ .

**Fig. 6** Illustration of MPC velocity reference customization at the transition stage. **a** In the world frame, UGV  $G_j$  moves at the angular speed of  $w$ , and the UAV is expected to move in the direction of  $-\hat{t}^d$  in the  $x$ - $y$  plane. **b** We can compute the UAV's expected velocity in the frame  $G_j$  based on its world frame expected velocity and  $G_j$ 's angular speed

To further eliminate the impact caused by the rotation of  $G_j$  at the transition stage, we customize the reference velocity  $\hat{v}_{s_t}^d$  based on the reference trajectory. As shown in Fig. 6, at the transition stage, the UAV is expected to move toward the target UGV on the  $x$ - $y$  plane in the world frame, i.e., the horizontal component of  $\hat{t}_{G_j A}$  is supposed to be

$$\hat{t}^d = -\frac{\hat{t}^d}{\|\hat{t}^d\|} v_0^d, \quad (29)$$

where  $\hat{t}^d = [t_{G_j A}^x, t_{G_j A}^y, 0]^T$ .  $v_0^d = (\|\hat{z}^{(1)x}(s, t)\|^2 + \|\hat{z}^{(1)y}(s, t)\|^2)^{\frac{1}{2}}$  is the reference horizontal velocity at the current timestamp  $t$ . Then we can obtain the desired UAV horizontal velocity  $\hat{v}_{s_t}^d$  in the frame  $G_j$  based on Eq. (4), denoted by

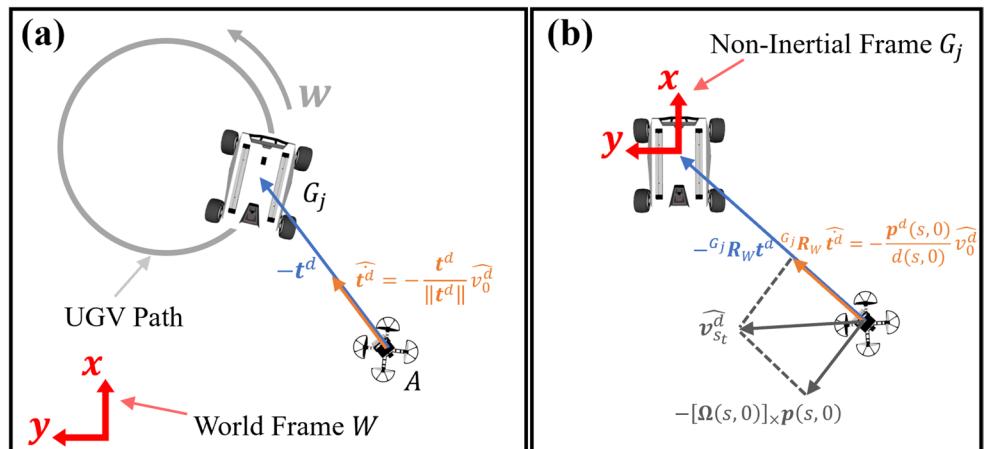
$$\begin{aligned} \hat{v}_{s_t}^d &= -[\Omega(s, 0)] \times p(s, 0) + {}^{G_j} R_W \hat{t}^d \\ &= -[\Omega(s, 0)] \times p(s, 0) + {}^{G_j} R_W \frac{\hat{t}^d}{\|\hat{t}^d\|} v_0^d \\ &= -[\Omega(s, 0)] \times p(s, 0) + \frac{p^d(s, 0)}{d(s, 0)} v_0^d, \end{aligned} \quad (30)$$

where  $p^d(s, 0) = [p^x(s, 0), p^y(s, 0), 0]^T$ . Finally,  $\hat{v}$  and the corresponding cost weight  $Q_v$  are denoted by

$$\hat{v}(s, k) = \begin{cases} [(\hat{v}_{s_t}^d)^x, (\hat{v}_{s_t}^d)^y, \hat{z}^{(1)z}(s, 0)]^T, & s = G_i G_j s, \\ \hat{z}^{(1)}(s, t + k\Delta t), & s = G_i s \text{ or } G_j s, \end{cases} \quad (31)$$

$$Q_v(s) = \begin{cases} \text{diag}(Q_{0v}^x, Q_{0v}^y, Q_{0v}^z), & s = G_i G_j s, k = 0, \\ \text{diag}(0, 0, Q_{0v}^z), & s = G_i G_j s, k > 0, \\ \text{diag}(Q_{1v}^x, Q_{1v}^y, Q_{1v}^z), & s = G_i s \text{ or } G_j s, \end{cases} \quad (32)$$

where  $(\hat{v}_{s_t}^d)^x$  and  $(\hat{v}_{s_t}^d)^y$  represent the  $x$ -axis and  $y$ -axis components of  $\hat{v}_{s_t}^d$ , respectively.  $Q_{0v}^*$  and  $Q_{1v}^*$  are constant cost weights. Noted that we only apply the horizontal velocity cost to the initial state ( $k = 0$ ) at stage  $G_i G_j s$ , as the



horizontal position of the UAV for  $k > 0$  cannot be predetermined to calculate the corresponding desired horizontal velocity.

As for the UAV's attitude, we make its head in the direction toward the target UGV at stage  ${}^{G_i}G_j s$ , and align its head with  $G_i$  ( $G_j$ ) at stage  ${}^{G_i}s$  ( ${}^{G_j}s$ ). So  $\hat{q}$  is denoted by

$$\hat{q}(s, k) = \begin{cases} [\cos(\hat{\psi}/2); 0; 0; \sin(\hat{\psi}/2)], & s = {}^{G_i}G_j s, \\ [1; 0; 0; 0], & s = {}^{G_i}s \text{ or } {}^{G_j}s, \end{cases} \quad (33)$$

where  $\hat{\psi} = \arctan(p^y(s, t)/p^x(s, t))$  is the desired yaw angle at the transition stage. The corresponding cost weight terms of  $q$  are denoted by

$$Q_q(s) = \begin{cases} Q_{0q}, & s = {}^{G_i}G_j s, \\ Q_{1q}, & s = {}^{G_i}s \text{ or } {}^{G_j}s, \end{cases} \quad (34)$$

where  $Q_{0q}$  and  $Q_{1q}$  are constant matrices.

The input cost weight matrix  $R$  are denoted by

$$R(s, k) = \begin{cases} e^{-\varepsilon_3 k} R_0, & s = {}^{G_i}G_j s, \\ e^{-\varepsilon_3 k} R_1, & s = {}^{G_i}s \text{ or } {}^{G_j}s, \end{cases} \quad (35)$$

where  $R_0$  and  $R_1$  are constant matrices.  $\varepsilon_3$  is the decay weight. We configure two sets of cost weights to tailor the UAV's performance to the specific requirements of each stage. For example, at stage  ${}^{G_i}G_j s$ , we prioritize smoother flight by assigning higher roll and pitch angular rate cost weights, i.e.,  $R_0^r > R_1^r$ ,  $R_0^p > R_1^p$ . And at stage  ${}^{G_j}s$ , we emphasize precise control of the UAV's z-axis position by setting a higher z-axis position cost weight, i.e.,  $Q_{0p}^z < Q_{1p}^z$ . As for the terminal cost matrices  $Q_N$  and  $Q_{Nd}$ , their configurations at each stage are similar to those of  $Q$  and  $Q_d$ .

With the configuration of the SAMPC controller, the implementation utilizes the ACADO toolkit (Houska et al., 2011) under the control frequency of 100 Hz. We obtain the optimal input sequence  $u_0^*, u_1^*, \dots, u_{N-1}^*$ , and  $u_0^*$  is selected as the UAV's current control input.

## 5 Experiments

To validate the proposed framework, we conduct benchmark experiments in simulation. Additionally, extensive real-world tests are performed to demonstrate the applicability of our method across diverse scenarios.

### 5.1 Simulation benchmarks

We benchmark our method against CoNi-MPC (Zhang et al., 2023) in simulation and also evaluate the performance of our method without utilizing the customized MPC velocity

reference at the transition stage (i.e.  $Q_{0v}^x = Q_{0v}^y = 0$ , noted by "w.o. Ref.  $V_{xy}$ "). The simulation is conducted on a laptop equipped with an Intel Core i9-13900 H CPU. The simulation environment is based on Gazebo, with the quadrotor model based on PX4-Autopilot,<sup>1</sup> and visualization is performed in Rviz.

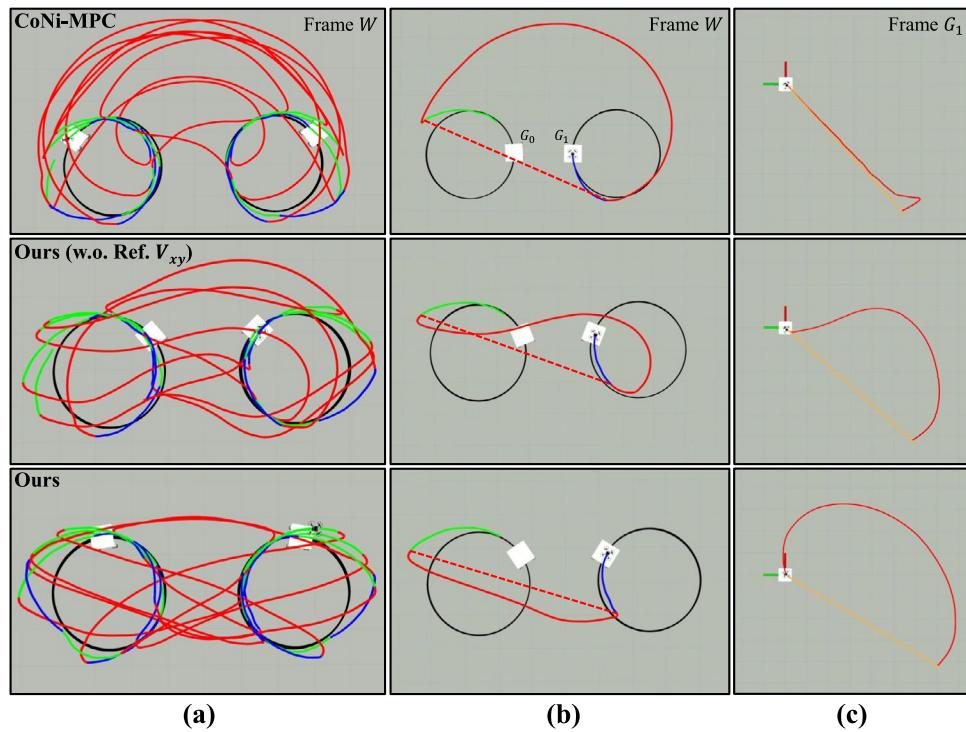
Firstly, to evaluate the performance of our method at the transition stage, we set up a scenario where the UAV cyclically transits across two UGVs  $G_0$  and  $G_1$ , i.e., the transition task  $\mathbb{T} = \{ {}^{G_0} \rightarrow {}^{G_1} S_0, {}^{G_1} \rightarrow {}^{G_0} S_1, {}^{G_0} \rightarrow {}^{G_1} S_2, \dots \}$ . Transition points are set to  $p_t({}^{G_i}s_k) = [0, 0, 1.5 \text{ m}]^T$ ,  $p_t({}^{G_i}G_j s_k) = [0, 0, 1.5 \text{ m}]^T$ , and  $p_t({}^{G_j}s_k) = [0, 0, 0.3 \text{ m}]^T$ , where  $k \in \{0, 1, 2, \dots\}$ ,  $G_i, G_j \in \{G_0, G_1\}$ . Each UGV moves in a circular path with a linear velocity of 0.5 m/s and an angular velocity of 0.35 rad/s. The maximum speed of the UAV in the reference frame is set to 1.0 m/s. We draw UAV's world frame history paths of each method, shown in Fig. 7a. As illustrated by the red transition paths, our method enables smooth and short-distance transitions under varying initial relative positions and velocities. And in one transition process  ${}^{G_0} \rightarrow {}^{G_1} S$ , the UAV's world frame paths and its non-inertial frame transition paths are shown in Fig. 7b and c, respectively. It can be observed that in CoNi-MPC, the detour issue at the transition stage is quite severe, which is caused by the requirement for the UAV to strictly follow the reference trajectory in the non-inertial frame. In contrast, our method effectively addresses the detour issue, allowing the UAV to achieve a relatively short transition path in the world frame while lacking knowledge of world frame information. Additionally, when comparing the transition paths with and without the customized MPC velocity reference, it is evident that using the customized velocity reference efficiently guides the UAV to move toward the target UGV.

Furthermore, we conduct cyclic transition experiments with different UGV linear and angular velocities. For each method, we compute the UAV's total transition path length (denoted by  $L_t$ ) in the world frame, the total shortest transition path length (the straight-line length from the start to the end position of the UAV at the transition stage, shown by the red dashed lines in Fig. 7b, denoted by  $L_s$ ), and their ratio ( $L_t/L_s$ ), shown in Table 1. We can see that in our method, the UAV's transition path in the world frame is close to the shortest path (the ratio is less than 1.1), while in CoNi-MPC the ratio is higher and increases greatly as the target UGV's angular speed grows. Meanwhile, it can be observed that the method without the customized velocity reference results in some detours, indicating that the customized velocity reference further shortens the transition path.

We also evaluate the impact of reference frame transition on the UAV's flight stability. Specifically, we analyze three

<sup>1</sup> <https://github.com/PX4/PX4-Autopilot>.

**Fig. 7** Transition path comparisons between CoNi-MPC and our method without and with customized MPC velocity reference at the transition stage. **a** The UAV's world frame paths of the three methods at the UGVs' linear speed of 0.5 m/s and angular speed of 0.35 rad/s. The green, red, and blue paths represent the UAV's paths at the initial, transition, and final stages, respectively. The black paths represent UGVs' paths. **b** The UAV's world frame paths in one transition process. The red dashed path represents the shortest transition path. **c** The UAV's transition paths and the reference trajectory (in orange) in the frame  $G_1$



(a)

(b)

(c)

**Table 1** Transition path length comparison

UGV ( $v, w$ )	Method	$L_s$ (m)	$L_t$ (m)	$L_t/L_s$
(0.5, 0.25)	CoNi-MPC	77.77	100.67	1.29
	Ours(w.o. Ref. $V_{xy}$ )	68.09	73.94	1.09
	Ours	62.18	64.06	<b>1.03</b>
(0.5, 0.35)	CoNi-MPC	63.30	120.14	1.90
	Ours(w.o. Ref. $V_{xy}$ )	62.33	79.22	1.27
	Ours	66.25	70.79	<b>1.07</b>
(0.8, 0.35)	CoNi-MPC	65.00	119.59	1.84
	Ours(w.o. Ref. $V_{xy}$ )	65.18	88.52	1.36
	Ours	66.91	72.58	<b>1.08</b>

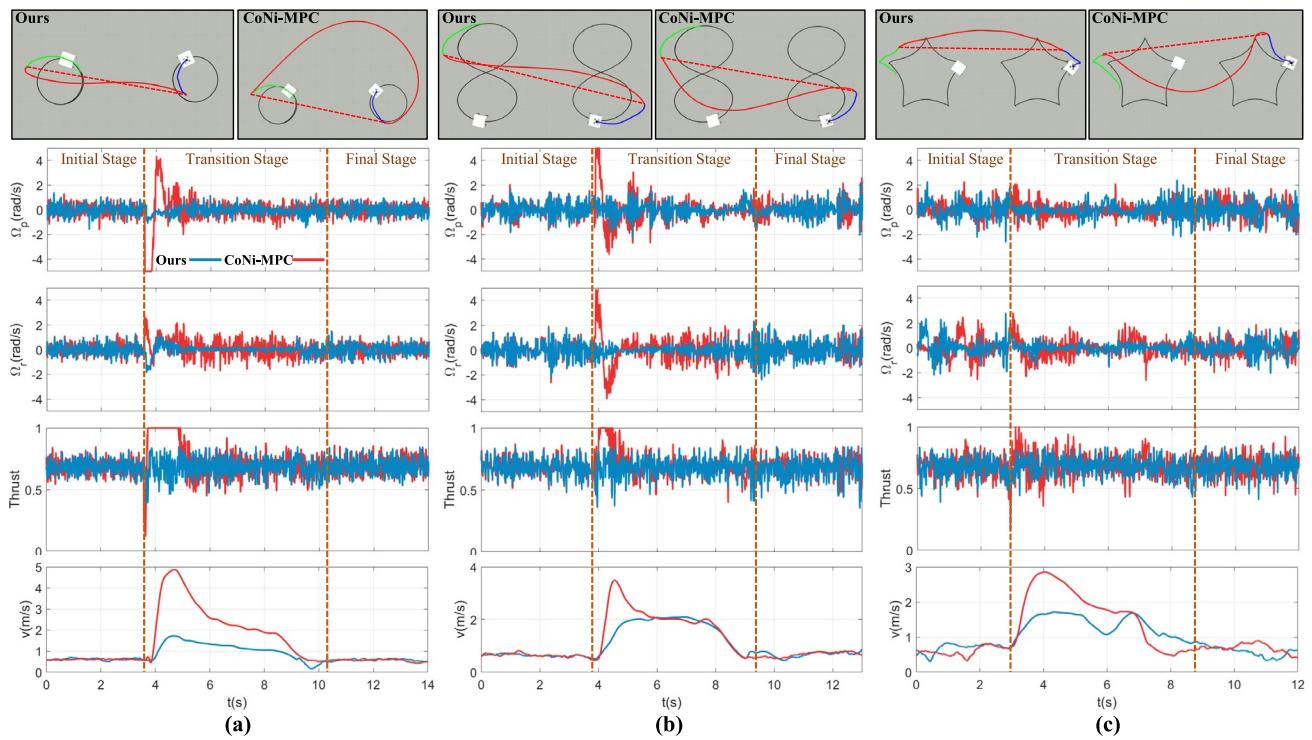
transition scenarios in which the two UGVs follow circular, eight-shaped, and star-shaped trajectories, respectively. The linear and angular velocities of the circular motion are 0.5 m/s and 0.5 rad/s, respectively. For the remaining two scenarios, the maximum velocity of the UGV is set to 0.8 m/s. We compare the transition paths of our method and CoNi-MPC across these scenarios, and plot the corresponding MPC input curves as well as the UAV's speed curves in the world frame, shown in Fig. 8. The results show that in our method, the UAV can maintain stable angular velocity and thrust inputs when the reference frame transits, and it completes the transition with a steady speed. This demonstrates that our method can effectively handle abrupt reference frame transitions under varying motion conditions. In contrast, in CoNi-MPC, the UAV exhibits severe fluctuations in angular velocity and thrust, and the flight speed increases rapidly, which indicates that reference frame transition

leads to unstable flight behaviors. Especially in the first scenario, the high angular velocity of the reference frame results in significantly increased MPC inputs. Furthermore, we plot the position trajectory curves of the UAV and the target UGV of the three transition scenarios, as shown in Fig. 9. It can be observed that our method enables the UAV to smoothly approach the target UGV, while CoNi-MPC results in unnecessary detours and noticeable instability of the UAV's altitude in the first scenario.

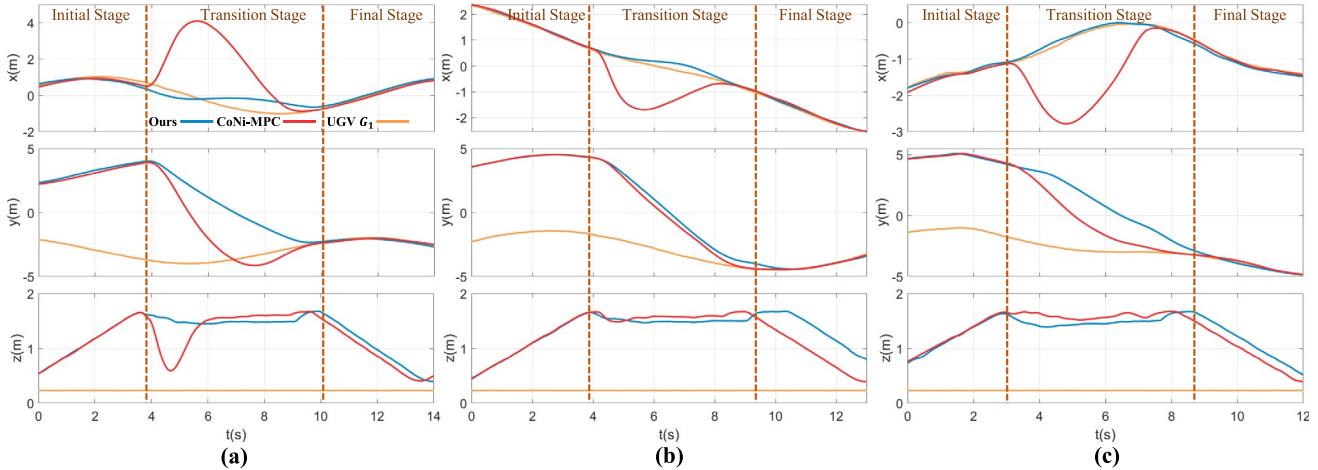
## 5.2 Real-world experiments

### 5.2.1 Indoor experiments

We conduct indoor experiments to verify our framework and test the system in different complex tasks, including cyclic transitions, cross-vehicle landing, delivery, and interaction with three UGVs. Our experiment platforms include a quadrotor UAV equipped with an airdrop mechanism and three UGVs ( $G_0$ ,  $G_1$ , and  $G_2$ ), shown in Fig. 10. The UAV has an onboard computer with an Intel Celeron J4125 processor (2–2.7 GHz) and 8GB RAM. The MPC controller takes around 5 ms computation time on average for one iteration, and the reference trajectory generation consumes around 18 ms on average at each stage. Relative state estimations between the UAV and UGVs are computed from the NOKOV motion capture system, and the IMU data of UGVs is shared with the UAV through WiFi. A first-order low-pass filter is applied to filter high-frequency noises in



**Fig. 8** Analysis of the impact of reference frame transition on UAV flight stability. We compare our method and CoNi-MPC in three transition scenarios, plotting their transition paths, MPC inputs (pitch rate, roll rate, and thrust) curves, and UAV's speed curves in the world frame



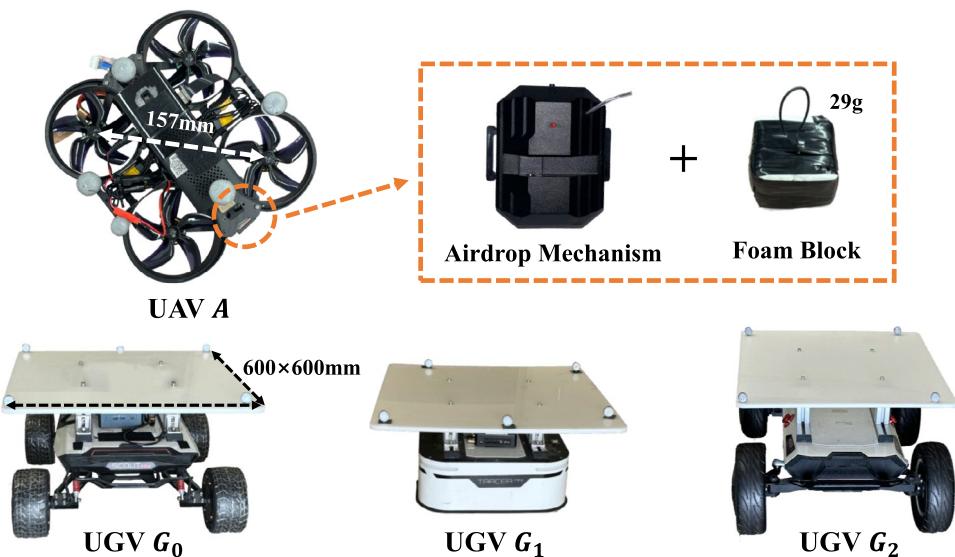
**Fig. 9** The trajectory curves of the UAV and the target UGV of the three transition scenarios

the IMU measurements. The UAV's maximum speed in the reference frame is set to 1.5 m/s in Scenarios 1 and 2, and 1.0 m/s in Scenarios 3 and 4. The cost weights of the SAMPC are listed in Table 2, and their values remain consistent across all scenarios.

*Scenario 1: Cyclic Transition Test.* We first conduct the cyclic transition test between UGV  $G_0$  and  $G_1$ . Transition points are set the same as in the simulation test. We set two motion patterns for UGVs: (1) circular motion at a linear speed of 0.8 m/s and an angular speed of 0.5 rad/s,

the distance between the two circle centers is 5 m, and (2) parallel reciprocating straight-line motion at a maximum speed of 1.0 m/s (maximum relative speed is 2.0 m/s), the distance between the two parallel routes is 2.8 m. The UAV completes more than 8 transition processes in each motion pattern. Its world frame paths and a composite image of one transition process are shown in Fig. 11. The transition path lengths of the UAV under the two motion patterns are shown in Table 3. It can be seen that the UAV is able to achieve no-detour and smooth transitions under different UGV motion

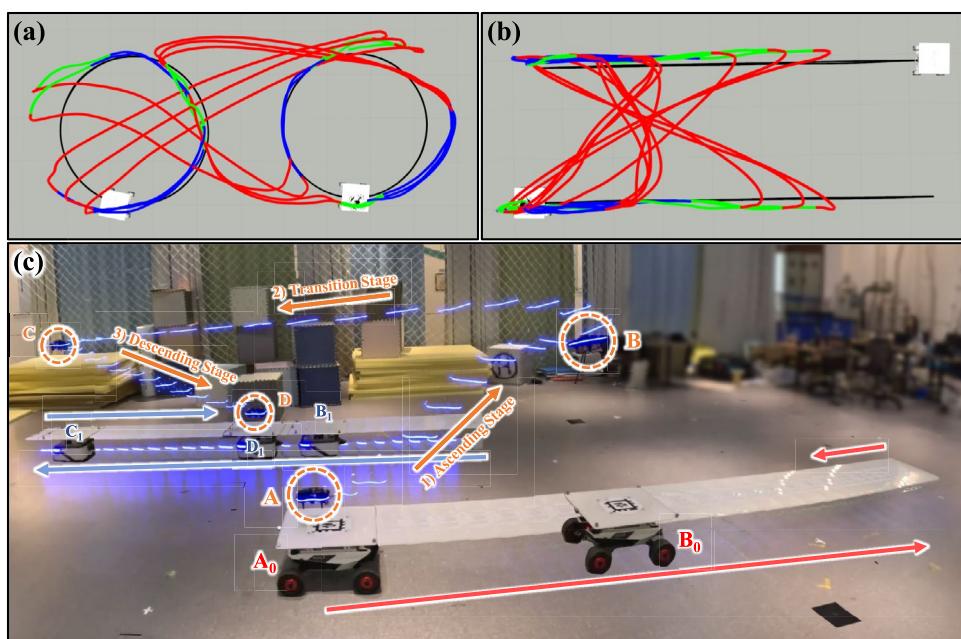
**Fig. 10** The UAV and UGV platforms in our real-world experiments. Top: The UAV is a quadrotor equipped with an airdrop mechanism. We use a foam block as the simulated parcel to perform delivery experiments. Bottom: The three UGV platforms, each equipped with a plastic plate on top, provide a surface for the UAV to take off and land



**Table 2** The cost weights of the SAMPC

$Q_{0p}/Q_{1p}$	$Q_{0v}/Q_{1v}$	$q_0/q_1$	$Q_{0g}/Q_{1g}$	$R_0/R_1$
[0, 0, 200]/ [200, 200, 300]	[350, 350, 100]/ [100, 100, 300]	200/ 100	[40, 40, 40, 40]/ [50, 50, 50, 50]	[1, 30, 30, 5]/ [1, 20, 20, 5]

**Fig. 11** In Scenario 1, the UAV cyclically transits across UGV  $G_0$  and  $G_1$ . The two UGVs have two motion patterns: circular motion and straight-line motion. **a** UAV's world frame paths when UGVs are in circular motion. **b** UAV's world frame paths when UGVs are in straight-line motion. **c** A composite image in one transition process, where the initial speed of the UAV at the transition stage approaches 2 m/s



patterns. Even with a high initial relative speed, the UAV can still maintain stable flight.

*Scenario 2: Cross-Vehicle Takeoff and Landing.* In this scenario, the UAV takes off from the top of UGV  $G_0$ , then transits to UGV  $G_1$ , and eventually lands on the top of UGV  $G_1$ , i.e.,  $\mathbb{T} = \{G_0 \rightarrow G_1 S\}$ . Transition points are set to  $p_t(G_0 s) = [0, 0, 1.0 \text{ m}]^T$ ,  $p_t(G_0 G_1 s) = [0, 0, 1.0 \text{ m}]^T$ , and  $p_t(G_1 s) = [0, 0, 0.05 \text{ m}]^T$  (the landing point). The UAV will arm automatically when reaching the landing point. The two UGVs move in opposite directions along S-shaped

curves at a maximum speed of 1.0 m/s, and the UAV's initial relative speed reaches 1.5 m/s at the transition stage. A composite image of the transition process is shown in Fig. 12, and the UAV's world frame path is shown in Fig. 13. It can be observed that the UAV transits effectively at a high speed along a short path and ultimately lands precisely on top of the target UGV.

*Scenario 3: Cross-Vehicle Delivery.* This scenario focuses on a parcel delivery task, where the UAV takes off from UGV  $G_0$  with a parcel, delivery the parcel to UGV  $G_1$ ,

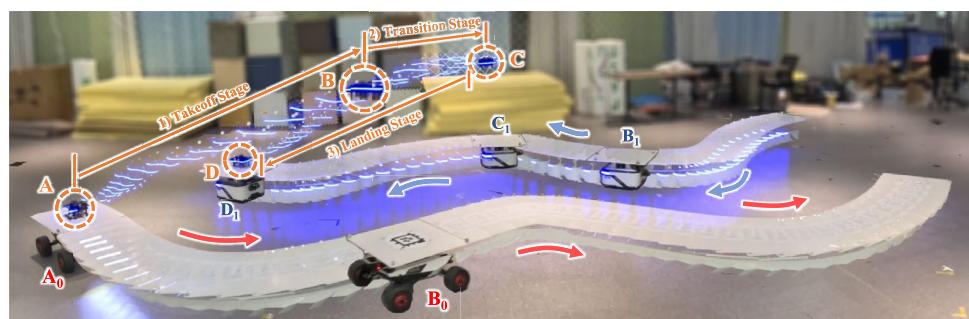
**Table 3** Transition path lengths in scenario 1

Motion Pattern	$L_s$ (m)	$L_t$ (m)	$L_t/L_s$
Circular	55.68	61.96	1.11
Straight-Line	57.67	61.79	1.07

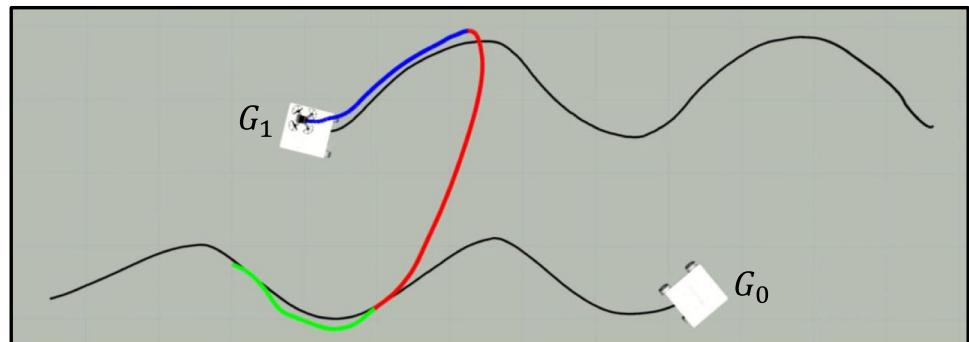
and finally lands back on UGV  $G_0$ , i.e.,  $\mathbb{T} = \{G_0 \rightarrow G_1 S_0, G_1 \rightarrow G_0 S_1\}$ . The two UGVs move in the straight-line motion pattern at a maximum speed of 0.8 m/s. Transition points are set to  $p_t(G_0 s_0) = [0, 0, 1.1 \text{ m}]^T$ ,  $p_t(G_0 G_1 s_0) = [0, 0, 1.1 \text{ m}]^T$ ,  $p_t(G_1 s_0) = [0.1 \text{ m}, 0, 0.25 \text{ m}]^T$  (the delivery point),  $p_t(G_1 s_1) = [0, 0, 1.1 \text{ m}]^T$ ,  $p_t(G_1 G_0 s_1) = [0, 0, 1.1 \text{ m}]^T$ , and  $p_t(G_0 s_1) = [0, 0, 0.05 \text{ m}]^T$  (the landing point). We use a foam block as the parcel, and an airdrop mechanism is mounted on the UAV for delivery, shown in Fig. 10. When the UAV reaches the delivery point, the parcel will be released by the airdrop mechanism. As shown in Fig. 14, the UAV successfully delivers the parcel into the basket on UGV  $G_1$  and lands back on UGV  $G_0$ .

*Scenario 4: Collaboration with Three UGVs* This scenario involves three UGVs, where the UAV takes off from UGV  $G_0$  with a parcel, delivers it to UGV  $G_1$ , and finally lands on UGV  $G_2$ , i.e.,  $\mathbb{T} = \{G_0 \rightarrow G_1 S_0, G_1 \rightarrow G_2 S_1\}$ . The UGVs move in circular paths with a linear speed of 0.8 m/s and an angular speed of 0.25 rad/s. The transition points are set similarly to those in Scenario 3. The screenshots of the two transition processes and the UAV's world frame paths are shown in Figs. 15 and 16, respectively. We can see that the UAV accurately delivers the parcel to  $G_1$  and completes its task by landing on the top of  $G_2$ . Its world frame paths indicate that it achieves short-distance and smooth transitions in the presence of multiple UGVs.

**Fig. 12** A composite image of Scenario 2. The UAV takes off from UGV  $G_0$  (from point A to B), transits to UGV  $G_1$  (from point B to C), and finally lands on UGV  $G_1$  (from point C to D). The two UGVs move along S-shaped curves, and the initial relative speed of the UAV reaches 1.5 m/s at the transition stage



**Fig. 13** The UAV's world frame path in Scenario 2

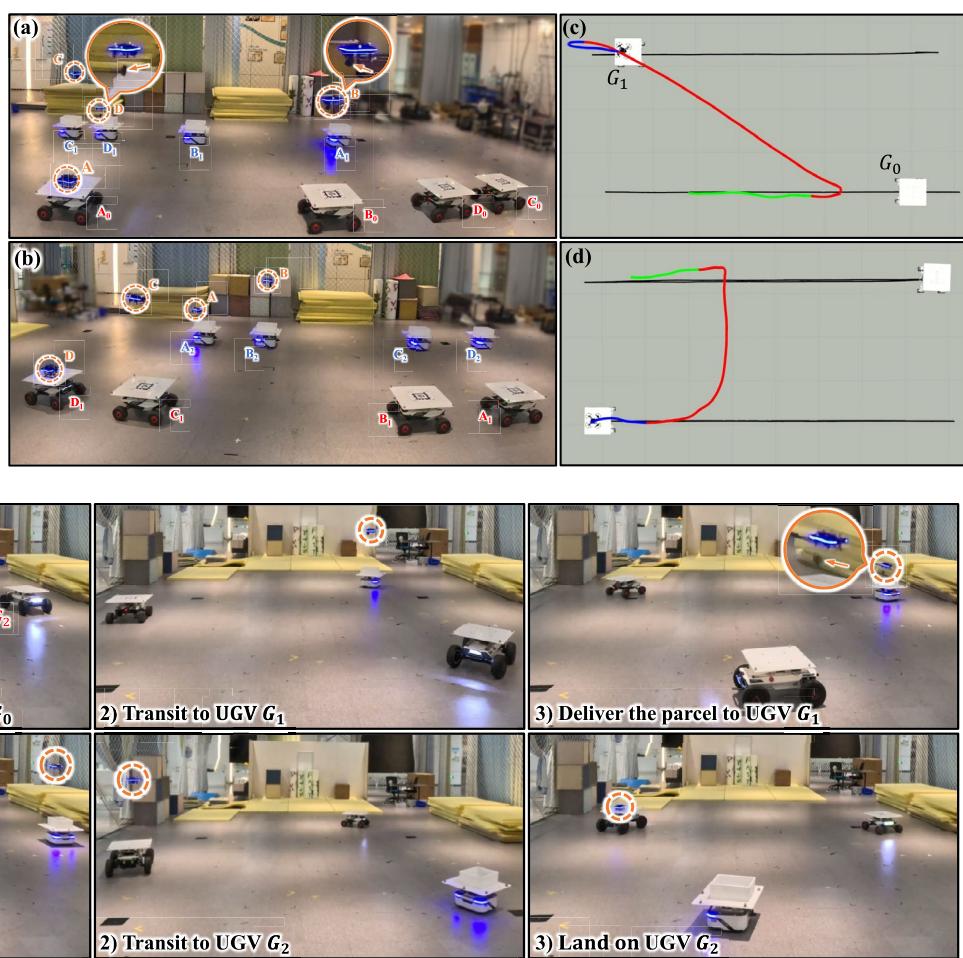


The above four experiment scenarios comprehensively validate the performance of the proposed framework, demonstrating its capability to achieve smooth, short-distance, and accurate cross-vehicle transitions. Among them, the cyclic transition test in Scenario 1 illustrates that our method effectively adapts to different UGV motion conditions (different motion directions and speeds), and mitigates the impact of high initial relative speed at the transition stage. In Scenario 2–4, the UAV successfully lands and delivers the parcel on a rapidly moving UGV, indicating that our approach balances the need for short-distance at the transition stage with the accuracy requirement at the final stage. To further demonstrate the performance of our method at the final stage, we draw the UAV's position and the corresponding reference trajectory curves and compute its average tracking errors, shown in Fig. 17 and Table 4. The value in bold indicates the smallest tracking error compared with results in other scenarios. We can see that the tracking error for each axis is within 0.08 m in Scenario 1, and is within 0.05 m in Scenarios 2 and 3, indicating that our method ensures good accuracy at the final stage. Additionally, Scenario 4 showcases the framework's ability to coordinate the UAV with multiple UGVs simultaneously (more than two), further highlighting its versatility in complex operational environments.

## 5.2.2 Outdoor experiments

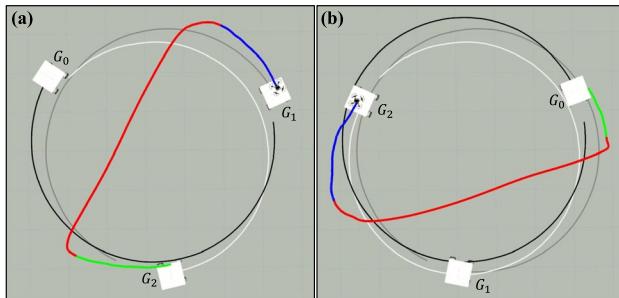
To further validate the practical value of our approach, we conduct outdoor experiments. The platforms are shown in Fig. 18, the UAV is a 250 mm quadrotor equipped with a

**Fig. 14** In Scenario 3, the UAV delivers a parcel from UGV  $G_0$  to  $G_1$  and lands back on  $G_0$ . **a** In the first transition process, the UAV drops the parcel (pointed by the orange arrow) on  $G_1$  at the point  $D$ . **b** In the second transition process, the UAV lands back on  $G_0$  from  $G_1$ . **c** and **d** The UAV's world frame paths in the first and second transition processes



**Fig. 15** Screenshots of each stage of the two transition processes in Scenario 4. Top: In the first transition process, the UAV takes off with a parcel from  $G_0$ , transits to  $G_1$ , and descends to deliver the parcel to

$G_1$ . Bottom: In the second transition process, the UAV ascends from  $G_1$ , transits to  $G_2$ , and finally lands on  $G_2$



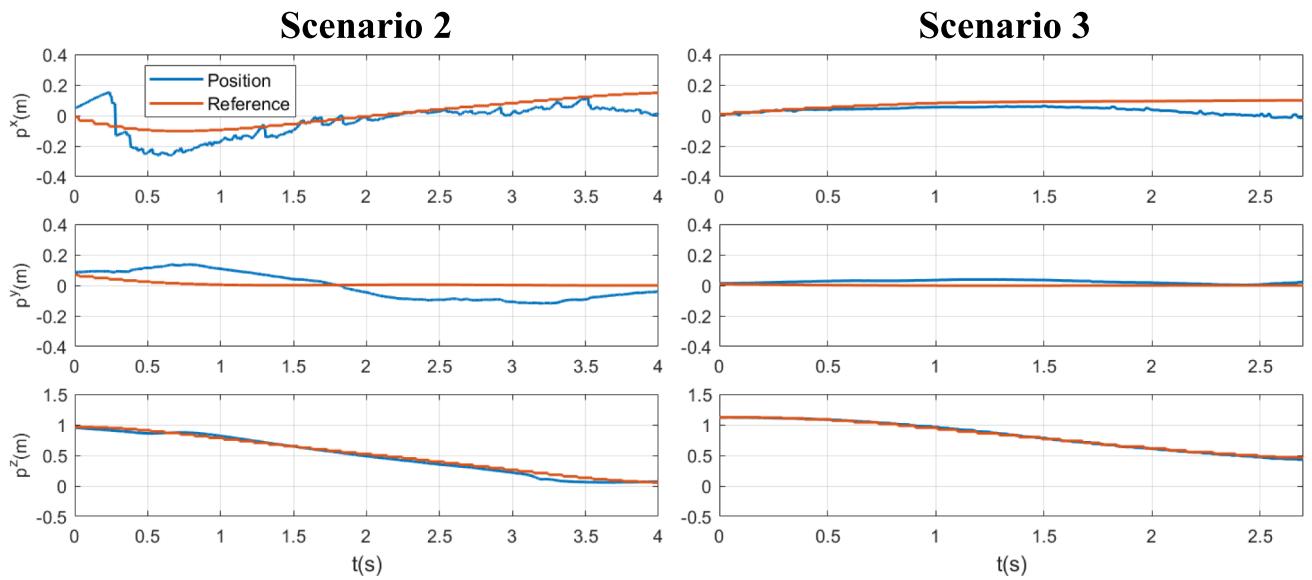
**Fig. 16** The UAV's world frame paths of the two transition processes in Scenario 4. The black, gray, and white paths represent the paths of UGV  $G_0$ ,  $G_1$ , and  $G_2$ , respectively. **a** and **b** represent the first and second transition processes, respectively

NUC-12WSKi7 onboard computer. The relative estimations between the UAV and UGVs are obtained utilizing the CREPES (Shi et al., 2023) relative localization system (shown in Fig. 18c), without the aid of GPS or any external positioning systems. The two UGVs move back and forth in a straight line at a maximum speed of 1 m/s, while the

UAV performs cyclic transitions between them with a maximum relative velocity of 1 m/s. Figure 19 illustrates the UAV transits from UGV  $G_0$  to  $G_1$ , and then transits back to UGV  $G_0$ . Figure 19a and b show the reference trajectories and UAV's paths at the two transition stages in the corresponding reference frame. The results demonstrate that, in outdoor environments, our method enables the UAV to achieve smooth and efficient cross-vehicle transitions using only relative observations with the UGVs.

## 6 Conclusion

In this paper, we propose COVER, a cross-vehicle transition framework for quadrotor control in AGC scenarios. In the framework, the UAV is directly controlled in UGVs' body frames, thus eliminating dependencies that are necessary in the world frame. We divide the transition process into three stages, defining the stage-varying system state as well as



**Fig. 17** UAV position curves and the corresponding reference trajectory curves at the final stage in Scenario 2 and Scenario 3

**Table 4** Average tracking errors at the final stage

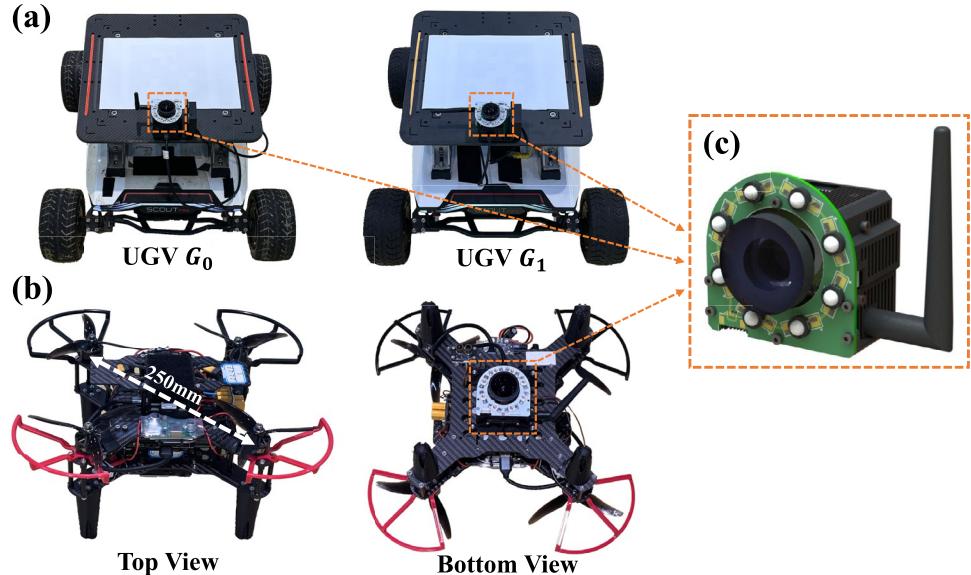
Scenario	UGV $v_{max}$	x-axis	y-axis	z-axis
Scenario 2	1.0 m/s	0.067 m	0.076 m	0.035 m
Scenario 3	0.8 m/s	0.042 m	<b>0.023 m</b>	<b>0.012 m</b>
Scenario 4	0.8 m/s	<b>0.037 m</b>	0.039 m	0.020 m

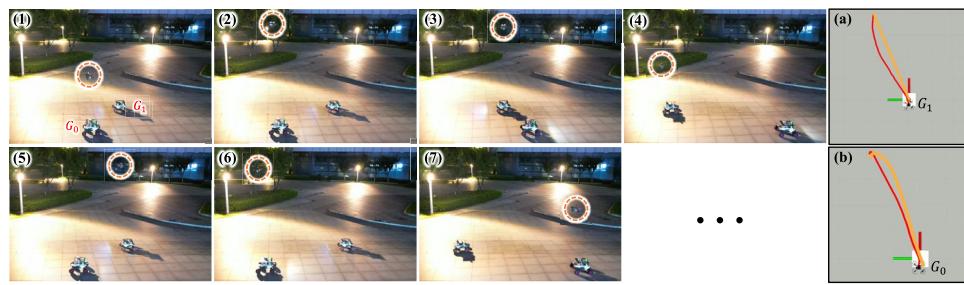
the differential system model. Then we generate an optimal reference trajectory from the UAV's initial position to the stage transition point by solving an NLP problem. The effect of the target UGV's rotation on the initial relative velocity is eliminated to obtain a more feasible transition reference trajectory. Finally, we design a stage-adaptive MPC controller, proposing a novel MPC position reference mode. The controller can balance the need for short-distance at the transition stage with the accuracy requirement at the final stage.

Extensive experimental results validate that our approach enables smooth, short-distance, and accurate cross-vehicle transitions.

However, this paper still has certain limitations. For instance, in this work, obstacle avoidance is not considered, which limits the applicability of our system to open environments. In our recent work (Zhang et al., 2025), we propose a local obstacle avoidance method that operates directly in the non-inertial frame. This method constructs a sample-based modulation matrix from LiDAR data to adjust the velocity output of a high-level planner, enabling the generation of collision-free and dynamically feasible UAV trajectories. In the future, we plan to integrate this method into our framework, enabling obstacle-aware transition trajectories through enhanced sensing and velocity

**Fig. 18** **a** and **b** The UAV and UGV platforms in the outdoor experiments. **c** The CREPES (Shi et al., 2023) relative localization system





**Fig. 19** The UAV transits from UGV  $G_0$  to  $G_1$ , and then transits back to UGV  $G_0$ . (1)–(4): The UAV ascends on UGV  $G_0$ , transits to UGV  $G_1$ , and descends on  $G_1$ . (5)–(7): The UAV ascends on UGV  $G_1$ , tran-

modulation. Furthermore, our current system supports only a single UAV. In future work, we plan to extend the framework to multi-UAV scenarios to enable collaborative operations between multiple UAVs and UGVs. Finally, we aim to deploy the proposed method in real-world applications such as logistics and agriculture to further demonstrate its practical value.

**Author Contributions** Qiuyu Ren wrote the main manuscript text and Miao Xu assisted with the real-world experiments. All authors reviewed the manuscript.

**Data availability** No datasets were generated or analysed during the current study.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

## References

- Bacheti, V. P., Brandão, A. S., & Sarcinelli-Filho, M. (2022). Leader-follower UGV-UAV formation as control paradigm for package delivery. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 772–778). IEEE.
- Chen, L., Xiao, J., Teo, C. W. R., Li, J., & Feroskhan, M. (2024). Air-ground collaborative control for angle-specified heterogeneous formations. *IEEE Transactions on Intelligent Vehicles*. <https://doi.org/10.1109/TIV.2024.3420408>
- Chen, N., Li, Z., Quan, L., Chen, X., Xu, C., Gao, F., & Cao, Y. (2024). Cost-effective swarm navigation system via close cooperation. *IEEE Robotics and Automation Letters*, 9(11), 9343–9350.
- DeVries, L., & Dawkins, J. (2018). Multi-vehicle target tracking and formation control in non-inertial reference frames. In *Annual American Control Conference (ACC)* (pp. 4026–4031).
- Di Caro, G. A., & Yousef, A. W. Z. (2021). Multi-robot informative path planning using a leader-follower architecture. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 10045–10051). IEEE.
- Gao, Y., Ji, J., Wang, Q., Jin, R., Lin, Y., Shang, Z., Cao, Y., Shen, S., Xu, C., & Gao, F. (2023). Adaptive tracking and perching for quadrotor in dynamic scenarios. *IEEE Transactions on Robotics*, 40, 499–519.
- Gao, X., Su, X., An, A., & Zhang, H. (2023). Robust model predictive tracking control for the wheeled mobile robot with boundary constraints. *IEEE Transactions on Cybernetics*, 53, 1–11.
- Han, Z., Guo, K., Xie, L., & Lin, Z. (2018). Integrated relative localization and leader-follower formation control. *IEEE Transactions on Automatic Control*, 64(1), 20–34.
- Houska, B., Ferreau, H. J., & Diehl, M. (2011). Acado toolkit—an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3), 298–312.
- Jennings, L. S., & Teo, K. L. (1990). A computational algorithm for functional inequality constrained optimization problems. *Automatica*, 26(2), 371–375.
- Ji, J., Yang, T., Xu, C., & Gao, F. (2022). Real-time trajectory planning for aerial perching. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 10516–10522). IEEE.
- Ji, J., Zhou, X., Xu, C., & Gao, F. (2021). CMPCC: Corridor-based model predictive contouring control for aggressive drone flight. In *Experimental Robotics: The 17th International Symposium* (pp. 37–46). Springer.
- Li, Z., Mao, R., Chen, N., Xu, C., Gao, F., Cao, Y. (2023). ColAG: A collaborative air-ground framework for perception-limited UGVs' navigation. arXiv preprint [arXiv:2310.13324](https://arxiv.org/abs/2310.13324)
- Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1), 503–528.
- Nascimento, T. P., Basso, G. F., Dórea, C. E., & Gonçalves, L. M. G. (2019). Perception-driven motion control based on stochastic nonlinear model predictive controllers. *IEEE/ASME Transactions on Mechatronics*, 24(4), 1751–1762.
- Niu, G., Yang, Q., Gao, Y., & Pun, M.-O. (2021). Vision-based autonomous landing for unmanned aerial and ground vehicles cooperative systems. *IEEE robotics and automation letters*, 7(3), 6234–6241.
- Pei, L., Lin, J., Han, Z., Quan, L., Cao, Y., Xu, C., & Gao, F. (2023). Collaborative planning for catching and transporting objects in unstructured environments. *IEEE Robotics and Automation Letters*, 9(2), 1098–1105.
- Romero, A., Sun, S., Foehn, P., & Scaramuzza, D. (2022). Model predictive contouring control for time-optimal quadrotor flight. *IEEE Transactions on Robotics*, 38(6), 3340–3356.
- Santos Cardoso, E., Bacheti, V. P., & Sarcinelli-Filho, M. (2023). Package delivery based on the leader-follower control paradigm for multirobot systems. In *2023 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 775–781). IEEE.
- Shi, Y., Hua, Y., Yu, J., Dong, X., Lü, J., & Ren, Z. (2023). Cooperative fault-tolerant formation tracking control for heterogeneous air-ground systems using a learning-based method. *IEEE Transactions on Aerospace and Electronic Systems*, 60(2), 1505–1518.

- Sun, L., & Huo, W. (2015). 6-DOF integrated adaptive backstepping control for spacecraft proximity operations. *IEEE Transactions on Aerospace and Electronic Systems*, 51(3), 2433–2443.
- Sun, Y., Yang, J., Zhao, D., Okonkwo, M. C., Zhang, J., Wang, S., & Liu, Y. (2024). Enhancing stability and safety: A novel multi-constraint model predictive control approach for forklift trajectory. *IET Cyber-Systems and Robotics*, 6(4), 70004. <https://doi.org/10.1049/csy2.70004>
- Vlantis, P., Marantos, P., Bechlioulis, C. P., & Kyriakopoulos, K. J. (2015). Quadrotor landing on an inclined platform of a moving ground vehicle. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2202–2207). IEEE.
- Wang, H., Li, H., Zhou, B., Gao, F., & Shen, S. (2024). Impact-aware planning and control for aerial robots with suspended payloads. *IEEE Transactions on Robotics*, 40, 2478–2497. <https://doi.org/10.1109/TRO.2024.3381555>
- Wang, P., Wang, C., Wang, J., & Meng, M.Q.-H. (2022). Quadrotor autonomous landing on moving platform. *Procedia Computer Science*, 209, 40–49.
- Wang, Z., Zhou, X., Xu, C., & Gao, F. (2022). Geometrically constrained trajectory optimization for multicopters. *IEEE Transactions on Robotics*, 38(5), 3259–3278.
- Xu, J., Liu, X., Jin, J., Pan, W., Li, X., & Yang, Y. (2024). Holistic service provisioning in a UAV-UGV integrated network for last-mile delivery. *IEEE Transactions on Network and Service Management*. <https://doi.org/10.1109/TNSM.2024.3487357>
- Xun, Z., Huang, J., Li, Z., Ying, Z., Wang, Y., Xu, C., Gao, F., & Cao, Y. (2023). Crepes: Cooperative relative pose estimation system. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5274–5281). <https://doi.org/10.1109/IROS55552.2023.10342523>
- Zhang, B., Chen, X., Chen, Q., Xu, C., Gao, F., & Cao, Y. (2025). Global-state-free obstacle avoidance for quadrotor control in air-ground cooperation. *IEEE Robotics and Automation Letters*. <http://doi.org/10.1109/LRA.2025.3568314>
- Zhang, B., Chen, X., Li, Z., Beltrame, G., Xu, C., Gao, F., & Cao, Y. (2023). CoNi-MPC: Cooperative non-inertial frame based model predictive control. *IEEE Robotics and Automation Letters*, 8(12), 8082–8089.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.