Leader–Follower-Based Design for Multi-Vehicle Cooperative Control

Mengxiang Chen, D.Munguntsetseg, M.Dovchinvanchig*

School of Applied Sciences, Mongolian University of Life Sciences, Ulaanbaatar 17024, Mongolia

**Corresponding Author*

Abstract: This paper presents a multi-UGV cooperative control scheme that integrates an improved Leader–Follower method with consensus principles, enabling dynamic formation keeping and real-time obstacle avoidance. The system operates on a ROS multi-master framework, using MAVLink and Pixhawk for command execution, combined with ultrasonic ranging and IMU/GPS fusion for state estimation. Both Gazebo simulations and indoor real-vehicle tests (one leader and two followers) were conducted. Results show that the average position error is about 0.069 m in simulation and 0.098 m in real-vehicle experiments, with increased formation error and maximum instantaneous error. Analysis indicates that the primary causes of performance degradation include sensor noise, communication latency, and model–reality mismatch. Several improvements are recommended, such as enhanced sensor fusion, rigorous calibration, and delay-robust or adaptive control. These measures are expected to reduce the gap between simulation and real-world performance.

Keywords: Multi-Robot Formation Control; Unmanned Ground Vehicle; Leader–Follower; Virtual Structure

1. Introduction

Driven by advances in AI, sensor technology, wireless communications, and embedded systems, unmanned systems are being widely deployed across various sectors, including the military, transportation, logistics, and agriculture [1]. While the intelligence of a single unmanned vehicle has improved significantly, its efficiency and flexibility remain limited for tasks such as large-area inspection, material transport, or emergency response in complex environments. In contrast, multi-vehicle cooperative formation systems—through inter-vehicle information sharing and coordinated control—can significantly enhance task efficiency, system robustness, and environmental adaptability [2].

In recent years, formation control in multi-agent systems (MAS) has attracted intensive attention. The core problem is to maintain prescribed relative positions among multiple autonomous vehicles while achieving path tracking and dynamic obstacle avoidance. Traditional centralized control strategies suffer from heavy computational load and poor scalability, whereas distributed control better matches the practical requirements of multi-vehicle systems [3].

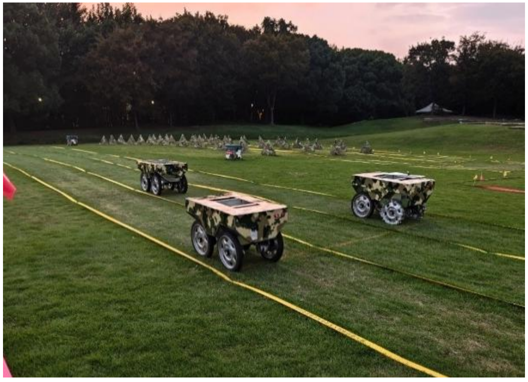

Figure 1. Physical Setup of the Vehicle Formation

2. Theories for Multi-Vehicle Cooperative Control

The essence of formation control is to maintain relative positions and orientations among vehicles via information exchange. Common models include consensus control, the Leader–Follower paradigm, and the Virtual Structure method [4].

2.1 Consensus Control

Consensus control forms the theoretical foundation of multi-agent systems. Its core idea is that, through local information exchange, all agents' states converge to agreement. In formation control, consensus maintains desired relative relationships between vehicles and is thus key to stable formations.

For a system of N unmanned vehicles, let the state and control input of vehicle i be x_{i} and u_i , respectively. A typical consensus control law is [5]:

$$u_i = \sum_{j \in \mathrm{N}_i} a_{ij}(x_j - x_i) \tag{1}$$

where N_i is the neighbor set of vehicle i and a_{ij} are communication weights. If the communication topology is a connected directed graph (or contains a directed spanning tree), then

$$\lim_{t \to \infty} \| x_i - x_j \| = 0, \forall i, j \tag{2}$$

i.e., all agents asymptotically reach consensus.

2.2 Leader–Follower Model

The Leader–Follower approach is one of the most commonly used formation methods. One or more leaders are designated, and followers regulate their motion with respect to the leader's state and a desired offset. Owing to its simplicity and scalability, it is widely used in multi-vehicle cooperation [6]. A standard formulation is:

$$\dot{x}_i = Ax_i + Bu_i, u_i = -k(x_i - x_{\text{leader}} - d_i) \tag{3}$$

where x_i is the state of follower i, x_{leader} is the leader's state, d_i is the desired relative offset, and k is the control gain that tunes stability and response speed. By feedback on the tracking error, each follower maintains the desired relative distance and bearing to the leader, ensuring a stable formation.

2.3 Virtual Structure Method

The virtual structure method treats the entire formation as a virtual rigid body. Each vehicle corresponds to a fixed point on the rigid body and moves with the body's overall motion. By centrally controlling the virtual rigid body, all vehicles move cooperatively while preserving formation geometry—especially suitable for regular formations (rectangular, V-shaped, circular) [7]. The core expression is:

$$p_i(t) = p_c(t) + R(t)p_i^0 \tag{4}$$

where:

$p_i(t)$: global position of vehicle $i$at time t;

$p_c(t)$: position of the formation (rigid-body) center;

$R(t)$: rotation matrix describing the rigid body's orientation;

p_i^0 : fixed relative position of vehicle i in the rigid-body frame (initial offset).

3. Model Scheme Design

3.1 Formation-Control Model

The core objective in a multi-UGV formation is to make all vehicles follow a leader (or a reference trajectory) while honoring formation constraints, keeping stable relative positions and headings. We adopt an improved Leader–Follower model that blends consensus ideas with the virtual-structure view to enable flexible formation keeping and dynamic obstacle avoidance across formations.

Leader–Follower formation model

Assume a fleet of N UGVs. Vehicle 1 is the Leader; vehicles $2, \ldots, N$ are Followers. Let the leader and follower states evolve as:

$$\mathbf{X}_L = f(\mathbf{X}_L, \mathbf{u}_L), \mathbf{X}_i = f(\mathbf{X}_i, \mathbf{u}_i), i = 2, \ldots, N \tag{5}$$

With

$$\mathbf{X}_L = [\frac{p_L}{\theta_L}], X_i = [\frac{p_i}{\theta_i}], p = [xy]. \tag{6}$$

The desired formation waypoint for follower $i$in the world frame is

$$\mathbf{p}_i^* = \mathbf{p}_L + R(\theta_L)\mathbf{d}_i \tag{7}$$

where $R(\theta_L)$ is the leader's rotation matrix and $\mathbf{d}_i$ is the desired offset of follower i $i$in the leader body frame.Define the errors

$$\mathbf{e}_{p,i} = \mathbf{p}_i^* - \mathbf{p}_i, e_{\theta,i} = wrap(\theta_L - \theta_i) \tag{8}$$

where $wrap(\cdot)$ normalizes an angle to $(-\pi, \pi]$.

(2) Leader maneuver feedforward

To precisely track a moving formation waypoint without changing the basic control structure, add a feedforward term of the leader's motion in the position channel:

$$\dot{\mathbf{p}}_i = \dot{\mathbf{p}}_L + R(\theta_L)J\mathbf{d}_i\dot{\theta}_L + k_p\mathbf{e}_{p,i}, \dot{\theta}_i = k_\theta e_{\theta,i} \tag{9}$$

with

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{10}$$

Parameters: θ_i is follower i 's heading; $\mathbf{d}_i \in R^2$ is the formation offset in the leader frame; $k_p > 0, k_\theta > 0$ are proportional gains (or their diagonal-matrix counterparts).

The feedforward term：

$$\dot{\mathbf{p}}_L + R(\theta_L)J\mathbf{d}_i\dot{\theta}_L \tag{11}$$

represents the desired waypoint's own velocity (including the orbital velocity induced by leader rotation). Adding it effectively removes steady-state errors during leader maneuvers, keeping a tight formation.

(3) Formation definitions

We parameterize the desired offsets as

$$\mathbf{d}_i = \begin{bmatrix} r_i cos & \alpha_i \\ r_i sin & \alpha_i \end{bmatrix} \tag{12}$$

where r_i is the radial distance and α_i the angular offset of follower $i$relative to the leader body frame. By adjusting r_i, α_i online, the formation can switch/layout rectangular, V-shaped, or circular patterns in real time.

3.2 Control-Logic Design

Using the leader's attitude as the reference, define formation offsets in the leader body frame and rotate them into the geographic N/E frame:

$$\begin{bmatrix} d_N \\ d_E \end{bmatrix} = \begin{bmatrix} cos\psi_L & -sin\psi_L \\ sin\psi_L & cos\psi_L \end{bmatrix} \begin{bmatrix} x_b \\ y_b \end{bmatrix} \tag{13}$$

Then the desired N/E coordinates of follower $i$are

$$\begin{bmatrix} N_i^* \\ E_i^* \end{bmatrix} = \begin{bmatrix} N_L \\ E_L \end{bmatrix} + \begin{bmatrix} d_N \\ d_E \end{bmatrix} \tag{14}$$

A simple saturated tracking law is

$$v_{NE} = sat_{Vm}\left(Kp \begin{bmatrix} N_i^* & -N_i \\ E_i^* & -E_i \end{bmatrix} \right),$$
$$\psi_i = sat_{\psi max}(K_\psi(\psi_L - \psi_i)). \tag{15}$$

In implementation, we send the *MAVLinkSET_POSITION_TARGET_LOCAL_NED* command with only v_N, v_E, v_D yaw rate (set v_D=0 for planar formations).

During formation switching, apply a first-order low-pass or S-curve limiter to the offsets to suppress transients and improve smoothness. Compared with "distance–bearing PID", this design keeps attitude and actuator loops in the inner flight controller, improving robustness and reducing noise sensitivity.

3.3 Obstacle-Avoidance Strategy

Local obstacle avoidance relies on ultrasonic ranging. When the front distance d_{obs} falls below the safety threshold d_{safe}, trigger:

$$v_i' = v_i \cdot \frac{d_{\text{obs}} - d_{min}}{d_{\text{safe}} - d_{min}}, \left(d_{min} < d_{\text{obs}} < d_{\text{safe}}\right) \tag{16}$$

$$\omega_i' = \omega_i + k_{\text{avoid}}(d_{\text{left}} - d_{\text{right}}) \tag{17}$$

where $d_{\text{left}}, d_{\text{right}}$ are the lateral ranges and k_{avoid} is a tuning gain. Simultaneous speed reduction and steering compensation yields smooth avoidance without breaking overall formation stability.

3.4 Algorithm-Flow Design

As depicted in Figure 2，The software runs on a ROS-distributed framework and follows a sense–decide–act–feedback loop for real-time, stable multi-UGV cooperation:

Initialization. Start ROS Master and multi-master sync; register topics and set up links (VPN/5G as needed).

State acquisition. The leader plans a path from task goals, publishes its pose/velocity/attitude periodically; followers subscribe to /leader_pose for synchronization.

Error computation. Each follower computes distance error e_d and heading error e_θ from leader data and onboard sensing.

PID output. Apply linear-velocity and

angular-velocity controllers to produce v_i and ω_i; publish to /cmd_vel and forward to Pixhawk for execution.

Obstacle check & correction. If ultrasonic sensing indicates risk, invoke the local avoidance branch to adjust speed/yaw.

Feedback & estimation. IMU/GPS/ultrasonic streams feed the state-estimation node for continuous pose updates, closing the loop at ~20 Hz.

Supervision & resilience. systemd manages key services (e.g., mavlink-router.service on boot, formation.service for control scripts with auto-restart on failure). Global clock sync via NTP/chrony aligns logs and sensor timestamps for reproducible analysis.

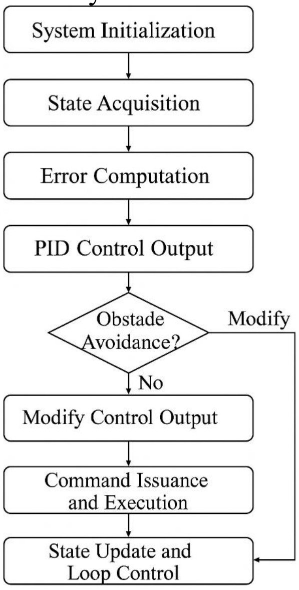

Figure 2. Flowchart of the Algorithm

4. Experiments and Result Analysis

4.1 Simulation Environment

Simulations are carried out on Gazebo 11 + ROS Noetic. OS: Ubuntu 20.04; visualization via Gazebo 11 and RViz; development in Python/C++. The host uses an Intel i7-9700 CPU with 16 GB RAM. A ROS multi-master setup (multimaster_fkie) enables real-time synchronization among vehicles. The vehicle model adopts a four-wheel differential-drive platform with virtual odometry, IMU, and laser interfaces. Three UGVs are instantiated (1 leader + 2 followers), each equipped with virtual ultrasonic sensors for obstacle sensing and avoidance. The map is 20 m × 10 m with static obstacles and randomized paths.

The experimental procedure begins with system initialization, where the ROS multi-master environment is started, and the Leader and Follower nodes are loaded. Subsequently, formation initialization is completed by selecting a desired formation, such as a “line” or “V-shape,” through the /formation_select topic. During the path planning phase, the Leader navigates along a predefined trajectory, for instance, a figure-eight or a circular path. In the control execution phase, Follower nodes continuously receive the Leader’s pose data and implement PID control and obstacle avoidance algorithms to maintain dynamic following and path tracking. Throughout the experiment, critical data, including vehicle position, velocity, tracking error, and communication latency, are logged via the /system_log topic. Finally, the formation trajectories, error variations, and communication performance are visualized and statistically analyzed using RViz and Matplotlib [8].

Obstacle scenes:

Scene A (single obstacle): 1 cubic obstacle, $0.3 \times 0.3 \times 0.3$ m.

Scene B (multiple obstacles): two spherical obstacles, diameter 0.1 m.

Scene C (narrow corridor): two walls, tightest width 1 m.

4.2 Simulation Results and Analysis

This study conducts simulation validation using a longitudinal “1”-shape (single-file) formation, in which Vehicle 1 serves as the leader (Leader), and Vehicles 2 and 3 act as followers (Follower1 and Follower2). Prior to the experiment, the three vehicles are placed at preset initial poses. By launching the navigation node of the Leader and the formation-control nodes of Follower1 and Follower2, the system enters normal operation. In the Gazebo simulation environment, vehicle states are monitored in real time to obtain the relative pose (position and orientation) between the Leader and the two Followers. Follower1 and Follower2 obtain output signals via TF nodes—including linear and angular velocity commands—and execute motion adjustments according to these control inputs. Under the action of the control algorithm, the two Followers then depart from their initial positions and continuously follow the Leader along the prescribed circular trajectory in a repetitive manner, thereby realizing a closed-loop formation-following process. During the experiment, the system continuously records vehicle pose data, control outputs, and error signals through ROS topics, providing data support for subsequent performance analysis and evaluation of control effectiveness.

(1)Formation Path-Tracking Results

Figure 3 illustrates the formation motion trajectories of the Leader and the two Followers in the “1”-shape path scenario. From the trajectory curves, it can be observed that each Follower accurately tracks the Leader’s motion path. No pronounced oscillations occur during the dynamic response of the formation, indicating favorable stability and consistency.

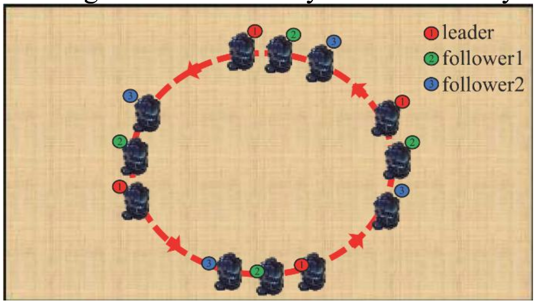

Figure3. Schematic of Multi-Vehicle Formation Trajectories in the Gazebo Simulation

Table 1. Simulation Performance

Vehicle	Mean pos. error (m)	Max Error (m)	Formation error (m)
Follower1	0.0683	0.1660	0.0526
Follower2	0.0705	0.1669	0.0526

The average position error is under 0.1 m, and the max instantaneous error is below 0.2 m, meeting the design target. The mean formation-keeping error is $E_f = 0.053\,m$.

4.3 Experimental Procedure

Power on vehicles → auto-attach to 5G → start ROS core and formation nodes, complete topic registration and link initialization. After initialization, the leader loads a preset path (straight, I-shape, or V-shape); followers receive formation configuration and enter standby to establish the initial formation at start. During the run, the leader follows the preset trajectory; followers receive leader pose via ROS topics and execute PID or the improved Leader–Follower controller to achieve position/heading tracking and dynamic formation motion. For avoidance tests, static obstacles (cylinders/boxes/walls) are placed randomly; followers engage the avoidance module to evaluate response and post-avoidance rejoining. A remote console logs pose, attitude, commands, and latency for later visualization and statistical analysis of tracking accuracy, formation keeping, and avoidance stability.

4.4 Results and Analysis

(1) Formation Motion Trajectories

Table2. Real-Vehicle Formation Performance

Vehicle	Mean error (m)	Max error (m)	Formation error (m)
Follower1	0.0950	0.2150	0.0814
Follower2	0.1018	0.2422	0.0834

Figure1 presents the motion trajectories of the leader and follower recorded in the real-vehicle experiments. The results demonstrate that the multi-vehicle system achieved satisfactory synchronous motion and maintained a stable formation in the indoor environment. The corresponding test results are summarized in Table 2.

Mean position errors are 0.095 m and 0.1018 m (overall mean 0.0984 m). Mean formation error ≈ 0.0824 m with ~0.002 m inter-vehicle difference, showing good consistency. Max instantaneous errors reach 0.215–0.242 m (mean 0.2287 m), indicating ~0.23 m peaks in worst moments that can be further reduced.

4.5 Comparison: Simulation vs. Real Vehicle

Table 3. Comparison of Simulation and Real-Vehicle Results

Metric	Simulation mean	Real-vehicle mean
Mean tracking error	0.0694 m	0.0984 m
Max error	0.1664 m	0.2287 m
Formation error	0.0526 m	0.0824 m

Discussion. Table 3 shows that real-vehicle performance degrades noticeably compared to simulation. The mean tracking error increases from 0.0694 m to 0.0984 m (Δ = +0.0290 m, +41.8%), the maximum error increases by 0.0623 m (+37.4%), and the formation error increases from 0.0526 m to 0.0824 m (Δ = +0.0298 m, +56.7%). These results indicate that, under real-world conditions, both point tracking and formation maintenance are significantly worse than in simulation. The primary causes are likely measurement noise and bias in onboard sensors, environmental disturbances (e.g., wind), additional communication latency and packet loss, and model–reality mismatch (unmodeled aerodynamics and actuator dynamics). To reduce the simulation–reality gap, we recommend reporting statistical dispersion (standard deviation and confidence intervals), increasing the number of experimental trials, improving sensor fusion (e.g., EKF/UKF), performing rigorous sensor/actuator calibration, and incorporating delay-robust or adaptive control schemes. Implementing these measures is expected to reduce both mean and peak errors in real tests.

5. Conclusions

This study designs and validates a practical cooperative formation-control system for multi-UGV teams. Theoretically, the approach integrates consensus control, Leader–Follower strategies, and the virtual-structure method, improving formation flexibility and stability. In terms of engineering implementation, the system is built on a ROS multi-master architecture with Pixhawk, employing saturated tracking laws and feedforward compensation to reduce steady-state errors, while ultrasonic sensing achieves local obstacle avoidance. Simulation and real-vehicle experiments confirm that the system maintains dynamic following and formation reconfiguration in complex scenes, although real-world performance shows increased average and peak errors compared with simulation. To enhance robustness, future work should focus on: (1) improving sensor-fusion and time-synchronization mechanisms (e.g., EKF/UKF); (2) enhancing communication reliability and delay compensation; (3) collecting more experimental data for statistical evaluation and controller tuning; and (4) validating adaptive or learning-based controllers in more complex environments. Overall, this research provides a feasible control framework and engineering pathway for multi-UGV cooperation in large-area inspection, logistics, and rescue applications.

References

[1] Zhang Y, Sun Q, Ren W. Survey of distributed formation control[J]. Unmanned Systems, 2024, 12(1): 1-35.

[2] Queralta J P, Taipalmaa J, Pullinen B, et al. Collaborative multi-robot search and rescue: A survey[J]. IEEE Access, 2020, 8: 191617-191643.

[3] Cao Y, Yu W, Ren W, Chen G. An overview of recent progress in distributed multi-agent coordination[J]. IEEE Transactions on Industrial Informatics, 2013, 9(1): 427-438.

[4] Olfati-Saber R, Fax J A, Murray R M. Consensus and cooperation in networked multi-agent systems[J]. Proceedings of the IEEE, 2007, 95(1): 215-233.

[5] Cao M, Morse A S, Anderson B D O. Coordination control of agents in a plane[J]. IEEE Transactions on Automatic Control, 2008, 53(4): 899-903.

[6] Fax J A, Murray R M. Information flow and cooperative control of vehicle formations[J]. IEEE Transactions on Automatic Control, 2004, 49(9): 1465-1476.

[7] Ren W, Beard R W. Formation feedback control for multiple spacecraft via virtual structures[J]. IEE Proceedings–Control Theory and Applications, 2004, 151(3): 357-368.

[8] Hernández S, Herrero F. Multimaster ROS Systems—Technical Report[R]. 2016: 1-14.