

# Style for IFAC Conferences & Symposia: Use Title Case for Paper Title

Alessandra Elisa Sindi Morando <sup>\*,\*\*</sup> Jossué Cariño <sup>\*\*</sup>  
Pedro Castillo <sup>\*\*</sup> Roberto Sacile <sup>\*</sup> Enrico Zero <sup>\*</sup>

<sup>\*</sup> University of Genoa, Genoa, Italy  
*alessandra.elisa.sindi.morando@edu.unige.it, roberto.sacile@unige.it,*  
*enrico.zero@dibris.unige.it*  
<sup>\*\*</sup> Université de technologie de Compiègne, CNRS, Heudiasyc,  
Compiègne, France  
*alessandra-elisa-sindi.morando@utc.fr, {jossue.escobar,*  
*pedro.castillo}@hds.utc.fr*

---

**Abstract:** ...

**Keywords:** ...

---

## 1. INTRODUCTION

Multi-agent systems (MASs) are nowadays used in a variety of applications from as they can perform complex tasks, which would otherwise impossible for a single robot. This is even more the case when considering heterogeneous fleet where the strengths of one type of robot can compensate for the limitations of another. For example, to monitor a large area, multiple ground agents can move from one point to another being supervised from above by a drone with a wide field of view. To deepen the subject and understand both the potential tasks and challenges of MASs, the reader can refer to the work of Maldonado et al. (2024).

One of the fundamental cooperative tasks when working with multi-agents is formation control. This topic has been studied for years. For example, already Lee and Chong (2009) already proposed a decentralized control algorithm for a team of two-wheeled robots to achieve achieve a geometric pattern. However, formation control remains an hot challenging topic to this day. Tran et al. (2021) experimentally validated a robust distributed control based on negative imaginary systems consensus theory, using both ground robots and an air-ground fleet. In the work of Güler and İsa E. Yıldırım (2023), each agent computes its control action in a leader-follower manner using local extended Kalman filter's estimates to achieve a desired shape together with other wheeled robots and a drone. An optimal distributed formation control algorithm for double-integrator multi-agents has been presented and validated through simulations by Huang et al. (2023). Aditya and Werner (2023) defined the formation problem for a group of double-integrator agents as a discrete-time game whose solution is given by a state-dependent Riccati equation. A robust distributed consensus controller is presented by Restrepo et al. (2023) to address the rendezvous problem and applied in simulation to a swarm of drones. In the recent years, reinforcement learning has used in many areas including formation control. Wang et al. (2020) addressed multi-particle formation control by combining

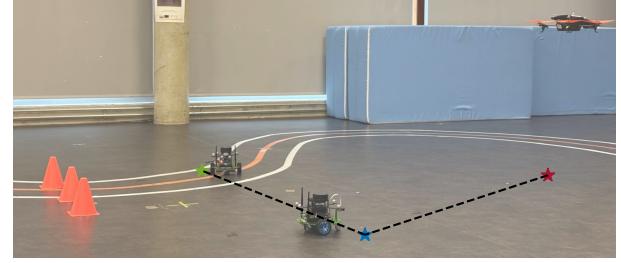


Fig. 1. This paper proposes a formation control for a heterogeneous fleet to attain a desired shape despite obstacles.

graph attention networks and multiple long short-term memories to achieve the desired shape and avoid collisions respectively. A position and an orientation robust controllers based on reinforcement learning are proposed and validated through simulations by Zhong et al. (2025) for the formation of a term of quadrotors.

In this context is placed this work which proposes a novel formation control scheme for a team consisting of a quadcopter and two unicycles, as show in Fig. 1. When addressing the multi-agent formation, there are two main challenges to solve: achieving the desired shape and avoiding collisions. On one hand, for the first challenge, a distributed controller combining Feedback Linearization (FL) and an optimal linear controller based on Linear Matrix Inequalities (LMIs) is presented. LMIs are a powerful mathematical tool used in many applications among which also formation control. In the work of Trejo et al. (2023), an LMI is solved to design the controller and observer gains, guaranteeing stable formation flight for a group of quadcopters. Deshpande et al. (2011) propose a formation control with artificial fixed delays for multiple double-integrators and verify the asymptotic stability of the scheme by checking the solvability of an LMI. Semsar-Kazerooni and Khorasani (2009) present an LMI formulation of the LQR problem to guarantee stable state consensus with an optimal control effort in a multi-agent system.

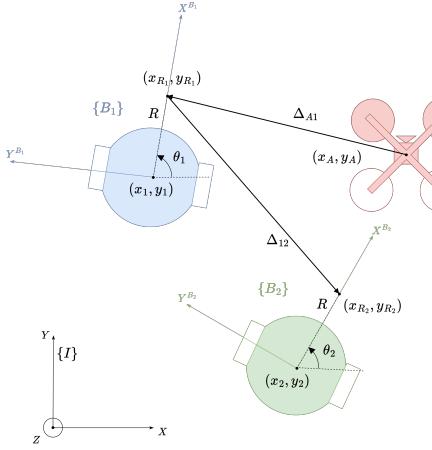


Fig. 2. Air-ground heterogeneous fleet.

On the other end, to avoid collision avoidance, the chosen solution is Artificial Potential Field (APF). In analogy with a particle moving in an electrostatic or gravitational field, virtual repulsive force fields are simulated to ward off the robot from an obstacle. This approach has been widely used in formation control of ground vehicles, see for example the work of Yongshen et al. (2018). However, in more recent works, e.g. Han et al. (2024) and Piet et al. (2025), APFs are employed to navigate around obstacles and prevent collisions while attaining the desired shape.

This paper represents the continuation of the previous work by Morando et al. (2025), where a control architecture combining Feedback Linearization (FL), a robust LMI-based controller and APFs was proposed and validated through simulations for the formation of three unicycles. The successive developments sought to manage the challenge of a heterogeneous team (replacing a ground vehicle with a quadcopter) by adapting the methodology and then experimentally validating the new solution. The contributions of this work are: a) a controller involving FL, a LMI-based controller and APFs is presented for the stable formation of an air-ground fleet; b) the proposed controller has been validated through MATLAB Simulink simulations and a comprehensive set of experiments in an indoor arena, including both static and dynamic obstacle scenarios. The remainder paper is structured as follows. In Section 2, the mathematical model used to describe the team is stated. The proposed formation controller is detailed in Section 3. The simulations and the experiments results are reported and analyzed in Section 4 and in Section 5 respectively. Finally, conclusions are drawn in Section 6.

## 2. HETEROGENEOUS FLEET MODELING

In this section, the dynamics of the agents composing the team (i.e., of the two unicycles and the quadrotor, as shown in Fig. 2.) are described and adapted to then apply the LMI approach. The well-known kinematic model of a two-wheeled ground robot  $G_i$  is

$$\begin{cases} \dot{x}_i = v_i \cos \theta_i \\ \dot{y}_i = v_i \sin \theta_i \\ \dot{\theta}_i = \omega_i \end{cases} \quad (1)$$

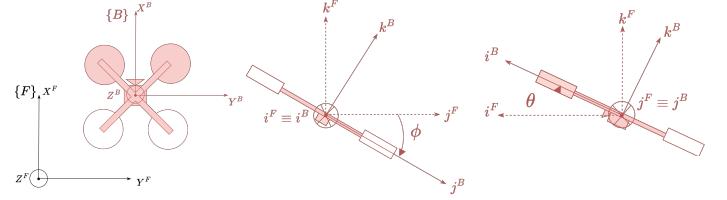


Fig. 3. Quadcopter's roll  $\phi$  and pitch  $\theta$  angles.

where  $(x_i, y_i, \theta_i)$  is the pose of the agent and  $(v_i, \omega_i)$  are the linear and angular velocities respectively. The equations (1) describing the motion of the center of rotation are non-linear. But still, if a point located at distance  $R > 0$  from  $(x_i, y_i)$  denote as  $(x_{R_i}, y_{R_i})$ , is considered and new virtual inputs are defined  $(u_{i,x}, u_{i,y})$ , the dynamics become linear:

$$\begin{cases} \dot{x}_{R_i} = u_{i,x} \\ \dot{y}_{R_i} = u_{i,y} \end{cases} \quad (2)$$

Moving on to the drone, as in the formation problem deals with the  $(X, Y)$ -dynamics, some simplifying hypothesis have been made. First, let us assume that there is an inner attitude controller. Hence, the focus is only on the translational dynamics which are:

$$\begin{cases} m\ddot{x}_A^F = -T \sin \theta \\ m\ddot{y}_A^F = T \cos \theta \sin \phi \\ m\ddot{z}_A^F = T \cos \theta \cos \phi - mg \end{cases} \quad (3)$$

Let analyze the equations with the help of Fig. 3. The state  $(x_A^F, y_A^F, z_A^F)$  represents the position of the drone w.r.t. the inertial frame  $\{F\}$ . The frame  $\{F\}$  has been used instead of  $\{I\}$  to be aligned with the framework employed to implement the controller for the drone, see Section 5. The angles  $\phi$  and  $\theta$  are the roll and pitch Euler angles of the body frame  $\{B\}$  (fixed to the drone) relative to the inertial frame  $\{F\}$ . Ignoring the small body forces, the only forces acting on the drone are the thrust produced by the motors  $T$  along the  $k^B$  direction and the gravity  $mg$  along the  $-k^F$  direction. Let suppose that the  $Z$ -controller

$$T = (r_1 + mg)/(\cos \phi \cos \theta) \quad (4)$$

is such that the drone's mass is compensated in a short time by  $r_1$ , i.e.  $r_1 \rightarrow 0$ . This implies that the  $(x, y)$ -dynamics will not include the mass of the drone, as the  $Z$ -controller already compensated it. Finally, let us make the hypothesis that the roll and pitch angles are small, i.e.  $\phi, \theta \approx 0$ . Then, the translational dynamics simplify to

$$\begin{cases} \ddot{x}_A^F = -g\theta \\ \ddot{y}_A^F = g\phi \end{cases} \quad (5)$$

At this point, all the agents can be described by the linear models (1) and (5). To define the formation problem, let us introduce the quantities

$$\Delta_{A1} = [\Delta_{A1,x} \ \Delta_{A1,y}]^T \quad (6a)$$

$$\Delta_{A1,x} = x_{R_1} - x_A = x_{R_1} - y_A^F \quad (6b)$$

$$\Delta_{A1,y} = y_{R_1} - y_A = y_{R_1} - x_A^F \quad (6c)$$

with  $(x_A, y_A)$  the position of the drone w.r.t the inertial frame  $\{I\}$ ,

$$\Delta_{12} = [\Delta_{12,x} \ \Delta_{12,y}]^T \quad (7a)$$

$$\Delta_{12,x} = x_{R_2} - x_{R_1}, \ \Delta_{12,y} = y_{R_2} - y_{R_1} \quad (7b)$$

the desired inter-distances  $\Delta_{A1}^d$  and  $\Delta_{12}^d$ , and the desired position of the drone  $(x_A^d, y_A^d)$  w.r.t the inertial frame

$\{I\}$  (constant over time). The overall dynamics of the formation's errors are as follows

$$\begin{cases} \ddot{e}_{A,x} = -g\theta \\ \ddot{e}_{A,y} = g\phi \\ \dot{e}_{\Delta_{A1,x}} = u_{1,x} - \dot{e}_{A,y} \\ \dot{e}_{\Delta_{A1,y}} = u_{1,y} - \dot{e}_{A,x} \\ \dot{e}_{\Delta_{12,x}} = u_{2,x} - u_{1,x} \\ \dot{e}_{\Delta_{12,y}} = u_{2,y} - u_{1,y} \end{cases} \quad (8)$$

with

$$e_{A,x} = x_A^F - y_A^d, e_{A,y} = y_A^F - x_A^d \quad (9)$$

$$e_{\Delta_{A1,x}} = \Delta_{A1,x} - \Delta_{A1,x}^d, e_{\Delta_{A1,y}} = \Delta_{A1,y} - \Delta_{A1,y}^d \quad (10)$$

$$e_{\Delta_{12,x}} = \Delta_{12,x} - \Delta_{12,x}^d, e_{\Delta_{12,y}} = \Delta_{12,y} - \Delta_{12,y}^d \quad (11)$$

Let us define the state and control vectors as follows

$$\mathbf{e}^T = [e_{A,x} \ \dot{e}_{A,x} \ e_{A,y} \ \dot{e}_{A,y} \ e_{\Delta_{A1,x}} \ \dot{e}_{\Delta_{A1,x}} \ e_{\Delta_{12,x}} \ \dot{e}_{\Delta_{12,y}}] \quad (12a)$$

$$\mathbf{u}_A = \begin{bmatrix} \phi \\ \theta \end{bmatrix}, \mathbf{u}_1 = \begin{bmatrix} u_{1,x} \\ u_{1,y} \end{bmatrix}, \mathbf{u}_2 = \begin{bmatrix} u_{2,x} \\ u_{2,y} \end{bmatrix}, \mathbf{u} = \begin{bmatrix} \mathbf{u}_A \\ \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \quad (12b)$$

Hence, the heterogeneous fleet's dynamics can be written in a matrix form:

$$\dot{\mathbf{e}} = A\mathbf{e} + B\mathbf{u} \quad (13)$$

By discretizing with sampling time  $\Delta T$  using the Euler method:

$$\mathbf{e}(k+1) = A_d\mathbf{e}(k) + B_d\mathbf{u}(k) \quad (14)$$

with  $A_d = I + \Delta T A$  and  $B_d = \Delta T B$ . To model real-world disturbances, it is assumed that an additive zero-mean disturbance  $\mathbf{d}(k)$  is acting on the input channel, i.e.

$$\mathbf{e}(k+1) = A_d\mathbf{e}(k) + B_d\mathbf{u}(k) + B_d\mathbf{d}(k) \quad (15)$$

Let us make the following hypothesis: a) the drone knows its position; b) the robot  $G_1$  knows its position w.r.t the quadcopter; c) the robot  $G_2$  knows its position w.r.t the other agent  $G_1$ ; d) all unicycles are aware of their distance from any obstacles on the ground; e) a hierarchy is defined between the ground robots:  $G_1$  has the priority over  $G_2$ .

### 3. FORMATION CONTROLLER

This section aims to detail the proposed formation controller. The two crucial issues, formation maintenance and collision avoidance, are addressed respectively in the next two subsections.

#### 3.1 Formation Maintenance: an LMI Approach

The first goal is to ensure the robots achieve the desired shape while using only local measurements and counter-acting the worst-case disturbance. Based on the previous hypotheses, the measurements of each agent are as follows:

$$\mathbf{z}_A = \begin{bmatrix} e_{A,x} \\ e_{A,y} \\ \dot{e}_{A,x} \\ \dot{e}_{A,y} \end{bmatrix} = C_A \mathbf{e} \quad (16a)$$

$$\mathbf{z}_1 = \begin{bmatrix} e_{\Delta_{A1,x}} \\ e_{\Delta_{A1,y}} \end{bmatrix} = C_1 \mathbf{e}, \mathbf{z}_2 = \begin{bmatrix} e_{\Delta_{12,x}} \\ e_{\Delta_{12,y}} \end{bmatrix} = C_2 \mathbf{e} \quad (16b)$$

with  $C_A$ ,  $C_1$ , and  $C_2$  the measurement matrices of the three. The idea is to formulate the robust formation control problem as a dynamic game between two players: one is the control action and the other opponent is the "nature". In the proposed solution, at each time step  $k$

the control action to be applied at the next step  $\mathbf{u}(k)$  is obtained by solving the following min-max problem defined over a one-step horizon.

$$\begin{aligned} \min_{\mathbf{u}(k)} \max_{\mathbf{e}(k) \neq 0} & \frac{J(k)}{\|\mathbf{e}(k)\|_2^2} \\ \text{s.t. } & \mathbf{z}_i(k) = C_i \mathbf{e}(k) \\ & \mathbf{u}_i(k) = \mu_i(\mathbf{z}_i(k)) \quad i \in \{A, 1, 2\} \end{aligned} \quad (17)$$

with

$$J(k) = \mathbf{u}(k)^T Q_u \mathbf{u}(k) + \mathbf{e}(k+1)^T Q_e \mathbf{e}(k+1) + \mathbf{e}(k)^T Q_e \mathbf{e}(k) \quad (18)$$

The cost function can be rewritten by substituting the evolution model equation (15).

$$J(k) = \begin{bmatrix} \mathbf{e}(k) \\ \mathbf{u}(k) \end{bmatrix}^T \begin{bmatrix} Q_e + A_d^T Q_e A_d & A_d^T Q_e B_d \\ B_d^T Q_e A_d & Q_u + B_d^T Q_e B_d \end{bmatrix} \begin{bmatrix} \mathbf{e}(k) \\ \mathbf{u}(k) \end{bmatrix} \quad (19)$$

Hence, the min-max problem to be solved is

$$\begin{aligned} \min_{\mathbf{u}(k)} \max_{\mathbf{e}(k) \neq 0} & \frac{\begin{bmatrix} \mathbf{e}(k) \\ \mathbf{u}(k) \end{bmatrix}^T \begin{bmatrix} Q_{ee} & Q_{eu} \\ Q_{ue} & Q_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{e}(k) \\ \mathbf{u}(k) \end{bmatrix}}{\|\mathbf{e}(k)\|_2^2} \\ \text{s.t. } & \mathbf{z}_i(k) = C_i \mathbf{e}(k) \\ & \mathbf{u}_i(k) = \mu_i(\mathbf{z}_i(k)) \quad i \in \{A, 1, 2\} \end{aligned} \quad (20)$$

with

$$Q_{ee} = Q_e + A_d^T Q_e A_d \quad (21)$$

$$Q_{eu} = A_d^T Q_e B_d, Q_{ue} = Q_{eu}^T \quad (22)$$

$$Q_{uu} = Q_u + B_d^T Q_e B_d \quad (23)$$

Note that the problem (20) has the same form as the one considered by Gattami et al. (2012) in their paper where they prove that the linear policy is optimal, i.e.

$$\mathbf{u}_i^*(k) = K_i^* \mathbf{z}_i(k) \quad i \in \{A, 1, 2\} \quad (24)$$

where the matrices  $K_i^*$  can be obtained by solving the following LMI:

$$\begin{aligned} \min_{\gamma} & \gamma \\ \text{s.t. } & K = \text{diag}(K_A, K_1, K_2) \\ & \begin{pmatrix} Q_{ee} - \gamma I + Q_{eu} K C + C^T K^T Q_{eu}^T & C^T K^T \\ K C & -Q_{uu}^{-1} \end{pmatrix} \leq 0 \end{aligned} \quad (25)$$

where  $C$  is the matrix obtained by stacking the single  $C_i$ . To conclude, please note that since the discrete-time system is time-invariant, the min-max problem (20) to be solved will be the same at each iteration. Hence, it is sufficient to solve the LMI (25) offline one time to obtain the optimal gain matrices  $K_i^*$ , and then multiply at each instant  $k$  by the local measurement  $\mathbf{z}_i(k)$  by  $K_i^*$  to obtain  $\mathbf{u}_i^*(k)$ .

#### 3.2 Collision and Obstacle Avoidance: APFs

Collisions can still occur when the agents attain the desired shape. Moreover, obstacles can be present in the robots' way. It is therefore crucial to propose a solution to avoid crashes for the experimental validation. This is where APFs come to help. For simplicity, let us consider at first a scenario where the ground is clean and collisions may occur between  $G_1$  and  $G_2$ . On one hand, since  $G_1$  has the highest priority, there is no need to correct the control action  $\mathbf{u}_1^*(k)$ . On the other hand, the robot  $G_2$  has to correct its way to avoid the other ground vehicle. A repulsive force can be used to attain this, with a direction opposite to the vector going from  $G_2$  to  $G_1$  and a magnitude as follows:

$$|F_{12}| = \begin{cases} \left( k_{Rep}/d_{Rep}^2 \right) \left[ (1/d_{12}) - (1/d_{Rep}) \right] & d_{12} < d_{Rep} \\ 0 & d_{12} \geq d_{Rep} \end{cases} \quad (26)$$

with  $d_{12} = \sqrt{\Delta_{12,x}^2 + \Delta_{12,y}^2}$  and  $d_{Rep}$  the distance after which the robot is not influenced by the obstacle. The final control inputs for  $G_2$  are:

$$\begin{aligned} u_{2,x}(k) &= u_{2,x}^*(k) + \frac{1}{2} \frac{F_{12,x}(k)}{M} \Delta T \\ u_{2,y}(k) &= u_{2,y}^*(k) + \frac{1}{2} \frac{F_{12,y}(k)}{M} \Delta T \end{aligned} \quad (27)$$

with  $M$  the two-wheeled robots mass. The same approach can be extended to a scenario with both static and dynamic obstacles. Suppose that there are  $N$  (static or dynamic) obstacles, then the final control inputs for both unicycles are:

$$\begin{aligned} u_{1,x}(k) &= u_{1,x}^*(k) + \frac{1}{2} \frac{\sum_{j=1}^N F_{j1,x}(k)}{M} \Delta T \\ u_{1,y}(k) &= u_{1,y}^*(k) + \frac{1}{2} \frac{\sum_{j=1}^N F_{j1,y}(k)}{M} \Delta T \\ u_{2,x}(k) &= u_{2,x}^*(k) + \frac{1}{2} \frac{F_{12,x}(k) + \sum_{j=1}^N F_{j2,x}(k)}{M} \Delta T \\ u_{2,y}(k) &= u_{2,y}^*(k) + \frac{1}{2} \frac{F_{12,y}(k) + \sum_{j=1}^N F_{j2,y}(k)}{M} \Delta T \end{aligned} \quad (28)$$

#### 4. SIMULATIONS RESULTS

The first validation of the proposed formation controller has been done via simulations in MATLAB R2022a Simulink. The tests have been carried out on a laptop with OS Ubuntu 20.04.6 LTS, equipped with 32 GB of RAM, an SSD NVMe, and a processor Intel Core i7-12800H. To choose the parameters, the target robots' characteristics have been taken into account from the beginning in the tuning phase. Two Waveshare JetBot Pro ROS AI and a Parrot AR.Drone 2.0 compose the experimental equipment specifically, see Section 5. First, both the linear and the angular velocities of the ground vehicles have been limited:  $|v_i| \leq 0.5 \text{ m s}^{-1}$  and  $|\omega_i| \leq 0.5 \text{ rad s}^{-1}$ . Concerning the distance of the point from the center of rotation, it has been chosen  $R = 0.18 \text{ m}$ . The continuous-time models have been discretized with  $\Delta T = 0.02 \text{ s}$ . The weighting matrices used in the definition of the LMI have been set to  $Q_e = 2e4 I_{8 \times 8}$  and  $Q_u = \text{blkdiag}(0.01 I_{2 \times 2}, 0.04 I_{4 \times 4})$ . The obtained  $K_i^*$  were

$$K_A^* = \begin{bmatrix} 0 & -0.0770 & 0 & -1.2440 \\ 0.0770 & 0 & 1.2440 & 0 \end{bmatrix} \quad (29)$$

$$K_1^* = \begin{bmatrix} -9.6717 & 0 \\ 0 & -9.6717 \end{bmatrix} \quad (30)$$

$$K_2^* = \begin{bmatrix} -15.0765 & 0 \\ 0 & -15.0765 \end{bmatrix} \quad (31)$$

which implies in closed loop the following eigenvalues of the closed-loop system

$$p = \text{eig}(A_d + B_d K^* C) = \begin{cases} 0.6985 & \mu = 2 \\ 0.7572 & \mu = 2 \\ 0.8066 & \mu = 2 \\ 0.9988 & \mu = 2 \end{cases} \quad (32)$$

which are all inside the unitary circle, check Fig. 4. Moving to the APFs, the mass of the Jetbot's robot is  $M = 1.4 \text{ kg}$  and it has been chosen  $k_{Rep} = 700$  and  $d_{Rep} = 5R$ . In

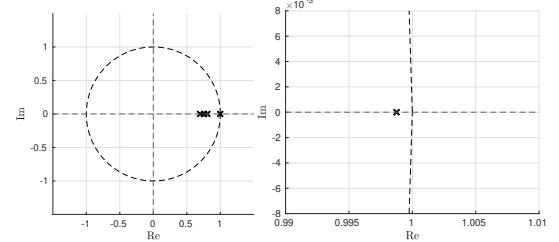


Fig. 4. Poles of the closed-loop system in the imaginary plane.

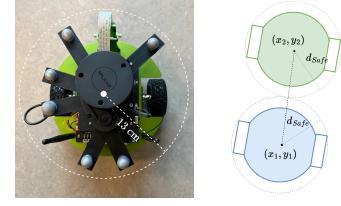


Fig. 5. The physical robot is contained entirely in a circle of radius  $d_{Safe} = 0.13 \text{ m}$ . Consequently, if the centers of rotation are at a distance less than  $2d_{Safe}$  it means that the agents collided.

the simulations, the unicycles are treated as mass points but nevertheless the objective is to verify if the physical robots will crash. For this reason, the robots are considered to have collided in the simulation if  $d_{12} \leq 2d_{Safe}$  (with  $d_{Safe} = 0.13 \text{ m}$ , see Fig. 5). For a quantitative analysis, the time required for the agents to reach the formation was studied. Let us define the key performance index  $\bar{t}$  as follows:

$$\forall t \geq \bar{t} : \begin{cases} |e_{A,x}(t)| < \varepsilon, |e_{A,y}(t)| < \varepsilon \\ |e_{\Delta_{A1,x}}(t)| < \varepsilon, |e_{\Delta_{A1,y}}(t)| < \varepsilon \\ |e_{\Delta_{12,x}}(t)| < \varepsilon, |e_{\Delta_{12,y}}(t)| < \varepsilon \end{cases} \quad (33)$$

with  $\varepsilon$  a negligible error. For the simulations, it has been set  $\varepsilon = 0.10 \text{ m}$ .

The first set of simulations considers a scenario with the ground clean and without disturbances, i.e.  $\mathbf{d}(k) = 0$ . The results are shown in Fig. 6. For each unicycle, they are plotted the 2D-trajectories of the center of rotation and of the point  $(x_{R_i}, y_{R_i})$  with a solid and a dotted line respectively. The desired inter distances drone- $G_1$  and  $G_1-G_2$  are plotted in solid black, while the actual inter distances at the end of the simulation are the dotted black lines. Finally, stars denote the desired final positions of the robots based on the definitions of  $(x_A^d, y_A^d)$ ,  $\Delta_{A1}^d$  and  $\Delta_{12}^d$ . From Fig. 6, it is straightforward to conclude that the agents make the formation avoiding collisions in all scenarios, which differ in initial conditions and desired

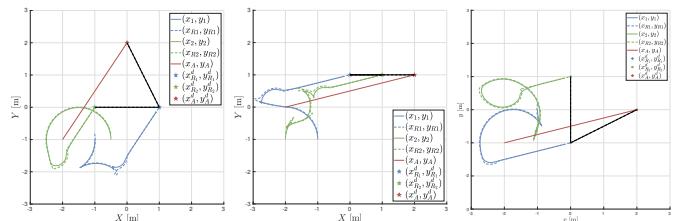
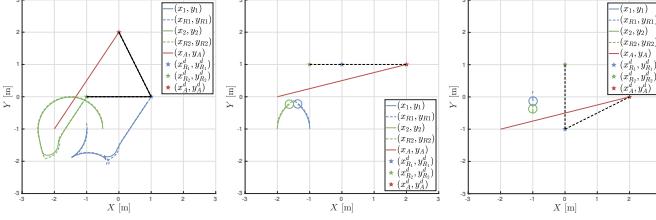
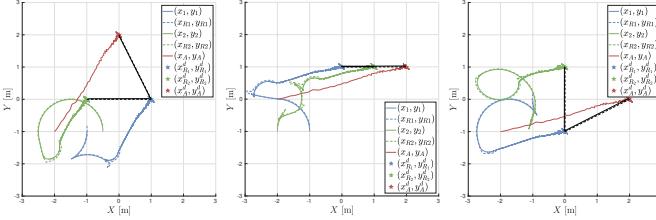


Fig. 6. Trajectories of the three agents with desired and actual interdistances in ideal simulations.



(a) Triangle Top      (b) Line Segment      (c) Triangle Right

Fig. 7. Trajectories of the three agents without APFs in ideal simulations.



(a) Triangle Top      (b) Line Segment      (c) Triangle Right

Fig. 8. Trajectories of the three agents with desired and actual interdistances in simulation with disturbances.

geometric shapes. The key performance index for the first scenario is  $\bar{t} = 54.76$  s, for the second case  $\bar{t} = 59.38$  s, and for the last case  $\bar{t} = 59.38$  s.

To demonstrate the importance of APFs, the simulations have been redone without repulsive forces. If a collision occurs this time, the ground vehicles will halt and remain stopped until the end of the test. Fig. 7 displays the resulting trajectories. In the first case, the curves are the same as Fig. 6a since the unicycles do not clash. However, the differences are obvious in the other two cases: compare Fig. 7b and Fig. 7c with Fig. 6b and Fig. 6c. It is clear that without APFs, the unicycles get too close and they must stop. Hence, even if the drone reaches the desired position  $(x_A^d, y_A^d)$ , the overall formation is not attained by the fleet. On the opposite, thanks to the artificial repulsive forces the robot  $G_2$  deviates its path to stay at a safe distance from  $G_1$ .

The last set of simulations aims to study the robustness of the proposed control scheme in the presence of the zero-mean disturbance  $\mathbf{d}(k)$ . The disturbance has been modeled in particular as a gaussian noise with variance  $\Sigma = 0.1I_{6 \times 6}$ . The results are presented in Fig. 8. The trajectories are similar to the ones in Fig. 6, but the agents in the end oscillate around the desired final positions due to disturbance  $\mathbf{d}(k)$  (though they remain within a small neighborhood). This is the reason why the solid and dotted lines do not exactly overlap: the dotted shape appears indeed to be a translated version of the solid one. However, collisions between  $G_1$  and  $G_2$  are avoided and the formation is reached after  $\bar{t} = 54.22$  s,  $\bar{t} = 112.14$  s, and  $\bar{t} = 112.1$  s respectively.

## 5. EXPERIMENTAL VALIDATION

Once analyzed the simulations and given the promising results, the next step has been to validate the formation control scheme through experiments. The tests have been carried out in an indoor arena equipped with a motion capture Optitrack system, used to measure the robots'

positions. For the communication between multi-agents, the TCP protocol and ROS1 Noetic facilities have been used. The control actions were computed into separate ROS nodes running on a ground station (i.e., the same laptop used for the simulations) sending then the commands to the separate robots. For controlling the quadcopter, the implementation of the inner controller has been done using the FL-AIR framework.

Initially, some preliminary tests were run using the gains used in the simulations, but the drone presented unwanted oscillations. Making an analogy with a PD controller, we would like indeed the gain  $K_A^*$  to have the following shape:

$$\begin{bmatrix} 0 & -K_P & 0 & -K_D \\ K_P & 0 & K_D & 0 \end{bmatrix} \quad (34)$$

Which is equivalent to two PD controllers, one for the X-dynamic and one for the Y-dynamic. Moreover, when tuning experimentally a PD attitude controller for the quadcopter, good performance were obtained with  $K_D \approx 1.222K_P = \alpha K_P$ . As tuning the weighting matrices to achieve the structure in (34) isn't trivial with the current definition of  $\mathbf{z}_A$  in (16a), the drone's measurement has been redefined to enforce this form, as follows:

$$\mathbf{z}'_A = \begin{bmatrix} e_{A,x} + \alpha \dot{e}_{A,x} \\ -e_{A,y} - \alpha \dot{e}_{A,x} \end{bmatrix} = C'_A \mathbf{e} \quad (35)$$

The min-max problem (20) has been consequently changed to:

$$\begin{aligned} \min_{\mathbf{u}(k)} \max_{\mathbf{e}(k) \neq 0} \frac{J(k)}{\|\mathbf{e}(k)\|_2^2} \\ \text{s.t. } \mathbf{z}'_A(k) = C'_A \mathbf{e}(k) \\ \mathbf{z}_i(k) = C_i \mathbf{e}(k) \quad i \in \{1, 2\} \\ \mathbf{u}_A(k) = \boldsymbol{\mu}_A(\mathbf{z}'_A(k)) \\ \mathbf{u}_i(k) = \boldsymbol{\mu}_i(\mathbf{z}_i(k)) \quad i \in \{1, 2\} \end{aligned} \quad (36)$$

With this new formulation, the weighting matrices have been chosen, following a tuning phase, as:

$$Q_e = \text{diag}(971.10, 971.10, 1000, 1000, 35, 35, 40, 40) \quad (37a)$$

$$Q_u = \text{diag}(0.01, 0.01, 0.08, 0.08, 0.08, 0.08) \quad (37b)$$

The resulting optimal gain matrices are

$$K'_A = \begin{bmatrix} -0.1258 & 0 \\ 0 & 0.1258 \end{bmatrix} \quad (38)$$

$$K'_1 = \begin{bmatrix} -7.8261 & 0 \\ 0 & -7.8261 \end{bmatrix} \quad (39)$$

$$K'_2 = \begin{bmatrix} -8.2934 & 0 \\ 0 & -8.2934 \end{bmatrix} \quad (40)$$

while the eigenvalues in closed-loop are:

$$p' = \text{eig}(A_d + B_d K'^* C') = \begin{cases} 0.8341 & \mu = 2 \\ 0.8435 & \mu = 2 \\ 0.9849 + 0.0163i & \mu = 2 \\ 0.9849 - 0.0163i & \mu = 2 \end{cases} \quad (41)$$

which are again inside the unitary circle, as shown in Fig 9.

The first experiments have been conducted with the ground free of obstacles. The resulting trajectories are shown in Fig. 10. The geometric shape is achieved by the fleet but still the quadrotor is not exactly in the desired position, denoted with the red star. The same behaviour has been observed in all the tests, showing that there

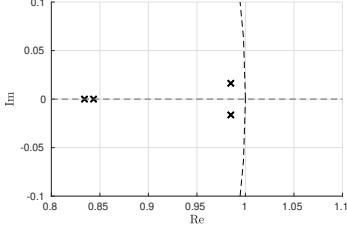


Fig. 9. Poles of the closed-loop system in the imaginary plane with the new drone's measurement definition.

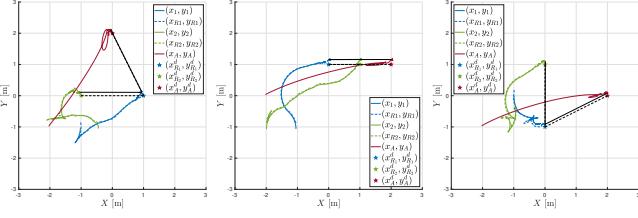
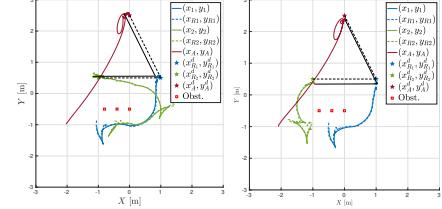


Fig. 10. Trajectories of the three agents with desired and actual interdistances in the experiments with the ground clear of obstacles.

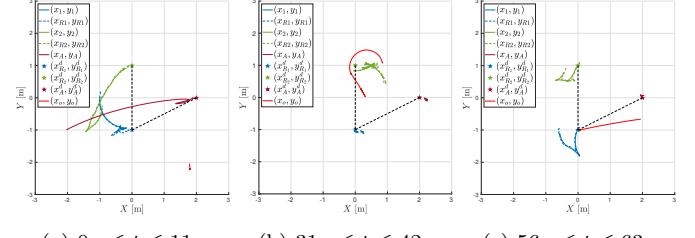
is still room for improvement. As it will be discussed in Section 6, a possible solution is to also include in the dynamics an integral action both in X and Y. The robots  $G_1$  and  $G_2$ , however, keep the desired distances from the drone and from  $G_2$ , respectively. For these reasons, the final geometry of the team (dotted black line) in Fig. 10 maintains the correct shape and orientation but translated from the desired configuration (solid line). Moreover, even if the desired final positions are inverted w.r.t the initial positions of the ground vehicles, they do not collide. For the quantitative analysis, the negligible error has been increased to  $\varepsilon = 0.20$  m. With this new definition, the formation is achieved after  $t = 8.92$  s,  $t = 10.04$  s,  $t = 10.34$  s in the three scenarios respectively. A video collecting the three experiments can be found at: **TODO**. At the end, even the ground vehicles appear to oscillate. This can be explained by the small oscillations of the drone around the desired position, which are then mirrored by the unicycles as they maintain the geometric shape. To prove this supposition, a trial was performed using only the ground robots, while the drone's position was mimicked by applying a constant setpoint throughout the test. The plots are not made for the sake of brevity, but the experiment is included in the previous video. In the video, it can be seen that once they reach the desired distance from the setpoint, the two Jetbot halt and hold their position.

Once the validation of formation maintenance and collisions prevention was complete, the next aspect to be investigated was obstacle avoidance, both still and in motion. To begin, static obstacles were considered. Specifically, the first experiment considered the geometric shape and the initial configuration shown in Fig.10a. To be accurate,  $y_A^d$  was increased by half a meter; otherwise, the unicycles could not have achieved the formation due to their close proximity to the obstacles. Moreover, this time three cones (in Fig.11, the red squares) were placed in front of the robots. The resulting trajectories are displayed in Fig.11a. Let it compare with Fig.10a. In both cases, the geometric



(a)  $R_2$  on the right      (b)  $R_2$  on the left

Fig. 11. Trajectories of the three agents with desired and actual interdistances in the experiments with static obstacles.



(a)  $0 \text{ s} \leq t \leq 11 \text{ s}$       (b)  $31 \text{ s} \leq t \leq 42 \text{ s}$       (c)  $56 \text{ s} \leq t \leq 63 \text{ s}$

Fig. 12. Trajectories of the three agents with desired and actual interdistances in the experiments with a dynamic obstacle.

shape is formed (in this case  $\bar{t} = 15.94$  s) without collisions between  $G_1$  and  $G_2$ . This time, however, the unicycles have to move avoiding the cones on the ground. Consequently, more time is required to reach the formation than in the obstacle-free case. On the one hand, without obstructions,  $G_1$  and  $G_2$  arrive at the desired distance from the drone and  $G_1$  in  $\bar{t}_1 = 6.60$  s and  $\bar{t}_2 = 8.92$  s, respectively. On the other hand, with the three cones, we have  $\bar{t}_1 = 9.80$  s and  $\bar{t}_2 = 15.94$  s. Moreover, the robot  $G_2$  does not go the shortest route towards the green star, but instead approaches it from the right to avoid collisions with  $G_1$ . To confirm this assumption, another experiment was conducted by swapping the ground robots at the start so they were already on the right side. Fig. 11b shows the results. As expected,  $G_2$  this time avoids the obstacles moving on the left-hand side of the cones. As expected,  $G_2$  this time avoids the obstacles by moving on the right-hand side of the cones. This is reflected in the key performance indexes:  $\bar{t} = 9.66$  s,  $\bar{t}_1 = 8.20$  s, and  $\bar{t}_2 = 9.66$  s. A video with the two experiments can be seen at: **TODO**.

In the final test, the cones were removed but a new ground vehicle playing the role of dynamic obstacles was introduced. While the other agents form the desired shape, the robot, guided using a joystick by an operator tries to disrupt the team. The geometric shape and the initial positions of the agents are the ones in Fig.10c. To better show and analyze the results, the robots' trajectories are plotted not for the entire test, but for some specific, interesting intervals of time. Let start by Fig. 12a. Initially, the dynamic obstacle is far from the agents, allowing them to reach the formation, as in the scenario free of obstacles in Fig. 10c. Let's now see what happens when the jammer gets close to the Jetbots. When the four-wheeled robot gets close to  $G_2$ , this last moves away, breaking the formation to keep a safe distance from the obstacle. Still when the jammer goes back to being far apart,  $G_2$  will position itself at the desired distance from  $G_1$ . When the obstacle gets

close to  $G_1$ , both the two-wheeled robots start moving but for two different reasons. On the one hand,  $G_1$  moves to be at a safe distance from the obstacle. On the other hand,  $G_2$  moves not because it is too close to the obstacle, but to keep the desired distance from  $G_1$ . A video of the test is available at: **TODO**.

## 6. CONCLUSION

...

## REFERENCES

- Aditya, P. and Werner, H. (2023). A distributed linear quadratic discrete-time game approach to formation control with collision avoidance. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, 1239–1244. doi:10.1109/CDC49753.2023.10384062.
- Deshpande, P., Menon, P.P., Edwards, C., and Postlethwaite, I. (2011). Formation control of multi-agent systems with double integrator dynamics using delayed static output feedback. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 3446–3451. doi:10.1109/CDC.2011.6161074.
- Gattami, A., Bernhardsson, B.M., and Rantzer, A. (2012). Robust team decision theory. *IEEE Transactions on Automatic Control*, 57(3), 794–798. doi:10.1109/TAC.2011.2168071.
- Güler, S. and Isa E. Yıldırım (2023). A distributed relative localization approach for air-ground robot formations with onboard sensing. *Control Engineering Practice*, 135, 105492. doi: <https://doi.org/10.1016/j.conengprac.2023.105492>.
- Han, L., Wang, Y., Yan, Z., Li, X., and Ren, Z. (2024). Event-triggered formation control with obstacle avoidance for multi-agent systems applied to multi-uav formation flying. *Control Engineering Practice*, 153, 106105. doi: <https://doi.org/10.1016/j.conengprac.2024.106105>.
- Huang, F., Duan, M., Su, H., and Zhu, S. (2023). Distributed optimal formation control of second-order multiagent systems with obstacle avoidance. *IEEE Control Systems Letters*, 7, 2647–2652. doi:10.1109/LCSYS.2023.3287950.
- Lee, G. and Chong, N.Y. (2009). Decentralized formation control for small-scale robot teams with anonymity. *Mechatronics*, 19(1), 85–105. doi: <https://doi.org/10.1016/j.mechatronics.2008.06.005>.
- Maldonado, D., Cruz, E., Abad Torres, J., Cruz, P.J., and Gamboa Benitez, S.d.P. (2024). Multi-agent systems: A survey about its components, framework and workflow. *IEEE Access*, 12, 80950–80975. doi:10.1109/ACCESS.2024.3409051.
- Morando, A.E.S., Bozzi, A., Zero, E., Sacile, R., and Castillo, P. (2025). Robust control architecture for fleet formation based on lmi and artificial potential fields. In *2025 20th Annual System of Systems Engineering Conference (SoSE)*, 1–6. doi:10.1109/SoSE66311.2025.11083835.
- Piet, B., Strub, G., Changey, S., and Petit, N. (2025). Control of an uav swarm in amorphous formation. In *2025 American Control Conference (ACC)*, 469–475. doi:10.23919/ACC63710.2025.11108054.
- Restrepo, E., Loría, A., Sarras, I., and Marzat, J. (2023). Robust consensus of high-order systems under output constraints: Application to rendezvous of underactuated uavs. *IEEE Transactions on Automatic Control*, 68(1), 329–342. doi:10.1109/TAC.2022.3144107.
- Semsar-Kazerooni, E. and Khorasani, K. (2009). An lmi approach to optimal consensus seeking in multi-agent systems. In *2009 American Control Conference*, 4519–4524. doi:10.1109/ACC.2009.5160268.
- Tran, V.P., Garratt, M.A., and Petersen, I.R. (2021). Multi-vehicle formation control and obstacle avoidance using negative-imaginary systems theory. *IFAC Journal of Systems and Control*, 15, 100117. doi: <https://doi.org/10.1016/j.ifacsc.2020.100117>.
- Trejo, J.A.V., Ponsart, J.C., Adam-Medina, M., Valencia-Palomo, G., and Theilliol, D. (2023). Distributed observer-based leader-following consensus control for lpv multi-agent systems: Application to multiple vtol-uavs formation control. In *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1316–1323. doi:10.1109/ICUAS57906.2023.10156012.
- Wang, H., Qiu, T., Liu, Z., Pu, Z., and Yi, J. (2020). Multi-agent formation control with obstacles avoidance under restricted communication through graph reinforcement learning. *IFAC-PapersOnLine*, 53(2), 8150–8156. doi: <https://doi.org/10.1016/j.ifacol.2020.12.2300>. 21st IFAC World Congress.
- Yongshen, L., Xuerong, Y., Yajun, Y., and Shengdong, P. (2018). Formation control of ugv based on artificial potential field. In *2018 37th Chinese Control Conference (CCC)*, 6830–6835. doi:10.23919/ChiCC.2018.8483701.
- Zhong, S., Tian, H., Xia, Y., and Liu, H. (2025). Quadrotor formation maneuver control via reinforcement learning. In *2025 44th Chinese Control Conference (CCC)*, 1–6. doi:10.23919/CCC64809.2025.11179569.