*Int. J. Appl. Math. Comput. Sci., 2022, Vol. 32, No. 3, 415–428*
DOI: 10.34768/amcs-2022-0030

# DECENTRALIZED MULTI–AGENT FORMATION CONTROL WITH POLE–REGION PLACEMENT VIA CONE–COMPLEMENTARITY LINEARIZATION

ANTONIO GONZÁLEZ [a,*], ANTONIO SALA [a], LEOPOLDO ARMESTO [b]

[a] University Institute of Control Systems and Industrial Computing (AI2)
Polytechnic University of Valencia
Camino de Vera, s/n, 46022 València, Spain
e-mail: angonsor@upv.es

[b] Design and Manufacturing Institute (IDF)
Polytechnic University of Valencia
Camino de Vera, s/n, 46022 València, Spain

An output-feedback decentralised formation control strategy is pursued under pole-region constraints, assuming that the agents have access to relative position measurements with respect to a set of neighbors in a graph describing the sensing topology. No communication between the agents is assumed; however, a shared one-way communication channel with a pilot is needed for steering tasks. Each agent has a separate copy of the same controller. A virtual structure approach is presented for the formation steering as a whole; actual formation control is established via cone-complementarity linearization algorithms for the appropriate matrix inequalities. In contrast to other research where only stable consensus is pursued, the proposed method allows us to specify settling-time, damping and bandwidth limitations via pole regions. In addition, a full methodology for the decoupled handling of steering and formation control is provided. Simulation results in the example section illustrate the approach.

**Keywords:** multi-agent autonomous system, formation control, decoupling, linear matrix inequalities, cone-complementarity linearization.

## 1. Introduction

The task of keeping a multi-agent system in the desired formation is known as formation control (Ren and Beard, 2008), which has played a crucial role in a great variety of applications over the last decades: UAV formation flight (Dong *et al.*, 2014; Zou *et al.*, 2018), underwater sensing networks (Bechlioulis *et al.*, 2019), cooperative transport (Bai and Wen, 2010), etc. The objective of formation control synthesis is to design a distributed control strategy based on local information exchange to achieve a geometrical formation shape in a coordinated fashion (Oh *et al.*, 2015). Formation control needs to operate jointly with *steering* tasks to set the translational and rotational degrees of freedom in complete multi-agent trajectory control problems.

A first approach to deal with these problems is to assume a virtual structure (Ren and Beard, 2004), where there exists a virtual reference frame that can be used to define each agent's position and velocity. A second approach is the leader–follower one (see He *et al.*, 2018; Kamel and Zhang, 2015; Farrera *et al.*, 2020), in which all agents try to reach given relative positions with respect to that of a 'leader.' The leader–follower approach is a particular case of more general *consensus* setups where a *virtual center* arises from properties of a graph which encodes the interaction's topology. When agents' manipulated variable is a speed command, they can be considered single integrators, so that the related problems are named first-order consensus (Ren and Sorensen, 2008); in the case where the agents' manipulated variable is acceleration, the problem under discussion is called second-order consensus (Ren and Atkins, 2005; Li *et al.*,

*Corresponding author

2015; Tian *et al*., 2018). Heterogeneous combinations of first- and second-order agents may also be considered (Rahimi *et al*., 2014; González-Sierra *et al*., 2022). There may exist a communication network supporting the formation control tasks so that agents can interchange states or reference values (Ren and Sorensen, 2008), or there may be communication-free structures where agents only have access to their own set of measurements (see, e.g., Dehghani and Menhaj, 2016). Obstacle-avoidance issues may also need to be considered (a behavior-based approach can be found in the work of Lee and Chwa (2018)), but that makes stability proofs more difficult; there are other neural/adaptive approaches for nonlinear agents (Peng *et al*., 2021), as well as developments including delay or fault tolerance (González *et al*., 2019). This work will discuss the communication-free case. Nonlinear setups, delay, faults, heterogeneous agents and obstacle avoidance issues will be intentionally left out of the scope of the present manuscript.

Control strategies for agents may be state-feedback ones (Ren and Beard, 2004; Ren and Atkins, 2005; Zhai *et al*., 2009), observer-based (Wen *et al*., 2016; Zhang and Chen, 2017), static output feedback (Zhai *et al*., 2009), or full output-feedback (Li *et al*., 2015; Tian *et al*., 2018). Stability proofs for the above formation-control tasks usually propose a fixed structure of each agent's control law and prove stability of an augmented canonical realization involving all agents in the linear case (Ren and Atkins, 2005; Wen *et al*., 2016), or a decrease in some Lyapunov function (Li *et al*., 2015; Tian *et al*., 2018). Zhai *et al*. (2009) use some matrix transformations to obtain stabilizing static output feedback controllers for a consensus problem via linear matrix inequalities (LMIs), actually, bilinear ones (BMIs) once decision variable products are acknowledged.

The last of the above works motivates the generalization we are addressing here; the goals of this paper are as follows: (a) stating the formation control problem as a three-task (consensus, translation, rotation) one where each of the tasks is decoupled from the other two ones; (b) allowing an arbitrary output-feedback consensus control with internal dynamics; (c) incorporating a two-degree-of-freedom setup, with feedforward terms from a virtual-structure setup for the steering tasks, in addition to the consensus-related feedback; (d) incorporating pole-region performance constraints apart from mere stability (other options are, of course, possible); (e) using a cone-complementarity linearization algorithm (El Ghaoui *et al*., 1997) to solve the resulting bilinear matrix inequalities.

In summary, the contribution of this paper is presenting a full methodology to address decentralized formation control (consensus) with only relative position feedback. Translation and rotational steering are also incorporated, decoupled from the consensus achievement. Our proposal allows specifying settling-time, and damping and maximum bandwidth constraints, contrarily to other research where only stability is pursued. Decoupling allows independent design of controllers for consensus, translation and rotation tasks. In-depth consideration of controller design for the steering tasks will not be considered in the scope of this work as, actually, they are double-integrator dynamics that, in principle, can be easily controlled.

The structure of the paper is as follows. The next section contains preliminaries about an agent's dynamics and sensor network as well as the problem statement, Section 3 discusses the steering component of the formation as a whole, whereas Section 4 details the formation-shape control error definition. Section 5 discusses the closed-loop dynamics under output-feedback plus steering feedforward control laws, and the decoupled representation of it. Section 6 details LMIs and the controller synthesis methodology. The paper ends with some examples in Section 7 and conclusions.

## 2. Preliminaries and problem statement

Consider a multiagent system, formed from $N$ identical agents, described by their acceleration dynamics as

$$\begin{aligned} \dot{p}_i &= v_i, \\ \dot{v}_i &= u_i + f_i, \end{aligned} \tag{1}$$

where $p_i \in \mathbb{R}^n$ and $v_i \in \mathbb{R}^n$ are respectively the position and velocity of each agent, $u_i \in \mathbb{R}^n$ is the control action, $f_i \in \mathbb{R}^n$ is an external disturbance, and $n$ represents the number of position variables (degrees of freedom). In the transfer-function notation, each agent would be described by $\frac{1}{s^2} I_n$, $s$ being the Laplace operator and $I_n$ the $n \times n$, identity matrix. Note that each component $p_{i,k}$, $k = 1, \ldots, n$, of $p_i$ is thus decoupled from the others, and can include any degree of freedom of the actual physical agent (spatial position coordinates, orientation ones, etc.).

The relative positions will be denoted as $\delta_{ij} := p_i - p_j$. We will assume that agents can only measure relative position data from a certain number of neighboring agents, i.e., only certain combinations of $i$ and $j$ will be available as measurements. The sensing capabilities of agents in the formation will be encoded in the *sensing* matrix:

$$\tilde{\mathcal{A}} = \begin{bmatrix} \tilde{\alpha}_{11} & \tilde{\alpha}_{12} & \cdots & \tilde{\alpha}_{1N} \\ \tilde{\alpha}_{21} & \tilde{\alpha}_{22} & \cdots & \tilde{\alpha}_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ \tilde{\alpha}_{N1} & \tilde{\alpha}_{N2} & \cdots & \tilde{\alpha}_{NN} \end{bmatrix}, \tag{2}$$

where $\tilde{\alpha}_{ij} = 1$, $[i, j] \in [1, \ldots, N] \times [1, \ldots, N]$ if agent $i$ can measure the relative position with respect to agent $j$, and $\tilde{\alpha}_{ij} = 0$ otherwise; diagonal elements will be assumed to be equal to 1. Obviously, the above matrix

has a graphical interpretation as the *adjacency matrix* of a directed *sensing graph* (Zhai *et al*., 2009; Ren and Beard, 2008; Farrera *et al*., 2020); hence $\tilde{A}$ will be named the adjacency matrix in the sequel.

For notational simplicity, we will assume the adjacency matrix to be identical for all degrees of freedom, although minor modifications in the developments would allow for different adjacency matrices; this common graph will entail identical dynamics in all degrees of freedom, so block-diagonal matrices will appear, using the Kronecker product notation; in particular, given a row vector $g = (g_1, \ldots, g_n)$, we set $g \otimes I_m = (g_1 I_m \; g_2 I_m \; \ldots g_n I_m)$, and $I_m \otimes g$ will be a block-diagonal matrix with $m$ repetitions of $g$ at the diagonal block elements; these expressions, among others, will be used in further developments. The mixed-product property of the Kronecker product $(A \otimes B) \cdot (C \otimes D) = (AC) \otimes (BD)$ will also be used later on.

It will be assumed that relative positions $\delta_{ij}$ can be acquired with respect to an 'absolute' inertial orientation reference frame, for instance, the one provided by an orientation-capable IMU in actual applications.

The formation pattern is defined through a set of desired (reference) relative position vectors $\delta^*_{ij}$, assumed to be known; actually, $\delta^*_{ij}$ may be non-constant functions of time to allow for rotation, scaling or other formation maneuvers in the so-called *steering* tasks.

Of course, the formation pattern's reference relative positions need not be "explicitly" set by the user for all pairs of agents, neither for those in the sensing graph with adjacency $\tilde{\mathcal{A}}$: indeed, as long as there is a directed graph $\mathcal{T}^*$ (nodes indicate agents, and edges between nodes $i$ and $j$ indicate the availability of the reference distance $\delta^*_{ij}$) with at least one spanning tree, then there is a path between any two arbitrary agents in $\mathcal{T}^*$; thus their relative distances can be computed (understanding $\delta^*_{ij} = -\delta^*_{ji}$ if a reverse path is taken on $\mathcal{T}^*$). Details are left to the reader. Actually, the need for such a spanning tree to be able to describe a formation leads to the assumption below (widespread in the literature cited in Introduction).

**Assumption 1.** There exists at least a spanning tree in the directed graphs described by the adjacency matrix $\tilde{\mathcal{A}}$.

**2.1. Problem statement.** The first objective of this paper is designing a dynamic feedback law which computes $u_i$ from available measurements (i.e., only $\delta_{ij}$ such that $\tilde{\alpha}_{ij} \neq 0$), in such a way that all agents can exponentially converge to a given formation pattern, defined through the set of desired relative position vectors between all agents, i.e., $\lim_{t\to\infty}(p_i - p_j) - \delta^*_{ij} = 0$ with some guaranteed exponential decay rate performance.

Additionally, our proposal will allow *decoupling* the whole problem of controlling an ensemble of multiple agents into (a) a "formation control" (also known as *consensus*) task (reaching prescribed relative trajectories with respect to the other agents with some pole-region performance objectives) and (b) a "steering" task in which a "virtual structure" (Ren and Beard, 2004) will be used to compute (as *feedforward* terms to the above formation control) the reference accelerations that each of the agents would have if the perfect formation had been attained. Subsequently, a second decoupling into *translational* vs. *rotational* steering will also be made at this level.

As the control structure to be proposed is decentralized, each agent's control algorithm will be intentionally made identical, only receiving shared linear and rotational acceleration commands from a 'pilot' (manual/automatic controller in charge of steering). Each agent's control will only be parameterized with some weights associated with the measurement graph and its relative position to a 'virtual center' of the formation's structure. We will propose matrix inequality conditions such that pole-region placement specifications can be proven for the consensus task, and an iterative algorithm based on cone-complementarity linearization (CCL) will be the choice for finding a feasible solution to these inequalities. The CCL algorithm will enable us to consider a generic quadratic Lyapunov function, instead of the diagonal-only case in other literature references, with, say, an agent-specific Lyapunov function (Rahimi *et al*., 2014).

## 3. Formation steering tasks

The movement of the formation as a whole will be denoted as steering, so an external controller (human/automatic 'pilot') will need to control the *position* and *orientation* of a virtual reference frame whose origin will be at a given *steering point*. Exploring which measurements and control strategy should be used for the steering tasks will not be discussed in this work, as they will have standard double-integrator dynamics which can be controlled with ease in most cases.

**3.1. Translational steering.** The 'pilot' will be assumed to provide linear acceleration $f^F$ commands based on his observations of the position and velocity of the steering point; indeed, the strategy to be developed in later sections, based in relative positions, will intentionally leave out double-integrator dynamics, uncontrollable from the proposed formation control strategy, but controllable by providing a common steering acceleration command $f^F$ to all agents.

**3.2. Rotational steering: A virtual structure.** Regarding the formation's orientation, we will assume the pilot to provide angular acceleration commands $\tau^F$, usually coming from observations of the orientation and angular velocity of the formation as a whole.

In particular, in a 3D setting (simpler formulas arise in planar motion, of course), the steering of the rotational degrees of freedom of the formation can be considered via integrating this rotational dynamics of a virtual rigid body, the so-called "virtual structure", in (Ren and Beard, 2004)

$$\dot{q}^F = \frac{1}{2}\Omega^q(\omega^F)\cdot q^F \tag{3}$$

$$\dot{\omega}^F = \tau^F, \tag{4}$$

where $q^F$ is a quaternion defining the formation's orientation, and $\Omega^q(\omega^F)$ is a $4\times 4$ matrix translating the concept of angular speed, $\omega^F = (\omega_1,\omega_2,\omega_3)^T$, to the quaternion framework:

$$\Omega(\omega^F) = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \tag{5}$$

$$\Omega^q(\omega^F) = \begin{pmatrix} \Omega(\omega^F) & \omega^F \\ -{\omega^F}^T & 0 \end{pmatrix}. \tag{6}$$

We have assumed that of inertia matrix the virtual structure is the identity matrix, in the same way, we assumed that each agent's mass was unity: we provide acceleration commands instead of 'virtual torques.'[1]

Now, the reference acceleration for each agent will be the sum of the translational acceleration and rotation-related components, in "absolute" coordinates in an inertial frame of reference.

Let us denote by $\bar{\delta}_i^*$ the constant relative position for agent $i$, in a reference inertial frame, with respect to a pre-defined "virtual center of rotation" (steering point).

Let us denote by $\mathcal{R}(q^F)$ the rotation matrix associated with quaternion $q^F$ defining orientation at a particular time instant. Then, the rotation of the formation would prescribe a reference position (relative to the center of rotation), speed and acceleration, for each agent, given by

$$\delta_i^* = \mathcal{R}(q^F)\cdot\bar{\delta}_i^* \tag{7}$$

$$\dot{\delta}_i^* = \Omega(\omega^F)\mathcal{R}(q^F)\cdot\bar{\delta}_i^* \tag{8}$$

$$\ddot{\delta}_i^* = \big(\Omega(\tau^F)\mathcal{R}(q^F) + \Omega(\omega^F)\Omega(\omega^F)\mathcal{R}(q^F)\big)\cdot\bar{\delta}_i^*, \tag{9}$$

$\Omega(\cdot)$ being a $3\times 3$ skew-symmetric matrix (5).

Note that acceleration in (9) contains only rotational-related elements, so the total desired acceleration for agent $i$ needs to add the translational component, yielding

$$u_i^F = \ddot{\delta}_i^* + f^F. \tag{10}$$

[1] Adding a virtual inertia matrix (different from a multiple of the identity matrix) is considered by Ren and Beard (2004) but, from an application point of view, it is perhaps detrimental, as the unstability of the "intermediate axis" rotation would entail steering problems in certain maneuvers that can be naively avoided with identity virtual inertia.

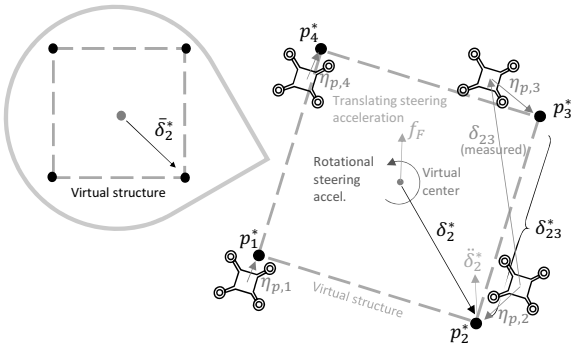



Fig. 1. Scheme of the different elements of the steering/formation control strategy. The left (not to scale) square represents the virtual structure in a fixed reference frame. The right square represents such a structure once translated and rotated.

The term $u_i^F$ represents a rigid virtual structure feedforward component, but an additional feedback component is needed to reach a given formation pattern. Figure 1 schematically depicts the above-discussed steering task elements.

## 4. Formation control error

Once the steering computations have been carried out, reference values for relative positions and velocities between agents (needed in formation control) trivially arise from defining

$$\delta_{ij}^* = \delta_i^* - \delta_j^*. \tag{11}$$

We will now discuss how the error with respect to the desired reference formation *shape* is understood so that each agent has a sort of reference point which depends on a weighted sum of the other agent's references (denoted as *consensus* in the literature).

Let us define the (weighted) formation control error for each agent's position as

$$\eta_{p,i} = \sum_{j=1}^{N}\alpha_{ij}\left(\delta_{ij} - \delta_{ij}^*\right), \tag{12}$$

where $\delta_{ij} = p_i - p_j$ is the error between *actual* agents positions, and $\alpha_{ij}$ are some weighting coefficients described by the following weighting matrix:

$$\mathcal{A} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1N} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_{N1} & \alpha_{N2} & \cdots & \alpha_{NN} \end{bmatrix}, \tag{13}$$

with $\alpha_{ij} = 0$ iff $\tilde{\alpha}_{ij} = 0$ for off-diagonal elements, and diagonal elements being arbitrary. In consequence, matrix $\mathcal{A}$ is a weighted version of the incidence matrix $\tilde{\mathcal{A}}$ in (2), with user-defined non-negative weights to encode the "importance" of some corrections, associated with, say, a "weighted sensing graph." Actually, weights $\alpha_{ij}$ will be useful to set a desired position of a "steering point" whose dynamics will be a double integrator, decoupled from formation control (Theorem 1 in Section 5, and the simulation example in Section 7).

Let us now denote by $\mathcal{L}$ the Laplacian matrix

$$\mathcal{L} = \mathcal{D} - \mathcal{A}, \tag{14}$$

with $\mathcal{D} = \operatorname{diag}_{i=1}^{N}\left(\sum_{j=1}^{N}\alpha_{ij}\right)$ being a matrix with incidence row sums as diagonal elements. From (14), it can be deduced that $\mathcal{L}$ is singular, and $1_{N\times 1}$ is always in its (right) null space. Also, Assumption 1 implies that the algebraic multiplicity of the null eigenvalue is actually 1, i.e., left and right null spaces of $\mathcal{L}$ are one-dimensional (Olfati-Saber and Murray, 2004).

The definition in (12) amounts to $\eta_p = \mathcal{L}p$, and that $\eta_p$ are invariant to translations (adding the column vector $\tau \in \mathbb{R}^n$ to all elements of $p$ would verify that $(\mathcal{L}\otimes I_n)\cdot p = (\mathcal{L}\otimes I_n)\cdot(p + 1_{N\times 1}\otimes\tau)$, as $\mathcal{L}\cdot 1_{N\times 1} = 0$).

Conversely, from $\eta_p$, positions can be recovered up to a translation. This is why driving $\eta_p$ to zero is understood as "formation control" because, when $\eta_p = 0$, relative distances $\delta_{ij}$ are equal to their reference values $\delta^*_{ij}$, and positions are determined up to a translation of the whole formation. The said translation of the whole formation will be accomplished via the control of the "steering point" mentioned in Section 3, yet in order to be decoupled from formation control, the steering point cannot be arbitrary but depending on the eigenvectors of $\mathcal{L}$, as discussed later.

In order to complete the understanding, the following corollary explains that pursuing $\eta_{p,i} = 0$ means that agent $i$ must track a certain "consensus" reference point $p^*_i$ (see (15) below) given by a weighted sum of positions of each of the agents (plus some constants). For notational brevity, shortcut $\sum_j$ will denote thereafter the sum $\sum_{j=1}^{N}$.

**Corollary 1.** *When the formation control error defined in* (12) *tends to zero, the position vector for each agent $p_i$ tends to*

$$p^*_i = \frac{1}{\sum_j \alpha_{ij}}\left(\sum_j \alpha_{ij}p_j + \sum_j \alpha_{ij}\delta^*_{ij}\right), \tag{15}$$

*where*

$$\frac{\sum_j \alpha_{ij}p_j}{\sum_j \alpha_{ij}}$$

*and*

$$\frac{\sum_j \alpha_{ij}\delta^*_{ij}}{\sum_j \alpha_{ij}}$$

*can respectively be viewed as convex combinations of the neighbor positions of the current agent (including the own agent $i$ if $\alpha_{ii} \neq 0$) and the references $\delta^*_{ij}$.*

*Proof.* The proof can be outlined from the equivalences

$$\begin{aligned}\eta_{p,i} &= \sum_j \alpha_{ij}\left(\delta_{ij} - \delta^*_{ij}\right)\\ &= \sum_j \alpha_{ij}\left(p_i - p_j - \delta^*_{ij}\right)\\ &= \left(\sum_j \alpha_{ij}\right)\cdot p_i - \sum_j \alpha_{ij}(p_j - \delta^*_{ij}).\end{aligned}$$

Hence

$$\eta_{p,i} = \left(\sum_j \alpha_{ij}\right)\cdot(p_i - p^*_i),$$

i.e., coordinates $\eta$ are proportional to the "error" between an agent's position and $p^*_i$, which can be interpreted as a "position reference" for the said agent. ■

An illustration of the above idea appears, too, in Fig. 1.

## 5. Closed-loop formation control with feedforward terms

**5.1. Open-loop state-space model.** The open-loop dynamics of the multiagent system is determined from (1) by the time-derivative of (12):

$$\dot{\eta}_{p,i} = \eta_{v,i} \tag{16}$$

$$\begin{aligned}\dot{\eta}_{v,i} = &\sum_j \alpha_{ij}\left(u_i - u_j\right) + \sum_j \alpha_{ij}\left(f_i - f_j\right)\\ &- \sum_j \alpha_{ij}\ddot{\delta}^*_{ij},\end{aligned} \tag{17}$$

where, implicitly, we have defined $\eta_{v,i} = \sum_j \alpha_{ij}(\dot{\delta}_{ij} - \dot{\delta}^*_{ij})$ to write (16), and velocities and accelerations of the reference relative positions (11) appear in (17).

Now, letting

$$\begin{aligned}\eta_p &= \begin{bmatrix}\eta_{p,1}^T & \cdots & \eta_{p,N}^T\end{bmatrix}^T,\\ \eta_v &= \begin{bmatrix}\eta_{v,1}^T & \cdots & \eta_{v,N}^T\end{bmatrix}^T,\\ u &= \begin{bmatrix}u_1^T & \cdots & u_N^T\end{bmatrix}^T,\\ f &= \begin{bmatrix}f_1^T & \cdots & f_N^T\end{bmatrix}^T,\\ \delta^* &= \begin{bmatrix}\delta_1^{*T} & \cdots & \delta_N^{*T}\end{bmatrix}^T,\end{aligned} \tag{18}$$

the above expressions can be written in compact form, for the whole formation, as

$$\begin{bmatrix}\dot{\eta}_p\\ \dot{\eta}_v\end{bmatrix} = \begin{bmatrix}0 & I_N\otimes I_n\\ 0 & 0\end{bmatrix}\begin{bmatrix}\eta_p\\ \eta_v\end{bmatrix} + \begin{bmatrix}0\\ \mathcal{L}\otimes I_n\end{bmatrix}\left(u + f - \ddot{\delta}^*\right). \tag{19}$$

**5.2. Output feedback control.** In the sequel, sensors for relative velocities will be assumed to be unavailable. Hence, an output feedback control scheme will be thereafter considered, using only relative position feedback from graph neighboring agents.

Consider the following dynamic output feedback control scheme of order $q$ with state and output equations

$$\dot{\xi}_i = F\xi_i + G\eta_{p,i}, \tag{20}$$

$$u_i^{FB} = K\xi_i + H\eta_{p,i}, \tag{21}$$

where $\xi_i \in \mathbb{R}^q$ is the controller's state variable of the vector for agent $i$. Accordingly, once feedforward steering terms (10) are added, the actual control action for each agent is

$$u_i = u_i^{FB} + u_i^F. \tag{22}$$

We assume a decentralized implementation, where the same controller is used by each agent. Thus, $F \in \mathbb{R}^{q\times q}$, $G \in \mathbb{R}^{q\times n}$, $K \in \mathbb{R}^{n\times q}$ and $H \in \mathbb{R}^{n\times n}$ are the controller parameters to be designed.

Note that the feedforward term is an absolute acceleration which depends on the position of the agent in the virtual structure, as well as the rotational position, velocity and acceleration of the virtual structure's reference frame. This term can thus be computed by a centralized "steering computer" or, if a model of the virtual structure is available to each agent, then it can carry out the computations (7)–(9) independently of the other agents.

**5.3. Closed-loop formation dynamics.** With the above closed-loop controllers, the closed-loop formation acceleration can be expressed as

$$\dot{\eta}_{v,i} = \sum_j \alpha_{ij}\left(u_i^{FB} - u_j^{FB}\right) + \sum_j \alpha_{ij}\left(f_i - f_j\right) \tag{23}$$

because the feedforward terms $u_i^F$ cancel out the reference accelerations $\ddot{\delta}^*$ in (19). Hence, a representation of the closed-loop system formed by (1) and the dynamic controller (20)–(22) is obtained as

$$\dot{x} = \bar{A}x + \bar{B}f, \tag{24}$$

where

$$\begin{aligned} x &= \begin{bmatrix}p^T & v^T & \xi^T\end{bmatrix}^T,\\ p &= \begin{bmatrix}p_1^T & \cdots & p_N^T\end{bmatrix}^T,\\ v &= \begin{bmatrix}v_1^T & \cdots & v_N^T\end{bmatrix}^T,\\ \xi &= \begin{bmatrix}\xi_1^T & \cdots & \xi_N^T\end{bmatrix}^T,\\ f &= \begin{bmatrix}f_1^T & \cdots & f_N^T\end{bmatrix}^T, \end{aligned} \tag{25}$$

$$\bar{A} = \begin{bmatrix}0 & I_N\otimes I_n & 0\\ \mathcal{L}\otimes H & 0 & (I_N\otimes K)_{nN\times qN}\\ \mathcal{L}\otimes G & 0 & I_N\otimes F\end{bmatrix}, \tag{26}$$

$$\bar{B} = \begin{bmatrix}0 & I_N\otimes I_n & 0\end{bmatrix}^T.$$

The closed-loop dynamics will be shown to have two uncontrollable poles per degree of freedom, associated to steering (Section 3). These poles must be "separated" from the dynamics for any stabilizing formation control design to be feasible; this is the objective of the next section.

**5.4. Decoupled representation of closed-loop dynamics.** The following theorem proves that there exists a "steering point" $p^s \in \mathbb{R}^n$ for the closed-loop formation control, whose acceleration is always zero in the absence of external forces and formation control is decoupled from the movements of this point.

**Theorem 1.** *Let $\gamma$ be a $1\times N$ row vector in the left null space of the Laplacian matrix $\mathcal{L}$ of the weighted sensing graph,* (14)*, such that $\gamma\cdot 1_{N\times 1} = 1$. Set $p^s = (\gamma\otimes I_n)\cdot p$, where $p$ is the column vector comprised of the juxtaposition of the $n$ positions; $p^s \in \mathbb{R}^n$ will be referred to as the 'steering point.'*

*If the initial conditions of the control state variables $\xi(0)$ satisfy*

$$(\gamma\otimes I_q)\cdot\xi(0) = 0, \tag{27}$$

*then, for all $t \geq 0$, the dynamics of the multiagent formation can be expressed as*

$$\left[\begin{array}{c}\dot{p}^s\\ \dot{v}^s\\ \hline \dot{z}\end{array}\right] = \left[\begin{array}{cc|c}0 & I_n & 0\\ 0 & 0 & 0\\ \hline 0 & 0 & A\end{array}\right]\left[\begin{array}{c}p^s\\ v^s\\ \hline z\end{array}\right] + \left[\begin{array}{c}0\\ \gamma\otimes I_n\\ \hline J_2\end{array}\right] f, \tag{28}$$

*where*

$$J_2 = \begin{bmatrix}0_{nN\times nN}\\ I_N\otimes I_n - \mathcal{W}(\gamma\otimes I_n)\\ 0_{qN\times nN}\end{bmatrix}, \tag{29}$$

*$z$ being the formation-specific state vector formed by the juxtaposition of the relative position and velocities with respect to the steering point (say, $p_i - p^s$ and $v_i - v^s$, $i = 1, \ldots, N$), expressed as*

$$z = \begin{bmatrix} (p - \mathcal{W} \cdot p^s)^T & (v - \mathcal{W} \cdot v^s)^T & \xi^T \end{bmatrix}^T, \tag{30}$$

*where $\mathcal{W} = 1_{N\times 1} \otimes I_n$.*

*Proof.* From the theorem statement, we have that $\gamma\mathcal{L} = 0$. Considering the full closed-loop state vector $\bar{x}$, note that $p^s = \bar{\gamma} \cdot x$ with $\bar{\gamma} = \begin{bmatrix} \gamma \otimes I_n & 0 & 0 \end{bmatrix}$.

Recalling that $x$ is of size $2nN + qN$ and $\bar{\gamma}$ is of size $n \times (2nN + qN)$, the acceleration of the steering point can be obtained as

$$\ddot{p}^s = \bar{\gamma} \cdot \ddot{x} = \bar{\gamma}\bar{A}^2 x + \bar{\gamma}\bar{A}\bar{B} f + \bar{\gamma}\bar{B}\dot{f},$$

where $\bar{A}$ and $\bar{B}$ are defined in (26). From $\bar{A}$ and $\bar{B}$ in (26), it can be deduced that

$$\begin{aligned} \bar{\gamma}\bar{A}^2 &= [0_{n\times nN} \quad 0_{n\times nN} \quad (\gamma \otimes I_n)_{n\times nN} \\ &\qquad \cdot (I_N \otimes K)_{nN\times qN}], \\ \bar{\gamma}\bar{A}\bar{B} &= \gamma \otimes I_n, \\ \bar{\gamma}\bar{B} &= 0. \end{aligned} \tag{31}$$

Applying the mixed-product property of the Kronecker product, one has that

$$(\gamma \otimes I_n)_{n\times nN} \cdot (I_N \otimes K)_{nN\times qN} = (\gamma \otimes K)_{n\times qN},$$

and therefore the acceleration of the pseudo-center of gravity yields

$$\ddot{p}^s = \Gamma x + (\gamma \otimes I_n) f \tag{32}$$

with $\Gamma = \begin{bmatrix} 0_{n\times N} & 0_{n\times N} & \gamma \otimes K \end{bmatrix}$.

Hence, from (32), with the initial condition for $\bar{\xi}$ satisfying (27), we can conclude that $\Gamma = 0$ because $(\gamma \otimes K) \cdot \xi = K \cdot (\gamma \otimes I_q) \cdot \xi$ for any matrix $K$.

This property also holds for higher time-derivatives of $p^s$. Indeed, $\mathrm{d}^k p^s / \mathrm{d}t^k = \bar{\gamma}\bar{A}^k x$.

Let us now prove, by induction, that $\bar{\gamma}\bar{A}^k = [0\ 0\ \gamma \otimes R^{(k)}]$ with $R^{(k)} = KF^{k-2}$, for $k = 2, 3, \ldots$.

Indeed, the assertion is true for $k = 2$, with $R^{(2)} = K$. Now,

$$\begin{aligned} &\begin{bmatrix} 0_{n\times nN} & 0_{n\times nN} & \gamma \otimes R^{(k-1)} \end{bmatrix} \bar{A} \\ &\quad = [-(\gamma \otimes R^{(k-1)}) \cdot (\mathcal{L} \otimes G) \quad 0 \\ &\qquad (\gamma \otimes R^{(k-1)}) \cdot (I_N \otimes F)] \end{aligned} \tag{33}$$

but

$$\begin{aligned} &-(\gamma \otimes R^{(k-1)}) \cdot (\mathcal{L} \otimes G) \\ &\qquad = -(\gamma \cdot \mathcal{L}) \otimes (R^{(k-1)} \cdot G) = 0, \end{aligned}$$

and, from the same mixed-product property, one has that

$$\begin{aligned} &-(\gamma \otimes R^{(k-1)}) \cdot (I_N \otimes F) \\ &\qquad = -(\gamma \cdot I_N) \otimes (R^{(k-1)} \cdot F). \end{aligned}$$

Henceforth, $\gamma \cdot I_N = \gamma$ and

$$R^{(k)} = R^{(k-1)} \cdot F = KF^{k-2}, \quad k \geq 2.$$

Hence, we have proved that $\bar{\gamma}\bar{A}^k = [0\ 0\ \gamma \otimes R^{(k)}]$ with $R^{(k)} = KF^{k-2}$ for all $k \geq 2$.

Note also that $(\gamma \otimes R^{(k)}) \cdot \xi = R^{(k)} \cdot (\gamma \otimes I_q) \cdot \xi$ for any matrix $R^{(k)}$ of compatible dimensions. Hence, if $(\gamma \otimes I_q) \cdot \bar{\xi} = 0$, then $(\gamma \otimes R^{(k)}) \cdot \bar{\xi} = 0$. Thus, with the initial condition for $\bar{\xi}$ satisfying (27), all derivatives of $p^s$ from the second onwards are zero under no external forces (because expressions $\bar{\gamma}\bar{A}^k\bar{B}$ appearing in higher-order derivatives of the steering point are zero), so the dynamics of $p^s$'s are those of a double integrator excited by an external force.

Once the steering point dynamics have been established, we will introduce the relative displacement vector $z$ from (30).

Then, it is easy to show that the joint dynamics of the steering point and of $z$, composed of agents' relative positions, velocities and controller states, are governed by

$$\begin{aligned} \left[\begin{array}{c} \dot{p}^s \\ \dot{v}^s \\ \hline \dot{z} \end{array}\right] &= \left[\begin{array}{ccc|c} 0 & I_n & 0 & \\ 0 & 0 & 0 & \\ \hline J_1 & 0 & A & \end{array}\right] \left[\begin{array}{c} p^s \\ v^s \\ \hline z \end{array}\right] \\ &\quad + \left[\begin{array}{c} 0 \\ \gamma \otimes I_n \\ \hline J_2 \end{array}\right] \bar{f}, \end{aligned} \tag{34}$$

where matrix $\bar{A}$ was defined in (26), and

$$\begin{aligned} J_1 &= \begin{bmatrix} 0_{nN\times n} \\ 0_{qN\times n} \\ (\mathcal{L} \otimes G) \cdot \mathcal{W} \end{bmatrix}, \\ J_2 &= \begin{bmatrix} 0_{nN\times nN} \\ I_N \otimes I_n - \mathcal{W}(\gamma \otimes I_n) \\ 0_{qN\times nN} \end{bmatrix}. \end{aligned} \tag{35}$$

Recalling that $1_{N\times 1}$ is the right eigenvector associated with the zero eigenvalue of the Laplacian matrix $\mathcal{L}$, it can be seen that $(\mathcal{L} \otimes G) \cdot \mathcal{W} = (\mathcal{L} \otimes G) \cdot (1_{N\times 1} \otimes I_n) = (\mathcal{L} \cdot 1_{N\times 1}) \otimes (G \cdot I_n) = 0$, leading to $J_1 = 0$ in (35). Consequently, as $J_1 = 0$, the expression (28) is obtained. ∎

**Minimal realization.** The interpretation of the above theorem is that the multi-agent system is seen as a plain 'double integrator' if only the steering point is taken into account, so formation control is "decoupled" from steering, and an external driver/pilot needs only

taking care of the movement of $p^s$. Formation control dynamics and those of the steering point are decoupled from the block diagonal structure of (28). The theorem shows that the stability analysis of the formation control must be addressed without considering the steering point's double-integrating behavior, i.e., analyze only the dynamics of $z$.

However, the realization (28) is nonminimal, as $p^s$ is a linear combination of agent states. Recalling that $p^s = \bar{\gamma}p$ and $v^s = \bar{\gamma}v$, it can be deduced that $\bar{\gamma} \cdot (p - \mathcal{W}\, p^s) = 0$, $\bar{\gamma} \cdot (v - \mathcal{W}\, v^s) = 0$ and, therefore, rewriting

$$\Gamma_\gamma \bar{z} = 0, \qquad \Gamma_\gamma := \begin{bmatrix} \bar{\gamma} & 0 & 0_{n\times qN} \\ 0 & \bar{\gamma} & 0_{n\times qN} \end{bmatrix} \tag{36}$$

indicates that there are $2n$ uncontrollable modes in formation control (weighted sum of relative positions and velocities) that can be readily eliminated using the change of variable $z = [\mathrm{null}(\Gamma_\gamma) \quad \Gamma_\gamma^T]\bar{z}_r$, discarding the last two elements of $\bar{z}_r$ (uncontrollable), compensating for the addition of $p^s$ and $v^s$ in (28). As the change of the variable is linear, the resulting reduced-order dynamics, whose state transition matrix will be denoted as $\bar{A}_r(F, G, K, H)$, will also be expressed as an affine function of controller matrices (decision variables), so the proposed control synthesis methodology will also apply to the transformed matrices with straightforward modifications.

Note that, actually, it is mandatory to work on the minimal realization of the formation control arising from $\bar{z}_r$; otherwise, the uncontrollable modes associated with the (marginally) unstable steering dynamics would preclude obtaining feasible asymptotically stable solutions to the matrix inequalities.

## 6. Control synthesis

The design of the parameters $F, G, K$ and $H$ in (20), (21) can be addressed by means of iterative convex optimization algorithms, once the independence from the steering task has been established by Theorem 1 and the cancellation of feedforward terms in (21).

As the translational/rotational steering dynamics reduce to a double integrator for each degree of freedom, a controller for it can be designed with well-known textbook methods or, possibly, left to an external "pilot" to care about it, maybe with visual feedback. Anyway, designing controllers for the steering task is intentionally out of the scope of this work.

**Closed-loop specifications.** The design of the controller parameters $F, G, K$ and $H$ is carried out with the objective being to ensure the convergence of the formation control to the desired position with a prescribed decay and damping.

Formally, specifications for control synthesis will be set up based on the D-stability criterion (Peaucelle *et al.*, 2000), prescribing a region (the shadowed area in Fig. 2) where closed-loop poles must lie. The parameter $r$ (the radius of the right circle) should be chosen to respect a certain bandwidth limitation (an upper bound to the natural frequency), while $\sigma = q_c + \rho$ (bounding the real part of the closed-loop poles) should be negative, and ensuring a minimum settling time (decay rate bound) to reach the formation pattern. The damping characteristics are determined by

$$\cos(\theta) = \frac{r^2 + q_c^2 - \rho^2}{2r|q_c|}.$$

Note that decreasing radius $\rho$ has the beneficial effect of improving both the decay rate and the damping coefficient.

Let $\bar{n}$ be the size of $\bar{A}_r$, $\bar{A}_r$ being the state-transition matrix of the reduced-order dynamics discussed at the end of Section 5.4, depending on the controller matrices. The existence of symmetric matrices $P_1, P_2, Q_1, Q_2 \in \mathbb{R}^{\bar{n}} > 0$ and control parameters $F, G, K, H$ (embedded into $\bar{A}_r$) satisfying the constraints $P_1Q_1 = I$, $P_2Q_2 = I$ and the linear matrix inequalities

$$\begin{bmatrix} -r^2P_1 & \bar{A}_r^T \\ (*) & -Q_1 \end{bmatrix} < 0, \qquad \begin{bmatrix} -\rho^2P_2 & \left(\bar{A}_r - q_cI\right)^T \\ (*) & -Q_2 \end{bmatrix} < 0 \tag{37}$$

ensures that all closed-loop poles are located in the convex set given in Fig. 2. Indeed, the Schur complement makes the above inequalities equivalent to $r^2P_1 - A^TP_1A > 0$, $\rho^2P_2 - (A - q_cI)^TP_2(A - q_cI) > 0$, $P_1 > 0$, $P_2 > 0$, which are well-known conditions for the eigenvalues of $A$ being in the disc with radius $r$ and those of $A - q_cI$ in the disc with radius $\rho$, both centered at the origin. Of course, other desired pole-regions shapes are also possible if desired, using the above-mentioned D-stability conditions.

In this work the final goal pole region will be iteratively reached, as discussed below, from an initial controller placing closed-loop poles in a larger region which will progressively shrink and/or move left.

**Integral action.** Apart from the translational steering acceleration $f_F$, the presence of arbitrary external disturbances as part of $f_i$ may change the formation's shape. If low-frequency disturbance rejection is desired, the control scheme (21) and (20) can be modified by introducing an integral action so as the effect of constant loads in the formation control can be steady-state rejected. In such a case, it is sufficient to choose the controller gains $F$ and $G$ in (20) of the following structure:

$$F = \begin{bmatrix} 0_n & 0 \\ \mathcal{G}_1 & \mathcal{F} \end{bmatrix}, \quad G = \begin{bmatrix} I_n \\ \mathcal{G}_2 \end{bmatrix} \tag{38}$$

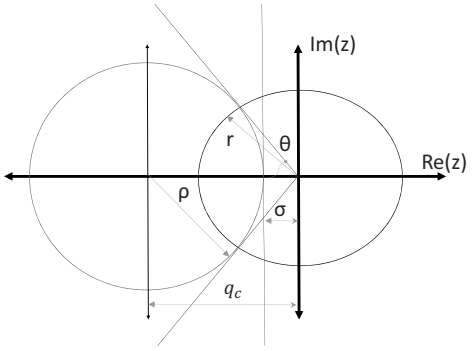



Fig. 2. Desired pole region, intersection of two discs $D(0, r)$ and $D(q_c, \rho)$. Design parameters are $r > 0$, $\rho > 0$, $q_c < 0$; see the decay rate ($\sigma$) and damping ($\theta$) formulas in the main text.

for some matrices $\mathcal{G}_1$, $\mathcal{F}$, $\mathcal{G}_2$ of appropriate dimensions. As (38) is a particular case of (20), incorporation of the integral action is straightforward if desired.

**Solution via the CCL+performance-shaping algorithm.** Note that the conditions (37) with the equality constraints $P_1Q_1 = I$, $P_2Q_2 = I$ render the problem non convex. Therefore, finding a feasible solution to ensure that all poles are located in the region corresponding to the intersection between both circles in Fig. 2 is addressed via the cone-complementarity linearization (CCL) algorithm (El Ghaoui *et al.*, 1997) by relaxing the equality constraints $P_lQ_l = I$ to

$$\begin{bmatrix} P_1 & I \\ I & Q_1 \end{bmatrix} \geq 0, \quad \begin{bmatrix} P_2 & I \\ I & Q_2 \end{bmatrix} \geq 0, \tag{39}$$

together with the objective function

$$\min \big(\mathrm{trace}\big(\sum_{l=1}^{2} \big(P_lQ_l^{(k-1)} + Q_lP_l^{(k-1)}\big)\big)\big) \tag{40}$$

to minimize at each CCL iteration $k = 1, 2, \ldots$ until the trace is $2\bar{n}$, where $P_l^{(0)}$, $Q_l^{(0)}$, $l = 1, 2$ are obtained from the feasible solution for a given initial controller and associated (relaxed) performance levels. Actually, CCL iterations are embedded into an outer iteration loop with progressively more stringent performance constraints (37) until CCL's lower trace bound ceases to be attainable.

An initial controller and associate performance levels are needed. Intuitively, a reasonably performing controller for a single agent is suggested.

This provides a sensible starting point so that CCL+performance shaping iterations can retune such a controller to accommodate the interaction with the rest of agents in the formation. Indeed, when inserted into the formation, LMI-guaranteed initial values of $\rho$ and $r$ will, expectedly, not be satisfactory, but the above performance requirements can be made more stringent by decreasing $\rho$ (better damping), increasing $\sigma$ (faster decay rate) or decreasing $r$ (lower natural frequency), setting up suitable step sizes and bisection-like optimization (details omitted for brevity). For simplicity, only the decrease in $\rho$ will be pursued in the case study in the next section.

## 7. Case study

The proposed formation control strategy will be illustrated by the following four-agent setup.

**System description.** Consider a multiagent system formed by $N = 4$ agents whose sensing graph is depicted in Fig. 3. The adjacency matrix $\tilde{\mathcal{A}}$ describing the sensing graph (2) and the weighted adjacency matrix in (13) are therefore defined by

$$\tilde{\mathcal{A}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{41}$$

$$\mathcal{A} = \begin{bmatrix} 0 & \tilde{\alpha}_{12} & 0 & 0 \\ 0 & 0 & \tilde{\alpha}_{23} & \tilde{\alpha}_{24} \\ \tilde{\alpha}_{31} & 0 & 0 & 0 \\ 0 & \tilde{\alpha}_{42} & 0 & 0 \end{bmatrix}, \tag{42}$$

where the weighting coefficients $\tilde{\alpha}_{ij} \neq 0$ can be designed to set the position of the steering point $p^s = (\gamma \otimes I_n) \cdot p$. In this example, we have

$$\gamma = [\tilde{\alpha}_{23}\tilde{\alpha}_{42}\tilde{\alpha}_{31} \quad \tilde{\alpha}_{42}\tilde{\alpha}_{31}\tilde{\alpha}_{12} \quad \tilde{\alpha}_{23}\tilde{\alpha}_{42}\tilde{\alpha}_{12} \quad \tilde{\alpha}_{12}\tilde{\alpha}_{24}\tilde{\alpha}_{31}].$$

Here, we wish to put the steering point in the center of the multiagent system's desired formation, say, $\gamma = 0.25 \cdot [1, 1, 1, 1]$. Under this requirement, a feasible option for $\tilde{\alpha}_{ij}$ is

$$\tilde{\alpha}_{12} = 0.5, \quad \tilde{\alpha}_{23} = 0.5, \quad \tilde{\alpha}_{24} = 1, \quad \tilde{\alpha}_{31} = 0.5, \quad \tilde{\alpha}_{42} = 1.$$

**CCL and performance-shaping initialization.** Initial controllers for the CCL algorithm have been set to the state-space representation of stabilizing controllers for a SISO double integrator with closed-loop poles inside a desired pole region, i.e., close to $s = -2$. An additional integral action was added via an augmented realization with an input disturbance. The initial control was designed via state-feedback+observer pole placement, so controller

$$G_r(s) = \frac{-84.88(s^2 + 1.02s + 0.42)}{s(s^2 + 10.20s + 41.61)}$$

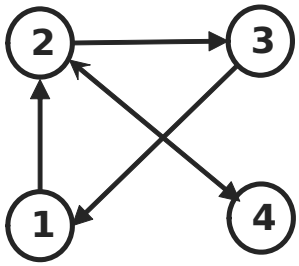



Fig. 3. Sensing graph for the example.

stabilizes a SISO double integrator with closed-loop poles at $[-2, -2.02, -2.04, -2.06, -2.08]$, while rejecting input disturbances (wind) with no steady-state error (indeed, an integral action is present in the controller). A transformation to a modal canonical form was used to express the state-space representation of $G_r(s)$ in the form of (38), with details left to the reader; this is required in our developments, in order to keep the controller's integrator fixed in the iterations.

Of course, due to the coupling between agents, the actual closed-loop poles of the formation-control state-transition matrix $\bar{A}_r$ will lie in a different position (otherwise, the problem's solution would be trivial), but the above initial setup is considered to be a sensible CCL starting point. Specifically, the initial control $G_r(s)$, with the given communication topology in (41), leads to a closed loop system with poles depicted as the lighter gray marks in Fig. 4. Thus, setting $\rho = 29.95$, $r = 28$, $q_c = -30$ (i.e., $\sigma = -0.05$) allowed obtaining the initial feasible performance levels and associated matrices $P_1^{(0)}$, $P_2^{(0)}$, $Q_1^{(0)}$, $Q_2^{(0)}$. The chosen values for $r$, $q_c$ and $\rho$ do overbound the actually achieved pole region: this is intentional in order to have room for improving the decay rate $\sigma$, which was selected as our main performance goal.

Note that single-agent decay-rate performance was, ideally, $-2$, but the coupling between agents (not considered in the initial controller) forced us to set a bound of $-0.05$: the naive, non-optimized, initial formation-control was at the verge of instability; low-damped poles were also present (see Fig. 4). This is why a performance improvement was intentionally shaped in the iterations via decreasing $\rho$, which improves both the decay rate and damping coefficients.

**Results.** After applying the above-described CCL algorithm by decreasing $\rho$ at each CCL iteration, we obtain a stabilizing dynamic formation control (20)–(21) after 82 performance-shaping iterations with all closed-loop poles in the region determined by the intersection of the circles of Fig. 2 with $r = 28$ and $q_c = -30$ and $\rho = 29.54$. In this way, the decay rate bound was improved from the initial value of $-0.05$ to a value of $\sigma = -0.46$, i.e., roughly a 9-fold increase. Furthermore, actually achieved decay rates (dominant poles) are better than that given by $\sigma$, as LMIs provide only a worst-case bound, given our two-circle setup, as discussed later.

The finally designed control parameters are

$$\begin{aligned} F &= \begin{bmatrix} 0 & 0 & 0 \\ -1.11 & -13.78 & 2.03 \\ -2.75 & 2.03 & -7.42 \end{bmatrix}, \\ G &= \begin{bmatrix} 1.00 \\ -3.15 \\ -11.02 \end{bmatrix}, \\ K &= \begin{bmatrix} -7.25 & -3.68 & -12.45 \end{bmatrix}, \\ H &= -24.81. \end{aligned} \tag{43}$$

If desired, the agent's formation controller can also be expressed in transfer function form as

$$G_r(s) = \frac{-24.81s^3 - 384.5s^2 - 411.9s - 160.1}{s^3 + 21.21s^2 + 98.16s}.$$

The lighter gray marks in Fig. 4 depict the pole location in the complex plane with the initial design, and the darker gray marks denote the pole location with the designed control parameters (43) via CCL. The displacement of the closed-loop poles at each algorithm iteration is also depicted in Fig. 4 as black lines (some "jagged" artifacts appear due to step-size/bisection steps, but that may be expected and poses no problem in our setup). Figure 5 shows the evolution of the real part of the dominant pole (top, the more negative, the better), worst-case damping coefficient $\cos(\theta)$ (bottom, the larger the better). It can be appreciated in Figs. 5 and 4 that the real part of the dominant pole (convergence ratio) is decreased from $-0.13$ to $-0.61$ with the control design (43). Moreover, the worst-case damping coefficient is increased from 0.1 to 0.37.

*Closed-loop simulation.* Two simulations have been carried out: The first one (Simulation 1) will be made from out-of-formation initial conditions, in the absence of external disturbance forces. In this case, from the results in this work, steering will not be considered as the steering point will remain fixed, not affected by formation control dynamics. Indeed, Fig. 6 shows that all agents effectively reach the desired formation target, and it can be appreciated that the steering point $p^s$ remains in the same location. The time evolution of the formation errors $\delta_{12}, \delta_{13}, \delta_{14}$ is depicted in Fig. 7, exponentially decaying to zero, as prescribed by the LMIs.

Simulation 2 illustrates the effect of external forces with initial conditions in the desired formation shape. In particular, the following external accelerations appear at instant $t = 1$ s, different for each of the agents: $f_1 = [-2.4, -2]$ m/s$^2$, $f_2 = [-2, 0.6]$ m/s$^2$, $f_3 = [3, 2.6]$ m/s$^2$ and $f_4 = [2.2, -6]$ m/s$^2$. As external forces do exert

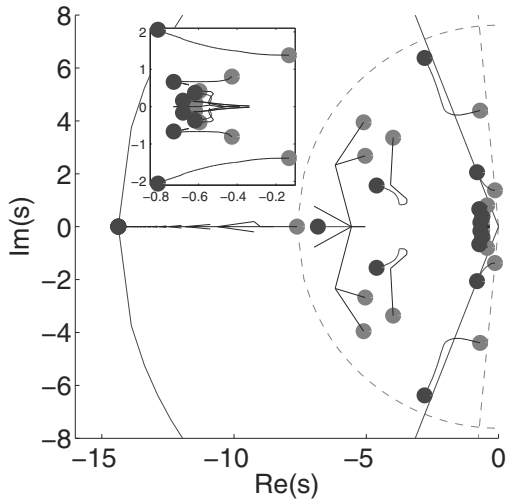



Fig. 4. Achieved regional pole constraints with the designed formation control. A zoomed version (top left) shows the rightward movements of the dominant poles (closer to the imaginary axis) as the bound for $\rho$ is made more stringent.

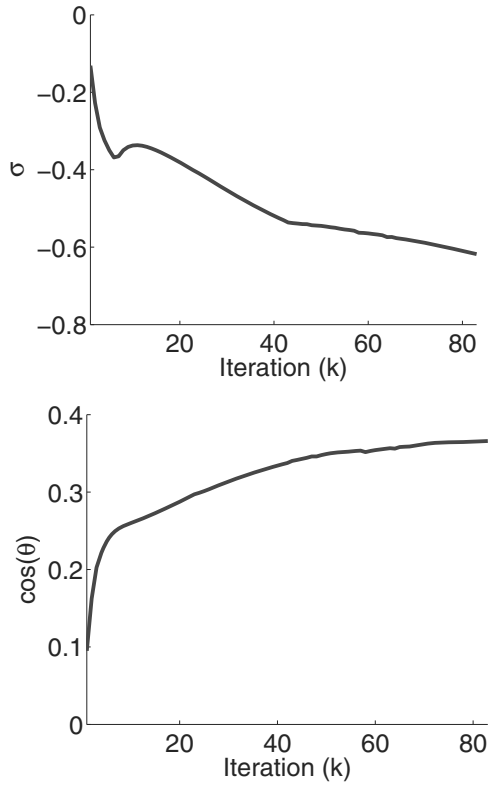



Fig. 5. Evolution of the real part of the dominant pole at each CCL iteration $k$ (top), evolution of the damping coefficient $cos(\theta)$ at each CCL iteration $k$ (bottom).

(virtual) accelerations over the steering point, a steering controller is needed; in this case,

$$G_{\text{steer}}(s) = \frac{10(s+0.2)(s+0.1)}{s(s+2)}$$

was chosen, for both horizontal and vertical degrees of freedom; this controller can be interpreted as a PID with a noise filter. The rotational degrees of freedom were kept constant, setting $\tau_F \equiv 0$ in (4), so the goal is keeping formation shape under a sort of uneven "wind" and, on top of that, the pilot has to steer the formation providing the necessary acceleration to counteract the effect of such wind over the steering point's position (steering controller has an integral action).

Results of this second simulation show that the external forces are steady-state rejected due to the integral action of both the formation control and steering strategies: the formation pattern is not broken after the transient, despite the existence of different external forces in each agent (see Fig. 8). The time evolution of the formation errors $\delta_{12}, \delta_{13}, \delta_{14}$ is depicted in the solid lines in Figs. 7 and 9. The dashed line represents the evolution of the steering point, with the reference position set to $(0, 0)$. Note that, as steering dynamics are the same in both the horizontal and vertical degrees of freedom, and forces are constant, the steering point follows a straight line trajectory, deviating from the initial position at the start of the simulation, but returning to it later on.

**Comparative analysis.** As most of the literature reviewed in Introduction pursues only consensus achievement (stability with no disturbances), settling time and low-frequency disturbance rejection are not part of their problem statement so performance is not guaranteed in such setups. For instance, the dashed black lines in Figs. 7 and 9 (labeled as $\delta_1^*$) depicts the modulus of the deviations (to avoid cluttering) using the observer-based technique from the work of Zhang and Chen (2017) where state-feedback was computed with pole-placement at $s = -2$ (as done in our initial controller here), and observer dynamics used the Riccati equation with $Q = R = I$ (see the cited work for details). Their resulting control is stable, of course, but its settling time is slower and there is a position error under disturbances, when compared with our CCL proposal (each deviation $\delta_{12}, \delta_{13}, \delta_{14}$ also depicted in the solid line style in Figs. 7 and 9 for comparative purposes).

## 8. Conclusions and perspectives

This paper presented a formation-control setup based on, first, decoupling the formation steering from the formation shape control, and second, building a minimal-realization for the formation shape problem (consensus) and proposing a pole-region placement

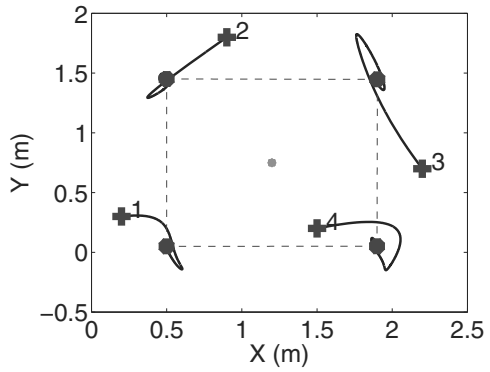



Fig. 6. Trajectories followed by each agent during formation control (Simulation 1).

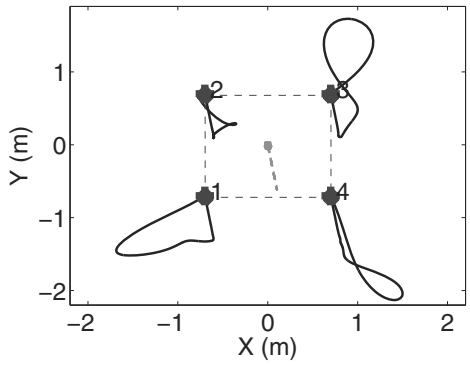



Fig. 8. Trajectories followed by each agent during formation control (Simulation 2).

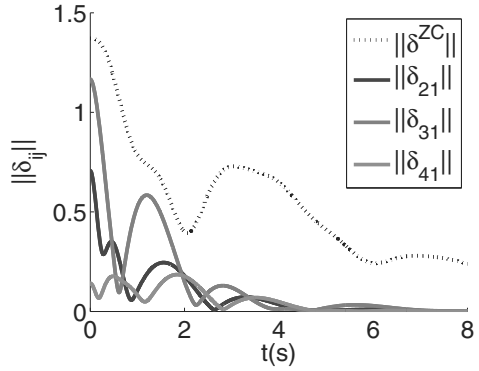



Fig. 7. Comparison of the formation control errors $||\delta_{12}||$, $||\delta_{13}||$ and $||\delta_{14}||$ with the formation control error $||\delta^{ZC}|| = ||\left(\delta_{12}^{ZC}, \delta_{13}^{ZC}, \delta_{14}^{ZC}\right)||$ obtained by Zhang and Chen (2017) (Simulation 1).

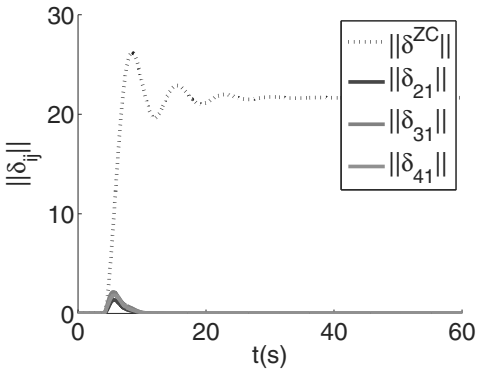



Fig. 9. Comparison of the formation control errors $||\delta_{12}||$, $||\delta_{13}||$ and $||\delta_{14}||$ with the formation control error $||\delta_1^{ZC}|| = ||\left(\delta_{12}^{ZC}, \delta_{13}^{ZC}, \delta_{14}^{ZC}\right)||$ obtained by Zhang and Chen (2017) (Simulation 2).

specification; a solution to the latter problem is obtained via the CCL algorithm with progressively more stringent specifications. The steering control, independent of formation control, was addressed via a virtual structure approach. In this way, a complete methodology was built for the output-feedback control of formations of multiple agents.

Numerical simulations illustrate the proposal's performance. Nevertheless, optimization with fixed-structure controllers is a nonconvex problem so local minima might be attained; also, as discussed in Introduction, we did not take into account the possibility of inter-agent communication (maybe with delays), or a detailed analysis of disturbance rejection. However, the use of matrix inequalities and CCL algorithms opens up more performance/robustness possibilities ($\mathcal{H}_\infty$, $\mu$-synthesis, Lyapunov–Krasovskii analysis) that will be explored in future work, from the perspective of providing a complete methodology ready for practical applications.

## Acknowledgment

The first author is grateful for the financial support via the grant GVA/2021/082 from Generalitat Valenciana. Part of the authors' research activity in related topics is funded via the grant PID2020-116585GB-I00 through MCIN/AEI/10.13039/501100011033 and by the European Union.

## References

Bai, H. and Wen, J.T. (2010). Cooperative load transport: A formation-control perspective, *IEEE Transactions on Robotics* **26**(4): 742–750.

Bechlioulis, C.P., Giagkas, F., Karras, G.C. and Kyriakopoulos, K.J. (2019). Robust formation control for multiple underwater vehicles, *Frontiers in Robotics and AI* **6**(2019): 90.

Dehghani, M.A. and Menhaj, M.B. (2016). Communication free leader–follower formation control of unmanned

aircraft systems, *Robotics and Autonomous Systems* **80**(2016): 69–75.

Dong, X., Yu, B., Shi, Z. and Zhong, Y. (2014). Time-varying formation control for unmanned aerial vehicles: Theories and applications, *IEEE Transactions on Control Systems Technology* **23**(1): 340–348.

El Ghaoui, L., Oustry, F. and AitRami, M. (1997). A cone complementarity linearization algorithm for static output-feedback and related problems, *IEEE Transactions on Automatic Control* **42**(8): 1171–1176.

Farrera, B., López-Estrada, F.-R., Chadli, M., Valencia-Palomo, G. and Gómez-Peñate, S. (2020). Distributed fault estimation of multi-agent systems using a proportional-integral observer: A leader-following application, *International Journal of Applied Mathematics and Computer Science* **30**(3): 551–560, DOI: 10.34768/amcs-2020-0040.

González, A., Aranda, M., López-Nicolás, G. and Sagüés, C. (2019). Robust stability analysis of formation control in local frames under time-varying delays and actuator faults, *Journal of the Franklin Institute* **356**(2): 1131–1153.

González-Sierra, J., Dzul, A. and Martínez, E. (2022). Formation control of distance and orientation based-model of an omnidirectional robot and a quadrotor UAV, *Robotics and Autonomous Systems* **147**(2022): 103921.

He, S., Wang, M., Dai, S.-L. and Luo, F. (2018). Leader–follower formation control of USVs with prescribed performance and collision avoidance, *IEEE Transactions on Industrial Informatics* **15**(1): 572–581.

Kamel, M.A. and Zhang, Y. (2015). Decentralized leader–follower formation control with obstacle avoidance of multiple unicycle mobile robots, *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Halifax, Canada*, pp. 406–411.

Lee, G. and Chwa, D. (2018). Decentralized behavior–based formation control of multiple robots considering obstacle avoidance, *Intelligent Service Robotics* **11**(1): 127–138.

Li, Z., Liu, H.H., Zhu, B. and Gao, H. (2015). Robust second-order consensus tracking of multiple 3-DOF laboratory helicopters via output feedback, *IEEE/ASME Transactions on Mechatronics* **20**(5): 2538–2549.

Oh, K.-K., Park, M.-C. and Ahn, H.-S. (2015). A survey of multi-agent formation control, *Automatica* **53**(2015): 424–440.

Olfati-Saber, R. and Murray, R. (2004). Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on Automatic Control* **49**(9): 1520–1533.

Peaucelle, D., Arzelier, D., Bachelier, O. and Bernussou, J. (2000). A new robust D-stability condition for real convex polytopic uncertainty, *Systems and Control Letters* **40**(1): 21–30.

Peng, C., Zhang, A. and Li, J. (2021). Neuro-adaptive cooperative control for high-order nonlinear multi-agent systems with uncertainties, *International Journal of Applied Mathematics and Computer Science* **31**(4): 635–645, DOI: 10.34768/amcs-2021-0044.

Rahimi, R., Abdollahi, F. and Naqshi, K. (2014). Time-varying formation control of a collaborative heterogeneous multi agent system, *Robotics and Autonomous Systems* **62**(12): 1799–1805.

Ren, W. and Atkins, E. (2005). Second-order consensus protocols in multiple vehicle systems with local interactions, *AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, USA*, p. 6238.

Ren, W. and Beard, R.W. (2004). Decentralized scheme for spacecraft formation flying via the virtual structure approach, *Journal of Guidance, Control, and Dynamics* **27**(1): 73–82.

Ren, W. and Beard, R.W. (2008). *Distributed Consensus in Multi-Vehicle Cooperative Control*, Springer, London.

Ren, W. and Sorensen, N. (2008). Distributed coordination architecture for multi-robot formation control, *Robotics and Autonomous Systems* **56**(4): 324–333.

Tian, B., Lu, H., Zuo, Z. and Yang, W. (2018). Fixed-time leader–follower output feedback consensus for second-order multiagent systems, *IEEE Transactions on Cybernetics* **49**(4): 1545–1550.

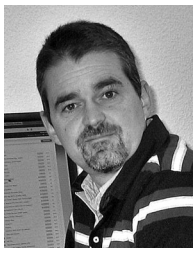
Wen, C., Liu, F., Song, Q. and Feng, X. (2016). Observer-based consensus of second-order multi-agent systems without velocity measurements, *Neurocomputing* **179**(2016): 298–306.

Zhai, G., Okuno, S., Imae, J. and Kobayashi, T. (2009). A matrix inequality based design method for consensus problems in multi-agent systems, *International Journal of Applied Mathematics and Computer Science* **19**(4): 639–646, DOI: 10.2478/v10006-009-0051-1.

Zhang, H. and Chen, J. (2017). Bipartite consensus of multi-agent systems over signed graphs: State feedback and output feedback control approaches, *International Journal of Robust and Nonlinear Control* **27**(1): 3–14.

Zou, Y., Zhou, Z., Dong, X. and Meng, Z. (2018). Distributed formation control for multiple vertical takeoff and landing UAVs with switching topologies, *IEEE/ASME Transactions on Mechatronics* **23**(4): 1750–1761.

**Antonio González** received a telecommunications engineer degree in 2001 and a PhD degree in automation and industrial informatics in 2012 from Universitat Politécnica de València (UPV). He was a postdoctoral researcher at the Laboratory of Industrial and Human Automation Control, Mechanical Engineering and Computer Science, CNRS, UMR 8201, Valenciennes, France, from 2013 to 2014. Currently he works as an associate professor at the Department of Systems Engineering and Automation at Universitat Politècnica de València (Spain). His research interests are within the broad area of time delay systems, robust control, networked control systems, multirobot systems and process control applications.



**Antonio Sala** was born in València, Spain, in 1968. He received his MSc degree in electrical engineering and his PhD degree in control engineering from the Polytechnic University of Valencia (UPV), Spain, in 1993 and 1998, respectively. He has been a full professor (chair) in the Systems and Control Engineering Department, UPV, since 2009. He has been teaching for over 30 years a wide range of subjects in the area, such as linear systems theory, multivariable process control, and intelligent control. He has supervised 12 PhD theses, and taken part in research and mobility projects funded by local industries, government, and the European community. He has published over 80 journal papers and 150 conference ones. Professor Sala has served as an associate editor of *IEEE Transactions on Fuzzy Systems, Fuzzy Sets and Systems* and *Revista Iberaamericana Automatica e Informatica Industrial*.

**Leopoldo Armesto** received his BSc degree in electronic engineering, his MSc degree in control systems engineering, and his PhD degree in automation and industrial computer science from Universitat Politècnica de València (UPV), Spain, in 1998, 2001, and 2005, respectively, where he had also held a PhD scholarship for three years at the Department of Systems and Control Engineering (DISA). Since 2004 and 2016, respectively, he has been an assistant professor and a senior lecturer at DISA, UPV. He is also a member of the Robotics and Automation Research Group of the Design and Manufacturing Institute at UPV. He is the author or a coauthor of several journal and conference papers. His current research interests are mobile robotics, optimal control, educational robotics and reinforcement learning.

Received: 4 February 2022
Revised: 25 March 2022
Accepted: 6 April 2022