

Formation Control via Rotation Symmetry Constraints

American Control Conference
New Orleans, LA

Zamir Martinez and **Daniel Zelazo**

The Stephen B. Klein Faculty of Aerospace Engineering, Technion

May 29, 2026



CONNECT LAB
Cooperative Networks and Controls



THE STEPHEN B. KLEIN
FACULTY OF
AEROSPACE ENGINEERING



TECHNION
Israel Institute
of Technology



Introduction



Formation control arises naturally in a wide range of multi-agent applications

- ▶ UAV Swarms: mapping and surveillance
- ▶ Satellite constellations: uniform spacing for coverage
- ▶ Multi-robot systems: distributed sensing and transportation



How can we define formations?

standard approach: specify geometric constraints between agents

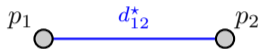


A formation can be described by desired distances between selected pairs of agents.



How can we define formations?

standard approach: specify geometric constraints between agents

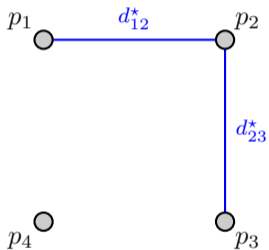


As we add distance constraints, the set of feasible configurations becomes smaller.



How can we define formations?

standard approach: specify geometric constraints between agents

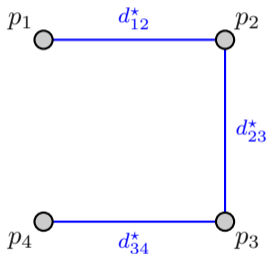


As we add distance constraints, the set of feasible configurations becomes smaller.



How can we define formations?

standard approach: specify geometric constraints between agents

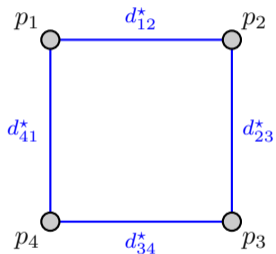


As we add distance constraints, the set of feasible configurations becomes smaller.



How can we define formations?

standard approach: specify geometric constraints between agents

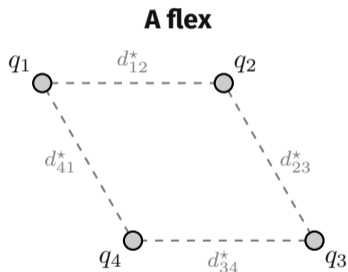
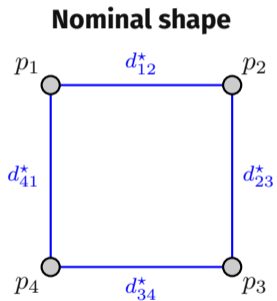


As we add distance constraints, the set of feasible configurations becomes smaller.



How can we define formations?

standard approach: specify geometric constraints between agents

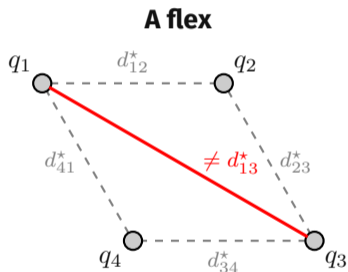
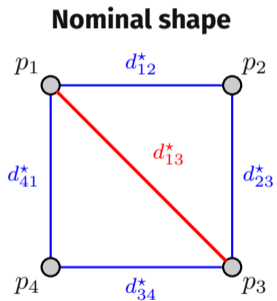


But too few constraints may still allow **flexes**: different shapes satisfying the same distances.



How can we define formations?

standard approach: specify geometric constraints between agents

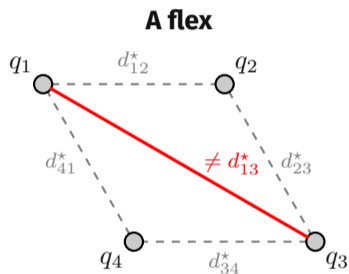
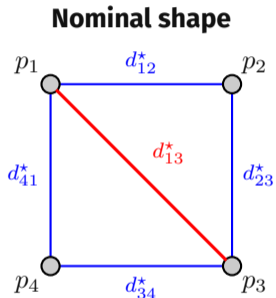


Adding an independent constraint can eliminate the flex.



How can we define formations?

standard approach: specify geometric constraints between agents



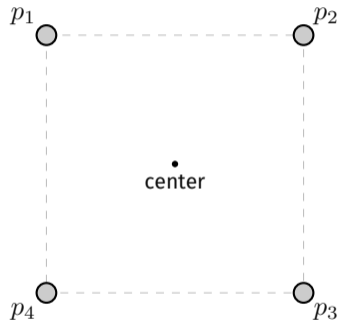
Rigidity theory formalizes when enough independent constraints have been added. In \mathbb{R}^2 , a minimally rigid formation of n agents requires

$$m = 2n - 3 \text{ distance constraints.}$$



How can we define formations?

Another approach: specify symmetry constraints

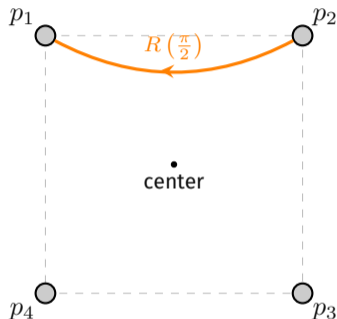


Many target formations contain structure that is not captured efficiently by pairwise distances.



How can we define formations?

Another approach: specify symmetry constraints



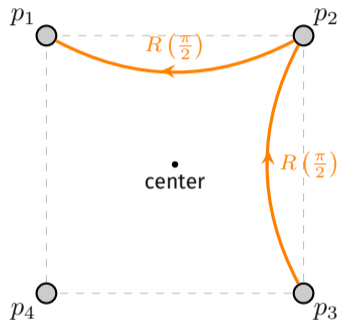
Here the formation is described by repeated rotations:

$$p_i = \underbrace{R\left(\frac{\pi}{2}\right)}_{\text{rotation matrix}} p_{i+1}.$$



How can we define formations?

Another approach: specify symmetry constraints



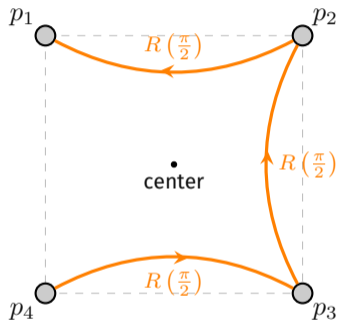
Here the formation is described by repeated rotations:

$$p_i = \underbrace{R\left(\frac{\pi}{2}\right)}_{\text{rotation matrix}} p_{i+1}.$$



How can we define formations?

Another approach: specify symmetry constraints



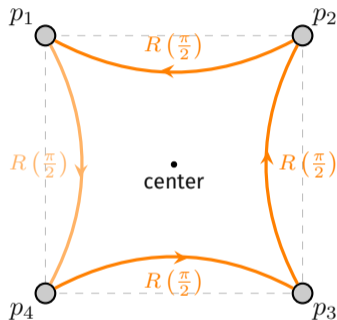
Here the formation is described by repeated rotations:

$$p_i = \underbrace{R\left(\frac{\pi}{2}\right)}_{\text{rotation matrix}} p_{i+1}.$$



How can we define formations?

Another approach: specify symmetry constraints



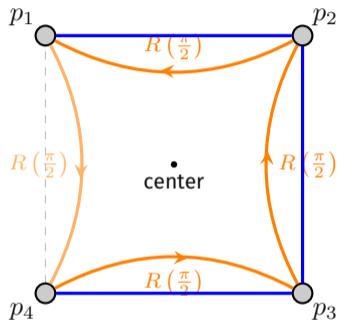
Here the formation is described by repeated rotations:

$$p_i = \underbrace{R\left(\frac{\pi}{2}\right)}_{\text{rotation matrix}} p_{i+1}.$$



How can we define formations?

Another approach: specify symmetry constraints



only a spanning tree of symmetry constraints

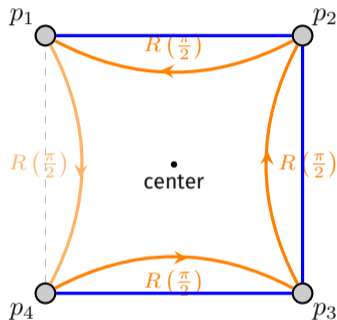
For cyclic formations, the target shape can be encoded using only

$$m = n - 1$$



How can we define formations?

Another approach: specify symmetry constraints



only a spanning tree of symmetry constraints

Can symmetry constraints alone be used to define a formation control strategy?



The Formation Control Problem

Formation Control Objective

For an ensemble of n agents with dynamics

$$\dot{p}_i = u_i,$$

with $p_i(t) \in \mathbb{R}^d$, information exchange graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and formation constraint function $F : \mathbb{R}^{nd} \rightarrow \mathbb{R}^M$, design a distributed control law for each agent $i \in \{1, \dots, n\}$ such that the set

$$\mathcal{F}(p) = \{p \in \mathbb{R}^{nd} \mid F(p) = F(\mathbf{p}^*)\},$$

is asymptotically stable.



The Formation Control Problem

Formation Control Objective

For an ensemble of n agents with dynamics

$$\dot{p}_i = u_i,$$

with $p_i(t) \in \mathbb{R}^d$, information exchange graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and formation constraint function $F : \mathbb{R}^{nd} \rightarrow \mathbb{R}^M$, design a distributed control law for each agent $i \in \{1, \dots, n\}$ such that the set

$$\mathcal{F}(p) = \{p \in \mathbb{R}^{nd} \mid F(p) = F(\mathbf{p}^*)\},$$

is asymptotically stable.

F will encode **inter-agent rotational symmetries** of the target configuration



The Formation Control Problem

Formation Control Objective

For an ensemble of n agents with dynamics

$$\dot{p}_i = u_i,$$

with $p_i(t) \in \mathbb{R}^d$, information exchange graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and formation constraint function $F : \mathbb{R}^{nd} \rightarrow \mathbb{R}^M$, design a distributed control law for each agent $i \in \{1, \dots, n\}$ such that the set

$$\mathcal{F}(p) = \{p \in \mathbb{R}^{nd} \mid F(p) = F(\mathbf{p}^*)\},$$

is asymptotically stable.

We need a mathematical language to talk about symmetries



Symmetry in Graphs

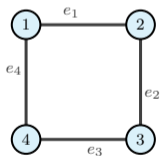
Automorphisms encode graph **symmetries**

Graph Automorphism

An **automorphism** of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a permutation $\psi : \mathcal{V} \rightarrow \mathcal{V}$ of its vertex set such that

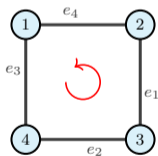
$$\{v_i, v_j\} \in \mathcal{E} \Leftrightarrow \{\psi(v_i), \psi(v_j)\} \in \mathcal{E}$$

Example: Cycle graph C_4



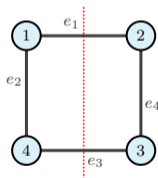
Identity:

$$\text{Id} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$



90° rotation:

$$\psi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}$$



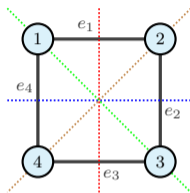
reflection:

$$\psi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$



Symmetry in Graphs

- ▶ Additional permutations can be found for the given graph considering all possible reflections and rotations
- ▶ The set of all automorphisms of \mathcal{G} form a *group* - $\text{Aut}(\mathcal{G})$
 - $\text{Aut}(\mathcal{G}) = \{\text{Id}, \psi_1, \psi_2, \dots\}$
- ▶ For any subgroup $\Gamma \subseteq \text{Aut}(\mathcal{G})$, we say that \mathcal{G} is Γ -*symmetric*, which define specific symmetries in \mathcal{G}





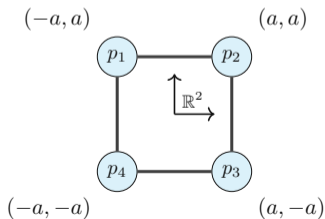
Symmetry in Frameworks

Definition

A framework (\mathcal{G}, p) in \mathbb{R}^d is called $\tau(\Gamma)$ -symmetric if

$$\tau(\gamma)(p_i) = p_{\gamma(i)} \quad \text{for all } \gamma \in \Gamma \quad \text{and all } i \in \mathcal{V}$$

The permutation symmetries of the graph are matched with geometric symmetries of the formation through a map $\tau : \Gamma \rightarrow O(d)$.





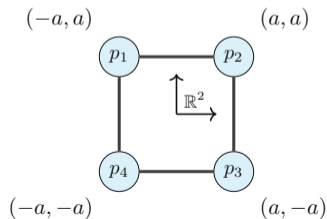
Symmetry in Frameworks

Definition

A framework (\mathcal{G}, p) in \mathbb{R}^d is called $\tau(\Gamma)$ -symmetric if

$$\tau(\gamma)(p_i) = p_{\gamma(i)} \quad \text{for all } \gamma \in \Gamma \quad \text{and all } i \in \mathcal{V}$$

The permutation symmetries of the graph are matched with geometric symmetries of the formation through a map $\tau : \Gamma \rightarrow O(d)$.



- Consider $\Gamma = \{\text{Id}, \psi_1, \dots\}$ (All rotations of C_4)

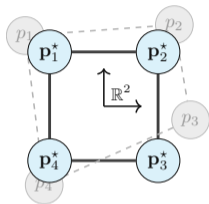
$$\blacktriangleright \tau(\psi_1)p_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} = \begin{bmatrix} -a \\ a \end{bmatrix} = p_1$$

$$\blacktriangleright \tau(\psi_1^2)p_3 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ -a \end{bmatrix} = \begin{bmatrix} -a \\ a \end{bmatrix} = p_1$$



Symmetry in Frameworks

For any cycle graph C_n :

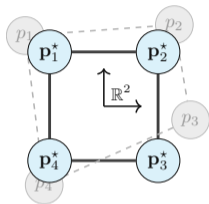


- ▶ A $\tau(\Gamma)$ -symmetric framework is called **C_n -symmetric** when the configuration's rotational symmetries match those of C_n



Symmetry in Frameworks

For any cycle graph C_n :

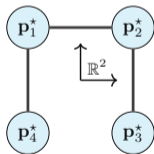
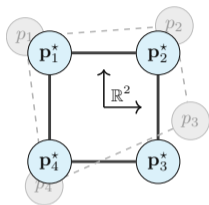


- ▶ A $\tau(\Gamma)$ -symmetric framework is called **C_n -symmetric** when the configuration's rotational symmetries match those of C_n
- ▶ In cycle notation:
$$\Gamma = \{\text{Id}, \psi_1, \psi_1^2, \dots, \psi_1^{n-1}\} \subseteq \text{Aut}(C_n)$$



Symmetry in Frameworks

For any cycle graph C_n :



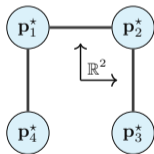
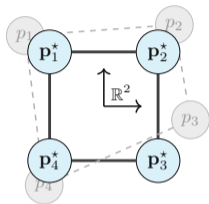
$$\mathbf{p}_4^* = \tau(\psi_1^3) \mathbf{p}_3^*$$

- ▶ A $\tau(\Gamma)$ -symmetric framework is called **C_n -symmetric** when the configuration's rotational symmetries match those of C_n
- ▶ In cycle notation:
 $\Gamma = \{\text{Id}, \psi_1, \psi_1^2, \dots, \psi_1^{n-1}\} \subseteq \text{Aut}(C_n)$
- ▶ A single representative agent recovers the full formation via successive applications of $\tau(\psi_1)$



Symmetry in Frameworks

For any cycle graph C_n :



$$p_4^* = \tau(\psi_1^3) p_3^*$$

- ▶ A $\tau(\Gamma)$ -symmetric framework is called **C_n -symmetric** when the configuration's rotational symmetries match those of C_n
- ▶ In cycle notation:
 $\Gamma = \{\text{Id}, \psi_1, \psi_1^2, \dots, \psi_1^{n-1}\} \subseteq \text{Aut}(C_n)$
- ▶ A single representative agent recovers the full formation via successive applications of $\tau(\psi_1)$

An *interaction graph* $\mathcal{G}_I = (\mathcal{V}, \mathcal{E}_I)$, chosen as a **spanning tree** of C_n is sufficient to define the target configuration



A Gradient Approach

Formation Control Objective

Design $u_i(t)$ such that for every edge $ij \in \mathcal{E}_I$,

$$\lim_{t \rightarrow \infty} \|p_i(t) - \tau(\gamma_{ji}) p_j(t)\| = 0$$



A Gradient Approach

Formation Control Objective

Design $u_i(t)$ such that for every edge $ij \in \mathcal{E}_I$,

$$\lim_{t \rightarrow \infty} \|p_i(t) - \tau(\gamma_{ji}) p_j(t)\| = 0$$

Define a symmetry-forcing potential function:

$$F(p(t)) = \frac{1}{2} \sum_{ij \in \mathcal{E}_I} \|p_i(t) - \tau(\gamma_{ji}) p_j(t)\|^2.$$

Gradient descent $u(t) = -\nabla F(p(t))$ defines the control law:

$$\dot{p}_i(t) = \sum_{ij \in \mathcal{E}_I} (\tau(\gamma_{ji}) p_j(t) - p_i(t))$$



Symmetry Based Control Law

A Linear Control Law:

$$\dot{p}(t) = -Qp(t)$$

$Q \in \mathbb{R}^{2n \times 2n}$ is a **symmetry-constrained matrix-weighted Laplacian** of \mathcal{G}_I , with block entries

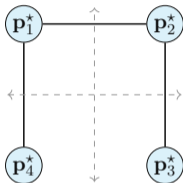
$$[Q]_{ij} = \begin{cases} d(i)I_2, & i = j, i \in \mathcal{V} \\ -\tau(\gamma_{ji}), & ij \in \mathcal{E}_I \\ 0, & \text{o.w.} \end{cases}$$

- ▶ $d(i)$ denotes the degree of node i



Symmetry Based Control Law

Example



$$Q = \begin{bmatrix} I_2 & -R(\frac{\pi}{2}) & 0 & 0 \\ -R(\frac{\pi}{2})^T & 2I_2 & -R(\frac{\pi}{2}) & 0 \\ 0 & -R(\frac{\pi}{2})^T & 2I_2 & -R(\frac{\pi}{2}) \\ 0 & 0 & -R(\frac{\pi}{2})^T & I_2 \end{bmatrix}$$

$$Q\mathbf{p}^* = 0$$

- in fact, any rotation, scaling, or translation of \mathbf{p}^* is also in kernel



Example

$$Q = \begin{bmatrix} I_2 & -R(\frac{\pi}{2}) & 0 & 0 \\ -R(\frac{\pi}{2})^T & 2I_2 & -R(\frac{\pi}{2}) & 0 \\ 0 & -R(\frac{\pi}{2})^T & 2I_2 & -R(\frac{\pi}{2}) \\ 0 & 0 & -R(\frac{\pi}{2})^T & I_2 \end{bmatrix}$$

Properties of Q

- ▶ Positive semi-definite
- ▶ Its nullspace corresponds to the trivial motions of the formation — **translations, uniform scalings, and rotations**



Symmetry Based Control Law

let

$$\mathcal{F} = \{p \in \mathbb{R}^{2n} \mid \tau(\gamma_{ji})p_i = p_{\gamma_{ji}(i)}, \quad \forall ij \in \mathcal{E}, i \in \mathcal{V}\},$$

be the set of \mathcal{C}_n -symmetric configurations

Theorem

For any initial condition $p(0) \in \mathbb{R}^{2n}$, the control law $\dot{p}(t) = -Qp(t)$ renders \mathcal{F} exponentially stable. Moreover, the agents converge to

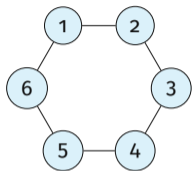
$$\lim_{t \rightarrow \infty} p_i(t) = \frac{1}{n} S_i \sum_{k=1}^n S_k^T p_k(0).$$

where S_i is the product of rotation matrices along the spanning tree path from node 1 to node i , and $p(\infty)$ is the orthogonal projection of $p(0)$ onto \mathcal{F} .

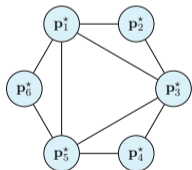


Example

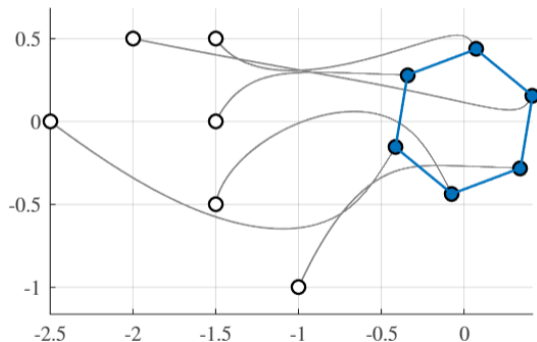
Setup: $n = 6$ agents, tasked with attaining a \mathcal{C}_6 -symmetric formation (hexagon)



Underlying Graph \mathcal{C}_6

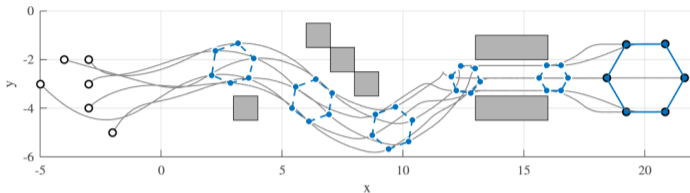


A distance based approach
requires 9 edges





Formation Maneuvering



- ▶ **Formation maneuvering** aims to satisfy the formation control objective while simultaneously moving the formation through space as a rigid body
- ▶ Secondary objective:

$$\lim_{t \rightarrow \infty} \|\dot{p}_i(t) - v_i(t)\| = 0$$

where $v_i \in \mathbb{R}^d$ is the desired rigid body velocity for each agent

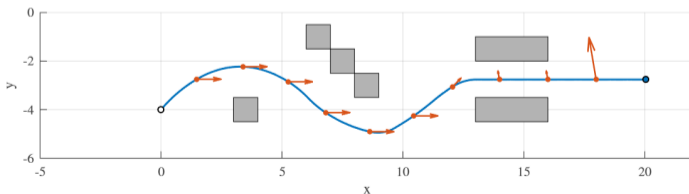


Formation Maneuvering

Assumption

All agents have access to a virtual trajectory predefined by a

- i) translation $r(t) \in \mathbb{R}^2$ with $\dot{r}(t) = v(t)$;
- ii) rotation $\mathcal{R}(t) \in SO(2)$ with $\dot{\mathcal{R}}(t) = \Omega(t)\mathcal{R}(t)$ where $\Omega(t) = \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix}$, and $\omega(t)$ is the desired angular velocity of the formation;
- iii) scale factor $s(t) \in \mathbb{R}^+$, with $\dot{s}(t) = \alpha(t)s(t)$, $\alpha(t) \in \mathbb{R}$.





Formation Maneuvering

Let $c_i(t) = p_i(t) - r(t)$ denote the position of agent i relative to the reference trajectory, and define the set of all shifted \mathcal{C}_n -symmetric configurations as

$$\mathcal{F}_c = \{p \in \mathbb{R}^{2n} \mid \tau(\gamma_{ji})c_i = c_{\gamma_{ji}(i)}, \forall ji \in \mathcal{E}, i \in \mathcal{V}\}$$

We introduce the state $\zeta(t) \in \mathbb{R}^{2n}$ to be the states $p(t) \in \mathbb{R}^{2n}$ expressed in a frame moving along the virtual trajectory,

$$\zeta(t) = \frac{1}{s(t)} (I_n \otimes \mathcal{R}(t)^T) c(t) \in \mathbb{R}^{2n}.$$



Formation Maneuvering

Let $c_i(t) = p_i(t) - r(t)$ denote the position of agent i relative to the reference trajectory, and define the set of all shifted \mathcal{C}_n -symmetric configurations as

$$\mathcal{F}_c = \{p \in \mathbb{R}^{2n} \mid \tau(\gamma_{ji})c_i = c_{\gamma_{ji}(i)}, \forall ji \in \mathcal{E}, i \in \mathcal{V}\}$$

We introduce the state $\zeta(t) \in \mathbb{R}^{2n}$ to be the states $p(t) \in \mathbb{R}^{2n}$ expressed in a frame moving along the virtual trajectory,

$$\zeta(t) = \frac{1}{s(t)} (I_n \otimes \mathcal{R}(t)^T) c(t) \in \mathbb{R}^{2n}.$$

Theorem

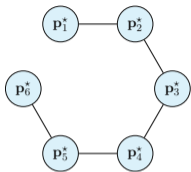
For any initial condition $p(0) \in \mathbb{R}^{2n}$, the augmented Control Law:

$$u(t) = -Qc(t) + \mathbf{1}_n \otimes v(t) + (I_n \otimes \Omega(t) + \alpha(t))c(t)$$

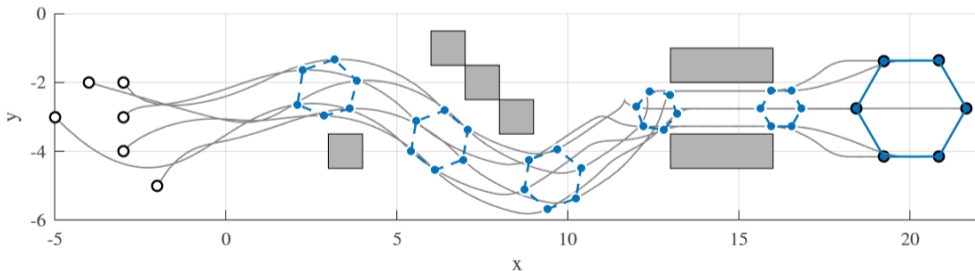
renders \mathcal{F}_c exponentially stable.



Example

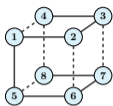
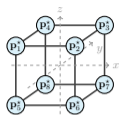


Setup: $n = 6$ agents tasked with attaining a C_6 -symmetric formation (hexagon) while maneuvering along a predefined virtual trajectory

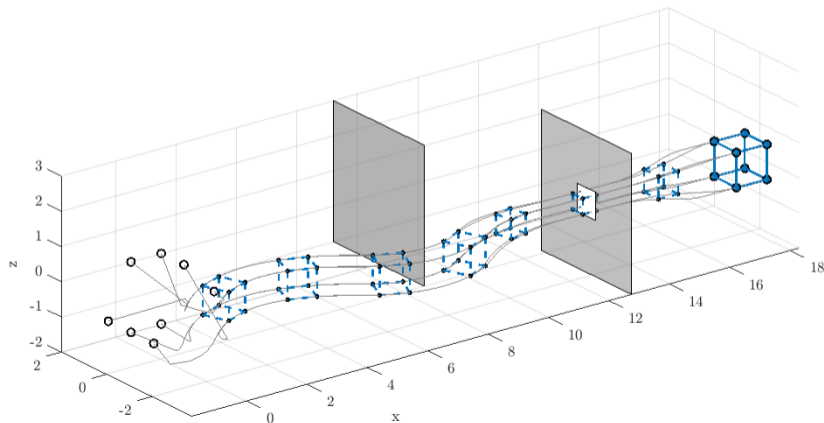




Extension to \mathbb{R}^3



Setup: $n = 8$ agents tasked with attaining a cube formation while maneuvering along a predefined virtual trajectory





Summary

- Cyclic formations can be achieved using only rotation symmetry constraints
- The proposed method requires only $n - 1$ edges (minimal connectivity), fewer than "classical" approaches

Future Work

- ▶ Extension to broader point-group symmetries in \mathbb{R}^3
- ▶ Explore leader-follower architectures enabling fully distributed agreement on time-varying virtual trajectories
- ▶ Incorporate additional geometric constraints alongside symmetry to enable a broader class of target formations



Thank you

- Z. Martinez and D. Zelazo, “Formation Control via Rotation Symmetry Constraints,” in American Control Conference, New Orleans, LA, USA, May 2026.
- Z. Martinez and D. Zelazo, “Symmetry-Based Formation Control on Cycle Graphs Using Dihedral Point Groups,” in IFAC World Congress, Busan, South Korea, Aug. 2026.