Networked Robotics Systems, Cooperative Control and Swarming (ROB-GY 6333 Section A)

- **Today's lecture:**
    - Nonlinear Formation Control

Standard Recipe

- This is the power of algebraic ___________!
  - Algebraic graph theory
  - Algebraic geometry

- Laplacian → Consensus Protocol
- ???????? → Formation Control

- We will meet a lot of old friends

What might a decentralized ***linear*** formation control law look like?

- Assume a 2D formation defined by a framework (graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$) and function $p(\mathcal{V})$ to locate vertices

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = -k \sum_{j \in \mathcal{N}_i} \left(\begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} - \begin{bmatrix} x_j(t) \\ y_j(t) \end{bmatrix} \right) - \left(\begin{bmatrix} p_{x,i} \\ p_{x,i} \end{bmatrix} - \begin{bmatrix} p_{y,j} \\ p_{y,j} \end{bmatrix} \right)$$

- Is it local/decentralized?
- Is it stable?
- What is the behavior like?

Formation that can only translate (no rotation)

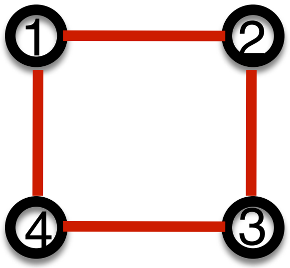
- Assume a formation defined by a framework with graph $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}_f)$ and function $p(\mathcal{V})$ to locate vertices.
- Assume that agents are connected through a communication graph $\mathcal{G}_{com} = (\mathcal{V}, \mathcal{E}_{com})$
- We want agents to be at the desired locations up to a common translation

$\dot{\mathbf{x}}_i = \mathbf{p}(v_i) + \boldsymbol{\tau}$, where the position of each agent $\mathbf{x}_i \in \mathbb{R}^d$

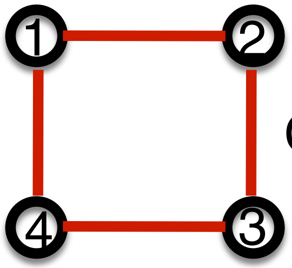
- Define variables $\tau_i(t) = x_i - p_i$.
  The formation is achieved when there is **consensus on translation** $\forall i, j,\ \tau_i = \tau_j$
- Results in control law

$$\boxed{\dot{\mathbf{x}}_i = -k \sum_{j \in \mathcal{N}_i} \left(\mathbf{x}_i(t) - \mathbf{x}_j(t)\right) - \left(\mathbf{p}_i - \mathbf{p}_j\right)}$$

- Convergence is guaranteed as long as $\mathcal{E}_f \subseteq \mathcal{E}_{com}$ and $\mathcal{G}_{com}$ is connected

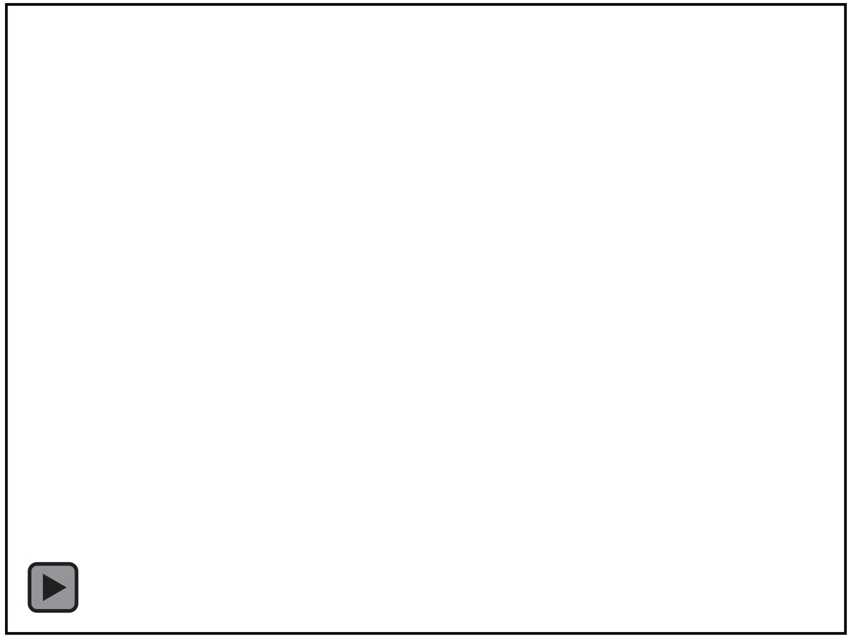


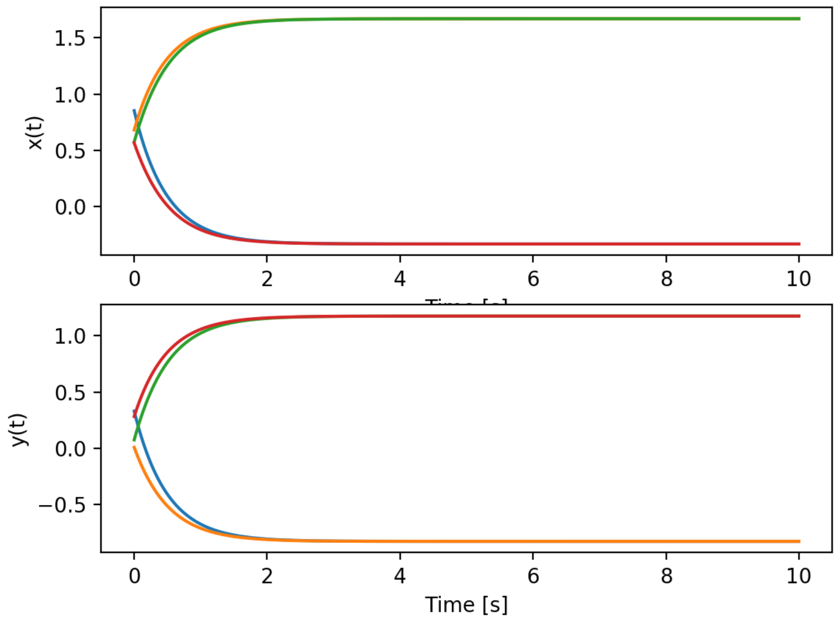
Desired formation

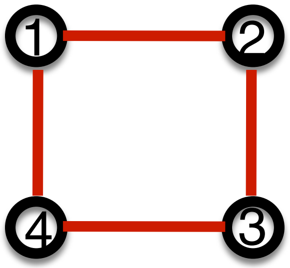


Communication graph

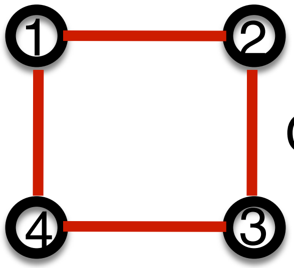
(gain) k = 1





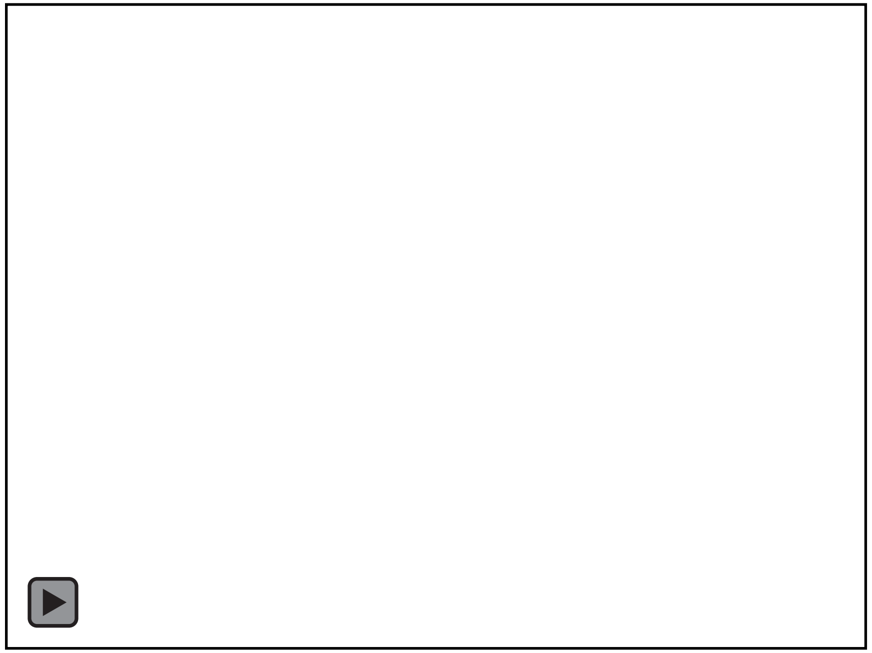


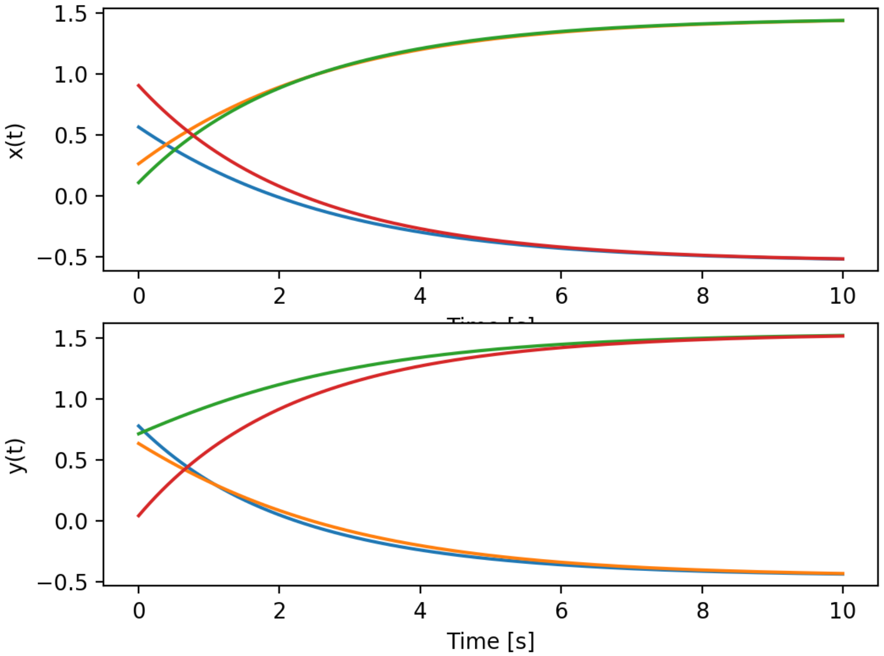
Desired formation

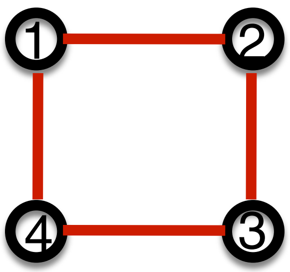


Communication graph

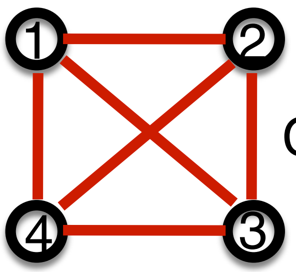
(gain) k = 0.2





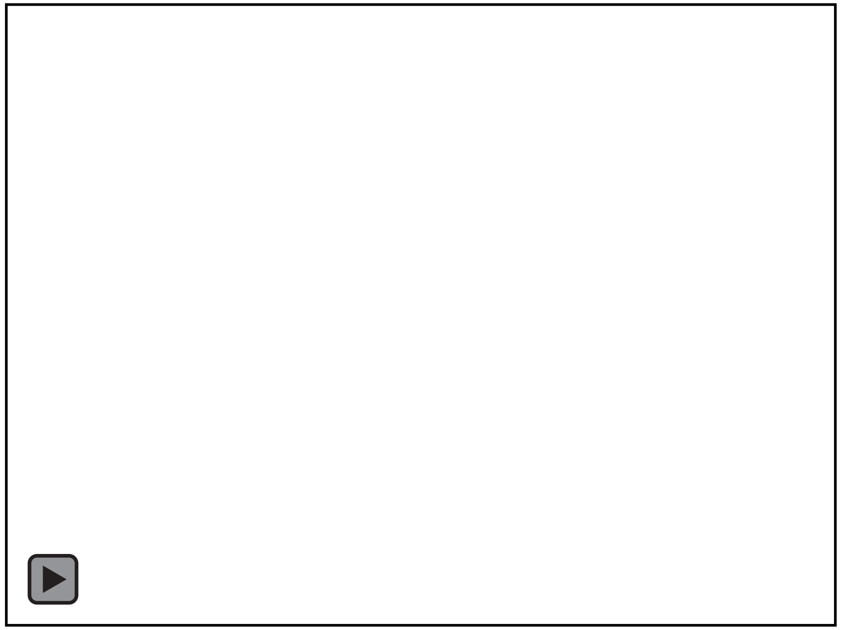


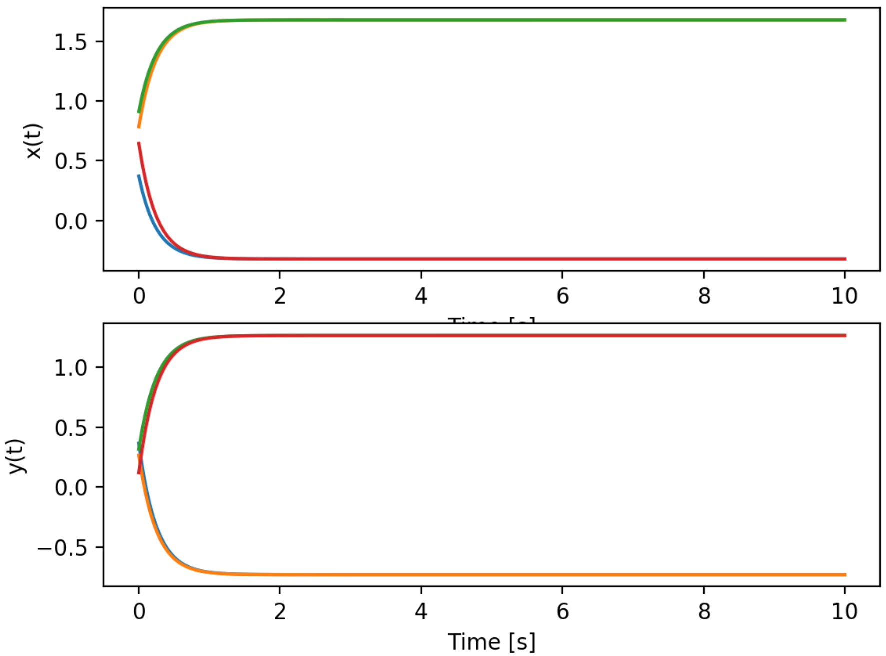
Desired formation



Communication graph

(gain) $k = 0.2$





Issues

- Needs the absolute position of all agents (e.g. using GPS)
- Does not allow rotations
- No guarantees that connectivity can be maintained at all times (especially with linear control)
- Quick fix when original connectivity graph is not good enough
    - First agreement protocol to get all agents to form a fully connected graph
    - Then apply formation control

Next level

- In rendezvous, the agents converge to something like a "center-of-mass"
- **Idea:** For formations without rotation, easy to define the "center-of-mass" motion, and then figure out the relative position of each agent relative to the "center-of-mass"
  - No need for absolute coordinates (GPS)
- As usual, first look for a simple linear control law

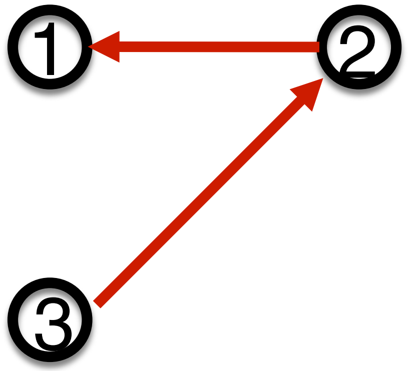
Formation control: relative state-based control

- Something closer to the original “center-of-mass” idea we started with:
  - Consider three agents and a fully connected formation, compute the relative states $z(t)$ and impose a reference z_{ref}

$$z(t) = \begin{bmatrix} x_1(t) - x_2(t) \\ x_2(t) - x_3(t) \end{bmatrix}$$

  - The **incidence matrix** returns here as the matrix that maps between the relative and actual state

$$z(t) = E^T x(t)$$



Formation control: relative state-based control

- Inspired by the Laplacian-based consensus protocol, you may be tempted to pick a control law of the form

$$\dot{\mathbf{x}} = -kL(\mathbf{x} - \mathbf{x}_{ref})$$

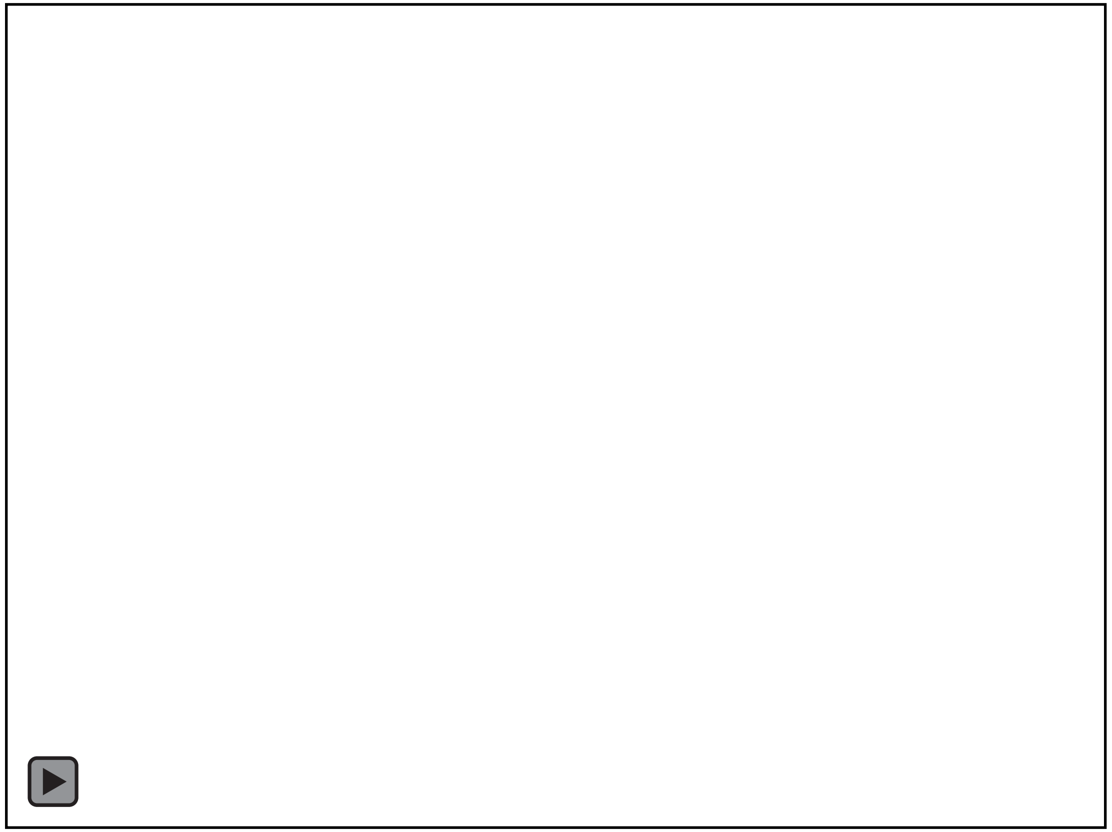
- However, $\mathbf{x}_{ref}$ doesn't really exist, because there is no absolute position we must converge. But if we play around with the idea and recall that $L = EE^T$

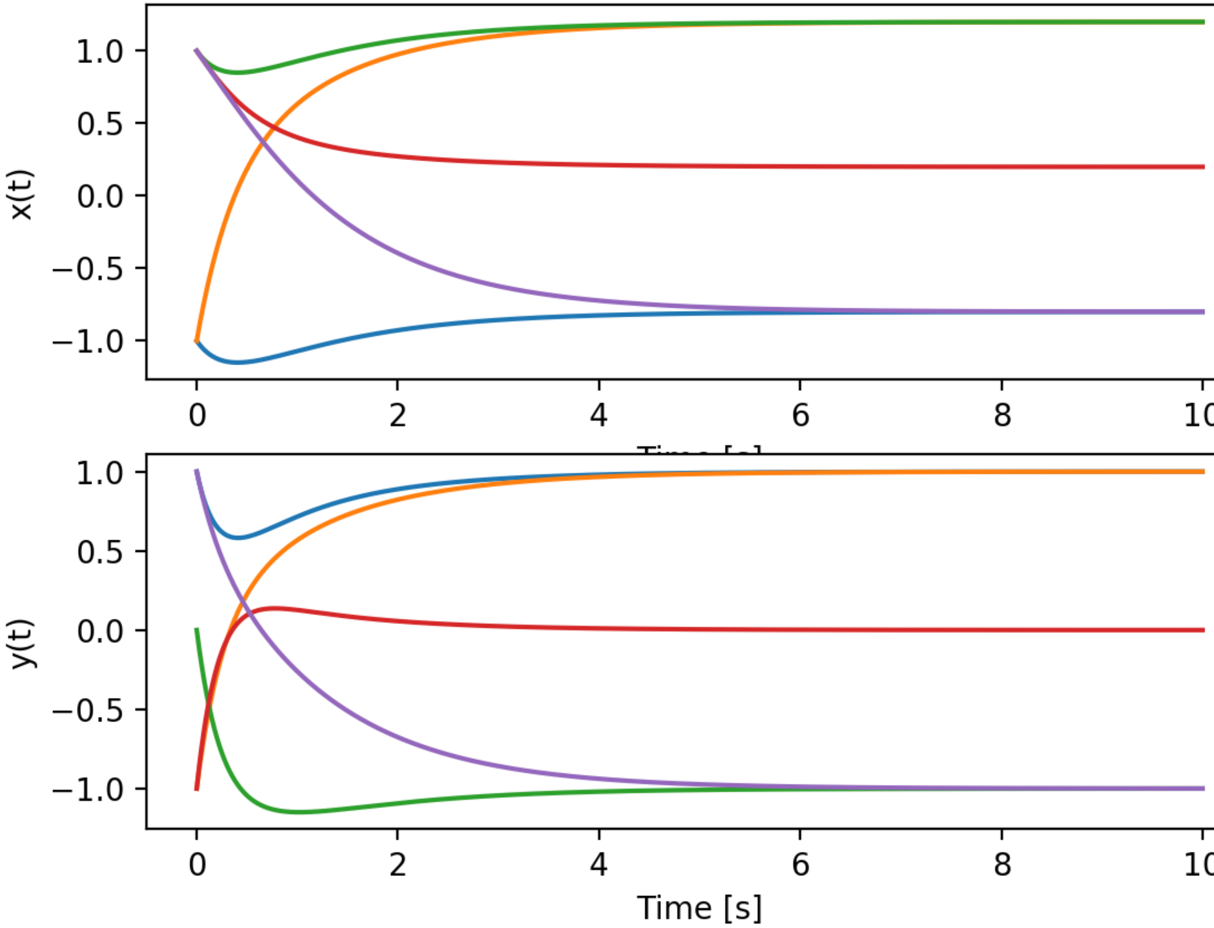
$$E\left(\mathbf{z}_{ref}\right) = E\left(E^T \mathbf{x}_{ref}\right) = L\mathbf{x}_{ref}$$

- You can obtain a local linear control law that only needs relative positions to converge onto $\mathbf{z}_{ref}$

$$\dot{\mathbf{x}} = -kL\mathbf{x} + kE\mathbf{z}_{ref}$$

Formation control: relative state-based control





Formation control: relative state-based control

- Previous slide: Single integrator dynamics (trivial dynamics, only control input)

$$\dot{\mathbf{x}} = \mathbf{u}$$

$$\mathbf{u} = -kL\mathbf{x} + kE\mathbf{z}_{ref}$$

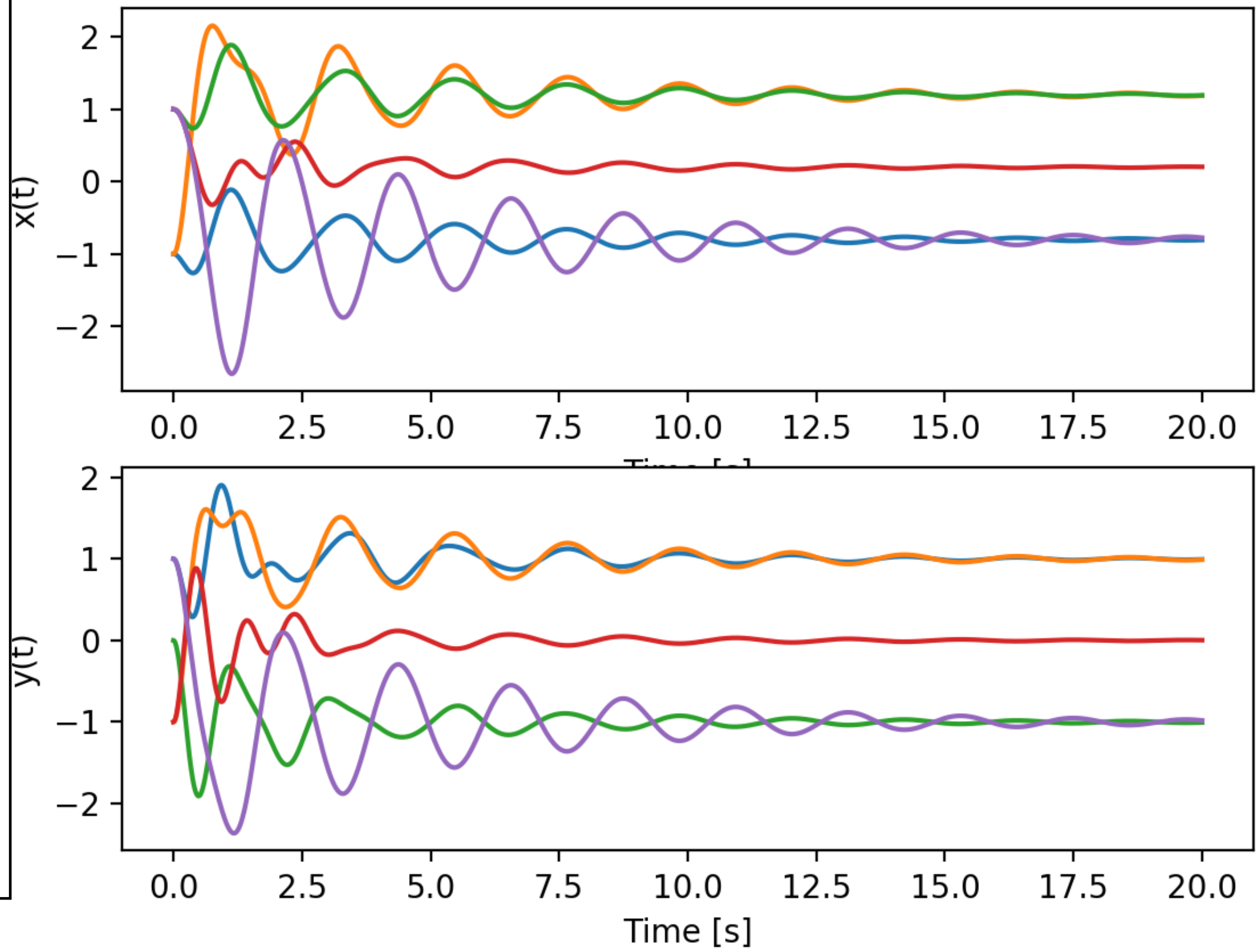
- Double integrator dynamics

$$\ddot{\mathbf{x}} = \mathbf{u}$$

$$\mathbf{u} = -kL\mathbf{x} + kE\mathbf{z}_{ref} - dL\dot{\mathbf{x}} + dE\dot{\mathbf{z}}_{ref}$$

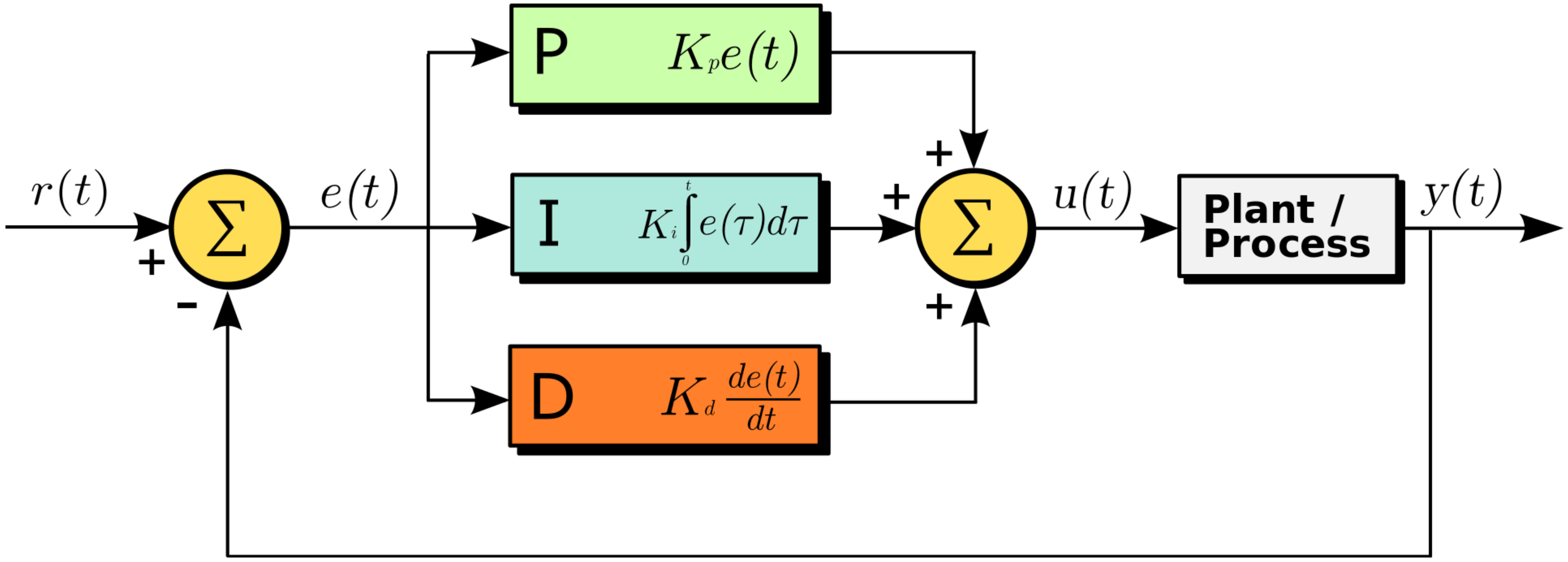
Converges to $z(t) = z_{ref}(t)$, where $k,\ d > 0$, if the connectivity undirected graph contains a spanning tree (similar to rooted out-branching, an acyclic graph that reaches all vertices, but no notion of root)

Formation control: relative state-based control



$k = 10, d = 0.5$

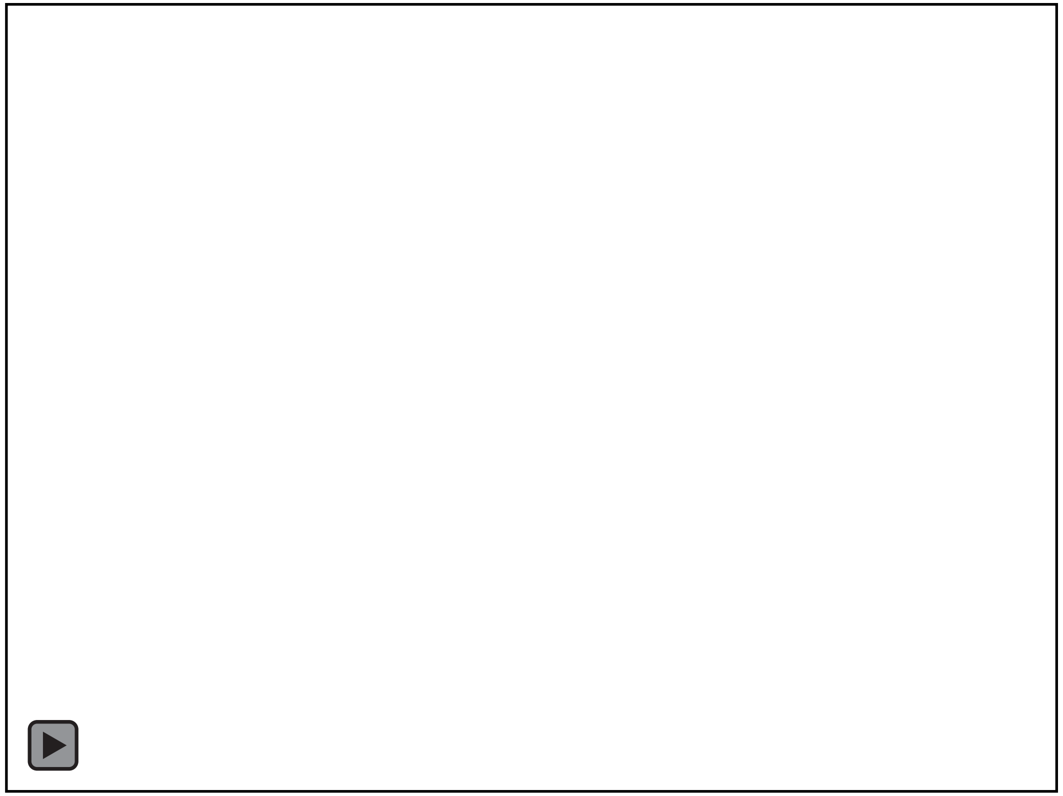
PID control and Integrator Dynamics

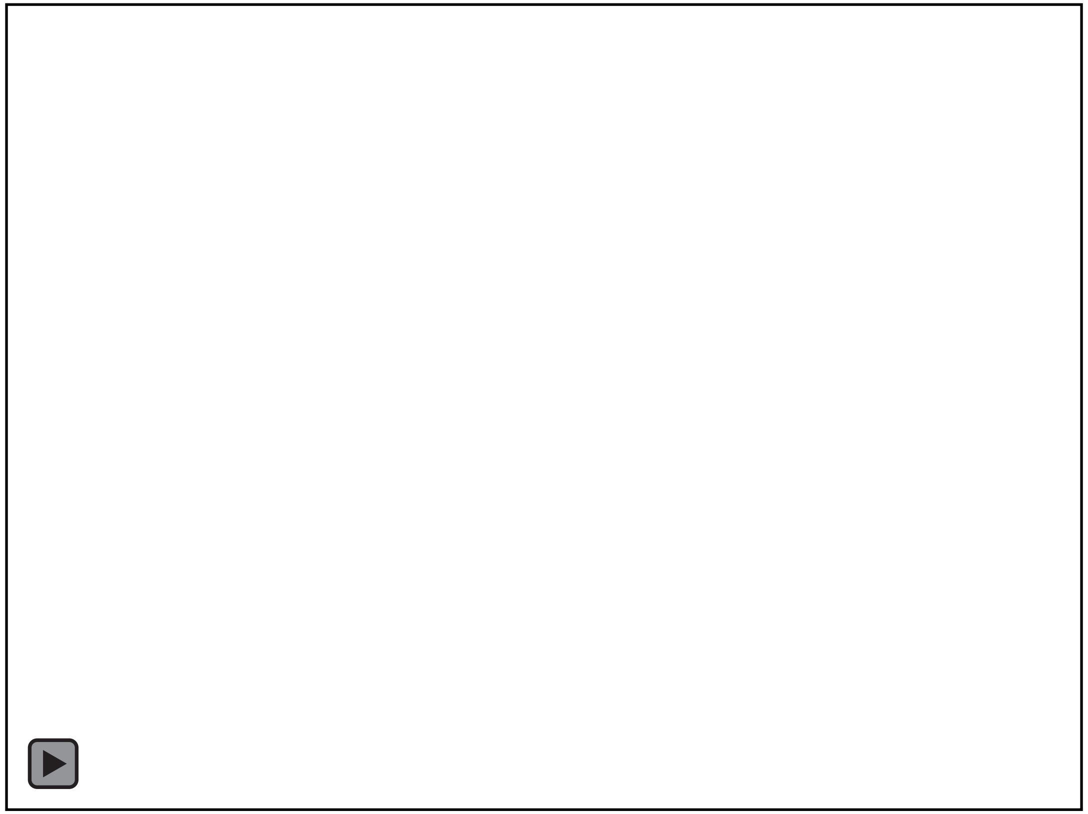


Rigidity-based control law

$$\dot{\mathbf{p}} = \mathbf{u}$$

$$\mathbf{u} = \mathbf{R}_{\mathcal{G}}^{T}(\mathbf{p})(\rho_{ref} - \rho_{\mathcal{G}}(\mathbf{p}))$$





Rigidity-based control law

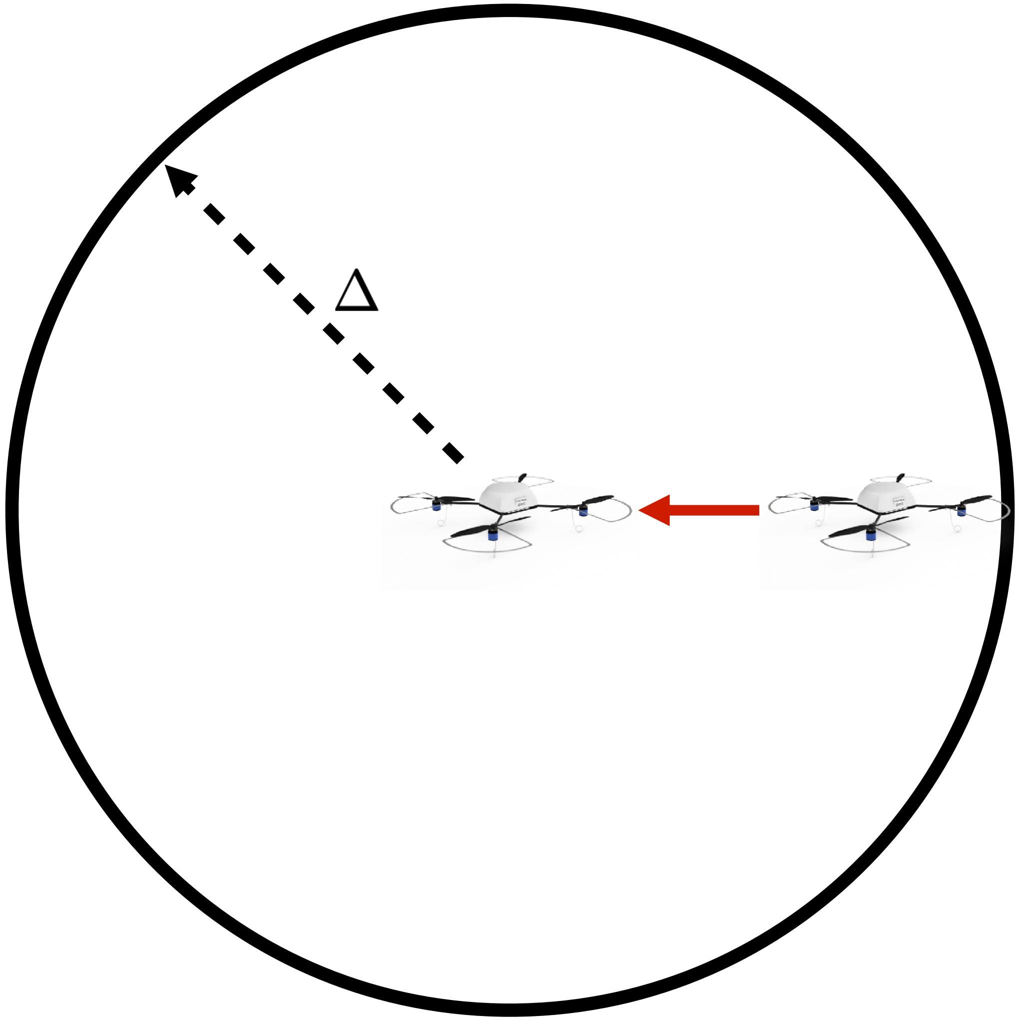
$$\dot{\mathbf{p}} = \mathbf{u}$$

$$\mathbf{u} = \mathbf{R}_{\mathcal{G}}^{T}(\mathbf{p})(\rho_{ref} - \rho_{\mathcal{G}}(\mathbf{p}))$$

- Is local
- Translation and rotation invariant
- Nonlinear control law based on rigidity
  - Nonlinear is not necessarily always better than linear, but it allows for more complicated behavior
- Can be problematic when rigidity matrix is singular

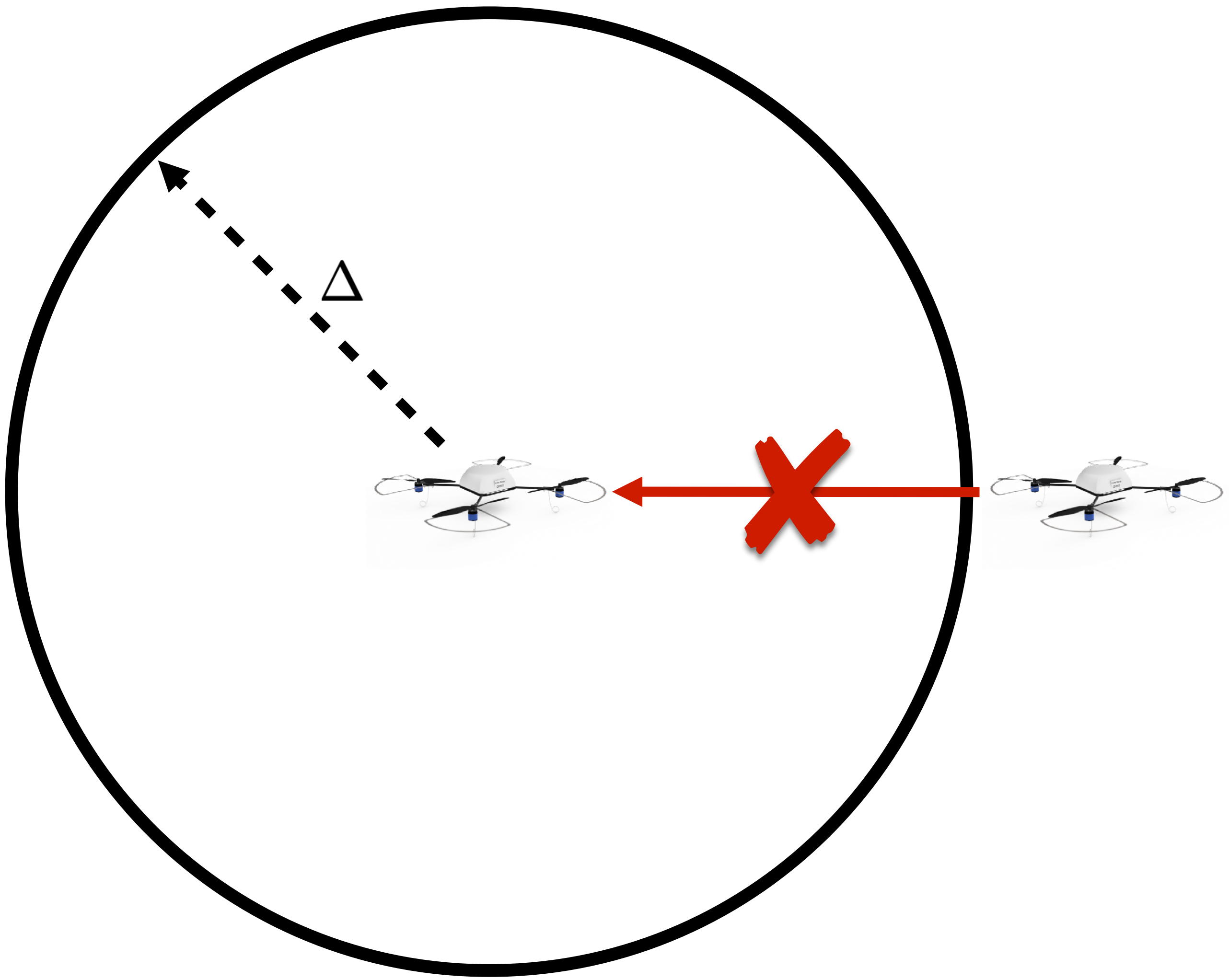
Formation control and real-world constraints

- Real robots will typically be able to communicate only locally



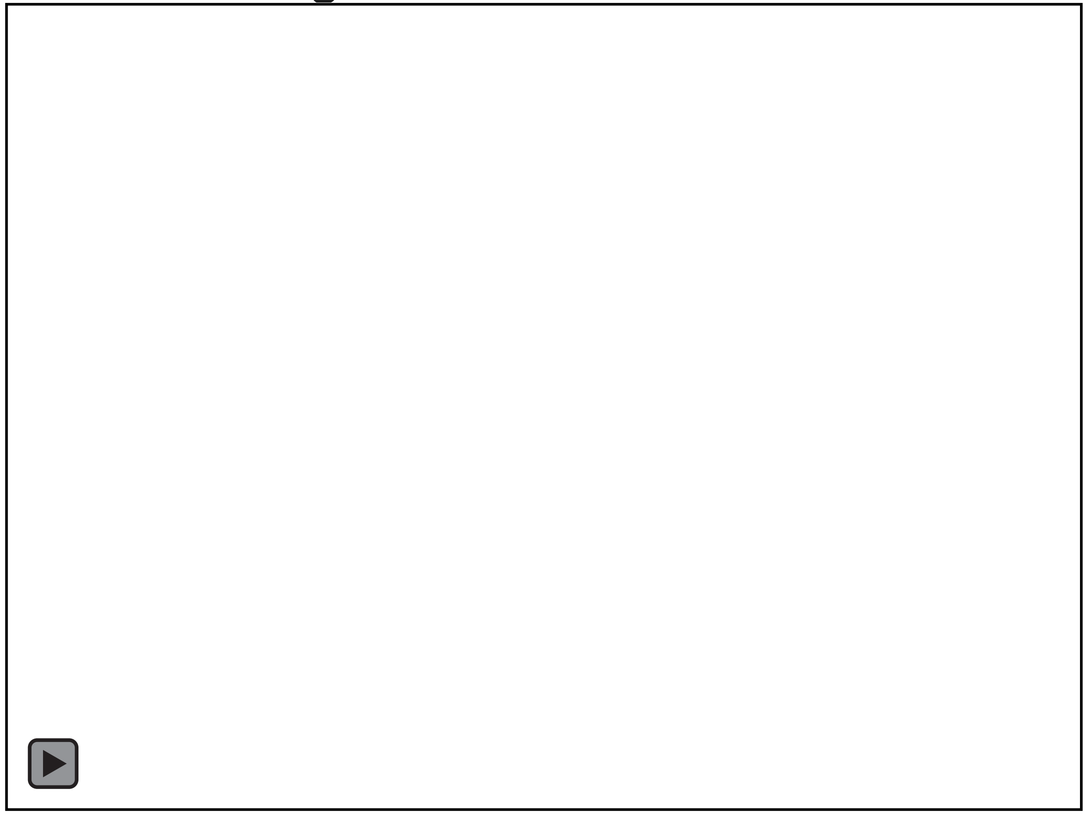
Formation control and real-world constraints

- Real robots will typically be able to communicate only locally



Formation control and real-world constraints

- Can we guarantee that during consensus or formation control the distance between neighbors does not exceed a certain radius Δ?



$$\Delta = 1.5$$

Formation control and real-world constraints

$$\Delta = 1.5$$

Weighted graph-based feedback

- Idea: extend formation control to take into account maximum distance constraints
  - Control law with single integrator dynamics:

$$\dot{\mathbf{x}}_i = \mathbf{u}_i$$

$$\mathbf{u}_i = -\sum_{j \in \mathcal{N}_i} f(\mathbf{x}_i - \mathbf{x}_j)$$

  where *f* is antisymmetric $f(\mathbf{x}_i - \mathbf{x}_j) = -f(\mathbf{x}_j - \mathbf{x}_i)$

    - Local (depends only on neighbors) and homogenous (same for all agents) control law
    - Only relative information is needed
    - Centroid of the states is guaranteed to be constant! $\overline{x} = \frac{1}{N}\sum_i \mathbf{x}_i$

Weighted graph-based feedback: Rendezvous

- How to pick the function $f(\cdot)$?
- **Idea:** Connection strength changes as a function of distance between agents

$$f(\mathbf{x}_i - \mathbf{x}_j) = -\omega(\mathbf{x}_i - \mathbf{x}_j)\cdot(\mathbf{x}_i - \mathbf{x}_j)$$

where $\omega(\mathbf{x}) : \mathbb{R}^p \rightarrow \mathbb{R}$ is a symmetric positive definite function

- This choice will guarantee a Laplacian-like control law

$$\dot{\mathbf{x}} = -E(\mathcal{G})W(x)E(\mathcal{G})^T \mathbf{x}$$

where $L_W = -E(\mathcal{G})W(x)E(\mathcal{G})^T \geq 0$ as long the G is connected (with only one zero eigenvalue)

Weighted graph-based feedback: Rendezvous

- $\mathcal{D}_{\mathcal{G},\delta}$ is the set of all swarm configurations (recall points $p \in \mathbb{R}^{Nd}$) where robots are always within a given distance $\delta > 0$ from each other.

$$\mathcal{D}_{\mathcal{G},\delta} = \{p \in \mathbb{R}^{Nd} \mid \| p_i - p_j \| \leq \delta, \forall v_i, v_j \in \mathcal{E}\}$$

Weighted graph-based feedback: Rendezvous

- $\mathcal{D}_{\mathcal{G},\delta}$ is the set of all swarm configurations (recall points $p \in \mathbb{R}^{Nd}$) where robots are always within a given distance $\delta > 0$ from each other.

$$\mathcal{D}_{\mathcal{G},\delta} = \{p \in \mathbb{R}^{Nd} \mid \| p_i - p_j \| \leq \delta, \forall v_i, v_j \in \mathcal{E}\}$$

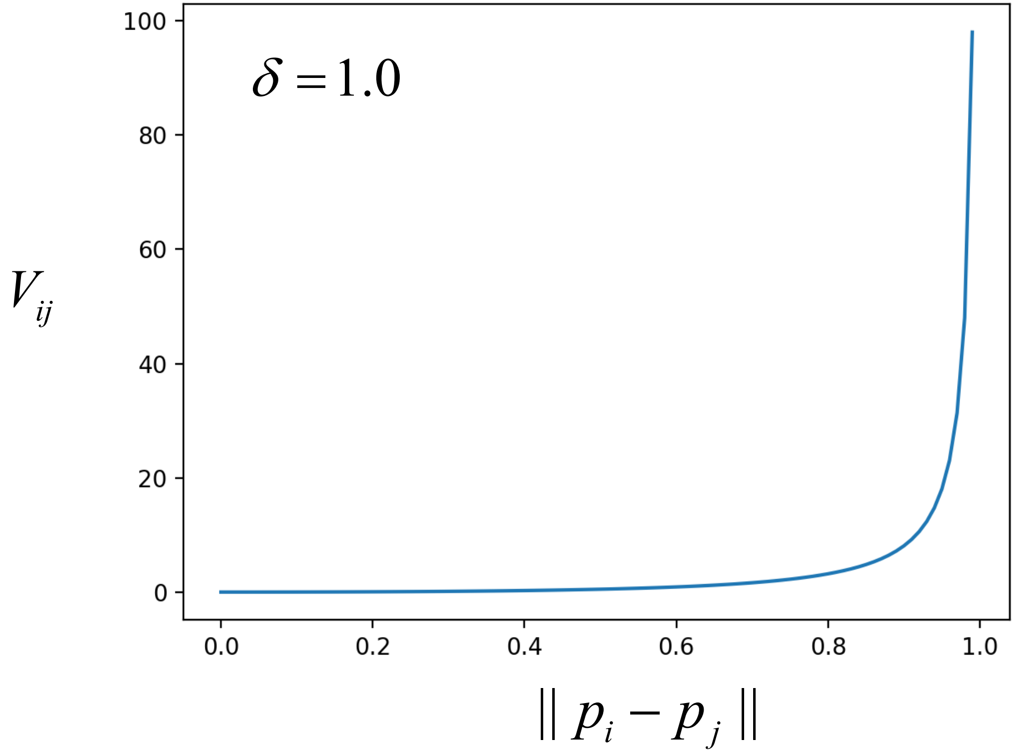
- $\mathcal{D}^{\varepsilon}_{\mathcal{G},\delta}$ is the set of all swarm configurations where robots are always within a given distance $\delta - \varepsilon > 0$ from each other. For now, think of ε as a bit of extra margin.

$$\mathcal{D}^{\varepsilon}_{\mathcal{G},\delta} = \{p \in \mathbb{R}^{Nd} \mid \| p_i - p_j \| \leq \delta - \varepsilon, \forall v_i, v_j \in \mathcal{E}\}$$

Weighted graph-based feedback: Rendezvous

- Define edge tension as a function

$$V_{ij}(\delta, p) = \begin{cases} \dfrac{\| p_i - p_j \|^2}{\delta - \| p_i - p_j \|} & \text{for } v_i, v_j \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$



- Define the total tension energy as $V(\delta, p) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} V_{ij}(\delta, p)$
  - Looks weird because the units are not in Joules

Weighted graph-based feedback: Rendezvous

- Recall: $\mathcal{D}^{\varepsilon}_{\mathcal{G},\delta} = \{p \in \mathbb{R}^{Nd} \mid \| p_i - p_j \| \leq \delta - \varepsilon, \forall v_i, v_j \in \mathcal{E}\}$ $\quad V(\delta, p) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} V_{ij}(\delta, p)$

- To make things simpler for control, I will define the system state $\mathbf{x}$ as the point p

$$\mathbf{x} \triangleq p$$

- Given an initial position $\mathbf{x}_0 \in \mathcal{D}^{\varepsilon}_{\mathcal{G},\delta}$ for $\varepsilon \in (0, \delta)$, if the graph $\mathcal{G}$ is connected, then

$$\Omega(\delta, \mathbf{x}_0) = \{\mathbf{x} \mid V(\delta, \mathbf{x}) \leq V(\delta, \mathbf{x}_0)\}$$

is an invariant set under the control law $\boxed{\mathbf{u}_i = -\sum_{j \in \mathcal{N}_i} \frac{2\delta - \| \mathbf{x}_i - \mathbf{x}_j \|}{(\delta - \| \mathbf{x}_i - \mathbf{x}_j \|)^2}(\mathbf{x}_i - \mathbf{x}_j)}$

where $x(t)$ will converge to the centroid $\bar{x} = \frac{1}{N}\sum_i \mathbf{x}_i$

Weighted graph-based feedback: Rendezvous

- Reflect: We just designed a swarm where everybody stays in touch
  - Not obvious from just looking at a control law
  - Required thinking back about invariant sets and Lyapunov functions

Why the margin ε?

- Graph dynamics based on proximity function

$$v_i, v_j \in \mathcal{E} \Leftrightarrow \| p_i - p_j \| \leq \Delta$$

- We can try to set $\delta = \Delta$ but we cannot switch the controller on as soon as the connection is made

Edge created

$$f(\mathbf{x}_i - \mathbf{x}_j) = 0$$

$$\xrightarrow{\| p_i - p_j \| \leq \Delta - \varepsilon}$$

Edge influence starts

$$f(\mathbf{x}_i - \mathbf{x}_j) = \frac{2\delta - \| \mathbf{x}_i - \mathbf{x}_j \|}{(\delta - \| \mathbf{x}_i - \mathbf{x}_j \|)^2} (\mathbf{x}_i - \mathbf{x}_j)$$

Weighted graph-based feedback: Rendezvous

Formation control with weights

- The translation invariant formation control law can be modified to ensure connectivity is maintained as

$$\boxed{\dot{x}_i(t) = -\sum_{j \in \mathcal{N}_{\mathcal{G}_d}(i)} \frac{2(\Delta - \|\boldsymbol{\tau}_i - \boldsymbol{\tau}_j\|) - \|\mathbf{x}_i(t) - \mathbf{x}_j(t) - (\boldsymbol{\tau}_i - \boldsymbol{\tau}_j)\|}{(\Delta - \|\boldsymbol{\tau}_i - \boldsymbol{\tau}_j\| - \|\mathbf{x}_i(t) - \mathbf{x}_j(t) - (\boldsymbol{\tau}_i - \boldsymbol{\tau}_j)\|)^2} (\mathbf{x}_i(t) - \mathbf{x}_j(t) - (\boldsymbol{\tau}_i - \boldsymbol{\tau}_j))}$$

where $\boldsymbol{\tau}_i - \boldsymbol{\tau}_j$ is the desired translation vector between two vertices i and j

for all i, j such that $v_i, v_j \in \mathcal{E}_d$

Formation control with weights

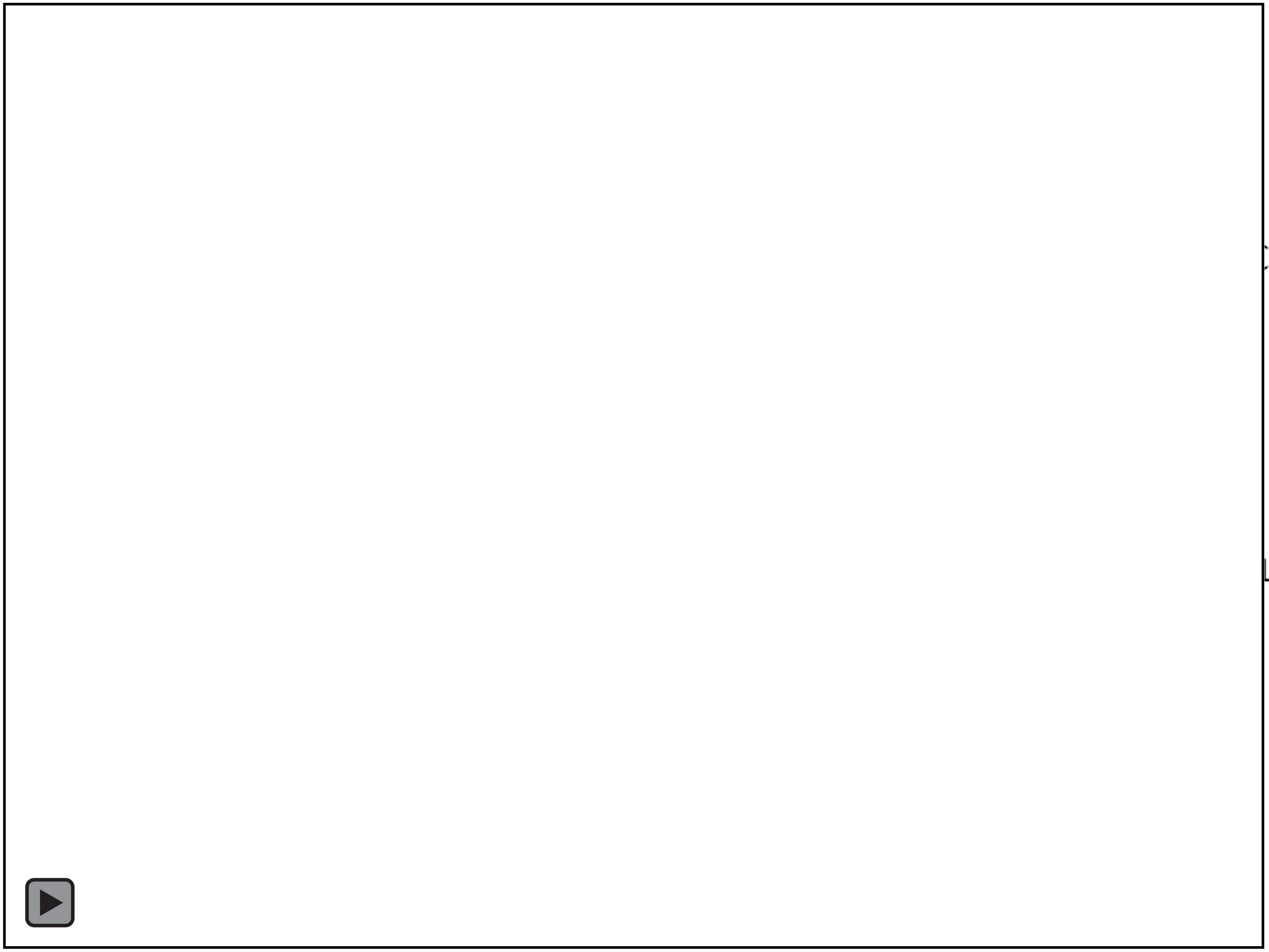
- The translation invariant formation control law can be modified to ensure connectivity is maintained as

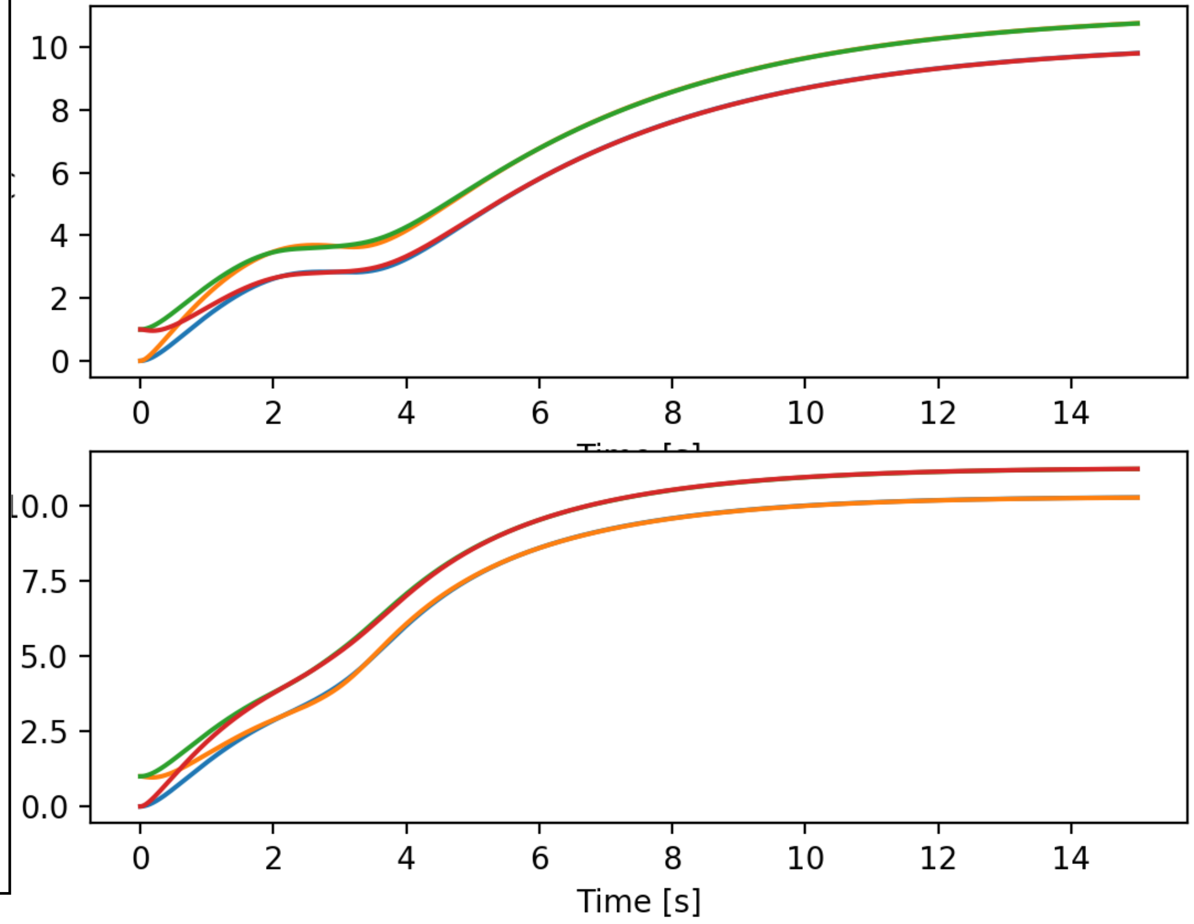
$$\boxed{\dot{x}_i(t) = -\sum_{j \in \mathcal{N}_{\mathcal{G}_d}(i)} \frac{2(\Delta - \| \underline{\boldsymbol{\tau}_i - \boldsymbol{\tau}_j} \|) - \| \underline{\mathbf{x}_i(t) - \mathbf{x}_j(t)} - (\underline{\boldsymbol{\tau}_i - \boldsymbol{\tau}_j}) \|}{(\Delta - \| \underline{\boldsymbol{\tau}_i - \boldsymbol{\tau}_j} \| - \| \underline{\mathbf{x}_i(t) - \mathbf{x}_j(t)} - (\underline{\boldsymbol{\tau}_i - \boldsymbol{\tau}_j}) \|)^2} (\underline{\mathbf{x}_i(t) - \mathbf{x}_j(t)} - (\underline{\boldsymbol{\tau}_i - \boldsymbol{\tau}_j}))}$$

where $\boldsymbol{\tau}_i - \boldsymbol{\tau}_j$ is the desired translation vector for the formation between two vertices i and j for all i, j such that $v_i, v_j \in \mathcal{E}_d$

- Think about how to make it rotation invariant

Control of formation + target + obstacle avoidance





Control Law Example

- Why is a second derivative an appropriate choice here?

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = F_{formation,i} + F_{target,i} + F_{obstacle,i}$$