# Mobile Robots

## Assembly and Control

Mathias Hudoba de Badyn

Advanced Topics in Control
May 16, 2022

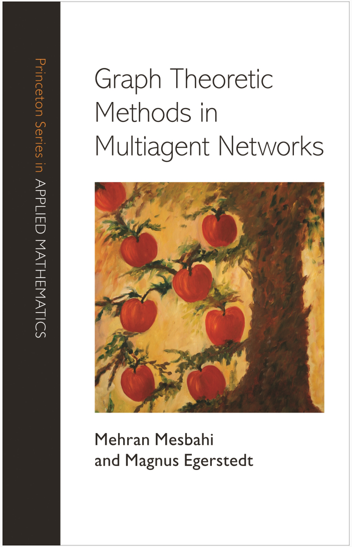
# Announcements

1. Homework 3 is extended, Vahid is coming at 17:45 to answer questions

# Announcements

1. Homework 3 is extended, Vahid is coming at 17:45 to answer questions
2. Project ideas should be finalized

# Recap from last week

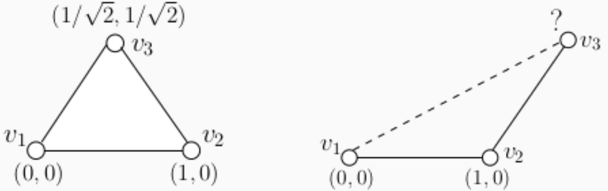

Princeton Series in APPLIED MATHEMATICS
Graph Theoretic Methods in Multiagent Networks
Mehran Mesbahi
and Magnus Egerstedt

# Recap of Rigidity

**Desired Interagent distances**

$$D = \{d_{ij} \in \mathbb{R} \ : \ d_{ij} > 0; i, j = 1, \ldots, n, i \neq j\}$$

$D$ must be **feasible:** there exist points $\Xi := \{\xi_1, \ldots, \xi_n \mid \xi_i \in \mathbb{R}^p\}$ such that $\|\xi_i - \xi_j\| = d_{ij}$,




$D = \{d_{12} = d_{13} = d_{23} = 1\}$

(a) Feasible formation

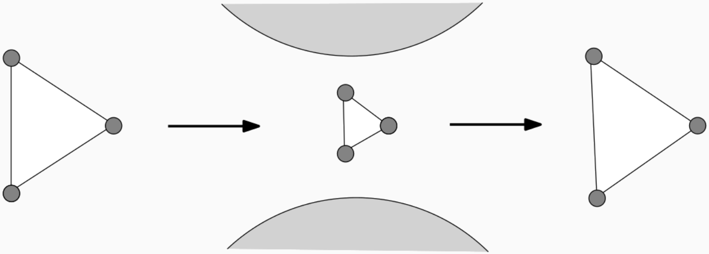
$D = \{d_{12} = d_{23} = 1, \ d_{13} = 3\}$

(b) Infeasible formation

# Recap of Rigidity

We refer to the **scale invariant** formation $D$ as any set of distances $D'$ such that $D' = \alpha D$ for any $\alpha \in \mathbb{R}_+$.

Used in 'cluttered' environments where strict interagent distances are not required.



# Recap of Rigidity

Interagent distances

$$D = \{d_{ij} = d_{ji} \geq 0,\ i, j = 1, \ldots, n,\ i \neq j\}$$

| formation | specification | interpretation |
|---|---|---|
| scale invariant | $D$ | $\|x_i - x_j\| = \alpha d_{ij}$ for some $\alpha > 0$ |
| rigid | $D$ | $\|x_i - x_j\| = d_{ij}$ |
| translational invariant | $\Xi$ | $x_i = \xi_i + \tau$ for some $\tau \in \mathbf{R}^p$ |

# Rigidity

Define a **formation graph** as $\mathcal{G}_f = (\mathcal{V}, E_f, w)$, where

- $\mathcal{V}$ is a set of vertices, denoting agents

# Rigidity

Define a **formation graph** as $\mathcal{G}_f = (\mathcal{V}, E_f, w)$, where

- $\mathcal{V}$ is a set of vertices, denoting agents
- $E_f$ is a set of edges, denoting interagent connections

# Rigidity

Define a **formation graph** as $\mathcal{G}_f = (\mathcal{V}, E_f, w)$, where

- $\mathcal{V}$ is a set of vertices, denoting agents
- $E_f$ is a set of edges, denoting interagent connections
- $w : \mathcal{E} \mapsto \mathbb{R}_+$ associates a **feasible** desired interagent distance to each edge

# Rigidity

Define a **formation graph** as $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}, w)$, where

- $\mathcal{V}$ is a set of vertices, denoting agents
- $\mathcal{E}$ is a set of edges, denoting interagent connections
- $w : \mathcal{E} \mapsto \mathbb{R}_+$ associates a **feasible** desired interagent distance to each edge

Given a feasible $\Xi$, define:

- A **framework**: $(\Xi, \mathcal{G}_f) := \mathcal{G}(\Xi)$

# Rigidity

Define a **formation graph** as $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}, w)$, where

- $\mathcal{V}$ is a set of vertices, denoting agents
- $\mathcal{E}$ is a set of edges, denoting interagent connections
- $w : \mathcal{E} \mapsto \mathbb{R}_+$ associates a **feasible** desired interagent distance to each edge

Given a feasible $\Xi$, define:

- A **framework**: $(\Xi, \mathcal{G}_f) := \mathcal{G}(\Xi)$
- A **framework trajectory** as a set of states $x_1(t), \ldots, x_n(t)$ such that $x_i(0) = \xi_i$, which...

# Rigidity

Define a **formation graph** as $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}, w)$, where

- $\mathcal{V}$ is a set of vertices, denoting agents
- $\mathcal{E}$ is a set of edges, denoting interagent connections
- $w : \mathcal{E} \mapsto \mathbb{R}_+$ associates a **feasible** desired interagent distance to each edge

Given a feasible $\Xi$, define:

- A **framework**: $(\Xi, \mathcal{G}_f) := \mathcal{G}(\Xi)$
- A **framework trajectory** as a set of states $x_1(t), \ldots, x_n(t)$ such that $x_i(0) = \xi_i$, which...
- is called **edge consistent** if $\|x_i(t) - x_j(t)\|$ is constant for all $ij \in \mathcal{E}, t \in \mathbb{R}_+$, or

# Rigidity

Define a **formation graph** as $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}, w)$, where

- $\mathcal{V}$ is a set of vertices, denoting agents
- $\mathcal{E}$ is a set of edges, denoting interagent connections
- $w : \mathcal{E} \mapsto \mathbb{R}_+$ associates a **feasible** desired interagent distance to each edge

Given a feasible $\Xi$, define:

- A **framework**: $(\Xi, \mathcal{G}_f) := \mathcal{G}(\Xi)$
- A **framework trajectory** as a set of states $x_1(t), \ldots, x_n(t)$ such that $x_i(0) = \xi_i$, which...
- is called **edge consistent** if $\|x_i(t) - x_j(t)\|$ is constant for all $ij \in \mathcal{E}, t \in \mathbb{R}_+$, or
- **rigid**, if $\|x_i(t) - x_j(t)\|$ is constant for all $i \neq j$.

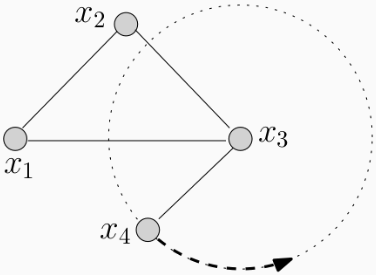
# Rigidity

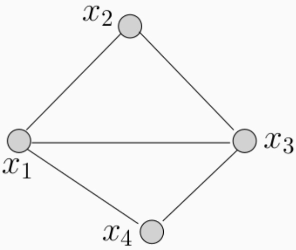
**Definition**

A framework $(\Xi, \mathcal{G}_f)$ is called **rigid** if and only if all edge-consistent trajectories of the framework are rigid trajectories.

Otherwise, the framework is called flexible.

# Rigidity




**Left:** Flexible framework with edge-consistent trajectories.

**Right:** Rigid framework

# Rigidity

**Definition**

A framework $\mathcal{G}(\Xi)$ is **infinitesimally rigid** if $R(\mathcal{G}(\Xi))u = 0$ for all $u$ satisfying

$$(u_i(t) - u_j(t))^T(x_i(t) - x_j(t)) = 0 \text{ for all } ij \in \mathcal{E}$$

Some properties of $R(\mathcal{G}(\Xi))$:

- Has $|\mathcal{E}|$ rows
- Has $pn$ columns ($p$ is the state dimension)
- Is an algebraic representation of a graph plus the points $\Xi$

**Theorem**

If $\mathcal{G}$ is rigid (i.e., there exists at least one $\Xi$ such that $\mathcal{G}(\Xi)$ is infinitesimally rigid), then the set of all $\Xi$ such that $\mathcal{G}(\Xi)$ is infinitesimally rigid is a dense, open subset of $\mathbb{R}^{pn}$.

1. Rigid graphs have a lot of feasible configurations!

# Rigid Graphs

**Theorem**

If $\mathcal{G}$ is rigid (i.e., there exists at least one $\Xi$ such that $\mathcal{G}(\Xi)$ is infinitesimally rigid), then the set of all $\Xi$ such that $\mathcal{G}(\Xi)$ is infinitesimally rigid is a dense, open subset of $\mathbb{R}^{pn}$.

1. Rigid graphs have a lot of feasible configurations!
2. Feasible configurations can be approximated arbitrarily well (remember what dense + open means)

# Mobile robotics: Outline

1. Today: Specifying formations by relative states

# Mobile robotics: Outline

1. Today: Specifying formations by relative states
2. Shape-based control (controlling to target $\Xi$)

# Mobile robotics: Outline

1. Today: Specifying formations by relative states
2. Shape-based control (controlling to target $\Xi$)
3. Relative state-based control (controlling to $\|x_i - x_j\| = d_{ij}$)

# Formation Specification via Relative States

$$z(t) = \begin{bmatrix} x_1(t) - x_2(t) \\ x_2(t) - x_3(t) \end{bmatrix} \in \mathbb{R}^4$$

This specifies **all** relative formation states, including

$$x_1(t) - x(3) = [x_1(t) - x_2(t)] + [x_2(t) - x_3(t)]$$

# Formation Specification via Relative States

We can also set a formation center $x_o(t)$ and control the states,

$$z(t) = \begin{bmatrix} x_o(t) - x_1(t) \\ x_1(t) - x_2(t) \\ x_2(t) - x_3(t) \end{bmatrix} \in \mathbb{R}^6$$

# Formation Specification via Relative States – Incidence Matrix

$$z(t) = \begin{bmatrix} x_o(t) - x_1(t) \\ x_1(t) - x_2(t) \\ x_2(t) - x_3(t) \end{bmatrix} \in \mathbb{R}^6$$

$$z(t) = D(\mathcal{D})^T x(t)$$

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

$$D(\mathcal{D}) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{bmatrix} \otimes I_2$$

# Formation Specification via Relative States – RSS

We need to be able to specify “enough” relative states to uniquely place the formation.

We need to be able to specify "enough" relative states to uniquely place the formation.

This can be done by choosing a **spanning subgraph** of the **complete directed graph** on $n$ nodes,

$$z(t) = \left(D(\mathcal{D})^T \otimes I_2\right) x(t),$$

# Formation Specification via Relative States – RSS

We need to be able to specify "enough" relative states to uniquely place the formation.

This can be done by choosing a **spanning subgraph** of the **complete directed graph** on $n$ nodes,

$$z(t) = \left(D(\mathcal{D})^T \otimes I_2\right) x(t),$$

Then we control $z(t)$ to some $z_{\text{ref}}(t)$.

# Formation Specification via Relative States – RSS

We need to be able to specify “enough” relative states to uniquely place the formation.

This can be done by choosing a **spanning subgraph** of the **complete directed graph** on $n$ nodes,

$$z(t) = \left(D(\mathcal{D})^T \otimes I_2\right) x(t),$$

Then we control $z(t)$ to some $z_{\text{ref}}(t)$.

This is called the **relative state specification (RSS)**.

# Formation Specification via Relative States – RSS Example

# Equivalence/Converting between RSS's

Suppose we have two spanning digraphs $\mathcal{D}_j$ and $\mathcal{D}_d$. Are these equivalent representations of an RSS?

# Equivalence/Converting between RSS's

Suppose we have two spanning digraphs $\mathcal{D}_j$ and $\mathcal{D}_d$. Are these equivalent representations of an RSS?

We can show equivalence by finding a linear transformation

$$T_{dj} D(\mathcal{D}_j)^T = D(\mathcal{D}_d)^T$$

# Equivalence/Converting between RSS's

Suppose we have two spanning digraphs $\mathcal{D}_j$ and $\mathcal{D}_d$. Are these equivalent representations of an RSS?

We can show equivalence by finding a linear transformation

$$T_{dj} D(\mathcal{D}_j)^T = D(\mathcal{D}_d)^T$$

**Proposition**

Let $\mathcal{D}_j$ be a directed spanning tree RSS and let $\mathcal{D}_d$ be an arbitrary RSS. Then,

$$T_{dj} = D(\mathcal{D}_d)^T D(\mathcal{D}_j) \left[ D(\mathcal{D}_j)^T D(\mathcal{D}_j) \right]^{-1}$$

# Equivalence/Converting between RSS's

Suppose we have two spanning digraphs $\mathcal{D}_j$ and $\mathcal{D}_d$. Are these equivalent representations of an RSS?

We can show equivalence by finding a linear transformation

$$T_{dj} D(\mathcal{D}_j)^T = D(\mathcal{D}_d)^T$$

**Proposition**

Let $\mathcal{D}_j$ be a directed spanning tree RSS and let $\mathcal{D}_d$ be an arbitrary RSS. Then,

$$T_{dj} = D(\mathcal{D}_d)^T D(\mathcal{D}_j) \left[ D(\mathcal{D}_j)^T D(\mathcal{D}_j) \right]^{-1}$$

This is the **projection** of the edge states of $\mathcal{D}_j$ onto those of $\mathcal{D}_d$.

# shape-based control

# Shape-based control – Preliminaries

Recall:

1. **Formation graph:** $\mathcal{G}_f(\mathcal{V}, E_f)$

# Shape-based control – Preliminaries

Recall:

1. **Formation graph:** $\mathcal{G}_f(\mathcal{V}, E_f)$
2. **Target locations:** $\Xi = \{\xi_1, \ldots, \xi_n\}$

# Shape-based control – Preliminaries

Recall:

1. **Formation graph:** $\mathcal{G}_f(\mathcal{V}, E_f)$
2. **Target locations:** $\Xi = \{\xi_1, \ldots, \xi_n\}$
3. **Goal:** $x_i(t) \to \xi_i + \boldsymbol{\tau}$, for some $\boldsymbol{\tau} \in \mathbb{R}^p$

# Shape-based control – Preliminaries

Recall:

1. **Formation graph:** $\mathcal{G}_f(\mathcal{V}, E_f)$
2. **Target locations:** $\Xi = \{\xi_1, \ldots, \xi_n\}$
3. **Goal:** $x_i(t) \to \xi_i + \boldsymbol{\tau}$, for some $\boldsymbol{\tau} \in \mathbb{R}^p$
4. $\boldsymbol{\tau}$ here represents an offset relative to some frame of reference

# Shape-based control – Preliminaries

Recall:

1. **Formation graph:** $\mathcal{G}_f(\mathcal{V}, E_f)$
2. **Target locations:** $\Xi = \{\xi_1, \ldots, \xi_n\}$
3. **Goal:** $x_i(t) \to \xi_i + \boldsymbol{\tau}$, for some $\boldsymbol{\tau} \in \mathbb{R}^p$
4. $\boldsymbol{\tau}$ here represents an offset relative to some frame of reference
5. **Communication graph:** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

# Shape-based control – Preliminaries

Recall:

1. **Formation graph:** $\mathcal{G}_f(\mathcal{V}, E_f)$
2. **Target locations:** $\Xi = \{\xi_1, \ldots, \xi_n\}$
3. **Goal:** $x_i(t) \to \xi_i + \boldsymbol{\tau}$, for some $\boldsymbol{\tau} \in \mathbb{R}^p$
4. $\boldsymbol{\tau}$ here represents an offset relative to some frame of reference
5. **Communication graph:** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

The goal of formation control:

# Shape-based control – Preliminaries

Recall:

1. **Formation graph:** $\mathcal{G}_f(\mathcal{V}, E_f)$
2. **Target locations:** $\Xi = \{\xi_1, \ldots, \xi_n\}$
3. **Goal:** $x_i(t) \to \xi_i + \boldsymbol{\tau}$, for some $\boldsymbol{\tau} \in \mathbb{R}^p$
4. $\boldsymbol{\tau}$ here represents an offset relative to some frame of reference
5. **Communication graph:** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

The goal of formation control:

(R1): $\|x_i(t) - x_j(t)\| \to d_{ij}$ for all $ij \in E_f$

Recall:

1. **Formation graph:** $\mathcal{G}_f(\mathcal{V}, E_f)$
2. **Target locations:** $\Xi = \{\xi_1, \ldots, \xi_n\}$
3. **Goal:** $x_i(t) \to \xi_i + \boldsymbol{\tau}$, for some $\boldsymbol{\tau} \in \mathbb{R}^p$
4. $\boldsymbol{\tau}$ here represents an offset relative to some frame of reference
5. **Communication graph:** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

The goal of formation control:

(R1): $\|x_i(t) - x_j(t)\| \to d_{ij}$ for all $ij \in E_f$

(R2): If the interaction graph $\mathcal{G}(t)$ is dynamic, then it should converge to $E_f \subseteq \mathcal{E}$ for all $t \geq T$, for finite $T$.

# the static graph case

# Shape-based control – The static graph case

Suppose (R2) is satisfied: all pairs of target relative states $ij \in E_j$ are also communicating, in that $i$ and $j$ exchange information ($ij \in \mathcal{E}$)

# Shape-based control – The static graph case

Suppose (R2) is satisfied: all pairs of target relative states $ij \in E_j$ are also communicating, in that $i$ and $j$ exchange information ($ij \in \mathcal{E}$)

Define $\boldsymbol{\tau}_i$ as the displacement of $x_i$ from its target $\xi_i$

$$\boldsymbol{\tau}_i(t) = x_i(t) - \xi_i, \;\; i \in \mathcal{N}$$

# Shape-based control – The static graph case

Suppose (R2) is satisfied: all pairs of target relative states $ij \in E_j$ are also communicating, in that $i$ and $j$ exchange information ($ij \in \mathcal{E}$)

Define $\boldsymbol{\tau}_i$ as the displacement of $x_i$ from its target $\xi_i$

$$\boldsymbol{\tau}_i(t) = x_i(t) - \xi_i, \;\; i \in \mathcal{N}$$

By reaching agreement over the $\boldsymbol{\tau}_i$'s, each agent would satisfy $x_i - \xi_i = \boldsymbol{\tau}$, and so $\|x_i - x_j\| = d_{ij}$ as required by (R1)

# Shape-based control – The static graph case & consensus

We can let each agent $i$ perform consensus on $\boldsymbol{\tau}_i$ with its neighbours in $E_f$:

$$\dot{\boldsymbol{\tau}}_i(t) = -\sum_{j \in N_f(i)} \left(\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_j(t)\right)$$

## Shape-based control – The static graph case & consensus

We can let each agent $i$ perform consensus on $\boldsymbol{\tau}_i$ with its neighbours in $E_f$:

$$\dot{\boldsymbol{\tau}}_i(t) = -\sum_{j \in N_f(i)} (\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_j(t))$$

However, we want this consensus to be over the *formation graph* $\mathcal{G}_f$, since this encodes our target set $\Xi$:

$$N_f(i) = \{j \in \mathcal{V} \ : \ ij \in E_f\}$$

# Shape-based control – The static graph case & consensus

We can let each agent $i$ perform consensus on $\boldsymbol{\tau}_i$ with its neighbours in $E_f$:

$$\dot{\boldsymbol{\tau}}_i(t) = -\sum_{j \in N_f(i)} (\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_j(t))$$

However, we want this consensus to be over the *formation graph* $\mathcal{G}_f$, since this encodes our target set $\Xi$:

$$N_f(i) = \{j \in \mathcal{V} \ : \ ij \in E_f\}$$

Since we assume that (R2) is satisfied, $E_f \subseteq \mathcal{E}$, and so this is a feasible communication strategy.

Note that $\dot{\tau}_i = \dot{x}_i$, and

$$\tau_i(t) - \tau_j(t) = x_i(t) - x_j(t) - (\xi_i - \xi_j).$$

Note that $\dot{\tau}_i = \dot{x}_i$, and

$$\tau_i(t) - \tau_j(t) = x_i(t) - x_j(t) - (\xi_i - \xi_j).$$

Therefore, we can write

$$\dot{x}_i(t) = \dot{\tau}_i(t) = -\sum_{j \in N_f(i)} (\tau_i(t) - \tau_j(t)) = -\sum_{j \in N_f(t)} \big(x_i(t) - x_j(t)\big) - \big(\xi_i - \xi_j\big)$$

## Shape-based control – The static graph case & consensus

**Theorem**

Consider the protocol,

$$\dot{x}_i(t) = -\sum_{j \in N_f(t)} \big(x_i(t) - x_j(t)\big) - (\xi_i - \xi_j)$$

on a connected formation graph $\mathcal{G}_f = (\mathcal{V}, E_f)$ encoding a set of target locations $\Xi$.

**Theorem**

Consider the protocol,

$$\dot{x}_i(t) = -\sum_{j \in N_f(t)} \big(x_i(t) - x_j(t)\big) - \big(\xi_i - \xi_j\big)$$

on a connected formation graph $\mathcal{G}_f = (\mathcal{V}, E_f)$ encoding a set of target locations $\Xi$.

If the communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ satisfies $E_f \subseteq \mathcal{E}$, then the above protocol drives the agents to a constant displacement of the target positions:

**Theorem**

Consider the protocol,

$$\dot{x}_i(t) = -\sum_{j \in N_f(t)} \big(x_i(t) - x_j(t)\big) - \big(\xi_i - \xi_j\big)$$

on a connected formation graph $\mathcal{G}_f = (\mathcal{V}, E_f)$ encoding a set of target locations $\Xi$.

If the communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ satisfies $E_f \subseteq \mathcal{E}$, then the above protocol drives the agents to a constant displacement of the target positions:

$$x_i(t) - \xi_i \to \tau, \ \forall i \in \mathcal{N}.$$

as $t \to \infty$.

# the dynamic graph case

# the dynamic graph case

(Way harder!)

# Shape-based control – The dynamic graph case graph case

Suppose $\mathcal{G}$ is dynamic, i.e. the edge set is a function of time: $\mathcal{E} = \mathcal{E}(t)$.

# Shape-based control – The dynamic graph case graph case

Suppose $\mathcal{G}$ is dynamic, i.e. the edge set is a function of time: $\mathcal{E} = \mathcal{E}(t)$.

The only simple result is if we assume $E_f \subseteq \mathcal{E}(t)$ for all $t$.

# Shape-based control – The dynamic graph case graph case

Suppose $\mathcal{G}$ is dynamic, i.e. the edge set is a function of time: $\mathcal{E} = \mathcal{E}(t)$.

The only simple result is if we assume $E_f \subseteq \mathcal{E}(t)$ for all $t$.

**Corollary**

Given a connected, target formation graph $\mathcal{G}_f = (\mathcal{V}, E_f)$, the above protocol will asymptotically drive all agents to a constant displace- ment of the target positions if for all $t \geq 0$, $E_f \subseteq \mathcal{E}(t)$.

# Shape-based control – The dynamic graph case graph case

Suppose $\mathcal{G}$ is dynamic, i.e. the edge set is a function of time: $\mathcal{E} = \mathcal{E}(t)$.

The only simple result is if we assume $E_f \subseteq \mathcal{E}(t)$ for all $t$.

**Corollary**

Given a connected, target formation graph $\mathcal{G}_f = (\mathcal{V}, E_f)$, the above protocol will asymptotically drive all agents to a constant displace- ment of the target positions if for all $t \geq 0$, $E_f \subseteq \mathcal{E}(t)$.

There are many results on graphs that vary randomly over time, but requires complicated machinerty to analyze

# relative state-based control

# Relative state-based control

Let's start with a feedback control design procedure.

Suppose each agent $i \in \mathcal{N}$ is a single integrator $\dot{x}_i = u_i$.

# Relative state-based control

Let's start with a feedback control design procedure.

Suppose each agent $i \in \mathcal{N}$ is a single integrator $\dot{x}_i = u_i$.

Given a spanning tree digraph $\mathcal{D}$, the **relative states** are defined as,

$$z(t) = D(\mathcal{D})^T x(t)$$

# Relative state-based control

Let's start with a feedback control design procedure.

Suppose each agent $i \in \mathcal{N}$ is a single integrator $\dot{x}_i = u_i$.

Given a spanning tree digraph $\mathcal{D}$, the **relative states** are defined as,

$$z(t) = D(\mathcal{D})^T x(t)$$

Let $z_{\text{ref}}$ be a constant **reference relative position** for the agents

# Relative state-based control

Let's start with a feedback control design procedure.

Suppose each agent $i \in \mathcal{N}$ is a single integrator $\dot{x}_i = u_i$.

Given a spanning tree digraph $\mathcal{D}$, the **relative states** are defined as,

$$z(t) = D(\mathcal{D})^T x(t)$$

Let $z_{\text{ref}}$ be a constant **reference relative position** for the agents

The formation error at time $t$ is defined as,

$$e(t) = z_{\text{ref}} - z(t).$$

# Relative state-based control

We can write the dynamics of the error as,

$$\dot{e}(t) = -D(\mathcal{D})^T u(t).$$

*Proof:*

# Relative state-based control

A common feedback design is to integrate proportional to the error:

$$u(t) = kD(\mathcal{D})e(t).$$

# Relative state-based control

A common feedback design is to integrate proportional to the error:

$$u(t) = kD(\mathcal{D})e(t).$$

Then, we can write,

$$\dot{e} = -kL_e(\mathcal{D})e(t)$$

*Proof:*

The edge Laplacian is positive-definite for a spanning tree, and so

$$\lim_{t \to \infty} e(t) = 0,$$

.

# Relative state-based control

The edge Laplacian is positive-definite for a spanning tree, and so

$$\lim_{t \to \infty} e(t) = 0,$$

and so the $n$ single integrators achieve $z_{\text{ref}}$ if $k > 0$ as $t \to \infty$.

# Relative state-based control – Dynamics

What do the dynamics of $x$ look like under this control?

# Relative state-based control – Dynamics

What do the dynamics of $x$ look like under this control?

**Lemma**

Under the control law $u(t) = kD(\mathcal{D})e(t)$, the state dynamics are,

$$\dot{x}(t) = -kL(\tilde{\mathcal{D}})x(t) + kD(\mathcal{D})z_{\text{ref}},$$

where $\tilde{\mathcal{D}}$ is the undirected version of the spanning tree.

*Proof:*

# Why does the Laplacian use the undirected version of $\mathcal{D}$?

# relative state-based control – double-integrator case

# Relative state-based control – Double Integrator

Consider the double-integrator control $\ddot{x}_i(t) = u_i(t)$, for all $i \in \mathcal{N}$.

Consider the double-integrator control $\ddot{x}_i(t) = u_i(t)$, for all $i \in \mathcal{N}$.

Suppose that desired relative positions and velocities are defined via a spanning tree digraph $\mathcal{D}$ as,

$$\begin{bmatrix} z_{\text{ref}} \\ \dot{z}_{\text{ref}} \end{bmatrix}.$$

# Relative state-based control – Double Integrator

Consider the double-integrator control $\ddot{x}_i(t) = u_i(t)$, for all $i \in \mathcal{N}$.

Suppose that desired relative positions and velocities are defined via a spanning tree digraph $\mathcal{D}$ as,

$$\begin{bmatrix} z_{\text{ref}} \\ \dot{z}_{\text{ref}} \end{bmatrix}.$$

We further assume that $\ddot{z}_{\text{ref}} = 0$, i.e. the reference velocity does not change with time.

# Relative state-based control – Double Integrator

Again, we set the error as $e(t) = z_{\text{ref}} - D(\mathcal{D})^T x(t)$.

Again, we set the error as $e(t) = z_{\text{ref}} - D(\mathcal{D})^T x(t)$.

We can show that,

$$\ddot{e} = -D(\mathcal{D})^T u(t).$$

*Proof:*

# Relative state-based control – Double Integrator

Define the (proportional-derivative) feedback controller,

$$u(t) = k \begin{bmatrix} D(\mathcal{D}) & D(\mathcal{D}) \end{bmatrix} \begin{bmatrix} e(t) \\ \dot{e}(t) \end{bmatrix}.$$

Define the (proportional-derivative) feedback controller,

$$u(t) = k \begin{bmatrix} D(\mathcal{D}) & D(\mathcal{D}) \end{bmatrix} \begin{bmatrix} e(t) \\ \dot{e}(t) \end{bmatrix}.$$

Then, the closed-loop system is

$$\begin{bmatrix} \dot{e}(t) \\ \ddot{e}(t) \end{bmatrix} = \begin{bmatrix} 0 & I \\ -kL_e(\mathcal{D}) & -kL_e(\mathcal{D}) \end{bmatrix} \begin{bmatrix} e(t) \\ \dot{e}(t) \end{bmatrix}$$

*Proof:*

# Stability of the closed-loop system

# Stability of the closed-loop system

*continued:*

# Dynamics of the closed-loop system

**Theorem**

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & I \\ -kL(\tilde{\mathcal{D}}) & -kL(\tilde{\mathcal{D}}) \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ kD(\mathcal{D}) & kD(\mathcal{D}) \end{bmatrix} \begin{bmatrix} z_{\text{ref}}(t) \\ \dot{z}_{\text{ref}} \end{bmatrix}$$

# Conclusion

1. Shape-based control (Take points $\Xi$ and control towards them)

# Conclusion

1. Shape-based control (Take points $\Xi$ and control towards them)
2. Relative state-based control (Define relative states $x_i - x_j$ and control them so $\|x_i - x_j\| = d_{ij}$.

# Conclusion

1. Shape-based control (Take points $\Xi$ and control towards them)
2. Relative state-based control (Define relative states $x_i - x_j$ and control them so $\|x_i - x_j\| = d_{ij}$.
3. Single/double integrator versions

# Homework: Extensions of these ideas

1. Internal dynamics: $\dot{x}_i = ax(t) + bu(t)$

# Homework: Extensions of these ideas

1. Internal dynamics: $\dot{x}_i = ax(t) + bu(t)$
2. Unicycle control: $r_i(t) = x_i(t) + jy_i(t)$

# Notifications

Reading Assignment

- Mesbahi & Egerstedt 6.2-6.6

# Notifications

## Reading Assignment

- Mesbahi & Egerstedt 6.2-6.6
- Mesbahi & Egerstedt 7.1-7.4

# Notifications

## Reading Assignment

- Mesbahi & Egerstedt 6.2-6.6
- Mesbahi & Egerstedt 7.1-7.4

## Exercise Session

Coverage Control with **Alberto**