# Formation control

REPLAN team

April 29, 2025

# Outline

# Outline

# What is a formation?

Aircraft formation

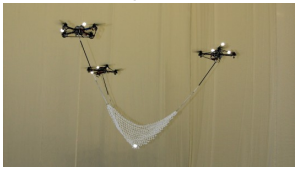
"Firefly" drone fromations

- a number of individual physical agents
- a group of agents displaying spatial pattern → inter-agents interactions
- interaction (often) 'local' but with desired global behavior

*More than just a collection, a formation has structure and purpose*

# Why formations?

Transportation



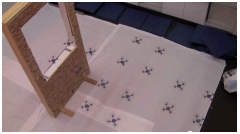
Search & Rescue

- *Coordinated Action:* Enabling a group to perform tasks more effectively than individual agents.
- *Enhanced Capabilities:* Achieving goals that a single robot might not be able to accomplish.
- *Robustness:* Increased resilience to failures if one member is lost.
- *Coverage:* Efficiently exploring or monitoring an area.

*Formations enable collective intelligence and action.*

# Unmanned/autonomous vehicles $\longrightarrow$ UAV formations

- Robots
  - ~~Robot manipulators~~
  - Mobile robots
    - Ground robots
      - **Wheeled robots**
      - ~~Legged robots~~
    - (Sub-)marine robots
    - **Aerial robots**

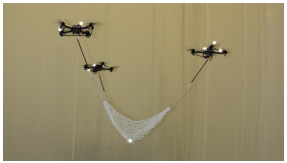


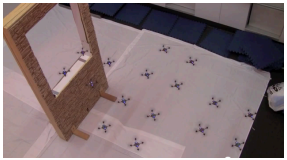
*Multi-agent systems (**MAS**) paradigm is often used in Robotic Control*

**Applications & projects aoround the world:**

- DGA - UAV for interception
- CEA - rescue drone
- precision agriculture
- surveillance(military, police, etc.)
- NTNU - autonomous ferries (USV)
- FEUP - MarineUAS (UAVs), NOPTILUS(UUVs)

# UAV formations: key task and motion tasks

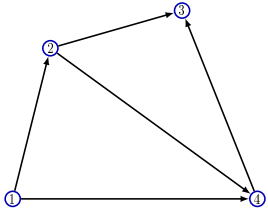




- main application : **localization of 'targets'**
- formation should be autonomous, or controlled through a single leader vehicle
- formation of 3 to 6 vehicles are typically, but... there are also swarms
- key 'motion tasks':
  - **formation shape control** (acquiring and maintaining shape)
  - moving formation as a whole in a certain direction (requiring **velocity consensus** or **flocking**)

# Formations - 'formal' definition

- A formation is a collection of agents ('point' agents) in two or three dimensional space
- A formation is **rigid** if the distance between each pair of agents cannot change over time
- In a *rigid* formation, normally only *some* distances should be explicitly maintained.




*Formation Control: It's All About Connections (Represented by **Graphs**!)*

# Outline

# Terminology

**Definition**

A graph $G$ is a pair $(V, E)$, where:

- $V$ - the set of **nodes**/points/*vertices* $V = \{v_1, \ldots, v_N\}$
- $E$ - the set of **edges**/arcs $E = \{(v_i, v_j) | v_i, v_j \in V\}$

**Node**/*Vertex*:

- fundamental/basic unit
- node (computer science), actor (sociology), "site" (physics)
- the set of nodes of a graph: $V(G)$ or $N(G)$

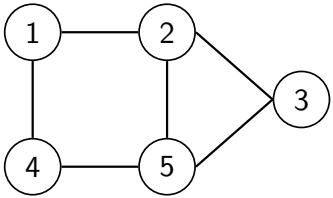
**Edge** (Arc):

- "line" connecting 2 nodes
- connection/link (computer science), bond (physics), link/tie (sociology).
- the set of arcs of a graph: $E(G)$ or $A(G)$

# Graphs - Properties 1/4

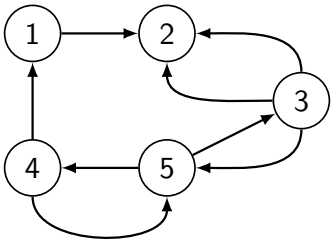
**Orientation of the edges** ⟵ transfer of energy/information/etc.

Undirected Graphs




- $V(G) = \{1, 2, 3, 4, 5\}$
- $E(G) = \{(1, 2), (1, 4), (2, 3), (2, 5), (3, 5), (4, 5)\}$
- $(i, j) = (j, i), \forall i, j \in V(G)$

Directed Graphs

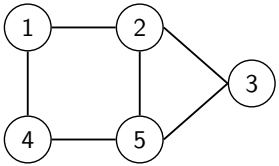



- $V(G) = \{1, 2, 3, 4, 5\}$
- $E(G) = \{(1, 2), (4, 1), (2, 3), (3, 2), (2, 5), (3, 5), (4, 5), \ldots\}$
- $(i, j) \neq (j, i), \forall i, j \in V(G)$

# Graphs - Properties 2/4

**Degree** of a node: the number of edges connected to that node

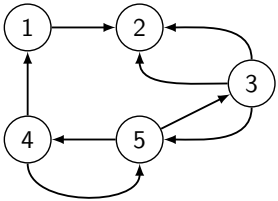
Undirected Graphs




- $V(G) = \{1, 2, 3, 4, 5\}$
- $\mathrm{k}(2) = 3$
- $\mathrm{k}(4) = 2$

$$\sum_{i \in V(G)} \mathrm{k}(i) = 2|E|$$

Directed Graphs



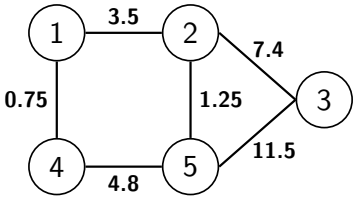

- $\mathrm{k}_{in}(2) = 3$
- $\mathrm{k}_{out}(2) = 0$
- $\mathrm{k}(2) = \mathrm{k}_{in}(2) + \mathrm{k}_{out}(2) = 3$

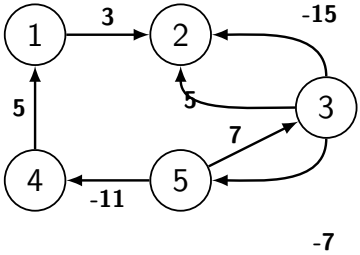
$$\sum_{i \in V(G)} \mathrm{k}_{in}(i) + \sum_{i \in V(G)} \mathrm{k}_{out}(i) = 2|E|$$

# Graphs - Properties 3/4

**Weighted Graphs**:

- $G = (V, E, f)$ where $f : E \mapsto \mathbb{R}$, $f(i, j) = w_{ij} \in \mathbb{R}$
- alternatively, $W = \left[w_{ij}\right]_{1<i,j<|V|} \in \mathbb{R}^{|V|\times|V|}$



# Graphs - Properties 4/4

**Definition**

A **walk** in the graph $G$ is a sequence of nodes with the property that any two consecutive nodes are connected by an edge.

**Classification** :

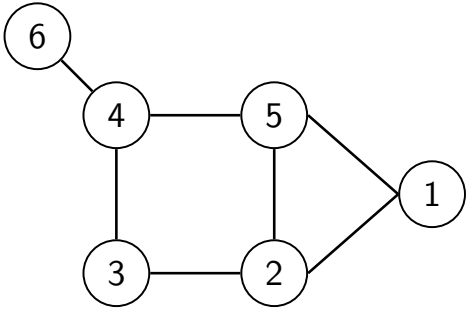
- based on edges
  - simple: all edges in the walk are distinct (no edge is repeated);
  - composed: some edges can be repeated multiple times.
- based on nodes:
  - elementary: all nodes are pairwise distinct;
  - non-elementary: nodes can be repeated.

***Path*** = simple elementary walk

**Definition**

A **cycle** in the graph $G$ is a simple walk in which the first node coincides with the last one.

# Paths within a graph - Examples

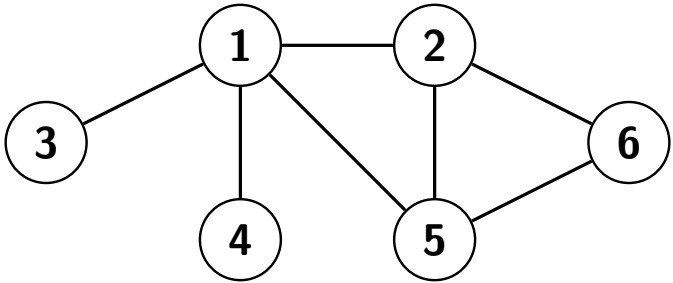



- walks:
  - 1-2-3-5-2-1-6
  - 2-5-4-1-2-3-5-2
- **paths**:
  - 1-2-3-5-4
  - 6-1-2-3
- cycles:
  - 2-3-5-2
  - 1-2-5-4-1

# Connectivity in Graphs

**Definition**

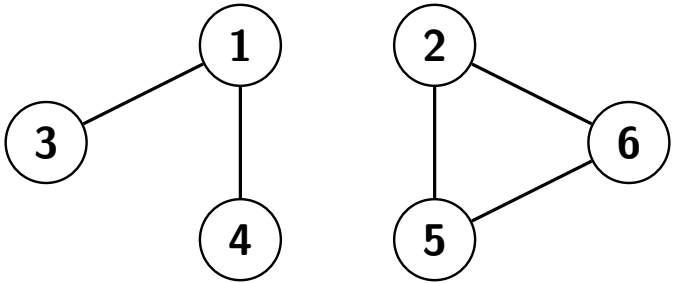
A graph $G = (V, E, f)$ is **connected** if $\forall i, j \in V$ there exists at least one path from $i$ to $j$ in the graph.



# Connectivity in Graphs

**Definition**

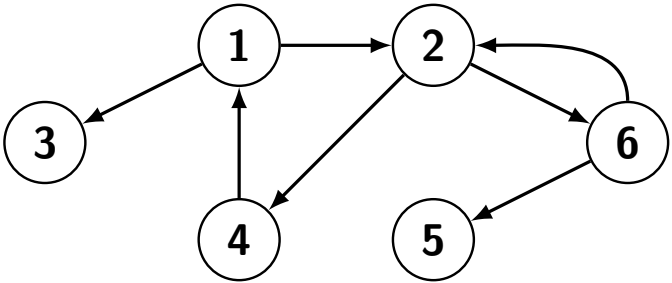
A graph $G = (V, E, f)$ is **connected** if $\forall i, j \in V$ there exists at least one path from $i$ to $j$ in the graph.



# Connectivity in Graphs

**Definition**

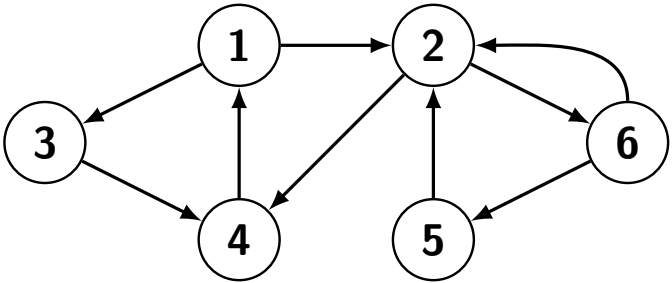
A graph $G = (V, E, f)$ is **connected** if $\forall i, j \in V$ there exists at least one path from $i$ to $j$ in the graph.



# Connectivity in Graphs

**Definition**

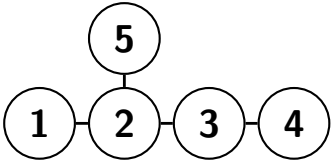
A graph $G = (V, E, f)$ is **connected** if $\forall i, j \in V$ there exists at least one path from $i$ to $j$ in the graph.

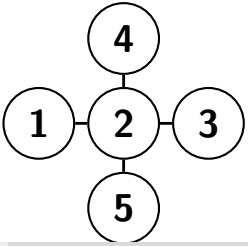


# Elementary Topologies 1/3

## Trees

- connected acyclic graphs
- any two nodes have exactly one path between them

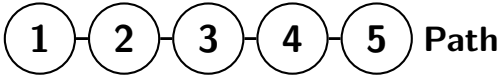



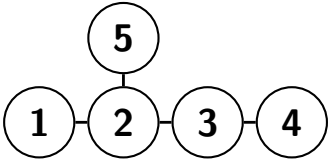
# Elementary Topologies 1/3

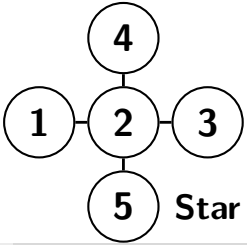
## Trees

- connected acyclic graphs
- any two nodes have exactly one path between them




Path



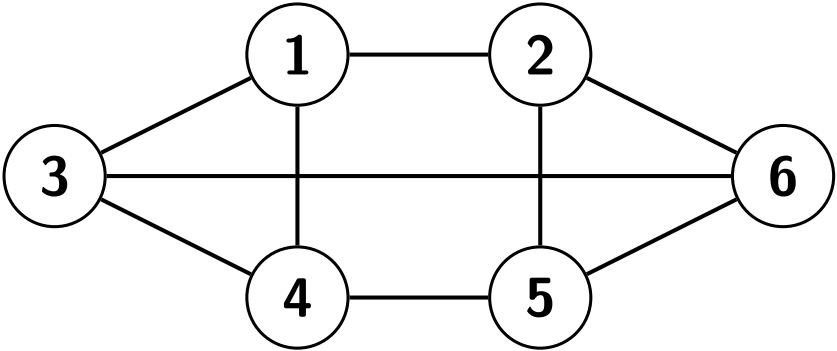




Star

# Elementary Topologies 2/3

## Regular Graphs

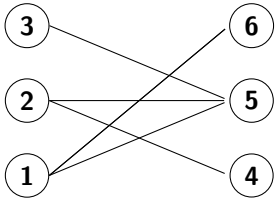
- connected graphs
- all nodes have the same degree ($k(i) = k, \forall i \in V$)



# Elementary Topologies 3/3

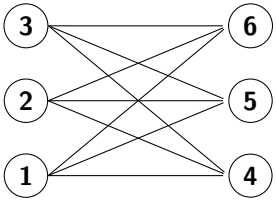
## Bipartite Graphs

- $V$ can be partitioned into $V_1$ and $V_2$ such that $\forall (u, v) \in E$ we have
  - either $u \in V_1$ and $v \in V_2$
  - or $v \in V_1$ and $u \in V_2$




## Complete Graphs

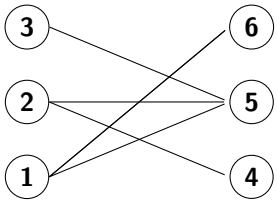
- any two nodes are adjacent
- $\forall i \neq j \in V, \exists (i, j) \in E$
- has $\frac{n(n-1)}{2}$ edges



# Elementary Topologies 3/3

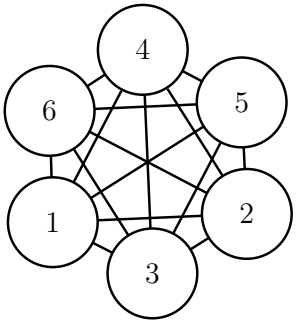
## Bipartite Graphs

- $V$ can be partitioned into $V_1$ and $V_2$ such that $\forall (u, v) \in E$ we have
  - either $u \in V_1$ and $v \in V_2$
  - or $v \in V_1$ and $u \in V_2$




## Complete Graphs

- any two nodes are adjacent
- $\forall i \neq j \in V, \exists (i, j) \in E$
- has $\frac{n(n-1)}{2}$ edges



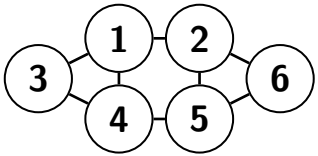
# Subgraphs and Spanning Graphs

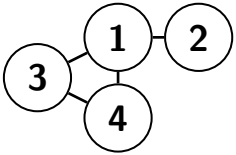
**Definition**

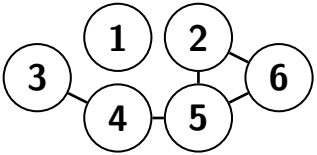
A graph $G_1 = (V_1, E_1)$ is a **subgraph** of $G = (V, E)$ if $V_1 \subset V$ and $E_1 = E(V_1)$.

**Definition**

A graph $G_1 = (V_1, E_1)$ is a **spanning graph** of $G = (V, E)$ if $V_1 = V$ and $E_1 \subset E$.



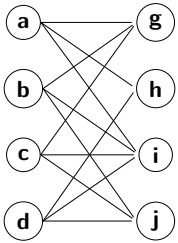
# Graph Isomorphism

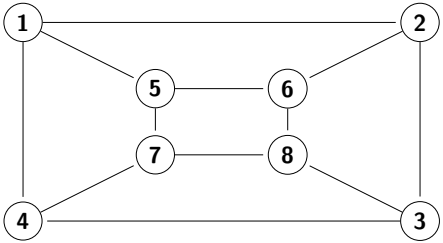
- 2 graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- a bijective function: $f : V_G \mapsto V_H$

**Definition**

The function $f$ is an **isomorphism** if $\forall (u, v) \in V_G$ we have $(f(u), f(v)) \in V_H$

- if the isomorphism $f$ exists, then $G$ and $H$ are called isomorphic



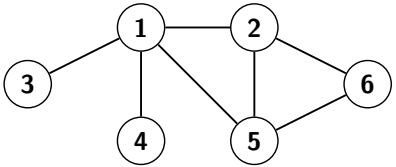




- the graphs look *very* different
- but they are isomorphic, with $f$:

f(a)=1 f(b)=6
f(c)=8 f(d)=3
f(g)=5 f(h)=2
f(i)=4 f(j)=7

# Graph Representation - Matrices

- *incidence* matrix
  - $V \times E$
  - the position $(node, arc)$ contains $0$ or $1$ (alternatively, the weight)
- **adjacency** matrix
  - $V \times V$
  - the position $(i, j)$ contains $0$ or $1$ (alternatively, the weight)

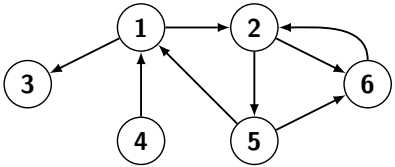



$$M_i = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_a = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

# Graph Representation - Matrices

- *incidence* matrix
  - $V \times E$
  - the position $(node, arc)$ contains $0$ or $1$ (alternatively, the weight)
- **adjacency** matrix
  - $V \times V$
  - the position $(i, j)$ contains $0$ or $1$ (alternatively, the weight)

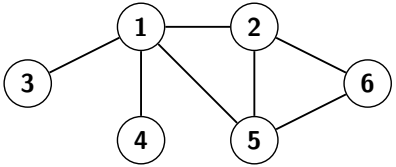



$$M_i = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_a = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Graph Representation - Lists

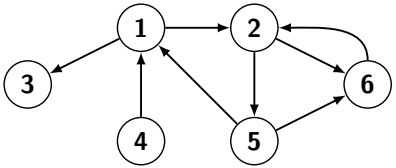
- list of edges
  - pairs of the form $(i, j) \in E$ + other data (e.g., weight, flow)
  - does *not* have a linked list structure
- **adjacency** list
  - a vector of $|V|$ lists, one for each node in $V$
  - $\forall u \in V$ the list $\mathrm{ADJ}(u)$ contains all nodes adjacent to $u$




$\mathrm{ADJ}(1):$ 2 3 4 5
$\mathrm{ADJ}(2):$ 1 5 6
$\mathrm{ADJ}(3):$ 1
$\mathrm{ADJ}(4):$ 1
$\mathrm{ADJ}(5):$ 1 2 6
$\mathrm{ADJ}(6):$ 2 5

# Graph Representation - Lists

- list of edges
  - pairs of the form $(i, j) \in E$ + other data (e.g., weight, flow)
  - does *not* have a linked list structure
- **adjacency** list
  - a vector of $|V|$ lists, one for each node in $V$
  - $\forall u \in V$ the list $\mathrm{ADJ}(u)$ contains all nodes adjacent to $u$




$\mathrm{ADJ}(1)$ : 2 3
$\mathrm{ADJ}(2)$ : 5 6
$\mathrm{ADJ}(3)$ :
$\mathrm{ADJ}(4)$ : 1
$\mathrm{ADJ}(5)$ : 1 6
$\mathrm{ADJ}(6)$ : 2

# Adjacency Matrix vs. Adjacency List

## Adjacency Matrix

- efficient if the matrix is "dense" (i.e., large number of edges)
- it is determined if $(i, j) \in E$ in $O(1)$ steps.
- traversing the list of edges from $\mathrm{ADJ}(i)$ in $O(n)$.

## Adjacency List

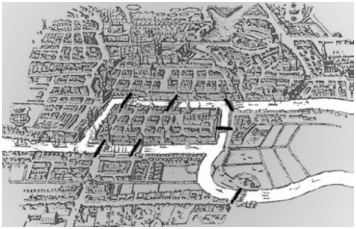
- efficient if the matrix is "sparse."
- it is determined if $j \in \mathrm{ADJ}(i)$ in $|\mathrm{ADJ}(i)|$ steps
- traversing the list of edges from $\mathrm{ADJ}(i)$ in $|\mathrm{ADJ}(i)|$ steps

# Konigsberg bridge problem 1/4

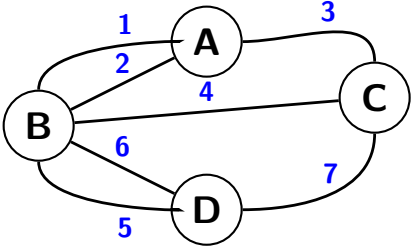
**Königsberg Bridge Problem**

Located in Prussia (1525-1947), the city of Königsberg spanned the Pregel River, encompassing its banks and two central islands. These **four** distinct land areas were interconnected by **seven** bridges.

*Is there a single walk that crosses all 7 bridges exactly once each?*



# Konigsberg bridge problem 2/4

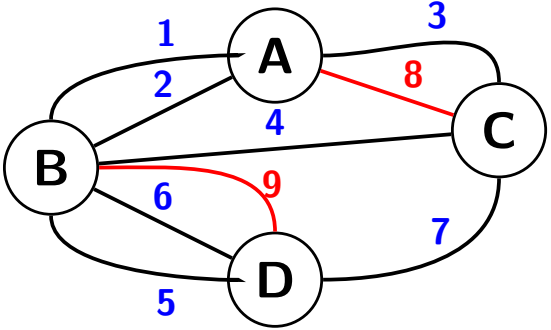



**Problem**

Is there a walk through the graph that starts at $A$ and ends at $A$, passing through each edge exactly once?

- such a walk = Eulerian cycle - *because Euler solved the puzzle*

# Konigsberg bridge problem 3/4

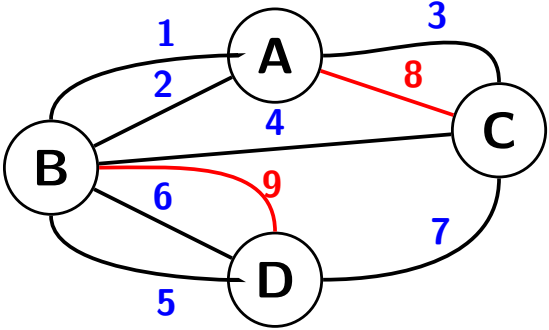



**Solution - Euler**

By adding 2 "bridges", the walk satisfying the problem constraints is:

$$A, 1, B, 5, D, 6, B, 4, C, 8, A, 3, C, 7, D, 9, B, 2, A$$

# Konigsberg bridge problem 4/4 - solution?

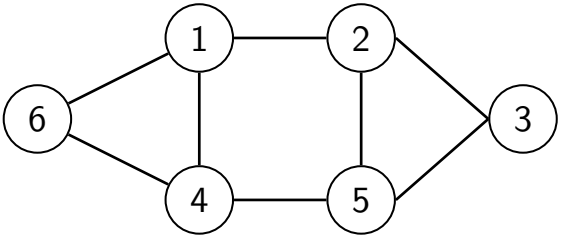



**Theorem**

An (undirected) graph has an Eulerian circuit if and only if:

1. the degree of every node is **even**
2. the graph is connected

# Shortest Path problem



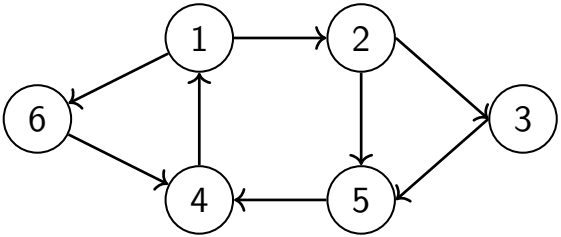

**Problem**

Let $G = (V, E)$ be a graph and two nodes $s$ - source and $t$ - destination.

*What is the shortest path through the graph from $s$ to $t$?*

- in practice, various criteria (e.g., Euclidean distance)
- multitude of algorithms (e.g., Dijkstra, A*, Bellman-Ford )

# Shortest Path problem




**Problem**

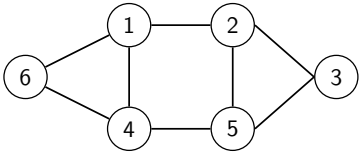
Let $G = (V, E)$ be a graph and two nodes $s$ - source and $t$ - destination.

*What is the shortest path through the graph from $s$ to $t$?*

- in practice, various criteria (e.g., Euclidean distance)
- multitude of algorithms (e.g., Dijkstra, A*, Bellman-Ford )

# Shortest Path problem - Categories

- **single-pair** shortest path problem
- **single-source** shortest path problem
- single-destination shortest path problem[a]
- **all-pairs shortest path problem**

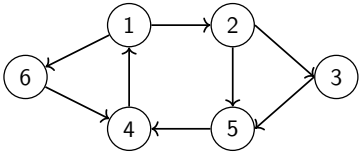



[a] This can be reduced to the single-source shortest path problem by reversing the arcs in the directed graph.

Algorithms designed for more general cases are usually significantly more efficient than simply running a "single-pair" type algorithm repeatedly.

# Shortest Path problem - Categories

- **single-pair** shortest path problem
- **single-source** shortest path problem
- single-destination shortest path problem[a]
- **all-pairs shortest path problem**




[a] This can be reduced to the single-source shortest path problem by reversing the arcs in the directed graph.

Algorithms designed for more general cases are usually significantly more efficient than simply running a "single-pair" type algorithm repeatedly.

# Shortest Path problem - Algorithms

**Important** algorithms (at least, some of them...):

- **Dijkstra's algorithm**
  - *single-source shortest path*
  - positive weights (**!**)
- **Bellman–Ford algorithm**
  - *single-source shortest path*
  - weights can be negative.
- **A* search algorithm**
  - *single-pair shortest path*
  - uses *heuristic methods* to "speed up".
- **Floyd–Warshall algorithm**
  - *all pairs shortest paths*
  - with positive or negative edge weights (but with no negative cycles)
- **Johnson's algorithm**
  - *all pairs shortest paths*
  - generally faster than Floyd–Warshall for sparse graphs
- **Viterbi algorithm**
  - *the shortest **stochastic** path problem*
  - with an additional weight for each vertex

# Shortest Path problem - Algorithms - Complexity

**Complexity** - depending on the number of:

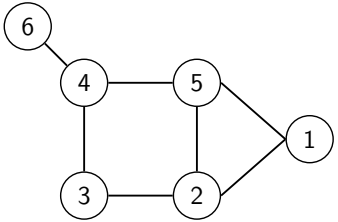
- edges - $O(g(E))$
- nodes - $O(g(V))$
- nodes and edges - $O(g(V, E))$

| Weights | Complexity | Algorithm (*undirected case with positive weights*) |
|---|---|---|
| $\mathbb{R}_+$ | $O(V^2)$ | Dijkstra |
| $\mathbb{R}_+$ | $O((E+V)\log V)$ | Johnson (binary heap) |
| $\mathbb{R}_+$ | $O(E+V\log V)$ | Fredman & Tarjan (Fibonacci heap) |
| $\mathbb{N}$ | $O(E)$ | Thorup |

Choosing the algorithm based on the characteristics of the graph. !!!!!

# Elements of Topology/Metrics

- geodesic path: the shortest path between 2 nodes
- topological distance: the number of edges in the geodesic path
  - $\forall p, q \in V$ topological distance is $d_{p,q}$
  - distance matrix $D = \left[d_{ij}\right]_{i,j \in V}$




$$D = \begin{bmatrix} 0 & 1 & 2 & 2 & 1 & 3 \\ 1 & 0 & 1 & 2 & 1 & 3 \\ 2 & 1 & 0 & 1 & 2 & 2 \\ 2 & 2 & 1 & 0 & 1 & 1 \\ 1 & 1 & 2 & 1 & 0 & 2 \\ 3 & 3 & 2 & 1 & 2 & 0 \end{bmatrix}$$

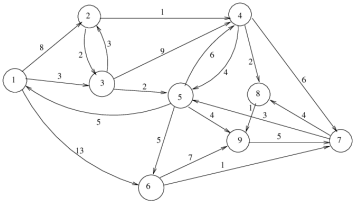
- diameter: the length (number of edges) of the longest geodesic path between any two nodes.
  - alternatively, diameter = average of geodesic distances
  - but they can be considered as distinct metrics

# To sum up: Graph-based descriptions

**Definitions**

A (*un*-)directed graph $\mathcal{G}$ is defined as a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of nodes, and $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$ denotes the set of (*un*-)ordered pairs of the nodes, called edges.

- A (*directed*) path is a sequence of ordered edges of the form $(v_i, v_{i1}), (v_{i1}, v_{i2}), \ldots, (v_{i(k-1)}, v_{ik}), (v_{ik}, v_j)$ where $v_{ik} \in \mathcal{V}$

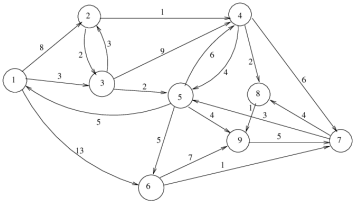


# To sum up: Graph-based descriptions

**Definitions**

A (*un*-)directed graph $\mathcal{G}$ is defined as a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of nodes, and $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$ denotes the set of (*un*-)ordered pairs of the nodes, called edges.

- A (*directed*) path is a sequence of ordered edges of the form $(v_i, v_{i1}), (v_{i1}, v_{i2}), \ldots, (v_{i(k-1)}, v_{ik}), (v_{ik}, v_j)$ where $v_{ik} \in \mathcal{V}$
- A (*di*-)graph is called (strongly) connected if there is a (*directed*) path from every node to every other node.

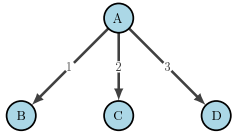


# To sum up: Graph-based descriptions

**Definitions**

A (*un*-)directed graph $\mathcal{G}$ is defined as a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of nodes, and $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$ denotes the set of (*un*-)ordered pairs of the nodes, called edges.

- A (*directed*) path is a sequence of ordered edges of the form $(v_i, v_{i1}), (v_{i1}, v_{i2}), \ldots, (v_{i(k-1)}, v_{ik}), (v_{ik}, v_j)$ where $v_{ik} \in \mathcal{V}$
- A (*di*-)graph is called (strongly) connected if there is a (*directed*) path from every node to every other node.
- A directed tree is a digraph, where every node has exactly one parent except for one node(called the root). The root has a directed path to every other node.

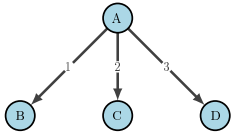


# To sum up: Graph-based descriptions

**Definitions**

A (*un*-)directed graph $\mathcal{G}$ is defined as a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of nodes, and $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$ denotes the set of (*un*-)ordered pairs of the nodes, called edges.

- A (*directed*) path is a sequence of ordered edges of the form $(v_i, v_{i1}), (v_{i1}, v_{i2}), \ldots, (v_{i(k-1)}, v_{ik}), (v_{ik}, v_j)$ where $v_{ik} \in \mathcal{V}$
- A (*di*-)graph is called (strongly) connected if there is a (*directed*) path from every node to every other node.
- A directed tree is a digraph, where every node has exactly one parent except for one node(called the root). The root has a directed path to every other node.
- A (*directed*) spanning tree of a digraph is a directed tree formed by graph edges that connect all the nodes of the graph.

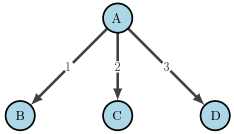


# To sum up: Graph-based descriptions

**Definitions**

A (*un*-)directed graph $\mathcal{G}$ is defined as a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of nodes, and $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$ denotes the set of (*un*-)ordered pairs of the nodes, called edges.

- A (*directed*) path is a sequence of ordered edges of the form $(v_i, v_{i1}), (v_{i1}, v_{i2}), \ldots, (v_{i(k-1)}, v_{ik}), (v_{ik}, v_j)$ where $v_{ik} \in \mathcal{V}$
- A (*di*-)graph is called (strongly) connected if there is a (*directed*) path from every node to every other node.
- A directed tree is a digraph, where every node has exactly one parent except for one node(called the root). The root has a directed path to every other node.
- A (*directed*) spanning tree of a digraph is a directed tree formed by graph edges that connect all the nodes of the graph.
- A graph has (or contains) a (directed) spanning tree if there exists a (directed) spanning tree that is a subset of the graph.



# Cooperation - Connectivity

**Axiom**

Shared information is a necessary condition for cooperation.

- e.g., *meet-for-dinner* problem

**Remark**

Typically, the interaction graph of the agents needs to be connected or, at least, have a spanning tree

Connectivity for *digraphs*:

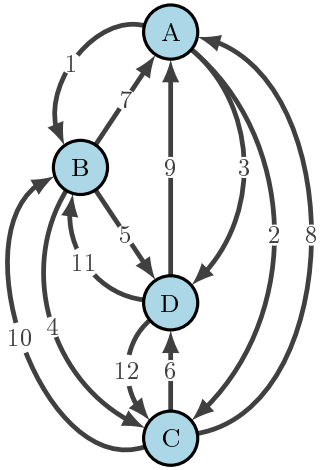
- Strong connectivity ("tare conex") = there is a path between any pair of nodes
- Weak connectivity ("slab conex") = the undirected counterpart is connected

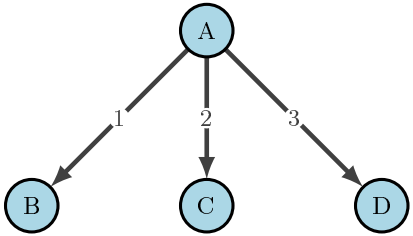
**Remark**

The existence of a spanning tree - between strong and weak connectivity.

# Spanning tree - Illustrative example

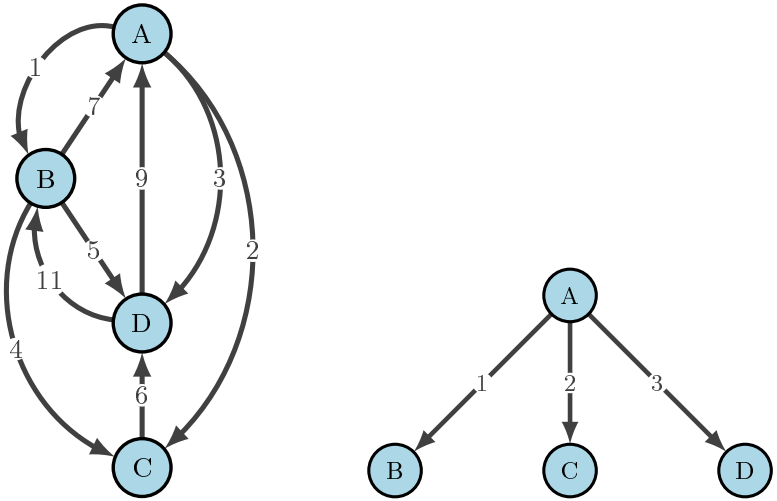
- minimal spanning tree for digraphs → **Edmond's Algorithm**



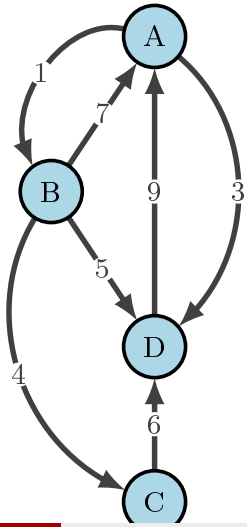
# Spanning tree - Illustrative example

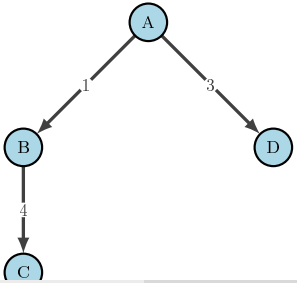
- minimal spanning tree for digraphs → **Edmond's Algorithm**



# Spanning tree - Illustrative example

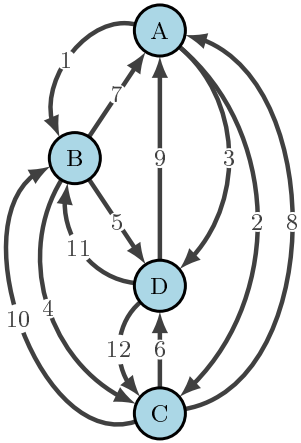
- minimal spanning tree for digraphs → **Edmond's Algorithm**

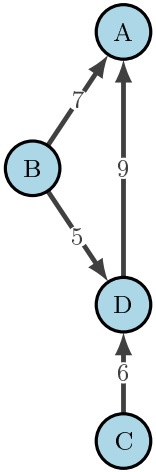


# Spanning tree - Illustrative example

- minimal spanning tree for digraphs → **Edmond's Algorithm**



# The Consensus Problem: How do de all agree?

**The Core Question**

How can a group of independent agents reliably agree on a single value or state?

How do we design systems where agreement is guaranteed?

**Challenges**:

- different starting points/views
- imperfect communication (delays, losses)
- potential failures (agents/links offline)

**The goal**

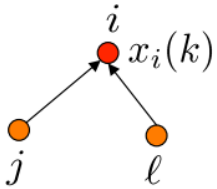
Achieve unified agreement despite the hurdles.

Why we need to solve it? $\longrightarrow$ Essential for:

- **coordinated action** (Formations, Robotics)
- data consistency (distributed Databases, *Blockchain*)
- fault tolerance (safety and security of systems)

# Basic setup

- $x_i(t)$ or $x_i(k)$ - $i$-th agent "position" at $t$ or $k$
- $\mathcal{N}_i$ - the neighbors of $i$




- available information for the agent $i$ :

$$\mathcal{I}_i^c(k) = \{x_j(k) | j \in \mathcal{N}_i(k)\} \longleftarrow \text{communication}$$

$$\mathcal{I}_i^r(k) = \{x_i(k) - x_j(k) | j \in \mathcal{N}_i(k)\} \longleftarrow \text{sensors}$$

- update rule : $x_i(k+1) = F_i(x_i(k), \mathcal{I}_i(k))$ or $\dot{x}_i(t) = F_i(x_i(t), \mathcal{I}_i(t))$

# Notations - unweighted (di)graphs

- degrees matrix: $D = \mathrm{diag}\left(\deg(v_1), \deg(v_2), \ldots, \deg(v_N)\right)$
- adjacency matrix:

$$A = \left[a_{ij}\right]_{i,j}, \text{ where } a_{ij} = \begin{cases} 1, & \exists(v_i, v_j) \\ 0, & \text{otherwise} \end{cases}$$

- incidence matrix

$$\mathcal{I} = \left[\iota_{ij}\right]_{i,j}, \text{where } \iota_{ij} = \begin{cases} 1, & \exists(v_i, v_j) \\ -1, & \exists(v_j, v_i) \\ 0, & \text{otherwise} \end{cases}$$

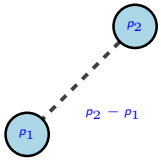
- **Laplacian matrix** :

$$\mathcal{L} = \underbrace{D - A}_{\text{for undirected}} = \mathcal{I}\mathcal{I}^\top$$

- $\mathcal{L}$ symmetric and positive semi-definite for undirected graphs

# Canonical problem: *Rendezvous*

- a collection of $N$ mobile agents
- $x_i = p_i, \forall i = 1 : N$ (state $\sim$ position)
- agent $i$ can measure only **relative** distance to/from $j \in \mathcal{N}_i$
- agents do not have global information (e.g., absolut coordinate)




**Problem**

All agents need to reach same state: $p_{\text{rendezvous}}$, i.e., $\exists T > 0$ for which $p_1(T) = p_1(T)$

- let $N = 2 \longrightarrow$ the graph is connected
- $p_{\text{rendezvous}} \in$ line segment defined by $p_1$ and $p_2$ (?)
- update rule: $\dot{p}_1 = \gamma_1(p_1 - p_2) \qquad \dot{p}_2 = \gamma_2(p_2 - p_1)$

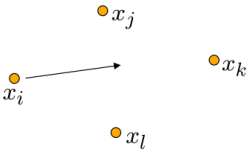
**Questions**

- How we can force $p_{\text{rendezvous}}$ to be the midpoint?
- $p_i(t) =?$ for $t > T$

# Canonical problem: *Rendezvous*

- general case $N$ agents

$$\dot{p}_i = -\gamma \sum_{j \in N_i} (p_i - p_j)$$




**Question**

Towards which position/coordinates will they head?

Remark

If (and only if) the graph is **connected**, the **consensus equation** leads agents to the same state value:

$$\lim_{t \to \infty} p_i(t) = \bar{p} = \frac{1}{N} \sum_{j=1}^{N} p_j(0)$$

# Why is the Laplacian important?

- in the **consensus equation** let $\gamma = 1$:

$$\dot{p}_i = -\sum_{j \in \mathcal{N}_i} (p_i - p_j), i = 1, \ldots, N$$

- or, equivalently:

$$\left.\begin{aligned} \dot{p}_i &= -\deg(n_i) p_i + \sum_{j=1}^{N} a_{ij} p_j \\ p &= \begin{bmatrix} p_1 & p_2 & \ldots & p_N \end{bmatrix}^\top \in \mathbb{R}^{dN} \end{aligned}\right\} \Rightarrow \dot{p} = -\mathcal{L} p$$

# Why is the Laplacian important?

- in the **consensus equation** let $\gamma = 1$:

$$\dot{p}_i = -\sum_{j \in \mathcal{N}_i} (p_i - p_j), i = 1, \ldots, N$$

- or, equivalently:

$$\left.\begin{aligned} \dot{p}_i &= -\deg(n_i) p_i + \sum_{j \in \mathcal{N}_i} a_{ij} p_j \\ p &= \begin{bmatrix} p_1 & p_2 & \ldots & p_N \end{bmatrix}^\top \in \mathbb{R}^{dN} \end{aligned}\right\} \Rightarrow \dot{p} = -\mathcal{L} p$$

- we have an autonomous system, whose convergence and stability **depends on the Laplacian**.

# Laplacian - properties

- If the graph is connected then:

$$\begin{aligned} \mathrm{eig}(\mathcal{L}) &= \{\lambda_1, \ldots, \lambda_N\}, & \text{and } 0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_N \\ \mathrm{eigv}(\mathcal{L}) &= \{\nu_1, \ldots, \nu_N\}, & \\ \mathrm{Ker}(\mathcal{L}) &= \mathrm{span}(\nu_1), & \mathrm{Ker}(\mathcal{L}) = \mathrm{span}(\mathbf{1}) \text{ (when } \gamma = 1) \end{aligned}$$

- stability = what happens when $t \to \infty$?
- LTI (linear time-invariant) system: $\dot{x} = Ax \quad (\dot{x} = -\mathcal{L}x)$
- stability determined by: $\lambda(A) = \{\lambda_1, \ldots, \lambda_n\}$

# [Recap] Notions of equilibrium and stability

For a system given as

$$\dot{x} = Ax, \qquad x \in \mathbb{R}^n$$

a equilibrium point $x^*$ is a constant of the differential equation (i.e. a solution of the equation $\dot{x}(t) = 0$), which means that

$$Ax(t) = Ax^* = 0$$

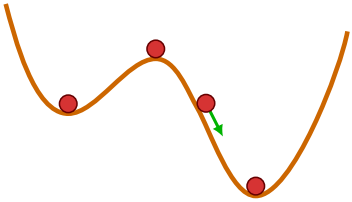
We identify the following cases:

- $rankA = n$: unique equilibrium $x^* = 0$
- $rankA < n$: a collection of equilibrium points given by the null space of $A$ ($kerA$):

$$x^* = \{x | \, x \in kerA\}$$

We may classify the equilibrium points in:

- stable
- unstable
- indifferent



**Ideally**, we are interested by the case $rankA = n$, to which corresponds $x^* = 0$.

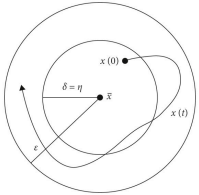
# [Recap] Stability in a linear system 1/2

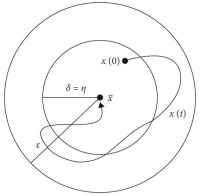
**Stability**

A system of the form $\dot{x} = f(x)$ is stable if, after a sufficiently long amount of time, the state no longer depends on the initial conditions, no matter what they are.

The definition by Lyapunov formalizes the stability as follows. The system is:

- *stable* if for every $\varepsilon > 0$, there exists $\delta > 0$ s.t. $\|x(0)\| < \delta \Rightarrow \forall t \geq 0, \|x(t)\| < \varepsilon.$
- *asymptotically stable* if it is *Lyapunov stable* and there exists $\delta > 0$ s.t. $\|x(0)\| < \delta \Rightarrow \lim_{t\to\infty} \|x(t)\| = 0.$

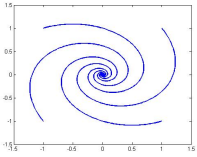


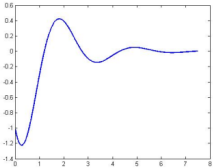



# Stability in a linear system 2/2

**Asimptotically stable**

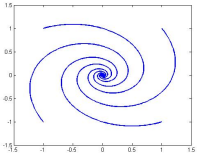
$$\mathrm{Re}(\lambda_i) < 0, \forall i \rightarrow \lim_{t\to\infty} x(t) = 0$$

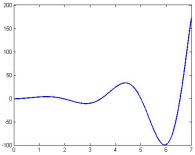




**Unstable**

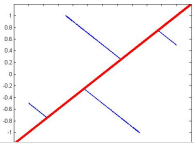
$$\exists i \text{ a.î. } \mathrm{Re}(\lambda_i) > 0 \rightarrow \lim_{t\to\infty} x(t) = \infty$$





**Critical stable** (e.g., consensus equation)

$$0 = \lambda_1 > \lambda_2 \geq \ldots \lambda_N \rightarrow \lim_{t\to\infty} x(t) \in \mathrm{Ker}(A)$$



# Static Consensus

**Remark**

If the graph is **static and connected**, then by consensus equation all the state will reach $\mathrm{Ker}(\mathcal{L})$.

$$\mathrm{Ker}(\mathcal{L}) = \mathrm{span}(\mathbf{1}), x \in \mathrm{Ker}(\mathcal{L}) \leftrightarrow x = \begin{bmatrix} \alpha \\ \alpha \\ \alpha \\ \alpha \\ \dots \end{bmatrix}, \alpha \in \mathbb{R}$$

- the consensus/rendezvous problem is solved! $\rightarrow$ "consensus state"

$$\dot{x}_i = -\sum_{j \in N_i} (x_i - x_j) \rightarrow \lim_{t \to \infty} x_i(t) = \frac{1}{N} \sum_{j=1}^{N} x_j(0)$$

# Convergence Rate

**Fiedler eigenvalue (Algebric connectivity)**

The second smallest eigenvalue from $\lambda(\mathcal{L})$,i.e., $\lambda_2$) plays a crucial role in convergence.

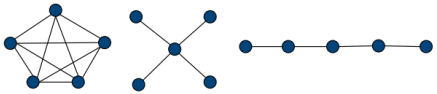
- $\lambda_2$: Fiedler eigenvalue
- convergence rate:

$$\|x(t) - \frac{1}{N}\mathbf{1}^\top x(0)\| \leq Ce^{-\lambda_2 t}$$

**Idea**

The more connected the graph, the faster the convergence (and the more information needs to be exchanged/shuffled over the network)

- complete graph: $\lambda_2 = n$
- star: $\lambda_2 = 1$
- path: $\lambda_2 < 1$



# Notations - **weighted** (di)graphs

- degrees matrix: $D = \mathrm{diag}\left(\deg(v_1), \deg(v_2), \ldots, \deg(v_N)\right)$
- adjacency matrix:

$$A = \left[a_{ij}\right]_{i,j}, \text{ where } a_{ij} = \begin{cases} 1, & \exists(v_i, v_j) \\ 0, & \text{otherwise} \end{cases}$$

- **weights matrix**:

$$W = \left[w_{ij}\right]_{i,j}, \text{ where } w_{ij} = \begin{cases} > 0, & \exists(v_i, v_j) \\ 0, & \text{otherwise} \end{cases}$$

- **Laplacian** matrix:

$$L = \left[\ell_{ij}\right]_{i,j}, \text{where } \ell_{ij} = \begin{cases} \sum\limits_{k \in \mathcal{N}_i} w_{ik}, & \text{if } i = j \\ \\ -\, w_{ij}, & \text{otherwise} \end{cases}$$

- $\mathcal{L}$ symmetric and positive semi-definite for **undirected graphs**

# Consensus protocols 1/2

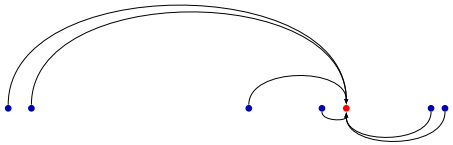
- single integrator: $\dot{p}_i = u_i$
- $p_i \in \mathbb{R}, u_i \in \mathbb{R}$
- first order consensus protocol

- the control law:
  $u_i = -\sum_{j=1}^{N} a_{ij} w_{ij} (p_i - p_j) = -\sum_{j \in \mathcal{N}_i} w_{ij} (p_i - p_j)$
- "update rule" for consensus:
  $\dot{p}_i = -\left( \sum_{j \in \mathcal{N}_i} w_{ij} \right) p_i + \sum_{j \in \mathcal{N}_i} w_{ij} p_j$

- $\xi = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix}^T$
- $\mathcal{L} = \begin{bmatrix} \sum_{j \in \mathcal{N}_1} w_{1j} & -w_{12} & \dots & -w_{1N} \\ -w_{21} & \sum_{j \in \mathcal{N}_2} w_{2j} & \dots & -w_{2N} \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}$

# Consensus protocols 1/2

- single integrator: $\dot{p}_i = u_i$
- $p_i \in \mathbb{R}, u_i \in \mathbb{R}$
- first order consensus protocol



- the control law:
  $u_i = -\sum_{j=1}^{N} a_{ij} w_{ij} (p_i - p_j) = -\sum_{j \in \mathcal{N}_i} w_{ij} (p_i - p_j)$
- "update rule" for consensus:

$$\dot{\xi} = -\mathcal{L}\xi$$

- $\xi = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix}^T$
- $\mathcal{L} = \begin{bmatrix} \sum_{j \in \mathcal{N}_1} w_{1j} & -w_{12} & \dots & -w_{1N} \\ -w_{21} & \sum_{j \in \mathcal{N}_2} w_{2j} & \dots & -w_{2N} \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}$

# Consensus protocols 1/2

- single integrator: $\dot{p}_i = u_i$
- $p_i \in \mathbb{R}, u_i \in \mathbb{R}$
- first order consensus protocol

- the control law:
  $u_i = -\sum_{j=1}^{N} a_{ij} w_{ij} (p_i - p_j) = -\sum_{j \in \mathcal{N}_i} w_{ij} (p_i - p_j)$
- "update rule" for consensus:

$$\dot{\xi} = -\mathcal{L}\xi$$

- consensus constraint: $\dot{\xi}_c = 0$
- i.e., $\mathcal{L}\xi_c = 0 \rightarrow \xi_c \in \ker(\mathcal{L})$
- $\ker(\mathcal{L}) = \mathrm{span}(\nu_1)$, but $\mathrm{Ker}(\mathcal{L}) \neq \mathrm{span}(\mathbf{1})$(!)
- $\nu = \left[\alpha_1, \ldots, \alpha_N\right]^\top$, $\alpha_i > 0$, $\sum_i \alpha_i = 1$

$$\xi_c^\star = \left( \sum_{i=1}^{N} \alpha_i \xi_i(0) \right) \mathbf{1}$$

# Consensus protocols 2/2

- double integrator: $\dot{p}_i = v_i, \dot{v}_i = u_i$
- $p_i \in \mathbb{R}, v_i \in \mathbb{R}, u_i \in \mathbb{R}$
- first order consensus protocol

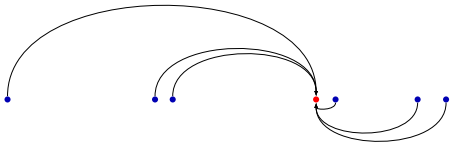
- the control law: $u_i = -\sum_{j \in \mathcal{N}_i} w_{ij}[(p_i - p_j) + \gamma(v_i - v_j)]$
- $\gamma$ - scaling factor
- the goal of consensus:

$$|p_i - p_j| \to 0,\ |v_i - v_j| \to 0,\ \textbf{for } t \to \infty$$

- $\xi = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix}^T$, $\zeta = \begin{bmatrix} v_1 & v_2 & \dots & v_N \end{bmatrix}^T$
- "update rule" for consensus: $\begin{bmatrix} \dot{\xi} \\ \dot{\zeta} \end{bmatrix} = \Gamma \begin{bmatrix} \xi \\ \zeta \end{bmatrix}$
- where: $\Gamma = \begin{bmatrix} O_n & I_n \\ -\mathcal{L} & -\gamma\mathcal{L} \end{bmatrix}$

# Consensus protocols 2/2

- double integrator: $\dot{p}_i = v_i, \dot{v}_i = u_i$
- $p_i \in \mathbb{R}, v_i \in \mathbb{R}, u_i \in \mathbb{R}$
- first order consensus protocol



- the control law: $u_i = -\sum_{j \in \mathcal{N}_i} w_{ij}[(p_i - p_j) + \gamma(v_i - v_j)]$
- $\gamma$ - scaling factor
- the goal of consensus:

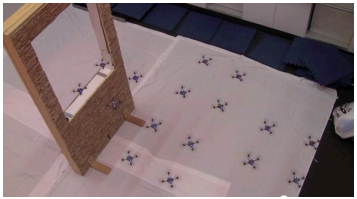
$$|p_i - p_j| \to 0,\ |v_i - v_j| \to 0,\ \textbf{for } t \to \infty$$

- $\xi = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix}^T$, $\zeta = \begin{bmatrix} v_1 & v_2 & \dots & v_N \end{bmatrix}^T$
- "update rule" for consensus: $\begin{bmatrix} \dot{\xi} \\ \dot{\zeta} \end{bmatrix} = \Gamma \begin{bmatrix} \xi \\ \zeta \end{bmatrix}$
- where: $\Gamma = \begin{bmatrix} O_n & I_n \\ -\mathcal{L} & -\gamma\mathcal{L} \end{bmatrix}$

# Outline

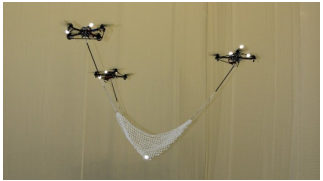
# Multi-agent formation control

- **MAS** $\longrightarrow$ control



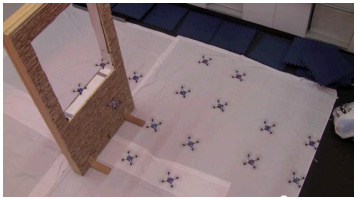
How we control MAS based on partial and relative information ***without an intervention*** of a central controller?

- **Formation control** $\longrightarrow$ aims to drive multiple agents to achieve **prescribed** constraints **on their states**



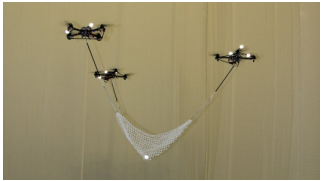
# Multi-agent formation control

- **MAS** $\longrightarrow$ control



How we control MAS based on partial and relative information ***with minimal intervention*** of a central controller?

- **Formation control** $\longrightarrow$ aims to drive multiple agents to achieve **prescribed** constraints **on their states**



# Types of formation control

Main aspects:

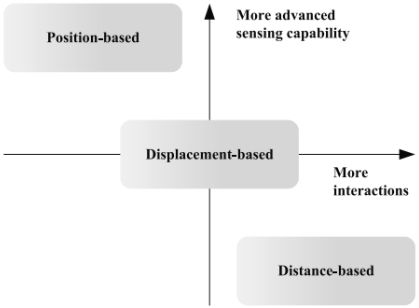
- **sensing capability**
- interaction topology

Which variables are sensed?

- local vs. global
- centralized vs. decentralized vs. distributed

**Types of formation control**:

- position-based formation control
- displacement-based formation control
- distance-based formation control



# Types of formation control

Main aspects:

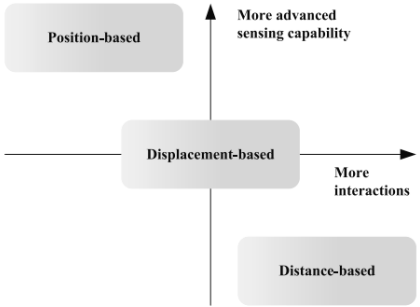
- sensing capability
- **interaction topology**

Which variables are actively controlled?

- local vs. global
- centralized vs. decentralized vs. distributed

**Types of formation control**:

- position-based formation control
- displacement-based formation control
- distance-based formation control



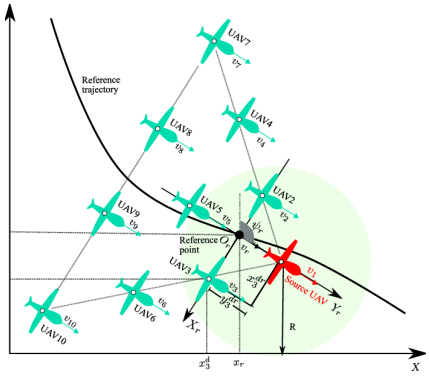
# Types of formation control

| | **Position-based** | **Displacement-based** | **Distance-based** |
|---|---|---|---|
| *Measured var.* | Positions of agents | Relative positions of $\mathcal{N}_i$ | Relative positions of $\mathcal{N}_i$ |
| *Controlled var.* | Positions of agents | Relative positions of $\mathcal{N}_i$ | Inter-agent distances |
| *Coord. system* | Global | Local(but aligned ) | Local |
| *Interaction graph* | Unnecessary | Connectivity or $\exists$ spanning tree | Rigidity or persistence |

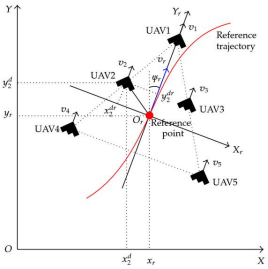
# Classifications 1/3

## w.r.t time-varying aspect

- **Formation producing problem**
  - *Objective*: agents to achieve a **prescribed** desired formation shape.
  - Approaches:
    - matrix theory based approach
    - Lyapunov based approach
    - graph rigidity approach
    - receding horizon approach
- **Formation tracking problems**
  - *Objective*: **Prescribed** reference trajectories for agents to track.
  - Approaches:
    - matrix theory based approach
    - potential function based approach
    - Lyapunov based approach
    - etc.



# Classifications 2/3



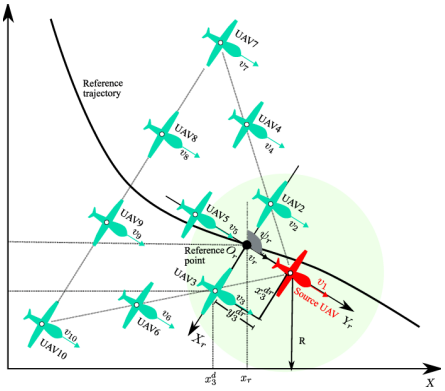

## w.r.t fundamental idea in control scheme

- **Leader-follower**
  - at least one agent is “leader”, others “followers”
  - “followers” track the position of the “leader” with some *offsets*
- **Behavioral approach**
  - various prescribed behaviors for agents
  - e.g., obstacle and collision avoidance
- **Virtual structure approach**
  - formation = “ a single object” - a virtual structure
  - a reference for the structure is given - desired formation is implicit for agents

# Classifications 3/3

**w.r.t how the formation is prescribed**

- **Morphous formation control**
  - explicit prescribed
  - sub-categories:
    - position - based formation control
    - displacement - based formation control
    - distance - based formation control
- **Amorphous formation control**
  - implicitly
  - various behaviors generate the formation
  - see *behavioral approach*



# General problem setup

- $N$ agents

$$\begin{cases} \dot{x}_i & = f_i(x_i, u_i) \\ y_i & = g_i(x_1, \ldots, x_N), i = 1 : N \\ z_i & = h_i(x_i) \end{cases}$$

- $x_i \in \mathbb{R}^{n_i}$ - state
- $u_i \in \mathbb{R}^{p_i}$ - control input
- $y_i \in \mathbb{R}^{q_i}$ - measurement
- $z_i \in \mathbb{R}^r$ - output of agent $i$

We consider:

- $z^* \in \mathbb{R}^{rN} \longrightarrow z^* = z^*(t)$ - output of MAS
- $F : \mathbb{R}^{rN} \mapsto \mathbb{R}^M$ - given
- the desired formation: $F(z) = F(z^*)$

**Problem**

Design a control law by using only measurements $y_i$ such that the set:

$$E_{z^*} = \{ x \mid F(z) = F(z^*) \}$$

is asymptotically stable w.r.t. the multi-agent system.

# Particular cases

position-based

- $y_i$ - **absolute** (global) variables
- $F(z) := z$
- $z \longrightarrow z^\star$
- variant to any modification of z

displacement-based

- $y_i$ - **relative** (global) variables
- $\left[\dots (z_i - z_j)^\top \dots\right]^\top$
- $F(z) \longrightarrow F(z^\star)$
- invariant to the translation of $z$

distance-based

- $y_i$ - **relative** (local) variables
- $\left[\dots ||z_i - z_j||^\top \dots\right]^\top$
- $F(z) \longrightarrow F(z^\star)$
- invariant to the translation and rotation of $z$

**Remark**

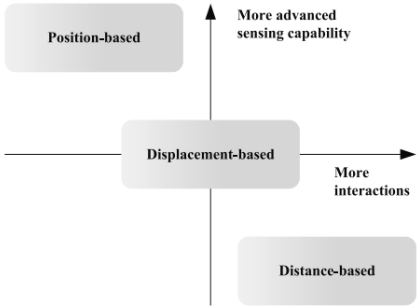
If $z$ - position vector, then $z^\star$ - desired position w.r.t. the global coordinates system.

# Outline

# Position-based formation control

Requirements:

- **Sensing capability**:
  - a global coordinate system
  - sense their absolute positions
- **Interaction topology**
  - is not necessarily required
  - but the interactions may help (e.g., enhancing control perfomance or addresing additional objectives)

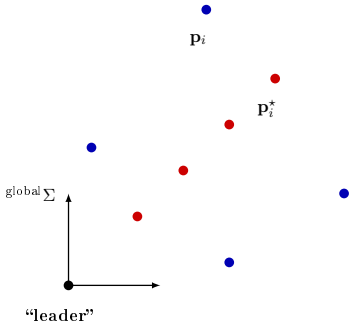



**Key aspect**

The interaction graph of the agents typically needs to be connected or have a spanning tree.

# Problem statement & Research directions

- $y_i$ - **absolute** (global) variables
- $F(z) := z$
- $z \longrightarrow z^\star$
- variant to any modification of z



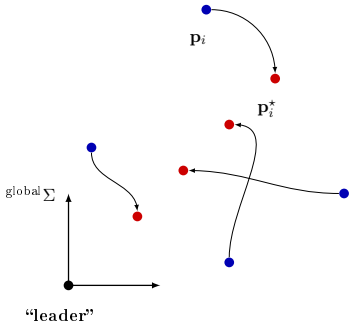

## Research directions

- enhance performance of formation control via interactions $\longrightarrow$ decentralized control
- use of a **global coordinator** $\longrightarrow$ centralized control [1]
  - take feedback from agents
  - provide the agents with appropiate commands

[1] useful when agents have limited actuation and/or sensing capabilities

# Problem statement & Research directions

- $y_i$ - **absolute** (global) variables
- $F(z) := z$
- $z \longrightarrow z^\star$
- variant to any modification of z




## Research directions

- enhance performance of formation control via interactions $\longrightarrow$ decentralized control
- use of a **global coordinator** $\longrightarrow$ centralized control [1]
  - take feedback from agents
  - provide the agents with appropiate commands

[1] useful when agents have limited actuation and/or sensing capabilities

# Global coordinator - example

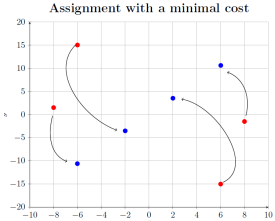
Idea: Static task asignment (source - target), i.e., assign agent $j$ to target position $p_{f,i}$ with a cost, e.g., $c_{ij} = (p_j - p_{f,i})^\top \mathbf{P}(p_j - p_{f,i})$:

$$\min_{\delta} \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}\delta_{ij}$$

$$\text{subject to: } \begin{cases} \sum_{i=1}^{N} \delta_{ij} = 1, & \forall\ i, \\ \sum_{j=1}^{N} \delta_{ij} = 1, & \forall\ j, \\ \delta_{ij} \in \{0, 1\}, \end{cases}$$

where $\delta_{ij}$ are the decision variables:

$$\delta_{ij} = \begin{cases} 1\ , & \text{target } x_{f,i} \text{ assigned to agent } j, \\ 0\ , & \text{otherwise.} \end{cases}$$

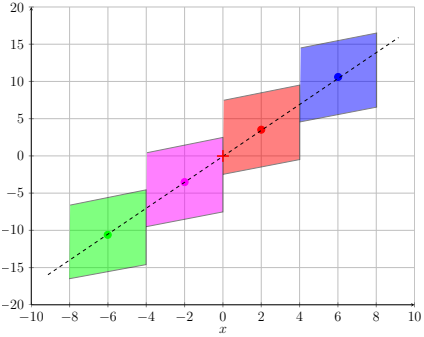


# Minimal configuration via global coord.

- Let $N = 4$ homogeneous agents
- MI optimization problem:

$$\min_{(x_i, u_i)} \quad \sum_{i=1}^{4} \|x_i\|_2^2$$

$$\text{subject to} \begin{cases} (x_i - x_j) \notin (\{-S_i\} \oplus S_j), \forall i \neq j \\ \dot{x}_i = Ax_i + Bu_i \end{cases}$$




Convergence towards to the minimal configuration: using task assignment and collision avoidance techniques.

- mobile sensors network
- coverage

# Mathematical Intermezzo: The Kronecker Product ($\otimes$)

## Definition

Let $\mathbf{A}$ be an $m \times n$ matrix and $\mathbf{B}$ be a $p \times q$ matrix. Their ***Kronecker product***, denoted $\mathbf{A} \otimes \mathbf{B}$, is the $(mp) \times (nq)$ block matrix defined as:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

Essentially, each element $a_{ij}$ of $\mathbf{A}$ is replaced by the block $a_{ij}\mathbf{B}$.

## A 2x2 Example

Let $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix}$. Then:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} 1 \cdot \mathbf{B} & 2 \cdot \mathbf{B} \\ 3 \cdot \mathbf{B} & 4 \cdot \mathbf{B} \end{bmatrix} = \left[\begin{array}{cc|cc} 1(0) & 1(5) & 2(0) & 2(5) \\ 1(6) & 1(7) & 2(6) & 2(7) \\ \hline 3(0) & 3(5) & 4(0) & 4(5) \\ 3(6) & 3(7) & 4(6) & 4(7) \end{array}\right] = \left[\begin{array}{cc|cc} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ \hline 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{array}\right]$$

# Mathematical Intermezzo: The Kronecker Product ($\otimes$)

## Definition

Let $\mathbf{A}$ be an $m \times n$ matrix and $\mathbf{B}$ be a $p \times q$ matrix. Their ***Kronecker product***, denoted $\mathbf{A} \otimes \mathbf{B}$, is the $(mp) \times (nq)$ block matrix defined as:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

Essentially, each element $a_{ij}$ of $\mathbf{A}$ is replaced by the block $a_{ij}\mathbf{B}$.

## Key Properties

- **Bilinearity**: Distributes over addition, scalars factor out.
- **Associativity**: $(\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C})$
- **Transpose**: $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$
- **Mixed Product Property**: $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$ (Requires dimensions compatible for $\mathbf{AC}$ and $\mathbf{BD}$)
- **Inverse**: $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$ (If $\mathbf{A}$, $\mathbf{B}$ are invertible)

# Single Integrator Example 1/2

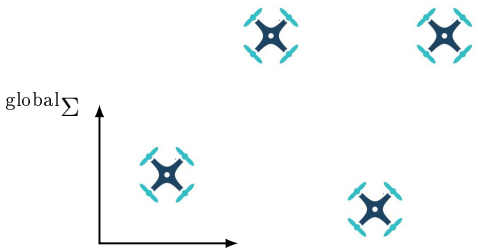
- $i$-th agent - single integrator ($i = 1 : N$):

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

- $p_i \in \mathbb{R}^d$ - position (w.r.t global)
- $u_i \in \mathbb{R}^d$ - control input (velocity)

Consider:

- $p^* \in \mathbb{R}^{dN} \longrightarrow p^* = p^*(t)$
- $F : \mathbb{R}^{dN} \mapsto \mathbb{R}^{dN}$, $F(p) := p$
- the desired formation: $p \longmapsto p^*$



# Single Integrator Example 1/2

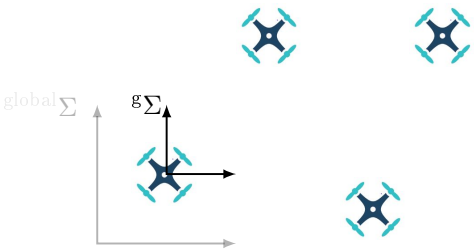
- $i$-th agent - single integrator ($i = 1 : N$):

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

- $p_i \in \mathbb{R}^d$ - position (w.r.t global)
- $u_i \in \mathbb{R}^d$ - control input (velocity)

Assume:

- agents sense their **absolute** position
- control law : $u_i = -k_p(p_i^\star - p_i),\ k_p > 0$
- error dynamics for $e_p := p^* - p$



# Single Integrator Example 1/2

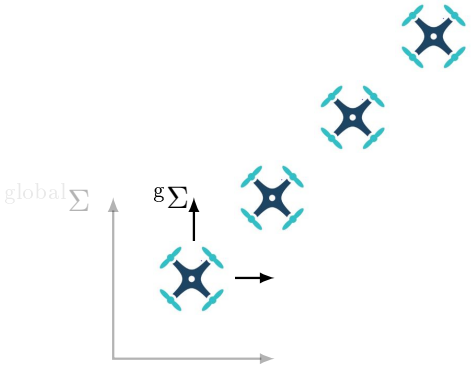
- $i$-th agent - single integrator ($i = 1 : N$):

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

- $p_i \in \mathbb{R}^d$ - position (w.r.t global)
- $u_i \in \mathbb{R}^d$ - control input (velocity)

Assume:

- agents sense their **absolute** position
- control law : $u_i = -k_p(p_i^\star - p_i),\ k_p > 0$
- error dynamics for $e_p := p^* - p$



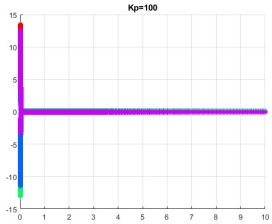
# Single Integrator Example 2/2

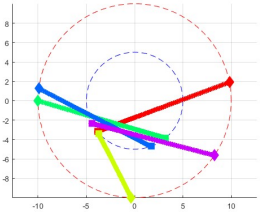
- without interaction:
  - error dynamics $\dot{e}_p = -k_p e_p$
  - exponential convergence of $p$ to $p^*$
  - convergence rate $\sim k_p$
- with interaction
  - additional control input $\sum_{j \in \mathcal{N}_i} w_{ij}(p_i - p_j)$
  - control law : $u_i = -k_p(p_i^\star - p_i) + \sum_{j \in \mathcal{N}_i} w_{ij}(p_i - p_j)$
  - error dynamics: $\dot{e}_{p_i} = -k_p e_{p_i} - \sum_{j \in \mathcal{N}_i} w_{ij} e_{p_i}$

**Gersgorini's disc theorem**

$\text{eig}(k_p I_n + L) \geq k_p$

- convergence ?



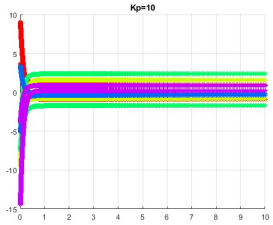
# Single Integrator Example 2/2

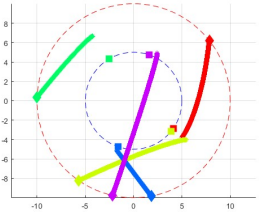
- without interaction:
  - error dynamics $\dot{e}_p = -k_p e_p$
  - exponential convergence of $p$ to $p^*$
  - convergence rate $\sim k_p$
- with interaction
  - additional control input $\sum_{j\in\mathcal{N}_i} w_{ij}(p_i - p_j)$
  - control law : $u_i = -k_p(p_i^\star - p_i) + \sum_{j\in\mathcal{N}_i} w_{ij}(p_i - p_j)$
  - error dynamics: $\dot{e}_{p_i} = -k_p e_{p_i} - \sum_{j\in\mathcal{N}_i} w_{ij} e_{p_i}$

**Gersgorini's disc theorem**

$\text{eig}(k_p I_n + L) \geq k_p$

- convergence ?



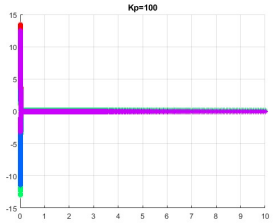
# Single Integrator Example 2/2

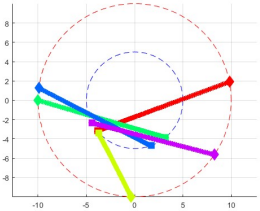
- without interaction:
  - error dynamics $\dot{e}_p = -k_p e_p$
  - exponential convergence of $p$ to $p^*$
  - convergence rate $\sim k_p$
- with interaction
  - additional control input $\sum_{j\in\mathcal{N}_i} w_{ij}(p_i - p_j)$
  - control law : $u_i = -k_p(p_i^\star - p_i) + \sum_{j\in\mathcal{N}_i} w_{ij}(p_i - p_j)$
  - error dynamics: $\dot{e}_p = -k_p e_p - (L \otimes I_n)e_p$

**Gersgorini's disc theorem**

$\text{eig}(k_p I_n + L) \geq k_p$

- convergence ? $k_p$ s.t. convergence !



# Double Integrator Example 1/2

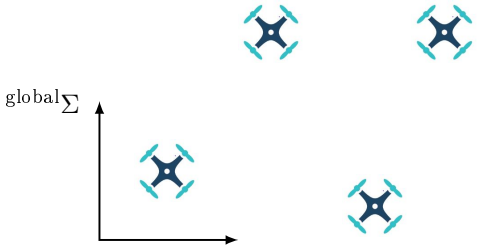
- $i$-th agent - "double integrator" ($i = 1 : N$):

$$\ddot{p}_i = u_i$$

- $p_i \in \mathbb{R}^n$ - position (w.r.t global)
- $u_i \in \mathbb{R}^n$ - control input (acceleration)

Consider:

- $p^* \in \mathbb{R}^{nN} \longrightarrow p^* = p^*(t)$
- $F : \mathbb{R}^{nN} \mapsto \mathbb{R}^{nN}$, $F(p) := p$
- the desired formation: $p \longmapsto p^*$



# Double Integrator Example 1/2

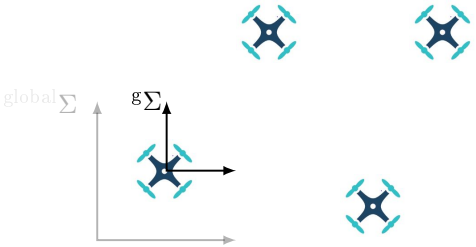
- $i$-th agent - "double integrator" ($i = 1 : N$):

$$\ddot{p}_i = u_i$$

- $p_i \in \mathbb{R}^n$ - position (w.r.t global)
- $u_i \in \mathbb{R}^n$ - control input (acceleration)

Assume:

- agents sense their **absolute** position
- control law : $u_i = -k_a k_v(\dot{p}_i^\star - \dot{p}_i) - k_a k_p(p_i^\star - p_i)$
- with $k_a, k_v, k_p > 0$



# Double Integrator Example 1/2

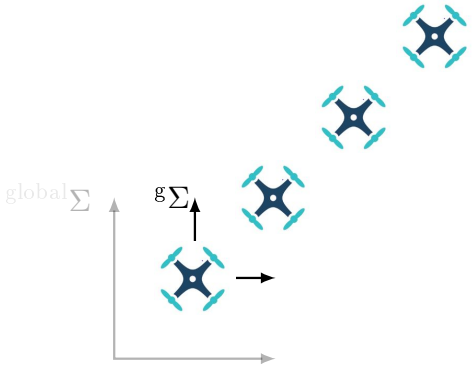
- $i$-th agent - "double integrator" ($i = 1 : N$):

$$\ddot{p}_i = u_i$$

- $p_i \in \mathbb{R}^n$ - position (w.r.t global)
- $u_i \in \mathbb{R}^n$ - control input (acceleration)

Assume:

- agents sense their **absolute** position
- control law : $u_i = -k_a k_v(\dot{p}_i^\star - \dot{p}_i) - k_a k_p(p_i^\star - p_i)$
- with $k_a, k_v, k_p > 0$

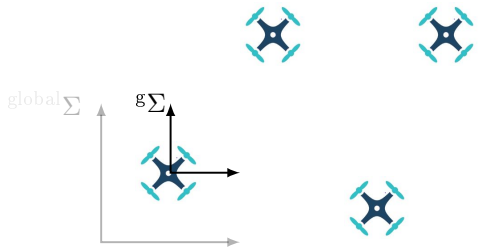


# Double Integrator Example 2/2

- without interaction:
  - error dynamics $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p)$
  - convergence of $p$ to $p^*$
  - convergence rate $\sim k_a, k_v, k_p$
- with interaction:
  - additional control input $u_i^a = \sum_{j\in\mathcal{N}_i} w_{ij}k_v(\dot{p}_j - \dot{p}_i) + w_{ij}k_p(p_j - p_i)$
  - control law : $u_i = -k_p(p_i^\star - p_i) + u_i^a$
  - error dynamics: $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p) - (L \otimes I_n)(k_v\dot{e}_p + k_p e_p)$

**Convergence**

- if G has a spanning tree

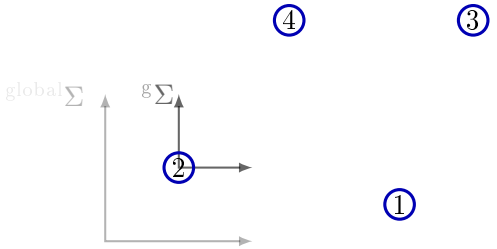


# Double Integrator Example 2/2

- without interaction:
  - error dynamics $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p)$
  - convergence of $p$ to $p^*$
  - convergence rate $\sim k_a, k_v, k_p$
- with interaction:
  - additional control input $u_i^a = \sum_{j\in\mathcal{N}_i} w_{ij}k_v(\dot{p}_j - \dot{p}_i) + w_{ij}k_p(p_j - p_i)$
  - control law : $u_i = -k_p(p_i^\star - p_i) + u_i^a$
  - error dynamics: $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p) - (L \otimes I_n)(k_v\dot{e}_p + k_p e_p)$

**Convergence**

- in practice, $\text{eig}(\dots) \leq \text{eig}(\dots)$

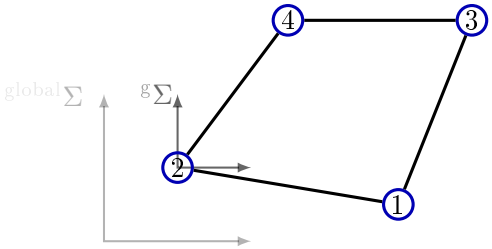


# Double Integrator Example 2/2

- without interaction:
  - error dynamics $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p)$
  - convergence of $p$ to $p^*$
  - convergence rate $\sim k_a, k_v, k_p$
- with interaction:
  - additional control input $u_i^a = \sum_{j\in\mathcal{N}_i} w_{ij}k_v(\dot{p}_j - \dot{p}_i) + w_{ij}k_p(p_j - p_i)$
  - control law : $u_i = -k_p(p_i^\star - p_i) + u_i^a$
  - error dynamics: $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p) - (L \otimes I_n)(k_v\dot{e}_p + k_p e_p)$

**Convergence**

- in practice, $\text{eig}(\dots) \leq \text{eig}(\dots)$

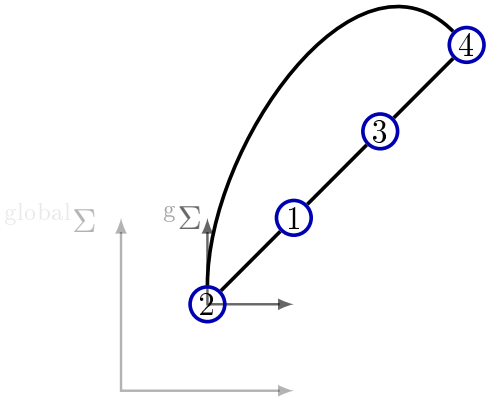


# Double Integrator Example 2/2

- without interaction:
  - error dynamics $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p)$
  - convergence of $p$ to $p^*$
  - convergence rate $\sim k_a, k_v, k_p$
- with interaction:
  - additional control input $u_i^a = \sum_{j\in\mathcal{N}_i} w_{ij}k_v(\dot{p}_j - \dot{p}_i) + w_{ij}k_p(p_j - p_i)$
  - control law : $u_i = -k_p(p_i^\star - p_i) + u_i^a$
  - error dynamics: $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p) - (L \otimes I_n)(k_v\dot{e}_p + k_p e_p)$

**Convergence rate**

- in practice, $\text{eig}(\dots) \leq \text{eig}(\dots)$

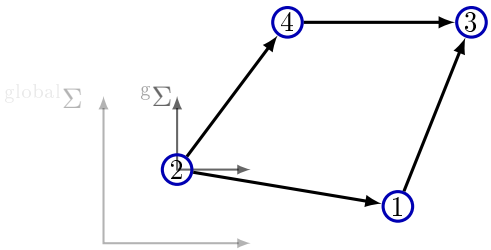


# Double Integrator Example 2/2

- without interaction:
  - error dynamics $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p)$
  - convergence of $p$ to $p^*$
  - convergence rate $\sim k_a, k_v, k_p$
- with interaction:
  - additional control input $u_i^a = \sum_{j\in\mathcal{N}_i} w_{ij}k_v(\dot{p}_j - \dot{p}_i) + w_{ij}k_p(p_j - p_i)$
  - control law : $u_i = -k_p(p_i^\star - p_i) + u_i^a$
  - error dynamics: $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p) - (L \otimes I_n)(k_v\dot{e}_p + k_p e_p)$

**Convergence rate**

- in practice, $\text{eig}(\dots) \leq \text{eig}(\dots)$

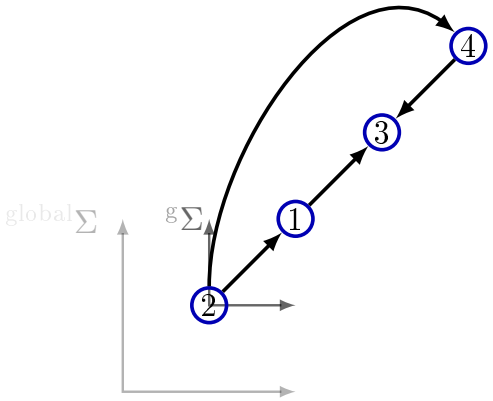


# Double Integrator Example 2/2

- without interaction:
  - error dynamics $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p)$
  - convergence of $p$ to $p^*$
  - convergence rate $\sim k_a, k_v, k_p$
- with interaction:
  - additional control input $u_i^a = \sum_{j\in\mathcal{N}_i} w_{ij}k_v(\dot{p}_j - \dot{p}_i) + w_{ij}k_p(p_j - p_i)$
  - control law : $u_i = -k_p(p_i^\star - p_i) + u_i^a$
  - error dynamics: $\ddot{e}_p = -k_a(k_v\dot{e}_p + k_p e_p) - (L \otimes I_n)(k_v\dot{e}_p + k_p e_p)$

**Convergence rate**

- in practice, eig$(\ldots) \leq$ eig$(\ldots)$

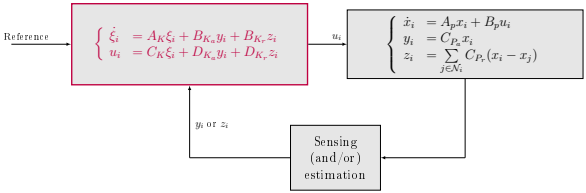


# LTI Dynamics Example 1/2

- $i$-th agent - linear time-invariant dynamics ($i = 1 : N$):

$$\begin{cases} \dot{x}_i &= A_p x_i + B_p u_i \\ y_i &= C_{P_a} x_i \\ z_i &= \sum_{j \in \mathcal{N}_i} C_{P_r}(x_i - x_j) \end{cases}$$

- $y_i \in \mathbb{R}^n$ - **absolute** measurement
- $z_i \in \mathbb{R}^n$ - **relative** measurement




Assume:

- *dynamic* control law ($i = 1 : N$):

$$\begin{cases} \dot{\xi}_i &= A_K \xi_i + B_{K_a} y_i + B_{K_r} z_i \\ u_i &= C_K \xi_i + D_{K_a} y_i + D_{K_r} z_i \end{cases}$$

- if $B_{K_r}, D_{K_r} = \mathbb{0}$, position-based
- if $B_{K_a}, D_{K_a} = \mathbb{0}$, displacement-based

# LTI Dynamics Example 2/2

- *dynamic* control law ($i = 1 : N$):

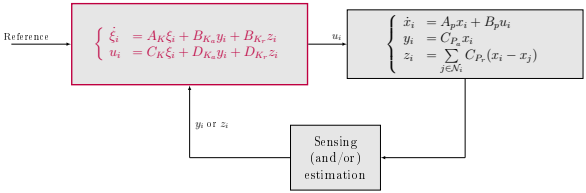
$$\begin{cases} \dot{\xi}_i &= A_K \xi_i + B_{K_a} y_i + B_{K_r} z_i \\ u_i &= C_K \xi_i + D_{K_a} y_i + D_{K_r} z_i \end{cases}$$

Remarks:

- if $B_{K_r}, D_{K_r} = \mathbb{O}$, position-based
- if $B_{K_a}, D_{K_a} = \mathbb{O}$, displacement-based
- stability of MAS is equiv. with assymp. stabilty of:

$$\begin{cases} \dot{x}_i &= A_p x_i + B_p u_i \\ y_i &= C_{P_a} x_i \\ z_i &= \lambda_i C_{P_r} (x_i - x_j) \end{cases}$$

- $\lambda_i = \mathrm{eig}(L)$

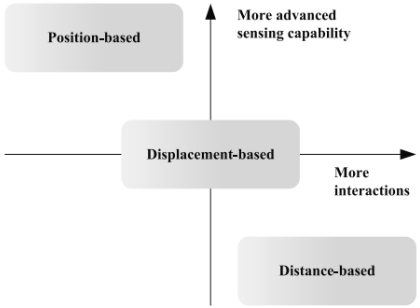


# Outline

# Displacement-based formation control

Requirements:

- **Sensing capability**:
  - each agent has its own local coordinate system
  - sense their **relative** positions ($\rightarrow$ displacements)
  - alignment of coordinate systems (!)
- **Interaction topology**
  - is required (mandatory)
  - needs to ensure achievement of the desired formation by controlling only the displacements of their neighbors




- desired displacement: $F(z) := \left[ \dots (z_i - z_j)^\top \dots \right]^\top$

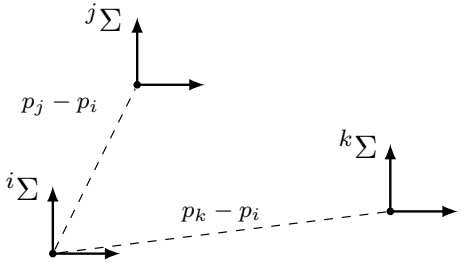
**Remark**

The interaction graph needs to be characterized by either connectedness or existence of a spanning tree.

# Displacement-based formation control

Requirements:

- **Sensing capability**:
  - each agent has its own local coordinate system
  - sense their **relative** positions ($\rightarrow$ displacements)
  - alignment of coordinate systems (!)
- **Interaction topology**
  - is required (mandatory)
  - needs to ensure achievement of the desired formation by controlling only the displacements of their neighbors




- desired displacement: $F(z) := \left[\dots (z_i - z_j)^\top \dots\right]^\top$

**Remark**

The interaction graph needs to be characterized by either connectedness or existence of a spanning tree.

# Single Integrator Example

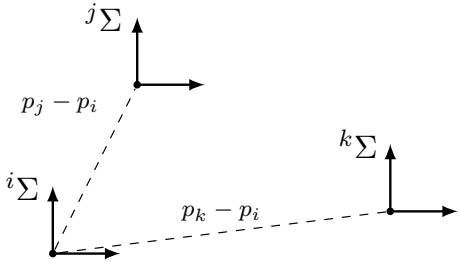
- $i$-th agent - "single integrator" ($i = 1 : N$):

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

- $p_i \in \mathbb{R}^n$ - position (w.r.t global)
- $u_i \in \mathbb{R}^n$ - control input (velocity)

Consider:

- $p^* \in \mathbb{R}^{nN} \longrightarrow p^* = p^*(t)$
- $F : \mathbb{R}^{nN} \mapsto \mathbb{R}^{nN},\ F(p) := \left[\dots (p_i - p_j)^\top \dots\right]^\top$
- the desired formation:
  $E_{p^\star} := \{p | p_i - p_j = p_i^\star - p_j^\star, \forall i, j \in V\}$
- $p \mapsto p^\star$



# Single Integrator Example

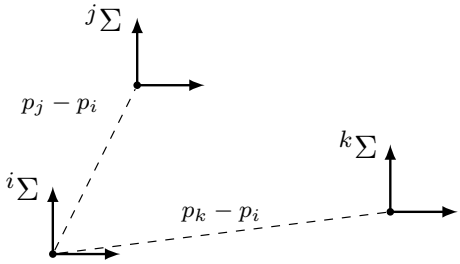
- $i$-th agent - "single integrator" ($i = 1 : N$):

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

- $p_i \in \mathbb{R}^n$ - position (w.r.t global)
- $u_i \in \mathbb{R}^n$ - control input (velocity)

Assume:

- agents sense their neighbors' **relative** position
- control law : $u_i = -k_p \sum_{j \in \mathcal{N}_i} (p_j - p_i - p_j^\star + p_i^\star),\ k_p > 0$
- error dynamics for $e_p := p^* - p \sim$ consensus
- $\dot{e}_p = -k_p (L \otimes I_n) e_p$



# Double Integrator 1/2

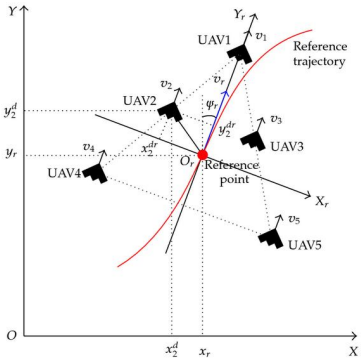
- $i$-th agent - "double integrator" ($i = 1 : N$):

$$\begin{cases} \dot{p}_i &= v_i \\ \dot{v}_i &= u_i \end{cases}$$

- $p_i \in \mathbb{R}^n$ - position, $v_i \in \mathbb{R}^n$ - velocity
- $u_i \in \mathbb{R}^n$ - control input (acceleration)

Assume:

- agents sense their neighbors' **relative** position and velocity
- the relative measurements: $z_i = \begin{bmatrix} p_i & v_i \end{bmatrix}^\top$
- desired formation: $z^* = \begin{bmatrix} \dots \begin{bmatrix} p_i^\star & v_i^\star \end{bmatrix}^\top \dots \end{bmatrix}^\top$




$$\begin{cases} \dot{p}^\star &= v^\star \\ \dot{v}^\star &= 0 \text{ !!!!!} \end{cases}$$

# Double Integrator 1/2

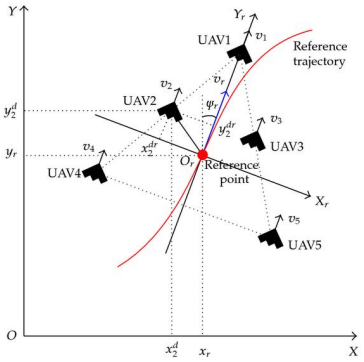
- $i$-th agent - "double integrator" ($i = 1 : N$):

$$\begin{cases} \dot{p}_i &= v_i \\ \dot{v}_i &= u_i \end{cases}$$

- $p_i \in \mathbb{R}^n$ - position, $v_i \in \mathbb{R}^n$ - velocity
- $u_i \in \mathbb{R}^n$ - control input (acceleration)

Consider:

- $z^* \in \mathbb{R}^{2nN} \longrightarrow z^* = z^*(t)$
- $F : \mathbb{R}^{nN} \mapsto \mathbb{R}^{nN},\ F(z) := \left[\dots (z_i - z_j)^\top \dots\right]^\top$
- $z \mapsto z^\star$
- $E_{p^\star,p^\star} := \{\begin{bmatrix} p^\top & v^\top \end{bmatrix}^\top | p_i - p_j = p_i^\star - p_j^\star,\ v_i - v_j = v_i^\star - v_j^\star, \forall i,j \in V\}$




$$\begin{cases} \dot{p}^\star &= v^\star \\ \dot{v}^\star &= 0 \text{ !!!!!} \end{cases}$$

# Double Integrator 1/2

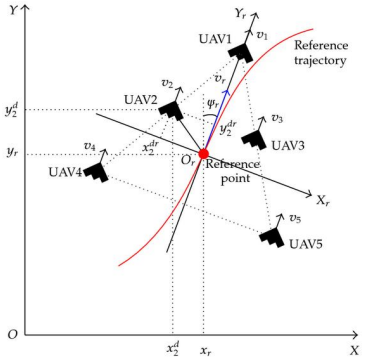
- $i$-th agent - "double integrator" ($i = 1 : N$):

$$\begin{cases} \dot{p}_i &= v_i \\ \dot{v}_i &= u_i \end{cases}$$

- $p_i \in \mathbb{R}^n$ - position, $v_i \in \mathbb{R}^n$ - velocity
- $u_i \in \mathbb{R}^n$ - control input (acceleration)

Control law & error dynamics :

$$u_i = -k_p \sum_{j \in \mathcal{N}_i} (p_i - p_j - p_i^\star + p_j^\star) - k_v \sum_{j \in \mathcal{N}_i} (v_i - v_j + v_i^\star - v_j^\star) k_p, k_v > 0$$




$e_p := p^\star - p,\ e_v := v^\star - v$

$$\begin{bmatrix} \dot{e}_p \\ \dot{e}_v \end{bmatrix} = \begin{bmatrix} \mathbf{0} & I_{nN} \\ -k_p(L \otimes I_n) & -k_v(L \otimes I_n) \end{bmatrix} \begin{bmatrix} e_p \\ e_v \end{bmatrix}$$

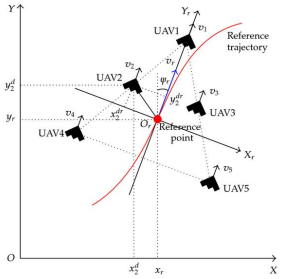
# Double Integrator 2/2

$$e_p := p^\star - p,\ e_v := v^\star - v$$

$$\begin{bmatrix} \dot{e}_p \\ \dot{e}_v \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & I_{nN} \\ -k_p(L \otimes I_n) & -k_v(L \otimes I_n) \end{bmatrix}}_{\Gamma} \begin{bmatrix} e_p \\ e_v \end{bmatrix}$$

- let $\{\lambda_1, ....\lambda_n\} = \text{eig}(L)$
- $\text{eig}(\Gamma) = \{\mu_{i\pm}\}_{i \leq n}$
- Due to properties of Kronecker product:

$$\mu_{i\pm} = \frac{k_v\lambda_i \pm \sqrt{k_v^2\lambda_i^2 + 4k_p\lambda_i}}{2}$$




**Important Remark**

Note that existence of a spanning tree in G is a necessary **but not sufficient** condition for achieving the desired formation

**Theorem**

$e_p \to 0$ and $e_v \to 0$ asymptotically if and only if $\Gamma$ has exactly $n$ zero eigenvalues and all the others have negative real parts.

# LTI Dynamics Example 1/2

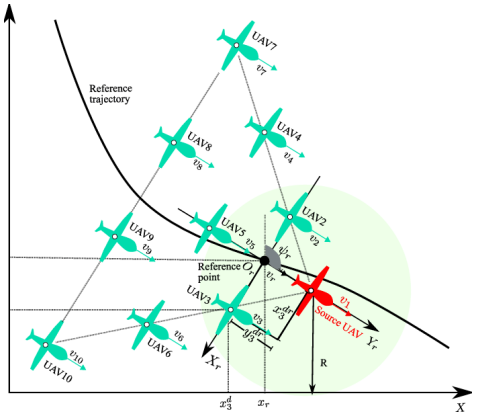
- $i$-th agent - general linear case ($i = 1 : N$):

$$\dot{x}_i = Ax_i + Bu_i$$

- $x_i \in \mathbb{R}^n$ - state (w.r.t. global coord. system)
- $u_i \in \mathbb{R}^m$ - control input

Assume:

- agent $i$ sense the relative values: $y_{ij} = C(x_i - x_j)$, $j \in \mathcal{N}_i$
- $C \in \mathbb{R}^{p \times n}$
- $x^\star \in \mathbb{R}^{nN}$ given as $\dot{x}^\star = (I_n \otimes A)x^\star$
- desired formation:
  $E_{x^\star} = \{x | x_i - x_j = x_i^\star - x_j^\star,\ i, j = 1 : N\}$



# LTI Dynamics Example 2/2

- control law ($i = 1 : N$):

$$u_i = KC \sum_{j \in \mathcal{N}_i} w_{ij}(x_j - x_i), \quad K \in \mathbb{R}^{m \times p}$$

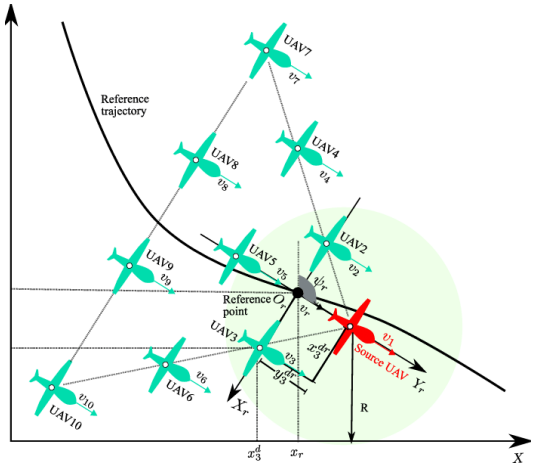
Remarks:

- $e_x := x^\star - x$
- error dynamics: $\dot{e}_x = (I_n \otimes A)e_x - (L \otimes BKC)e_x$

**Theorem**

The desired formation is achieved if

- $G$ has a spanning tree
- $A - \lambda_i BKC$ are Hurwitz, where $\lambda_i \in \text{eig}(L)$ and $\lambda_i \neq 0$

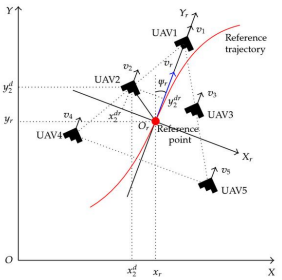


# Non-holonomic agent dynamics example

- each agent $i$ dynamics is described by:

$$\begin{cases} \dot{x}_i = v_i \cos\theta_i \\ \dot{y}_i = v_i \sin\theta_i, \qquad i = 1:N \\ \dot{\theta}_i = \omega_i \end{cases}$$

- $p_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^\top \in \mathbb{R}^2$ - position w.r.t global frame
- $\theta_i \in (-\pi, \pi]$ - heading angle w.r.t global frame
- $v_i \in \mathbb{R}$ and $\omega_i \in \mathbb{R}$ control inputs
- the desired formation: $E_{p^\star} := \{p \mid p_i - p_j = p_i^\star - p_j^\star, \forall i, j \in V\}$
- $p \mapsto p^\star$




For the unicycles,[a] propose:

$$\begin{cases} v_i = k \left[\cos\theta_i \sin\theta_i\right] \sum_{j \in \mathcal{N}_i} (p_j - p_i + p_i^\star - p_i^\star) \\ \omega_i = \cos t \end{cases}$$

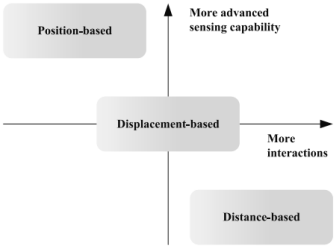
[a] Lin, Francis, and Maggiore 2005.

# Outline

# Distance-based formation control

Requirements:

- **Sensing capability**:
  - each agent has its own local coordinate system
  - sense their neighbors' **relative** positions
  - orientation of coordinate systems not aligned (!)
- **Interaction topology**
  - is required (mandatory)
  - desired formation $\longrightarrow$ desired distance between any pair of agents
  - formation can be treated as a given rigid body
  - formation invariant to combinations of translation and rotation




- desired formation:
  $F(z) := \left[ \dots \|z_i - z_j\|^\top \dots \right]^\top$

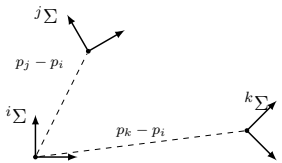
**Remark**

Agents' model linear + distance-based formation control $\rightarrow$ **nonlinear** control laws (!) .

# Distance-based formation control

Requirements:

- **Sensing capability**:
  - each agent has its own local coordinate system
  - sense their neighbors' **relative** positions
  - orientation of coordinate systems not aligned (!)
- **Interaction topology**
  - is required (mandatory)
  - desired formation $\longrightarrow$ desired distance between any pair of agents
  - formation can be treated as a given rigid body
  - formation invariant to combinations of translation and rotation




- desired formation:
  $F(z) := \left[ \dots \|z_i - z_j\|^\top \dots \right]^\top$

**Remark**

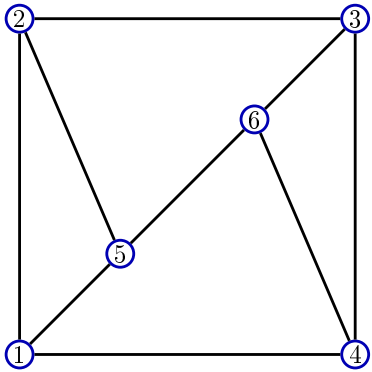
The interaction graph need to be **rigid** or *persistent*.
In the literature, one treats separately ***undirected*** and *directed* formations.

# Graph rigidity - Preliminaries 1/3

Let $G = (V, E)$ - **undirected**

**Definitions**

- $p \in \mathbb{R}^{dN}$ is **a realization** of $G$ in $\mathbb{R}^d$, if to each $i \in V(G)$ a point $p_i \in \mathbb{R}^d$ is assigned.
- $(G, p)$ - a framework of $G$ in $\mathbb{R}^d$
- an edge function $g_G : \mathbb{R}^{dN} \mapsto \mathbb{R}^M$ associated to $(G, p)$ is defined as: $g_G(p) := \frac{1}{2}\left[\dots \|p_i - p_j\|^2 \dots\right]^\top$, $i, j \in E$




Similarities:

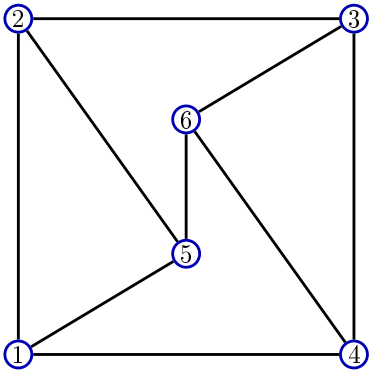
- framework $\rightarrow$ realization of $G \sim$ isomorphism
- $g_G(p) \sim F(p)$
- $M = ?$

# Graph rigidity - Preliminaries 1/3

Let $G = (V, E)$ - **undirected**

**Definitions**

- $p \in \mathbb{R}^{dN}$ is **a realization** of $G$ in $\mathbb{R}^d$, if to each $i \in V(G)$ a point $p_i \in \mathbb{R}^d$ is assigned.
- $(G, p)$ - a framework of $G$ in $\mathbb{R}^d$
- an edge function $g_G : \mathbb{R}^{dN} \mapsto \mathbb{R}^M$ associated to $(G, p)$ is defined as: $g_G(p) := \frac{1}{2}\left[\dots \|p_i - p_j\|^2 \dots\right]^\top$, $i, j \in E$




Similarities:

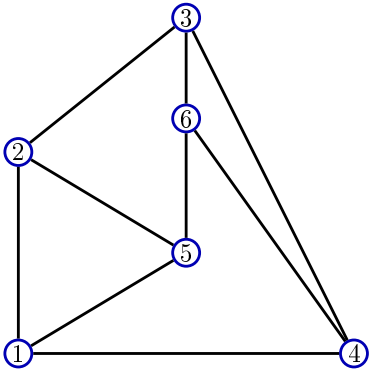
- framework $\rightarrow$ realization of $G \sim$ isomorphism
- $g_G(p) \sim F(p)$
- $M = ?$

# Graph rigidity - Preliminaries 1/3

Let $G = (V, E)$ - **undirected**

**Definitions**

- $p \in \mathbb{R}^{dN}$ is **a realization** of $G$ in $\mathbb{R}^d$, if to each $i \in V(G)$ a point $p_i \in \mathbb{R}^d$ is assigned.
- $(G, p)$ - a framework of $G$ in $\mathbb{R}^d$
- an edge function $g_G : \mathbb{R}^{dN} \mapsto \mathbb{R}^M$ associated to $(G, p)$ is defined as: $g_G(p) := \frac{1}{2}\left[\dots \|p_i - p_j\|^2 \dots\right]^\top$, $i, j \in E$




Similarities:

- framework $\rightarrow$ realization of $G \sim$ isomorphism
- $g_G(p) \sim F(p)$
- $M = ?$

# Graph rigidity - Preliminaries 2/3

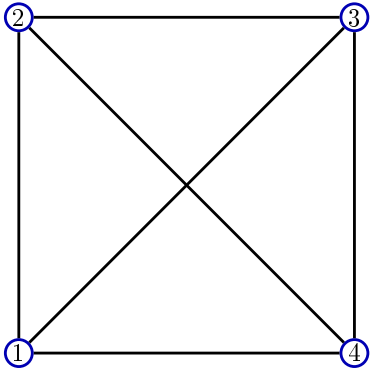
**Theorem**

$(G, p)$ is ***rigid*** if there $\exists$ a neighborhood $U_p$ of $p \in \mathbb{R}^{dN}$ such that $\forall q \in U_p$ we have :

$$(G, p) \equiv (G, q) \longrightarrow (G, p) \cong (G, q)$$

Remarks:

- $(G, p) \equiv (G, q)$ if $g_G(p) = g_G(q)\ \forall (i, j) \in E$
- $(G, p) \cong (G, q)$ if $g_G(p) = g_G(q)\ \forall i, j \in V$



# Graph rigidity - Preliminaries 2/3

**Theorem**

$(G, p)$ is ***rigid*** if there $\exists$ a neighborhood $U_p$ of $p \in \mathbb{R}^{dN}$ such that $\forall q \in U_p$ we have :

$$(G, p) \equiv (G, q) \longrightarrow (G, p) \cong (G, q)$$

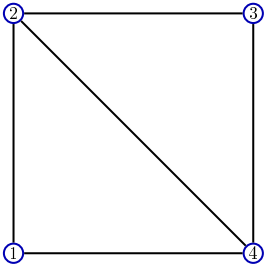
Remarks:

- $(G, p) \equiv (G, q)$ if $g_G(p) = g_G(q)\ \forall (i, j) \in E$
- $(G, p) \cong (G, q)$ if $g_G(p) = g_G(q)\ \forall i, j \in V$

**Minimal rigidity**

No edge of G can be removed without losing the rigidity of $(G, p)$

- A graph is said to be *minimally rigid* if it is rigid and if there is no rigid graph having the same vertices but fewer edges.



# Graph rigidity - Preliminaries 2/3

**Theorem**

$(G, p)$ is ***rigid*** if there $\exists$ a neighborhood $U_p$ of $p \in \mathbb{R}^{dN}$ such that $\forall q \in U_p$ we have :

$$(G, p) \equiv (G, q) \longrightarrow (G, p) \cong (G, q)$$

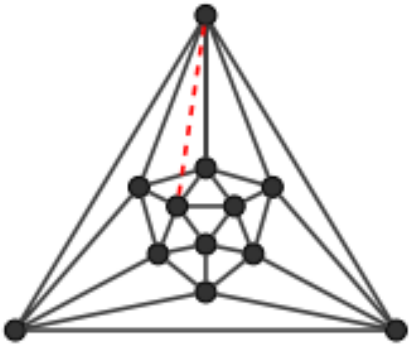
Remarks:

- $(G, p) \equiv (G, q)$ if $g_G(p) = g_G(q)\ \forall (i, j) \in E$
- $(G, p) \cong (G, q)$ if $g_G(p) = g_G(q)\ \forall i, j \in V$

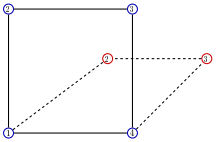
**Minimal rigidity - Theorem**

$(G, p)$ is ***minimally rigid*** if it is **rigid** and

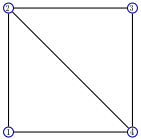
$$|E| = dN - \frac{d(d+1)}{2}$$



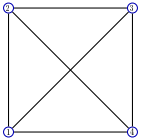
# Graph rigidity - Preliminaries 3/3



(i)



(ii)



(iii)

(i) **not rigid**

- $p$ is not unique
- even if all the edges length are fixed, the graph can be deformed

(ii) **rigid**

- by adding an edge to **(i)**
- $p$ becomes locally unique up to congruence

(iii) **globally rigid**

- $p$ is globally unique
- the graph is complete (!)

# Illustrative Example - single integrator case 1/2

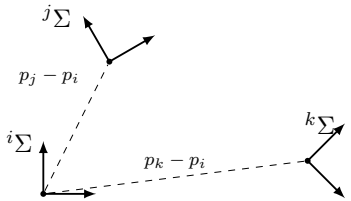
- $i$-th agent - "single integrator" ($i = 1 : N$):

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

- $p_i \in \mathbb{R}^n$ - position (in ${}^g\Sigma$)
- $u_i \in \mathbb{R}^n$ - control input (velocity, in ${}^g\Sigma$)

Consider :

- agents sense relative position of their neighbors
- the interaction graph $G$ - undirected
- the desired formation:
  $E_{p^\star} := \{p | \|p_i - p_j\| = \|p_i^\star - p_j^\star\|, \forall i, j \in V\}$
- ~~$p \mapsto p^\star$~~, but the set of realizations that are congruent to $p^\star$



# Illustrative Example - single integrator case 1/2

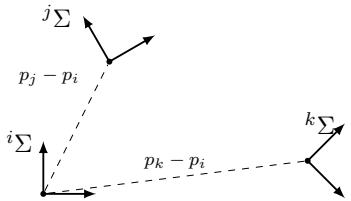
- $i$-th agent - "single integrator" ($i = 1 : N$):

$$\dot{p}_i^i = u_i^i$$

- $p_i^i \in \mathbb{R}^n$ - position (in ${}^i\Sigma$)
- $u_i^i \in \mathbb{R}^n$ - control input (velocity, in ${}^i\Sigma$)

Consider :

- agents sense relative position of their neighbors
- the interaction graph $G$ - undirected
- the desired formation:
  $E_{p^\star} := \{p | \|p_i - p_j\| = \|p_i^\star - p_j^\star\|, \forall i, j \in V\}$
- ~~$p \mapsto p^\star$~~, but the set of realizations that are congruent to $p^\star$




**Remark**

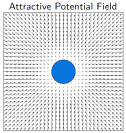
$$p_{ji}^i := p_j^i - p_i^i \equiv p_i^j, \ \forall j \in \mathcal{N}_i$$

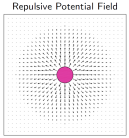
# Illustrative Example - single integrator case 2/2

## Control laws in distance-based setup

**Remark**

Gradient control laws have been **popularly** used to achieve the desired formation in distance-based control.




- for agent $i$, we can define a local potential function $\varphi_i : \mathbb{R}^{d(\mathcal{N}_i+1)} \to \mathbb{R}_+$ with:

$$\varphi_i(p_i^i, \ldots, p_i^j, \ldots) := \frac{k_p}{2} \sum_{j \in \mathcal{N}_i} \gamma_{ij}(\|p_j^i - p_i^i\|), \; k_p > 0 \text{ and } \gamma_{ij} : \mathbb{R} \to \mathbb{R}_+ \text{ differentiable}$$

- control law:

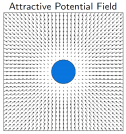
$$u_i^i = -\nabla_{p_i^i} \varphi_i(p_i^i, \ldots, p_i^j, \ldots) = k_p \sum_{j \in \mathcal{N}_i} \frac{\partial \gamma_{ij}(\|p_j^i - p_i^i\|)}{\partial \|p_j^i - p_i^i\|} \frac{p_j^i - p_i^i}{\|p_j^i - p_i^i\|}$$

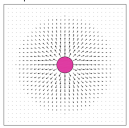
# Illustrative Example - single integrator case 2/2

## Control laws in distance-based setup

**Remark**

Gradient control laws have been **popularly** used to achieve the desired formation in distance-based control.




- for agent $i$, we can define a local potential function $\varphi_i : \mathbb{R}^{d(\mathcal{N}_i+1)} \to \mathbb{R}_+$ with:

$$\varphi_i(p_i^i, \ldots, p_i^j, \ldots) := \frac{k_p}{2} \sum_{j \in \mathcal{N}_i} \gamma_{ij}(\|p_j^i - p_i^i\|), \; k_p > 0 \text{ and } \gamma_{ij} : \mathbb{R} \to \mathbb{R}_+ \text{ differentiable}$$

- control law:

$$u_i^i = -\nabla_{p_i^i} \varphi_i(p_i^i, \ldots, p_i^j, \ldots) = k_p \sum_{j \in \mathcal{N}_i} \frac{\partial \gamma_{ij}(\|p_j^i - p_i^i\|)}{\partial \|p_j^i - p_i^i\|} \frac{p_j^i - p_i^i}{\|p_j^i - p_i^i\|}$$

**Remark**

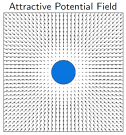
${}^i\Sigma$ cannot give information about stability $\longrightarrow$ change to ${}^g\Sigma$

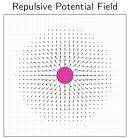
# Illustrative Example - single integrator case 2/2

## Control laws in distance-based setup

**Remark**

Gradient control laws have been **popularly** used to achieve the desired formation in distance-based control.




- for agent $i$, we can define a local potential function $\varphi_i : \mathbb{R}^{d(\mathcal{N}_i+1)} \to \mathbb{R}_+$ with:

$$\varphi_i(p_i^i, \ldots, p_i^j, \ldots) := \frac{k_p}{2} \sum_{j \in \mathcal{N}_i} \gamma_{ij}(\|p_j^i - p_i^i\|), \; k_p > 0 \text{ and } \gamma_{ij} : \mathbb{R} \to \mathbb{R}_+ \text{ differentiable}$$

- control law:

$$u_i^i = -\nabla_{p_i} \varphi_i(p_i, \ldots, p^j, \ldots) = \nabla_{p_i} \Phi(p)$$

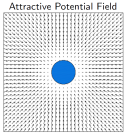
- $\Phi : \mathbb{R}^{dN} \to \mathbb{R}_+, \qquad \Phi(p) := \sum_{(i,j) \in E} \gamma_{ij}(\|p_j - p_i\|)$
- Obviously, it is required that set of the critical points of $\Phi(p)$ include $E_{p^\star}$

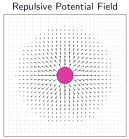
# Illustrative Example - single integrator case 2/2

## Control laws in distance-based setup

**Remark**

Gradient control laws have been **popularly** used to achieve the desired formation in distance-based control.




- for agent $i$, we can define a local potential function $\varphi_i : \mathbb{R}^{d(\mathcal{N}_i+1)} \to \mathbb{R}_+$ with:

$$\varphi_i(p_i^i, \ldots, p_i^j, \ldots) := \frac{k_p}{2} \sum_{j \in \mathcal{N}_i} \gamma_{ij}(\|p_j^i - p_i^i\|), \; k_p > 0 \text{ and } \gamma_{ij} : \mathbb{R} \to \mathbb{R}_+ \text{ differentiable}$$

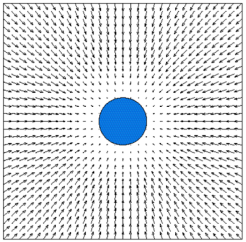
- control law:

$$u_i^i = -\nabla_{p_i} \varphi_i(p_i, \ldots, p^j, \ldots) = \nabla_{p_i} \Phi(p) \longrightarrow \dot{p} = u = -\nabla\Phi(p)$$

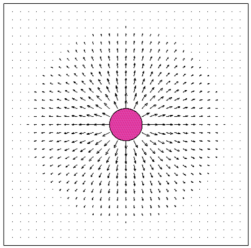
- $\Phi : \mathbb{R}^{dN} \to \mathbb{R}_+, \qquad \Phi(p) := \sum_{(i,j) \in E} \gamma_{ij}(\|p_j - p_i\|)$
- Obviously, it is required that set of the critical points of $\Phi(p)$ include $E_{p^\star}$

# How we choose $\gamma_{ij}$?

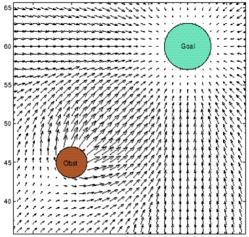
Attractive Potential Field



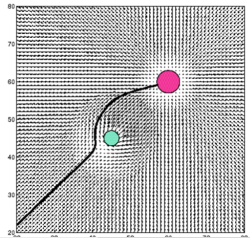
Repulsive Potential Field



Vector Sum of Two Fields




Resulting Robot Trajectory



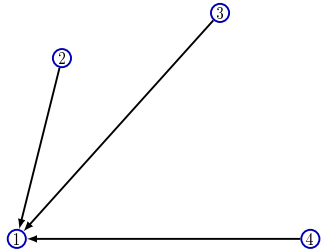
# Distance-based directed formation

Let $G = (V, E)$ - **directed**

- $p_i \in \mathbb{R}^d$ assigned to $i \in V$
- $(G, p)$ - directed framework in $\mathbb{R}^d$
- define $E^c := \{(i,j) | (j,i) \in E \text{ and } (i,j) \notin E\}$
- the underlying undirected graph of $G$ is $G^c = (V, E \cup E^c)$

**Idea**

Study the directed interaction graph $G$ via its undirected counterpart $G^c$.




- **not rigid**
- is not appropriate for distance-based control

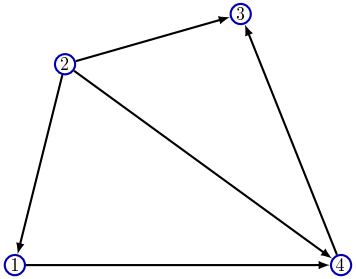
# Rigidity of digraphs 1/3 - Realizability

***"Recap"***

- the incidence matrix $H = \{h_{ki}\} \in \mathbb{R}^{m \times n}$

$$h_{ki} = \begin{cases} 1, & \text{if edge } k \text{ sinks at node } i, \\ -1, & \text{if edge } k \text{ sources at node } i, \\ 0, & \text{otherwise.} \end{cases}$$

- the Laplacian $\mathcal{L} = H^\top H$. If the graph is connected:
  - $\text{rank}(\mathcal{L}) = n - 1$
  - $\ker(\mathcal{L}) = \ker(H) = \text{span}\{\mathbf{1}_n\}$
- $p = [p_1^\top, \ldots, p_N^\top] \in \mathbb{R}^{dN}$ - representation of $G$




**Realizability**

Defining a set of distances $\mathcal{D} = \{\delta_{ij} \mid \delta_{ij} = \|p_i - p_j\|, \forall (i,j) \in \mathcal{E}\}$ at a given $t$, then $p$ is called realization of $G$, and $\mathcal{D}$ is realizable.

***Obvious*** **Remark**

The desired formation is given by a desired distance set $\mathcal{D}$

# Rigidity of digraphs 2/3 - from incidence to rigidity

- we define the relative offset between agents $i$ and $j$: $z_{ij} = p_i - p_j$
- the relative position vector $z$ as:

$$z = [z_1^\top \dots z_m^\top] \in \mathbb{R}^{dm} \text{ or, compactly } z = \mathbf{H}p$$

- $\mathbf{H} = (H \otimes I_d)$
- for the distance-based formation control approach, we consider the mapping:

$$\varsigma : \mathbb{R}^{dn} \mapsto \mathbb{R}^m, \quad \varsigma(p) = (\|p_i - p_j\|^2)_{(i,j)\in\mathcal{E}} = Z^\top z,$$

- $Z = \mathrm{blk}(z_1, z_2, \dots, z_m) \in \mathbb{R}^{dm\times m}$ - block diagonal matrix
- in order to determine rigidity, we need to analyze the Jacobian matrix of $\varsigma(p)$

# Graph rigidity - for digraphs 3/3

- the Jacobian matrix $\Longrightarrow$ the rigidity matrix:

$$R(p) = \frac{1}{2}\frac{\partial \varsigma(p)}{\partial p} \in \mathbb{R}^{m \times dn}$$

- can also be calculated as: $R(p) = Z^\top \mathbf{H}$

**What is rigidity?**

A representation $p$ is rigid if there exists $\varepsilon > 0$ such that for all realizations $p^\bullet$ of the distance set induced by $p$ and satisfying $\max_{i \in \mathcal{V}} \|p_i - p_i^\bullet\| < \varepsilon$ are congruent to $p$.

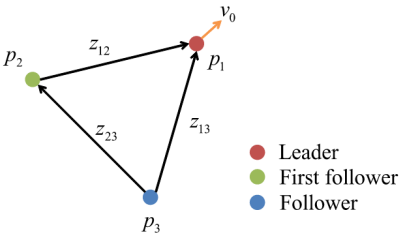
- rigidity matrix captures the infinitesimal changes in distances w.r.t. agent positions
- a framework $(\Gamma, p)$ with $N \geq d$ agents is infinitesimally rigid if:

$$\text{rank}(R(p)) = dN - \frac{d(d+1)}{2}$$

# Example: LFF 1/4 - description

Consider an acyclic LFF(Leader-First-Follower) formation:

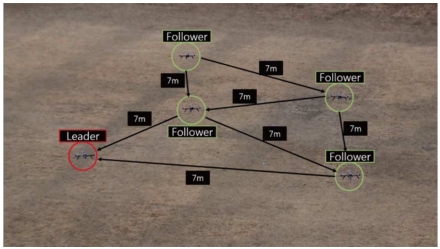
- agent $i = 1$ - *leader*
- agent $i = 2$ - *first-follower*
- the other agents $i \geq 3$ - *ordinary followers*
- $m = 1 + 2(n-2)$ edges
- the sets of neighbors for each agent are as follows:
  - $\mathcal{N}_1 = \emptyset$
  - $\mathcal{N}_2 = \{1\}$
  - $\mathcal{N}_i = \{i-2, i-1\}, \forall i \geq 3$
- all agents are in the plane $\Leftrightarrow$ $d = 2 \Rightarrow$ $p_i = [x_i\ y_i], \forall i$



# Example: LFF 1/4 - description

Consider an acyclic LFF(Leader-First-Follower) formation:

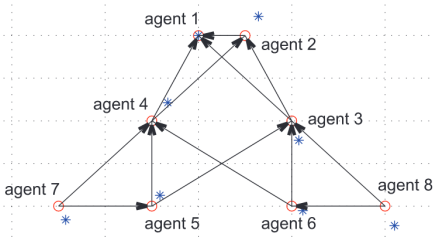
- agent $i = 1$ - *leader*
- agent $i = 2$ - *first-follower*
- the other agents $i \geq 3$ - *ordinary followers*
- $m = 1 + 2(n - 2)$ edges
- the sets of neighbors for each agent are as follows:
  - $\mathcal{N}_1 = \emptyset$
  - $\mathcal{N}_2 = \{1\}$
  - $\mathcal{N}_i = \{i - 2, i - 1\}, \forall i \geq 3$
- all agents are in the plane $\Leftrightarrow$ $d = 2 \Rightarrow$ $p_i = [x_i\ y_i], \forall i$



# Example: LFF 1/4 - description

Consider an acyclic LFF(Leader-First-Follower) formation:

- agent $i = 1$ - *leader*
- agent $i = 2$ - *first-follower*
- the other agents $i \geq 3$ - *ordinary followers*
- $m = 1 + 2(n - 2)$ edges
- the sets of neighbors for each agent are as follows:
  - $\mathcal{N}_1 = \emptyset$
  - $\mathcal{N}_2 = \{1\}$
  - $\mathcal{N}_i = \{i - 2, i - 1\}, \forall i \geq 3$
- all agents are in the plane $\Leftrightarrow$ $d = 2 \Rightarrow$ $p_i = [x_i\ y_i], \forall i$



# Example: LFF 2/4 - Rigidity analysis $N = 3$

$$H = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \Longrightarrow R(p) = Z^\top (H \otimes I_d) = \begin{bmatrix} z_{1x} & z_{1y} & -z_{1x} & -z_{1y} & 0 & 0 \\ z_{2x} & z_{2y} & 0 & 0 & -z_{2x} & -z_{2y} \\ 0 & 0 & z_{3x} & z_{3y} & -z_{3x} & -z_{3y} \end{bmatrix}$$

- $z_k = \begin{bmatrix} z_{kx} & z_{ky} \end{bmatrix}^\top \in \mathbb{R}^2, \quad k = 1:3$ are the relative offsets between agents
- for $d = 2, N = 3$ the rigidity condition becomes $\text{rank}(R(p)) = 3 \leftrightarrow$ there must be at least one non-zero 3rd order minor
- This condition is met for all representations $p \in \mathbb{R}^6$ where positions are non-collinear.
- $\Rightarrow$ the condition on the distances set $\bar{\mathcal{D}}$ to ensure rigidity:

$$\bar{\delta}_{12} \neq \bar{\delta}_{13} + \bar{\delta}_{23}$$

- the reasoning can be extended for $N \geq 4$: the condition being applied "locally", ensuring the non-colinearity of any tuple $(i-2, i-1, i)$

**Remark**

- for any formation in the plane($d = 2$) with $N$ agents the rigidity condition becomes $\text{rank}(R(p)) = 2N - 3$.
- since $R(p) \in \mathbb{R}^{2N-3 \times 2m}$ the condition imply $R(p)$ is a matrix of maximum rank.

# Example: LFF 3/4 - Dynamic analysis - Single Integrator case

- for a given formation we define the squared distance error $e_{ij}$ for a specific edge $k(i,j)$:

$$e_{ij} = e_{k(i,j)} = \|z_{k(i,j)}\|^2 - \bar{\delta}^2_{k(i,j)} = \|z_{ij}\|^2 - \bar{\delta}^2_{ij},$$

- $N \geq 3$ agents described by single integrator dynamics:

$$\dot{p} = \underbrace{O_{2N}}_{A} p + \underbrace{I_{2N}}_{B} u$$

- $u = \begin{bmatrix} u_1^\top & u_2^\top & \dots & u_N^\top \end{bmatrix}^\top \in \mathbb{R}^{2N}$ are the inputs, i.e. the velocities of each agent.
- $u_1 = v_0 \Leftrightarrow$ the *leader* is moving with constant velocity
- $u_i = \hat{v}_i + \alpha_i \sum_{j \in \mathcal{N}_i} e_{ij} z_{ij}, \quad \forall i \in \mathcal{V}$, where $\alpha_i > 0$, usually $\alpha_2 > 1$ and $\alpha_i = 1$
- **velocity estimator**: $\dot{\hat{v}}_i = \sum_{j \in \mathcal{N}_i} e_{ij} z_{ij}$
- $v_0 - \hat{v}_i \to 0, \forall i \in \mathcal{V}$ as $t \to \infty$
- for stability and convergence we employ a Lyapunov function:

$$V_i = \frac{1}{4} \sum_{i \in \mathcal{N}_i} e_{ij}^2 + \frac{1}{2} \|v_0 - \hat{v}_i\|^2.$$

# Example: LFF 4/4- Dynamic analysis - Double Integrator case

- $N \geq 3$ agents described by single integrator dynamics:

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \underbrace{\begin{bmatrix} O_{2N} & I_{2N} \\ O_{2N} & O_{2N} \end{bmatrix}}_{A} \begin{bmatrix} p \\ v \end{bmatrix} + \underbrace{\begin{bmatrix} O_{2N} \\ I_{2N} \end{bmatrix}}_{B} u$$

- $v \in \mathbb{R}^{2N}$ - velocity vector, $u \in \mathbb{R}^{2N}$ - acceleration/input vector
- $\ddot{p}_1 = a_0$ - leader has constant acceleration
- $u_i = \hat{a}_i + \alpha_i^p \sum_{j \in \mathcal{N}_i} e_{ij} z_{ij} + \alpha_i^v \sum_{j \in \mathcal{N}_i} \|v_i - v_j\|^2 (v_i - v_j), \forall i \in \mathcal{V}$
- $\alpha_i^p$ and $\alpha_i^v$ influence the rate of convergence
- **acceleration estimator**:

$$\dot{\hat{a}}_i = \sum_{j \in \mathcal{N}_i} e_{ij} z_{ij} + \sum_{j \in \mathcal{N}_i} \|v_i - v_j\|^2 (v_i - v_j)$$

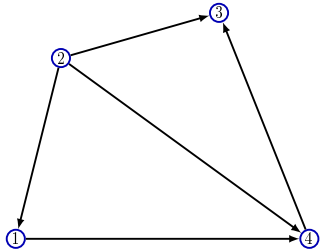
# Distance-based directed formation

Let $G = (V, E)$ - **directed**

- $p_i \in \mathbb{R}^d$ assigned to $i \in V$
- $(G, p)$ - directed framework in $\mathbb{R}^d$
- define $E^c := \{(i,j) | (j,i) \in E \text{ and } (i,j) \notin E\}$
- the underlying undirected graph of $G$ is $G^c = (V, E \cup E^c)$

**Idea**

~~Study the directed interaction graph $G$ via its undirected counterpart $G^c$.~~




- **rigid**
- **but** node 2 has **too** much responsability, (i.e., has to control the "lengths" of 3 edges)
- **then** the rigidity is not sufficient !

# Distance-based directed formation

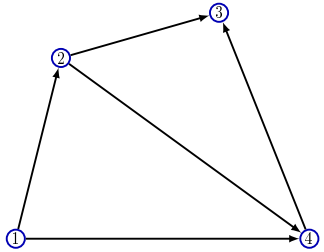
Let $G = (V, E)$ - **directed**

- $p_i \in \mathbb{R}^d$ assigned to $i \in V$
- $(G, p)$ - directed framework in $\mathbb{R}^d$
- define $E^c := \{(i,j) | (j,i) \in E \text{ and } (i,j) \notin E\}$
- the underlying undirected graph of $G$ is $G^c = (V, E \cup E^c)$

**Idea**

see graph persistence[a]

[a] Hendrickx, Anderson, Delvenne, and Blondel 2007.




- **rigid** and ***persistent***
- responsibility for controlling edge lengths is well distributed
- i.e., every node can control its outgoing edge length

# Outline

# Flocking

**Motivation**: many collective behaviors are indeed based on relatively simple interactions among individuals.

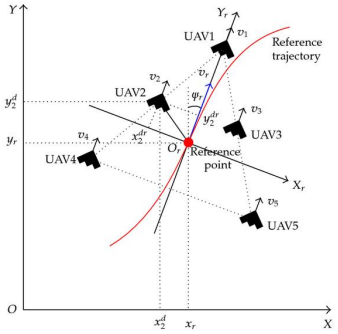
**Reynolds' rules**[a]:

- **Cohesion**: stay close to *nearby neighbors*
- **Separation**: avoid collisions *with nearby neighbors*
- **Alignment**: match velocity *with nearby neighbors*

**Alignment** $\Longrightarrow$ FLOCKING

*Usually*, implemented by means of **velocity consensus of agents**.

[a] Reynolds 1987.



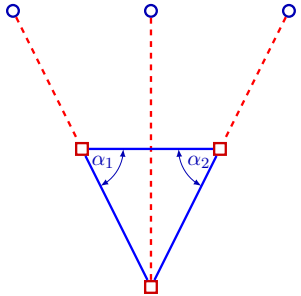
# Angle-based formation control

Let $N = 3$ agents:

- $\alpha$ instead of $p$
- $\alpha^\star = \begin{bmatrix} \alpha_1^\star & \alpha_2^\star & \alpha_3^\star \end{bmatrix}^\top$
- $\alpha_i^\star$ - the desired angle subtended at agent $i$ by the other agents
- $\alpha_1^\star + \alpha_2^\star + \alpha_3^\star = \pi$

**Remarks**

- agents do not share a common heading
- an agent measures the positive (negative) counter-clockwise (clockwise) from their local $p_i$-direction to agent $j$ angle: $\varphi_{ij} \in (-\pi, \pi)$, $\forall j \in \mathcal{N}_i$
- $\theta_i = |\varphi_{ik} - \varphi_{ij}|$ - angle subtended at agent $i$ by agents $k$ and $i$



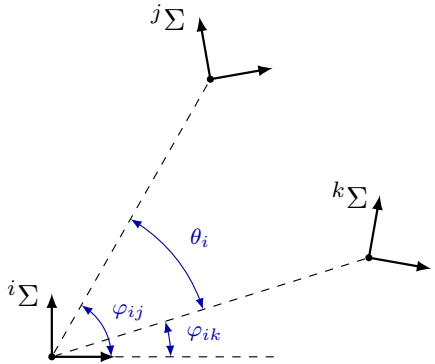
# Angle-based formation control

Let $N = 3$ agents:

- $\alpha$ instead of $p$
- $\alpha^\star = \begin{bmatrix} \alpha_1^\star & \alpha_2^\star & \alpha_3^\star \end{bmatrix}^\top$
- $\alpha_i^\star$ - the desired angle subtended at agent $i$ by the other agents
- $\alpha_1^\star + \alpha_2^\star + \alpha_3^\star = \pi$

**Remarks**

- agents do not share a common heading
- an agent measures the positive (negative) counter-clockwise (clockwise) from their local $p_i$-direction to agent $j$ angle: $\varphi_{ij} \in (-\pi, \pi)$, $\forall j \in \mathcal{N}_i$
- $\theta_i = |\varphi_{ik} - \varphi_{ij}|$ - angle subtended at agent $i$ by agents $k$ and $i$



# Angle-based formation control

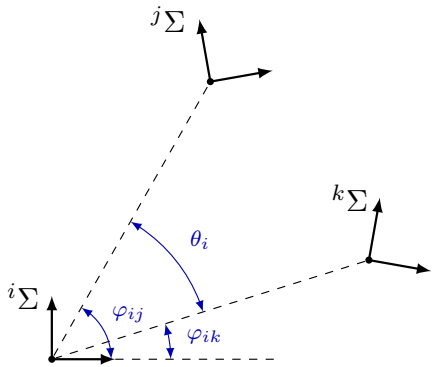
Let $N = 3$ agents:

- $\alpha$ instead of $p$
- $\alpha^\star = \begin{bmatrix} \alpha_1^\star & \alpha_2^\star & \alpha_3^\star \end{bmatrix}^\top$
- $\alpha_i^\star$ - the desired angle subtended at agent $i$ by the other agents
- $\alpha_1^\star + \alpha_2^\star + \alpha_3^\star = \pi$

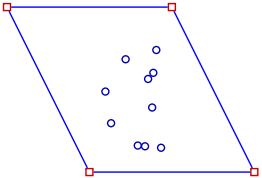
**Remarks**

- $\varphi_{ij} \in (-\pi, \pi),\ \forall j \in \mathcal{N}_i$
- $\theta_i = |\varphi_{ik} - \varphi_{ij}| \in (0, 2\pi)$

$$\alpha_i = \begin{cases} \theta_i, & \text{if } \theta_i \leq \pi \\ 2\pi - \theta_i, & \text{otherwise} \end{cases}$$



# Containment control



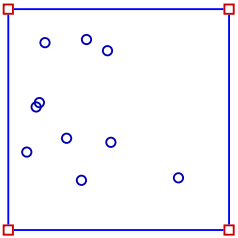
**Idea**

Follower agents are driven into the convex hull spanned by leader agents.

- leaders' behavior ← autonomously.
- followers ← **consensus protocol**

**The main benefit**: follower agents **do not** require expensive sensors.

# Containment control



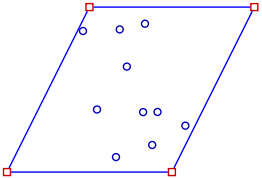
**Idea**

Follower agents are driven into the convex hull spanned by leader agents.

- leaders' behavior ← autonomously.
- followers ← **consensus protocol**

**The main benefit**: follower agents **do not** require expensive sensors.

# Containment control



**Idea**

Follower agents are driven into the convex hull spanned by leader agents.

- leaders' behavior ← autonomously.
- followers ← **consensus protocol**

**The main benefit**: follower agents **do not** require expensive sensors.

# Outline

# Conclusions

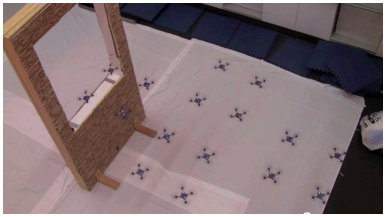
**Application and/or variation of *Formation control* :**

- Collaborative teams of robots
- Collision avoidance
- Connectivity maintenance
- Coverage
- etc.

There is a vast amount of the literature surveys, e.g.,

- R. Olfati-Saber, J. A. Fax, and R. M. Murray (2007). "Consensus and cooperation in networked multi-agent systems". In: *Proceedings of the IEEE* 95.1, pp. 215–233
- K.-K. Oh, M.-C. Park, and H.-S. Ahn (2015). "A survey of multi-agent formation control". In: *Automatica* 53, pp. 424–440

and so many others.




*Complex networks/formations: the study of complex patterns of interaction*

# Conclusions

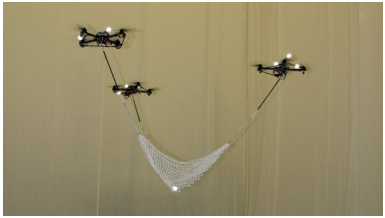
**Application and/or variation of *Formation control* :**

- Collaborative teams of robots
- Collision avoidance
- Connectivity maintenance
- Coverage
- etc.

There is a vast amount of the literature surveys, e.g.,

- R. Olfati-Saber, J. A. Fax, and R. M. Murray (2007). "Consensus and cooperation in networked multi-agent systems". In: *Proceedings of the IEEE* 95.1, pp. 215–233
- K.-K. Oh, M.-C. Park, and H.-S. Ahn (2015). "A survey of multi-agent formation control". In: *Automatica* 53, pp. 424–440

and so many others.



*Complex networks/formations: the study of complex patterns of interaction*