Graph Theory

— The absolute basics —

— . . . and some example decentralization problems —

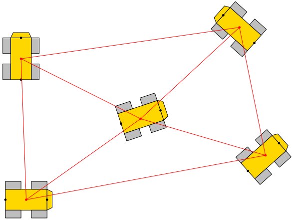
Charles C. Cossette

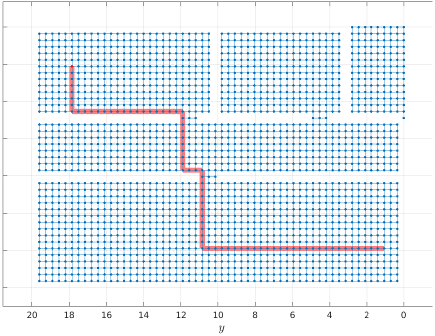
McGill University, Department of Mechanical Engineering

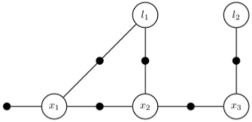
June 9, 2022

Introduction

Graph theory is ubiquitous in robotics and computer science.







Multi-robot problems, path planning, and batch estimation factor graphs[1] all make use of graph theory.

[1] Figure taken from https://gtsam.org/tutorials/intro.html.

Introduction

The examples we will cover are

1. decentralized consensus (largely taken from [1]),
2. decentralized multi-robot formation control (largely taken from [2]).

Some initial definitions

Definition (set)

A *set* is an **unordered list** of **unique** mathematical objects. A set is denoted with curly $\{\}$ brackets.

$$\mathcal{S} = \{1, 2, 3\} = \{2, 1, 1, 3\}.$$

Definition (ordered pair)

An *ordered pair, 2-tuple*, or simply a *pair* is a list of mathematical objects where the **order of the objects has significance.** An ordered pair or any n-tuple is denoted with circular $()$ brackets.

$$\mathcal{P} = (a, b) \neq (b, a).$$

Definition (Cartesian product)

The Cartesian product of two sets $\mathcal{A} \times \mathcal{B}$ is the set of all ordered pairs of the elements of $\mathcal{A}$ and $\mathcal{B}$.

$$\mathcal{A} \times \mathcal{B} = \{(a, b) \mid a \in \mathcal{A},\ b \in \mathcal{B}\}$$

Some initial definitions

Notation (number of elements)

The number of elements in a set $\mathcal{S}$ is denoted $|\mathcal{S}|$.

Graphs: Definitions

Definition (Directed Graph)

A *directed graph* or *digraph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of

- a set of nodes $\mathcal{V} = \{1, \ldots N\}$ and
- a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.

The set of edges consists of ordered pairs

$$(i, j) \in \mathcal{E} \tag{1}$$

between some of the nodes.

- j is called the *child* of i .
- i is called the *parent* of j.
- Arrows usually point from the parent to the child.

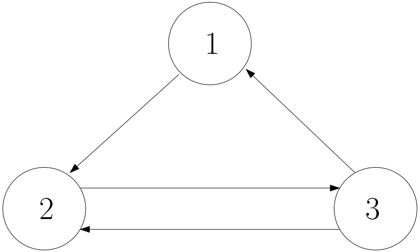


Figure 1: A simple graph with three nodes.

$$\mathcal{V} = \{1, 2, 3\}, \mathcal{E} = \{(1, 2), (2, 3), (3, 1), (3, 2)\}$$

Graphs: Definitions

Definition (Neighbors)

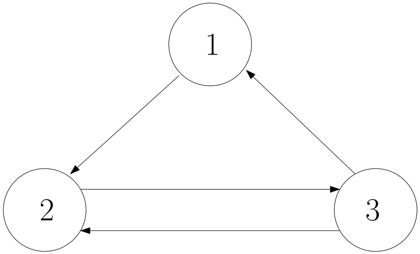
The set of *out-neighbors* or *successors* $\mathcal{N}_i^{\text{out}}$ of i is equivalent to all of its children, and is defined as,

$$\mathcal{N}_i^{\text{out}} = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}\}.$$

The set of *in-neighbors* or *predecessors* $\mathcal{N}_i^{\text{in}}$ of i is equivalent to all of its parents, and is defined as,

$$\mathcal{N}_i^{\text{in}} = \{j \in \mathcal{V} \mid (j,i) \in \mathcal{E}\}.$$

Henceforth, the shorthand $\mathcal{N}_i$ will refer to the out-neighbors.



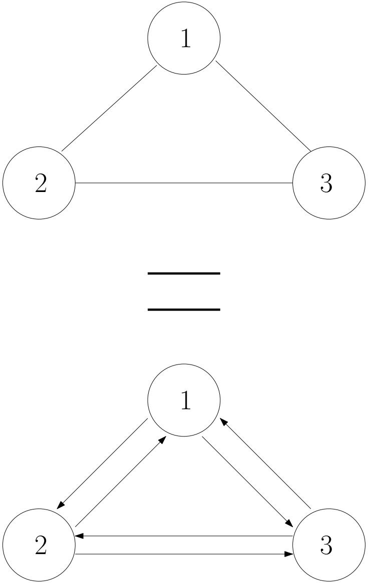
$$\mathcal{N}_1 = \{2\}$$
$$\mathcal{N}_2 = \{3\}$$
$$\mathcal{N}_3 = \{1,2\}$$

Graphs: Definitions

Definition (Undirected graph)

An undirected graph can be viewed as a directed graph, where any pair of connected nodes have an edge in both directions. That is, if

$$(i,j) \in \mathcal{E} \quad \text{then} \quad (j,i) \in \mathcal{E}.$$



Definition (Subgraph)

A subgraph $\mathcal{S} = (\mathcal{V}_s, \mathcal{E}_s)$ of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a subset of nodes of $\mathcal{G}$, and all edges in $\mathcal{G}$ connecting pairs of nodes in that subset. That is,

$$\mathcal{V}_s \subset \mathcal{V},$$

$$\mathcal{E}_s = \{(i,j) | (i,j) \in \mathcal{E} \text{ and } i,j \in \mathcal{V}_s\}.$$

Graphs: Definitions

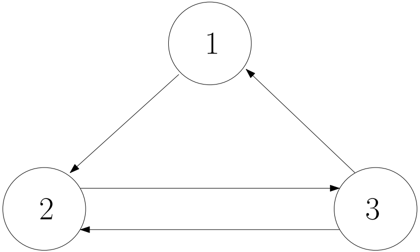
Definition (Directed path)

A *directed path* of $\mathcal{G}$ is a sequence of the form

$$(i, j),\ (j, k),\ \dots\ ,\ (\ell, m). \tag{2}$$

Definition (Strongly connected)

A graph is said to be *strongly connected* if there is a directed path from any node to any other node.



Definition (Strongly connected components)

A *strongly connected component* of $\mathcal{G}$ is a subgraph that is strongly connected, and the addition of other nodes/edges to the subgraph **will** break strong-connectedness.

Weighted Graphs

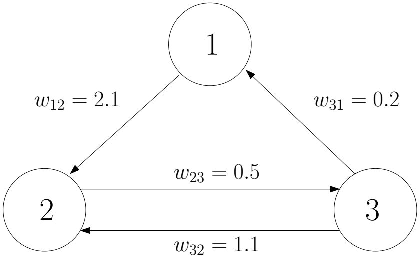
- It is possible to associate *weights* with the edges in a graph. That is, the weight $w_{ij} \in \mathbb{R}_{\geq 0}$ can be associated with the edge (i, j).

- The weights are defined such that

$$w_{ij} > 0 \text{ if } (i, j) \in \mathcal{E}$$

  and $w_{ij} = 0$ otherwise.

- For unweighted graphs, one can assume all non-zero weights are equal to 1.



Matrix Representations of Graphs: Adjacency Matrix

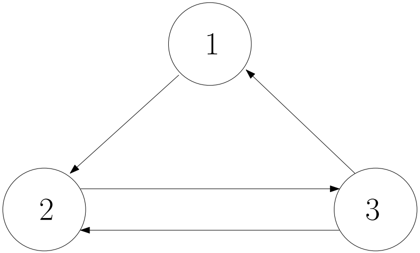
Definition (Adjacency Matrix)

The *adjacency matrix* $\mathbf{A}$ is a $|\mathcal{V}| \times |\mathcal{V}|$ matrix with elements $a_{ij} = w_{ij}$.

- That is,

$$\mathbf{A} = \begin{bmatrix} 0 & w_{12} & \cdots & w_{1N} \\ w_{21} & 0 & & \\ \vdots & & & \\ w_{N1} & & & 0 \end{bmatrix},$$

  where $w_{ii} = 0$ if there is no self-edge.
- For undirected graphs, the adjacency matrix is symmetric.



$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

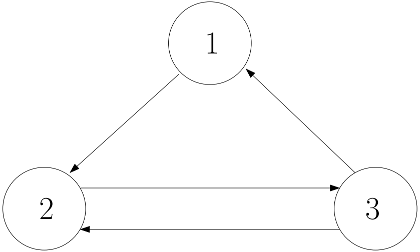
Matrix Representations of Graphs: Incidence Matrix

Definition (Incidence Matrix)

The *incidence matrix* $\mathbf{\Pi}$ is a $|\mathcal{V}| \times |\mathcal{E}|$ matrix, where each column corresponds to an edge. If a certain column k corresponds to the edge (i, j) then $\pi_{ik} = -1$, and $\pi_{jk} = 1$.

- The column ordering is arbitrary.
- By definition, $\mathbb{1}^\mathsf{T}\mathbf{\Pi} = \mathbf{0}$, where $\mathbb{1}$ is a column of 1's.

$$\mathbb{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$



For columns corresponding to edges $(1,2), (2,3), (3,1), (3,2) \ldots$

$$\mathbf{\Pi} = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 \\ 0 & 1 & -1 & -1 \end{bmatrix}.$$

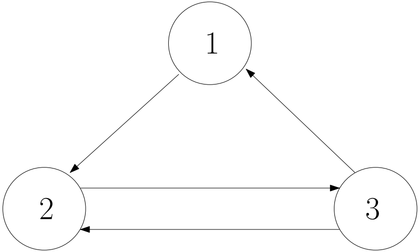
Matrix Representations of Graphs: Degree Matrix

Definition (Degree Matrix)

The *degree matrix* is defined as

$$\mathbf{D} = \mathrm{diag}\Big(\sum_{k \in \mathcal{N}_1} w_{1k}\,,\ \ldots\,,\ \sum_{k \in \mathcal{N}_N} w_{Nk}\Big).$$

That is, it is a diagonal matrix containing the sum of each node's edges' weights.



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

Matrix Representations of Graphs: Laplacian Matrix

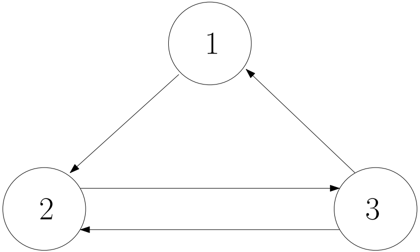
Definition (Laplacian Matrix)

The *Laplacian* matrix $\mathbf{L}$ is an $|\mathcal{V}| \times |\mathcal{V}|$ matrix with elements given by [3]

$$\ell_{ij} = \begin{cases} \sum_{k \in \mathcal{N}_i} w_{ik}, & \text{if } i = j \\ -w_{ij}, & \text{if } i \neq j \end{cases}.$$

- It can be shown that

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \tag{3}$$



$$\mathbf{L} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Some Properties of the Laplacian Matrix

Trivial Eigenvalue

Since row sums of $\mathbf{L}$ are always zero, $\mathbf{L}\mathbb{1} = \mathbf{0}$. It follows that $\lambda = 0$ and $\mathbb{1}$ are **always** an eigenvalue and eigenvector of $\mathbf{L}$, respectively. This is called the *trivial eigenvalue*.

Eigenvalues and Connectivity [1]

Let $\mathcal{G}$ be a strongly connected graph with N nodes. Then $\lambda = 0$ is the single zero eigenvalue of $\mathbf{L}$ (i.e., no repeated zero eigenvalues), and $\mathrm{rank}(\mathbf{L}) = N - 1$.

If $\mathcal{G}$ has c strongly connected components, then the trivial eigenvalue will have multiplicity c and $\mathrm{rank}(\mathbf{L}) = N - c$.

Non-negative real part of eigenvalues [1]

All nontrivial eigenvalues of $\mathbf{L}$ have positive real parts.

Consensus

In a team of robots or agents, *consensus* refers to the robots all agreeing on a certain quantity of interest.

Examples:

- The agents need to achieve consensus on the room temperature, after each taking a noisy measurement.
- Robots needing to achieve consensus on the heading direction and velocity to travel in.
- Robots needing to agree on a "rendezvous" location.

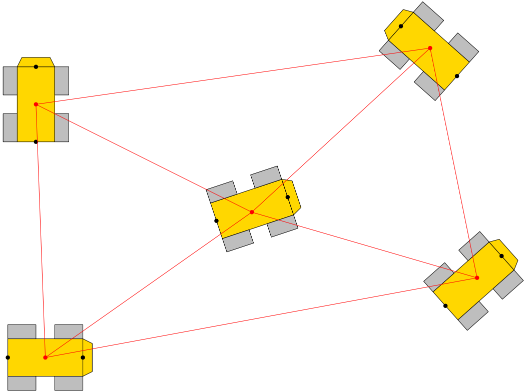


Figure 2: A team of autonomous ground vehicles, with communication links represented as a graph.

Consensus

- Consider N agents, each possessing some sort of consensus state variable x_i.
- The task is to find some sort of consensus update law u_i where

  $$\dot{x}_i(t) = u_i, \qquad x_i(0) = x_{0_i}, \qquad i = 1, \ldots, N,$$

  such that consensus is achieved, meaning all x_i converge to the same value,

  $$x_1 = x_2 = \ldots = x_N \triangleq \alpha.$$

- The communication links are represented by the weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
- The update law for agent i, $u_i = u_i(x_{j \in \mathcal{N}_i})$, must strictly be a function of information from the agent's neighbors.

Consensus

- In [1], the following consensus law is proposed

$$\dot{x}_i = -\sum_{j \in \mathcal{N}_i} w_{ij}(x_i - x_j). \tag{4}$$

- Consider, for example, the first agent

$$\begin{aligned}
\dot{x}_1 &= -\sum_{j \in \mathcal{N}_1} w_{1j}(x_1 - x_j) \\
&= -\left(w_{12}(x_1 - x_2) + \ldots + w_{1N}(x_1 - x_N)\right) \\
&= -\begin{bmatrix} \left(\sum_{j \in \mathcal{N}_1} w_{1j}\right) & -w_{12} & \ldots & -w_{1N} \end{bmatrix} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}}_{\triangleq \mathbf{x}} \\
&= -\left[\ell_{11}\ \ell_{12}\ \ldots\ \ell_{1N}\right] \mathbf{x},
\end{aligned}$$

  where ℓ_{ij} are the elements of the Laplacian matrix $\mathbf{L}$ of the graph $\mathcal{G}$.

- The collective dynamics are therefore

$$\dot{\mathbf{x}} = -\mathbf{L}\mathbf{x}.$$

Consensus

- Since all eigenvalues of $-\mathbf{L}$ are in the open left-hand plane, the system $\dot{\mathbf{x}} = -\mathbf{L}\mathbf{x}$ is stable.
- Assuming that the interaction graph is strongly connected, then the system will only have a single non-zero equilibrium point.
- Let $\boldsymbol{\gamma} = [\gamma_1 \dots \gamma_N]^\mathsf{T}$ be a left-eigenvector of $\mathbf{L}$ (i.e. such that $\boldsymbol{\gamma}^\mathsf{T}\mathbf{L} = \mathbf{0}$). The term

$$y(t) \triangleq \boldsymbol{\gamma}^\mathsf{T}\mathbf{x}(t) \tag{5}$$

  is invariant since $\dot{y}(t) = -\boldsymbol{\gamma}^\mathsf{T}\mathbf{L}\mathbf{x}(t) = \mathbf{0}$.
- Hence, the final value that all the states converge to can be obtained from

$$\lim_{t\to\infty} y(t) = y(0) \tag{6}$$

$$\boldsymbol{\gamma}^\mathsf{T}\mathbb{1}\alpha = \boldsymbol{\gamma}^\mathsf{T}\mathbf{x}_0 \tag{7}$$

$$\alpha = \frac{\boldsymbol{\gamma}^\mathsf{T}\mathbf{x}_0}{\sum_{i=1}^{N}\gamma_i} \tag{8}$$

Consensus

In the case of an undirected graph, $\gamma = \mathbb{1}$ and

$$\alpha = \frac{\sum_{i=1}^{N} x_{0_i}}{N}, \tag{9}$$

i.e. the agents each converge to the average initial condition!

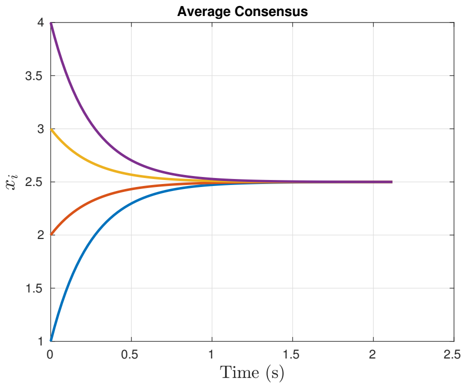
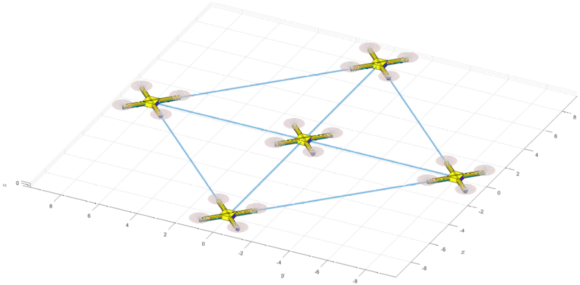


Figure 3: Each agent's individual consensus variable over time.

Formation control

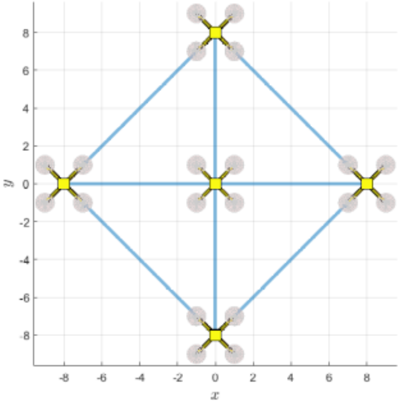
- We seek to maintain a group of robots in a specified geometry.



- This can be convenient for coordinating groups of robots.
  - Specify only the position and attitude of the group, instead of each agent.

Formation Control

- We can stay in formation simply by controlling the inter-robot distances.
- If all inter-robot distances are constant, the formation is said to be *rigid*.
- Therefore, we must design a controller that will keep all these inter-robot distances constant.



Formation Control

- Not all inter-robot distances must be regulated, but we still require a minimum.
- We can define an **undirected graph** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes are the robots, and there is an edge if the distance between two robots is regulated.
- **Note:** now that we are working with an undirected graph, if $(i, j) \in \mathcal{E}$ then so is (j, i).

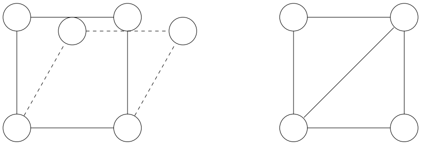


Figure 4: Flexible (left) and rigid (right) formations.

Formation control

- Let $\mathbf{r}_a^{iw}$ denote the position of robot i, relative to some common reference point w, in some common frame a.
- Let $\mathbf{r} = [\mathbf{r}_a^{1w^\mathsf{T}}, \ldots, \mathbf{r}_a^{Nw}]^\mathsf{T}$. Define the *rigidity function* as the list of all squared inter-robot distances[2]:

$$\phi(\mathbf{r}) = \begin{bmatrix} \vdots \\ \frac{1}{2}\left\|\mathbf{r}_a^{ij}\right\|^2 \\ \vdots \end{bmatrix} \in \mathbb{R}^{|\mathcal{E}|}, \qquad (i, j) \in \mathcal{E}. \tag{10}$$

- Approximating to first order,

$$\phi(\mathbf{r} + \delta\mathbf{r}) \approx \phi(\mathbf{r}) + \mathbf{R}(\mathbf{r})\delta\mathbf{r} \tag{11}$$

where $\mathbf{R}(\mathbf{r}) = \frac{\partial\phi(\mathbf{r})}{\partial\mathbf{r}}$ will be called the *rigidity matrix*.
- We want the inter-robot distances to remain constant with a small change in position $\delta\mathbf{r}$, hence $\phi(\mathbf{r} + \delta\mathbf{r}) = \phi(\mathbf{r})$, leading to

$$\mathbf{R}(\mathbf{r})\delta\mathbf{r} = \mathbf{0}. \tag{12}$$

---

[2] Only include one of either (i, j) or (j, i) in the rigidity function.

Formation control

- For n dimensions (i.e. $\mathbf{r}_a^{iw} \in \mathbb{R}^n$), the rigidity matrix will be $|\mathcal{E}| \times n|\mathcal{V}|$.
- In 3D, there are 6 degrees of freedom (3 translation, 3 rotation) that the entire formation can move, without changing inter-robot distances.
- Hence, we require that

$$\mathrm{rank}(\mathbf{R}(\mathbf{r})) = 3|\mathcal{V}| - 6 \tag{13}$$

  for a formation to be rigid.

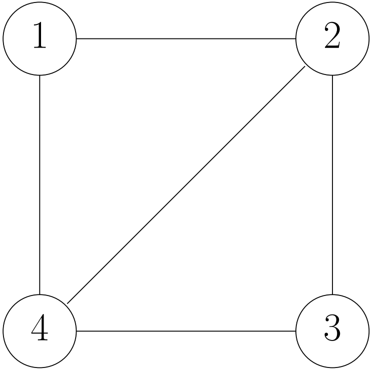
Definition (Minimally rigid)

A rigid graph is said to be *minimally rigid* if the removal of a single edge causes it to lose rigidity.

In this case, in 3D, $|\mathcal{E}| = 3|\mathcal{V}| - 6$ and $\mathbf{R}(\mathbf{r})$ will be full row rank.

Formation control: Rigidity matrix example

$$\mathbf{R}(\mathbf{r}) = \begin{bmatrix} (\mathbf{r}_1-\mathbf{r}_2)^\mathsf{T} & -(\mathbf{r}_1-\mathbf{r}_2)^\mathsf{T} & & \\ (\mathbf{r}_1-\mathbf{r}_4)^\mathsf{T} & & & -(\mathbf{r}_1-\mathbf{r}_4)^\mathsf{T} \\ & (\mathbf{r}_2-\mathbf{r}_3)^\mathsf{T} & -(\mathbf{r}_2-\mathbf{r}_3)^\mathsf{T} & \\ & (\mathbf{r}_2-\mathbf{r}_4)^\mathsf{T} & & -(\mathbf{r}_2-\mathbf{r}_4)^\mathsf{T} \\ & & (\mathbf{r}_3-\mathbf{r}_4)^\mathsf{T} & -(\mathbf{r}_3-\mathbf{r}_4)^\mathsf{T} \end{bmatrix}$$



where

$$\mathbf{r}_i \triangleq \mathbf{r}_a^{iw}$$

Formation control

- Consider the error in squared distances $\mathbf{e}(t) = (\phi(\mathbf{r}(t)) - \phi_{\mathrm{des}})$, as well as the Lyapunov function candidate

$$V(t) = \frac{1}{2}\mathbf{e}(t)^{\mathsf{T}}\mathbf{e}(t).$$

- The time derivative of $V(t)$ is given by

$$\begin{aligned} \dot{V}(t) &= \frac{\partial V}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial \mathbf{r}} \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} \\ &= \mathbf{e}(t)^{\mathsf{T}}\mathbf{R}(\mathbf{r})\dot{\mathbf{r}}. \end{aligned} \tag{14}$$

- If we assume that we control robot velocities then $\dot{\mathbf{r}} = \mathbf{u}$, and if we choose the control law to be $\mathbf{u} = -\mathbf{R}(\mathbf{r})^{\mathsf{T}}\mathbf{e}(t)$ then

$$\dot{V}(t) = -\mathbf{e}(t)^{\mathsf{T}}\mathbf{R}(\mathbf{r})\mathbf{R}(\mathbf{r})^{\mathsf{T}}\mathbf{e}(t) \leq 0$$

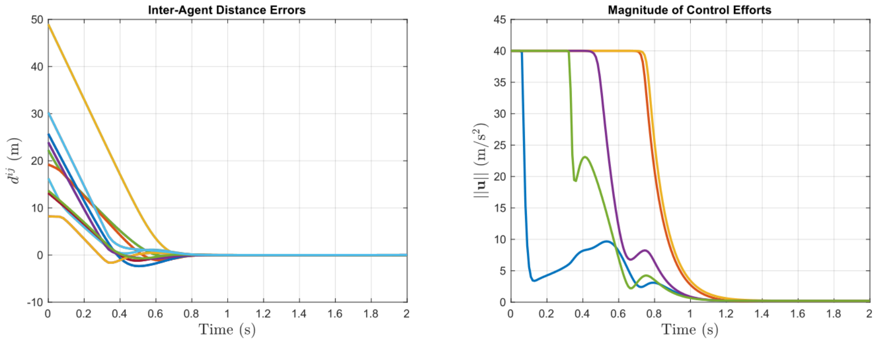
  since $\mathbf{R}(\mathbf{r})\mathbf{R}(\mathbf{r})^{\mathsf{T}}$ is positive semi-definite.

- If the graph is minimally rigid, then $\mathbf{R}(\mathbf{r})\mathbf{R}(\mathbf{r})^{\mathsf{T}}$ is positive definite and $\mathbf{e}(t) \to \mathbf{0}$ as $t \to \infty$.

Formation control

- The "stacked" control law is $\mathbf{u} = -\mathbf{R}(\mathbf{r})^\mathsf{T}\mathbf{e}(t)$ where $\mathbf{u} = [\mathbf{u}_1 \; \dots \; \mathbf{u}_N]^\mathsf{T}$
- Breaking this down into components we have

$$\mathbf{u}_i = -\frac{1}{2} \sum_{j \in \mathcal{N}_i} \left(\left\| \mathbf{r}_a^{ij} \right\|^2 - {d_{\text{des}}^{ij}}^2 \right) \mathbf{r}_a^{ij}. \tag{15}$$



Just the basics! Some other terms to watch out for

- Trees, spanning trees, forests
- Cliques
- Connectivity, algebraic connectivity

Searching Algorithms (shortest path algorithms)

- Breadth-first search
- Depth-first search
- Dijkstra's algorithms
- A* search

References

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[2] M. de Queiroz, X. Cai, and M. Feemster, *Formation Control of Multi-Agent Systems: A Graph Rigidity Approach*. Chichester, West Sussex, United Kingdom: John Wiley & Sons, Ltd, 2019.

[3] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Princeton University Press, 2014.

Theorem (Non-negative real part of eigenvalues)

For a directed weighted graph, all eigenvalues of $\mathbf{L}$ *have non-negative real part.*

Proof.

The proof follows directly from use of Gershgorin's disk theorem, which states that

$$\operatorname{spec}(\mathbf{L}) \subset \bigcup_{i \in \{1,\ldots,n\}} \left\{ z \in \mathbb{C} \,\middle|\, \|z - \ell_{ii}\| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |\ell_{ij}| \right\}.$$

For the Laplacian matrix,

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} |\ell_{ij}| = \ell_{ii} \geq 0$$

since the weights are all strictly positive. Therefore, all of the Gershgorin disks contain the origin, but are strictly in the closed right-hand plane. Therefore, all eigenvalues of $\mathbf{L}$ lie in the closed right-hand plane, and thus have non-negative real component. □