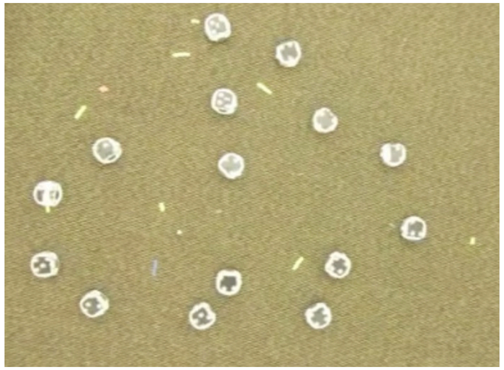
Networked Control
Modeling, Design, and Applications

Magnus Egerstedt

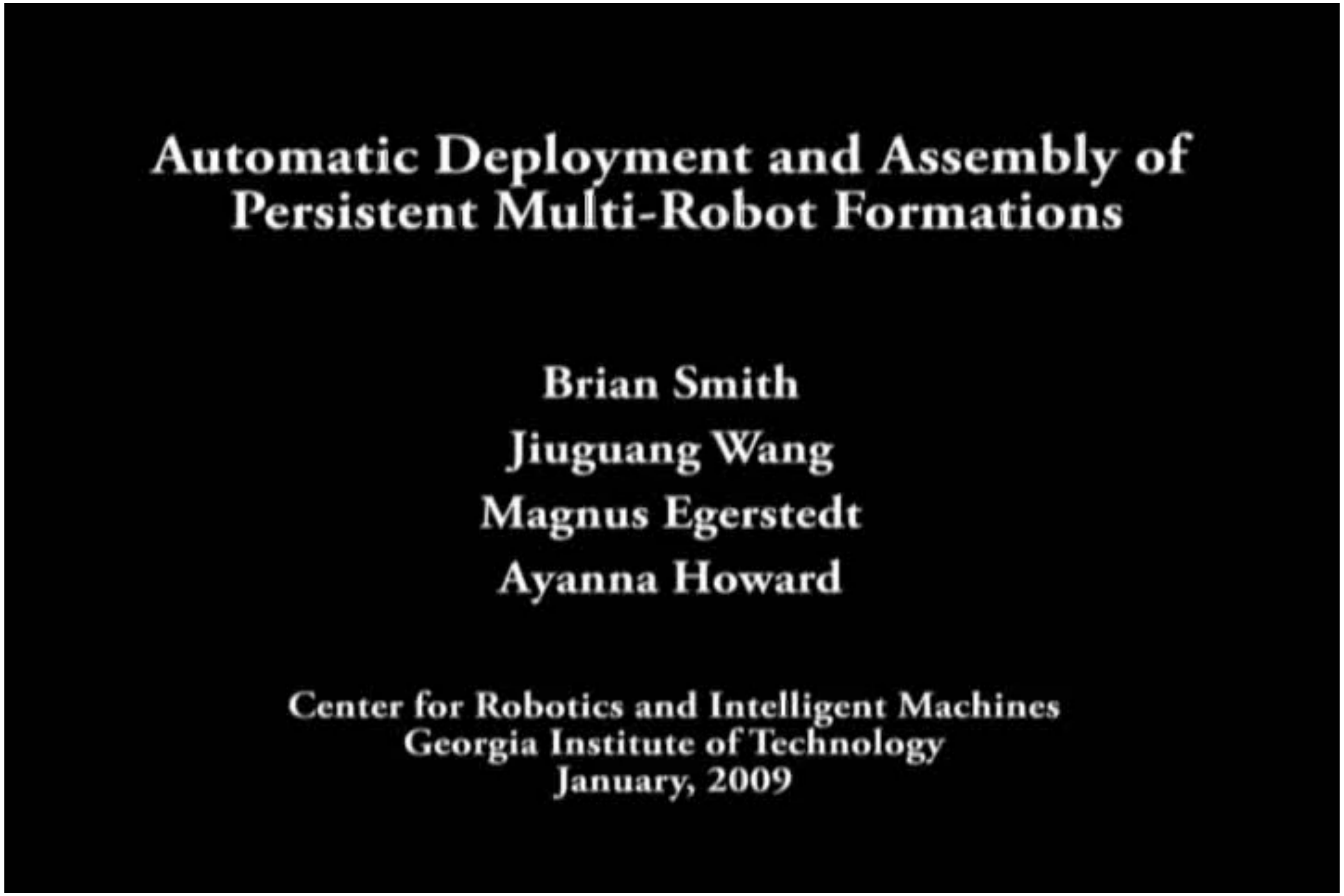
GRITSLab
Electrical and Computer Engineering
Georgia Institute of Technology
www.ece.gatech.edu/~magnus

Outline:

1. Graph-Based Control
2. Multi-Agent Networks
3. Control of Robot Teams
4. Sensor Networks

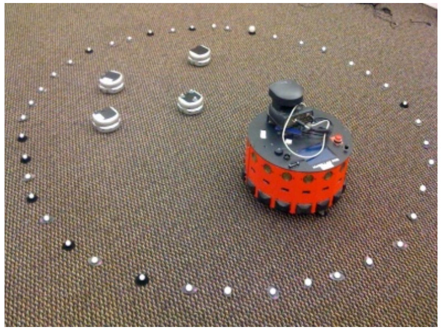


A Mood Picture

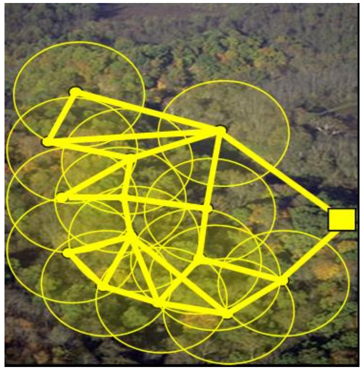


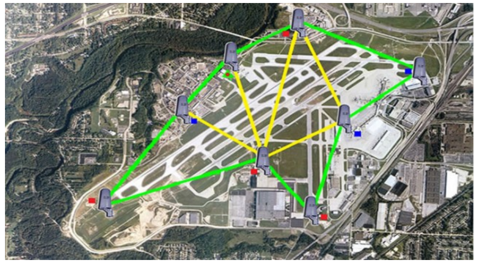
Ruining the Mood…

Application Domains

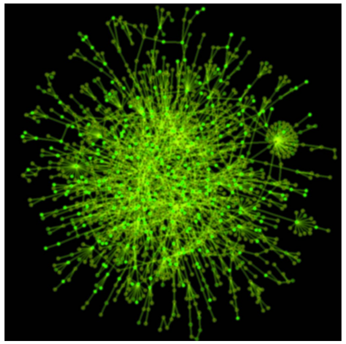


Multi-agent robotics

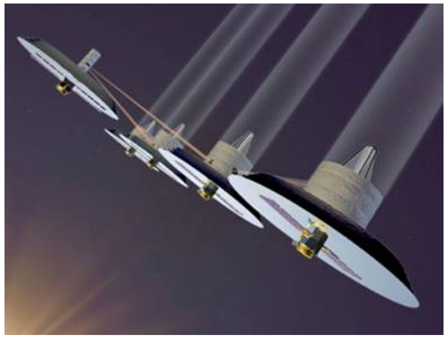




Sensor and communications networks



Biological networks



Coordinated control

The Mandatory Bio-Slide

- As sensor webs, large-scale robot teams, and networked embedded devices emerge, algorithms are needed for inter-connected systems with ***limited communication, computation, and sensing capabilities***



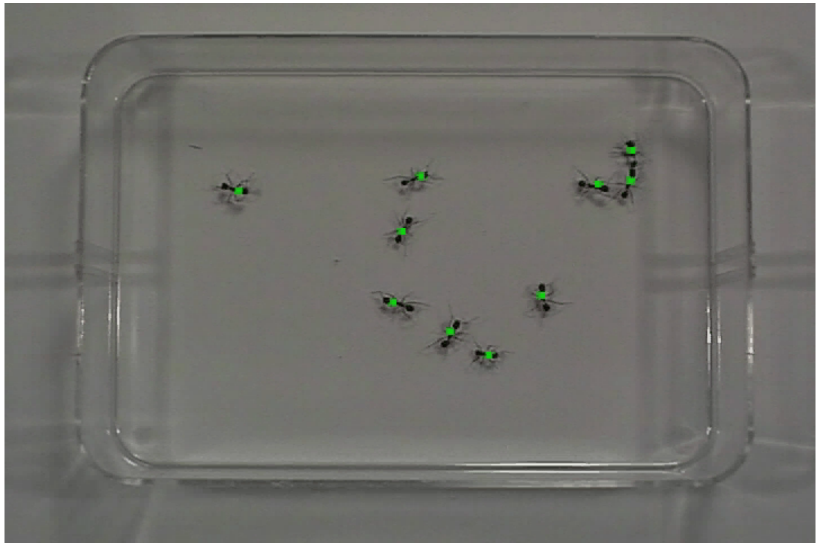


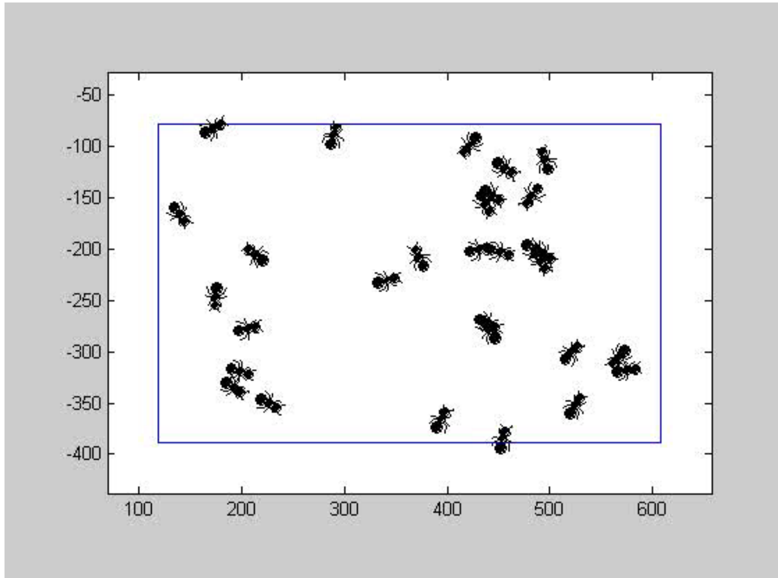
- How to effectively control such systems?
  - What is the correct model?
  - What is the correct mode of interaction?
  - Does every individual matter?

The Starting Point

SESSION 1
GRAPH-BASED CONTROL

Why I Started Caring About Multi-Agent Systems



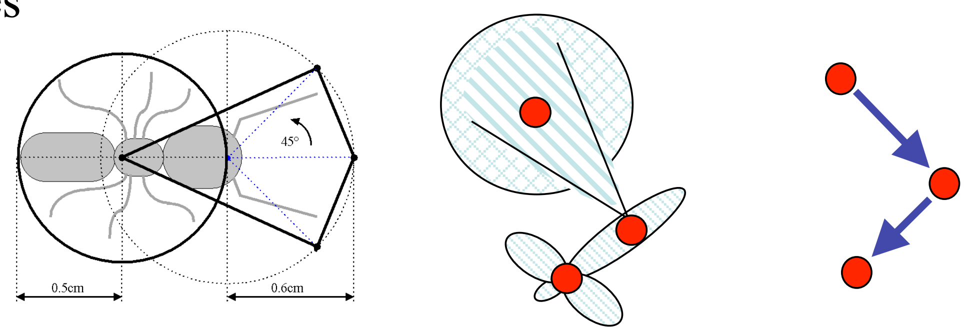


?

"They look like ants."
– Stephen Pratt, Arizona State University

Graphs as Network Abstractions

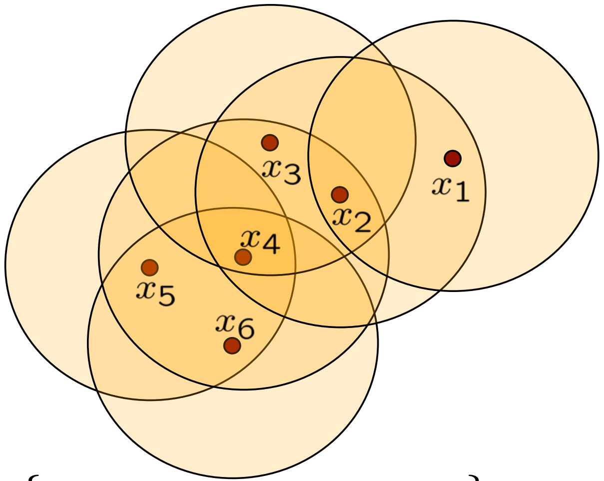
- A networked sensing and actuation system consists of
  - **NODES** - physical entities with limited resources (computation, communication, perception, control)
  - **EDGES** - virtual entities that encode the flow of information between the nodes

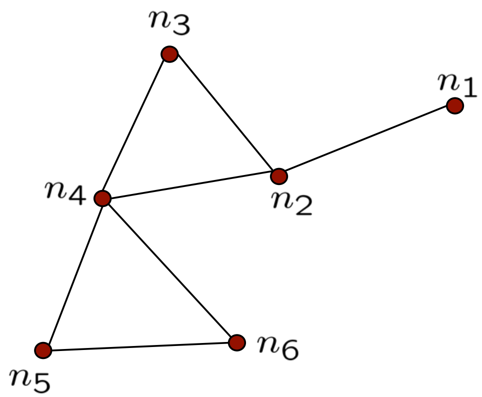


- The "right" mathematical object for characterizing such systems at the network-level is a **GRAPH**
  - Purely ***combinatorial*** object (no ***geometry*** or ***dynamics***)
  - The characteristics of the information flow is abstracted away through the (possibly weighted and directed) edges

Graphs as Network Abstractions

- The connection between the combinatorial graphs and the geometry of the system can for instance be made through geometrically defined edges.
- Examples of such proximity graphs include **disk-graphs**, **Delaunay graphs, visibility graphs**, and **Gabriel graphs** [1].

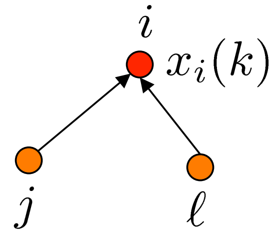




$$\mathcal{N} = \{n_1, n_2, n_3, n_4, n_5, n_6\}$$
$$\mathcal{E} = \{(n_1, n_2), (n_2, n_3), (n_3, n_4), (n_2, n_4), (n_4, n_5), (n_4, n_6), (n_5, n_6)\}$$

The Basic Setup

- $x_i(k) =$ "state" at node i at time k
- $N_i(k) =$ "neighbors" to agent i



- Information "available to agent i

$$I_i^c(k) = \{x_j(k) \mid j \in N_i(k)\}$$ ← common ref. frame (comms.)

or

$$I_i^r(k) = \{x_i(k) - x_j(k) \mid j \in N_i(k)\}$$ ← relative info. (sensing)

- Update rule:

$$x_i(k+1) = F_i(x_i(k), I_i(k))$$ ← discrete time

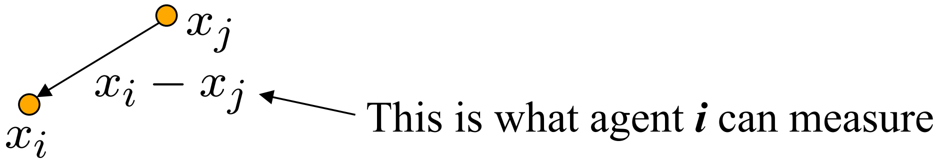
or

$$\dot{x}_i(t) = F_i(x_i(t), I_i(t))$$ ← continuous time

- ***How pick the update rule?***

Rendezvous – A Canonical Problem

- Given a collection of mobile agents who can only measure the relative displacement of their neighbors (no global coordinates)



- Problem: Have all the agents meet at the same (unspecified) position

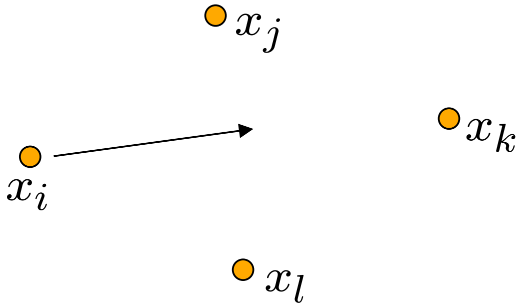
- If there are only two agents, it makes sense to have them drive towards each other, i.e.

$$\dot{x}_1 = -\gamma_1(x_1 - x_2)$$
$$\dot{x}_2 = -\gamma_2(x_2 - x_1)$$

- If $\gamma_1 = \gamma_2$ they should meet halfway

Rendezvous – A Canonical Problem

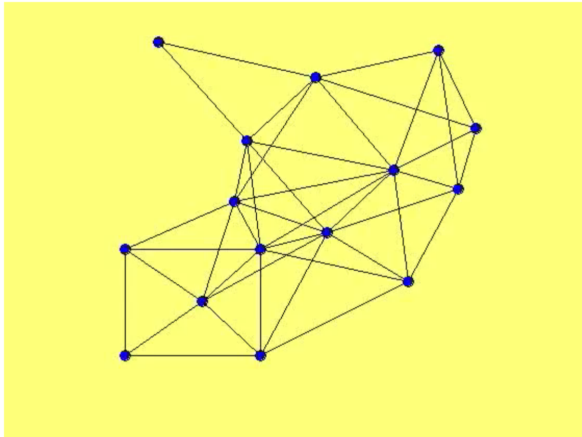
- If there are more than two agents, they should probably aim towards the centroid of their neighbors (or something similar)



$$\dot{x}_i = -\gamma \sum_{j \in \mathcal{N}_i} (x_i - x_j)$$

Fact [2-4]: As long as the graph is connected (iff), the ***consensus equation*** drives all agents to the same state value

$$\lim_{t \to \infty} x_i(t) = \bar{x} = \frac{1}{N} \sum_{j=1}^{N} x_j(0)$$



Algebraic Graph Theory

- To show this, we need some tools…
- Algebraic graph theory provides a bridge between the combinatorial graph objects and their matrix representations
  - **Degree matrix**:

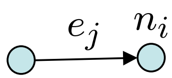
$$D = \mathrm{diag}(\deg(n_1), \ldots, \deg(n_N))$$

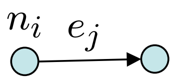
  - **Adjacency matrix**:

$$A = [a_{ij}], \;\; a_{ij} = \begin{cases} 1 & \text{if} \\ 0 & \text{o.w.} \end{cases}$$

  - **Incidence matrix** (directed graphs):

$$\mathcal{I} = [\iota_{ij}], \;\; \iota_{ij} = \begin{cases} 1 & \text{if} \\ -1 & \text{if} \\ 0 & \text{o.w.} \end{cases}$$

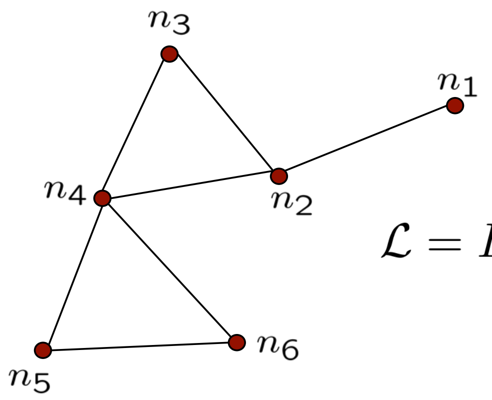




  - **Graph Laplacian**:

$$\mathcal{L} = D - A = \mathcal{I}\mathcal{I}^T$$

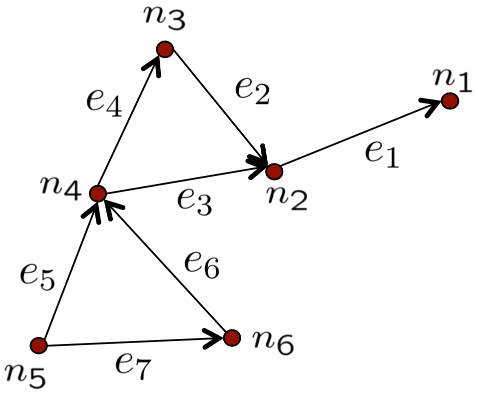
Algebraic Graph Theory - Example



$$\mathcal{L} = D - A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Algebraic Graph Theory - Example



$$\mathcal{I} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

$$\mathcal{L} = \mathcal{I}\mathcal{I}^T = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

The Consensus Equation

- The reason why the graph Laplacian is so important is through the already seen "consensus equation"

$$\dot{x}_i = -\sum_{j \in \mathcal{N}_i} (x_i - x_j), \;\; i = 1, \ldots, N$$

or equivalently (W.L.O.G. scalar agents)

$$\left.\begin{array}{l} \dot{x}_i = -\mathrm{deg}(n_i) x_i + \sum_{j=1}^{N} a_{ij} x_j \\ x = \left[\begin{array}{cccc} x_1 & x_2 & \cdots & x_N \end{array}\right]^T \end{array}\right\} \Rightarrow \; \dot{x} = -\mathcal{L} x$$

- This is an autonomous LTI system whose convergence properties depend purely on the spectral properties of the Laplacian.

Graph Laplacians: Useful Properties

- It is orientation independent
- It is symmetric and positive semi-definite
- If the graph is ***connected*** then

$$\text{eig}(\mathcal{L}) = \{\lambda_1, \ldots, \lambda_N\}, \text{ with } 0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N$$

$$\text{eigv}(\mathcal{L}) = \{\nu_1, \ldots, \nu_N\}, \text{ with } null(\mathcal{L}) = \text{span}\{\nu_1\} = \text{span}\{\mathbf{1}\}$$

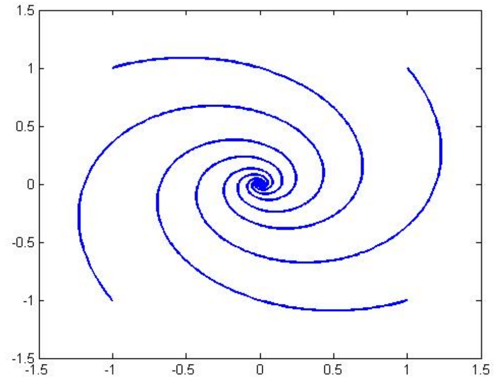
Stability - Basics

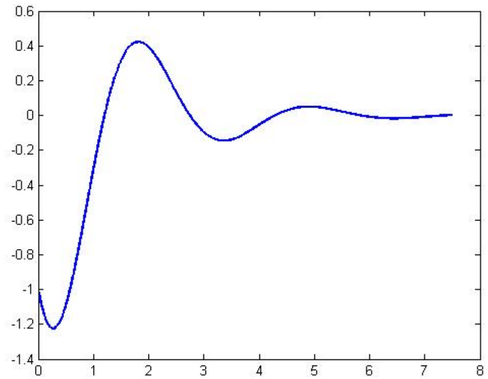
- The stability properties (what happens as time goes to infinity?) of a linear, time-invariant system is completely determined by the eigenvalues of the system matrix

$$\dot{x} = Ax \quad (\dot{x} = -Lx)$$

- Eigenvalues $\lambda(A) = \lambda_1, \ldots, \lambda_n$

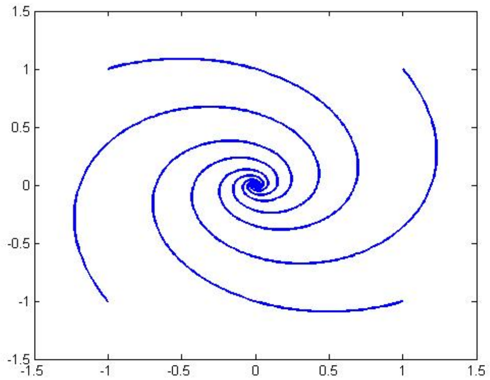
- Asymptotic stability: $\mathrm{Re}(\lambda_i) < 0,\ i = 1, \ldots, n \ \Rightarrow \ \lim_{t \to \infty} x(t) = 0$

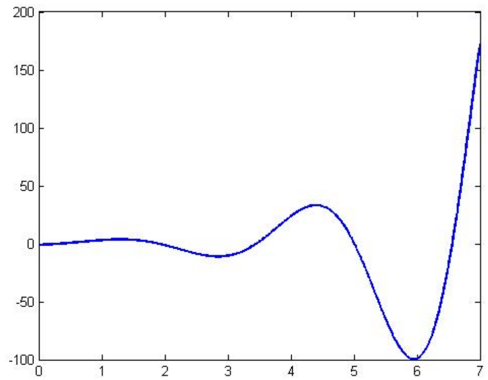




Stability - Basics

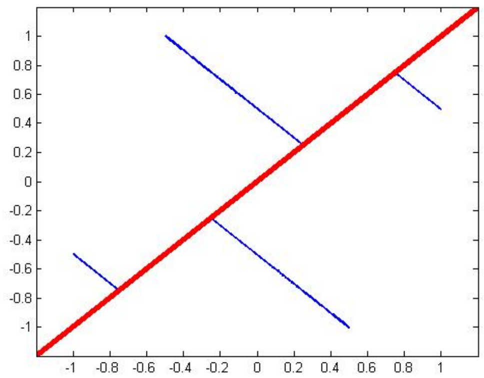
- Unstable: $\exists i \text{ s.t. } \mathrm{Re}(\lambda_i) > 0, \quad \Rightarrow \quad \lim_{t\to\infty} x(t) = \infty$





- Critically stable:

$$0 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n, \quad \Rightarrow$$
$$\lim_{t\to\infty} x(t) \in \ \mathrm{null}(A)$$



This is the case for the consensus equation

Static Consensus

- If the graph is static and connected, under the consensus equation, the states will reach null(L)
- Fact (again):

$$\text{null}(L) = \text{span}\{\mathbf{1}\}, \; x \in \text{null}(L) \;\Leftrightarrow\; x = \begin{bmatrix} \alpha \\ \alpha \\ \vdots \\ \alpha \end{bmatrix}, \; \alpha \in \Re$$

- So all the agents state values will end up at the same value, i.e. the consensus/rendezvous problem is solved!

$$\dot{x}_i = -\sum_{j \in N_i} (x_i - x_j) \;\Rightarrow\; \lim_{t \to \infty} x_i(t) = \frac{1}{n} \sum_{j=1}^{n} x_j(0) = \frac{1}{n} \mathbf{1}^T x(0)$$

Formation Control

- Being able to reach consensus goes beyond solving the rendezvous problem.
- Formation control:

$$\underbrace{x_1, \ldots, x_N}_{\text{agent positions}} \longrightarrow \underbrace{y_1, \ldots, y_N}_{\text{target positions}}$$

- But, formation achieved if the agents are in any translated version of the targets, i.e.,

$$x_i = y_i + \tau, \ \forall i, \ \text{for some } \tau$$

- Enter the consensus equation [5]:

$$e_i = x_i - y_i$$

$$\dot{e}_i = -\sum_{j \in N_i} (e_i - e_j)$$

$$e_i(\infty) = e_j(\infty) = \tau$$

$$\dot{x}_i = \sum_{j \in N_i} [(x_i - x_j) - (y_i - y_j)]$$

$$x_i(\infty) = y_i + \tau, \ \forall i$$

Formation Control

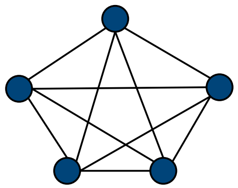
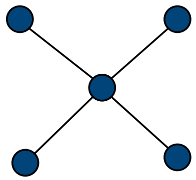
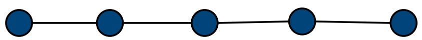
Convergence Rates

- The second smallest eigenvalue of the graph Laplacian is really important!
- Algebraic Connectivity (= 0 if and only if graph is disconnected)
- Fiedler Value (robustness measure)
- ***Convergence Rate:***

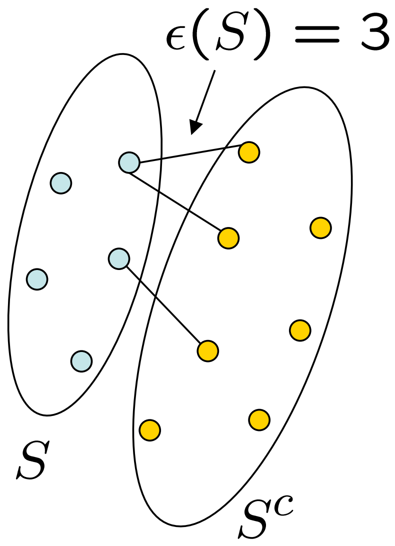
$$\|x(t) - \frac{1}{n}\mathbf{1}^T x(0)\| \leq Ce^{-\lambda_2 t}$$

- **Punch-line:** The more connected the network is, the faster it converges (and the more information needs to be shuffled through the network)

- Complete graph: $\lambda_2 = n$
- Star graph: $\lambda_2 = 1$
- Path graph: $\lambda_2 < 1$

Cheeger's Inequality



$$\phi(S) = \frac{\epsilon(S)}{\min\{|S|, |S^c|\}}$$

(measures how many edges need to be cut to make the two subsets disconnected as compared to the number of nodes that are lost)

isoperimetric number:

$$\phi(G) = \min_S \phi(S)$$

(measures the robustness of the graph)

$$\phi(G) \geq \lambda_2 \geq \frac{\phi(G)^2}{2\Delta(G)}$$

Beyond Static Consensus

- So far, the consensus equation will drive the node states to the same value if the graph is static and connected.
- But, this is clearly not the case in a number of situations:
  - **Edges = communication links**
    - Random failures
    - Dependence on the position (shadowing,…)
    - Interference
    - Bandwidth issues
  - **Edges = sensing**
    - Range-limited sensors
    - Occlusions
    - Weirdly shaped sensing regions

Summary I

- Graphs are natural abstractions (combinatorics instead of geometry)
- Consensus problem (and equation)
- Static Graphs:
    - Undirected: Average consensus iff G is connected
- Need richer network models!