CSCI 8535 Multi Robot Systems

Dr. Ramviyas N. Parasuraman

Assistant Professor, Computer Science

Jan 10, 2019

Talk Outline

- Recap on Course Introduction, Syllabus, etc.
- Announcements
- Graph Theory – fundamentals
- Rendezvous problem

Course Introduction

This is primarily a **research oriented, seminar-style** course covering the topics of control, communication, cooperation, and coordination aspects in multi-robot systems.

It enables students to understand, devise, and solve problems in multi-robot systems and the course will have project-based assignments.

Course Outline

General topics to be covered:

- Multi-robot Rendezvous and Formation Control
- Multi-agent Cooperation and Coordination
- Security and adversarial actions
- Applications of Multi-Robot Systems

Goals of the Course

Graduate-only course.

- Give you a good intuition of ***Multi Robot Systems (MRS)*** modeling and control
  - The essential theoretical tools for MRS
  - How to implement and simulate MRS
  - How to solve real-world multi-robots problems

- You will be able to work on a MRS projects

- After the course, you will:
  - Know the essential theoretical tools for MRS
  - Know how to implement and simulate MRS
  - Know how to solve real-world problems
  - Develop and present a research project
  - Learn something about mobile robots

Requisites of the Course

Requirements:

- (Hard) Programming background and skills (Python or C++)
- (Soft) Working knowledge of simulation tools (Matlab or V-REP or ROS Gazebo, etc.)
- (Hard) Rudimentary mathematical analysis
- (Hard) Linear algebra
- (Soft) Some control theory
- (Soft) Graph theory fundamentals
- (Soft) Probability theory fundamentals

Course Style

- Seminar-style lectures
  - Each student will be assigned a paper to read and present it to the class (as if it’s their own work)
  - Each student will need to critically and constructively review the papers not assigned to them

- Project-based practical assignments and exam

Grading Criteria

In-class participation and Attendance: 10%

Assignments/Paper Reviews: 20%

Paper Presentations: 20%

Mini Project (Midterm): 20% (Project assigned by the Instructor)

Research Project (Final Project): 30% (Project chosen by the student in teams)

Office hours

Office hours of the instructor:

Tuesday and Thursday 2 -3 pm.

Location: Boyd Room 519/520.

If this schedule does not work for you, then send me an email to set up an appointment.

Announcements

First paper assignment for review (by all students):

Presenter: Instructor

Paper: "Consensus Control of Distributed Robots Using Direction of A rrival of Wireless Signals" R. Parasuraman and BC. Min, Intl. Sym. On Distributed Autonomous Robotic Systems (DARS), 2018.

Paper accessible here.
http://cobweb.cs.uga.edu/~ramviyas/pdf/DARS2018_Parasuraman.pdf

Also, uploaded to *eLC* and *Slack.*

Graph Theory, Rendezvous Problem, and Formation Control Problem

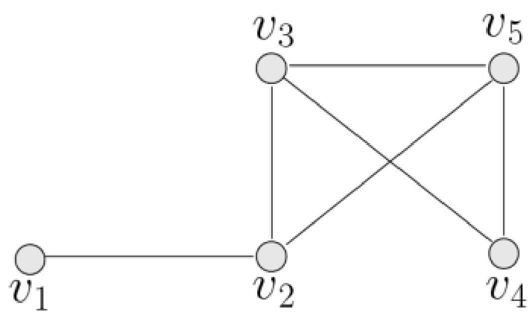
Courtesy of slides from Dr. Andrej Pronobis, U. Washington and KTH Sweden

https://www.pronobis.pro/teaching/mas/

Graph Theory

- Great tool for analyzing networks

Undirected Graphs



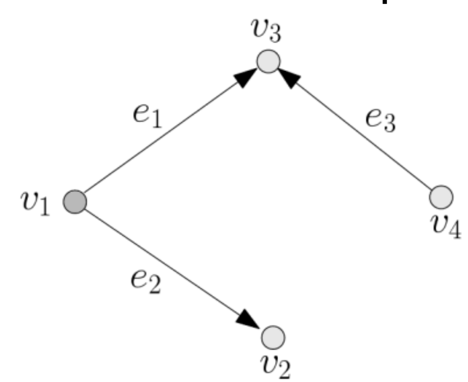
$\mathcal{G} = (V, E)$

Set of vertices

Set of edges

Unordered pair $\{v_i, v_j\} \in E \subseteq [V]^2$

Directed Graphs



$\mathcal{D} = (V, E)$

Ordered pair $(v_i, v_j) \in E$

Tail

Head

- Neighborhood of a vertex $N(i) = \{v_j \in V \mid v_i v_j \in E\}$

Slides courtesy of Dr. Andrej Pronobis

Algebraic Graph Theory

- Adjacency matrix

$$[A(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in E, \\ 0 & \text{otherwise.} \end{cases} \qquad [A(\mathcal{D})]_{ij} = \begin{cases} 1 & \text{if } (v_j, v_i) \in E(\mathcal{D}) \\ 0 & \text{otherwise,} \end{cases}$$

- Degree matrix (undirected graph)

Degree of vertex $d(v_i)$ represents cardinality of neighborhood set $N(i)$

$$\Delta(\mathcal{G}) = \begin{pmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d(v_n) \end{pmatrix}$$

- In-degree matrix (directed graph)

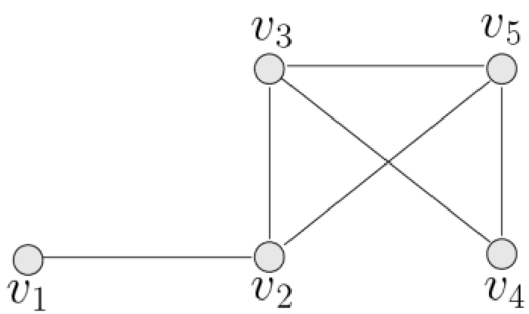
$d(v_i)$ represents the in-degree (counts incoming edges only)

Slides courtesy of Dr. Andrej Pronobis

Graph Laplacian

- For undirected graphs

$$L(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G})$$



$$L(\mathcal{G}) = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{pmatrix}$$

- For directed graphs – In-degree Laplacian

$$L(\mathcal{D}) = \Delta(\mathcal{D}) - A(\mathcal{D})$$

Slides courtesy of Dr. Andrej Pronobis

Properties of Laplacian

- Symmetric and positive semi-definite
- Eigenvalues can be ordered as

$$\lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq \cdots \leq \lambda_n(\mathcal{G})$$

- Smallest eigenvalue is always zero

$$\lambda_1(\mathcal{G}) = 0$$

- Is the graph connected?
  - If for every pair of vertices there is a path
  - IFF $\lambda_2(\mathcal{G}) > 0$
  - As many connected sub-graphs as zero eigenvalues

Slides courtesy of Dr. Andrej Pronobis

Types of Graphs

A graph is said to be connected when there is at least one path (connection/link) from every vertex to every other vertex.

- Path graph or a line graph: each node is connected only once (without a loop)
- Cycle graph: graph structure looks like a cycle (loop)
- Complete graph: all vertices (nodes) are connected to all other vertices
- Tree graph: a connected graph without any cycles

Quick references for Graph Theory

Nice internet resource available in the internet for learning graph theory notations and definitions:

http://www.personal.kent.edu/~rmuhamma/GraphTheory/MyGraphTheory/defEx.htm

Rendezvous Problem – Agreement

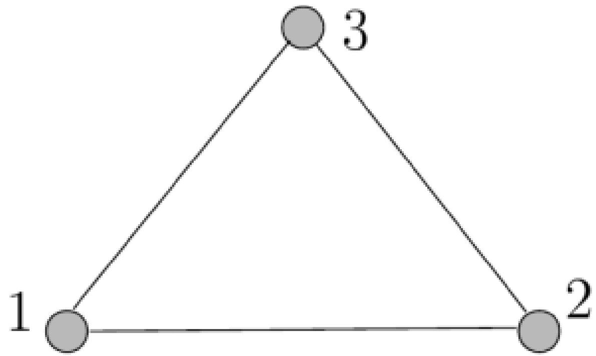
- Agents agree on a value of a parameter
- Definition
  - n dynamic agents
  - Interconnected via relative links
  - Agent's state depends on the sum of its relative states w.r.t. a subset of other agents
- Applications
  - Distributed estimation in sensor networks
  - Flocking/swarming

Slides courtesy of Dr. Andrej Pronobis

Rendezvous Problem

- Rendezvous problem
  - Agent’s state is its location – mobile robot
  - Agents should meet at one point in space
- Example: agreement protocol over a triangle

$$\dot{x}_3 = (x_1 - x_3) + (x_2 - x_3)$$



$$\dot{x}_1 = (x_2 - x_1) + (x_3 - x_1) \qquad \dot{x}_2 = (x_1 - x_2) + (x_3 - x_2)$$

- Links pull robots towards each other

Slides courtesy of Dr. Andrej Pronobis

State-space Representation

- Continuous-time state-space model

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

$$\boxed{\dot{x}(t) = -L(\mathcal{G})\, x(t)}$$

- For directed graphs: use in-degree Laplacian

$$\boxed{\dot{x}(t) = -L(\mathcal{D})x(t)}$$

Slides courtesy of Dr. Andrej Pronobis

State-space Models in Matlab

- Dynamics of a system specified using continuous time-invariant state-space model

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$$

- Create state-space model

```
sys = ss(A,B,C,D)
```

- In our case:

$$\dot{x}(t) = -L(\mathcal{G})\,x(t) \qquad \dot{x}(t) = -L(\mathcal{D})x(t)$$

A

Slides courtesy of Dr. Andrej Pronobis

Simulating

- Simulate
  - Initial condition response

```
[y,t,x] = initial(sys,x0,t)
```

$$\dot{\mathbf{x}}(t) = A\mathbf{x} \qquad \mathbf{y}(t) = C\mathbf{x}$$

  - Response to arbitrary inputs

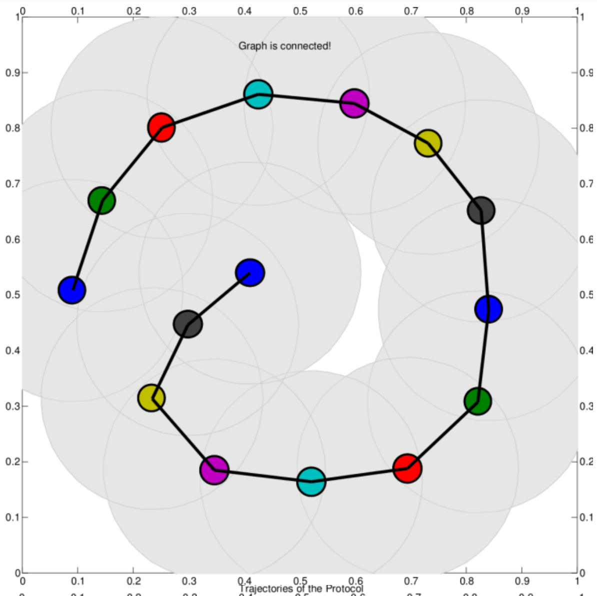
```
[y,t,x] = lsim(sys,u,t,x0)
```

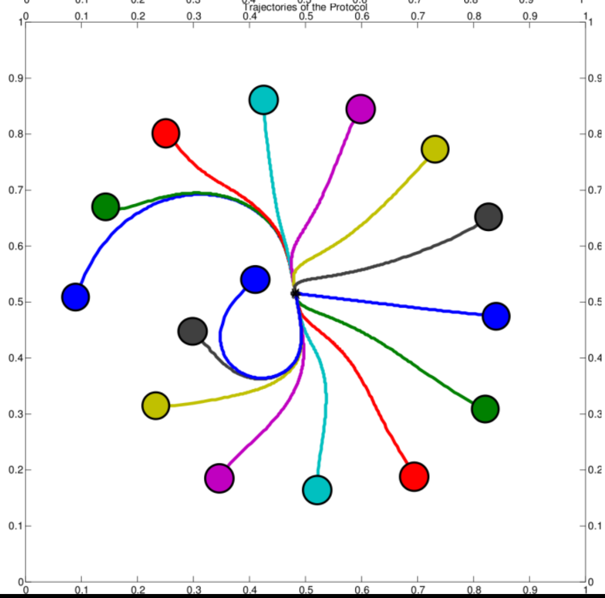
$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \qquad \mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$$

- Basic toolkit for defining and visualizing problems available in course materials

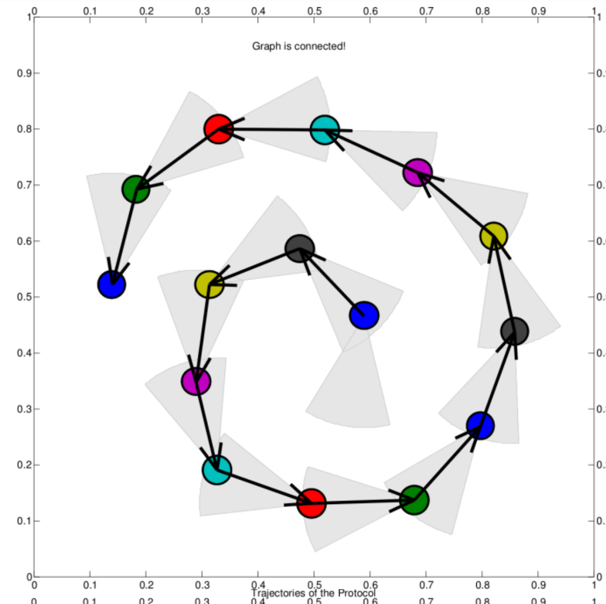
Slides courtesy of Dr. Andrej Pronobis

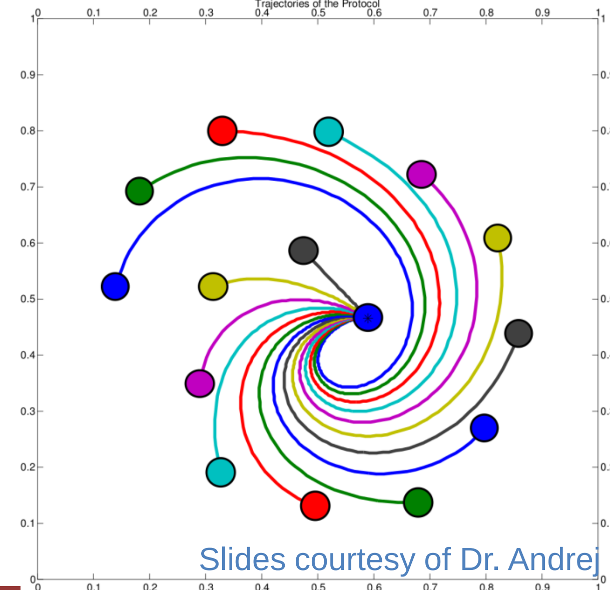
Simulating Trajectories





Run in Matlab





Slides courtesy of Dr. Andrej Pronobis

Formation Control Problem

- Mobile agents move in order to realize a geometrical pattern (formation)
  - Appear often in biological systems (e.g. geese)
- Formations can be specified in several ways
  - **Shape**
    - Specified in terms of points

$$\Xi = \{\xi_1, \ldots, \xi_n\},\ \xi_i \in \mathbf{R}^p,\ i = 1, \ldots, n,$$

    - Translationally invariant

$$x_i = \xi_i + \tau$$

Slides courtesy of Dr. Andrej Pronobis

State-space Representation

- Let's define τ_i as displacement from target

$$\tau_i(t) = x_i(t) - \xi_i,\ i = 1, \ldots, n$$

- Now, apply the agreement protocol to τ_i

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

- Since $\dot{\tau}_i(t) = \dot{x}_i(t)$, $\tau_i(t) - \tau_j(t) = x_i(t) - x_j(t) - (\xi_i - \xi_j)$

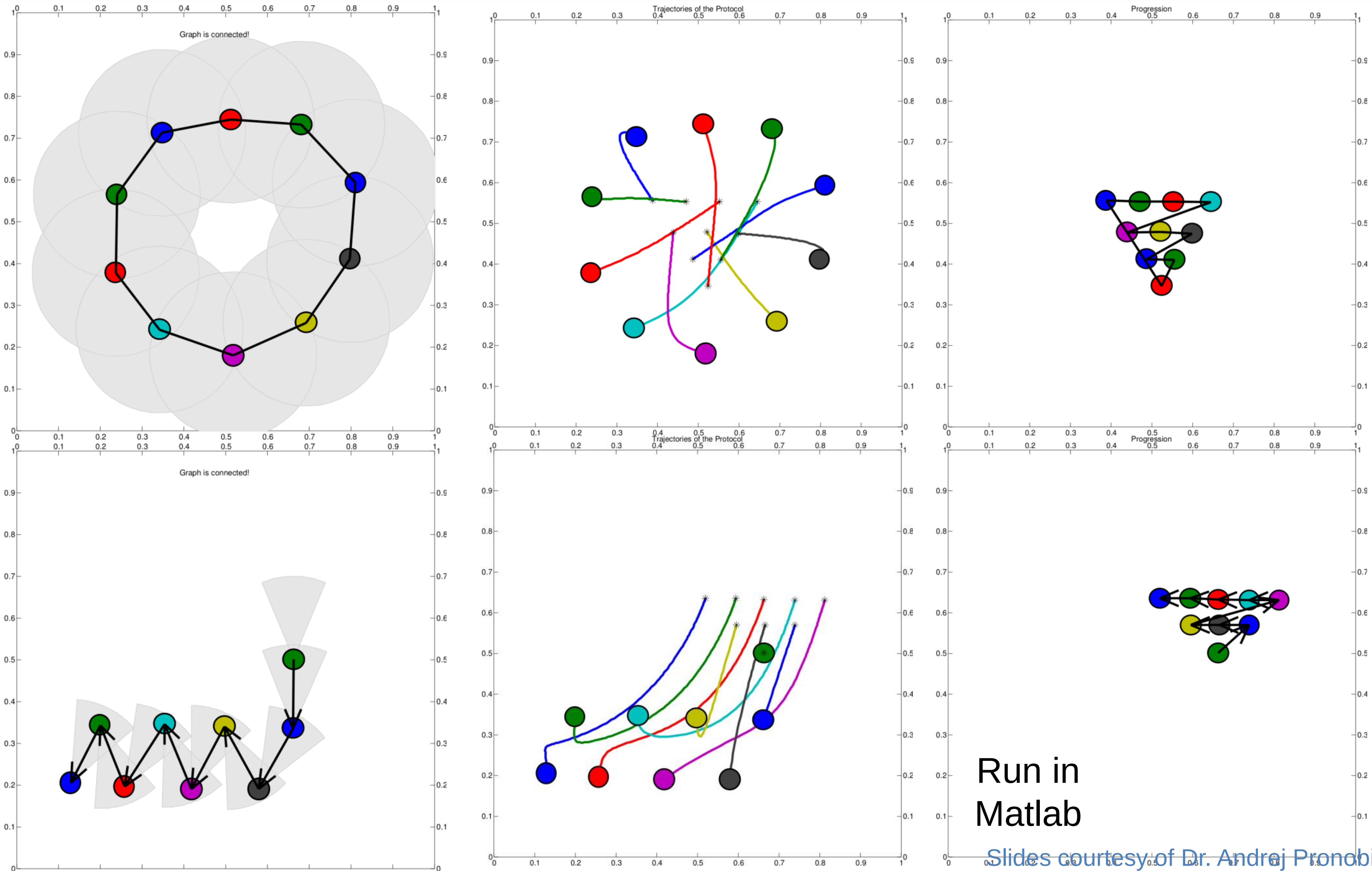
$$\dot{x}_i(t) = - \sum_{j \in N_f(i)} (x_i(t) - x_j(t)) - (\xi_i - \xi_j)$$

$$\dot{x}(t) = -L(\mathcal{G})\, x(t) + L(\mathcal{G})\, \Xi$$

Analogous for directed graphs

Slides courtesy of Dr. Andrej Pronobis

Simulating Trajectories



Run in Matlab

Slides courtesy of Dr. Andrej Pronobis

Summary

- Intuition about what multi-agent and multi-robot systems are and how to solve control problems
  - From theory to simulations
- Multiple applications in robotics
  - When no global maps and central coordination
- What’s next?
  - Dynamic and random networks
  - Switching between formations and control problems
  - Networks as systems (with inputs & outputs)
- Try this at home!

Slides courtesy of Dr. Andrej Pronobis