Autonomous Multi-Agent Systems (Spring 2020)[1]

Class Project

The class project for E6899 carries 60% of the final grade and consists of four components: i) project proposal, ii) midterm progress report, iii) final report, and iv) conference-style presentation. Additionally, you will need to keep a logbook to record your weekly activity. Details of each of these tasks can be found below. The goal of this course is to encourage critical thinking and discussions about open problems in the area of multi-agent systems as well as to find novel areas of application. As such, the project grade will reflect the effort put into the project and the clarity of exposition, as much as progress towards a solution.

Projects should typically be carried out individually, but groups of two are possible with the instructor's permission, and only if there is clear division of labour. In such cases, both students will have to submit individual proposals and progress reports, but the presentation can be created and given jointly (and will be allocated a longer time-slot), as can the final report.

The goal of the class project is to explain, model, and analyze/control a multi-agent system. Projects may be theoretical, computational, or a mixture of both. The subject of the project may be chosen from those we discuss in class, or it could be something completely different. Applying methods we discussed in class to your own research is encouraged. Projects will likely fall into one (or more) of the following categories:

1. **Theoretical:** Developing new, or extending existing theory
2. **Computational:** Developing a software toolbox to implement existing methods
3. **Applied:** Using existing methodologies in new application areas

Students are strongly encouraged to discuss project ideas with the instructor **before** submitting their proposal.

Project structure

The proposal and two reports should all be written using LaTeX. Templates will be available on the course website.

Project proposal

The proposal should be a maximum of two pages in single column format, excluding references, and is due by **February 19th**. The proposal should include:

- Motivation for the choice of project
- Acknowledgement of prior work and relevant papers
- A clear problem statement
  - Be ambitious, but be somewhat realistic; solving the Riemann Hypothesis is probably not going to happen in this class.
- Deliverables

[1] **Instructor:** James Anderson   james.anderson@columbia.edu

  - E.g. a software toolbox for..., an extension of a theorem to a new class of system, simulations that back up your hypothesis, a network model for a new application, etc.

- The approach to solving the problem
  - Decompose the problem into manageable subproblems – define incremental goals.

A well thought-through proposal will make the project much easier and less daunting.

Midterm progress report

The progress report should be a maximum of 3 pages excluding references and is due **March 13**th. Items to include:

- Description of progress towards achieving goals set out in the proposal
  - This could include a reformulation of the problem statement or change of deliverables. State new refined or expanded goals.
- Preliminary results
  - E.g. simulation results or observations, theoretical observations, new conjectures, solution to a restricted version of the problem, etc.
- Statement of any difficulties you have encountered thus far
- Outline of your path to completion
  - List of tasks left to be completed with expected duration of each. Do you anticipate any difficulties, and if so, how do you plan to handle these?

Final report

The final report will be written in the style of a conference paper (8 pages max including references). The deadline for the final report is **May 7**th. Note this is after the presentations; this will allow you to incorporate any feedback. A conference paper is a document that stands on its own, i.e. your classmates should be able to follow it without having read the proposal and mid-term report. A conference paper format will include the following sections:

- Introduction
- Literature review
- Problem statement
- Results
- Numerical results/experiments (if applicable)
- Conclusion and future work

Presentation

Each individual project will be assigned a 20-minute presentation slot; paired projects will have 30 minutes. There will be an additional 5 minutes of questions from the audience. The presentations will take place during the last weeks of class; exact dates will depend on the number of projects. All presentation will take the form of slides, i.e. not chalk talks.

Logbook

To encourage good research practice and to help with project management, each student should maintain a logbook with weekly entries. At a minimum, each week include the following entries:

- **Progress:** list what you have achieved this week
- **Reading:** list the papers/books/forums/etc. you read (if any)
- **Problems:** what in particular has caused you trouble? how can this be fixed?
- **Goals:** list goals for the next week
- **Impediments:** What do you foresee that may prevent you reaching the goals for next week?

Logbooks will be submitted at the same time as final project reports.

Choosing a topic

A good way to start is to take a look at the papers on the reading list and define a project that begins with the results in that paper. Recreating numerical experiments from the papers is a good way to learn the material and is also a valid part of a project. Below is a list of topics and links to researchers working in that area. Take a look at the problems that are being worked on and see what interests you. Note, almost everyone is working in many areas, but they're only listed once here. This is also a tiny snapshot of the people working in this area.

Robotic coordination

- Francesco Bullo: (planning, control, power networks, security)
- Sonia Martinez: (multi-vehicle sensor systems, motion coordination)
- Ali Jadbabaie: (control, networks, learning)
- Jorge Cortes: (distributed control, privacy, power)
- Claire Tomlin: (UAVs, air traffic control, reachability/safety)
- Raffaello D'Andrea:(UAVs, distributed control, planning)
- Vijay Kumar: (UAVs, swarms)
- Aaron Ames: (robotics, control, safety)
- George Pappas: (hybrid systems, UAVs, embedded systems, privacy)

Cascading failures in power systems

- Dan Bienstock
- Gil Zussman
- Alessandro Zocca (cascading failures, EV charging)
- Michael (Misha) Chertkov (control of power networks)

Electric vehicle charging

- Steven Low (EV charging, frequency control)
- Na Li (battery swapping for EV, control of power networks)
- Yorie Nakahira (EV charging algorithms, information theory, neuroscience)

Distributed Control

- James Anderson (SLS distributed control, power networks, power data privacy)
- Nikolai Matni (SLS distributed control, learning for control)
- Anders Rantzer (Distributed control, positive systems, ...)
- Mihailo Jovanovic (Sparsity promoting distributed control, power networks, vehicle platoons)
- Bassam Bamieh (Distributed control, vehicle platoons)
- Sanjay Lall (Distributed control, quadratic invariance, nonlinear)
- Laurent Lessard (Distributed control, quadratic invariance, algorithm design)
- Javad Lavaei (Distributed control, optimal power flow, power networks)

Incorporating information theory

- Takashi Tanaka
- Sanjoy Mitter
- Victoria Kostina

Transportation systems

- Sam Coogan: (transportation, power, safety)
- Ram Rajagopal: (transportation, power)

Social networks

- Florian Dorfler (Opinion dynamics, peer review, power network control)
- Ali Jadbabaie: (spread of misinformation, robotic coordination)
- Asuman Ozdaglar: (networks, game theory, social and economic networks)