

Massachusetts Institute of Technology

16.412/6.834J Cognitive Robotics

PS 4: Final Project Description

Due Dates: Project Proposal - April 11th
Project Presentations – May 9th and 11th.
Final Project Reports – May 11th.

Objectives

The purpose of the project is for you to develop a deep understanding of how to apply a focused set of methods towards the creation of a simple cognitive robot, to be able to communicate these methods to others in a simple and intuitive manner, and to be able to innovate upon these methods, or to lend novel insight into their behavior.

More specifically, you should demonstrate the ability to:

- Clearly state and motivate an interesting, focused innovation to cognitive robotics. An innovation may be an innovative application, a novel algorithmic extension and/or the evaluation of an important analytical question (already exercised in PS2).
- Extract and evaluate the relevant literature using the web and library resources (already exercised in PS3, your advanced lecture).
- Provide a simple, clean tutorial explanation for one or two novel algorithms used in your project, and extract simple pedagogical examples that highlight the key features of the algorithm.
- Implement and demonstrate the application of your algorithm in support of your project goals.
- Evaluate the approach analytically and/or empirically.

Project Grading

- A – represents mastery: the ability to analyze, extend and apply existing methods in a way that is novel and insightful; the ability to explain and motivate in a manner that is particularly intuitive.
- B – represents solid competence: the ability to articulately motivate, explain, implement and evaluate a focused set (i.e., 1 or 2) of intelligent embedded systems methods.

C – represents partial competence of the above.

General

- The results of your project are to be captured in a written document of roughly 10 to 30 pages (depending on team size - ~ 10 pages per member) and are to be presented orally, with slides.
- You may, if negotiated with me in advance, feel free to use your work in this class to extend or complement the work you do within some other project.
- You are welcome to work together in teams, just like in the real world. This will allow you to tackle more ambitious projects. Teams of two typically work best.

Each team may produce a single written reports are to be done, as everyone is to have that experience.

Important Deadlines

The project includes a project proposal (due **Monday, April 11th**), a presentation of results (**May 9th and 11th**) and a final report (due **May 11th**). The deadline for each is given below.

Project Proposal (due April 11th):

The proposal will be up to 6 pages. It will be due in class on **Monday, April 11th**. The purpose is for you to jump start the project before it is too late to get anything done, so take it seriously. Use the project proposal time to do all your background work.

Plan your project in two spirals, each of roughly equal length. In the first spiral you should implement a complete, but simplified version of your system. Stub out most of the difficult components, but make sure that you have place holders for all modules. In addition, make sure that you have exercised all interfaces. The purpose of the first spiral is to place on the table all key issues.

During your second spiral, elaborate the modules to their full capability, in light of the time available. At the end of the first spiral, make sure to carefully assess where you stand, and the difficulty of each module, and then adjust the scope of your project accordingly.

Your proposal should include:

- Cover page
- Introduction, including motivation for the project and assessment of its value to the technical community.
- Concise problem statement outlining of the objectives of the project.
- Statement of previous work and methods that you build on. This is where you do your background work.
- Outline of the proposed technical approach.
- Plans for Spiral 1 and Spiral 2.
- Planning, including division of labor and time schedule.

Also feel free to send me a one page email bouncing your ideas off me. I'll be on travel for a week, but will look at them when I return.

Project Presentations (due May 9th and 11th):

You will give a brief presentation of what you've learned in your project. These presentations will be **May 9th and 11th**. The exact length of the presentation depends on the number of projects, but estimate roughly 8 minutes per person.

Project Reports (due 11th):

See the project objectives at the beginning of this handout for the key elements of your project report.

Your final project report will be due at the end of the day during the final day of classes, **May 11th**. Please submit your final project to the course Secretary, Brian O'Connell. Include a hardcopy of your final project, and a CD or .zip file, containing your project writeup, project presentation, source code for your project, and any explanation required to run your code.

Types of Projects

Projects can be one of the following types:

- **A simulated area exam.** This concept is most familiar to Course 6 students. You read a focused set of papers and do a scholarly critique. This should consist of one to two primary papers, or at most three, that describe a couple, closely related methods. Part of the process is to pick a set of dimensions along which to compare and contrast the papers. To demonstrate competence you should clearly explain the approaches and perform a careful analysis, often including an implementation of one or two of the methods and some empirical evaluation. This is an enhancement of what you did in problem set 2.

To demonstrate mastery you should move beyond comparison, highlighting novel insights about the work and by proposing interesting ways in which the different research approaches can be synthesized or extended. You conclude with a description of the degree to which you are moved, and why.

For example, you could contrast a Bayes net approach to state estimation to estimation using probabilistic hybrid automata; you could compare two recent papers on planning based on propositional encodings, plan graphs and model-checking; or you could contrast

techniques for tracking hidden Markov models both from the POMDP and model-based reasoning literature. When you have a rough idea of what you might want to do, please bounce the idea off me. I can suggest directions to focus, or perhaps recommend particular papers.

- **An extension.** You pick one of the papers that describe an implemented idea for controlling a cognitive robot and you reimplement it, reporting on the surprises that emerge as you understand the work at the level required to write code. You then create and report upon a novel extension to that approach, reporting on the improvements you make. Alternatively you propose a novel integration of two methods for creating a cognitive robot.

For example, you could redo any number of the papers presented in the course: a model-based diagnosis system, a temporal planning and execution capability, a distributed system for solving POMDPs, ...

Remember that implementation has its risks, because everyone seems to overestimate what they can do by the end of the semester. On the other hand, if sufficient detail is given, and you are a good coder, this can be a lot of fun. So assess your programming skills and select accordingly.

- **A novel cognitive robot application.** You pick one or more methods from the papers you've read, perhaps find implementations available over the web, and you design and implement a novel cognitive robot. You demonstrate this capability in simulation, or if REALLY ambitious, on real hardware. You report the innovations of the design and any experimental results and findings.

For example, you might want to prototype a simulated version of a soccer ball size micro-spacecraft that navigates around space station, looking for leaks in its hull. You might focus on the path planning problem, evaluating a variety of the path planning algorithms such as PLR, or you might try different approaches to reactive control, such as Jim Firby's RAPS system.

This project can be risky if you plan on integrating with a real system. This is well beyond the call of duty. Using a simulator is much safer, but still has risks. You can accomplish more by working in teams, and this is encouraged. Be careful not to be exposed to weak-link problems exacerbated by the part-time, conflicting pull nature of student labor.

Alternatively you might propose a novel cognitive robot, but then only implement a selected critical piece to assess feasibility. Caution is advised, along with especially clear and modest goals. On the other hand, if you succeed this can be extremely rewarding. Once you come up with an idea I might be able to suggest places where you can get code or additional papers.

Programming

Your project should demonstrate some level of implementation, either by implementing a cognitive robotics algorithm or by applying that algorithm to the creation of an interesting cognitive robot. The language you use is of your choice, and you may choose to use implementations of existing algorithms that authors may make available on the web. Languages typically used include Java, C++, Lisp, Scheme, C, Prolog and Matlab.

Project Do's and Don'ts

It is easy to select an over ambitious project. Based on past experience, the following are some useful lessons:

- Maintain a tight focus. Go deep on two or even just one method. Beyond two methods it is very easy to produce a project that offers no deep insights. Likewise, in an innovative extension or application, focus on one or at most two innovations.
- Carefully evaluate what is practical from an implementation standpoint, based on your programming skills. Most algorithms take two to four times longer to implement than most might expect.
- Create a descope plan. First brainstorm what you would like to accomplish in your project. Then create three plans:

- An enhanced plan: What you would like to accomplish in your wild dreams.
 - Baseline plan: What you expect to accomplish, given a modicum of conservatism.
 - Minimal plan: A minimum project you need to accomplish to meet your learning goals. Don't be surprised if this is all you accomplish.
- Start early, start early, start early

Suggested Project Topics and Readings

As a result of preparing for your advanced lectures, you should have a deep well of recent readings and methods to draw from. A list of additional [suggested readings for projects](#) is included on the web. This list of readings was recommend by a variety of researchers in autonomous systems, planning and scheduling, propositional reasoning and search, constraint and reactive programming, Bayesian inference, model-based reasoning, hybrid systems and formal verification.