

# Massachusetts Institute of Technology

## 16.412J/6.834J Cognitive Robotics

### Advanced Lecture Proposal

Lawrence Bush, Brian Bairstow and Tony Jimenez  
{ bushl2, bairstow }@mit.edu, tjimenez@alum.mit.edu

March 27, 2005

#### Part A:

Title: Solving POMDPs through Macro Decomposition

Team Member List:

- ☐ Lawrence Bush
- ☐ Brian Bairstow
- ☐ Tony Jimenez

Topics:

- ☐ An introduction to the fundamentals of POMDPs (30 minutes)
- ☐ Demonstration:  
  
Understanding POMDPs through visualization: Visualizations of MDP (as a primer) and POMDP policy generation and execution. (20 Minutes)
- ☐ A review and a paper representing the state of the art in POMDP research and a pedagogical explanation of the respective algorithm. (30 minutes)

#### Part B: Covered Paper

In our lecture, we will be covering the following paper:

Georgios Theodorou and Leslie Pack Kaelbling, "Approximate Planning in POMDPs with Macro-Actions," *Advances in Neural Information Processing Systems 16*, Vancouver, 2004 (NIPS-03).

#### Part C: Abstract

This lecture will cover partially observable Markov decision processes (POMDP), in particular, as they pertain to macro actions. This algorithm is applied to a robot navigation task, which has a temporal component. Integral to this algorithm are belief compression using grid approximation and reinforcement learning with a model (Real Time Dynamic Programming). The specifics of these techniques will be covered.

POMDPs are used to model intelligent agents in an uncertain environment. The agent observes its environment, develops a belief state (a probabilistic state estimate) and chooses an action to maximize the expected future reward. POMDPs are powerful because all environments are uncertain to varying degrees. Consequently, POMDPs more closely model reality.

Unfortunately, creating a policy is currently intractable due to the continuous nature of the belief state space. This has kept POMDPs from being used in real-world applications. One way to ease the computational complexity is to reduce the size of the state space using macro actions. This technique exploits the fact that an agent usually only travels through a small portion of its belief space.

Macro actions are commands like "move down the hallway" or "move to the site" rather than "move forward two inches" or "turn 30 degrees clockwise." Using macro actions effectively turns the state space from a grid to a graph, reducing the number of possible states and thus the number of belief states. Macro actions can be implemented in concert with belief compression and reinforcement learning to create more efficient POMDP implementation.

## Part D: Background Publications

N. Roy, G. Gordon and S. Thrun. "Finding Approximate POMDP solutions Through Belief Compression". *Journal of Artificial Intelligence Research*, 23: 1-40, 2005.  
<http://web.mit.edu/nickroy/www/papers/jair05.pdf>

Supplementing the above paper:

N. Roy, "PhD Thesis: Finding Approximate POMDP Solutions Through Belief Compression," Robotics Institute, Carnegie Mellon University, 2003.  
<http://mapleleaf.csail.mit.edu/~nickroy/thesis/>

Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach (Second Edition)," Prentice Hall, 2002  
<http://aima.cs.berkeley.edu/>

Leslie Pack Kaelbling, Michael L. Littman and Anthony R. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains," *Artificial Intelligence*, Vol. 101, 1998.  
<http://people.csail.mit.edu/people/lpk/papers/aij98-pomdp.pdf>

Hiller and Lieberman, "Introduction to Operations Research (Fourth Edition)," Holden-Day, Inc., 1986  
<http://www.mhhe.com/engcs/industrial/hillier/>

Jaakkola, T., Singh, S., and Jordan, M., "Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems," *Advances In Neural Information Processing Systems*, MIT Press, 1995.  
<http://www.eecs.umich.edu/~baveja/Papers/Nips94b.pdf>

Georgios Theodorou, Kevin Murphy, and Leslie Pack Kaelbling, "Representing hierarchical POMDPs as DBNs for multi-scale robot localization," *International Conference on Robotics and Automation*, 2004.  
<http://people.csail.mit.edu/people/lpk/papers/theochar-icra04.pdf>

## Part E: Division of Labor

Primary Activities:

- ❑ POMDP Explanation (Brian)
- ❑ Paper representing state of the art research (Tony)
- ❑ Demonstration (Larry)

Overarching Activities:

- ❑ Pedagogical Slide Annotations
- ❑ Slide Construction
- ❑ Presentation Flow and Cohesiveness
- ❑ Presentation Effectiveness

## Part F: Demonstration

We will be giving a pedagogical demonstration of POMDPs. This is to take the form of a visualizations of the evolving decision process and final action sequence of a POMDP on an instructive (small) UAV navigation problem. This edification process is intended to instruct the listener while building and executing the plan.

### Demonstration Implementation

In preparation, we investigated using existing software for this demonstration. In particular, we reviewed, compiled and ran the following software implementations:

- [1] Perseus: a set of Matlab functions implementing a randomized point-based approximate value iteration algorithm for Partially Observable Markov Decision Processes (POMDPs).

<http://staff.science.uva.nl/~mtjspaen/pomdp/>

- [2] Pomdp-solve: This code implements the following POMDP solution algorithms:

- Enumeration algorithms
- Sondik's two-pass algorithm
- Cheng's linear support algorithm
- The witness algorithm
- Incremental pruning

<http://www.cs.brown.edu/research/ai/pomdp/code/index.html>

These implementations, however, did not suit our needs because they were either approximate implementations, they were advanced algorithms that did not lend themselves to pedagogical explanation, or the code did not lend itself to extension for visualization purposes.

Instead, we implemented a POMDP implementation ourselves. Our implementation required that we construct a POMDP finite horizon belief state space search followed by policy execution. Our implementation also included two versions, one which ran stand-alone and output its results as text, and one which we equipped with visualization capabilities to demonstrate what the algorithm was doing as it was running. The visualizations involve mapping a simple, generic problem to a

suitable UAV navigation environment. The problem mapping serves to motivate the problem and engage the viewer. The output of the visualization is a movie showing the solution development and policy execution (a UAV navigating to an airport). The movie displays the evolving belief state throughout the process.

We also mapped, solved, executed the policy of and visualized the same problem as an MDP. The basic policy iteration functionality for this implementation was provided by a Markov Decision Process (MDP) Toolbox for MATLAB.

<http://www.cs.ubc.ca/~murphyk/Software/MDP/mdp.html>

The output of this visualization is also a movie showing the solution development and policy execution. In this case, the movie displays the state utilities during the solution process and the UAV position during the policy execution (navigation) process.