16.412
Problem Set 1
Shen Qu
2/21/2005

Part A

Planning with observations/sensor data
Traditional planning assumes that we have complete knowledge of the world in which the plan runs in. However, this assumption is unrealistic for real world scenarios where parts of the environment may be unknown or uncertain. There are several approaches to dealing with planning which involves sensor data that that can only be obtained after the initiation of a plan: reactive planning, contingency planning, etc. But the one I'm more interested in is when a system interleaves planning and execution. This method requires the planner to make as complete a plan as possible prior to execution, and modifies incomplete parts of the plan as information becomes available. There are benefits and issues associated with this approach which will be covered more in section D.

Human Robot interactions
This is a complicated subject that involves various types of reasoning. The first thought that comes to mind when thinking of human robot interaction would be the method of communication between the 2 parties which could involve techniques such as image mapping, sound recognition, body language/identifications of gestures, etc. Another could be the functionality of the robot and how it may be useful to humans. However, a question always on the back of my mind is the optimal yet safe level and method of control that a human should have over a robot with autonomous capabilities. This question originates from the aircraft example of who gets final authority between human pilot and onboard computer system. And I'm not sure how much research was done on this topic within robotics, or whether it's even an issue. But as robots get more advanced with faster and better reasoning algorithms, it is possible to face situations where a robot's analysis of the state of a system is different from that of the human controlling or accompanying it. Then the questions becomes, if a human issues a command that conflicts with the robots internal reasoning system, should the robot be able to override it? If not, and a conflicting command is indeed issued, how would that affect the robot's logic and reasoning, and how should the robot deal with it? Hopefully, I can explore this question a little more through this course and other opportunities.

Learning Algorithms
For me, rather than a topic of interest, machine learning can probably be more accurately described as a topic of fear. Through some basic knowledge, state-of-art algorithms, and recognition of patterns from real world experience, a system can build on its knowledge and effectively become "smarter." Now, combine that with the possibility of a robot with authority or override human command (as described above). This is probably where all sci-fi nightmares began….

Part D

1. “An Analysis of Sensor-based task planning.” By Olawsky, Krebsbach, Gini 1995

This is an older paper (not published within the last 5 years). However, I am presenting it here because it gives a good, comprehensive overview of interleaving planning and execution along with major drawbacks and issues to consider. More recent papers, on the other hand, tend to dive deeply into specific methods and details and fail to address the big picture. (Or it is possible that more recent yet comprehensive papers exist, and I have simply failed to uncover them within the time available to research this topic.)

Traditionally, the planning process is completed before execution begins; but this paper describes the approach of incorporating sensor data collected at execution time into the plan when it is beneficial to do so. In the case where planner lacks sufficient information, it may either assume a default value or defer that portion of the plan to a later time when sensor data can be obtained. The former can increase the uncertainty in the plan; however, it does allow the planner to generate a more complete plan prior to execution. Deferring, on the other hand, can greatly reduce uncertainty but it does present problems of its own: 1) deferring may be prohibitively expensive, 2) early execution of actions without certain required information may interfere with goals not yet considered, and 3) data later collected may still be incorrect or uncertain.

Given these considerations the authors identified three key factors for consideration when selecting a strategy for making such a plan:
1. Goal ordering – what goals the planner should try to satisfy first, not necessarily the order of execution, but the order in which a particular portion of plan is completed. The key strategy here is to sense as much of the information as possible in the early parts of the plan.
2. When to sense – when to switch between planning and execution. There are various methods, but the authors choose to defer goals where further planning requires sensing and only begin execution when plans for all goals are either completed or deferred.
3. What to sense – what to sense and what to default. These decisions are made based on default reliability, sensor reliability, planning costs, execution costs, and the cost of human intervention if applicable.

The authors also provided two metrics for assessing the quality of a plan generated: execution cost, cost of actions during all phases of execution; success rate, percentage of problems where no action needs to be undone due to premature execution of a plan.

Given these factors of consideration and plan quality measurements, authors then proceed to dive into their test world call the Tool Box and to provide detailed cost and success analysis on tests performed in this world. The major flaw of this paper is that the Tool Box world is much more simplified than the real world, and their planner exploit domain depended properties of the Tool Box world not useful for a generic world; there are also various simplifying assumptions made throughout the analysis. Therefore, although the

Tool Box does demonstrate the functionalities of interleaving planning and execution, it does not serve as convincing evidence that this method would be feasible for more general situations.

Finally, the paper concludes on a particularly helpful note by briefly covering other methods of dealing with missing information such as reactive and contingency planning, emphasizing the difference between uncertainty and missing information, and defining open world and close world assumptions used.

2. "Interleaving Execution and Planning via Symbolic Model Checking." Bertoli, Cimatti, Traverso. 2003

This is practically the only recently presented paper I found that considers interleaving execution and planning. Most of the other research in the area seems to focus on dealing with uncertainty, conformant planning, and how to reduce the large, impractical conditional tree plans so that one may hope to generate the entire plan off-line, prior to execution. But although this paper focuses on interleaving planning and execution, the method presented is very different than that described in the previous one.

Like the title suggests this paper presents an interleaving algorithm that uses symbolic model checking to plan over nondeterministic, partially observable domains. In order to achieve this, it uses a state-of-the-art off-line conditional plan generator between executions, and uses observations made during executions to further propagate the plan. An oversimplified version of the algorithm follows:

1. Start with a given belief state
2. Generate a partial conditional plan up to the next observation point. (A conditional plan is a tree structured plan which branches over non-deterministic domains and noisy sensor inputs. Each branch in the tree transverses a series of belief states)
3. The system executes the plan and is guaranteed to follow one of the branches in the partial conditional plan.
4. The system takes in the new current belief state and generates another conditional plan.
5. Algorithm continues until termination.

One key point about this algorithm is that it is guaranteed to terminate because there is a condition in the algorithm which states that every run must transverse a belief state that have not been previously visited. However, the major weakness of this algorithm is that it does not guarantee a plan which reaches the goal state. This is to say, one of the termination points could be a state from where the goal cannot be reached; so the algorithm is guaranteed to terminate at the goal state only within safely explorable domains, where the algorithm cannot get trapped in a state where no strong path to the goal exist.

3. "Complexity of Planning with Partial Observability." Rintanen. 2003

Perhaps the greatest weakness of this paper (for the purpose of this assignment) is it contains nothing directly applicable to the design of a planner or my potential final project for this class. While the first 2 papers discussed above either present an algorithm or a planner capable of performing planning interleaved with execution, this third paper is much more theoretical in nature and focuses on the complexity of the planning problem. However, I feel that the complexity bound is what gives us the bottom-line evaluation metric for the difficulty of a problem. Thus, although not suited for practical use, the material presented in this paper is no less important than the other 2 and thus deserves to be reviewed.

The main finding presented in this paper is, "for non-probabilistic (success probability 1) partially observable planning the plan existence problem is 2-EXP-complete." The rest of the paper is mostly spent on proving this finding as well as providing more direct proves for the EXP-hardness of conditional planning with full observability and the EXPSPACE-hardness of conditional planning without observability. This paper also pointed out that, "plan existence for classical planning (deterministic, one initial state) is [only] PSPACE-complete." At least for me, this really put into scope just how difficult planning for real or realistic scenarios is and the need for better, faster algorithms if we wish to accomplish reasonable planning outside the controlled lab environment.

The paper also leaves open the question of complexity of plans with success probability less than 1 and the difficulty level of optimal planning.

Part E

The somewhat obvious project that derives from parts B and C would be to implement a planner which interleaves planning and execution and uses observations to revise or complete plans. The exact type of planner and algorithm that I would use is still unknown at this point. And depending on its complexity, other features can be added or removed, such as algorithm to identify target (or goals), a simulation (via on screen display) where the plan generated is actually carried out, etc. Sensory data can be taken from a simple piece of hardware linked to the planner or simulated using keyboard inputs or even data files.

The likelihood for me to pursue this project depends largely on what I learn about this type of planning through the course of the semester. I may also be swayed by what interesting projects other people came up with. (I would very much like to do something involving human interaction with robot!) And the direction of my research will also play a role in project selection.