Model-Based Reactive Planning for Autonomous Systems

Seung H. Chung & Brian C. Williams

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

http://www.ai.mit.edu

The Problem: To achieve robustness, autonomous systems, such as deep space explorers, and embedded systems, such as intelligent automobiles, must be able to repair or reconfigure their underlying hardware as failures occur. This requires planning algorithms that are able to reason from engineering models, in order to generate novel command sequences. These planners are similar to STRIPs-style planners, with the added complexity that they must reason through the indirect effects of hardware component interactions. To handle novel situations these planners must operate at reactive time scales, generating novel actions within the sense-act loop. The challenge is that even STRIPS-style planning is known to be NP Hard. Methods for precompiling reactive plans are efficient, such as universal plans. However, universal plans have size quadratic in the number of system states, and are impractical for real-world systems. This research addresses the problem of generating reactive plans that are compact, fast and offer hard, real-time guarantees.

Motivation: Future space exploration will require autonomous systems that are able to act alone, responding to novel situations and failures, sometimes within the fraction of a second. In our past work we demonstrated NASA's first fully autonomous spacecraft called Deep Space 1 mission, which acts by reasoning from basic engineering hardware and operations models. We continue this line of work in the upcoming Air Force T21 multi-spacecraft mission and NASA's Space Technology 7 space probe(ST7).

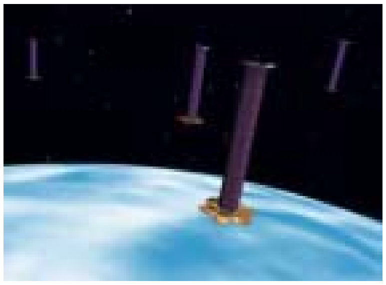

Figure 1: Autonomy for the TechSat 21 multi-spacecraft interferometer

We are creating highly autonomous explorers that use a *model-based executive* to plan actions to achieve a continuous stream of requested goals and to react seamlessly in the event of failure. This implies that the executive cannot operate on predefined execution sequences but must be able to deduce the necessary execution sequence from models of the system being controlled and the intended goals.

Previous Work: This research is based on our earlier development of the Burton model-based reactive planner [5]. Burton generates a set of hierarchical reactive plans that are compact and efficient, but at the cost of completeness. Our current research achieves completeness and generality, by combining Burton's hierarchal reactive plans with recent methods for BDD-based planning [2][3][4], inspired by work on symbolic verification [1].

Approach: The purpose of the Burton, model-based reactive planner is to generate command sequences that lead a physical system from the current state to an intended goal state. These physical systems are represented as a network of interacting components, modeled by a set of concurrent state machines that communicate through shared constraints. As commands are executed, failures may occur, hence at every step the reactive planner must be able to infer the current new state and replan accordingly.

Burton generates control actions in average case constant time. It computes multi-step recovery sequences one step at a time, allowing it to compensate for anomalies in execution at every step. Burton captures five desiderata for reactive planning:

1. It only generates non-destructive / reversible actions.
2. It never proposes actions that lead to dead-end plans.
3. It generates all possible plans for the system being controlled.
4. It ensures progress towards the intended goal.
5. It operates at reactive time scales.

A limitation of Burton is that it can only generate plans for systems in which the input / output relationship between components form an acyclic causal ordering graph. We extend Burton to accommodate cycles in the causal ordering graph. If a causal graph is acyclic, then the planning task can be hierarchically decomposed. However, a cycle in a causal graph represents a set of components that are mutually dependent, and not easily decomposed. These cycles may be eliminated by composing the components into a single super component, but results in an exponential growth in the component's states. To minimize state explosion we use a Binary Decision Diagram is used to encode a reactive plan for the super component. This composition process reduces the physical system to one with an acyclic causal graph, which may then be controlled using our past work on generating hierarchical plans.

Keys to Burton's performance are its offline hierarchical policy construction and compact, BDD-based encoding of these policies. A policy is simply a table that lists the actions and / or subgoals required to transition from the current state to the goal state. Generating a plan that transitions the system to the goal state only requires a simple table look-up of the policies.

Impact: Burton provides one of the capabilities necessary for spacecraft or Earth bound embedded systems to continuously operate autonomously in the event of failure. Such capabilities will improve the robustness of spacecraft systems and enable ambitious mission scenarios that would otherwise be infeasible due to long communication-time delays or uncertainty in the spacecraft's operational environment.

Future Work: Burton currently uses concurrent constraint automata to represent a model of the physical system being controlled. Burton will be extended to accommodate models of systems with complex behaviors through the use of hierarchical concurrent constraint automata.

Research Support: This research is supported by the DARPA Mobies program under contract F33615- 00-C-1702 and by NASA under contract NAG2-1388.

References:

[1] R. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 8(C-35):677–691, 1986.

[2] A. Cimatti, E. Giunchiglia, F. Giunchiglia, and P. Traverso. Planning via model checking: A decision procedure for ar. In *Proceedings of the 4th European Conference on Planning (ECP-97)*, 1997.

[3] A. Cimatti, M. Roveri, and P. Traverso. Automatic obdd-based generation of universal plans in non-deterministic domains. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, pages 875–881, 1998.

[4] R. Jensen and M. Veloso. Obdd-based universal planning: Specifying and solving planning problems for synchronized agents in non-deterministic domains. *Journal of Artificial Intelligence Research,* Accepted 1999.

[5] B. Williams and P. Nayak. A reactive planner for a model-based execution. In *Proceedings 15th International Joint Conference AI (IJCAI'97)*, 1997.