Computational Challenges for Model-based Autonomous Systems

Prof. Brian C. Williams

Massachusetts Institute of Technology

Artificial Intelligence & Space Systems Laboratories
Idea: Support programmers with embedded languages that avoid commonsense mistakes, by reasoning from hardware models.

Reactive Model-based Programming

Polar Lander Leading Diagnosis:

- Legs deployed during descent.
- Noise spike on leg sensors latched by software monitors.
- Laser altimeter registers 50ft.
- Begins polling leg monitors to determine touch down.
- Latched noise spike read as touchdown.
- Engine shutdown at ~50ft.
Model-based Executives automate all reasoning about system interactions.

- Scheduling
- Command confirmation
- Diagnosis
- Commanding
- Configuration
- Repair . . .
Model-based Programs Interact Directly with State

Embedded programs interact with plant sensors and actuators:

- Read sensors
- Set actuators

Programmers must map between states and sensors/actuators.

Model-based programs interact with plant state:

- Read state
- Write state

Model-based executives map automatically between states and sensors/actuators.
Example: The model-based program sets the state to thrusting, and the model-based executive . . . .

Oxidizer tank       Fuel tank

Deduces that thrust is off, and the engine is healthy

Plans actions to open six valves

Deduces that a valve failed - stuck closed

Determines that valves on the backup engine will achieve thrust, and plans needed actions.
Modeling Complex Behaviors through Probabilistic Concurrent Constraint Automata

• Complex, discrete behaviors
  • modeled through concurrency, hierarchy and non-determinism.

• Anomalies and uncertainty
  • modeled by probabilistic transitions

• Physical interactions
  • modeled by discrete and continuous constraints

• Timing
  • modeled by simple temporal networks

inflow = outflow = 0
Model-based Autonomy Architecture

Model-based Program

Control Program
- Executes concurrently
- Preempts
- Asserts and queries states
- Chooses based on reward
- Expresses temporal and resource constraints

System Model

Computational Challenges:
- Propositional Satisfiability
- Optimal CSPs
- Graph-based Planning
- Scheduling

Model-based Executive (Livingstone, Titan, Kirk...)

Control Program Sequencer
Searches for optimal feasible threads of execution
- Plans
- Failures
- Performs lazy scheduling

State estimates
- Mode Estimation
  Tracks likely state trajectories

State goals
- Mode Reconfiguration
  Finds best target
  Plans reactively

Deductive Controller

Plant

Observations

Commands
Future autonomy requires reasoning about hybrid discrete/continuous systems

Detecting subtle failures

Coordinating fleets of agile vehicles

Compute hybrid of:
- HMM belief update
- Kalman filtering

Compute hybrid of:
- temporal constraint problem
- mixed integer linear program

\[ x_p - x_q \geq d - M b_{pq1} \]
\[ x_q - x_p \geq d - M b_{pq2} \]
\[ y_p - y_q \geq d - M b_{pq3} \]
\[ y_q - y_p \geq d - M b_{pq4} \]
\[ \sum_{k=1}^{4} b_{pqk} \leq 3 \]