Temporal Logic Motion Planning for Systems with Complex Dynamics

Amit Bhatia, Matthew R. Maly, Morteza Lahijanian, Erion Plaku, Lydia E. Kavraki, and Moshe Y. Vardi

Department of Computer Science, Rice University
“After inspecting the contaminated areas $C_1$ and $C_2$, visit the decontamination station $D$, and then return to one of the base stations $B_1$ or $B_2$”

“Take measurements from all the stations, but make sure to go to $A$ before going to $B$, and go to $C$ before going to $D$ or $E$, and leave station $F$ at the end”

“Complex” robots + Temporal goals $\Rightarrow$ Increased capabilities
Approach, Environment, Dynamics

- Simple
- Dynamic
- Static
- Complex

System Dynamics

Environment

Approach
Approach, Environment, Dynamics

- Environment
  - Dynamic
  - Static
  - Simple
  - Complex
- Approach
  - Automaton based

GR(1)
Approach, Environment, Dynamics

Environment

Approach

System Dynamics

Environment

dynamic

static

simple

complex

automaton based

GR(1)

this talk

simple

complex
Temporal Logic Planning

- **Automaton based:**
  - Construct an automaton from specification
  - Construct a finite abstraction of the motion of the robot in the environment
  - Plan over product of automaton and abstraction

- **GR(1):**
  - Construct a game structure using input variables of the system and environment
  - Use the definition of $\mu$-calculus over the game structure
  - Find a $\mu$-calculus formula that characterizes the set of winning states of the system
  - Construct winning states from this formula
  - By saving intermediate values in the computation, construct a winning strategy and synthesize an automaton (FDS) that implements the goal
GR(1) Synthesis for Robotics

- Discrete abstraction of environment
- Assume a (bi)simulation relation – provably correct controllers to move between regions

  GR(1) specification

```plaintext
# Define when and how to radio
Do radio if and only if you are sensing person
If you are activating radio or you activated radio then stay there

# Patrol goals
If you are not activating carrying_item and you are not activating radio then visit dining
:.
If you are activating carrying_item and you are not activating radio then visit porch
```

- Using discrete abstraction and GR(1) formula, synthesize a hybrid controller
  - state machine:
    - edges correspond to sensor readings
    - vertices correspond to discrete robot states

“hide and seek” example

[Kress-Gazit et al. TRO 2009, RAM 2011]
GR(1) Synthesis for Robotics

- Discrete abstraction of environment
- Assume a (bi)simulation relation – provably correct controllers to move between regions

GR(1) specification

Using discrete abstraction and GR(1) formula, synthesize a hybrid controller

- state machine:
  - edges correspond to sensor readings
  - vertices correspond to discrete robot states

Reactive Synthesis
Need assumptions on both robot and environment

Very powerful

“hide and seek” example

[Kress-Gazit et al. TRO 2009, RAM 2011]
Synergistic Framework for Motion Planning with LTL

- Allows for complex dynamics – no local steering assumed

- Creates **product automaton** consisting of:
  - geometric abstraction of workspace
  - automaton translated from specification

- Method
  - Use discrete search in the product automaton to produce a **guide**
  - **Turn the guide into a motion plan** using sampling-based motion planning methods

[Plaku et al., TRO 2010; Bhatia et al., ICRA 2010; Bhatia et al., Rob. Auto. Mag. 2011; Bhatial et al., CDC 2011; Maly et. al., HSCC 2013]
Sampling-based Motion Planning

Roadmaps:
PRM [Kavraki, Svestka, Latombe, Overmars ’96]
Obstacle based PRM [Amato, Bayazit, Dale ’98]
Medial Axis PRM [Wilmarth, Amato, Stiller ’98]
Gaussian PRM [Boor, Overmars, van der Stappen ’01]
Bridge Building Planner [Hsu, Jiang, Reif, Sun ’03]
Hierarchical PRM [Collins, Agarwal, Harer ’03]
Improving PRM Roadmaps [Morales, Rodriguez, Amato ’03]
Entropy guided Path-planning [Burns, Brendan, Brock ’04]
RESAMPL [Rodriguez, Thomas, Pearce, Amato ’06]
Probab. foundations of PRM [Hsu, Latombe, Kurniawati ’06]
Adaptive PRM [Kurniawati et al. ’08]
Multi-model planning [Hauser et al. ’10]
Small-tree PRM [Lanteigne et al. ’11]
Rapidly-exploring Random Roadmap [Alterovitz et al. ’11]
and many others

Trees: (continued):
Multiparticle RRT [Zucker et al. ’07]
TC-RRT [Stillman et al. ’07]
RRT-JT [Vande Wege et al ’07]
DSLX [Plaku, Kavraki, Vardi ’08]
KPIECE [Sucan, Kavraki ’08]
RPDST [Tsianos, Kavraki ’08]
BiSpace [Diankov et al. ’08]
GRRT [Chakravorty, Kumar ’09]
IKBiRRT [Berenson et al.’09]
CBiRRT [Berenson et al.’09]
J+RRT [Vahrenkamp ’09]
RG-RRT [Shkolnik et al. ’09]
PCA-RRT [Li, Bekris ‘10]
T-RRT [Jailet et al. ‘10]
SyCLoP [Plaku et al. ‘10]
RRT* [Karaman et al, 10]
RRG [Karaman et al, 10]
PRM* [Karaman et al, 10]
Bi-RRT* [Akgun et al. ’11]
SR-RRT [Lee et al. ‘12]
RRT# [Arslan et al. ‘13]
STRIDE [Gipson et al. ‘13]
and many others

These method provide probabilistic completeness
Motion Planning Problem

Given a dynamical system, the motion planning problem of “Reach-the-Destination” is as follows.

\[ \text{MPP} = (S, S_0, \text{INVALID}, \text{Goal}, U, f) \]

Synergistic Planning

Symbiotic Combination of Discrete Planning and Motion Planning:

- **Rapidly-exploring Random Tree (RRT)** (Kuffner & LaValle, ’99, ’01)
  - “Pull” search tree toward unexplored parts of the state space
  - Use random samples, distance metric, nearest neighbors

- **Expansive-Space Tree (EST)** (Hsu et al., ’97, ’00)
  - “Expand” search tree toward unexplored parts of the state space
  - Use probability distribution, distance metric, nearest neighbors

Synergistic Planner in Action

Sampling-Based Tree Planners

- **Synergistic Planner (ExCAPE)**
- **Computational Speedup**

Synergistic Planner VS Sampling-Based Tree Planners

Planning for Hybrid Systems

- Fits naturally in the synergistic planning framework
- Include modes and discrete transitions in the abstraction.

- Temporal Logic Motion Planning

Syntactically Co-Safe LTL: \[ \phi := \left\{ \pi | \neg \pi \lor \phi \lor \psi \right\} \text{next} \text{until} \text{eventually} \]

DFA can be constructed

Hybrid System Falsification

- Verification of hybrid systems is decidable only in simple cases
- Impractical for complex systems

- Falsification Approach: find a trajectory that violates safety

- Use synergistic motion-planning framework to compute a witness trajectory to an unsafe state

References

- Plaku et al., RSS ’08
- Plaku et al., Formal Methods ’09
- Plaku et al., TACAS ’09
- Plaku et al., TRO ’10
- Bhatia et al., ICRA ’10
- Bhatia et al., Rob. Auto. Mag. ’11
- Bhatia et al., CDC ’11
- Maly et al., HSCC ’13
Next Talk

- Description of the synergistic framework
- Demonstration of some of its capabilities using
  - Hybrid systems
  - Partially unknown environments
Thank you!