cost driven motion planner synthesis for a robotic swarm

Antonio Iannopollo

EXCAPE ANNUAL MEETING
10-11 MARCH 2014
UC BERKELEY
Problem definition

Given

• a set of specifications expressed in Linear Temporal Logic (LTL)
• a set of robots with different characteristics (speed, motion cost)
• a set of tasks, where each task is an ordered sequence of spatial regions to be visited without violating the LTL specs

we want to synthesize a motion planner that is able to satisfy specifications deciding how to allocate tasks to different robots according to a given cost function.
Solution Strategy

The presented solution is based on the **LTLMoP toolkit**

**Assumptions:**
- Each robot must be able to **fully satisfy specifications**
  - *E.g. specifications cannot require two regions to be visited at the same time*
- Robots **do not interfere** each other
  - *E.g. a robot will not block another one while executing a task*
- Once scheduled, a task remains active **until it is completed**
  (according to safety specifications)
Solution Strategy – Tasks & Robots

Each task is characterized by
• An ordered set of regions to be visited
• A formula (on environment variables) that will “activate” the task
• A priority index
A task is considered completed when all regions have been visited

Each robot is characterized by
• Speed
• Motion cost
Solution Strategy - Synthesis

Using
- task and robot information
- environment assumptions and additional (safety) robot specifications
- LTLMoP input (regions and simulation configuration)

two types of controller are synthesized:
- A controller for each robot
- (part of) a centralized scheduler

Communication between robots and scheduler is realized through additional variables

During simulation, for each task being activated, the scheduler will activate the robot that is not busy and that minimize the function

\[
\text{task\_priority} \cdot \text{moving\_cost} + (1 - \text{task\_priority})) \cdot \text{est\_moving\_time}
\]
Solution Strategy

Configuration

- Task:
  - priority
  - regions
  - activation vars

- Robot:
  - moving cost
  - speed
  - id

- General specs:
  - Environment assumptions and system specs

Scheduler specs

Robot complete specs:
- include task specs

Scheduler specs

Environment

Robot

Config file

Synthesis

LTLMoP input

Region file

Simulation configuration

cost function: \([p*c + (1 - p)]*t\)

p = task priority

\(c = \text{moving cost}\)

\(t = \text{estimate moving time}\)
Experimental Results

2 tasks:
  task1: r1 and then r2
  task2: r3 and then r4

v2

task1 has low priority (minimum cost)
task2 has high priority (minimum time)

v1

2 robots:
  Faster but with higher cost
  Slower but cheaper
Conclusion

- **Hierarchical** approach to synthesis from LTL, with introduction of decisions based on a cost function
- Allow to **parallelize** a task list execution on swarm of robots
- Possible improvements are related both to practical and theoretical aspects, such as the implementation of queues of tasks for a single robot, a more formal definition of problem and solution, and a stronger integration with principles of platform-based design
Questions?