Implan: Scalable Incremental Motion Planning for Multi-Robot Systems

Indranil Saha
UC Berkeley and UPenn

Joint work with
Rattanachai Ramaithitima (UPenn), Vijay Kumar (UPenn), George Pappas (UPenn) and Sanjit Seshia (UC Berkeley)

Annual Meeting 2015
Motion Planning Problem

An input problem instance $\mathcal{P} = \langle R, I, PRIM, Workspace, \xi \rangle$

- R - The set of robots
- I - Initial state of the group of robots
- $PRIM = [PRIM_1, PRIM_2, \ldots, PRIM_{|R|}]$
- Workspace - Workspace dimension, position of obstacles
- $\xi$ - Specification given in Linear Temporal Logic

Definition (Motion Planning Problem)

Given an input problem $\mathcal{P}$ with specification $\xi$, synthesize a trajectory that satisfies $\xi$
Complan

(COMpositional Motion PLANner)

http://www.seas.upenn.edu/~isaha/complan.shtml

\[ \Phi(0) \xrightarrow{Prim_1} \Phi(1) \xrightarrow{Prim_2} \Phi(2) \ldots \Phi(L-1) \xrightarrow{Prim_L} \Phi(L) \]

Constraints: \((\Phi(0) \in I) \land \|Transition\| \land \|Specification\|\)

Boolean combination of constraints from Linear Arithmetic and Equality with Uninterpreted Functions theories

Complan solves for the \(L\) motion primitives using an SMT solver
The motion planning problem is reduced to a monolithic SMT solving problem.

The number of constraints for collision avoidance increases quadratically with the number of robots.

Complan has limited scalability.
Implan
(Incremental Motion PLANner)

Specification: $\neg collision \cup reach$
Outline of the Algorithm

- Synthesize optimal trajectory for each robot independently
- Assign Priority to find a feasible ordering for the robots
- Synthesize final trajectories according to the assigned priorities
  - Treat the robots with higher priorities as dynamic obstacles
  - Introduce minimum delay to execute the optional trajectory to avoid collision
Priority Assignment Algorithm

**InitObs**(j) : The set of robots whose initial locations obstruct the optimal trajectory of Robot \( R_j \)

**FinalObs**(j) : The set of robots whose final locations obstruct the optimal trajectory of Robot \( R_j \)

**Algorithm:**

1. Generate the following constraints:

\[
\forall R_j \in R, \forall R_k \in R \setminus \{R_j\} : \\
R_k \in \text{InitObs}(j) \rightarrow \text{prio}(k) > \text{prio}(j)
\]

\[
\forall R_j \in R, \forall R_k \in R \setminus \{R_j\} : \\
R_k \in \text{FinalObs}(j) \rightarrow \text{prio}(j) > \text{prio}(k)
\]

2. Solve the constraints using an SMT solver
Example 1: Motion Plan Synthesis for 6 Robots

Ordering:
1: $R_3$  2: $R_6$  3: $R_2$
4: $R_4$  5: $R_1$  6: $R_5$
Example 1: Motion Plan Synthesis for 6 Robots

Ordering:
1: $R_3$  2: $R_6$  3: $R_2$
4: $R_4$  5: $R_1$  6: $R_5$

Computation Time:
- Complan: 27m40s
- Implan: 7s
What if a priority assignment does not exist?
What if a priority assignment does not exist?

\[ \text{prio}(1) < \text{prio}(2) \land \text{prio}(1) > \text{prio}(2) \]
What if a priority assignment does not exist?

\[ \text{prio}(1) < \text{prio}(2) \land \text{prio}(1) > \text{prio}(2) \]

Dependent robots are placed into a group

Groups are found by \textit{unsatisfiable cores} generated by the SMT solver
Example 2: Motion Plan Synthesis for 6 Robots

Ordering:
1: \( R_2 \)
2: \( R_4, R_5, R_6 \)
3. \( R_1, R_3 \)
Example 2: Motion Plan Synthesis for 6 Robots

Ordering:
1: $R_2$
2: $R_4, R_5, R_6$
3: $R_1, R_3$

Computation Time:
Complan: 3hr13m
Implan: 27m
Generated trajectories are cost optimal, but may not be time optimal.

The worst case computational complexity of Implan is the same as Complan.
Example 3: Motion Planning in a Compact Workspace

25 quadrotors moving in a closed place

**Specification:**
\[ \neg \text{collision} \cup \text{reach} \]

**Workspace Size:**
15.2m × 16.8m

**Computation Time:**
4hr12m
Example 4: Incremental Motion Planning in a Free Workspace

50 quadrotors moving in a free place

Specification:
\[ \neg \text{collision} \cup \text{reach} \]

Workspace Size:
15.2m × 16.8m

Computation Time:
30m19s
To Extend **Implan** to arbitrary LTL/MTL properties
Thank You!!

http://www.seas.upenn.edu/~isaha