Compositional Synthesis of Multi-Robot Motion Plans via SMT Solving

Indranil Saha
UC Berkeley and UPenn

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

March 10, 2014
Goal: I1 → F1, I2 → F2, I3 → F3, I4 → F4

Invariants:
- Maintain a rectangular formation
- Maintain a precedence relationship
  - The X co-ordinate of the quadrotors at I1 and I2 will be always less than the X coordinate of the quadrotors at I3 and I4
- Maintain a minimum distance
  - The distance between two quadrotors is always greater than one unit
To synthesize motion plans automatically for
- a group of robots
- complex dynamics
- complex specification

Specification is given in Linear Temporal Logic (LTL)
Existing Solutions for LTL Motion Planning

- Generate a finite abstraction for the robot dynamics
- Generate a finite model for the property
- Apply a game theoretic algorithm to generate a high level plan
- Generate low level control signals that satisfy the bisimulation property

Computationally expensive.. Not suitable for multi-robot systems
Our Approach

- We assume availability of a set of precomputed control laws for each robot
  - motion primitives

- We use an off-the-shelf SMT solver to generate motion plans composing these motion primitives
 Motion Primitive

A motion primitive is formally defined as a 7-tuple: \( \langle u, \tau, q_i, q_f, X_{rf}, W, cost \rangle \).

- \( u \) - a precomputed control input
- \( \tau \) - the duration for which the control signal is applied
- \( q_i \) - initial velocity configuration
- \( q_f \) - final velocity configuration
- \( X_{rf} \) - relative final position
- \( W \) - the set of relative blocks through which the robot may pass
- \( cost \) - an estimated energy consumption for executing the control law

\( PRIM_i \) - the set of all primitives for robot \( i \)
An input problem instance $\mathcal{P} = \langle N, I, F, PRIM, OBS, \xi \rangle$

- $N$ - Number of robots
- $I$ - Initial state of the group of robots
- $F$ - Final state of the group of robots
- $PRIM = [PRIM_1, PRIM_2, \ldots, PRIM_N]$
- $OBS$ - the set of obstacles
- $\xi$ - $\square \psi$, conjunction of a set of invariant properties
A motion plan of a multi-robot system for an input problem instance $\mathcal{P} = \langle N, I, F, PRIM, OBS, \square \psi \rangle$ is defined as a sequence of states $\Phi = (\Phi(0), \Phi(1), \ldots, \Phi(L))$ such that

- $\Phi(0) \in I$
- $\Phi(L) \in F$
- $\Phi(0) \models \psi$

and the states are related by the transitions in the following way:

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

Definition (Motion Planning Problem)

Given an input problem $\mathcal{P}$ and a positive integer $L$, synthesize a motion plan of length $L + 1$. 

ExCAPE Annual Meeting 2014 Indranil Saha
Transition Constraints

State of a robot: $\langle q, X \rangle$

- $q$ - Velocity configuration
- $X$ - Position

$\Phi_1 = [\phi_{11}, \ldots, \phi_{1N}]$, $\Phi_2 = [\phi_{21}, \ldots, \phi_{2N}]$

$Prim = [\text{prim}_1, \ldots, \text{prim}_N]$, where $\text{prim}_i \in PRIM_i$.

A transition

$\Phi_1 \xrightarrow{Prim} \Phi_2$

is associated with the following constraints:

- $\forall i \in \{1, \ldots, N\} : \phi_{1i}.q = \text{prim}_i.q_i$
- $\forall i \in \{1, \ldots, N\} : \phi_{2i}.q = \text{prim}_i.q_f$
- $\forall i \in \{1, \ldots, N\} : \phi_{2i}.X = \phi_{1i}.X + \text{prim}_i.pos_f$
- $\text{obstacle}_\text{avoidance}(\Phi_1, \Phi_2, Prim, OBS)$
- $\text{collision}_\text{avoidance}(\Phi_1, \Phi_2, Prim)$
- $(\Phi_1 \models \psi) \rightarrow (\Phi_2 \models \psi)$
Goal: \((I_1 \text{ and } I_2) \rightarrow B \quad (I_3 \text{ and } I_4) \rightarrow A\)

Invariants:

- Maintain a rectangular or linear formation
- Maintain a minimum distance
- The distance between two quadrotors is always greater than one unit

No motion plan that satisfies the formation constraint exists
Goal: \((I1 \text{ and } I2) \rightarrow B\) 
\((I3 \text{ and } I4) \rightarrow A\)

Invariants:
- Maintain a minimum distance
- The distance between two quadrotors is always greater than one unit
Finding Optimal Trajectory

- Find the least number of motion primitives that can generate a valid trajectory.

- Among all trajectories that use the least number of motion primitives, find the one that incurs the least cost.
Goal: (I1 and I2) → B  
(I3 and I4) → A

Invariants:
- Maintain a minimum distance
- The distance between two quadrotors is always greater than one unit
Thank You!!