Compositional Synthesis of Multi-Robot Motion Plans via SMT Solving

Indranil Saha

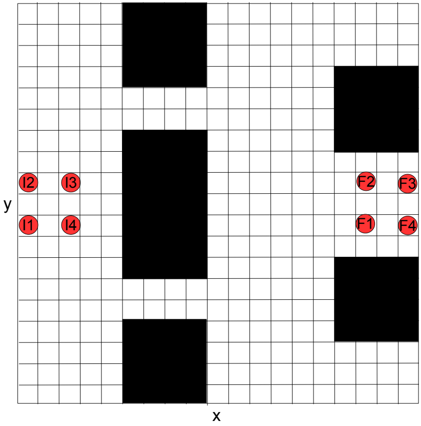
UC Berkeley and UPenn

Joint work with

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

Dagstuhl Seminar on Verification of Cyber-Physical Systems

March 17-21, 2014



Start movie

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 I1and 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

Goal

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

Work by Kress-Gazit, Fainekos, Pappas, Karaman, Frazzoli, Kavraki, Verdi, Topcu, Murray, Belta, Rus and others..

Computationally expensive.. Not suitable for multi-robot systems

Our Approach

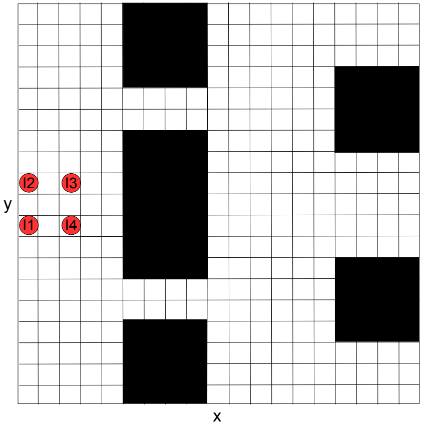
- We assume availability of a set of precomputed control laws for each robot
  - motion primitives
- We use an off-the-shelf SMT solver to generate motion plans composing these motion primitives

State of a Multi-Robot System

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

- q - Velocity configuration
- X - Position

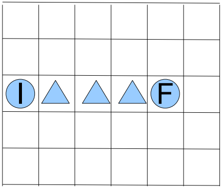
State of the multi-robot system:
$\Phi = [\phi_1, \ldots, \phi_N]$

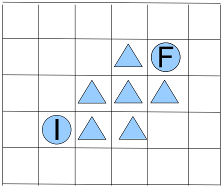


Motion Primitive

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
- τ - 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





Note: Motion Primitives are position oblivious

Motion Planning Problem

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
- ξ - $\Box\Psi$, conjunction of a set of invariant properties

Definition (Motion Planning Problem)

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

Motion Plan

A *motion plan* of a multi-robot system for an input problem instance $\mathcal{P} = \langle N, I, F, PRIM, OBS, \Box\Psi \rangle$ 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:

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

Transition Constraints

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

$Prim = [prim_1, \ldots, prim_N]$, where $prim_i \in PRIM_i$.

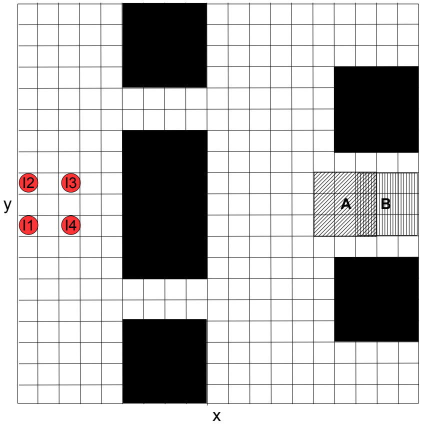
A transition

$$\Phi_1 \xrightarrow{Prim} \Phi_2$$

is associated with the following constraints:

- $\forall i \in \{1, \ldots, N\} : \phi_{1i}.q = prim_i.q_i$
- $\forall i \in \{1, \ldots, N\} : \phi_{2i}.q = prim_i.q_f$
- $\forall i \in \{1, \ldots, N\} : \phi_{2i}.X = \phi_{1i}.X + prim_i.X_{rf}$
- $obstacle\_avoidance(\Phi_1, \Phi_2, Prim, OBS)$
- $collision\_avoidance(\Phi_1, \Phi_2, Prim)$
- $(\Phi_1 \models \Psi) \rightarrow (\Phi_2 \models \Psi)$

Specification 2

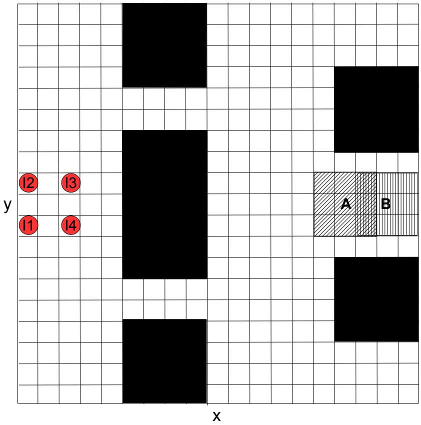


Goal: (I1 and I2) → B
(I3 and I4) → A

Invariants:

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

Specification 2



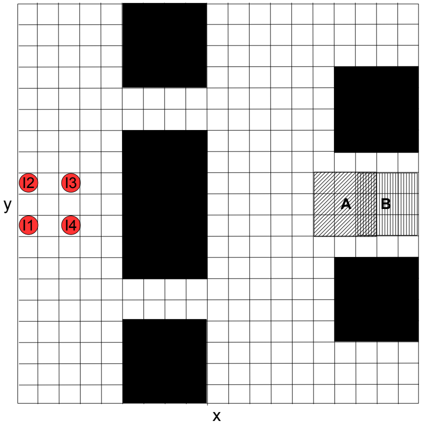
Goal: (I1 and I2) $\rightarrow$ B
(I3 and I4) $\rightarrow$ A

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

Specification 2



Goal: (I1 and I2) $\rightarrow$ B
(I3 and I4) $\rightarrow$ A

Invariants:

- Maintain a minimum distance
  - The distance between two quadrotors is always greater than one unit

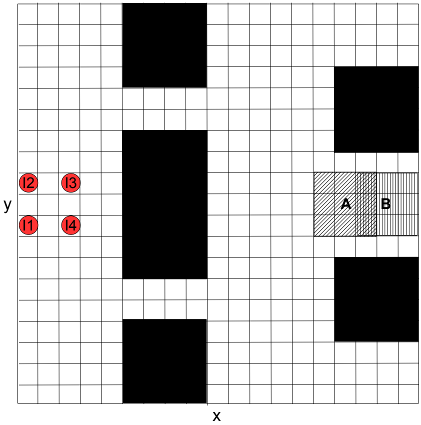
Start movie

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

Specification 2



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

Start movie

Completeness

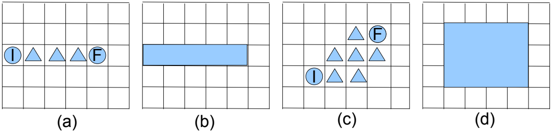
The completeness is with respect to the given set of motion primitives.

Given a positive integer L, If there exists a trajectory of length L using the given set of motion primitives, our technique is able to generate that trajectory.

Runtime

Destination Specification	Without rectangular abstraction	With rectangular abstraction
Spec 1	4m51s	2m06s
Spec 2	5m59s	3m25s

Table: Experimental results on two case studies.



Current and Future Work

- How to handle arbitrary LTL specification in our framework?

  - Persistent Surveillance
  [Belta and others, ICRA 2012, CDC 2012, ...]

- How to deal with change in environment?

  - Patching task level robot controllers
  [LivingstonPrabhakarJoseMurray, ICRA 2013]

- How to scale our framework for a large number of robots?

  - Distributed motion planning
  [TurpinMichaelKumar, ICRA 2012, JRR 2014]

Thank You!!