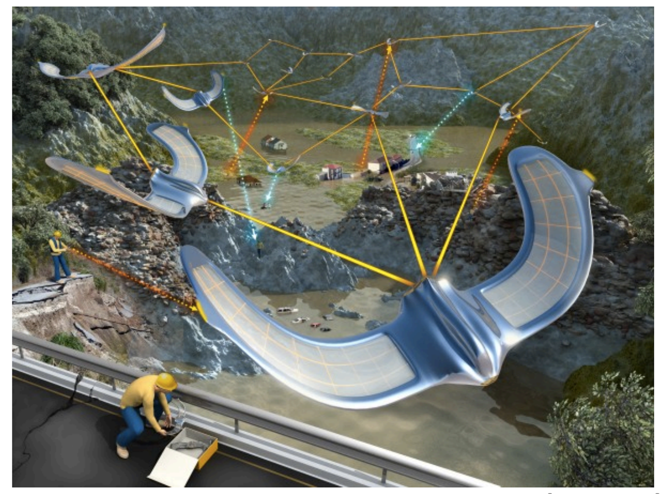
Reactive Switching Protocols for Multi-Robot High-Level Tasks

Vasumathi Raman
California Institute of Technology (vasu@caltech.edu)

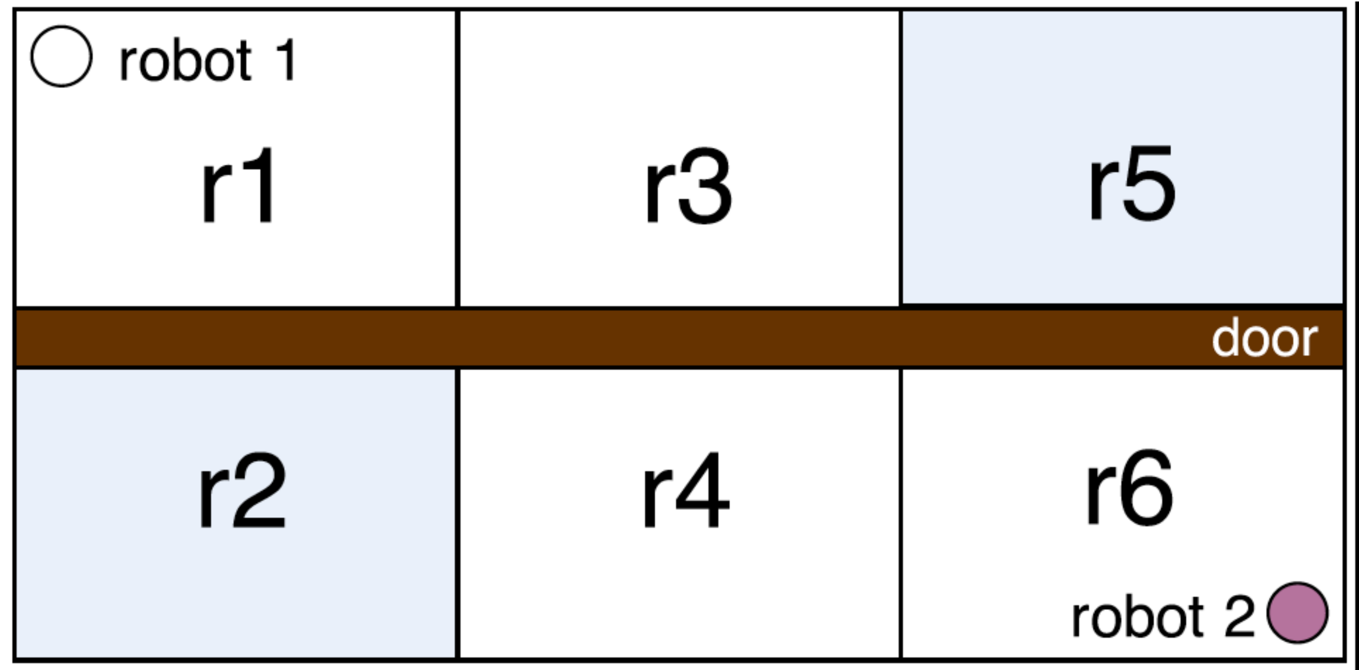
Motivation

Automatically generate provably correct control from high-level specifications for teams of interchangeable robots


Kiva Systems


Disaster-Response UAVs (EPFL)

Example:



Robot 1 starts in r1, robot 2 starts r6. If the door is closed, the robots cannot move through it.
Regions r2 and r5 are ``intrusion-sensitive'' -- when an intruder is detected in them, one of the robots must go to that region to investigate.

Overview

- model robot team as a switched system

$$\dot{x}(t) = f_{\sigma(t)}(x(t)),$$

mode = task assignment

- construct motion controllers for each mode
- synthesize switching protocol to realize φ

Contributions

Novelty: concurrent task reassignment and planning via reactive synthesis

Computation: switched system representation yields exponential improvement during synthesis

Virtualization: explicit separation between motion controllers and robots allows solution of otherwise infeasible tasks

Reactive Switching Protocols for Multi-Robot High-Level Tasks

Vasumathi Raman

Automatically generating provably correct control from high-level specifications for teams of interchangeable robots

Reactive Synthesis

Specification
$\varphi = \varphi_e \Rightarrow \varphi_s$

↓

Synthesis Engine [1]

↓

System $\mathcal{M}$ such that $\mathcal{M} \models \varphi$

- Formulas in Linear Temporal Logic (LTL)*
  - » Propositional logic +

    $\mathcal{U}$ (until) $\bigcirc$ (next) $\square$ (always) $\diamond$ (eventually)
- Propositions partitioned into controlled ($\mathcal{Y}$) and uncontrolled ($\mathcal{X}$) sets
- If the operating environment obeys φ_e, the system satisfies φ_s.
- Generalized Reactivity (1) (GR(1))

$$\varphi_e^i \wedge \varphi_e^t \wedge \varphi_e^g \Rightarrow \varphi_s^i \wedge \varphi_s^t \wedge \varphi_s^g$$

Related Work

Temporal Logic Synthesis for Multi-Robot Systems

Kloetzer and Belta, T-Ro 2007, 2010	not reactive
Chen & Belta, T-Ro 2010	smaller class of specifications
Loizou and Kyriakopoulo, CDC 2004	restrict non-motion actions to be continuous
Raman and Kress-Gazit, ICRA 2014	expensive encoding of individual robot motion in LTL

Concurrent task assignment and planning

Turpin and Kumar, ICRA 2013	does not consider non-determinism in the environment
Ayanian, Rus and Kumar, NecSys 2013	task is just a goal configuration for the team

Synthesis of switching protocols

Liu, Ozay, Topcu and Murray, TAC 2013	for general switched systems

Motion Control

- Need a controller for driving the team of robots from any current configuration to the goal configuration for each permutation of the goals
- Can use approach in [2] -- decompose the configuration space into obstacle-free polytopes, generate local smooth feedback laws that drive the team of robots from one cell to an adjoining one, sequence these local controllers using A* or incremental D* to reach the goal.

[1] Roderick Bloem, Barbara Jobstmann, Nir Piterman, Amir Pnueli, Yaniv Sa'ar. *Synthesis of Reactive(1) designs*. J. Comput. Syst. Sci. 78(3): 911-938 (2012)

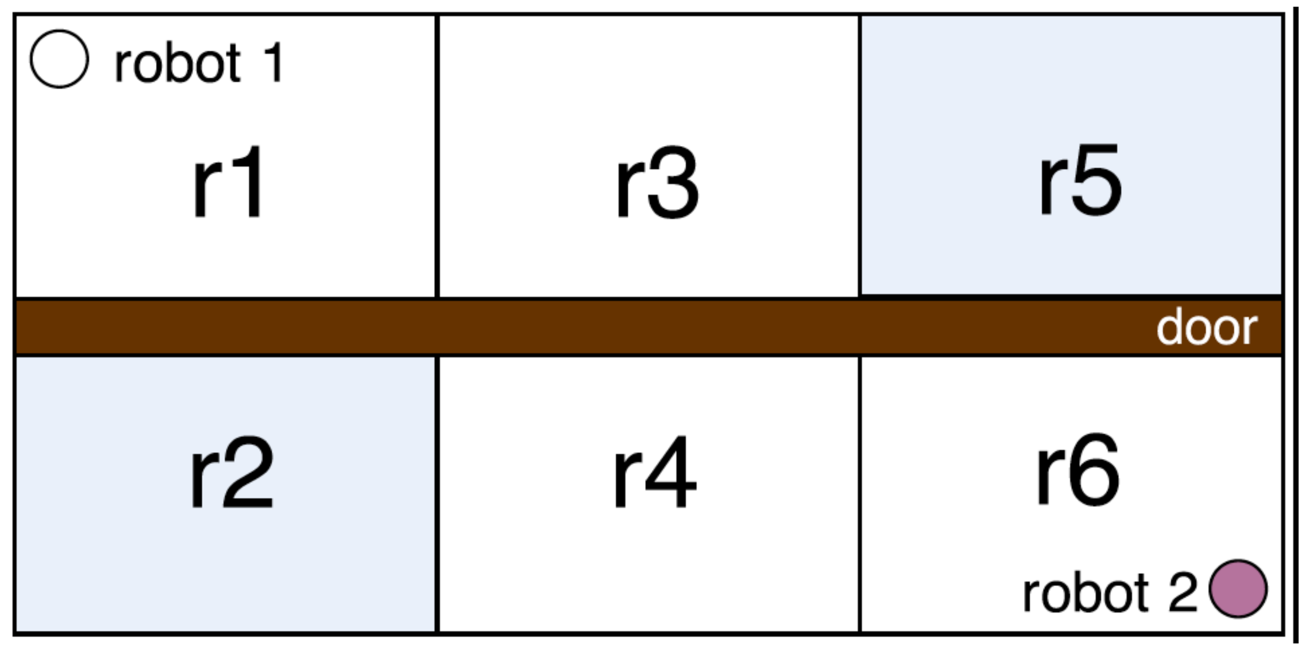
[2] Nora Ayanian, Daniela Rus, and Vijay Kumar. *Decentralized multirobot control in partially known environments with dynamic task reassignment*. In NecSys, pages 311–316, Santa Barbara, CA, 2012.

Reactive Switching Protocols for Multi-Robot High-Level Tasks

Vasumathi Raman

Automatically generating provably correct control from high-level specifications for teams of interchangeable robots

Example:



Robot 1 starts in r1, robot 2 starts r6.
If the door is closed, the robots cannot move through it.
Regions r2 and r5 are ``intrusion-sensitive'' -- when an intruder is detected in them, one of the robots must go to that region to investigate.

Discrete Abstraction

$\mathcal{G} = \{r_2, r_5\}$ goals

$\mathcal{P}_N = \{12, 21\}$ goal permutations

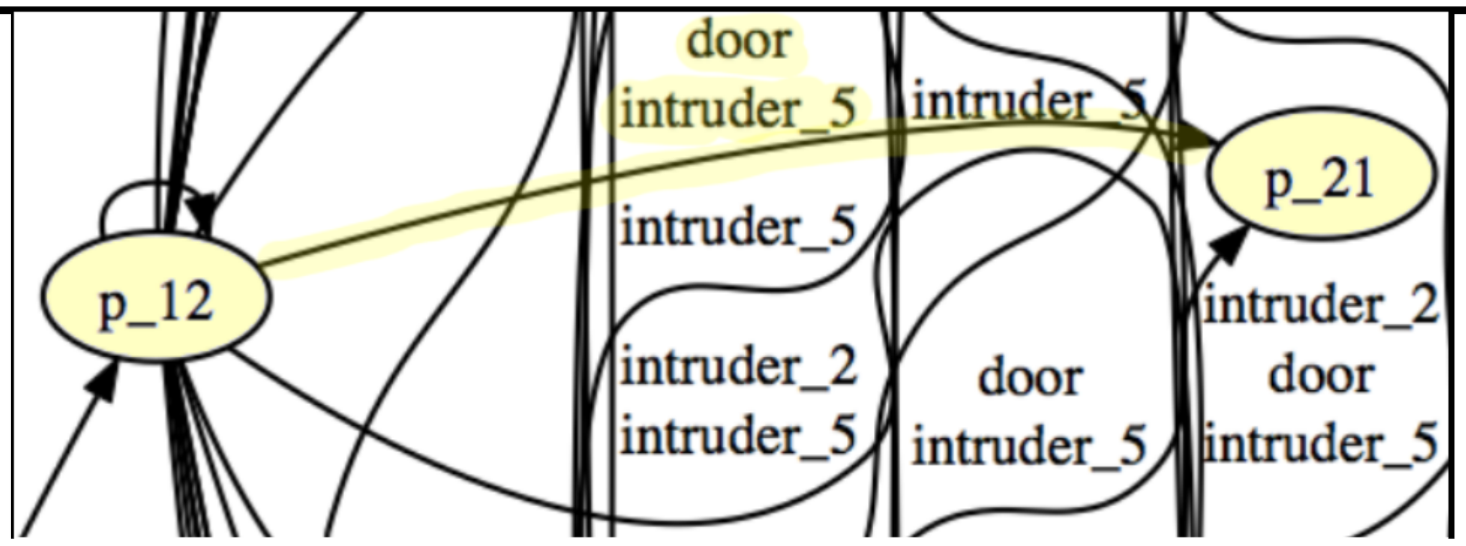
$\mathcal{X} = \{\pi_{intruder_2}, \pi_{intruder_5}, \pi_{door}\}$ $\quad \mathcal{Y} = \{\pi_p \mid p \in \{12, 21\}\}$

$\cup\{\pi_{i_r} \mid i \in \{1,2\}, r \in \{1,2,3,4,5,6\}\}$

completion of motion of robot *i* to room *r*

Synthesized Automaton (Excerpt)

Robot 1 is heading towards r2, robot 2 towards r5.



The system switches modes from p12 to p21 when the door is closed and an intruder is sensed in r5.

Specification (Excerpt)

$(\varphi 1_{r_1} \wedge \varphi 2_{r_6})$ #Initial (Environment)

(Robot 1 starts in r1, Robot 2 in r6)

$(\neg\pi_{intruder_2} \wedge \neg\pi_{intruder_5} \wedge \neg\pi_{door})$ #Initial (Environment)

(Initially no intruders, open door)

(π_{12}) #Initial (System)

(Robot 1 is initially assigned goal r_2, Robot 2 is initially assigned goal r_5)

$\wedge \quad \Box(\varphi 1_{r_1} \wedge \bigcirc \pi_{door} \Rightarrow \bigcirc \neg\varphi 1_{r_2})$ #Safety (Environment)

(If door closed, Robot 1 can't move from r_1 to r_2)

$\wedge \quad \Box(\varphi 1_{r_2} \wedge \bigcirc \pi_{door} \Rightarrow \bigcirc \neg\varphi 1_{r_1})$ #Safety (Environment)

(If door closed, Robot 1 can't move from r_2 to r_1)

...

$\wedge \quad \Box(\varphi 2_{r_5} \wedge \bigcirc \pi_{door} \Rightarrow \bigcirc \neg\varphi 2_{r_6})$ #Safety (Environment)

(If door closed, Robot 2 can't move from r_5 to r_6)

$\wedge \quad \Box(\varphi 2_{r_6} \wedge \bigcirc \pi_{door} \Rightarrow \bigcirc \neg\varphi 2_{r_5})$ #Safety (Environment)

(If door closed, Robot 2 can't move from r_6 to r_5)

1 $\quad \wedge \quad \Box\Diamond(\pi_{intruder2} \Rightarrow \bigcirc(\varphi 1_{r_2} \vee \varphi 2_{r_2}))$ #Liveness (System)

(If an intruder is detected in r_2, either Robot 1 or 2 should go to r_2)

2 $\quad \wedge \quad \Box\Diamond(\pi_{intruder5} \Rightarrow \bigcirc(\varphi 1_{r_5} \vee \varphi 2_{r_5}))$ #Liveness (System)

(If an intruder is detected in r_5, either Robot 1 or 2 should go to r_5)

Reactive Switching Protocols for Multi-Robot High-Level Tasks

Vasumathi Raman

Automatically generating provably correct control from high-level specifications for teams of interchangeable robots