Formal Methods, Machine Learning, and Cyber-Physical Systems

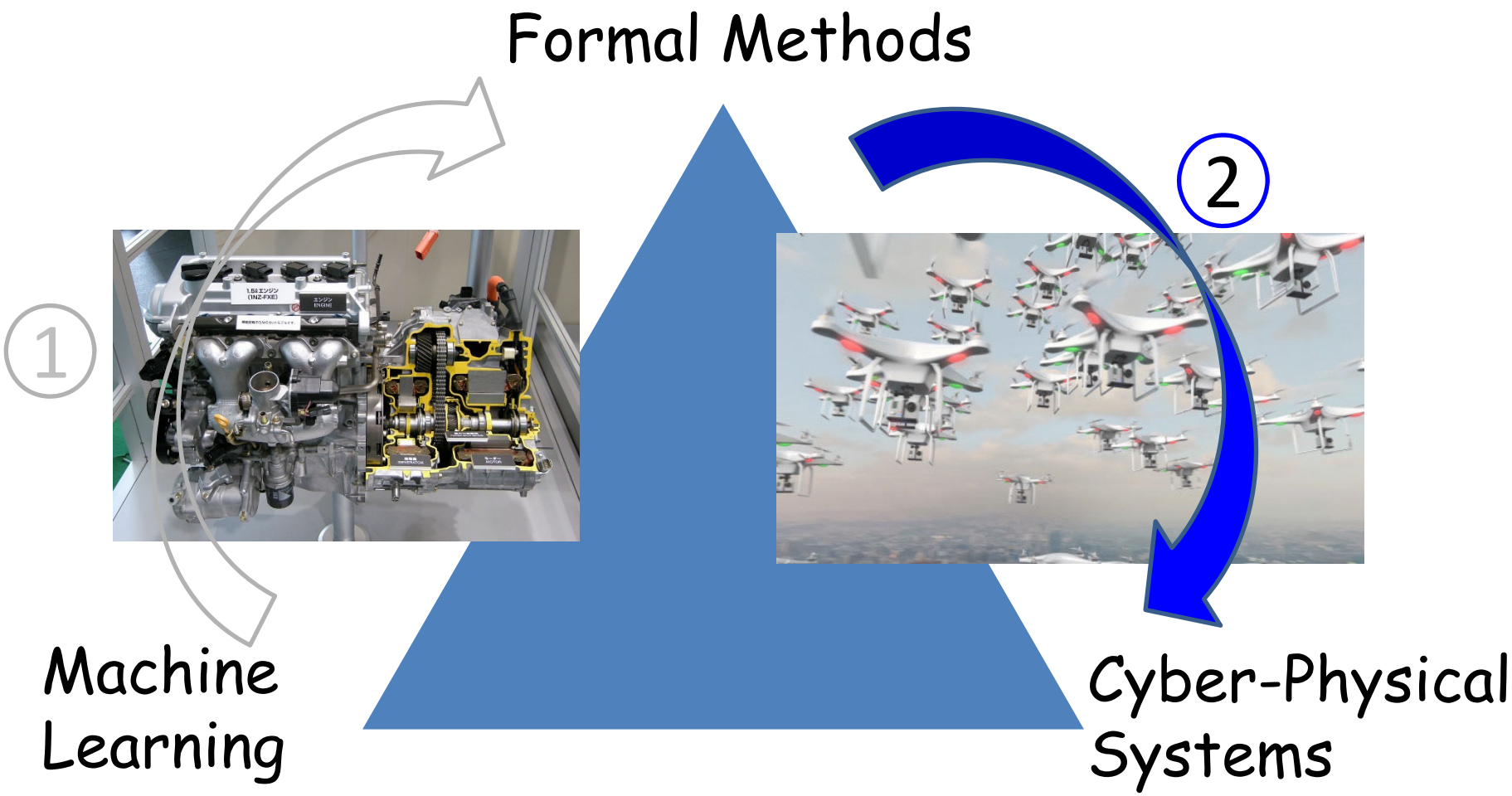
Sanjit A. Seshia

Professor

UC Berkeley

ATVA 2018 Tutorial
October 7, 2018

Connections in this Lecture



Synthesis of Controllers/Plans for Cyber-Physical Systems

Joint work with:

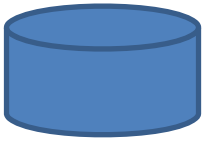
Ankush Desai, Shromona Ghosh, Pierluigi Nuzzo, Indranil Saha, Yasser Shoukry, Vijay Kumar, George Pappas, Shaz Qadeer, Alberto Sangiovanni-Vincentelli, Paulo Tabuada

[Shoukry et al., HSCC 2017, CDC 2017, Proc. IEEE 2018; Desai et al. ICCPS 2017; Saha et al., IROS 2014]

Goal: Correct-by-Construction Motion Planning for Robotics

Declarative Task Specification (Temporal Logic) [+ Examples]

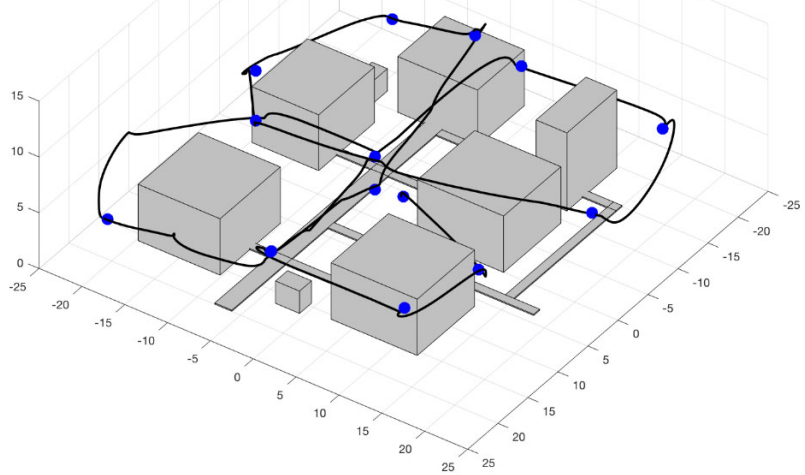
Compiler



Component Library

Safe, Correct Executable Software

Drone executing plan in ROS/Gazebo Simulator



Challenges

- Scalable synthesis of motion plans blending high-level (discrete) and low-level (continuous) control

- Implementation in software on top of networked robotics platforms

- Dealing with untrusted components, uncertainty in environment, and ML-based perception

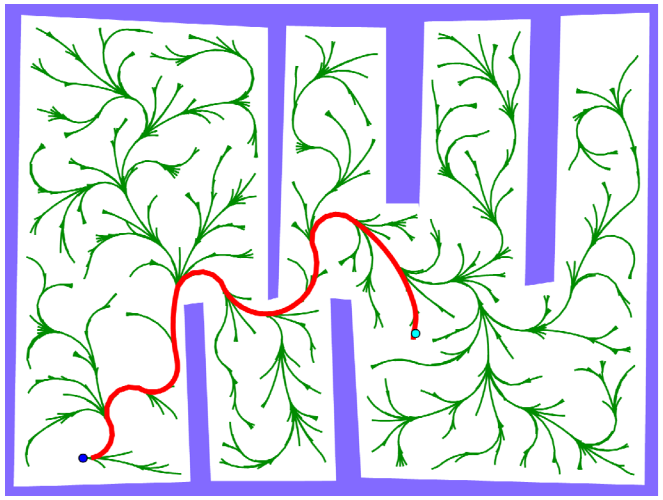
Motion Planning Problem

- Problem: Compute an obstacle-free trajectory that is feasible with the given robot dynamics within the target workspace.
- Core problem for autonomous vehicles
- Challenges:
  - Handling high-level tasks (captured by formalisms such as Linear Temporal Logic)
  - Handling distributed team of robots
  - Handling uncertainty

[Shoukry et al., HSCC 2017, CDC 2017, Proc. IEEE 2018; Saha et al., IROS 2014]

Related Work: Sampling-based Methods

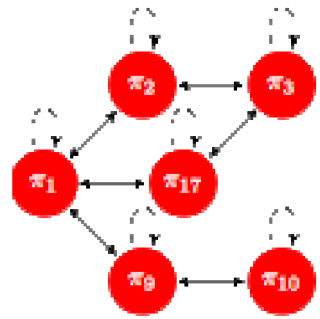
- S. Karaman and E. Frazzoli, “Optimal kinodynamic motion planning using incremental sampling-based methods,” CDC 2010.
- A. Bhatia, L. E. Kavraki, and M. Y. Vardi, “Sampling-based motion planning with temporal goals,” ICRA 2010.
- A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano Perez, “LQR-RRT: Optimal sampling-based motion planning with automatically derived extension heuristics,” ICRA 2012
- Do not handle high-level specifications given by formal models, such as LTL.



Related Work: Abstraction Based Techniques

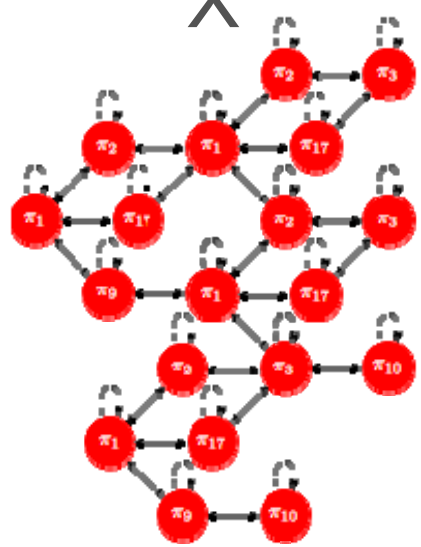
Discrete ← Continuous

- P. Tabuada and G. J. Pappas, "Linear time logic control of discrete-time linear systems," TAC 2006.
- M. Kloetzer and C. Belta, "Temporal Logic Planning and Control of Robotic Swarms by Hierarchical Abstractions," TAC 2007.
- G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," Automatica 2009.
- ...



$$x^{(t+1)} = Ax^{(t)} + Bu^{(t)}$$
$$y^{(t)} = Cx^{(t)}$$

X

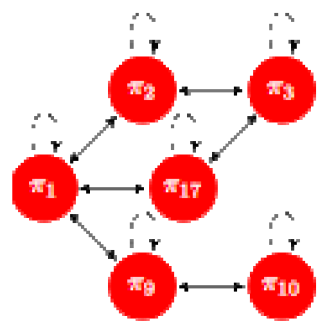


Scales poorly as the number of continuous states increases

Related Work: Optimization Based Techniques

Discrete → Continuous

- Alberto Bemporad and Manfred Morari, “Control of systems integrating logic, dynamics, and constraints,” Automatica, 35(3), 1999.
- E. M. Wolff, U. Topcu, and R. M. Murray, “Optimization-based trajectory generation with linear temporal logic specifications,” ICRA 2014.
- V. Raman, A. Donze, D. Sadigh, R. Murray, S. Seshia, “Reactive synthesis from signal temporal logic specifications,” HSCC 2015.
- …



$$x^{(t+1)} = Ax^{(t)} + Bu^{(t)}$$
$$y^{(t)} = Cx^{(t)}$$

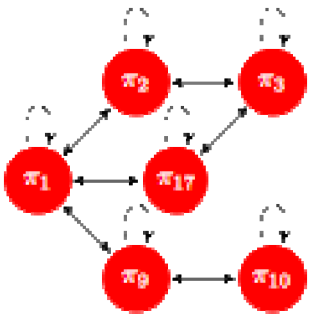
Scales poorly as the number of discrete states increases

$$\begin{aligned}
&\min \quad J(x) \\
&\text{subject to:} \\
&x^{(t+1)} = Ax^{(t)} + Bu^{(t)} \\
&-b_1^{(t)} + b_2^{(t+1)} + b_9^{(t+1)} = 0 \\
&-b_2^{(t)} + b_1^{(t+1)} + b_3^{(t+1)} = 0 \\
&\vdots \\
&x^{(0)} - \overline{x} = Mb_1^{(0)}
\end{aligned}$$

Two Extremes

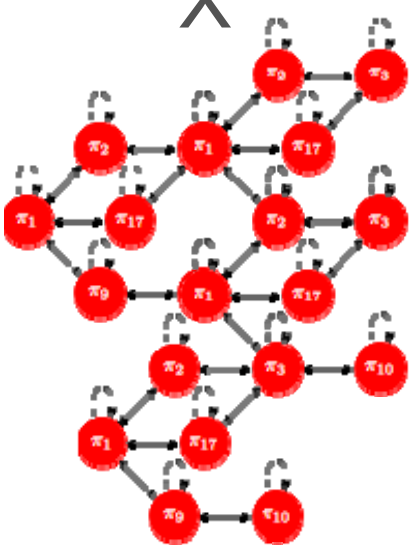
Discrete ← Continuous

(Abstraction based)



$$x^{(t+1)} = Ax^{(t)} + Bu^{(t)}$$
$$y^{(t)} = Cx^{(t)}$$

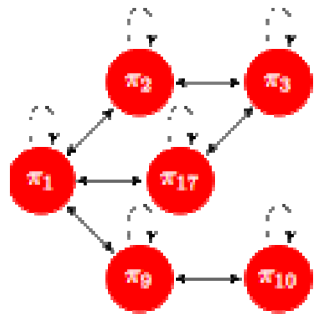
X



Scales poorly as the number of continuous states increases

Discrete → Continuous

(Optimization based)



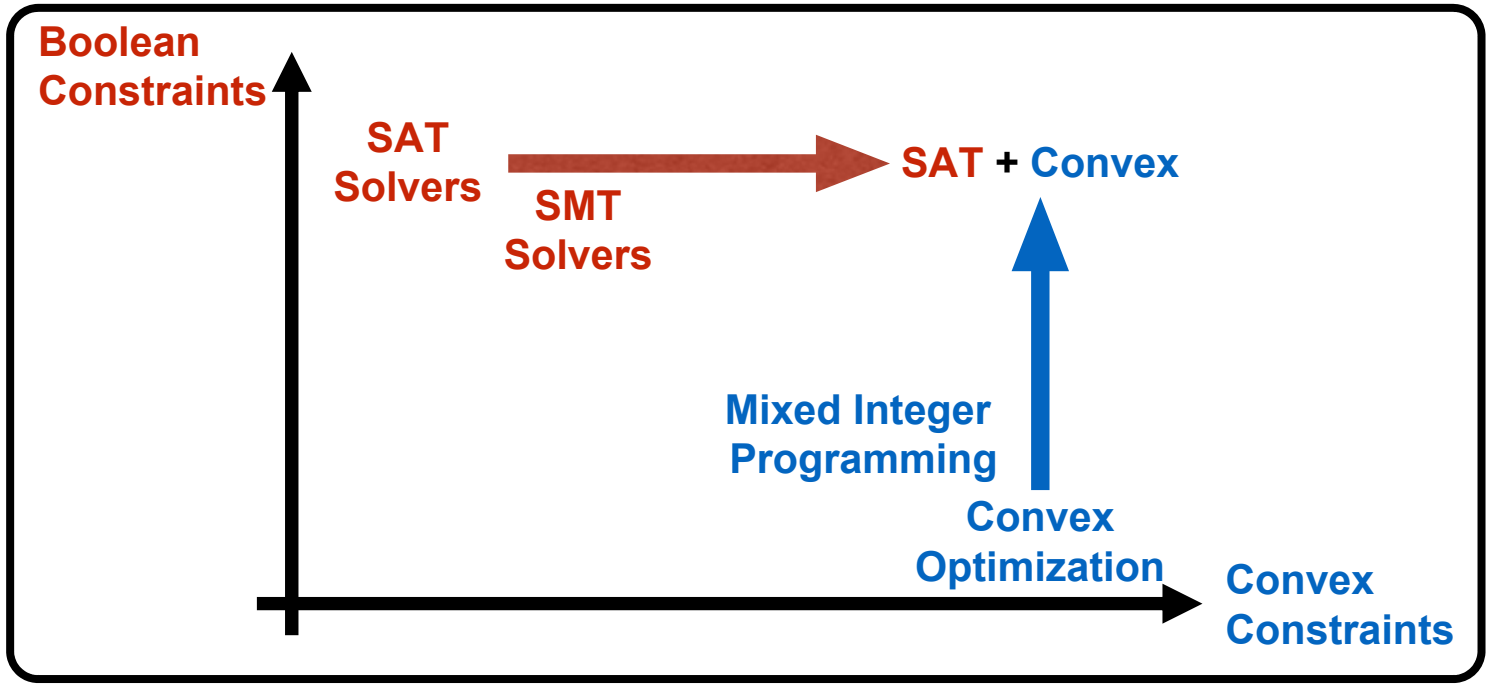
$$x^{(t+1)} = Ax^{(t)} + Bu^{(t)}$$
$$y^{(t)} = Cx^{(t)}$$

$$\begin{aligned} &\min \quad J(x) \\ &\text{subject to:} \\ &x^{(t+1)} = Ax^{(t)} + Bu^{(t)} \\ &-b_1^{(t)} + b_2^{(t+1)} + b_9^{(t+1)} = 0 \\ &-b_2^{(t)} + b_1^{(t+1)} + b_3^{(t+1)} = 0 \\ &\vdots \\ &x^{(0)} - \overline{x} = Mb_1^{(0)} \end{aligned}$$

Scales poorly as the number of discrete states increases

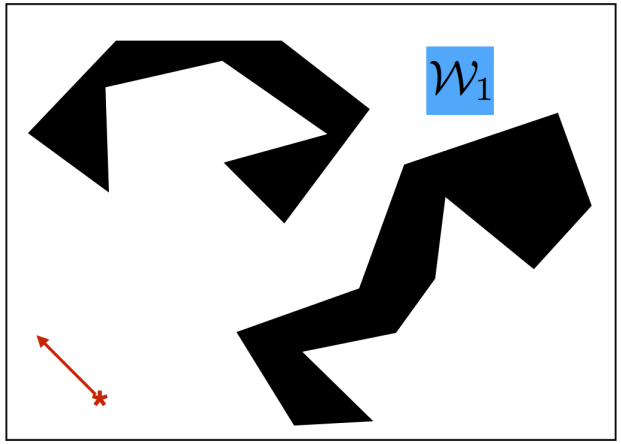
Satisfiability Modulo Convex Programming [Shoukry et al., HSCC 2017]

- SAT Solvers: central tools in formal methods to reason about discrete dynamics.
- Convex Optimization: central tools in control theory to reason about continuous dynamics.
- Approach for hybrid dynamics: SAT Solving + Convex Programming



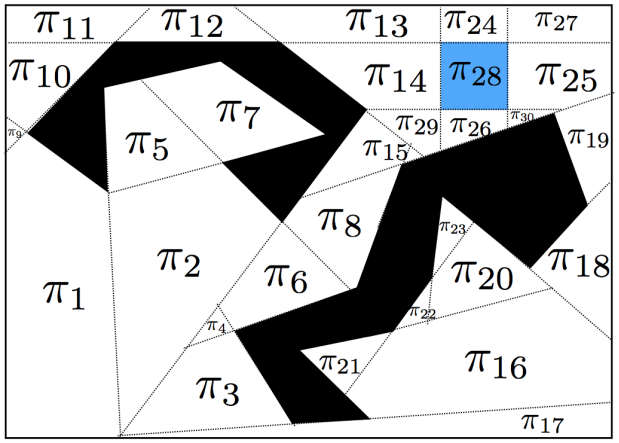
Motivating Example: Obstacle Avoidance

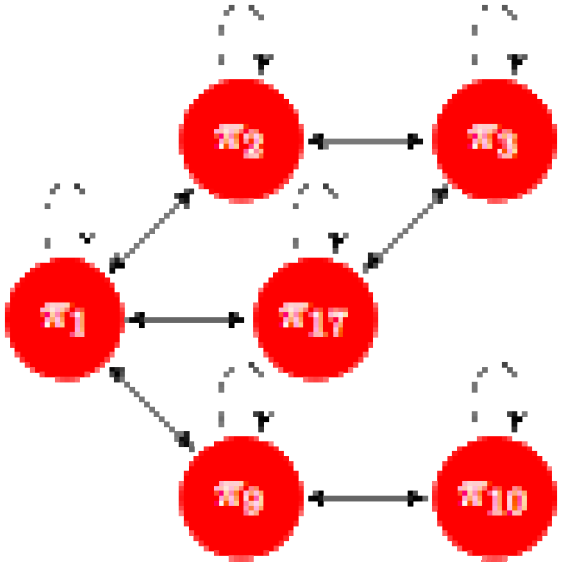
- Given:
  - Robot dynamics (linear)
  - Input constraints
  - Initial and final states
- Generate the input sequence (controller)



Motivating Example: Obstacle Avoidance

- Given:
  - Robot dynamics (linear)
  - Input constraints
  - Initial and final states
- Generate the input sequence (controller)

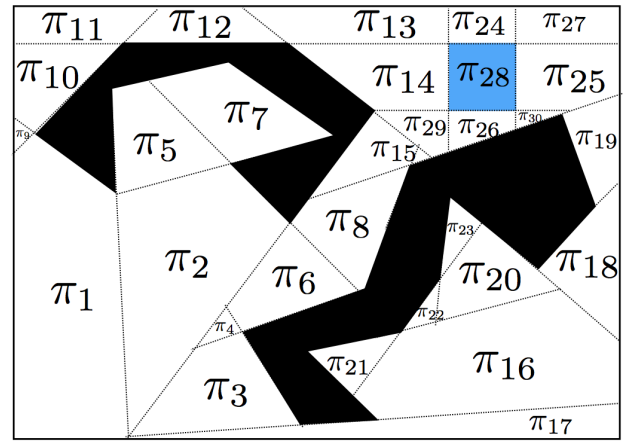


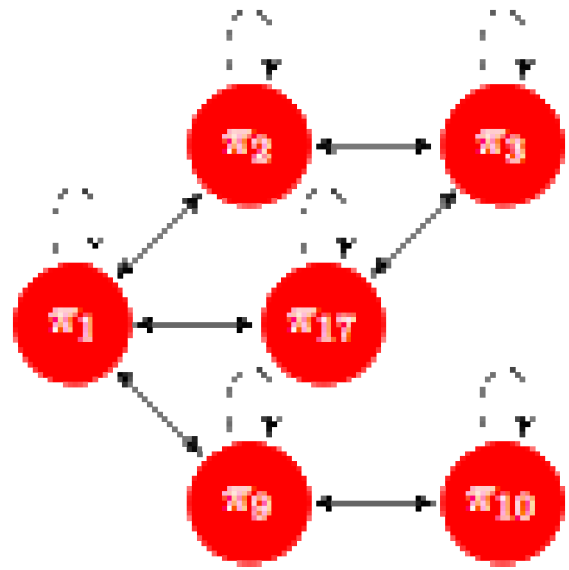


Motivating Example: Obstacle Avoidance

$\varphi ::= b^0_{\text{start}}$ (Initial partition)

$\wedge\, b^L_{\text{goal}}$ (Goal partition)





Motivating Example: Obstacle Avoidance

$$
\begin{aligned}
\varphi ::= \; & b^0_{\text{start}} && \text{(Initial partition)} \\
\wedge \; & b^L_{\text{goal}} && \text{(Goal partition)} \\
\wedge \; & b^j_i \rightarrow \bigvee_{i' \in \Pi(i)} b^{j+1}_{i'} && \forall j \in \{0, \ldots, L-1\}, \\
& && i \in \{1, \ldots, m\} \\
& && \text{(Adjacency constraints)} \\
\wedge \; & \sum_{i=1}^{m} b^j_i = 1 && \forall j \in \{0, \ldots, L-1\} \\
\wedge \; & x^{j+1} = Ax^j + Bu^j && \forall j \in \{0, \ldots, L-1\} \\
& && \text{(Robot dynamics)} \\
\wedge \; & \|u^j\| \leq \overline{u} && \forall j \in \{0, \ldots, L-1\} \\
& && \text{(Input constraints)} \\
\wedge \; & x^0 = x && \text{(Initial state)} \\
\wedge \; & b^j_i \rightarrow x^j \in \mathcal{P}_i && \forall j \in \{0, \ldots, L-1\} \\
& && \text{(Region constraints)}
\end{aligned}
$$

$\phi(b)$

$g(x) \leq 0$

$\phi'(b) \vee g'(x) \leq 0$

Motivating Example: Obstacle Avoidance

Definition: Monotone SMC Formula

$$
\begin{aligned}
\mathit{formula} &::= \{\mathit{clause} \wedge\}^{*}\mathit{clause} \\
\mathit{clause} &::= (\{\mathit{literal} \vee\}^{*}\mathit{literal}) \mid \mathit{pB\_predicate} \\
\mathit{literal} &::= \mathit{bool\_var} \mid \neg\mathit{bool\_var} \mid \top \mid \bot \mid \\
&\qquad \mathit{conv\_constraint} \\
\mathit{conv\_constraint} &::= \mathit{equation} \mid \mathit{inequality} \\
\mathit{equation} &::= \mathit{affine\_function} = 0 \\
\mathit{inequality} &::= \mathit{convex\_function}\ \mathit{relation}\ 0 \\
\mathit{relation} &::= < \mid \leq
\end{aligned}
$$

$\phi(b)$

$g(x) \leq 0$

$\phi'(b) \vee g'(x) \leq 0$

Applications: Controller Synthesis

Obstacle Avoidance

$\phi ::=$ initial partition
$\wedge$ goal partition
$\wedge$ adjacency partition
$\wedge$ robot dynamics
$\wedge$ input constraints
$\wedge$ initial state
$\wedge$ region constraints

LTL Motion Planning

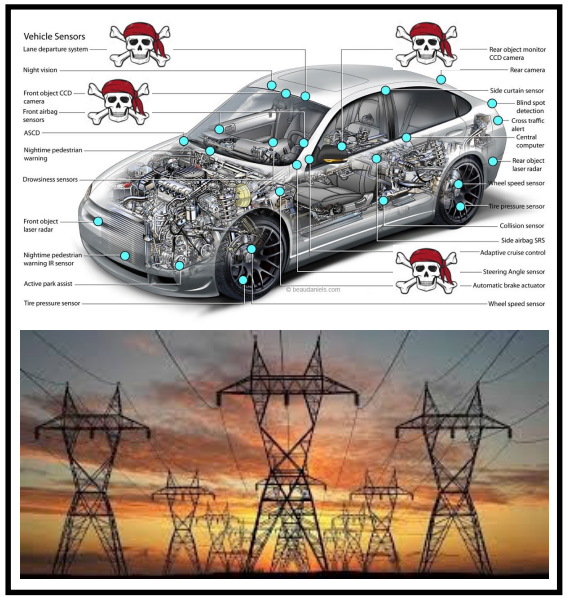
$\phi ::=$ LTL BMC encoding
$\wedge$ adjacency partition
$\wedge$ robot dynamics
$\wedge$ input constraints
$\wedge$ initial state
$\wedge$ region constraints

Multi-robot Motion Planning

$\phi ::=$ LTL BMC encoding
$\wedge$ adjacency partition
$\wedge$ robot dynamics
$\wedge$ input constraints
$\wedge$ initial state
$\wedge$ region constraints
$\wedge \big(x_{\text{robot 1}} - x_{\text{robot 2}} > \epsilon$
$\vee\ x_{\text{robot 2}} - x_{\text{robot 1}} > \epsilon$
$\vee\ y_{\text{robot 1}} - y_{\text{robot 2}} > \epsilon$
$\vee\ y_{\text{robot 2}} - y_{\text{robot 1}} > \epsilon$
$\big)$

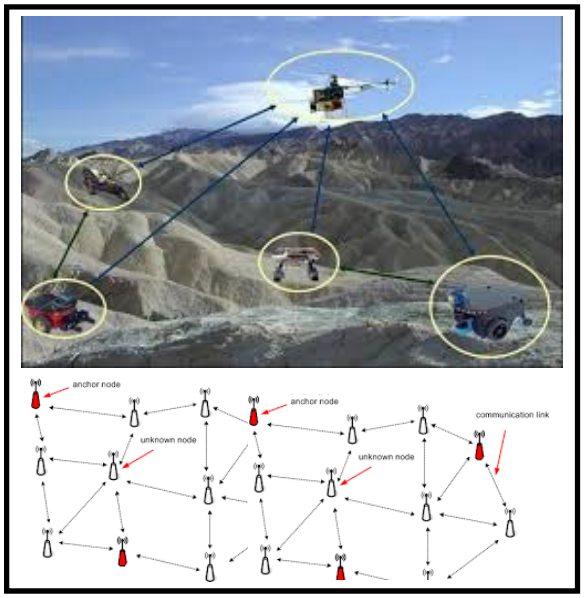
Applications: CPS Security

Secure State Estimation



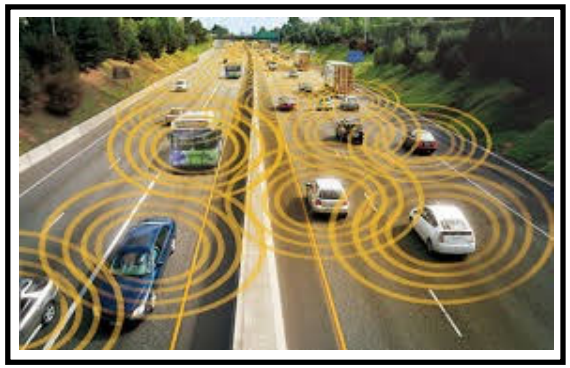
$$\phi(b, x) ::= \bigwedge_{i=1}^{p} \left(\neg b_i \Rightarrow \| Y_i - \mathcal{O}_i x \|_2^2 \leq 0 \right) \wedge \left(\sum_{i=1}^{p} b_i \leq \bar{s} \right).$$

Secure Localization



$$\phi(b, X) ::= \bigwedge_{(i,j)\in\mathcal{E}} \left(\neg b_{ij} \Rightarrow |g_{ij}^T \, Z \, g_{ij} - m_{ij}^2| \leq 0 \right) \wedge Z = \begin{bmatrix} Y & X^T \\ X & I \end{bmatrix} \succeq 0 \wedge \left(\sum_{(i,j)\in\mathcal{E}}^{p} b_{ij} \leq \bar{s} \right).$$

Secure Traffic Routing



$b_i = (0,0)$ ith car is honest

$b_i = (0,1)$ ith car physically exist but dishonest

$b_i = (1,0)$ ith car is Sybil

if car i does not report car j

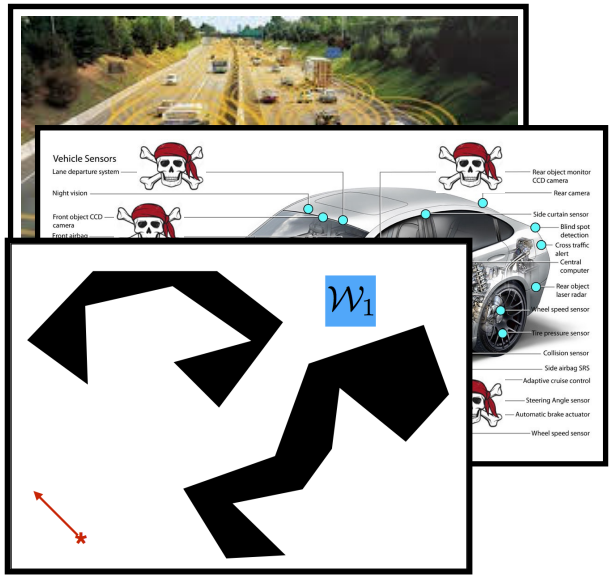
then both i and j can not be honest

...

for the cluster $\mathcal{I}$ of honest cars

$$\frac{1}{N}\sum_{k=1}^{N}(v_{\mathcal{I}}(k) - v_{loop}(k))(v_{\mathcal{I}}(k) - v_{loop}(k))^T \succeq \frac{1}{N}\sum_{k=1}^{N}(v_{\overline{\mathcal{I}}}(k) - v_{loop}(k))(v_{\overline{\mathcal{I}}}(k) - v_{loop}(k))^T$$

Satisfiability Modulo Convex Programming



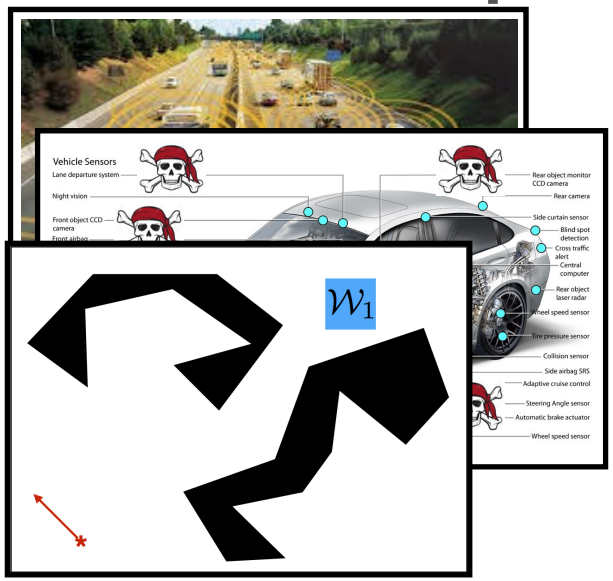
The satisfiability of the monotone SMC formula can always be cast as a feasibility problem for a finite disjunction of convex constraints.

$$
\begin{aligned}
& \big((g_1(x) \leq 0) \wedge \phi_1(b)\big) \\
\vee \quad & \big((g_2(x) \leq 0) \wedge \phi_2(b)\big) \\
\vee \quad & \ldots \\
\vee \quad & \big((g_k(x) \leq 0) \wedge \phi_k(b)\big)
\end{aligned}
$$

Monotone SMC Formula

Theorem:
Any feasibility problem for a finite disjunction of convex programs can be posed as a satisfiability problem for a monotone SMC formula.

Complexity of Solving SMC



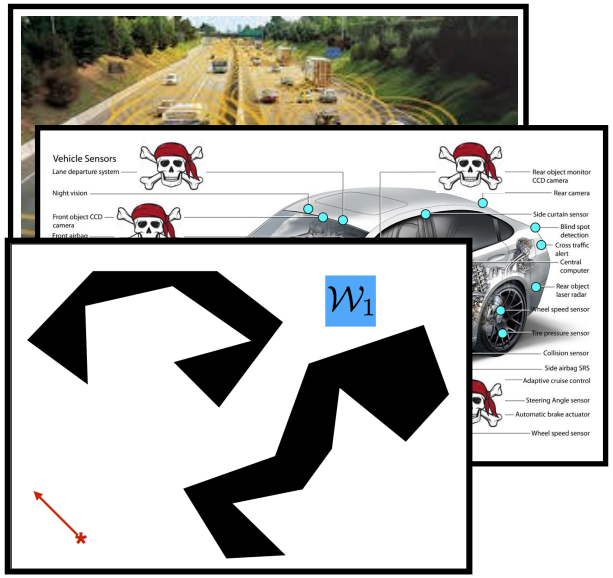
$$\text{Complexity} = \#\text{Iterations} \times (\text{Time}_{\phi(b)} + \text{Time}_{g(x)\leq 0})$$

$$\begin{aligned} & \big((g_1(x) \leq 0) \wedge \phi_1(b)\big) \\ \vee \quad & \big((g_2(x) \leq 0) \wedge \phi_2(b)\big) \\ \vee \quad & \ldots \\ \vee \quad & \big((g_k(x) \leq 0) \wedge \phi_k(b)\big) \end{aligned}$$

Monotone SMC Formula

Reduce the number of iterations?

Solving SMC



$$\text{Complexity} = \#\text{Iterations} \times \left(\underbrace{\text{Time}_{\phi(b)}}_{\text{“small”}} + \underbrace{\text{Time}_{g(x)\leq 0}}_{\text{polynomial}}\right)$$

$$\begin{aligned} & \big((g_1(x) \leq 0) \wedge \phi_1(b)\big) \\ \vee \quad & \big((g_2(x) \leq 0) \wedge \phi_2(b)\big) \\ \vee \quad & \ldots \\ \vee \quad & \big((g_k(x) \leq 0) \wedge \phi_k(b)\big) \end{aligned}$$

Monotone SMC Formula

Key idea:
abstraction-based search

Boolean Expansion

- Step 0:
  - Given a monotone SMC formula $\phi(b, x)$, construct its Boolean expansion:

$$\phi'(a, b, x) = \phi_B \wedge \bigwedge_{i=1}^{|C|} (a_i \implies (g_i(x) \leq 0))$$

where ϕ_B is obtained from $\phi(b, x)$ by replacing each convex constraint $g_i(x) \leq 0$ with a Boolean variable a_i

Lemma:

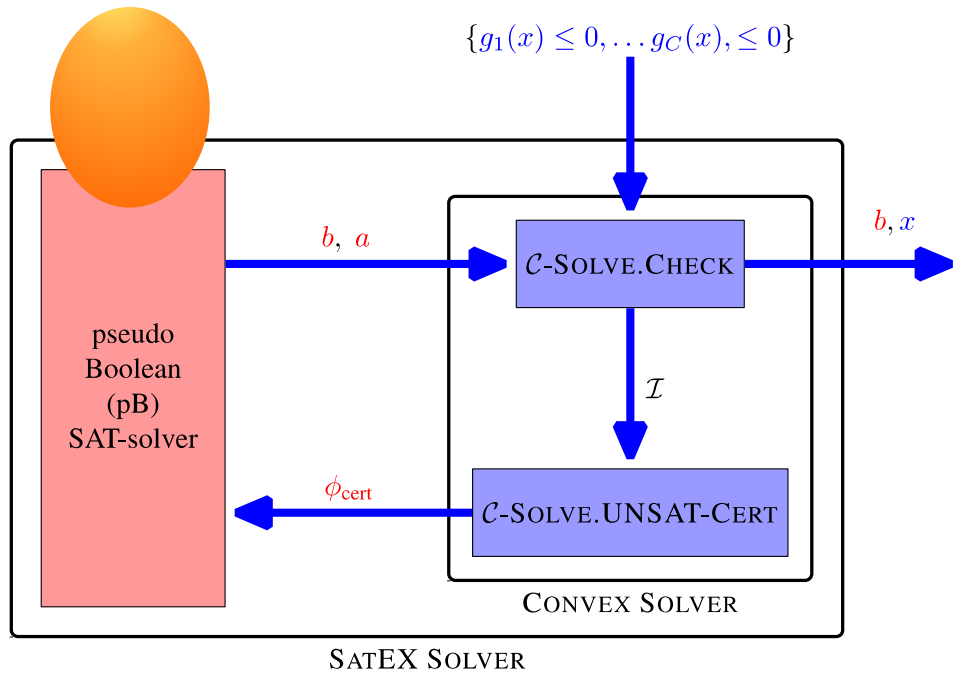
$\phi(b, x)$ and $\phi'(a, b, x)$ are equi-satisfiable.

Solver Operation

- Step I: Solve the "Boolean abstraction" of the problem.
- Step II: Extract which convex constraints are active.
- Step III: Check the satisfiability of:

$$\min_{x \in \mathbb{R}^n} \quad 1$$
$$\text{s.t.} \quad g_j(x) \leq 0$$
$$j \in \{1, \ldots, |C|, a_j = 1\}$$

$$\phi'(a, b, x) = \phi_B \wedge \bigwedge_{i=1}^{|C|} (a_i \implies (g_i(x) \leq 0))$$

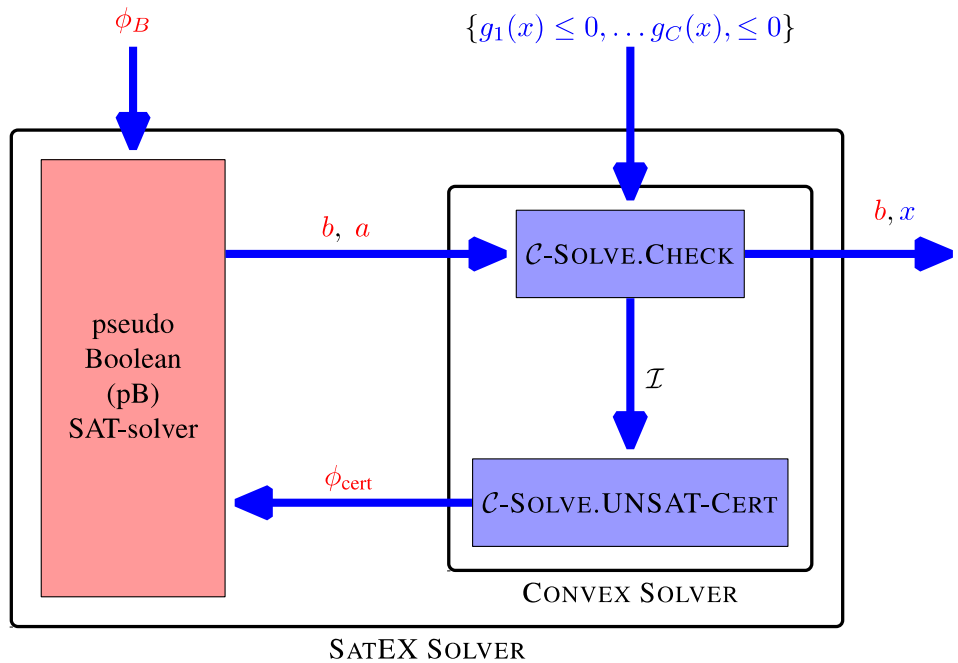


- Step IV: Generate UNSAT certificate:

$$\phi_{\text{trivial-ce}} = \bigvee_{j \in \text{supp}(a)} \neg a_j$$

Solver Operation

$$\phi'(a, b, x) = \phi_B \wedge \bigwedge_{i=1}^{|C|} (a_i \implies (g_i(x) \leq 0))$$



Performance?

$$\text{Complexity} = \#\text{Iterations} \times (\text{Time}_{\phi(b)} + \text{Time}_{g(x) \leq 0})$$

Minimal UNSAT Certificate

- Finding a minimal UNSAT certificate is equivalent to finding an Irreducible Infeasible Set (IIS) of:

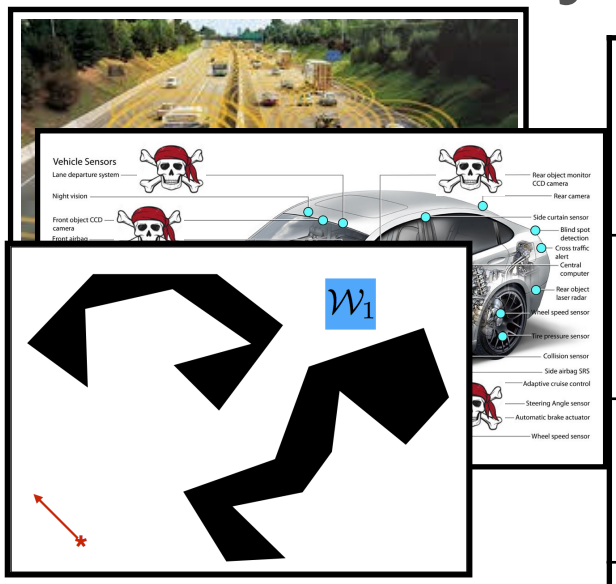
$$\min_{x \in \mathbb{R}^n} \quad 1 \qquad \text{s.t.} \quad g_j(x) \leq 0 \qquad j \in \{1, \ldots, |C|, a_j = 1\}$$

- NP-hard in general.

- However, our problem has more structure than general IIS problems.

- Can we exploit the structure of the Boolean constraints to help find the IIS of the convex program?

Summary of UNSAT certificates



$$\begin{aligned} & \big((g_1(x) \le 0) \wedge \phi_1(b)\big) \\ \vee \quad & \big((g_2(x) \le 0) \wedge \phi_2(b)\big) \\ \vee \quad & \ldots \\ \vee \quad & \big((g_k(x) \le 0) \wedge \phi_k(b)\big) \end{aligned}$$

Monotone SMC Formula

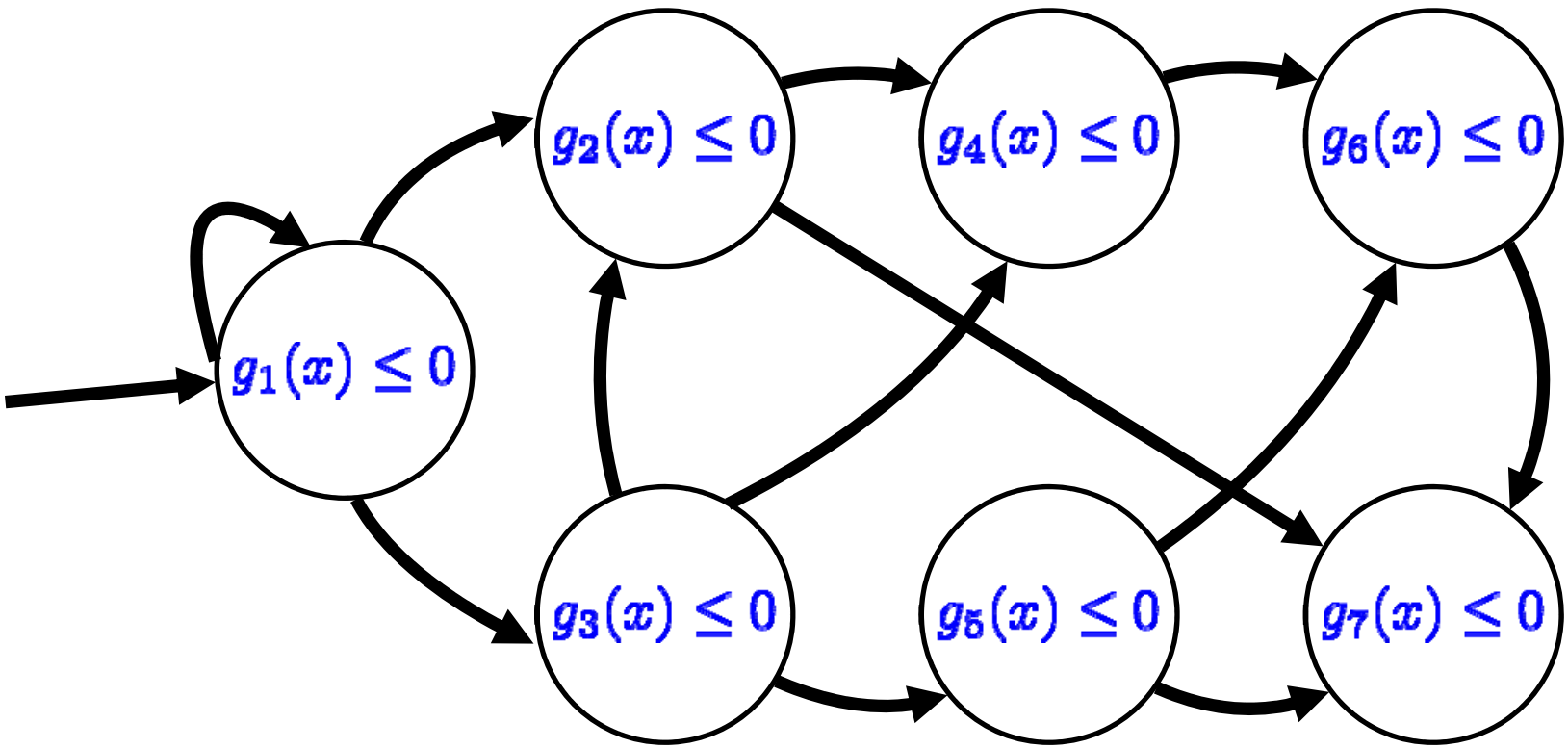
UNSAT Certificate	Minimal	Complexity (number of convex problems)
Trivial	No	Constant
IIS	Yes	Exponential
Sum of Slacks	Yes*	Linear
Minimum Prefix	Yes*	Constant

* under reasonable technical assumptions

$$\text{Complexity} = \#\text{Iterations} \times \big(\underbrace{\text{Time}_{\phi(b)}}_{\text{“small”}} + \underbrace{\text{Time}_{g(x)\le 0}}_{\text{polynomial}}\big)$$

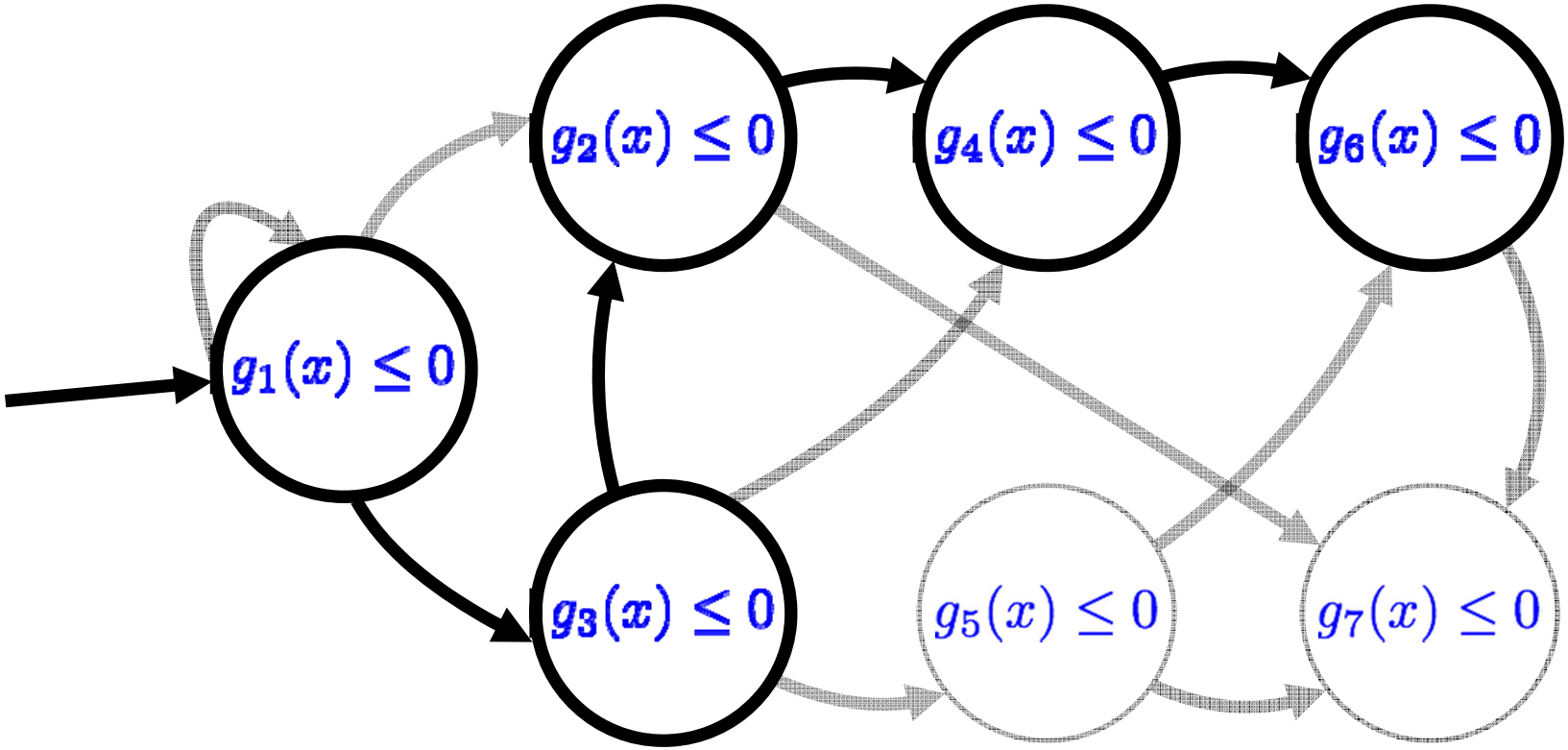
Minimum Prefix Certificates

Example: switched linear/convex systems, motion planning



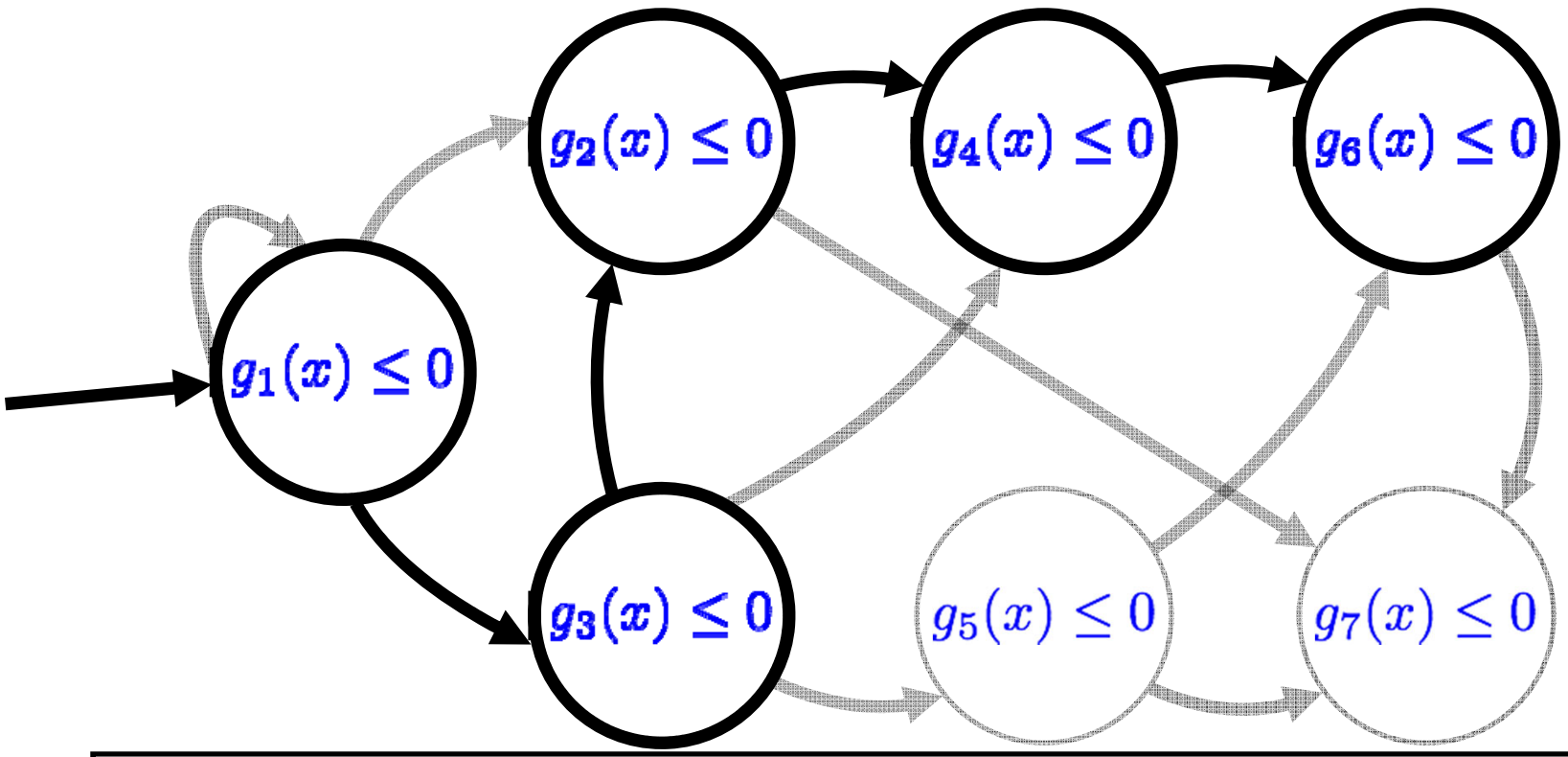
Minimum Prefix Certificates

Example: switched linear/convex systems, motion planning



Minimum Prefix Certificates

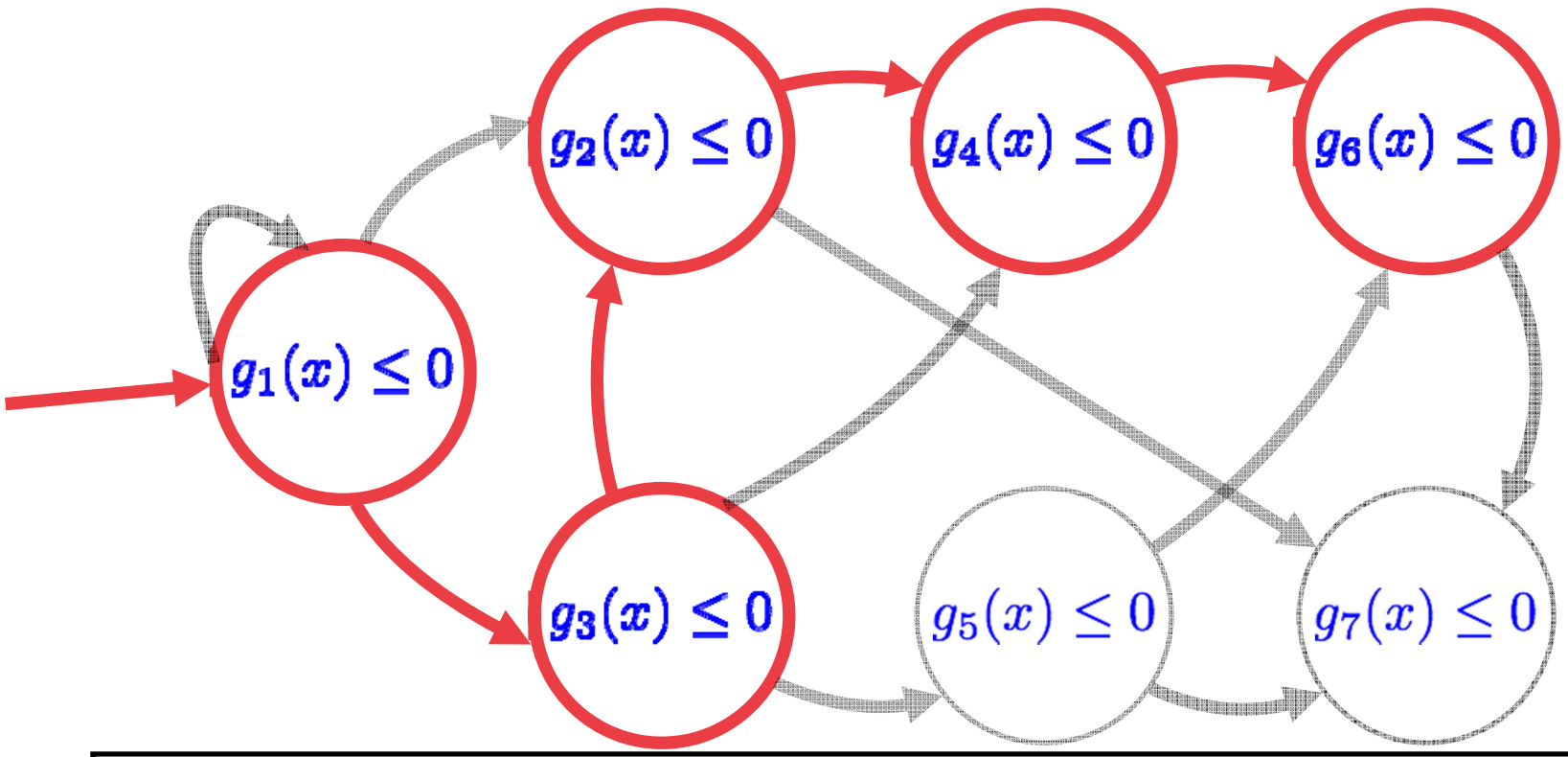
Example: switched linear/convex systems, motion planning



$$\min_{x \in \mathbb{R}^n} 1 \quad \text{s.t.} \quad (g_1(x) \leq 0) \wedge (g_3(x) \leq 0) \wedge$$
$$(g_2(x) \leq 0) \wedge (g_4(x) \leq 0) \wedge (g_6(x) \leq 0)$$

Minimum Prefix Certificates

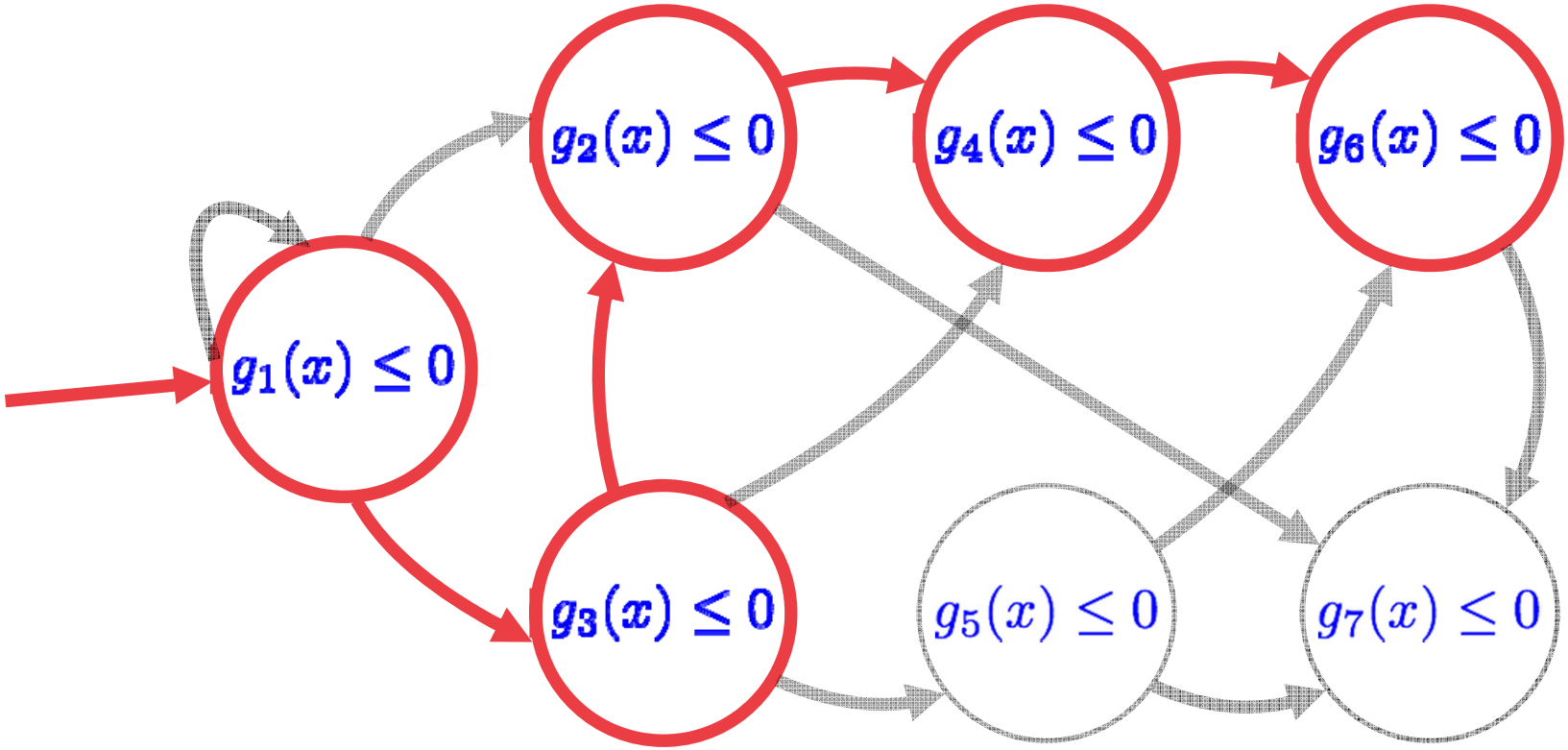
Example: switched linear/convex systems, motion planning



$$\min_{x \in \mathbb{R}^n} 1 \quad \text{s.t.} \quad (g_1(x) \leq 0) \wedge (g_3(x) \leq 0) \wedge (g_2(x) \leq 0) \wedge (g_4(x) \leq 0) \wedge (g_6(x) \leq 0)$$

Minimum Prefix Certificates

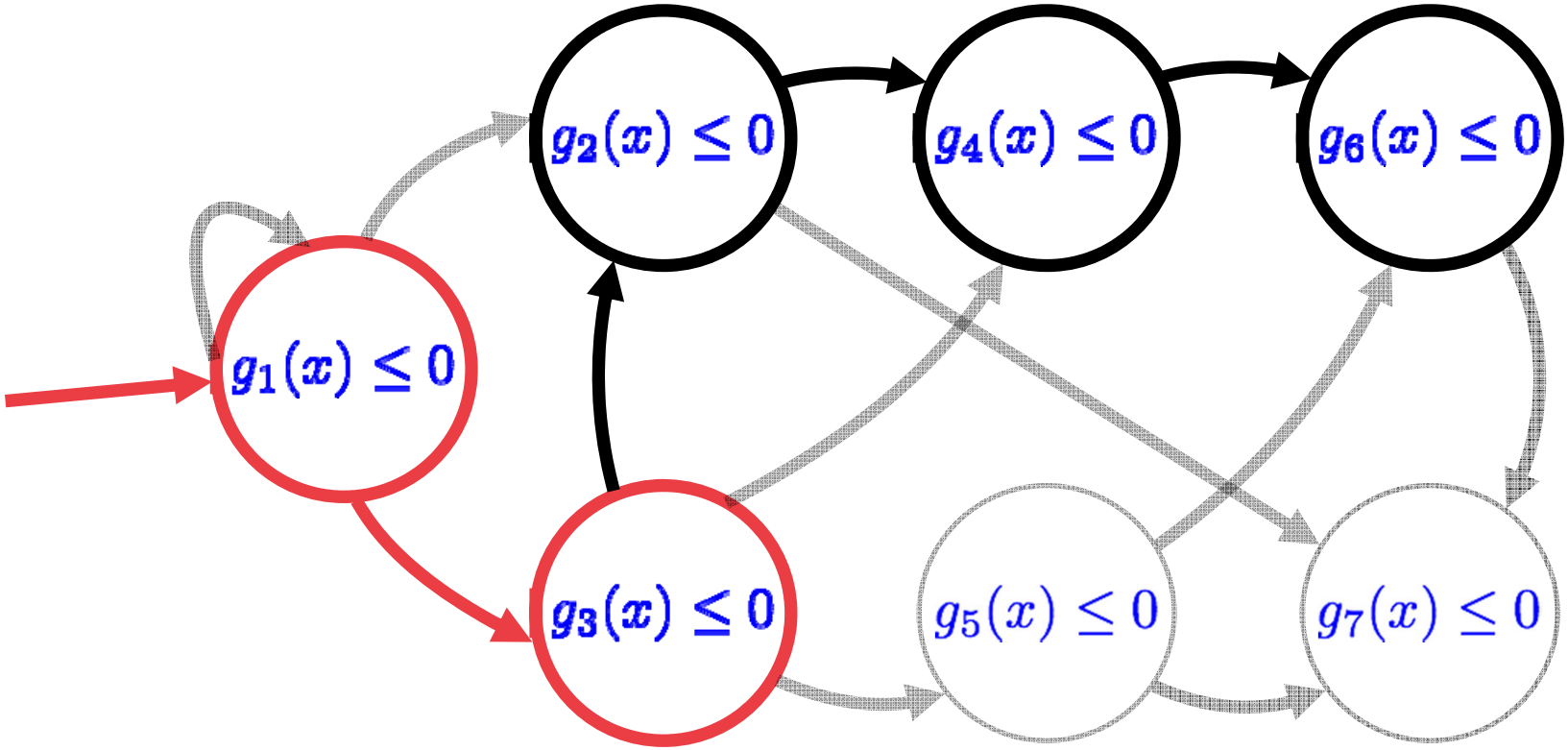
Example: switched linear/convex systems, motion planning



Key idea: Find the *"shortest"* UNSAT certificate (e.g., the shortest prefix witness of a safety property violation).

Minimum Prefix Certificates

Example: switched linear/convex systems, motion planning



Key idea: Find the *"shortest"* UNSAT certificate (e.g., the shortest prefix witness of a safety property violation).

Minimum Prefix Certificates

Definition: Positively Ordered Unate (POU) Function

A function is said to be POU with respect to an ordering of its variables if for all values of b_i, we have

$$u(0,\ldots,0) \leq u(b_1,0,\ldots,0) \leq u(b_1,b_2,0,\ldots,0) \leq \ldots$$
$$\ldots \leq u(b_1,b_2,\ldots,b_{m-1},0) \leq u(b_1,b_2,\ldots,b_{m-1},b_m).$$

Theorem:

Consider a monotone SMC formula $\phi(b,x)$ and its Boolean abstraction ϕ_B.

If:

(1) ϕ_B is positively ordered unate

(2) the domain of the real variables is bounded

then minimal UNSAT certificates exist and can be computed in constant time.

Minimum Prefix Certificates

$$\min_{x \in \mathbb{R}^n} \quad 1 \qquad \text{s.t.} \quad g_j(x) \leq 0 \qquad j \in \{1, \ldots, |C|, a_j = 1\}$$

$$\min_{\substack{s_1, \ldots, s_L \in \mathbb{R} \\ x \in \mathcal{W} \subseteq \mathbb{R}^n}} \sum_j |s_i|$$

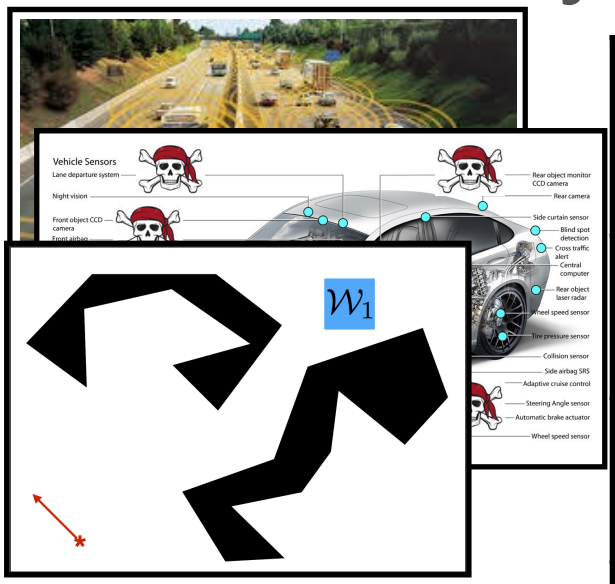
$$\text{s.t.} \quad g_j(x) \leq s_j, \qquad j \in \{1, \ldots, |C|, a_j = 1\}$$

$$\beta\left(\sum_{k=1}^{i-1} |s_k|\right) \leq |s_i| \qquad j \in \{1, \ldots, |C|, a_j = 1\}$$

Theorem:

Let s* be the solution of the above optimization problem. If k* is the minimal index such that |s*|> 0, then the certificate: $\phi_{\text{prefix}} = \bigvee_{j=1}^{k^*} \neg a_j$ is the minimum prefix certificate.

Summary of UNSAT certificates



$$\begin{aligned} & \big((g_1(x) \le 0) \wedge \phi_1(b)\big) \\ \vee\ & \big((g_2(x) \le 0) \wedge \phi_2(b)\big) \\ \vee\ & \ldots \\ \vee\ & \big((g_k(x) \le 0) \wedge \phi_k(b)\big) \end{aligned}$$

Monotone SMC Formula

UNSAT Certificate	Minimal	Complexity (number of convex problems)
Trivial	No	Constant
IIS	Yes	Exponential
Sum of Slacks	Yes*	Linear
Minimum Prefix	Yes*	Constant

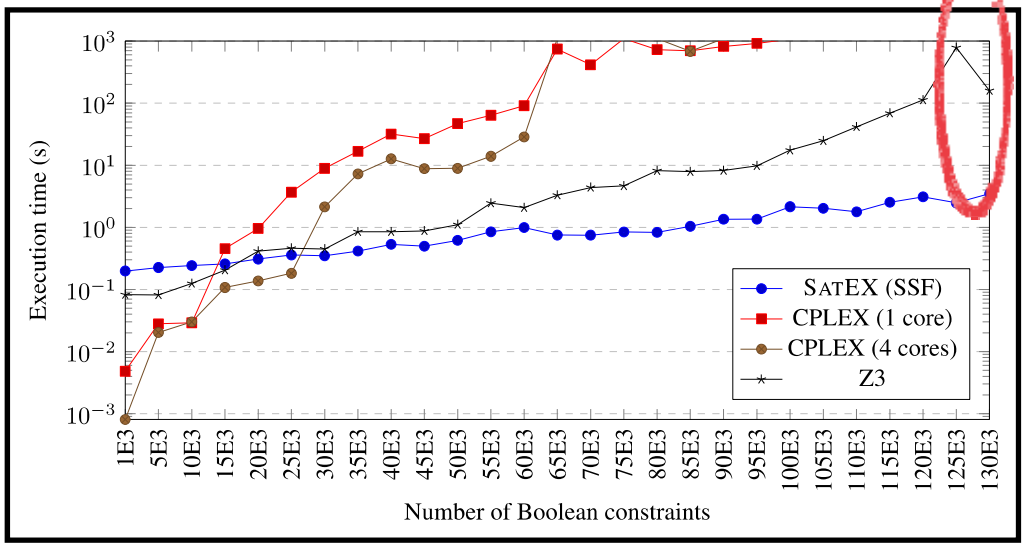
* under reasonable technical assumptions

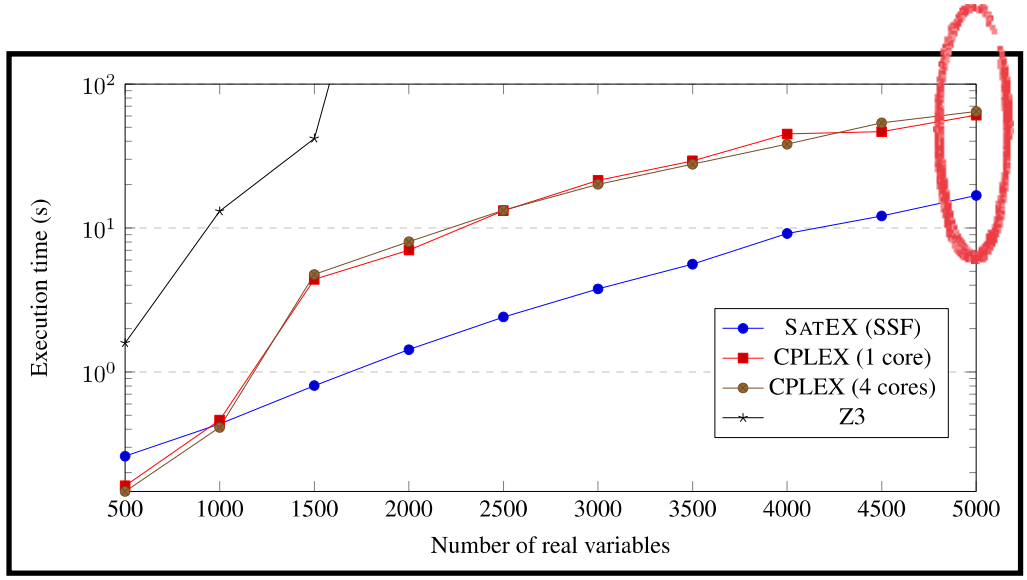
$$\text{Complexity} = \#\text{Iterations} \times \big(\underbrace{\text{Time}_{\phi(b)}}_{\text{“small”}} + \underbrace{\text{Time}_{g(x)\le 0}}_{\text{polynomial}}\big)$$

Scalability Results

10000 x

Increase the number of
Boolean constraints
#Boolean variables = 4800
#Real variables = 100

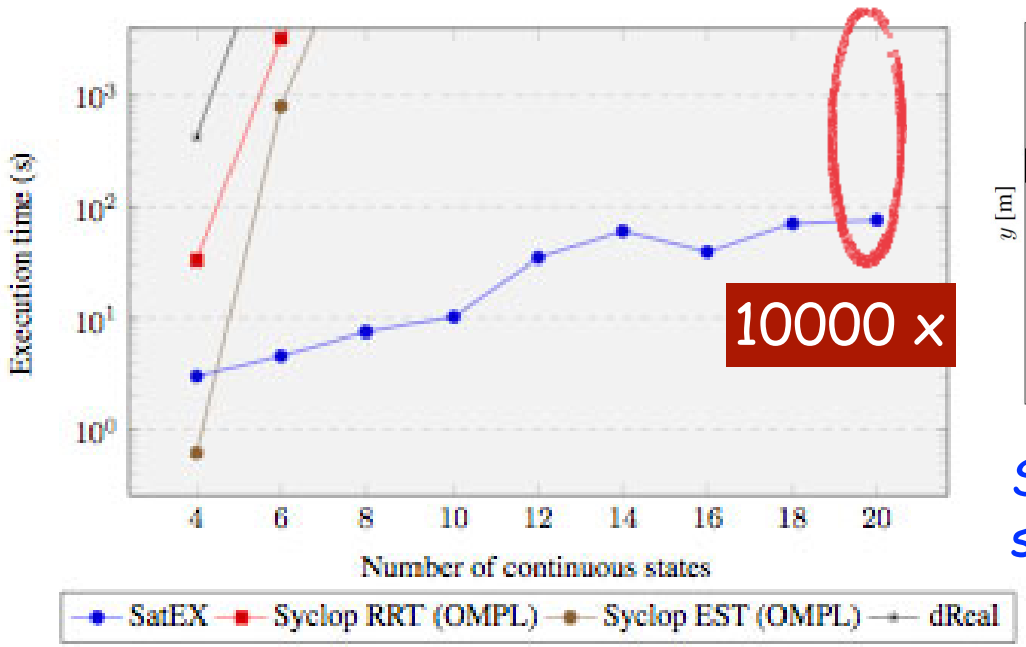


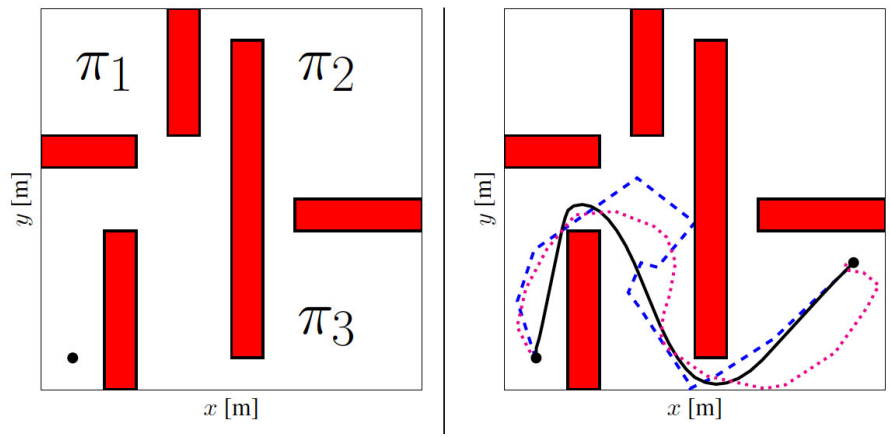


http://yshoukry.bitbucket.io/SatEX

Increase the number of
Real variables
#Boolean variables = 4800
#Boolean constraints = 7000

Results(1): Single Robot, Reach-Avoid



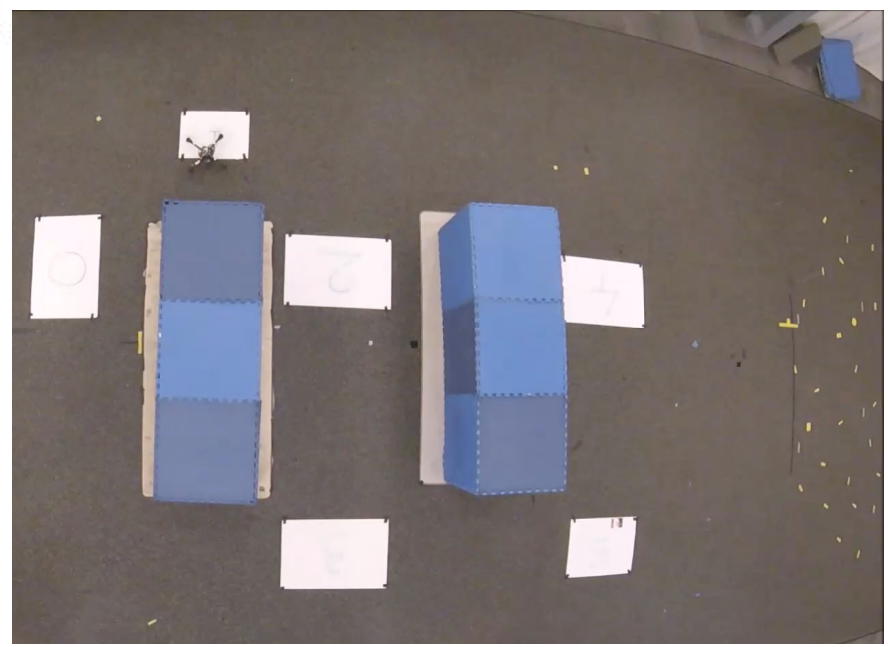


SMC generated trajectories (blue) ar
smoother than Syclop generated traje

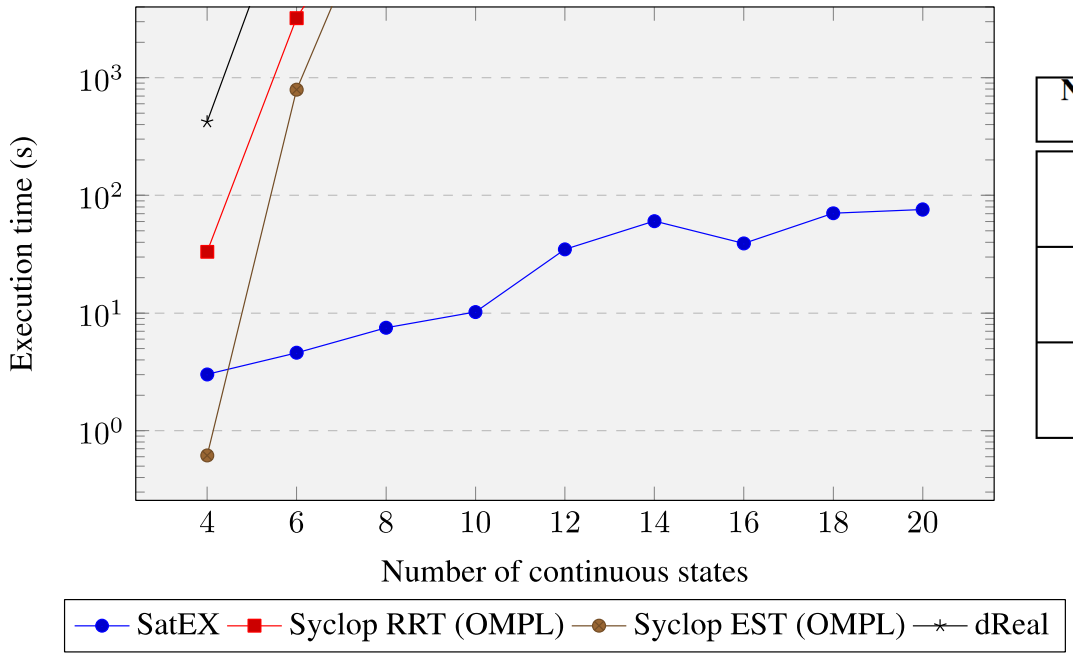
Syclop (Synergistic Combination of Layers Of Planning):
- High level planner + low level RRT/EST
- outperform traditional sampling-based algorithms by orders of magnitude

Z3 and MILP (LTL OPT) times out

time out =1 hour

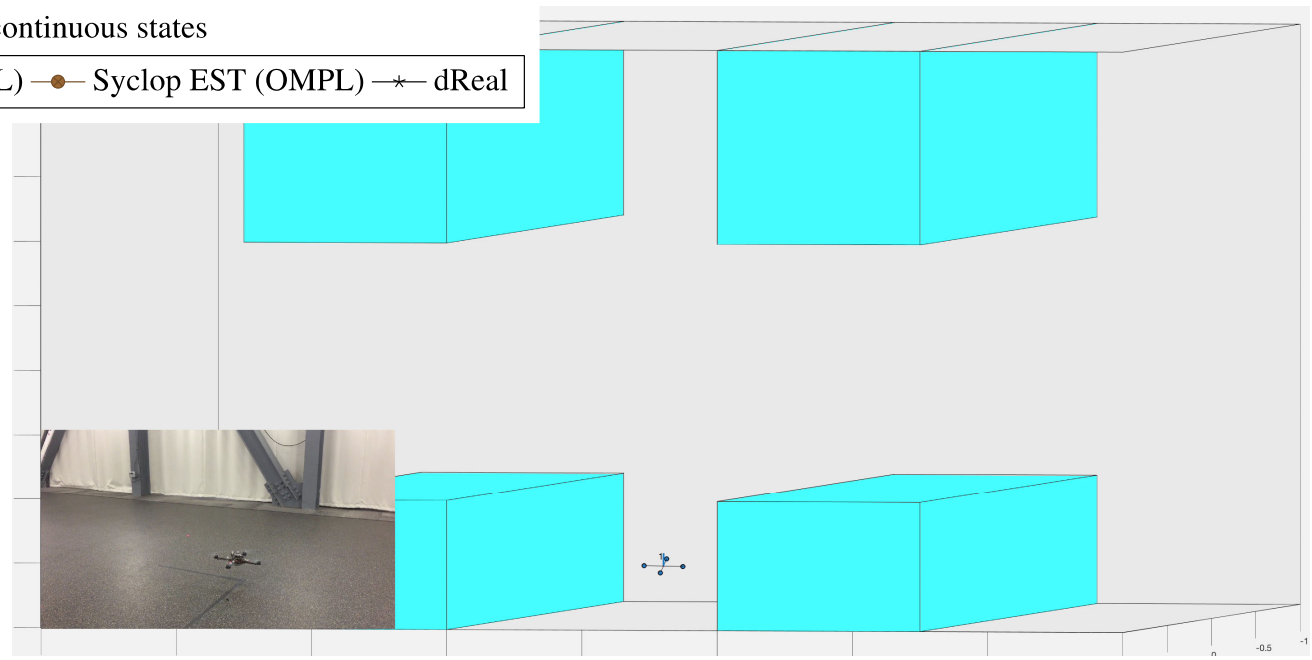


Result (2): LTL Motion Planning



Number of States	Number of Propositions	SMC-Based [s]	Synergistic LTL [s]
4	1	6.6614	2.4978
	2	16.0456	44.3815
	3	36.651	153.389
6	1	25.2670	13.003
	2	54.517	timeout
	3	73.0913	timeout
8	1	6.280	timeout
	2	26.385	timeout
	3	225.255	timeout

$$\phi ::= \Box\Diamond\pi_1 \wedge \Box\Diamond\pi_2 \wedge \Box\neg\pi_{obstacle}$$



Result (3): Multi-Robot, LTL

$\Box \quad \neg$obstacles

$\wedge \Diamond\Box$ at least one robot patrols the middle

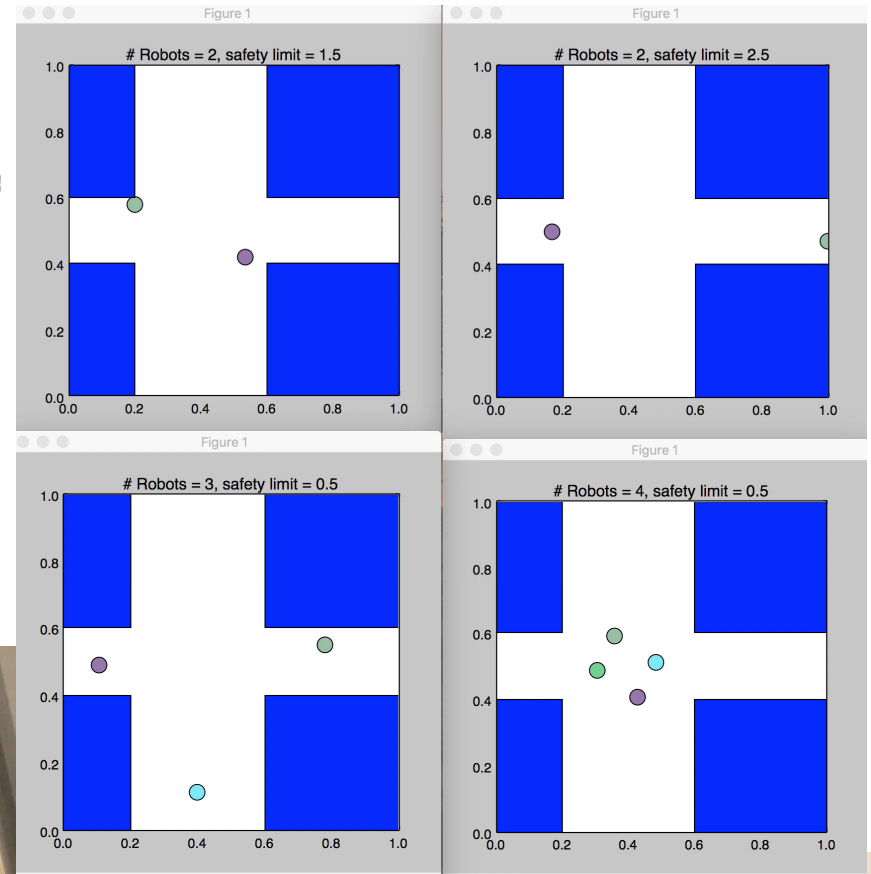
$\wedge \Box\Diamond$ 1st robot visits charging station #1

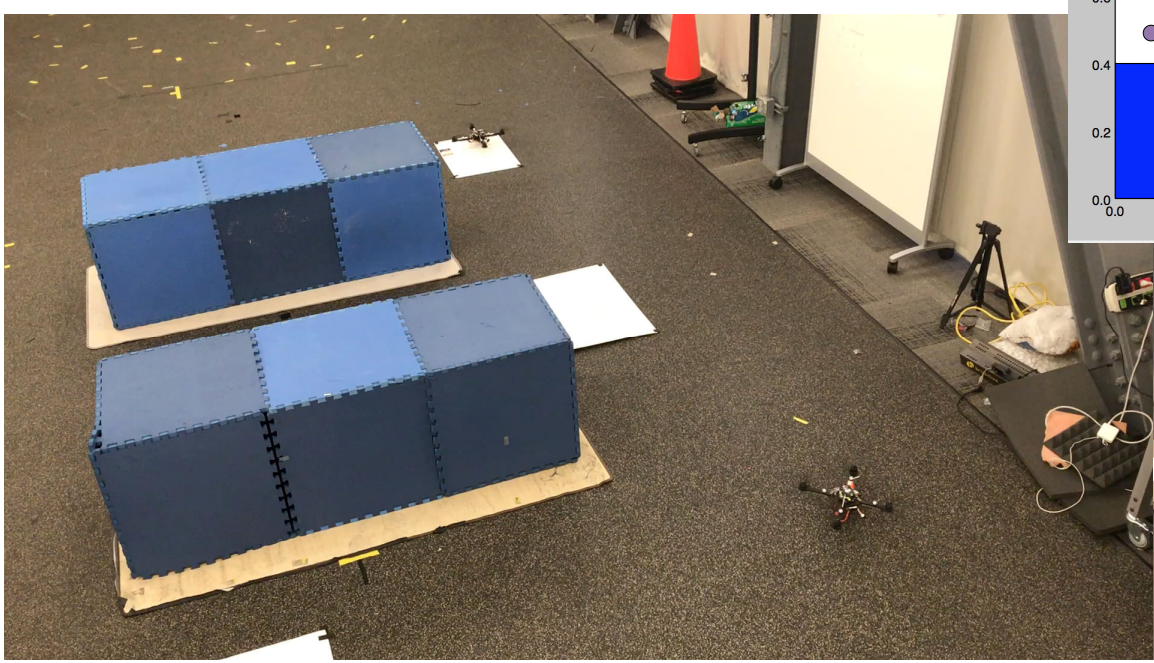
$\wedge \Box\Diamond$ 2nd robot visits charging station #2

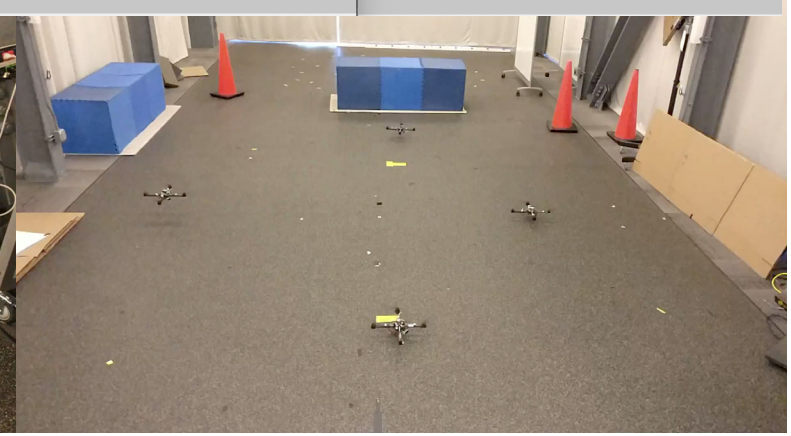
$\wedge \Box$ 1.5m seperation

$\wedge \Box$ velocity $\leq 2.5m/s$

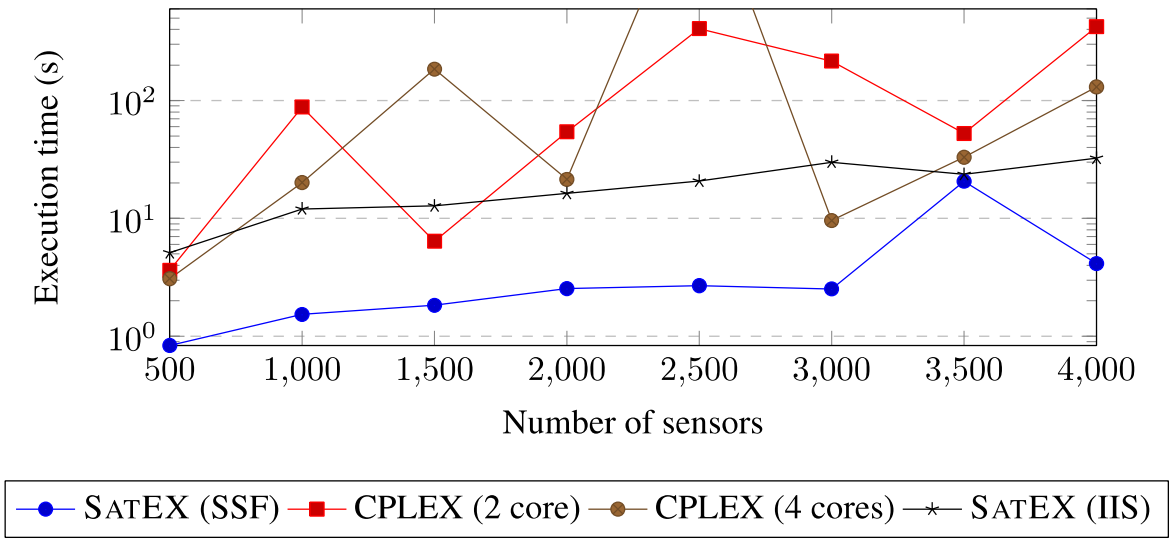
$\wedge \Box$ acceleration $\leq 1m/s^2$

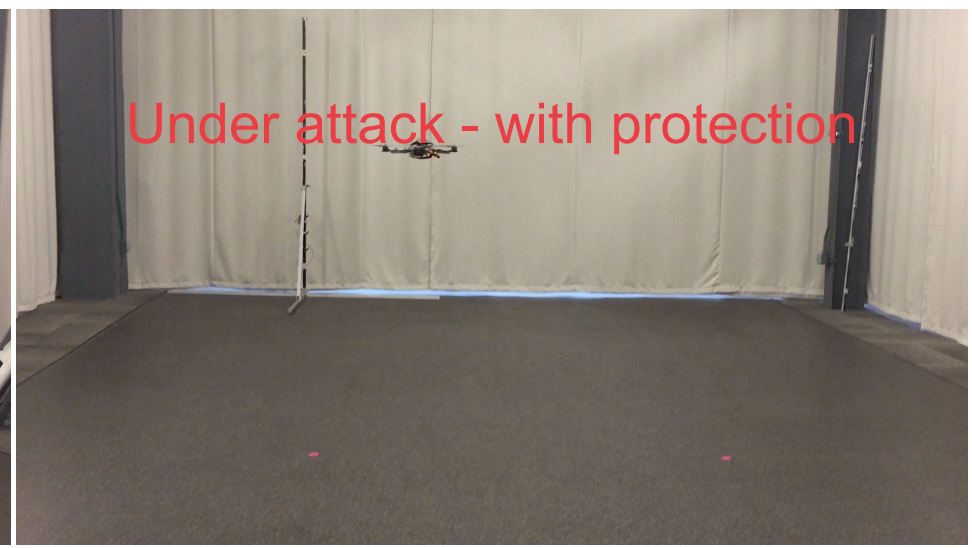






Result (4): Secure CPS

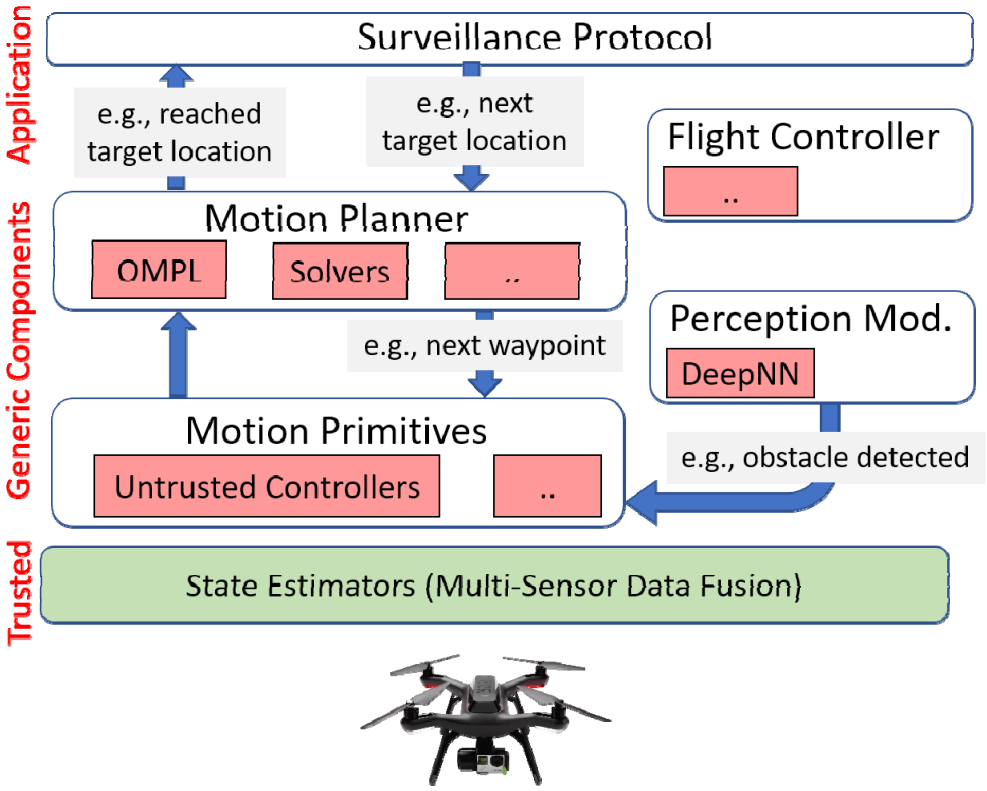




Challenges

- Scalable synthesis of motion plans blending high-level (discrete) and low-level (continuous) control
- Implementation in software on top of networked robotics platforms
- Dealing with untrusted components, uncertainty in environment, and ML-based perception

Robotics Software Stack



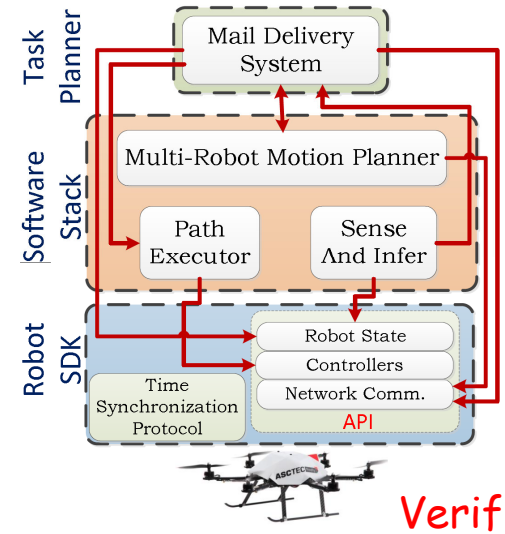
Challenges

- High-Level Programming of Event-Driven Software
- Reliable Networked Behavior
- Verification in the Presence of Black/Gray-box Components
- Guaranteeing Safety via Run-time assurance

Drona: A Software Framework for Distributed Mobile Robotics

Contributions:

1. *Provably correct* decentralized motion planner that can perform on-the-fly collision-free planning *robust* against network clock-synchronization errors.
2. *A verified* software stack for distributed mobile robotics.
3. High-level ModP language, back-end model checker, and code generation to ROS/other SDKs.

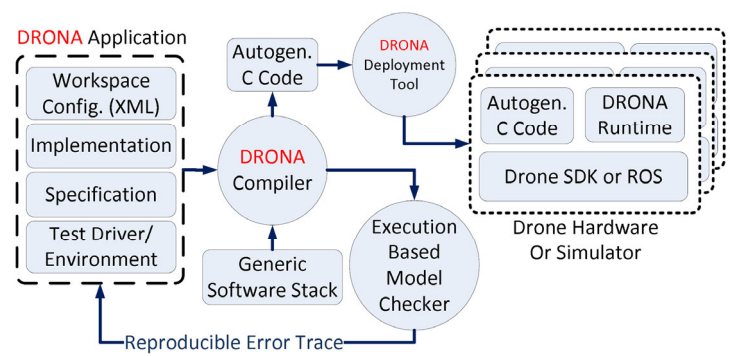


Verified software stack

Achievements:

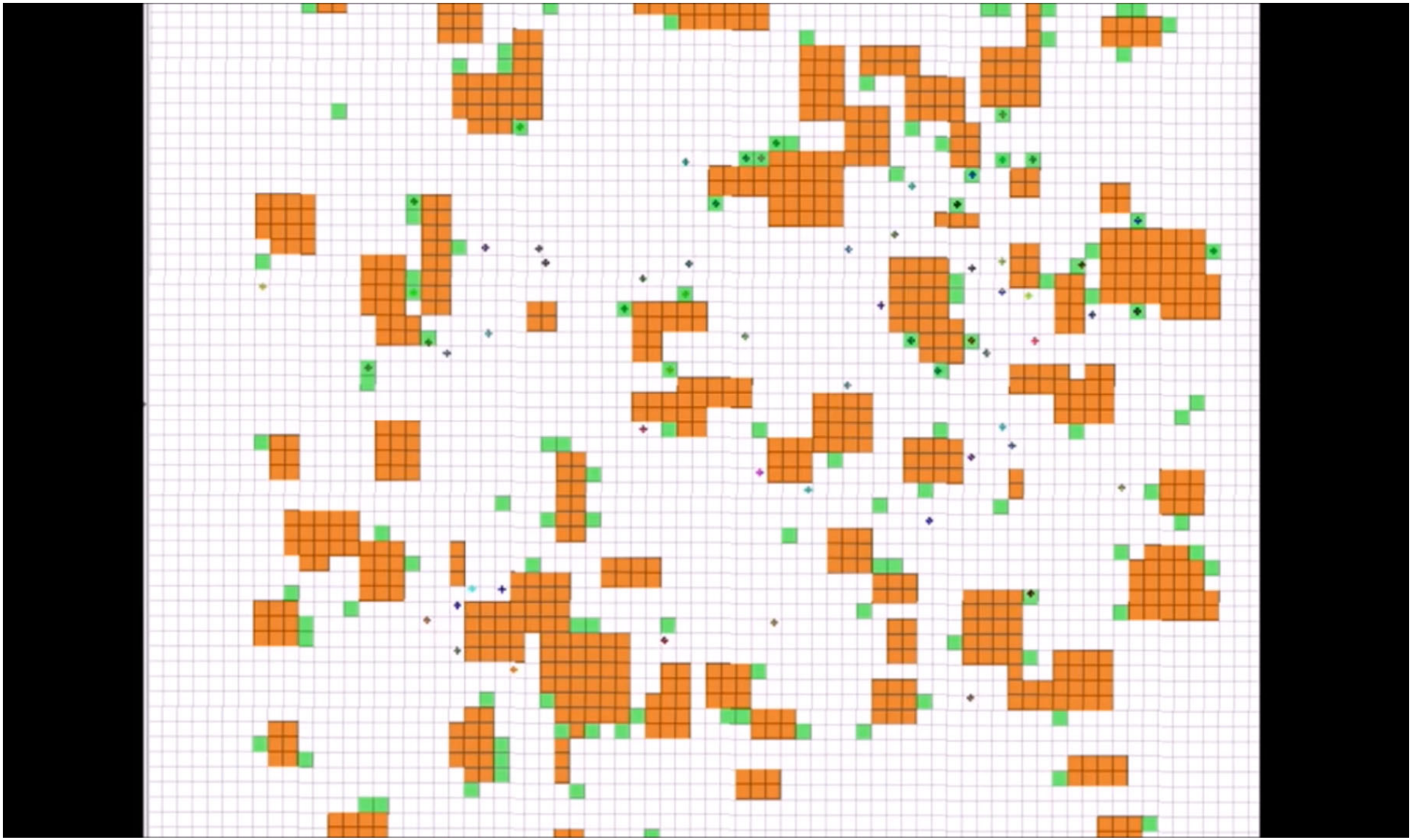
✓ Programmed multiple drone platforms (Astec-Firefly, CrazyFlie, ROS, etc.).

✓ Framework efficiently scales for large workspaces with 128 robots.



"DRONA: A Framework for Safe Distributed Mobile Robotics", A. Desai et al., ICCPS 2017.

Drona scalability experiment: 64 drones in simulation



Challenges

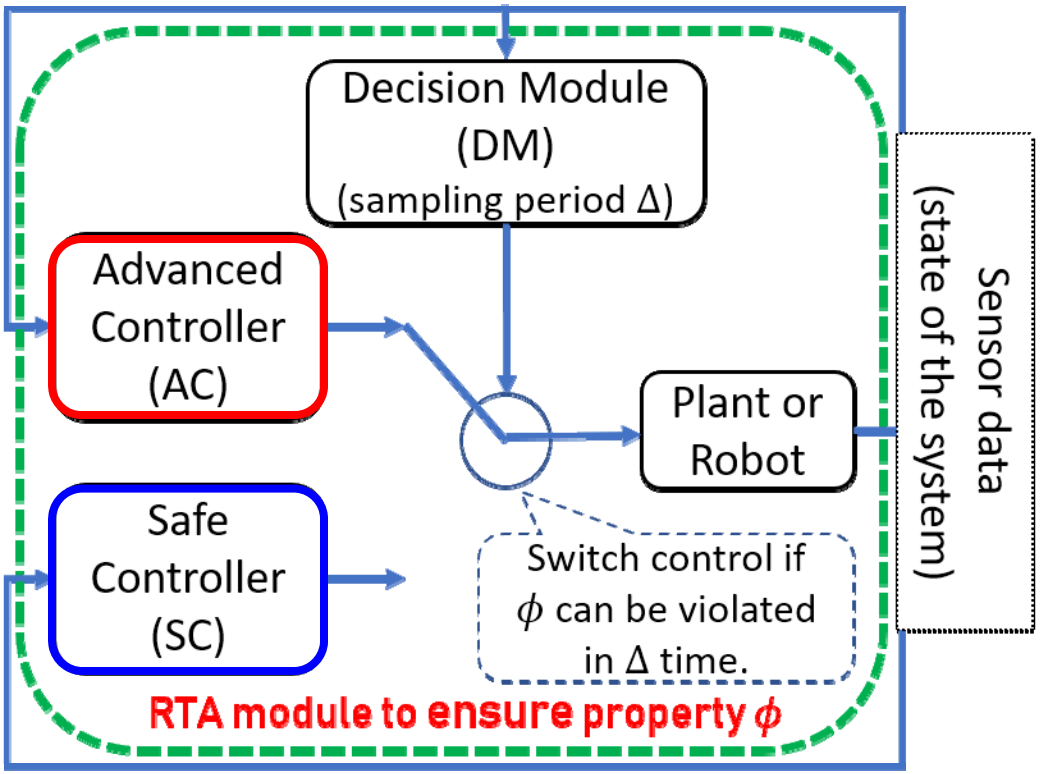
- Scalable synthesis of motion plans blending high-level (discrete) and low-level (continuous) control

- Implementation in software on top of networked robotics platforms

- Dealing with untrusted components, uncertainty in environment, and ML-based perception

Simplex Architecture for Run-Time Assurance of Periodic Real-Time Systems

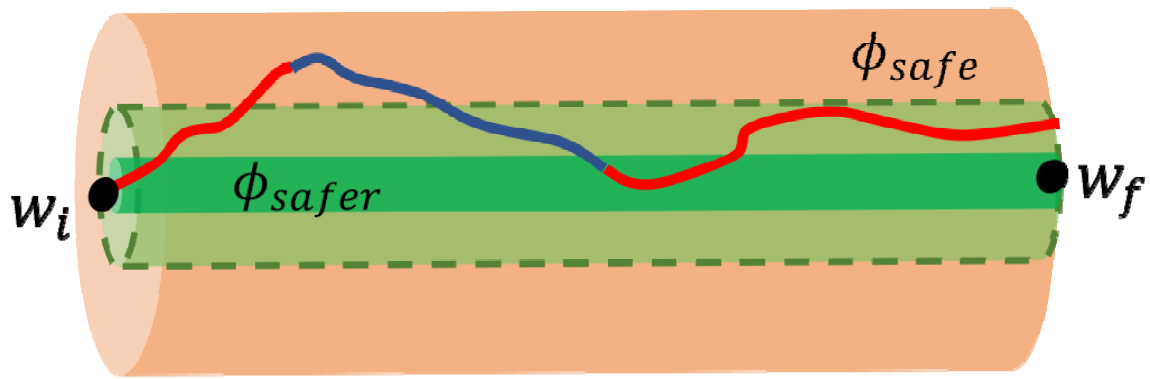
[Lui Sha, RTSS'98]



Already used in fault-tolerant avionics systems

Our Approach: Controlled Reachable Sets for Reach-Avoid Objectives

Period Δ



<u>Theorem:</u> The following is an inductive invariant:

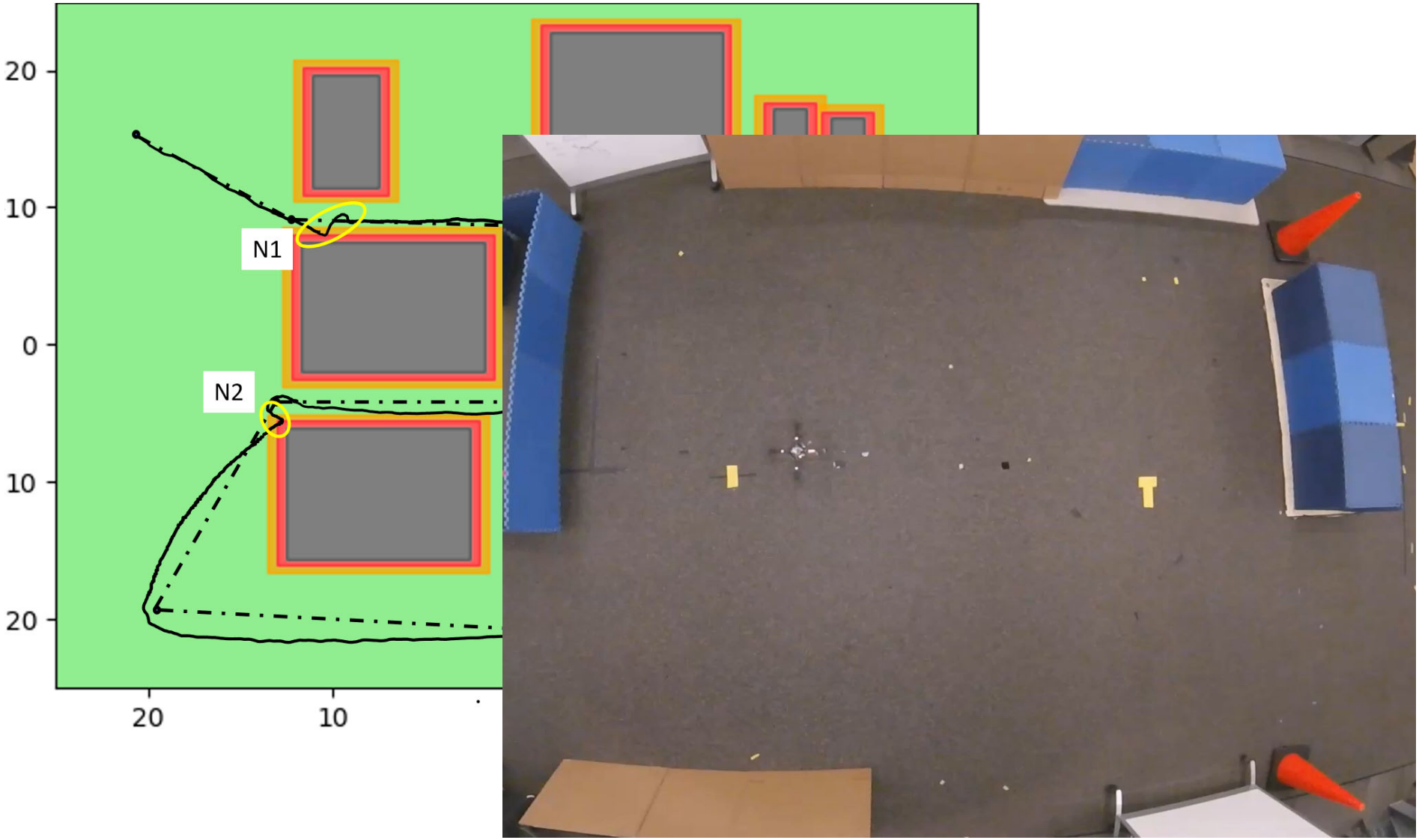
$$\text{Mode} = \text{SC} \wedge s \in \varphi_{safe}$$

$$\vee$$

$$\text{Mode} = \text{AC} \wedge \text{Reach}(s,*,\Delta) \in \varphi_{safe}$$

[Desai et al., "SOTER: Programming Safe Robotics Systems using Runtime Assurance", 2018.

Results: Drone Navigating in an Unknown Workspace (reach-avoid problems)



Summary

- Scalable synthesis of motion plans blending high-level (discrete) and low-level (continuous) control → SMC

- Implementation in software on top of networked robotics platforms → Drona

- Dealing with untrusted components, uncertainty in environment, and ML-based perception → Run-time assurance + *more… See tomorrow's talk*!

Summary of the Lecture