Generating Longest Counterexample: At the Cross-roads of MILP and SMT

Manish Goyal[1], David Bergman[2], and Parasara **Sridhar** Duggirala[1]

[1] Department of Computer Science, University of North Carolina at Chapel Hill

[2]School of Business, University of Connecticut, Storrs

American Control Conference (ACC), July 2020

Outline

- Background

  Verification, falsification and control

- Introduction

  Longest counterexample

- Preliminaries

  Simulation-equivalent analysis, reachable set computation

- Methodology

  Constraint Propagation

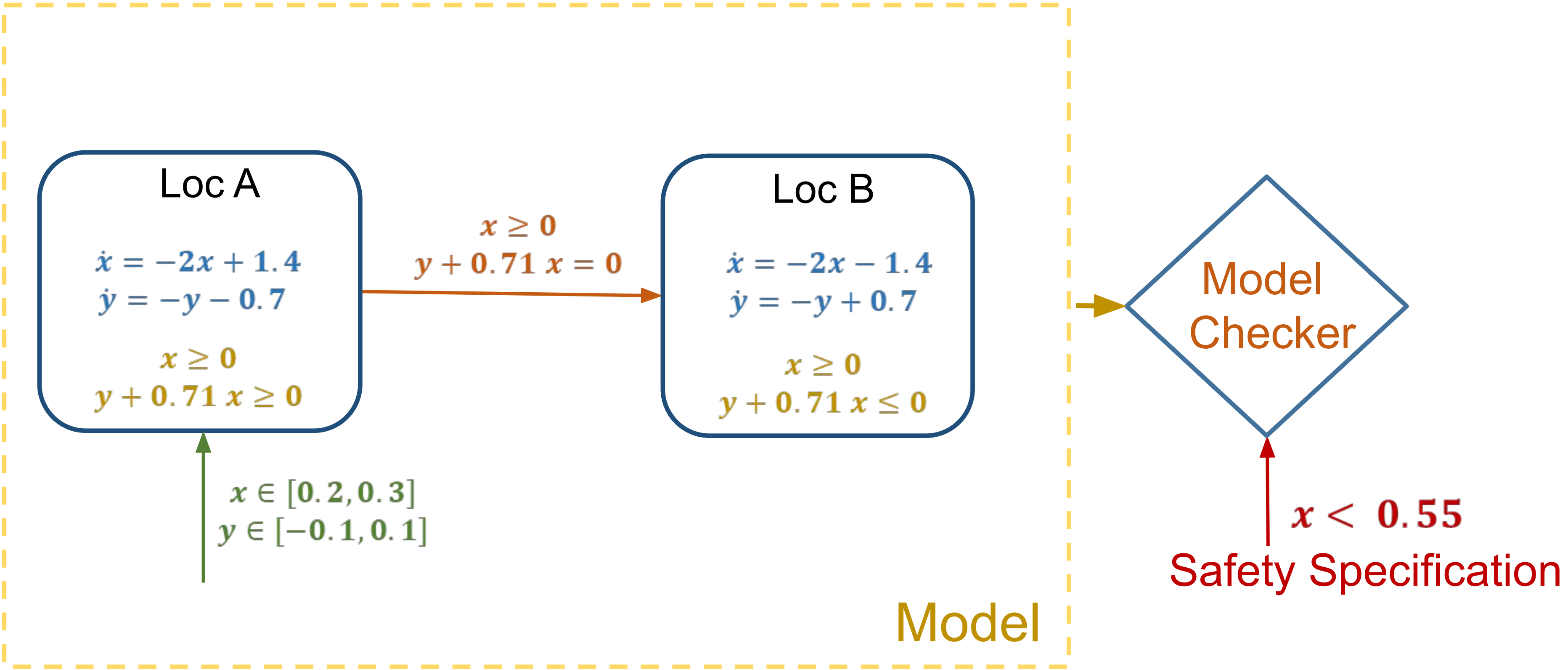
- Frameworks

  MILP-based and SMT-based

- Re [illegible] work

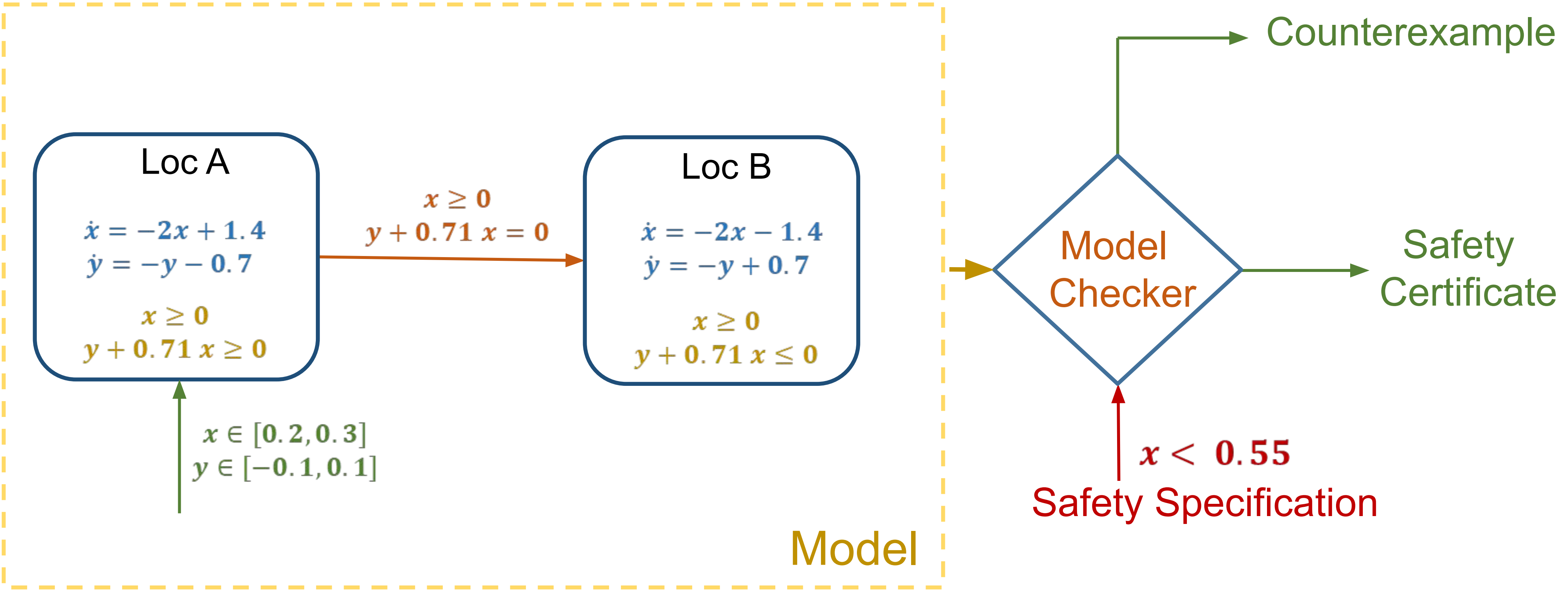
Verification

Analogous to Reachability

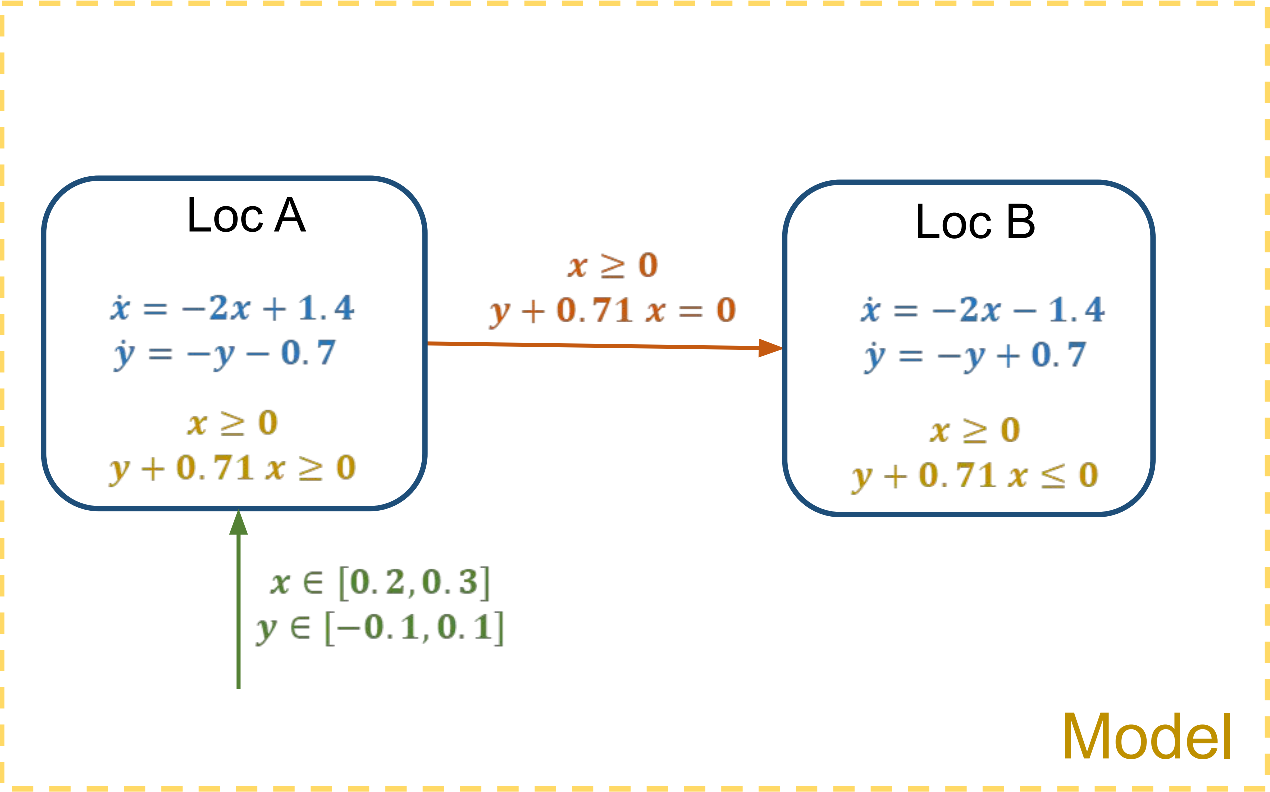


Verification

Analogous to Reachability



Verification

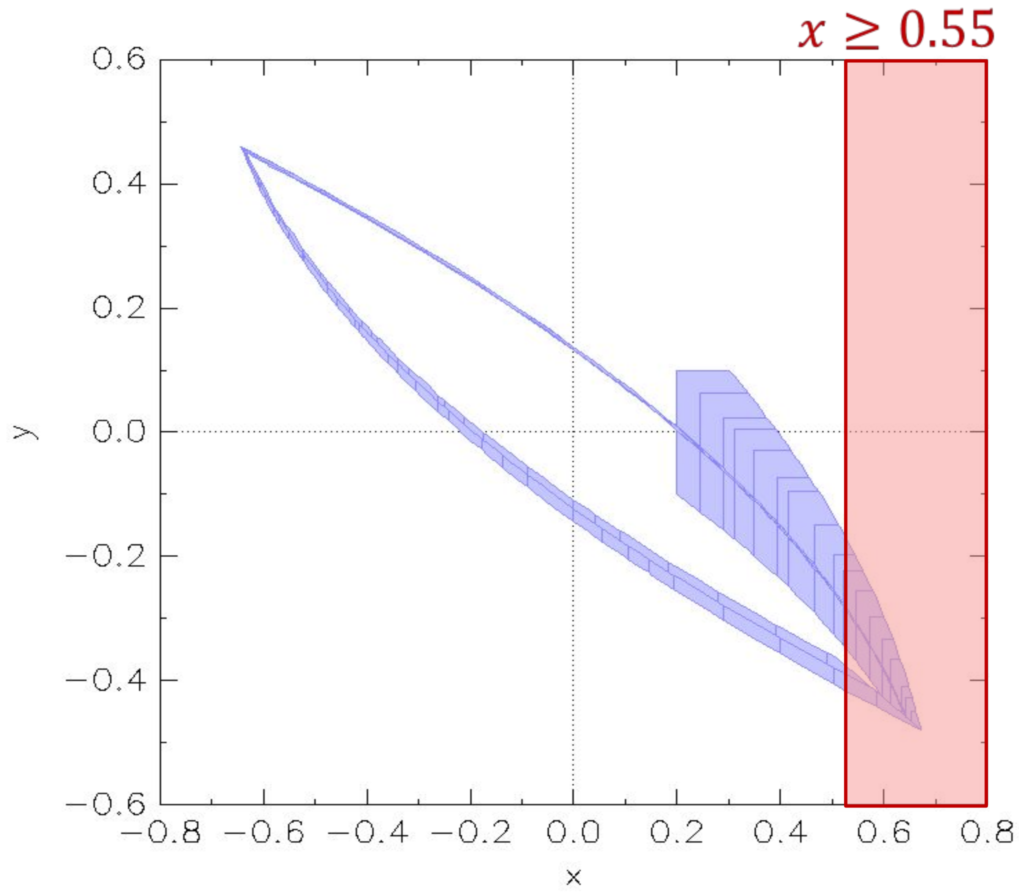


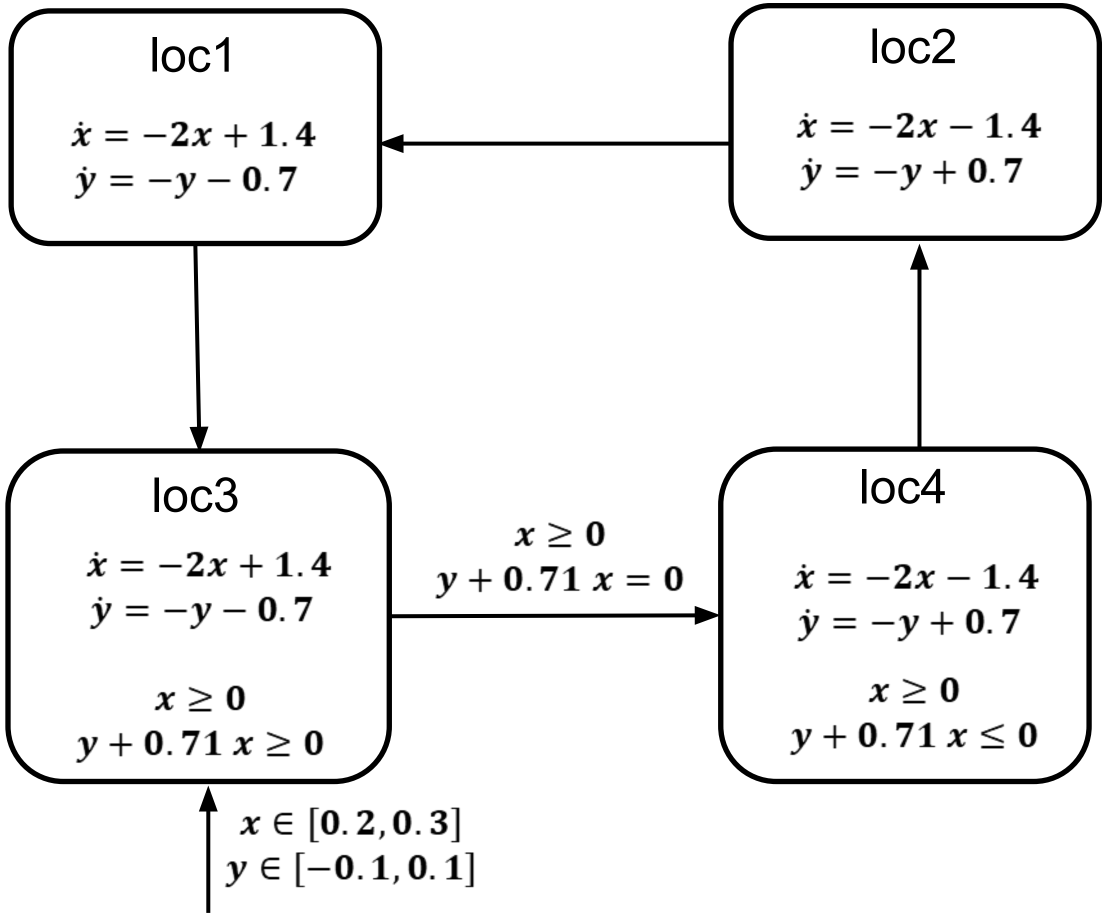
Hybrid Automaton

- Locations
- Flow
- Invariant
- Transitions
- Guard Condition
- Initial Condition

Verification

SpaceEx*: Filtered Oscillator**



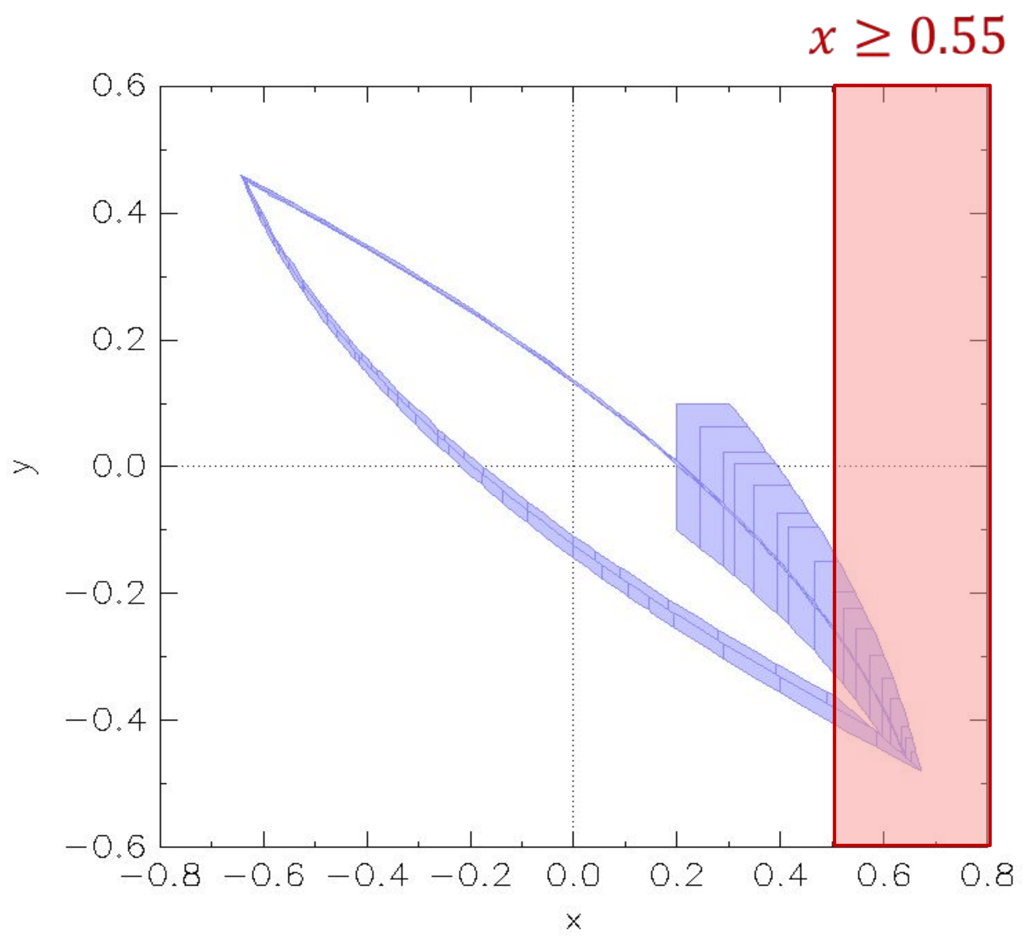


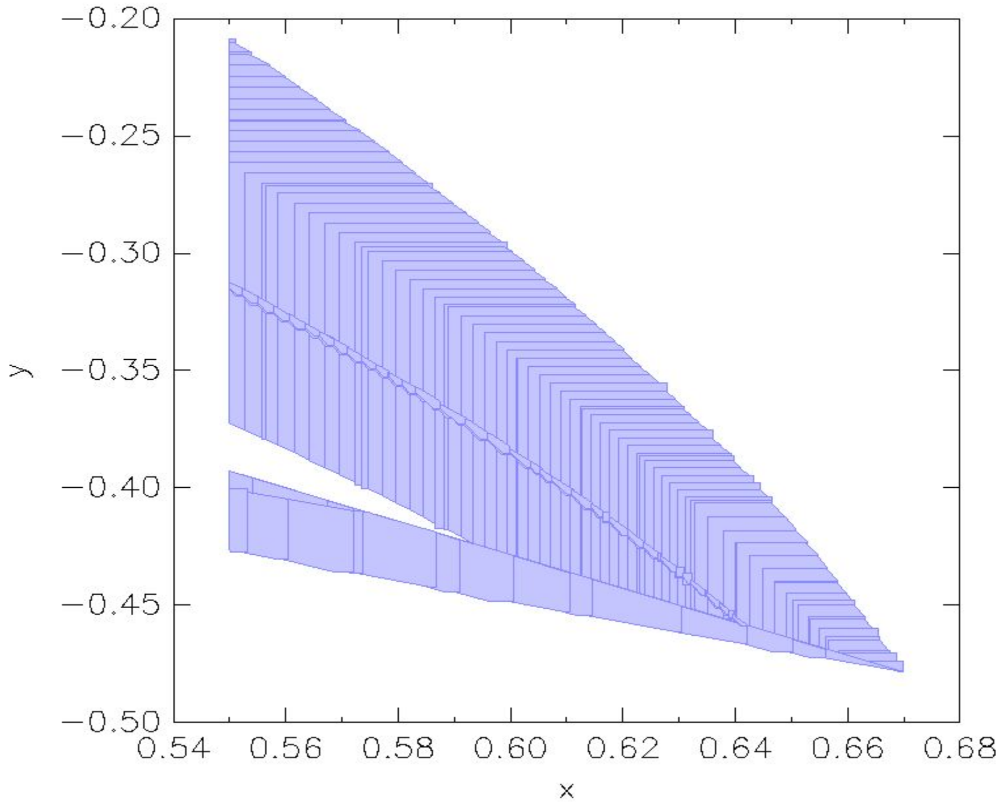
*G. Frehse et al, Spaceex: Scalable verification of hybrid systems. CAV 2011.

**https://ths.rwth-aachen.de/research/projects/hypro/filtered-oscillator/

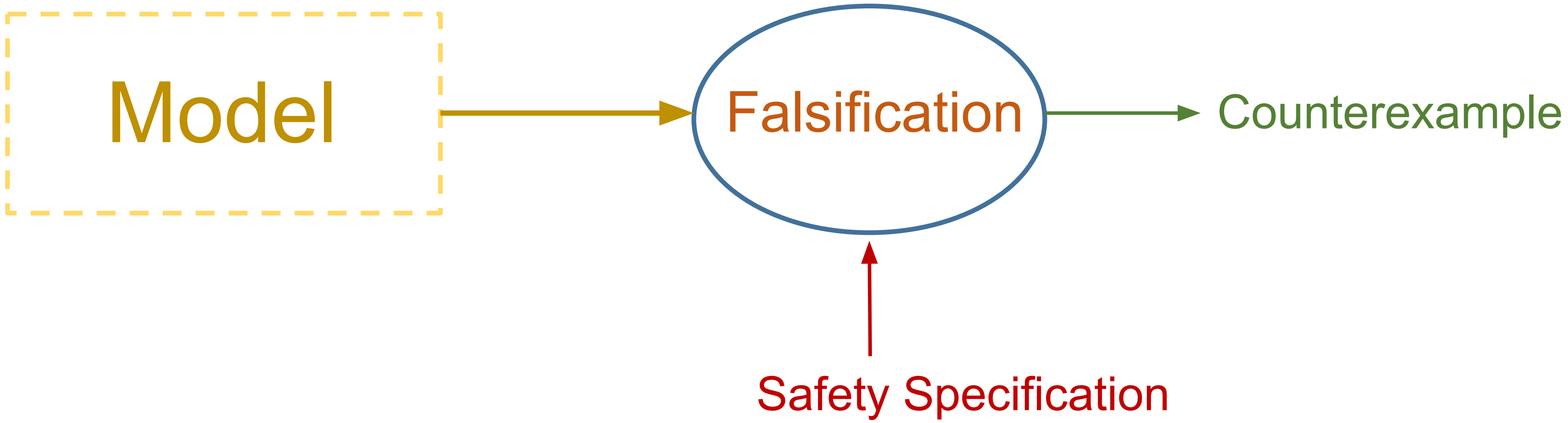
Verification

SpaceEx: Filtered Oscillator



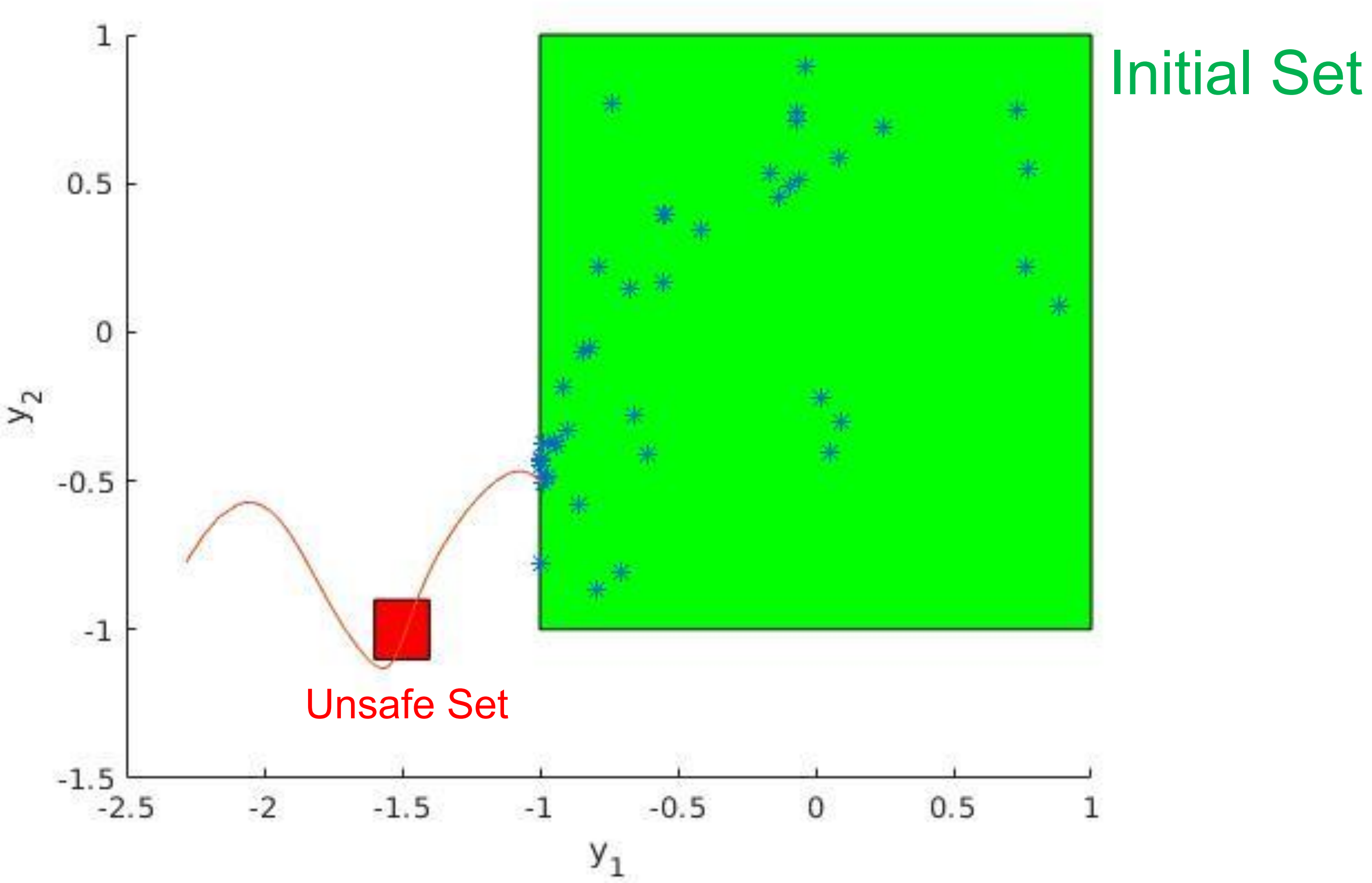


Falsification



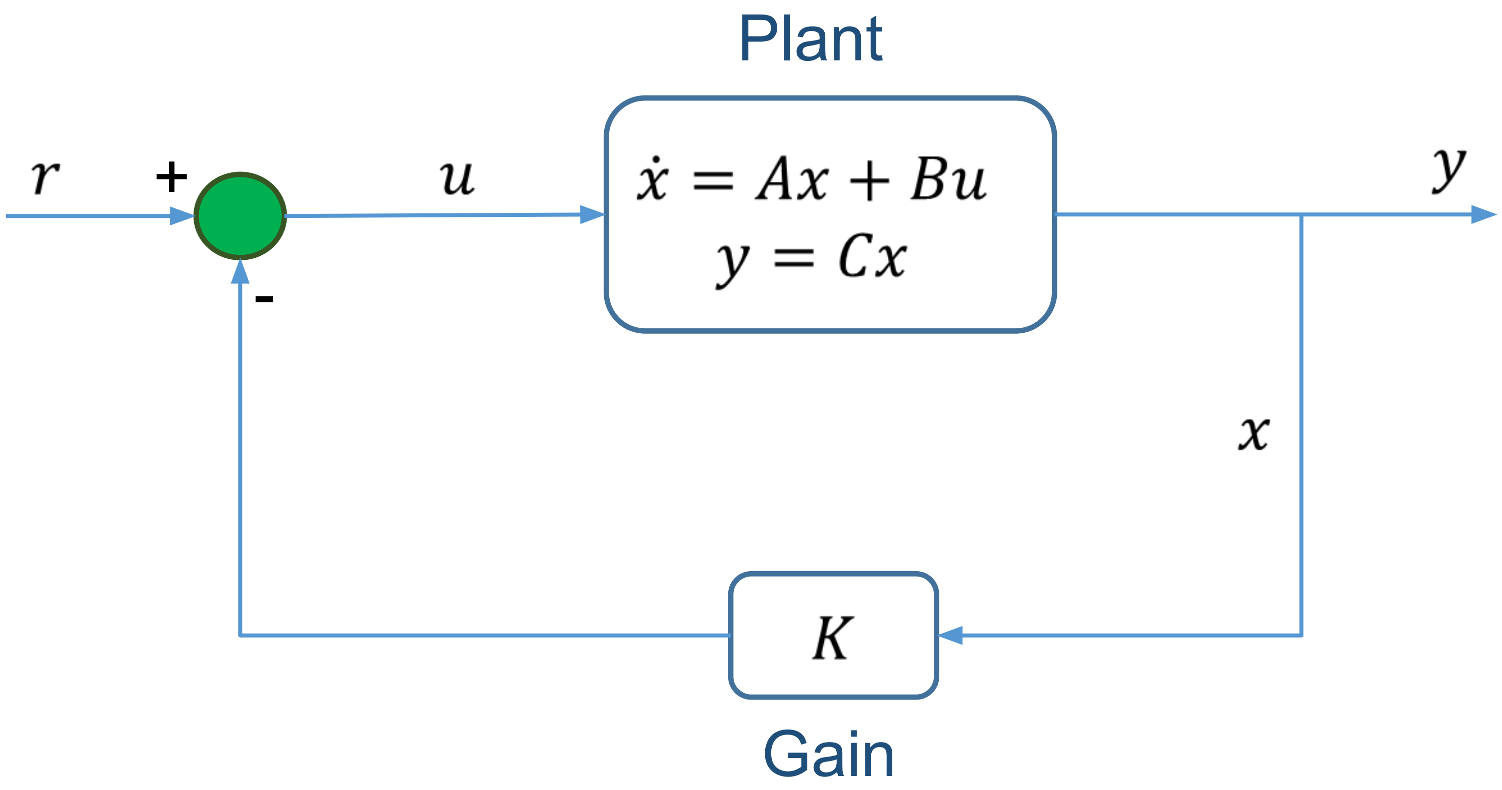
Falsification

S-TaLiRo*

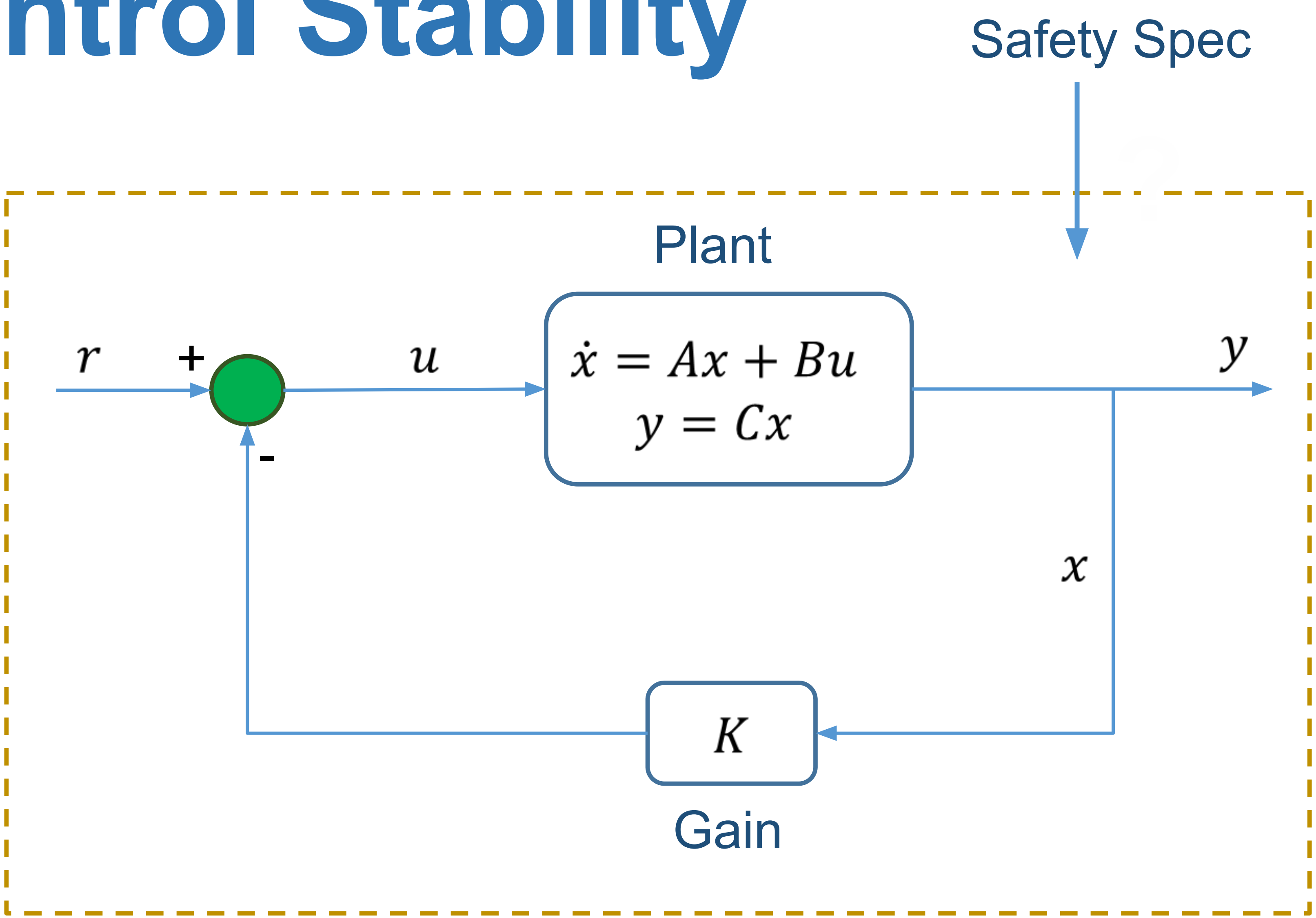


*Y. Annapureddy et al, S-TaLiRo: A Tool for Temporal Logic Falsification for Hybrid Systems. TACAS 2011.

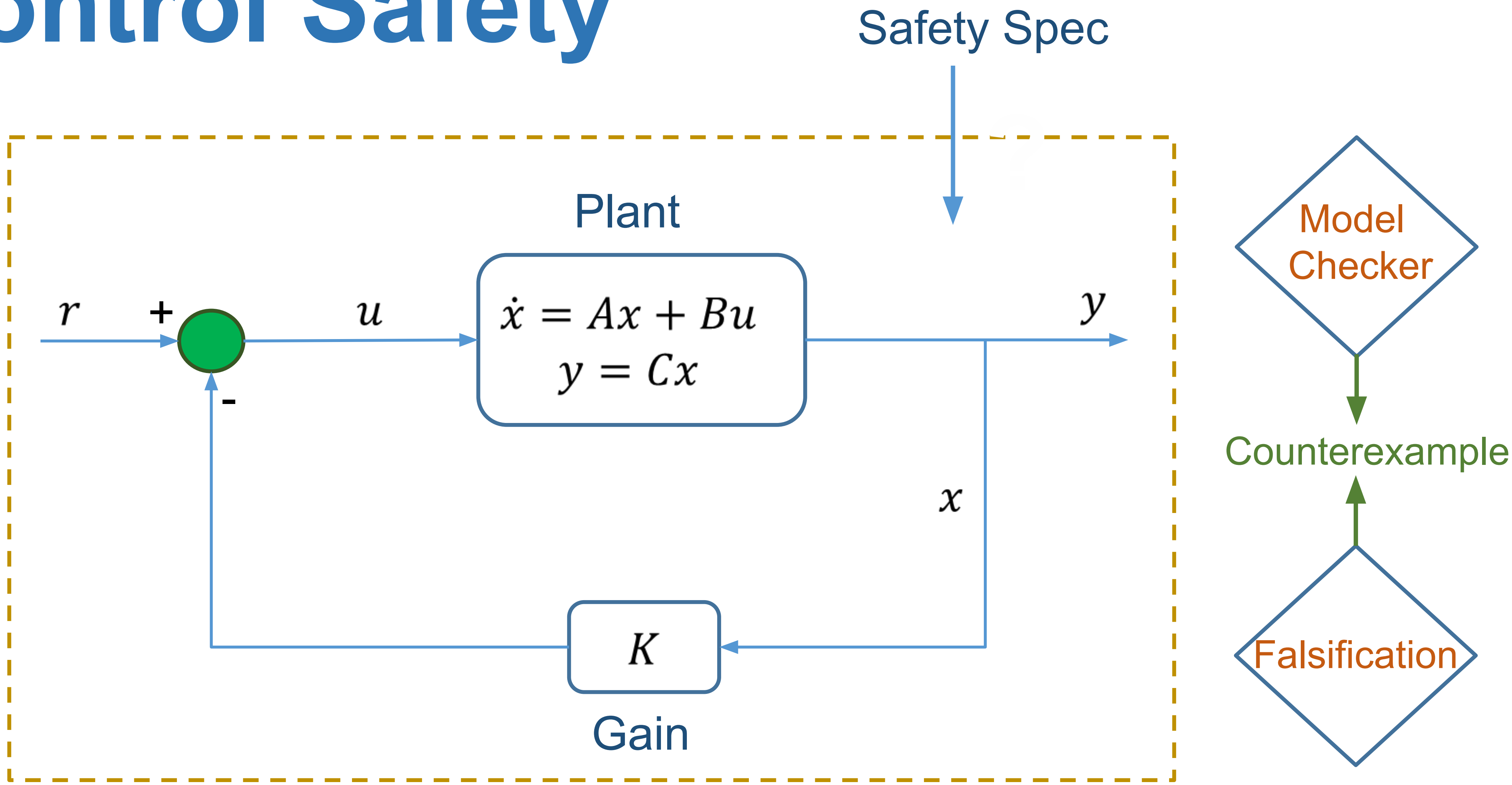
Control Stability



Control Stability



Control Safety



Motivation

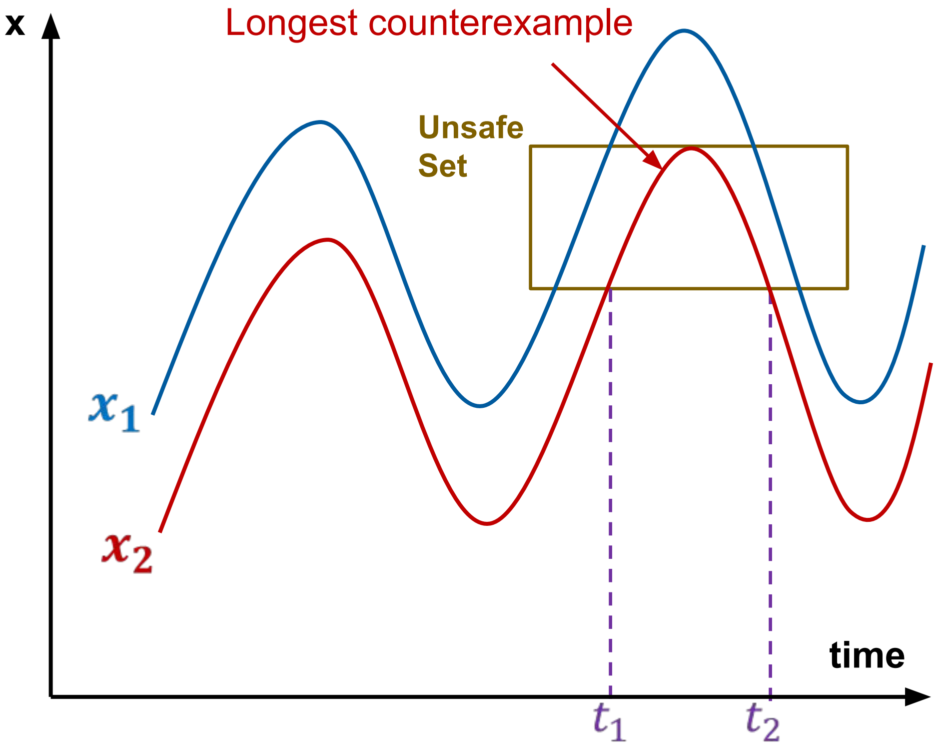

Server: 13 FPS
Client: 25 FPS
Vehicle: Lincoln Mkz2017
Map: Town01
Simulation time: 0:00:22
Speed: 21 km/h
Heading: -0° N
Location: (112.6, 133.0)
Height: -0 m
Throttle:
Steer:
Brake:
Reverse:
Hand brake:
Collision:
Number of vehicles: 2
Nearby vehicles:
20m Nissan Patrol
CARLA


Server: 13 FPS
Client: 25 FPS
Vehicle: Lincoln Mkz2017
Map: Town01
Simulation time: 0:00:14
Speed: 9 km/h
Heading: -0° N
Location: (107.6, 133.0)
Height: 0 m
Throttle:
Steer:
Brake:
Reverse:
Hand brake:
Collision:
Number of vehicles: 2
Nearby vehicles:
25m Nissan Patrol
CARLA

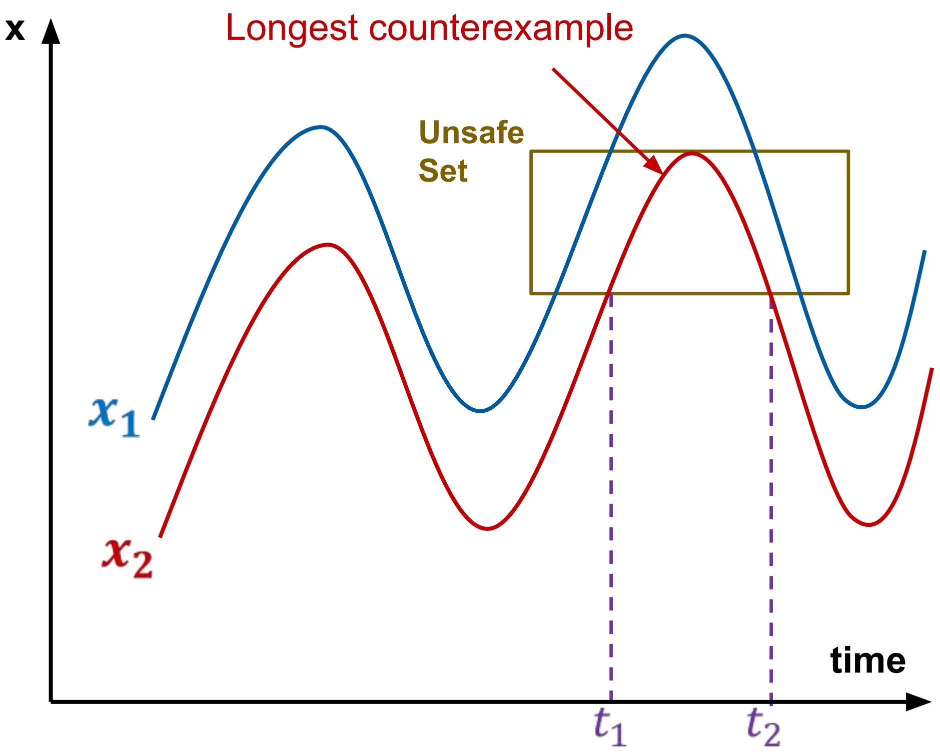
Outline

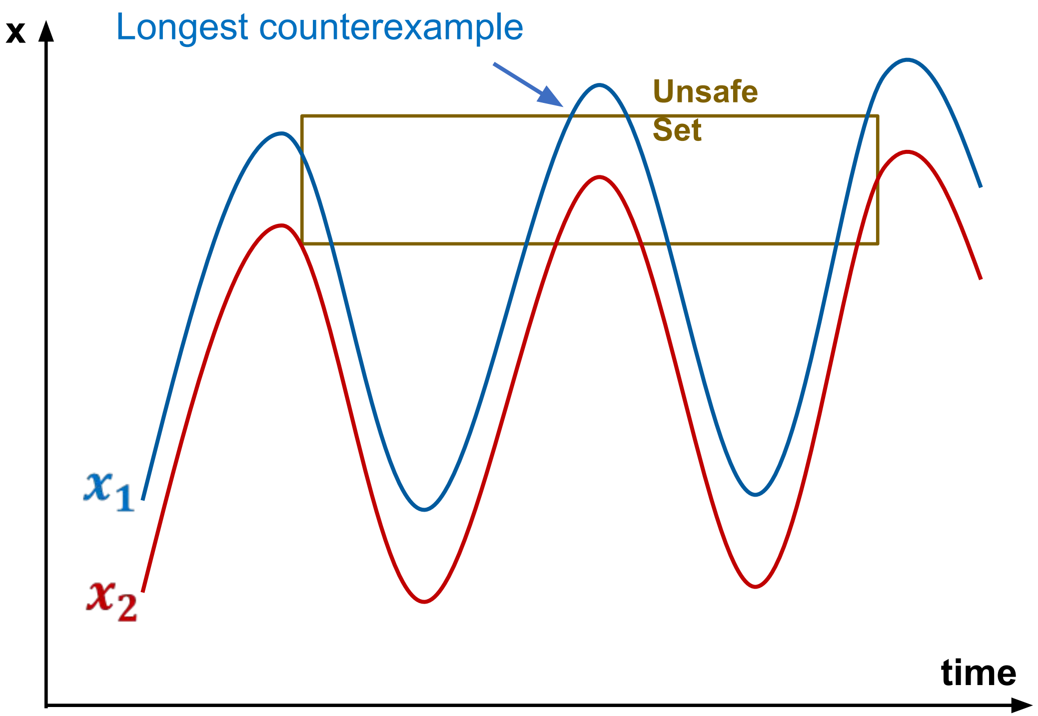
- ✔ Background
- Introduction
- Preliminaries
- Methodology
- Frameworks
- Results

Introduction



Introduction



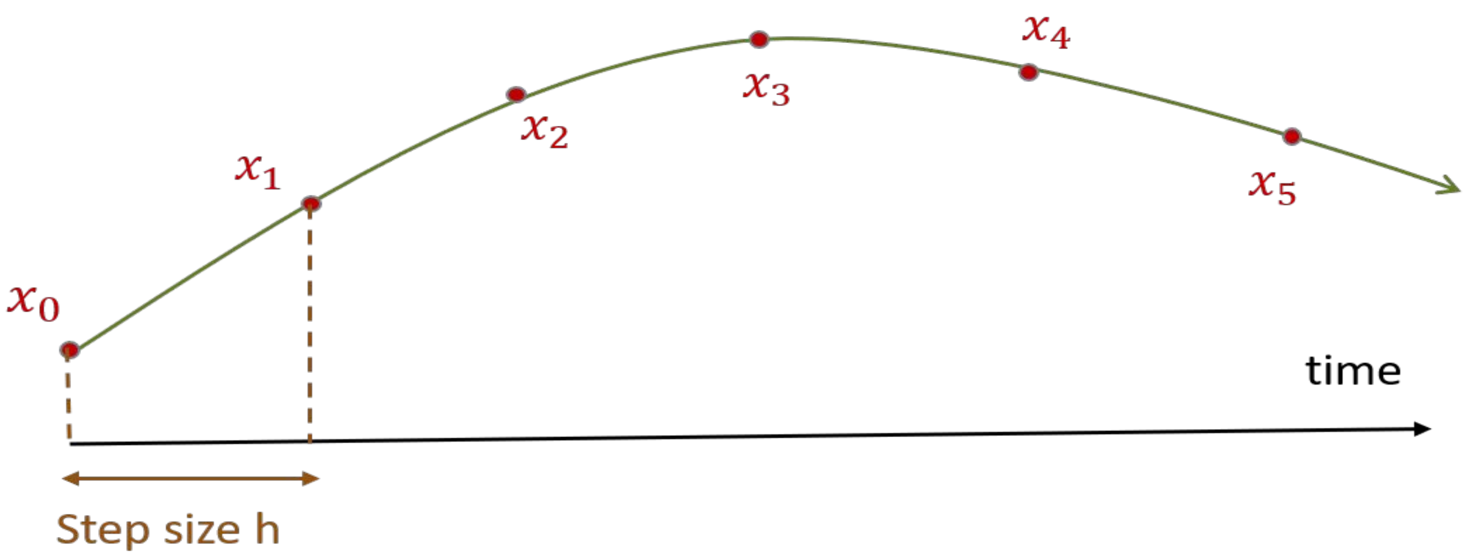


Outline

- ✔ Background
- ✔ Introduction
- Preliminaries
- Methodology
- Frameworks
- Results

Simulation-equivalent Analysis

For a dynamical system H with affine dynamics $\dot{x} = Ax + B$, the simulation starting from a state x_0 is computed as a sequence $\tau_H(x_0, h)$ of states at discrete time steps with step size h.

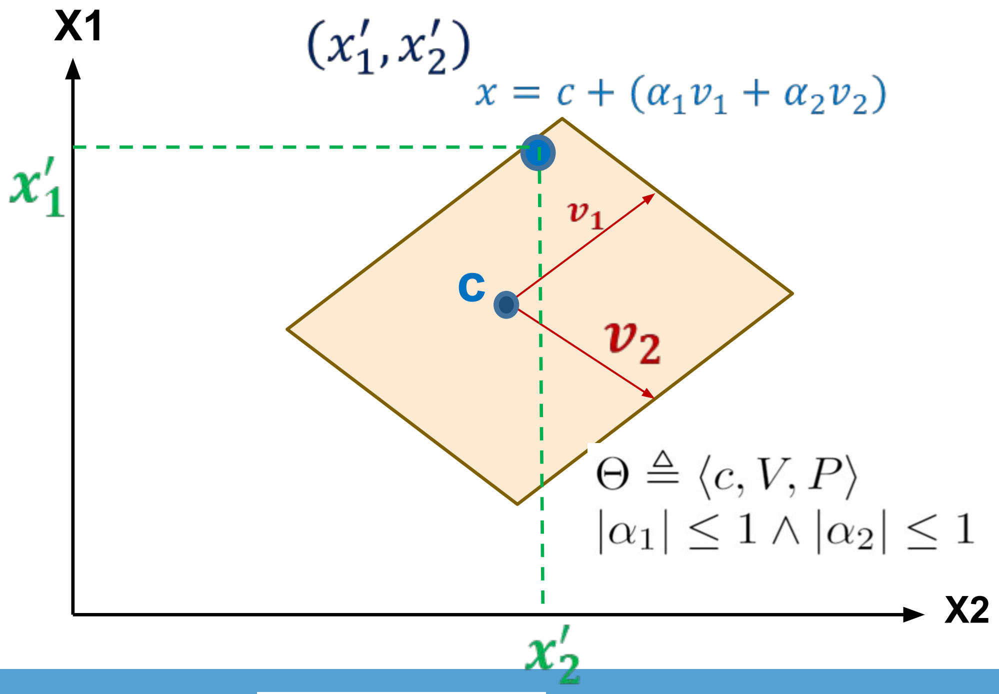


In the sequence $\tau_H(x_0, h)$ = $x_0, x_1, x_2, \ldots,$ each pair (x_i, x_{i+1}) corresponds to a continuous trajectory starting at x_i and reaching x_{i+1} after h time units.

Star Representation

A *generalized star* Θ is a tuple $\langle c, V, P\rangle$ where $c \in \mathbb{R}^n$ is called the *center*, $V = \{v_1, v_2, \ldots, v_m\}$ is a set of m $(\leq n)$ vectors in $\mathbb{R}^n$ called the *basis vectors*, and $P : \mathbb{R}^n \to \{\top, \bot\}$ is a predicate, defined as

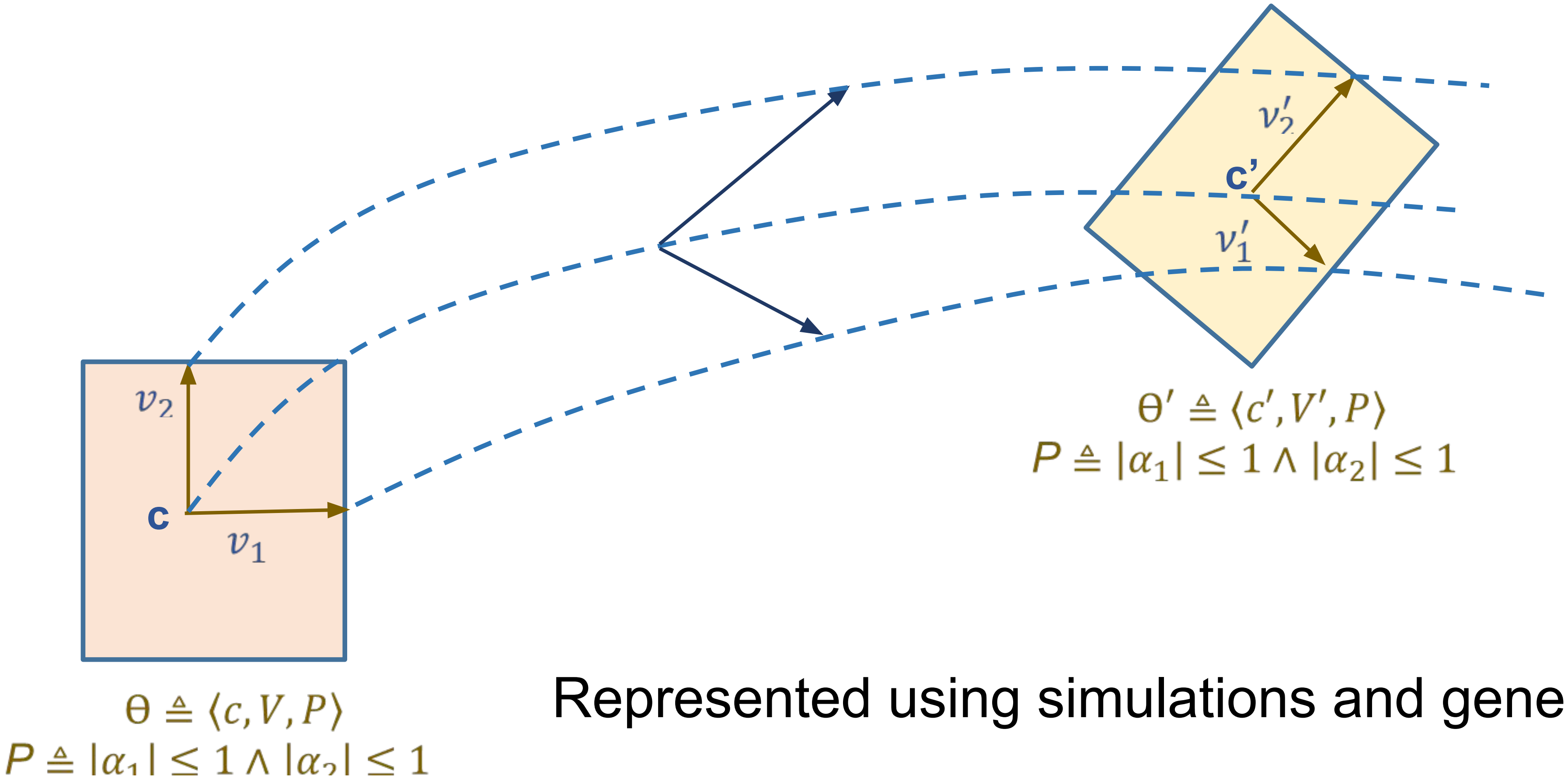
$$[\![\Theta]\!] = \{x \mid \exists \bar{\alpha} = [\alpha_1, \ldots, \alpha_m]^T \text{ such that } x = c + \Sigma_{i=1}^{n} \alpha_i v_i \text{ and } P(\bar{\alpha}) = \top\}$$



Variables

- Orthonormal: x_1' and x_2'
- Basis: α_1 and α_2

Reachable Set Computation

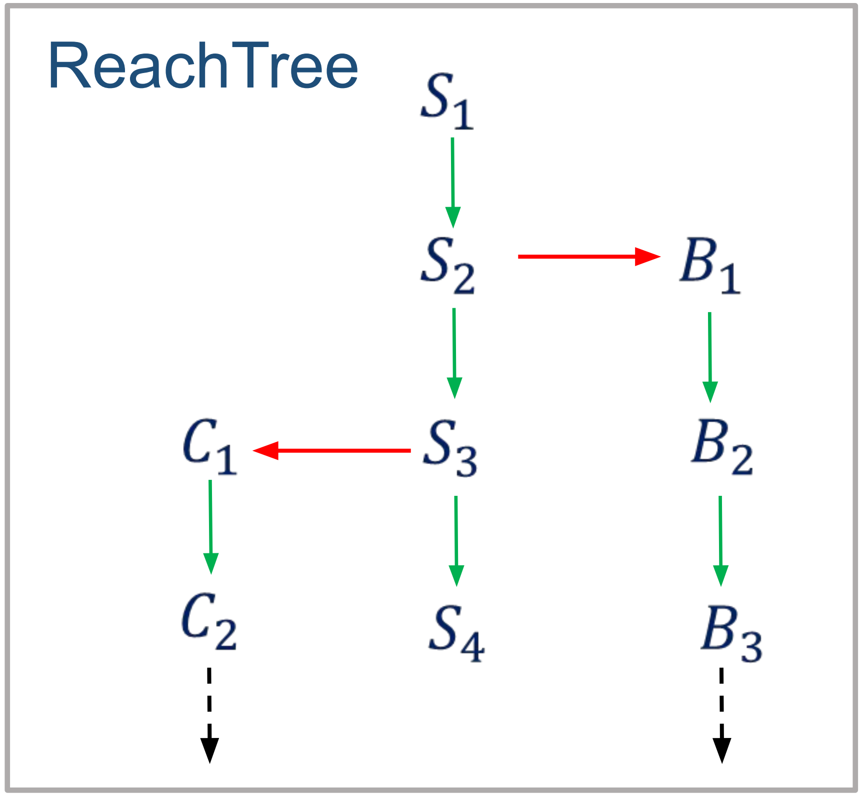


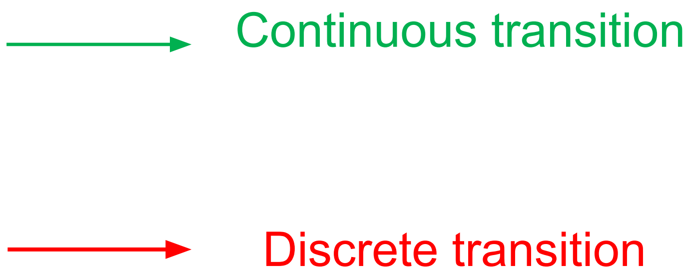
Represented using simulations and generalized star

The predicate P remains the same

Simulation-equivalent Analysis

Linear Hybrid Systems

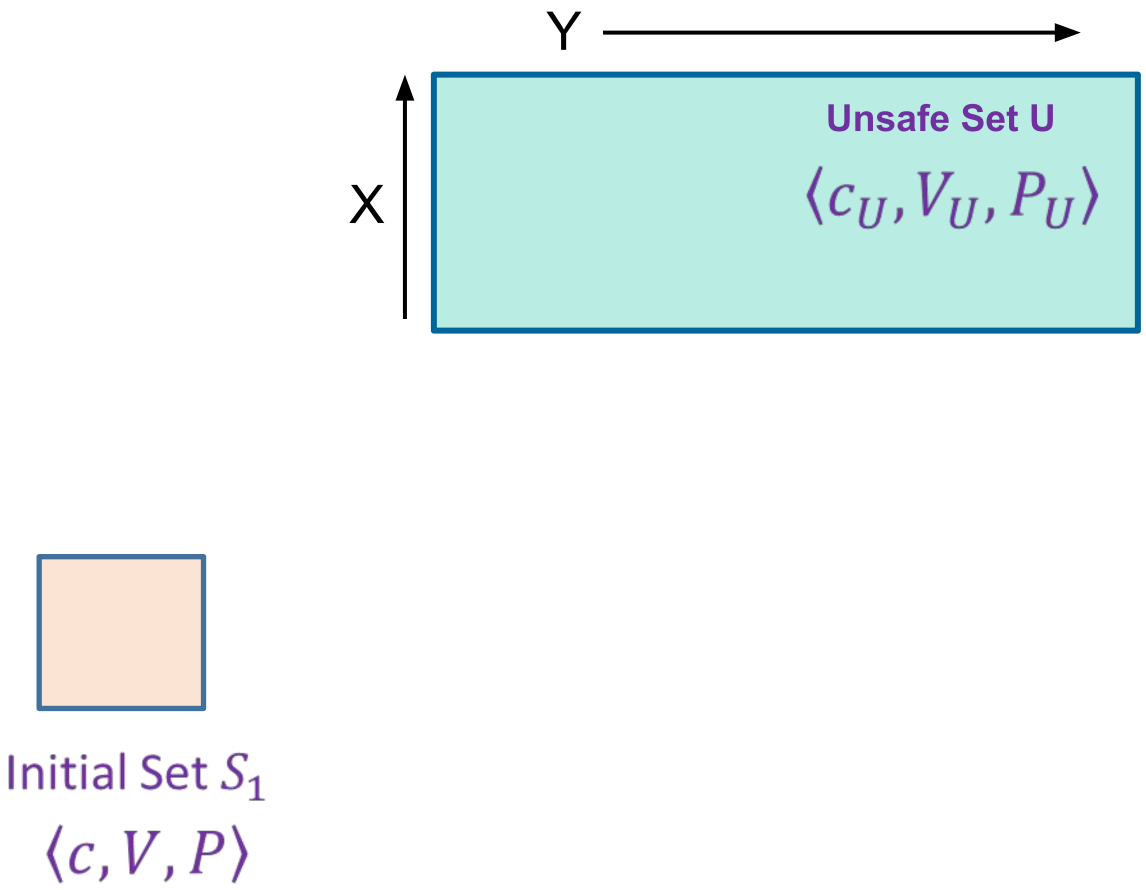




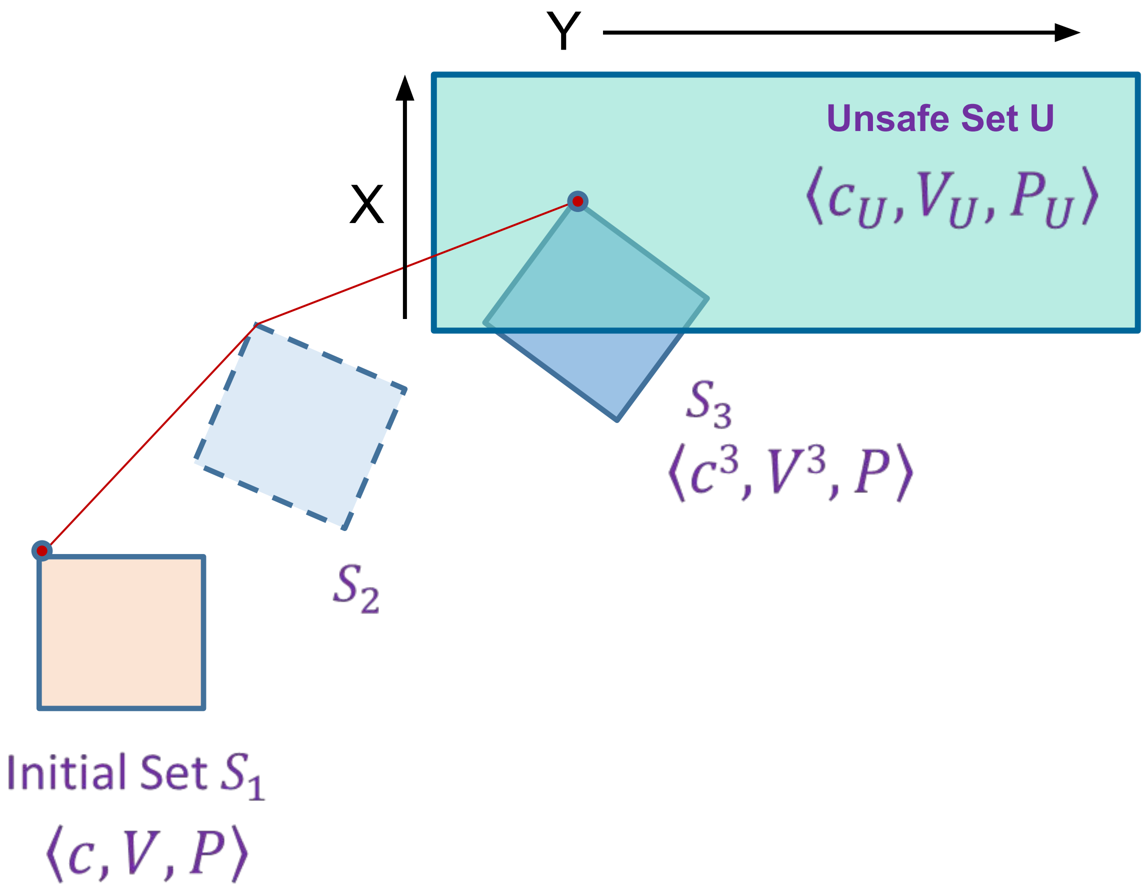
Outline

- ✔ Background
- ✔ Introduction
- ✔ Preliminaries
- Methodology
- Frameworks
- Results

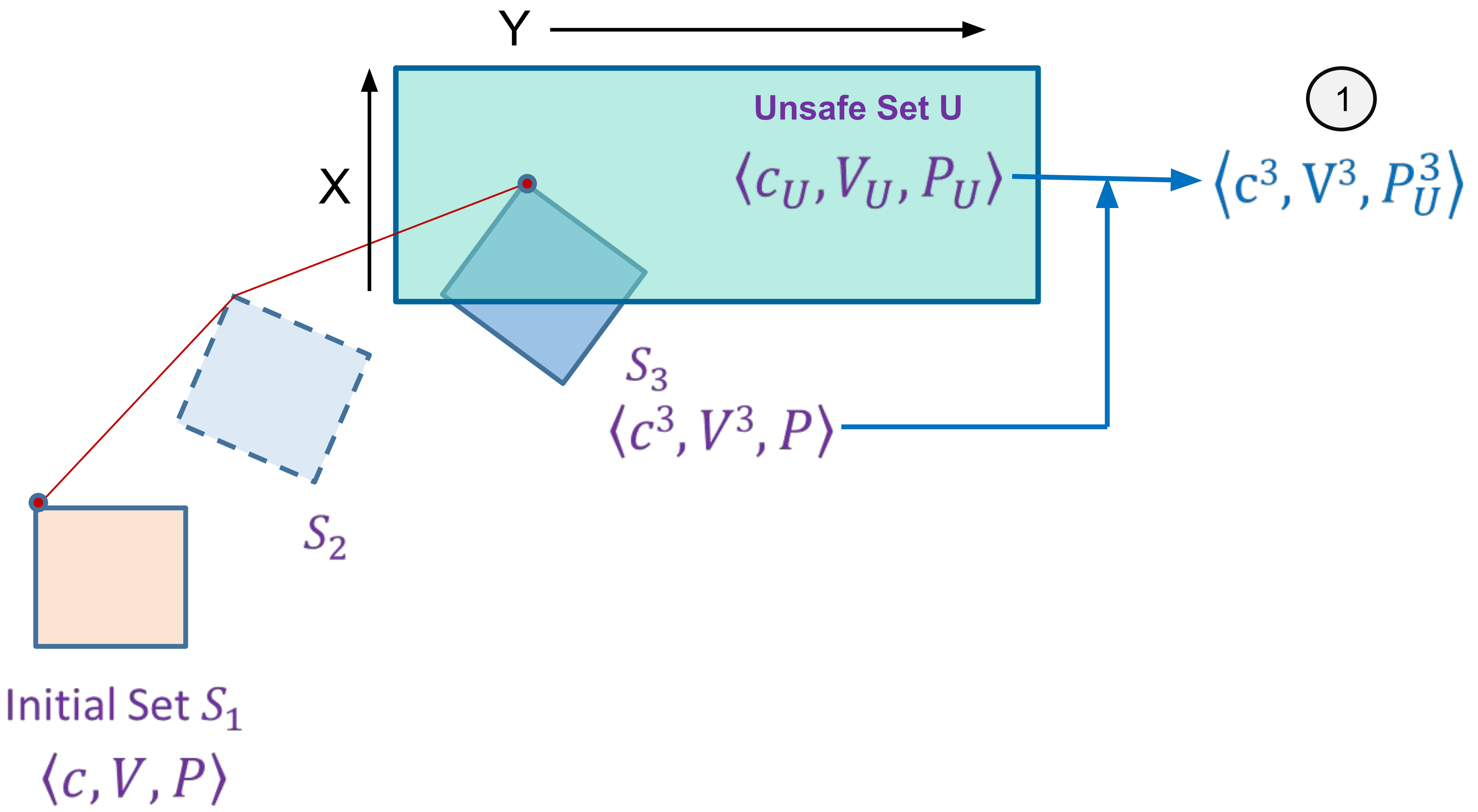
Constraint Propagation



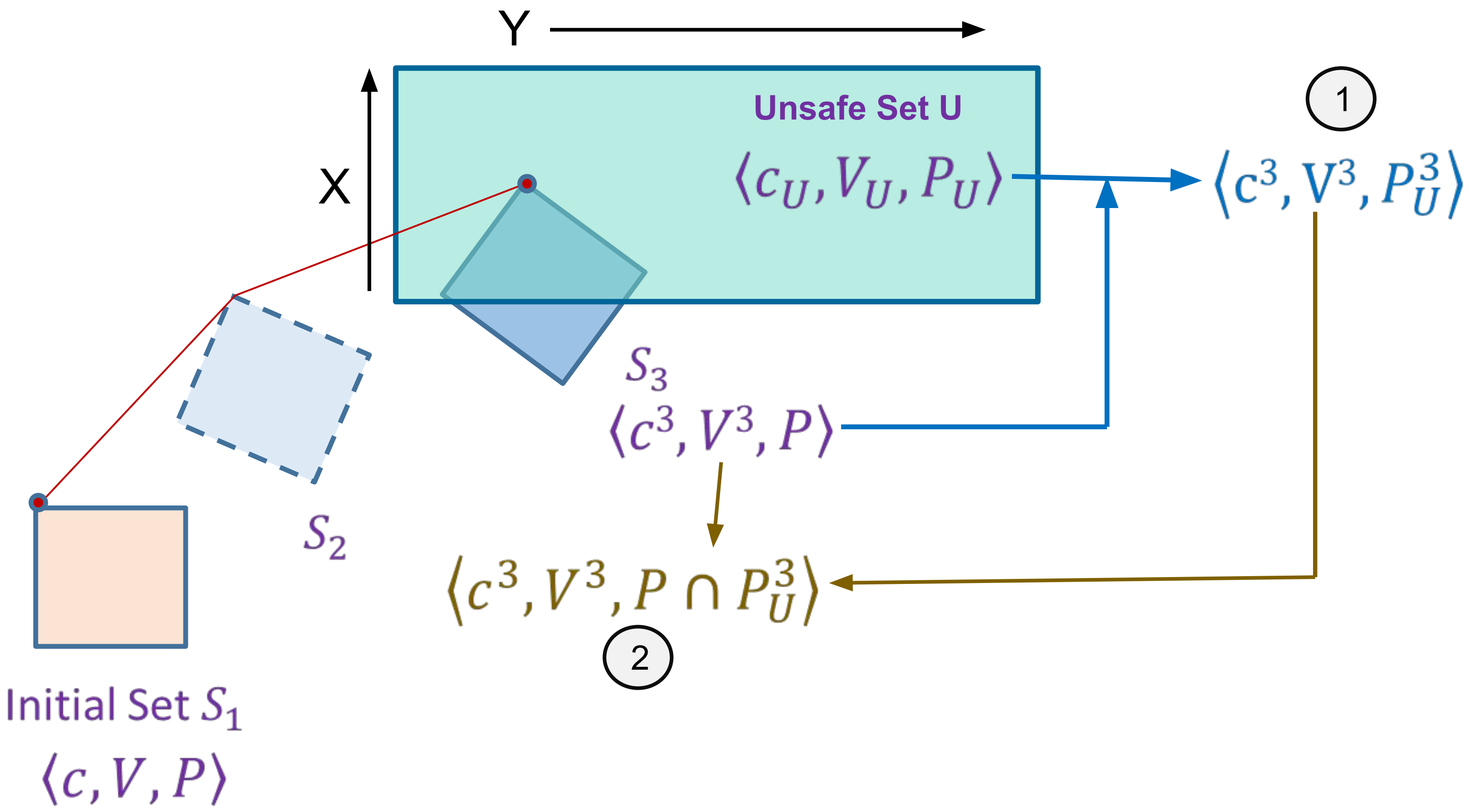
Constraint Propagation



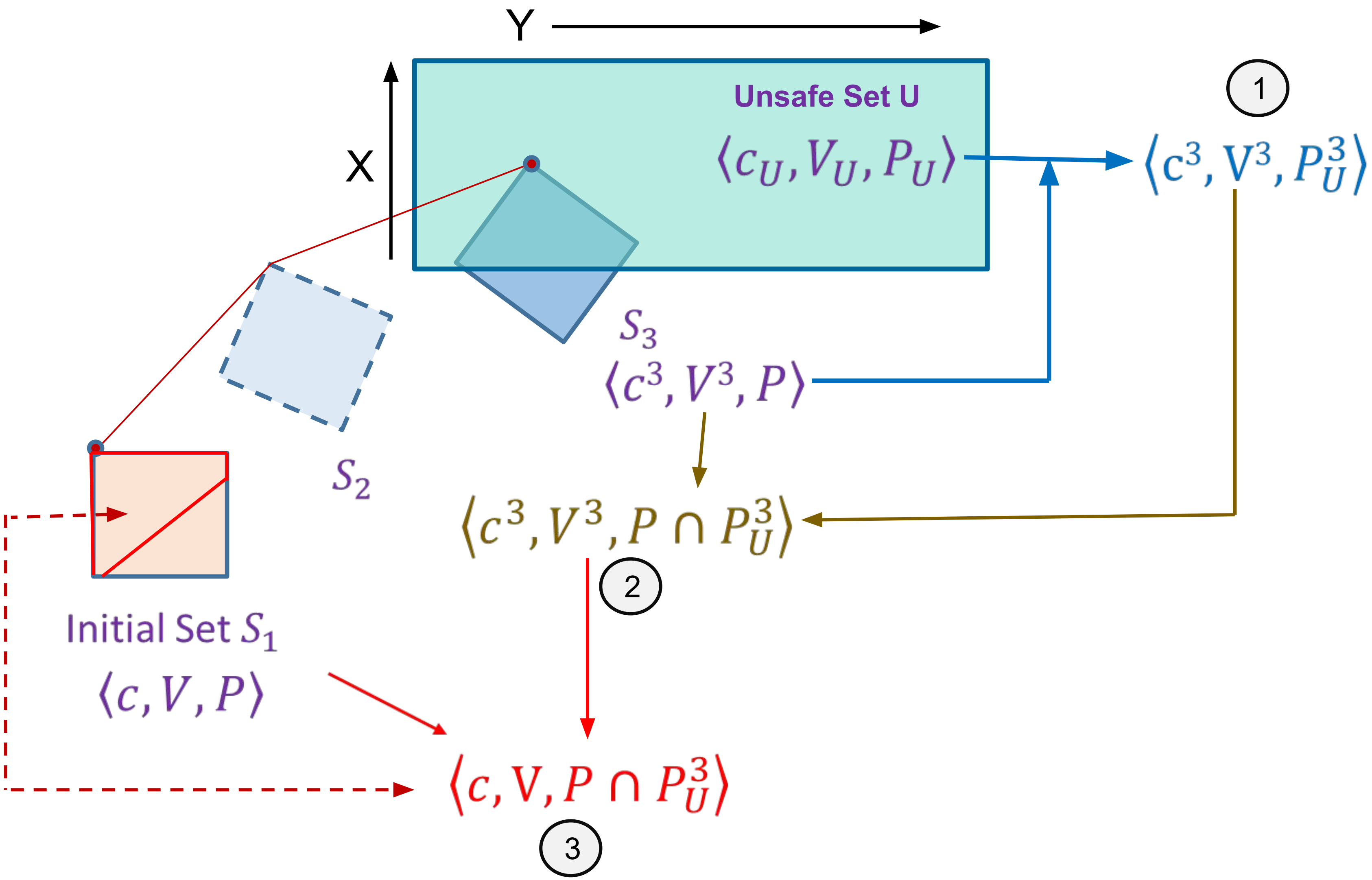
Constraint Propagation



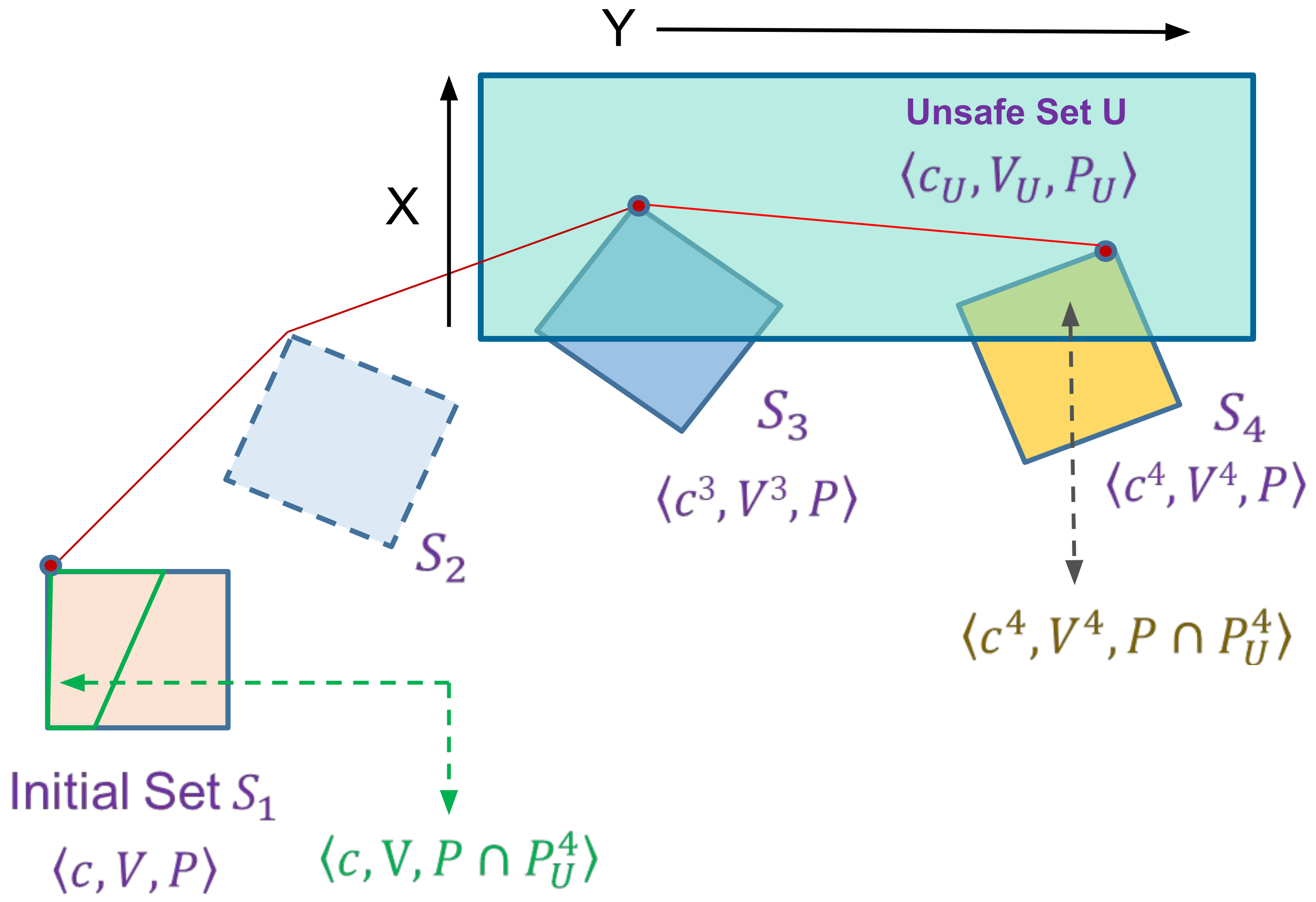
Constraint Propagation



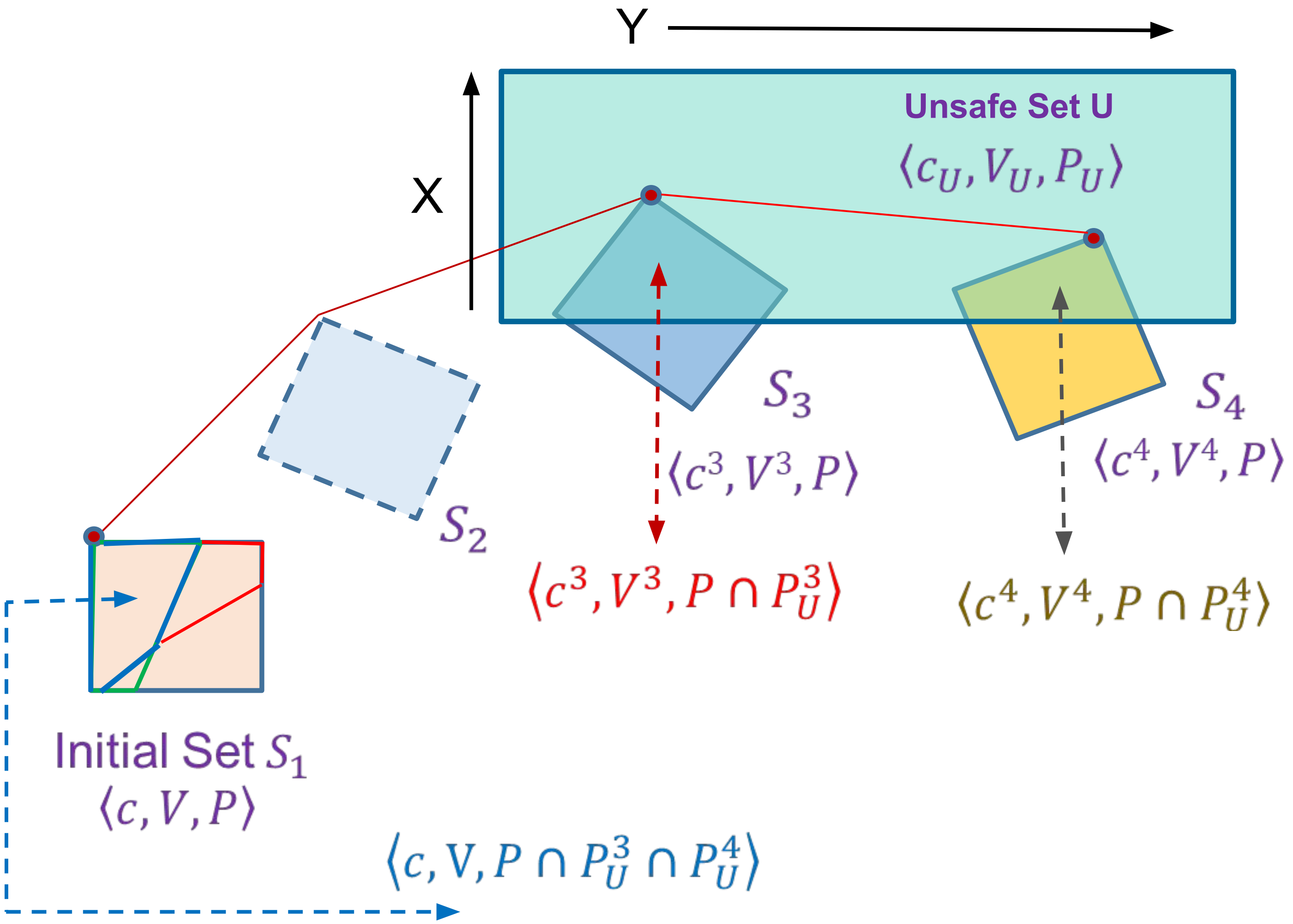
Constraint Propagation



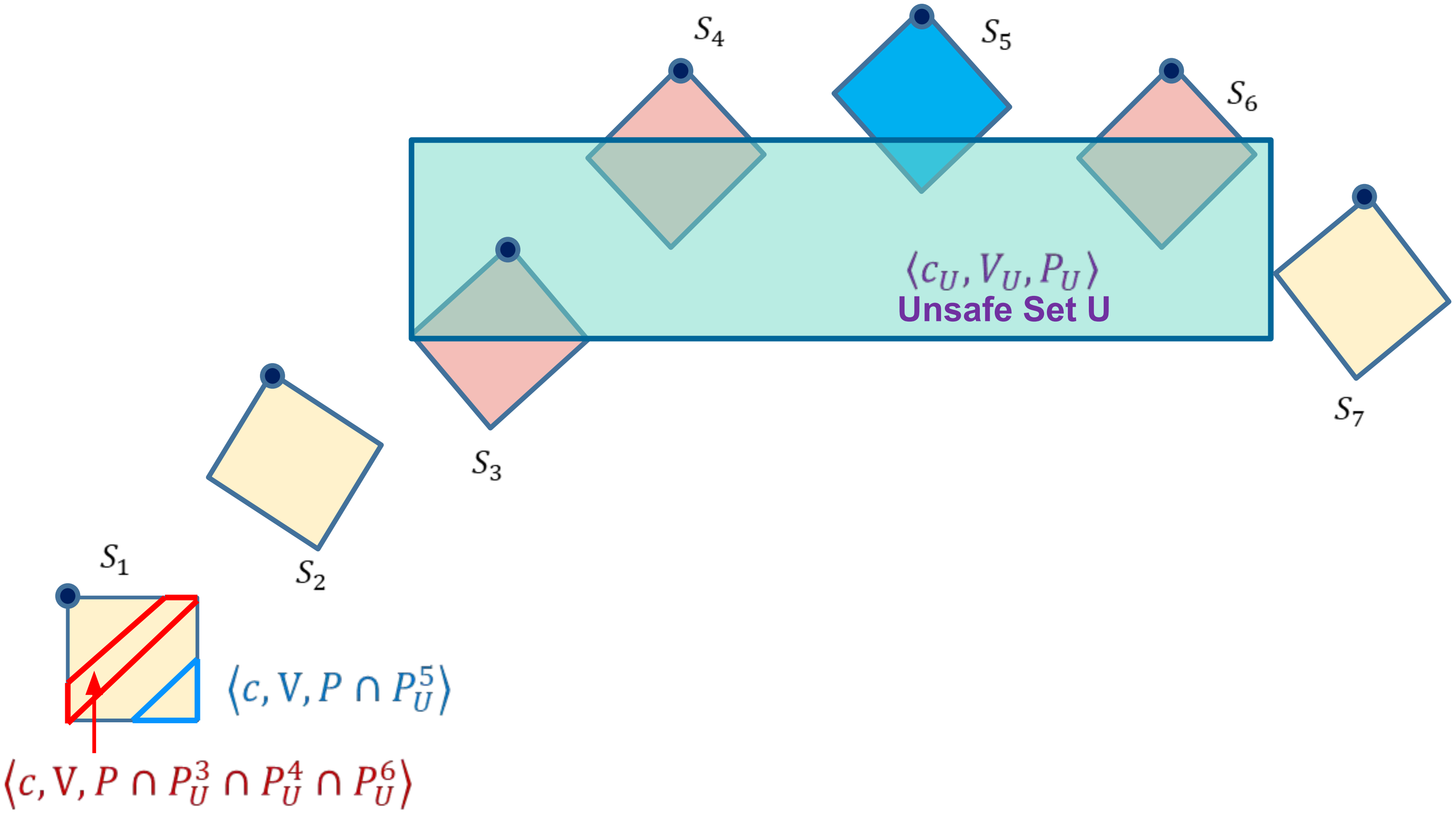
Constraint Propagation



Constraint Propagation

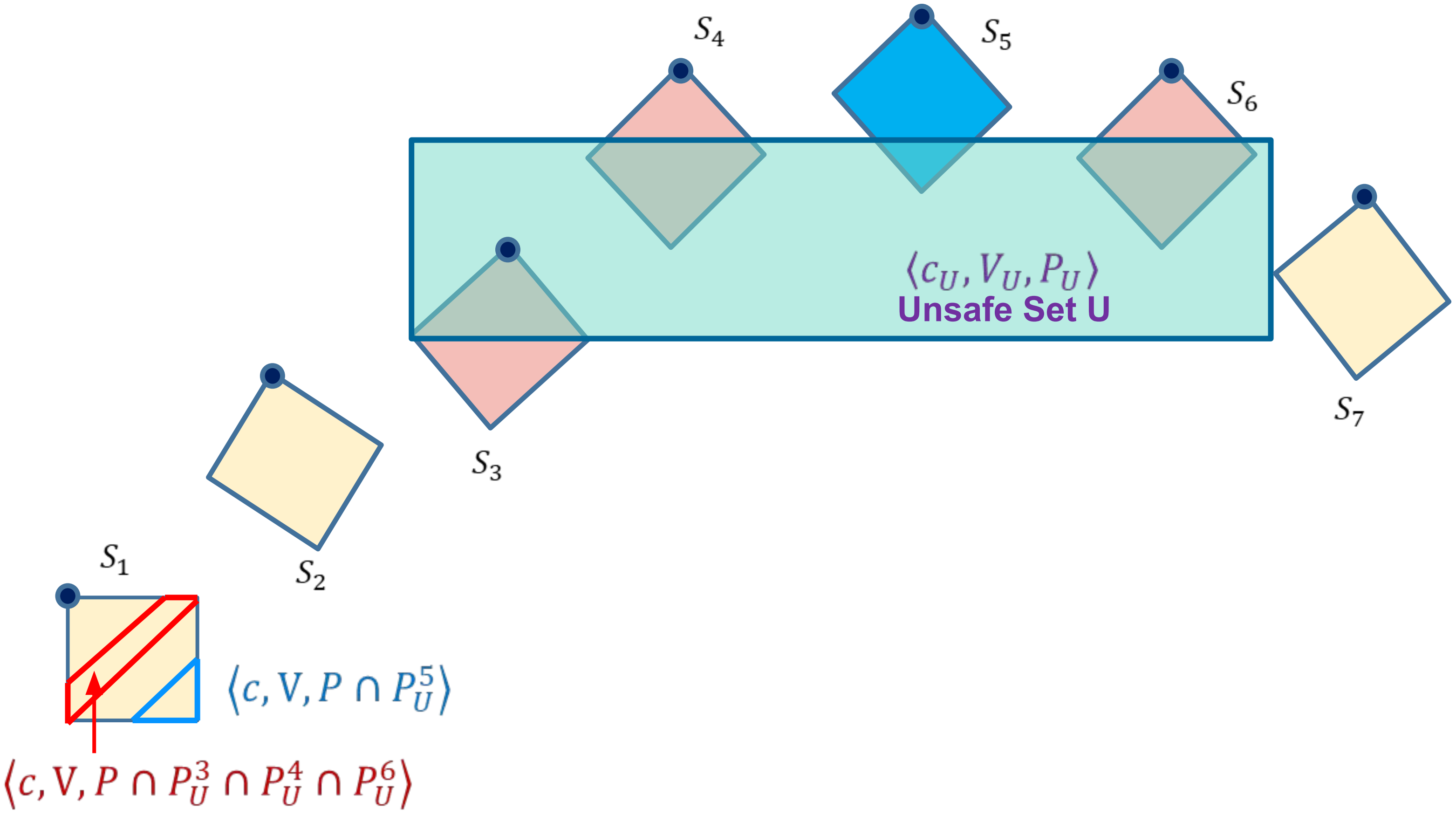


Constraint Propagation



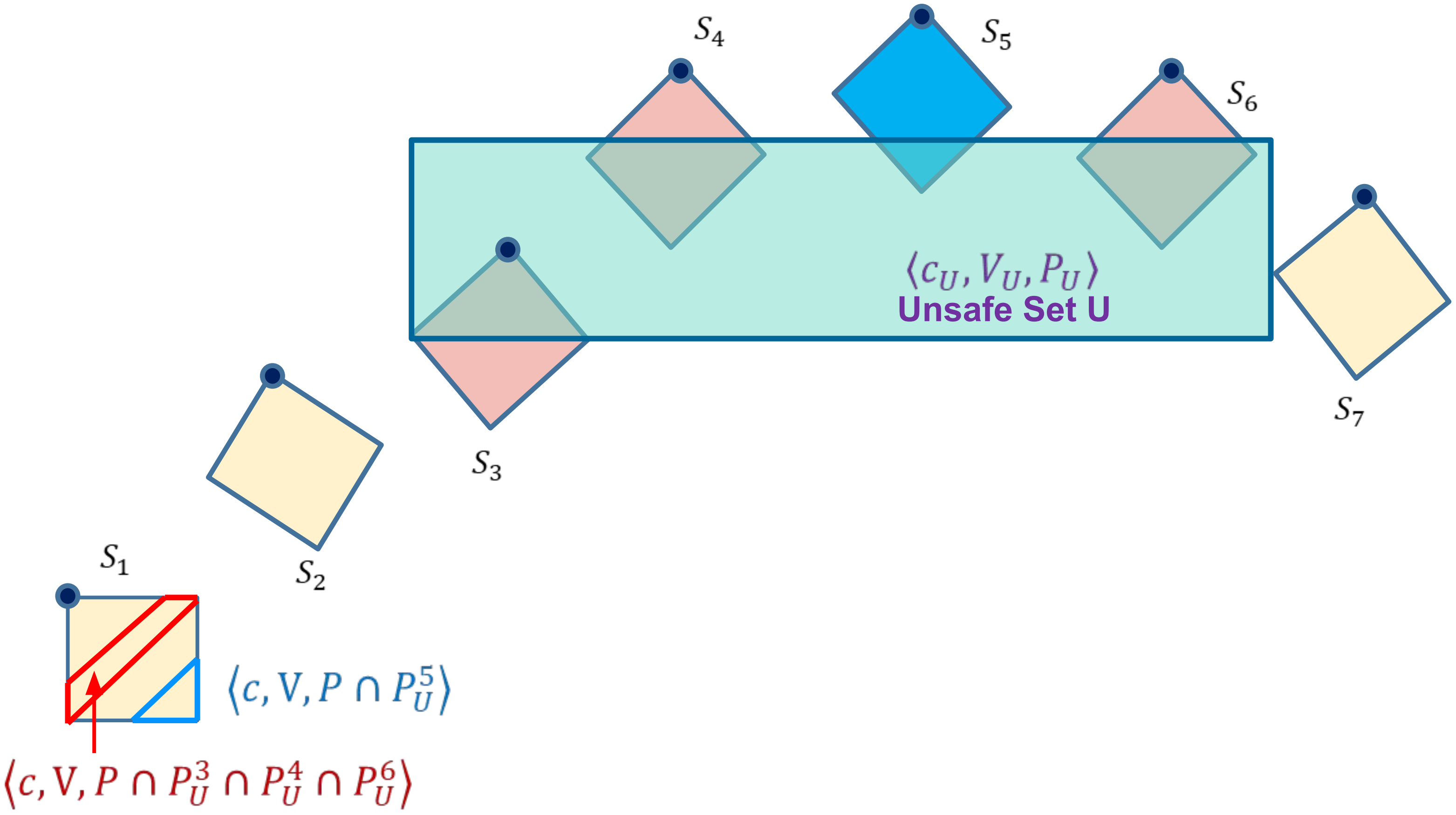
Constraint Propagation

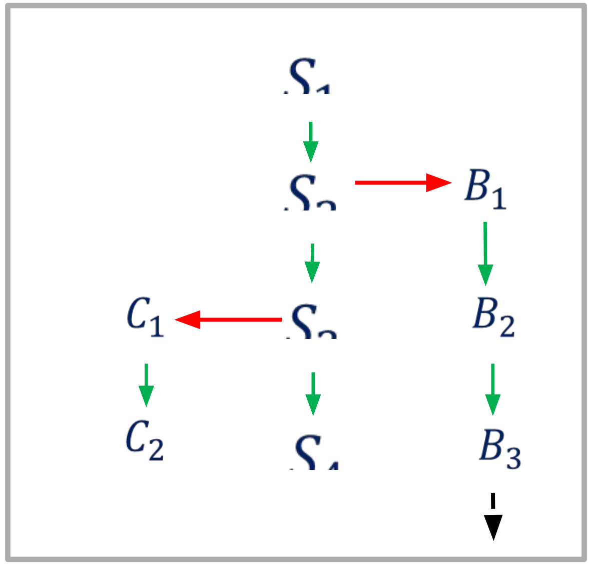
Computationally hard!



Constraint Propagation

Computationally hard!





Outline

- ✔ Background
- ✔ Introduction
- ✔ Preliminaries
- ✔ Methodology
- Frameworks
- Results

MILP-based Framework

for *each path* Γ *in* $ReachTree$ **do**

  $\Pi \triangleq \{\mathbb{S}_i | \mathbb{S}_i \in \Gamma, \mathbb{S}_i \cap U \neq \emptyset\}$;

  Introduce $|\Pi|$ decision variables $z_1, z_2 \ldots z_{|\Pi|}$;

  $\mathcal{C}_\Pi \leftarrow \emptyset$;

  Transform U into $\langle c^i, V^i, P_U^i \rangle$ where

   $\Pi[i] \triangleq \langle c^i, V^i, P^i \rangle$;

  $\mathcal{C}_\Pi \leftarrow \bigwedge_{i=1}^{|\Pi|} \bigwedge_{c \in P_U^i \wedge P^i} c + M(1 - z_i)$;

  $length_\Pi \leftarrow max \sum_i z_i$ while C_Π is feasible;

  if $length_\Pi > length_{max}$ **then**

    $length_{max} \leftarrow length_\Pi$;

    $\bar{\alpha}_{len} \leftarrow feasible(\mathcal{C}_\Pi)$;

  end

end

SMT-based Framework

for *each path* Γ *in* $ReachTree$ **do**

$\Pi \triangleq \{\mathbb{S}_i | \mathbb{S}_i \in \Gamma, \mathbb{S}_i \cap U \neq \emptyset\}$;

Introduce $|\Pi|$ binary variables $b_1, b_2 \ldots b_{|\Pi|}$;

$\mathcal{C}_\Pi \leftarrow \triangle_{i=1}^{|\Pi|} b_i$; **Soft constraints**

Transform U into $\langle c^i, V^i, P_U^i \rangle$ where $\Pi[i] \triangleq \langle c^i, V^i, P^i \rangle$;

$\mathcal{C}_\Pi \leftarrow C_\Pi \bigwedge_{i=1}^{|\Pi|} (b_i == (P_U^i \wedge P^i))$; **Hard constraints**

$length_\Pi \leftarrow Optimize_{SMT}(\mathcal{C}_\Pi)$ while C_Π is feasible;

if $length_\Pi > length_{max}$ **then**

$length_{max} \leftarrow length_\Pi$;

$\bar{\alpha}_{len} \leftarrow feasible(\mathcal{C}_\Pi)$;

end

end

Outline

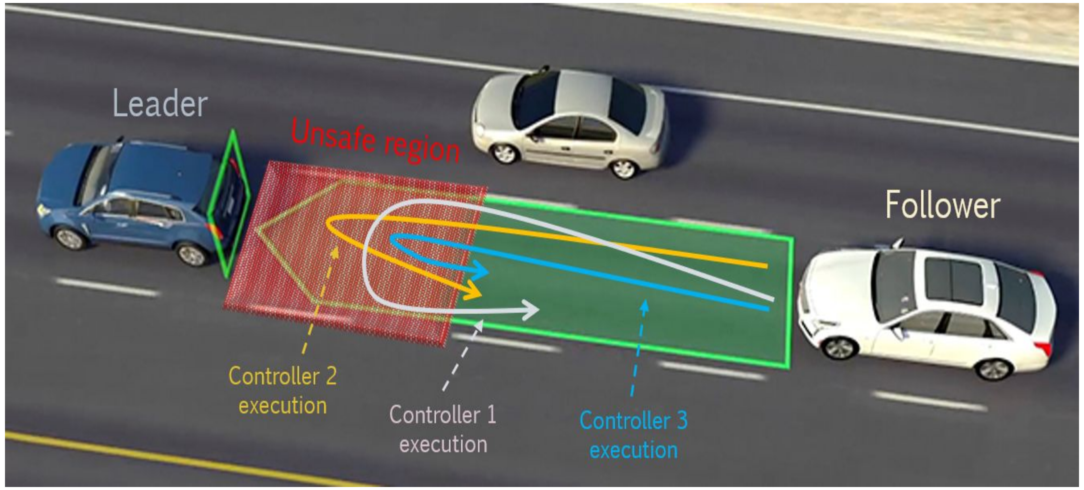
- ✔ Background
- ✔ Introduction
- ✔ Preliminaries
- ✔ Methodology
- ✔ Frameworks
- Results

Evaluation

Adaptive Cruise Control*

$$\dot{s} = (v_f - v)$$
$$\dot{v} = a$$
$$\dot{a} = g_1 a + g_2(v - v_f) + g_3(s - (v + 10))$$

v_f : leading car's velocity
v : follower's speed
a: follower's acceleration
s: distance

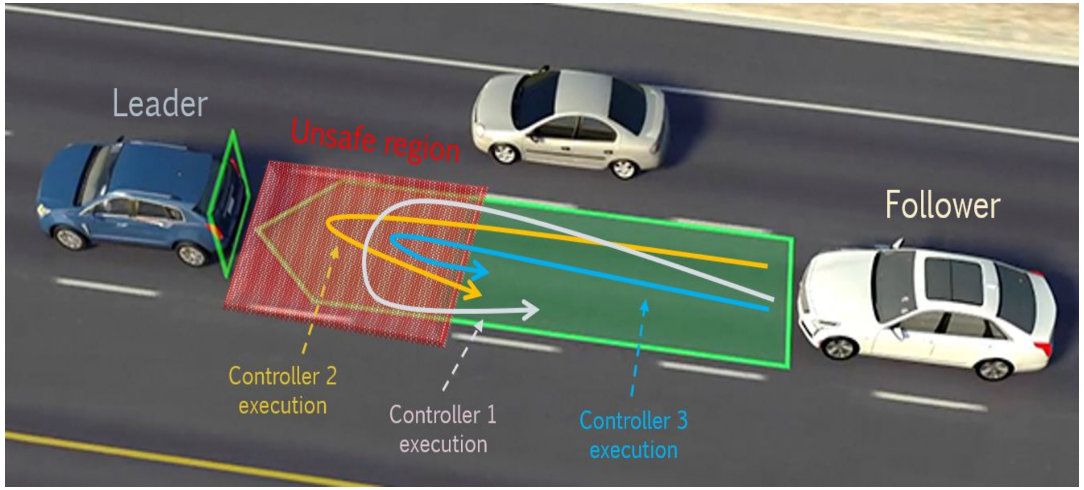


*A. Tiwari, Approximate reachability for linear systems. HSCC, 2003

Evaluation

Adaptive Cruise Control*

$$\dot{s} = (v_f - v)$$
$$\dot{v} = a$$
$$\dot{a} = g_1 a + g_2(v - v_f) + g_3(s - (v + 10))$$



v_f : leading car's velocity
v : follower's speed
a: follower's acceleration
s: distance

Initial Values

$s \in [2, 5]$
$v \in [18, 22]$
$v_f = 20$
$a \in [-1, 1]$

Controller 1

$g_1 = -3$
$g_2 = -3$
$g_3 = 1$

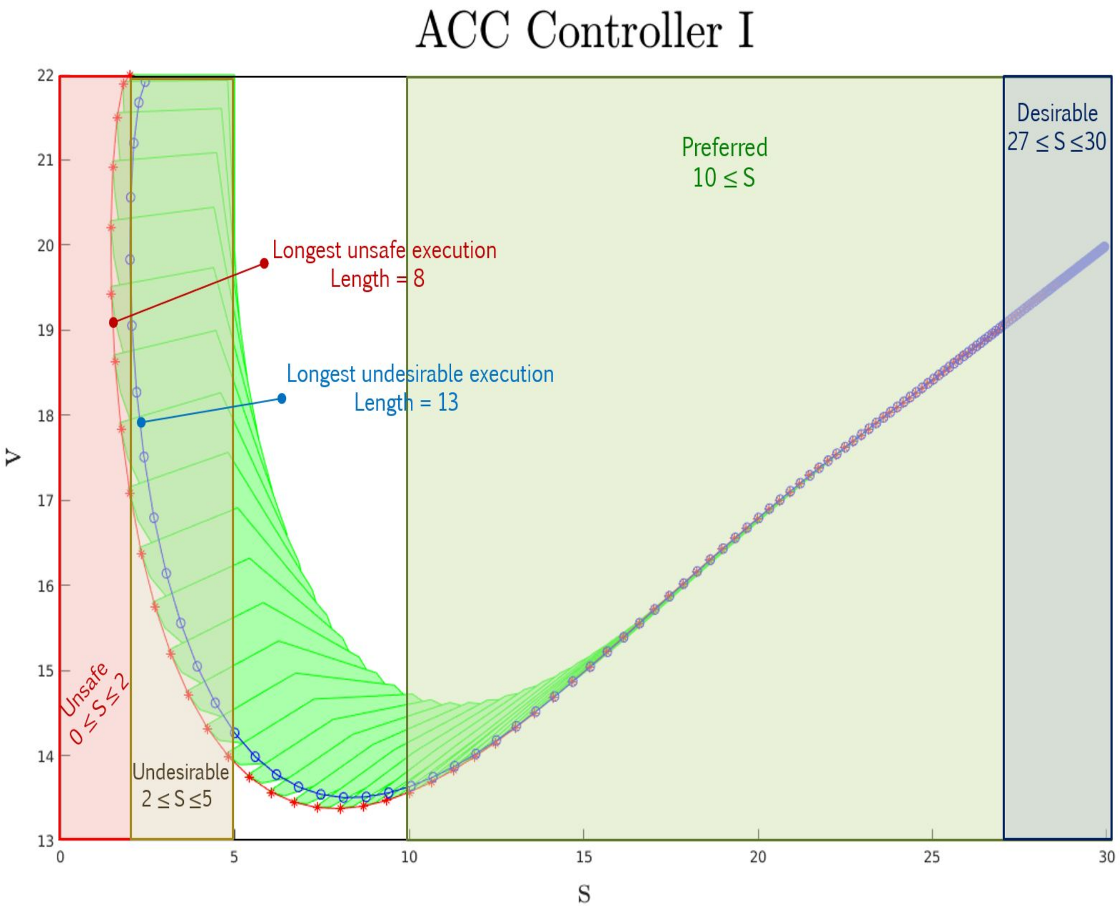
Controller 2

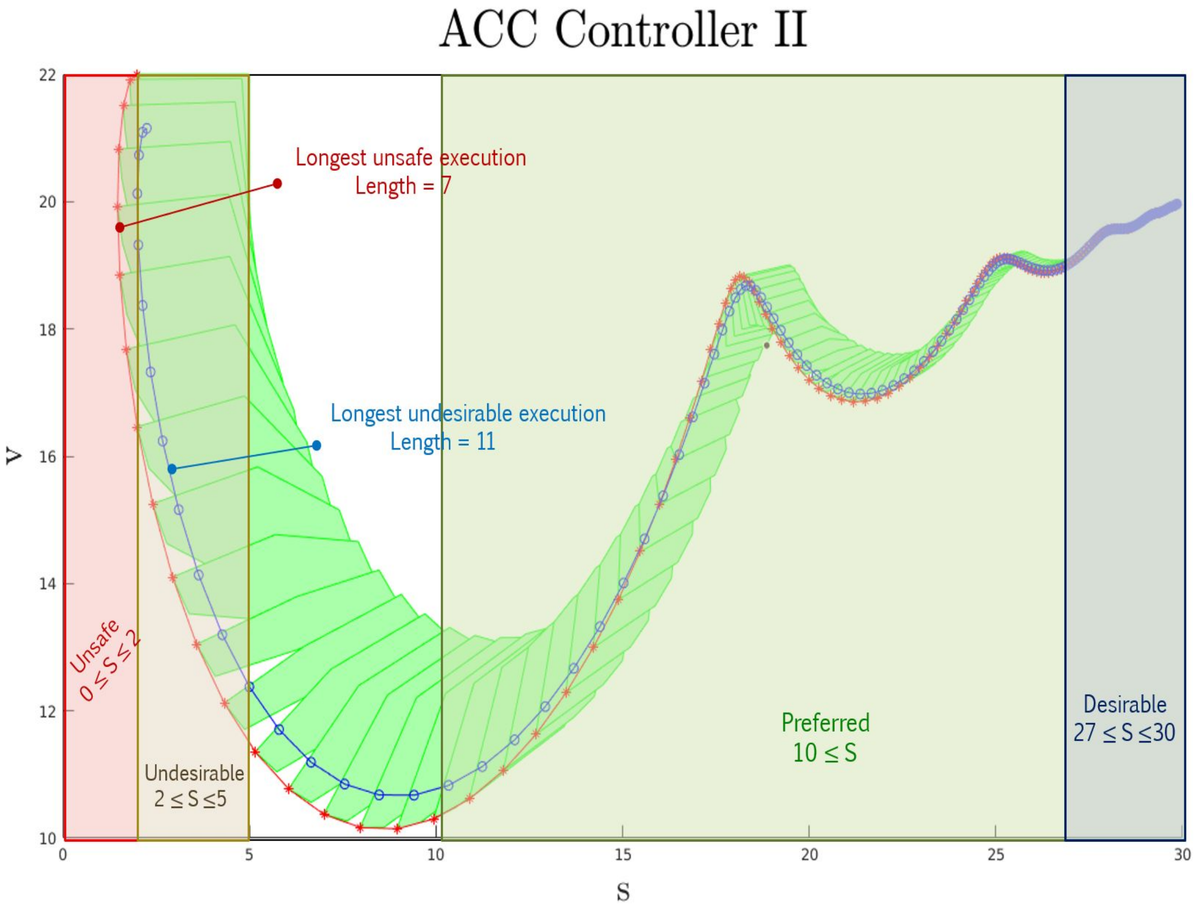
$g_1 = -1$
$g_2 = -3$
$g_3 = 1$

*A. Tiwari, Approximate reachability for linear systems. HSCC, 2003

Evaluation

$$\dot{s} = (v_f - v)$$
$$\dot{v} = a$$
$$\dot{a} = g_1 a + g_2(v - v_f) + g_3(s - (v + 10))$$





*M. Goyal, P. S. Duggirala, Extracting counterexamples induced by safety violation in linear hybrid systems, Automatica, 07/2020

Evaluation

Model	Dims, Modes	Actual Inter. Duration	LCE Duration		Counterexample		Verification Time (sec)	LCE Gen. Time	
			MILP	SMT	MILP	SMT		MILP	SMT
Damped Oscillator 1	2, 1	[5 10] [34 44] [66 74]	[5 9] [35 44] [66 73]	[5 9] [35 44] [66 73]	[-5.28 0.764]	[-5.321 0.865]	0.44	0.04	0.51
Damped Oscillator 2	2, 1	[3 10] [29 49] [59 100]	[3 10] [30 49] [59 100]	[3 9] [29 49] [59 100]	[-5.0 0.398]	[-5.0 0.606]	0.59	0.04	0.55
Buck Converter	4, 6	**cl1:** [13 21] **op1:** [22 50] **cl2:** [51]	[13 21] [22 50] [51]	[13 21] [22 50] [51]	il = 1.0 vc = 0 t = 0, gt = 0	il = 0.6892 vc = 0 t= 0, gt = 0	0.66	0.04	0.60
Filtered Oscillator	34, 4	**loc3:** [3 5] **loc3:** [7 21] **loc4:** [26]	[5] [7 21] [26]	[5] [7 21] [26]	[0.2069 0.07 0...]	[0.205 0.07 0...]	37	2.14	49

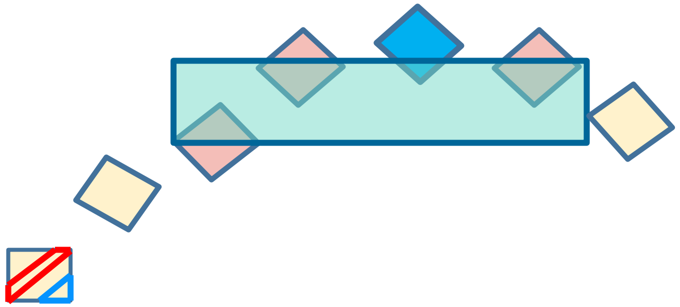
Takeaways

- Use the artifacts from verification
- Search in the space of basis variables defining the initial set
- Longest counterexample may not be unique
- MILP-based framework is faster
- SMT-based formalism provides guarantees

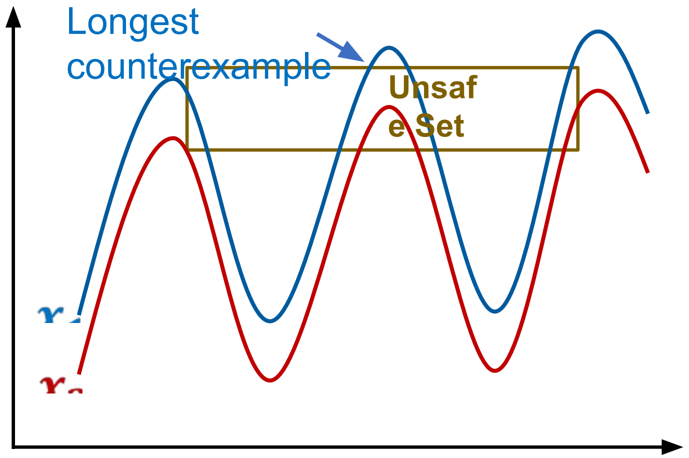
Future work

- Explore BDD-based optimization techniques
- Use various counterexamples in controller synthesis

Generating Longest Counterexample: On the Cross-roads of MILP and SMT

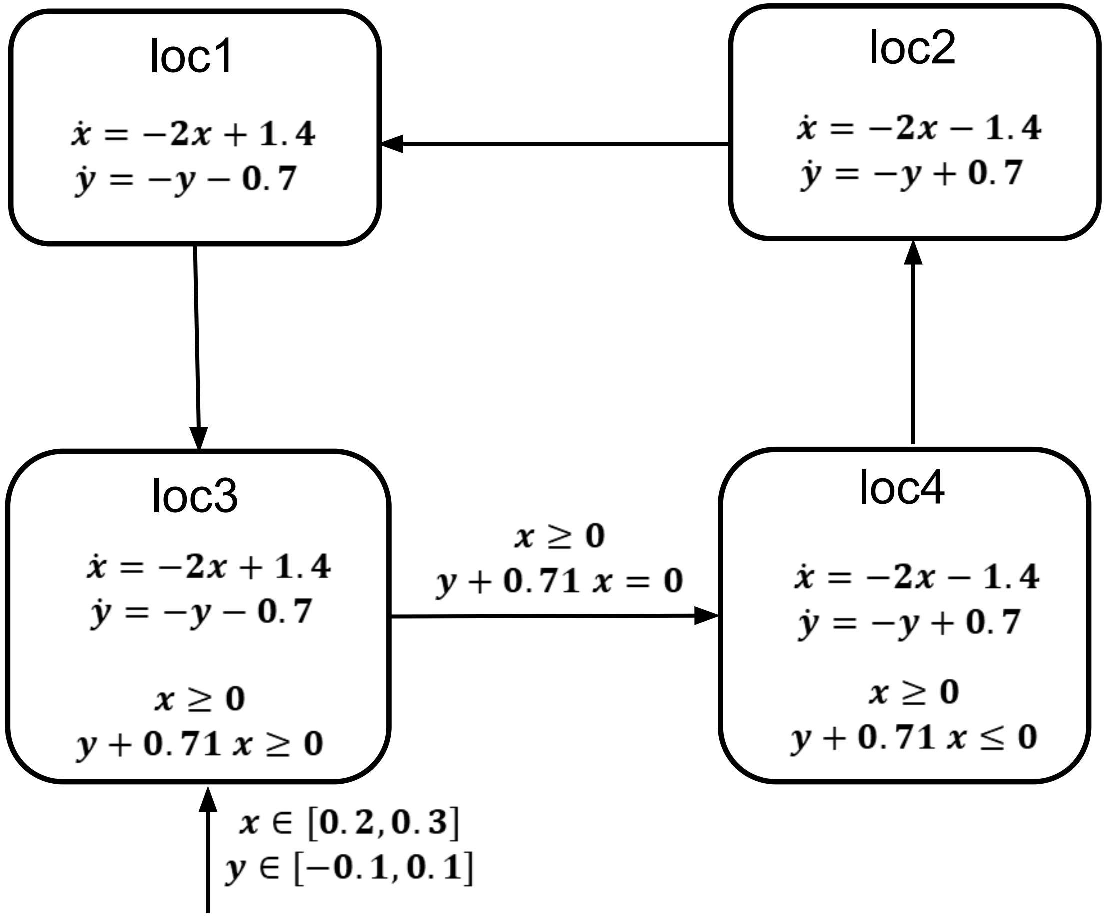


Constraint propagation



manish.goyal@cs.unc.edu
david.bergman@uconn.edu
psd@cs.unc.edu

Verification



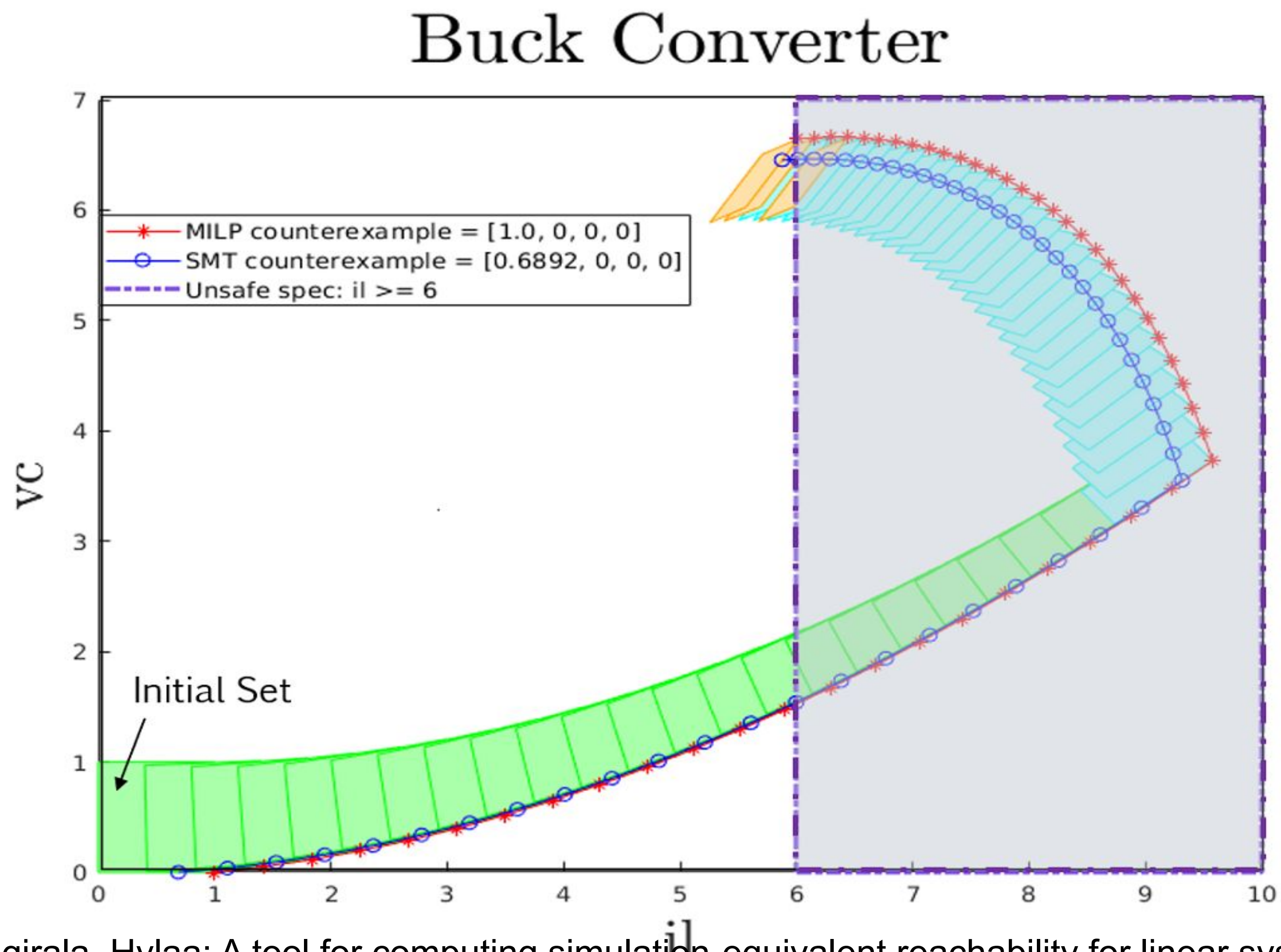
Filtered Oscillator

What to do!

The entire ‘unsafe’ reachable set

Randomly generated counterexample

Evaluation*: HyLaa



*S. Bak and P. S. Duggirala, Hylaa: A tool for computing simulation-equivalent reachability for linear systems. HSCC 2017