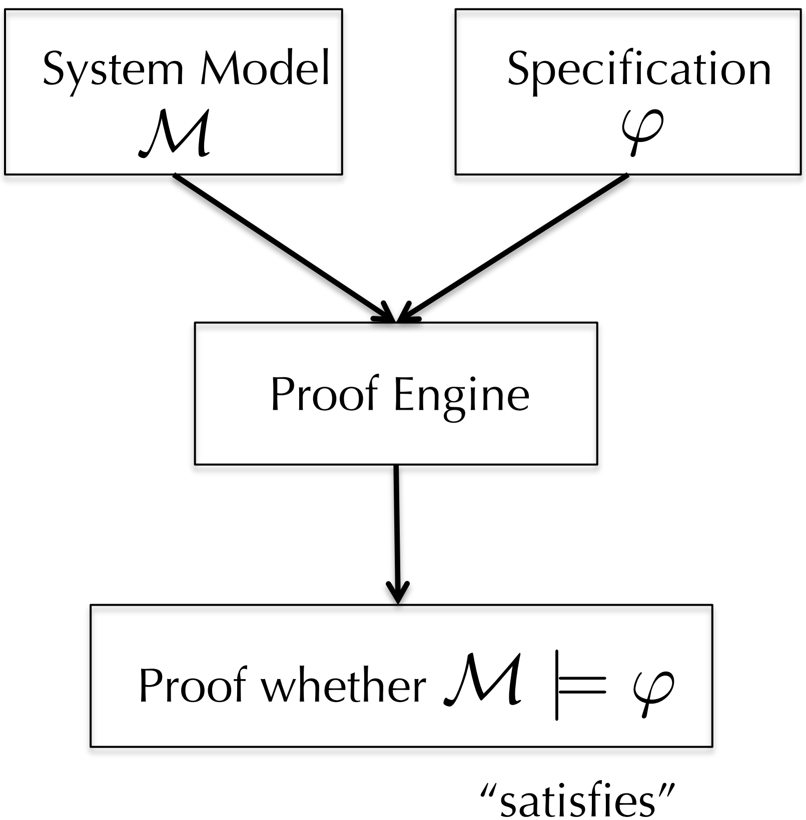
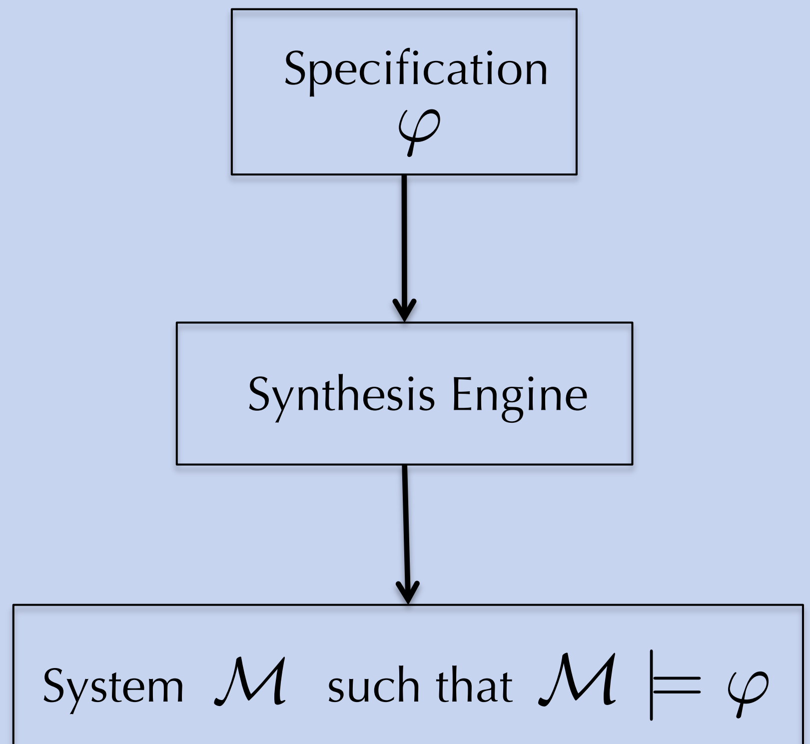
Formal Methods: Two Perspectives

Verification



Synthesis

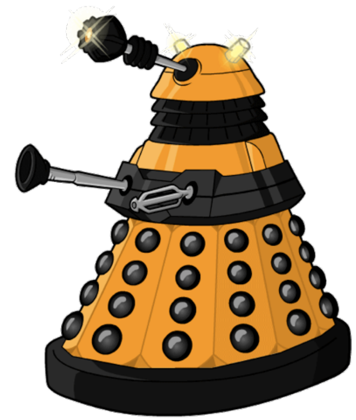


Reactive Synthesis

Most systems do not operate in isolation!

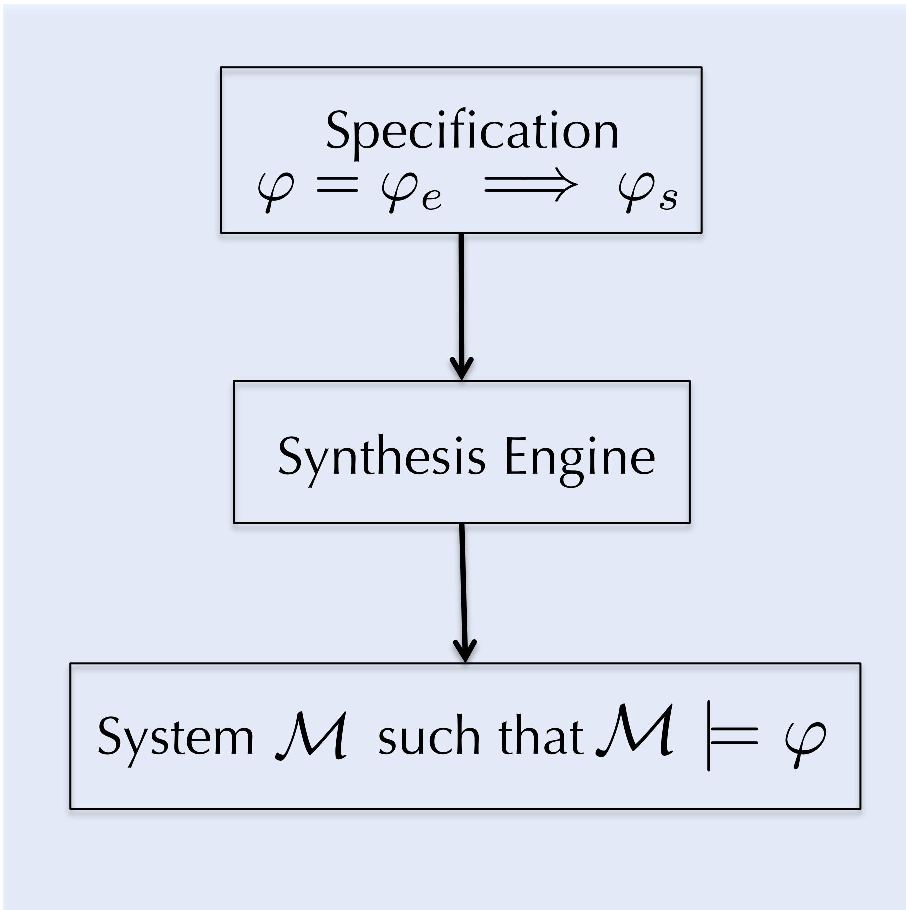
Given $\varphi = \varphi_e \implies \varphi_s$

environment assumptions (for φ_e) — system guarantees (for φ_s)



synthesize a **strategy** $\mathcal{M}$ such that $\mathcal{M} \models \varphi$

Contributions



Signal Temporal Logic

MILP solver

Control input maximizing satisfaction

Highlights

- Counterexample-Guided Synthesis
- Model Predictive Control
- Semi-decision procedure for reactive synthesis from STL

Reactive Synthesis from Temporal Logic

- Robotics
  - Kress-Gazit, Fainekos and Pappas, ICRA 2007
- Autonomous Cars
  - Wongpiromsarn, Topcu and Murray, HSCC 2010
- Aircraft Electric Power Systems
  - Nuzzo et al, IEEE Access 2013
- Switched Systems
  - Liu, Ozay, Topcu and Murray, IEEE TAC 2013

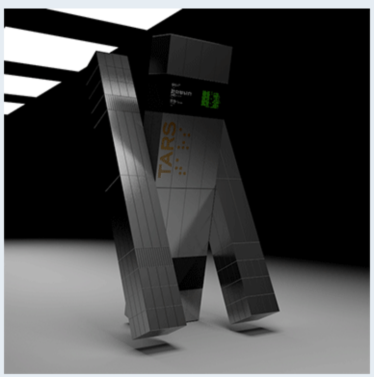
Based on **Linear Temporal Logic** Synthesis

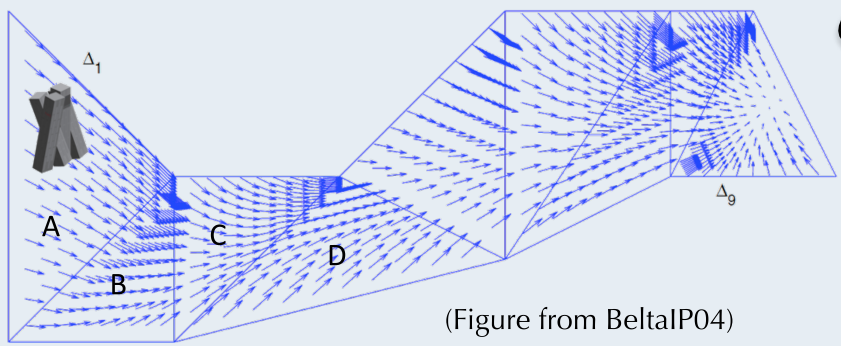
LTL Synthesis for Hybrid Systems

Create discrete **abstraction** → **Synthesize** correct discrete controller → Continuously **implement** discrete solution

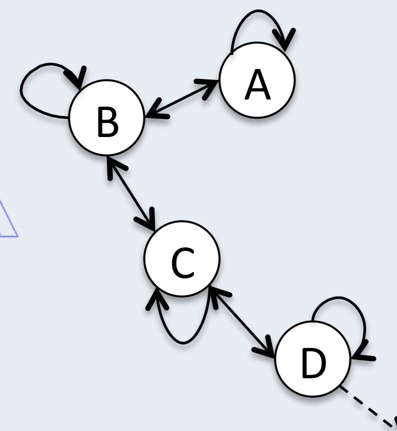
LTL Synthesis for Hybrid Systems

Create discrete **abstraction** → **Synthesize** correct discrete controller → Continuously **implement** discrete solution





(Figure from BeltaIP04)



Dynamical system → Labeled transition system

[AlurHLP00, BeltaH06, HabetsCS06, KaramanF09, Kress-GazitFP07, KloetzerB08, TabuadaP06, WongpiromsarnTM12,…]

LTL Synthesis for Hybrid Systems

Create discrete **abstraction** → **Synthesize** correct discrete controller → Continuously **implement** discrete solution

BUT

- **Discrete abstraction** is expensive
- LTL is inconvenient for specifying
  - properties of **continuous signals**
  - **temporal duration** of events

Signal Temporal Logic (STL)

[Maler and Nickovic 04]

- Continuous predicates: $\boxed{\mu(\mathbf{x}) > 0}$

No need to discretize the state space!

- Boolean operators:

$\wedge$ (and) $\vee$ (or) $\neg$ (not) $\implies$ (implies)

- Temporal operations:

$\varphi \, \mathcal{U}_{[a,b]} \psi$ (until) $\boxed{\Box_{[a,b]}}\varphi$ (always) $\Diamond_{[a,b]} \varphi$ (eventually)

"At all times between a and b from now"

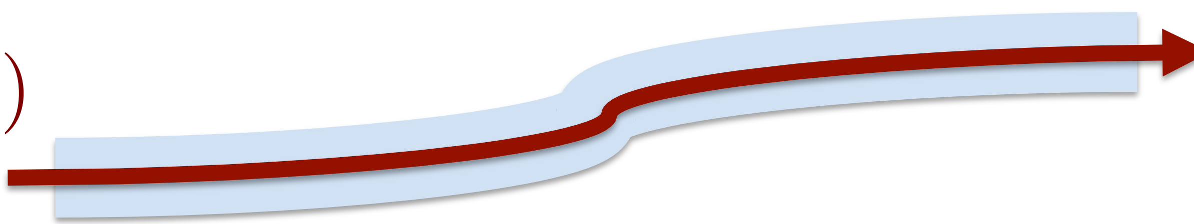
Quantitative Semantics for STL

[Donzé and Maler '10]

Robustness function $\rho^{\varphi} : \mathcal{X} \times \mathbb{N} \to \mathbb{R}$

$$(\mathbf{x}, t) \models \varphi \equiv \rho^{\varphi}(\mathbf{x}, t) > 0$$

$$|\mathbf{x}'_t - \mathbf{x}_t| < \rho^{\varphi}(\mathbf{x}, t)$$
$$\Rightarrow (\mathbf{x}', t) \models \varphi$$



Example: STL Quantitative Semantics

$$\mu_1 \equiv x - 3 > 0 \qquad \varphi = \Box_{[0,2]} \mu_1$$

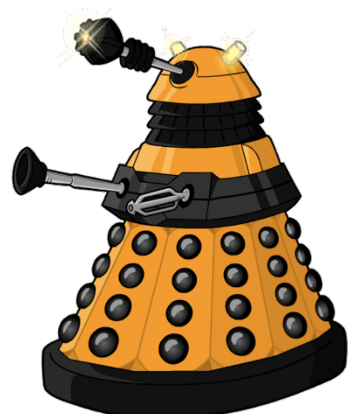
- Define $\rho^{\varphi} : \mathcal{X} \times \mathbb{N} \to \mathbb{R}$
  such that $(\mathbf{x}, t) \models \varphi \equiv \rho^{\varphi}(\mathbf{x}, t) > 0$
  - $\rho^{\mu_1}(x, 0) = x(0) - 3$
  - $\rho^{\mu_1 \wedge \mu_2}(x, t) = \min(\rho^{\mu_1}, \rho^{\mu_2})$
  - $\rho^{\varphi}(x, t) = \min_{t \in [0,2]} \rho^{\mu_1}(x, t) = \min_{t \in [0,2]} x(t) - 3$

Reactive Synthesis from STL

$$\varphi = \varphi_e \implies \varphi_s$$

environment assumptions

system guarantees



min

max

$$(\mathbf{x}, t) \models \varphi \equiv \rho^{\varphi}(\mathbf{x}, t) > 0$$

Preliminaries

- Continuous time hybrid system

$$\dot{x} = f(x, u, w)$$

$$\begin{aligned} x &\in \mathcal{X} \subseteq (\mathbb{R}^{n_c} \times \{0,1\}^{n_l}) \\ u &\in U \subseteq (\mathbb{R}^{m_c} \times \{0,1\}^{m_l}) \\ w &\in W \subseteq (\mathbb{R}^{e_c} \times \{0,1\}^{e_l}) \end{aligned}$$

- Assume discrete-time approximation

$$x(t_{k+1}) = f_d(x(t_k), u(t_k), w(t_k))$$

$$t_{k+1} - t_k = \Delta t > 0$$

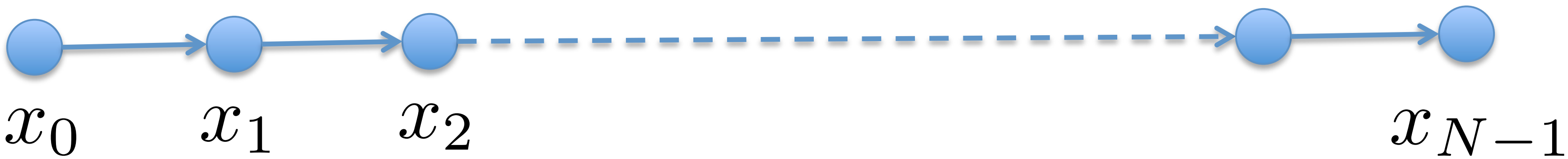
- A **run**

$$\xi(x_0, \mathbf{u}, \mathbf{w}) = (x_0 u_0 w_0)(x_1 u_1 w_1)(x_2 u_2 w_2)...$$

$$\xi(x_0, \mathbf{u}^N, \mathbf{w}^N) = (x_0 u_0 w_0)(x_1 u_1 w_1)(x_2 u_2 w_2)...(x_N u_N w_N)$$

Finite Run Parametrization

- Bounded-length *N* based on formula



$$x(t_{k+1}) = f_d(x(t_k), u(t_k), w(t_k))$$

$$N \geq Bound(\varphi)$$

e.g. $Bound(\Box_{[0,10]} \Diamond_{[1,6]} \psi) = 16$

- Inspired by bounded model checking

  [Biere et al. 99, Biere et al. 06, Clarke et al. 01]

STL Reactive Synthesis

Given:

Continuous-time system $\dot{x} = f(x, u, w)$

STL specification $\varphi = \varphi_e \implies \varphi_s$

Initial state x_0

Cost function J on runs

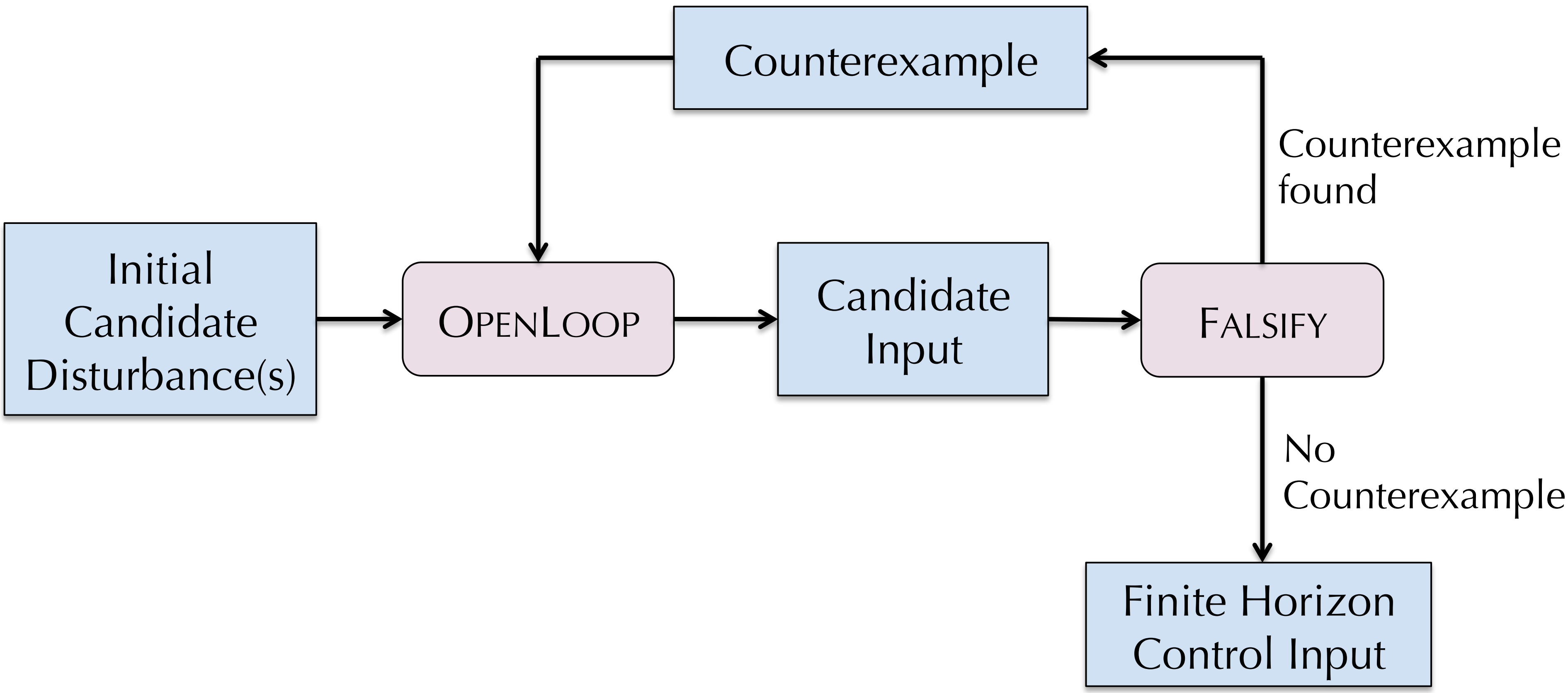
Compute:

$$\operatorname*{argmin}_{\mathbf{u}^N} \quad \max_{\mathbf{w}^N \in \{\mathbf{w} \in W^N | \mathbf{w} \models \varphi_e\}} \quad J(\xi(x_0, \mathbf{u}^N, \mathbf{w}^N))$$

$$\text{s.t.} \quad \forall \mathbf{w}^N \in W^N, \quad \xi(x_0, \mathbf{u}^N, \mathbf{w}^N) \models \varphi$$

CEGIS: Counterexample-Guided Inductive Synthesis

[Solar-Lezama et al, ASPLOS 2006]



CEGIS: Counterexample-Guided Inductive Synthesis

procedure $\mathrm{CEGIS}(\xi, x_0, N, \varphi, J)$

Initial Guess

  Let $\mathbf{w}^0 = (w_1^0, w_2^0, \ldots w_{N-1}^0)$, s.t. $\mathbf{w}^N \models \varphi_e$
  $W_{cand} = \{\mathbf{w}^0\}$
  while `True` **do**

OpenLoop

$$\mathbf{u}^0 \leftarrow \underset{\mathbf{u} \in U^N}{\operatorname{argmin}} \quad \max_{\mathbf{w}^0 \in W_{cand}} (J(\xi(x_0, \mathbf{u}, \mathbf{w}^0))) \quad \text{s.t. } \forall \mathbf{w}^0 \in W_{cand},\ \xi(x_0, \mathbf{u}, \mathbf{w}^0) \models \varphi_s,$$

    if $\mathbf{u}^0 ==$ `null` **then**
      Return `INFEASIBLE`
    end if

Falsify

$$\mathbf{w}^1 \leftarrow \operatorname{argmin}_{\mathbf{w} \in W^N} \quad \rho^{\varphi}(\xi(x_0, \mathbf{u}^0, \mathbf{w}), 0) \quad \text{s.t. } \mathbf{w}^1 \models \varphi_e$$

    if $\rho^{\varphi}(\xi(x_0, \mathbf{u}^0, \mathbf{w}^1)) > 0$ **then**
      Return $\mathbf{u}^0$

Success!

Inductive Step

    else
      $W_{cand} \leftarrow W_{cand} \cup \{\mathbf{w}^1\}$
    end if
  end while
end procedure

OpenLoop and Falsify

OpenLoop

$$\mathbf{u}^0 \leftarrow \underset{\mathbf{u}\in U^N}{\text{argmin}} \quad \max_{\mathbf{w}^0 \in W_{cand}} (J(\xi(x_0, \mathbf{u}, \mathbf{w}^0)))$$
$$\text{s.t. } \forall \mathbf{w}^0 \in W_{cand},\ \xi(x_0, \mathbf{u}, \mathbf{w}^0) \models \varphi_s,$$

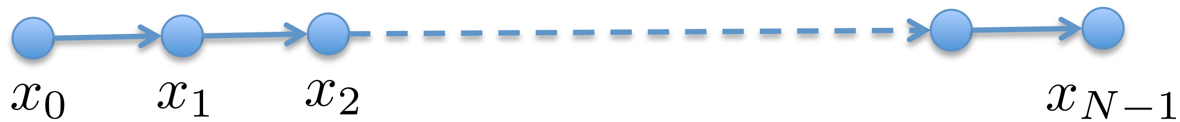
Falsify

$$\mathbf{w}^1 \leftarrow \text{argmin}_{\mathbf{w}\in W^N} \quad \rho^{\varphi}(\xi(x_0, \mathbf{u}^0, \mathbf{w}), 0)$$
$$\text{s.t. } \mathbf{w}^1 \models \varphi_e$$

<u>Problem</u>: find a bounded-time, open-loop sequence of inputs/disturbances

- satisfying an STL formula
- minimizing a cost

Open-Loop Synthesis



Problem: find a bounded-time sequence of inputs/disturbances

- satisfying an STL formula
- minimizing a cost

Solution: Encode everything (including the formula) as a Mixed Integer Linear Program

[Raman et al CDC 2014]

Example: STL Encoding

$$(\mathbf{x}, 0) \models \varphi \Leftrightarrow r_0^{\varphi} > 0$$

$$\varphi = \Box_{[0,10]} \Diamond_{[0,5]} ((T > 20) \wedge (T < 30))$$

$$r_t^{(T>20)} = T - 20$$

$$r_t^{(T<30)} = 30 - T$$

$$r_t^{((T>20)\wedge(T<30))} = \min(r_t^{(T>20)}, r_t^{(T<30)})$$

$$r_t^{\Diamond_{[0,5]}((T>20)\wedge(T<30))} = \max_{i=0,\ldots,5}(r_i^{((T>20)\wedge(T<30))})$$

$$r_t^{\Box_{[0,10]}\Diamond_{[0,5]}((T>20)\wedge(T<30))} = \min_{i=0,\ldots,10}(r_i^{\Diamond_{[0,5]}((T>20)\wedge(T<30))})$$

CEGIS: Counterexample-Guided Inductive Synthesis

Initial Guess

OpenLoop

Falsify

Inductive Step

procedure CEGIS$(\xi, x_0, N, \varphi, J)$

  Let $\mathbf{w}^0 = (w_1^0, w_2^0, \ldots w_{N-1}^0)$, s.t. $\mathbf{w}^N \models \varphi_e$

  $W_{cand} = \{\mathbf{w}^0\}$

  while True **do**

    $\mathbf{u}^0 \leftarrow \underset{\mathbf{u} \in U^N}{\operatorname{argmin}} \quad \max_{\mathbf{w}^0 \in W_{cand}} (J(\xi(x_0, \mathbf{u}, \mathbf{w}^0)))$ ✸

      s.t. $\forall \mathbf{w}^0 \in W_{cand}, \ \xi(x_0, \mathbf{u}, \mathbf{w}^0) \models \varphi_s,$

    if $\mathbf{u}^0 ==$ null **then**

      Return INFEASIBLE

    end if

    $\mathbf{w}^1 \leftarrow \operatorname{argmin}_{\mathbf{w} \in W^N} \quad \rho^\varphi(\xi(x_0, \mathbf{u}^0, \mathbf{w}), 0)$ ✸

      s.t. $\mathbf{w}^1 \models \varphi_e$

    if $\rho^\varphi(\xi(x_0, \mathbf{u}^0, \mathbf{w}^1)) > 0$ **then**

      Return $\mathbf{u}^0$

    else

      $W_{cand} \leftarrow W_{cand} \cup \{\mathbf{w}^1\}$

    end if

  end while

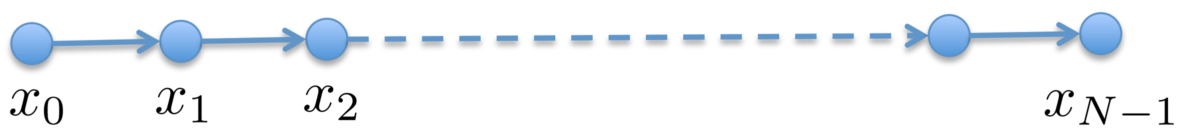
end procedure

Success!

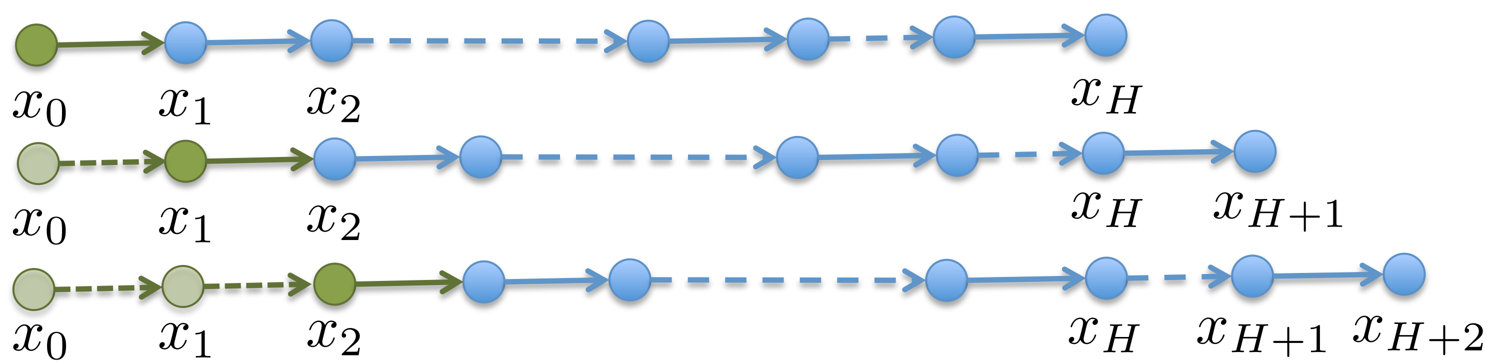
✸ **Solved using bounded-time open-loop synthesis, Raman et al, CDC'14**

Receding Horizon Synthesis

- Open-loop: generate a finite signal



- Receding horizon (MPC): do this repeatedly



STL Receding Horizon Reactive Synthesis

Given:

Continuous-time system $\dot{x} = f(x, u, w)$

STL specification $\varphi = \varphi_e \implies \varphi_s$

Initial state x_0

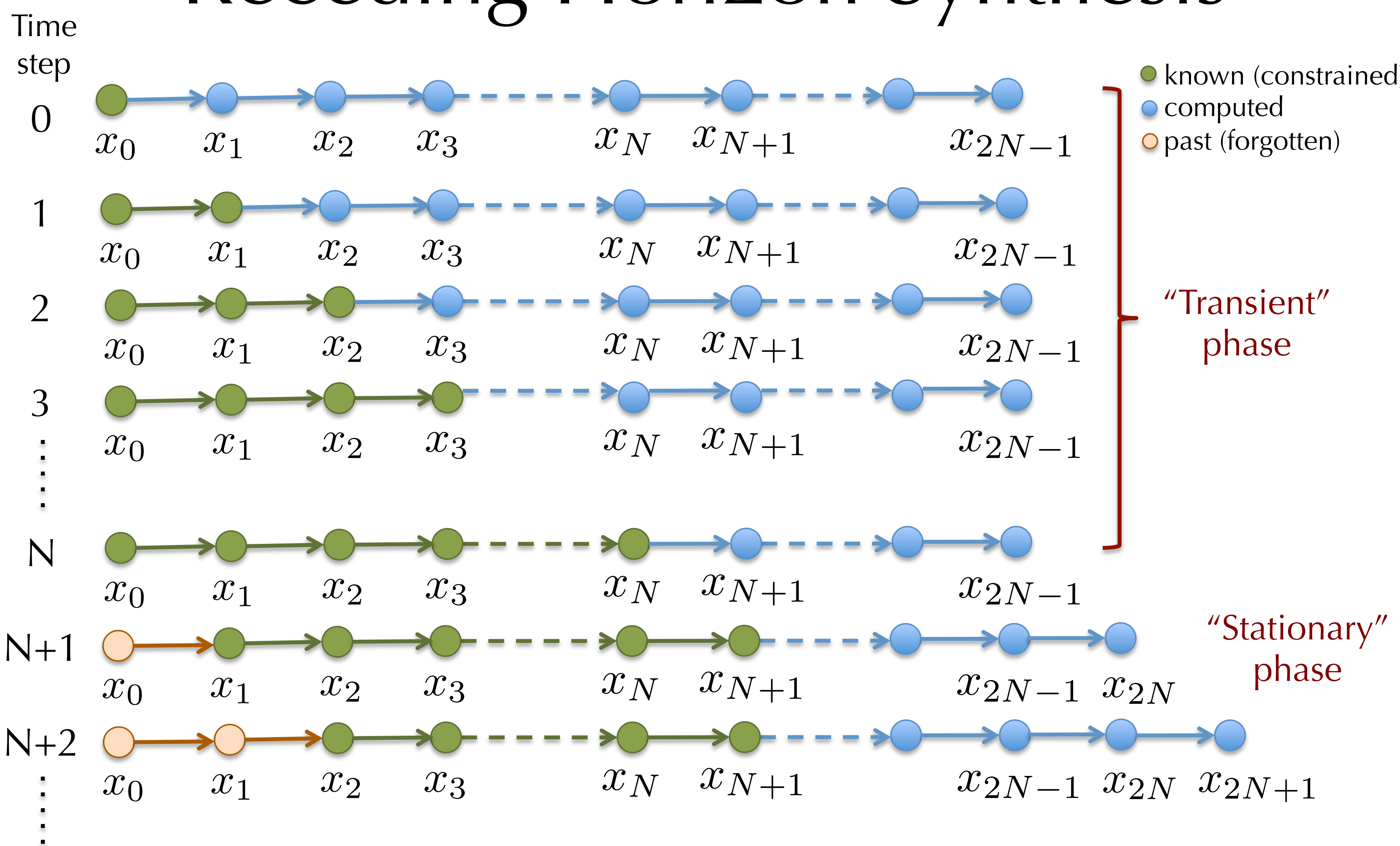
Cost function J on runs

Horizon H

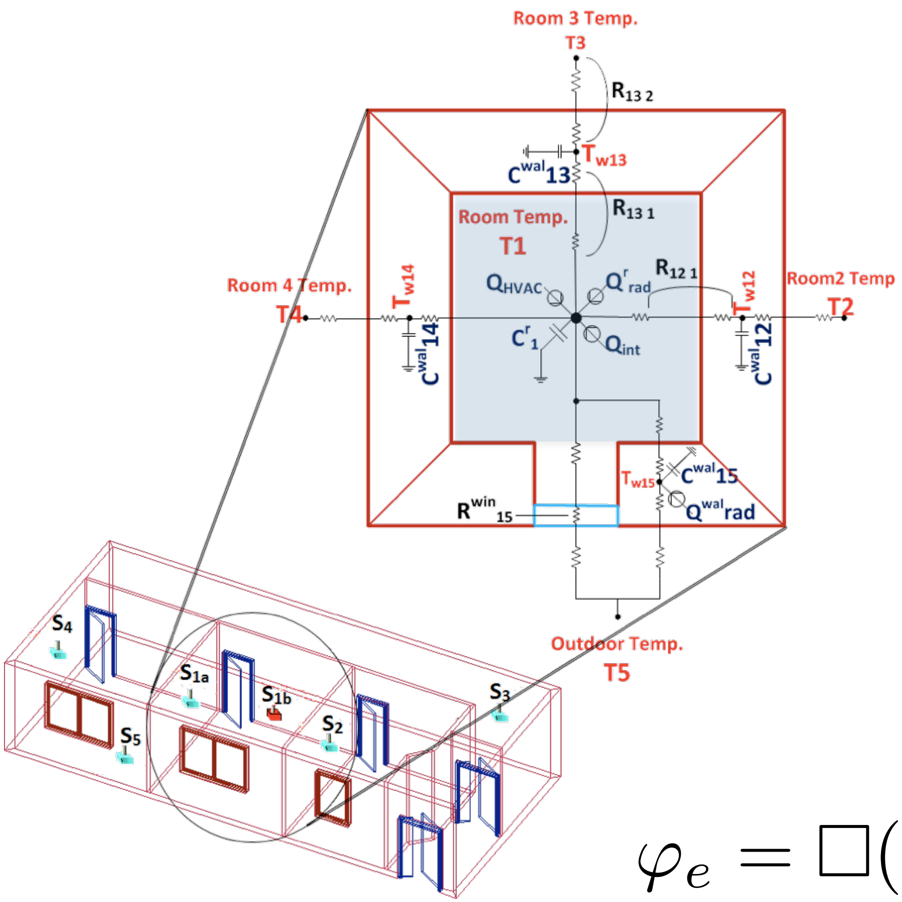
Compute:

$$\begin{aligned} \underset{\mathbf{u}^{H,k}}{\operatorname{argmin}} \quad & \max_{\mathbf{w}^{H,k} \in \{\mathbf{w} \in W^H \mid \mathbf{w} \models \varphi_e\}} \quad J(\xi(x_k, \mathbf{u}^{H,k}, \mathbf{w}^{H,k})) \\ \text{s.t.} \quad & \forall \mathbf{w} \in W^\omega, \quad \xi(x_0, \mathbf{u}, \mathbf{w}) \models \varphi \end{aligned}$$

Receding Horizon Synthesis



Case Study: HVAC Control



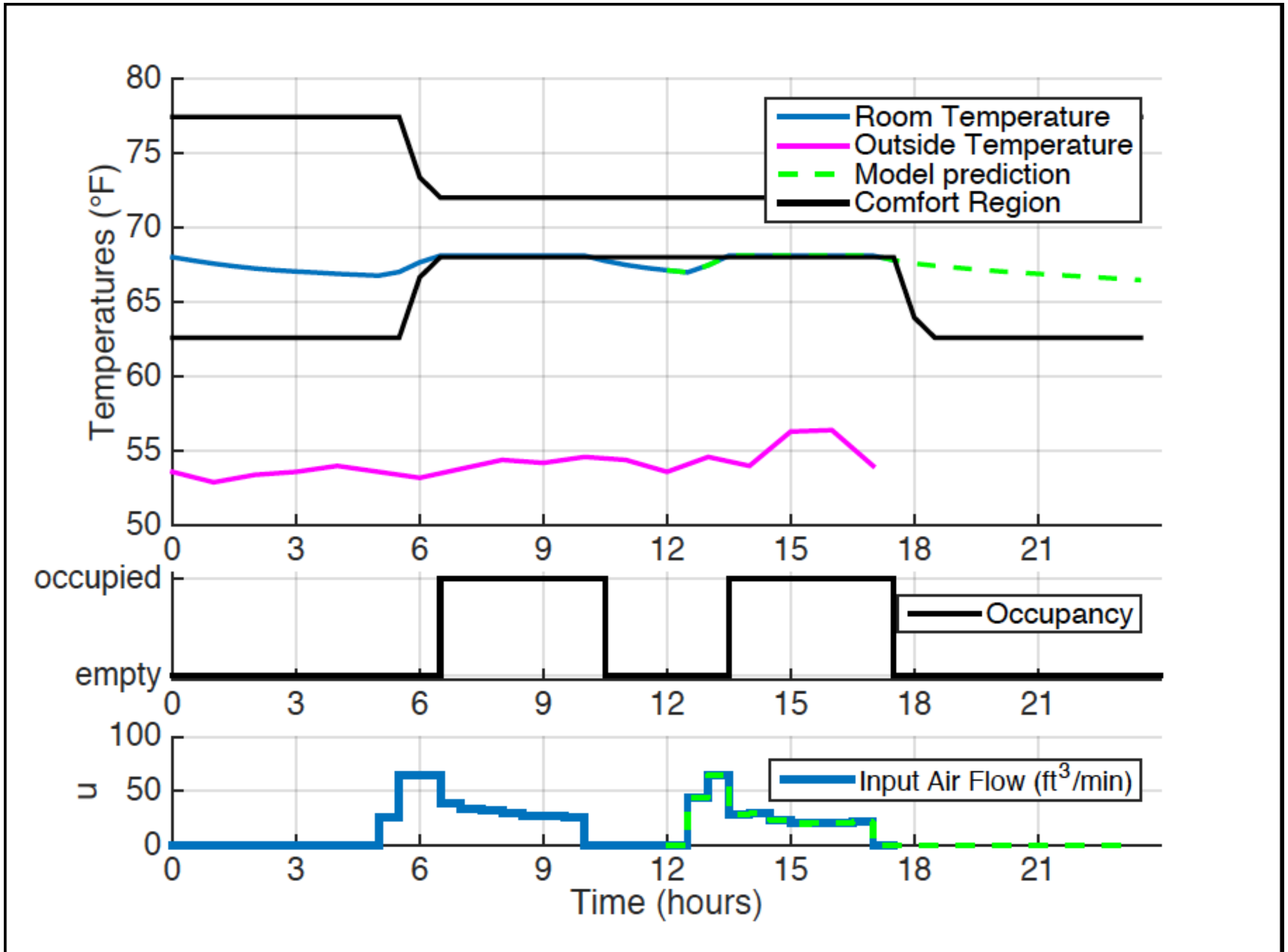
Minimize the input (air flow)

subject to

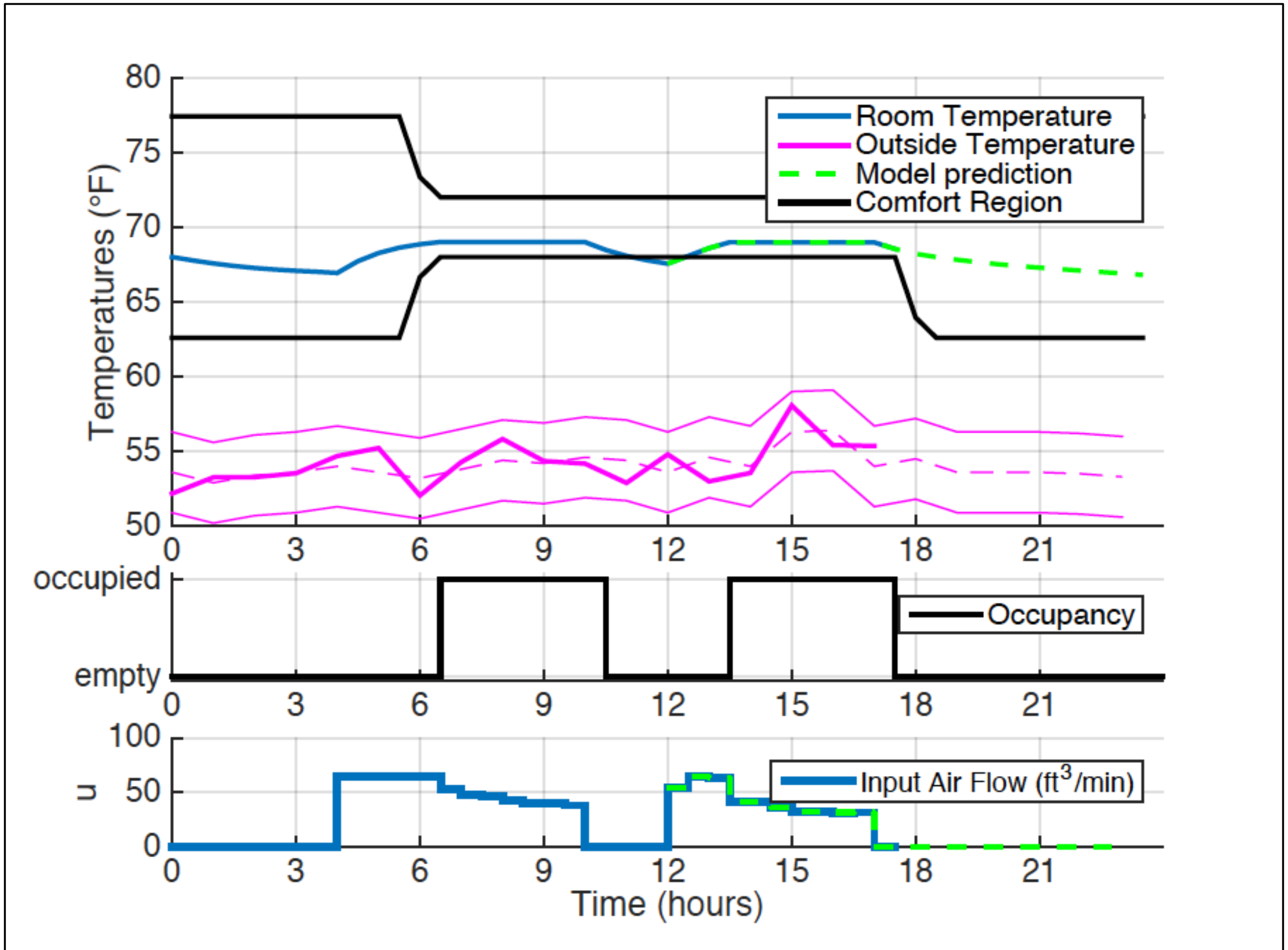
"If the occupancy of a room is > 0, the temperature should be above the comfort level"

$$\varphi_e = \Box(|\mathbf{w} - \mathbf{w}^{\text{ref}}| < 5)$$
$$\varphi_s = \Box((\text{occ}_t > 0) \Rightarrow (T_t > T_t^{\text{comfort}}))$$

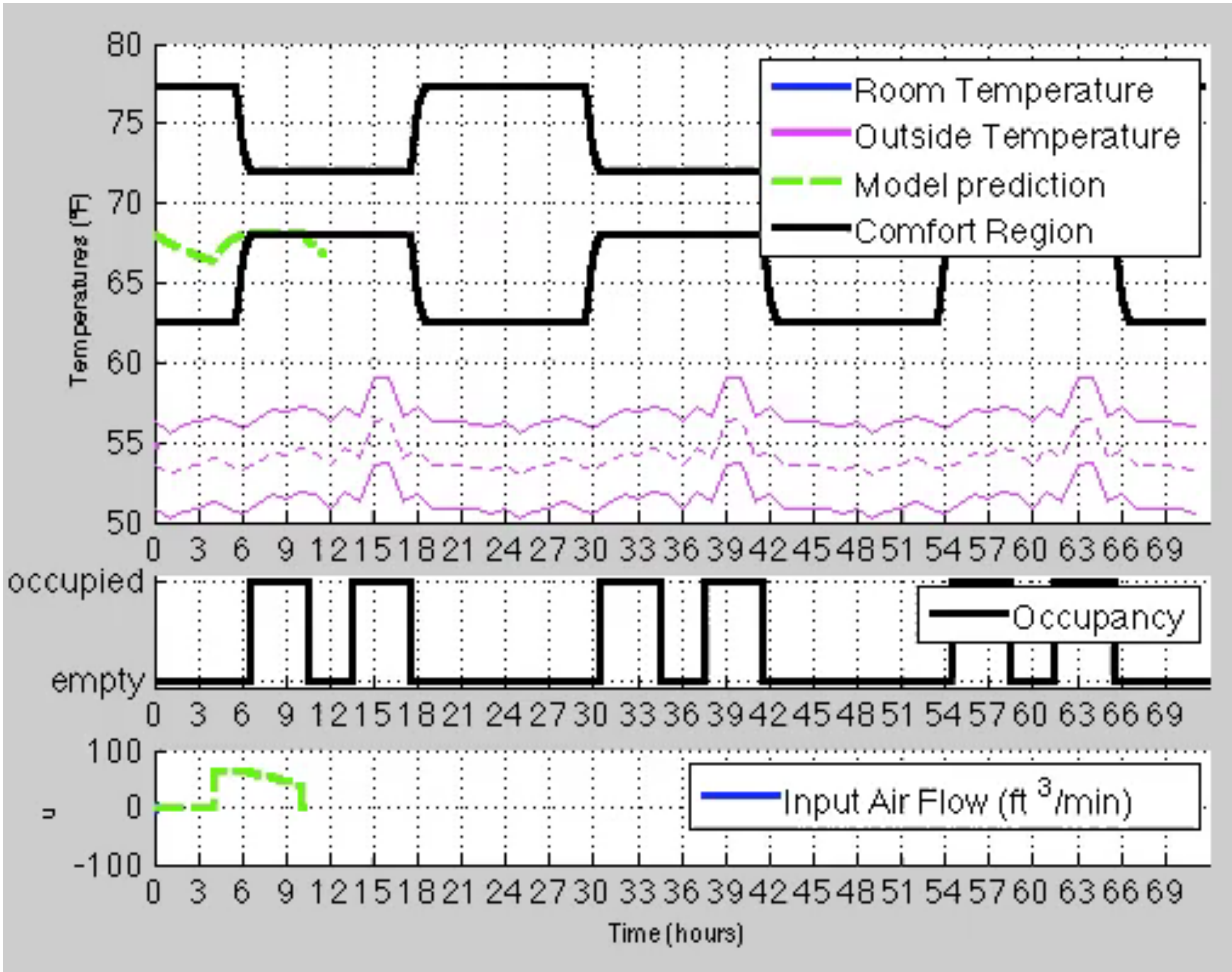
Case Study: HVAC Control



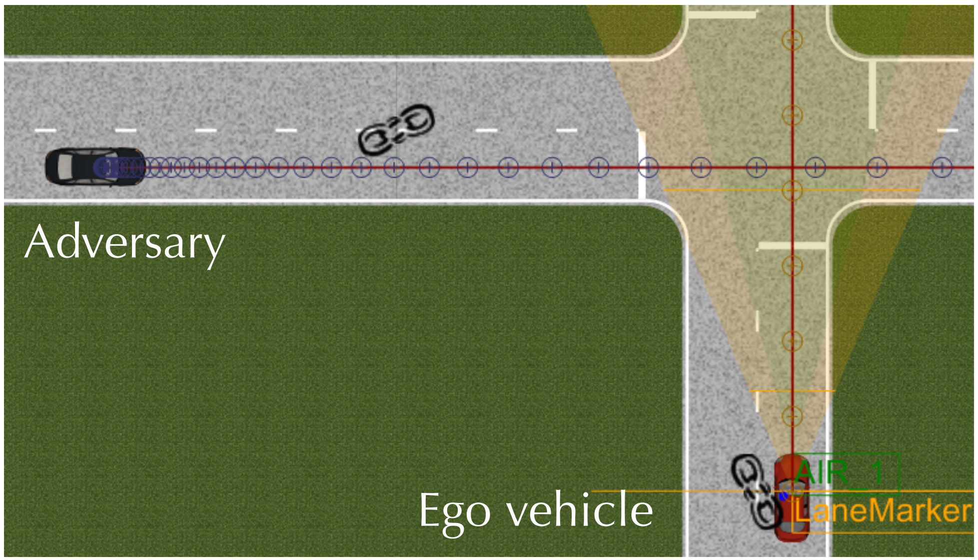
Case Study: HVAC Control



Case Study: HVAC Control



Case Study: Autonomous Driving



$$\begin{bmatrix} \dot{x}^{\text{ego}} \\ \dot{y}^{\text{ego}} \\ \dot{v}^{\text{ego}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^{\text{ego}} \\ y^{\text{ego}} \\ v^{\text{ego}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{u}$$

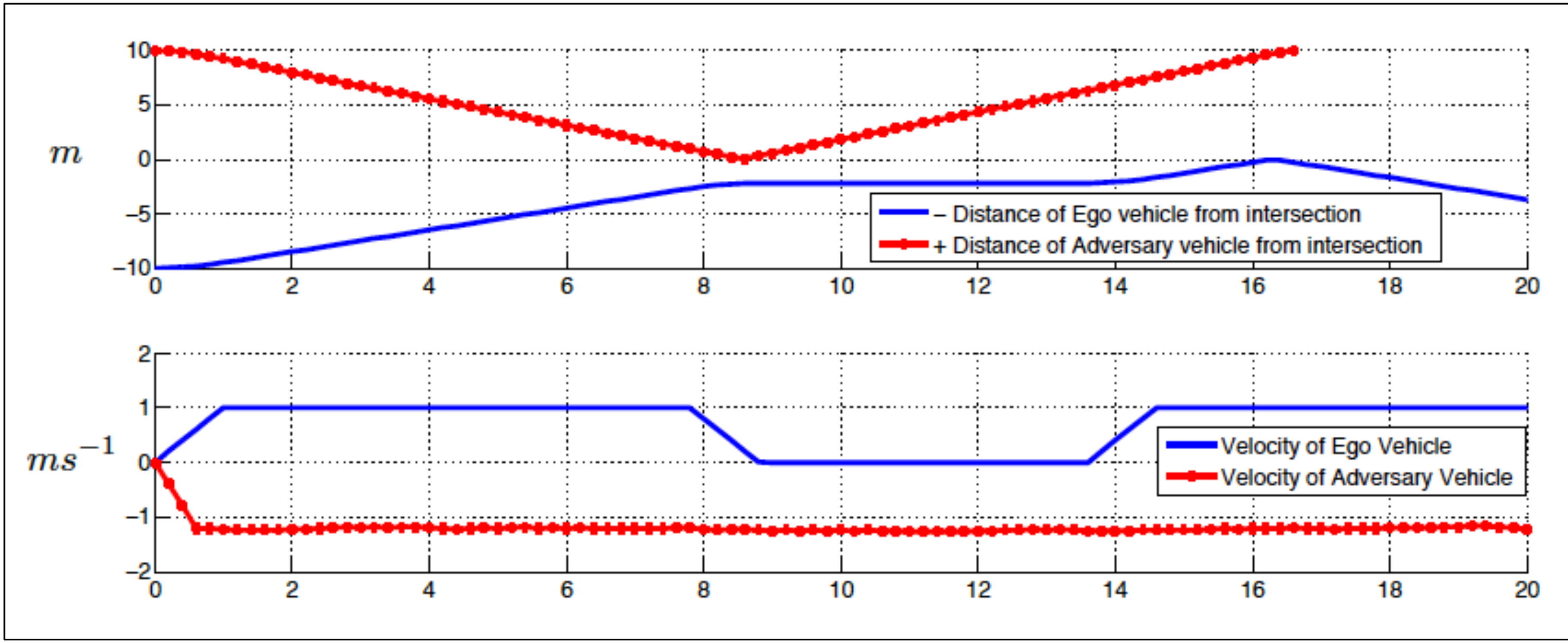
$$\begin{bmatrix} \dot{x}^{\text{adv}} \\ \dot{y}^{\text{adv}} \\ \dot{v}^{\text{adv}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^{\text{adv}} \\ y^{\text{adv}} \\ v^{\text{adv}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{w}$$

$$\varphi_e = \Box(|\mathbf{w} - \mathbf{w}^{\text{ref}}| < 0.1)$$

$$\varphi_s = \Box(|y_t^{\text{ego}} - x_t^{\text{adv}}| < 2) \Rightarrow \Box_{[0,2]}(|v_t^{\text{ego}}| < 0.1)$$

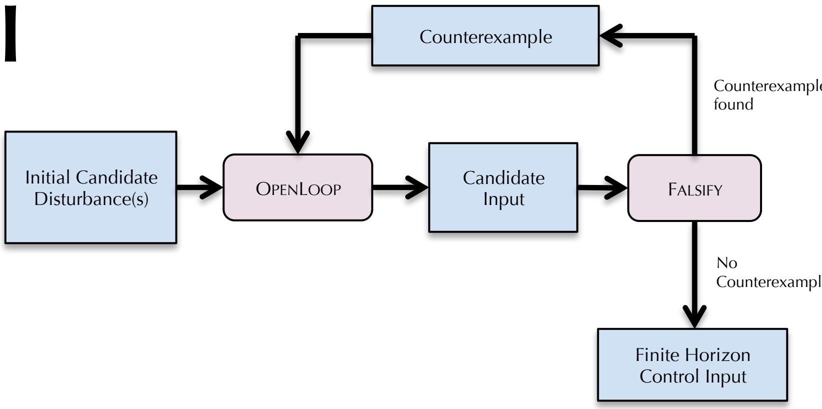
$$J(\xi(x_t, \mathbf{u}^H, \mathbf{w}^H)) = \sum_{l=0}^{H-1} |v_{t+l}^{\text{ego}} - 1|$$

Case Study: Autonomous Driving

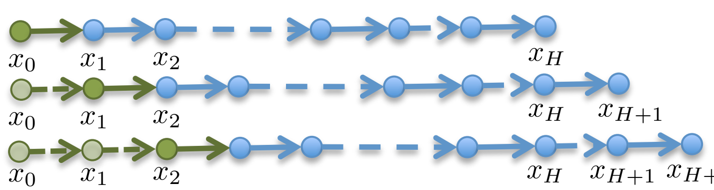


Recap + Discussion

- **Optimization-based reactive synthesis for Signal Temporal Logic**
  - No discrete abstraction
  - CEGIS scheme



- **Receding Horizon Control** for top-level unbounded "always"



- Future Work:
  - Other unbounded formulas
  - Synthesis for stochastic systems

Try it Out!

BluSTL: Controller Synthesis from Signal Temporal Logic Specifications *(presented at ARCH 2015)*

https://github.com/BluSTL/BluSTL
(BSD License)

[See *Zoolander*, 2001]

Thanks!

Reactive Synthesis from Signal Temporal Logic Specifications

Vasu Raman, Alexandre Donzé, Dorsa Sadigh,
Richard M. Murray, Sanjit A. Seshia

Contact: vasu@caltech.edu

https://github.com/BluSTL/BluSTL

HSCC

16 April 2015