Model Predictive Control with Signal Temporal Logic Specifications

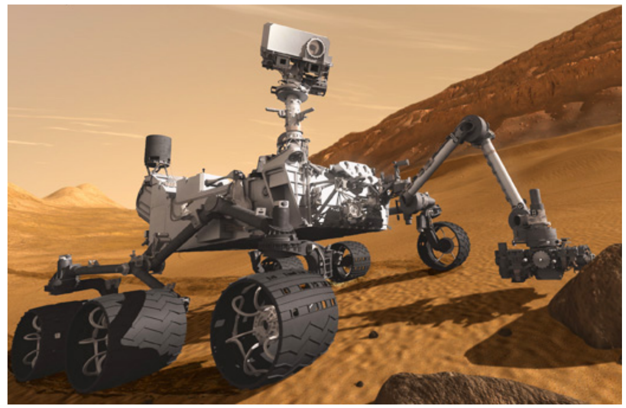
Vasu Raman[1], Alexandre Donzé[2], Mehdi Maasoumy[2]
Richard M. Murray[1], Alberto Sangiovanni-Vincentelli[2], Sanjit A. Seshia[2]

[1]California Institute of Technology
[2]University of California at Berkeley

CDC
15 December 2014

Modern Cyber-Physical Systems

Caltech DUC vehicle



NASA/JPL-Caltech Rover

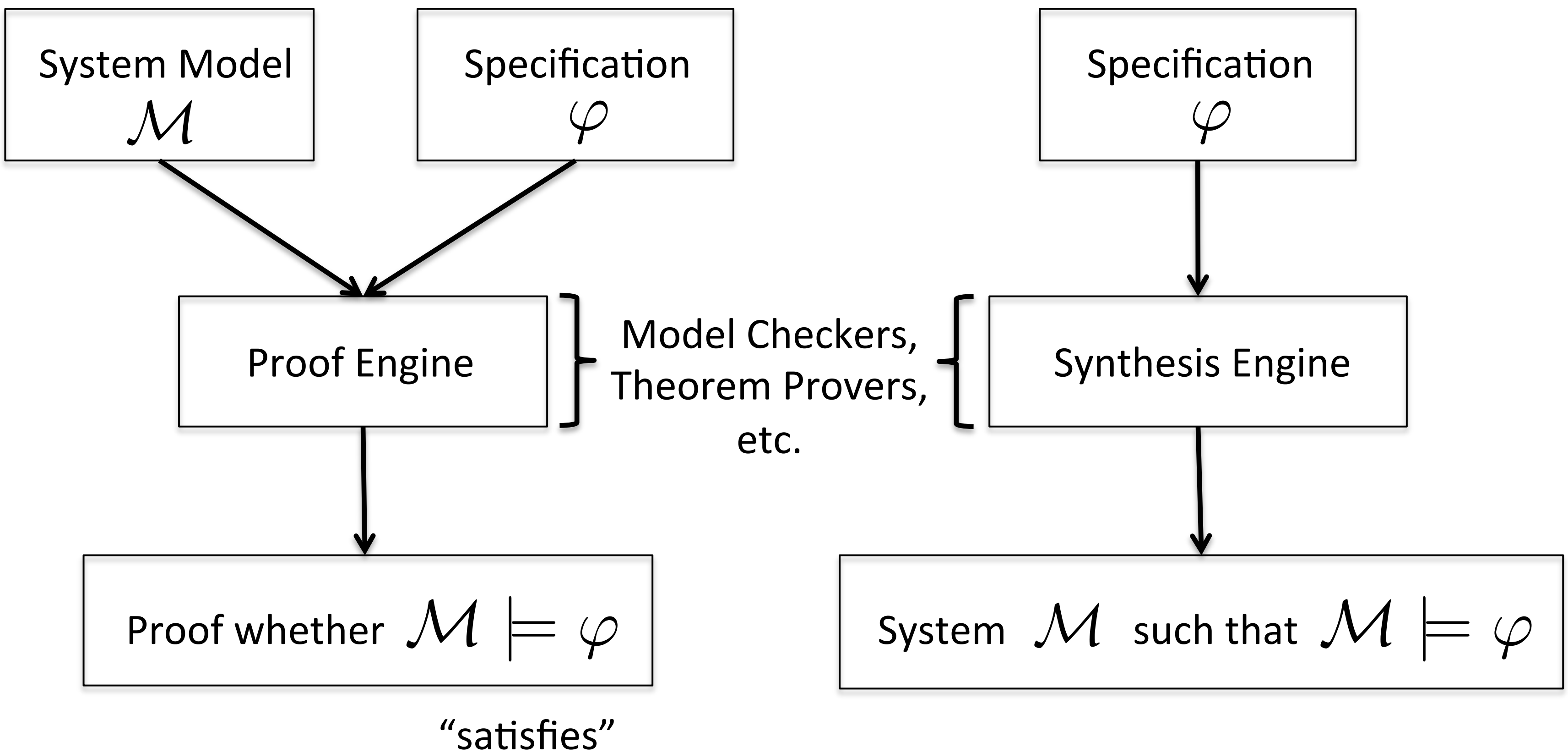
Smart Grid
(automationfederation.org)

- Operate **autonomously**
- Fulfill **complex** requirements
- Need to **specify** and **enforce** guarantees on behavior

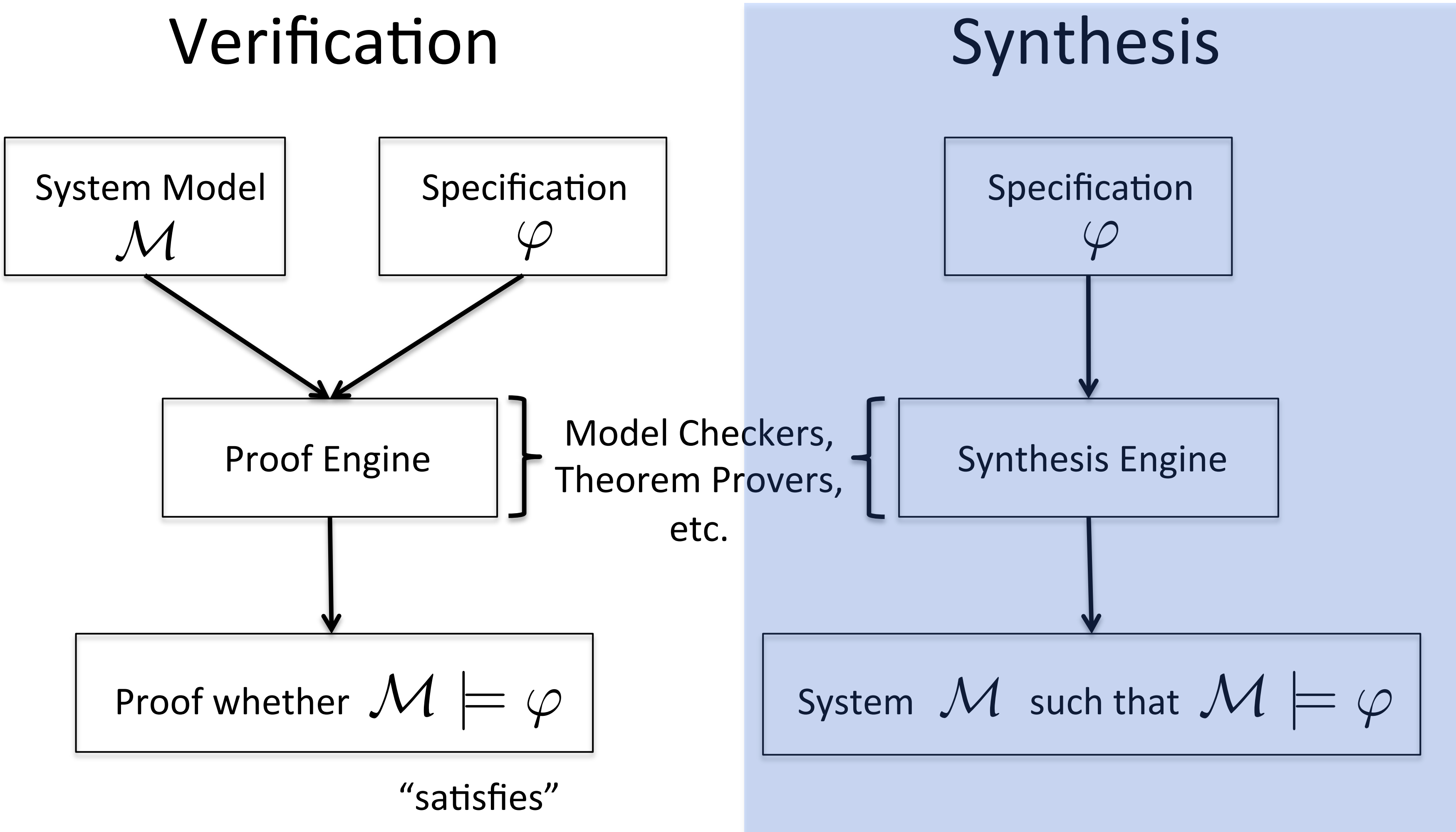
Formal Methods: Two Perspectives

Verification

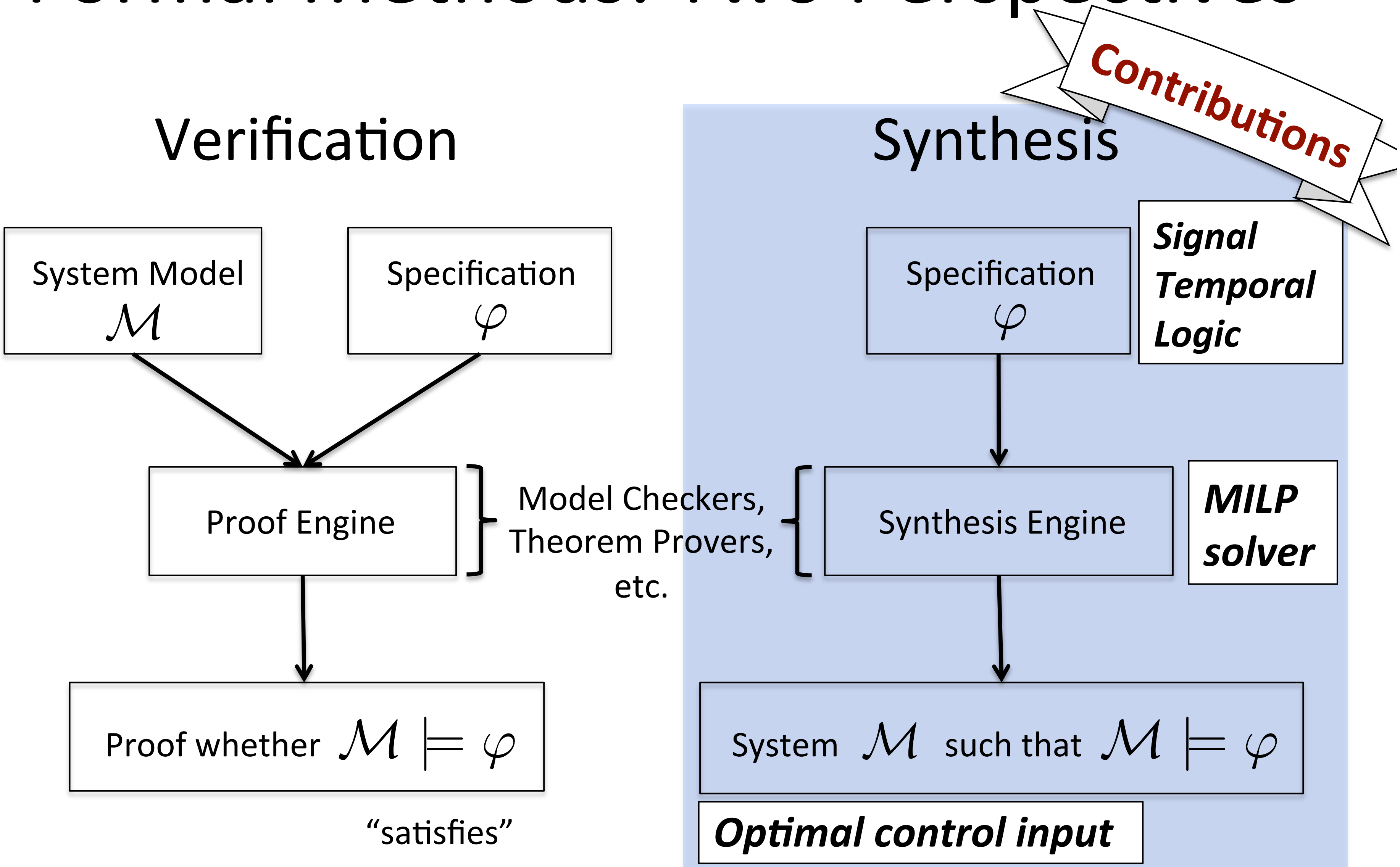
Synthesis



Formal Methods: Two Perspectives



Formal Methods: Two Perspectives



Synthesis for Cyber-Physical Systems

- Robotics
  - Kress-Gazit, Fainekos and Pappas, ICRA 2007
  - Kloetzer and Belta, TAC 2008
  - Karaman and Frazzoli, CDC 2009
  - Bhatia, Kavraki and Vardi, ICRA 2010
- Autonomous Cars
  - Wongpiromsarn, Topcu and Murray, HSCC 2010
- Aircraft Electric Power Systems
  - Nuzzo et al, IEEE Access 2013

Synthesis for Cyber-Physical Systems

- Robotics
  - Kress-Gazit, Fainekos and Pappas, ICRA 2007
  - Kloetzer and Belta, TAC 2008
  - Karaman and Frazzoli, CDC 2009
  - Bhatia, Kavraki and Vardi, ICRA 2010
- Autonomous Cars
  - Wongpiromsarn, Topcu and Murray, HSCC 2010
- Aircraft Electric Power Systems
  - Nuzzo et al, IEEE Access 2013

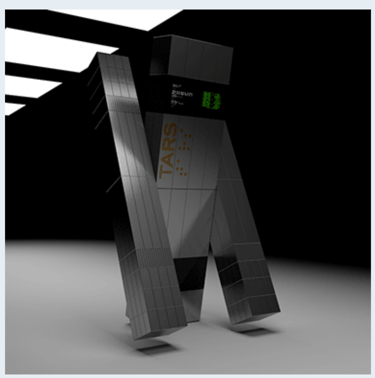
Based on **Linear Temporal Logic** Synthesis

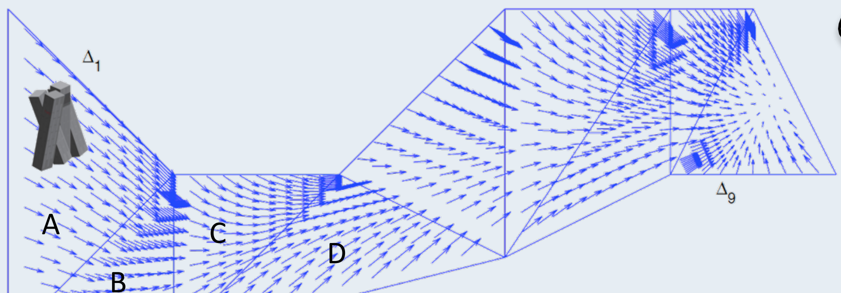
Temporal Logic Synthesis for CPS

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

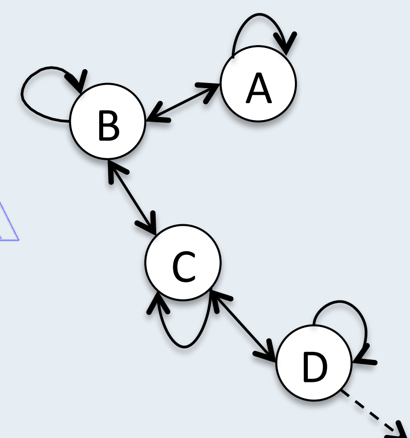
Temporal Logic Synthesis for CPS

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,…

Temporal Logic Synthesis for CPS

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

BUT

- **Discrete abstraction** is too expensive for high-dimensional dynamical systems
- Linear Temporal Logic is inconvenient for specifying
  - properties of **continuous signals**
  - **temporal duration** of events

Temporal Logic Synthesis for CPS
(What is lacking?)

- Continuous signals
  - "If temperature falls below 20°C, get it back above 20°C in the next time step"

$$\Box(\mathrm{T\_less\_than\_20} \implies \bigcirc(\neg \mathrm{T\_less\_than\_20}))$$

Temporal Logic Synthesis for CPS
(What is lacking?)

- Temporal duration
  - "Infinitely often visit A and no more than 5 time steps later visit B"

$$\square \lozenge (A \wedge \bigcirc B \vee \bigcirc \bigcirc B \vee \bigcirc \bigcirc \bigcirc B \vee \bigcirc \bigcirc \bigcirc \bigcirc B \vee \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc B)$$

  - "All visits to A and B should be no more than 5.1s apart"

$$\square (A \implies \lozenge (\text{clock\_less\_than\_5.1} \wedge B))$$

Signal Temporal Logic (STL)

[Maler and Nickovic 04]

- Continuous predicates: $\mu(\mathbf{x}) > 0$
- Boolean Operators: $\wedge, \vee, \implies, \neg$
- Bounded Temporal Operators:

ALWAYS	EVENTUALLY	UNTIL
$\Box_{[a,b]}\varphi$	$\Diamond_{[a,b]}\,\varphi$	$\varphi_1 \, \mathcal{U}_{[a,b]} \, \varphi_2$
φ holds at all $t \in [a, b]$	φ holds at some $t \in [a, b]$	

- We restrict to discretized time and linear predicates

Examples

- If temperature falls below 20°C, get it back above 20°C within 5 time steps

$$\Box(\mathrm{T\_less\_than\_20} \implies \bigcirc(\neg \mathrm{T\_less\_than\_20}))$$

- Infinitely often visit A and no more than five time steps later visit B

$$\Box\Diamond(A \wedge \bigcirc B \vee \bigcirc\bigcirc B \vee \bigcirc\bigcirc\bigcirc B \vee \bigcirc\bigcirc\bigcirc\bigcirc B \vee \bigcirc\bigcirc\bigcirc\bigcirc\bigcirc B)$$

- All visits to A and B should be no more than 5.1 seconds steps apart

$$\Box(A \implies \Diamond(\mathrm{clock\_less\_than\_5.1} \wedge B))$$

Examples

- If temperature falls below 20°C, get it back above 20°C within 5 time steps

$$\Box(T < 20 \implies \Diamond_{[0,5]}(T > 20))$$

- Infinitely often visit A and no more than five time steps later visit B

$$\Box\Diamond(A \wedge \Diamond_{[0,5]} B)$$

- All visits to A and B should be no more than 5.1 seconds steps apart

$$\Box(A \implies \Diamond_{[0,5.1]} B)$$

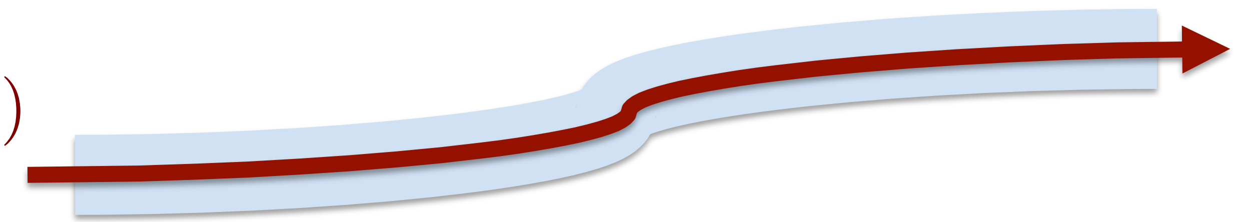
Quantitative Semantics for STL

[Donzé and Maler 10]

- In what neighborhood of the signal do we still satisfy φ?
- Robustness function $\rho^\varphi : \mathcal{X} \times \mathbb{N} \to \mathbb{R}$

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

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



- Examples: $\mu_1 \equiv x - 3 > 0 \qquad \varphi = \Box_{[0,2]}\mu_1$
  - $\rho^{\mu_1}(x, 0) = x(0) - 3$
  - $\rho^{\mu_1 \wedge \mu_2}(x, t) = \min(\rho^{\mu_1}, \rho^{\mu_2})$
  - $\rho^\varphi(x, t) = \min\limits_{t \in [0,2]} \rho^{\mu_1}(x, t) = \min\limits_{t \in [0,2]} x(t) - 3$

Optimal Control Synthesis from STL

<u>Given</u>:

Discrete time continuous system $x_{t+1} = f(x_t, u_t)$

STL specification φ

Initial state x_0

Cost function J on runs of the system

<u>Compute</u>:

$$\begin{aligned} \arg\min_{\mathbf{u}} \quad & J(\mathbf{x}(x_0, \mathbf{u}), \mathbf{u}) \\ & \text{s.t. } \mathbf{x}(x_0, \mathbf{u}) \models \varphi \end{aligned}$$

Maximally Robust Synthesis from STL

Given:

Discrete time continuous system $x_{t+1} = f(x_t, u_t)$

STL specification φ

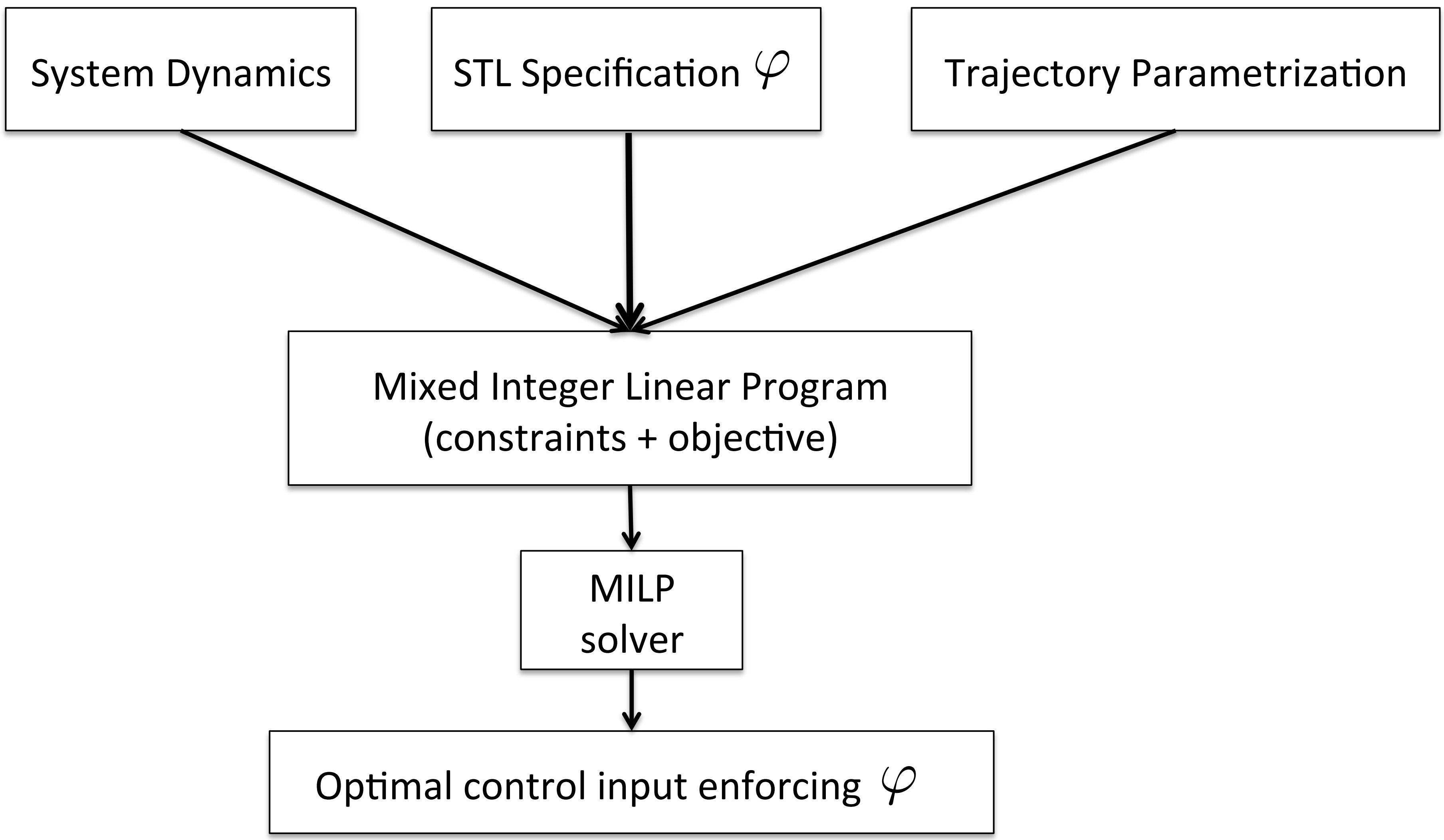
Initial state x_0

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

Compute:

$$\begin{aligned} \arg\max_{\mathbf{u}} \quad & \rho^{\varphi}(x_0, 0) \\ & \text{s.t. } \mathbf{x}(x_0, \mathbf{u}) \models \varphi \end{aligned}$$

Solution Overview



Trajectory Parametrization

- Bounded-length *N* based on formula

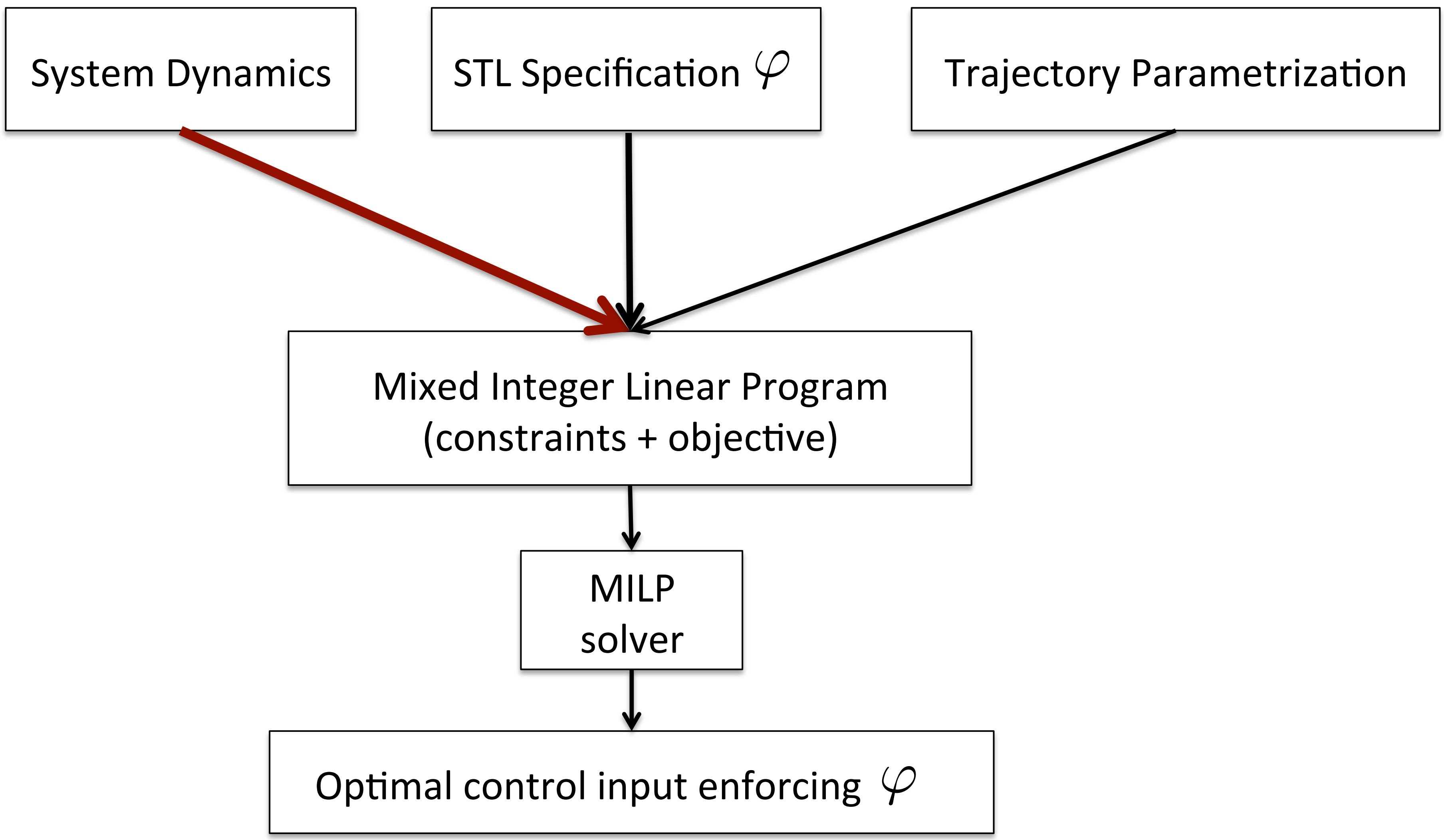
$$N \geq Bound(\varphi)$$

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

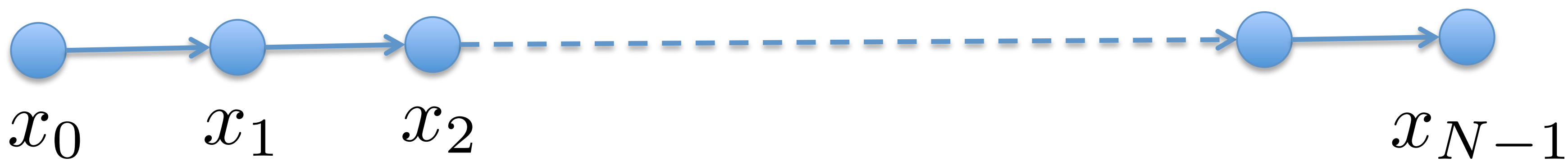
- Inspired by bounded model checking

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

Solution Overview



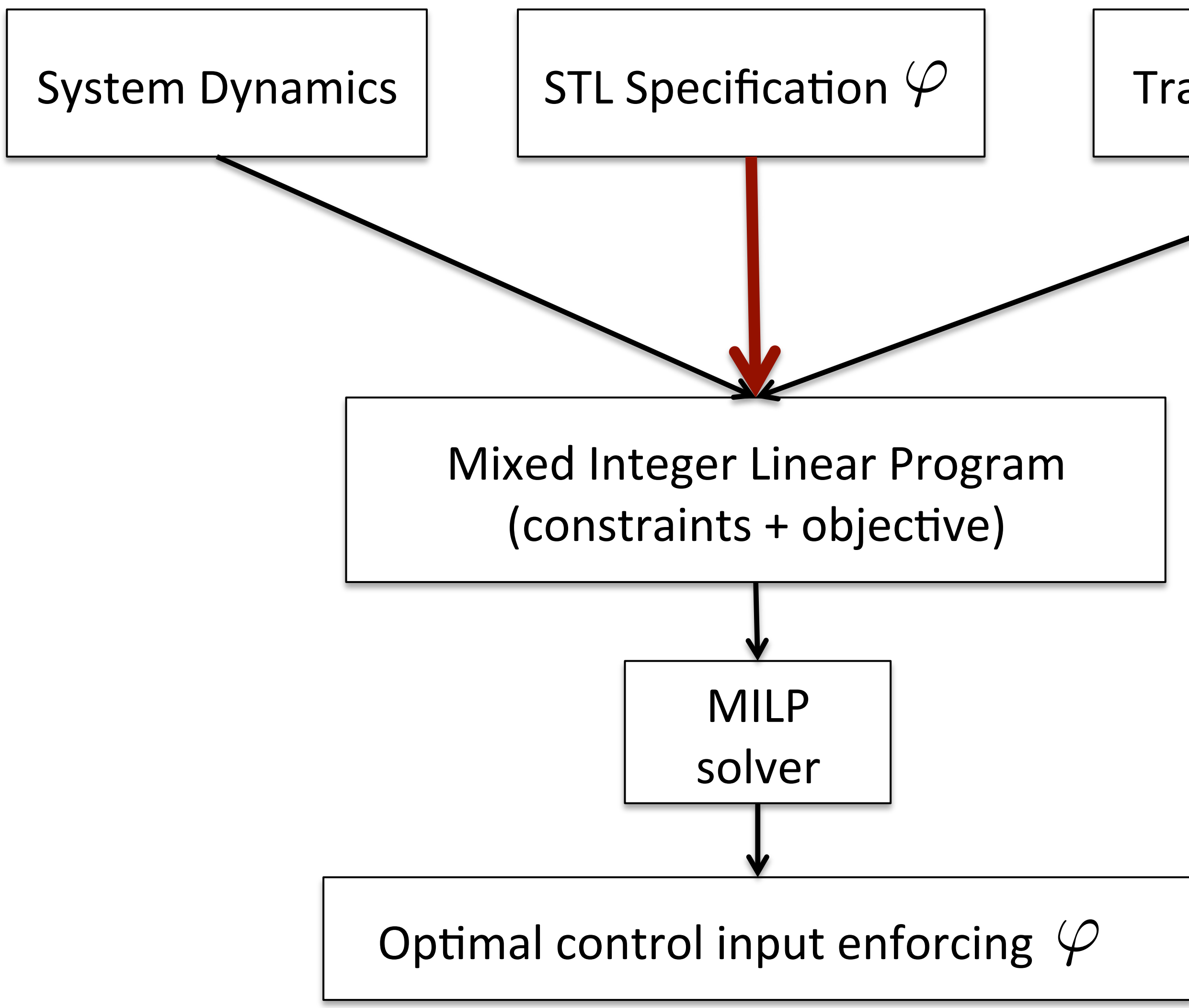
System Dynamics



$$x_{t+1} = f(x_t, u_t)$$

- Encoding similar to, e.g. [Bemporad & Morari 99]

Solution Overview



Two encodings:
- **Boolean**
- **quantitative**

Related: MILP encodings for other temporal logics
[Karaman et al. 08, Kwon and Agha 08, Wolff et al. 14]

Encoding STL as MILP Constraints

Given a formula ψ with subformulas denoted by φ

	Boolean encoding	Robustness encoding
Introduce	z_t^φ	r_t^φ
Constrained such that	$z_t^\varphi = 1 \Leftrightarrow (\mathbf{x}, t) \models \varphi$	$r_t^\varphi = \rho^\varphi(\mathbf{x}, t)$
Enforce	$z_0^\psi = 1$	$r_0^\psi > 0$

Recursively generate the MILP constraints corresponding to z_0^ψ or r_0^ψ

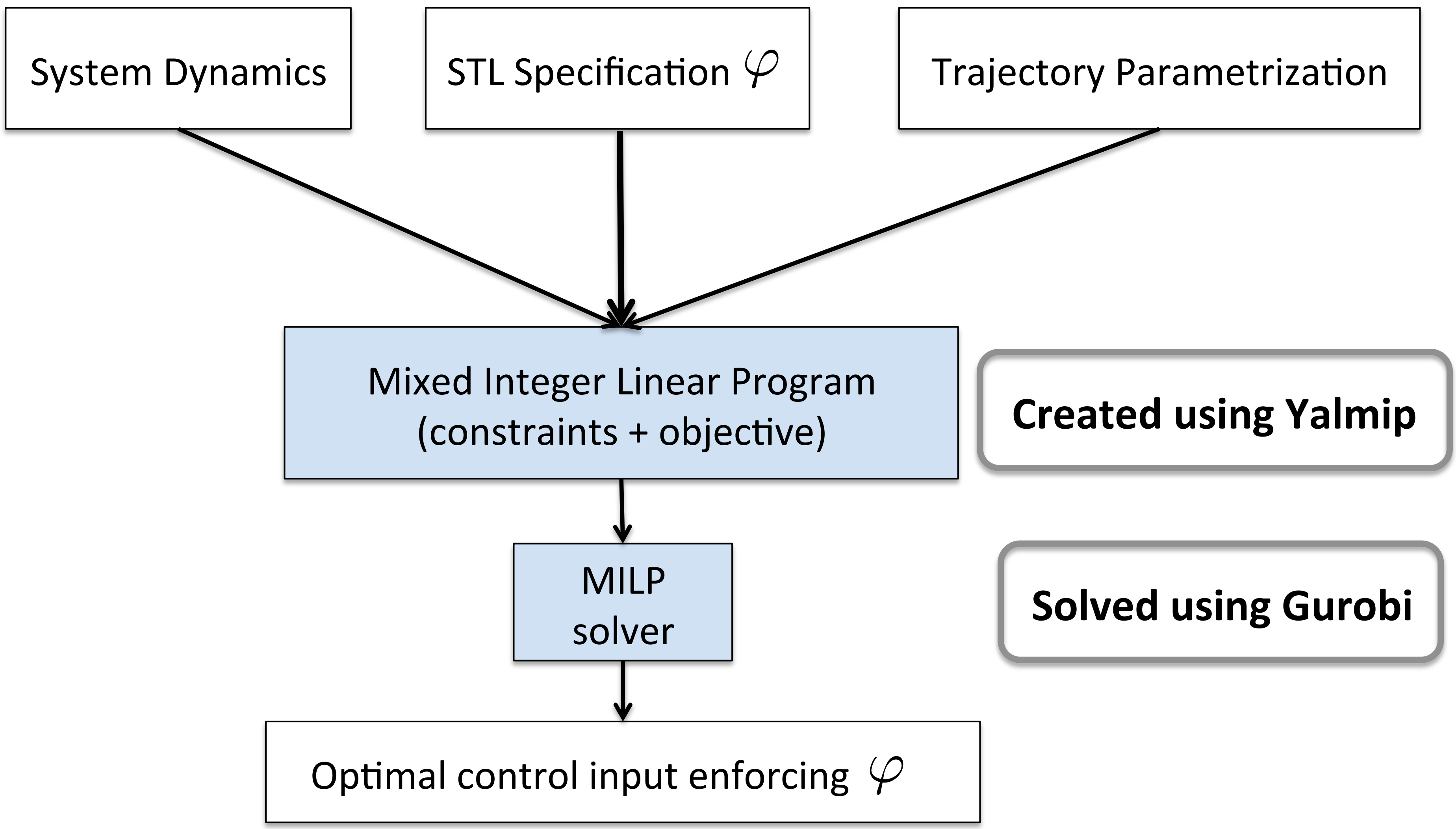
Boolean Operations

	Boolean encoding	Robustness encoding
Predicates	$\begin{array}{rcl} \mu(x_t) & \leq & M_t z_t^\mu - \epsilon_t \\ -\mu(x_t) & \leq & M_t(1 - z_t^\mu) - \epsilon_t \end{array}$	$r_t^\mu = \mu(x_t)$
Negation	$z_t^{\neg\varphi} = 1 - z_t^\varphi$	$r_t^{\neg\varphi} = -r_t^\varphi$
Conjunction $\psi = \wedge_{i=1}^m \varphi_i$	$z_t^\psi \leq z_{t_i}^{\varphi_i}, i = 1, ..., m,$ $z_t^\psi \geq 1 - m + \sum_{i=1}^m z_{t_i}^{\varphi_i}$	$\sum_{i=1}^m p_{t_i}^{\varphi_i} = 1$ $r_t^\psi \leq r_{t_i}^{\varphi_i}, i = 1, ..., m$ $r_{t_i}^{\varphi_i} - (1 - p_{t_i}^{\varphi_i})M \leq r_t^\psi$ $r_t^\psi \leq r_{t_i}^{\varphi_i} + M(1 - p_{t_i}^{\varphi_i})$

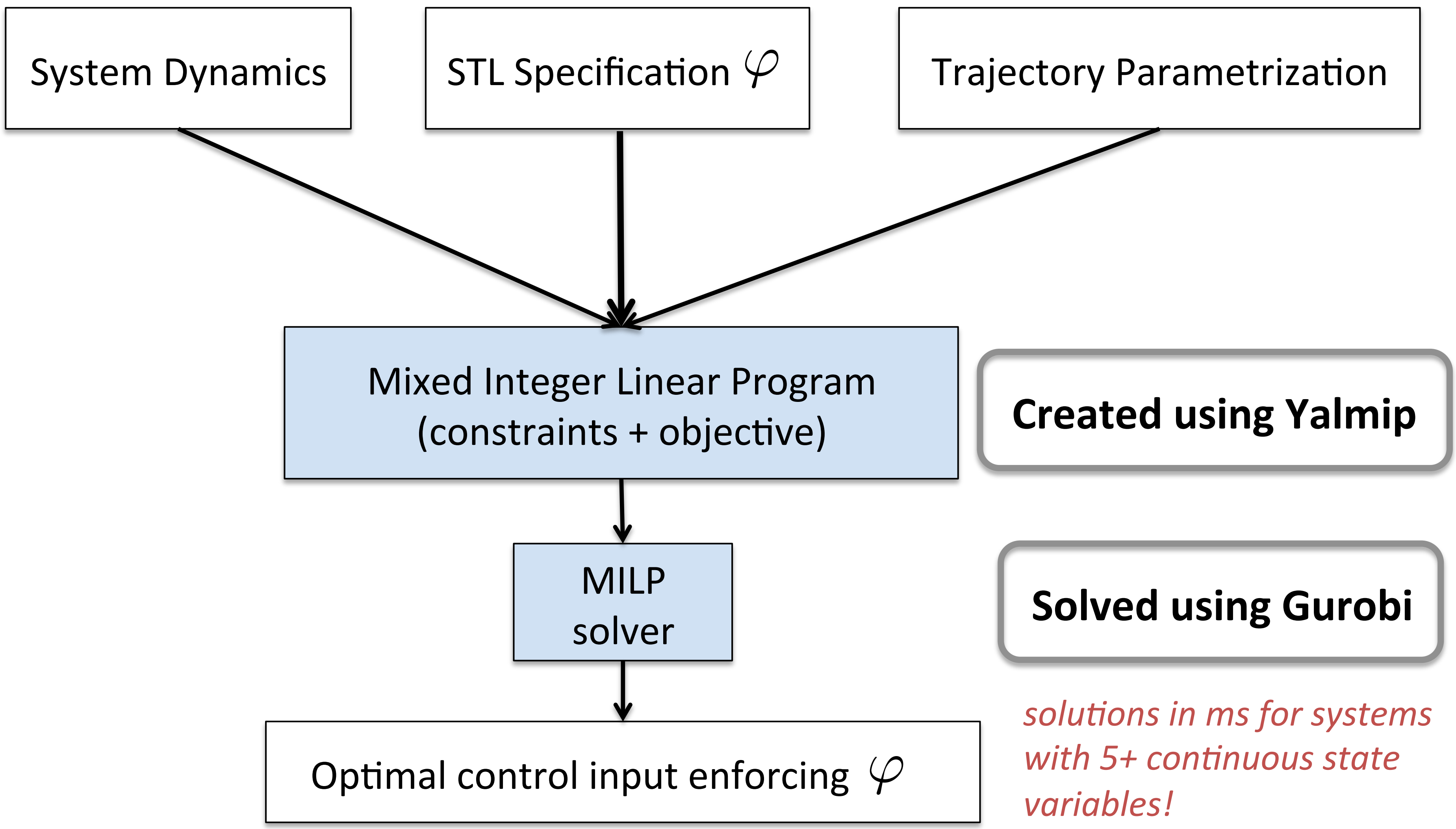
Temporal Operations

	Both encodings
Always $\psi = \Box_{[a,b]}\,\varphi$	$z_t^{\psi} = \bigwedge_{i=a_t^N}^{b_t^N} z_i^{\varphi}$
Eventually $\psi = \Diamond_{[a,b]}\,\varphi$	$z_t^{\psi} = \bigvee_{i=a_t^N}^{b_t^N} z_i^{\varphi}$
Until $\psi = \varphi_1\,\mathcal{U}_{[a,b]}\,\varphi_2$	$\varphi_1\,\mathcal{U}_{[a,b]}\,\varphi_2 = \Box_{[0,a]}\,\varphi_1 \wedge \Diamond_{[a,b]}\,\varphi_2 \wedge \Diamond_{[a,a]}(\varphi_1\,\mathcal{U}\,\varphi_2)$

Solution Overview



Solution Overview



Runtimes

$$\mathbf{x} = \mathbf{u} \quad x_t = x_t^{(1)} x_t^{(2)} x_t^{(3)} \quad J(\mathbf{x}, \mathbf{u}) = \sum_{k=1}^{N} \|u_{t_k}\|_1$$

$$\varphi_1 = \square_{[0,0.1]}\, x_t^{(1)} > 0.1$$

$$\varphi_2 = \square_{[0,0.1]}(x_t^{(1)} > 0.1) \wedge \square_{[0,0.1]}(x_t^{(2)} < -0.5)$$

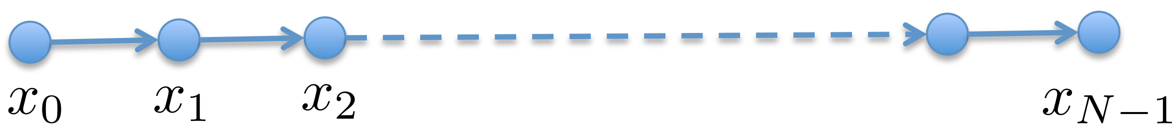
$$\varphi_3 = \square_{[0,0.5]} \lozenge_{[0,0.1]}(x_t^{(1)} > 0.1)$$

$$\varphi_4 = \lozenge_{[0,0.2]}(x_t^{(1)} > 0.1 \wedge (\lozenge_{[0,0.1]}(x_t^{(2)} > 0.1) \wedge \lozenge_{[0,0.1]}(x_t^{(3)} > 0.1)))$$

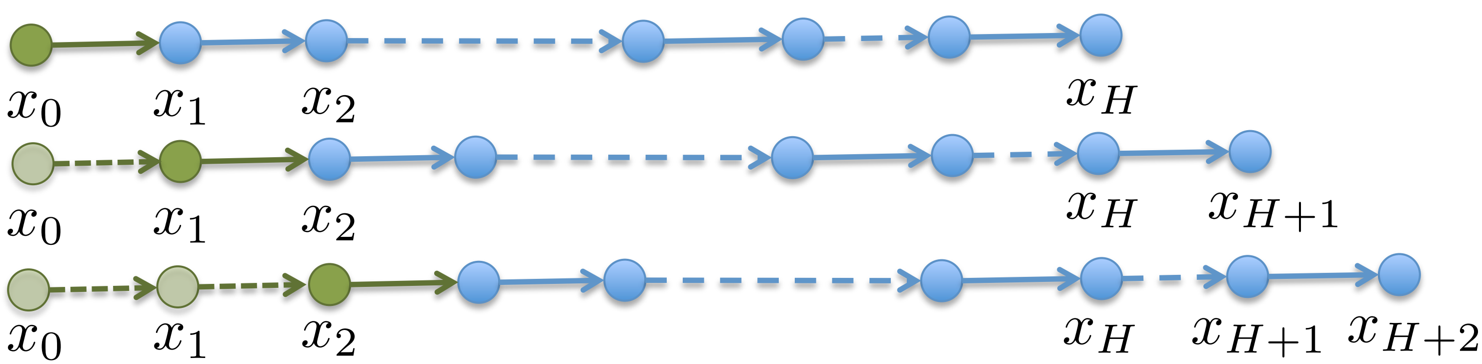
Formula	#constraints		Yalmip Time (s)		Solver time (s)	
φ_1	154	488	1.71	2.04	0.0070	0.0085
φ_2	364	897	1.94	2.69	0.0115	0.0229
φ_3	244	1282	1.84	3.15	0.0064	0.1356
φ_4	574	1330	2.29	3.37	0.2167	238.6
			Boolean	**Robustness**		

Model Predictive Control

- So far: generated open-loop finite trajectory



- Better: repeatedly generate a finite trajectory



- Related: MPC for mixed logical dynamical systems

[Bemporad and Morari 99]

Optimal Control Synthesis from STL

<u>Given</u>:

Discrete time continuous system $x_{t+1} = f(x_t, u_t)$

STL specification φ

Initial state x_0

Cost function J on runs of the system

<u>Compute</u>:

$$\arg\min_{\mathbf{u}} \quad J(\mathbf{x}(x_0, \mathbf{u}), \mathbf{u})$$
$$\text{s.t. } \mathbf{x}(x_0, \mathbf{u}) \models \varphi$$

Model Predictive Control from STL

Given:

Discrete time continuous system $x_{t+1} = f(x_t, u_t)$

STL specification φ

Initial state x_0

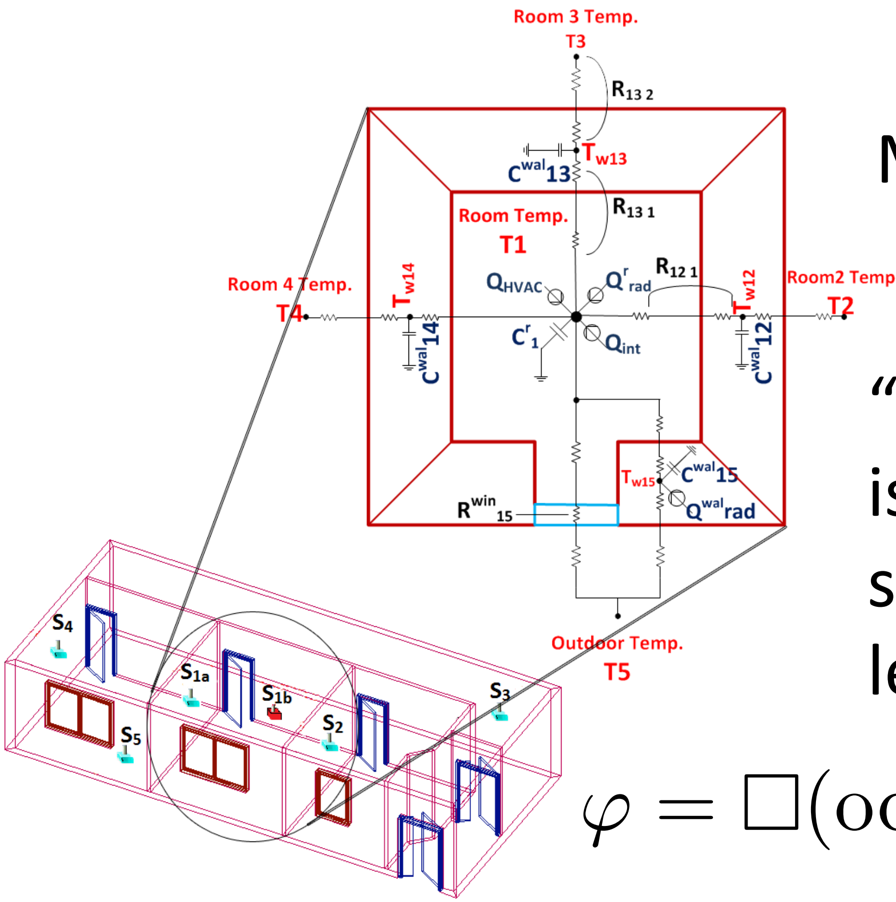
Cost function J on runs of the system

Horizon *H*

Compute:

$$\arg\min_{\mathbf{u}_t^H} \quad J(\mathbf{x}^H(x_t, \mathbf{u}_t^H), \mathbf{u}_t^H))$$
$$\text{s.t. } \mathbf{x}(x_0, \mathbf{u}) \models \varphi,$$

Example: HVAC system



Minimize the input (air flow)

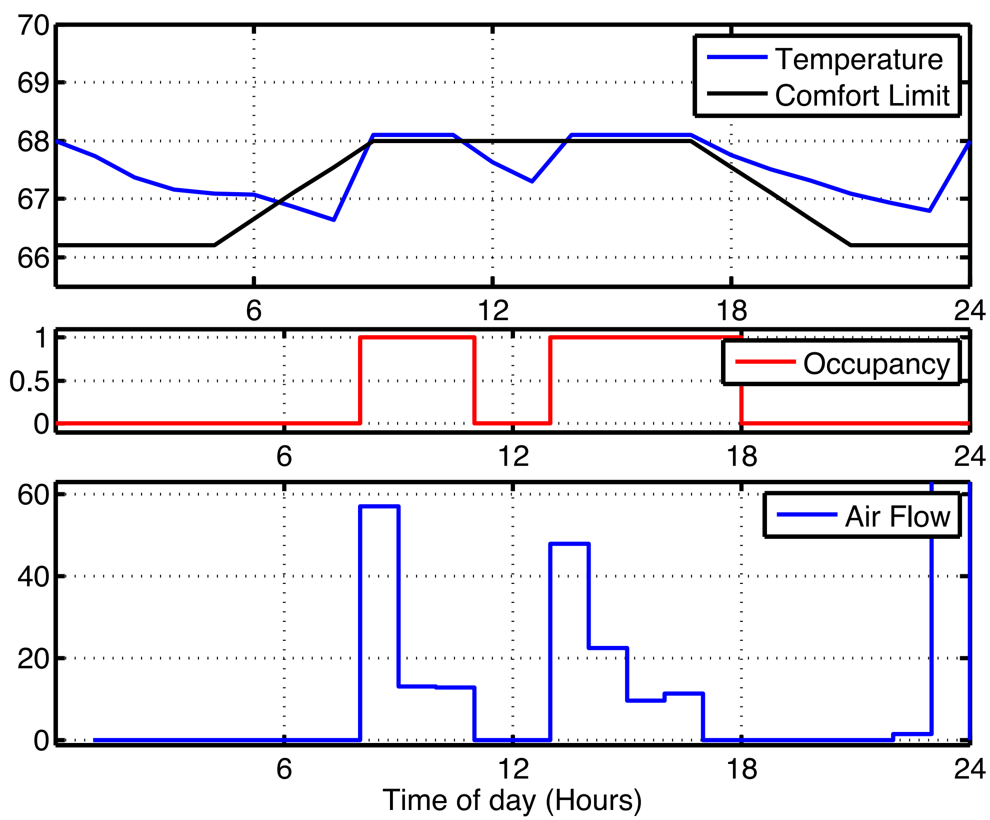
subject to

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

$$\varphi = \Box(\mathrm{occ}_t > 0) \Rightarrow (T_t > T_t^{\mathrm{comfort}}))$$

Example: HVAC system

$$\varphi = \Box_{[0,H]}(\mathrm{occ}_t > 0) \Rightarrow (T_t > T_t^{\mathrm{comfort}}))$$



Model Predictive Control

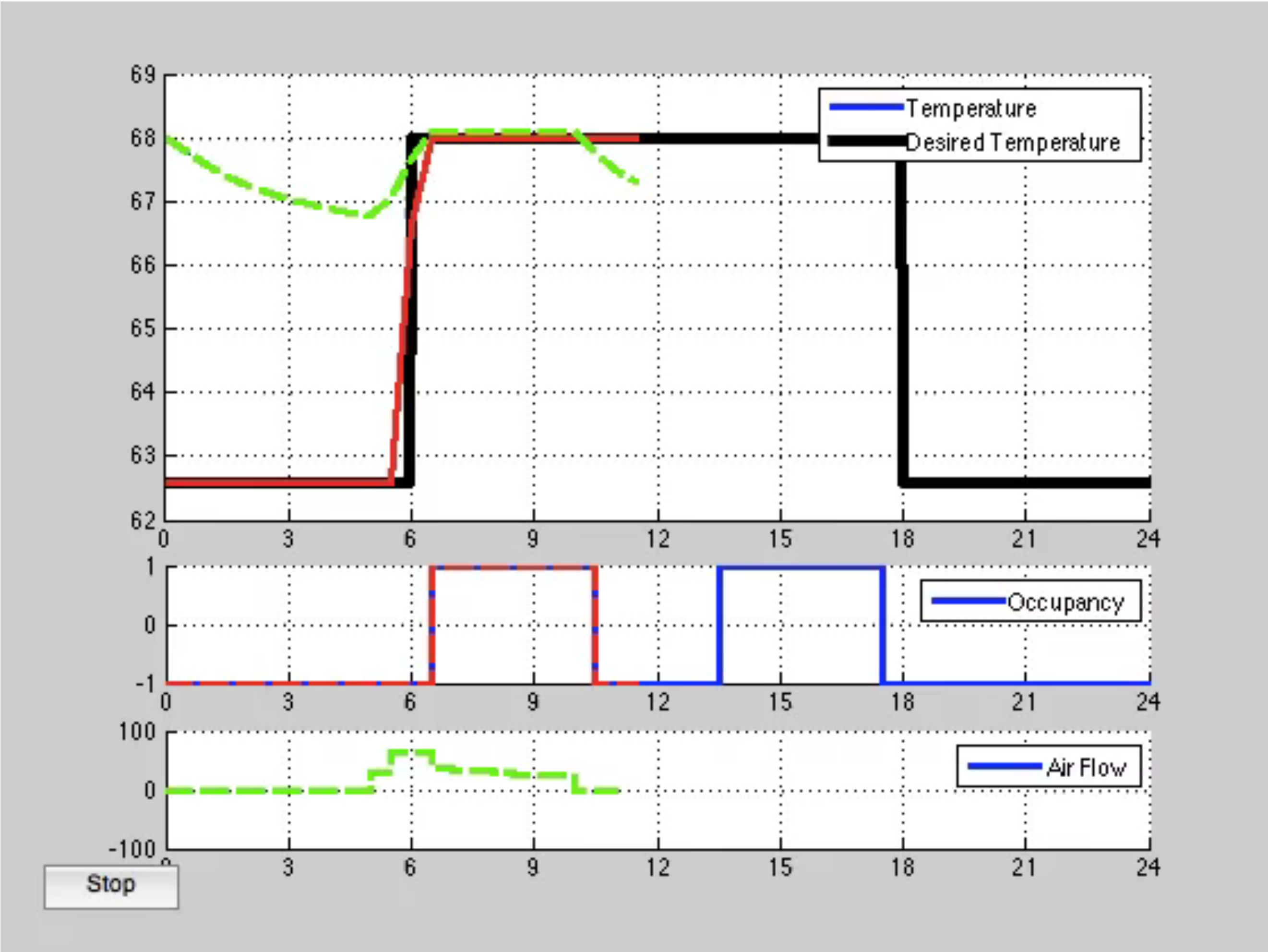
$$\min_{\vec{u}_t} \sum_{k=0}^{H-1} \|u_{t+l}\| \quad \text{s.t.}$$

$$x_{t+k+1} = f(x_{t+k}, u_{t+k}, d_{t+k}),$$

$$x_t \models \varphi$$

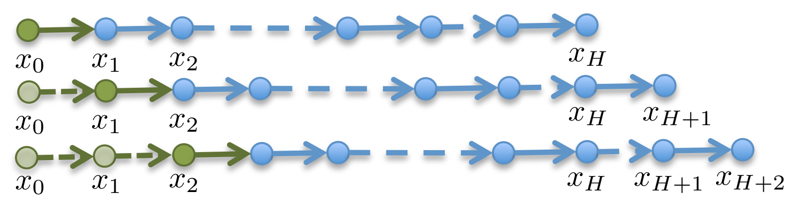
$$u_{t+k} \in \mathcal{U}_{t+k}, \;\; k = 0, \ldots, H-1$$

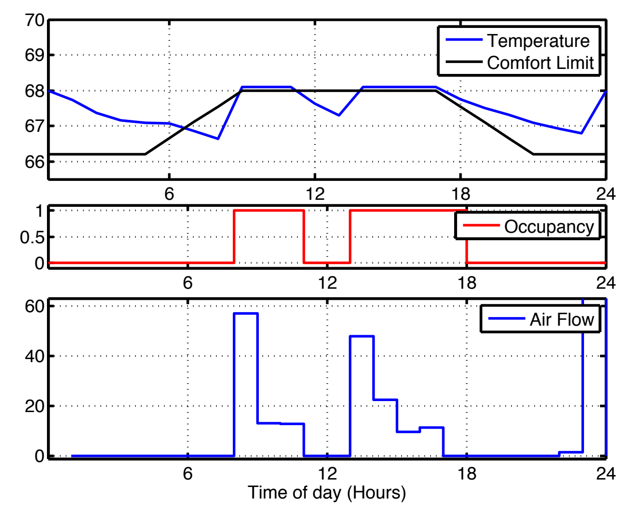
Example: HVAC system



Conclusions

- **Optimization-based synthesis for Signal Temporal Logic**
  - No discrete abstraction
  - Quantitative satisfaction
- **Model Predictive Control** for robustness and scalability
- Future Work:
  - uncertain dynamics/adversarial environment (reactive synthesis)
  - stochastic systems





Thanks!

Model Predictive Control with Signal Temporal Logic Specifications

Vasu Raman, Alexandre Donzé, Mehdi Maasoumy,
Richard M. Murray, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia

Contact: vasu@caltech.edu

CDC
15 December 2014