

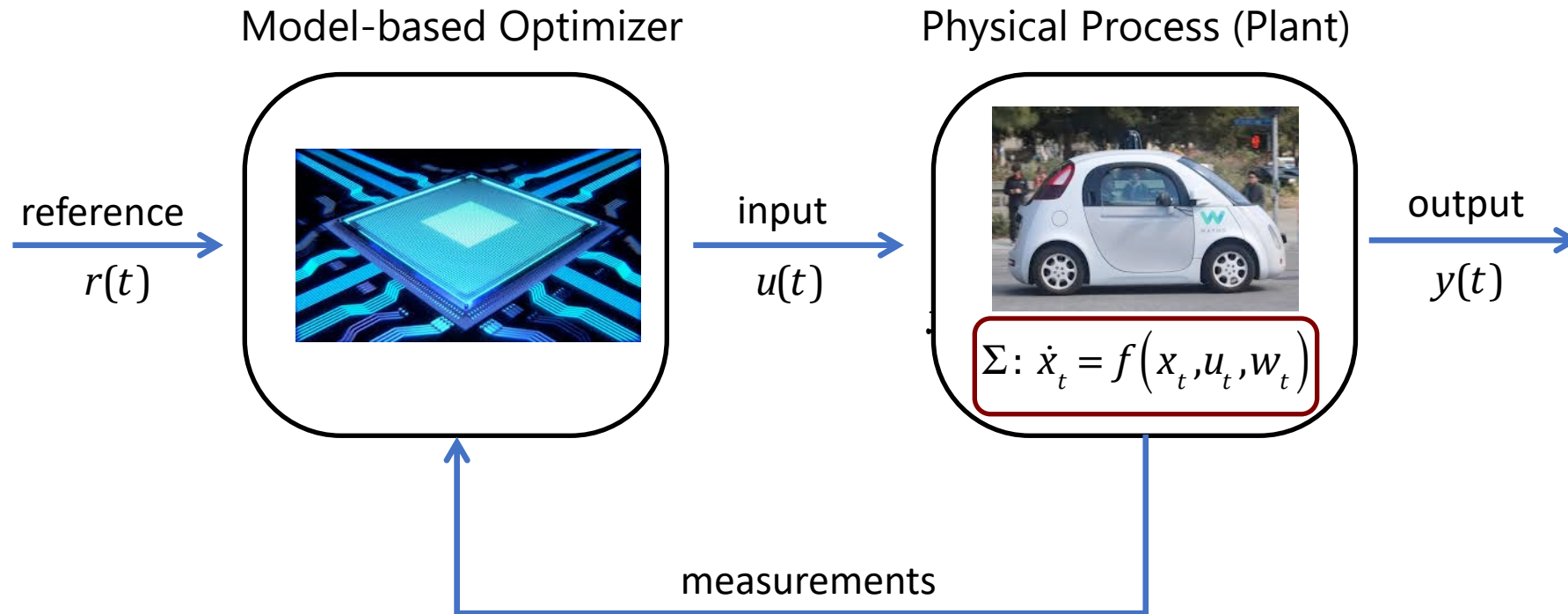
# Cyber-Physical Systems

Laura Nenzi

Università degli Studi di Trieste  
I Semestre 2023

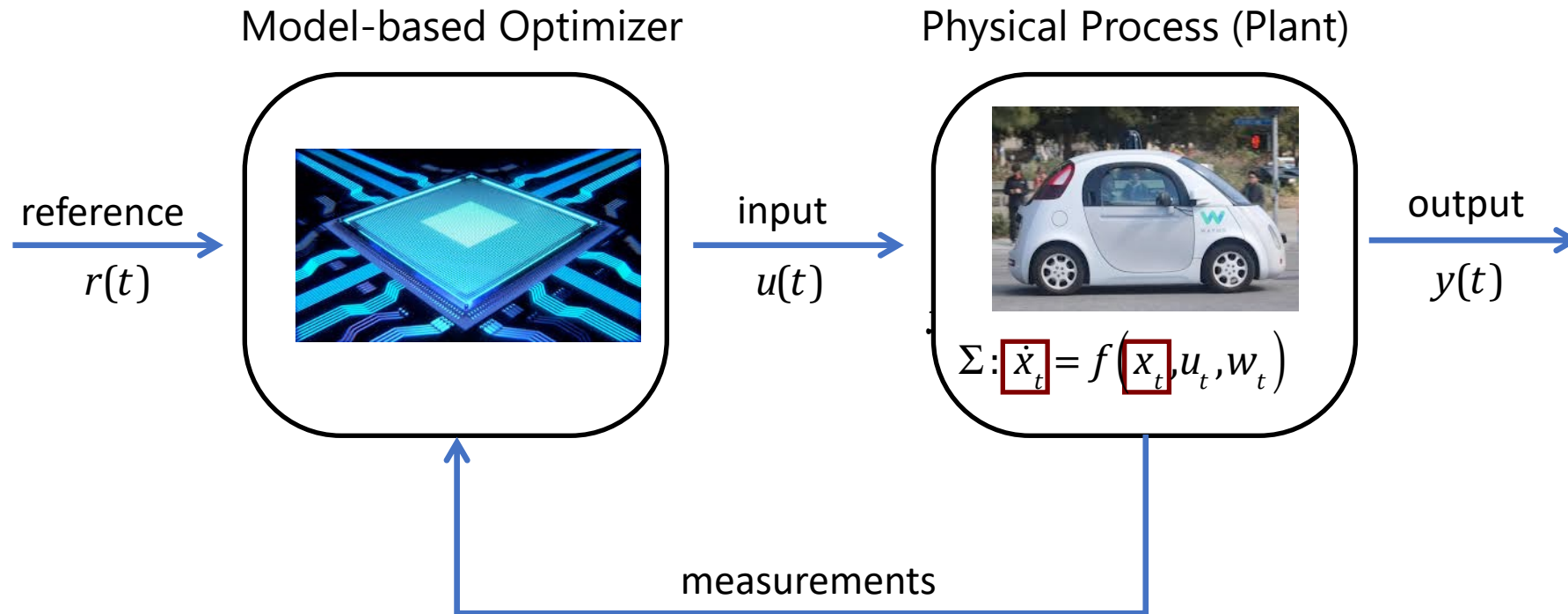
## Lecture 24: Control Synthesis with STL

# Model Predictive Control



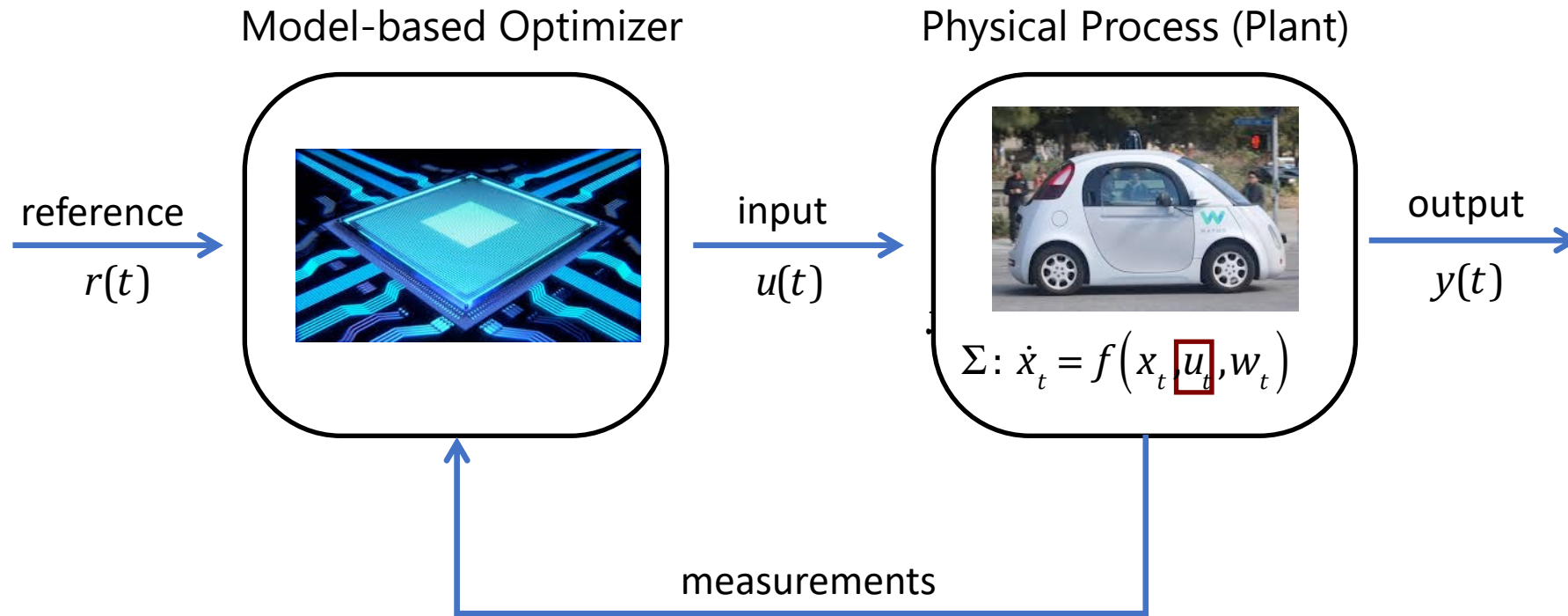
**MDP uses the dynamical model of the process to predict its future evolution and optimize consequently the control input signal**

# Model Predictive Control



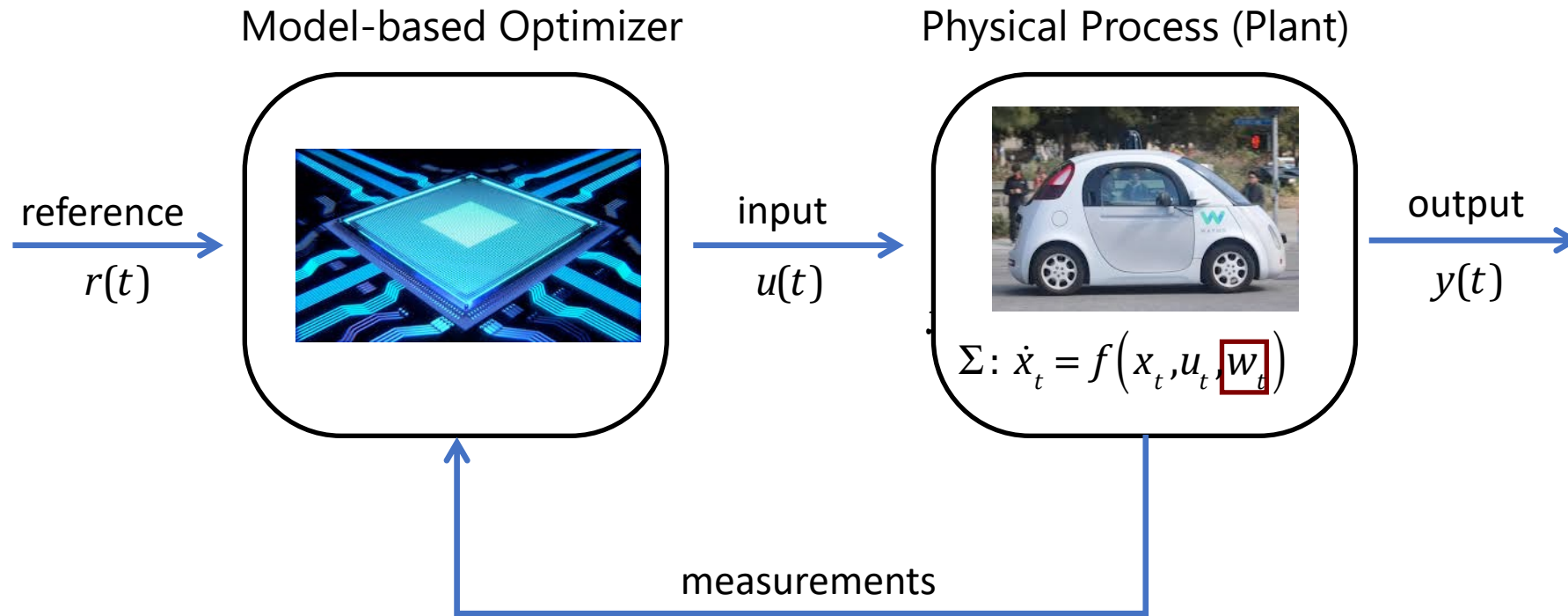
$\dot{x}_t \in X \subseteq \left( \mathbb{R}^{n_c} \times \{0,1\}^{n_l} \right)$  are continuous/logical states

# Model Predictive Control



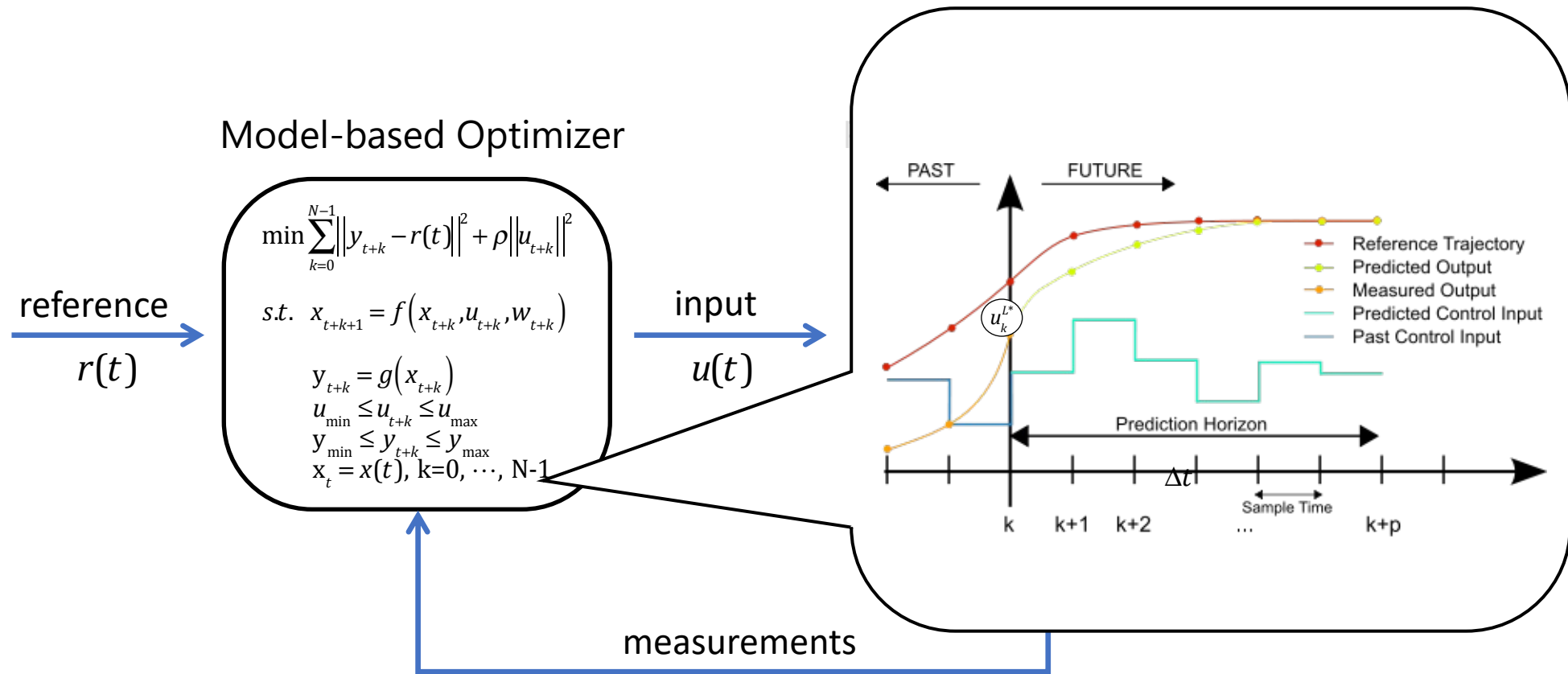
$\dot{u}_t \in U \subseteq \left( \mathbb{R}^{m_c} \times \{0,1\}^{m_l} \right)$  are continuous/logical inputs

# Model Predictive Control



$w_t \in W \subseteq \left( \mathbb{R}^{e_c} \times \{0,1\}^{e_l} \right)$  are the external environmental inputs

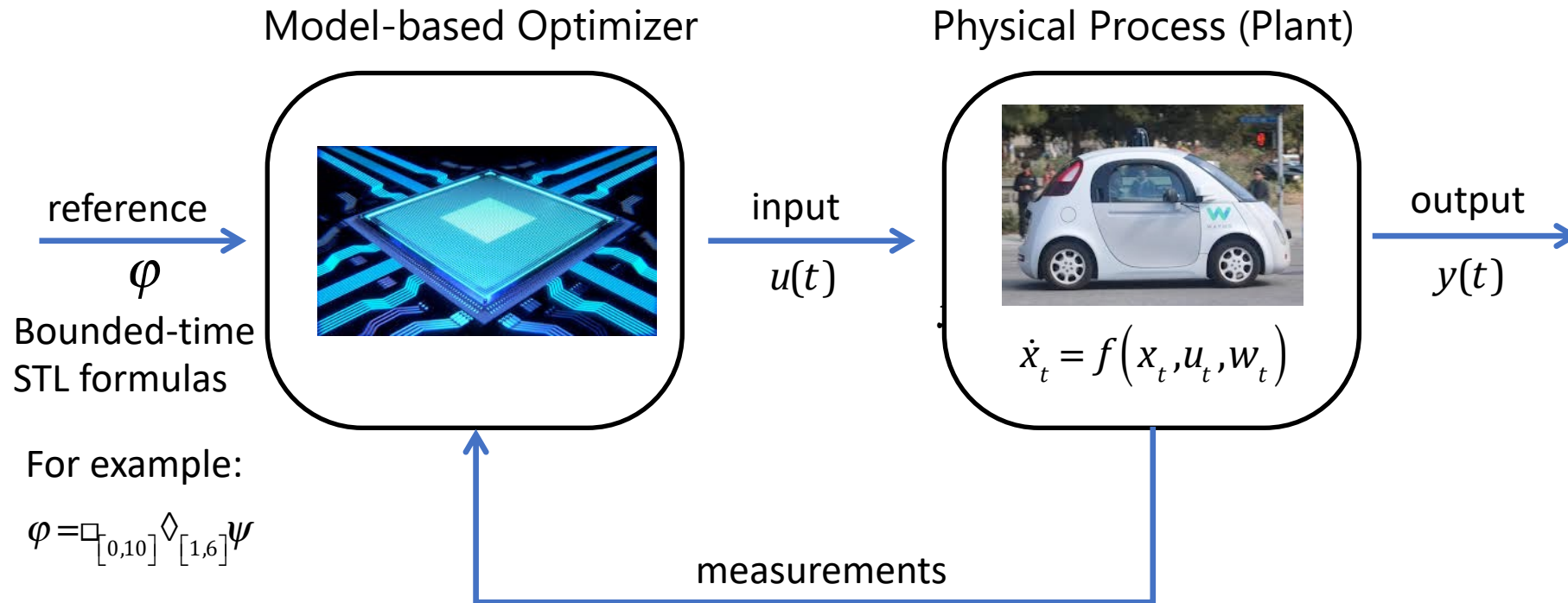
# Example of Receding Horizon Control



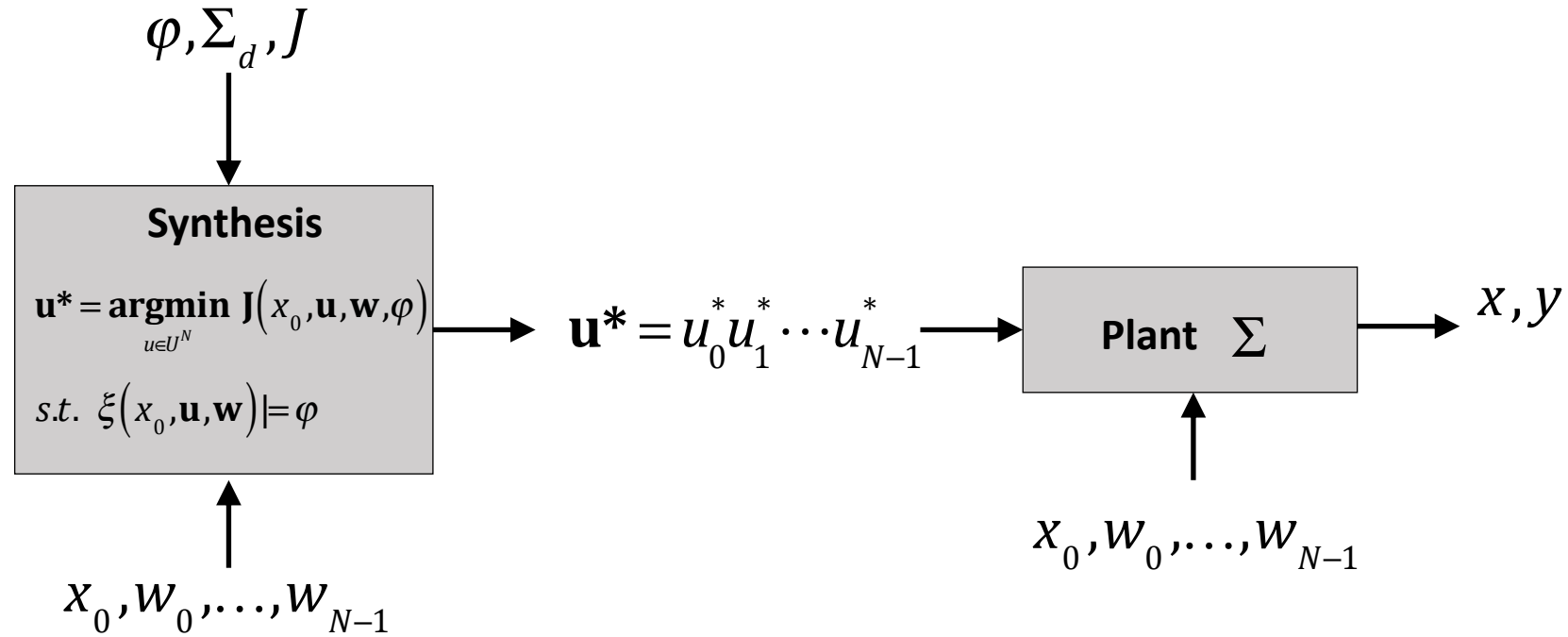
We consider a discrete-time approximation:  $\Sigma_d: x(t_k) = f_d(x(t_k), u(t_k), w(t_k)), \forall k > 0, t_{k+1} - t_k = \Delta t$

A run of the system is:  $\xi = (x_0 u_0 w_0)(x_1 u_1 w_1)(x_2 u_2 w_2) \dots$  where  $x_k = x(t_k), u_k = u(t_k), w_k = w(t_k),$

# Receding Horizon Control with STL



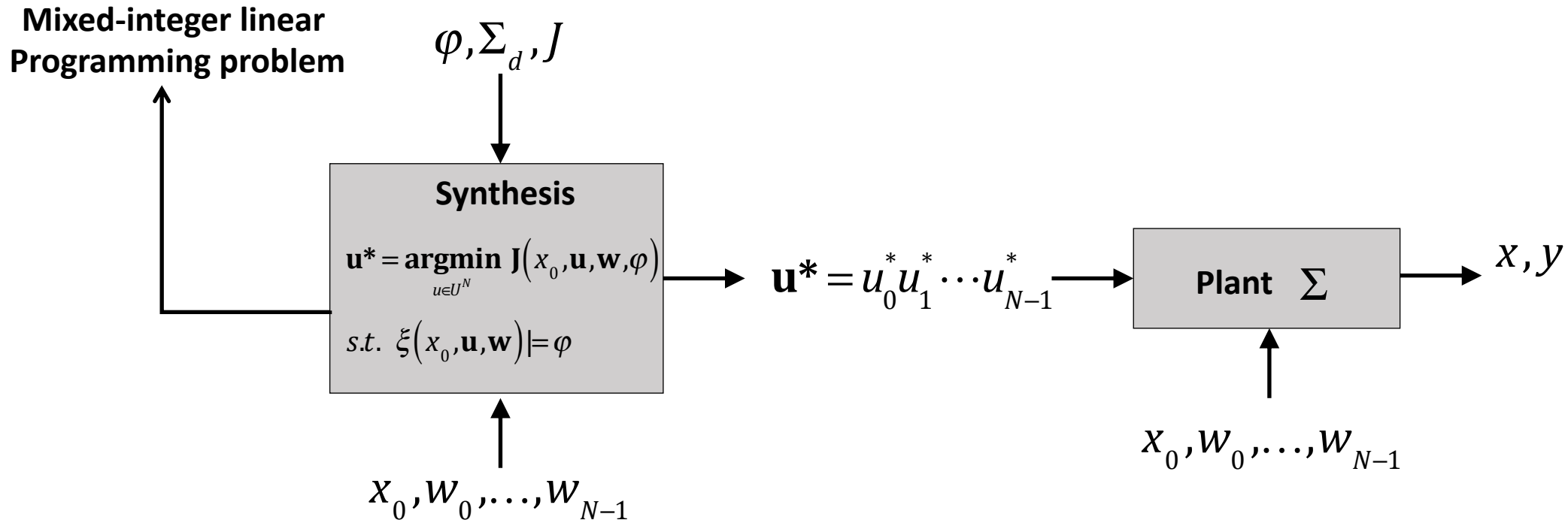
# Open-loop Controller Synthesis



$\mathbf{J}(x_0, \mathbf{u}, \mathbf{w}, \varphi) \in \mathbb{R}$  is a cost function

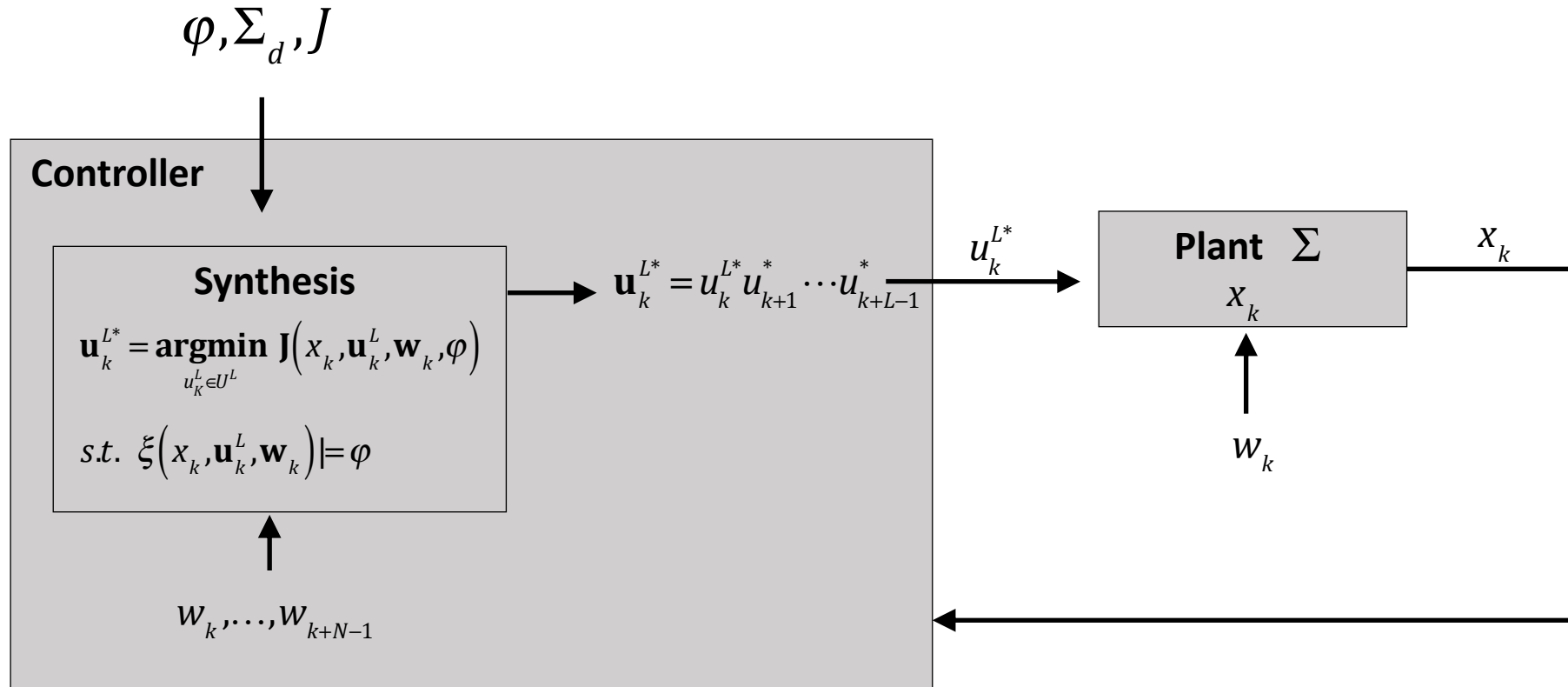


# Open-loop Controller Synthesis



$J(x_0, \mathbf{u}, \mathbf{w}, \varphi) \in \mathbb{R}$  is a cost function

# Closed-loop Controller Synthesis

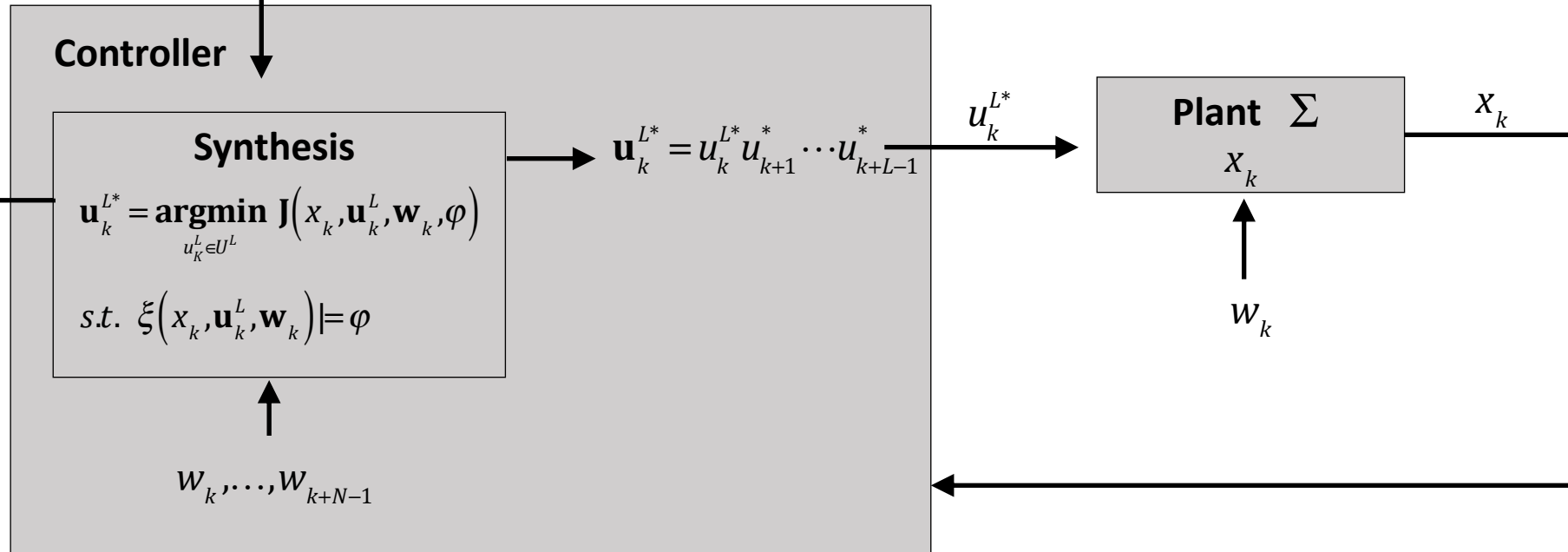


$J(x_0, \mathbf{u}, \mathbf{w}, \varphi) \in \mathbb{R}$  is a cost function

# Closed-loop Controller Synthesis

Mixed-integer linear  
Programming problem

$\varphi, \Sigma_d, J$



$\mathbf{J}(x_0, \mathbf{u}, \mathbf{w}, \varphi) \in \mathbb{R}$  is a cost function

# Mixed-Integer Linear Programming

$$\min c^T x$$

$$Ax \sim b$$
$$x \geq 0$$

$$x_i \in \mathbb{Z} \quad \forall i \in I$$

$$\sim = \{ \leq, =, \geq \}$$

maximize  $3x + 2y$

subject to

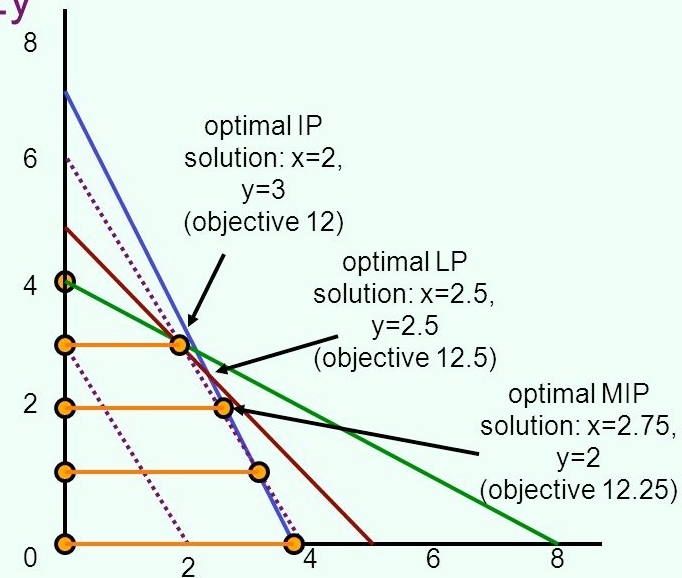
$$4x + 2y \leq 15$$

$$x + 2y \leq 8$$

$$x + y \leq 5$$

$$x \geq 0$$

$$y \geq 0, \text{ integer}$$



# Generating Systems Constraints

Given an horizon  $1, \dots, N$  and  $x_0, w_0, \dots, w_{N-1}$

$$x_1 = f_d(x_0, u_0, w_0)$$

$$x_2 = f_d(x_1, u_1, w_1)$$

$\vdots$

$$x_{N-1} = f_d(x_{N-2}, u_{N-2}, w_{N-2})$$

# Boolean Encoding of STL constraints

Given a formula  $\varphi$  we introduce a variable  $z_t^\varphi$

$$z_t^\varphi = 1 \Leftrightarrow \xi(x_t, u, w) \models \varphi$$

We recursively generate the MILP constraints corresponding to  $z_0^\varphi$

$$\varphi ::= \mu(x_t) > 0 \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \square_{[a,b]}\varphi_1 \mid \diamond_{[a,b]}\varphi_1 \mid \varphi_1 U_{[a,b]}\varphi_2$$

# Boolean Encoding of STL constraints

Given a formula  $\varphi$  we introduce a variable  $z_t^\varphi$

$$z_t^\varphi = 1 \Leftrightarrow \xi(x_t, u, w) \models \varphi$$

We recursively generate the MILP constraints corresponding to  $z_0^\varphi$

$$\varphi ::= \underbrace{\mu(x_t) > 0}_{\text{atomic}} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \square_{[a,b]}\varphi_1 \mid \diamond_{[a,b]}\varphi_1 \mid \varphi_1 U_{[a,b]}\varphi_2$$

$$\mu(x_t) \leq M_t(z_t^\mu) - \epsilon_t$$

$$-\mu(x_t) \leq M_t(1 - z_t^\mu) - \epsilon_t$$

Where  $M_t$  are sufficiently large positive numbers and  $\epsilon_t$  are sufficiently small positive numbers to bound  $\mu(x_t)$  away from zero

# Boolean Encoding of STL constraints

Given a formula  $\varphi$  we introduce a variable  $z_t^\varphi$

$$z_t^\varphi = 1 \Leftrightarrow \xi(x_t, u, w) \models \varphi$$

We recursively generate the MILP constraints corresponding to  $z_0^\varphi$

$$\varphi ::= \mu(x_t) > 0 \mid \underbrace{\neg \varphi}_{\psi} \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \square_{[a,b]} \varphi_1 \mid \diamond_{[a,b]} \varphi_1 \mid \varphi_1 U_{[a,b]} \varphi_2$$

$$\psi = \neg \varphi$$

$$z_t^\psi = 1 - z_t^\varphi$$



# Boolean Encoding of STL constraints

Given a formula  $\varphi$  we introduce a variable  $z_t^\varphi$

$$z_t^\varphi = 1 \Leftrightarrow \xi(x_t, u, w) \models \varphi$$

We recursively generate the MILP constraints corresponding to  $z_0^\varphi$

$$\varphi ::= \mu(x_t) > 0 \mid \neg \varphi \mid \underbrace{\varphi_1 \wedge \varphi_2}_{\psi} \mid \varphi_1 \vee \varphi_2 \mid \square_{[a,b]} \varphi_1 \mid \diamond_{[a,b]} \varphi_1 \mid \varphi_1 U_{[a,b]} \varphi_2$$

$$\psi = \bigwedge_{i=1}^m \varphi_i$$

$$z_t^\psi \leq z_{t_i}^{\varphi_i}, i = 1, \dots, m$$

$$z_t^\psi \geq 1 - m + \sum_{i=1}^m z_{t_i}^{\varphi_i}$$


# Boolean Encoding of STL constraints

Given a formula  $\varphi$  we introduce a variable  $z_t^\varphi$

$$z_t^\varphi = 1 \Leftrightarrow \xi(x_t, u, w) \models \varphi$$

We recursively generate the MILP constraints corresponding to  $z_0^\varphi$

$$\varphi ::= \mu(x_t) > 0 \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \square_{[a,b]}\varphi \mid \diamond_{[a,b]}\varphi \mid \varphi_1 U_{[a,b]}\varphi_2$$


$$\psi = \bigvee_{i=1}^m \varphi_i$$

$$z_t^\psi \geq z_{t_i}^{\varphi_i}, i = 1, \dots, m$$

$$z_t^\psi \leq \sum_{i=1}^m z_{t_i}^{\varphi_i}$$

# Boolean Encoding of STL constraints

Given a formula  $\varphi$  we introduce a variable  $z_t^\varphi$

$$z_t^\varphi = 1 \Leftrightarrow \xi(x_t, u, w) \models \varphi$$

We recursively generate the MILP constraints corresponding to  $z_0^\varphi$

$$\varphi ::= \mu(x_t) > 0 \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \underbrace{\square_{[a,b]} \varphi \mid \diamond_{[a,b]} \varphi \mid \varphi_1 U_{[a,b]} \varphi_2}$$

$$a_t^N = \min(t+a, N) \quad b_t^N = \min(t+b, N)$$

Compute  $z_t^\psi$  such that:

$$z_t^\psi = \bigwedge_{i=a_t^N}^{b_t^N} z_i^\varphi$$

# Boolean Encoding of STL constraints

Given a formula  $\varphi$  we introduce a variable  $z_t^\varphi$

$$z_t^\varphi = 1 \Leftrightarrow \xi(x_t, u, w) \models \varphi$$

We recursively generate the MILP constraints corresponding to  $z_0^\varphi$

$$\varphi ::= \mu(x_t) > 0 \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \square_{[a,b]} \varphi \mid \underbrace{\diamond_{[a,b]} \varphi}_{\text{red bracket}} \mid \varphi_1 U_{[a,b]} \varphi_2$$

$$a_t^N = \min(t+a, N) \quad b_t^N = \min(t+b, N)$$

Compute  $z_t^\psi$  such that:

$$z_t^\psi = \bigvee_{i=a_t^N}^{b_t^N} z_i^\varphi$$

# Boolean Encoding of STL constraints

Given a formula  $\varphi$  we introduce a variable  $z_t^\varphi$

$$z_t^\varphi = 1 \Leftrightarrow \xi(x_t, u, w) \models \varphi$$

We recursively generate the MILP constraints corresponding to  $z_0^\varphi$

$$\varphi ::= \mu(x_t) > 0 \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \square_{[a,b]}\varphi \mid \diamond_{[a,b]}\varphi \mid \underbrace{\varphi_1 U_{[a,b]}\varphi_2}$$

Compute  $z_t^\psi$  such that:

$$\psi = \varphi_1 U_{[a,b]}\varphi_2 = \square_{[0,a]}\varphi_1 \wedge \diamond_{[a,b]}\varphi_2 \wedge \diamond_{[a,a]}(\varphi_1 U \varphi_2)$$

$$\langle\langle \varphi_1 U \varphi_2 \rangle\rangle_t = \begin{cases} \left\{ z_t^{\varphi_2} \vee \left( z_t^{\varphi_1} \wedge \langle\langle \varphi_1 U \varphi_2 \rangle\rangle_{t+1} \right) \right\}, & t=1, \dots, N-1 \\ z_N^{\varphi_2} \end{cases}$$

# Algorithm for Open-Loop

---

**Algorithm 1** Algorithm for Problem **1**

---

```
1: procedure OPEN_LOOP( $f, x_0, \mathbf{w}, N, \varphi, J$ )
2:   LOOP_CONSTRAINTS  $\leftarrow$  Sec. IV-B
3:   SYSTEM_CONSTRAINTS  $\leftarrow$  Sec. IV-A
4:   STL_CONSTRAINTS  $\leftarrow$  Sec. IV-C2 OR Sec. IV-D
5:
    $\mathbf{u}^* \leftarrow \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}^N} J(x_0, \mathbf{u}, \mathbf{w}, \varphi)$ 
   s.t. LOOP_CONSTRAINTS
        SYSTEM_CONSTRAINTS
        STL_CONSTRAINTS
   Return  $\mathbf{u}^*$ 
6: end procedure
```

---

# Quantitative Encoding of STL constraints

$$\varphi ::= \underbrace{\mu(x_t) > 0}_{r_t^\mu} \mid \underbrace{\neg\varphi \mid \varphi_1 \wedge \varphi_2}_{r_t^\psi = -r_t^\varphi} \mid \varphi_1 \vee \varphi_2 \mid \square_{[a,b]} \varphi_1 \mid \diamond_{[a,b]} \varphi_1 \mid \varphi_1 U_{[a,b]} \varphi_2$$

# Quantitative Encoding of STL constraints

$$\varphi ::= \mu(x_t) > 0 \mid \neg\varphi \mid \underbrace{\varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2}_{\text{Binary variable}} \mid \square_{[a,b]}\varphi_1 \mid \diamond_{[a,b]}\varphi_1 \mid \varphi_1 U_{[a,b]}\varphi_2$$

$$\left( \sum_{i=1}^m p_{t_i}^{\varphi_i} = 1 \right) \longrightarrow \text{Binary variable}$$

$$r_t^\psi \leq r_{t_i}^{\varphi_i}, i = 1, \dots, m$$

$$r_{t_i}^{\varphi_i} - (1 - p_{t_i}^{\varphi_i})M \leq r_t^\psi \leq r_{t_i}^{\varphi_i} + M(1 - p_{t_i}^{\varphi_i})$$

$$\longrightarrow r_t^\psi = \min_i (r_{t_i}^{\varphi_i})$$



# Quantitative Encoding of STL constraints

$$\varphi ::= \mu(x_t) > 0 \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \underbrace{\varphi_1 \vee \varphi_2}_{\text{OR}} \mid \square_{[a,b]}\varphi_1 \mid \diamond_{[a,b]}\varphi_1 \mid \varphi_1 U_{[a,b]}\varphi_2$$

$$\sum_{i=1}^m p_{t_i}^{\varphi_i} = 1$$

$$r_t^\psi \geq r_{t_i}^{\varphi_i}, i = 1, \dots, m$$

$$r_{t_i}^{\varphi_i} - (1 - p_{t_i}^{\varphi_i})M \leq r_t^\psi \leq r_{t_i}^{\varphi_i} + M(1 - p_{t_i}^{\varphi_i})$$



$$r_t^\psi = \max_i (r_{t_i}^{\varphi_i})$$

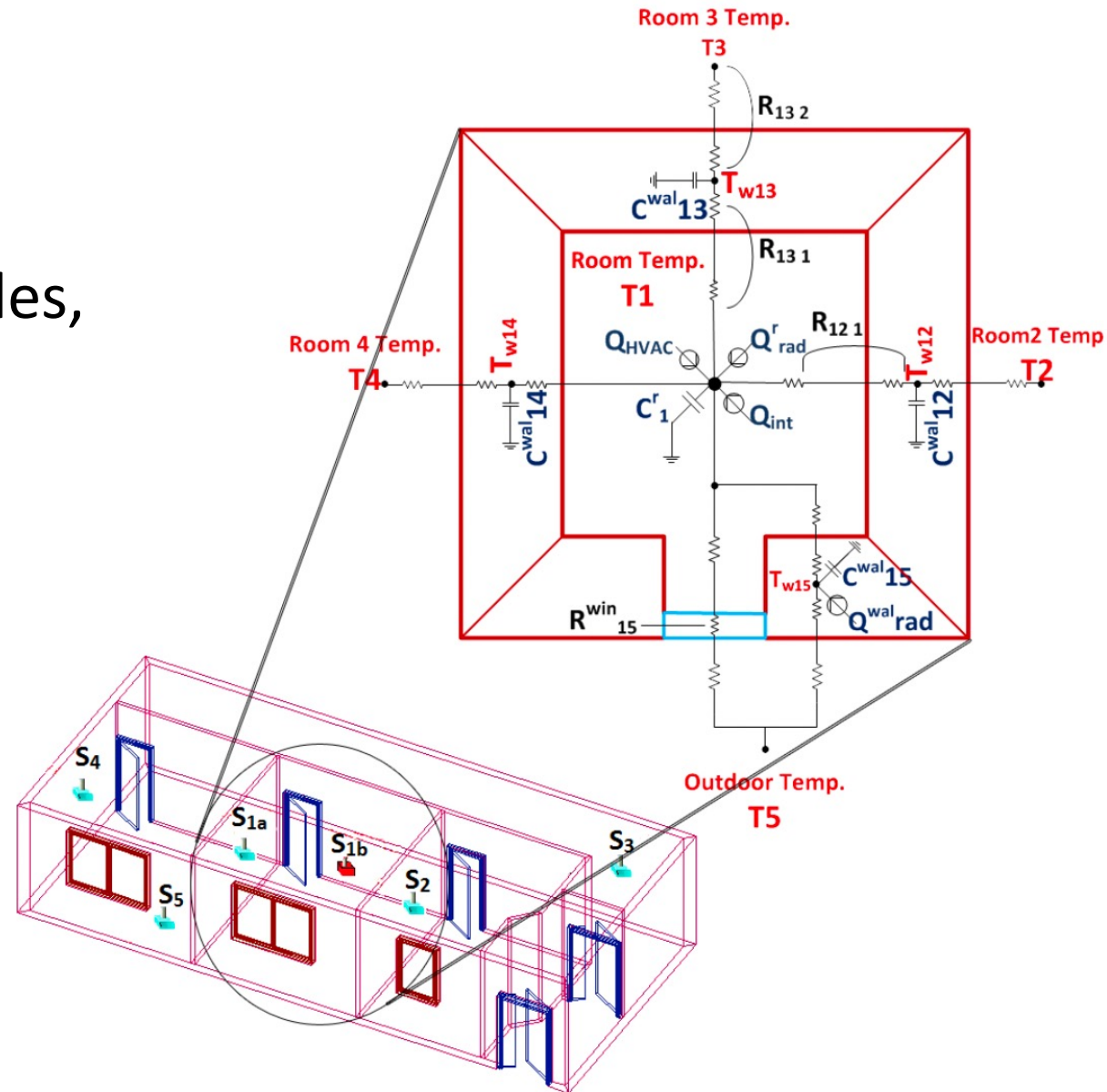
# Quantitative Encoding of STL constraints

$$\varphi ::= \mu(x_t) > 0 \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \boxed{\square_{[a,b]} \varphi_1 \mid \diamond_{[a,b]} \varphi_1 \mid \varphi_1 U_{[a,b]} \varphi_2}$$

As defined before

# Case Study: Building Climate Control

- ▶ Building a thermal model
- ▶ a resistor capacitor circuit with  $n$  nodes,  $m$  rooms  
 $n - m$  are walls.
- ▶  $T_{r_i}$  temperature of room  $r_i$
- ▶  $w_{i,j}$  wall between rooms  $i$  and  $j$
- ▶  $T_{w_{i,j}}$  temperature of wall  $w_{i,j}$



# Building Climate Model

$$C_{i,j}^w \frac{dT_{w_{i,j}}}{dt} = \sum_{k \in \mathcal{N}_{w_{i,j}}} \frac{T_{r_k} - T_{w_{i,j}}}{R_{i,jk}} + r_{i,j} \alpha_{i,j} A_{w_{i,j}} Q_{rad_{i,j}} \quad (6)$$

$$C_i^r \frac{dT_{r_i}}{dt} = \sum_{k \in \mathcal{N}_{r_i}} \frac{T_k - T_{r_i}}{R_{i,k_i}} + \dot{m}_{r_i} c_a (T_{s_i} - T_{r_i}) + w_i \tau_{w_i} A_{win_i} Q_{rad_i} + \dot{Q}_{int_i}, \quad (7)$$

# Building Climate Model

$$\frac{d}{dt}x_t = f(x_t, u_t, w_t),$$

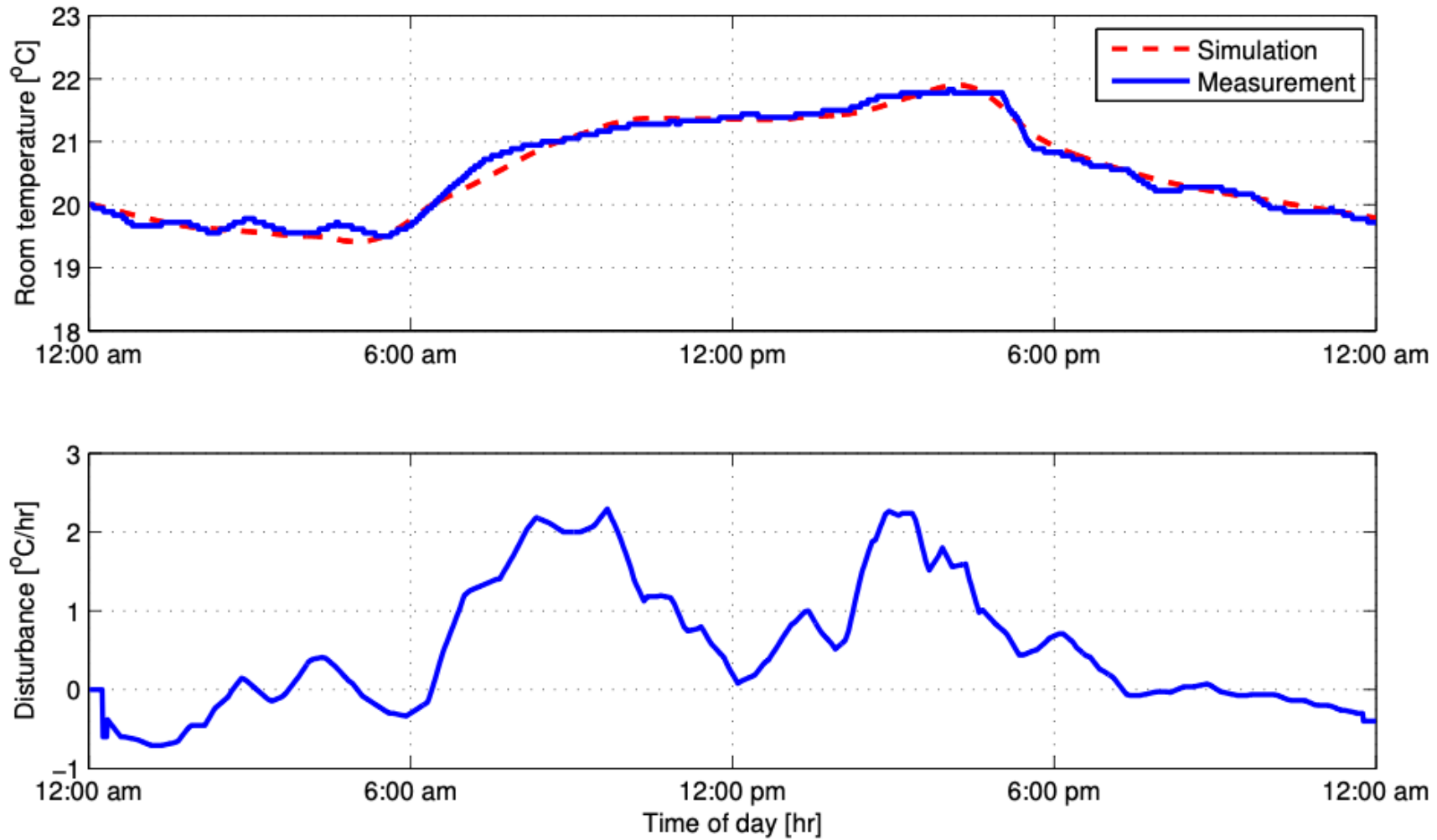
Where state vector  $x_t$  representing the temperature of the nodes in the thermal network

$u_t$  is the input vector representing the air mass flow rate and discharge air temperature of conditioned air into each thermal zone

$w_t$  stores the estimated disturbance values, aggregating various unmodelled dynamic

$y_t$  is the output vector, representing the temperature of the thermal zones.

# Simulation Building Model



Simulated temperature, measured temperature and unmodelled dynamics of a thermal zone in Bancroft library on UC Berkeley campus.

# MPC for Building Climate Control

Maintaining a comfort temperature given by  $T_t^{comf}$  whenever the room is occupied while minimizing the cost of heating.

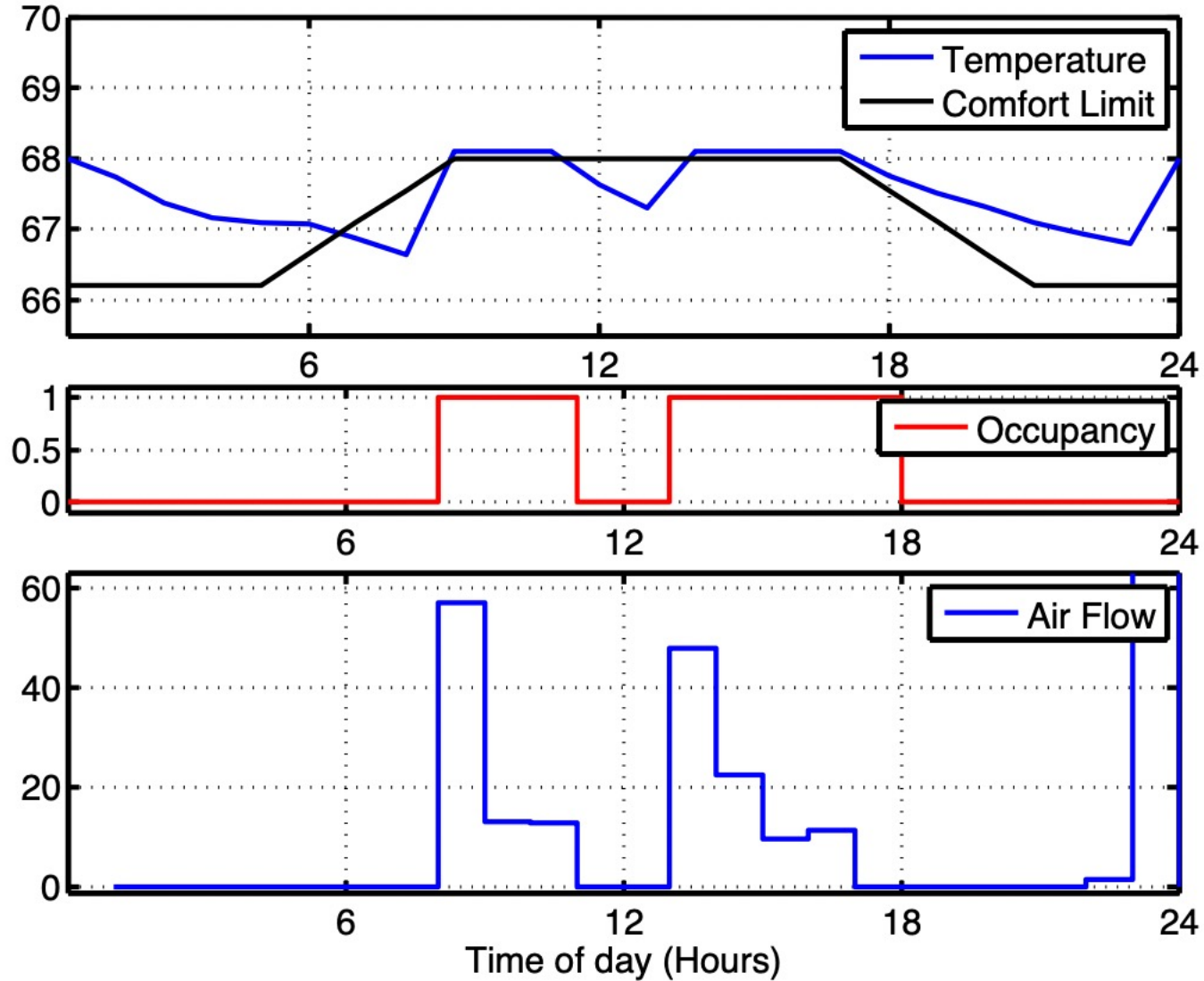
$$\min_{\vec{u}_t} \sum_{k=0}^{H-1} \|u_{t+k}\| \quad \text{s.t.}$$

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

$$x_t \models \varphi \quad \text{with} \quad \varphi = \square_{[0,H]}((occ_t > 0) \Rightarrow (T_t > T_t^{comf}))$$

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

# MPC for Building Climate Control





# BluSTL tool



EPiC Series in Computer Science

Volume 34, 2015, Pages 160–168

ARCH14-15. 1st and 2nd International Workshop on  
Applied verification for Continuous and Hybrid Systems



## BluSTL: Controller Synthesis from Signal Temporal Logic

### Specifications

Alexandre Donzé<sup>1</sup> and Vasumathi R

<sup>1</sup> Department of Electrical Engineering and Comp  
UC Berkeley,

Berkeley, CA 94720, USA [donze@berkeley](mailto:donze@berkeley)

<sup>2</sup> California Institute of Technology, Pasadena, CA, USA

#### Abstract

We present BluSTL, a MATLAB toolbox for automatically generating controllers written in Signal Temporal Logic (STL). The toolbox takes as input a specification expressed in STL and constructs an open-loop or a closed-loop (in a reactive fashion) controller that enforces these constraints on the system function. The controller can also be made reactive or robust to some disturbances. The toolbox is available at <https://github.com/BluSTL/BluSTL>.

### 2.1 System dynamics

We consider a continuous-time system  $\Sigma$  of the form

$$\dot{x} = Ax + B_u u + B_w w \quad (1)$$

$$y = Cx + D_u u + D_w w \quad (2)$$

where

- $x \in \mathcal{X} \subseteq \mathbb{R}^n$  is the *system state*,
- $u \in \mathcal{U} \subseteq \mathbb{R}^m$  is the *control input*,
- $w \in \mathcal{W} \subseteq \mathbb{R}^l$  is the *external input*,
- $y \in \mathcal{Y} \subseteq \mathbb{R}^o$  is the *system output*.

Given a sampling time  $\Delta t > 0$ , we discretize  $\Sigma$  into  $\Sigma_d$  of the form

$$x(t_{k+1}) = A^d x(t_k) + B_u^d u(t_k) + B_w^d w(t_k) \quad (3)$$

$$y(t_k) = C^d x(t_k) + D_u^d u(t_k) + D_w^d w(t_k) \quad (4)$$

where for all  $k > 0$ ,  $t_{k+1} - t_k = \Delta t$  and  $t_0 = 0$ . Given an integer  $N > 0$ ,  $x_0 \in \mathcal{X}$ , and two sequences  $\mathbf{u} \in \mathcal{U}^{N-1}$  and  $\mathbf{w} \in \mathcal{W}^{N-1}$  noted

$$\mathbf{u} = u_0 u_1 \dots u_{N-1}$$

$$\mathbf{w} = w_0 w_1 \dots w_{N-1}$$

we denote by  $\xi(x_0, \mathbf{u}, \mathbf{w}) \in \mathcal{X}^N$  the 4-tuple of sequences  $(\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{w}) = \xi(x_0, \mathbf{u}, \mathbf{w})$  such that  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{u}$  and  $\mathbf{w}$  satisfy (3-4) with  $x(t_k) = x_k$ ,  $y(t_k) = y_k$ ,  $u(t_k) = u_k$  and  $w(t_k) = w_k$  for all  $k$ .  $\xi(x_0, \mathbf{u}, \mathbf{w})$ , or sometimes simply  $\xi$  is called a run of  $\Sigma_d$ .