Cyber-Physical Systems

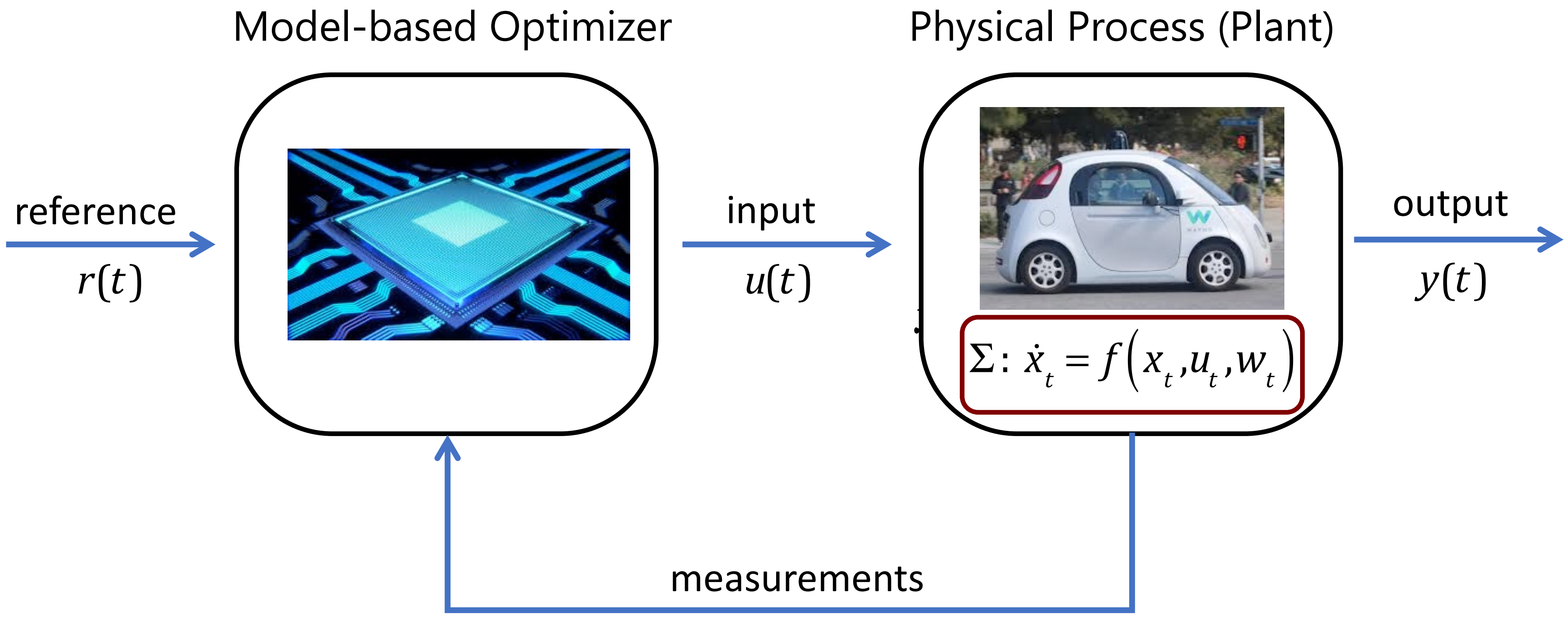
Laura Nenzi

Università degli Studi di Trieste
II Semestre 2020

Lecture 23: Control Synthesis with STL

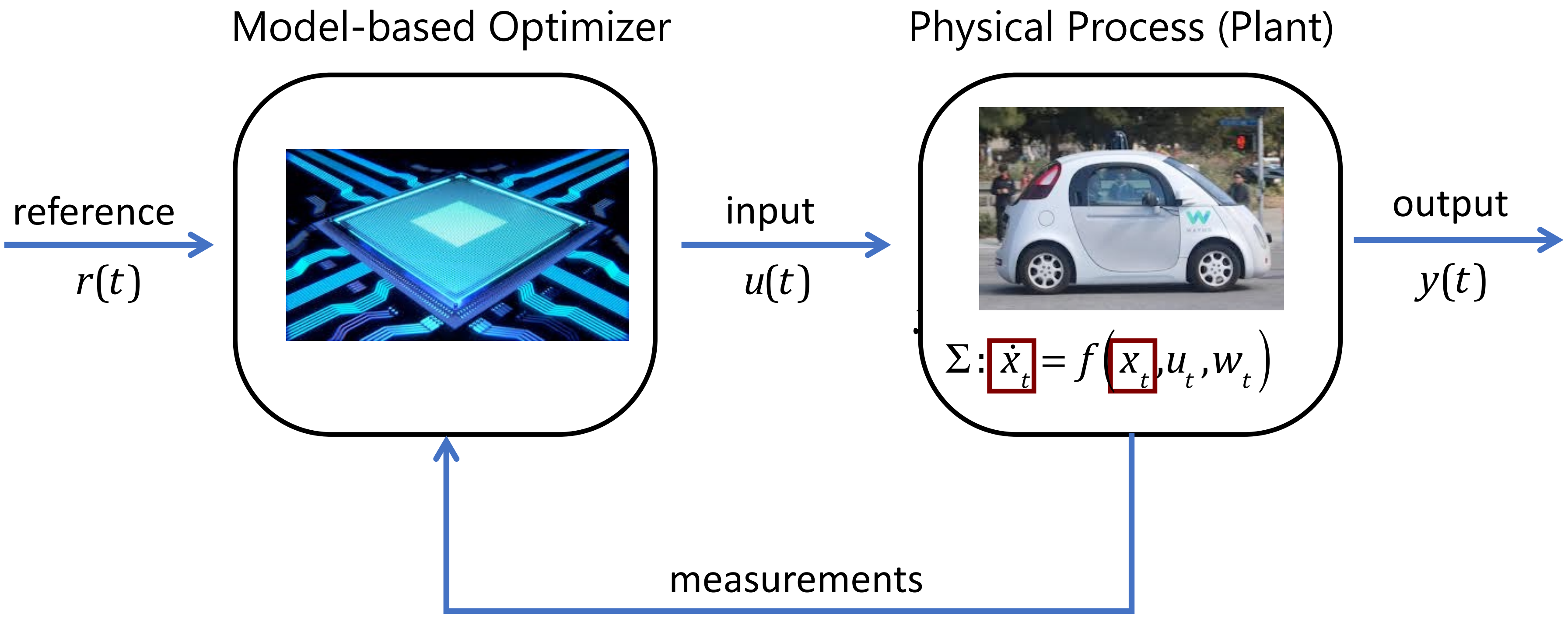
[Many Slides due to Ezio Bartocci]

Model Predictive Control



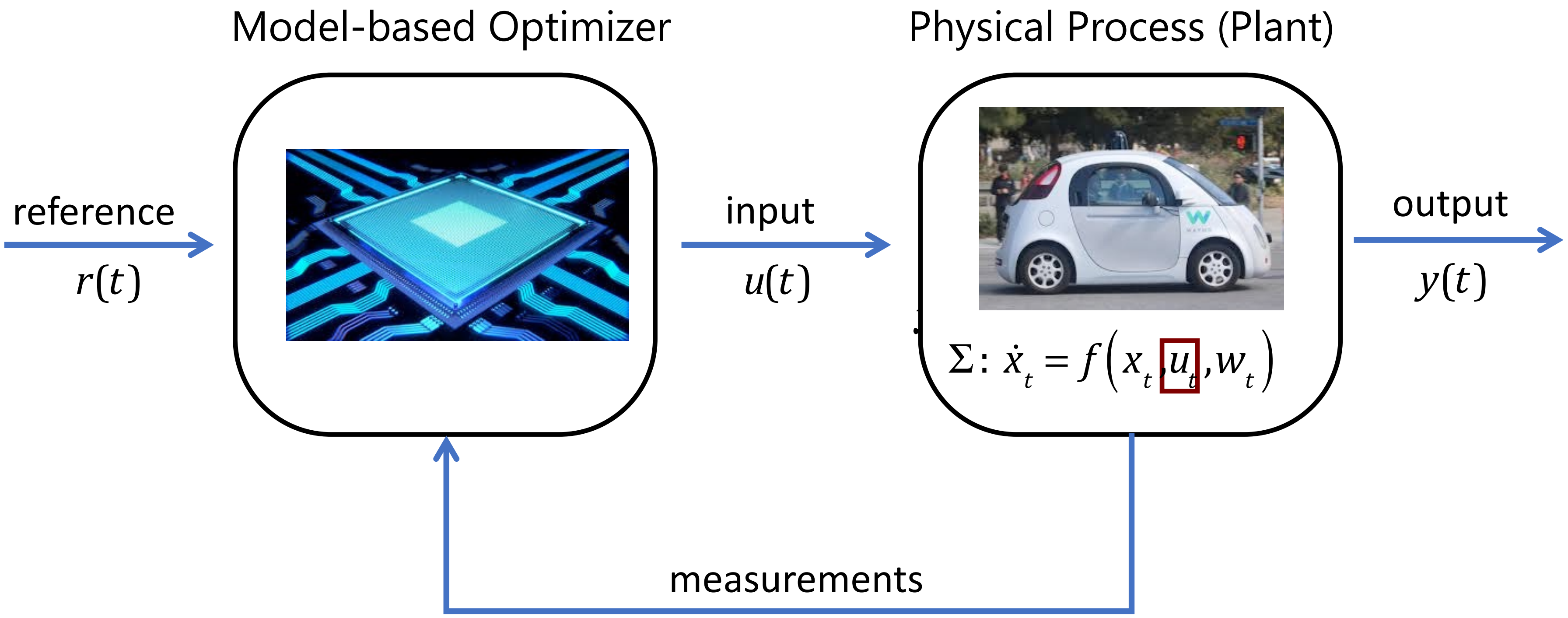
The idea is to use the dynamical model of the process to predict its future evolution and optimize consequently the control input signal

Model Predictive Control



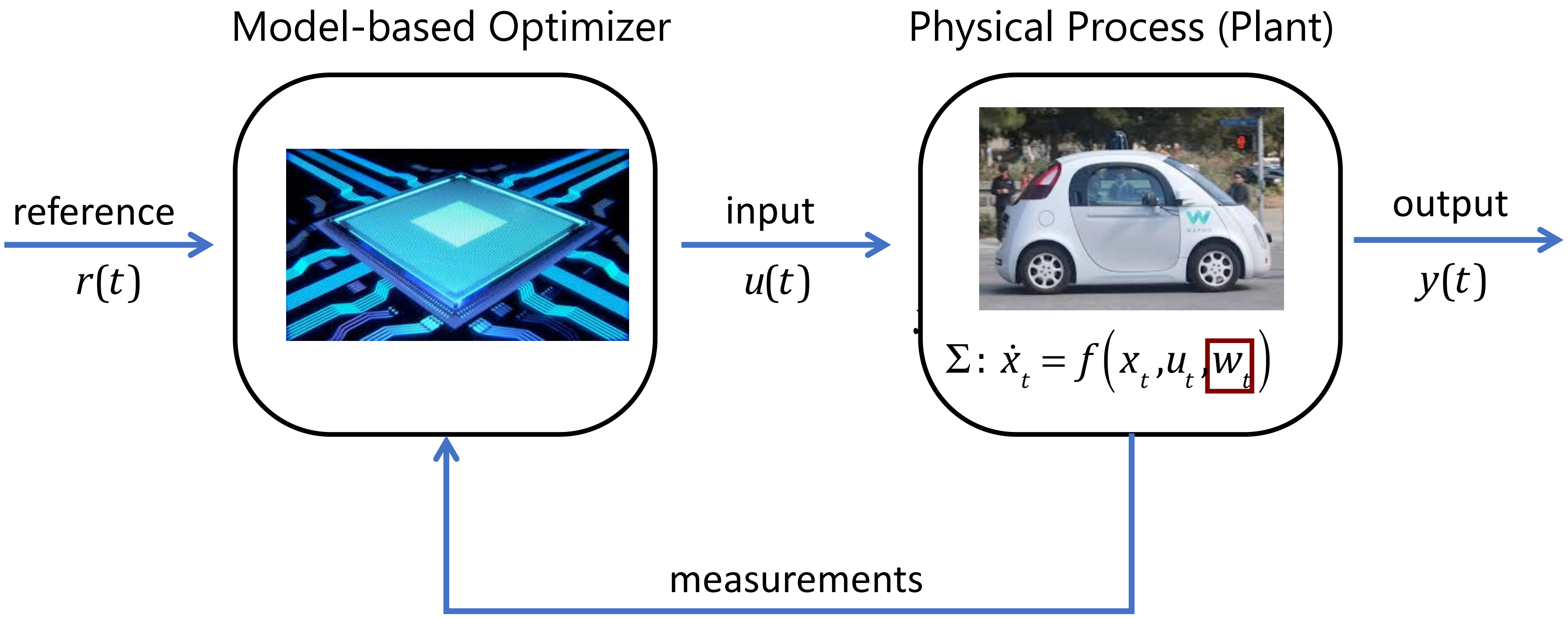
$\dot{x}_t \in \mathrm{X} \subseteq \left(\mathbb{R}^{n_c} \times \{0,1\}^{n_l} \right)$ are continuous/logical states

Model Predictive Control



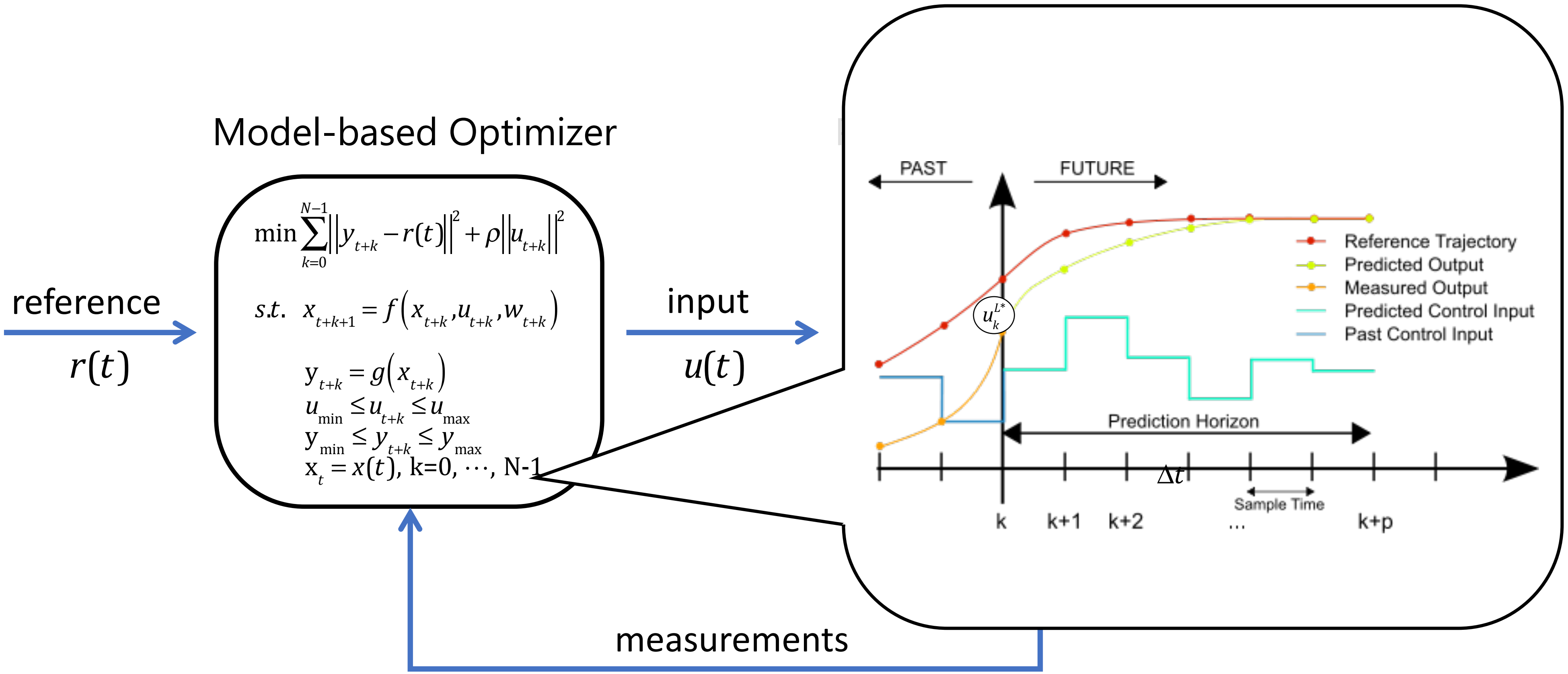
$\dot{u}_t \in U \subseteq \left(\mathbb{R}^{m_c} \times \{0,1\}^{m_l}\right)$ are continuous/logical inputs

Model Predictive Control



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

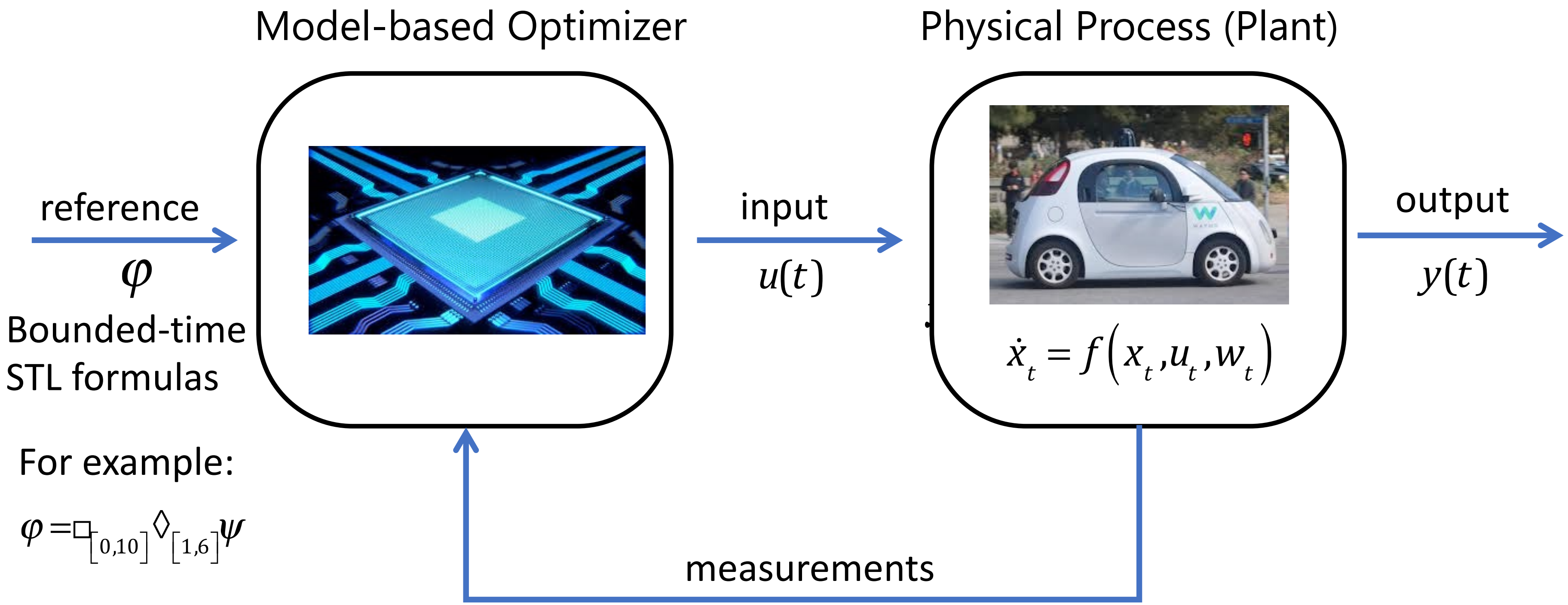
Example of Receding Horizon Control



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

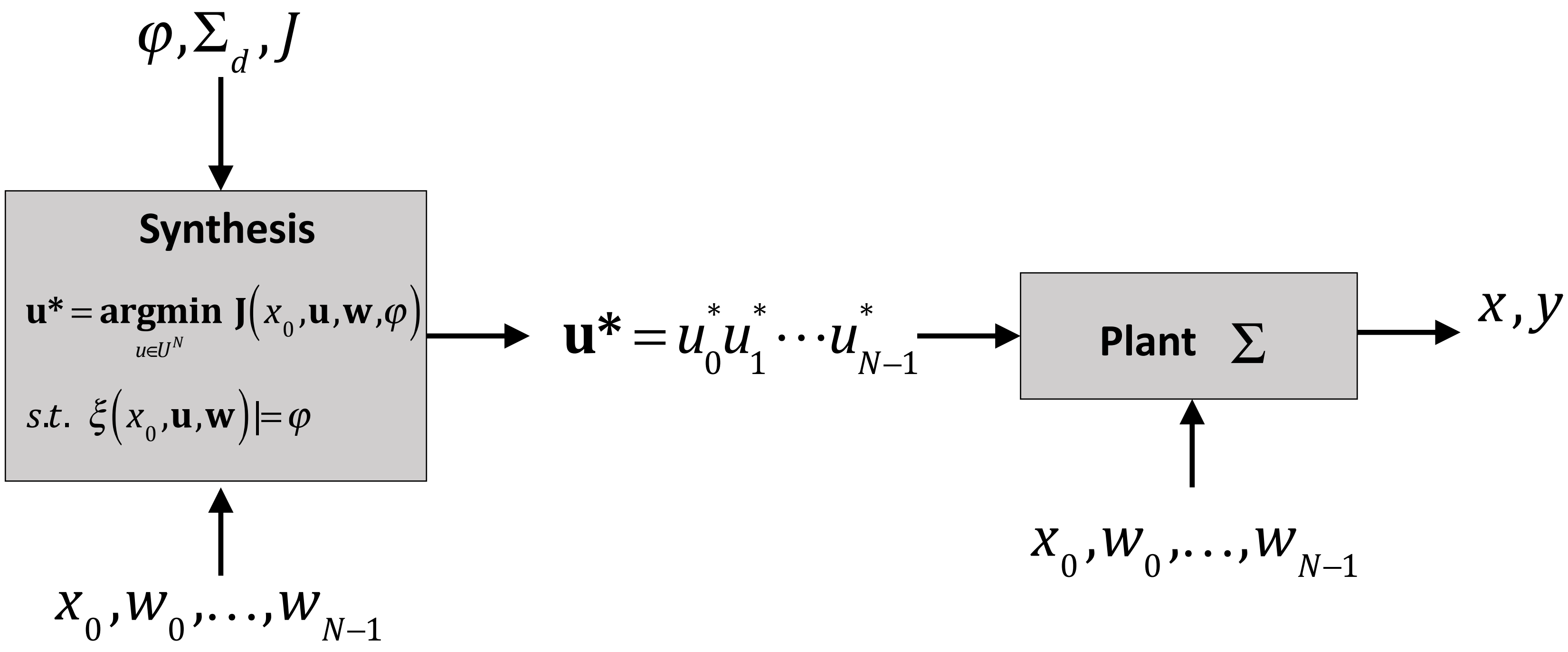
A run of the system is: $\xi=(x_0u_0w_0)(x_1u_1w_1)(x_2u_2w_2)\ldots$ where $x_k=x(t_k),\ u_k=u(t_k),\ w_k=w(t_k),$

Receding Horizon Control with STL



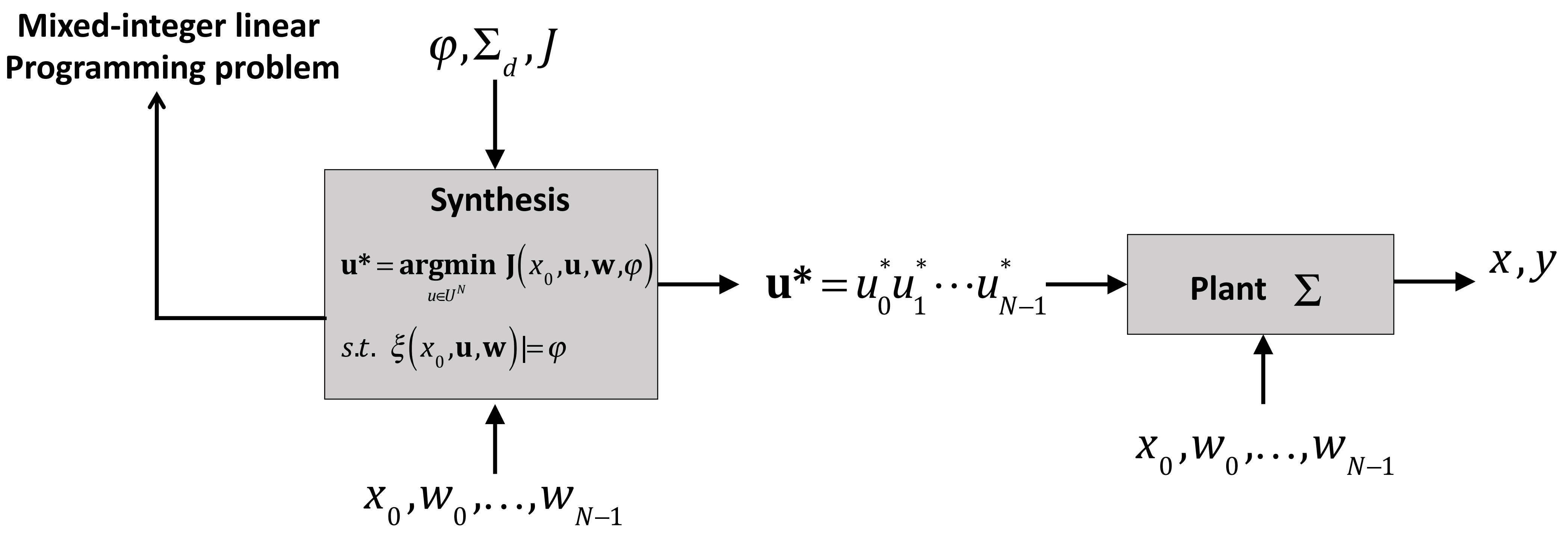
Vasumathi Raman, Alexandre Donzé, Mehdi Maasoumy, Richard M. Murray, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia: Model predictive control with signal temporal logic specifications. CDC 2014: 81-87

Open-loop Controller Synthesis



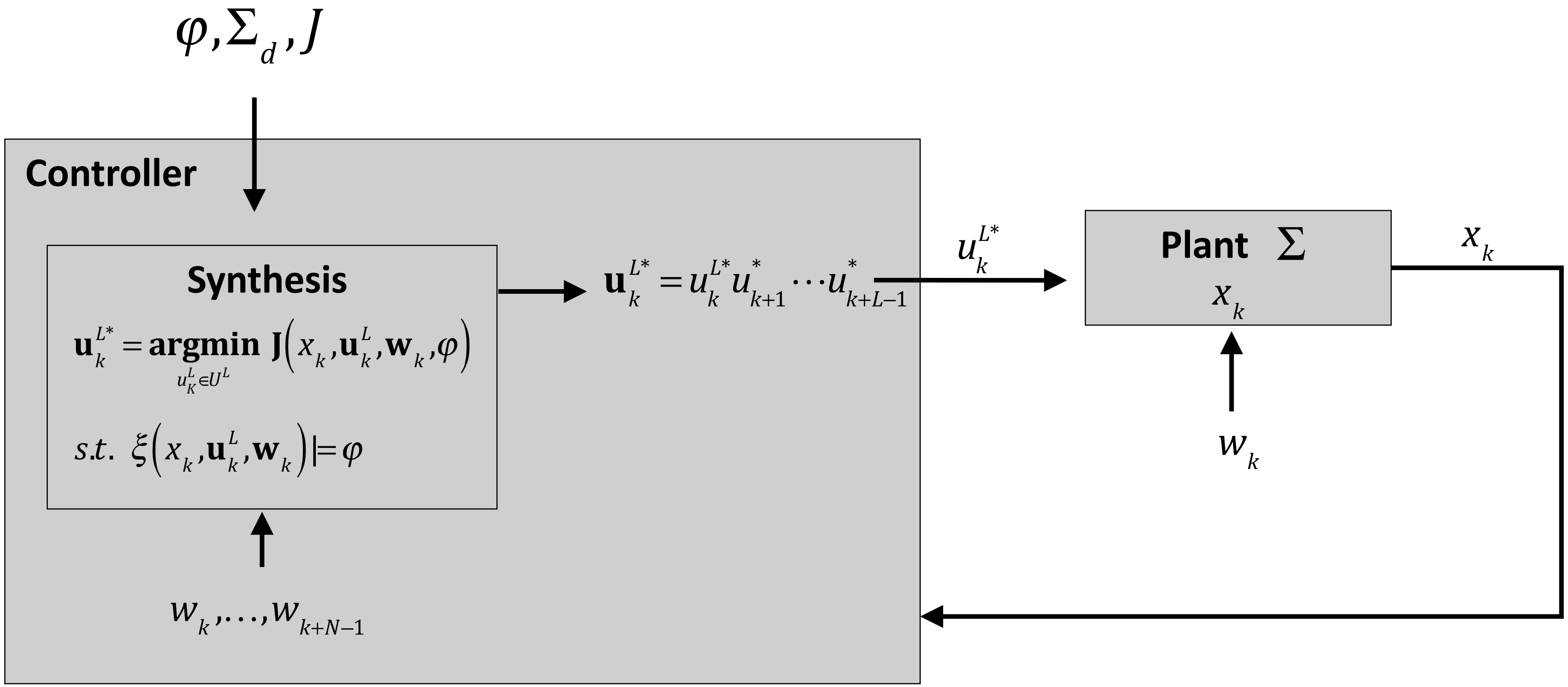
$\mathbf{J}(x_0, \mathbf{u}, \mathbf{w}, \varphi) \in \mathbb{R}$ **is a cost function**

Open-loop Controller Synthesis



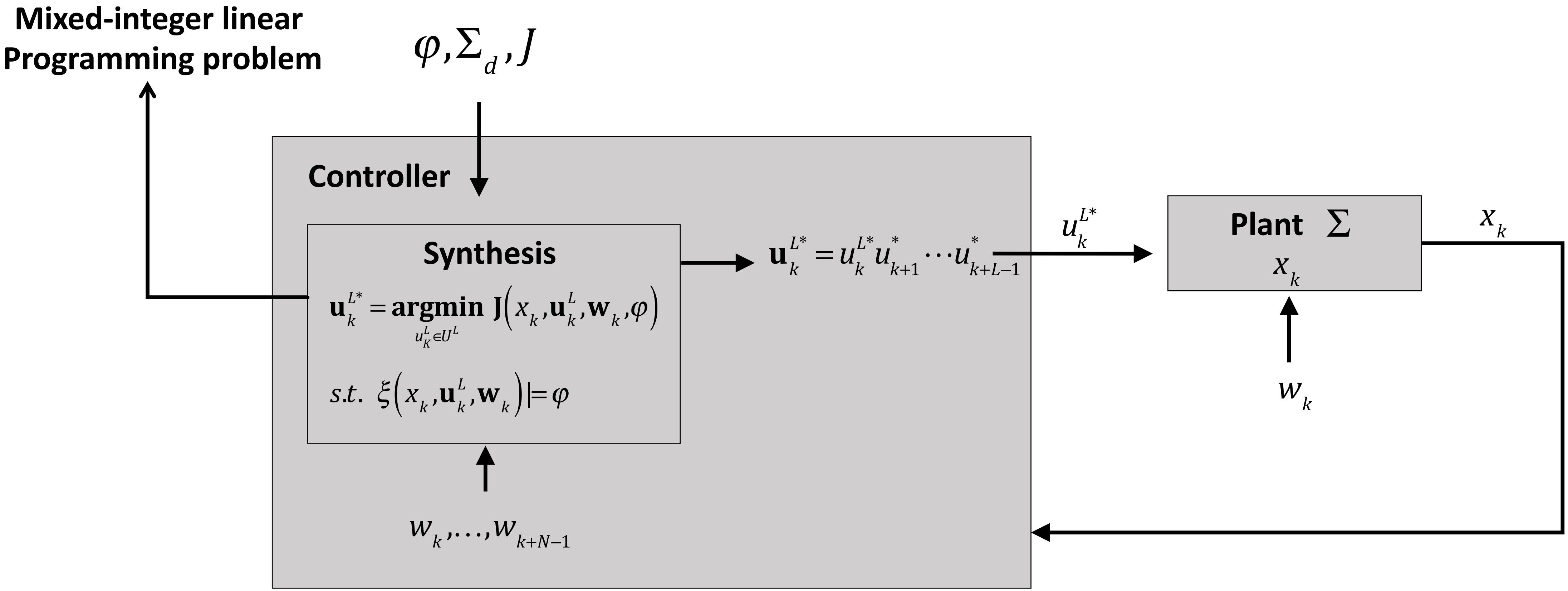
$\mathbf{J}(x_0, \mathbf{u}, \mathbf{w}, \varphi) \in \mathbb{R}$ **is a cost function**

Closed-loop Controller Synthesis



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

Closed-loop Controller Synthesis



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

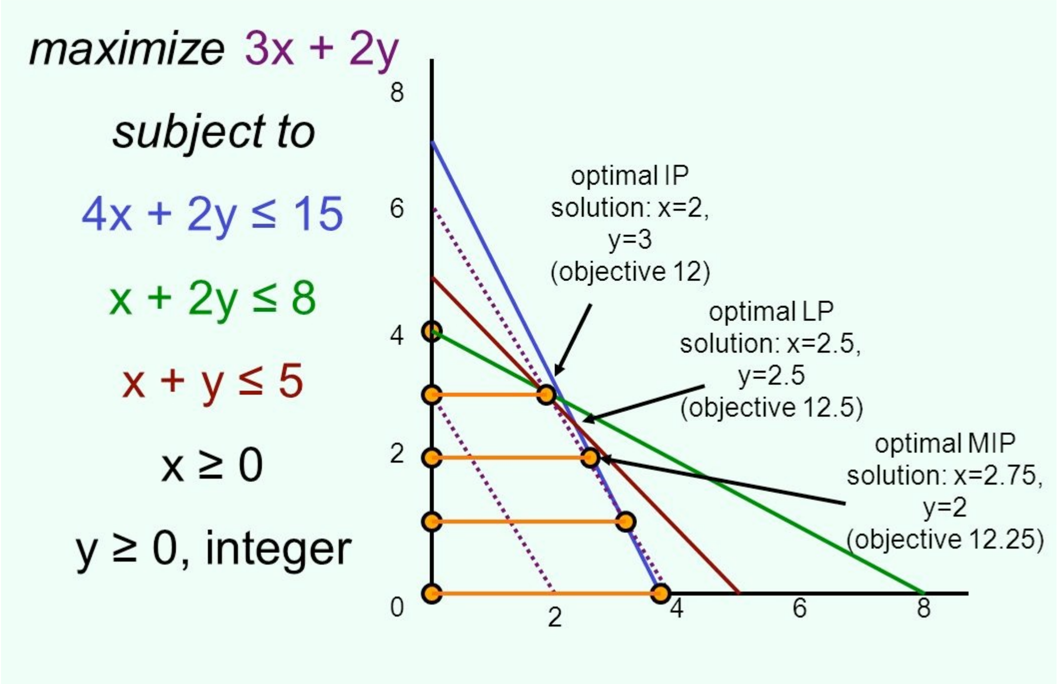
Mixed-Integer Linear Programming

$$\min\ c^T x$$

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

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

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



Generating Systems Constraints

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

$$x_1 = f_d\left(x_0, u_0, w_0\right)$$

$$x_2 = f_d\left(x_1, u_1, w_1\right)$$

$$\vdots$$

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

Boolean Encoding of STL constraints

Given a formula φ we introduce a variable z_t^{φ}

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

We recursively generate the MILP constraints corresponding to z_0^{φ}

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

Boolean Encoding of STL constraints

Given a formula φ we introduce a variable z_t^{φ}

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

We recursively generate the MILP constraints corresponding to z_0^{φ}

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

$$\mu(x_t) \leq M_t(z_t^{\mu}) - \varepsilon_t$$

$$-\mu(x_t) \leq M_t(1 - z_t^{\mu}) - \varepsilon_t$$

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

Boolean Encoding of STL constraints

Given a formula φ we introduce a variable z_t^{φ}

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

We recursively generate the MILP constraints corresponding to z_0^{φ}

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

$$\psi = \neg\varphi$$

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

Boolean Encoding of STL constraints

Given a formula φ we introduce a variable z_t^{φ}

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

We recursively generate the MILP constraints corresponding to z_0^{φ}

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

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

$$z_t^{\psi} \le z_{t_i}^{\varphi_i}, i = 1, \cdots, m$$

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

Boolean Encoding of STL constraints

Given a formula φ we introduce a variable z_t^{φ}

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

We recursively generate the MILP constraints corresponding to z_0^{φ}

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

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

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

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

Boolean Encoding of STL constraints

Given a formula φ we introduce a variable z_t^{φ}

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

We recursively generate the MILP constraints corresponding to z_0^{φ}

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

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

Compute z_t^{ψ} such that:

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

Boolean Encoding of STL constraints

Given a formula φ we introduce a variable z_t^{φ}

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

We recursively generate the MILP constraints corresponding to z_0^{φ}

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

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

Compute z_t^{ψ} such that:

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

Boolean Encoding of STL constraints

Given a formula φ we introduce a variable z_t^{φ}

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

We recursively generate the MILP constraints corresponding to z_0^{φ}

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

Compute z_t^{ψ} such that:

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

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

Algorithm for Open-Loop

Algorithm 1 Algorithm for Problem 1

```
procedure OPEN_ LOOP(f, x_0, w, N, φ, J)
    LOOP_CONSTRAINTS ← Sec. IV-B
    SYSTEM_CONSTRAINTS ← Sec. IV-A
    STL_CONSTRAINTS ← Sec. IV-C2 OR Sec. IV-D
```

$$\begin{aligned} \mathbf{u}^* \leftarrow \underset{\mathbf{u}\in\mathcal{U}^N}{\operatorname{argmin}} \quad & J(x_0, \mathbf{u}, \mathbf{w}, \varphi) \\ \text{s.t.} \quad & \text{LOOP\_CONSTRAINTS} \\ & \text{SYSTEM\_CONSTRAINTS} \\ & \text{STL\_CONSTRAINTS} \end{aligned}$$

```
    Return u*
end procedure
```

Quantitative Encoding of STL constraints

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

Quantitative Encoding of STL constraints

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

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

→ Binary variable

$$r_t^{\psi} \le r_{t_i}^{\varphi_i}, i = 1, \ldots, m$$

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

$$\Rightarrow \quad r_t^{\psi} = \min_i (r_{t_i}^{\phi_i})$$

Quantitative Encoding of STL constraints

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

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

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

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

$$\Rightarrow \quad r_t^{\psi} = \max_i (r_{t_i}^{\phi_i})$$

Quantitative Encoding of STL constraints

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

As defined before

Model Predictive Control for Closed-Loop

Algorithm 2 MPC Algorithm for Problem 2

procedure MPC($f, x_0, \phi = \Box \varphi_{MPC}, J$)
Let M be a large positive constant.
Let H be the bound of φ_{MPC}.
Set $P_0 = 0$ and $P_i = -M\ \forall 0 < i \le H$.
$\mathbf{w}^t \leftarrow$ PREDICT_W(0).
Compute $\mathbf{u}^0 = u_0^0, u_1^0, ..., u_{2H-1}^0$ as:
$\mathbf{u}^0 \leftarrow$ OPEN_LOOP*$(f, x_0, \mathbf{w}^0, 2H, \Box_{[0,H]}\, \varphi_{MPC}, J, \mathbf{P}^H, \emptyset)$
for t=1; t¡=H;t=t+1 **do**
Set $\mathbf{u}_{old}^t = u_0^0, u_1^1, u_2^2, ..., u_{t-1}^{t-1}$.
Set $P_i = 0$ for $0 \le i \le t$, $P_i = -M\ \forall t < i \le H$.
$\mathbf{w}^t \leftarrow$ PREDICT_W(t).
Compute $\mathbf{u}^t = u_0^t, u_1^t, ..., u_{2H-1}^t$ as:
$\mathbf{u}^t \leftarrow$ OPEN_LOOP*$(f, x_t, \mathbf{w}^t, 2H, \Box_{[0,H]}\, \varphi_{MPC}, J, \mathbf{P}^H, \mathbf{u}_{old}^t)$
end for
while True **do**
Set $\mathbf{u}_{old}^t = u_1^{t-1}, u_2^{t-1}, u_3^{t-1}, ..., u_t^{t-1}$.
Set $P_i = 0$ for $0 \le i \le H$.
$\mathbf{w}^t \leftarrow$ PREDICT_W(t).
$\mathbf{u}^t \leftarrow$ OPEN_LOOP*$(f, x_t, \mathbf{w}^t, 2H, \Box_{[0,H]}\, \varphi_{MPC}, J, \mathbf{P}^H, \mathbf{u}_{old}^t)$
end while
end procedure

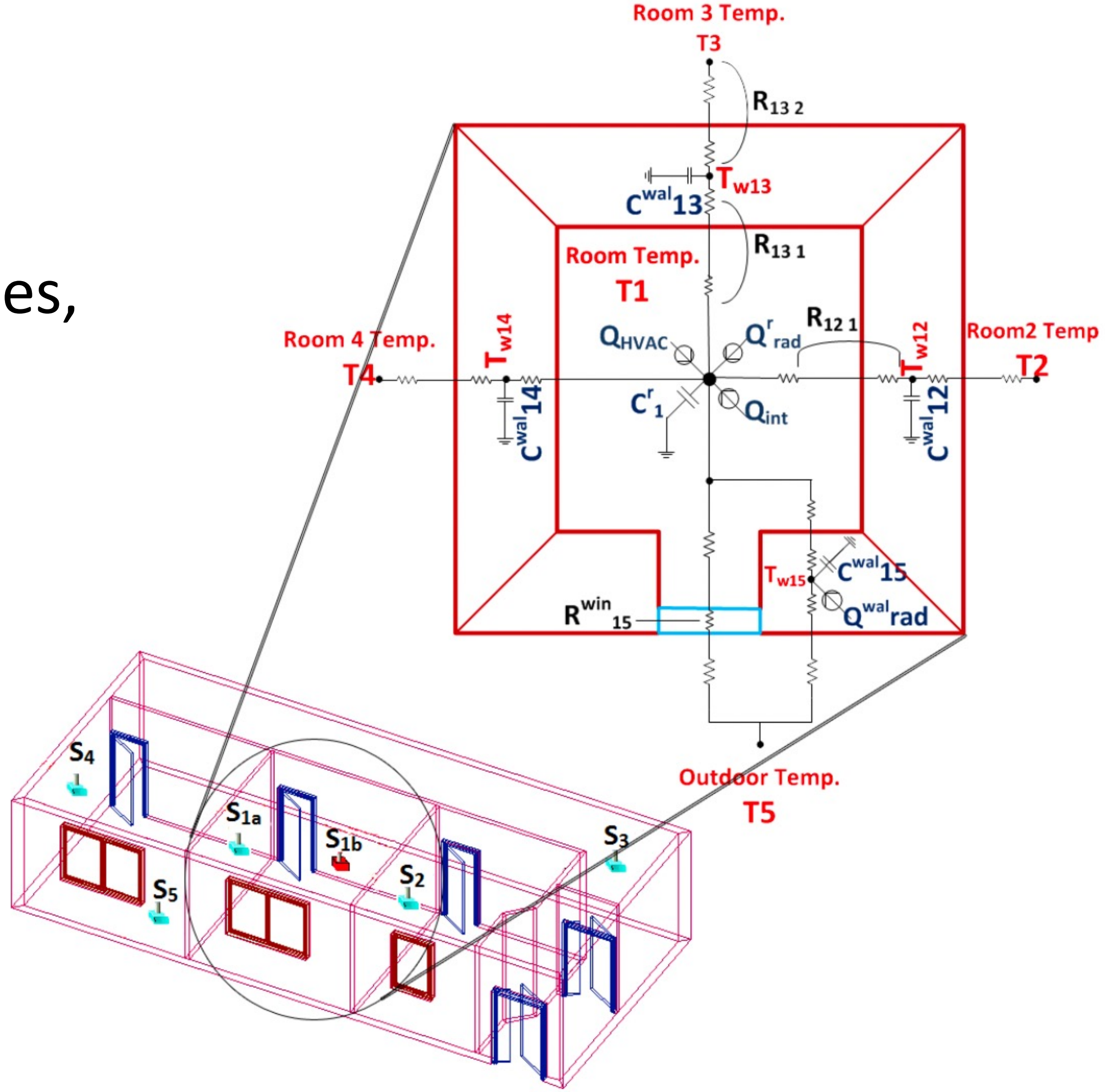
Transient Phase applies until an initial control sequence of length H has been computed

Stationary Phase

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

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

$$C_{i}^{r}\frac{dT_{r_i}}{dt}=\sum_{k\in\mathcal{N}_{r_i}}\frac{T_k-T_{r_i}}{R_{i,k_i}}+\dot{m}_{r_i}c_a(T_{s_i}-T_{r_i})+$$

$$w_i\tau_{w_i}A_{win_i}Q_{rad_i}+\dot{Q}_{int_i}, \tag{7}$$

Building Climate Model

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

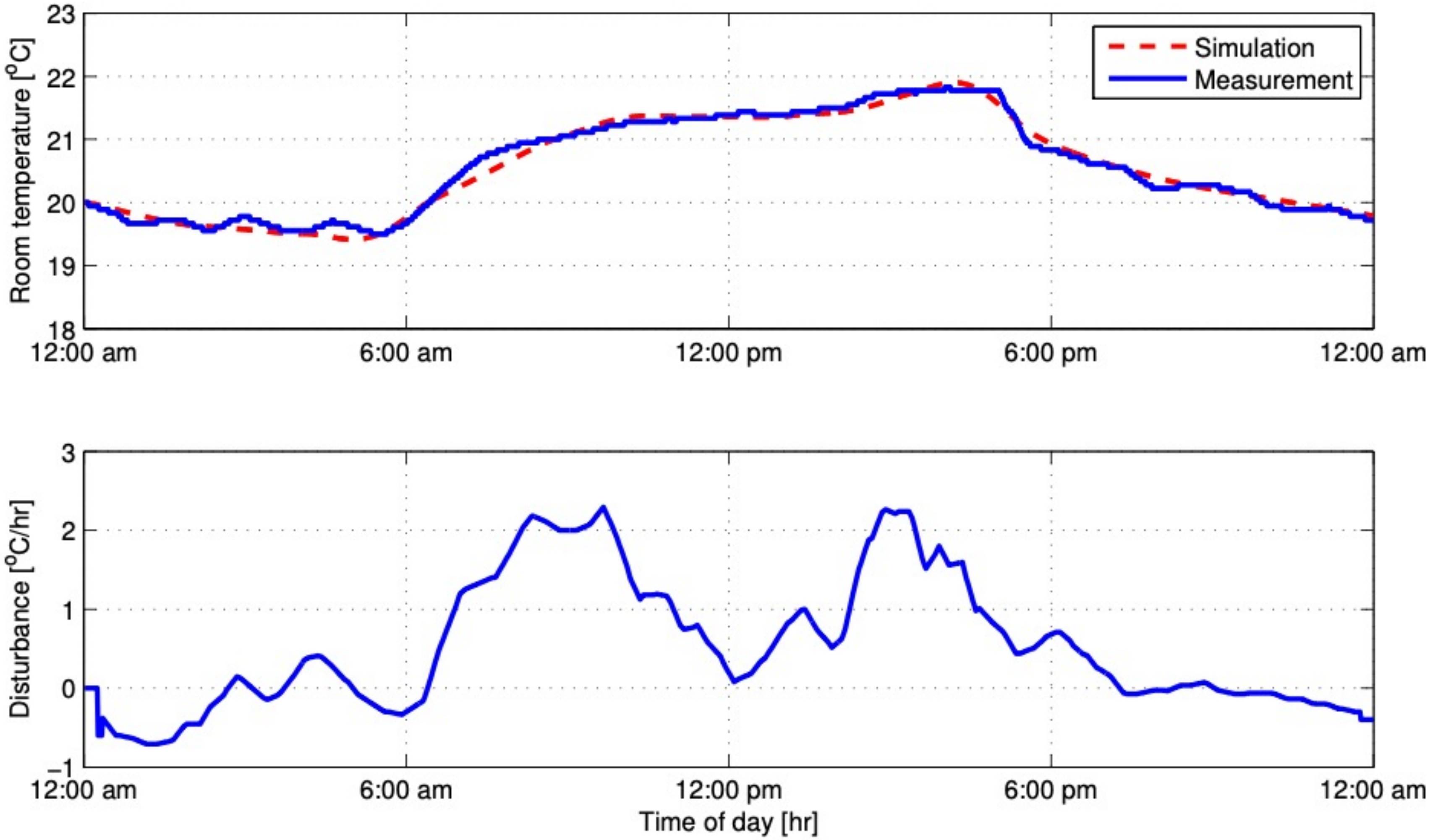
Where state vector x_t representing the temperature of the nodes in the thermal network

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.

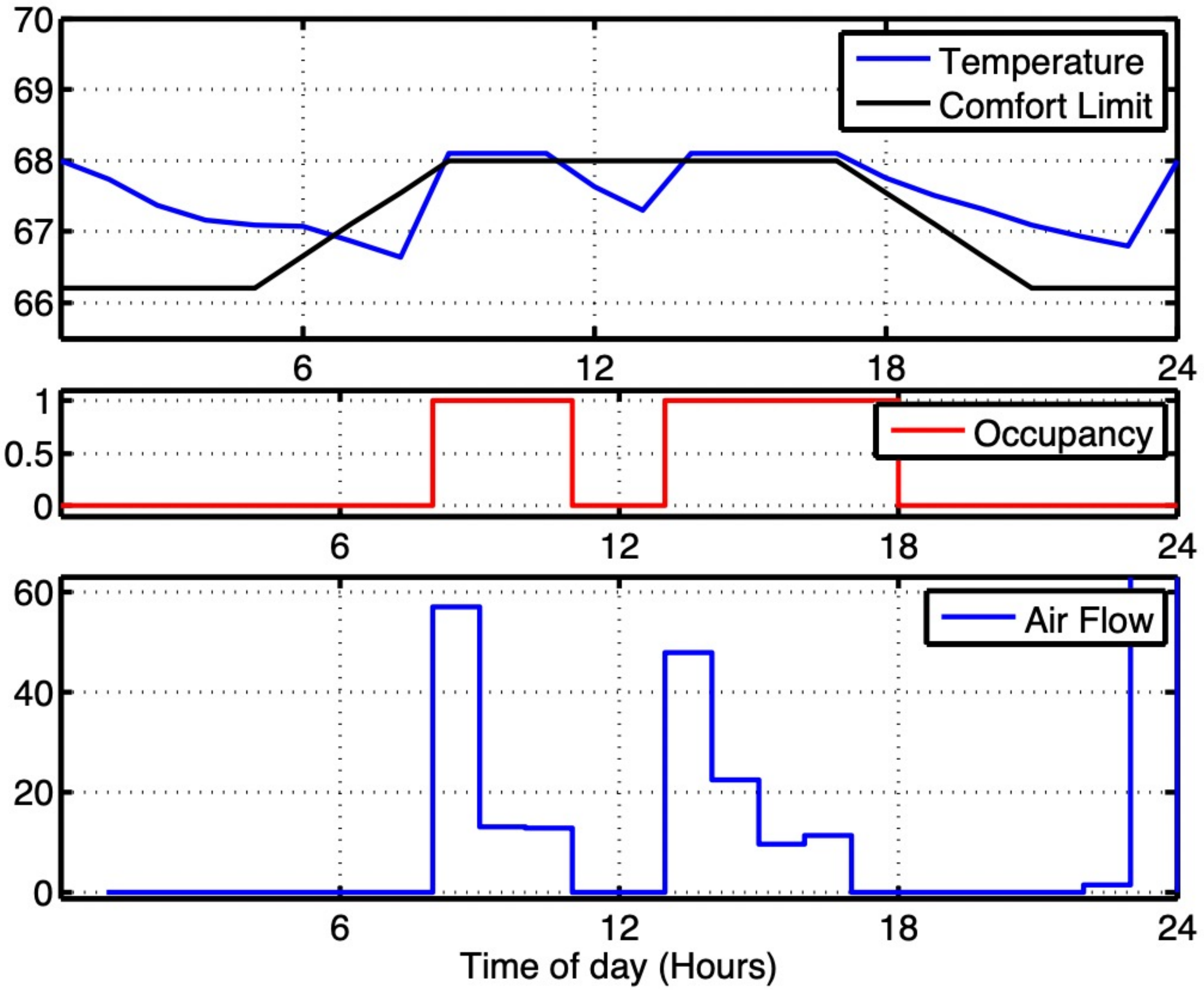
$$\min_{\vec{u}_t} \sum_{k=0}^{H-1} \|u_{t+k}\| \quad \text{s.t.}$$

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

$$x_t \models \varphi \quad \textbf{with} \quad \varphi = \Box_{[0,H]}((\text{occ}_t > 0) \Rightarrow (T_t > T_t^{\text{comf}})$$

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

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[1] and Vasumathi R

[1] Department of Electrical Engineering and Comp
UC Berkeley,
Berkeley, CA 94720, USA donze@berkeley
[2] California Institute of Technology, Pasadena, CA, USA

Abstract

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

2.1 System dynamics

We consider a continuous-time system Σ 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 Σ into Σ_d of the form

$$x(t_{k+1}) = A^d x(t_k) + B^d_u u(t_k) + B^d_w w(t_k) \quad (3)$$
$$y(t_k) = C^d x(t_k) + D^d_u u(t_k) + D^d_w w(t_k) \quad (4)$$

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-uple 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 ξ is called a run of Σ_d.

Limitations and other recent works

- **Solving MILP problems is NP-hard**
- **Only physical systems with linear dynamics**