Safety control with performance guarantees of cooperative systems using compositional abstractions

Pierre-Jean Meyer Antoine Girard Emmanuel Witrant

University of Grenoble, France

SDH-CPNL, June 9^{th} 2015

Outline

System description

Nonlinear control system:

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

- x: state
- u: control input
- w: disturbance input

Trajectories:

$$\Phi(\cdot, x_0, \mathbf{u}, \mathbf{w})$$

- x_0: initial state
- $\mathbf{u}$, $\mathbf{w}$: control and disturbance functions
- $\Phi(t, x_0, \mathbf{u}, \mathbf{w})$: state reached at time t

Bounded inputs:

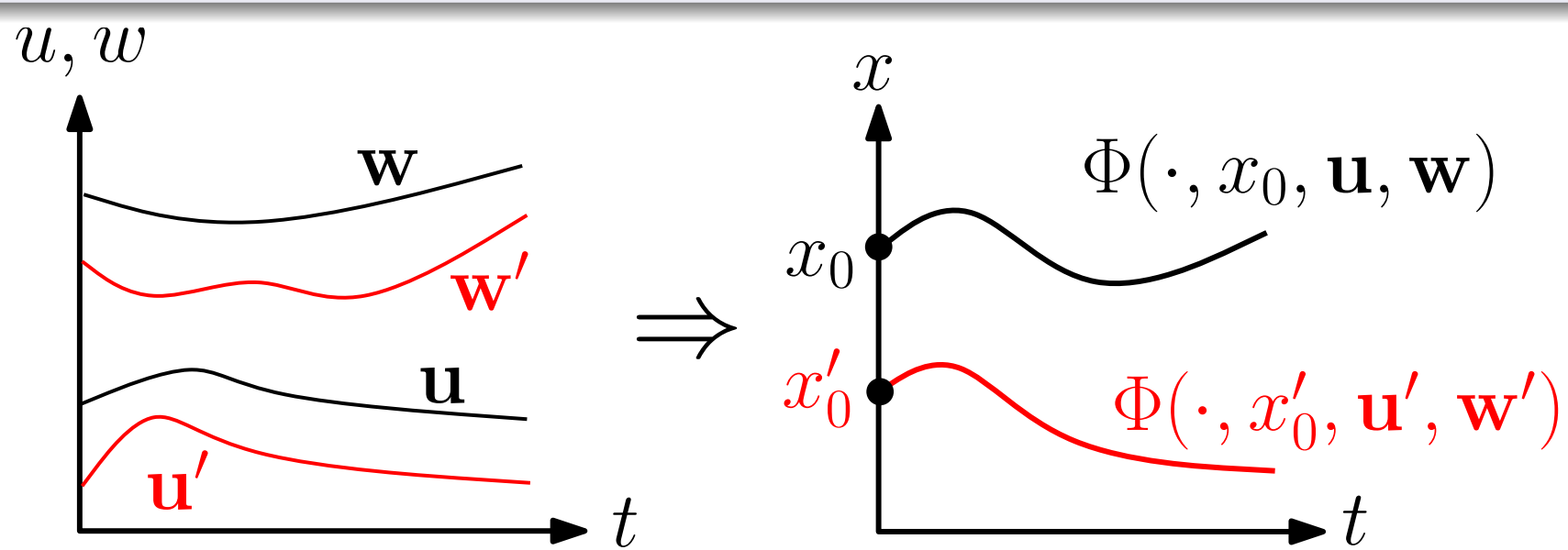
$$u \in [\underline{u}, \overline{u}], \;\; w \in [\underline{w}, \overline{w}]$$

Cooperative system

Definition (Cooperativeness)

The system is cooperative if Φ preserves the componentwise inequality:

$$\mathbf{u} \geq \mathbf{u}', \ \mathbf{w} \geq \mathbf{w}', \ x_0 \geq x_0' \Rightarrow \ \forall t \geq 0, \ \Phi(t, x, \mathbf{u}, \mathbf{w}) \geq \Phi(t, x', \mathbf{u}', \mathbf{w}')$$



Cooperative systems can be characterized using partial derivatives of f (Angeli and Sontag, 2003).

Outline

Problem formulation

Sampled dynamics (constant period τ)

$\longrightarrow$ non-deterministic transition system S:

$$x \xrightarrow{u} x' \Leftrightarrow \exists \mathbf{w} : [0, \tau] \to [\underline{w}, \overline{w}] \mid x' = \Phi(\tau, x, u, \mathbf{w})$$

Safety specification for S:

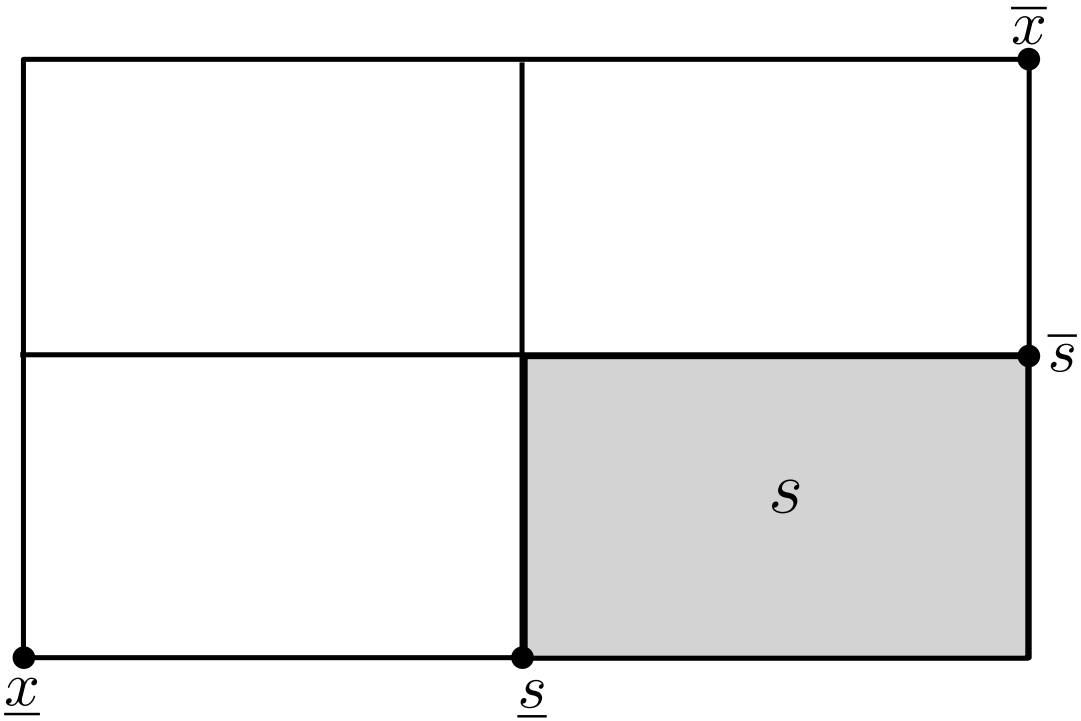
$$\forall k \in \mathbb{N},\ x(k\tau) \in [\underline{x}, \overline{x}]$$

Goal: synthesize a safety controller of S using an abstraction of the system

Symbolic abstraction

Goal: get a **finite** abstraction S_a of the sampled system S
All transitions of S must have an equivalence in S_a

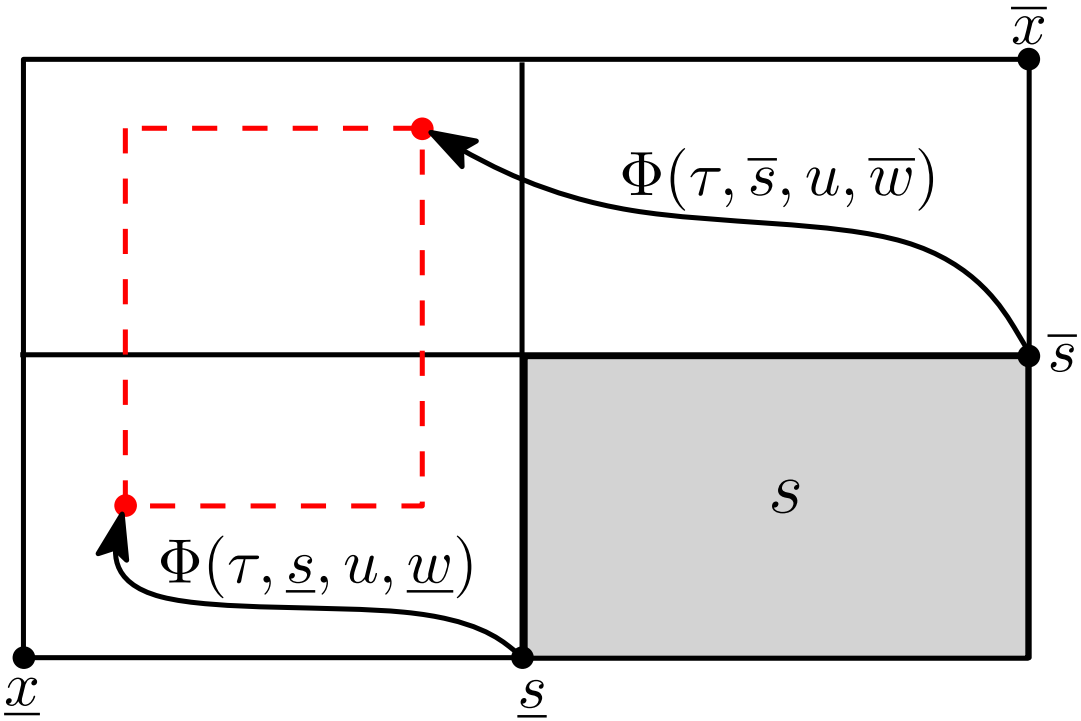
- Discretization of the control space $[\underline{u}, \overline{u}]$
- Partition $\mathcal{P}^0$ of the safe interval $[\underline{x}, \overline{x}]$ into **symbols**



Symbolic abstraction

Goal: get a **finite** abstraction S_a of the sampled system S
All transitions of S must have an equivalence in S_a

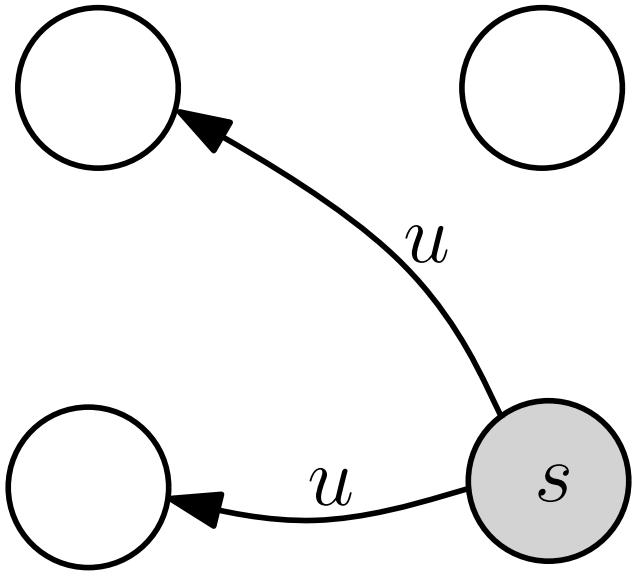
- Discretization of the control space $[\underline{u}, \overline{u}]$
- Partition $\mathcal{P}^0$ of the safe interval $[\underline{x}, \overline{x}]$ into **symbols**
- Over-approximation of the reachable set



Symbolic abstraction

Goal: get a **finite** abstraction S_a of the sampled system S
All transitions of S must have an equivalence in S_a

- Discretization of the control space $[\underline{u}, \overline{u}]$
- Partition $\mathcal{P}^0$ of the safe interval $[\underline{x}, \overline{x}]$ into **symbols**
- Over-approximation of the reachable set
- Intersection with the partition



Safety synthesis

Safety of S_a in the partition $\mathcal{P}^0$
Fixed-point algorithm on the operator:

$$F_{\mathcal{P}^0}(Z) = \{s \in Z \cap \mathcal{P}^0 \mid \exists\ u,\ \forall\ s \xrightarrow[a]{u} s',\ s' \in Z\}$$

Safety synthesis

Safety of S_a in the partition $\mathcal{P}^0$
Fixed-point algorithm on the operator:

$$F_{\mathcal{P}^0}(Z) = \{s \in Z \cap \mathcal{P}^0 \mid \exists\ u,\ \forall\ s \xrightarrow[a]{u} s',\ s' \in Z\}$$

Fixed-point Z_a reached in **finite time**
Z_a is the **maximal safe set** for S_a, associated with the safety controller :

$$C_a(s) = \{u \mid \forall\ s \xrightarrow[a]{u} s',\ s' \in Z_a\}$$

Theorem

C_a is a safety controller for S in Z_a.

Performance criteria

Performance criterion on a trajectory $(x^0, u^0, x^1, u^1, \dots)$ of S:

$$\sum_{k=0}^{+\infty} \lambda^k g(x^k, u^k)$$

with a cost function g and a **discount factor** $\lambda \in (0, 1)$

Performance criteria

Performance criterion on a trajectory $(x^0, u^0, x^1, u^1, \dots)$ of S:

$$\sum_{k=0}^{+\infty} \lambda^k g(x^k, u^k)$$

with a cost function g and a **discount factor** $\lambda \in (0, 1)$

Cost function on S_a: $g_a(s, u) = \max_{x \in s} g(x, u)$

Finite horizon of N sampling periods

Approximation of the performance criterion, if $\lambda^{N+1} \ll 1$:

$$\sum_{k=0}^{N} \lambda^k g_a(s^k, u^k)$$

Performance optimization

Dynamic programming algorithm:

$$J_a^N(s) = \min_{u \in C_a(s)} g_a(s,u)$$

$$J_a^k(s) = \min_{u \in C_a(s)} \left(g_a(s,u) + \lambda \max_{s \xrightarrow[a]{u} s'} J_a^{k+1}(s') \right), \ \forall k < N$$

$J_a^0(s)$ is the **worst-case minimization** of $\sum_{k=0}^{N} \lambda^k g_a(s^k, u^k)$

Performance optimization

Dynamic programming algorithm:

$$J_a^N(s) = \min_{u \in C_a(s)} g_a(s,u)$$

$$J_a^k(s) = \min_{u \in C_a(s)} \left(g_a(s,u) + \lambda \max_{s \xrightarrow[a]{u} s'} J_a^{k+1}(s') \right), \ \forall k < N$$

$J_a^0(s)$ is the **worst-case minimization** of $\sum_{k=0}^{N} \lambda^k g_a(s^k, u^k)$

Receding horizon controller:

$$C_a^*(s) = \arg\min_{u \in C_a(s)} \left(g_a(s,u) + \lambda \max_{s \xrightarrow[a]{u} s'} J_a^1(s') \right)$$

Performance guarantee

Let $M_a = \max_{s \in Z_a} \min_{u \in C_a(s)} g_a(s,u)$

Theorem

Let $(x^0, u^0, x^1, u^1, \dots)$ be a trajectory of S controlled with C_a^*.
Let $s^0, s^1, \dots$ such that $x^k \in s^k$, for all $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$,

$$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq J_a^0(s^k) + \frac{\lambda^{N+1}}{1-\lambda} M_a.$$

Performance guarantee

Let $M_a = \max_{s \in Z_a} \min_{u \in C_a(s)} g_a(s,u)$

Theorem

Let $(x^0, u^0, x^1, u^1, \dots)$ be a trajectory of S controlled with C_a^*.
Let $s^0, s^1, \dots$ such that $x^k \in s^k$, for all $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$,

$$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq J_a^0(s^k) + \frac{\lambda^{N+1}}{1-\lambda} M_a.$$

Worst-case optimization on finite horizon:

$$\sum_{j=0}^{N} \lambda^j g_a(s^{k+j}, u^{k+j}) \leq J_a^0(s^k)$$

Performance guarantee

Let $M_a = \max_{s \in Z_a} \min_{u \in C_a(s)} g_a(s,u)$

Theorem

Let $(x^0, u^0, x^1, u^1, \dots)$ be a trajectory of S controlled with C_a^*.
Let $s^0, s^1, \dots$ such that $x^k \in s^k$, for all $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$,

$$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq J_a^0(s^k) + \frac{\lambda^{N+1}}{1-\lambda} M_a.$$

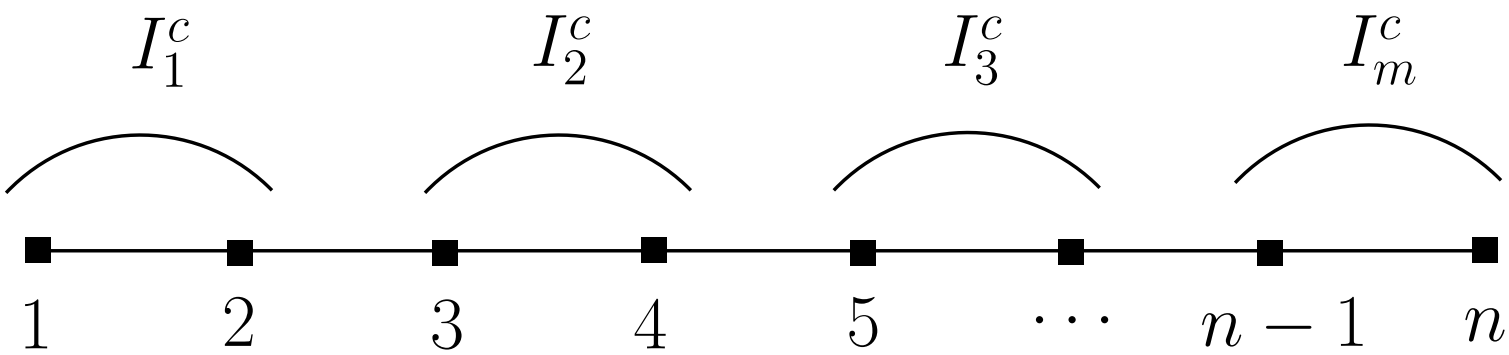
Worst-case of remaining cost, with receding horizon
we minimize at least the current step:

$$\sum_{j=N+1}^{+\infty} \lambda^j g_a(s^{k+j}, u^{k+j}) \leq \sum_{j=N+1}^{+\infty} \lambda^j \max_{s \in Z_a} \min_{u \in C_a(s)} g_a(s,u) = \frac{\lambda^{N+1}}{1-\lambda} M_a.$$

Outline

Decomposition: states

Decomposition into m subsystems:
Partition $(I_1^c, \ldots, I_m^c)$ of the state dimensions $\{1, \ldots, n\}$

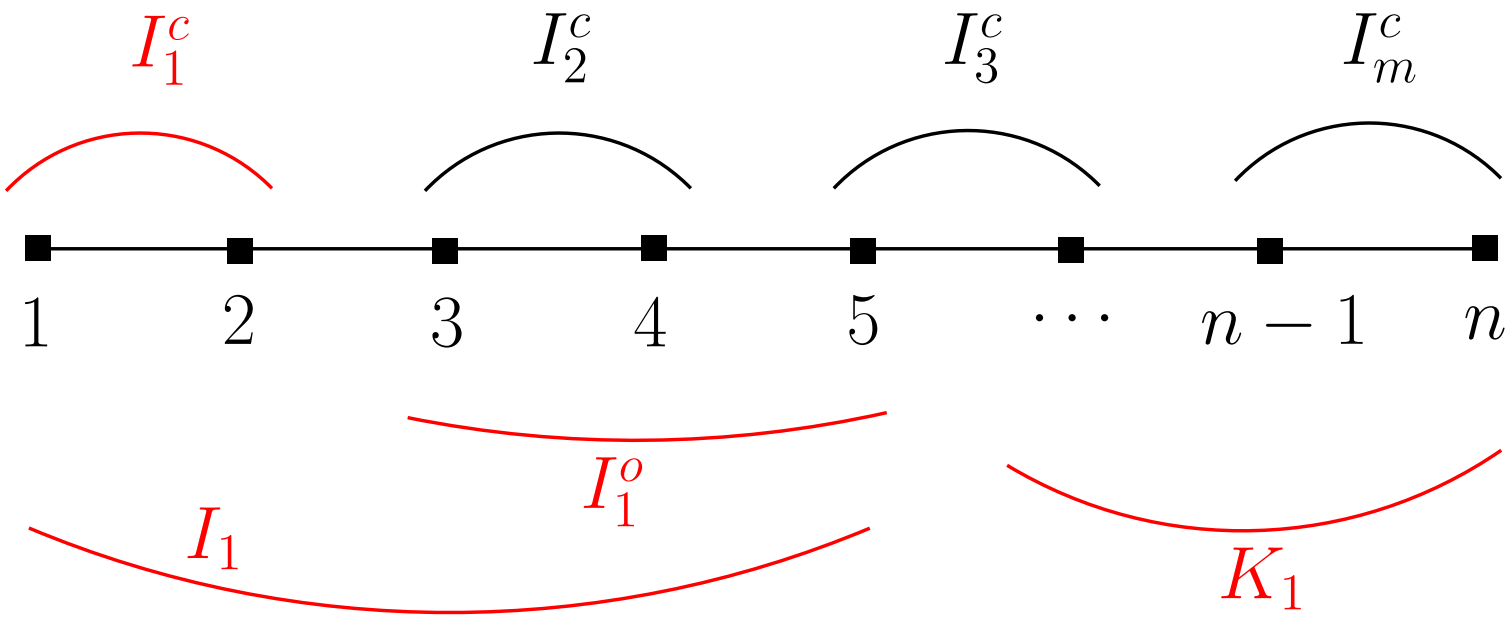


Decomposition: states

Decomposition into m subsystems:
Partition $(I_1^c, \ldots, I_m^c)$ of the state dimensions $\{1, \ldots, n\}$

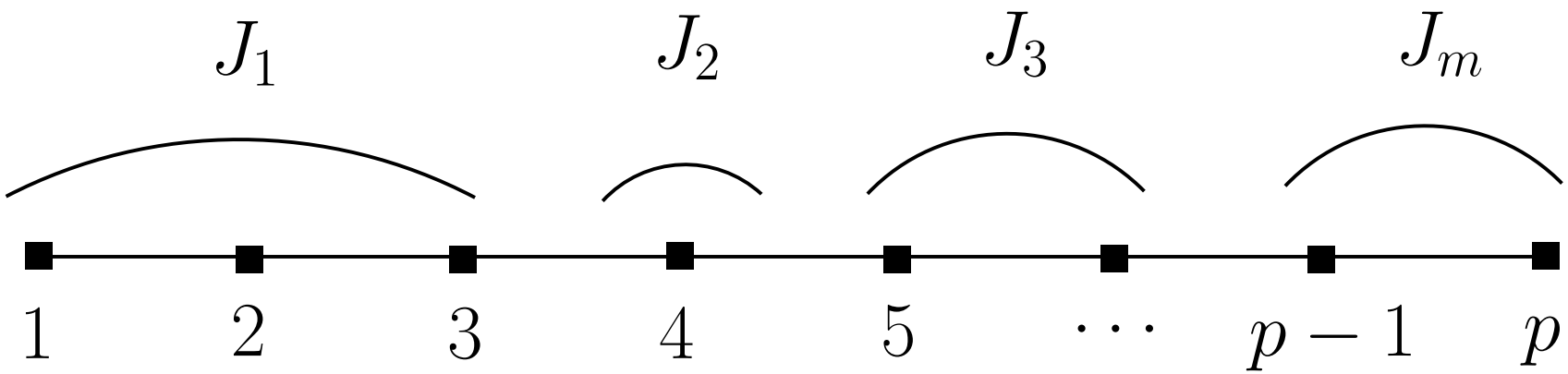
Subsystem $i \in \{1, \ldots, m\}$:

- I_i^c: controlled states
- I_i^o: observed but uncontrolled states
- $I_i = I_i^c \cup I_i^o$: all modeled states
- K_i: unobserved states (disturbances)



Decomposition: inputs

Decomposition into m subsystems:
Partition $(J_1, \ldots, J_m)$ of the input dimensions $\{1, \ldots, p\}$

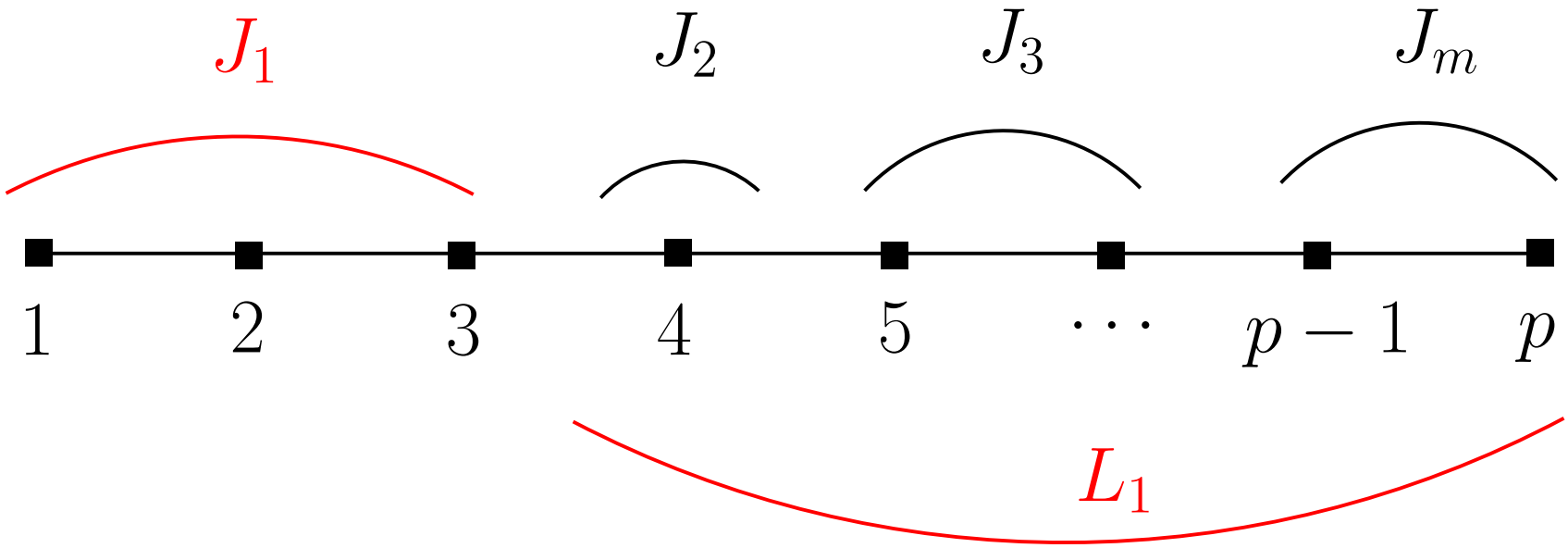


Decomposition: inputs

Decomposition into m subsystems:
Partition $(J_1, \ldots, J_m)$ of the input dimensions $\{1, \ldots, p\}$

Subsystem $i \in \{1, \ldots, m\}$:

- J_i: controlled inputs
- L_i: unobserved inputs (disturbances)



Subsystem abstraction

Subsystem $i \in \{1, \ldots, m\}$: create a symbolic abstraction S_i
Classical method, but with 2 **assume-guarantee obligations**:

Subsystem abstraction

Subsystem $i \in \{1, \ldots, m\}$: create a symbolic abstraction S_i
Classical method, but with 2 **assume-guarantee obligations**:

A/G Obligation (K_i)

Unobserved states: $x_{K_i} \in [\underline{x}_{K_i}, \overline{x}_{K_i}]$

Need **bounded disturbances**: w, x_{K_i}, u_{L_i}

For cooperative system $\Rightarrow$ larger over-approximation of the reachable set

$$\begin{cases} \Phi(\tau, (\underline{s}_{I_i}, \underline{x}_{K_i}), (u_{J_i}, \underline{u}_{L_i}), \underline{w}) \leq \Phi(\tau, \underline{s}, u, \underline{w}), \\ \Phi(\tau, (\overline{s}_{I_i}, \overline{x}_{K_i}), (u_{J_i}, \overline{u}_{L_i}), \overline{w}) \geq \Phi(\tau, \overline{s}, u, \overline{w}). \end{cases}$$

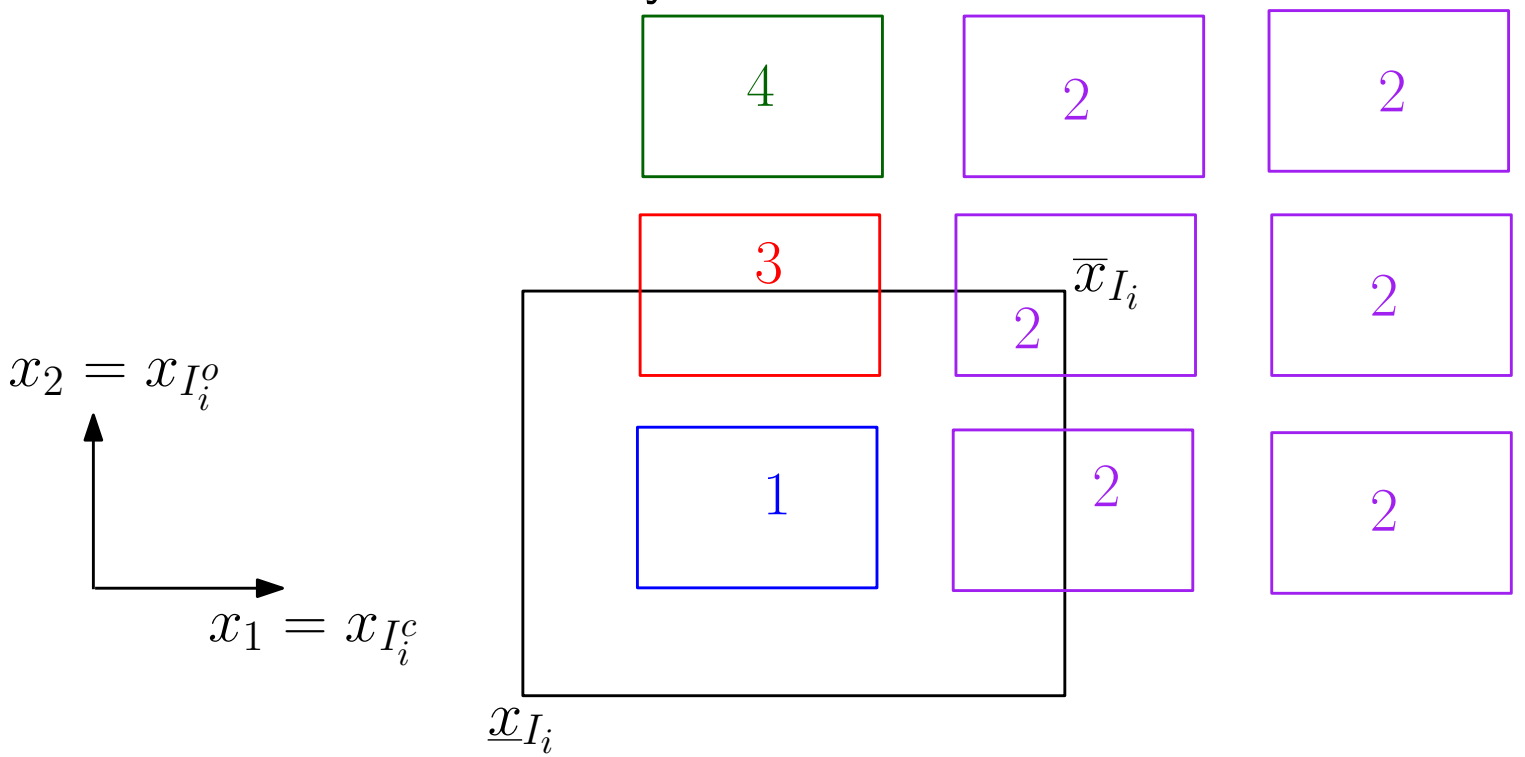
Subsystem abstraction

Subsystem $i \in \{1, \ldots, m\}$: create a symbolic abstraction S_i
Classical method, but with 2 **assume-guarantee obligations**:

A/G Obligation (I_i^o)

Observed but uncontrolled states: $x_{I_i^o} \in [\underline{x}_{I_i^o}, \overline{x}_{I_i^o}]$

Remove transitions where only uncontrolled states violate the safety



Controller synthesis

Same methods than the centralized approach
with a focus only on controlled states: $x_{I_i^c}$

Safety in the partition of $[\underline{x}_{I_i}, \overline{x}_{I_i}]$

- (safety already realized for uncontrolled states $x_{I_i^o}$)
- maximal safe set: Z_i
- safety controller: C_i

Performances

- cost function $g_i(s_{I_i}, u_{J_i})$
- g_i only depends on the controlled components $s_{I_i^c}$
- deterministic controller: C_i^*

Composition

S_c: composition of all subsystems S_i

- Safe transitions: $s \xrightarrow[c]{u} s' \iff \forall i \in \{1, \ldots, m\},\ s_{l_i} \xrightarrow[i]{u_{J_i}} s'_{l_i}$
- Unsafe transition: $s \xrightarrow[c]{u} Out \iff \exists i \in \{1, \ldots, m\} \mid s_{l_i} \xrightarrow[i]{u_{J_i}} Out_i$

Composition

S_c: composition of all subsystems S_i

- Safe transitions: $s \xrightarrow[c]{u} s' \iff \forall i \in \{1, \ldots, m\},\ s_{l_i} \xrightarrow[i]{u_{J_i}} s'_{l_i}$
- Unsafe transition: $s \xrightarrow[c]{u} Out \iff \exists i \in \{1, \ldots, m\} \mid s_{l_i} \xrightarrow[i]{u_{J_i}} Out_i$

Proposition (Alternating simulation)

Let a symbol s, a state $x \in s$ and a discrete input u.
For any successor x' in S ($x \xrightarrow{u} x'$), the corresponding symbol ($s' \mid x' \in s'$) is a successor in S_c ($s \xrightarrow[c]{u} s'$).

Safety

Composition of safe sets and safety controllers:

- $Z_c = Z_1 \Cap \cdots \Cap Z_m$
- $\forall s \in Z_c,\ C_c(s) = C_1(s_{I_1}) \times \cdots \times C_m(s_{I_m})$

Safety

Composition of safe sets and safety controllers:

- $Z_c = Z_1 \Cap \cdots \Cap Z_m$
- $\forall s \in Z_c,\ C_c(s) = C_1(s_{I_1}) \times \cdots \times C_m(s_{I_m})$

Theorem

C_c is a safety controller for S in Z_c.

Safety

Composition of safe sets and safety controllers:

- $Z_c = Z_1 \Cap \cdots \Cap Z_m$
- $\forall s \in Z_c,\ C_c(s) = C_1(s_{I_1}) \times \cdots \times C_m(s_{I_m})$

Theorem

C_c is a safety controller for S in Z_c.

Proposition (Safety comparison)

$Z_c \subseteq Z_a$.

Performances

Composition of the deterministic controllers:

- $\forall s \in Z_c,\ C_c^*(s) = (C_1^*(s_{I_1}), \ldots, C_m^*(s_{I_m}))$
- Let $M_i = \max_{s_i \in Z_i} \min_{u_i \in C_i(s_i)} g_i(s_i, u_i)$

Performances

Composition of the deterministic controllers:

- $\forall s \in Z_c,\ C_c^*(s) = (C_1^*(s_{I_1}), \dots, C_m^*(s_{I_m}))$
- Let $M_i = \max_{s_i \in Z_i} \min_{u_i \in C_i(s_i)} g_i(s_i, u_i)$

Theorem (Performance guarantee)

Let $(x^0, u^0, x^1, u^1, \dots)$ be a trajectory of S controlled with C_c^*.
Let $s^0, s^1, \dots$ such that $x^k \in s^k$, for all $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$,

$$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq \sum_{i=1}^{m} J_i^0(s_{I_i}^k) + \frac{\lambda^{N+1}}{1-\lambda} \sum_{i=1}^{m} M_i.$$

Performances

Composition of the deterministic controllers:

- $\forall s \in Z_c,\ C_c^*(s) = (C_1^*(s_{I_1}), \dots, C_m^*(s_{I_m}))$
- Let $M_i = \max\limits_{s_i \in Z_i} \min\limits_{u_i \in C_i(s_i)} g_i(s_i, u_i)$

Theorem (Performance guarantee)

Let $(x^0, u^0, x^1, u^1, \dots)$ be a trajectory of S controlled with C_c^*.
Let $s^0, s^1, \dots$ such that $x^k \in s^k$, for all $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$,

$$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq \sum_{i=1}^{m} J_i^0(s_{I_i}^k) + \frac{\lambda^{N+1}}{1-\lambda} \sum_{i=1}^{m} M_i.$$

Proposition (Guarantees comparison)

$$\forall s \in Z_c, \qquad J_a^0(s) + \frac{\lambda^{N+1}}{1-\lambda} M_a \leq \sum_{i=1}^{m} J_i^0(s_{I_i}) + \frac{\lambda^{N+1}}{1-\lambda} \sum_{i=1}^{m} M_i$$

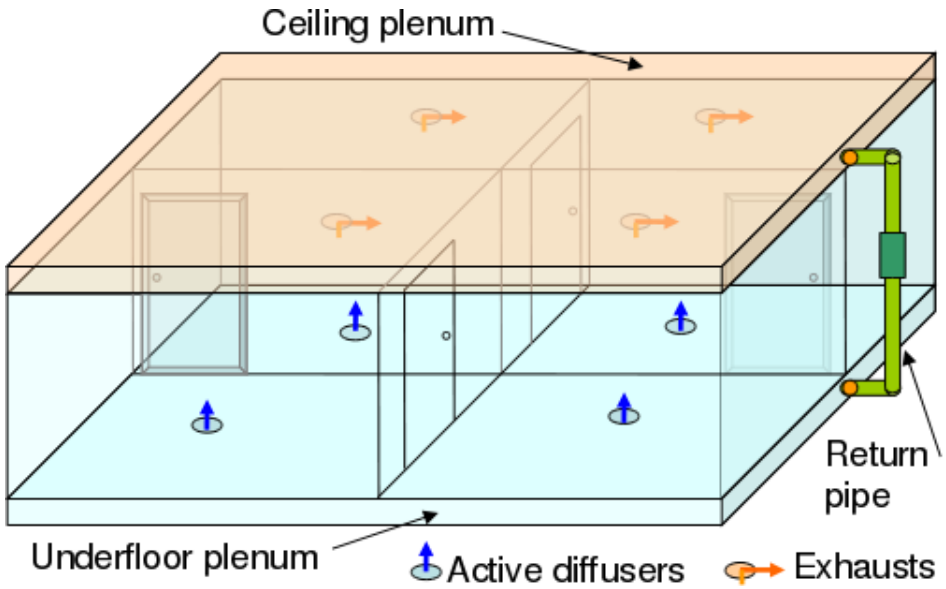
Complexity

- n: state space dimension
- p: control space dimension
- $\alpha_x \in \mathbb{N}$: precision of the state partition (per dimension)
- $\alpha_u \in \mathbb{N}$: precision of the input discretization (per dimension)
- $|\cdot|$: cardinality of a set

	Method	
	Centralized	Compositional
Abstraction (successors computed)	$2\alpha_x^n \alpha_u^p$	$\sum_{i=1}^{m} 2\alpha_x^{\lvert I_i \rvert} \alpha_u^{\lvert J_i \rvert}$
Dynamic programming (max iterations)	$N\alpha_x^{2n} \alpha_u^p$	$\sum_{i=1}^{m} N\alpha_x^{2\lvert I_i \rvert} \alpha_u^{\lvert J_i \rvert}$

Outline

UnderFloor Air Distribution



- Underfloor air cooled down
- Sent into the rooms by fans
- Air excess pushed through the ceiling exhausts
- Returned to the underfloor

- Disturbances: heat sources; opening of doors

Model

Assume a **uniform temperature** in each room
Temperature variations obtained from **energy and mass conservation** equations in each room

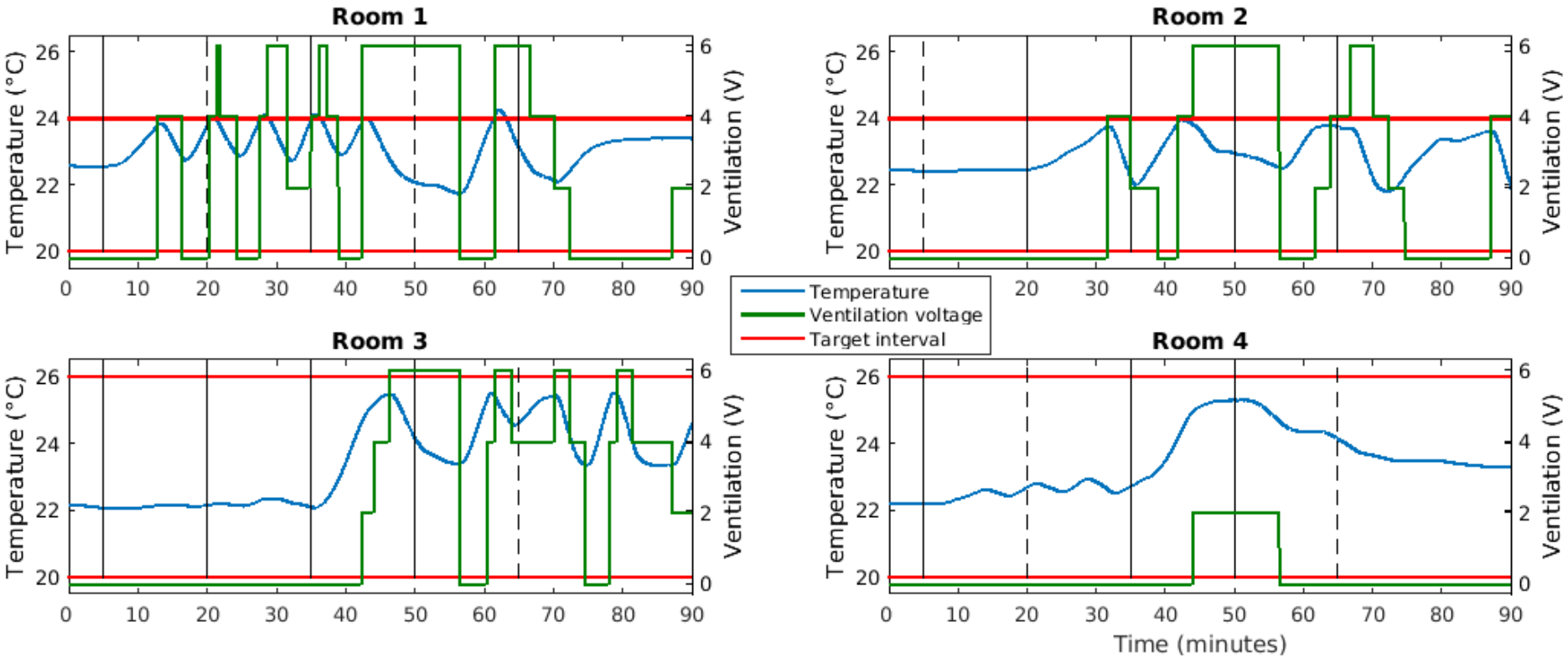
Our model: $\dot{T} = f(T, u, w, \delta)$

- $T \in \mathbb{R}^4$: state (temperature)
- $u \in \mathbb{R}^4$: controlled input (fan air flow)
- w: exogenous input (other temperatures)
- δ: discrete disturbance

The system is **cooperative**

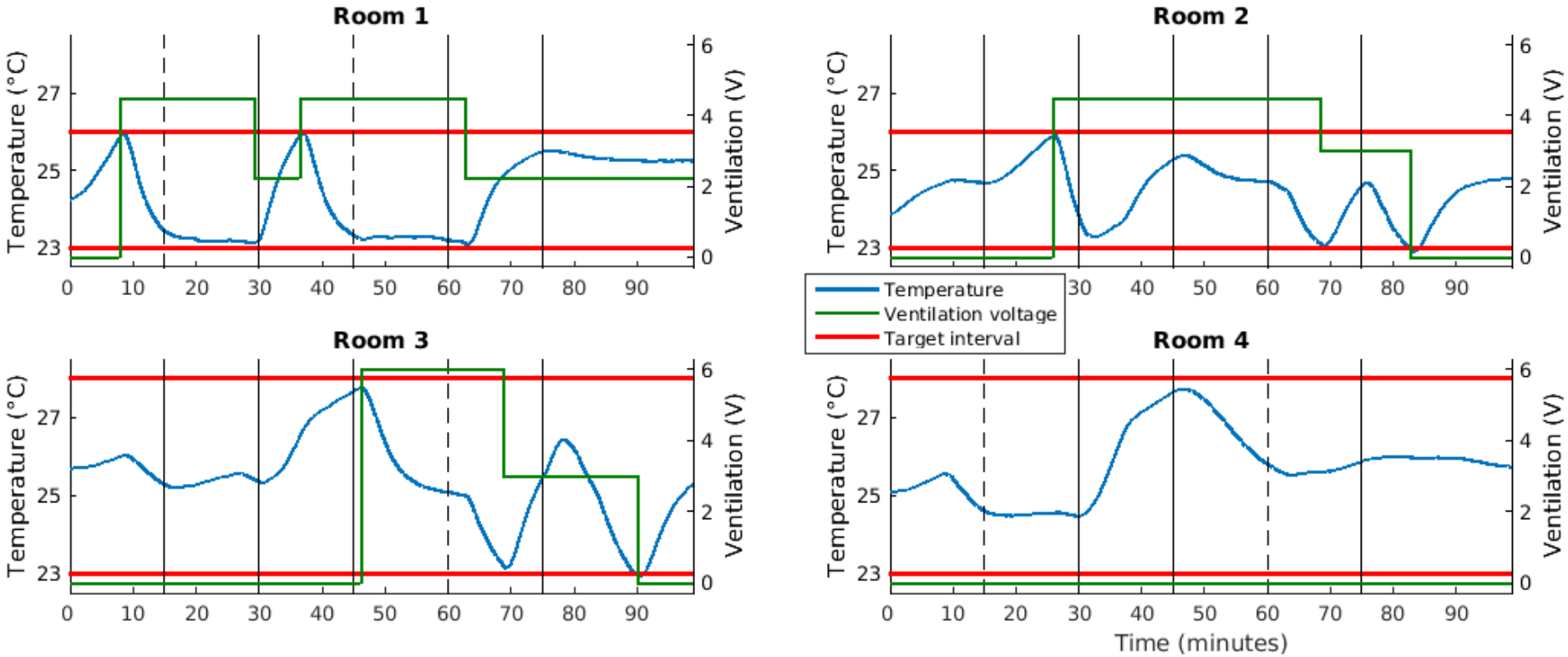
Centralized symbolic control

- $\alpha_x = 10$, $\alpha_u = 4$, $\tau = 34\ s$
- $g_a(s^k, u^k, u^{k-1}) = \|u^k\| + \|u^k - u^{k-1}\| + \|s_*^k - T_*\|$
- $N = 5$, $\lambda = 0.5$: $\lambda^{N+1} \approx 3\%$
- Computation time: more than 2 days



Compositional symbolic control

- $1D$ subsystems: $I_i = I_i^c = J_i = i$ and $I_i^o = \emptyset$
- $\alpha_x = 20$, $\alpha_u = 9$, $\tau = 10\ s$
- Computation time: $1.1\ s$



Conclusion

Compositional approach to addresses scalability issue of symbolic control:

- Decomposition into partial description of the system
- Tradeoff between model accuracy and complexity reduction
- Classical symbolic methods applied to each subsystem under assume-guarantee obligations
- Composition preserves safety and performance guarantees

Conclusion

Compositional approach to addresses scalability issue of symbolic control:

- Decomposition into partial description of the system
- Tradeoff between model accuracy and complexity reduction
- Classical symbolic methods applied to each subsystem under assume-guarantee obligations
- Composition preserves safety and performance guarantees

Perspectives

- Extension to observed but uncontrolled inputs
- Adaptive symbolic control framework:
  - measure the disturbance; tight estimation of its future bounds
  - synthesize compositional controller on the more accurate abstraction
  - apply controller until the next measure

Safety control with performance guarantees of cooperative systems using compositional abstractions

Pierre-Jean Meyer Antoine Girard Emmanuel Witrant

University of Grenoble, France

SDH-CPNL, June 9^{th} 2015