

# Sandboxing Controllers for Stochastic Cyber-Physical Systems

**Bingzhuo Zhong**, Technical University of Munich, Germany

**Majid Zamani**, CU Boulder, USA & Ludwig Maximilian University of Munich, Germany

**Marco Caccamo**, Technical University of Munich, Germany

FORMATS 2019, Amsterdam

August 29, 2019



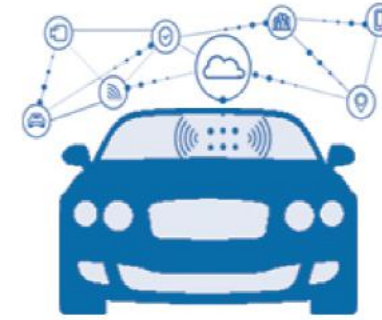
# Motivation



Unmanned Aerial Vehicles



Internet of Things

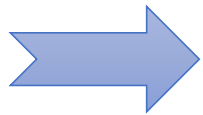


Autonomous Driving

In modern cyber-physical systems, lots of high performance, but unverified controllers are required to be used for complex tasks, e.g. deep neural network.

To ensure the safety, we exploit the idea of **sandbox** from the community of computer security.

- **(Isolation)** Restrict the behaviour of the untrusted component by isolating it from the critical part of a digital controller.
- **(Supervision)** It can only access the critical part when it follows the rules given by the sandboxing mechanism.



**Sandboxing unverified controllers for functionality and safety**



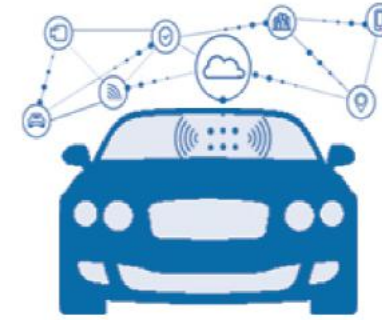
# Motivation



Unmanned Aerial Vehicles

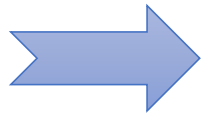


Internet of Things



Autonomous Driving

In modern cyber-physical systems, lots of high performance, but unverified controllers are required to be used for complex tasks, e.g. deep neural network.



**Sandboxing unverified controllers for functionality and safety**

In this work, we focus on

- Discrete-time, stochastic systems, i.e.,  $x(t+1) = f(x(t), u(t), \omega(t))$ , where  $\omega(t)$  is a **sequence of (independent and) identical distributed random variables**, possibly unbounded.
- A typical specification: invariance.



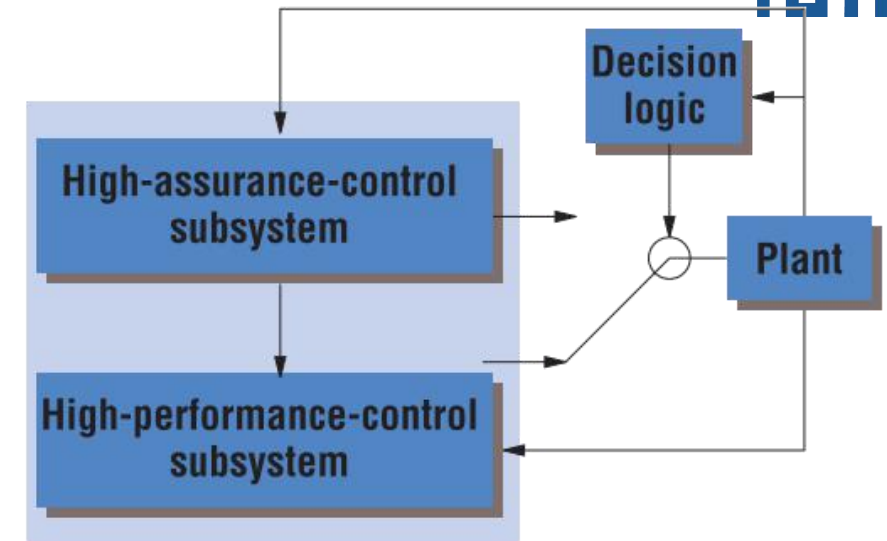
## Basic idea

### Safety advisor:

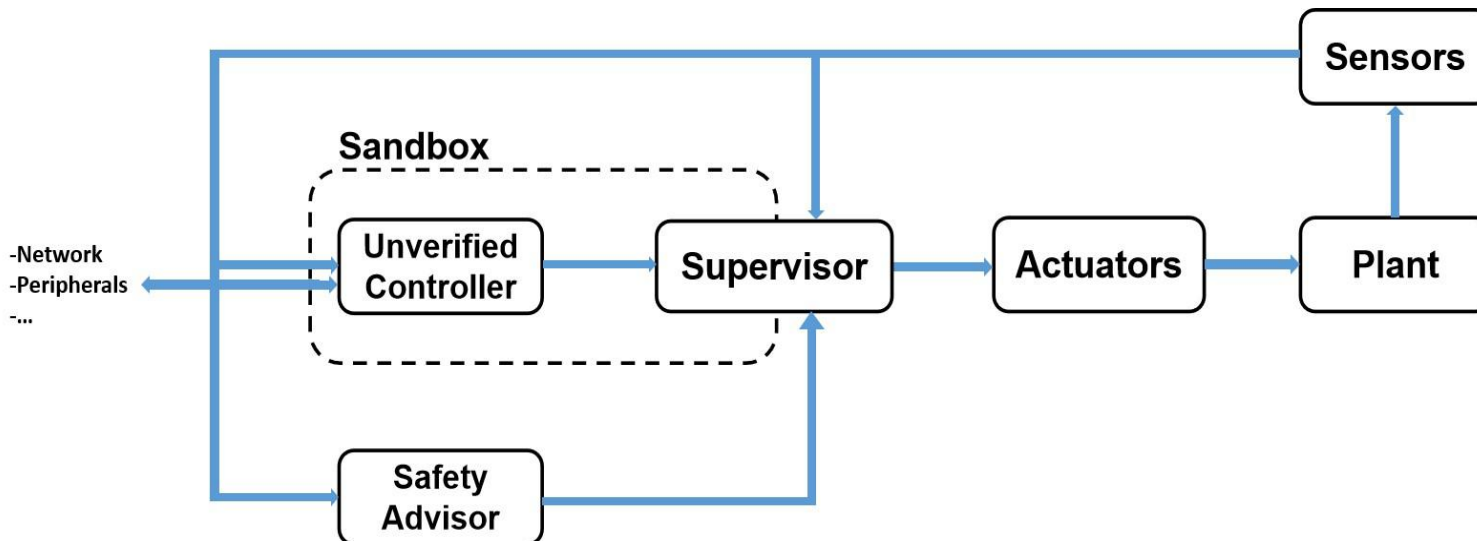
- Only focus on safety, aim at maximizing the probability of safety

### Supervisor:

- Check inputs from the unverified controller
- Feeding input provided by safety advisor as fallback action once input from the unverified control is hazardous



**Simplex architecture**




**Safe-visor architecture**

### Novelties:

- Stochastic systems
- Providing probabilistic guarantee for fulfilling safety specification
- More flexible for compromise between safety probability and functionality

# Definition

## Discrete time stochastic system

$$x(t+1) = f(x(t), u(t), \omega(t))$$


## Controlled discrete time Markov process

$$\mathfrak{D} = (X, U, \underbrace{\{U(x)\}_{x \in X}}_{\text{Set of Input executable at state } x}, T_{\mathfrak{D}})$$

State space      Input space      Borel-measurable stochastic kernel

$$T_{\mathfrak{D}}(x, u, x') = \mathbb{P}(x(k+1) = x' | x(k) = x, u(k) = u)$$

We focus on the case where  $U(X) = U$ .

Invariance specification: The system is expected to stay within a safety set.


For controlled discrete time Markov process:

- Figure out Markov policy which
  - maximize the possibility for the system staying in the safety set or
  - minimize the possibility for the system reaching the unsafety set in finite time horizon.



# Definition

## Discrete time stochastic system

$$x(t+1) = f(x(t), u(t), \omega(t))$$


## Controlled discrete time Markov process

$$\mathfrak{D} = (X, U, \underbrace{\{U(x)\}_{x \in X}}_{\text{Set of Input executable at state } x}, T_{\mathfrak{D}})$$

State space      Input space      Borel-measurable stochastic kernel

$$T_{\mathfrak{D}}(x, u, x') = \mathbb{P}(x(k+1) = x' | x(k) = x, u(k) = u)$$

We focus on the case where  $U(X) = U$ .

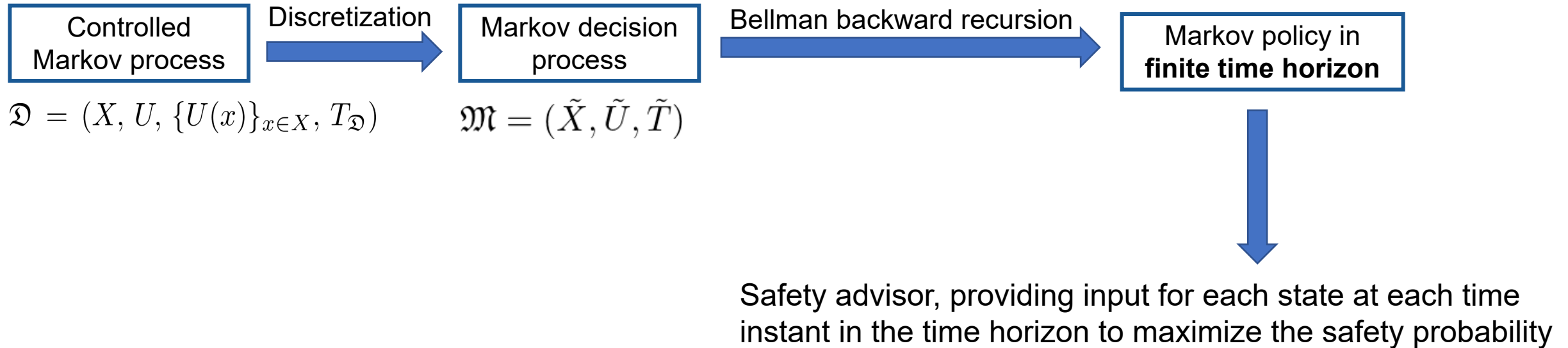
Invariance specification: The system is expected to stay within a safety set.

For controlled discrete time Markov process:

- Figure out Markov policy which
  - maximize the possibility for the system staying in the safety set or
  - minimize the possibility for the system reaching the unsafety set
 in finite time horizon.



# Safety Advisor

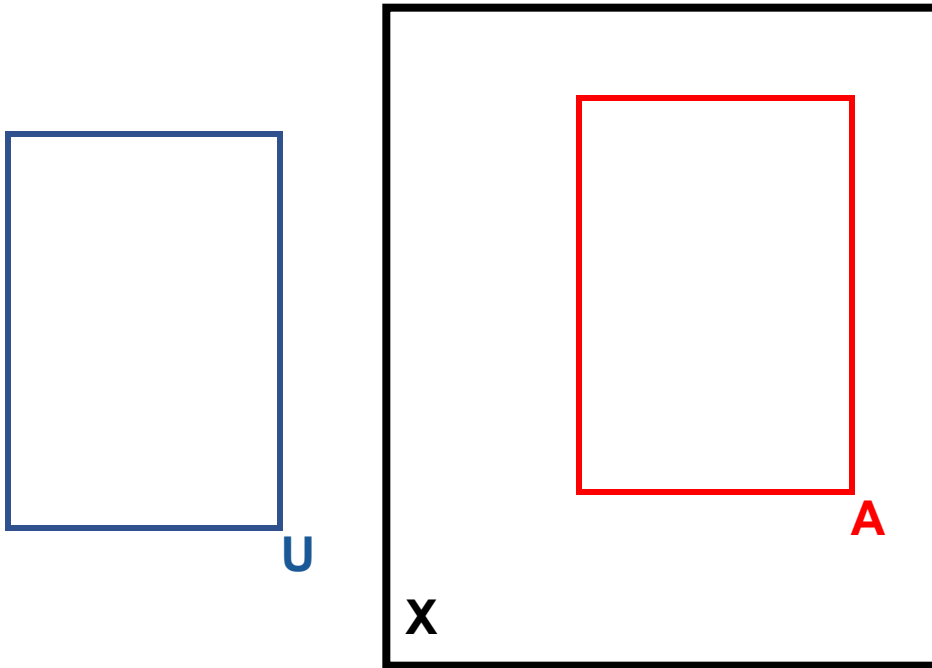
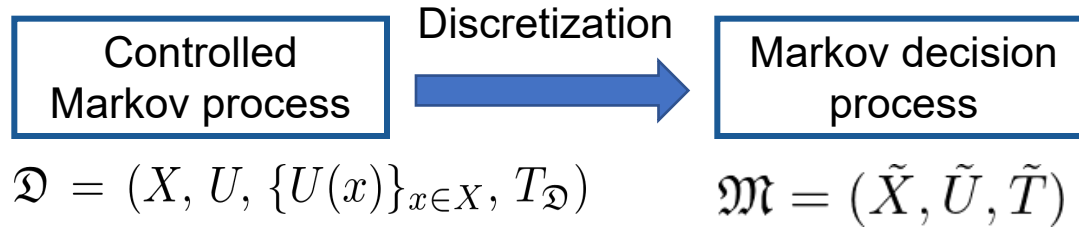


## Remarks:

- Length of the time horizon is tunable regarding the selected maximal tolerable probability of reaching unsafe states.

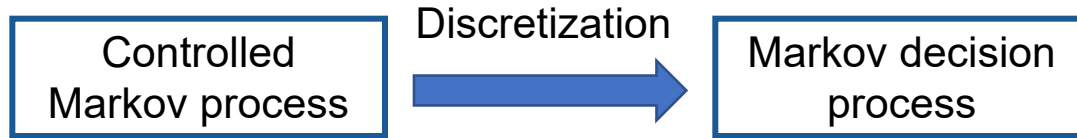


# Discretization of Controlled Markov process



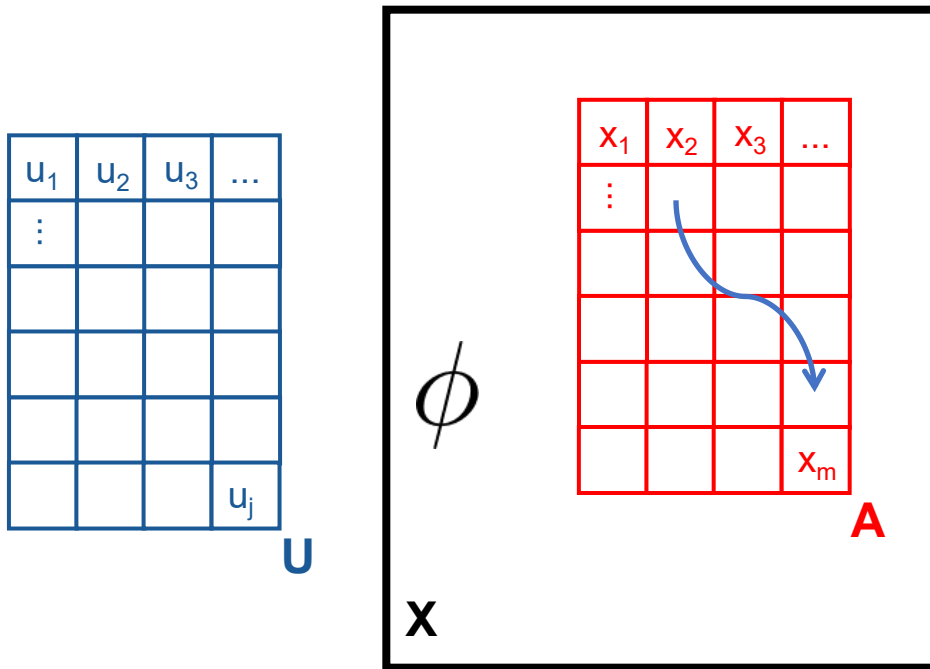


# Discretization of Controlled Markov process



$$\mathfrak{D} = (X, U, \{U(x)\}_{x \in X}, T_{\mathfrak{D}})$$

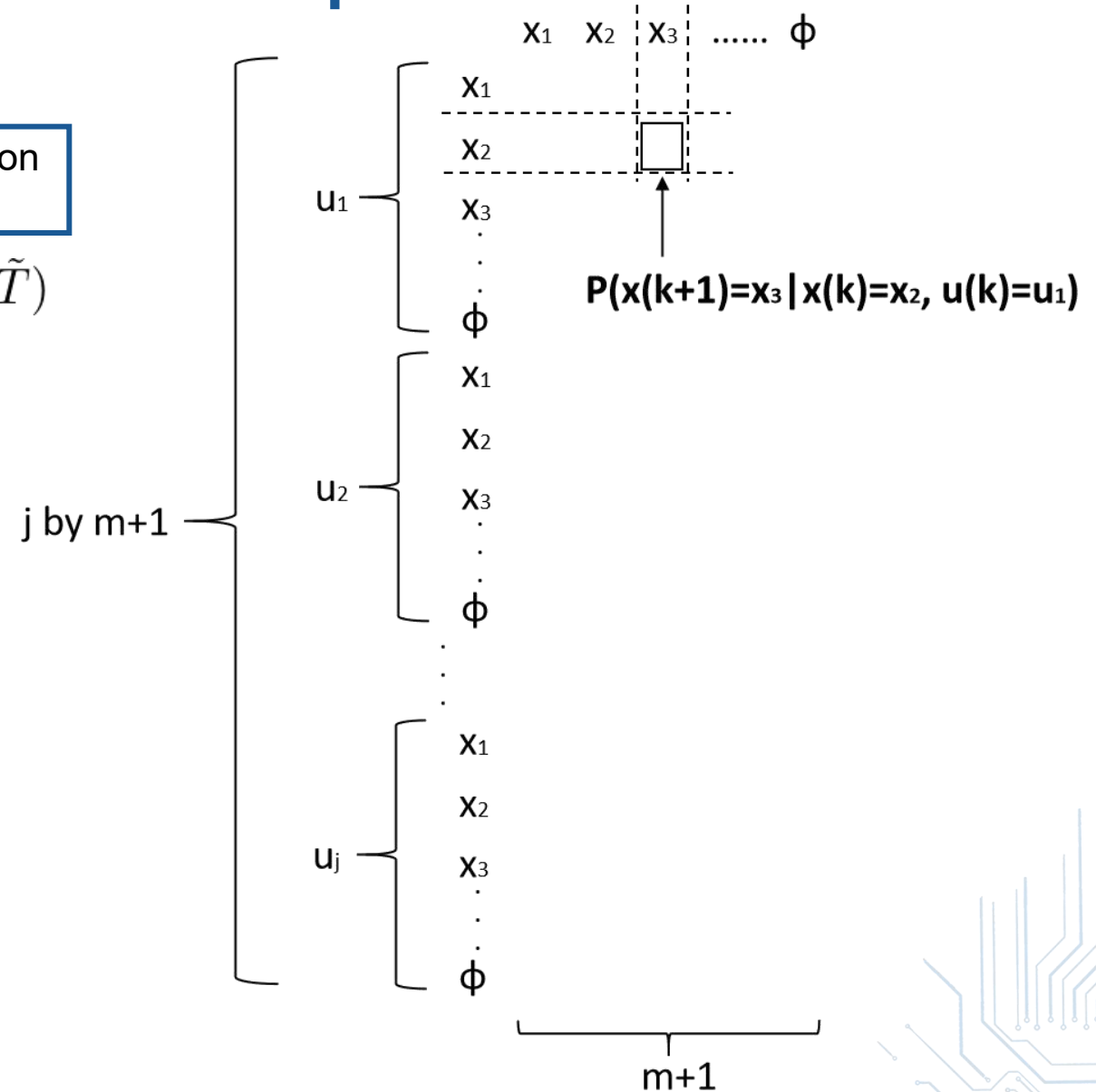
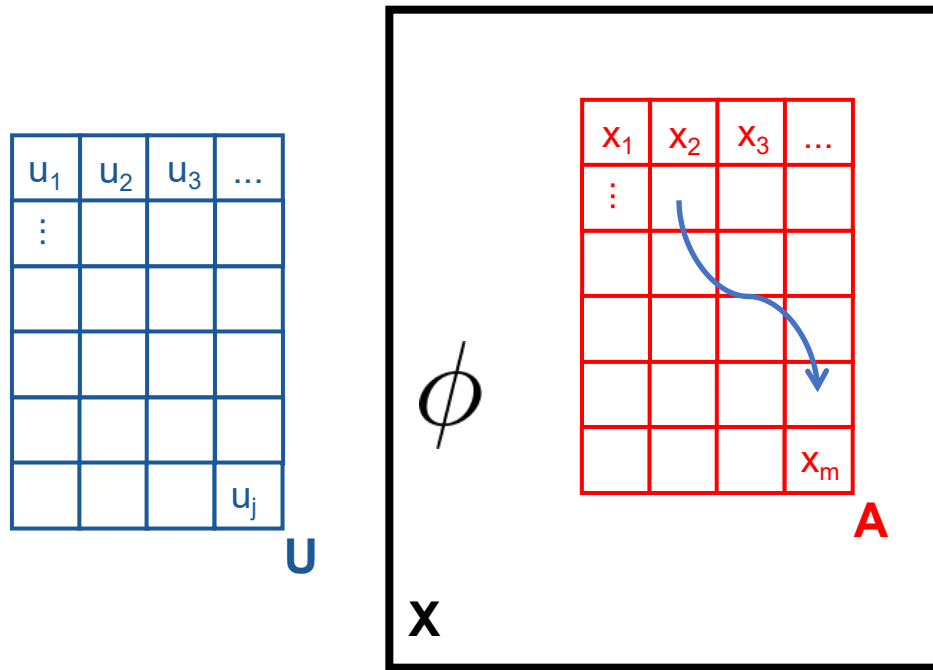
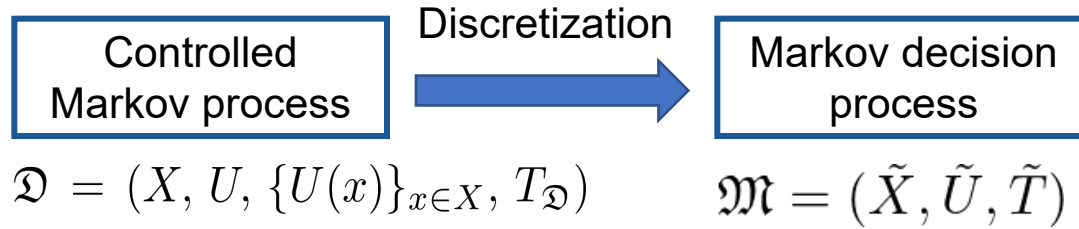
$$\mathfrak{M} = (\tilde{X}, \tilde{U}, \tilde{T})$$



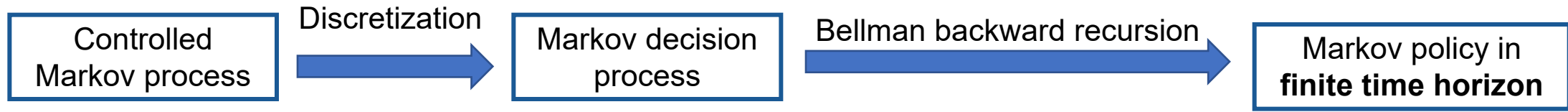
$$\tilde{T}(\tilde{x}_m | \tilde{x}_i, \tilde{u}_j) = \begin{cases} T_{\mathfrak{D}}(\tilde{X}_m | \tilde{x}_i, \tilde{u}_j) & \text{if } \tilde{x}_i, \tilde{x}_m \in \{\tilde{x}_i\}_{i=1}^N, \tilde{u}_j \in \tilde{U} \\ T_{\mathfrak{D}}(\mathcal{A}^c | \tilde{x}_i, \tilde{u}_j) & \text{if } \tilde{x}_i \in \{\tilde{x}_i\}_{i=1}^N, \tilde{x}_m \in \{\phi\}, \tilde{u}_j \in \tilde{U} \\ 1 & \text{if } \tilde{x}_i, \tilde{x}_m \in \{\phi\}, \tilde{u}_j \in \tilde{U} \\ 0 & \text{if } \tilde{x}_i \in \{\phi\}, \tilde{x}_m \in \{\tilde{x}_i\}_{i=1}^N, \tilde{u}_j \in \tilde{U} \end{cases}$$

sink state

# Discretization of Controlled Markov process



# Markov Policy in finite time horizon



$$\mathfrak{D} = (X, U, \{U(x)\}_{x \in X}, T_{\mathfrak{D}}) \quad \mathfrak{M} = (\tilde{X}, \tilde{U}, \tilde{T})$$

Given a time horizon  $H$ , the safety advisor (Markov Policy in finite time horizon) for the finite MDP is a matrix as the following:

$x_1$	$\mu_{*,0}(x_1)$	$\mu_{*,1}(x_1)$	$\mu_{*,2}(x_1)$	$\mu_{*,3}(x_1)$	.....	$\mu_{*,H-2}(x_1)$	$\mu_{*,H-1}(x_1)$
$x_2$	$\mu_{*,0}(x_2)$	$\mu_{*,1}(x_2)$	$\mu_{*,2}(x_2)$	$\mu_{*,3}(x_2)$	.....	$\mu_{*,H-2}(x_2)$	$\mu_{*,H-1}(x_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_{m-1}$	$\mu_{*,0}(x_{m-1})$	$\mu_{*,1}(x_{m-1})$	$\mu_{*,2}(x_{m-1})$	$\mu_{*,3}(x_{m-1})$	.....	$\mu_{*,H-2}(x_{m-1})$	$\mu_{*,H-1}(x_{m-1})$
$x_m$	$\mu_{*,0}(x_m)$	$\mu_{*,1}(x_m)$	$\mu_{*,2}(x_m)$	$\mu_{*,3}(x_m)$	.....	$\mu_{*,H-2}(x_m)$	$\mu_{*,H-1}(x_m)$
$\phi$	$\mu_{*,0}(\phi)$	$\mu_{*,1}(\phi)$	$\mu_{*,2}(\phi)$	$\mu_{*,3}(\phi)$	.....	$\mu_{*,H-2}(\phi)$	$\mu_{*,H-1}(\phi)$
	$t=0$	$t=1$	$t=2$	$t=3$	.....	$t=H-2$	$t=H-1$

where  $\forall k \in \overline{0, H}, \tilde{x} \in \tilde{X}, \mu_{*,k}(\tilde{x}) \in \tilde{U}$  Fill in all entries of the matrix.

# Markov Policy in finite time horizon

$x_1$	$\mu_{*,0}(x_1)$	$\mu_{*,1}(x_1)$	$\mu_{*,2}(x_1)$	$\mu_{*,3}(x_1)$	.....	$\mu_{*,H-2}(x_1)$	$\mu_{*,H-1}(x_1)$
$x_2$	$\mu_{*,0}(x_2)$	$\mu_{*,1}(x_2)$	$\mu_{*,2}(x_2)$	$\mu_{*,3}(x_2)$	.....	$\mu_{*,H-2}(x_2)$	$\mu_{*,H-1}(x_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_{m-1}$	$\mu_{*,0}(x_{m-1})$	$\mu_{*,1}(x_{m-1})$	$\mu_{*,2}(x_{m-1})$	$\mu_{*,3}(x_{m-1})$	.....	$\mu_{*,H-2}(x_{m-1})$	$\mu_{*,H-1}(x_{m-1})$
$x_m$	$\mu_{*,0}(x_m)$	$\mu_{*,1}(x_m)$	$\mu_{*,2}(x_m)$	$\mu_{*,3}(x_m)$	.....	$\mu_{*,H-2}(x_m)$	$\mu_{*,H-1}(x_m)$
$\phi$	$\mu_{*,0}(\phi)$	$\mu_{*,1}(\phi)$	$\mu_{*,2}(\phi)$	$\mu_{*,3}(\phi)$	.....	$\mu_{*,H-2}(\phi)$	$\mu_{*,H-1}(\phi)$
	t=0	t=1	t=2	t=3	.....	t=H-2	t=H-1

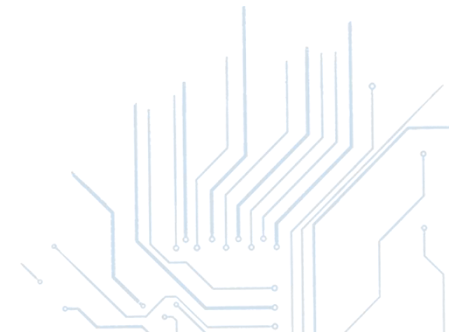
To determine the proper input in each entry, the following value function is introduced:

$$\tilde{V}_{*,n+1}(\tilde{x}) = 1_{\{\phi\}}(\tilde{x}) + 1_{\{\phi\}^c}(\tilde{x}) \min_{\tilde{u} \in \tilde{U}} \sum_{\tilde{y} \in \tilde{X}} \tilde{V}_{*,n}(\tilde{y}) \tilde{T}(\tilde{y}|\tilde{x}, \tilde{u})$$

initialized with  $\tilde{V}_{*,0} = 1_{\{\phi\}}(\tilde{x})$ .

Then the safety advisor can be recursively synthesized as the following:

$$\mu_{*,H-n-1}(\tilde{x}) \in \arg \min_{\tilde{\mu}_{H-n-1}} \sum_{\tilde{y} \in \tilde{X}} (1_{\{\phi\}^c}(\tilde{y}) + 1_{\{\phi\}}(\tilde{y}) \tilde{V}_{*,n}(\tilde{y})) \tilde{T}(d\tilde{y}|\tilde{x}, \tilde{\mu}_{H-n-1}(\tilde{x}))$$



# Markov Policy in finite time horizon

$x_1$	$\mu_{*,0}(x_1)$	$\mu_{*,1}(x_1)$	$\mu_{*,2}(x_1)$	$\mu_{*,3}(x_1)$	.....	$\mu_{*,H-2}(x_1)$	$\mu_{*,H-1}(x_1)$
$x_2$	$\mu_{*,0}(x_2)$	$\mu_{*,1}(x_2)$	$\mu_{*,2}(x_2)$	$\mu_{*,3}(x_2)$	.....	$\mu_{*,H-2}(x_2)$	$\mu_{*,H-1}(x_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_{m-1}$	$\mu_{*,0}(x_{m-1})$	$\mu_{*,1}(x_{m-1})$	$\mu_{*,2}(x_{m-1})$	$\mu_{*,3}(x_{m-1})$	.....	$\mu_{*,H-2}(x_{m-1})$	$\mu_{*,H-1}(x_{m-1})$
$x_m$	$\mu_{*,0}(x_m)$	$\mu_{*,1}(x_m)$	$\mu_{*,2}(x_m)$	$\mu_{*,3}(x_m)$	.....	$\mu_{*,H-2}(x_m)$	$\mu_{*,H-1}(x_m)$
$\phi$	$\mu_{*,0}(\phi)$	$\mu_{*,1}(\phi)$	$\mu_{*,2}(\phi)$	$\mu_{*,3}(\phi)$	.....	$\mu_{*,H-2}(\phi)$	$\mu_{*,H-1}(\phi)$
	t=0	t=1	t=2	t=3	.....	t=H-2	t=H-1

$$\tilde{V}_{*,n+1}(\tilde{x}) = 1_{\{\phi\}}(\tilde{x}) + 1_{\{\phi\}^c}(\tilde{x}) \min_{\tilde{u} \in \tilde{U}} \sum_{\tilde{y} \in \tilde{X}} \tilde{V}_{*,n}(\tilde{y}) \tilde{T}(\tilde{y}|\tilde{x}, \tilde{u})$$

initialized with  $\tilde{V}_{*,0} = 1_{\{\phi\}}(\tilde{x})$ . The safety advisor

$$\mu_{*,H-n-1}(\tilde{x}) \in \arg \min_{\tilde{\mu}_{H-n-1}} \sum_{\tilde{y} \in \tilde{X}} (1_{\{\phi\}^c}(\tilde{y}) + 1_{\{\phi\}}(\tilde{y}) \tilde{V}_{*,n}(\tilde{y})) \tilde{T}(d\tilde{y}|\tilde{x}, \tilde{\mu}_{H-n-1}(\tilde{x}))$$

Remarks:  $V_{*,n}(x)$  indicates the probability of reaching the unsafe set within  $\overline{0, n}$ , i.e.,

$$V_{*,n}(x) = \inf_{\pi \in \Pi} P_x^\pi(\diamond^{\leq n} \mathcal{A}^c)$$



# Markov Policy in finite time horizon

$x_1$	$\mu_{*,0}(x_1)$	$\mu_{*,1}(x_1)$	$\mu_{*,2}(x_1)$	$\mu_{*,3}(x_1)$	.....	$\mu_{*,H-2}(x_1)$	$\mu_{*,H-1}(x_1)$
$x_2$	$\mu_{*,0}(x_2)$	$\mu_{*,1}(x_2)$	$\mu_{*,2}(x_2)$	$\mu_{*,3}(x_2)$	.....	$\mu_{*,H-2}(x_2)$	$\mu_{*,H-1}(x_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_{m-1}$	$\mu_{*,0}(x_{m-1})$	$\mu_{*,1}(x_{m-1})$	$\mu_{*,2}(x_{m-1})$	$\mu_{*,3}(x_{m-1})$	.....	$\mu_{*,H-2}(x_{m-1})$	$\mu_{*,H-1}(x_{m-1})$
$x_m$	$\mu_{*,0}(x_m)$	$\mu_{*,1}(x_m)$	$\mu_{*,2}(x_m)$	$\mu_{*,3}(x_m)$	.....	$\mu_{*,H-2}(x_m)$	$\mu_{*,H-1}(x_m)$
$\phi$	$\mu_{*,0}(\phi)$	$\mu_{*,1}(\phi)$	$\mu_{*,2}(\phi)$	$\mu_{*,3}(\phi)$	.....	$\mu_{*,H-2}(\phi)$	$\mu_{*,H-1}(\phi)$
	t=0	t=1	t=2	t=3	.....	t=H-2	t=H-1

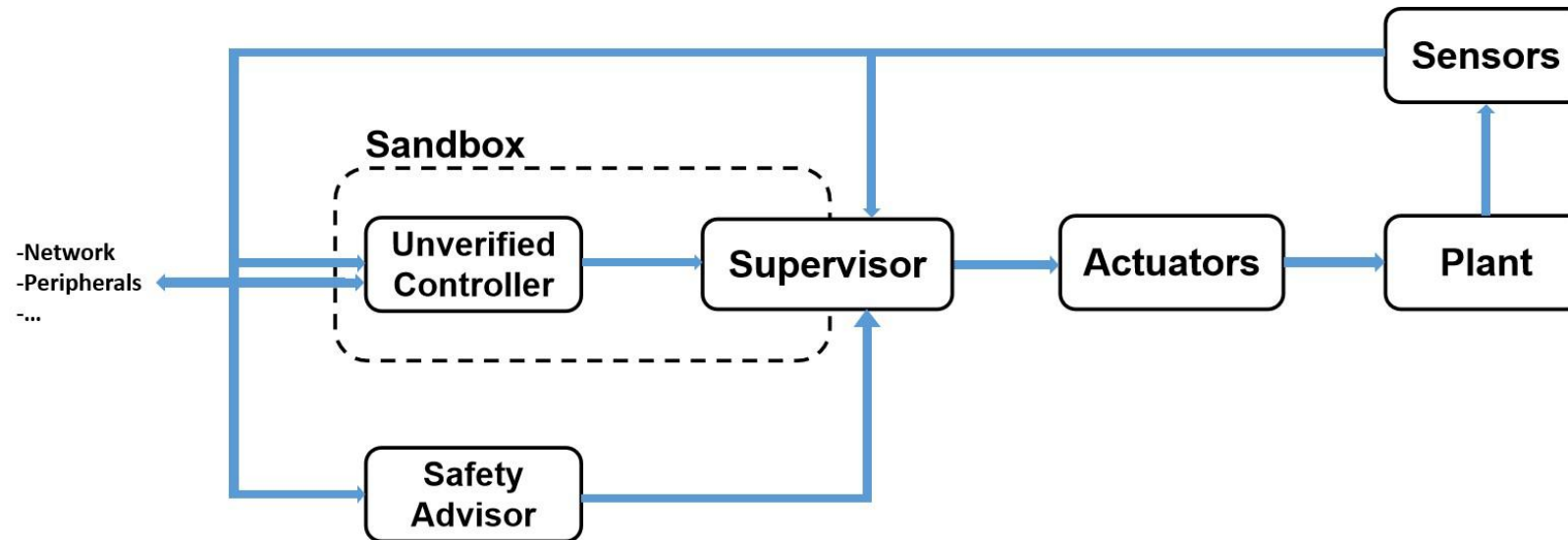
In our implementation, the time horizon  $\overline{0, H}$  of the Safety Advisor is determined in a way such that:

$$\forall \tilde{x} \in \tilde{X} \setminus \{\phi\}, \tilde{V}_{*,H}(\tilde{x}) \leq \rho \text{ and } \exists \tilde{x} \in \tilde{X} \setminus \{\phi\}, \tilde{V}_{*,H+1}(\tilde{x}) > \rho$$

where  $\rho$  is the maximal tolerable probability of reaching the unsafe set.



# History-based Supervisor



Key idea: at **every time instant** during the execution, check the feasibility of the inputs from unverified controller based on history path.

Example: at time  $t = k$ , the history path up to time  $t = k$  is:

$$\omega = (\omega_x(0), \omega_u(0), \omega_x(1), \omega_u(1), \dots, \omega_x(k-1), \omega_u(k-1), \omega_x(k))$$

where  $\omega_x(t) = x(t)$  and  $\omega_u(t) = u(t)$ .



# History-based Supervisor

At time  $t = k$ , given the history path up to time  $t = k$ :

$$\omega = (\omega_x(0), \omega_u(0), \omega_x(1), \omega_u(1), \dots, \omega_x(k-1), \omega_u(k-1), \omega_x(k))$$

current input given by the unverified controller can only be accepted when the following inequality holds:

$$\underbrace{\prod_{t=1}^k \sum_{\tilde{x} \in \tilde{X} \setminus \{\phi\}} \tilde{T}(\tilde{x} | \omega_x(t-1), \omega_u(t-1))}_{\text{Noise is (i.i.d.) random variable (or } \tilde{T}_t)} \left( 1 - \sum_{\tilde{x} \in \tilde{X}} \underbrace{\tilde{V}_{*, H-k-1}(\tilde{x})}_{\text{In case that we keep using safety advisor afterwards}} \underbrace{\tilde{T}(\tilde{x} | \omega_x(k), u_{uc}(\omega_x(k), k))}_{\text{If inputs from unverified controller is accepted}} \right) \geq 1 - \rho$$

$$P(x(t) \in \bar{X} \setminus \{\Phi\} | x(t-1) = \omega_u(t-1), u(t-1) = \omega_u(t-1))$$

Keep idea: At every time instant, make sure whether  $\rho$  can be respected by **keep using safety advisor afterward.**



# Case Study – Temperature Control Problem

Considering a room is equipped with a heater, the dynamic of the system is

$$x(t+1) = (1 - \beta - \gamma u(t))x(t) + \gamma T_h u(t) + \beta T_e + \omega(t)$$

$x(t)$ : The temperature of the room at time  $t$

$u(t)$ : The input to the room at time  $t$

$\beta$  : Conduction factor between the external environment and the room

$\gamma$  : Conduction factor between the heater and the room

$T_e$  : Temperature of the external environment

$T_h$  : Temperature of the heater

$\omega$  : Gaussian white noise

Safety specification :  $x(t) \in [19, 21]$

## Problem setting

$u(t) \in [0, 0.6]$

$\beta = 0.022$

$\gamma = 0.05$

$T_e = -1^\circ\text{C}$

$T_h = 50^\circ\text{C}$

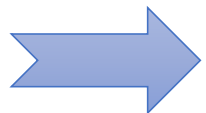
$\omega$  : mean is 0 and variance is 0.04

Sampling time period : 9 min

$\delta_x : 1.0 \times 10^{-3}$

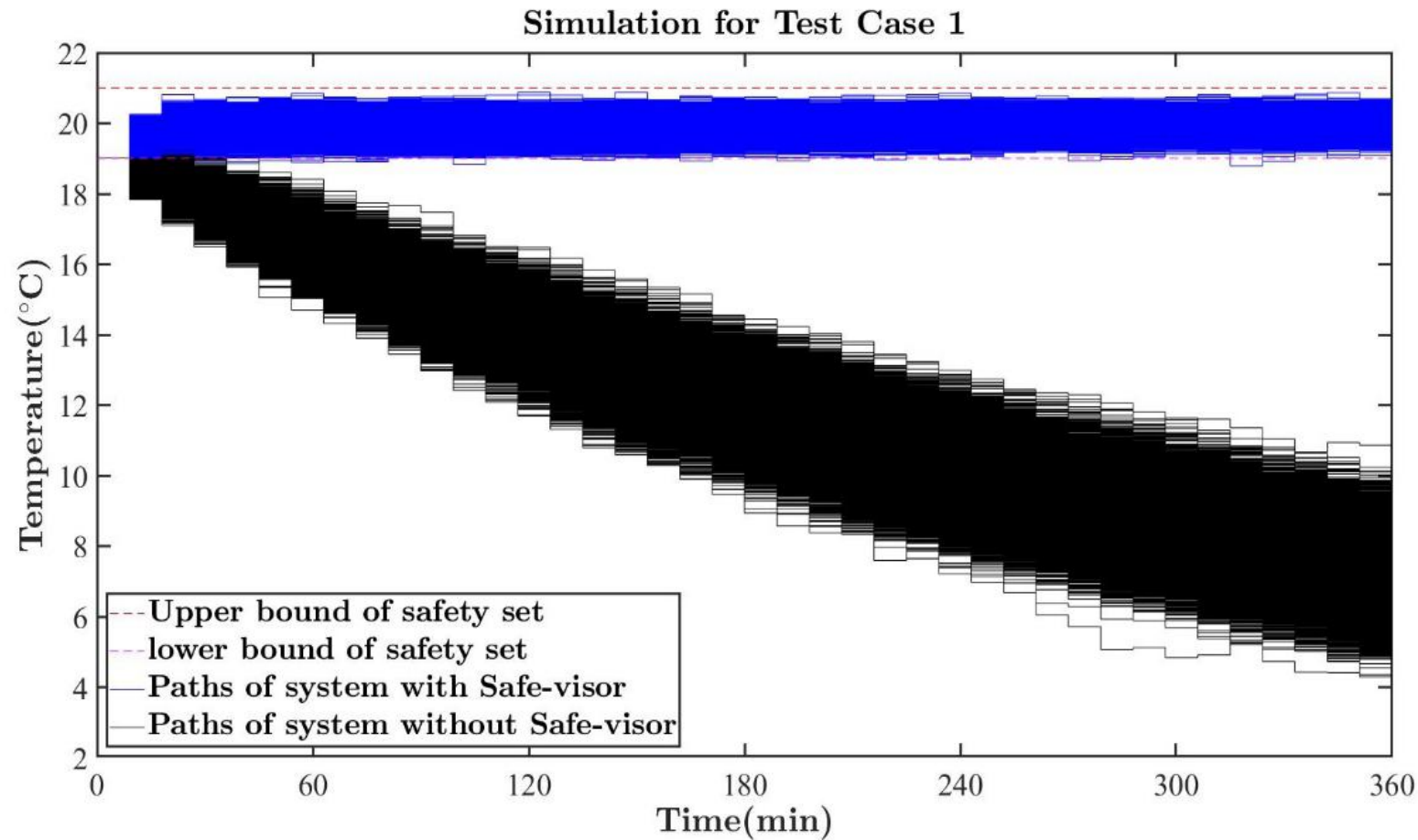
$\delta_u : 2.4 \times 10^{-2}$

Safety guarantee : 99%



Time horizon for the safety advisor:  $[0, 40]$  (6h)

# Case Study – Temperature Control Problem



Initial state	19.01°C
Unverified controller	u is 0 all the time
Percentage of paths in safety set (with Safe-visor)	99.02%
Average acceptance rate of unverified controller	19.12%
Percentage of paths in safety set (without Safe-visor)	0%
Percentage of paths in safety set (purely with Safety Advisor)	99.18%
Average execution time for History-based Supervisor	33.42 $\mu$ s

Number of simulation :  $1.0 \times 10^6$

Safety specification :  $x(t) \in [19, 21]$

## Case Study – Traffic Control Problem

Considering a road traffic control containing a cell with 2 entries and 1 exit, the dynamic of the system is

$$x(t + 1) = \left(1 - \frac{\tau v}{l} - q\right) x(t) + e_1 u(t) + \sigma(t) + e_2$$

$x(t)$ : The density of traffic at time  $t$

$u(t)$ : The input to the room at time  $t$  (1 means the green light is on while 0 means the red light is on)

$\tau$  : Sampling time interval of the system

$v$  : Flow speed of the vehicle on the road

$l$  : Temperature of the external environment

$q$  : Percentage of cars which leave the cell through the exit\*

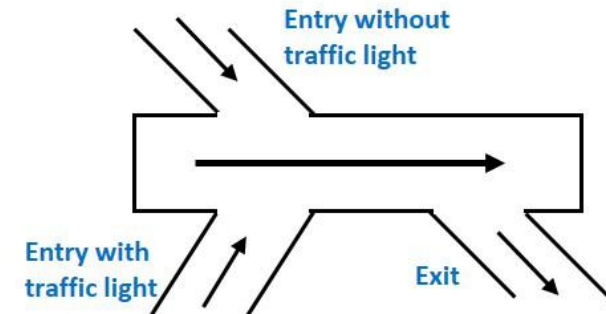
$e_1$  : Number of cars pass the entry controlled by the traffic light\*

$e_2$  : Number of cars pass the entry without the traffic light\*

$\sigma$  : Gaussian white noise

Safety guarantee : 99.95%

Time horizon for the safety advisor: [0,8186] (13.64h)



Safety specification  $x(t) \leq 20$

### Problem setting

$\tau$  : 6s

$v$  : 25 m/s

$l$  : 500 m

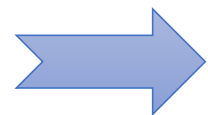
$q$  : 10%

$e_1$  : 6

$e_2$  : 3

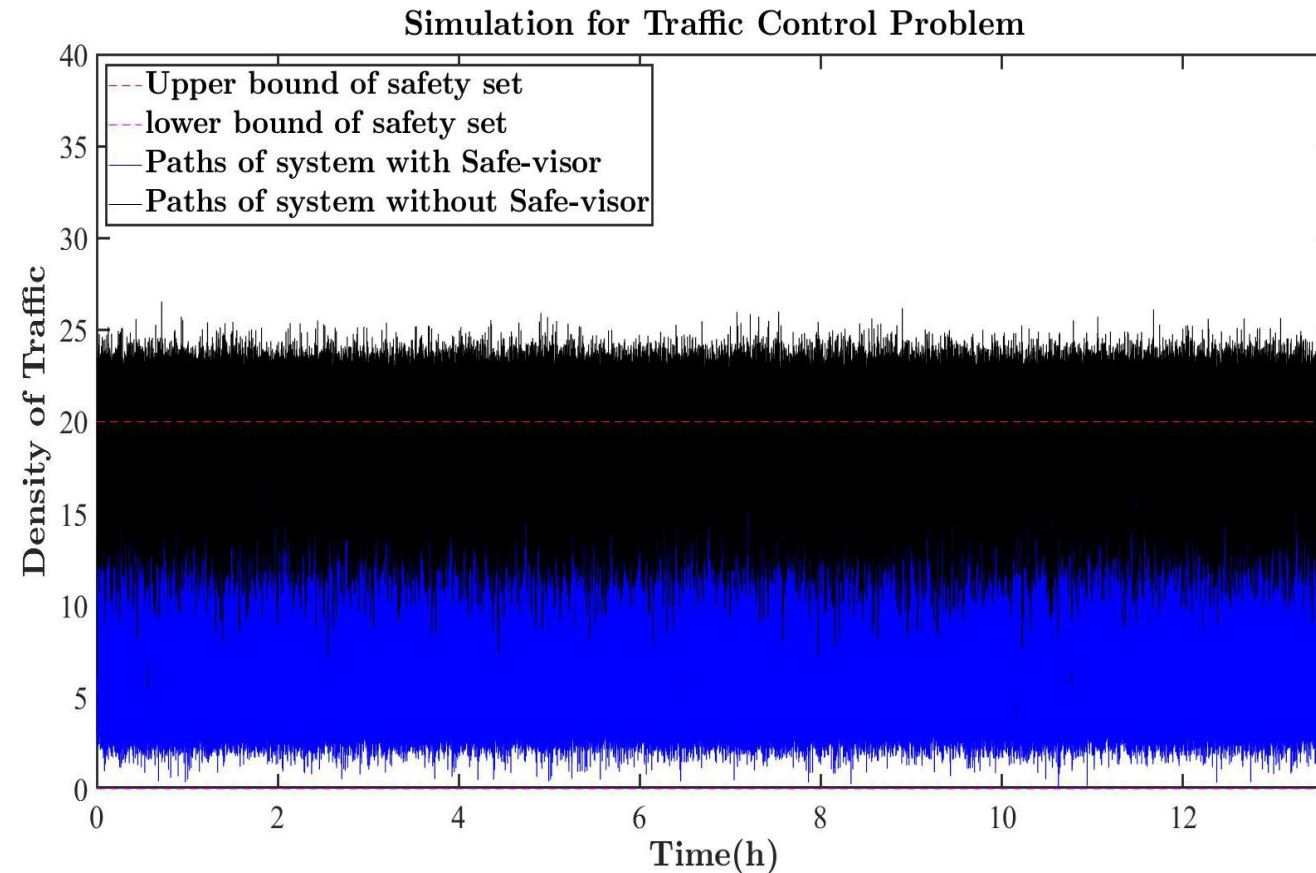
$\sigma$  : mean is 0 and variance is 2

$\delta_x$  :  $1.0 \times 10^{-3}$



\* in one sampling interval

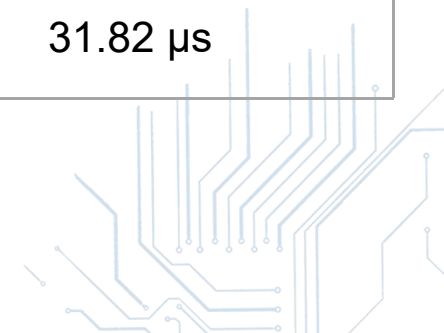
# Case Study – Traffic Control Problem



Number of simulation :  $1.0 \times 10^6$

Safety specification  $x(t) \leq 20$

Initial state	9
Unverified controller	$u(t) = 1$ when $t$ is odd number, otherwise 0
Percentage of paths in safety set (with Safe-visor)	99.958%
Average acceptance rate of unverified controller	8.5114%
Percentage of paths in safety set (without Safe-visor)	0%
Percentage of paths in safety set (purely with Safety Advisor)	99.989%
Average execution time for History-based Supervisor	31.82 $\mu$ s



## Perspective

Extending our method to

- 1) systems modeled by partially observable Markov decision process.
- 2) more general safety specification, e.g. co-safe linear temporal logic.



# Acknowledgements

## Funding:

- H2020 ERC Starting Grant AutoCPS (grant agreement No 804639)
- German Research Foundation (DFG) (grants ZA 873/1-1 and ZA 873/4-1).
- German Federal Ministry of Education and Research & Alexander von Humboldt Foundation:  
Alexander von Humboldt Professorship



## Q & A

