SpaceEx:
Scalable Verification of Hybrid Systems

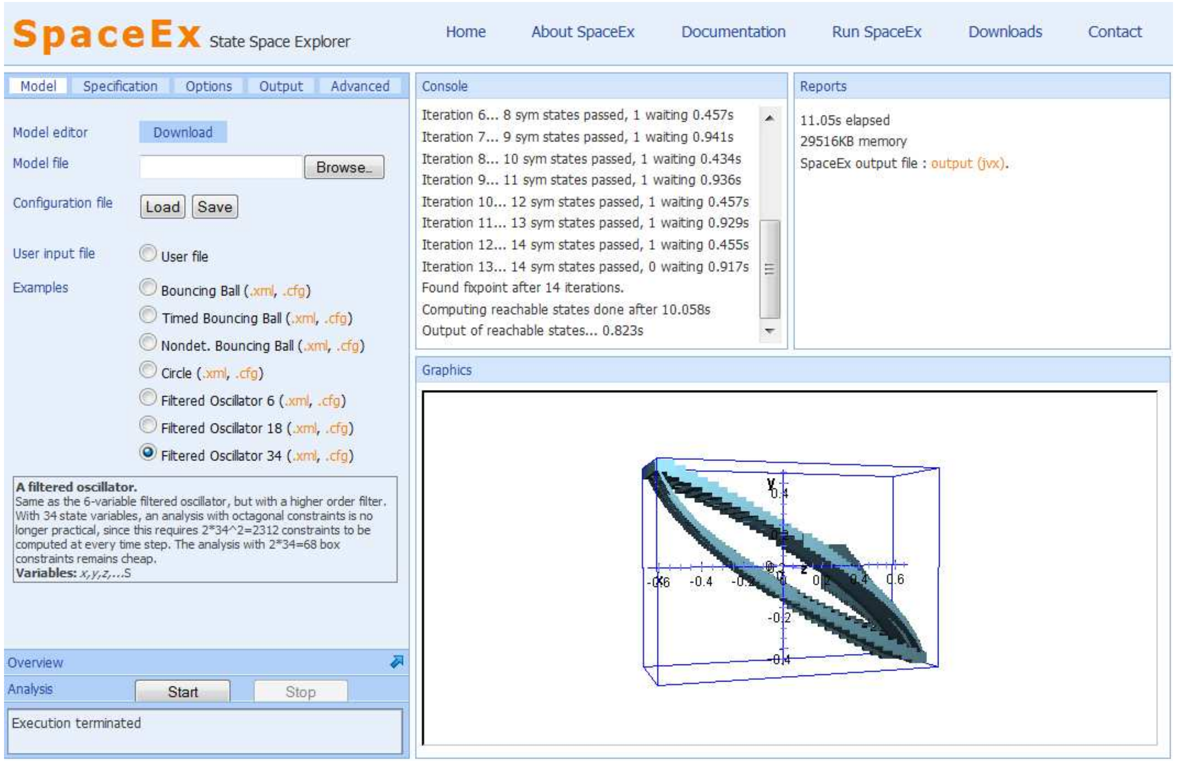
Colas Le Guernic

April 29, 2011

joint work with:
Goran Frehse, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel,
Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler.

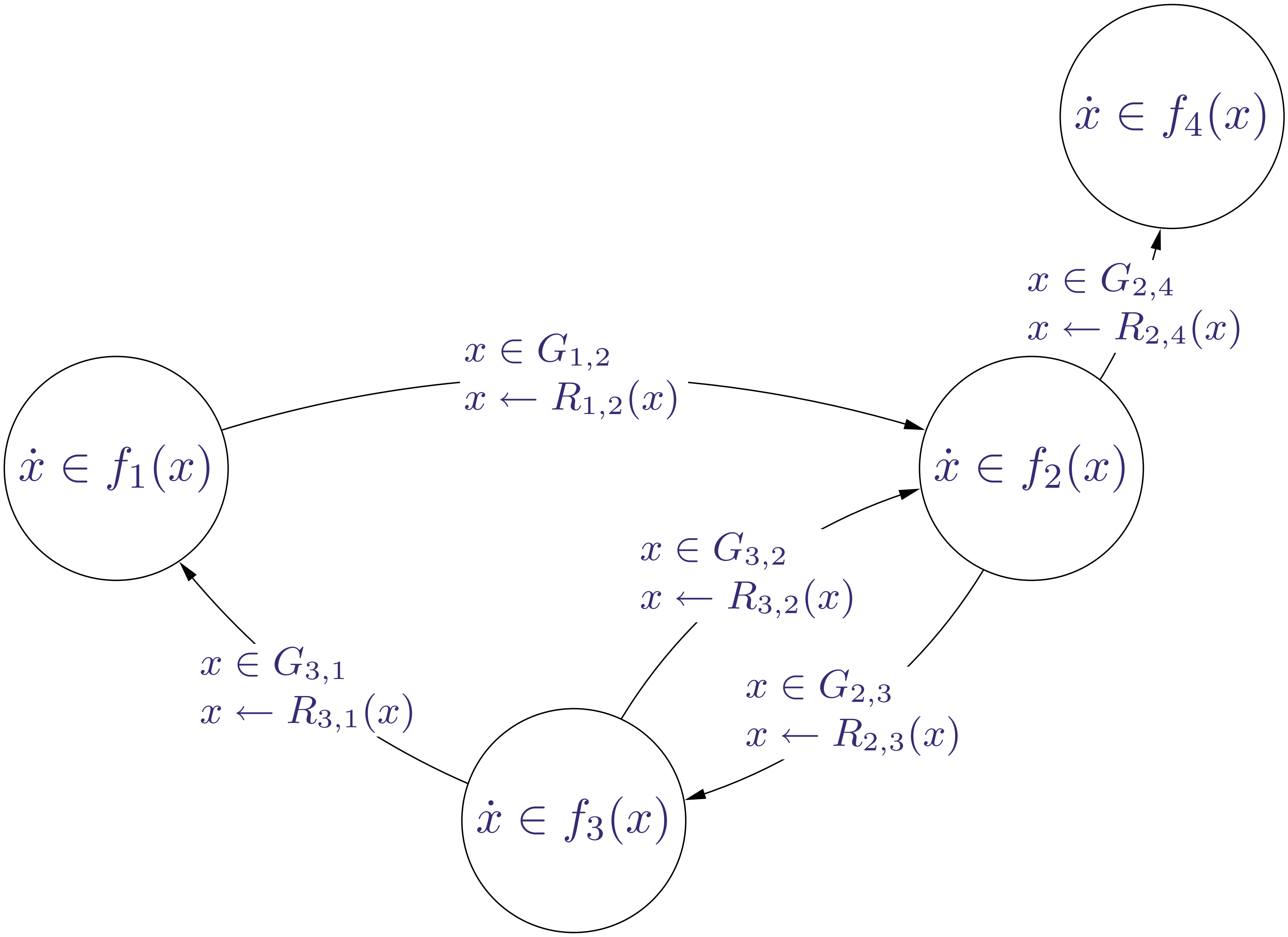
SpaceEx

SpaceEx is a software platform for reachability and safety verification of hybrid systems developed at Verimag.

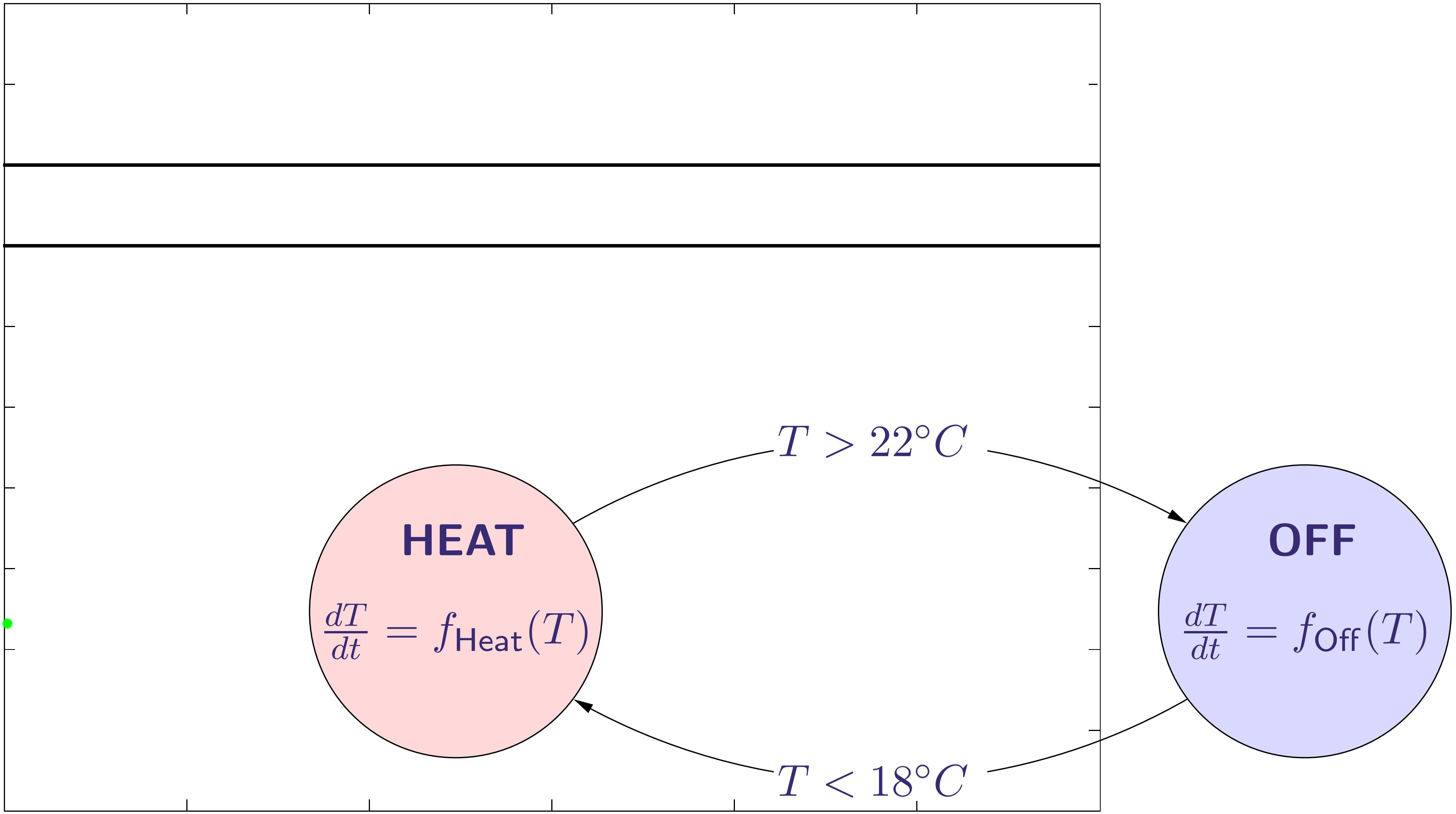


http://spaceex.imag.fr/

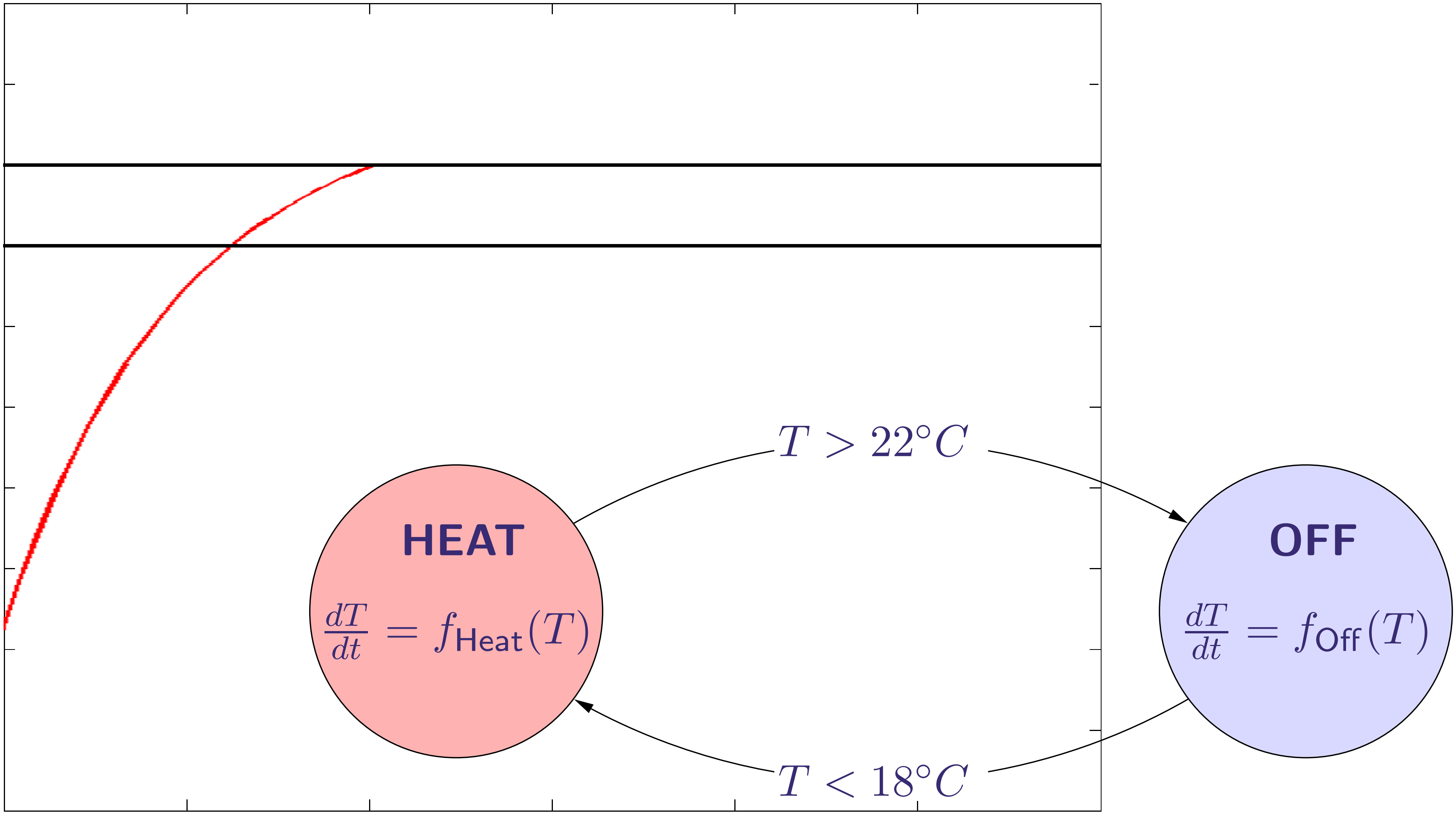
Hybrid Systems



Example

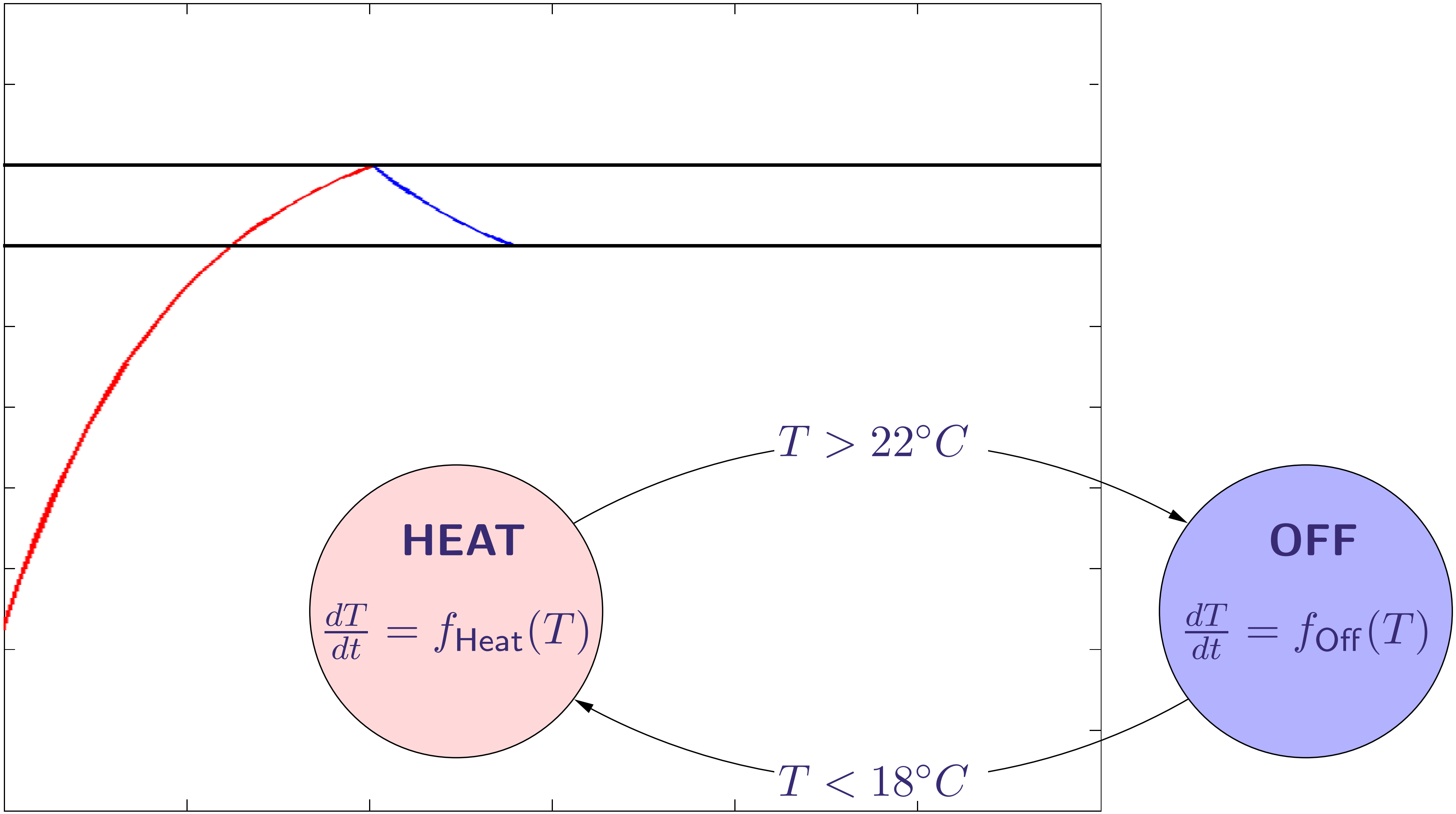

HEAT
$\frac{dT}{dt} = f_{\mathsf{Heat}}(T)$
OFF
$\frac{dT}{dt} = f_{\mathsf{Off}}(T)$
$T > 22°C$
$T < 18°C$

Example



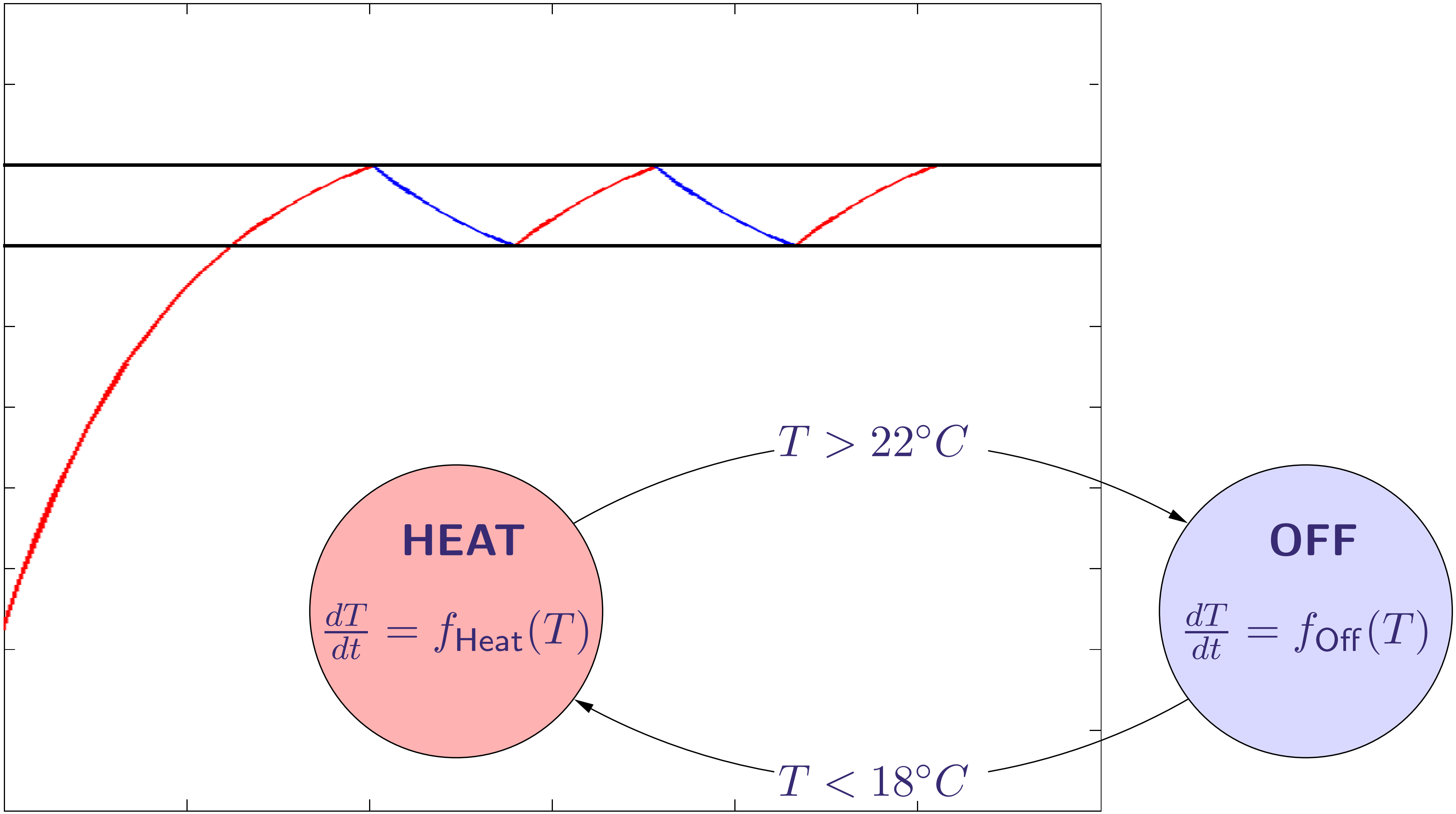
$T > 22°C$
HEAT
$\frac{dT}{dt} = f_{\mathsf{Heat}}(T)$
OFF
$\frac{dT}{dt} = f_{\mathsf{Off}}(T)$
$T < 18°C$

Example

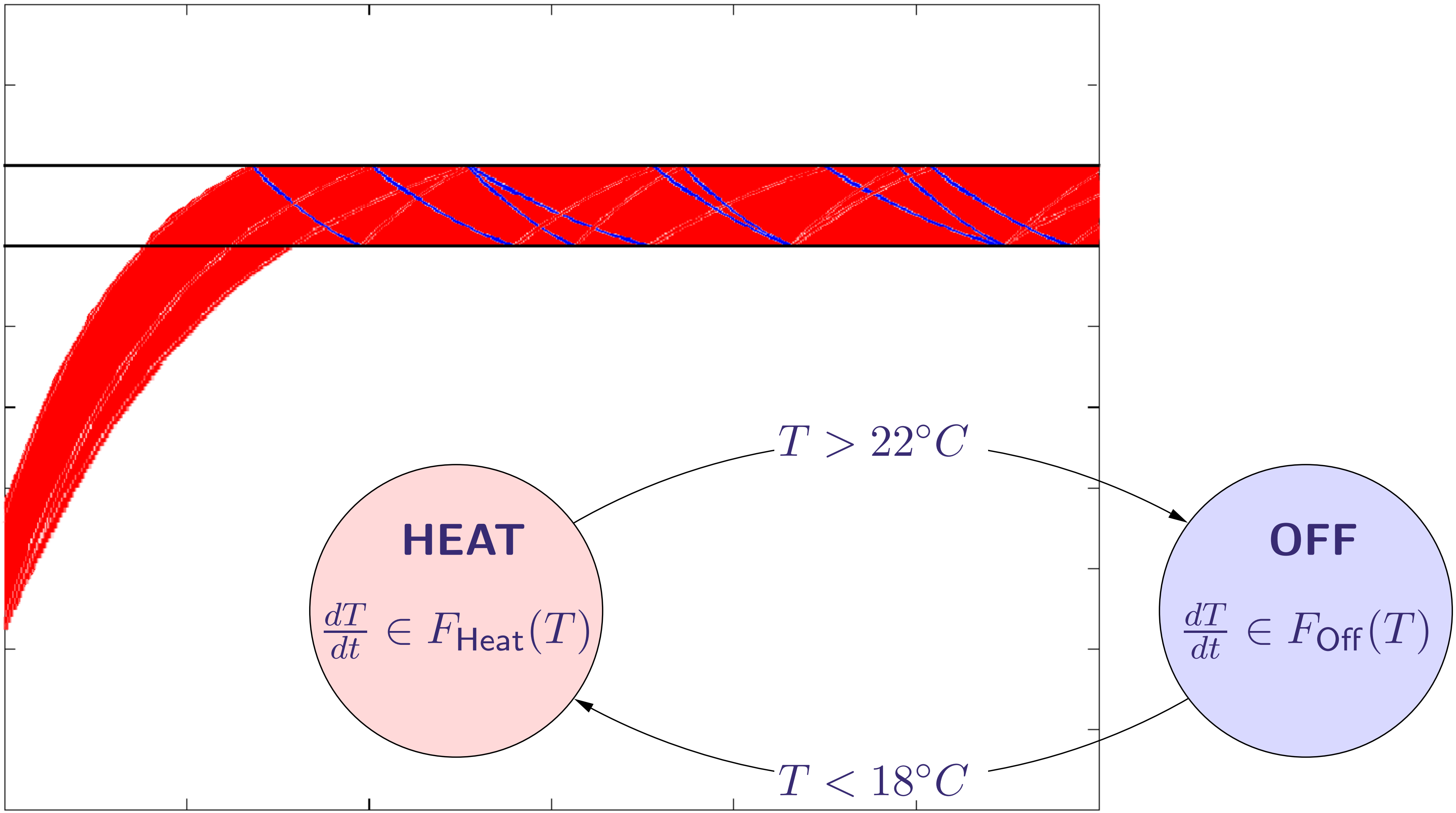


$T > 22°C$
HEAT
$\frac{dT}{dt} = f_{\mathsf{Heat}}(T)$
OFF
$\frac{dT}{dt} = f_{\mathsf{Off}}(T)$
$T < 18°C$

Example


$T > 22°C$
HEAT
$\frac{dT}{dt} = f_{\text{Heat}}(T)$
OFF
$\frac{dT}{dt} = f_{\text{Off}}(T)$
$T < 18°C$

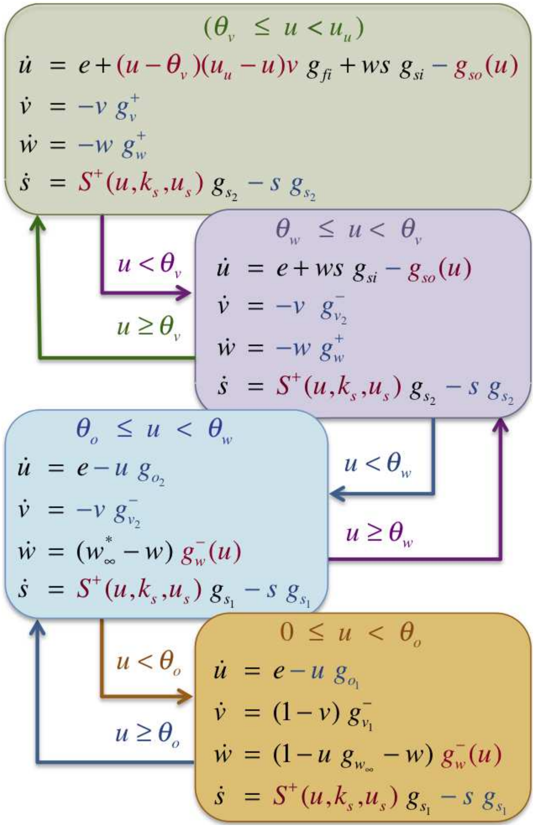
Example



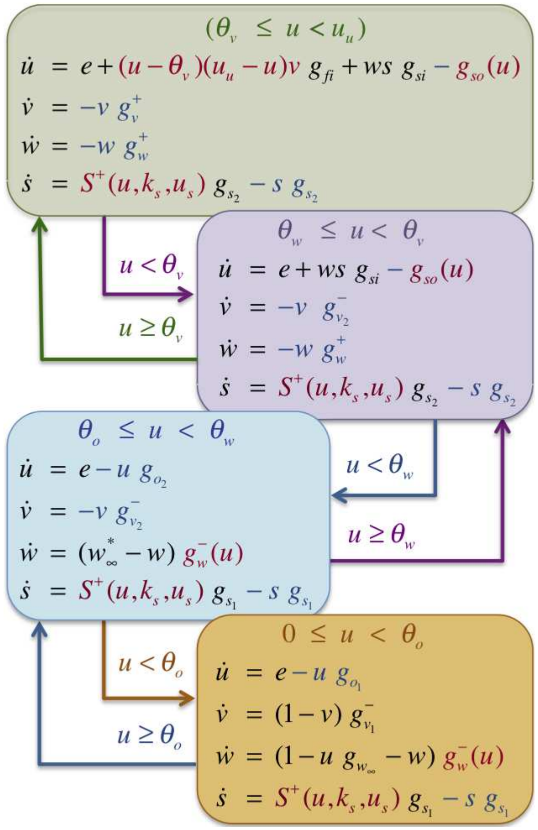
$T > 22°C$
HEAT
$\frac{dT}{dt} \in F_{\text{Heat}}(T)$
OFF
$\frac{dT}{dt} \in F_{\text{Off}}(T)$
$T < 18°C$

NYU

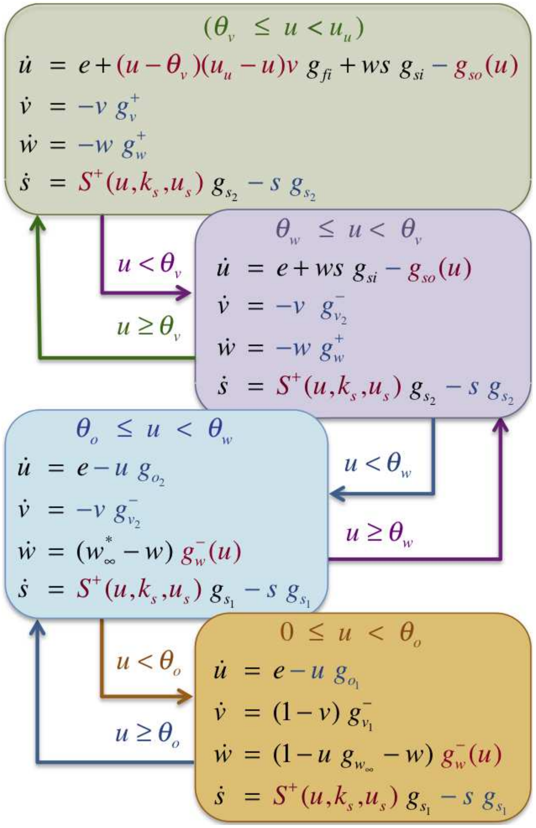
CMACS



$(\theta_v \leq u < u_u)$
$\dot{u} = e + (u - \theta_v)(u_u - u)v\, g_{fi} + ws\, g_{si} - g_{so}(u)$
$\dot{v} = -v\, g_v^+$
$\dot{w} = -w\, g_w^+$
$\dot{s} = S^+(u, k_s, u_s)\, g_{s_2} - s\, g_{s_2}$
$u < \theta_v$
$u \geq \theta_v$
$\theta_w \leq u < \theta_v$
$\dot{u} = e + ws\, g_{si} - g_{so}(u)$
$\dot{v} = -v\, g_{v_2}^-$
$\dot{w} = -w\, g_w^+$
$\dot{s} = S^+(u, k_s, u_s)\, g_{s_2} - s\, g_{s_2}$
$\theta_o \leq u < \theta_w$
$\dot{u} = e - u\, g_{o_2}$
$\dot{v} = -v\, g_{v_2}^-$
$\dot{w} = (w_\infty^* - w)\, g_w^-(u)$
$\dot{s} = S^+(u, k_s, u_s)\, g_{s_1} - s\, g_{s_1}$
$u < \theta_w$
$u \geq \theta_w$
$0 \leq u < \theta_o$
$\dot{u} = e - u\, g_{o_1}$
$\dot{v} = (1 - v)\, g_{v_1}^-$
$\dot{w} = (1 - u\, g_{w_\infty} - w)\, g_w^-(u)$
$\dot{s} = S^+(u, k_s, u_s)\, g_{s_1} - s\, g_{s_1}$
$u < \theta_o$
$u \geq \theta_o$



Parameter Estimation with RoVerGeNe.



Parameter Estimation with RoVerGeNe.

Based on abstractions by discrete automata.

Parameter Estimation in SpaceEx

SpaceEx, Reachability for:

LHA	HA with linear dynamics
$\dot{x} \in \mathcal{P}$	$\dot{x} \in \{Ax + u \mid u \in \mathcal{U}\}$

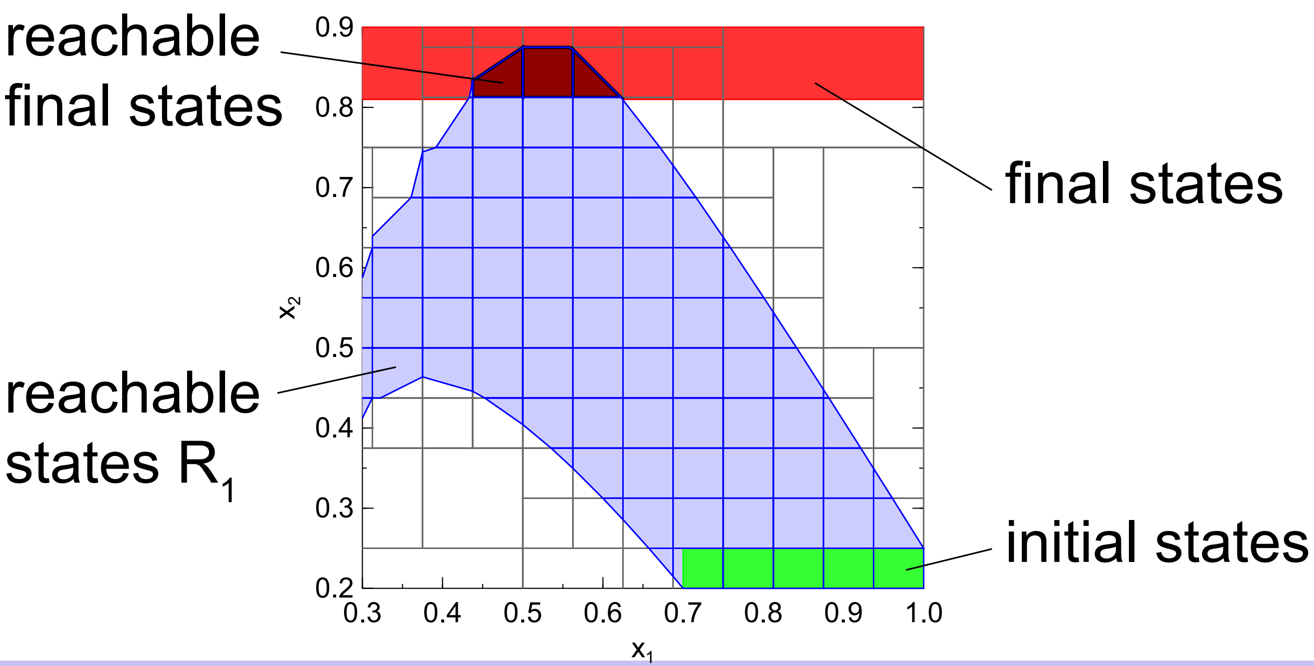
No parameter estimation.

Parameter Estimation in SpaceEx

SpaceEx, Reachability for:

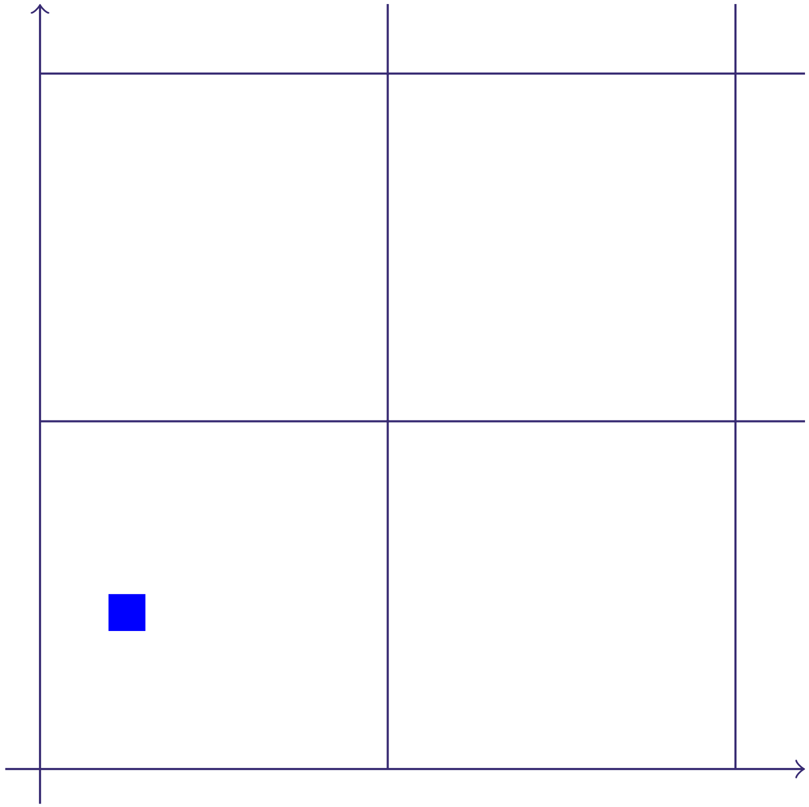
LHA	HA with linear dynamics
$\dot{x} \in \mathcal{P}$	$\dot{x} \in \{Ax + u \mid u \in \mathcal{U}\}$

Parameters as variables with 0 derivative.



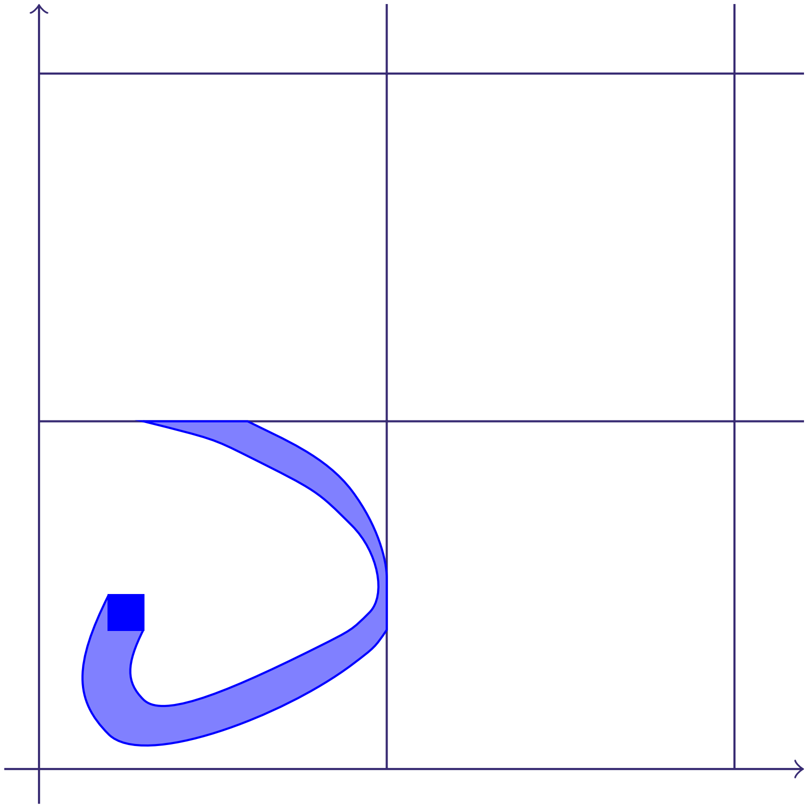
Reachability Analysis

- Continuous Dynamics: $\dot{x} \in Ax \oplus \mathcal{U}$ and $x(t) \in \mathcal{I}$
- Hyperplanar guards
- Affine Reset Maps



- Continuous Dynamics: $\dot{x} \in Ax \oplus \mathcal{U}$ and $x(t) \in \mathcal{I}$
- Hyperplanar guards
- Affine Reset Maps

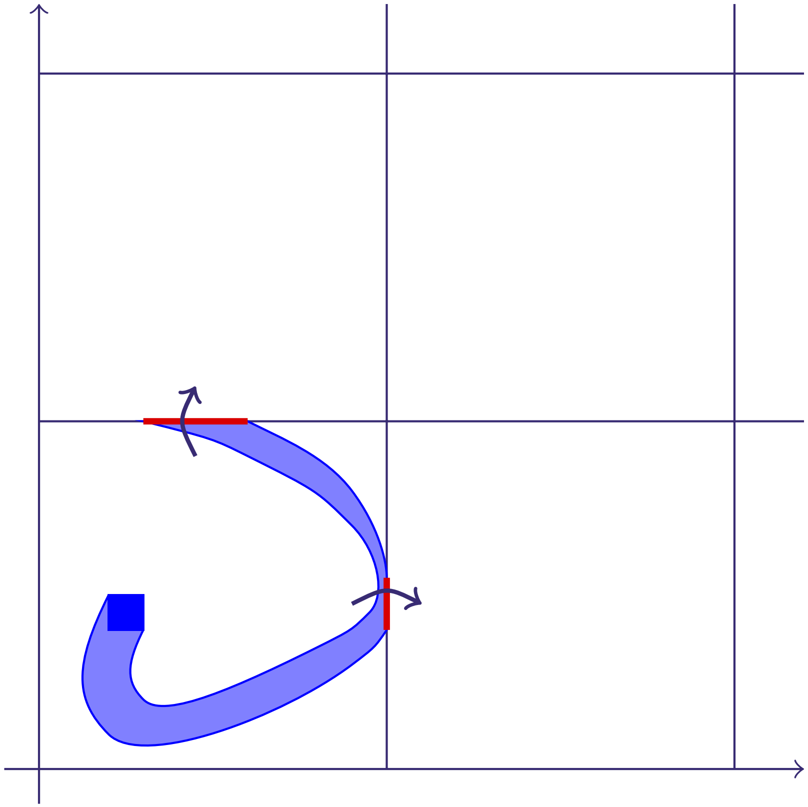
Post$_c$: Continuous evolution



Reachability Analysis

- Continuous Dynamics: $\dot{x} \in Ax \oplus \mathcal{U}$ and $x(t) \in \mathcal{I}$
- Hyperplanar guards
- Affine Reset Maps

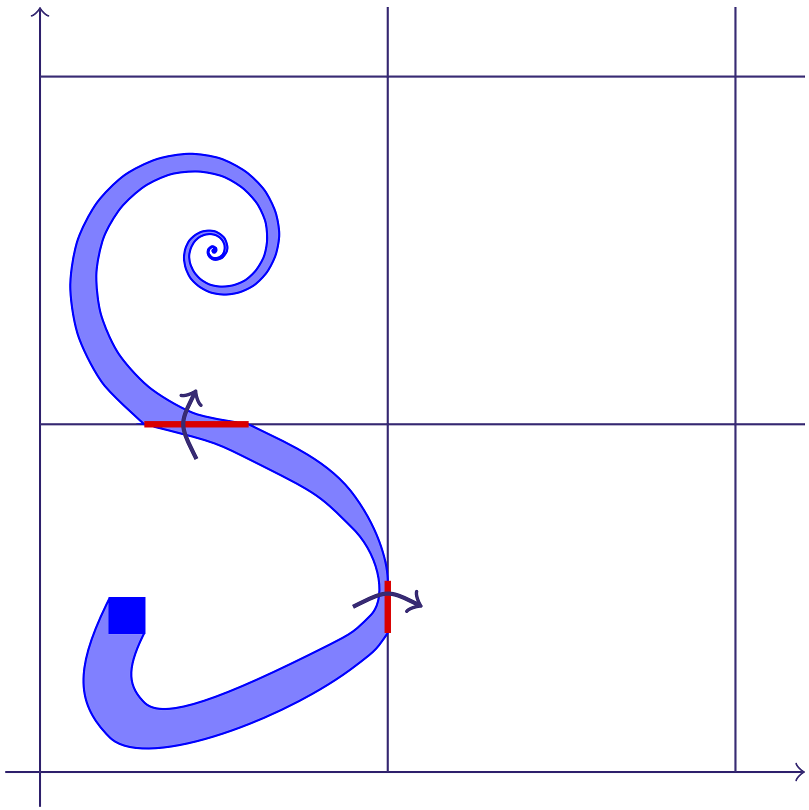
Post_c : Continuous evolution
Post_d : Discrete transition



Reachability Analysis

- Continuous Dynamics: $\dot{x} \in Ax \oplus \mathcal{U}$ and $x(t) \in \mathcal{I}$
- Hyperplanar guards
- Affine Reset Maps

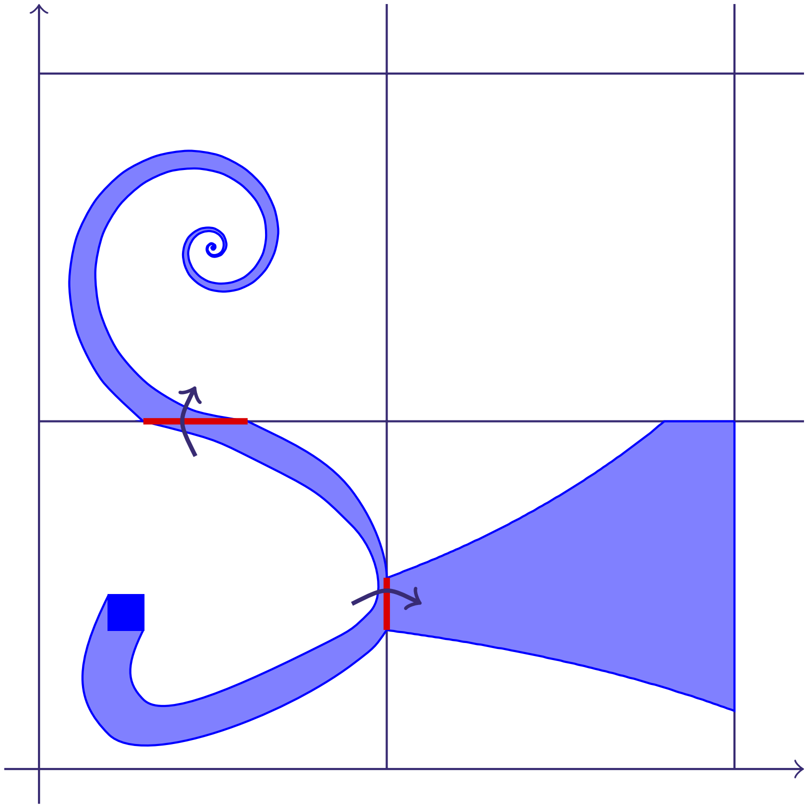
Post_c : Continuous evolution
Post_d : Discrete transition



Reachability Analysis

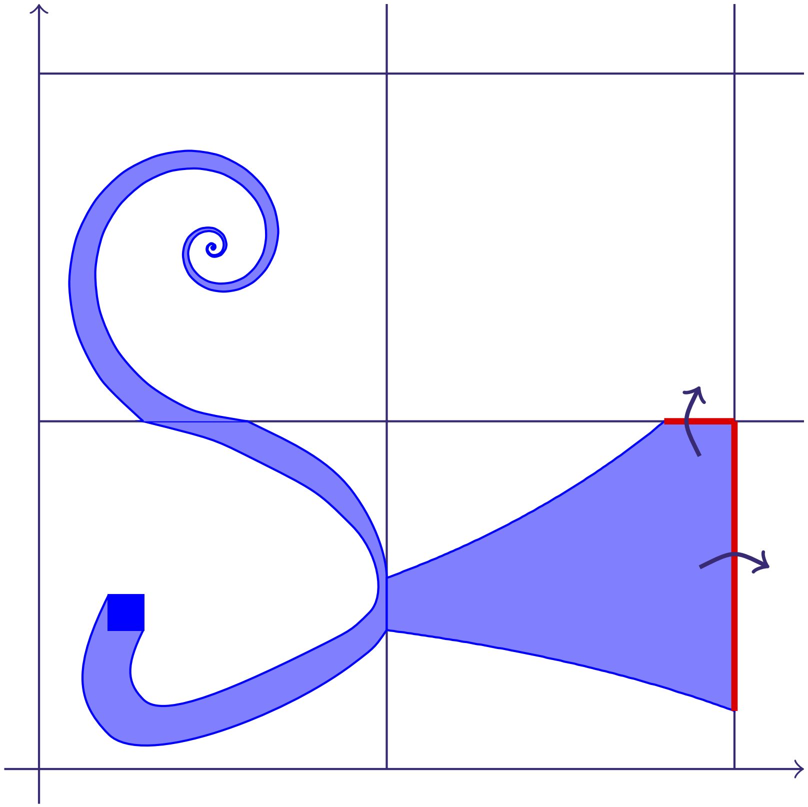
- Continuous Dynamics: $\dot{x} \in Ax \oplus \mathcal{U}$ and $x(t) \in \mathcal{I}$
- Hyperplanar guards
- Affine Reset Maps

Post_c : Continuous evolution
Post_d : Discrete transition



- Continuous Dynamics: $\dot{x} \in Ax \oplus \mathcal{U}$ and $x(t) \in \mathcal{I}$
- Hyperplanar guards
- Affine Reset Maps

Post_c : Continuous evolution
Post_d : Discrete transition



Reachability for Linear Time Invariant systems

Describe all $x(t) \in \mathbb{R}^d$ for any t in $[0; T]$ such that :

$$\dot{x}(t) = Ax(t) + u(t) \quad \text{with } x(0) \in \mathcal{X}_0 \text{ and } u(t) \in \mathcal{U}$$

Reachability for Linear Time Invariant systems

Describe all $x(t) \in \mathbb{R}^d$ for any t in $[0;T]$ such that :

$$\dot{x}(t) = Ax(t) + u(t) \quad \text{with } x(0) \in \mathcal{X}_0 \text{ and } u(t) \in \mathcal{U}$$

Analytical solution for a given input function u:

$$x(t) = e^{tA}x(0) + \int_0^t e^{(t-s)A}u(s)\,ds$$

Reachability for Linear Time Invariant systems

Describe all $x(t) \in \mathbb{R}^d$ for any t in $[0; T]$ such that :

$$\dot{x}(t) = Ax(t) + u(t) \quad \text{with } x(0) \in \mathcal{X}_0 \text{ and } u(t) \in \mathcal{U}$$

Analytical solution for a given input function u:

$$x(t) = e^{tA}x(0) + \int_0^t e^{(t-s)A}u(s)\,ds$$

Temporal discretization:

$$\mathrm{Reach}_{[t_k, t_k+\delta_k]}(\mathcal{X}_0) = e^{At_k}\mathrm{Reach}_{[0,\delta_k]}(\mathcal{X}_0) \oplus \mathrm{Reach}_{[t_k,t_k]}(\{0\})$$

Reachability for Linear Time Invariant systems

Describe all $x(t) \in \mathbb{R}^d$ for any t in $[0;T]$ such that :

$$\dot{x}(t) = Ax(t) + u(t) \quad \text{with } x(0) \in \mathcal{X}_0 \text{ and } u(t) \in \mathcal{U}$$

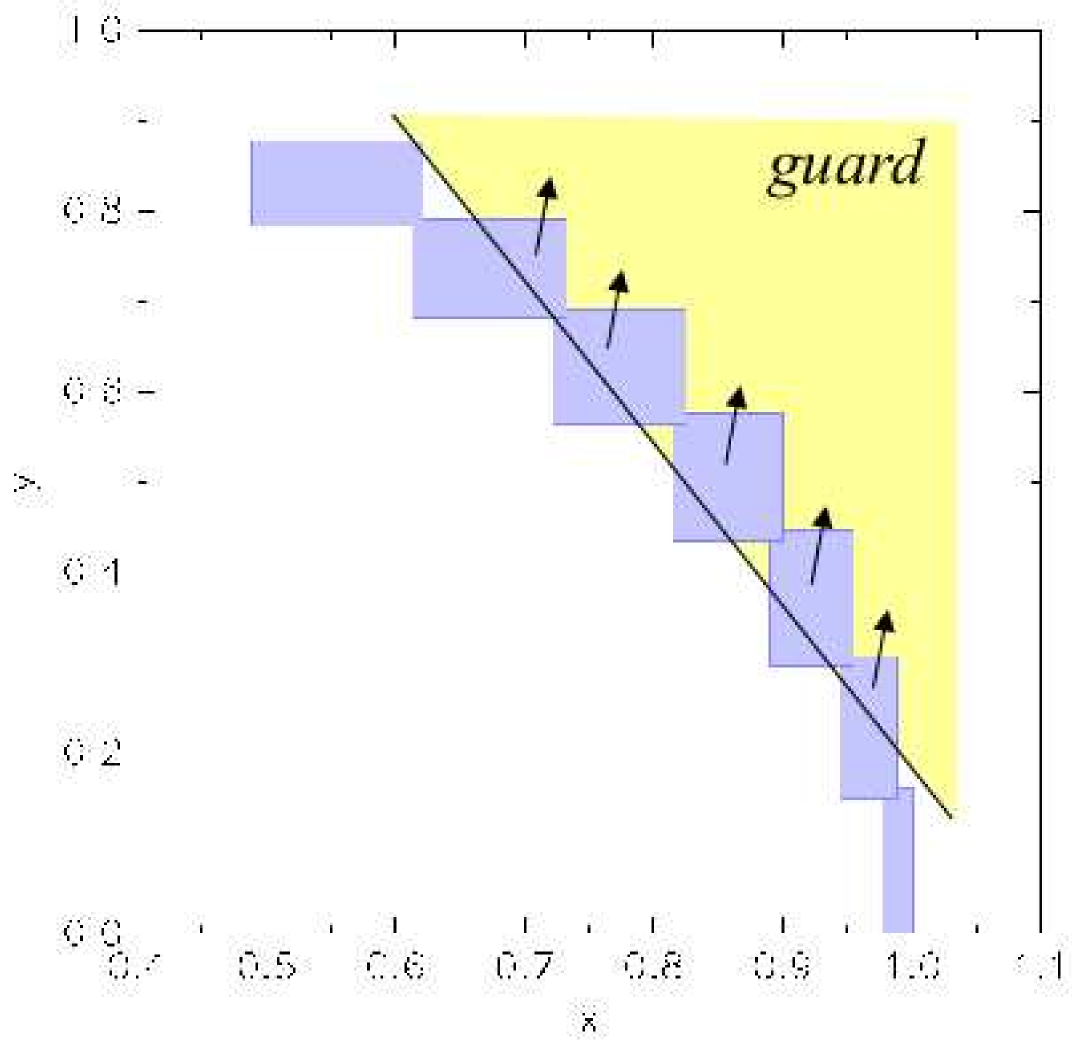
Analytical solution for a given input function u:

$$x(t) = e^{tA}x(0) + \int_0^t e^{(t-s)A}u(s)\,ds$$

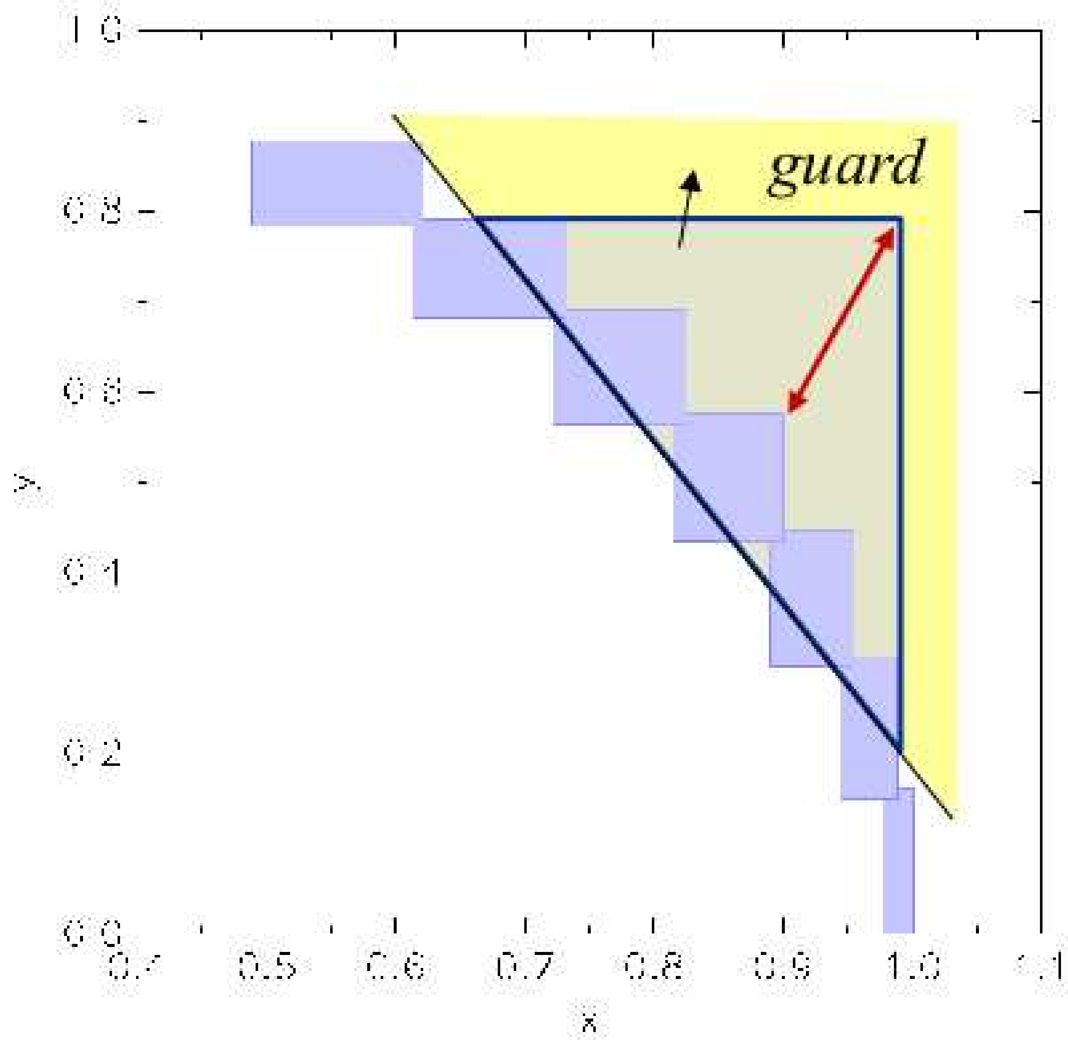
Temporal discretization:

$$\begin{aligned} \Psi_{k+1} &= \Psi_k \oplus e^{At_k}\Psi_{\delta_k}(\mathcal{U}) \\ \Omega_k &= e^{At_k}\Omega_{[0,\delta_k]}(\mathcal{X}_0, \mathcal{U}) \oplus \Psi_k \end{aligned}$$

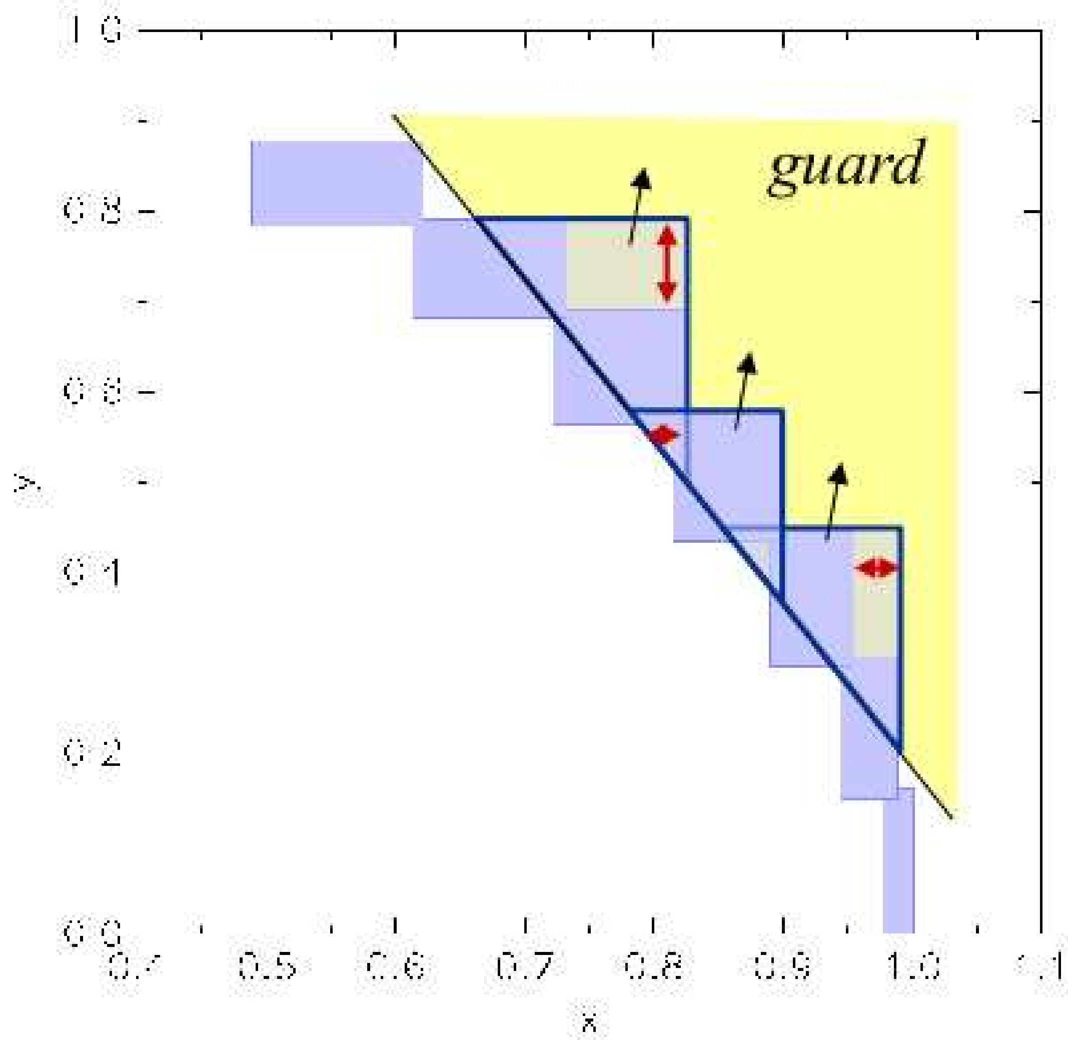
Guards



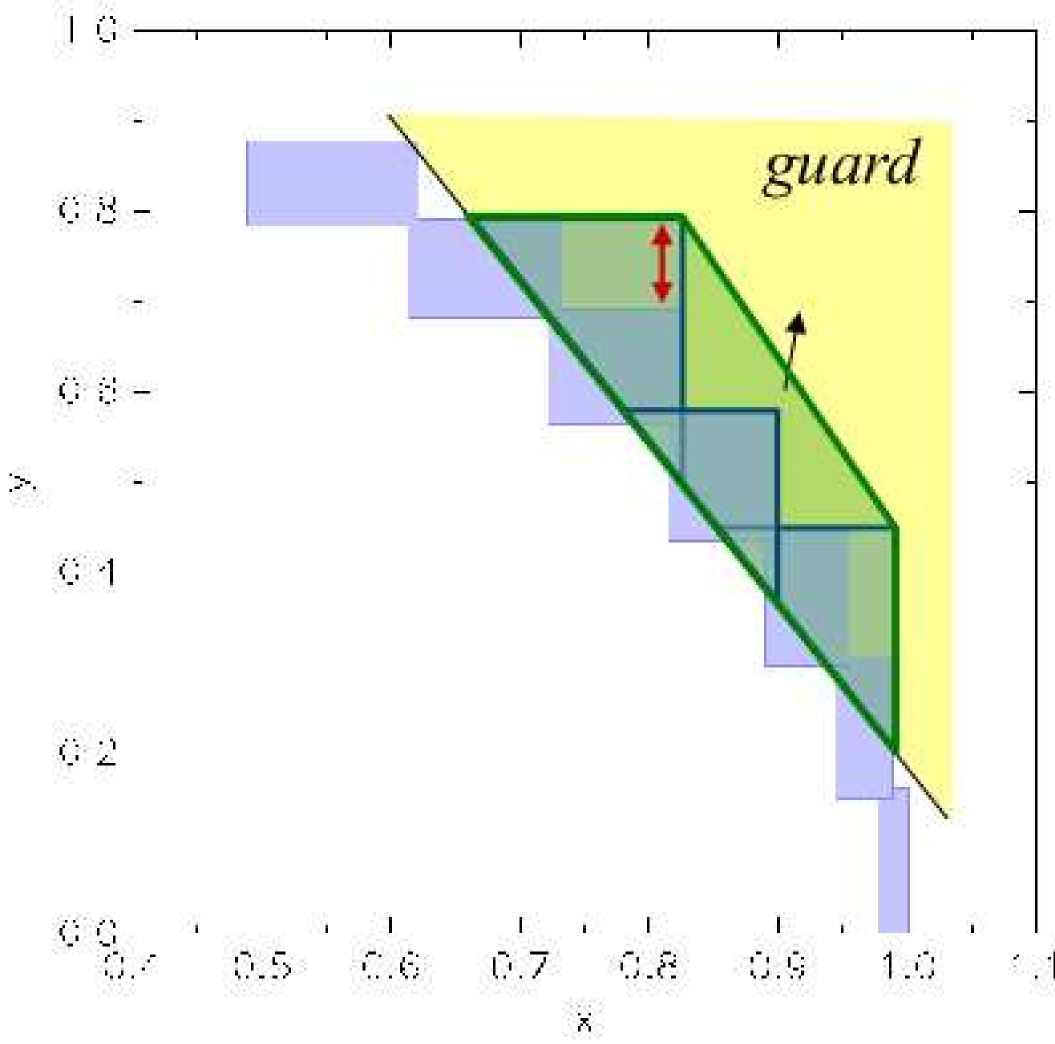
Guards



Guards



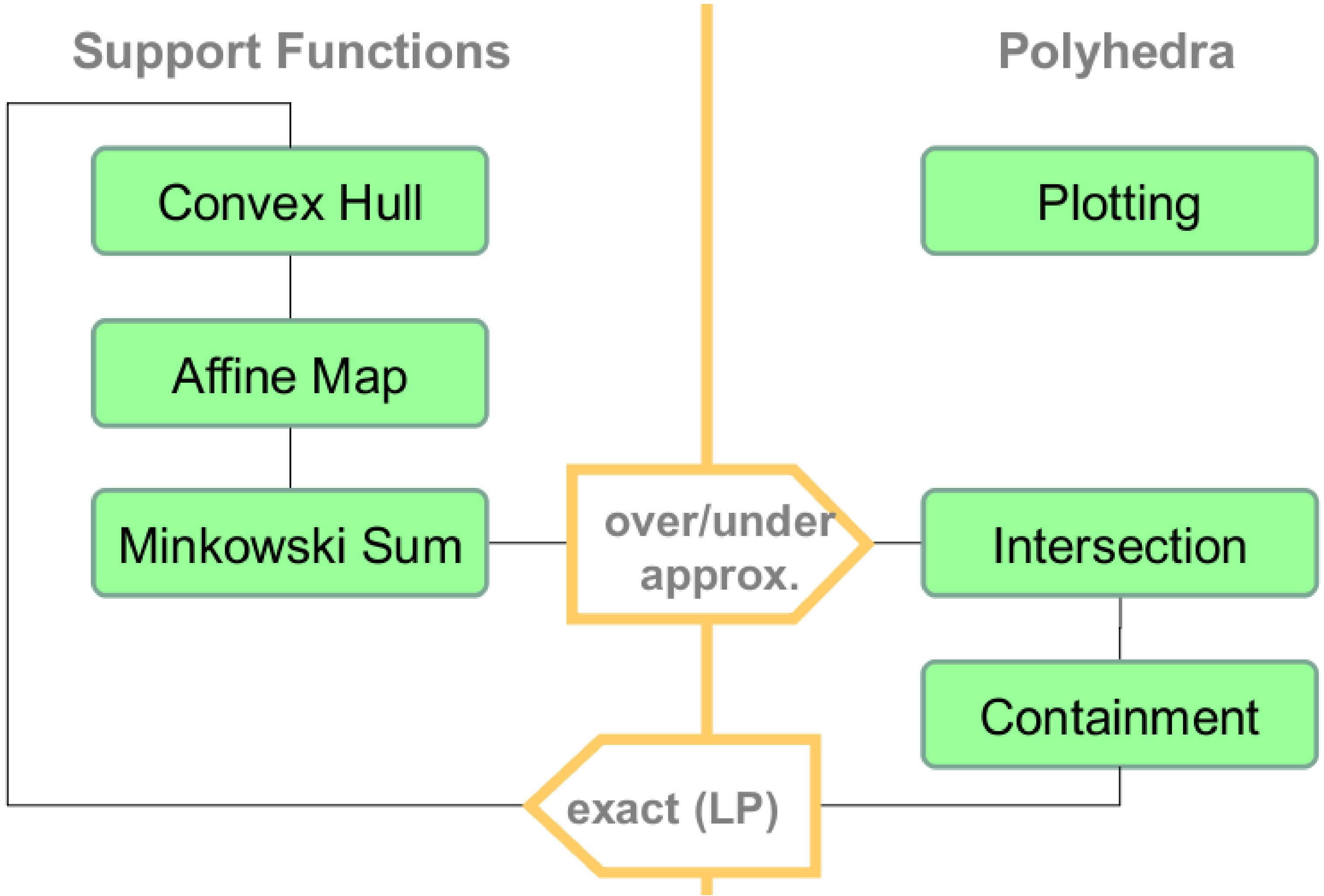
Guards



Representing Sets

Operators	Polyhedra		Zonotopes	Support Functions
	Constraints	Vertices		
Affine transform	-	++	++	++
Minkowski sum	--	-	++	++
Intersection	++	--	--	-
Containment	++	--	?	--
Convex hull	--	++	--	++

Scalable Computation by Transformation



Support Function

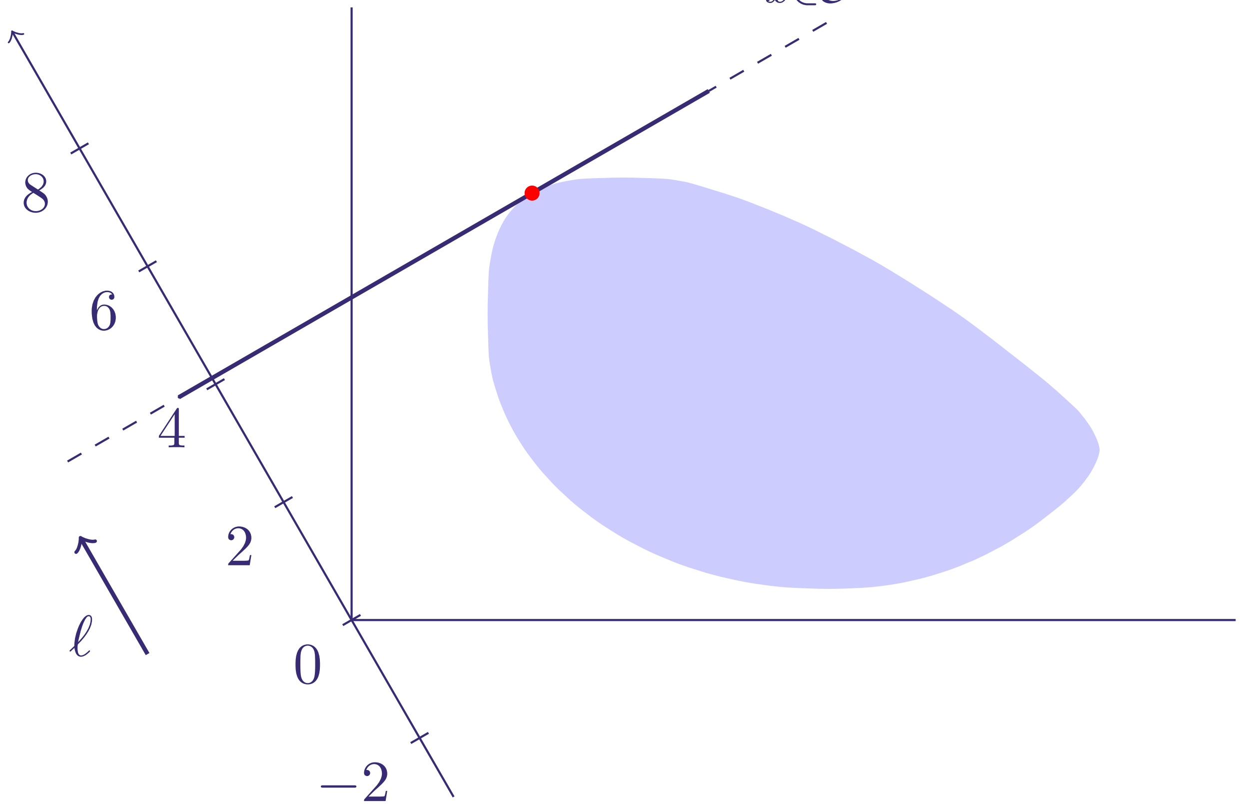
Definition

The support function of a compact convex set $\mathcal{S} \subseteq \mathbb{R}^d$, denoted $\rho_{\mathcal{S}}$, is defined as:

$$\begin{array}{rccc} \rho_{\mathcal{S}} : & \mathbb{R}^d & \longrightarrow & \mathbb{R} \\ & \ell & \longmapsto & \max_{x \in \mathcal{S}} \ell \cdot x \end{array}$$

The support function of a compact convex set $\mathcal{S} \subseteq \mathbb{R}^d$, denoted $\rho_\mathcal{S}$, is defined as:

$$\begin{aligned} \rho_\mathcal{S} : \quad \mathbb{R}^d &\rightarrow \mathbb{R} \\ \ell &\mapsto \max_{x \in \mathcal{S}} \ell \cdot x \end{aligned}$$



Examples

- support function of the unit cube $\mathcal{B}_\infty$:

$$\rho_{\mathcal{B}_\infty}(\ell) = \|\ell\|_1 = \sum_{i=0}^{d-1} |\ell_i|$$

- support function of a ball $\mathcal{S}$ of center c and radius r:

$$\rho_{\mathcal{S}}(\ell) = c \cdot \ell + r\|\ell\|_2$$

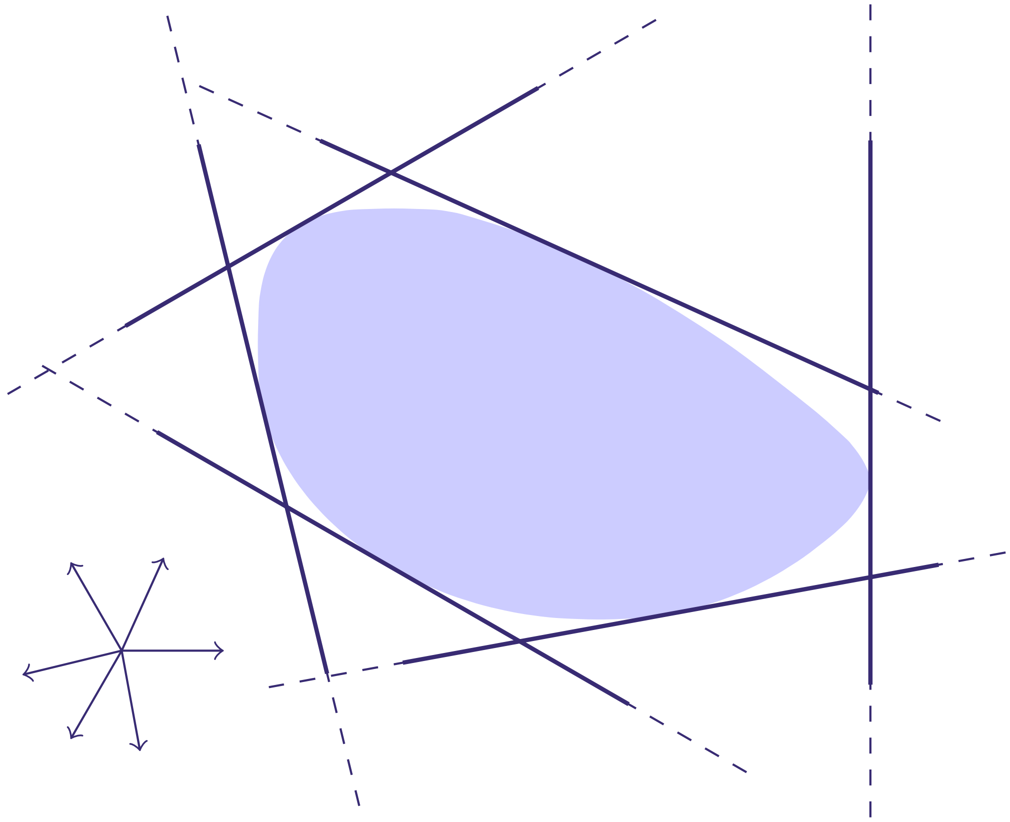
- support function of a polytope $\mathcal{P} = \{x : Ax \leq b\}$: any LP algorithm solving:

$$\begin{cases} \max x \cdot \ell \\ Ax \leq b \end{cases}$$

Representing sets

If $\mathcal{S}$ is convex, then:

$$\mathcal{S} = \bigcap_{\ell \in \mathbb{R}^d} \{x : x \cdot \ell \leq \rho_{\mathcal{S}}(\ell)\}$$



Properties

- Linear transformation:

$$\rho_{A\mathcal{S}}(\ell) = \rho_{\mathcal{S}}(A^\top \ell)$$

- Minkowski sum:

$$\mathcal{X} \oplus \mathcal{Y} = \{x + y : x \in \mathcal{X} \text{ and } y \in \mathcal{Y}\}$$

$$\rho_{\mathcal{X} \oplus \mathcal{Y}}(\ell) = \rho_{\mathcal{X}}(\ell) + \rho_{\mathcal{Y}}(\ell)$$

- Convex union:

$$\rho_{CH(\mathcal{X} \cup \mathcal{Y})}(\ell) = \max(\rho_{\mathcal{X}}(\ell), \rho_{\mathcal{Y}}(\ell))$$

$\rho_{\Omega_k}(\ell)$

$$\Psi_{k+1} = \Psi_k \oplus e^{At_k}\Psi_{\delta_k}(\mathcal{U})$$
$$\Omega_k = e^{At_k}\Omega_{[0,\delta_k]}(\mathcal{X}_0,\mathcal{U}) \oplus \Psi_k$$

For any direction ℓ:

$$\psi_{k+1}(\ell) = \psi_k(\ell) + \rho_{\Psi_{\delta_k}}((e^{At_k})^\top \ell)$$
$$\rho_{\Omega_k}(\ell) = \rho_{\Omega_{[0,\delta_k]}}((e^{At_k})^\top \ell) \oplus \psi_k(\ell)$$

$\Omega_{[0,\delta_k]}(\mathcal{X}_0, \mathcal{U})$

Let $\lambda \in [0,1]$, and $\Omega_\lambda(\mathcal{X}_0, \mathcal{U}, \delta)$ be the convex set defined by :

$$\Omega_\lambda(\mathcal{X}_0, \mathcal{U}, \delta) = (1-\lambda)\mathcal{X}_0 \oplus \lambda e^{\delta A}\mathcal{X}_0 \oplus \lambda\delta\mathcal{U} \oplus \left(\lambda\mathcal{E}_\Omega^+(\mathcal{X}_0, \delta) \cap (1-\lambda)\mathcal{E}_\Omega^-(\mathcal{X}_0, \delta)\right) \oplus \lambda^2\mathcal{E}_\Psi(\mathcal{U}, \delta)$$

$$\text{where } \mathcal{E}_\Omega^+(\mathcal{X}_0, \delta) = \boxdot\left(\Phi_2(|A|, \delta) \boxdot \left(A^2\mathcal{X}_0\right)\right)$$
$$\text{and } \mathcal{E}_\Omega^-(\mathcal{X}_0, \delta) = \boxdot\left(\Phi_2(|A|, \delta) \boxdot \left(A^2 e^{\delta A}\mathcal{X}_0\right)\right)$$
$$\text{and } \mathcal{E}_\Psi(\mathcal{U}, \delta) = \boxdot\left(\Phi_2(|A|, \delta) \boxdot (A\mathcal{U})\right).$$

Then $\mathrm{Reach}_{\lambda\delta,\lambda\delta}(\mathcal{X}_0, \mathcal{U}) \subseteq \Omega_\lambda(\mathcal{X}_0, \mathcal{U}, \delta)$. If we define $\Omega_{[0,\delta]}(\mathcal{X}_0, \mathcal{U})$ as:

$$\Omega_{[0,\delta]}(\mathcal{X}_0, \mathcal{U}) = \mathrm{CH}\Big(\bigcup_{\lambda\in[0,1]} \Omega_\lambda(\mathcal{X}_0, \mathcal{U}, \delta)\Big),$$

then $\mathrm{Reach}_{0,\delta}(\mathcal{X}_0) \subseteq \Omega_{[0,\delta]}(\mathcal{X}_0, \mathcal{U})$.

$$\Omega_\lambda(\mathcal{X}_0,\mathcal{U},\delta) = (1-\lambda)\mathcal{X}_0 \oplus \lambda e^{\delta A}\mathcal{X}_0 \oplus \lambda\delta\mathcal{U} \oplus \left(\lambda\mathcal{E}^{+}_{\Omega}(\mathcal{X}_0,\delta) \cap (1-\lambda)\mathcal{E}^{-}_{\Omega}(\mathcal{X}_0,\delta)\right) \oplus \lambda^2\mathcal{E}_{\Psi}(\mathcal{U},\delta)$$

$$\Omega_{[0,\delta]}(\mathcal{X}_0,\mathcal{U}) = \mathrm{CH}\Big(\bigcup_{\lambda\in[0,1]} \Omega_\lambda(\mathcal{X}_0,\mathcal{U},\delta)\Big),$$

$\rho_{\Omega_{[0,\delta_k]}}(\ell)$

$$\Omega_\lambda(\mathcal{X}_0,\mathcal{U},\delta) = (1-\lambda)\mathcal{X}_0 \oplus \lambda e^{\delta A}\mathcal{X}_0 \oplus \lambda\delta\mathcal{U} \oplus \left(\lambda\mathcal{E}_\Omega^+(\mathcal{X}_0,\delta) \cap (1-\lambda)\mathcal{E}_\Omega^-(\mathcal{X}_0,\delta)\right) \oplus \lambda^2\mathcal{E}_\Psi(\mathcal{U},\delta)$$

$$\Omega_{[0,\delta]}(\mathcal{X}_0,\mathcal{U}) = \mathrm{CH}\Big(\bigcup_{\lambda\in[0,1]} \Omega_\lambda(\mathcal{X}_0,\mathcal{U},\delta)\Big),$$

$$\rho_{\Omega_{[0,\delta]}}(\ell) = \max_{\lambda\in[0,1]} \Big((1-\lambda)\rho_{\mathcal{X}_0}(\ell) + \lambda\rho_{\mathcal{X}_0}((e^{\delta A})^\top\ell) + \lambda\delta\rho_{\mathcal{U}}(\ell) + \rho_{\lambda\mathcal{E}_\Omega^+\cap(1-\lambda)\mathcal{E}_\Omega^-}(\ell) + \lambda^2\rho_{\mathcal{E}_\Psi}(\ell)\Big)$$

Time step adaptation with error bounds

- A user defined time step is arbitrary
- Time step guided by requested quality of approximation:

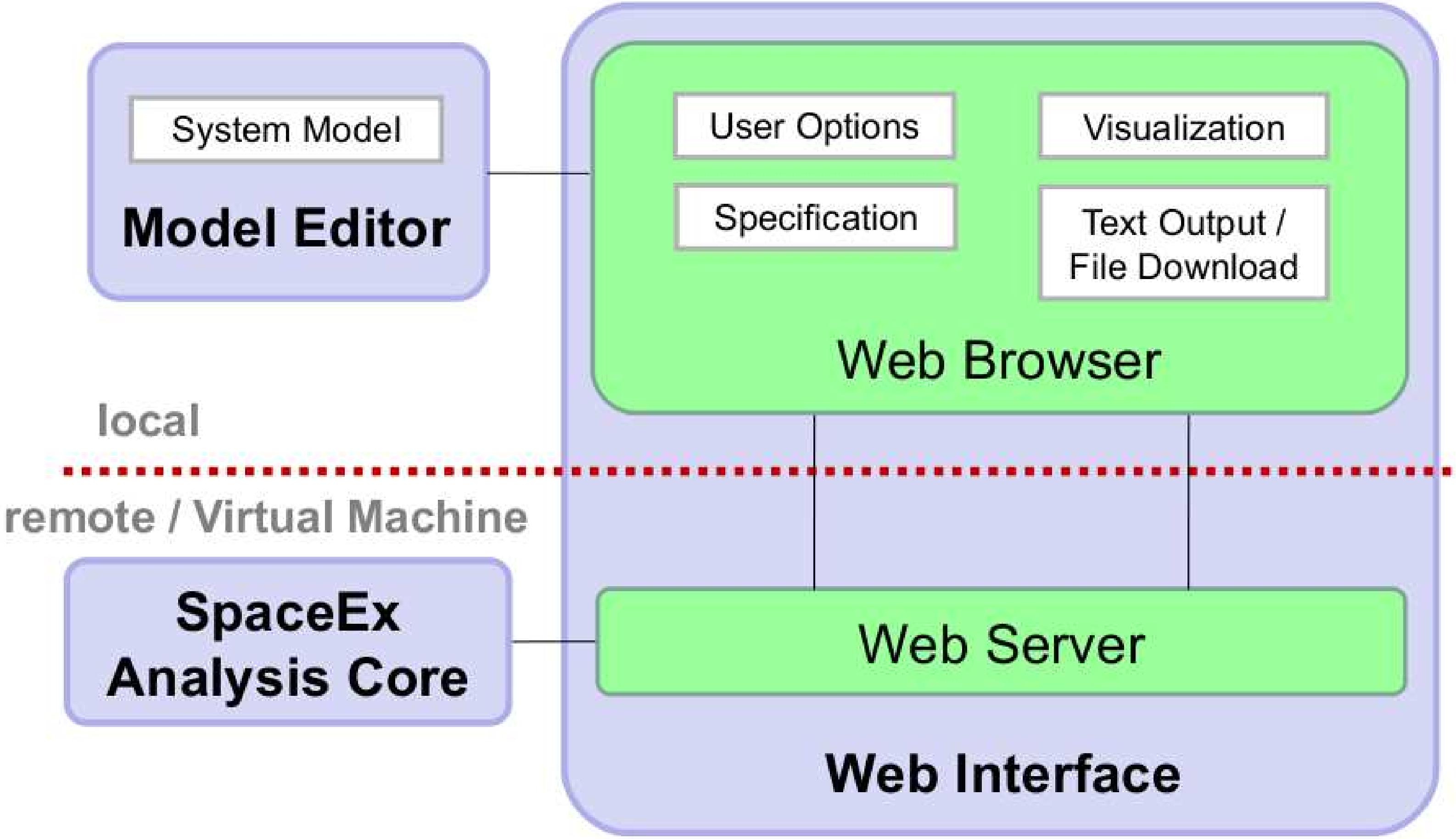
$$\epsilon_{\Omega_k}(\ell) = \rho(\ell, \Omega_k) - \rho\left(\ell, \mathrm{Reach}_{t_k,t_{k+1}}(\mathcal{X}_0)\right)$$

- linear accumulation of errors

$$\epsilon_{\Psi_k}(\ell) + \epsilon_{\Psi_{\delta_k^\Psi}(\mathcal{U})}(e^{At_k^\Psi}{}^\top \ell) \leq \frac{t_k^\Psi + \delta_k^\Psi}{T}\hat{\epsilon}_\Psi$$

- computed independently for each direction

SpaceEx Platform - Architecture



Thank you