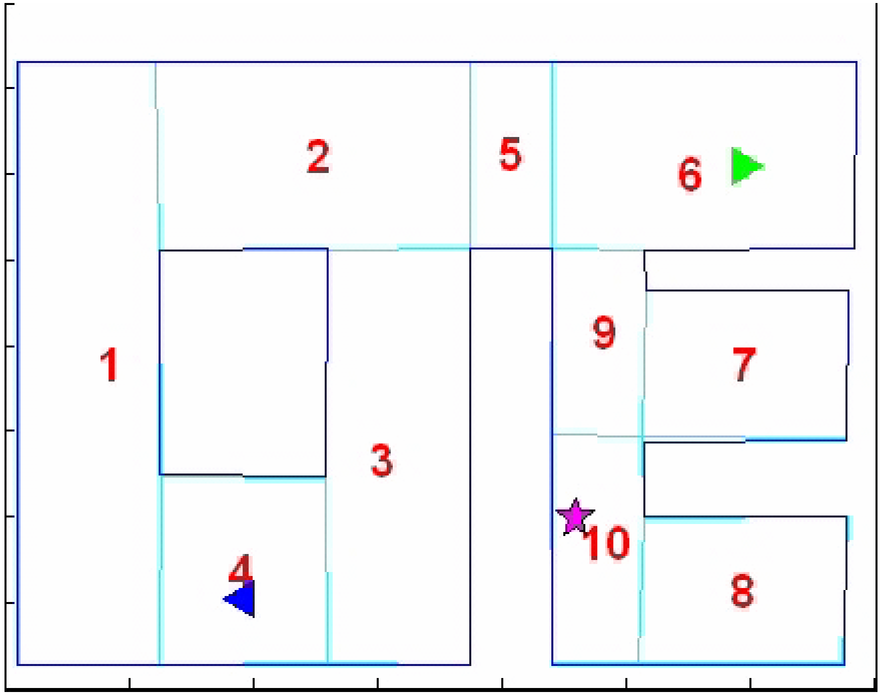
Hierarchical Synthesis of Hybrid Controllers from Temporal Logic Specifications

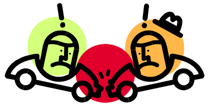
Georgios E. Fainekos, Antoine Girard and George J. Pappas

Search & Rescue Scenario

UAVs search for accidents.
If accident is found, then
they summon the ambulance.



✕ Accident



▶ UAV 1

◀ UAV 2

☆ Ambulance

$$\dot{z}(t) = v(t),\ \|v(t)\| \le v$$

$$y'(t) = z(t)$$

$$S'$$

$$z(t) \in \Re^2,\ z(0) = x_0,\ y'(t) \in \Re^2$$

Kress-Gazit, Fainekos and Pappas, *Where's waldo? sensor-based temporal logic motion planning*, ICRA 2007

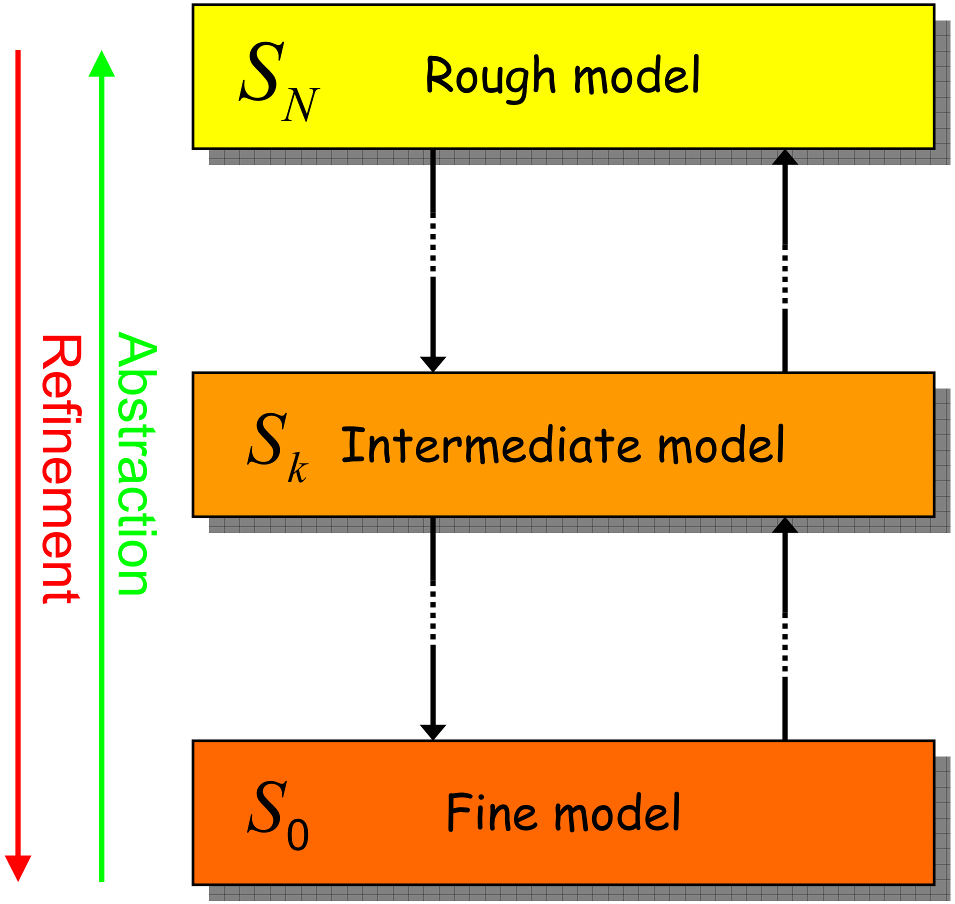
Search & Rescue Scenario

Two issues:

1) How can you construct robust trajectories wrt the temporal logic specifications?

2) How can we handle complex environments with complicated specifications beyond kinematics models?

Hierarchical Modeling of Control Systems



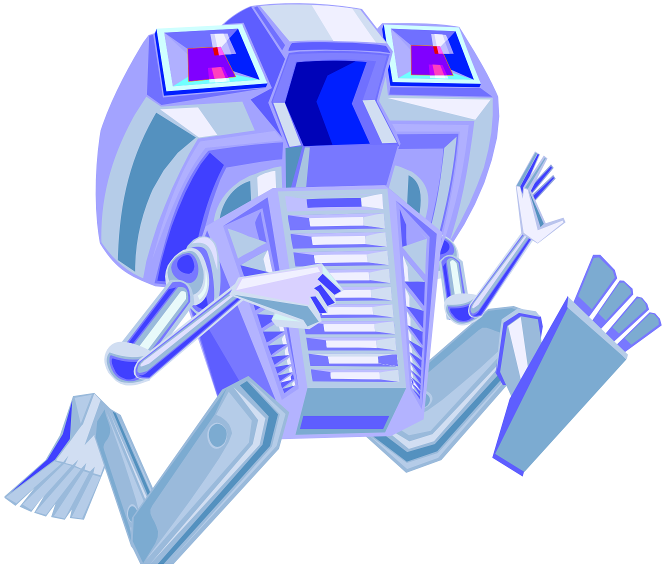
$$\dot{z}(t) = g(z(t), v(t)),\ v(t) \in V$$
$$y'(t) = k(z(t))$$
$$S' \xrightarrow{y'(t)}$$

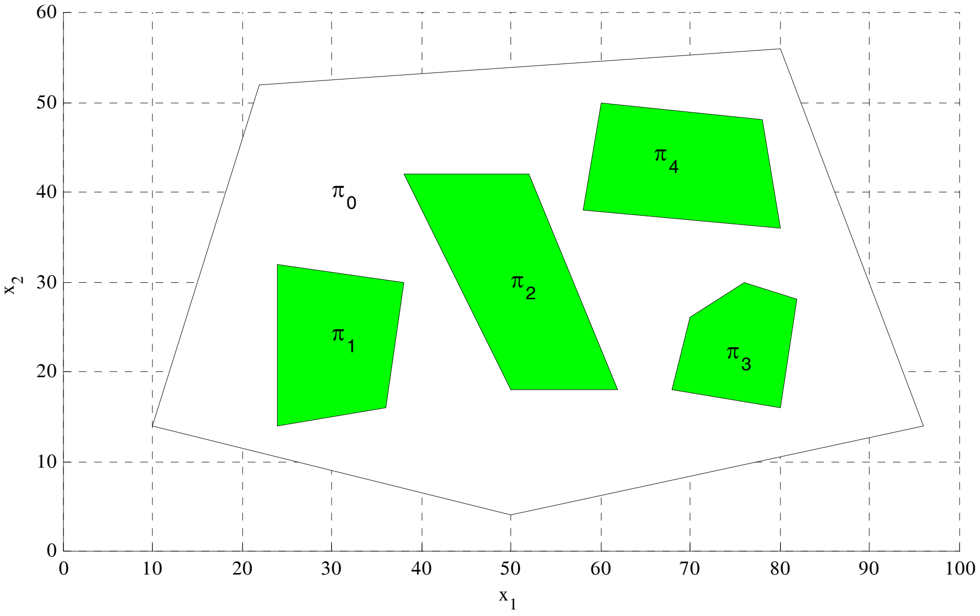
$$z(t) \in \Re^m,\ z(0) = z_0,\ y'(t) \in \Re^p$$

$$\dot{x}(t) = f(x(t), u(t)),\ u(t) \in U$$
$$y(t) = h(x(t))$$
$$S \xrightarrow{y(t)}$$

$$x(t) \in \Re^n,\ x(0) = x_0,\ y(t) \in \Re^p$$

Motivation - Motion Planning





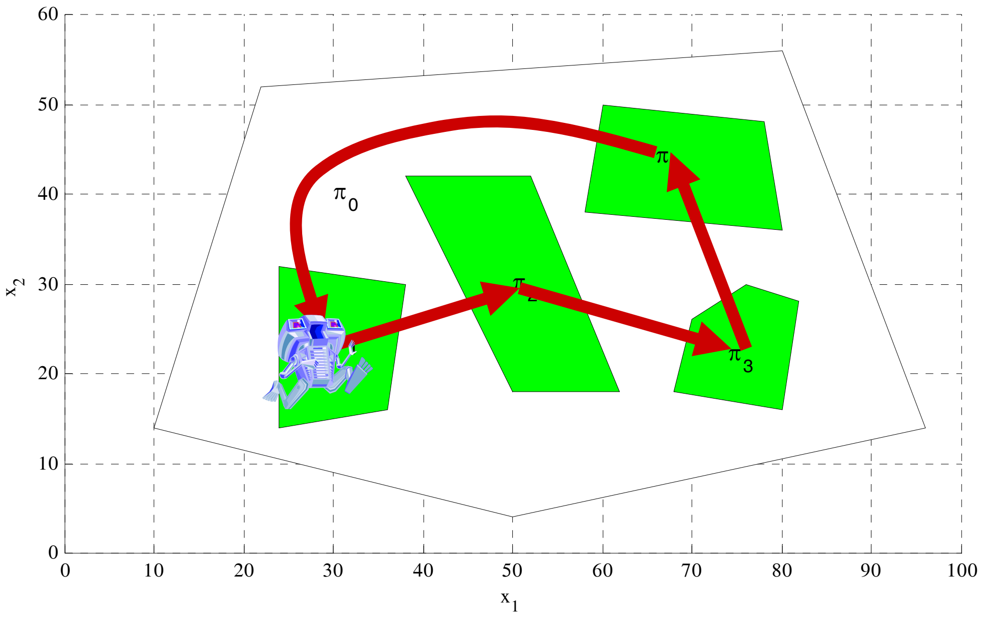
$$\ddot{x}(t) = u(t), \ \|u(t)\| \leq \mu$$

$$y(t) = x(t)$$

S

$$x(t) \in \Re^2, x(0) \in X_0, \dot{x}(0) = 0, y(t) \in \Re^2.$$

Motivation – Motion Planning

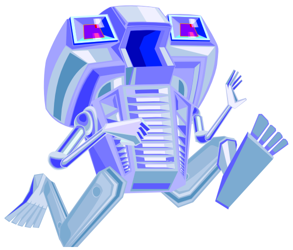


Task: "Stay always in π_0 and visit area π_2, then area π_3, then area π_4 and, finally, return to and stay in region π_1 while avoiding areas π_2 and π_3"

RTL: $\square\pi_0\wedge\Diamond(\pi_2\wedge\Diamond(\pi_3\wedge\Diamond(\pi_4\wedge(\neg\pi_2\wedge\neg\pi_3)U\square\pi_1)))$

Overview of solution

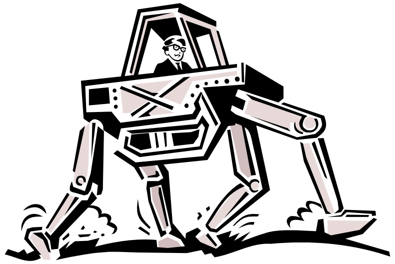
Agile robot



Abstraction

$$\xrightarrow{\varphi'}$$

Sluggish robot



$$\ddot{x}(t) = u(t),\ \|u(t)\| \le \mu$$

$$y(t) = x(t)$$

S

$y(t)$

$$x(t) \in \Re^2,\ x(0) = x_0,\ \dot{x}(0) = 0,\ y(t) \in \Re^2$$

$$\varphi \in RTL$$

$$\dot{z}(t) = v(t),\ \|v(t)\| \le \nu$$

$$y'(t) = z(t)$$

S'

$y'(t)$

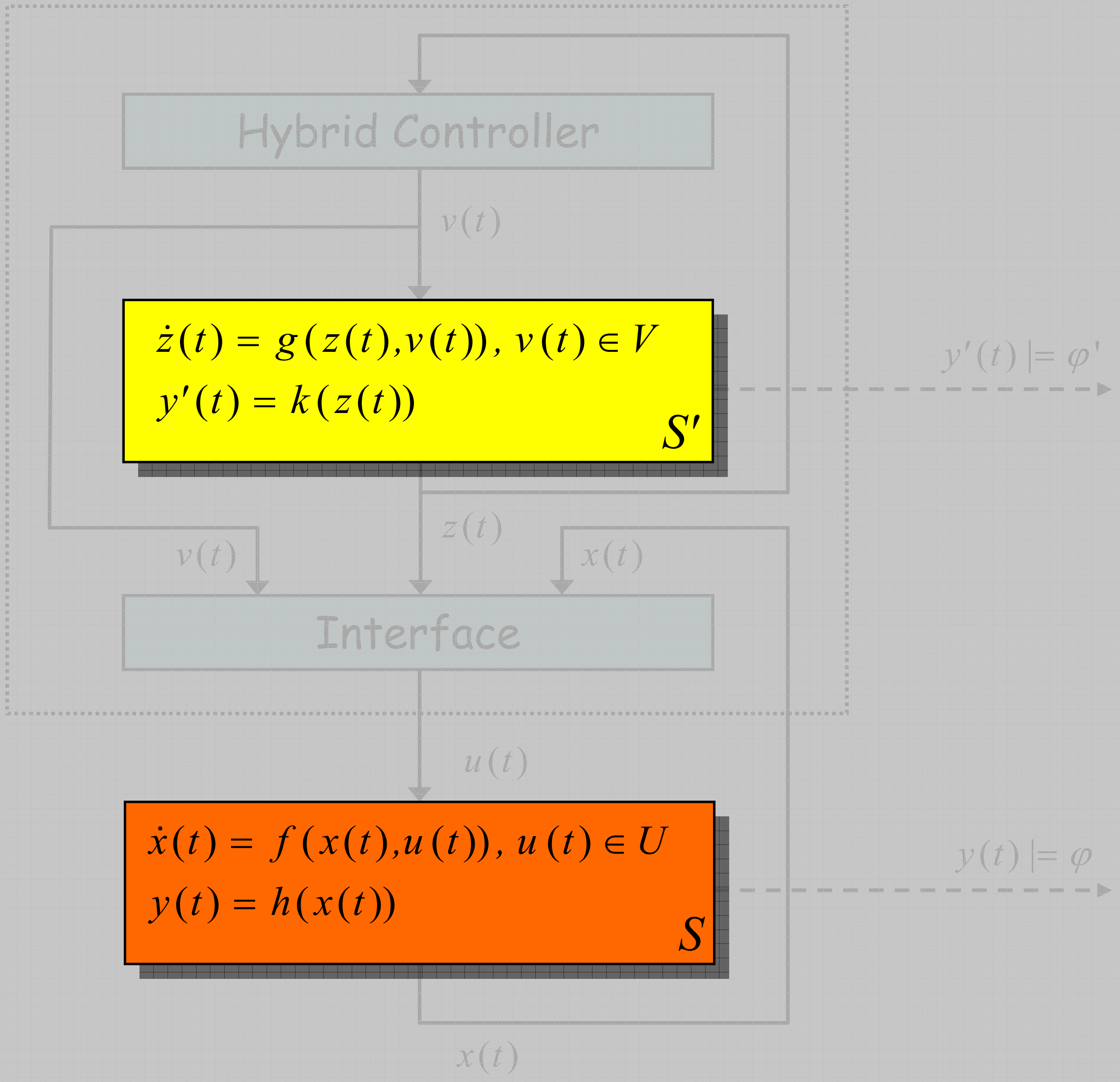
$$z(t) \in \Re^2,\ z(0) = x_0,\ y'(t) \in \Re^2$$

$$\varphi' \in RTL$$

$\longleftarrow$ Refinement

Girard, Pappas
CDC 2005, CDC 2006, Automatica

Fainekos, Kress-Gazit, Pappas
ICRA 2005, CDC 2005, CDC 2006
Kress-Gazit, Fainekos, Pappas
ICRA 2007



Hybrid Controller

$v(t)$

$$\dot{z}(t) = g(z(t), v(t)),\ v(t) \in V$$

$$y'(t) = k(z(t))$$

S'

$y'(t) \models \varphi'$

$v(t)$

$z(t)$

$x(t)$

Interface

$u(t)$

$$\dot{x}(t) = f(x(t), u(t)),\ u(t) \in U$$

$$y(t) = h(x(t))$$

S

$y(t) \models \varphi$

$x(t)$

A. Girard & G.J. Pappas, *Approximation Metrics for Discrete and Continuous Systems*, IEEE TAC, to appear.

Approximate Simulation Relation

$$\dot{x}(t) = f(x(t), u(t)),\ u(t) \in U$$
$$y(t) = h(x(t))$$

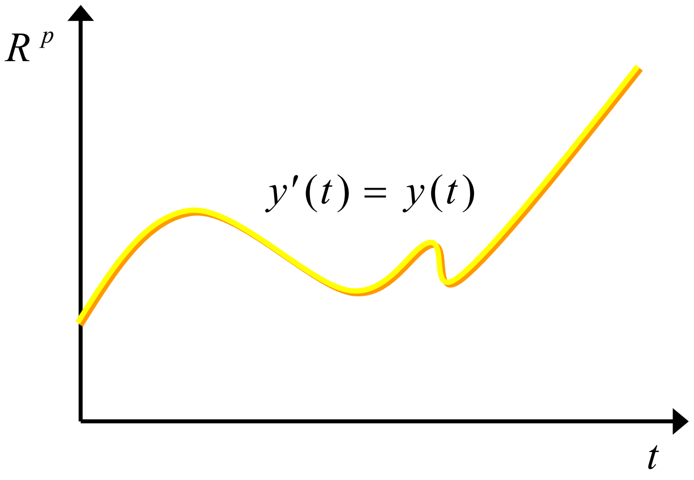
S $\xrightarrow{y(t)}$

$$x(t) \in \Re^n,\ x(0) = x_0,\ y(t) \in \Re^p$$

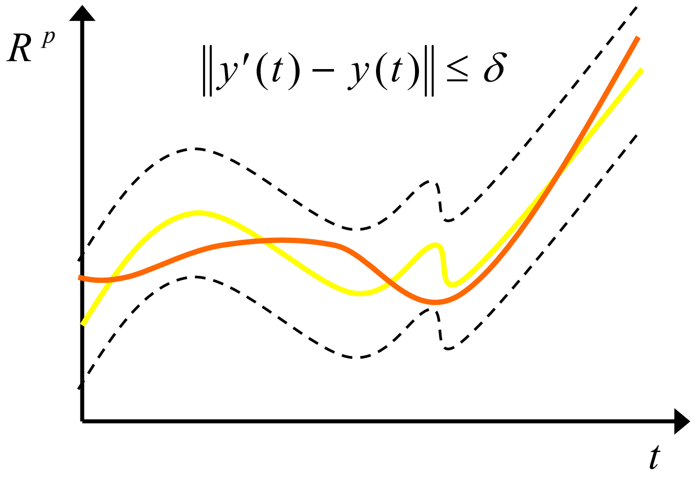
$$\dot{z}(t) = g(z(t), v(t)),\ v(t) \in V$$
$$y'(t) = k(z(t))$$

S' $\xrightarrow{y'(t)}$

$$z(t) \in \Re^m,\ z(0) = z_0,\ y'(t) \in \Re^p$$



Exact simulation



Approximate simulation

Approximate Simulation Relation

$$\dot{x}(t) = f(x(t),u(t)),\; u(t) \in U$$
$$y(t) = h(x(t))$$
$$S \xrightarrow{y(t)}$$
$$x(t) \in \Re^{n},\; x(0) = x_0,\; y(t) \in \Re^{p}$$

$$\dot{z}(t) = g(z(t),v(t)),\; v(t) \in V$$
$$y'(t) = k(z(t))$$
$$S' \xrightarrow{y'(t)}$$
$$z(t) \in \Re^{m},\; z(0) = z_0,\; y'(t) \in \Re^{p}$$

- A relation $R \subseteq \mathbf{R}^m \times \mathbf{R}^n$ is a δ-approximate simulation if for all $(z,x) \in R$,

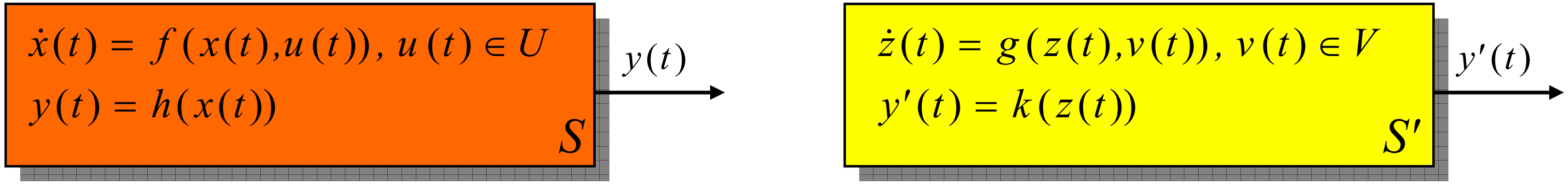
  1. $\|k(z) - h(x)\| \leq \delta$
  2. for all $T \geq 0$, for all trajectories $z(t)$ of *S'* such that $z(0)=z_0$, there exists a trajectory $x(t)$ of *S* such that $x(0)=x_0$ satisfying:

$$\forall t \in [0,T],\; (z(t), x(t)) \in R.$$

If initially the states of S and S' are in R (i.e. $(z_0,x_0) \in R$), then

Any observed trajectory of S' has an observed trajectory of S in its δ-neighborhood.

Simulation Functions



$$\dot{x}(t) = f(x(t),u(t)),\ u(t) \in U$$
$$y(t) = h(x(t))$$

$$x(t) \in \Re^{n},\ x(0) = x_0,\ y(t) \in \Re^{p}$$

$$\dot{z}(t) = g(z(t),v(t)),\ v(t) \in V$$
$$y'(t) = k(z(t))$$

$$z(t) \in \Re^{m},\ z(0) = z_0,\ y'(t) \in \Re^{p}$$

- We can construct approximate simulation relations using simulation functions.

- A simulation function $W: \mathbf{R}^{m} \times \mathbf{R}^{n} \to \mathbf{R}^{+}$ satisfies:

$$W(z,x) \geq \left\| k(z) - h(x) \right\|^{2}$$

$$\sup_{v \in V} \inf_{u \in U} \left(\frac{\partial W(z,x)}{\partial z} g(z,v) + \frac{\partial W(z,x)}{\partial x} f(x,u) \right) \leq 0$$

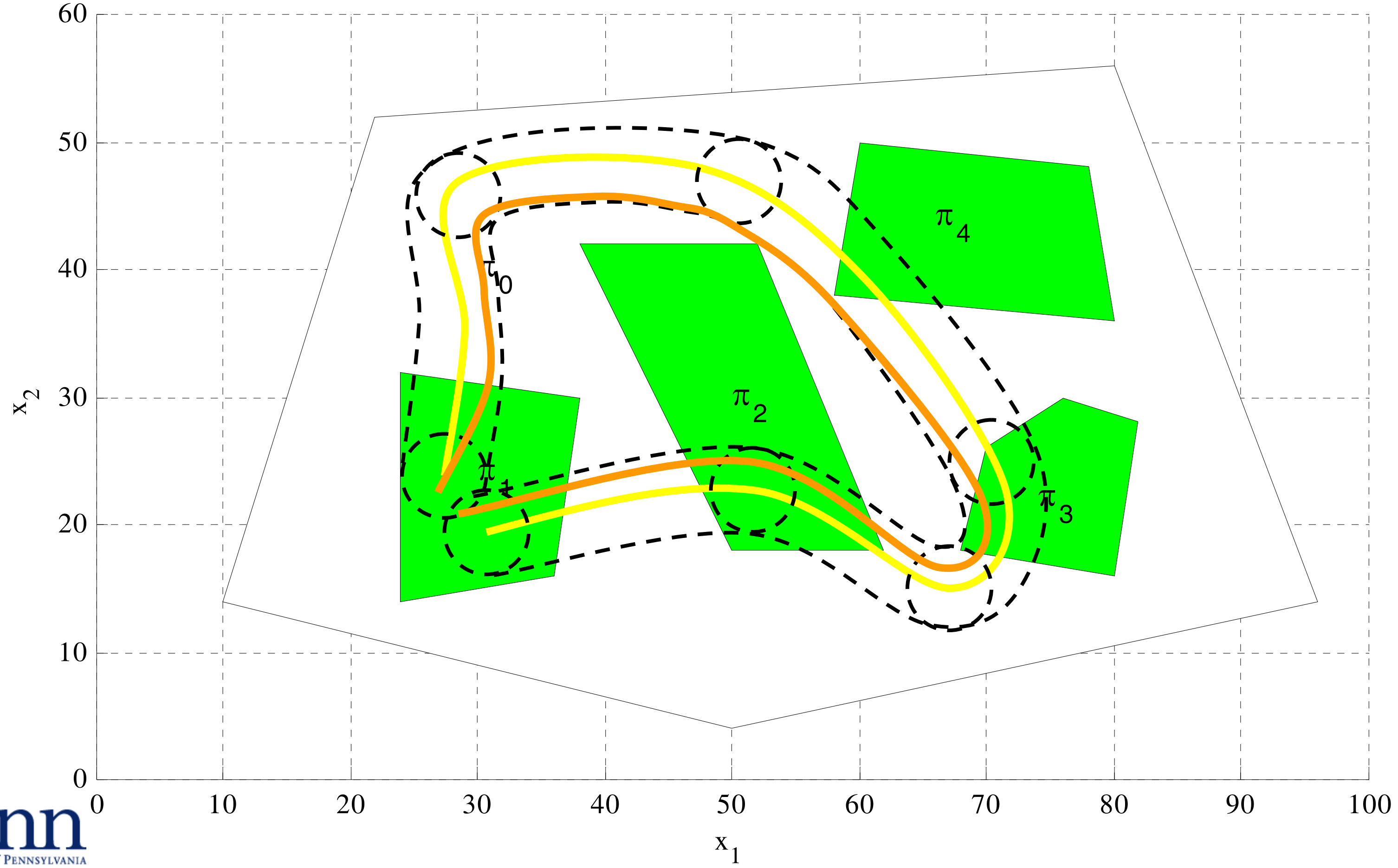
- Effective characterization of approximate simulation relations

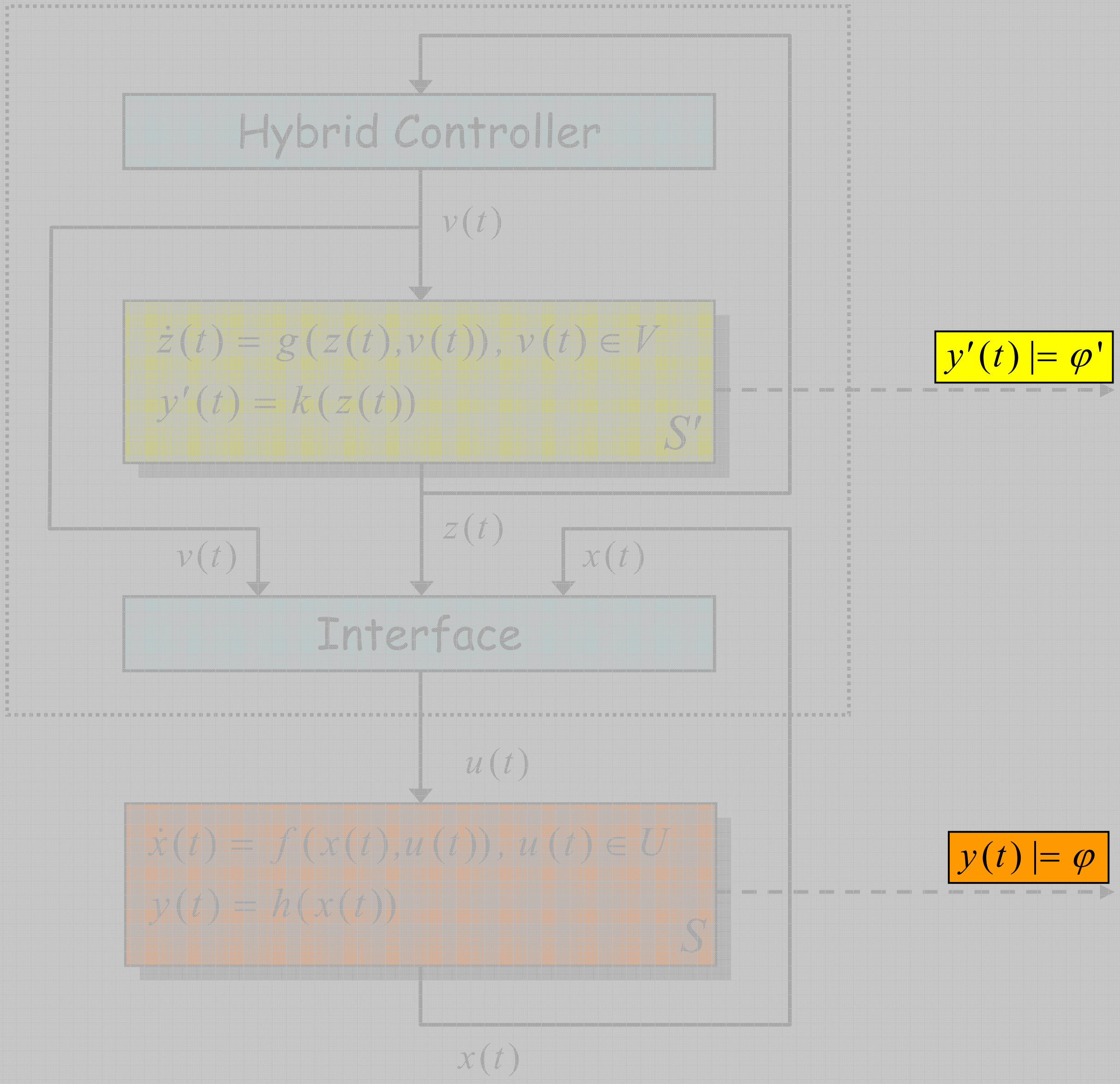
$R = \left\{ (z,x) \middle| \, W(z,x) \leq \delta^{2} \right\}$ is an approximate simulation relation of precision δ.

Abstraction

- Let S' be an abstraction of S such that there is simulation function W.
- Let $\delta^2 = W(z_0, x_0)$, then

Any observed trajectory of S' has an observed trajectory of S in its δ-neighborhood.




Hybrid Controller
$v(t)$
$\dot{z}(t) = g(z(t), v(t)),\ v(t) \in V$
$y'(t) = k(z(t))$
S'
$y'(t) \models \varphi'$
$v(t)$
$z(t)$
$x(t)$
Interface
$u(t)$
$\dot{x}(t) = f(x(t), u(t)),\ u(t) \in U$
$y(t) = h(x(t))$
S
$y(t) \models \varphi$
$x(t)$

Linear Temporal Logic: Continuous Time

Like propositional logic, but also reasoning wrt time …

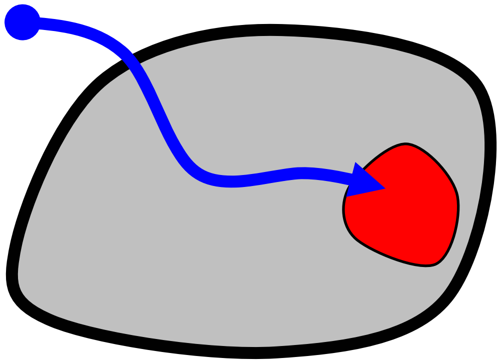
Basic operators:

Propositional: ∧, ∨, ¬

Temporal: ◊, □, U, R

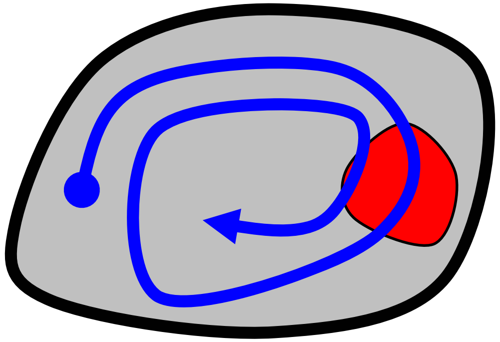
◊(red)

Eventually red



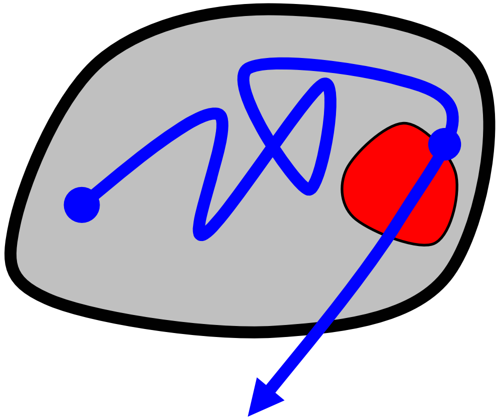
□(gray)

Always gray

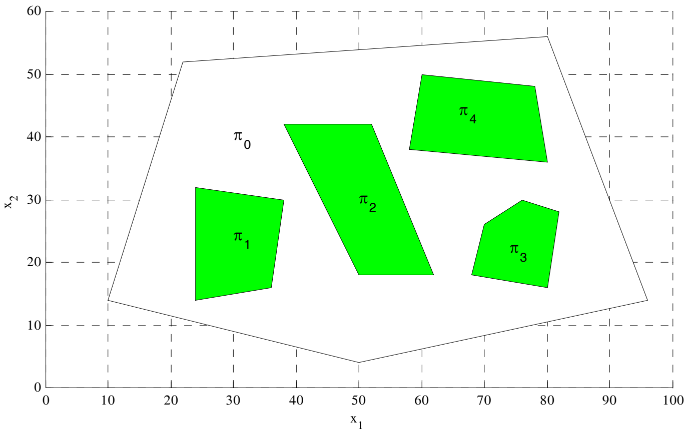


(gray) U (red)

gray Until red



LTL with continuous time semantics (RTL)



Let Π be a set of atomic propositions.

Define the denotation $[[.]] : \Pi \rightarrow P(X)$.

Syntax in NNF: $\varphi ::= \pi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 U \varphi_2 \mid \varphi_1 R \varphi_2$

Semantics:

$(y,[[.]]) \vDash \pi$ iff $y(0) \in [[\pi]]$

$(y,[[.]]) \vDash \varphi_1 \wedge \varphi_2$ iff $(y,[[.]]) \vDash \varphi_1$ and $(y,[[.]]) \vDash \varphi_2$

$(y,[[.]]) \vDash \varphi_1 \vee \varphi_2$ iff $(y,[[.]]) \vDash \varphi_1$ or $(y,[[.]]) \vDash \varphi_2$

$(y,[[.]]) \vDash \varphi_1 U \varphi_2$ iff $\exists t \geq 0$ s.t. $(y|_t,[[.]]) \vDash \varphi_2$ and $\forall s \in [0,t]$ $(y|_s,[[.]]) \vDash \varphi_1$

$(y,[[.]]) \vDash \varphi_1 R \varphi_2$ iff $\forall t \geq 0$ $(y|_t,[[.]]) \vDash \varphi_2$ or $\exists s \in [0,t]$ s.t. $(y|_s,[[.]]) \vDash \varphi_1$

Some notation: $y|_t(s) = y(t+s)$ $\Diamond\varphi = T U \varphi$ $\Box\varphi = F R \varphi$

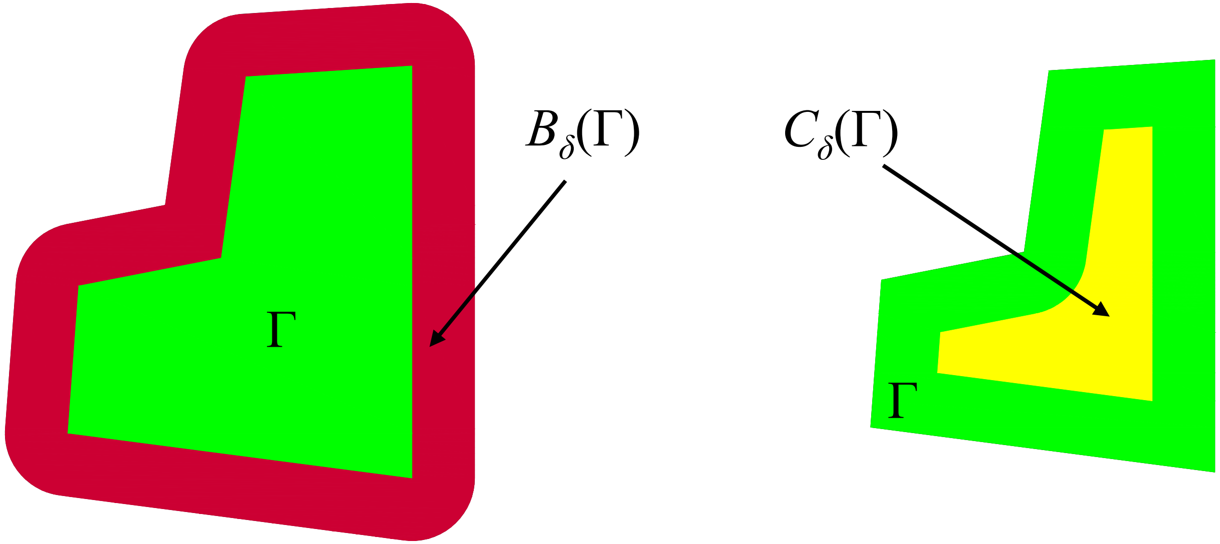
Contraction - Expansion

For δ>0, the δ-ball centered at a∈A is defined as:

$$B_\delta(\alpha)=\{\beta\in A:\|\alpha-\beta\|\le\delta\}$$

For Γ⊆A, the contraction and expansion are defined as:

$$C_\delta(\Gamma)=\{\alpha\in A:B_\delta(\alpha)\subseteq\Gamma\}$$
$$B_\delta(\Gamma)=\{\alpha\in A:B_\delta(\alpha)\cap\Gamma\neq\varnothing\}$$



"Robustifying" an RTL formula

Define a translation function

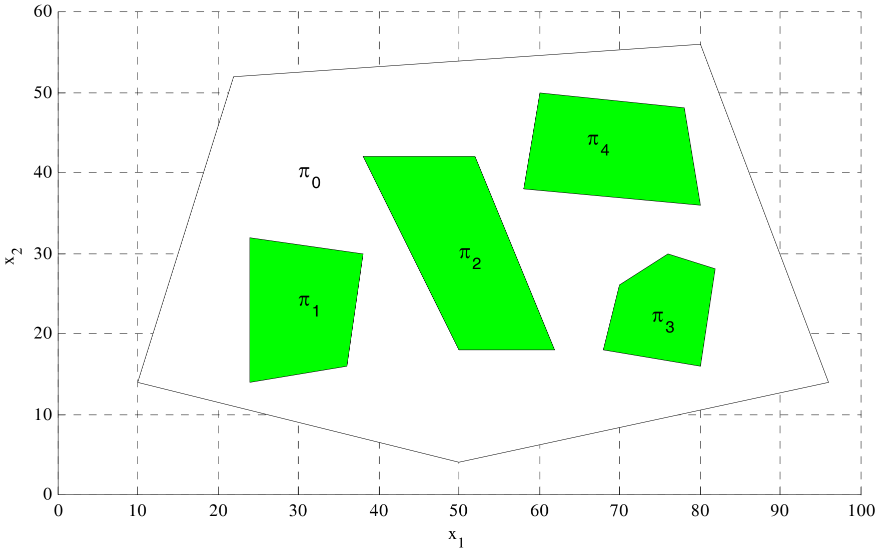
rob : $\Pi \rightarrow \Xi_{\Pi}$

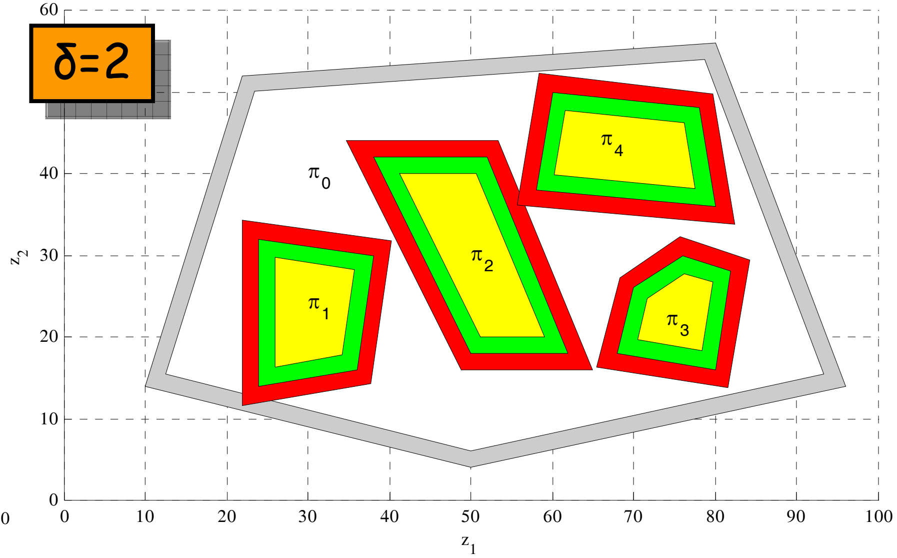
where

$\Xi_{\Pi} = \{\xi_{\alpha} | \alpha=\pi \text{ or } \alpha=\neg\pi \text{ for } \pi\in\Pi\}$

Define new map $[[.]]_{\delta} : \Xi_{\Pi} \rightarrow P(X)$

$$[[\xi]]_{\delta} := \begin{cases} C_{\delta}\left([[\pi]]^{c}\right) & if \quad \xi = \xi_{\neg\pi} \\ C_{\delta}\left([[\pi]]\right) & if \quad \xi = \xi_{\pi} \end{cases}$$





RTL: $\Box\xi_{\pi 0}\wedge\Diamond(\xi_{\pi 2}\wedge\Diamond(\xi_{\pi 3}\wedge\Diamond(\xi_{\pi 4}\wedge(\xi_{\neg\pi 2}\wedge\xi_{\neg\pi 3})U\Box\xi_{\pi 1})))$

Main Result

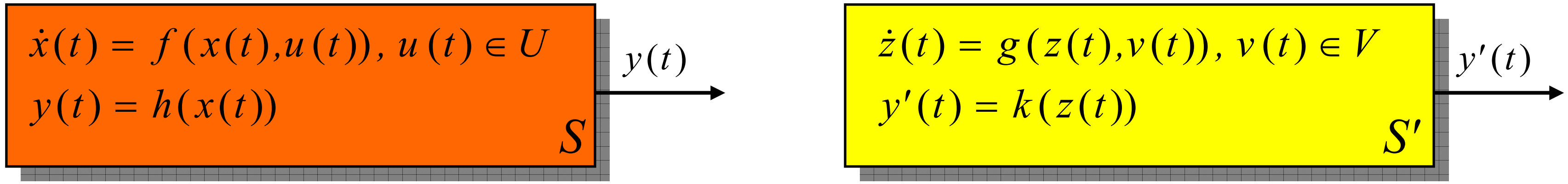
Theorem:

Consider an RTL formula φ, a map $[[.]] : \Pi \rightarrow P(X)$ and a number $\delta > 0$, then for all functions $y(t), y'(t)$ from R^+ to R^p such that for all $t \geq 0$,

$$||y(t) - y'(t)|| \leq \delta$$

it is

$$(y', [[.]]_\delta) \models \mathbf{rob}(\varphi) \quad \Longrightarrow \quad (y, [[.]]) \models \varphi$$



$$\dot{x}(t) = f(x(t),u(t)),\ u(t) \in U$$
$$y(t) = h(x(t))$$

S $\xrightarrow{y(t)}$

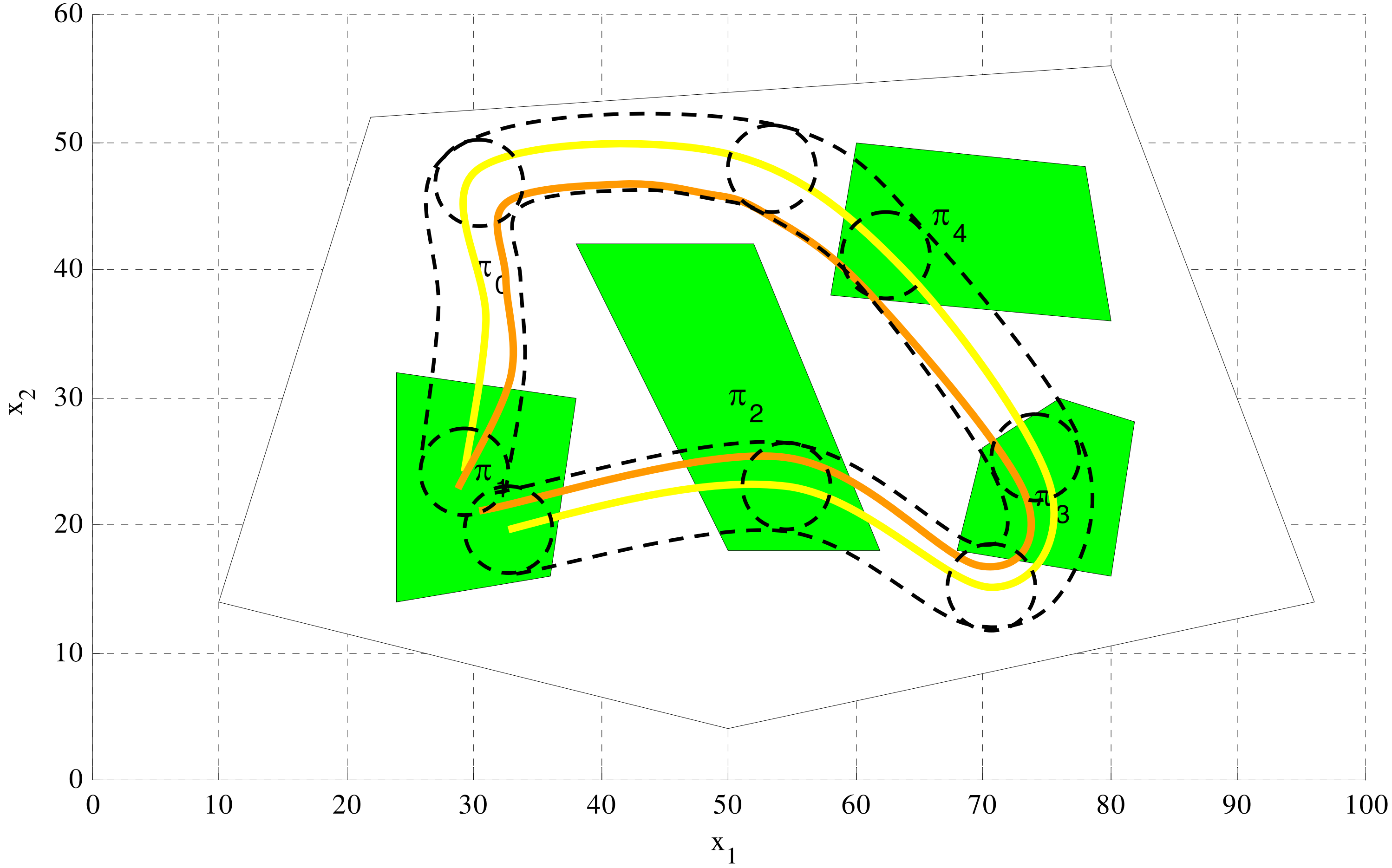
$$x(t) \in \Re^n,\ x(0) = x_0,\ y(t) \in \Re^p$$

$$\dot{z}(t) = g(z(t),v(t)),\ v(t) \in V$$
$$y'(t) = k(z(t))$$

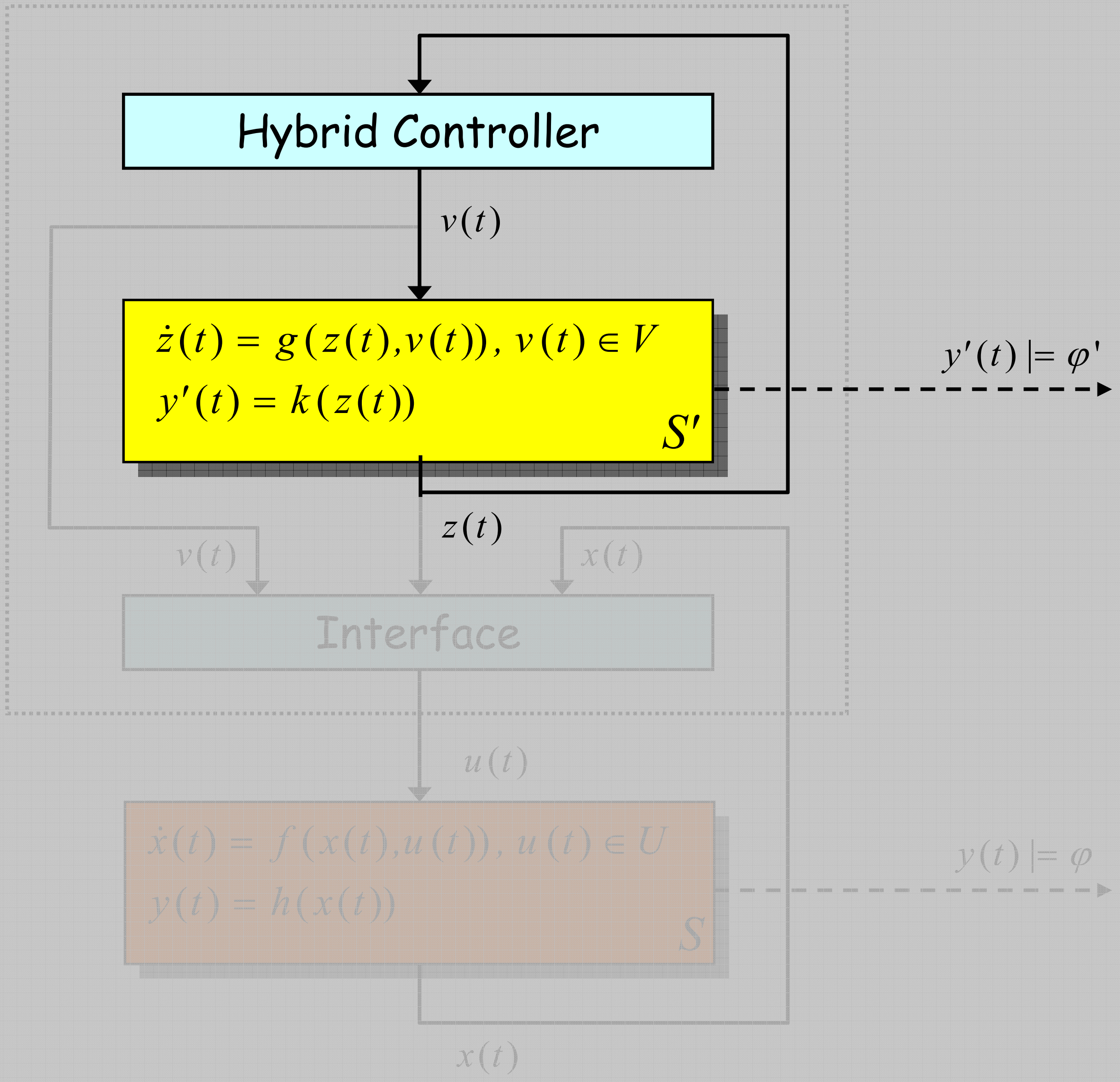
S' $\xrightarrow{y'(t)}$

$$z(t) \in \Re^m,\ z(0) = z_0,\ y'(t) \in \Re^p$$

Back to our simple example



[Gupta, Henzinger, Jagadeesan, HART '97]



Hybrid Controller

$v(t)$

$$\dot{z}(t) = g(z(t), v(t)),\ v(t) \in V$$
$$y'(t) = k(z(t))$$

S'

$y'(t) \models \varphi'$

$z(t)$

$v(t)$

$x(t)$

Interface

$u(t)$

$$\dot{x}(t) = f(x(t), u(t)),\ u(t) \in U$$
$$y(t) = h(x(t))$$

S

$y(t) \models \varphi$

$x(t)$

G. E. Fainekos, S. G. Loizou and G. J. Pappas, *Translating Temporal Logic to Controller Specifications*, CDC 2006

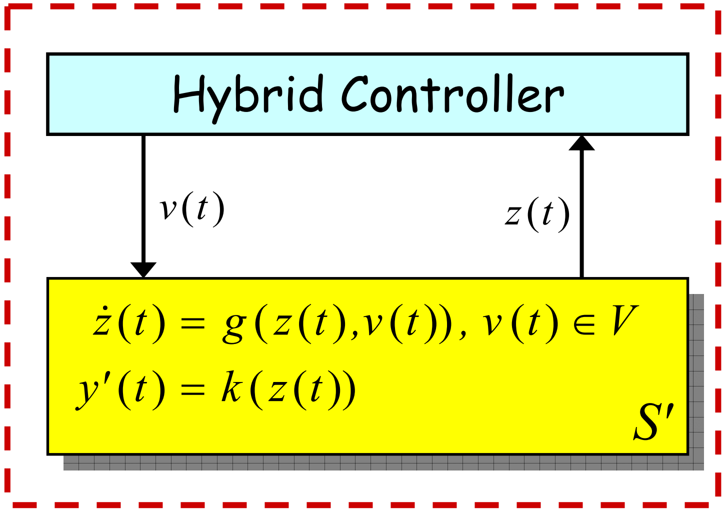
Hybrid Controller

A hybrid controller is a tuple

H' = (Z, V, L, E, Inv, Out, Init, Guard)

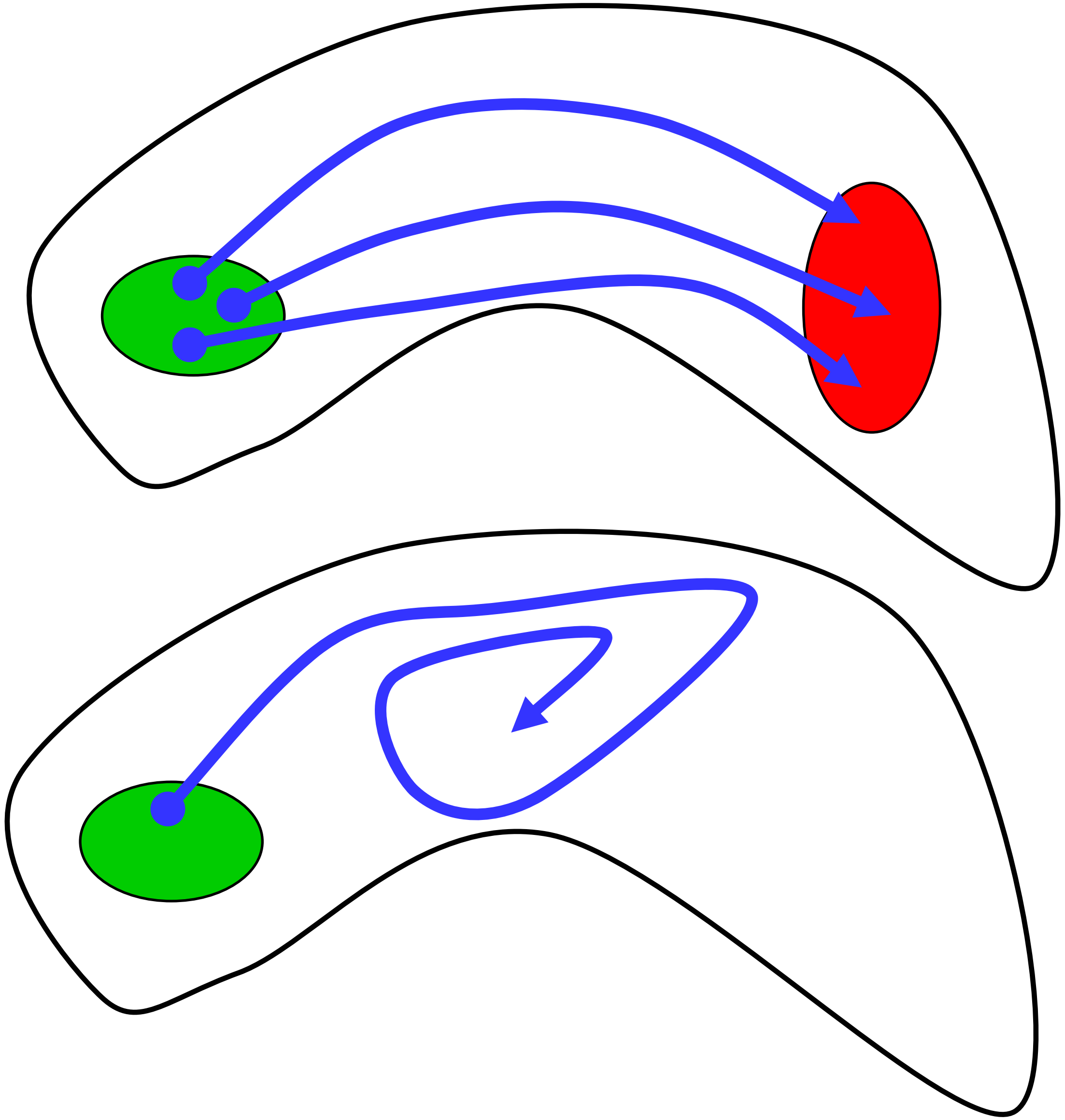
where

- Z is the state space of the system S'
- L is the set of control locations,
- $E \subseteq L \times L$ is the set of control switches
- Inv : $L \rightarrow P(Z)$ assigns an invariant set to each location
- Out : $L \times Z \rightarrow V$ is the control input for S'
- Init : $L \rightarrow P(Z)$ assigns to each control location a set of initial conditions
- Guard : $E \rightarrow P(Z)$ is the guard condition that enables a control switch e in E

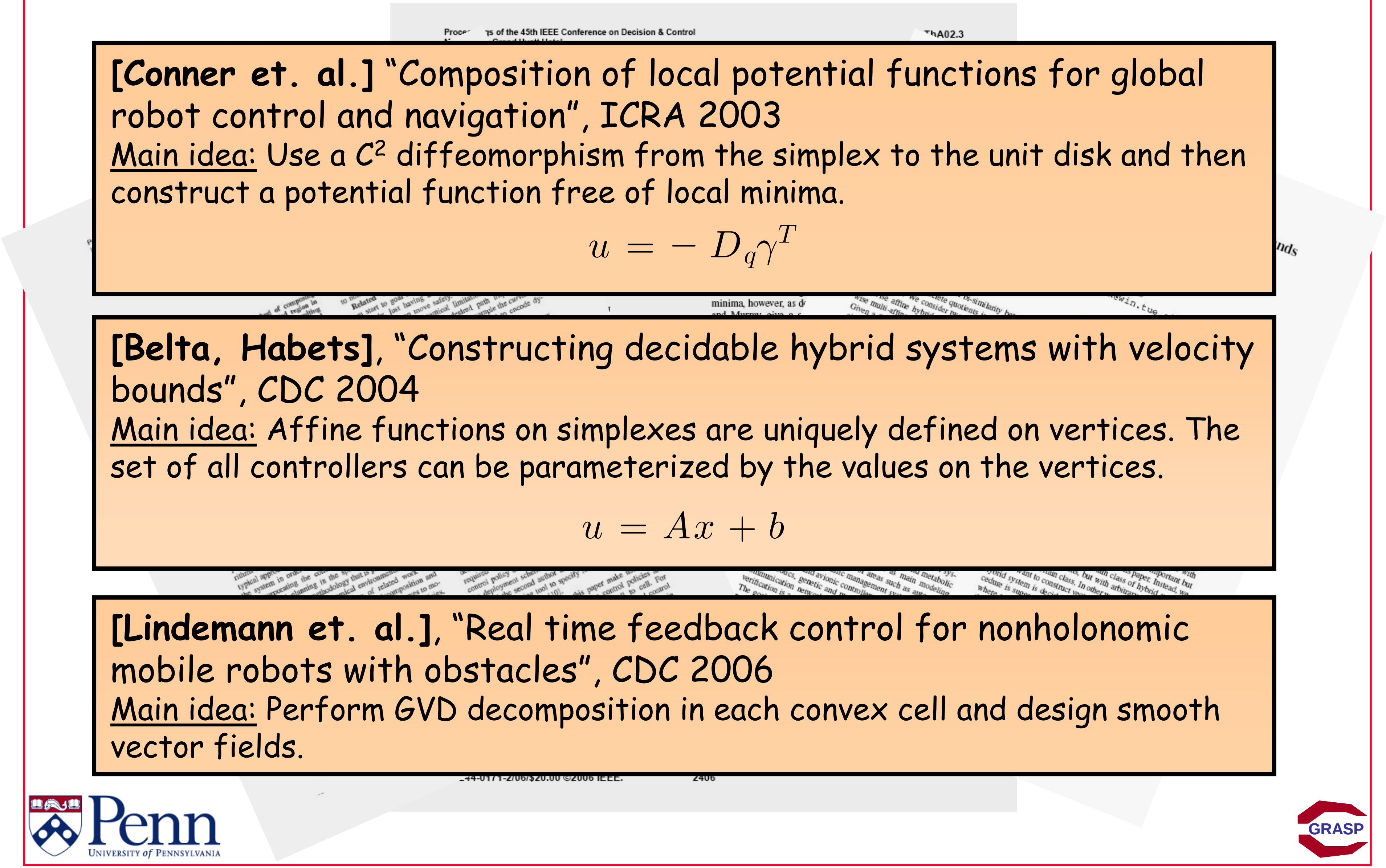


Hybrid Automaton

Each control location



How can you design such controllers?



[Conner et. al.] "Composition of local potential functions for global robot control and navigation", ICRA 2003

Main idea: Use a C^2 diffeomorphism from the simplex to the unit disk and then construct a potential function free of local minima.

$$u = -D_q\gamma^T$$

[Belta, Habets], "Constructing decidable hybrid systems with velocity bounds", CDC 2004

Main idea: Affine functions on simplexes are uniquely defined on vertices. The set of all controllers can be parameterized by the values on the vertices.

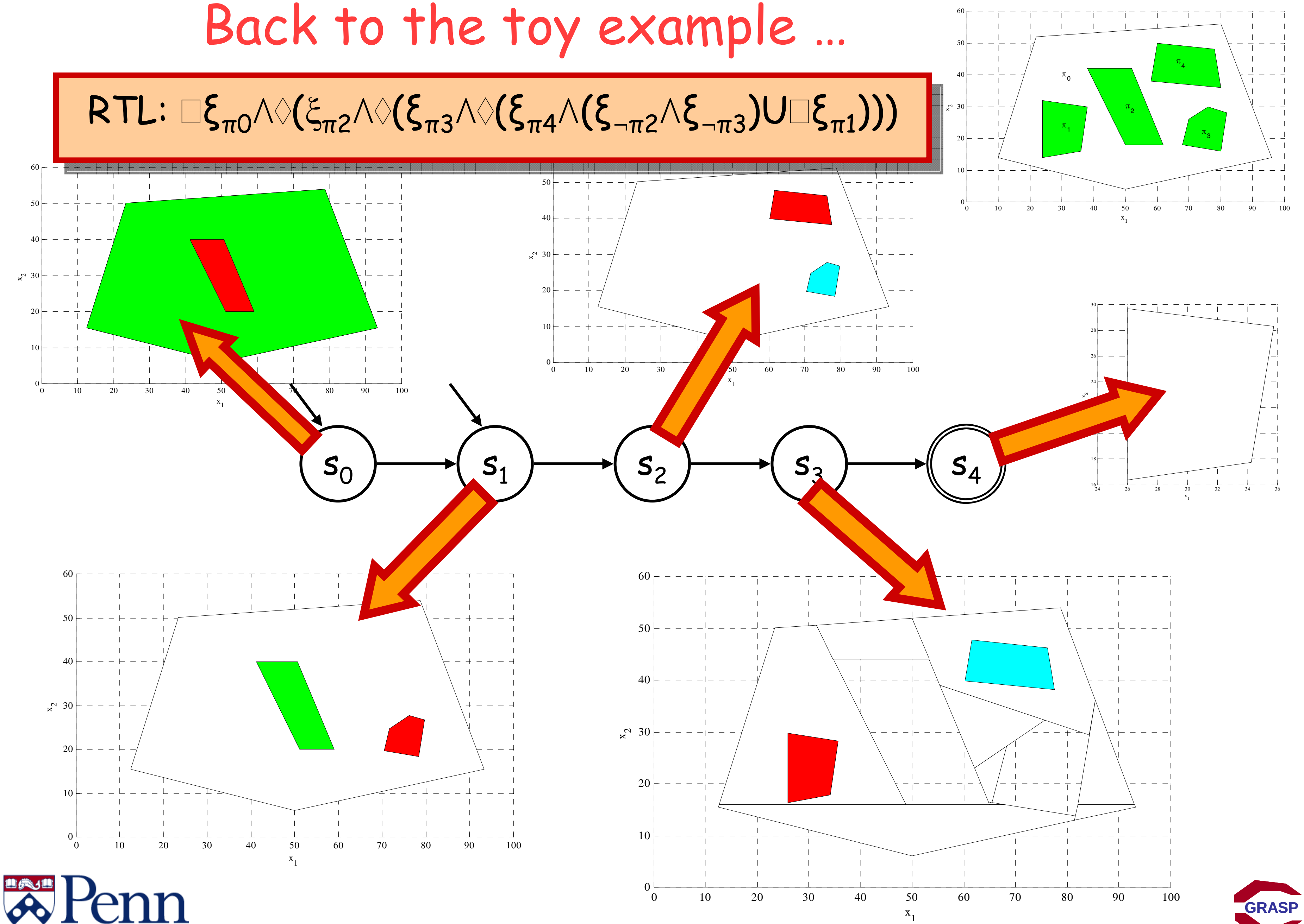
$$u = Ax + b$$

[Lindemann et. al.], "Real time feedback control for nonholonomic mobile robots with obstacles", CDC 2006

Main idea: Perform GVD decomposition in each convex cell and design smooth vector fields.

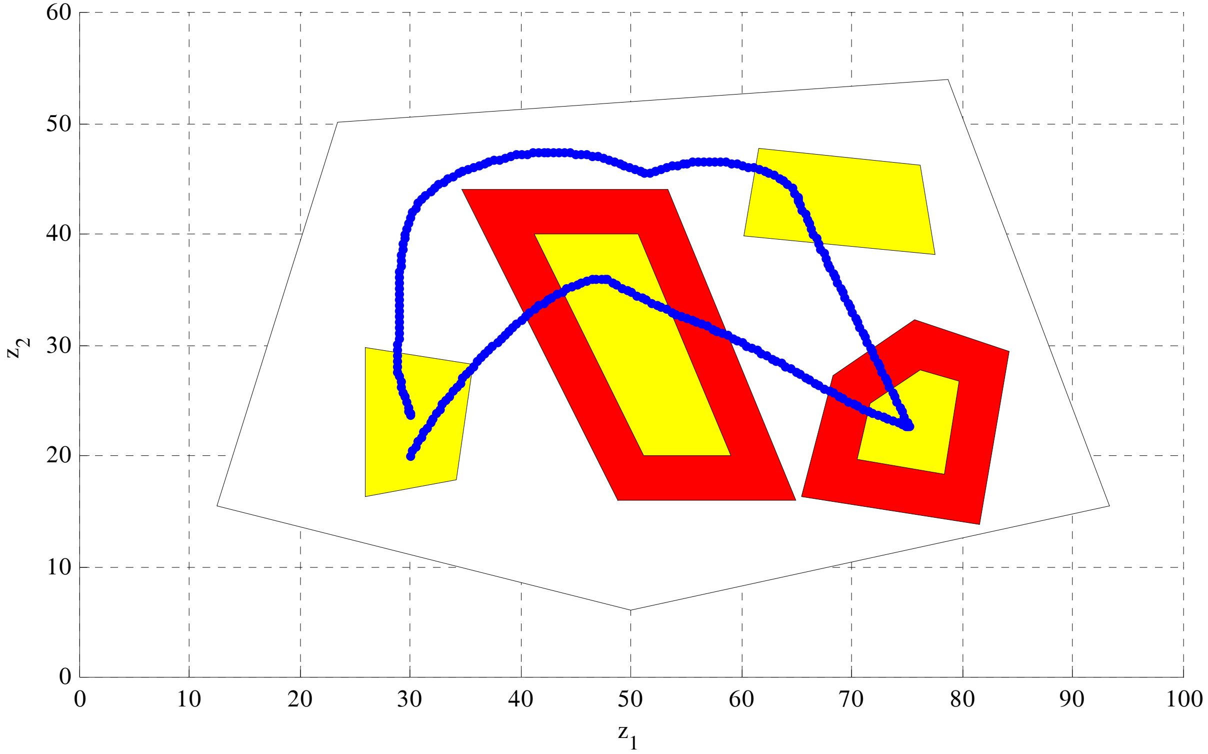
Back to the toy example …

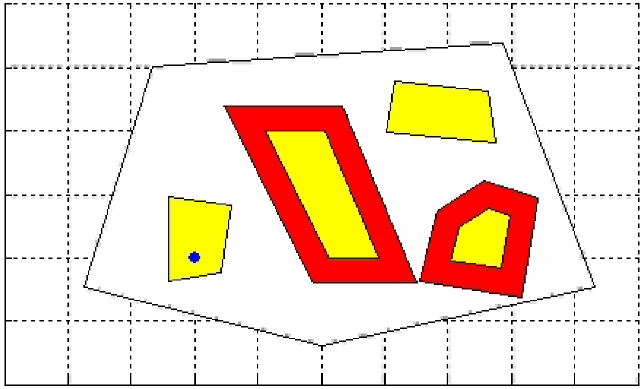
RTL: $\square \xi_{\pi 0} \wedge \lozenge(\xi_{\pi 2} \wedge \lozenge(\xi_{\pi 3} \wedge \lozenge(\xi_{\pi 4} \wedge (\xi_{\neg\pi 2} \wedge \xi_{\neg\pi 3}) \mathcal{U} \square \xi_{\pi 1})))$

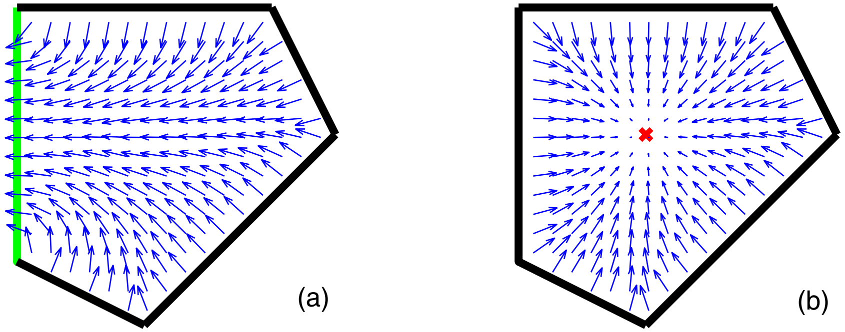


Back to the toy example …

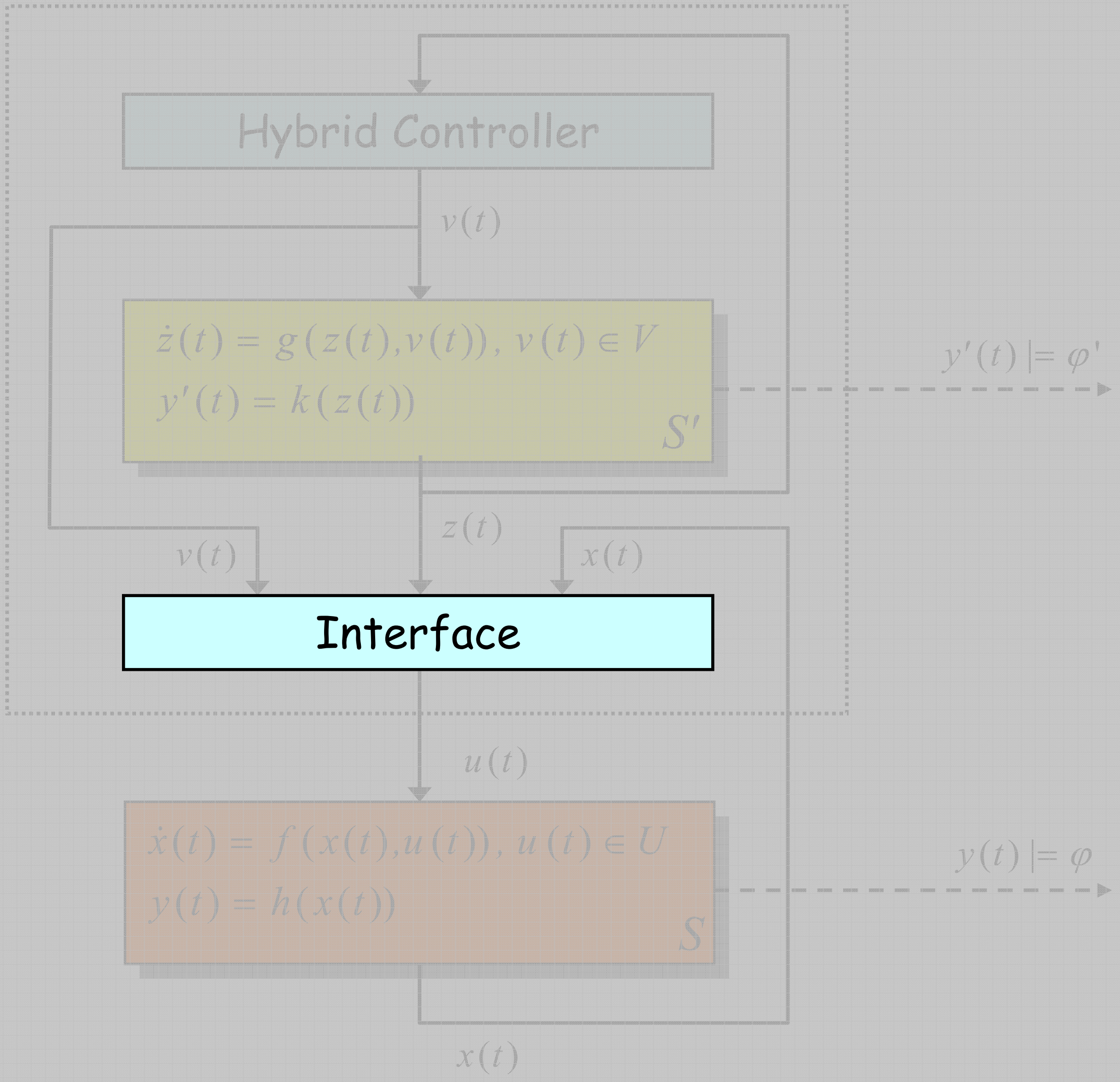
δ=2







Conner et. al. IROS 2003



Hybrid Controller

$v(t)$

$\dot{z}(t) = g(z(t), v(t)),\ v(t) \in V$

$y'(t) = k(z(t))$

S'

$y'(t) \models \varphi'$

$v(t)$

$z(t)$

$x(t)$

Interface

$u(t)$

$\dot{x}(t) = f(x(t), u(t)),\ u(t) \in U$

$y(t) = h(x(t))$

S

$y(t) \models \varphi$

$x(t)$

A. Girard & G.J. Pappas, *Hierarchical Control using Approximate Simulation Relations*, presented in CDC 2006

Refinement

$$\dot{x}(t) = f(x(t), u(t)),\ u(t) \in U$$
$$y(t) = h(x(t))$$

S $\xrightarrow{y(t)}$

$$x(t) \in \Re^n,\ x(0) = x_0,\ y(t) \in \Re^p$$

$$\dot{z}(t) = g(z(t), v(t)),\ v(t) \in V$$
$$y'(t) = k(z(t))$$

S' $\xrightarrow{y'(t)}$

$$z(t) \in \Re^m,\ z(0) = z_0,\ y'(t) \in \Re^p$$

- Can we synthesize $u(.)$ from the input $v(.)$?
- Interface $u_W : \mathbf{R}^m \times \mathbf{R}^n \times V \rightarrow U$ associated with simulation function W :

$$W(z, x) \geq \left\| k(z) - h(x) \right\|^2$$

$$\sup_{v \in V} \left(\frac{\partial W(z, x)}{\partial z} g(z, v) + \frac{\partial W(z, x)}{\partial x} f(x, u_W(z, x, v)) \right) \leq 0$$

Back to our simple example

$$\ddot{x}(t) = u(t),\ \|u(t)\| \leq \mu \qquad y(t) = x(t) \qquad S \xrightarrow{y(t)}$$

$$x(t) \in \Re^2,\ x(0) = x_0,\ \dot{x}(0) = 0,\ y(t) \in \Re^2$$

$$\dot{z}(t) = v(t),\ \|v(t)\| \leq \nu \qquad y'(t) = z(t) \qquad S' \xrightarrow{y'(t)}$$

$$z(t) \in \Re^2,\ z(0) = x_0,\ y'(t) \in \Re^2$$

- Simulation function and associated interface :

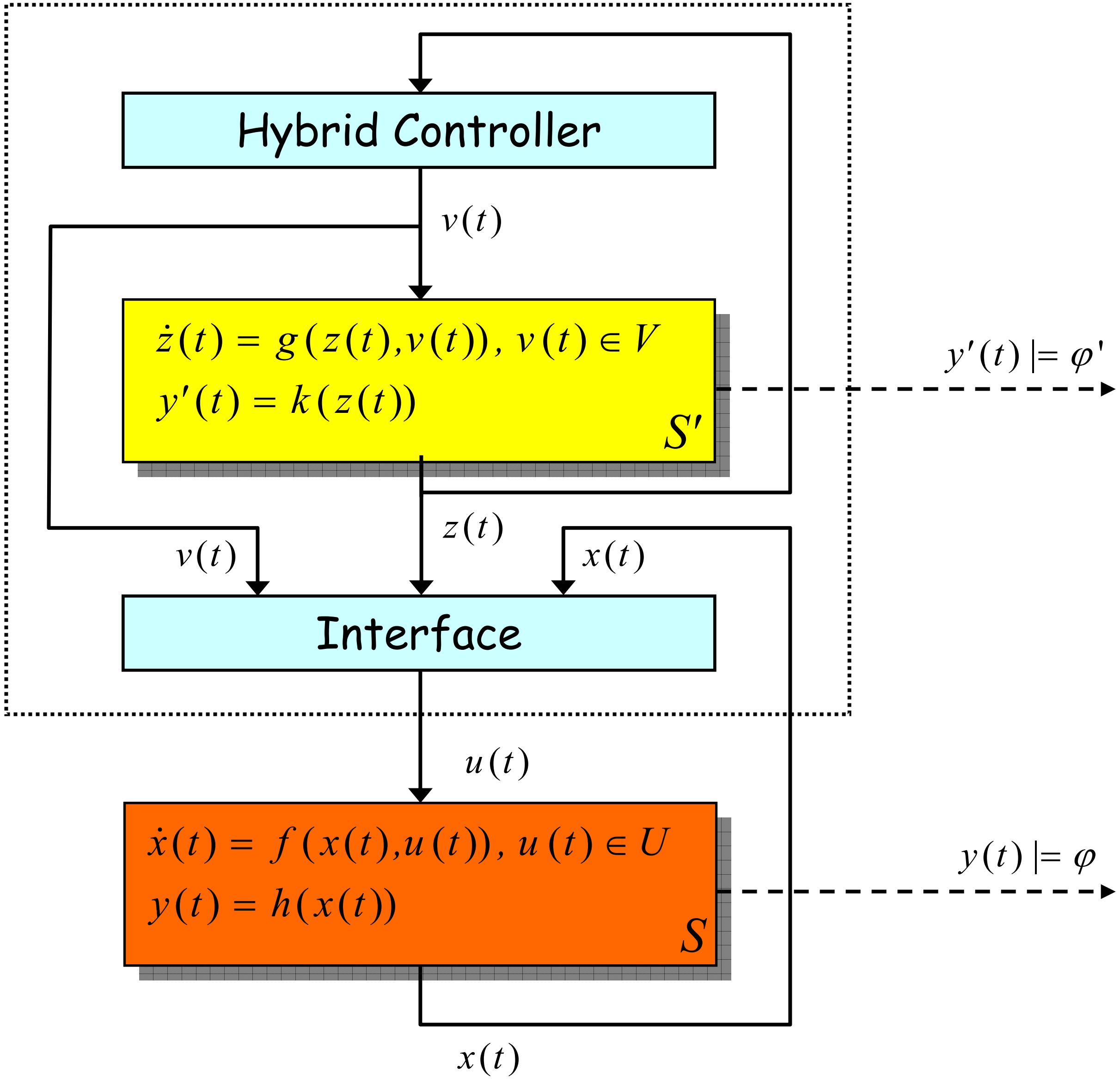
$$W(z, x, \dot{x}) = \max\left(\|z - x\|^2 + \alpha\|z - x - 2\dot{x}\|^2, 4\nu^2\right)$$

$$u_W(z, x, \dot{x}, v) = \frac{v}{2} + \frac{1+\alpha}{4\alpha}(z - x) - \dot{x}$$

- Condition :

$$\frac{\nu}{2}\left(1 + \left|1 - \frac{1}{\alpha}\right| + \frac{2}{\sqrt{\alpha}}\right) \leq \mu$$

Then R is an approximate simulation relation of precision *2ν* of S′ by S.



Hybrid Controller

$v(t)$

$\dot{z}(t) = g(z(t), v(t)),\ v(t) \in V$

$y'(t) = k(z(t))$

S'

$y'(t) \models \varphi'$

$v(t)$

$z(t)$

$x(t)$

Interface

$u(t)$

$\dot{x}(t) = f(x(t), u(t)),\ u(t) \in U$

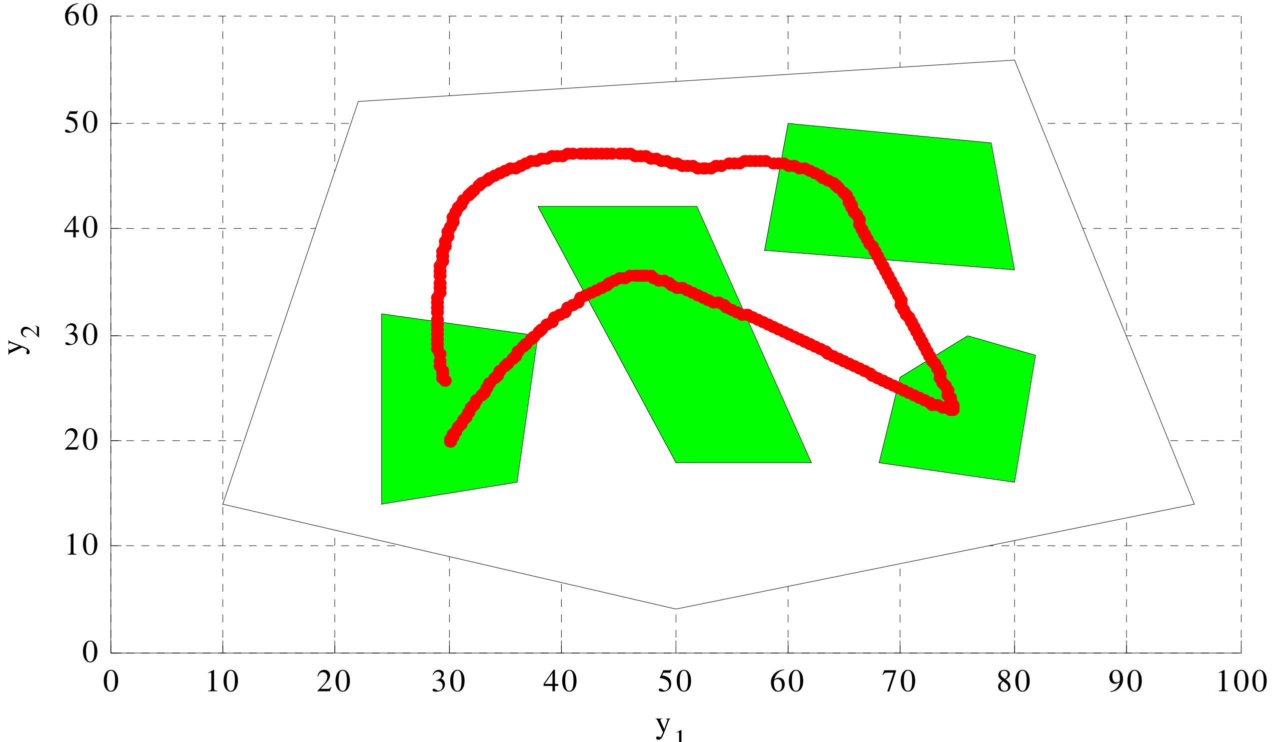
$y(t) = h(x(t))$

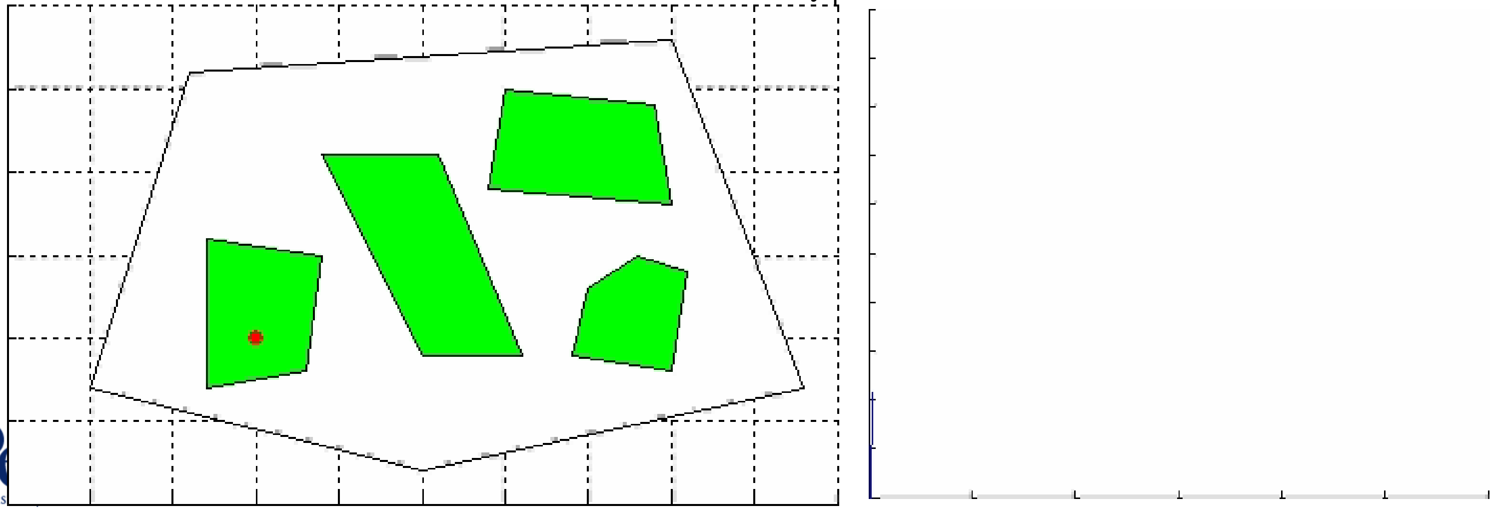
S

$y(t) \models \varphi$

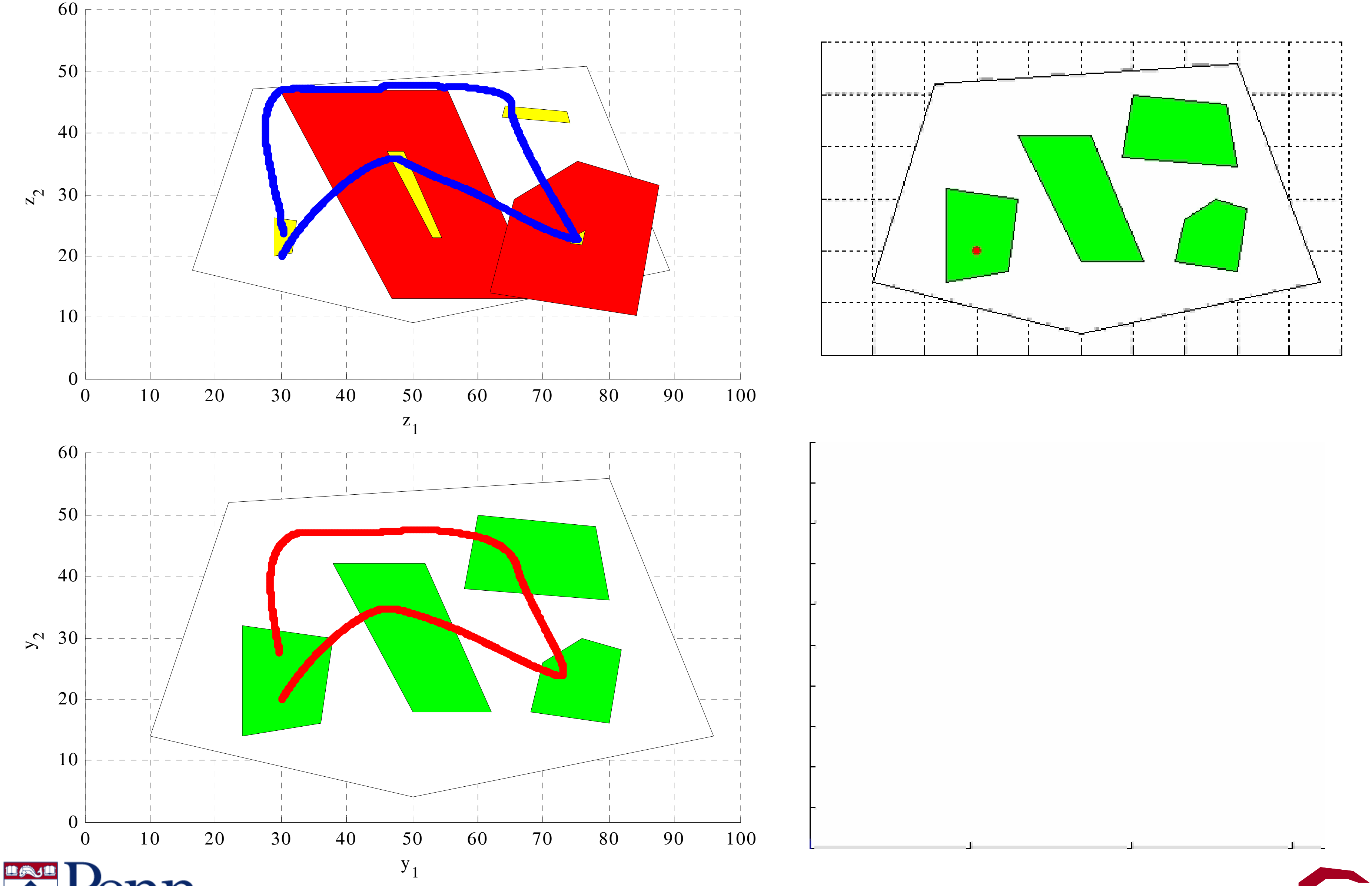
$x(t)$

Final Motion for v=1 (or δ=2)

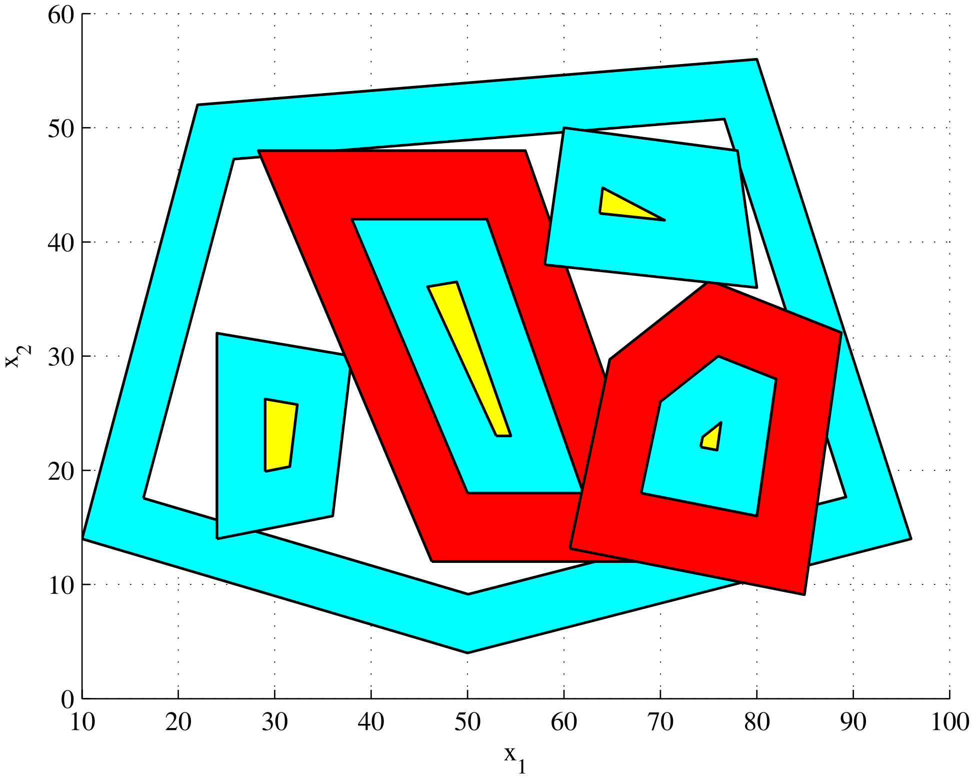




Example for v=2.5 (or δ=5)



If the abstract system is too fast …



Conclusions \ Future work

- ☑ Hierarchical framework for the synthesis of HA from temporal logic
  - ☑ Hierarchical controller
  - ☑ Approximate simulation relations
  - ☑ Robustness of temporal logic formulas

  - ☑ Takes advantage of existing methods for the design of HA from TL

  - ☑ Can provide solutions for complicated specifications, for systems with complicated continuous dynamics that operate in complicated environments

- ☐ Compute interfaces for other systems

- ☐ Design for distributed specifications and for open systems

- ☐ Make a public distribution of the prototype toolbox

Thank You!

Any Question(s) ?