

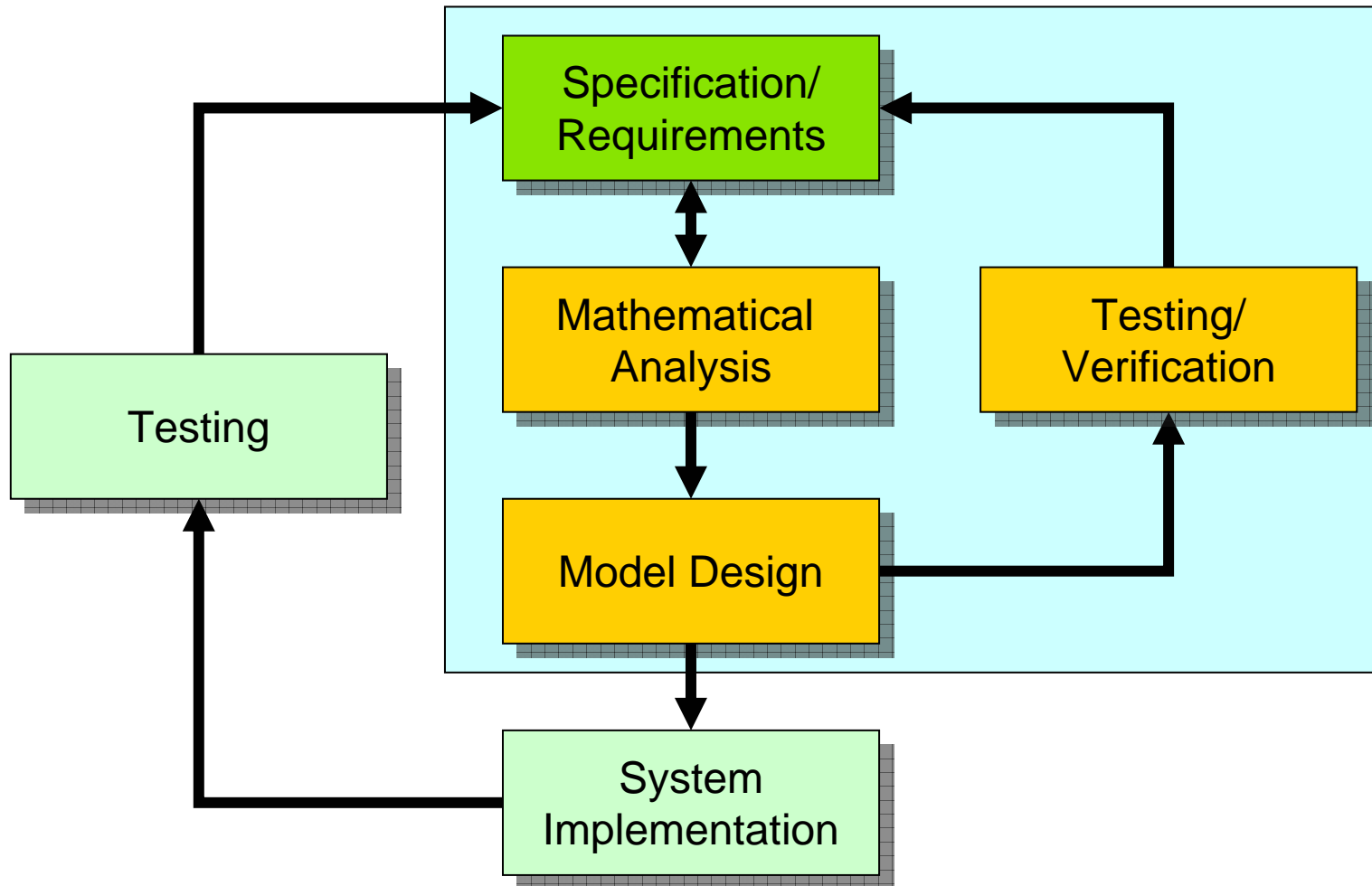
Robust Test Generation and Coverage for Hybrid Systems

A. Agung Julius, **Georgios E. Fainekos**, Madhukar Anand,
Insup Lee and George J. Pappas

Departments of ESE and CIS
University of Pennsylvania



Hybrid System Design Cycle





Testing VS Verification: continuous time

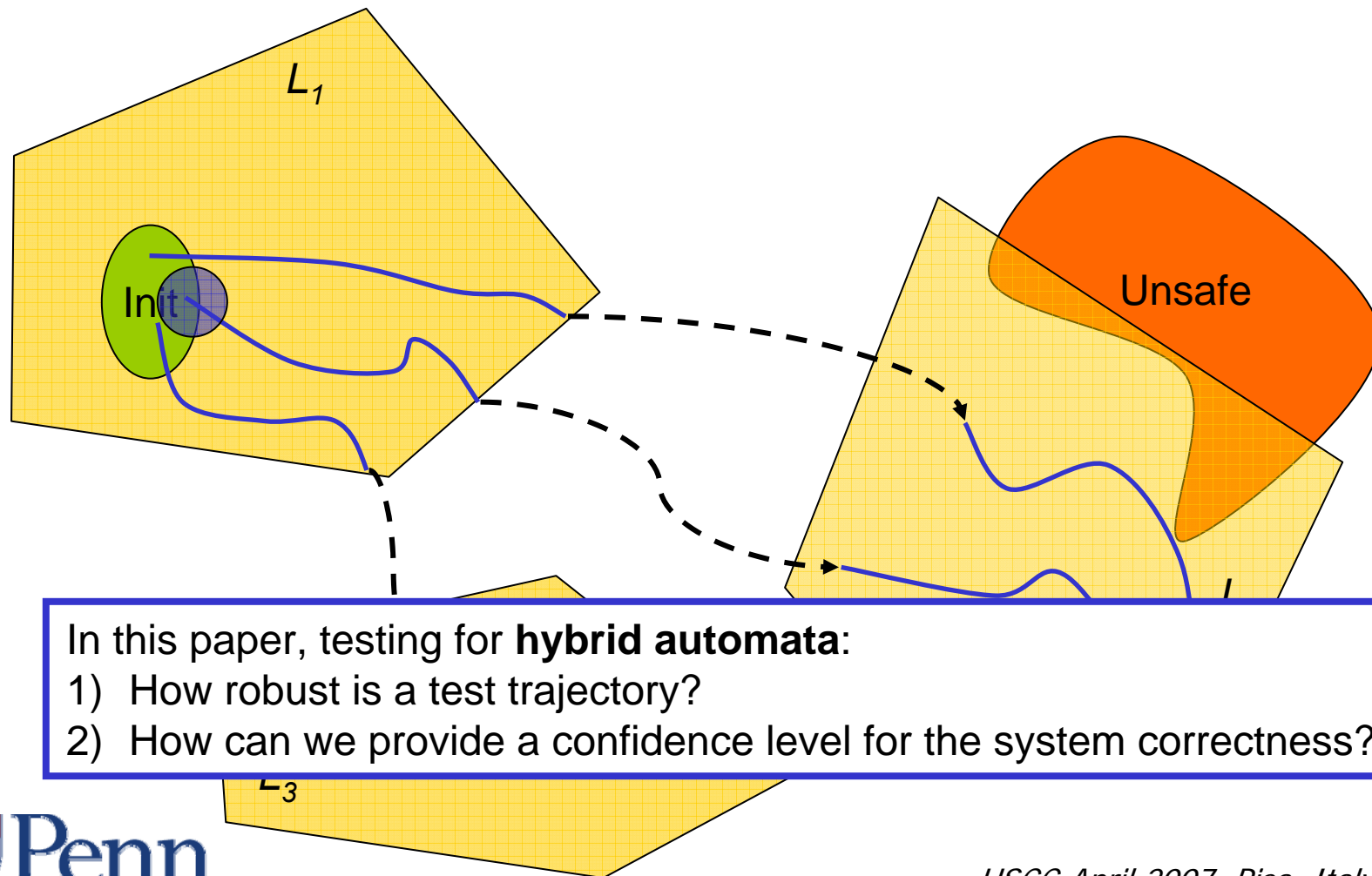
Systematic Testing

Badban, Franzle, Peleska, Teige SOQUA'06
Cheng, Kumar WAFR'06
Kim, Esposito ACC'05
Krichen, Tripakis FORMATS'04
Kapinski, Krogh, Maler, Stursberg HSCC'03
Branicky, Curtiss, Levine, Morgan, Yale Workshop'05
Bhatia, Frazzoli, HSCC'04

Testing

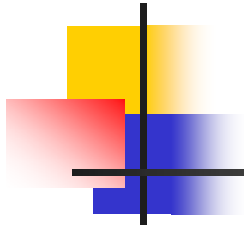
Tan, Kim, Sokolsky, Lee IRI'04
TorX [Bohnenkamp.et al]
Maler, Nickovic FORMATS'04
Briones, Brinksma FATES'04
van Osch FATES\RV'06
Krichen, Tripakis SPIN'04
UPPAAL-TRON [Mikucionis. et al]

Testing of hybrid systems



In this paper, testing for **hybrid automata**:

- 1) How robust is a test trajectory?
- 2) How can we provide a confidence level for the system correctness?



Defining the robustness of a simulation trajectory

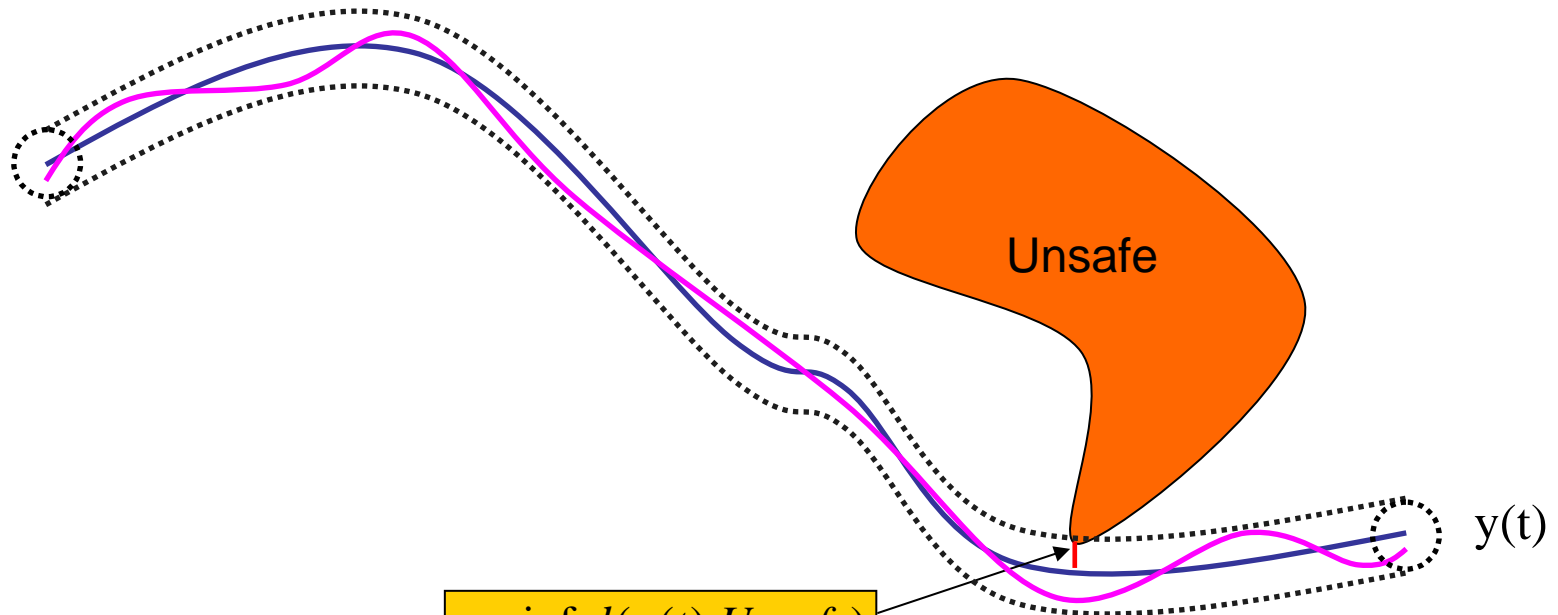
How robust is a test trajectory?

Consider the dynamical system:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) \\ y(t) &= g(x(t)) \end{aligned} \quad \mathcal{S} \xrightarrow{y(t)}$$

Fainekos, Girard, Pappas:
Temporal Logic Verification Using Simulation, FORMATS 2006

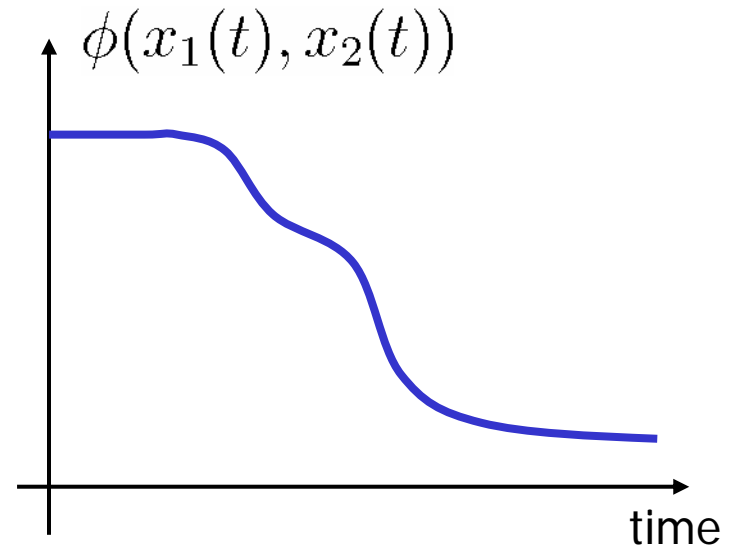
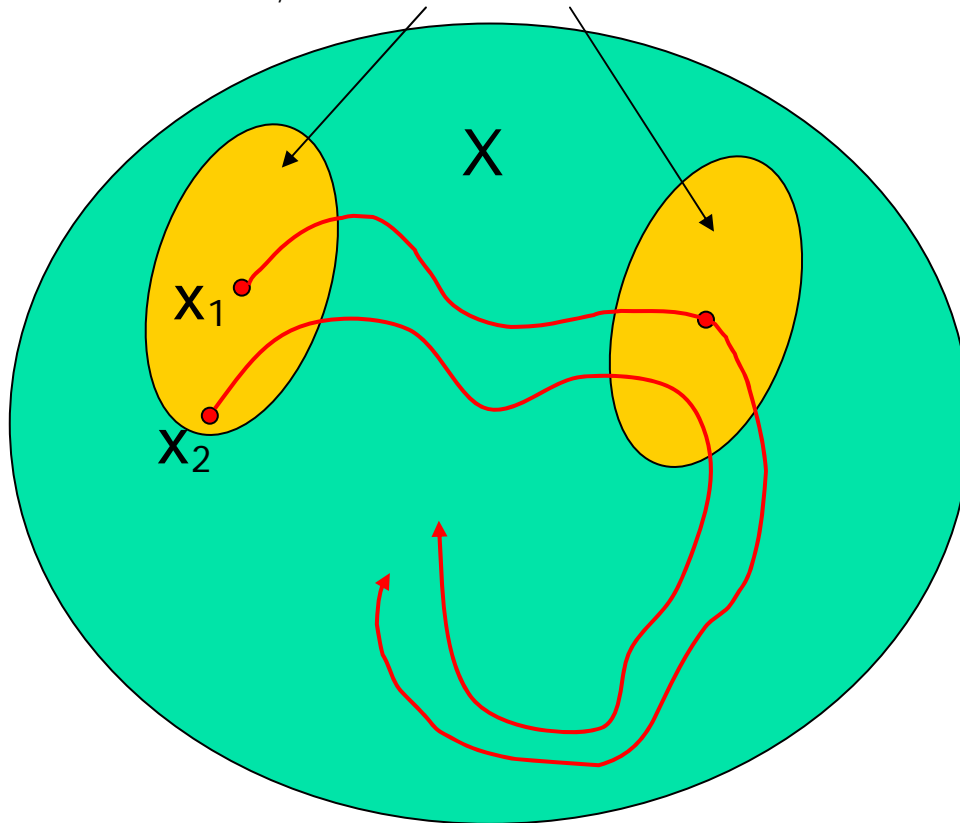
$$x \in \mathcal{R}^n, x(0) \in I, y \in \mathcal{R}^p$$



$$\epsilon = \inf_{t \geq 0} d(y(t), \text{Unsafe})$$

Bisimulation functions

$$B_\phi(x_1, \varepsilon) = \{x_2 \in X : \phi(x_1, x_2) \leq \varepsilon\}$$



A bisimulation function is **nonincreasing** along any two trajectories of the system.



Bisimulation functions

- *The function $\phi : X \times X \rightarrow \mathbb{R}_+$ is a bisimulation function if the following properties hold*
 - for all $x_1, x_2 \in X$ it is $\|g(x_1) - g(x_2)\|^2 \leq \phi(x_1, x_2)$
 - for all $x_1, x_2 \in X$ it is

$$\frac{\partial \phi(x_1, x_2)}{\partial x_1} f(x_1) + \frac{\partial \phi(x_1, x_2)}{\partial x_2} f(x_2) \leq 0$$

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

Connection to *contraction metrics*.

W. Lohmiller, J.J.E. Slotine, On contraction analysis for nonlinear systems, *Automatica*, 35, pp 683-696, 1998.



Systems with affine dynamics

In the case that the dynamics is affine,

$$f(x) = Ax + b,$$
$$x \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^{n \times 1},$$

we can propose that the bisimulation function assumes the form

$$\phi(x_1, x_2) = (x_1 - x_2)^T M (x_1 - x_2),$$

where M is a symmetric matrix.



Lyapunov equation

$$\begin{aligned} \phi(x_1, x_2) &\geq 0, \\ \frac{\partial \phi(x_1, x_2)}{\partial x_1} f(x_1) + \frac{\partial \phi(x_1, x_2)}{\partial x_2} f(x_2) &\leq 0. \end{aligned}$$



$$\begin{aligned} M &\geq 0, \\ A^T M + M A &\leq 0. \end{aligned}$$

Lyapunov equation ... always has a solution for stable A .

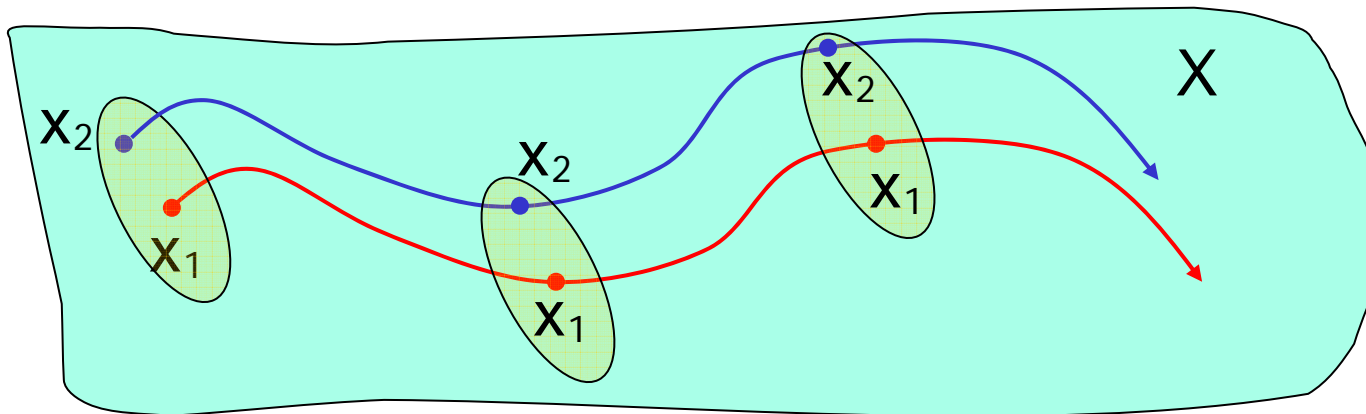
Invariance property

- The bisimulation function

$$\phi(x_1, x_2) = (x_1 - x_2)^T M (x_1 - x_2)$$

is a metric.

- The invariance property implies that the gap between two trajectories is bounded by the ellipsoids of the level sets.

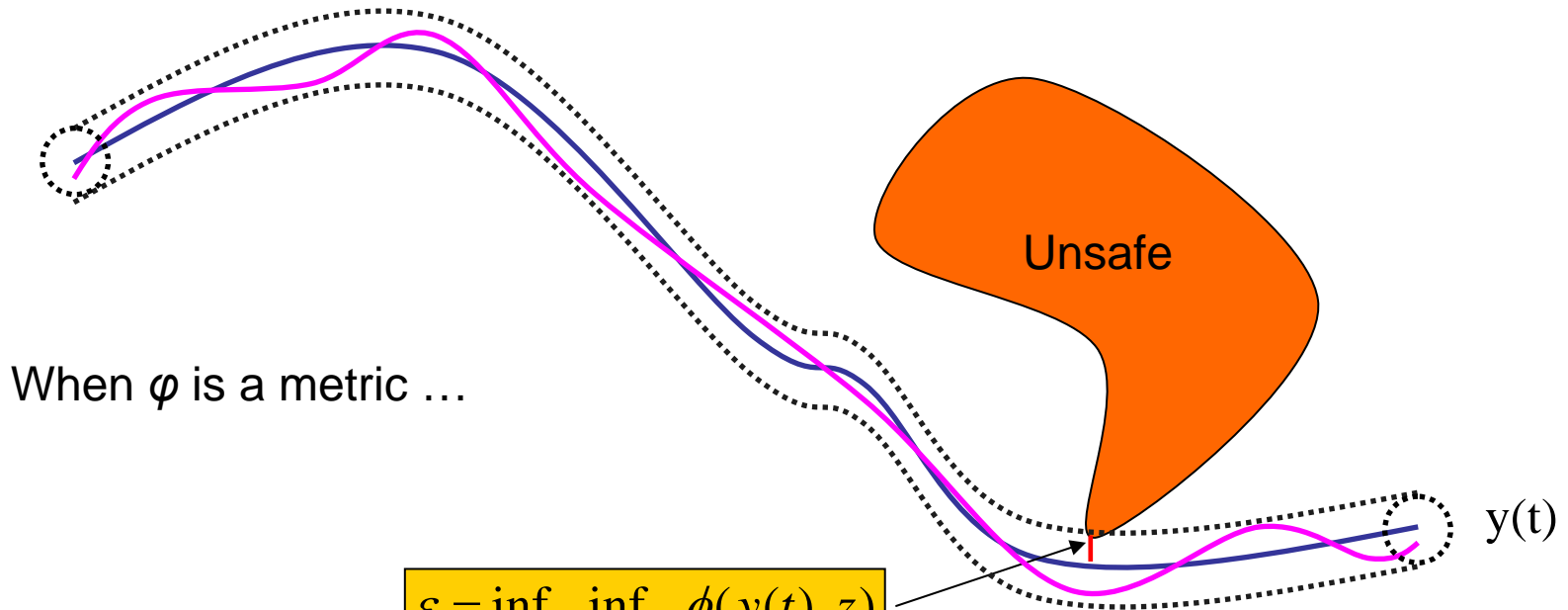


How robust is a test trajectory?

Consider the dynamical system:

$$\begin{array}{l} \dot{x}(t) = f(x(t)) \\ y(t) = g(x(t)) \end{array} \quad \mathcal{S} \quad \xrightarrow{y(t)}$$

$$x \in \mathcal{R}^n, x(0) \in I, y \in \mathcal{R}^p$$



$$\varepsilon = \inf_{t \geq 0} \inf_{z \in \text{Unsafe}} \phi(y(t), z)$$

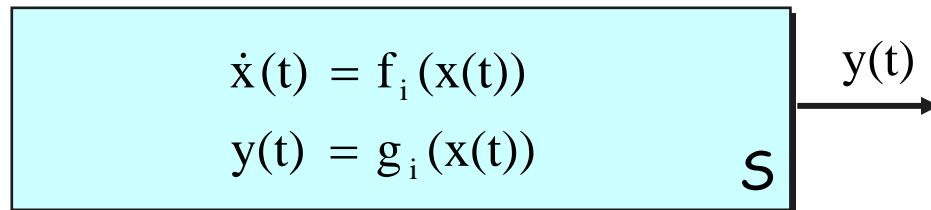
How robust is a hybrid test trajectory?

A hybrid automaton is a tuple

$$H = (X, L, E, f, g, U, \text{Inv}, \text{Init}, G, R, \text{Unsafe})$$

where

- X is the continuous state space
- L is the set of control locations
- $E \subseteq L \times L$ is the set of control switches
- $\text{Inv} : L \rightarrow \mathcal{P}(X)$ assigns an invariant set to each location
- $\text{Out} : L \times Z \rightarrow V$ is the control input for S'
- $\text{Init} \subseteq X_0 \times L$ is the set of initial conditions
- $G : E \rightarrow \mathcal{P}(\text{bd}(\text{Inv}(l)))$ is the guard condition that enables transition $e=(l,l') \in E$
- $R : E \rightarrow \text{Inv}(l')$ is the reset map for the transition $e=(l,l') \in E$
- $\text{Unsafe} \subseteq X_0 \times L$ is the unsafe region
- f, g



$$x \in \mathcal{R}^n, x(0) \in I, y \in \mathcal{R}^p$$



One step of the algorithm

- Suppose there is a time lag ϵ such that $\xi(\tau + \epsilon, x_0) \notin \text{Inv}(l_i)$.
- We define

$$d_{out} := \inf_{y \in g_1} \phi(\xi(\tau + \epsilon, x_0), y),$$

$$d_i := \inf_{0 \leq t \leq \tau + \epsilon} \inf_{y \in g_i^{act}} \phi(\xi(t, x_0), y), i = 2, 3, \dots, n,$$

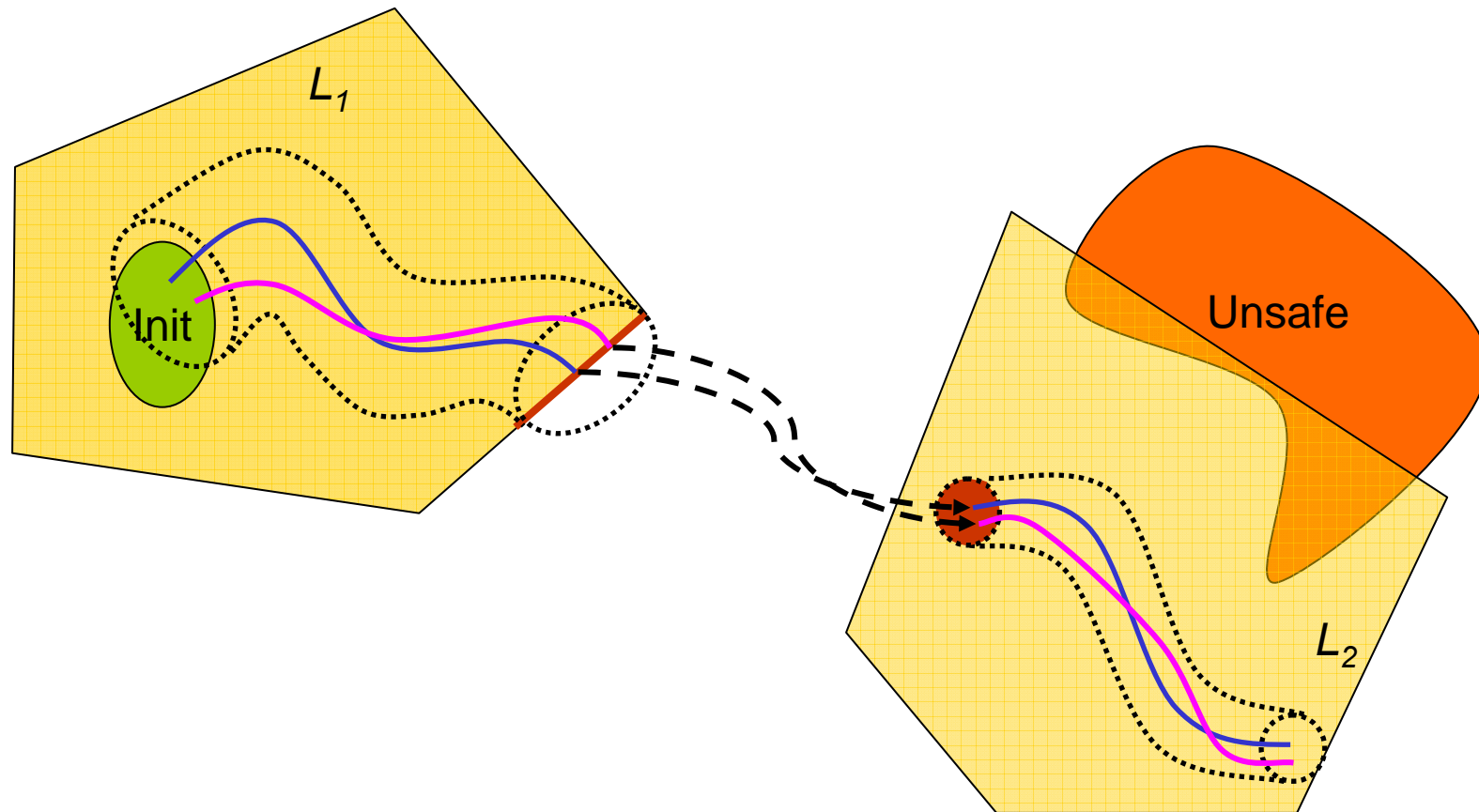
$$d_{unsafe} := \inf_{0 \leq t \leq \tau + \epsilon} \inf_{y \in \text{Inv}(l) \cap \text{Unsafe}^{act}} \phi(\xi(t, x_0), y),$$

$$d_{min} := \min\{d_{out}, d_{unsafe}, d_2, d_3, \dots, d_n\},$$

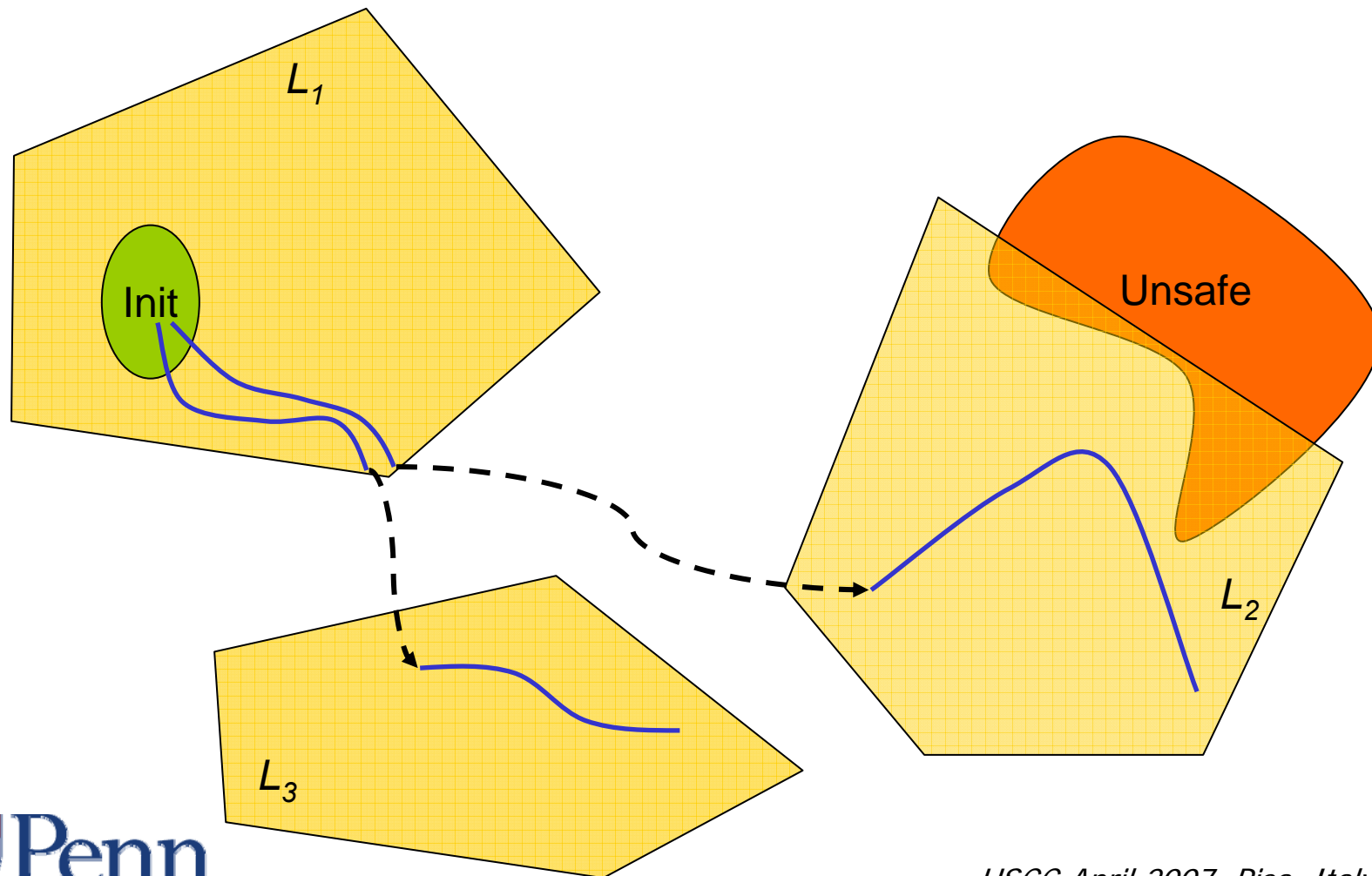
$$\hat{\epsilon} := \inf \{\delta > 0 \mid B_\phi(\xi(\tau - \delta, x_0), d_{min}) \subset \underline{\text{Inv}}(l)\}.$$

What about neighboring trajectories?

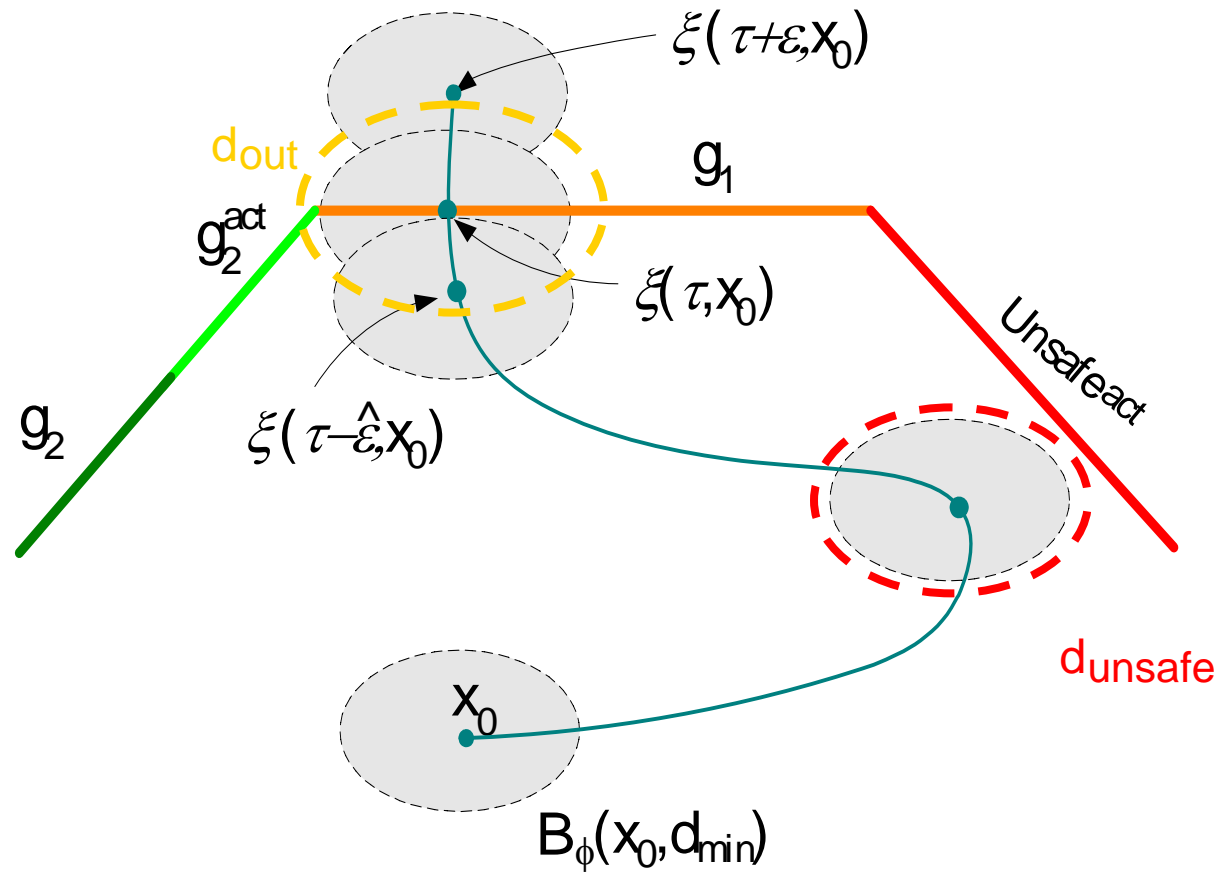
Bisimulation metric takes care of that ...



Robustness implies same qualitative behavior



We have timing guarantees, too.



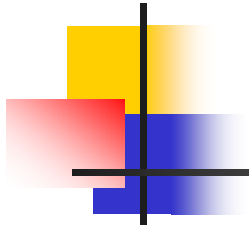


Main result: Loop Invariance

For any $x'_0 \in B_\phi(x_0, d_{\min}) \cap Inv(l_i)$:

- the state trajectory $\xi(t, x'_0)$ exits $Inv(l_i)$ through transition e_1 ,
- at time $t \in [\tau - \hat{\varepsilon}, \tau + \varepsilon]$,
- and is safe at least until it exits location l_i .

■ Thus, a guarantee on the **qualitative behavior** and **timing**.

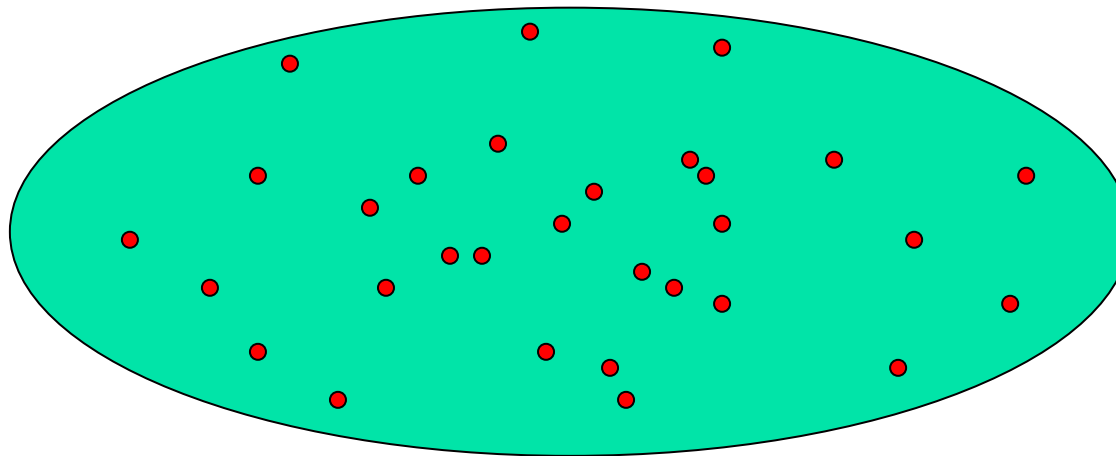


A Testing algorithm for Hybrid Automata



Covering of the parameter space

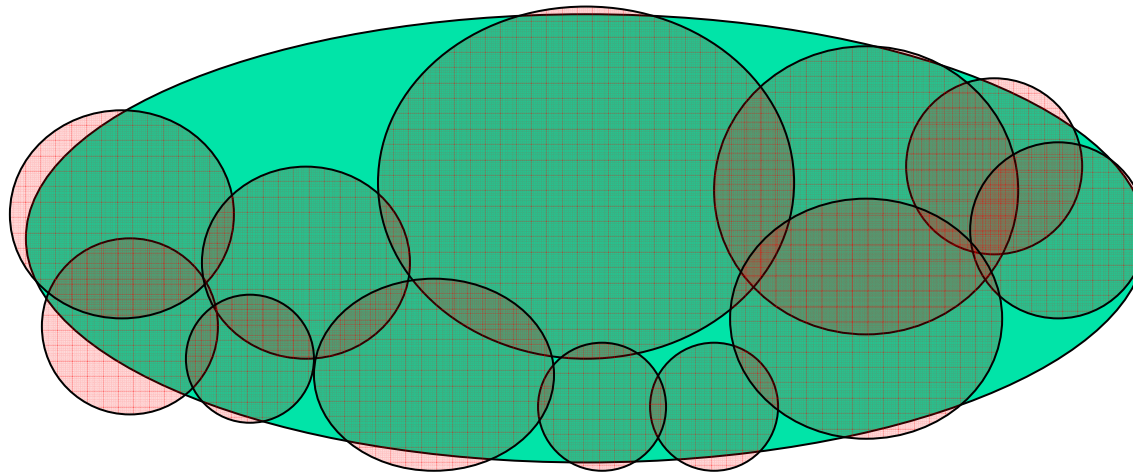
- It is impossible to cover an uncountable testing parameter space with points.



Initial Conditions

Covering with robust tests

- Each test represents a (nonzero measure) neighborhood of testing parameters.



Initial Conditions

- Parameters that lead to tests with the same qualitative properties are grouped together.



Covering with robust tests

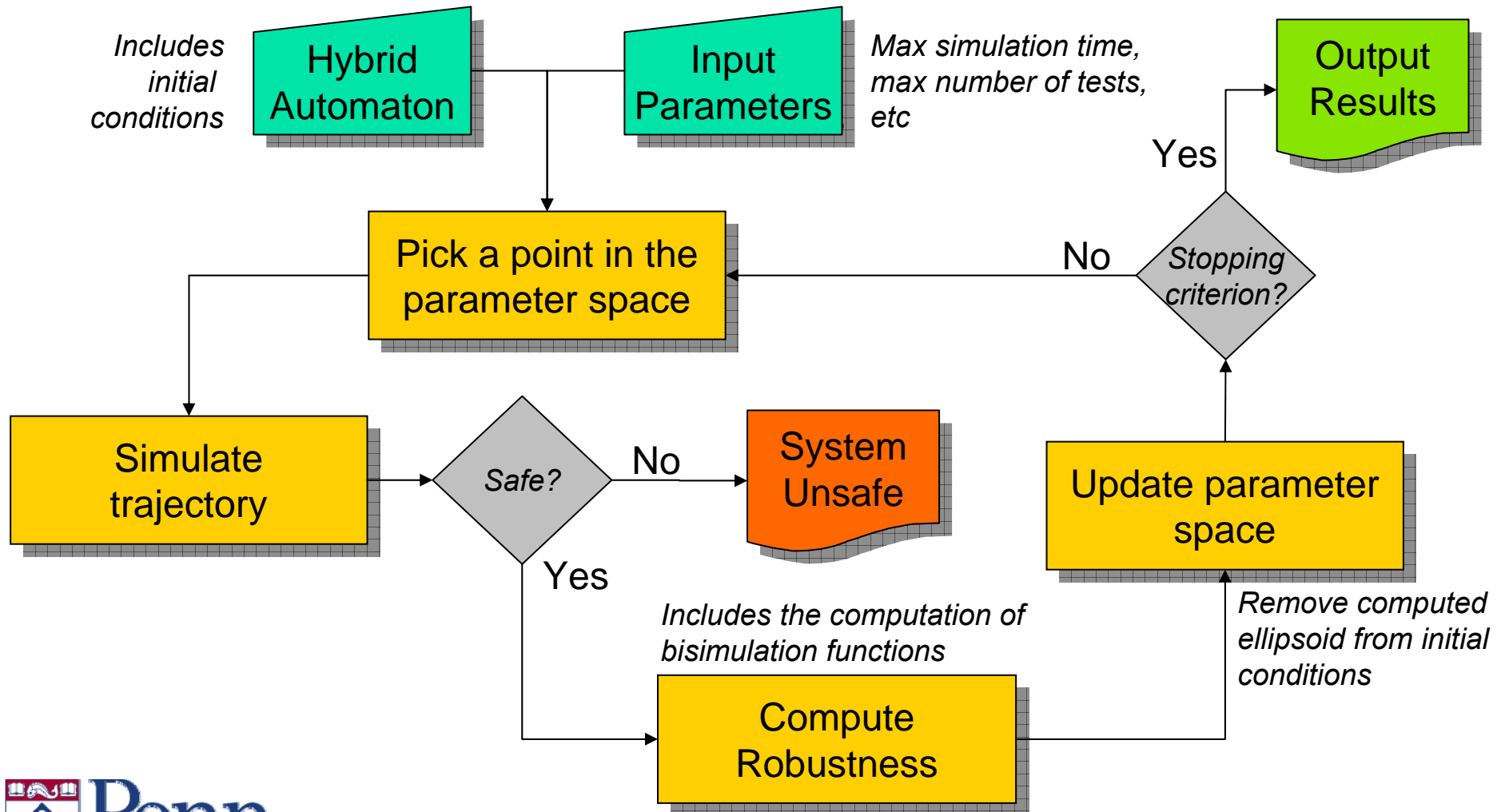
- Each test represents a (nonzero measure) neighborhood of testing parameters.

Finite covering is possible!

Only if the system is robust.

- Parameters that lead to tests with the same qualitative properties are grouped together.

Overview of algorithm





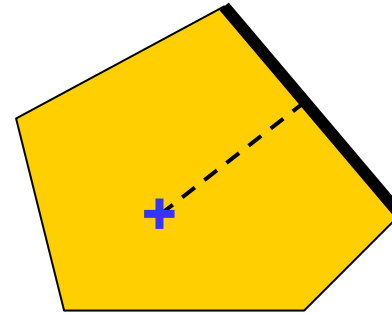
Coverage strategies

- **Randomized strategy**: easy to implement, almost impossible to get 100% coverage.
- **Grid based strategy**: easy to implement, suffers from curse of dimensionality.
- **Minimal dispersal**: based on partitioning the parameter space with **weighted** Voronoi partitions.

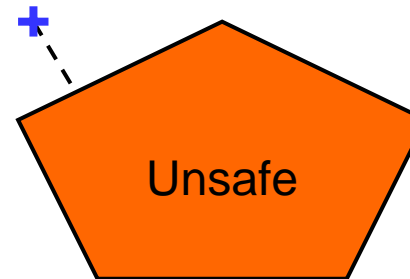
Computing distances

linear projections

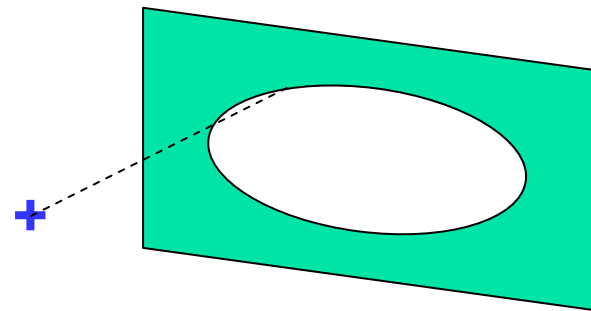
(least squares when we consider the location dynamics)

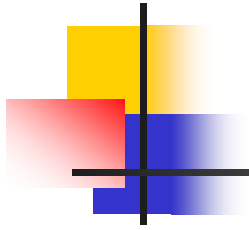


quadratic programming



semidefinite programming





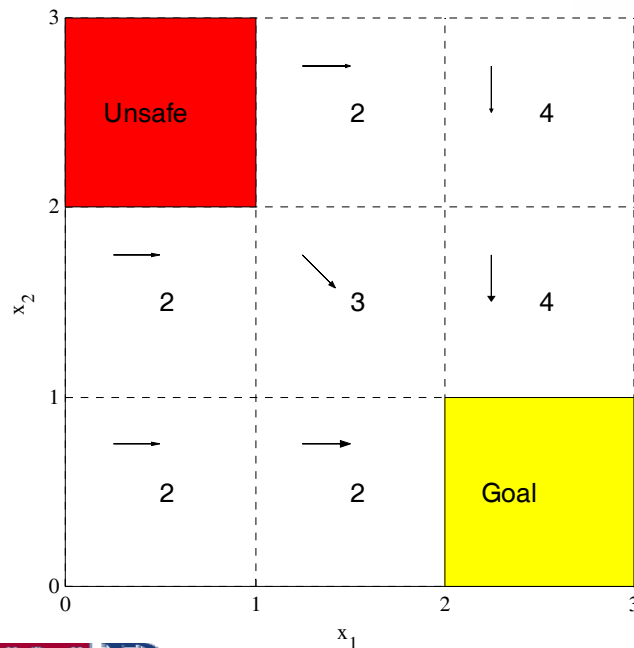
Some Examples

Navigation benchmark

A. Fehnker, F. Ivancic, Benchmarks for hybrid verifications, in *Hybrid Systems: Computation and Control 2004*, pp. 326-341.

Hybrid system with 3x3 locations.

The dynamics:



$$x = [x_1 \ x_2 \ v_1 \ v_2]^T, \dot{x} = Ax - Bu(i, j)$$

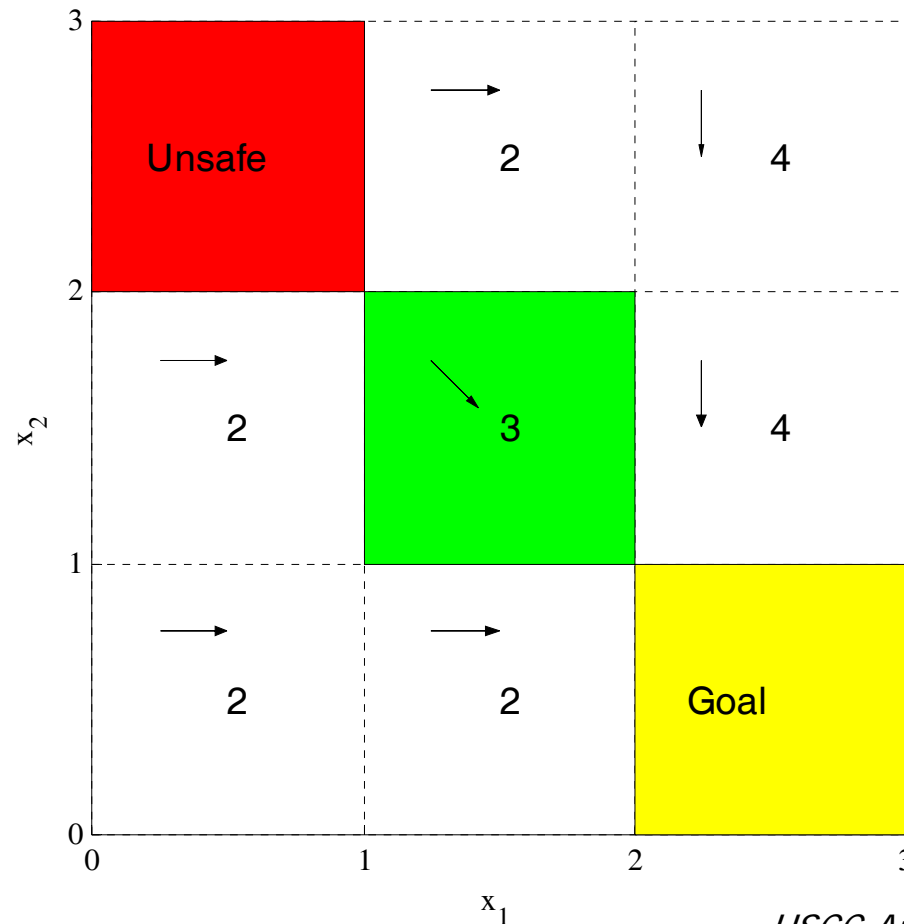
$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1.2 & 0.1 \\ 0 & 0 & 0.1 & -1.2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1.2 & 0.1 \\ 0.1 & -1.2 \end{bmatrix}$$

$$u(i, j) = [\sin(\pi C(i, j)/4) \ \cos(\pi C(i, j)/4)]^T$$

$$C_1 = \begin{bmatrix} U & 2 & 4 \\ 4 & 3 & 4 \\ 2 & 2 & G \end{bmatrix} \quad C_2 = \begin{bmatrix} 2 & 3 & 6 \\ 3 & 3 & G \\ 2 & 2 & U \end{bmatrix} \quad C_3 = \begin{bmatrix} U & 2 & 4 \\ 2 & 2 & 4 \\ 1 & 1 & G \end{bmatrix}$$

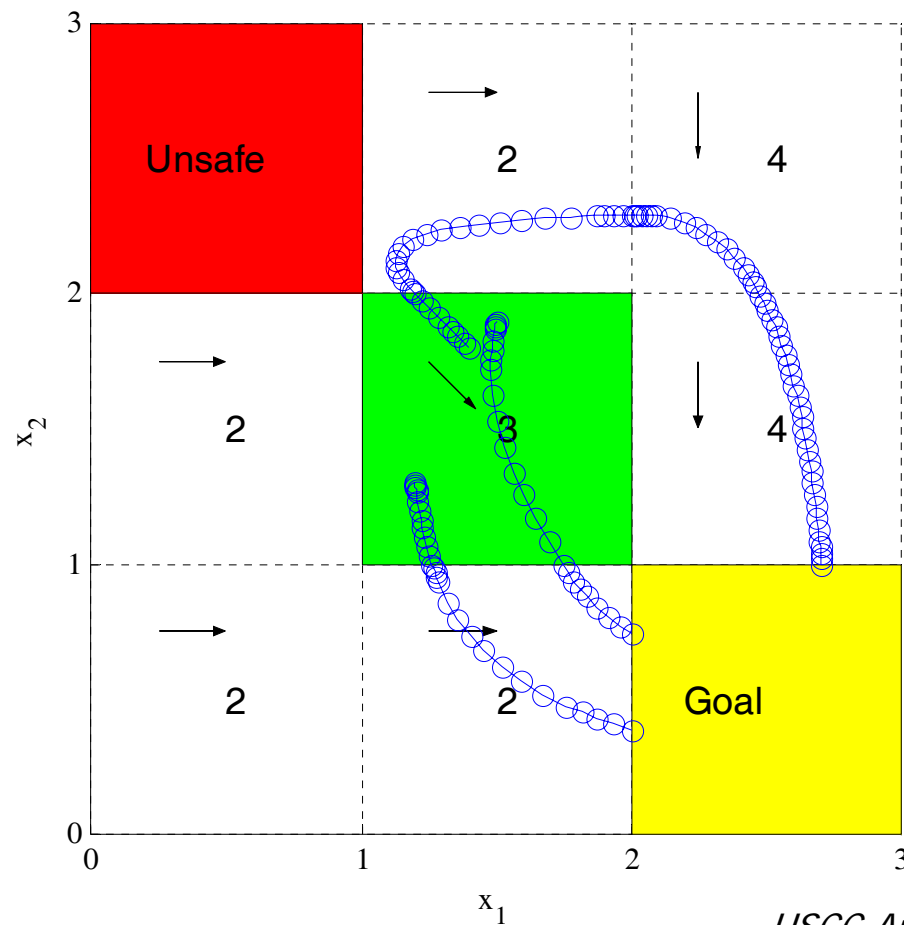
Navigation benchmark

A. Fehnker, F. Ivancic, Benchmarks for hybrid verifications, in *Hybrid Systems:Computation and Control 2004*, pp. 326-341.

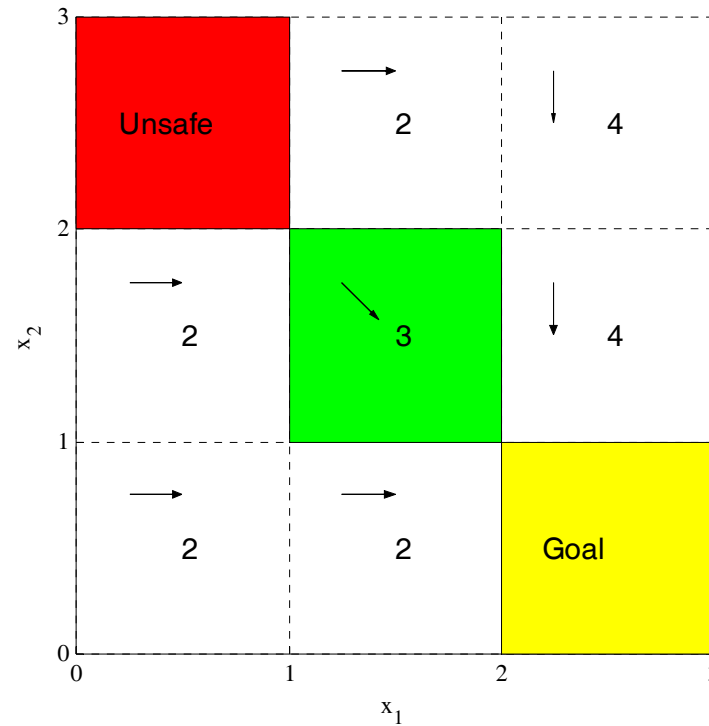


Navigation benchmark

A. Fehnker, F. Ivancic, Benchmarks for hybrid verifications, in *Hybrid Systems: Computation and Control 2004*, pp. 326-341.

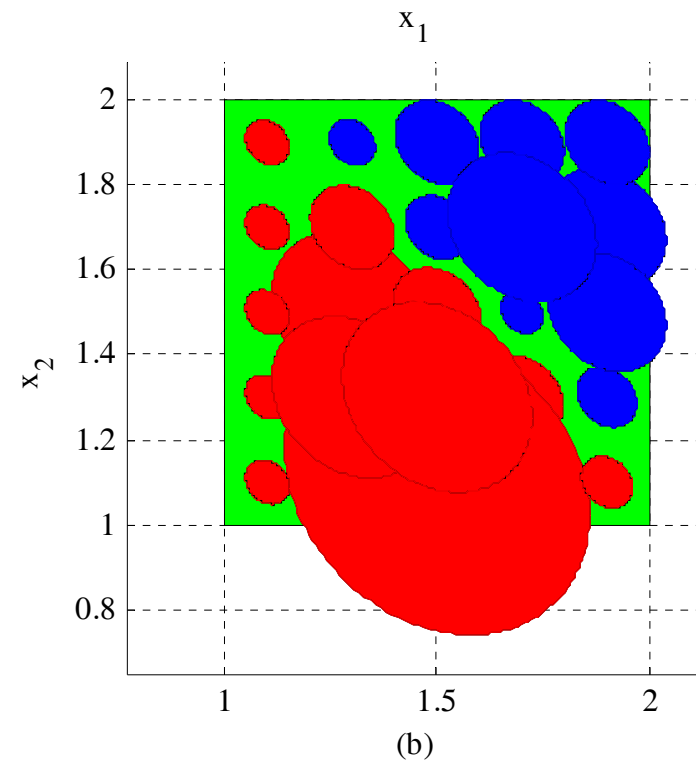
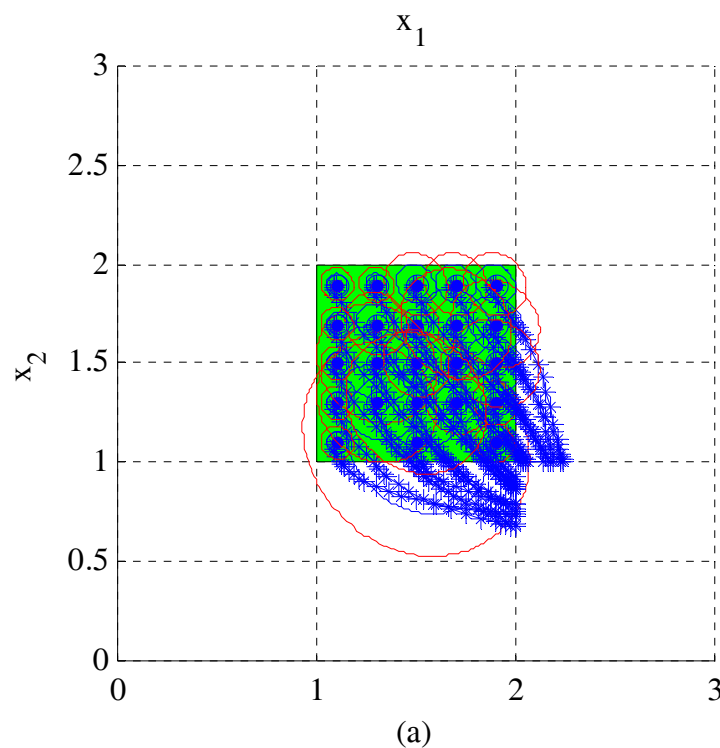


Navigation benchmark 1



$$\mathcal{X}_0 = [1, 2] \times [1, 2] \times \{-0.2\} \times \{0\}$$

Navigation benchmark 1

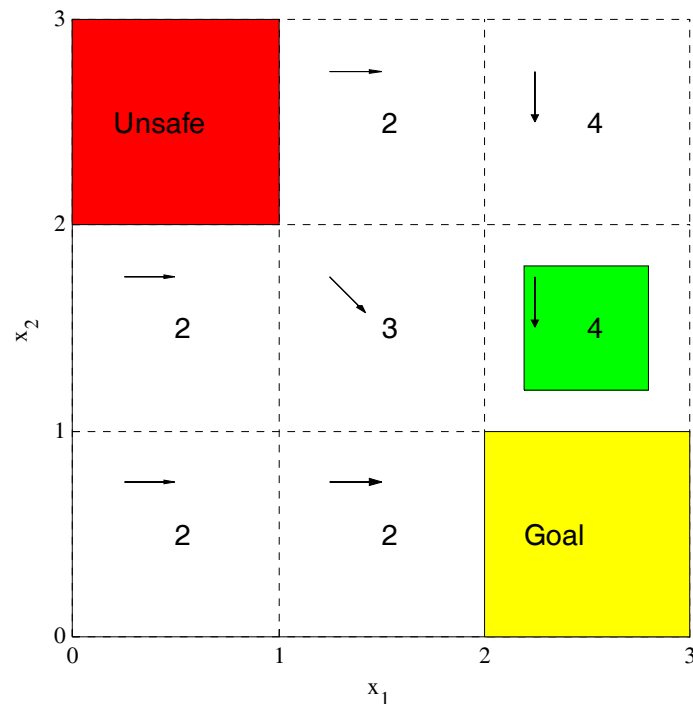


With 25 runs, we cover **>48%** of the initial set.

Notice that there is a clear **divide** in the initial set, due to different transitions.

Benchmark problem 2

A. Fehnker, F. Ivancic, Benchmarks for hybrid verifications, in *Hybrid Systems:Computation and Control 2004*, pp. 326-341.

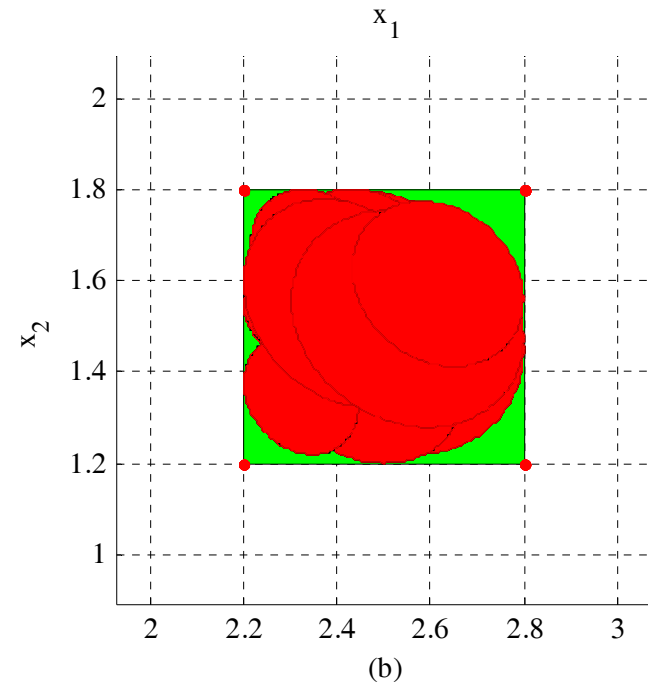
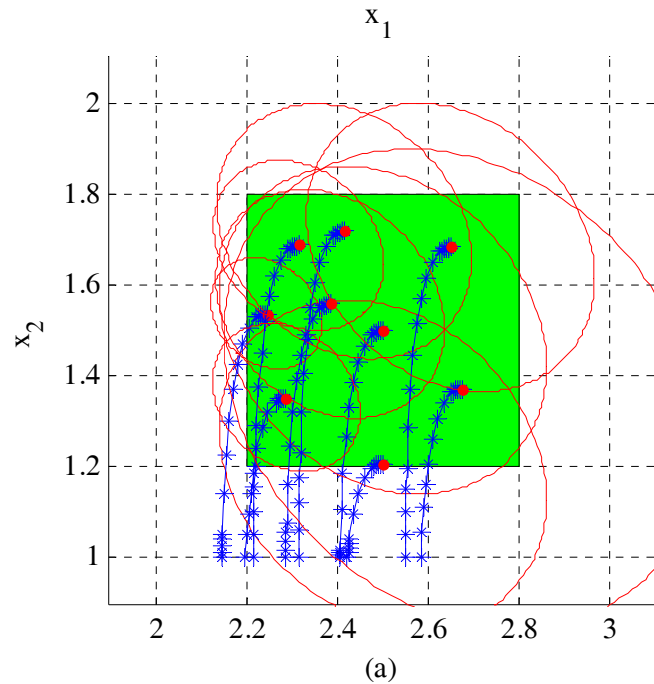


Initial condition:

$$[2.2, 2.8] \times [1.2, 1.8] \times \{-0.2\} \times \{0\}$$

Verified to be safe with CHARON

Benchmark problem 2

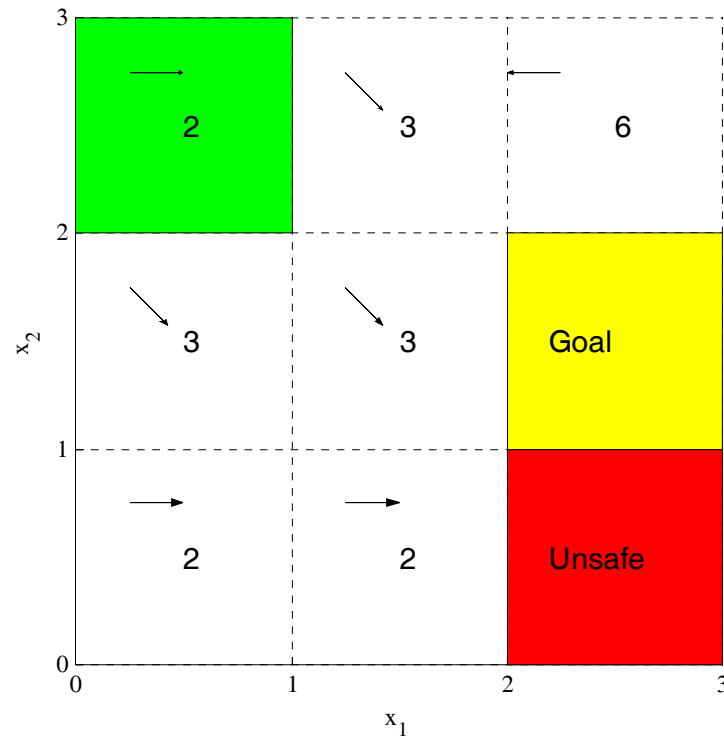


Safety verified after **9 tests!**

(All traces have the same qualitative behavior and the system is robust wrt to the unsafe set. Termination guaranteed similar to Girard & Pappas HSCC'06, Fainekos et al FORMATS 2006)

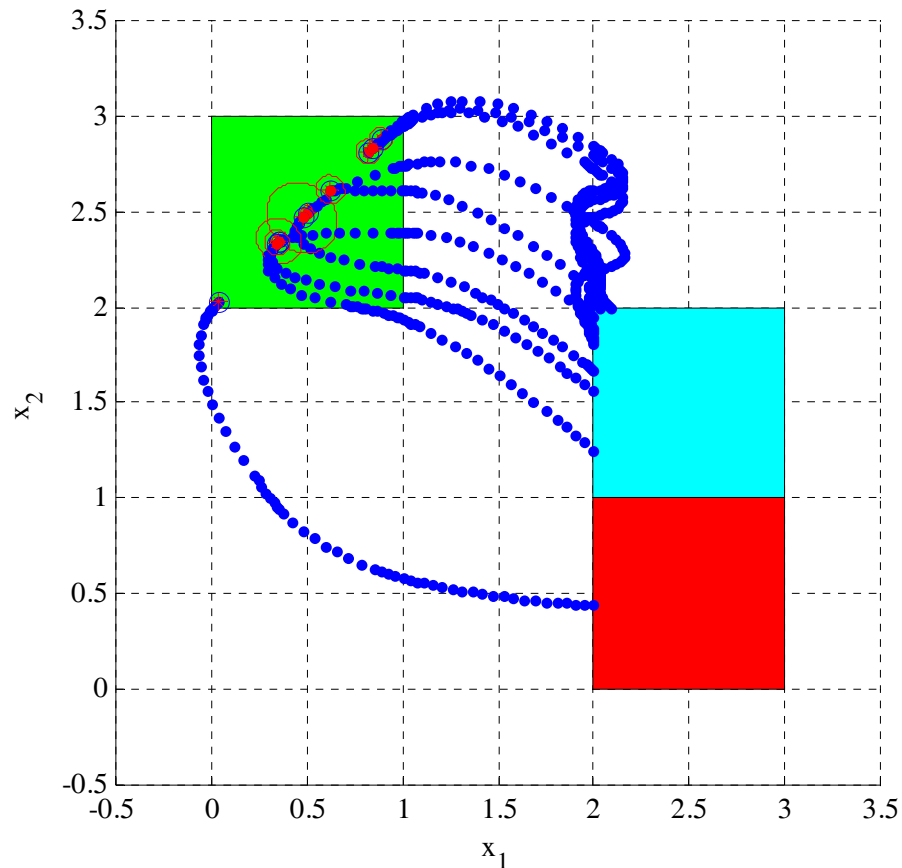
Numerically, we compute a coverage estimate of 72%.

Navigation benchmark 3

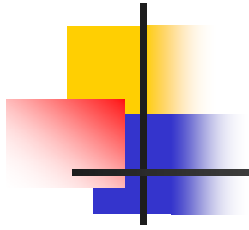


$$\mathcal{X}_0 = [0, 1] \times [2, 3] \times [-1, 1] \times [-1, 1]$$

Navigation benchmark 3



- Test generation using Voronoi with weights
- We verified unsafety with **10 tests**.



Conclusions and Discussion



Conclusions & Discussion

- We have introduced :
 - a notion of robustness for test trajectories of hybrid systems
 - An algorithm that computes confidence levels for hybrid systems
- A toolbox that helps the exploration of a hybrid system
 - Early stages of HS design
- The algorithm is **automatic** for hybrid systems with affine dynamics
- The framework can be effectively **parallelized**



Future Extensions

- Temporal logic testing of hybrid systems
 - Fainekos, Girard, Pappas: *Temporal Logic Verification Using Simulation*, in FORMATS 2006
- Probabilistic testing
 - Julius: *Approximate abstraction of stochastic hybrid automata*, in HSCC 2006
- Nonlinear systems
 - Girard, Pappas: *Approximate bisimulations for nonlinear dynamical systems*, in CDC 2005
- Hybrid Systems with bounded input (noise)
 - Girard, Pappas: *Verification using simulation*, in HSCC 2006

