

Robustness of Temporal Logic Specifications

(and an application to verification using simulation)



Georgios E. Fainekos

joint work with Antoine Girard and George J. Pappas

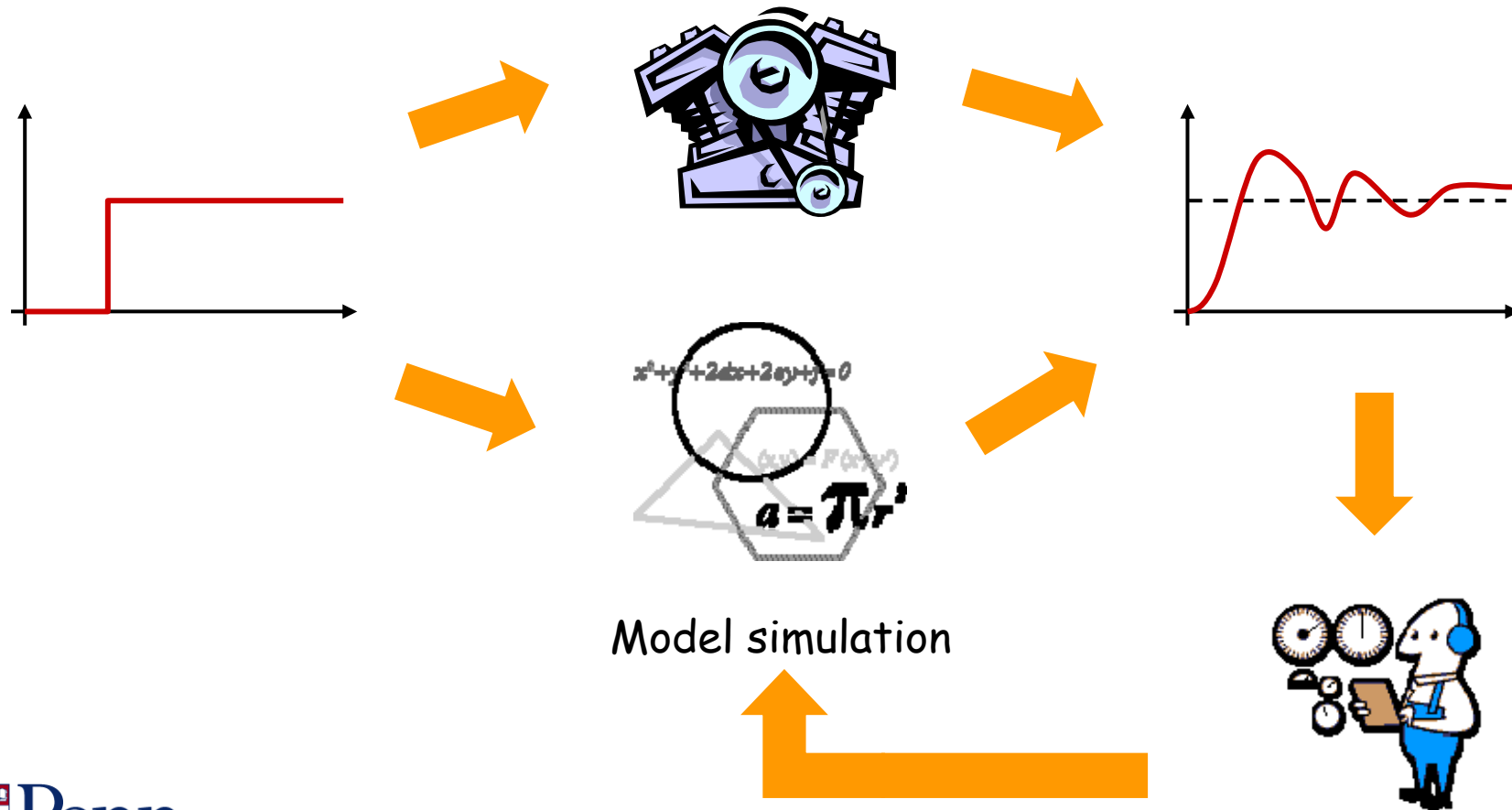
GRASP Laboratory
Department of Computer and Information Science
University of Pennsylvania

✉ fainekos @ seas.upenn.edu

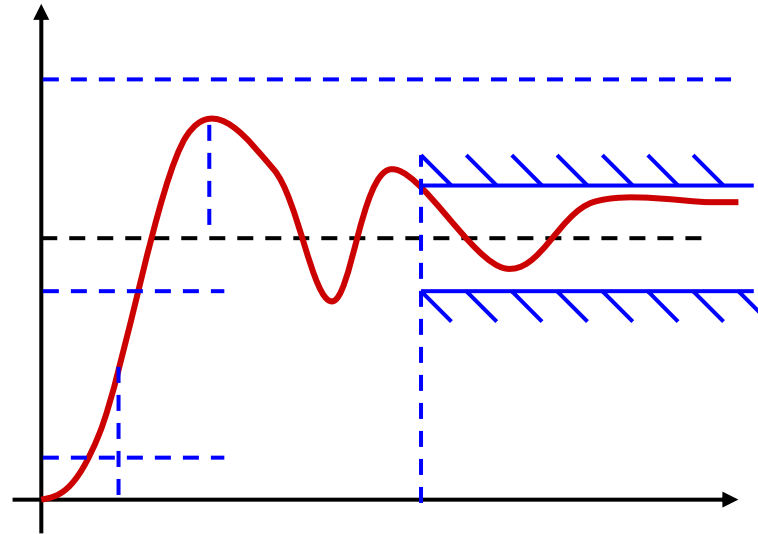
🌐 <http://www.seas.upenn.edu/~fainekos/>

Motivation I - a study of transient dynamics

Black-box controller tuning



Motivation I - a study of transient dynamics

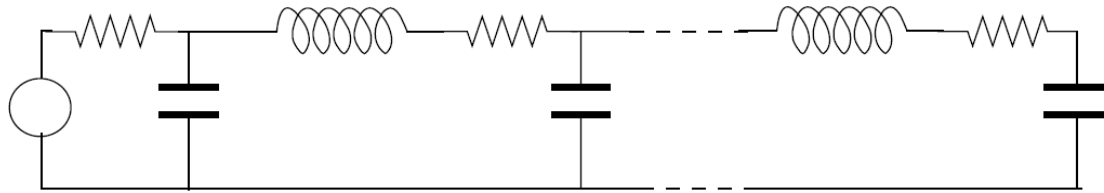


Desired Performance Characteristics

1. (percent) Overshoot
2. Rise time
3. Delay time
4. Settling time
5. Constraints on input/states
6. Response sensitivity

Use Linear or Metric
Temporal Logic

Motivation I - a study of transient dynamics*



System:

$$\dot{x}(t) = Ax(t) + bU_{in}(t)$$

$$U_{out}(t) = Cx(t)$$

Step input ($t > 0$):

$$U_{in}(t) = 1$$

Steady state at $t = 0^-$:

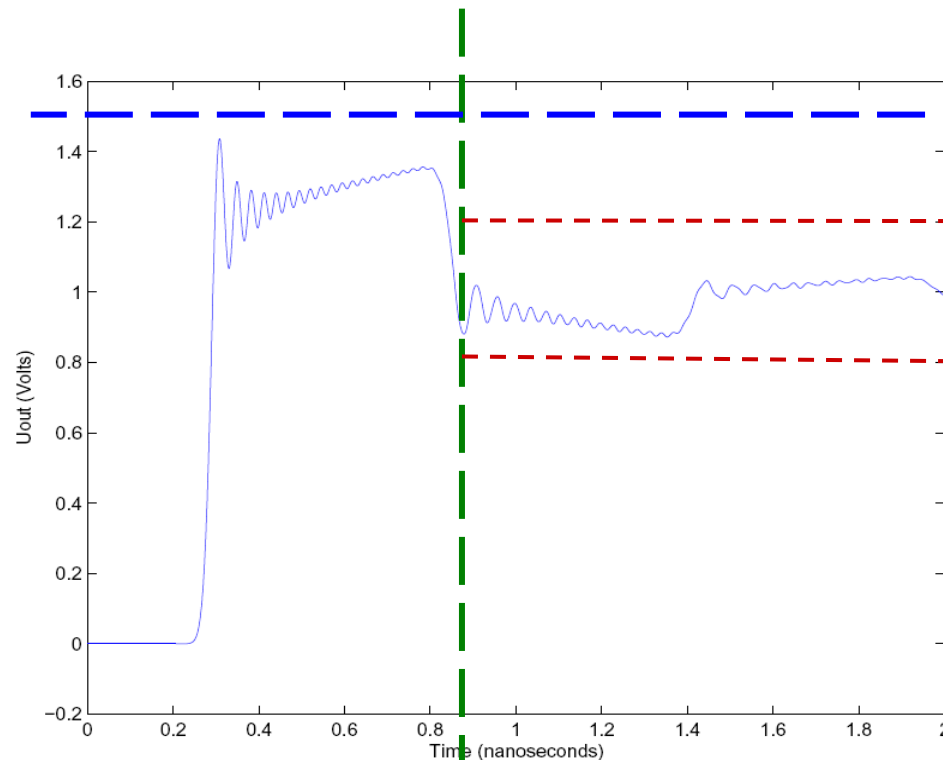
$$x(0) = -A^{-1}bU_{in}(0)$$

Property:

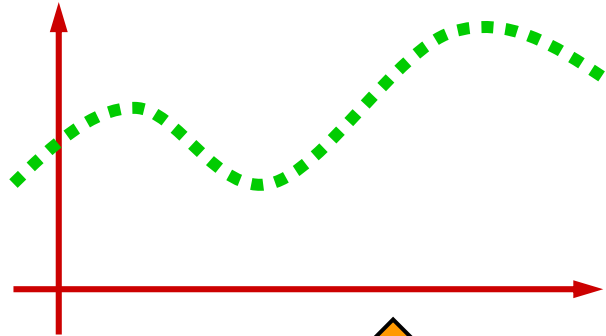
$$\phi = \square\pi_1 \wedge \diamond_{[0,T]}\square\pi_2$$

$$\mathcal{O}(\pi_1) = [-\theta, \theta]$$

$$\mathcal{O}(\pi_2) = [0.8, 1.2]$$



Boolean Problem Formulation



LTL / MTL
 $\varphi = \square((x \leq -10) \rightarrow \diamond_{\leq 2}(x \geq 10))$

Boolean
Tester

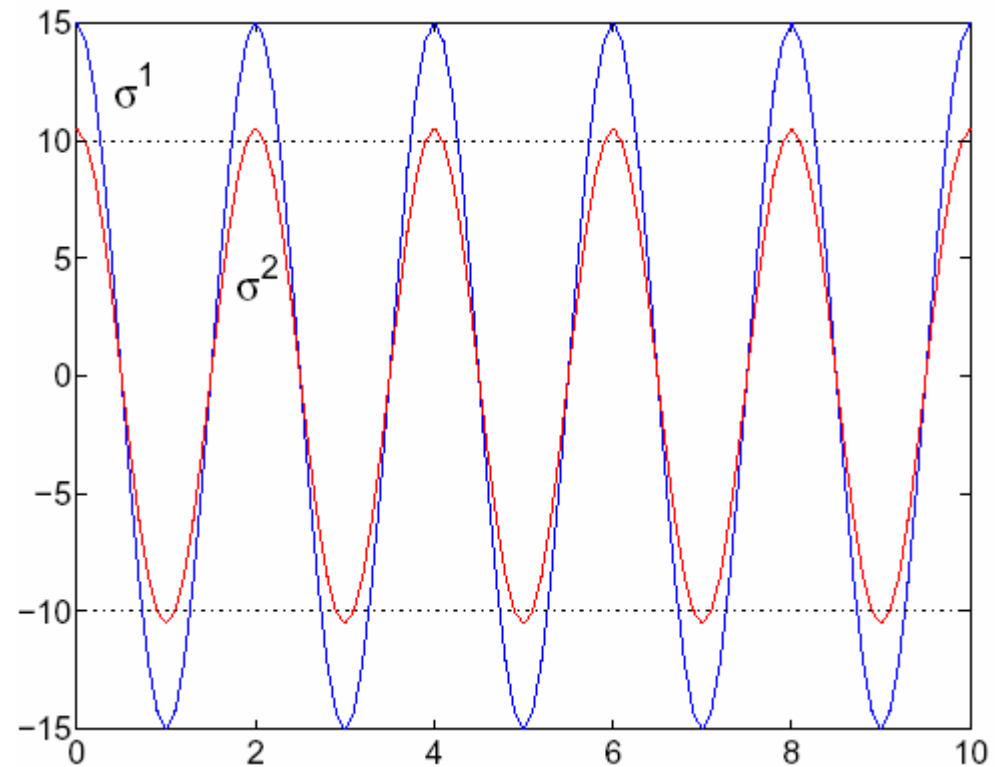
Truth Value
{0,1}

[Maler and Nickovic '04]
[Thati and Rosu '04]
[Rosu and Havelund '05]
[Geilen '01]
many others ...

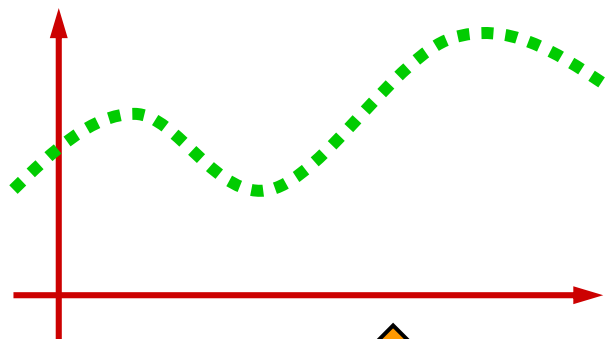
But the Boolean truth value is not enough ...

MTL Spec:

$$\square((x \leq -10) \rightarrow \diamond_{\leq 2}(x \geq 10))$$



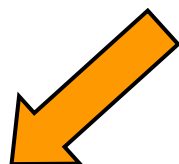
Problem formulation



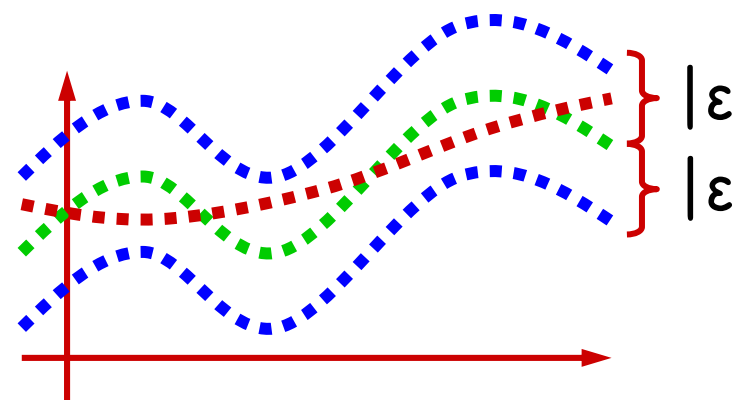
LTL / MTL
 $\varphi = \square((x \leq -10) \rightarrow \diamond_{\leq 2}(x \geq 10))$



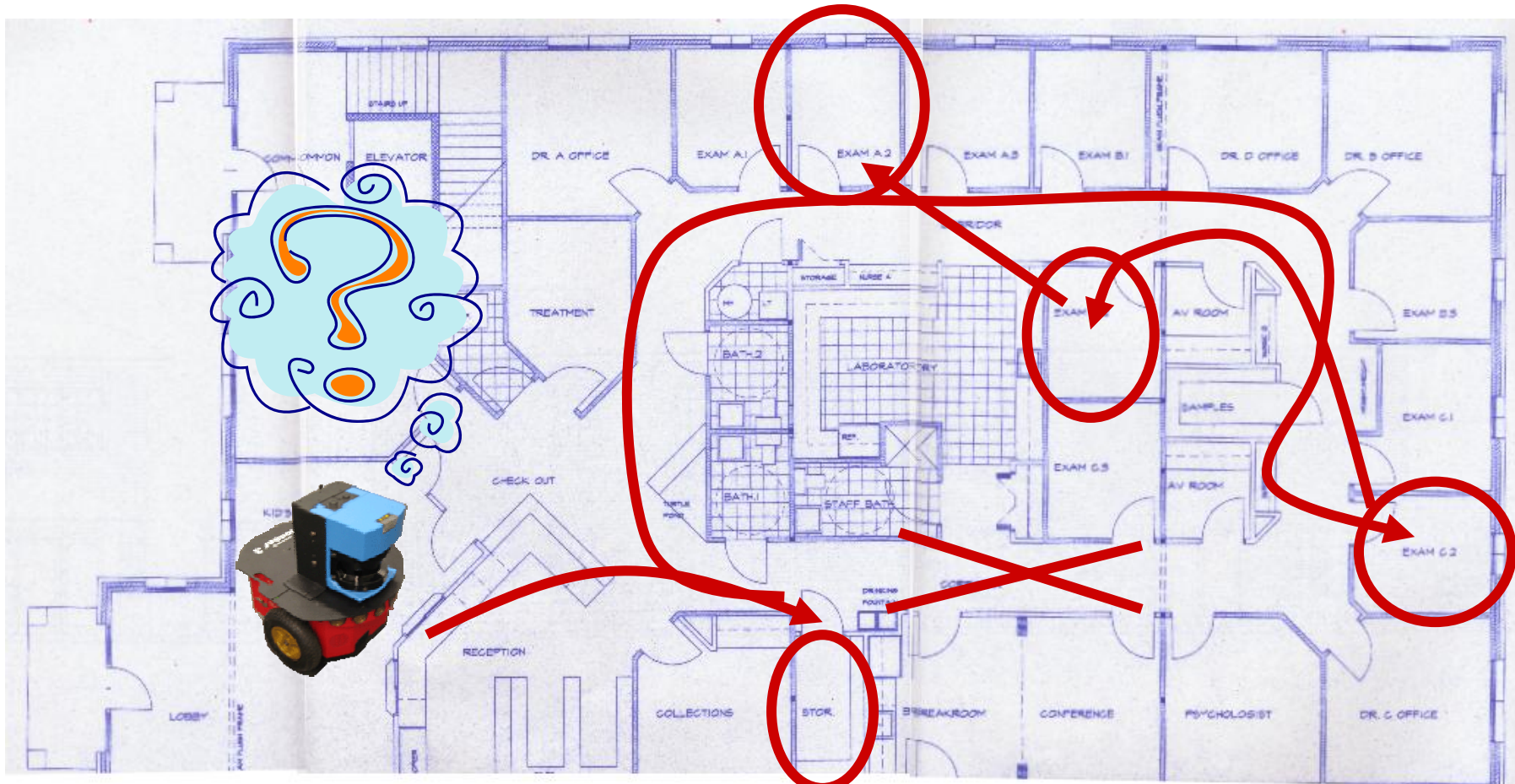
Monitor/Tester



Robustness parameter
 $\varepsilon \in \mathbb{R} \cup \{\pm\infty\}$



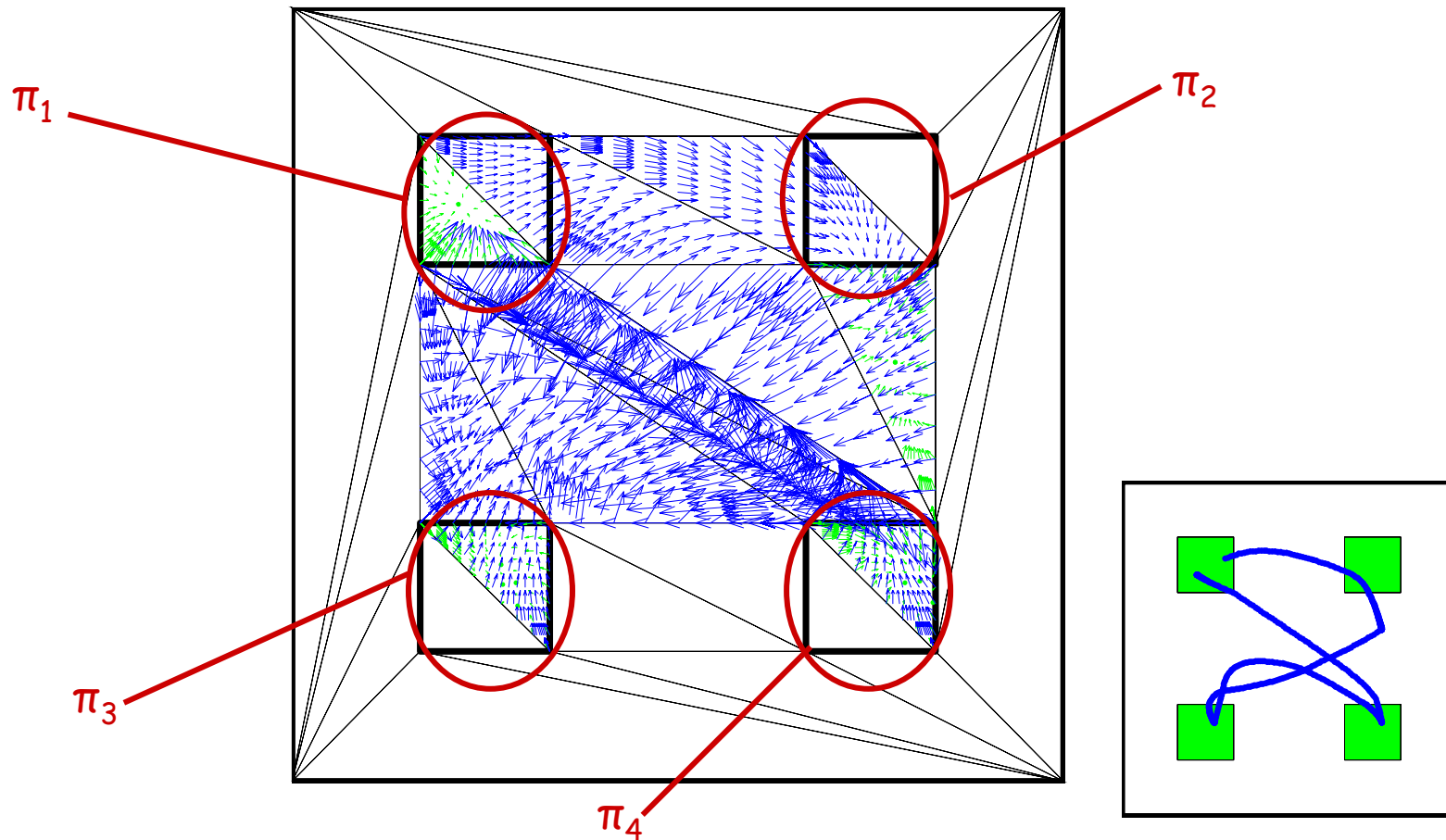
Motivation II - Humans & Robots in the near future



$$\varphi = \diamond (stor \wedge (\neg corr_1) U (exam_{c2} \wedge \wedge \diamond exam_{a2} \wedge \diamond exam_{b2}))$$

Motivation II - Hybrid Automaton

$$\diamond(\pi_2 \wedge \diamond(\pi_3 \wedge \diamond(\pi_4 \wedge \neg(\pi_3 \vee \pi_4) U \pi_1)))$$



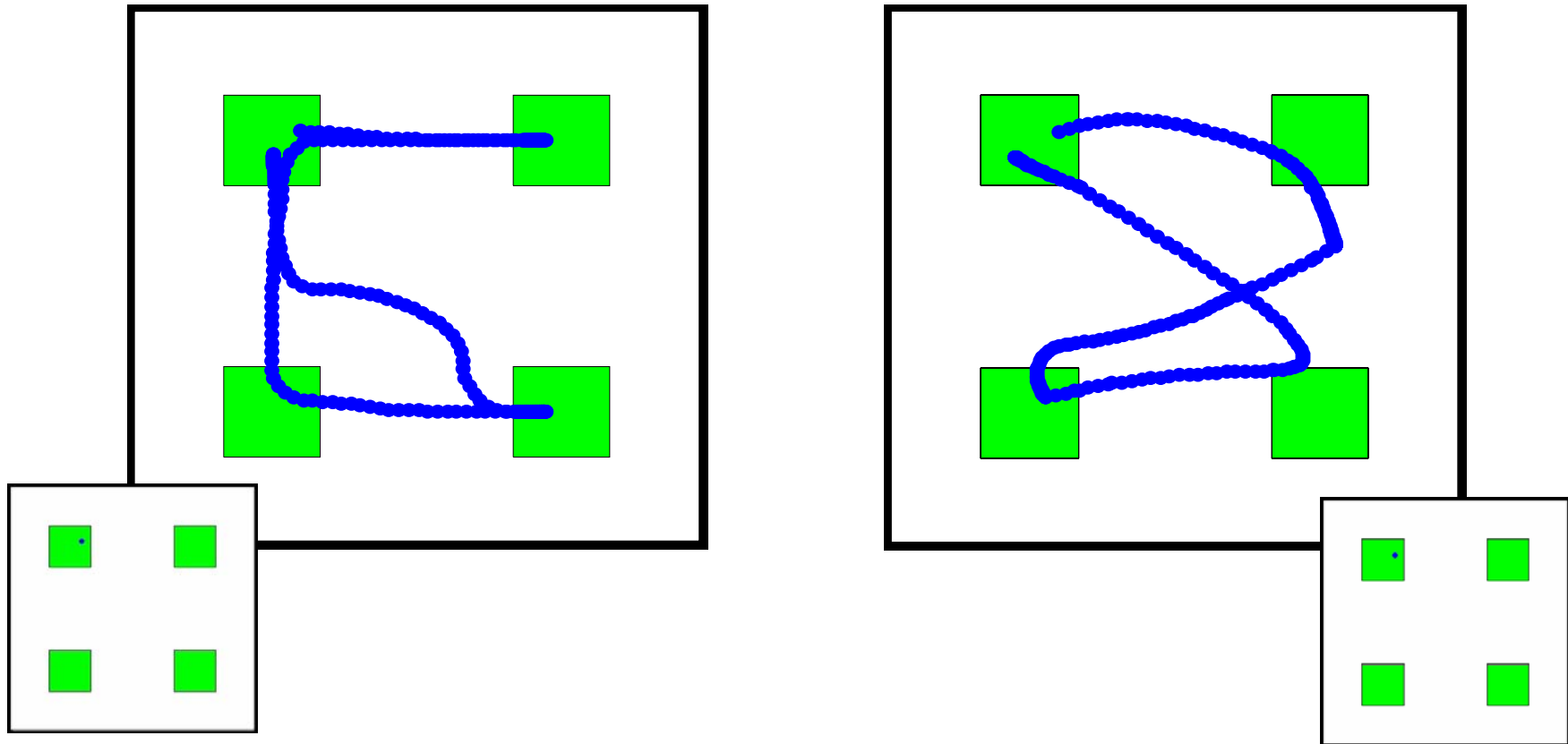
Fainekos, Kress-Gazit, Pappas, *Temporal Logic Motion Planning for Mobile Robots*, ICRA 2005
Fainekos, Kress-Gazit, Pappas, *Hybrid Controllers for Path Planning*, CDC 2005



Motivation I - LTL to motion planning

How do we quantify which trajectory is more robust ???

$$\diamond(\pi_2 \wedge \diamond(\pi_3 \wedge \diamond(\pi_4 \wedge \neg(\pi_3 \vee \pi_4) U \pi_1)))$$



Motivation/
Problem Formulation



LTL/MTL on traces
over metric spaces



Verification using
simulation



Roadmap

Robust Semantics



Computation of
robustness estimate

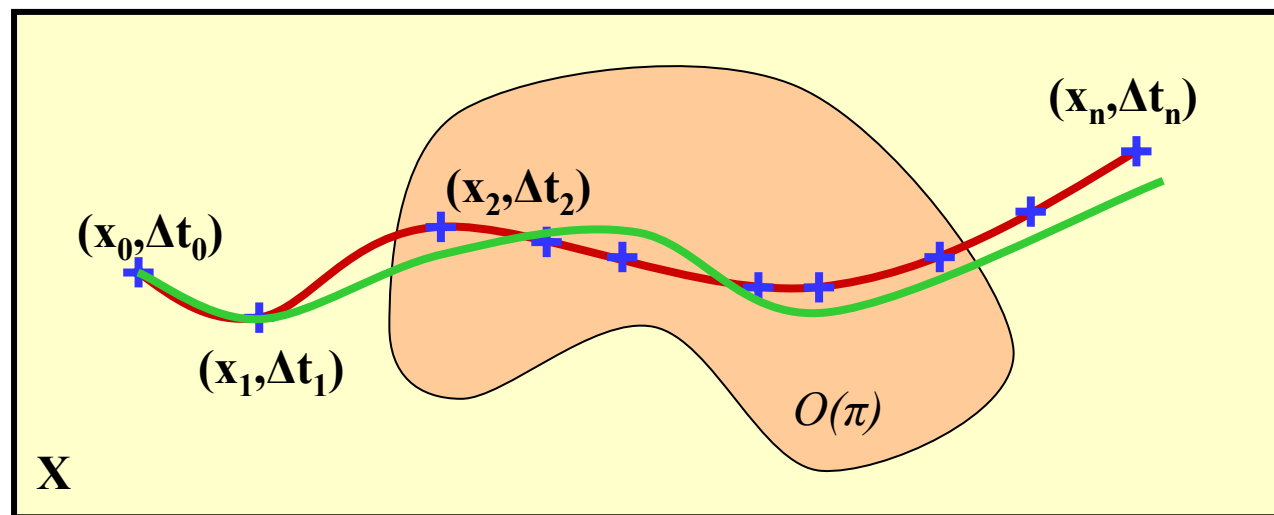
Part I: Robustness of Temporal Logic Specifications



(Finite) Timed State Sequences (TSS)

A timed state sequence T is a tuple (σ, τ, O) where

- σ is a sequence of states x_0, x_1, \dots, x_n
- τ is a sequence of time periods $\Delta t_0, \Delta t_1, \dots, \Delta t_n$
- $O: AP \rightarrow 2^X$ is a predicate mapping



Metric Temporal Logic (MTL)

Syntax:

$$\phi ::= \top \mid \pi \mid \neg\phi_1 \mid \phi_1 \vee \phi_2 \mid \bigcirc_I \phi_1 \mid \phi_1 \mathcal{U}_I \phi_2$$

I can be of any bounded or unbounded interval of \mathbb{Q}^+ .

i.e. $I = [0, +\infty)$, $I = [2.5, 9.8]$

Boolean Semantics:

$$\langle\langle \cdot \rangle\rangle : \Phi_{\text{MTL}} \times \text{TS} \rightarrow \{\top, \perp\}$$

$$\langle\langle v \rangle\rangle(\mathcal{T}) := v$$

$$\langle\langle \pi \rangle\rangle(\mathcal{T}) := \sigma_0 \in \mathcal{O}(\pi)$$

$$\langle\langle \neg\psi \rangle\rangle(\mathcal{T}) := \neg\langle\langle \psi \rangle\rangle(\mathcal{T})$$

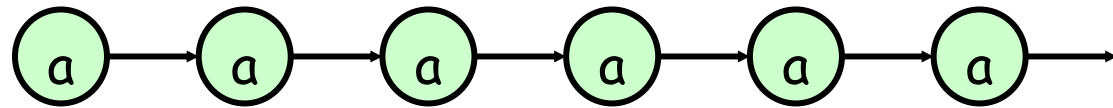
$$\langle\langle \phi_1 \vee \phi_2 \rangle\rangle(\mathcal{T}) := \langle\langle \phi_1 \rangle\rangle(\mathcal{T}) \vee \langle\langle \phi_2 \rangle\rangle(\mathcal{T})$$

$$\langle\langle \bigcirc_I \psi \rangle\rangle(\mathcal{T}) := \begin{cases} (\tau_1 \in I) \wedge \langle\langle \psi \rangle\rangle(\mathcal{T} \uparrow_1) & \text{if } |\mathcal{T}| > 1 \\ \perp & \text{otherwise} \end{cases}$$

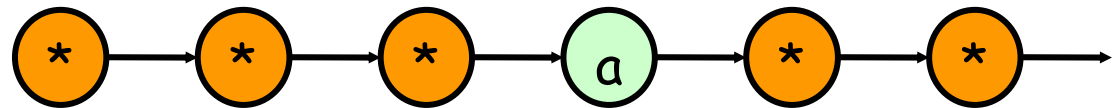
$$\langle\langle \phi_1 \mathcal{U}_I \phi_2 \rangle\rangle(\mathcal{T}) := \bigvee_{i=0}^{|\mathcal{T}|-1} (\tau_i \in K_I^{\mathcal{T}}) \wedge \langle\langle \phi_2 \rangle\rangle(\mathcal{T} \uparrow_i) \wedge \bigwedge_{j=0}^{i-1} \langle\langle \phi_1 \rangle\rangle(\mathcal{T} \uparrow_j)$$

Semantic Intuition of Linear Time Properties

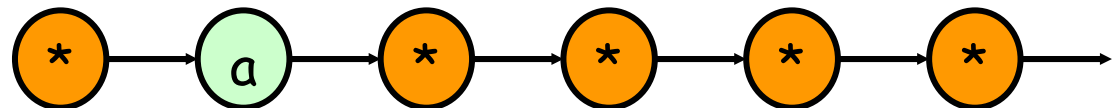
$\square a$ - always a



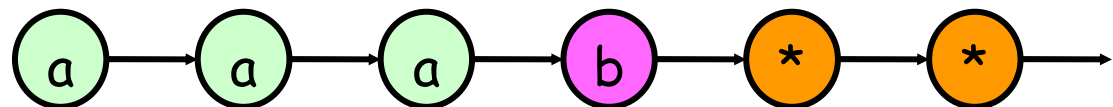
$\diamond a$ - eventually a



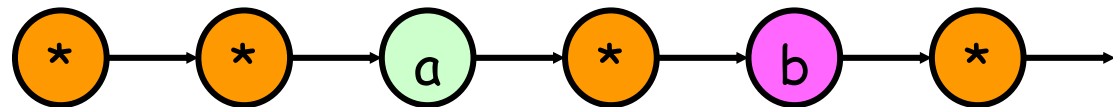
$\circ a$ - next state a



$a \cup b$ - a until b



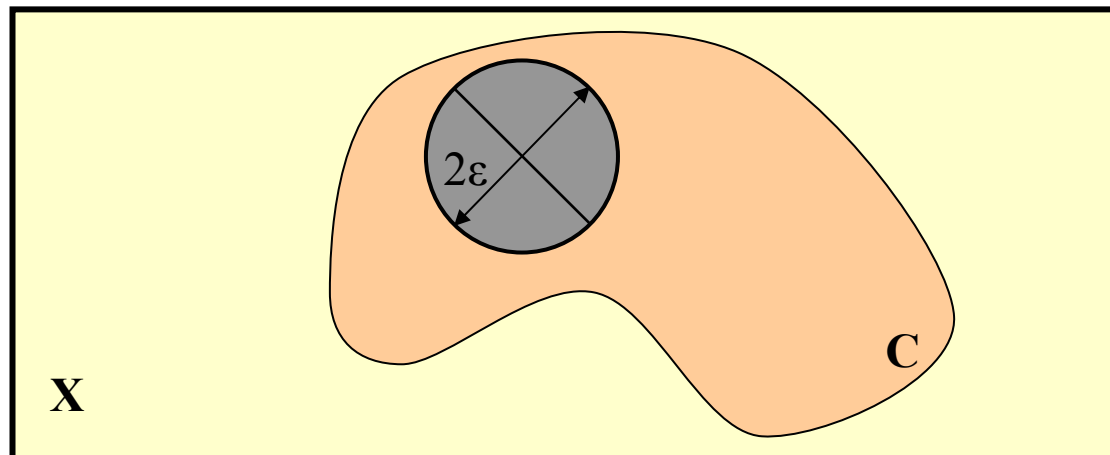
$a \text{ B } b$ - a before b



Metric Spaces

- A *metric space* (X, d) is a set X with a metric d
- A *metric* on a set X is a positive function $d: X \times X \rightarrow \mathbb{R}^+$, such that the three following properties hold
 - for all $x_1, x_2, x_3 \in X$ it is $d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3)$
 - for all $x_1, x_2 \in X$ it is $d(x_1, x_2) = 0$ iff $x_1 = x_2$
 - for all $x_1, x_2 \in X$ it is $d(x_1, x_2) = d(x_2, x_1)$
- Given a metric d , a radius $\varepsilon \in \mathbb{R}^+$ and a point $x \in X$, then the *open ε -ball* centered at x is defined as

$$B_d(x, \varepsilon) = \{ y \in X \mid d(x, y) < \varepsilon \}$$



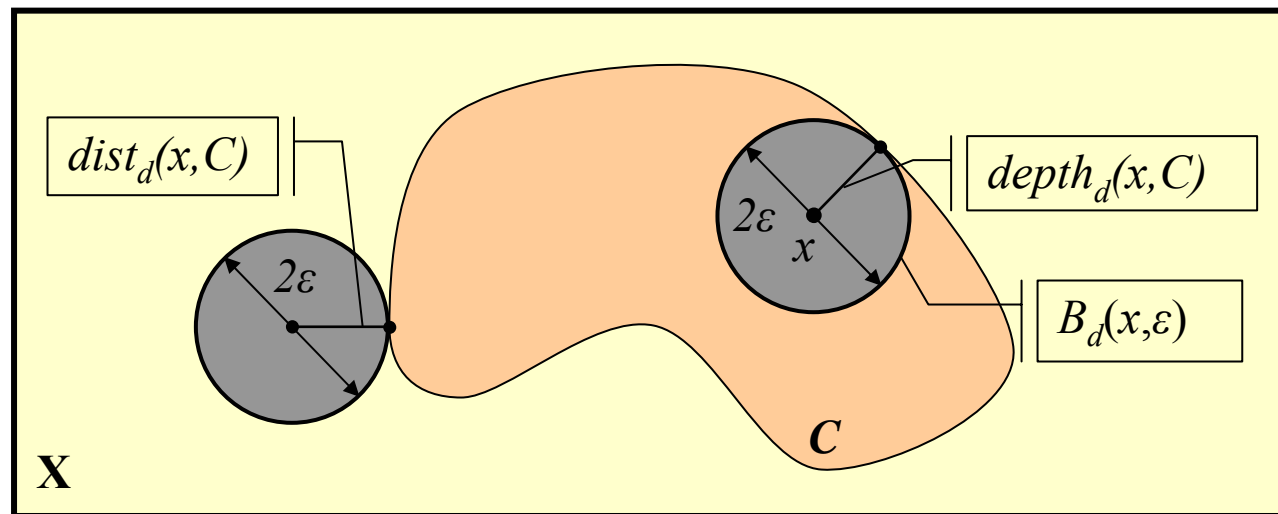
(Signed) Distance

Let $x \in X$ be a point, $C \subseteq X$ be a set and d be a metric. Then we define

$$\text{dist}_d(x, C) := \inf\{d(x, y) \mid y \in \text{cl}(C)\}$$

$$\text{depth}_d(x, C) := \text{dist}_d(x, X \setminus C)$$

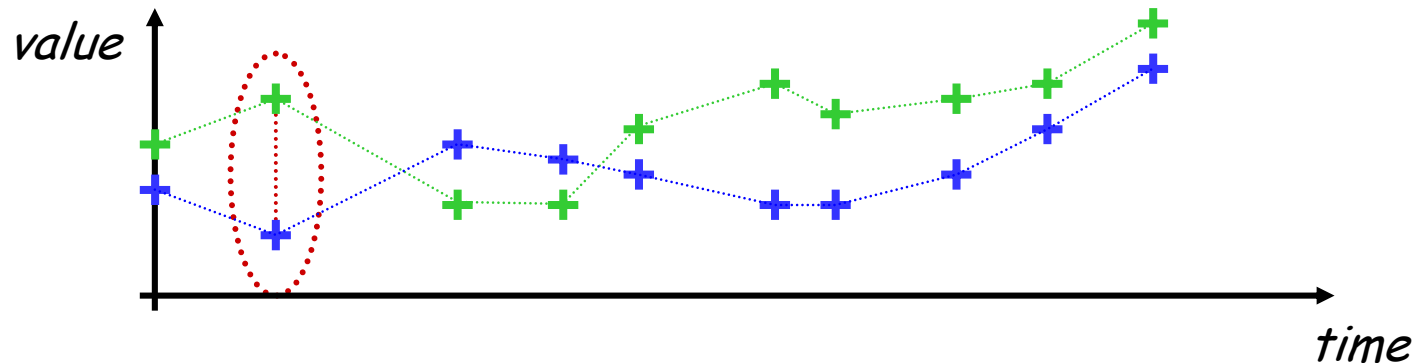
$$\text{Dist}_d(x, C) := \begin{cases} -\text{dist}_d(x, C) & \text{if } x \notin C \\ \text{depth}_d(x, C) & \text{if } x \in C \end{cases}$$



Trace distances

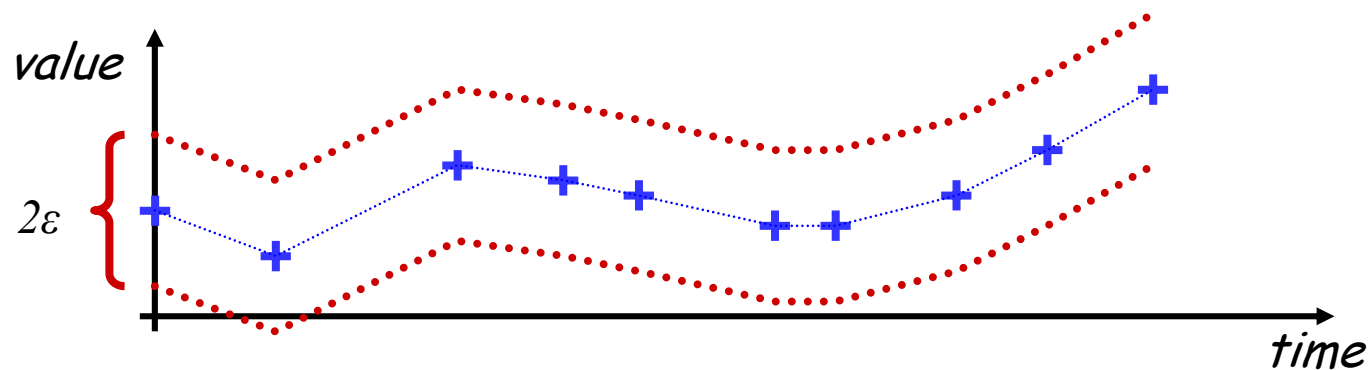
Lifting the metric d to sequences in the infinity sense

$$\rho(\tau, \sigma) = \max \{d(\tau_i, \sigma_i) \mid i=0, \dots, \text{length}(\tau)\}$$



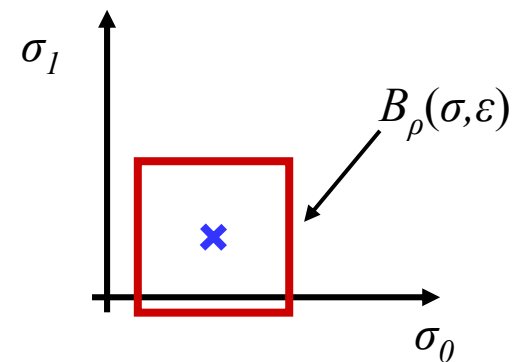
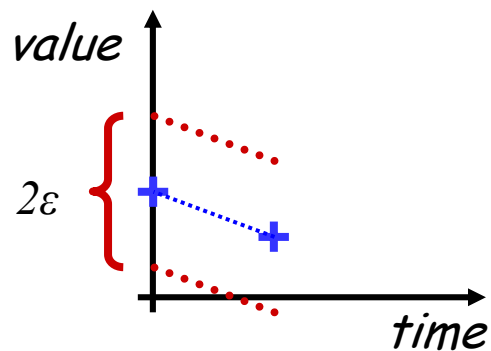
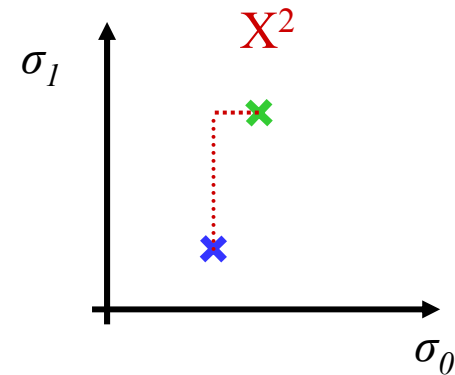
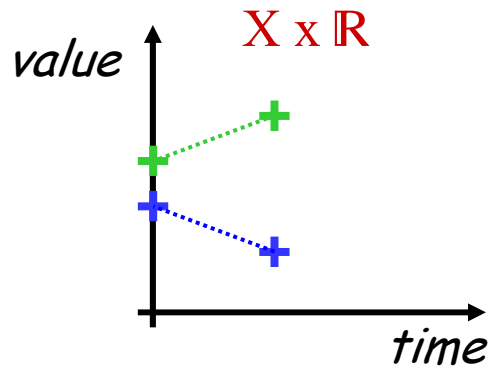
We can define a tube around a timed state sequence $T = (\sigma, \tau, \mathcal{O})$

$$TS_\varepsilon(T) = \{(\sigma', \tau, \mathcal{O}) \in TS(T) \mid \sigma' \in B_\rho(\sigma, \varepsilon)\}$$



Another point of view ...

- Think of (timed) state sequences as points in a multidimensional space
 - dimension = length of state sequence * dimension of X



Robust Satisfaction of an MTL Specification

Example: Assume $X=\mathbb{R}$, $O(a)=[1,2]$, $O(b)=[0,1)$, $length(T)=2$ and $\Phi=aUb$

Possible observations:

b true $\Rightarrow [0,1) \times \mathbb{R}$

a b $\Rightarrow [1,2] \times [0,1)$

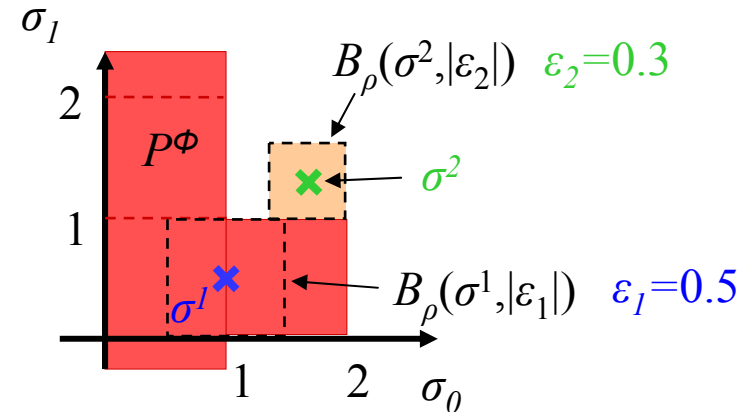
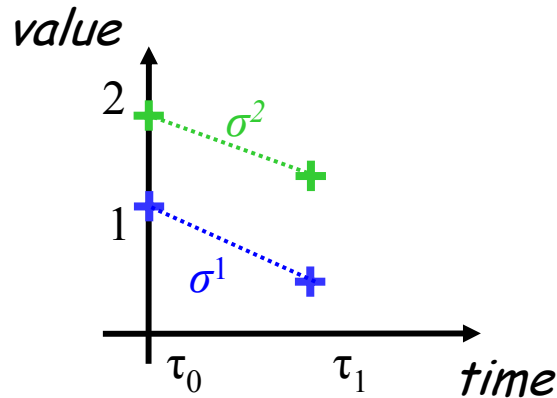


The set of SS that satisfy Φ

$$P\Phi = [0,1] \times \mathbb{R} \cup [1,2] \times [0,1)$$

$\sigma^2=1.7,1.3$

$\sigma^1=1.0,0.5$



The **robustness degree** $\varepsilon \in \mathbb{R} \cup \{-\infty, +\infty\}$ is the radius of the largest neighborhood of the state sequence

Definition of Robustness

Let φ be an MTL formula. Then we can define the following set of state seq.

$$P^\Phi = \{ S \in TS(T) \mid \langle\langle \varphi \rangle\rangle(S) = \mathbf{T} \}$$

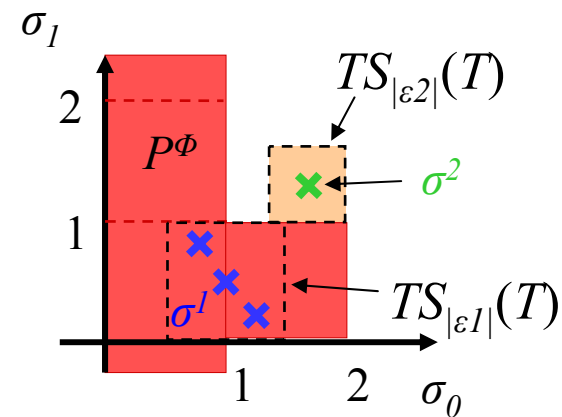
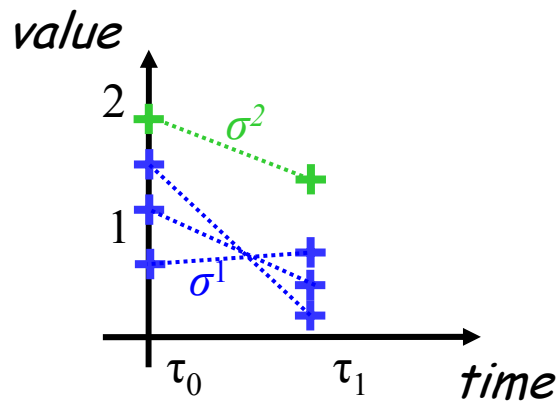
Then, given a state sequence σ , we can define the *robustness degree* as

$$\varepsilon := \mathbf{Dist}_\rho(\sigma, P^\Phi)$$

Robust Satisfaction of an MTL Specification

Proposition: Let Φ be an MTL formula, S be a timed state sequence and $|\varepsilon| > 0$ be the robustness degree, then for all T with $T \in B_\rho(S, \varepsilon)$ we have that

$$\langle\langle \Phi \rangle\rangle(S) = \langle\langle \Phi \rangle\rangle(T)$$



Robust Satisfaction of an MTL Specification

- But can we compute P^Φ and then $\mathbf{Dist}_\rho(\sigma, P^\Phi)$?
not with continuous time semantics!

With discrete time semantics:

- We must generate all the strings of a certain length in the language of Φ
 - LTL: construct finite automaton [Giannakopoulou, Havelund '01]
 - ♦ Get an approximation of the set of words [Kannan et al '95]
 - MTL: construct the timed automaton [Ouaknine, Worrell '05], [Maler et al '06]
- Find their set representation P^Φ in the space X
 - i.e. $P^\Phi = O(b) \times O(\text{true}) \cup O(a) \times O(b)$
- Compute the distance
 - doable even in very high dimensional spaces since we use infinity norm
- Solution:

Compute a *robustness estimate*

Robust Semantics for MTL

Syntax:

$$\phi ::= \top \mid \pi \mid \neg\phi_1 \mid \phi_1 \vee \phi_2 \mid \bigcirc_I \phi_1 \mid \phi_1 \mathcal{U}_I \phi_2$$

I can be of any bounded or unbounded interval of \mathbb{Q}^+ .

i.e. $I = [0, +\infty)$, $I = [2.5, 9.8]$

Robust Semantics:

$$\llbracket v \rrbracket(\mathcal{T}) := v$$

$$\llbracket \pi \rrbracket(\mathcal{T}) := \mathbf{Dist}_d(\sigma_0, \mathcal{O}(\pi))$$

$$\llbracket \neg\psi \rrbracket(\mathcal{T}) := -\llbracket \psi \rrbracket(\mathcal{T})$$

$$\llbracket \phi_1 \vee \phi_2 \rrbracket(\mathcal{T}) := \llbracket \phi_1 \rrbracket(\mathcal{T}) \sqcup \llbracket \phi_2 \rrbracket(\mathcal{T})$$

$$\llbracket \bigcirc_I \psi \rrbracket(\mathcal{T}) := \begin{cases} \mathbf{mv}(\tau_1 \in I) \sqcap \llbracket \psi \rrbracket(\mathcal{T} \uparrow_1) & \text{if } |\mathcal{T}| > 1 \\ -\infty & \text{otherwise} \end{cases}$$

$$\llbracket \phi_1 \mathcal{U}_I \phi_2 \rrbracket(\mathcal{T}) := \bigsqcup_{i=0}^{|\mathcal{T}|-1} (\mathbf{mv}(i \in K_I^{\mathcal{T}}) \sqcap \llbracket \phi_2 \rrbracket(\mathcal{T} \uparrow_i) \sqcap \prod_{j=0}^{i-1} \llbracket \phi_1 \rrbracket(\mathcal{T} \uparrow_j))$$

$$\llbracket \cdot \rrbracket : \Phi_{\text{MTL}} \times \text{TS} \rightarrow \mathbb{R} \cup \{-\infty, +\infty\}$$

$$\sqcup : \overline{\mathbb{R}} \times \overline{\mathbb{R}} \rightarrow \overline{\mathbb{R}} \quad \sqcap : \overline{\mathbb{R}} \times \overline{\mathbb{R}} \rightarrow \overline{\mathbb{R}}$$

$$x \sqcup y := \max(\{x, y\})$$

$$x \sqcap y := \min(\{x, y\})$$

Robust and Boolean Semantics for MTL

Proposition: Let ϕ be an MTL formula and \mathcal{T} be a signal, then

- | | | | |
|-----|-----------------------------------------------------------------------------------------------|-----|--------------------------------------------------------------------------------------------------|
| (1) | $[[\phi]](\mathcal{T}) > 0 \Rightarrow \langle\langle\phi\rangle\rangle(\mathcal{T}) = \top$ | (2) | $\langle\langle\phi\rangle\rangle(\mathcal{T}) = \top \Rightarrow [[\phi]](\mathcal{T}) \geq 0$ |
| (3) | $[[\phi]](\mathcal{T}) < 0 \Rightarrow \langle\langle\phi\rangle\rangle(\mathcal{T}) = \perp$ | (4) | $\langle\langle\phi\rangle\rangle(\mathcal{T}) = \perp \Rightarrow [[\phi]](\mathcal{T}) \leq 0$ |

Main results

Theorem: Let Φ be an MTL formula and T be a TSS, then

$$-\text{depth}_\rho(\tau, N_T^\Phi) \leq \llbracket \Phi \rrbracket(T) \leq \text{depth}_\rho(\tau, P_T^\Phi)$$

$$|\llbracket \Phi \rrbracket(T)| \leq |\mathbf{Dist}_\rho(\tau, P_T^\Phi)|$$

Theorem: Let Φ be an MTL formula and T be a TSS, then

$$\text{if } \llbracket \Phi \rrbracket(T) = \varepsilon \text{ and } |\varepsilon| > 0,$$

then $\llbracket \Phi \rrbracket(\mathcal{S}) = \llbracket \Phi \rrbracket(T)$ for all $\mathcal{S} \in \mathcal{TS}_{|\varepsilon|}(T)$

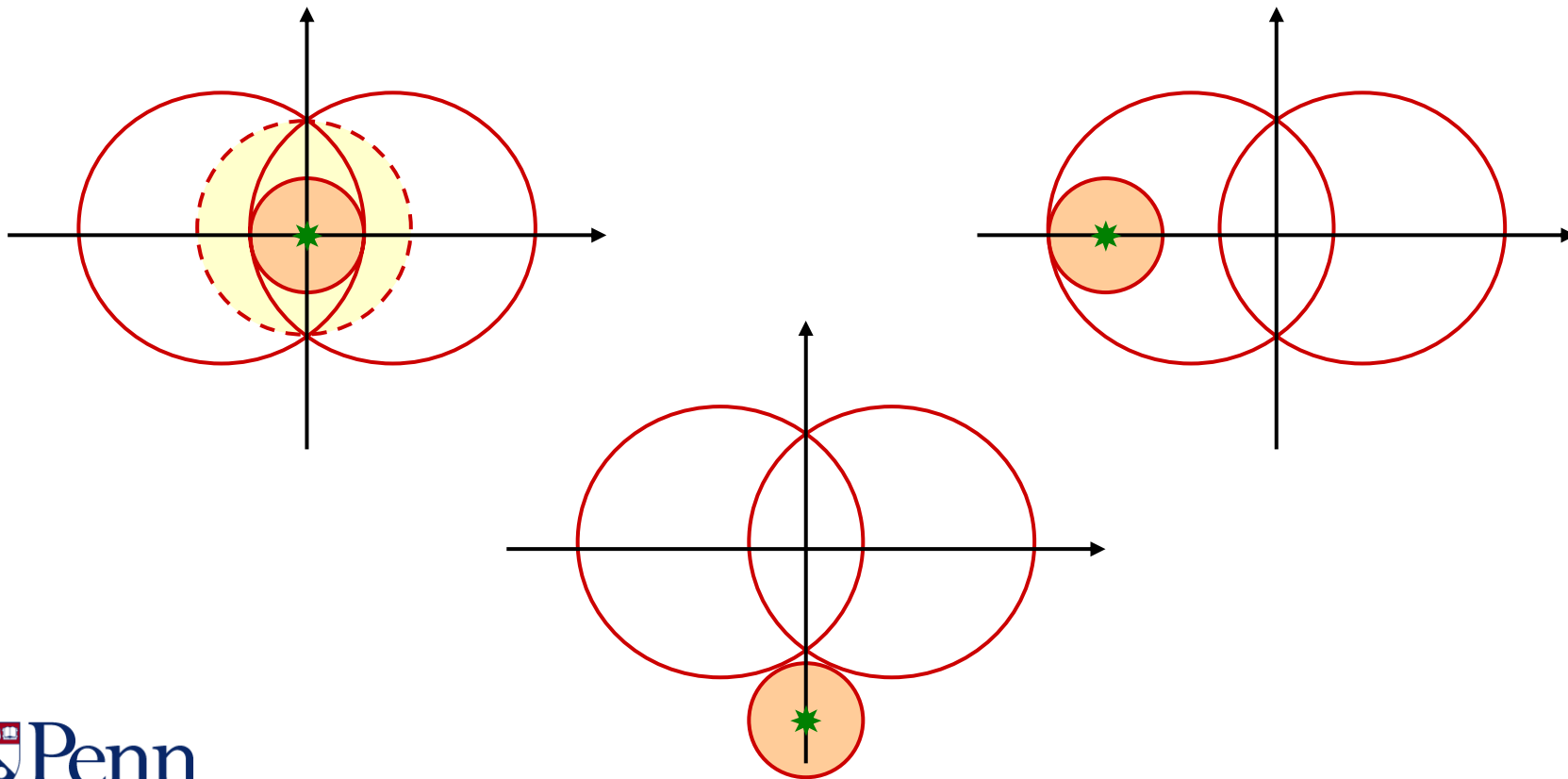
Sketch of Proof

Proposition: Let $\Phi = \Phi_1 \vee \Phi_2$ be an MTL formula, T be a TSS and assume

$$-\text{dist}_\rho(\sigma, P_T^{\phi_i}) \leq \llbracket \phi_i \rrbracket(T) \leq \text{dist}_\rho(\sigma, N_T^{\phi_i})$$

then

$$-\text{dist}_\rho(\sigma, P_T^\Phi) \leq \llbracket \Phi \rrbracket(T) \leq \text{dist}_\rho(\sigma, N_T^\Phi)$$



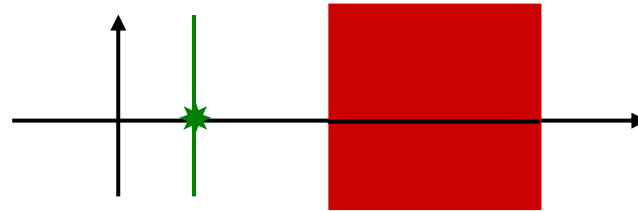
Sketch of Proof

Proposition: Let $\varphi = \pi$ be an MTL formula and T be a TSS, then

$$\text{Dist}_\rho(\sigma, P_T^\pi) = \text{Dist}_d(\sigma_0, \mathcal{O}(\pi))$$

Proposition: Let $\varphi = \bigcirc^i \psi$ be an MTL formula and T be a TSS, then

$$\text{Dist}_\rho(\sigma, P_T^\varphi) = \text{Dist}_\rho(\sigma \uparrow_1, P_{T \uparrow_1}^\psi)$$



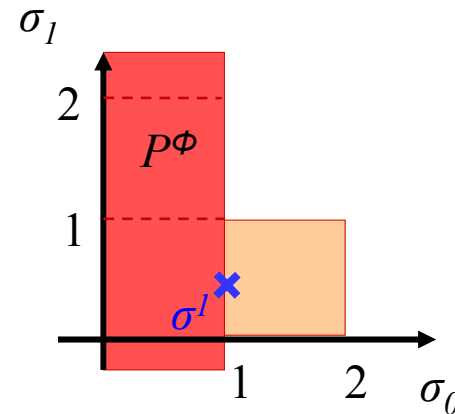
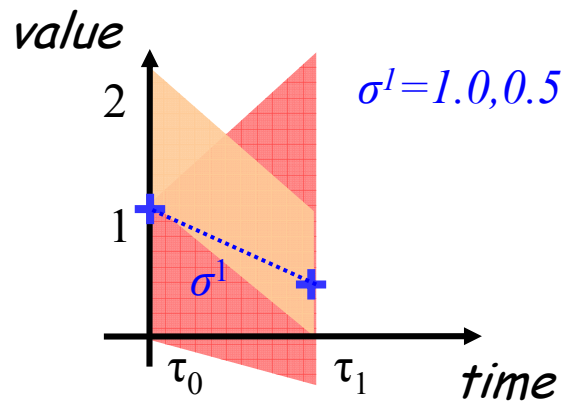
For the case $\varphi = \varphi_1 U_T \varphi_2$ of the main theorem, we point out that

$$\llbracket \phi \rrbracket(\mathcal{T}) = \bigsqcup_{i=0}^{|\mathcal{T}|} (\llbracket i \in K_{\mathcal{T}}^T \rrbracket(\mathcal{T}) \sqcap \llbracket \bigcirc^i \phi_2 \rrbracket(\mathcal{T}) \sqcap \prod_{j=0}^{i-1} \llbracket \bigcirc^j \phi_1 \rrbracket(\mathcal{T}))$$

First remark on the equality

Remark 1: Assume NNF. The equality is lost due to

- the disjunctions at the level of the atomic propositions: i.e. $\pi_1 \vee \pi_2$
 - (theoretically) can be fixed by letting $O(\pi_{12}) = O(\pi_1) \cup O(\pi_2)$
- the disjunctions at the level of subformulas: i.e. $\varphi_1 \vee \varphi_2$
- tautologies: i.e. $\pi \vee \neg\pi$
 - $\pi \vee \neg\pi \equiv \top$ but $[[\pi \vee \neg\pi]](\top) < +\infty$
- because Φ can be satisfied at different points in time
 - This is a real problem



$$[[\phi]](\mathcal{T}_1) = 0$$

Second remark on the equality

Remark 2: We can get equality for very restricted fragments of MTL.

- for the fragment (\wedge, \square) when $\langle\langle\phi\rangle\rangle(\mathcal{I}) = \top$
- for the fragment (\vee, \diamond) when $\langle\langle\phi\rangle\rangle(\mathcal{I}) = \perp$

When the negations appear in front of the atomic propositions and the sets in space X are convex.

Computing the robustness estimate

We re-write the temporal operator U in a recursive way

$$\llbracket \phi_1 \mathcal{U}_{\mathcal{I}} \phi_2 \rrbracket (\mathcal{T}) = \begin{cases} ((0 \in \mathcal{I}) \wedge \llbracket \phi_2 \rrbracket (\mathcal{T})) \vee \\ \vee (\llbracket \phi_1 \rrbracket (\mathcal{T}) \wedge \llbracket \phi_1 \mathcal{U}_{\mathcal{I}-\tau_1} \phi_2 \rrbracket (\mathcal{T}\uparrow_1)) & \text{if } |\mathcal{T}| > 1 \\ (0 \in \mathcal{I}) \wedge \llbracket \phi_2 \rrbracket (\mathcal{T}) & \text{otherwise} \end{cases}$$

$$\llbracket \phi_1 \mathcal{U}_{\mathcal{I}} \phi_2 \rrbracket (\mathcal{T}) = \begin{cases} (\llbracket 0 \in \mathcal{I} \rrbracket (\mathcal{T}) \sqcap \llbracket \phi_2 \rrbracket (\mathcal{T})) \sqcup \\ \sqcup (\llbracket \phi_1 \rrbracket (\mathcal{T}) \sqcap \llbracket \phi_1 \mathcal{U}_{\mathcal{I}-\tau_1} \phi_2 \rrbracket (\mathcal{T}\uparrow_1)) & \text{if } |\mathcal{T}| > 1 \\ \llbracket 0 \in \mathcal{I} \rrbracket (\mathcal{T}) \sqcap \llbracket \phi_2 \rrbracket (\mathcal{T}) & \text{otherwise} \end{cases}$$

We define hybrid semantics

$$\text{Negation: } \neg(v, \varepsilon) := (\neg v, -\varepsilon)$$

$$\text{Disjunction: } (v_1, \varepsilon_1) \vee (v_2, \varepsilon_2) := (v_1 \vee v_2, \max(\varepsilon_1, \varepsilon_2))$$

Required simplifications at each iteration

$$\phi \wedge (\top, +\infty) \equiv \phi$$

$$\phi \vee (\perp, -\infty) \equiv \phi$$

$$\phi \vee (\top, +\infty) \equiv (\top, +\infty)$$

$$\phi \wedge (\perp, -\infty) \equiv (\perp, -\infty)$$

Computing the robustness estimate

Algorithm 1 Monitoring Timed State Sequences

Input: The MTL formula ϕ and the timed state sequence $\mathcal{T} = (\sigma, \tau, \mathcal{O})$

Output: The formula's Boolean truth value and the robustness parameter

```
1: procedure MONITOR( $\phi, \mathcal{T}$ )
2:   if  $|\mathcal{T}| > 1$  then return  $\phi \leftarrow \text{PROGRESS}(\phi, \sigma_0, \tau_1, \perp, \mathcal{O})$ 
3:   else return  $\phi \leftarrow \text{PROGRESS}(\phi, \sigma_0, 0, \top, \mathcal{O})$ 
4:   end if
5:   if  $\phi = (v, \varepsilon)$  then return  $(v, \varepsilon)$   $\triangleright v \in \{\top, \perp\}$  and  $\varepsilon \in \overline{\mathbb{R}}$ 
6:   else return MONITOR( $\phi, \mathcal{T} \uparrow_1$ )
7:   end if
8: end procedure
```

Theorem: $H[\phi](\mathcal{T}) = H[\text{Progress}(\phi, \sigma_0, \tau_1, \dots)](\mathcal{T} \uparrow_1)$

Computing the robustness estimate

Algorithm 2 Formula Progression Algorithm

Input: The MTL formula ϕ , the current state s , the time period Δt for the next state, a variable $last$ indicating whether the next state is the last and the mapping \mathcal{O}

Output: The MTL formula ϕ that has to hold at the next state

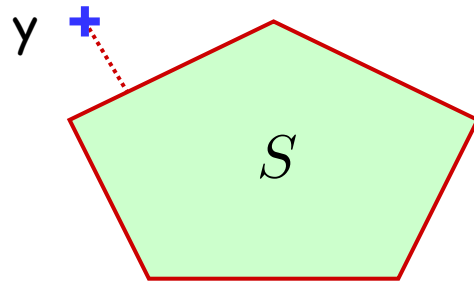
```
1: procedure PROGRESS( $\phi, s, \Delta t, last, \mathcal{O}$ )
2:   if  $\phi = (v, \varepsilon) \in \{\perp, \top\} \times \mathbb{R}$  then return  $(v, \varepsilon)$ 
3:   else if  $\phi = \pi$  then return  $(s \in \mathcal{O}(\pi), \mathbf{Dist}_d(s, \mathcal{O}(\pi)))$ 
4:   else if  $\phi = \neg\psi$  then return  $\neg$ PROGRESS( $\psi, s, \Delta t, last, \mathcal{O}$ )
5:   else if  $\phi = \phi_1 \vee \phi_2$  then
6:     return PROGRESS( $\phi_1, s, \Delta t, last, \mathcal{O}$ )  $\vee$  PROGRESS( $\phi_2, s, \Delta t, last, \mathcal{O}$ )
7:   else if  $\phi = \bigcirc_{\mathcal{I}}\psi$  then return HYBRID( $\neg last \wedge (\Delta t \in \mathcal{I})$ )  $\wedge \psi$ 
8:   else if  $\phi = \phi_1 \mathcal{U}_{\mathcal{I}}\phi_2$  then
9:      $\alpha \leftarrow$  HYBRID( $0 \in \mathcal{I}$ )  $\wedge$  PROGRESS( $\phi_2, s, \Delta t, last, \mathcal{O}$ )
10:     $\beta \leftarrow$  HYBRID( $\neg last \wedge (0 \in \overleftarrow{\mathcal{I}})$ )  $\wedge$  PROGRESS( $\phi_1, s, \Delta t, last, \mathcal{O}$ )  $\wedge \phi_1 \mathcal{U}_{\mathcal{I}-\Delta t}\phi_2$ 
11:    return  $\alpha \vee \beta$ 
12:   end if
13: end procedure

1: function HYBRID( $Bool$ )
2:   if  $Bool = \top$  return  $(\top, +\infty)$  else return  $(\perp, -\infty)$  end if
3: end function
```

Computing Distances

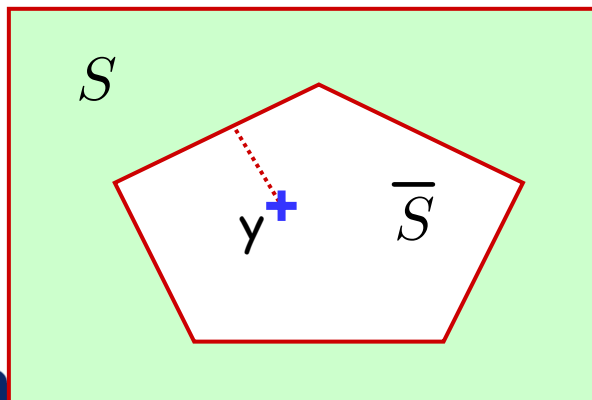
For computational reasons we allow only **convex** or **concave** sets which are usually described using **intersections** OR **unions** of **halfspaces**

$$S = \{x \mid \bigwedge_{j \in J} (a_j \cdot x \leq b_j)\}$$



$$S = \{x \mid \bigvee_{j \in J} (a_j \cdot x \leq b_j)\}$$

such that \overline{S} is convex



Assume S is convex

if $y \notin S$

Solve QP:

$$\min (x-y)^T(x-y)$$

$$\text{s.t. } \bigwedge_j a_j \cdot x \leq b_j$$

$$\text{Dist}(y,S) = -\text{minvalQP}$$

else

for $j \in J$

$$d_j = |b_j - a_j \cdot y| / \text{norm}(a_j)$$

end for

$$\text{Dist}(y,S) = \min \{d_j\}_{j \in J}$$

end if

Complexity?

- [Thati and Rosu] The best known bounds for monitoring algorithms use canonical forms consisting of exclusive disjunction of conjunctions
 - MTL: space $O(m2^m)$, time $O(|\tau|m^32^{3m})$
 - LTL: space: $O(|\phi|2^{|\phi|})$, time $O(|\tau||\phi|^32^{3|\phi|})$
- [Boyd and Vandenberghe] How easy is to compute the signed distance?
 - If $X=\mathbb{R}$, the set C is an interval and $d(x,y)=|x-y|$, then the problem reduces to finding the minimum of two values.
 - If $X=\mathbb{R}^n$ and $d(x,y)=(x-y)^T(x-y)$, then
 - ♦ If C is a closed and convex set we can calculate the distance by solving a convex optimization problem.

$$\begin{array}{ll} \min & \|x-x_0\|_2 \\ \text{s.t} & f_i(x)\leq 0 \quad i=1,\dots,k \quad (\text{convex inequalities}) \\ & Ax=b \quad (\text{linear equalities}) \end{array} \quad O(1)\sqrt{\theta(B)} \log \left(1 + \frac{\mu_0\theta(B)}{\varepsilon} \right)$$
 - ♦ If C is a polytope, then we solve a QP
 - ♦ If C is a hyperplane $C = \{x \mid a^T x=b\}$ or a halfspace $C = \{x \mid a^T x\leq b\}$, then there exist analytical solutions

Example I - Toy

MTL Spec:

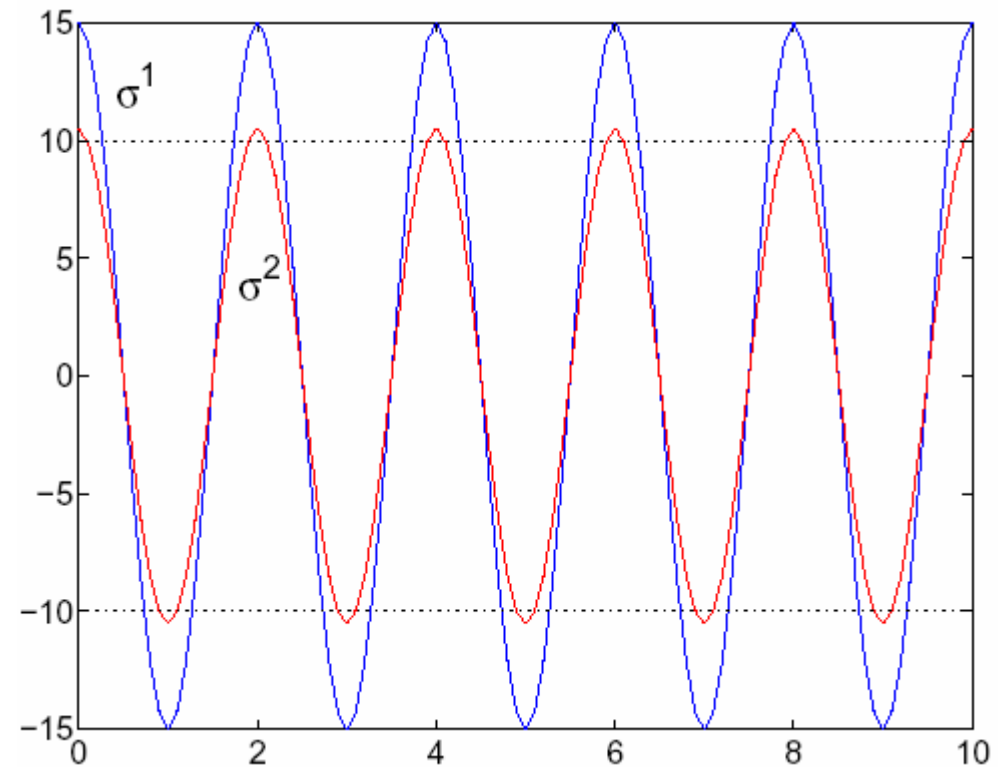
$$\square((x \leq -10) \rightarrow \diamond_{\leq 2}(x \geq 10))$$

Robustness estimate σ^1 :

$$\varepsilon^1 = 5$$

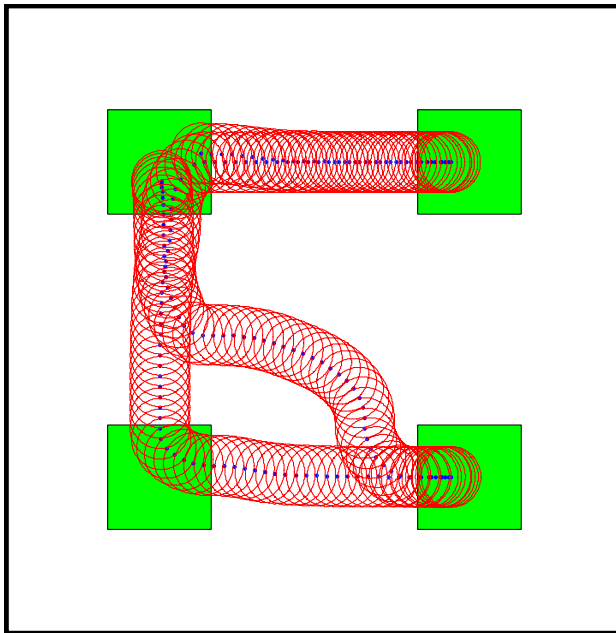
Robustness estimate σ^2 :

$$\varepsilon^2 = 0.5$$



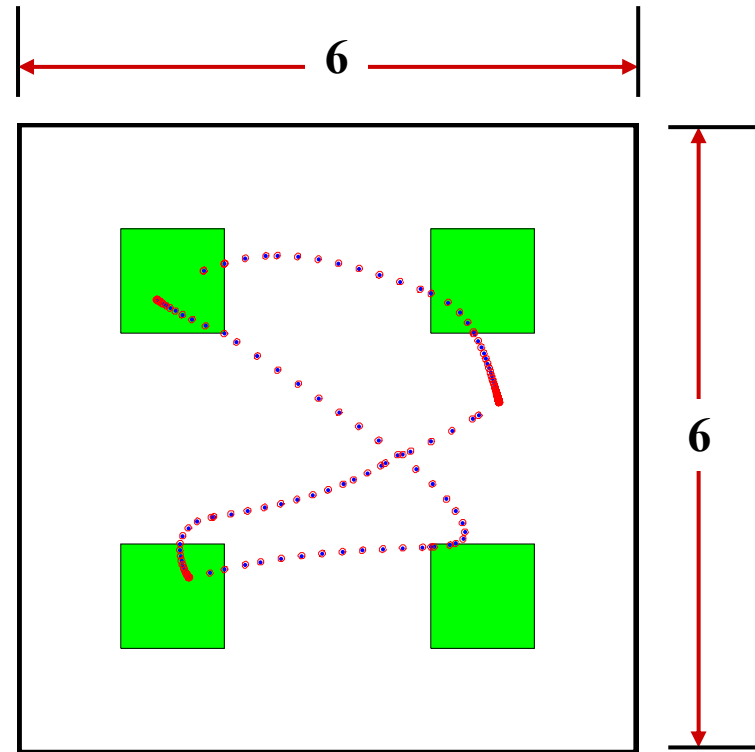
Example II - Robustness of Trajectories

$$\diamond_{I_2}(\pi_2 \wedge \diamond_{I_3}(\pi_3 \wedge \diamond_{I_4}(\pi_4 \wedge \neg(\pi_3 \vee \pi_4) U_{II} \pi_1))$$



Robustness estimate

$$\varepsilon = 0.28872$$

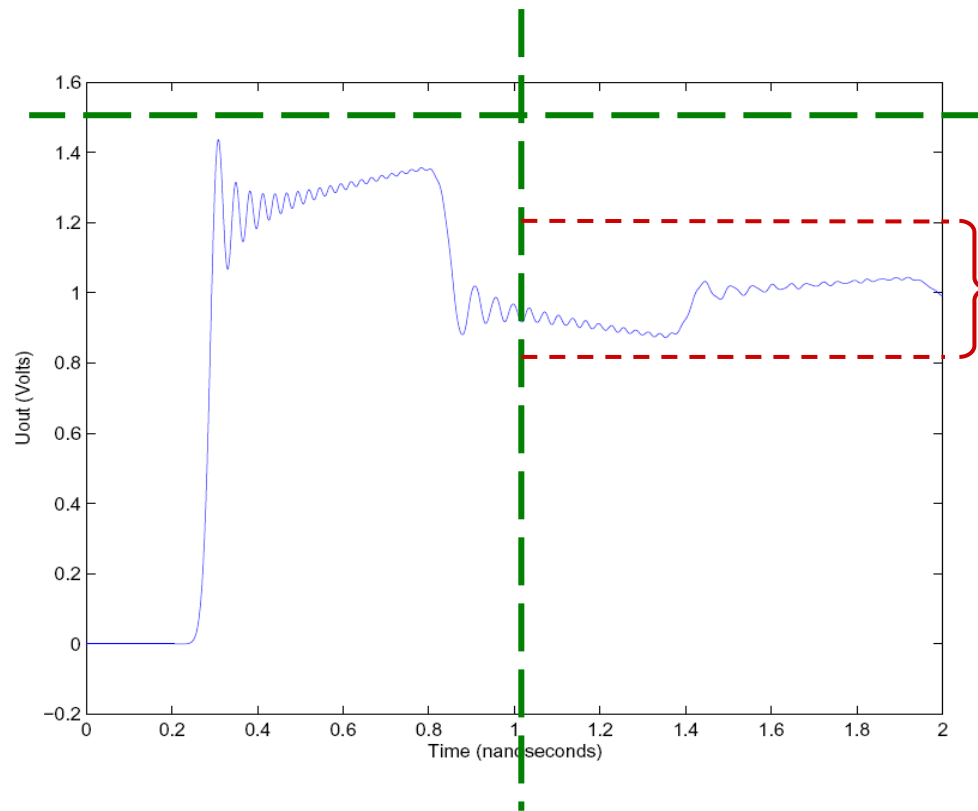


Robustness estimate

$$\varepsilon = 0.031333$$

Example III - Transient Dynamics*

*from Zhi Han's PhD Thesis 2005



System:

$$\dot{x}(t) = Ax(t) + bU_{in}(t)$$

$$U_{out}(t) = Cx(t)$$

Step input ($t = 0^+$):

$$U_{in}(t) = 1$$

Steady state at $t = 0$:

$$x(0) = -A^{-1}bU_{in}(0)$$

Property:

$$\phi = \square \pi_1 \wedge \diamond [0, T] \square \pi_2$$

$$\mathcal{O}(\pi_1) = [-\theta, \theta]$$

$$\mathcal{O}(\pi_2) = [0.8, 1.2]$$

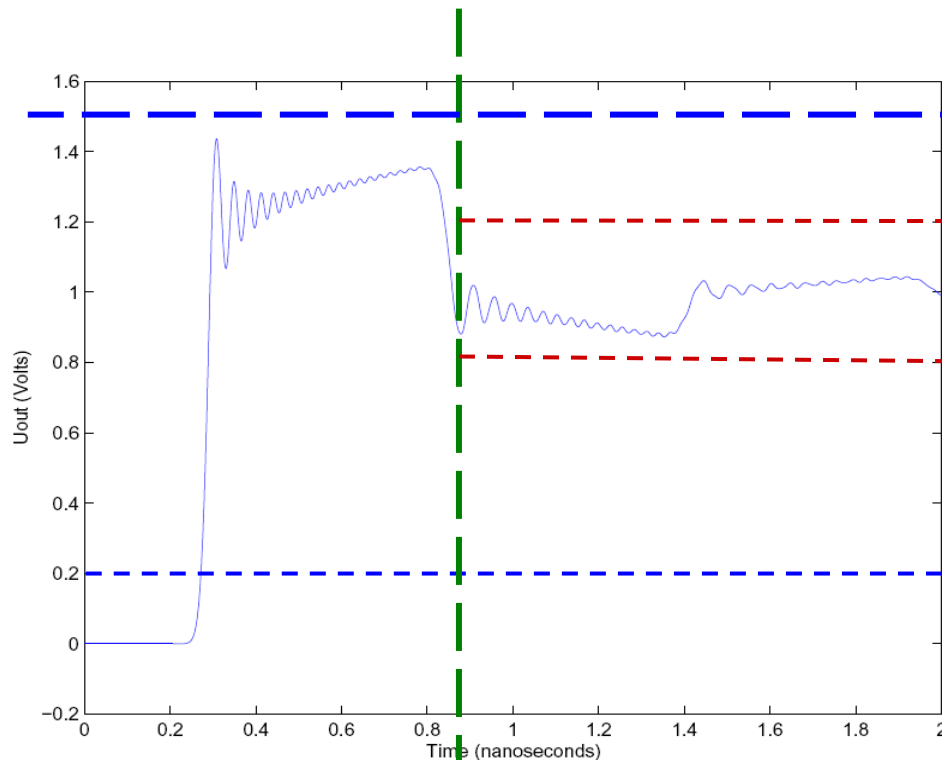
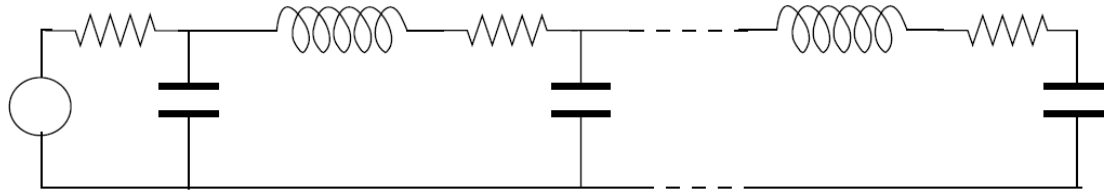
Robustness estimate for $T=1$ and $\theta=1.5$

$$\varepsilon = 0.0702$$

Part II: Verification using Simulations



Verification Using Simulation - Motivation



System:

$$\dot{x}(t) = Ax(t) + bU_{in}(t)$$

$$U_{out}(t) = Cx(t)$$

Step input ($t = 0^+$):

$$U_{in}(t) = 1$$

Steady state at $t = 0^-$:

$$x(0) = -A^{-1}bU_{in}(0)$$

Property:

$$\phi = \square \pi_1 \wedge \diamond_{[0,T]} \square \pi_2$$

$$\mathcal{O}(\pi_1) = [-\theta, \theta]$$

$$\mathcal{O}(\pi_2) = [0.8, 1.2]$$

Initial conditions:

$$U_{in}(0) \in [-0.2, 0.2]$$

How do we solve this problem?

Closed-loop system Σ :

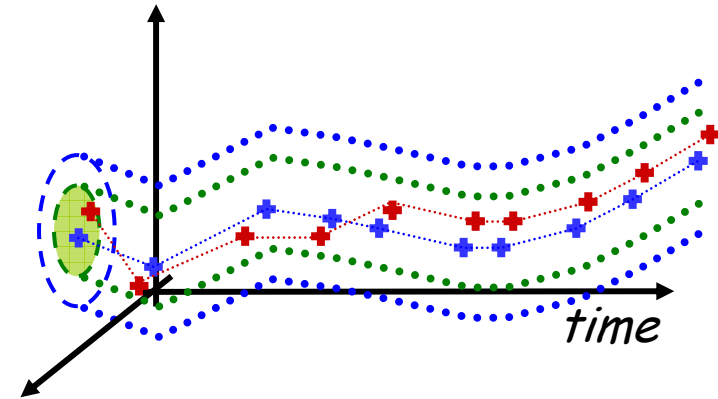
$$\dot{x} = f(x)$$

$$y = g(x)$$

$$X_0 \subseteq X$$

Specification Φ

$$\mathcal{L}(\Sigma') \subseteq \mathcal{L}(\Phi)$$



$$Proj_0(P^\Phi \cap \mathcal{L}(\Sigma'))$$

$$X_0$$

$$\left[\begin{array}{l} \llbracket \phi \rrbracket(\mathcal{T}) = \varepsilon \\ B_\rho(\sigma, |\varepsilon|) \end{array} \right]$$

How do we solve this problem?

Closed-loop system Σ :

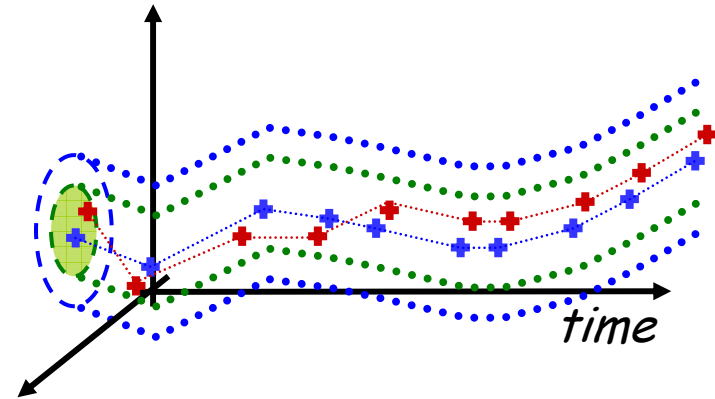
$$\dot{x} = f(x)$$

$$y = g(x)$$

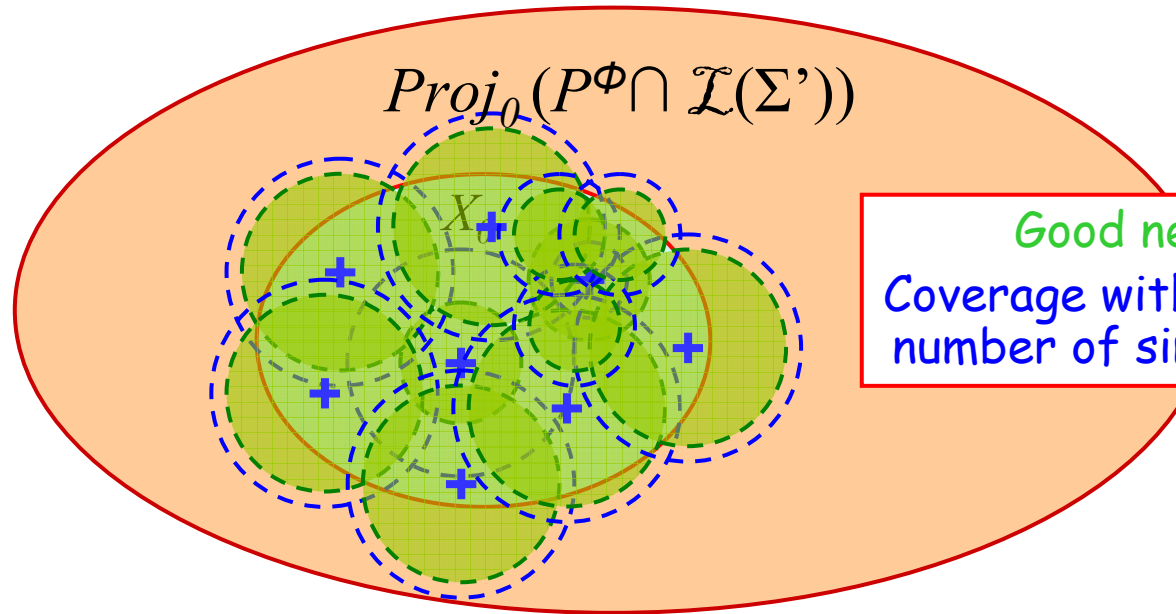
Specification Φ

$$X_0 \subseteq X$$

$$\mathcal{L}(\Sigma') \subseteq \mathcal{L}(\Phi)$$



$$Proj_0(P^\Phi \cap \mathcal{L}(\Sigma'))$$



Good news!
Coverage with a finite
number of simulations

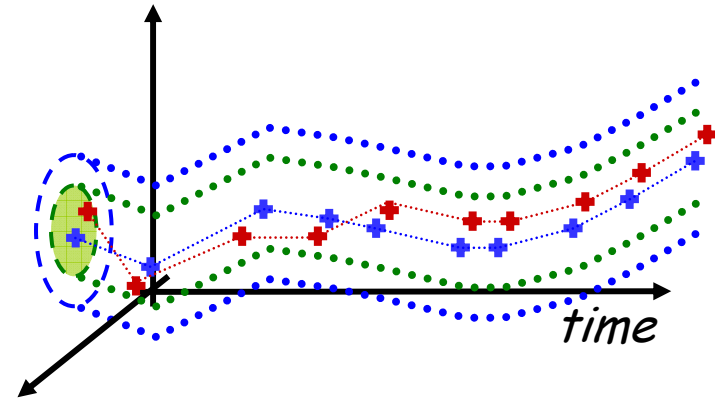
How do we solve this problem?

Closed-loop system Σ :

$$\begin{aligned} \dot{x} &= f(x) \\ y &= g(x) \end{aligned} \quad X_0 \subseteq X$$

Specification Φ

$$\mathcal{L}(\Sigma') \subseteq \mathcal{L}(\Phi)$$



$$Proj_0(P^\Phi \cap \mathcal{L}(\Sigma'))$$

X_0

Even better news!
It is possible to verify
the system with
just one simulation

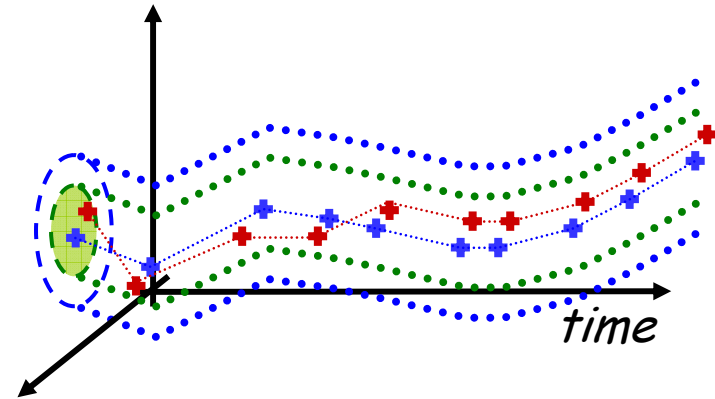
How do we solve this problem?

Closed-loop system Σ :

$$\begin{aligned} \dot{x} &= f(x) \\ y &= g(x) \end{aligned} \quad X_0 \subseteq X$$

Specification Φ

$$\mathcal{L}(\Sigma') \subseteq \mathcal{L}(\Phi)$$



$$\text{Proj}_0(P^\Phi \cap \mathcal{L}(\Sigma'))$$

$$|\llbracket \Phi \rrbracket(\mathcal{T})| \leq |\text{Dist}_\rho(\sigma, P_T^\Phi)|$$

X_0

Bad news!

We do not get completeness

$$\llbracket \Phi \rrbracket(\mathcal{T}) = 0 \not\Rightarrow \text{Dist}_\rho(\sigma, P_T^\Phi) = 0$$

$$\varepsilon := \text{Dist}_\rho(\tau, P_T^\Phi)$$

How do we solve this problem?

Closed-loop system Σ :

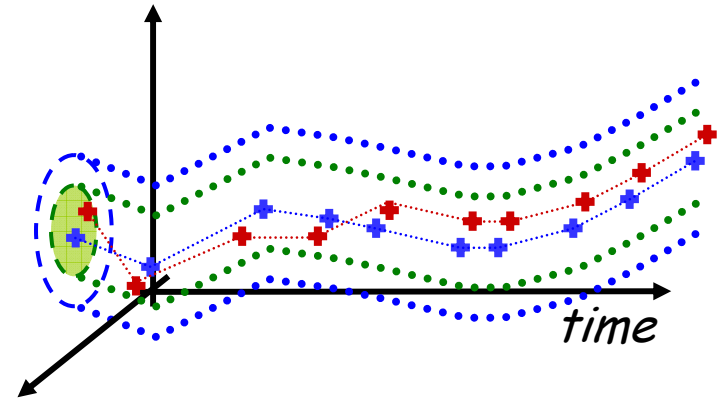
$$\dot{x} = f(x)$$

$$y = g(x)$$

Specification Φ

$$X_0 \subseteq X$$

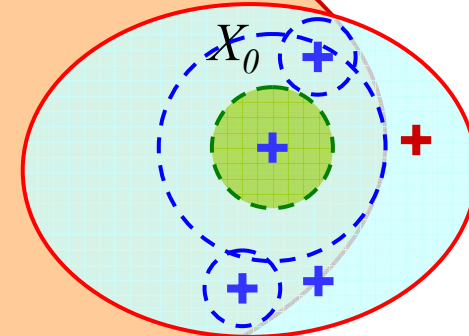
$$\mathcal{L}(\Sigma') \subseteq \mathcal{L}(\Phi)$$



$$Proj_0(P^\Phi \cap \mathcal{L}(\Sigma'))$$

Observation!

A robust system with respect to the property requires less simulations



Bisimulation functions

- A *bisimulation function* on a set X is a positive continuous function $V: X \times X \rightarrow \mathbb{R}^+$, such that the three following properties hold
 - 1) for all $x \in X$ it is $V(x,x) = 0$
 - 2) for all $x_1, x_2 \in X$ it is $d(g(x_1),g(x_2)) \leq V(x_1,x_2)$
 - 3) for all $x_1, x_2 \in X$ it is

$$\frac{\partial V}{\partial x_1} \cdot f(x_1) + \frac{\partial V}{\partial x_2} \cdot f(x_2) \leq 0$$

- Intuitively, a bisimulation function:
 - bounds the distance between the observations
 - does not increase during the evolution of the system

Computing bisimulation functions

Quadratic Bisimulation Functions for Deterministic Linear Systems

$$\begin{array}{l} \dot{x} = Ax \\ y = Cx \end{array} \rightarrow y$$

$V(x) = \sqrt{x^T M x}$
is a bisimulation function if

$$M \geq C^T C$$

$$A^T M + M A \leq 0$$

Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

WeA16.4

Approximate Bisimulations for Constrained Linear Systems

Antoine Girard and George J. Pappas

Bisimulation Functions using Sum Of Squares Relaxation

$$\begin{array}{l} \dot{x} = f(x) \\ y = g(x) \end{array} \rightarrow y$$

$$V(x_1, x_2) = \sqrt{q(x_1, x_2)}$$

is a bisimulation function if

$$q(x_1, x_2) - \|g_1(x_1) - g_2(x_2)\|^2 \text{ is SOS}$$

$$-\frac{\partial q(x_1, x_2)}{\partial x_1} f_1(x_1) - \frac{\partial q(x_1, x_2)}{\partial x_2} f_2(x_2) \text{ is SOS}$$

Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

MoB01.3

Approximate Bisimulations for Nonlinear Dynamical Systems

Antoine Girard and George J. Pappas

Some math ...

Theorem: Let V be a bisimulation function, (x_1, y_1) and (x_2, y_2) be trajectories of Σ and T_1, T_2 be the associated TSS, then

$$(\exists i \in \{1, 2\}. V(x_i(0), x_i(0)) < |[\![\phi]\!](T_i)|) \implies (\llbracket \phi \rrbracket(T_1) = \llbracket \phi \rrbracket(T_2))$$

Proposition: Let V be a bisimulation function. For any compact set of initial conditions $I \subseteq \mathbb{R}$, for all $\delta > 0$, there exists a finite set of points $\{x_1, \dots, x_r\} \subseteq I$ such that

for all $x \in I$, there exists x_i , such that $V(x, x_i) \leq \delta$

Theorem: Let $(x_1, y_1), \dots, (x_r, y_r)$ be trajectories of Σ such that $\mathbf{Disc}(I, \delta) = \{x_1(0), \dots, x_r(0)\}$. Let T_1, \dots, T_r be the associated TSS. Then,

$$(\forall i = 1, \dots, r. [\![\phi]\!](T_i) > \delta) \implies (\forall T \in \mathcal{L}_\tau(\Sigma). \llbracket \phi \rrbracket(T) = \top)$$

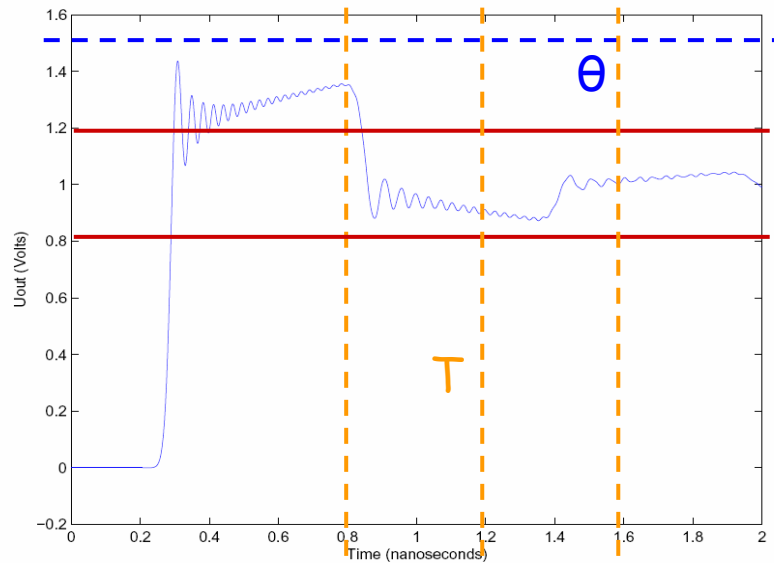
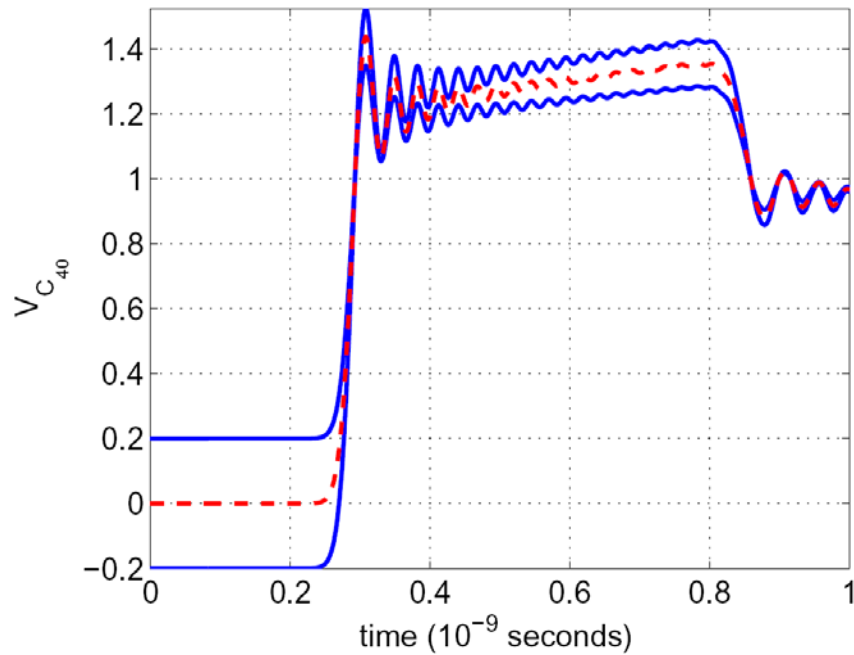
The Verification Algorithm

Algorithm 1 Temporal Logic Verification Using Simulation

Input: A dynamical system $\Sigma = (N, P, f, g, I, AP, \mathcal{O})$, an MTL formula ϕ , a sequence of time stamps τ and numbers $\delta > 0$, $r \in (0, 1)$ and $K \in \mathbb{N}$.

```
1: procedure VERIFY( $\Sigma, \phi, \tau, \delta, r, K$ )
2:    $P \leftarrow \text{Disc}(I, \delta)$ ,  $C \leftarrow \emptyset$ ,  $k \leftarrow 0$ 
3:   while  $k \leq K$  and  $P \neq \emptyset$  do
4:      $P' \leftarrow \emptyset$ 
5:     for  $x \in P$  do
6:       Pick  $\mathcal{T} \in \mathcal{L}_\tau(\Sigma)$  with  $\sigma_0 = x$  ▷ Simulate  $\Sigma$  for initial state  $x$ 
7:       if  $\llbracket \phi \rrbracket(\mathcal{T}) < 0$  then return " $\mathcal{L}_\tau(\Sigma) \not\subseteq \mathcal{L}(\phi)$ " ▷  $\phi$  does not hold on  $\Sigma$ 
8:       else if  $\llbracket \phi \rrbracket(\mathcal{T}) > r^k \delta$  then  $C \leftarrow C \cup N_V(x, r^k \delta)$ 
9:       else  $P' \leftarrow P' \cup \text{Disc}(I \cap N_V(x, r^k \delta), r^{k+1} \delta)$ 
10:      end if ▷ In lines 8,9:  $N_V(x, \delta) = \{x' \in \mathbb{R}^N \mid V(x, x') \leq \delta\}$ 
11:    end for
12:     $k \leftarrow k + 1$ ,  $P \leftarrow P'$ 
13:  end while
14:  if  $P = \emptyset$  then return " $\mathcal{L}_\tau(\Sigma) \subseteq \mathcal{L}(\phi)$ " ▷  $\phi$  holds on  $\Sigma$ 
15:  else return " $\mathcal{L}_\tau(\Sigma') \subseteq \mathcal{L}(\phi)$ " ▷  $\phi$  holds on  $\Sigma' = (N, P, f, g, I \cap C, AP, \mathcal{O})$ 
16:  end if
17: end procedure
```

from Zhi Han's PhD Thesis 2005



Experimental Results

Bisimulation function:

$$V(x_1, x_2) = \sqrt{(x_1 - x_2)^T M (x_1 - x_2)}$$

Property:

$$\phi = \square \pi_1 \wedge \diamond [0, T] \square \pi_2$$

$$\mathcal{O}(\pi_1) = [-\theta, \theta]$$

$$\mathcal{O}(\pi_2) = [0.8, 1.2]$$

	T=0.8	T=1.2	T=1.6
$\theta=1.4$	False 1	False 7	False 7
$\theta=1.5$	False 1	True 15	True 9
$\theta=1.6$	False 1	True 15	True 7



Conclusions

- ✓ LTL/MTL properties of dynamical systems in metric spaces cannot be effectively characterized by Boolean truth values
- ✓ **(Solution)** Define robust LTL/MTL semantics
 - ✓ **(Algorithms)** How do we compute robust semantics ?
 - ✓ **(Analysis)** Given trajectory, what is the maximum ϵ ?
- ✓ Model Checking and Exhaustive Verification are the holy grail, but **expensive** for Continuous and Hybrid Systems (at least for now)
 - ✓ **Light-weight verification** can help practitioners
- ✓ **(Solution)** A **new approach** to system testing using simulations
 - ✓ **(Main theme)** A system that is robust with respect to an MTL property is easier to verify!
 - ✓ If completeness is not achieved, we get coverage guarantees

Conclusions

- ✓ LTL/MTL properties of dynamical systems in metric spaces cannot be effectively characterized by Boolean truth values
- ✓ (Solution) Develop a robust MTL semantics
 - ✓ (Algorithms) Efficient algorithms for robust semantics?
 - ✓ (Analysis) Given a property, what is the maximum ϵ ?
- ✓ Model Checking and Exploration are the holy grail, but **expensive** for Continuous systems (at least for now)
 - ✓ **Light-weight verification** can help
- ✓ (Solution) A new approach to system verification: **robustness** and **coverage** calculations
 - ✓ (Main theme) A system that is robust with respect to a LTL property is easier to verify!
 - ✓ If completeness is not achieved, we get coverage guarantees

COMPLETED

Future work

LTL/MTL Semantics

- Moving from discrete time to continuous time
- Relaxing timing constraints (Robustness wrt time)
 - [Huang et al '06], [Henzinger et al '06], [Bouyer et al '05], ...
- Derive complexity bounds
- Design canonical forms
- Find topological conditions for which *rob. estimate* = *rob. degree*



Analysis

- ☑ Given $\epsilon > 0$, is a given trajectory ϵ -robust?

Robotics Application

- Design trajectories (controllers) to maximize ϵ

Testing Using Simulation

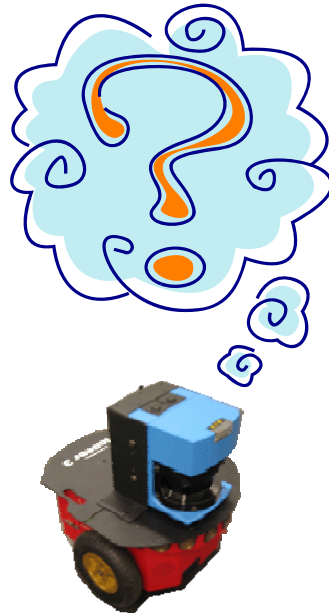
- Extend current results to
 - Hybrid Systems
 - Non-deterministic systems



Thank You!

•References

- Fainekos, Pappas: *Robustness of Temporal Logic Specifications*, in Proceedings of Formal Approaches to Testing and Runtime Verification, Aug 2006
- Fainekos, Girard, Pappas: *Temporal Logic Verification Using Simulation*, to appear in 4th International Conference on Formal Modeling and Analysis of Timed Systems, Sept 2006



Any Question(s) ?