Testing System Conformance for Cyber-Physical Systems

“Testing systems by walking the dog”

Rupak Majumdar
Max Planck Institute for Software Systems

Joint work with
Vinayak Prabhu (MPI-SWS) and Jyo Deshmukh (Toyota)
Cyber-Physical Systems

- Software controlled interactions with the (continuous) physical world

- Safety critical

- Software is the hard part
  - Expensive, brittle
  - Low productivity, High QA cost
  - Major part of development cost
The Conformance Problem

Fundamental problem in verification
- Equivalence verification for circuits
- Translation validation in compilers
The Conformance Problem

**Problem:** Given two systems, check they have equivalent behaviors

**Notion of equivalence:** Isomorphism

**Example:**
- Combinational equivalence checking of hardware circuits

**Transference of properties:**
- All properties are preserved
The Conformance Problem

**Problem:** Given two systems, check they have equivalent behaviors

**Solution:** Bisimulation or trace equivalence

**Example:**
- Sequential verification, process algebras, timed automata, …

**Transference of properties:**
- All properties in temporal logics such as CTL (bisimulation) or LTL (trace equivalence) are preserved
The Conformance Problem in CPS

Model 1: Complicated but precise dynamics

Model 2: Model-order reduced dynamics
The Conformance Problem in CPS

Model 1: Use fixed-step ODE solver
Model 2: Use dynamic-step ODE solver
The Conformance Problem in CPS

Model 1: Floating point implementation
Model 2: Fixed point implementation
The Conformance Problem

Problem: Given two systems, check they have equivalent behaviors

Challenges:
- Physical world and software may not match exactly (sensor noise, discrete modeling of continuous states, …)
- Bisimulation is too exact

Solution: ??? [This Talk]
Testing for Conformance

Test input

Model 1

Model 2

Test Generator

Bug

==

no

generate more tests

yes
Testing for Conformance

Test Generator

Model 1

Model 2

Test input

Bug

no
generate more tests

yes
Inputs and outputs are time-sampled traces of values in $\mathbb{R}^n$ completed using linear interpolation.

Simple “exact matching” does not work.

Key: Define a *metric* on traces, check if the output traces are close in the metric.
Which Metric?

1. Easily computed on traces

2. Preserves a large class of properties
Which Metric?

Strawman 1: Max of pointwise differences

\[ \sup_{t \in [0, T]} D(x(t), y(t)) \]

Over-estimates the distance due to *timing jitters*
Which Metric?

Strawman 2: Fix a finite set of (STL) properties

Check that both traces (closely) satisfy the same properties

What is a representative set of properties?
Skorokhod Metric
Pointwise distance on a rubber sheet
Skorokhod Metric
Skorokhod Metric

Timing discrepancy
Retiming functions stretch or compress time

A retiming function $r: [0,T] \rightarrow [0, T]$ is a continuous, strictly increasing, bijective map.

Metric: Compare values under a retiming
Towards Skorokhod

Retiming functions stretch or compress time

Given retiming $r$, maximize value difference:

$$\sup_{t \in [0,T]} D(x(t), y(r(t)))$$

$L_1, L_2, L_\infty$
Towards Skorokhod

Retiming functions stretch or compress time

Given retiming $r$, maximize value difference

But penalize timing discrepancies:

$$\max \left( \sup_{t \in [0, T]} |t - r(t)|, \sup_{t \in [0, T]} D(x(t), y(r(t))) \right)$$
Skorokhod Metric

Retiming functions stretch or compress time

Given retiming $r$, maximize value difference

But penalize timing discrepancies

Minimize over all retimings:

$$\inf_r \max \left( \sup_{t \in [0, T]} |t - r(t)|, \sup_{t \in [0, T]} D(x(t), y(r(t))) \right)$$
Skorokhod Metrics

\[ D_S(x, y) = \inf_{r: \text{retiming}} \max \left( \sup_{t \in [0, T]} |r(t) - t|, \sup_{t \in [0, T]} D(x(t), y(r(t))) \right) \]
Skorokhod Metrics

\[ D_S(x, y) = \inf_{r: \text{retiming}} \max \left( \sup_{t \in [0, T]} |r(t) - t|, \sup_{t \in [0, T]} D(x(t), y(r(t))) \right) \]

Not a new notion:
- Used to define a metric on cadlag functions
- Used to provide semantics to hybrid systems [Caspi, Broucke]
Skorokhod Metric: Properties

\[ D_S(x, y) = \inf_{r: \text{retiming}} \max \left( \sup_{t \in [0, T]} |r(t) - t|, \sup_{t \in [0, T]} D(x(t), y(r(t))) \right) \]

- Original trace \( y \) and retimed \( y \ast r \) have events in the same order
- A retimed trace need not be piecewise linear!
- Space of retimings is infinite
  - So it is not clear we can compute the distance
Polytime Computation

Theorem [M.Prabhu15]

1. The Skorokhod distance between two traces can be computed in time polynomial in number of dimensions of values (n) and number of time points (m)

2. There is a streaming sliding window algorithm with complexity $O(nmW)$ for window size $W$ for $L_1$, $L_2$, $L_\infty$ norms
Which Metric?

✓ Easily computed on traces
  - Fully polynomial time on traces
  - Linear time monitoring for fixed dimension and window size

2. Preserves a large class of properties
Transference of Properties

“Close systems satisfy close properties”

Timed (Quantitative) Linear Temporal Logic = LTL + Freeze quantifiers + Value predicates

\[ z_1 \cdot (|v_1| < 5) \rightarrow \diamond z_2 \cdot ((v_2^2 + v_3^2 \in [3, 7]) \land (z_1^2 + z_2^2 \leq 16)) \]
TLTL: Expressiveness

Subsumes Metric temporal logic and Signal temporal logic

\[ pU_{[a,b]}q \equiv x. (pUy. ((y \leq x + b) \land (y \geq x + a) \land q)) \]
Transference Theorem

Theorem: [DeshmukhM.Prabhu] There is a function $r_x^\delta$ such that for every TLTL formula $\phi$,
if trace $\sigma_1$ satisfies $\phi$ and Skorokhod metric between $\sigma_1$ and $\sigma_2$ is at most $\delta$,
then $\sigma_2$ satisfies $r_x^\delta(\phi)$

$r_x$ “expands” distances by $\delta$ but maintains LTL structure

$pU_{[a,b]}q$ expands to $pU_{[a-2\delta,b+2\delta]}q$
Which Metric?

- Easily computed on traces
  - Fully polynomial time on traces
  - Linear time monitoring for fixed dimension and window size

- Preserves a large class of properties
Simulink Conformance Tester

S-Taliro: Test generation based on gradient ascent

Test input

Simulink Model 1

Simulink Model 2

Compute Sk. dist. and compare to tolerance

dist high?

Bug

generate more tests
Case Studies

1. LQR control for aircraft pitch control
   A. Continuous-time model
   B. Digital implementation with sensor delay

2. Air-fuel ratio controller for an ECU (from Toyota)
   A. Continuous time nonlinear model
   B. Polynomial approximation to the nonlinear dynamics (but without formal guarantees)

3. Engine block model with numerical integrators
   A, B. Two different integration procedures
Toyota Air-Fuel Ratio Controller

An industrial challenge benchmark from Toyota

In simulations, the two models were found to be “close” w.r.t. a pre-selected set of properties.

Our tool found an input with high Skorokhod distance (relative to the engineering tolerance).

Time horizon 10s, 300 time points, 8 min total, 4 min simulation time.
What does this have to do with dog walking?
Algorithms for Skorokhod Metrics

How can you compute the Skorokhod metric between two finite traces?

\[ D_S(x, y) = \inf_{r: \text{retiming}} \max \left( \sup_{t \in [0, T]} |r(t) - t|, \sup_{t \in [0, T]} D(x(t), y(r(t))) \right) \]

1. Space of retimings is infinite

2. Retimed traces may be very complicated (not even polynomial)
Fréchet Metric
Skorokhod and Fréchet

\[
\mathcal{D}_S(x, y) = \inf_{r: \text{retiming}} \max \left( \sup_{t \in [0, T]} |r(t) - t|, \sup_{t \in [0, T]} \mathcal{D}(x(t), y(r(t))) \right)
\]

\[
\mathcal{D}_F(f, g) = \inf_{\alpha_f: [0,1] \rightarrow [0, T], \alpha_g: [0,1] \rightarrow [0, T]} \max_{0 \leq \theta \leq 1} \left\| f(\alpha_f(\theta)) - g(\alpha_g(\theta)) \right\|
\]

Trick to add reparameterization penalty:
- Add current time as a new component to the state
- Compare states using a combination of max-norm and \( \mathcal{D} \)
Reduce Skorokhod to Fréchet

Define the Dmax distance:

\[ D_{\text{max}}((x, tx), (y, ty)) = \max(D(x, y), |tx - ty|) \]

Fréchet Metric

\[ D_S(x, y) = \inf_{r: \text{retiming}} \max \left( \sup_{t \in [0, T]} |r(t) - t|, \sup_{t \in [0, T]} D(x(t), y(r(t))) \right) \]

\[ D_F(f, g) = \inf_{\alpha_f: [0,1] \rightarrow [0, T]} \max_{0 \leq \theta \leq 1} \| f(\alpha_f(\theta)) - g(\alpha_g(\theta)) \| \]

\( x: (x, tx) \rightarrow f \) and \( y: (y, ty) \rightarrow g \)

Define the Dmax distance:

\[ D_{\text{max}}((x,tx), (y,ty)) = \max(D(x,y), |tx - ty|) \]

\[ D_S(x,y) = D_F^{\text{max}}(f, g) \]
Computing the Fréchet Distance

1. Decision problem: Given $f$, $g$, and $\delta$, check if $D(f,g) \leq \delta$

2. Characterize a finite set of “critical” delta values and compute them for each geometry ($L_1$, $L_2$, $L_\infty$)

3. Binary search over this set
The Decision Problem

[AltGodau95] Reduce the problem to a two dimensional geometric problem

- Pairwise comparison of linear segments

- Key Step: *Free space diagram*

- [AltGodau95] did this for $L_2$ and $\mathbb{R}^2$

- We extend it to $L_1$, $L_2$, $L_\infty$ and $\mathbb{R}^n$
Free Space Diagram

\[ \text{Free}_\delta(f, g) = \left\{ (\rho_f, \rho_g) \in [0, T]^2 \text{ such that } \|f(\rho_f) - g(\rho_g)\| \leq \delta \right\} \]

Positions in the two curves where values differ by at most \( \delta \)

If there is a monotone increasing path from \((0,0)\) to \((T,T)\)

Then \( \mathcal{D}(f, g) \leq \delta \)
Positions in the two curves where values differ by at most $\delta$

If there is a monotone increasing path from $(0,0)$ to $(T,T)$

Then $D(f, g) \leq \delta$

Increasing in both parameters: can traverse the two curves without reversing, while ensuring value difference is at most $\delta$
Free Space as a Product

Free space for affine segment pairs: \((\rho_f, \rho_g) \in \text{Free}_\delta(f[i], g[j])\)
if
\[ f(\rho_f) \text{ in } i\text{-th affine segment.} \]
\[ g(\rho_g) \text{ in } j\text{-th affine segment.} \]

Free space for affine segment pairs:
\[
\text{Free}_\delta(f, g) = \bigcup \text{Free}_\delta(f[i], g[j]).
\]

Suffices to analyze pairs of linear segments.

Free space for affine segment pairs:
\[
\text{Free}_\delta(f[i], g[j]) \text{ is convex.}
\]

Compute \(\text{Free}_\delta(f[i], g[j])\) only at cell boundaries.
Computing $\text{Free}_\delta(f_i, g_j)$ at Boundaries

Geometric primitives depending on the metric

$\mathbb{R}^n$: $L^\text{max}_2$ quadratic equations.

$L^\text{max}_\infty$ intersection inequalities.

$L^\text{max}_1$ naive gives exponential.

We do it in $O(n^2)$. 

$\mathbb{R}^n$: Decide $\mathcal{D}_\mathcal{F}(f, g) \leq \delta$?

- $L^\text{max}_2, L^\text{max}_\infty$ in $O(nm^2)$.
- $L^\text{max}_1$ in $O(n^2m^2)$.

$m$ is number of affine segments.

- Sliding window based: Linear in $m$. 

19 / 28
Critical Values

\(\delta\) is critical if \(\text{Free}_\delta(fi, gj)\) becomes non-empty at a cell boundary.

Geometrically, \(\delta\) for which a horizontal line can go from cell \((i,j)\) to cell \((k,j)\).

Reduces to geometric primitives
Computable in polynomial time.
**Theorems: Skorokhod Metric**

**Compute Distance: Polygonal Traces in $\mathbb{R}^n$**

- $L_2$: $O(m^3 (n + \log(m)))$.
- $L_1, L_\infty$: $O(m^3 (\text{poly}(n) + \log(m)))$.

$m$ is number of affine segments.

**Monitor Polygonal $\mathbb{R}^n$ Traces: Decide $D_S(x, y) \leq \delta$?**

- $L_2, L_\infty$ in $O(nm^2)$.
- $L_1$ in $O(n^2 m^2)$.
- Sliding window based: **Linear in $m$**.

And that’s how dog walking applies to safe CPS…
Extensions: Tubes

Given two sets $F_1$, $F_2$ of trajectories, define

$$\mathcal{D}(F_1, F_2) = \sup_{f_1 \in F_1, f_2 \in F_2} \mathcal{D}_S(f_1, f_2)$$

Can we compute the distance between two sets of trajectories?

In practice, we get *reachability tubes* that over-approximate $F_1$ and $F_2$
Extensions: Tubes

Theorem [M.Prabhu16]

Given polygonal reachability tubes $F_1$ and $F_2$, and parameter $\varepsilon$, one can compute lower and upper bounds on $D(F_1, F_2)$

With bound $\varepsilon$ in polynomial time in $F_1$, $F_2$, and $\varepsilon$
Conclusion

Skorokhod distances provide a quantitative generalization of trace equivalence that is well-suited to cyber-physical systems

- Tractable to compute between traces
- Preserves logical properties approximately
Thank You

http://www.mpi-sws.org/~rupak/

References:
HSCC 2015, CAV 2015, HSCC 2016
End-to-End Arguments

Carry the *mathematical arguments* for correctness for control systems down to *software implementations*

Control Theory + Program Analysis = Reliable Embedded Systems