On Systematic Simulation of Open Systems

Oded Maler

CNRS-VERIMAG

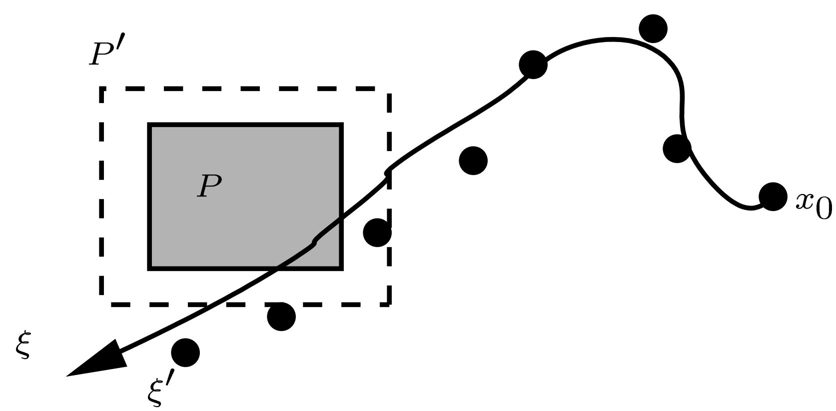
Grenoble, France

Joint work with Jim Kapinksi, Bruce Krogh (CMU) and Olaf Stursberg (Dortmund)

A Problem

Given $\dot{\mathbf{x}} = f(\mathbf{x})$ in $\mathbb{R}^n$ and $\mathbf{x}_0$, is there some time where the trajectory reaches a set P ?

Postulate 1. [Simulation is Fine] *An intelligent mortal can solve the reachability problem for a closed continuous system using a finite amount of numerical simulation.*

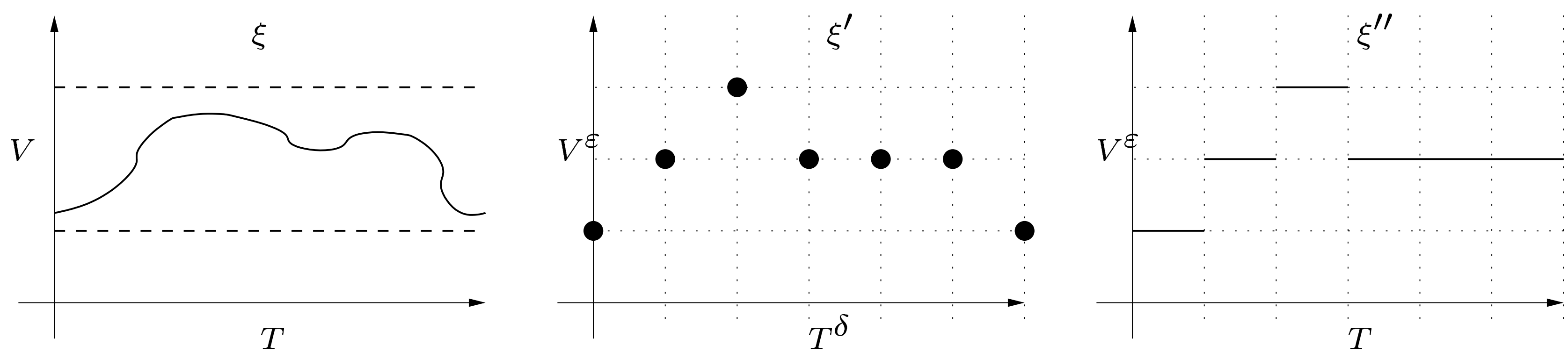


The Problem

How to do it for open systems $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{v})$?

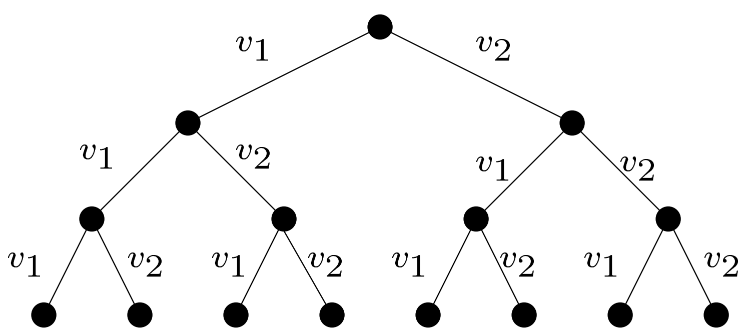
An input signal is a function $\psi : T \to V$.

For each given signal it is reduced to the simulation of a closed system but how to do it exhaustively (for $(2^{\aleph_0})^{2^{\aleph_0}}$ elements)? The first step is time and space discretization.

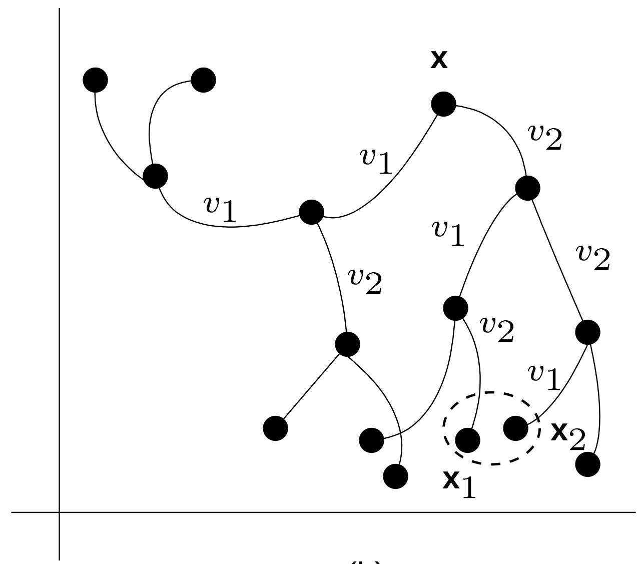


Vive Le Monoïde Libre

The set of all discretzied signals is one of the basic objects of CS, the set V^* of all sequences over a finite alphabet, . the free monoid generated by V. Notation: $v_1v_2v_1 \cdot v_1v_1 = v_1v_2v_1v_1v_1$. Can be viewed as a tree:



(a)



(b)

Simulation-Based Reachability

Notation: the action of a sequence $\psi \in \bar{V}^*$ on a state $\mathbf{x}$: as $\mathbf{x} \cdot \psi = \mathbf{x}'$

The set of points reachable from $\mathbf{x}$ at time k is

$$R^k(\mathbf{x}) = \{\mathbf{x} \cdot \psi : \psi \in \bar{V}^k\}$$

Points reachable until time k is

$$R^{\leq k}(\mathbf{x}) = \{\mathbf{x} \cdot \psi : \psi \in \bar{V}^{\leq k}\}$$

Semigroup property $\mathbf{x} \cdot (\psi \cdot v) = (\mathbf{x} \cdot \psi) \cdot v$ allows us to “re-use” partial simulations.

Simulation-Based Reachability - the Algorithm

Algorithm 1. [Reachability for Discretized Input Signals]

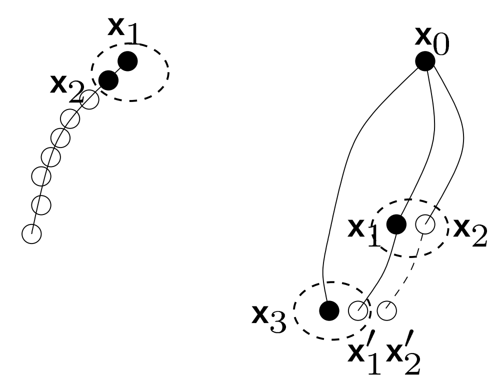
```
Reached:=New:=∅;
Waiting:={x_0};
Repeat k = 0, 1, ...
  For each x ∈ Waiting
    For each v ∈ V̄
      Compute x' = x · v;
      Insert x' into New;
    Move x from Waiting to Reached;
  Waiting:=New; New:=∅;
Forever
```

Exponential in bounded time, number of simulation grows indefinitely.

Reducing the Number of Simulations

Observation from automata: if $\mathbf{x} \cdot \psi_1 = \mathbf{x} \cdot \psi_2$
then for every v: $\mathbf{x} \cdot (\psi_1 \cdot v) = \mathbf{x} \cdot (\psi_2 \cdot v)$ So we need not simulate with extensions of ψ_2.

In Continuous systems equality is rare and should be replaced by closeness.



(a) (b)

Operations on Neighborhoods

Neighborhood: $N(\mathbf{x}) = \{\mathbf{x}' : \rho(\mathbf{x}, \mathbf{x}') \leq \varepsilon\}$

The action of input v on a neighborhood $N(\mathbf{x})$ is

$$N(\mathbf{x}) \cdot v = N'(\mathbf{x}')$$

where $\mathbf{x}' = \mathbf{x} \cdot v$ and v induces a homeomorphism between $N(\mathbf{x})$ and $N'(\mathbf{x}')$.

Two over-approximation operations:

$$Next(N(\mathbf{x}), v) \supseteq N(\mathbf{x}) \cdot v$$

$$Join(N_1(\mathbf{x}_1), N_2(\mathbf{x}_2)) \supseteq N_1(\mathbf{x}_1) \cup N_2(\mathbf{x}_2)$$

Simulation-Based Reachability with Neighborhoods

Reached:=New:=$\emptyset$;
Waiting:=$\{N(\mathbf{x}_0)\}$;
Repeat $k = 0, 1, \ldots$
  For each $N(\mathbf{x}) \in$ Waiting
    For each $v \in \bar{V}$
      Compute $N'(\mathbf{x}') = Next(N(\mathbf{x}), v)$;
      If $\mathbf{x}' \in \hat{N}(\hat{\mathbf{x}})$ for some $\hat{N}(\hat{\mathbf{x}}) \in$ Reached $\cup$ New
      Then
        Compute $N^*(\mathbf{x}^*) = Join(N'(\mathbf{x}'), \hat{N}(\hat{\mathbf{x}}))$
        If $N^*(\mathbf{x}^*) \neq \hat{N}(\hat{\mathbf{x}}))$ **Then**
          Insert $N^*(\mathbf{x}^*)$ into New
          Remove $\hat{N}(\hat{\mathbf{x}})$ from Reached $\cup$ New
      Else
        Insert $N'(\mathbf{x}')$ into New;
    Move $N(\mathbf{x})$ from Waiting to Reached;
  Waiting:=New; New:=$\emptyset$;
Forever

For every k, $R^{\leq k}(\mathbf{x}_0) \subseteq \bigcup_{N(\mathbf{x}) \in Reach} N(\mathbf{x})$ holds at the end of the k^{th} iteration of the main loop.

Concrete Implementation - I

Stable linear systems in discrete time: $\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{v}_k$

$$N_r(\mathbf{x}) = \{\mathbf{x}' : (\mathbf{x}' - \mathbf{x})^T M(\mathbf{x}' - \mathbf{x}) \leq r^2\}$$

where M is the solution of $A^T MA - M = -I$.

$$Next(N_r(\mathbf{x}), v) = N_{r'}(\mathbf{x}')$$

with $\mathbf{x}' = \mathbf{x} \cdot v$ and $r' = \alpha r$ where

$$\alpha = \sqrt{\frac{\hat{\lambda} - 1}{\hat{\lambda}}}$$

and $\hat{\lambda}$ is the largest eigenvalue of M.

Concrete Implementation - II

$$Join(N_{r_1}(\mathbf{x}_1), N_{r_1}(\mathbf{x}_2)) = \begin{cases} N_{r_1}(\mathbf{x}_1) & \text{if} \quad r_1 \geq r_2 \wedge \hat{r} \leq (r_1 - r_2) \\ N_{r_2}(\mathbf{x}_2) & \text{if} \quad r_2 > r_1 \wedge \hat{r} \leq (r_2 - r_1) \\ N_{r^*}(\mathbf{x}^*) & \text{otherwise.} \end{cases}$$

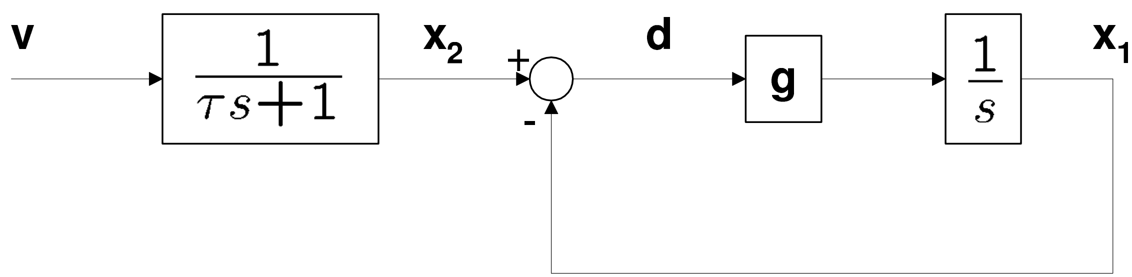
$$r^* = \frac{\hat{r} + r_1 + r_2}{2}$$

$$\mathbf{x}^* = \beta \mathbf{x}_1 + (1 - \beta)\mathbf{x}_2$$

$$\beta = \frac{r^* - r_2}{\hat{r}}$$

$$\hat{r}^2 = (\mathbf{x}_1 - \mathbf{x}_2)^T M (\mathbf{x}_1 - \mathbf{x}_2)$$

Example: Servo I



$$A = \begin{bmatrix} -g & g \\ 0 & -\frac{1}{\tau} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \frac{1}{\tau} \end{bmatrix}$$

Input space: $\{0.0, 0.5, 1.0\}$, want to avoid $P = |x_1 - x_2| > 1$

Example: Servo II

In discrete time:

$$\mathbf{x}_{k+1} = \Phi \mathbf{x}_k + \Gamma v_k$$

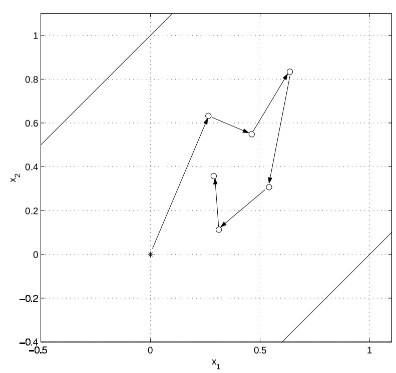
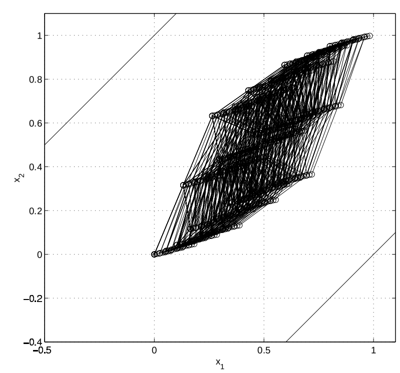
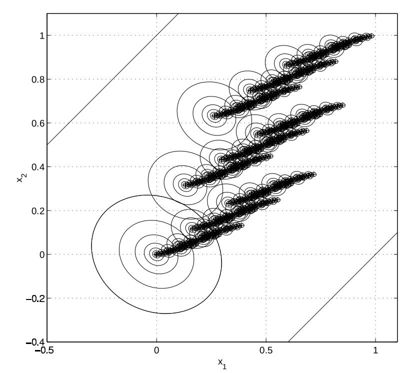
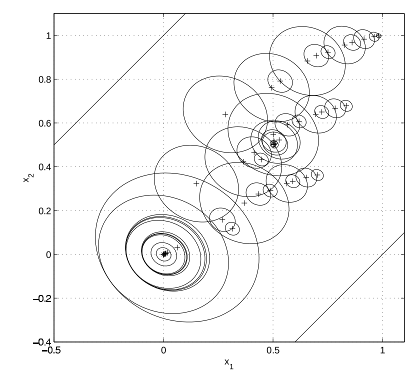
$$\Phi = e^{Ap} \qquad \Gamma = A^{-1}(e^{Ap} - I)B$$

For $g = 10$, $\tau = 0.1$, and $p = 0.1$:

$$\Phi = \begin{bmatrix} 0.368 & 0.368 \\ 0.000 & 0.368 \end{bmatrix} \qquad \Gamma = \begin{bmatrix} 0.264 \\ 0.632 \end{bmatrix}$$

Experimental Results

We apply 6 iterations. The exhaustive methods requires 1092 simulations. Our method requires only 231 and keeps only 48 elements in $Reach$.

$\psi = 1.0 \cdot 0.5 \cdot 1.0 \cdot 0.0 \cdot 0.0 \cdot 0.5$