Test Coverage for Continuous and Hybrid Systems

Tarik Nahhal and Thao Dang,
VERIMAG
Grenoble, France

Introduction

- **Hybrid systems**: appropriate high-level model for **embedded systems**
- **Testing**: commonly-used validation method in industry; it suffers less from the 'state explosion' problem and can be applied to the real system and not only to its model.
- **Testing of a reactive system**: control the inputs and check whether the corresponding behaviors are as expected.
- Infiniteness of the admissible input space of a hybrid system $\Rightarrow$ notion of coverage
- In **software testing**, syntactic coverage measures, such as statement coverage and if-then-else branch coverage, path coverage

Plan

1. Introduction: Hybrid systems testing problem
2. Test coverage
3. Coverage-guided test generation
4. Experimental results

Plan

1. **Introduction: Hybrid systems testing problem**
2. Test coverage
3. Coverage-guided test generation
4. Experimental results

Hybrid Automata

- $\mathcal{X} \subseteq \mathbb{R}^n$ is the **continuous state space**
- A set Q of **discrete locations**. In location q, the evolution of the continuous variables: $f_q(x(t), \dot{x}(t), u(t), p) = 0$ where $u(t) \in U_q$ (**input set**), $p \in W_q$ (**parameter set**). $\mathcal{I}_q \subseteq \mathbb{R}^n$ is the **staying condition** of location q.
- A set of $E \subseteq Q \times Q$ of **discrete transitions**. A discrete transition $e = (q, q')$, $\mathcal{G}_e \subseteq \mathcal{I}_q$ specifies the guard condition and $\mathcal{R}_e$ is the associated reset map.
- A **hybrid state** (q, x) can change in 2 ways: by **continuous evolution** and by **discrete evolution**
- This model allows to capture **non-determinism**

Testing Problem

- A **system under test** (modeled by a hybrid automaton) often operates within some environment.
- The **tester** plays the role of the **environment**. The tester generates the continuous inputs and control discrete transition.
- Implement the tester as a computer program $\Rightarrow$ continuous inputs are assumed to be piecewise-constant with a fixed period h (**time step**).
- Hence, there are two types of **control actions** the tester can perform: **continuous** (such as (f_q, u)) and **discrete** (such as (q, q')).

Specification and System under Test

We assume that the **specification** is modeled as a hybrid automaton $\mathcal{A}$ and the **system under test** (such as an implementation) by another hybrid automaton $\mathcal{A}_s$ such that:

- The **discrete states** of $\mathcal{A}_s$ is **observable**.
- Concerning the sets of **observable continuous variables**: $V_o(\mathcal{A}) \subseteq V_o(\mathcal{A}_s)$ and
- Concerning the sets of all **admissible control action sequences**: $S_{\mathcal{C}}(\mathcal{A}) \subseteq S_{\mathcal{C}}(\mathcal{A}_s)$.

Note: we do not assume that we know the model $\mathcal{A}_s$.

Conformance

Given an admissible control sequence γ

- $S_{\mathcal{O}}(\mathcal{A}, \gamma)$ is the set of **observation sequences of the specification $\mathcal{A}$**
- $\pi(S_{\mathcal{O}}(\mathcal{A}_s, \gamma))$ is the set of **observation sequences of the system under test** $\mathcal{A}_s$ under γ **projected** on the observable variables of $\mathcal{A}$.

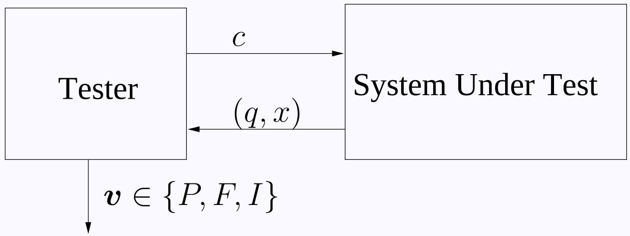
We say that the system under test $\mathcal{A}_s$ is **conform** to the specification $\mathcal{A}$, denoted by $\mathcal{A} \approx \mathcal{A}_p$, iff for all admissible control sequences γ

$$\pi(S_{\mathcal{O}}(\mathcal{A}_s, \gamma), V_o(\mathcal{A})) \subseteq S_{\mathcal{O}}(\mathcal{A}, \gamma).$$

Note: a control sequence which is admissible for the specification $\mathcal{A}$ is also admissible for the system under test $\mathcal{A}_s$.

Test case

Test case: **tree** where each **node** is associated with an **observation** and each **edge** is associated with a **control action**.



The observation sequences of the trees are grouped into three disjoint sets:

- the set O_p of observation sequences that cause a "pass" verdict
- the set O_f that cause a "fail" verdict
- the set O_i that cause an "inconclusive" verdict.

Infinite number of infinite traces $\Rightarrow$ select a finite portion of the input space of the specification $\mathcal{A}$ and test the conformance of $\mathcal{A}_s$ w.r.t. this portion.

The selection is done using a **coverage criterion** (see next).

Plan

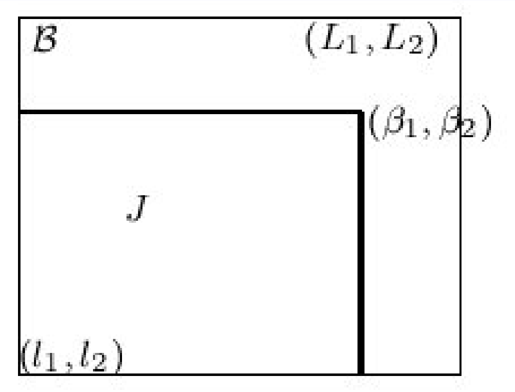
1. Introduction: Hybrid systems testing problem
2. **Test coverage**
3. Coverage-guided test generation
4. Experimental results

Test coverage

- **Test coverage** is a way to evaluate testing quality.
- We are interested in **state coverage** and focus on a measure that describes how 'well' the visited states represent the reachable set.
- This measure is defined using the **star discrepancy** notion in statistics, which characterises the uniformity of the distribution of a point set within a region.
- The star discrepancy is an important notion in equidistribution theory as well as in quasi-Monte Carlo techniques

Star discrepancy

- Let P be a **set of k points** inside $\mathcal{B} = [l_1, L_1] \times \ldots \times [l_n, L_n]$.
- A **subbox** $J = \prod_{i=1}^{n} [l_i, \beta_i]$ with $\beta_i \in [l_i, L_i]$. Let Γ be the set of all such subboxes
- The **local discrepancy**: $D(P, J) = |\frac{A(P,J)}{k} - \frac{\lambda(J)}{\lambda(\mathcal{B})}|$ where $A(P, J)$=number of points inside J, and $\lambda(J)$=volume of J.
- The **star discrepancy**: $D^*(P, \mathcal{B}) = sup_{J \in \Gamma} D(P, J)$. Note that $0 < D^*(P, \mathcal{B}) \leq 1$.



Test Coverage for Hybrid Systems

- Let $\mathcal{P} = \{(q, P_q)\}$ be the set of states. We define the **coverage** of $\mathcal{P}$ as:

$$Cov(\mathcal{P}) = \frac{1}{||Q||} \sum_{q \in Q} 1 - D^*(P_q, \mathcal{I}_q)$$

where $||Q||$ is the number of locations in Q.

- A large value of $Cov(\mathcal{P})$ indicates a good **space-covering** quality. If $\mathcal{P}$ is the set of states visited by a test suit, our objective is to maximize $Cov(\mathcal{P})$.

Plan

1. Introduction: Hybrid systems testing problem
2. Test coverage
3. **Coverage-guided test generation**
4. Experimental results

Test generation

Essence behind the solution we propose

- **Randomized** exploration, inspired by probabilistic **motion planning** techniques **RRT** (Random Rapidly-Exploring Trees) in robotics
- **Coverage criteria** reflects testing quality
- **Guided** by coverage criteria

Test generation algorithm

$\mathcal{T}.init(s_0)$, $j = 1$ /* s_0: initial state */

Repeat

  $s_{goal} = \textsc{Sampling}(\mathcal{S})$ /* $\mathcal{S}$: hybrid state space */

  $s_{near} = \textsc{Neighbor}(\mathcal{T}, s_{goal})$

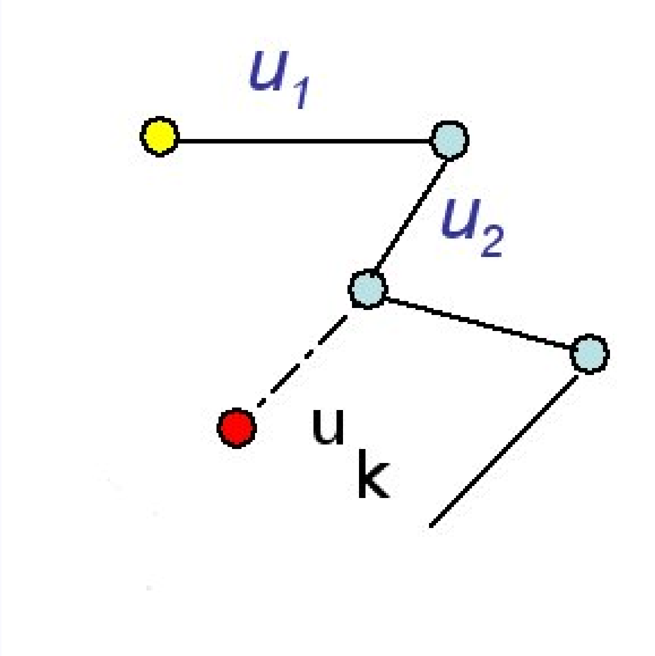
  $(s_{new}, u_{q_{near}}) = \textsc{ContinuousStep}(s_{near}, h)$ /* h: time step */

  $\textsc{DiscreteSteps}(\mathcal{T}, s_{new})$, $j{+}{+}$

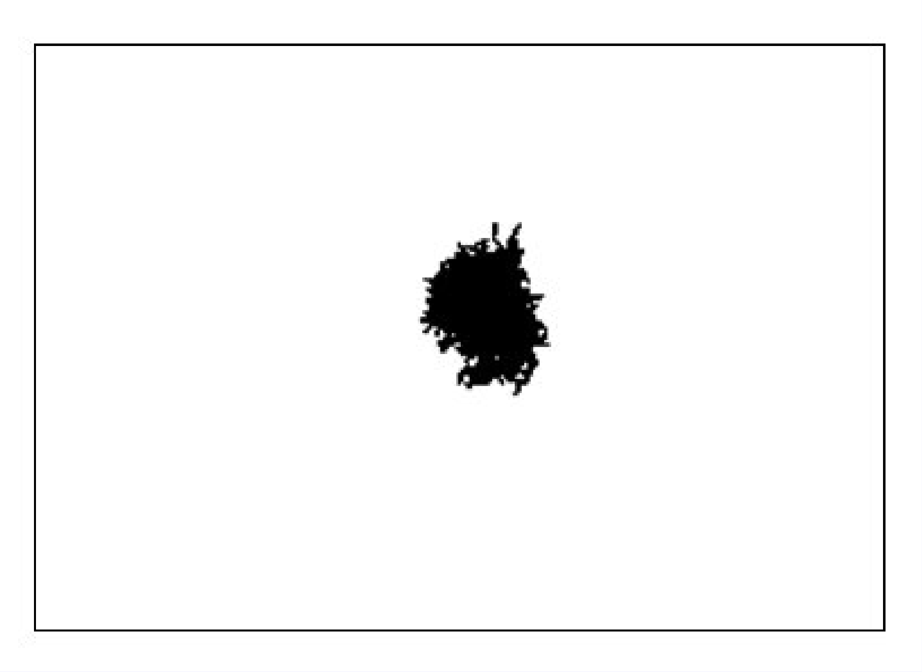
Until $j \geq J_{max}$

- It is natural to choose s_{near} to be a state near s_{goal} $\Rightarrow$ We need to define the distance between hybrid states.
- The procedure ContinuousStep tries to find the input $u_{q_{near}}$ to take the system from s_{near} towards s_{goal} as closely as possible.
- In the classic (continuous) RRT algorithms, sampling is often uniform, Neighbor is defined using the Euclidian distance

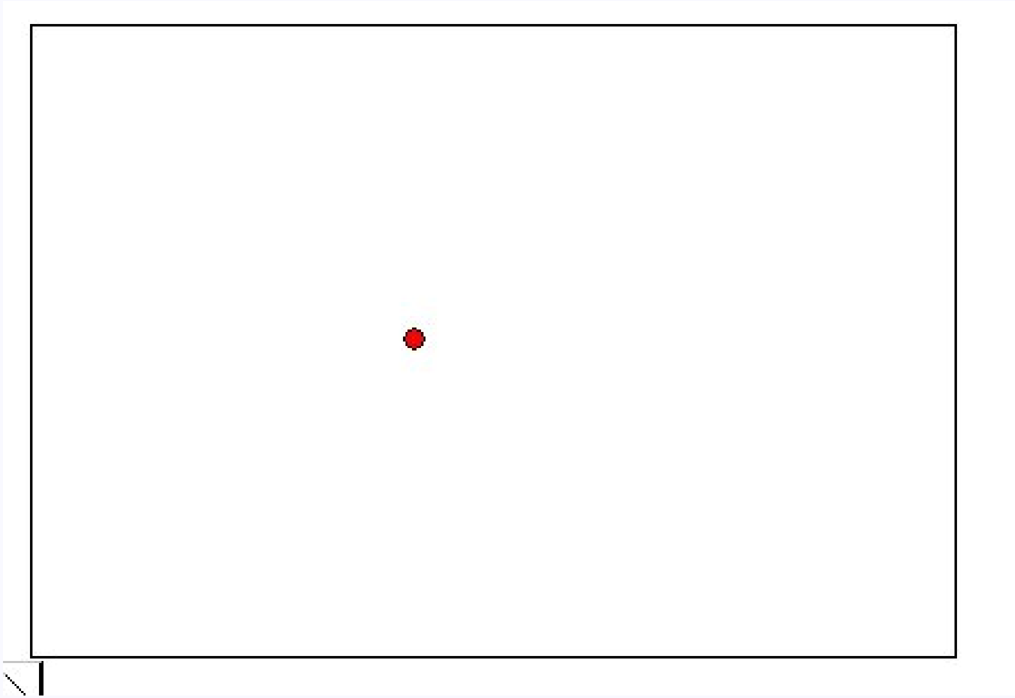
Simple randomized exploration



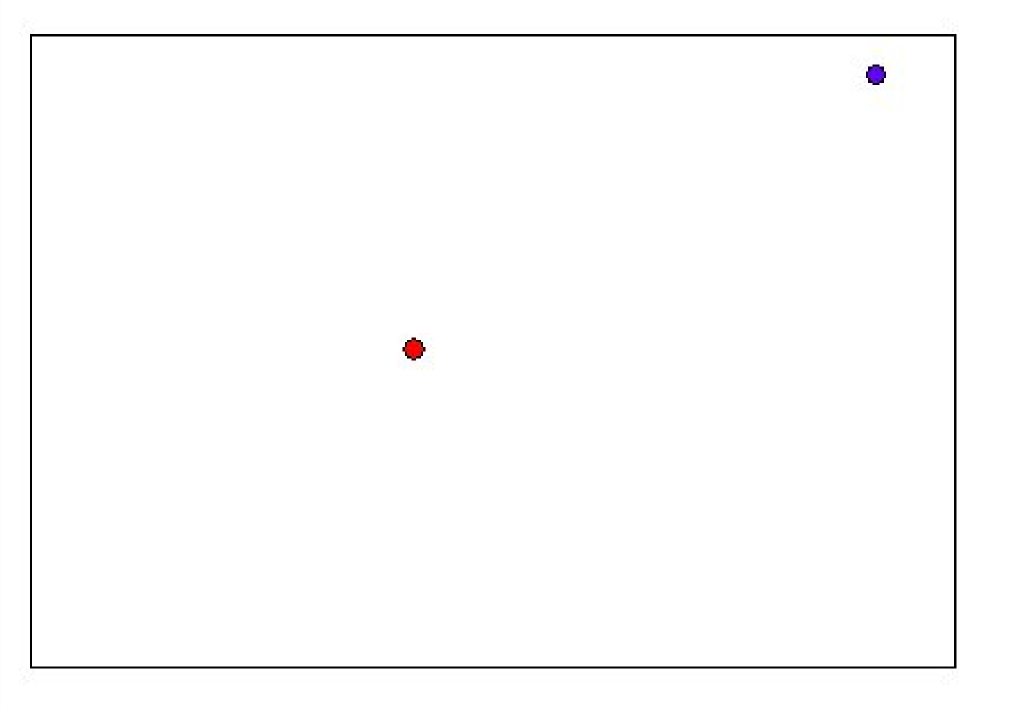
RRT-based exploration



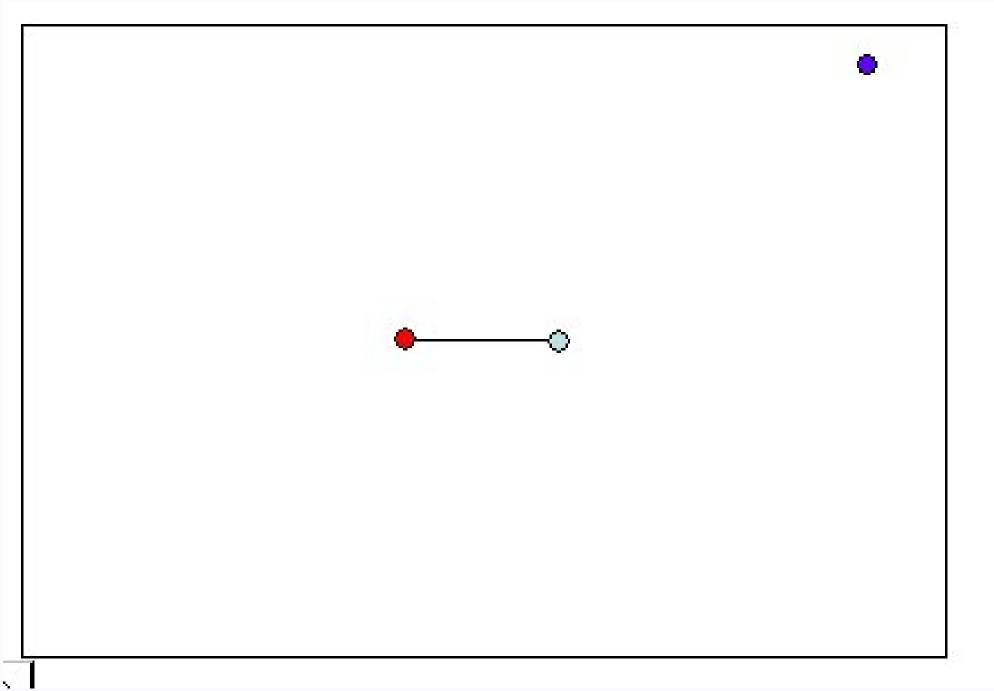
RRT simulation



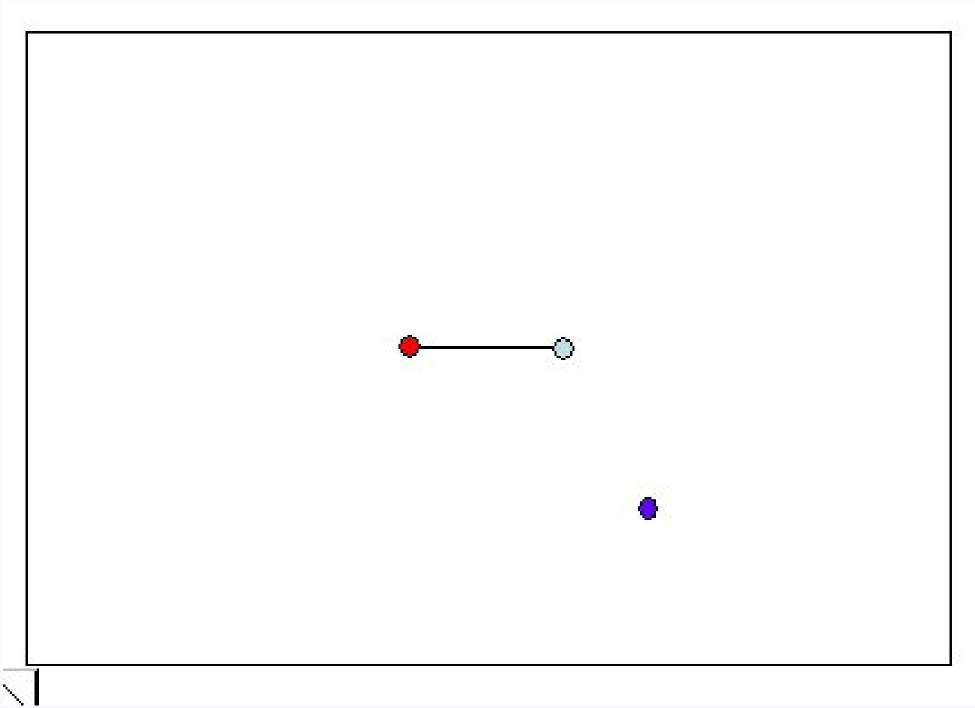
RRT simulation



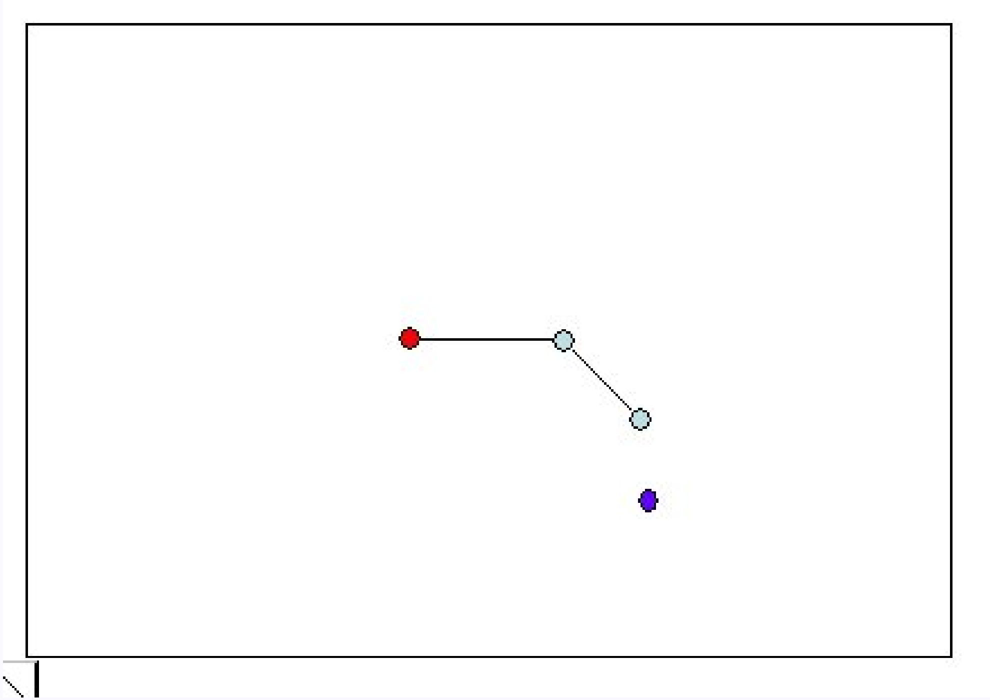
RRT simulation



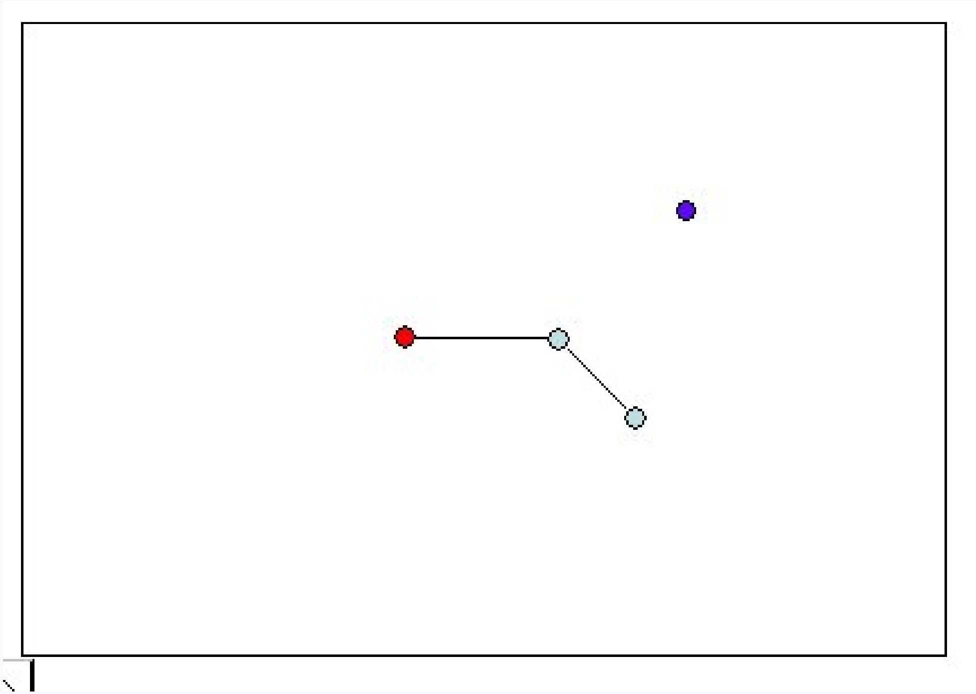
RRT simulation



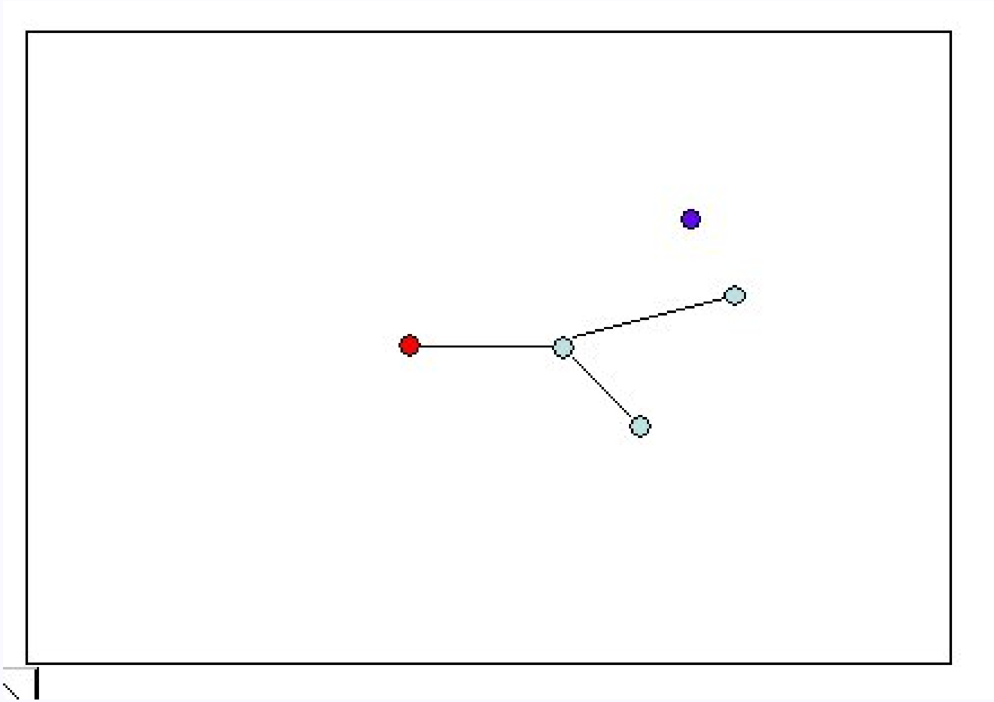
RRT simulation



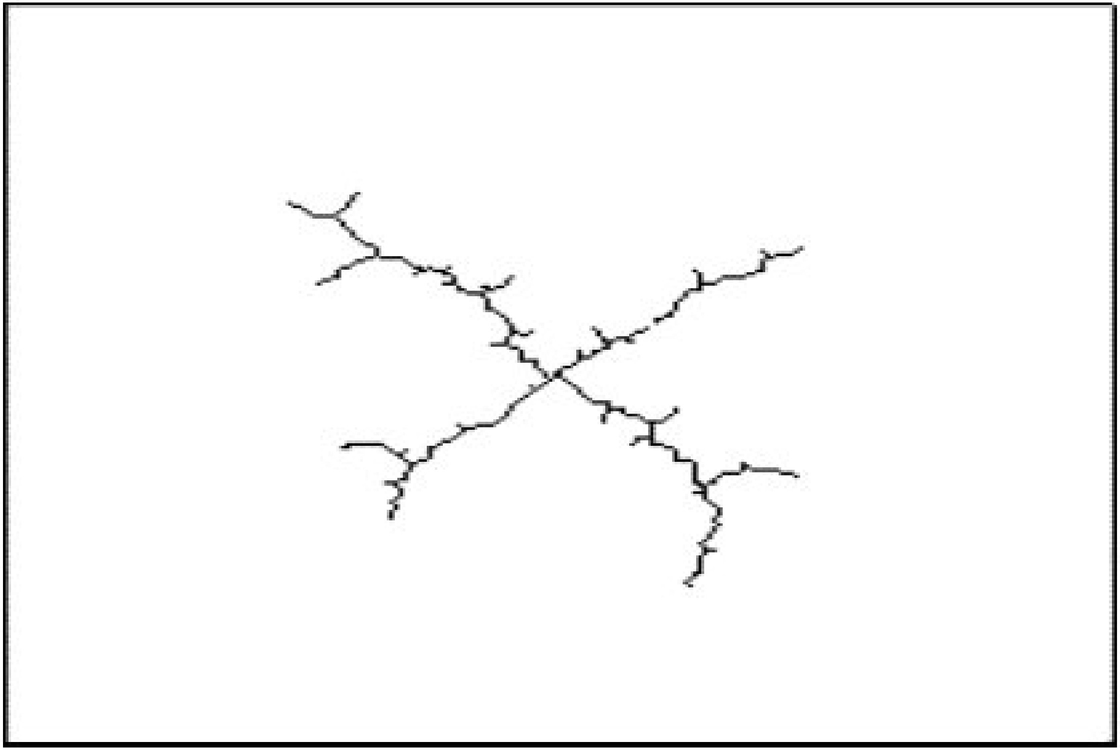
RRT simulation



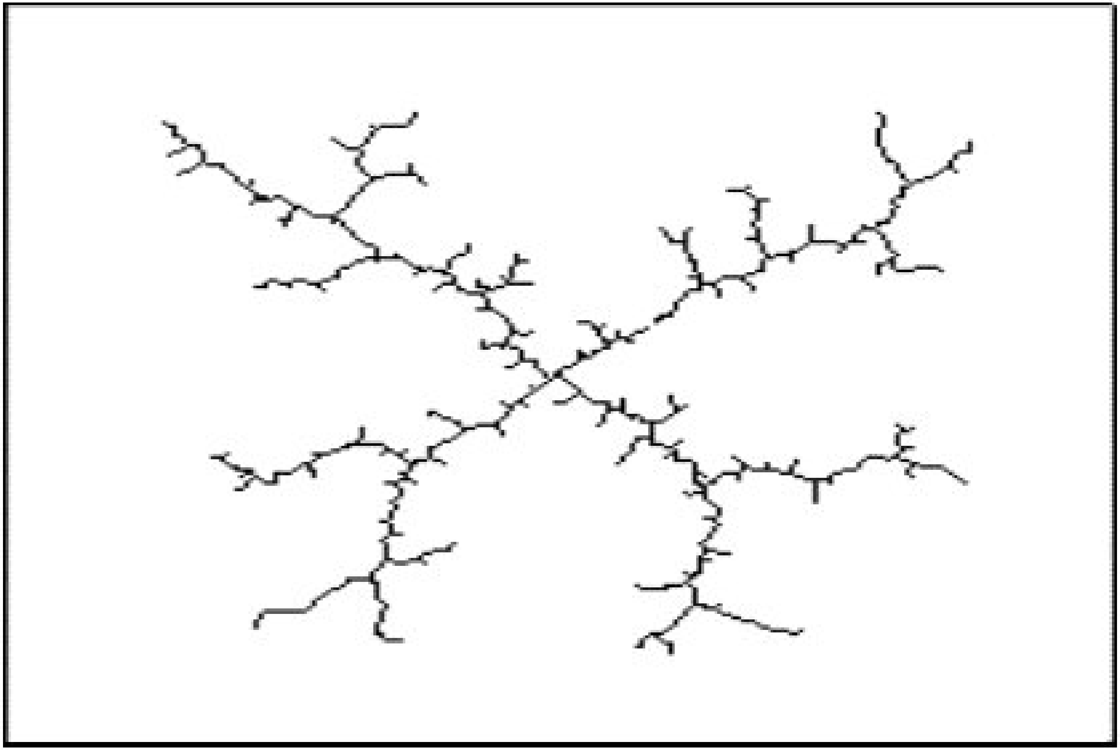
RRT simulation



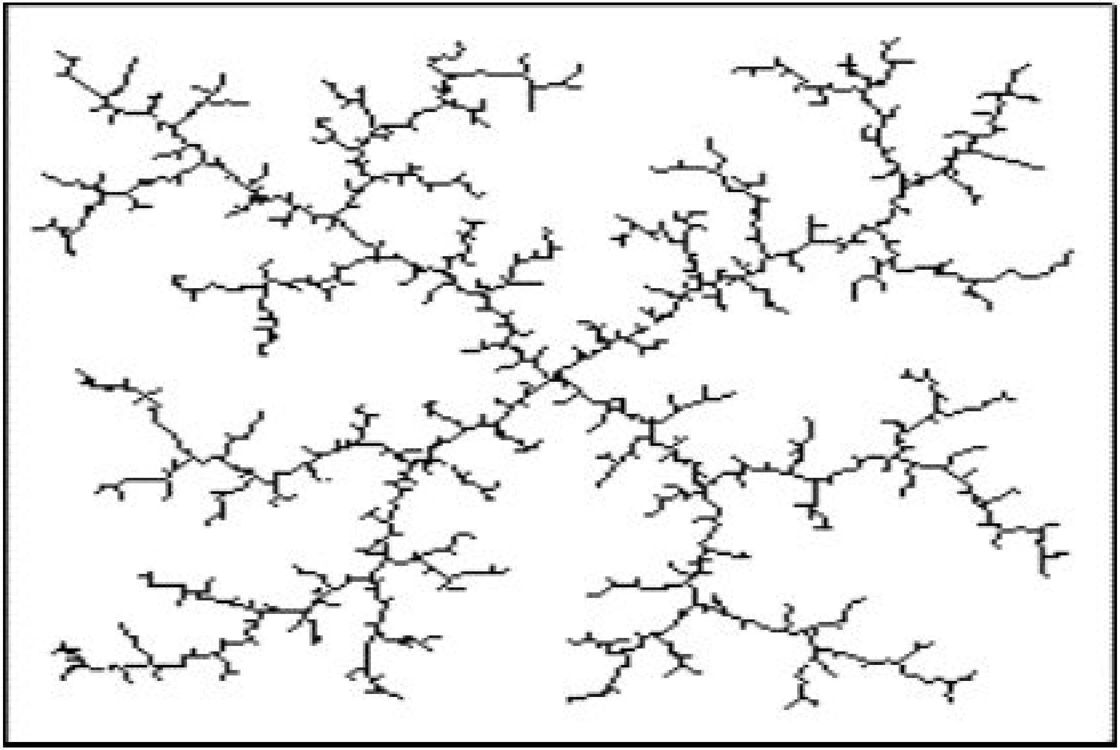
RRT simulation - example



RRT simulation - example



RRT simulation - example



Hybrid distance

- Two transitions $e = (q, q')$ and $e' = (q', q'')$, we define $\sigma(e, e') = \overline{d}(\mathcal{R}_{(l,l')}(\mathcal{G}_{(l,l')}), \mathcal{G}_{(l',l'')})$ where $\overline{d}$ is the Euclidian distance between their centroids.
- A path $\gamma = e_1, e_2, \ldots e_m$, **average length** $len(\gamma) = \sum_{i=1}^{m-1} \sigma(e_i, e_{i+1})$.
- Two **hybrid states** $s = (q, x)$ and $s' = (q', x')$,
  - if $q = q'$, the **hybrid distance** $d_H(s, s')$ is the Euclidian distance between x and x': $d_H(s, s') = ||x - x'||$.
  - If $q \neq q'$,

$$d_H(s, s') = \begin{cases} \min_{\gamma \in \Gamma(q,q')} \overline{d}(x, fG(\gamma)) + len(\gamma) + \overline{d}(x', lR(\gamma)) & \text{if } \Gamma(q, q') \neq \\ \infty & \text{otherwise.} \end{cases}$$

$fG(\gamma) = \mathcal{G}_{(l_1,l_2)}$ (first guard), and $lR(\gamma) = \mathcal{R}_{(l_k,l_{k+1})}(\mathcal{G}_{(l_k,l_{k+1})})$.

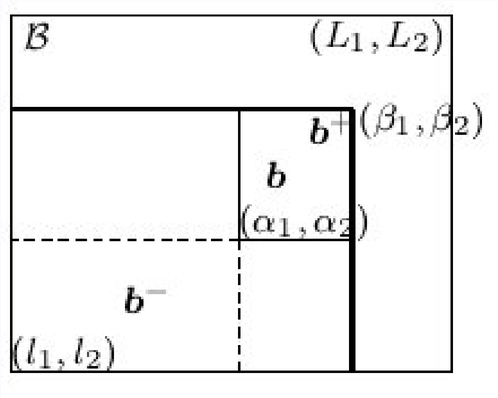
- NEIGHBOR can then be computed using this hybrid distance.

Coverage Estimation

- We estimate a lower and upper bound.
- Let Π be a box partition of $\mathcal{B}$. Given a box $\boldsymbol{b} = [\alpha_1, \beta_2] \times \ldots \times [\alpha_n, \beta_n] \in \Pi$, we define $\boldsymbol{b}^+ = [l_1, \beta_1] \times \ldots \times [l_n, \beta_n]$ and $\boldsymbol{b}^- = [l_1, \alpha_1] \times \ldots \times [l_n, \alpha_n]$.
- $C(P, \Pi) \leq D^*(P, \mathcal{B}) \leq B(P, \Pi)$ [THIEMARD01]

$$B(P, \Pi) = \max_{\boldsymbol{b} \in \Pi} \max\{\frac{A(P, \boldsymbol{b}^+)}{k} - \frac{\lambda(\boldsymbol{b}^-)}{\lambda(\mathcal{B})}, \frac{\lambda(\boldsymbol{b}^+)}{\lambda(\mathcal{B})} - \frac{A(P, \boldsymbol{b}^-)}{k}\}$$

$$C(P, \Pi) = \max_{\boldsymbol{b} \in \Pi} \max\{|\frac{A(P, \boldsymbol{b}^-)}{k} - \frac{\lambda(\boldsymbol{b}^-)}{\lambda(\mathcal{B})}|, |\frac{A(P, \boldsymbol{b}^+)}{k} - \frac{\lambda(\boldsymbol{b}^+)}{\lambda(\mathcal{B})}|\}$$



Coverage-Guided Sampling

- **Bias the goal state sampling** distribution according to the current coverage of the visited states.
- To sample a hybrid state, we first sample a discrete location and then a continuous state.
- Let $\mathcal{P} = \{(q, P_q) \mid q \in Q \wedge P_q \subset \mathcal{I}_q\}$ be the current set of visited states.
- The **discrete location sampling distribution** depends on the current continuous state coverage of each location:

$$Pr[q_{goal} = q] = \frac{D^*(P_q, \mathcal{I}_q)}{\sum_{q' \in Q} D^*(P_{q'}, \mathcal{I}_{q'})}.$$

Coverage-Guided Sampling (cont'd)

- Suppose that we have already sampled a discrete location $q_{goal} = q$.
- The **sampling of a continuous state** consists of two steps:
  1. Sample a box $\boldsymbol{b}_{goal}$ in the box partition Π
  2. Sample a point x_{goal} in $\boldsymbol{b}_{goal}$ **uniformly**.
- The **box sampling** distribution in the first step is biased in order to **optimize the coverage**.
- Strategy: **reduce** both the **lower bound** $C(P, \Pi)$ and the **upper bound** $B(P, \Pi)$, P be the current set of visited points at location q

Coverage-Guided Sampling (cont'd)

$$C(P,\Pi) = \max_{\boldsymbol{b}\in\Pi} \max\{|\frac{A(P,\boldsymbol{b}^-)}{k} - \frac{\lambda(\boldsymbol{b}^-)}{\lambda(\mathcal{B})}|, \ |\frac{A(P,\boldsymbol{b}^+)}{k} - \frac{\lambda(\boldsymbol{b}^+)}{\lambda(\mathcal{B})}|\}$$

Define a number $A^*(\boldsymbol{b})$ s.t. $\dfrac{\lambda(\boldsymbol{b})}{\lambda(\mathcal{B})} = \dfrac{A^*(\boldsymbol{b})}{k}$. Let $\Delta_A(\boldsymbol{b}) = A(P,\boldsymbol{b}) - A^*(\boldsymbol{b})$
$\Rightarrow C(P,\Pi) = \frac{1}{k}\max_{\boldsymbol{b}\in\Pi}\{\max\{|\Delta_A(\boldsymbol{b}^+)|, |\Delta_A(\boldsymbol{b}^-)|\}\}$.

Potential influence on the lower bound:

$$\xi(\boldsymbol{b}) = \frac{1 - \Delta_A(\boldsymbol{b}^+)/k}{1 - \Delta_A(\boldsymbol{b}^-)/k}$$

Intepretation: **(1)** If $\Delta_A(\boldsymbol{b}^+) < 0$ and $|\Delta_A(\boldsymbol{b}^+)|$ large, the 'lack' of points in $\boldsymbol{b}^+$ is significant $\Rightarrow \xi(\boldsymbol{b})$ large, meaning that the selection of $\boldsymbol{b}$ is favored. **(2)** If $\Delta_A(\boldsymbol{b}^-) < 0$ and $|\Delta_A(\boldsymbol{b}^-)|$ is large, it is preferable not to select $\boldsymbol{b}$ to increase the chance of adding new points in $\boldsymbol{b}^-$.

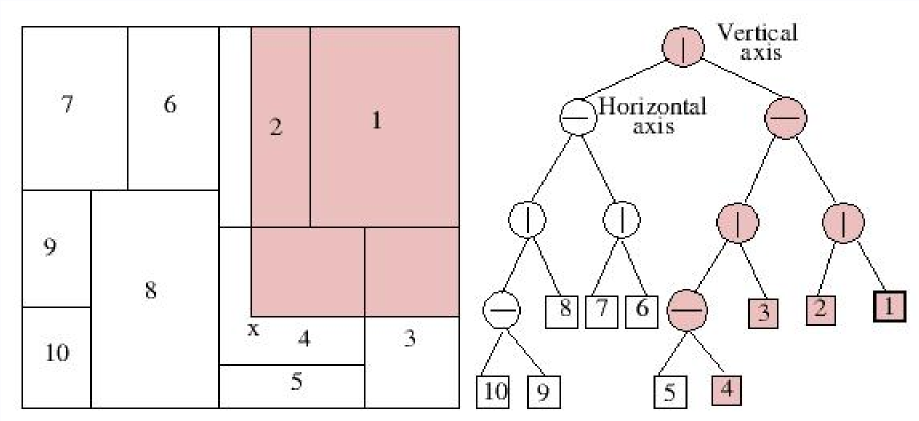
Implementation of gRRT

Using a hierachical box-partition of the state space, similar to a k-d tree.

- Approximate neighbors: to find a neighbor of x, we find the box $\boldsymbol{b}$ containing x and the neighbor is the point in $\boldsymbol{b}$ closest to x. Error control by fine tuning the partition granularity.
- Update the discrepancy estimation.
- Box splitting

Update the discrepancy estimation

- To update the star discrepancy estimation $\Rightarrow$ find all elementary boxes $\boldsymbol{b}$ s.t. the new point has increased the number of points in $\boldsymbol{b}^-$ and $\boldsymbol{b}^+$.
- These boxes are indeed those which intersect with the box $B_x = [x_1, L_1] \times \ldots \times [x_n, L_n]$.
  - If $\boldsymbol{b}$ is a subset of B_x, increment the numbers of points in both $\boldsymbol{b}^+$ and $\boldsymbol{b}^-$
  - If $\boldsymbol{b}$ intersects with B_x but is not entirely inside B_x, only increment the number of points in $\boldsymbol{b}^+$.



Reachability Completeness

In **motion planning**

- Given $\varepsilon > 0$, for any point x in the free state space, the probability that $\mathcal{T}^k$ at step k contains a vertex which is ε-close to x

$$lim_{k \to \infty} P[x \in N(\mathcal{T}^k, \varepsilon)] = 1$$

- The free state space is assumed to be controllable

In **reachability analysis**, not all points in the state space X is controllable. We derived **more general conditions for completeness**:

- Sampling: any subset of X with **positive volume** has a non-null probability of being sampled
- Control: Non-null probability that **each reachable direction is selected**. If the control input set is finite, this means $\forall u \in U : P[u^k = u] > 0$.

Plan

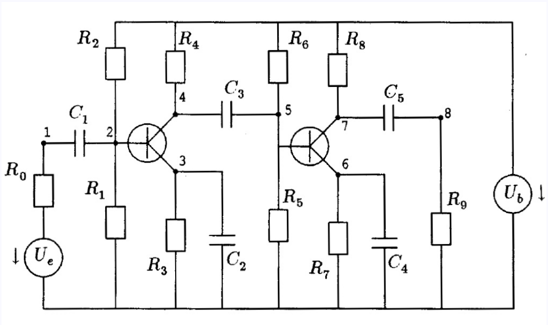
1. Introduction: Hybrid systems testing problem
2. Test coverage
3. Coverage-guided test generation
4. **Experimental results**

Transistor Amplifier

The circuit equations are a system of DAEs of index 1 with 8 continuous variables: $M\dot{y} = f(y,u)$ where M and f are:

$$\begin{pmatrix} -C_1 & C_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ C_1 & -C_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -C_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -C_3 & C_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_3 & -C_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -C_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -C_5 & C_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & C_5 & -C_5 \end{pmatrix}, \begin{pmatrix} -U_e/R_0 + y_1/R_0 \\ -U_b/R_2 + y_2(1/R_1 + 1/R_2) - (\alpha - 1)g(y_2 - y_3) \\ -g(y_2 - y_3) + y_3/R_3 \\ -U_b/R_4 + y_4/R_4 + \alpha g(y_2 - y_3) \\ -U_b/R_6 + y_5(1/R_5 + 1/R - 6) - (\alpha - 1)g(y_5 - y_6) \\ -g(y_5 - y_6) + y_6/R_7 \\ -U_b/R_8 + y_7/R_8 + \alpha g(y_5 - y_6) \\ y_8/R_9 \end{pmatrix}$$

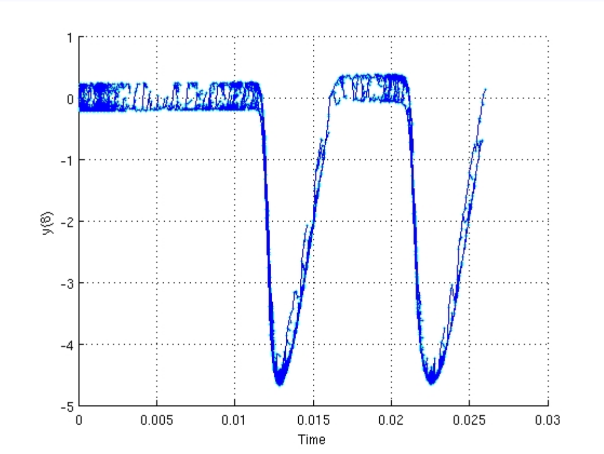
The circuit parameters are: $U_b = 6$; $U_F = 0.026$; $R_0 = 1000$; $R_k = 9000$, $k = 1, \ldots, 9$; $C_k = k10^{-6}$; $\alpha = 0.99$; $\beta = 10^{-6}$.
The initial state $y_{init} = (0, U_b/(R_2/R_1 + 1), U_b/(R_2/R_1 + 1), U_b, U_b/(R_6/R_5 + 1), U_b/(R_6/R_5 + 1), U_b, 0)$. The input signal $U_e(t) = 0.1 sin(200\pi t)$.

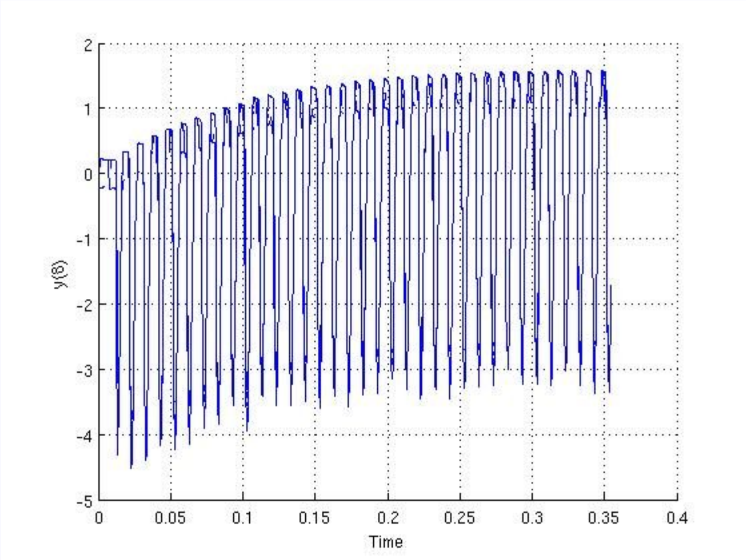


Transistor Amplifier - Results

Circuit parameter uncertainty: perturbation in the relation between the current through the source of the two transistors and the voltages at the gate and source $I_S = g(U_G - U_S) = \beta(e^{\frac{U_G - U_S}{U_F}} - 1) + \epsilon$, with $\epsilon \in [-5e-5, 5e-5]$.

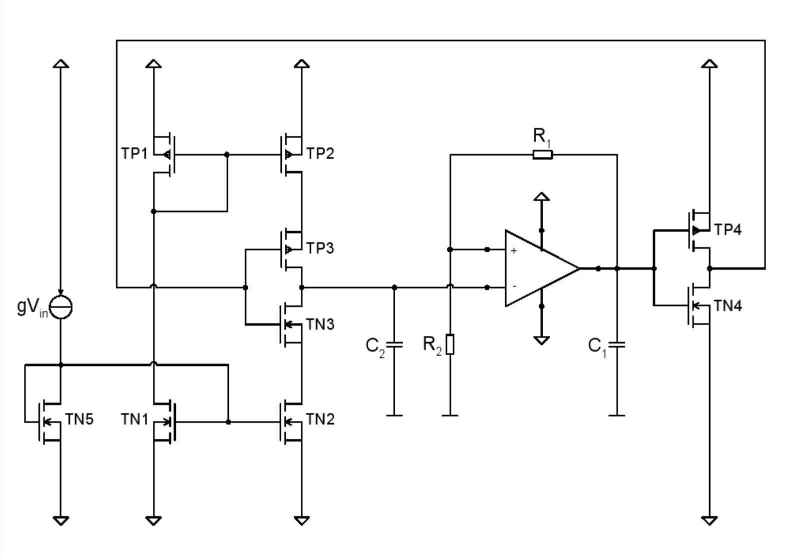
We used the gRRT algorithm to generate a test case $\Rightarrow$ presence of **overshoots** (the acceptable interval of U_8 in the non-perturbed circuit is $[-3.01, 1.42]$).

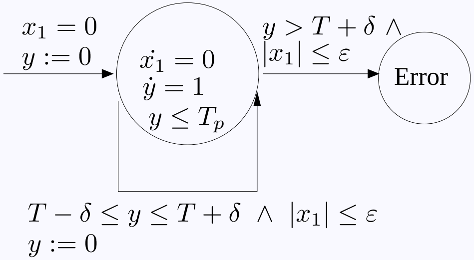




Voltage Controlled Oscillator

Circuit equations are DAEs with 55 continuous variables.

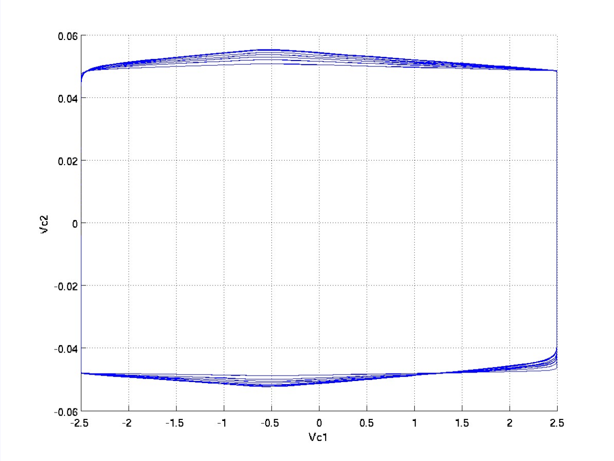




Voltage Controlled Oscillator - Results

We consider a constant input voltage $u_{in} = 1.7$ and a **time-variant deviation** of C_2 which ranges within $\pm 10\%$ of the value of $C_2 = 0.1e-4$

The generated test case shows that after the transient time, the variables v_{C_1} and v_{C_2} oscillate with the period $T \in [1.25, 1.258]s$ (with $\varepsilon = 2.8e{-}4$).

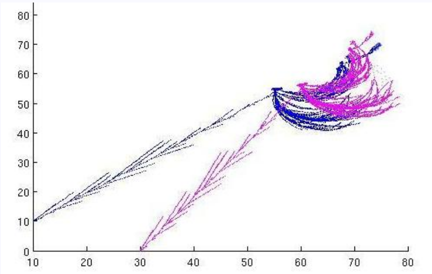


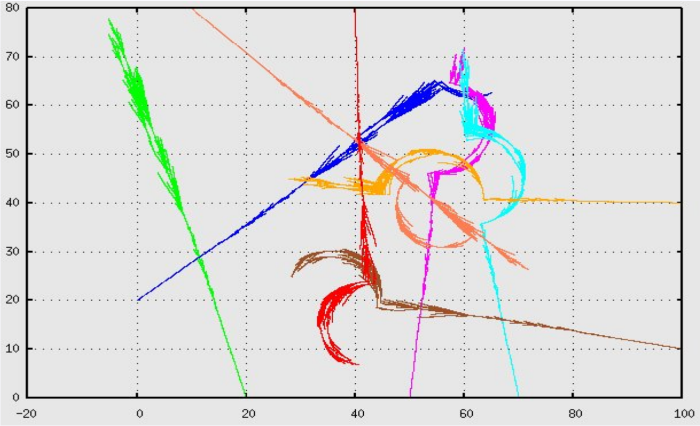
As a mixed-signal circuit example, we also tested on the Delta-Sigma modulator circuit.

Aircraft collision avoidance [MitchellTomlin00]

- **Continuous dynamics** of each aircraft: $\dot{x}_i = vcos(\theta_i) + d_1 sin(\theta_i) + d_2 cos(\theta_2)$, $\dot{y}_i = vsin(\theta_i) - d_1 cos(\theta_i) + d_2 sin(\theta_2)$, $\dot{\theta}_i = \omega$
where x_i, y_i: position, θ_i: relative heading. The continuous inputs are d_1 and d_2 are external disturbances.
- **Three discrete modes**: Mode 1, each aircraft begins in straight flight with a fixed heading. Mode 2: each makes an instantaneous heading change of 90 degrees, and begins a circular flight for π time units. Mode 3: each makes another instantaneous heading change of 90 degrees and resumes its original headings. For N aircrafts $\Rightarrow 3N + 1$ continuous variables (one for modeling a clock).
- $N = 2$ aircrafts, collision distance is 5. No colission was detected after visiting 10000 states. The computation time was 0.9 min.
- $N = 10$ aircrafts, the computation time was 10 min and a collision was detected after visiting 50000 states.

Aircraft collision avoidance





Higher dimensional systems

Tested systems $\dot{x}(t) = Ax(t) + u(t)$ were randomly generated. Matrix A in Jordan canonical form

dim n	Lower bound		Upper bound	
	gRRT	RRT	gRRT	RRT
3	0.451	0.546	0.457	0.555
5	0.462	0.650	0.531	0.742
10	0.540	0.780	0.696	0.904

dim n	Time (min)
5	1
10	3.5
20	7.3
50	24
100	71

Conclusions

Results

- **gRRT**: preserves the completeness property of RRT and is more time- and coverage-efficient
- Encouraging experimental results

Ongoing and Future work

- Partial observability

End

Thank You For Your Attention