# Probabilistic Analysis of Time Sensitive Systems

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 OCT 2014</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

4. TITLE AND SUBTITLE

**Probabilistic Analysis of Time Sensitive Systems**

5. AUTHOR(S)

6. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

**Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213**

7. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. SUPPLEMENTARY NOTES

The original document contains color images.

13. ABSTRACT

14. SUBJECT TERMS

15. SECURITY CLASSIFICATION OF:

<table>
<thead>
<tr>
<th>a. REPORT</th>
<th>b. ABSTRACT</th>
<th>c. THIS PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>unclassified</td>
<td>unclassified</td>
<td>unclassified</td>
</tr>
</tbody>
</table>

16. LIMITATION OF ABSTRACT

SAR

17. NUMBER OF PAGES

1

19a. NAME OF RESPONSIBLE PERSON

The original document contains color images.
**Probabilistic Analysis of Time Sensitive Systems**

**Problem Statement**

Time-sensitive systems in uncertain environments have complex behaviors. How do we assure correctness of such systems?

- Exact probabilistic verification is infeasible due to model size.
- Black box testing does not yield bounded predictions.
- Need formal approach for dealing with uncertainty.
- Accurate, bounded, probabilistic results.
- In reasonable time even for rarely occurring errors.

**SMC is a rigorous simulation-based approach for estimating that a property holds in a system.**

- System properties described in formal language (BLTL, etc.)
- Property is tested on “sample trajectories” (sequence of states).
- Each outcome treated as a Bernoulli trial (i.e., coin flip).

**SMC Basics**

- Indicator function \( l(\vec{x}) = 1 \) iff property holds for input \( \vec{x} \).
- Relative Error \( RE(\hat{p}) = \frac{|\hat{p} - p^*|}{p^*} \) is a measure of accuracy.
- Draw random samples from input distribution \( f(\vec{x}) \) until target Relative Error is met.
- Estimated probability that property holds is:
  \[
  \hat{p} = \frac{1}{N} \sum_{i=1}^{N} l(\vec{x}_i) = \frac{1}{10} = 0.1 \quad RE(\hat{p}) = \frac{0.32}{0.1} = 3.2
  \]

**Importance Sampling**

- Modify input distribution to make rare properties more visible.
- Weighting function \( W(\vec{x}) \) maps solution back to original problem.
- Reduced relative error with same number of samples:
  \[
  \hat{p} = \frac{1}{N} \sum_{i=1}^{N} l(\vec{x}_i)W(\vec{x}_i) = \frac{0.2 + 0.5 + 0.3}{10} = 0.1
  \]
  \[
  RE = \frac{0.18}{0.1} = 1.8
  \]

**Final Probability Estimate**

- Raw Probability Estimate \( \hat{p} = p^* \hat{p}_{raw} = 0.00047 \)
- \( RE(\hat{p}) = 0.01 \)

---

**Semantic Importance Sampling**

**A New Approach to Importance Sampling**

**Input Specification in C**

```c
#include "numaxis_client.h"
//@dist a=uniform(min=0,max=5)
//@dist b=normal(mean=1,std=1,min=0,max=5)
void simple()
{
    double a = INPUT_D("a");
    double b = INPUT_D("b");
    double c = a + b;
    double d = (a - b)/2.0;
    ASSERT(sin(c)*cos(d/2) < 0.995);
}
```

**SMT2 Model**

```plaintext
(set-logic QF_UFO)
(declare-fun a () Real)
(declare-fun b () Real)
(declare-fun c () Real)
(declare-fun d () Real)
(assert (= a_1 (+ a_1 b_1)))
(assert (= b_1 b))
(assert (= a_1 a))
(assert (= c_1 (+ a_1 b_1)))
(assert (> d_2 (- -2.1 b_1 2.1)))
(assert (< (* (min d_1) (cos d_1)) 0.9))
(smt-sat)
(smt-ok)
```

**Input Generation**

- Use \( I^*(\vec{x}) \) to generate random input vectors:
  - Randomly pick SAT cube
  - Randomly pick point in cube

- Raw Probability Estimate \( \hat{p}_{raw} = 0.024 \)
- \( RE(\hat{p}_{raw}) = 0.01 \)

**Simulation effort decreases exponentially with recursion depth.**

---

Jeffrey Hansen – jhansen@sei.cmu.edu