A Coverage-Guided Test Generation Tool for Hybrid Systems

Tarik Nahhal and Thao Dang

VERIMAG, 2 avenue de Vignate
38610 Gières, France
{Tarik.Nahhal,Thao.Dang}@imag.fr

Abstract. In this paper we describe a coverage-guided test generation tool for hybrid systems. The tool has been successfully applied to a number of case studies in control systems and in analog and mixed-signal circuits.

1 Testing Problem

Hybrid systems have been recognized as a high-level model appropriate for embedded systems, since this model can describe, within a unified framework, the logical part and the continuous part of an embedded system. The design of the tool follows the model-based design approach. Our test generation method is based on a randomized search and guided by a coverage measure. In particular, we use hybrid automata as a model for embedded systems. The hybrid automata we consider can admit continuous and discrete inputs. We assume that the tester can observe the discrete state and a subset of continuous variables. The system under test $\mathcal{A}_s$ often operates within some environment. In our testing problem, the tester plays the role of the environment and it performs experiments on $\mathcal{A}_s$ in order to study the relation between the specification $\mathcal{A}$ and the system under test $\mathcal{A}_s$. The tester emits an admissible input sequence to $\mathcal{A}_s$ and measures the resulting observation sequence in order to produce a verdict. Conformance is an important property of the relation between the system under test and the specification. In our framework, we assume that an input sequence which is admissible for $\mathcal{A}$ is also admissible for $\mathcal{A}_s$. Thus, *the system under test* $\mathcal{A}_s$ *is conform to the specification* $\mathcal{A}$, iff for every admissible input sequence, the set of observation sequences of $\mathcal{A}_s$ is included in that of $\mathcal{A}$.

A test case is represented by a tree where each node is associated with an observation and each edge of the tree is associated with an input action. Since a hybrid automaton might have an infinite number of infinite traces; therefore, we need to select a finite portion of the input space of $\mathcal{A}$ and test the conformance of $\mathcal{A}_s$ with respect to this portion. The selection is done using the following coverage criterion.

Test Coverage. The coverage measure is defined using the *star discrepancy* notion [2] in statistics. The star discrepancy of a set of points is a measure for the irregularity of the points over a region. When the region is a hyper-cube, the star discrepancy measures how badly the point set estimates the volume of the cube. Since a hybrid automaton can only evolve within the staying sets of its locations, we define the test coverage using the star discrepancy of the visited points with respect to these sets.

2 Test Generation

Our test generation is a combination of the Rapidly-exploring Random Tree RRT algorithm (a successful robot motion planning technique [4]) and a guiding tool used to achieve a good coverage of the system's behaviors we want to test. We call the resulting algorithm gRRT. In this section, we present only the main ideas of gRRT, and a detailed description of the algorithm can be found in [5].

The algorithm constructs a tree, the root of which corresponds to the initial state. In each iteration, a hybrid state s_{goal} is sampled to indicate the direction towards which the tree is expected to evolve. Expanding the tree towards s_{goal} is done by first finding a starting state s_{near} for the current iteration. It is natural to choose s_{near} to be a state near s_{goal}. The distance between two hybrid states is defined as an average length of the traces from s_{near} to s_{goal}. Next, we try to find an input to take the system from s_{near} towards s_{goal} as closely as possible. A new edge from s_{near} to the new state s_{new}, labeled with the associated input action is then added to the tree. To find s_{new}, when the input set is not finite it can be sampled, or one can solve a local optimal control problem. The algorithm terminates after reaching a satisfactory coverage value or after some maximal number of iterations. To extract a test case from the tree, we project the states at the nodes on the observable variables of $\mathcal{A}$. In this algorithm, the goal state sampling plays the role of guiding the exploration, with the goal to achieve a good coverage of the visited states.

Coverage-Guided Sampling. To evaluate the coverage, we estimate a lower and upper bound of the star discrepancy of a point set (exact computation is not an easy problem). These bounds as well as the information obtained from their estimation are used to decide which parts of the state space have been 'well explored' and which parts need to be explored more. We then bias the goal state sampling distribution according to the current coverage of the visited states. A detailed description of the guiding method can be found in [5].

Probabilistic completeness. Our test generation algorithm preserves the *probabilistic completeness* of the classic RRT algorithm. Roughly speaking, the probabilistic completeness property states that if the trace we seach for is feasible, then the probability that the algorithm finds it approaches 1 as the number k of iterations approaches infinity. This property is a way to explain a good space-covering property of the algorithm.

3 Implementation and Applications

We have implemented the above described algorithm in a test generation tool. To store the test cases, we use a data structure, similar to a k-d tree, which facilitates the required operations (such as, finding neighbor states, update the star discrepancy information). In most classic versions of hybrid automata, continuous dynamics are defined using ordinary differential equations (ODEs). To analyze analog and mixed-signal circuits, the behavior of which are described using differential algebraic equations (DAEs), we adapt the model to capture this particularity and use the well-known RADAU algorithm for solving DAEs [3]. Our test generation tool has been tested on various examples, which proved its scalability to high dimensional systems [5]. In addition, we have also successfully applied the tool to a number of case studies in control systems (in particular, an aircraft coordination problem with 31 continuous variables) and of in analog and mixed-signal circuits (in particular, a Voltage Controlled Oscillator VCO circuit with 55 continuous variables [5, 6]).

As future work, in order to facilitate the application to practical circuits, we need a tool for automatic generation of hybrid automata from commonly-used circuit descriptions, such as Spice netlists. On the other hand, as mentioned earlier, an efficient and reliable simulation method is a key ingredient in our approach. The state-of-the-art SPICE simulator is prone to convergence problems when dealing with circuit components with stiff characteristics. In collaboration with researchers at INRIA Rhône-Alpes, a topic of our undergoing research is to integrate in our test generation tool a simulation algorithm based on *the non-smooth approach* [1].

References

1. V. Acary and F. Pérignon. SICONOS: A software platform for modeling, simulation, analysis and control of Non Smooth Dynamical system. *Proc. of MATHMOD*, ARGSIM Verlag, 2006.
2. J. Beck and WWL. Chen. Irregularities of Distribution. Cambridge Univ. Press, 1987.
3. E. Hairer, C. Lubich and M. Roche. The Numerical Solution of Differential-Algebraic Systems by Runge Kutta Methods. Lecture Notes in Mathematics 1409, Springer-Verlag, 1989.
4. S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Directions*, pages 293–308. A K Peters, Wellesley, MA, 2001.
5. T. Nahhal and T. Dang. Test Coverage for Continuous and Hybrid Systems In *Computer Aided Verification* CAV'07, LNCS, Berlin, 2007 (to appear).
6. T. Nahhal and T. Dang. Test generation for analog and mixed-signal circuits using hybrid system models Technical report, VERIMAG, April 2007.