Demo: S-TaLiRo: A tool for Testing and Verification for Hybrid Systems: Recent Functionality and Additions

Bardh Hoxha Houssam Abbas Adel Dokhanchi Georgios Fainekos
Arizona State University, Tempe, AZ, USA
{bhoxha, hyabbas, adokhanc, fainekos}@asu.edu

ABSTRACT

In this demo, we will demonstrate the latest features of S-TaLiRo, a modular software tool that provides various methods of verification and testing of hybrid systems, using a combination of stochastic optimization algorithms, and local descent methods.

1. INTRODUCTION

S-TaLiRo is a modular software tool for the verification and testing of hybrid systems. It can analyze arbitrary Simulink models, user-defined functions and blackbox models. At the heart of the tool, we use randomized testing based on stochastic optimization techniques. The tool can be seamlessly run inside the Matlab environment. S-TaLiRo enables finding trajectories of systems that falsify Metric Temporal Togic Specifications [1]. Notable improvements include the following:

1. Finding control inputs to a *stochastic* system that make it minimally robust with respect to a given specification.
2. Parameter estimation of temporal logic specifications for state and timing parameters.

We will demonstrate the features presented above to the attendees on a modified version of the Automatic Transmission model provided by Mathworks as a Simulink demo[1]. We will demonstrate how the tool is setup, the creation of a system model, the creation of a specification, and how to falsify that specification. This includes stochastic systems.

We also demonstrate the use of S-TaLiRo in a model-based design. Other improvements to S-TaLiRo include:

- A graphical user interface.
- Search over input space that allows for variable distribution of control points.
- Added support for the use of the parallel toolbox in stochastic optimization.

In the following sections we will provide an overview of each of the main functionalities.

2. DEMO SETUP

We will bring a laptop and show how to use S-TaLiRo with different benchmark examples. We will also demonstrate the new graphical user interface.

[1]Available at: `http://www.mathworks.com/products/simulink/demos.html`

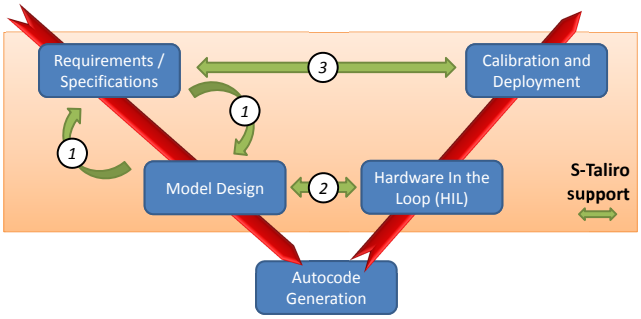

Figure 1: S-TaLiRo support for Model-Based Design.

3. FALSIFICATION

Temporal verification involves the ability to prove as well as to falsify temporal logic properties of systems. S-TaLiRo searches for counterexamples to Metric Temporal Logic (MTL) properties for non-linear hybrid systems through global minimization of a robustness metric [2].

At its core, it integrates robustness computation for traces of hybrid systems (TaLiRo) [2] with stochastic simulation. The search returns the simulation trace with the smallest robustness value that was found. Traces with positive - but low - robustness values are closer in distance to falsifying traces, using a mathematically well-defined notion of distance between trajectories and temporal logic properties. Such traces provide valuable insight to the developer on why a given property holds, or how to refocus a search for a counter-example.

4. ROBUSTNES OF STOCHASTIC HYBRID SYSTEMS

We have extended this work to *stochastic* hybrid systems. Stochasticity is inherent in many hybrid systems, and might arise as the result of actuator inaccuracies, sensor readings, rates of arrivals, component failures, or even by design to mitigate attacks on the system, etc. One important question is how such random phenomena can affect the functional correctness properties of a hybrid system.

In the past, Statistical Model Checking (SMC) for stochastic hybrid systems was proposed [6]: given a probability distribution on the parameters of the stochastic hybrid system and a specification φ in a temporal logic, SMC computes a probability that φ holds on the system along with confidence intervals. The probability of success of a formal specification

is not sufficient in all applications. For example, consider the following important requirement for car manufacturers: the normalized air-to-fuel (A/F) ratio should always be within 1 ± 0.1. We need to distinguish between designs that satisfy this requirement to varying degrees: all else being equal, a system design for which the worst expected ratio stays the closest to 1, and the probability of failure is low, should be preferred over all other correct designs. Moreover, in many cases, we do not just need to analyze the system behavior under typical input scenarios, but also to discover the inputs that induce the worst (expected) system behavior.

We will demonstrate how S-TaLiRo finds solutions to the two aforementioned challenges, and what guarantees we have on the quality of these solutions. Also, we demonstrate the use of S-TaLiRo in an MBD process, where the system design is improved based on the least robust traces.

Formally, S-TaLiRo has been enhanced to address the following problem: a *system* Σ as a mapping from a set of *initial operating conditions* X_0 and discrete-time *input signals* $\mathfrak{u} \in U^{\mathbb{T}}$ to discrete-time *output signals* in $Y^{\mathbb{T}}$, where X_0, U, Y are subsets of $\mathbb{R}^n$, and $\mathbb{T} \subset \mathbb{N}$ is a time-set. Discrete system variables like counters and flags are modeled as integers, so X_0 and U can be "hybrid" spaces. The input signals (if any) are parameterizable using a finite-dimensional parameter vector λ. Define the *decision space* $\Theta \triangleq X_0 \times \Lambda$.

Let $(\Omega, \mathcal{A}, P)$ be a probability space. The random events $\omega \in \Omega$ model the sources of randomness in Σ. Corresponding to every decision $\theta = (x_0, \lambda) \in \Theta$, the output of the system is modeled as a *discrete-time stochastic process* $\mathbf{Y}(t, \omega; \theta)$ parametrized by θ. Given a system property expressed in MTL and a sample path y, its robustness is modeled as a functional which takes in a sample path y of $\mathbf{Y}$, and produces a robustness value ρ_φ:

$$\rho_\varphi : y(\cdot, \omega; \theta) \mapsto \rho_\varphi(y(\cdot, \omega; \theta)) \equiv \rho_\varphi(\omega, \theta) \in [-\infty, \infty]$$

Robustness is positive if y satisfies φ, and negative otherwise. Minimizing robustness leads to trajectories that are closest to violating the specification. For stochastic systems, we use average robustness.

Problem 1. *Take a stochastic hybrid system Σ, an MTL specification φ, a test duration $T > 0$, and a decision $\theta \in \Theta$. Define the expected robustness U of the stochastic hybrid system w.r.t. φ as*

$$U(\theta) = \mathbb{E}_P[\rho_\varphi(\omega, \theta)] = \int_\Omega \rho_\varphi(\omega, \theta) dP(\omega) \qquad (1)$$

Compute the minimum expected robustness of the system with respect to the MTL specification: $U_* = \inf\{U(\theta) | \theta \in \Theta\}$

To solve Problem 1, we use a variant of Simulated Annealing (SA) adapted for objective functions that are expectations [4]. The rate of convergence of SA to the global minimum has known bounds [3]. In practice, this means that after any number of iterations, we can lower-bound the probability that the algorithm will choose a point $(x_0, \lambda) \in \Theta$ whose U-value is ϵ-close to the global minimum $\inf_{\theta \in \Theta} U(\theta)$. This can then be used as a stopping criterion. This SA variant can also provide bootstrapping estimates of the variance of $\rho_\varphi(\theta)$. This estimate is valuable feedback to the designer, since a positive but small average robustness with a large variance indicates a probability of failing the specification.

5. PARAMETER ESTIMATION

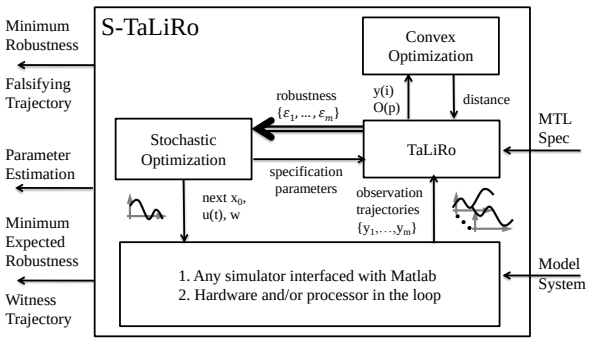

Figure 2: Architecture of S-TaLiRo.

In Model Based Development (MBD) of embedded systems, it is often desirable to not only verify/falsify certain formal system specifications, but also to automatically explore the properties that the system satisfies. Namely, given a parametric specification, we would like to automatically infer the ranges of parameters for which the property holds/does not hold on the system. We consider parametric specifications in Metric Temporal Logic (MTL). Using robust semantics for MTL, the parameter estimation problem can be converted into an optimization problem which can be solved by utilizing stochastic optimization methods.

Problem 2 (MTL Parameter Estimation Problem). *Given an MTL formula $\phi[\theta]$ with a vector of unknown parameters $\theta \in \Theta = [\theta_m, \theta_M]$, a hybrid system Σ, and a maximum testing time T, find an optimal range $\Theta^* = [\theta^*_m, \theta^*_M]$ such that for any $\zeta \in \Theta^*$, $\phi[\zeta]$ does not hold on Σ, i.e., $\Sigma \not\models \phi[\zeta]$.*

We demonstrate a method for solving this problem for specifications whose robustness function is monotonic with respect to the set of parameters λ. The method is explained in detail in [5].

Acknowledgments The work benefited from the input of Raymond Turin, James Kapinski and Jyotirmoy Deshmukh. This work was partially funded by NSF awards CNS 1116136, CNS 1319560, IIP-0856090 and the NSF I/UCRC Center for Embedded Systems.

6. REFERENCES

[1] Y. S. R. Annapureddy, C. Liu, G. E. Fainekos, and S. Sankaranarayanan. S-taliro: A tool for temporal logic falsification for hybrid systems. In *Tools and algorithms for the construction and analysis of systems*, volume 6605 of *LNCS*, pages 254–257. Springer, 2011.

[2] G. E. Fainekos and G. J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, 2009.

[3] A. Lecchini, J. Lygeros, and J. M. Maciejowski. Stochastic optimization on continuous domains with finite-time guarantees by markov chain monte carlo methods. *IEEE Transactions on Automatic Control*, 55:2858–2863, Dec. 2010.

[4] P. Muller. Simulation based optimal design. In B. J.O., J. M. Bernardo, A. Dawid, and S. A. F. M., editors, *Proc. 6th Valencia Int. Meeting on Bayesian Statistics*, pages 459–474. Oxford, 1999.

[5] H. Yang, B. Hoxha, and G. Fainekos. Querying parametric temporal logic properties on embedded systems. In *Testing Software and Systems*, pages 136–151. Springer, 2012.

[6] P. Zuliani, A. Platzer, and E. M. Clarke. Bayesian statistical model checking with application to simulink/stateflow verification. In *13th ACM International Conference on Hybrid Systems: Computation and Control*, pages 243–252, 2010.