EPiC Series in Computing
Volume 48, 2017, Pages 170–174
ARCH17. 4th International Workshop on Applied
Verification of Continuous and Hybrid Systems

ARCH-COMP17 Category Report: Preliminary Results on the Falsification Benchmarks

Adel Dokhanchi[1], Shakiba Yaghoubi[1], Bardh Hoxha[2], and Georgios Fainekos[1]

[1] School of Computing, Informatics and Decision Systems Engineering,
Arizona State University, Tempe, AZ, U.S.A.
{adokhanc, syaghoub, fainekos}@asu.edu
[2] Department of Computer Science,
Southern Illinois University, Carbondale, IL, U.S.A.
bhoxha@cs.siu.edu

1 Introduction

This report presents some preliminary base results from the 2017 friendly competition in the ARCH workshop [2] for the falsification of temporal logic specifications over Cyber-Physical Systems category. The benchmarks are available on the ARCH website (cps-vo.org/group/ARCH). In this report, we present results on a powertrain model developed by Toyota Technical Center which contains a complex automatic air-fuel control subsystem [7].

2 Falsification Tool: S-TaLiRo

S-TaLiRo [4] is a Matlab toolbox that searches for system behaviors that falsify (do not satisfy) specifications presented in Signal Temporal Logic (STL) [8]. It can analyze arbitrary Simulink models or user-defined black box systems, e.g., autonomous vehicles modeled in a robotics simulator. S-TaLiRo performs automated randomized test case generation based on stochastic optimization techniques guided by formal requirements in STL. Among the advantages of the toolbox is the seamless integration inside the Matlab environment, which is widely used in the industry for model-based development. For a recent overview of the S-TaLiRo functionality see [6]. The tool is publicly available on-line at [1] under General Public License (GPL).

3 Benchmark Results

Our experiments were conducted on a 64-bit Intel Xeon CPU (2.5GHz) with 64-GB RAM and Windows Server 2012. We used MATLAB 2015a to run the falsification toolbox S-TaLiRo [1]. For our experiments, we used the following stochastic optimization methods: Simulated Annealing (SA) [3], Cross-Entropy (CE) optimization [9], and Uniform Random (UR) sampling. We remark that all the experiments were performed with the default parameters for each optimization method. It would be expected that further improvements can be achieved by tuning the performance of the optimization algorithms for each benchmark problem. All the benchmark problems are available with the S-TaLiRo distribution [1] or from the ARCH workshop repository [2].

3.1 Powertrain Control

The Powertrain Control benchmark presented in this report was first introduced in [7]. The benchmark provides a high complexity model of an automatic air-fuel control system. It consists of an air-fuel controller and a mean-value engine model. The closed loop system takes two exogenous inputs: the throttle angle θ_{in} and, the engine speed ω. It has 3 continuous-valued states associated with the controller and 5 continuous-valued states associated with the plant. In addition, there are states which are introduced by the variable delay.

The controller has 4 modes of operation: "Startup", "Normal", "Power" and "Fault". Depending on the operation mode, the system should satisfy different requirements. We used a slightly modified version of the requirements presented in Eq. (27) of the paper by Jin et al. [7]. The following specification needs to be satisfied when the system is in the "Normal" mode:

$$\phi_{PB} = \Box_{(\tau_s,T)}((rise(a) \vee fall(a)) \rightarrow \Box_{(\eta,\zeta)}(|\mu| < \beta))$$

where $a = 40$, $rise(a) \equiv (\theta_{in} \leq 8.8°) \wedge \Diamond_{(0,\epsilon)}(\theta_{in} \geq a)$ for a small enough ϵ, $fall(a)$ is defined similarly, $\tau_s = 11$ is the necessary time for the system to enter the "Normal" mode from the "Startup" mode, $T = 50$ is the total simulation time, $\eta = 1$ is the settling time required after a $rise$ or $fall$ event happens, $\zeta = 5$ is the end of the current time interval in which the input is kept constant, and, finally, μ is the normalized error signal that indicates the error in the value of the state Air/Flow ratio from a reference value.

The formula states that whenever event $rise$ or $fall$ happens (the antecedent, which is over the input signal), μ should remain in the specified bound after the settling time η, and before other changes are made to the input (after time ζ). The antecedent of the formula is over the input signals of the system. In this report, the acceptable error bound β is reduced to 0.008 to make falsification feasible. Note that abrupt changes in the value of the input signal are acceptable and necessary here to satisfy the antecedent but, frequent changes in the input are not (less than time ζ). As a matter of fact, increasing the frequency of the changes renders the problem less interesting since falsification becomes easier.

We compare results for two falsification algorithms. One is a general falsification algorithm, where the optimizer minimizes the robustness value with respect to the given STL specification. This is the standard method used in S-TaLiRo. The second one is Vacuity Aware Falsification (VAF) [5]. In VAF, for reactive specifications, as a first step in falsification, we attempt to satisfy the antecedent and, then, falsify the specification. The S-TaLiRo VAF support has not been released yet to the public S-TaLiRo repository (SVN Revision 91 [1]), but it is provided with the Repeatability Evaluation Package submitted to the ARCH workshop and it is archived there. Since the antecedent can be satisfied at any time after τ_s, in our S-TaLiRo implementation, in general, we attempt to satisfy the antecedent in a fraction of T ($T/2$ here) so that there is enough time in the future to falsify the whole formula (even though in this particular benchmark this may not be of consequence).

We used 50 runs (experiments) for each algorithm with 100 tests for each run. The experimental results are presented in Tables 1 and 2, and a sample falsifying input and trajectory are shown in Fig. 1. In the tables, "Min Tests" indicates the minimum number of tests in the case of falsification, while "Min Rob." indicates the minimum best robustness values achieved for

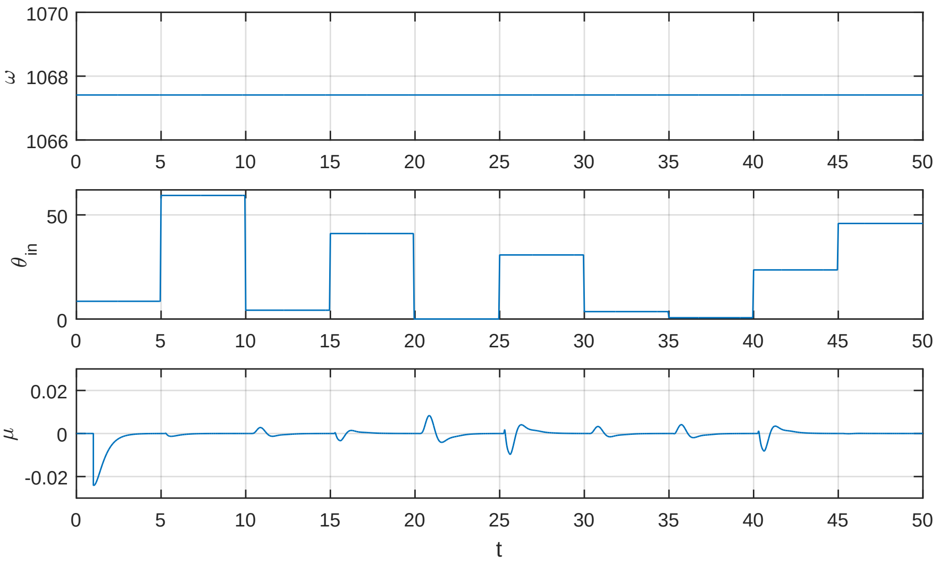

Figure 1: A falsifying piecewise constant input signal (θ_{in}) and the corresponding output trajectory (μ) of the powertrain system for specification ϕ_{PB}. The specification ϕ_{PB} is falsified at time t=20.785, and the robustness value is -6.12×10^{-5}. The input focuses on antecedent falsification up to time $t = 25$. The first 11 sec are ignored based on the requirements in ϕ_{PB}.

the cases without falsification. This gives an idea on how close these cases were to falsification. Using VAF, we achieve a slight improvement on the performance of the algorithm. This is due to the fact that the challenge in this falsification benchmark is mainly related to the consequent rather than the antecedent. Generally, if we heuristically force the antecedent to occur in the first half of the trace, then we observe a considerable increase in the number of falsifications. However, we cannot claim that because we enforce the antecedent to occur earlier, there is more time to search for the consequent. In this benchmark example, the consequent must occur within 5 time units of the antecedent being activated. Therefore, we believe that there is space for improvement in the falsification rate even for pure black-box methods and that this is a challenging benchmark which can drive forward the competition in the falsification category of the ARCH workshop.

Table 1: General Falsification

Optim.	Fals	Min Tests	Max Tests	Avg Tests	Min Rob.	Max Rob.	Avg Rob.
UR	7/50	18	93	52	1.7×10^{-5}	0.0035	8.81×10^{-4}
SA	9/50	13	83	50	3.54×10^{-5}	0.0042	0.0012
P-SA	4/50	34	80	55	7.41×10^{-6}	0.0051	0.0016

We also designed another experiment in which the antecedent is always satisfied when we are sampling for new input signals. Since the antecedent is satisfied whenever $rise$ or $fall$ happens, we used a single pulse as the input signal (shown in Fig. 2). The search space in S-TaLiRo is over the times t_1 and t_2 and the signal values x_1, x_2 and x_3 such that the antecedent is

Table 2: Vacuity Aware Falsification

Optim.	Fals.	Min Tests	Max Tests	Avg Tests	Min Rob.	Max Rob.	Avg Rob.
UR	9/50	12	96	63	3.4×10^{-6}	0.003	0.00086
SA	29/50	7	95	39	2.38×10^{-6}	0.0043	0.0013

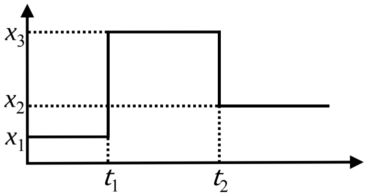

Figure 2: Pulse input to satisfy antecedent of ϕ_{PB}.

always satisfied ($t_2 < t_1 + 5$, $x_1, x_3 < 8.8$ and $x_2 > a$). Note that we can also try the case for the inverse pulse in which $x_1, x_3 > a$ and $x_2 < 8.8$. Allowing the test case generator in S-TaLiRo to choose either the first set of constraints or the second set of constraints during search would make the search space non-convex and, in that case, the search space sampling problem becomes more challenging. More generally, if desired, S-TaLiRo can search over the input signal space of a finite number of arbitrary magnitude and duration pulses which satisfy the *rise* and *fall* events sequentially.

Even though in this experiment any input signal sampled is constrained to satisfy the antecedent, S-TaLiRo has been able to falsify ϕ_{PB} in only 4 out of 50 runs. This result is shown in Table 1 under the name "P-SA". This indicates that the challenge in the problem is really in the consequent as opposed to activating the antecedent. A falsifying input is shown in Fig 3.

4 Conclusions

We have presented some preliminary base results for the falsification competition of the ARCH workshop. The results indicate that black box search based test generation methods do not perform much better than random sampling on this challenging benchmark. On the other hand, utilizing some information on the structure of the specification can help in at least doubling the rate of falsifications. We hope that this is viewed as a motivation that there is space for improvement in black box or gray box falsification methods and a competition on falsification.

References

[1] S-TaLiRo : https://sites.google.com/a/asu.edu/s-taliro/.

[2] Workshop on Applied Verification for Continuous and Hybrid Systems (ARCH) http://cps-vo.org/group/ARCH.

[3] H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivančić, and A. Gupta. Probabilistic temporal logic falsification of cyber-physical systems. *ACM Trans. Embed. Comput. Syst.*, 12(2s):95:1–95:30, May 2013.

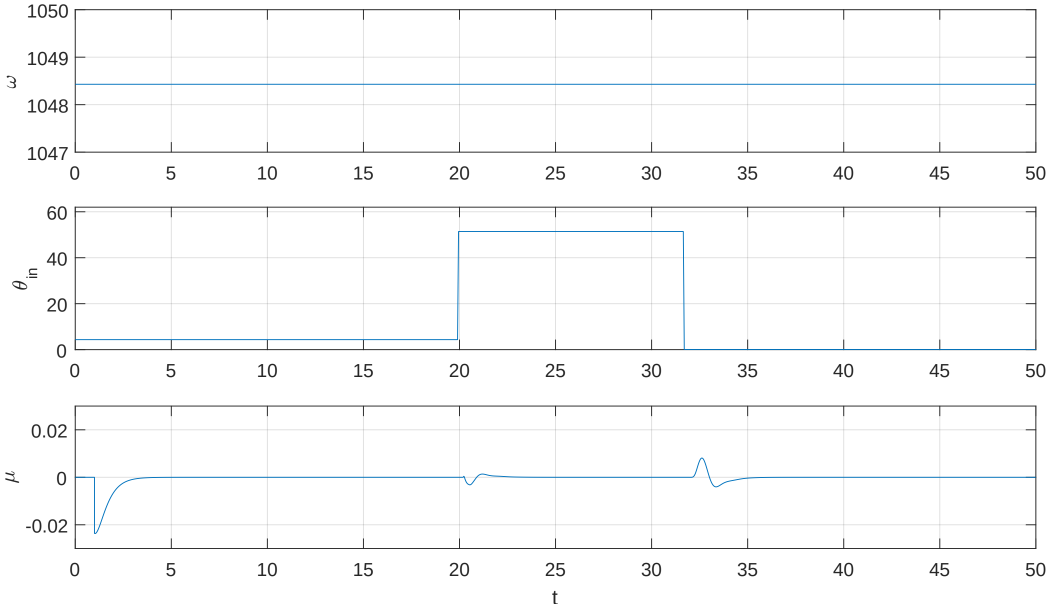

Figure 3: A sample falsifying pulse input signal and the corresponding trajectory of the powertrain system for ϕ_{PB}. The formula is falsified at time t=32.6425, and the robustness value is -4.083×10^{-5}.

[4] 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.

[5] A. Dokhanchi, S. Yaghoubi, B. Hoxha, and G. Fainekos. Vacuity aware falsification for MTL request-response specifications. In *IEEE International Conference on Automation Science and Engineering*, 2017.

[6] B. Hoxha, H. Bach, H. Abbas, A. Dokhanchi, Y. Kobayashi, and G. Fainekos. Towards formal specification visualization for testing and monitoring of cyber-physical systems. In *Int. Workshop on Design and Implementation of Formal Tools and Systems*. October 2014.

[7] X. Jin, J. V. Deshmukh, J. Kapinski, K. Ueda, and K. Butts. Powertrain control verification benchmark. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 253–262. ACM, 2014.

[8] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Proceedings of FORMATS-FTRTFT*, volume 3253 of *LNCS*, pages 152–166, 2004.

[9] S. Sankaranarayanan and G. Fainekos. Falsification of temporal properties of hybrid systems using the cross-entropy method. In *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '12, pages 125–134, New York, NY, USA, 2012. ACM.