ARCH-COMP18 Category Report: Results on the Falsification Benchmarks

Adel Dokhanchi[1], Shakiba Yaghoubi[1], Bardh Hoxha[2], Georgios Fainekos[1], Gidon Ernst[3], Zhenya Zhang[4], Paolo Arcaini[4], Ichiro Hasuo[4], and Sean Sedwards[5]

[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
[3] Computing and Information Systems,
University of Melbourne, VIC, Australia
gidon.ernst@unimelb.edu.au
[4] National Institute of Informatics, Tokyo, Japan
{zhangzy,arcaini,hasuo}@nii.ac.jp
[5] Intelligent Systems Engineering Lab,
University of Waterloo, ON, Canada
sean.sedwards@uwaterloo.ca

1 Introduction

This report presents results from the 2018 friendly competition in the ARCH workshop [3] for the falsification of temporal logic specifications over Cyber-Physical Systems category. The results extend those of the competition of the previous year 2017 by including an additional approach to falsification. 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 [10].

2 Falsification Tools

S-TaLiRo [5] is a Matlab toolbox that searches for system behaviors that falsify (do not satisfy) specifications presented in Signal Temporal Logic (STL) [11]. 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 [9]. The tool is publicly available on-line at [2] under General Public License (GPL).

FalStar is an experimental prototype of a falsification tool that explores the idea to construct falsifying inputs incrementally in time, thereby exploiting potential time-causal dependencies in the problem. It implements several algorithms: time-staging [8], a two layered framework combining Monte-Carlo tree search with stochastic optimization [13], and a probabilistic algorithm that adapts to the difficulty of the problem. The latter algorithm was used for this competition. The code is publicly available under the BSD license [1].

3 Benchmark: Powertrain Control

The Powertrain Control benchmark presented in this report was first introduced in [10]. 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. [10]. 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.

4 Experimental Results

The experiments with S-TaLiRo [2] were conducted on a 64-bit Intel Xeon CPU (2.5GHz) with 64-GB RAM and Windows Server 2012 running MATLAB 2015a. For these experiments, we used the following stochastic optimization methods: Simulated Annealing (SA) [4], Cross-Entropy (CE) optimization [12], 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 [2] or from the ARCH workshop repository [3].

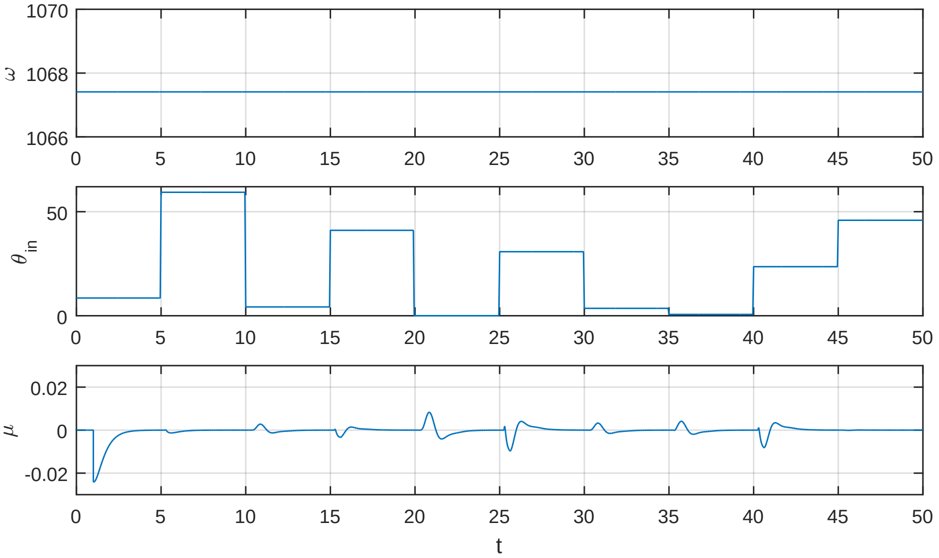

Figure 1: A falsifying piecewise constant input signal (θ_{in}) found by S-TaLiRo and the corresponding output trajectory (μ) of the powertrain system for specification ϕ_{PB}. The specification ϕ_{PB} is falsified at time $t = 20.785$, and the robustness value is -6.12×10^{-5}. The input focuses on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

The experiments added to the report in 2018 from FalStar were conducted on a 64-bit Intel i7-7600U CPU (2.8GHz with 4 cores and 16-GB RAM and Ubuntu 16.04, running MATLAB 2018a. The same Simulink model was used so that the results are comparable. We measure the success rate of finding a falsifying input as a primary indicator of the performance of the tools. Moreover, we state the number of simulations required for success. Note that the computational overhead of the falsification tools is negligible in comparison to running the simulations, so that the number of simulations is indicative of the performance of the tools.

We compare results for the following 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) [6]. In VAF, for reactive specifications, as a first step in falsification, we attempt to satisfy the antecedent and, then, falsify the specification. In this report we review last year's results [7], which demonstrate that S-TaLiRo can search over complex constraint input spaces. The S-TaLiRo VAF is publicly available [2]. 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).

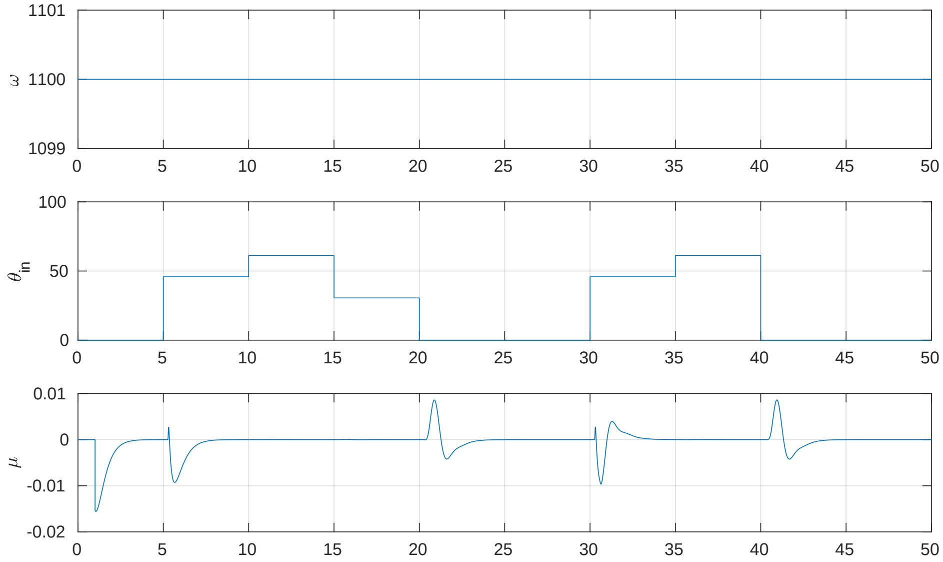

Figure 2: A falsifying trace found by FalStar corresponding to the trial with Min Tests = 1 in Table 2. The specification ϕ_{PB} is falsified the by three spikes after $t = 11$.

The third is a novel "Adaptive" algorithm implemented in FalStar that constructs input signals incrementally by constructing a tree on the search space. The key feature of the algorithm is that it tries "obvious" input signals first, i.e., such that take extreme input values $\theta = 0.0$ or $\theta = 61.1$. Only if such values turn out not to be useful, the search gradually switches to more fine-grained choices. Moreover, the algorithm is biased towards extending prefixes that have lead to less robust traces previously.

We used 50 runs (experiments) for each algorithm with 100 tests for each run. The experimental results are presented in Tables 1 and 2. A sample falsifying input and trajectory for S-TaLiRo is shown in Fig. 1, and one for FalStar is shown in Fig. 2. 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 the cases without falsification (not applicable for the adaptive algorithm as it always finds a trace with negative robustness). This gives an idea on how close these cases were to falsification.

Using VAF, we achieve a slight improvement on the performance of S-TaLiRo. 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
Adaptive	50/50	1	23	6	n/a	n/a	n/a

Table 2: Vacuity Aware Falsification from ARCH-COMP17 [7]

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

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. 3). 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 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 4.

5 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 in VAF can help in at least doubling the rate of falsifications.

Furthermore, the adaptive algorithm shows that random sampling under a strong bias towards extreme solutions can in fact lead to high falsification rates. For the powertrain benchmark, this is not too surprising: A large deviation of the air to fuel rate from the reference value is likely linked to extreme changes in the throttle.

Acknowledgement FalStar is supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST; and Grants-in-Aid No. 15KT0012, JSPS.

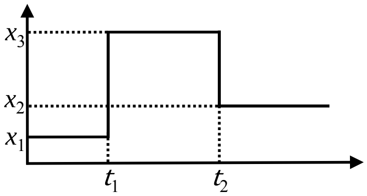

Figure 3: Pulse input to satisfy antecedent of ϕ_{PB}.

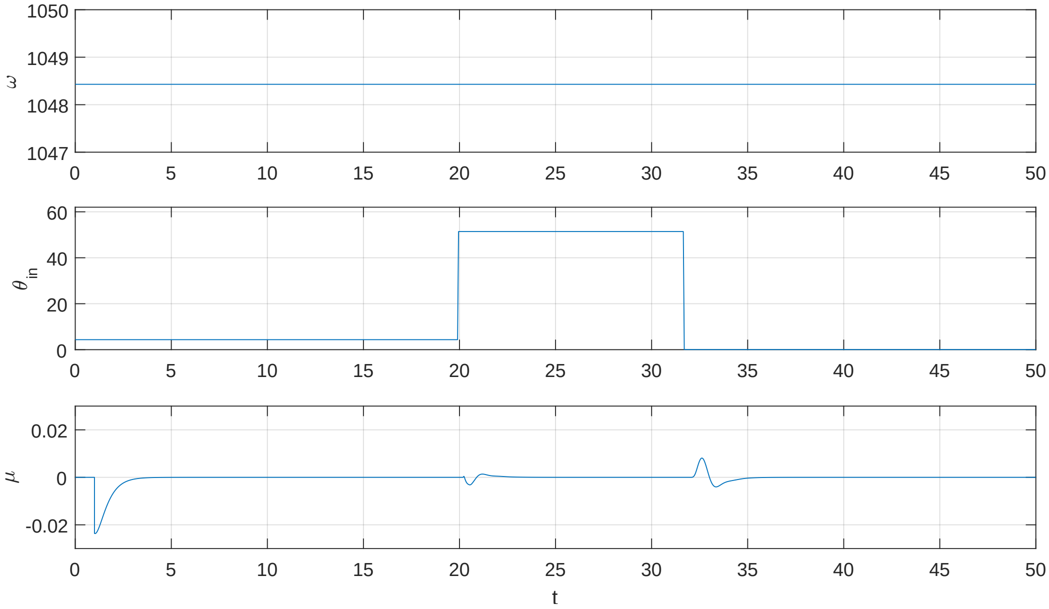

Figure 4: A sample falsifying pulse input signal and the corresponding trajectory of the powertrain system for ϕ_{PB}. The formula is falsified at time t=32.6425, and the robustness value is -4.083×10^{-5}.

References

[1] FalStar : https://github.com/ERATOMMSD/falstar.

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

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

[4] 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.

[5] 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.

[6] 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.

[7] A. Dokhanchi, S. Yaghoubi, B. Hoxha, and G. E. Fainekos. ARCH-COMP17 category report: Preliminary results on the falsification benchmarks. In *ARCH17. 4th International Workshop on Applied Verification of Continuous and Hybrid Systems, collocated with Cyber-Physical Systems Week (CPSWeek) on April 17, 2017 in Pittsburgh, PA, USA*, pages 170–174, 2017.

[8] G. Ernst, I. Hasuo, Z. Zhang, and S. Sedwards. Time-staging enhancement of hybrid system falsification. In *Symbolic and Numerical Methods for Reachability Analysis (SNR)*, EPTCS, 2018.

[9] 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.

[10] 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.

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

[12] 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.

[13] Z. Zhang, I. Hasuo, G. Ernst, and S. Sedwards. Two-layered falsification of hybrid systems guided by monte carlo tree search. Preprint, http://arxiv.org/abs/1803.06276.06276, 2018.