Poster Abstract: STRONG: A Trajectory-based Verification Toolbox for Hybrid Systems

Yi Deng
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute

Akshay Rajhans
Department of Electrical and Computer Engineering
Carnegie Mellon University

A. Agung Julius
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute

1. INTRODUCTION

STRONG (System Testing using RObust Neighborhood Generation) is a MATLAB toolbox for hybrid system verification. It implements the idea of robust test generation and coverage [1], i.e., covering the initial-state set with the interior of level sets of bisimulation functions, so that reachability/safety properties of a compact set of initial states can be obtained from a finite number of simulations. This approach scales well for high-dimensional systems unlike state set gridding approaches. In this poster, we present key features and functionalities of the toolbox.

2. FEATURES AND FUNCTIONALITIES

State-space exploration by simulating trajectories is used widely for analyzing dynamical systems due to its scalability despite being incomplete and informal. To bridge the divide between simulation and formal verification, tools that combine simulation with some formal analysis are recently being developed [2, 3]. In the similar spirit, we have been developing STRONG, which has the following key functionalities.

•*Model Consistency Checking*
Detecting and correcting some forms of ill-posed in the model.
•*Trajectory Simulation*
For every trajectory of hybrid systems with either linear or nonlinear dynamics, the tool gathers the information about its continuous evolution, transition events, and event times.
•*Robustness Computation*
The trajectory robustness bounds in time and space are computed for each segment of the simulated trajectory within every discrete location. This functionality, including the computation of bisimulation functions using YALMIP toolbox [4], can be automated for linear affine dynamics.
•*Initial Set Coverage Strategy*
The portion of the initial set covered by a robust ball around a simulated initial state leads to trajectories with the same safety and reachability properties as the simulated one. Covering the compact initial set effectively using simulated trajectories and their robust neighborhoods requires smartly choosing initial states for the simulated trajectories. Currently, the strategy implemented is to generate random points as initial states.
•*Parallelization of Robust Simulations*
The robust simulations can be performed independently of each other and the initial set can be covered in a highly parallel manner.
•*Graphical Output*
The results of robust simulations, including trajectories and robust balls, can be visualized for 2-D systems.

We summarize distinct features of the toolbox as below:
+ Robust test generation provides a formal bound on the trajectory divergence used by the tool.
+ The tool is simulation-based, and does not use gridding; therefore it scales well for high-dimensional systems.
+ The tool supports multi-core parallel computation using MATLAB's Parallel Computing Toolbox to achieve further speedup.

Acknowledgments

YD and AAJ would like to acknowledge the support of NSF CAREER grant CNS-0953976. AR would like to acknowledge the support of NSF grants CNS-1035800 and CCF-0926181.

3. REFERENCES

[1] Julius, A.A., Fainekos, G.E., Anand, M., Lee, I., Pappas, G.J.: Robust test generation and coverage for hybrid systems. In: Hybrid Systems: Computation and Control. Springer (2007) 329–342
[2] Donzé, A.: Breach, a toolbox for verification and parameter synthesis of hybrid systems. In: CAV. (2010) 167–170
[3] Annapureddy, Y.S.R., Liu, C., Fainekos, G.E., Sankaranarayanan, S.: 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., Springer (2011) 254–257
[4] Lofberg, J.: Yalmip : A toolbox for modeling and optimization in matlab. In: IEEE Intl. Symp. on Computer Aided Control Systems Design. (2004) 284–289