Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems

Alexandre Donzé

Verimag Laboratory 2, Avenue de Vignates, 38610 Gières France
donze@imag.fr
http://www-verimag.imag.fr/~donze/tool.shtml

Abstract. We describe `Breach`, a Matlab/C++ toolbox providing a coherent set of simulation-based techniques aimed at the analysis of deterministic models of hybrid dynamical systems. The primary feature of `Breach` is to facilitate the computation and the property investigation of large sets of trajectories. It relies on an efficient numerical solver of ordinary differential equations that can also provide information about sensitivity with respect to parameters variation. The latter is used to perform approximate reachability analysis and parameter synthesis. A major novel feature is the robust monitoring of metric interval temporal logic (MITL) formulas. The application domain of `Breach` ranges from embedded systems design to the analysis of complex non-linear models from systems biology.

1 Introduction

Model-based analysis and design techniques for complex systems with parameters uncertainty rely mostly on extensive simulation. Hybrid systems feature a mix of continuous and discrete components and most often their number of possible behaviors is infinite, rendering formal design by exhaustive simulation impossible. Instead, reachability analysis is used to generate over-approximations of the set of possible behaviors to prove that they all satisfy a given property. Efficient techniques and tools exist for hybrid systems with linear continuous dynamics [ADF+06] but to the best of our knowledge, no tool can be readily scalable for hybrid non-linear dynamics, as can be, for instance, simulation. Hence the original idea (also following [KKMS03, GP]) that lead to the development of `Breach` was to estimate dense sets reachable by the system based only on a finite (though possibly large) number of simulations. `Breach` implements this idea and was used, e.g., to produce the results presented in [DKR, DCL09]. It has now matured into a more general exploration tool for hybrid dynamical systems with uncertain parameters, with a convenient graphical user interface and the possibility to write MITL formulas and efficiently monitor their satisfaction robustness.

2 Hybrid Systems Definition and Simulation

`Breach` deals with piecewise-continuous hybrid systems specified as

T. Touili, B. Cook, and P. Jackson (Eds.): CAV 2010, LNCS 6174, pp. 167–170, 2010.
© Springer-Verlag Berlin Heidelberg 2010

$$\begin{cases} \dot{\mathbf{x}} = f(q, \mathbf{x}, \mathbf{p}),\ \mathbf{x}(0) = \mathbf{x}_0 \\ q^+ = e(q^-, \lambda),\ q(0) = q_0 \\ \lambda = \text{sign}(g(\mathbf{x})) \end{cases} \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{p} \in \mathcal{P} \subset \mathbb{R}^{n_p}$ is the *parameter vector*, $q \in \mathcal{Q}$ is the discrete state, g is the guard function mapping $\mathbb{R}^n$ to $\mathbb{R}^{n_g}$, and sign is the usual sign function extended to vectors. The function e is the *event function* which updates the discrete state when the sign of one component of g changes. A trajectory $\xi_{\mathbf{p}}$ is a function from $\mathbb{T} = \mathbb{R}^+$ to $\mathbb{R}^n$ satisfying (1) for all t in $\mathbb{T}$. For convenience, the initial state $\mathbf{x}_0$ is included in the parameter vector $\mathbf{p}$. Thus $\xi_{\mathbf{p}}(0) = \mathbf{x}_0 = (x_{0,1}, x_{0,2}, \ldots, x_{0,n})$ where for all $i \leq n$, $x_{0,i} = p_i$. `Breach` implements a standard discontinuity locking method for the simulation of such systems, based on CVodes ODE solver[1]. *Sensitivity analysis* consists in measuring the influence of a parameter change $\delta\mathbf{p}$ on a trajectory $\xi_{\mathbf{p}}$. A first-order approximation can be obtained by the Taylor expansion

$$\xi_{\mathbf{p}+\delta\mathbf{p}}(t) = \xi_{\mathbf{p}}(t) + \frac{\partial \xi_{\mathbf{p}}}{\partial \mathbf{p}}(t)\ \delta\mathbf{p} + \varphi(t, \delta\mathbf{p}) \text{ where } \varphi(t, \delta\mathbf{p}) = \mathcal{O}\big(\|\delta\mathbf{p}\|^2\big) \quad (2)$$

The derivative of $\xi_p(t)$ with respect to $\mathbf{p}$ in the right hand side of (2) is called the *sensitivity matrix* and denoted as $S_{\mathbf{p}}(t) = \frac{\partial \xi_{\mathbf{p}}}{\partial \mathbf{p}}(t)$. CVodes implements a common method to compute it for an ODE, by integrating a linear time varying ODE satisfied by $S_{\mathbf{p}}(t)$. For hybrid systems such as (1), the sensitivity equation can be solved between two consecutive events but S_p is discontinuous when a guard is crossed. `Breach` implements the computation of the discontinuity jumps provided that the guard functions are smooth enough at the crossing point (see [DKR] for more details).

3 Main Features Overview

Reachability using sensitivity analysis. The reachable set induced by a set of parameters $\mathcal{P}$ at time t is $\mathcal{R}_t(\mathcal{P}) = \bigcup_{\mathbf{p}\in\mathcal{P}} \xi_{\mathbf{p}}(t)$. We showed in [DKR] that it can be approximated by using sensitivity analysis. Let $\mathbf{p}$ and $\mathbf{p}'$ be two parameter vectors in $\mathcal{P}$ and assume that we computed the trajectory $\xi_{\mathbf{p}}$ and the sensitivity matrix $S_{\mathbf{p}}$ at time t. Then we can use $\xi_{\mathbf{p}}(t)$ and $S_{\mathbf{p}}(t)$ to estimate $\xi_{\mathbf{p}'}(t)$. We denote this estimate by $\hat{\xi}^{\mathbf{p}}_{\mathbf{p}'}(t)$. The idea is to drop higher order terms in the Taylor expansion (2), which gives $\hat{\xi}^{\mathbf{p}}_{\mathbf{p}'}(t) = \xi_{\mathbf{p}}(t) + S_{\mathbf{p}}(t)(\mathbf{p}' - \mathbf{p})$. If we extend this estimate to all parameters $\mathbf{p}'$ in $\mathcal{P}$, we get the following estimate for the reachable set $\mathcal{R}_t(\mathcal{P})$: $\hat{\mathcal{R}}^{\mathbf{p}}_t(\mathcal{P}) = \bigcup_{\mathbf{p}'\in\mathcal{P}} \hat{\xi}_{\mathbf{p}'}(t) = \{\xi_{\mathbf{p}} - S_{\mathbf{p}}(t)\mathbf{p}\} \oplus S_{\mathbf{p}}(t)\mathcal{P}$. If the approximation with one trajectory is too coarse, it can be improved by refining $\mathcal{P}$ into smaller sets until an error tolerance factor is satisfied [DKR]. The process is illustrated in Fig.1. It converges quadratically, although the error cannot be formally bounded in general (as is the case with numerical simulation). In [DKR], a local iterative refinement of the parameters boxes for which the reachable set intersects a bad set is used to synthesize sets of trajectories satisfying a safety property.

[1] See `https://computation.llnl.gov/casc/sundials/main.html`

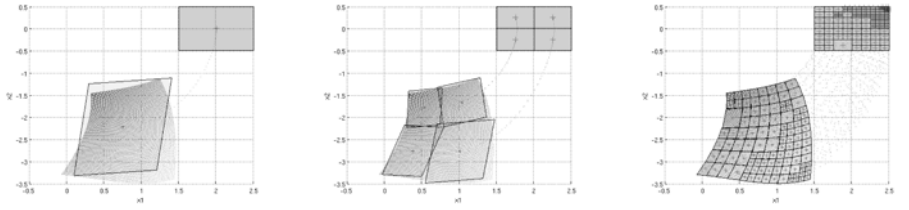

Fig. 1. Approximation of the reachable set for a Van der Pol equation using one trajectory, four trajectories, and an automatic refinement produced by the reachability routine of `Breach` with control of the error

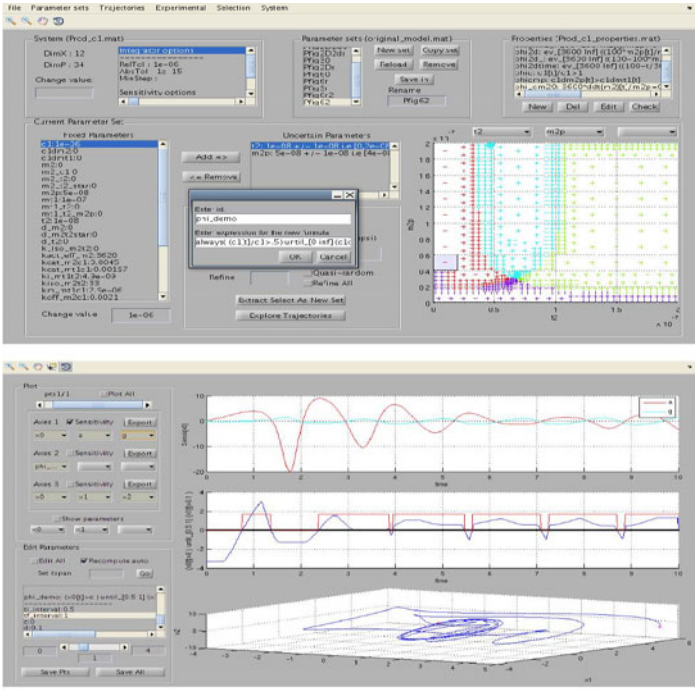

Fig. 2. Top: main window of `Breach`, manipulating parameter sets and properties. Each cross in the plot is for a given parameter vector and all parameters with the same color satisfy the same set of properties. Bottom: Trajectories explorer window. Each plot can display either the evolution of the state variables, their sensitivities or the robust satisfaction of MITL formulas. All parameters can be modified and the trajectories recomputed on-the-fly.

Property-driven parameters synthesis. Recently, we implemented an extension to support formulas of Signal Temporal Logic (STL), an analog extension of the Metric Interval Temporal Logic (MITL) which has the following core grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi_1 \mathcal{U}_I \varphi \tag{3}$$

The specificity of STL lies in the nature of its atomic predicates p, which are expressions of the form $y(x_0, x_1, \ldots, x_n, t) > 0$. A parser allows to specify

textually formulas with a richer grammar. E.g., a valid expression is `(x0[t]>0)=> eventually_[1,2] ((x1[t]-1>0) until_[1,2](-x2[t]-.1>0))`. `Breach` is able to monitor the Boolean satisfaction as well as the quantitative satisfaction. For p, the former is given by the sign of y (similarly to the tool AMT[2]) and the latter is given by $|y|$. For a compound formula the semantics follows that of [FP07]. We are not aware of other tools implementing this feature, except for TaLiRo[3] which `Breach` outperformed in our experiments. In particular, `Breach` does not suffer from the memory explosion problem reported in the user guide of Taliro for larger formulas. The computational time is experimentally linear in the size of the trace and the size of the formula, which is a light overhead to the cost of the simulation (see `Breach` website for examples and data). The use of sensitivity analysis for properties other than safety is still a work in progress, but `Breach` provides heuristics to find separations between parameter regions satisfying different properties. The GUI is shown on Fig. 2. In addition to defining systems and interfacing the above mentioned methods, it allows to explore the behaviors and monitor their properties by tuning the parameters on-the-fly.

4 Discussion and Future Work

`Breach` is still in very active development. For the moment, the GUI allows to specify models as general nonlinear ODES. For hybrid systems, the user still has to provide write small portions of C code to implement transitions (an example is given on the web site). Also, although in [DKR] we demonstrated the feasibility of analysing Simulink models with `Breach` , this feature still need some work in particular when the system is hybrid.

References

[ADF+06] Asarin, E., Dang, T., Frehse, G., Girard, A., Le Guernic, C., Maler, O.: Recent progress in continuous and hybrid reachability analysis (2006)

[DCL09] Donzé, A., Clermont, G., Langmead, C.: Parameter synthesis for nonlinear dynamical systems: Application to systems biology. Journal of Computational Biology 410(42) (2009)

[DKR] Donzé, A., Krogh, B., Rajhans, A.: Parameter synthesis for hybrid systems with an application to simulink models. In: Majumdar, R., Tabuada, P. (eds.) HSCC 2009. LNCS, vol. 5469, pp. 165–179. Springer, Heidelberg (2009)

[FP07] Fainekos, G., Pappas, G.: Robust sampling for mitl specifications. In: Raskin, J.-F., Thiagarajan, P.S. (eds.) FORMATS 2007. LNCS, vol. 4763, pp. 147–162. Springer, Heidelberg (2007)

[GP] Girard, A., Pappas, G.: Verification Using Simulation. In: Hespanha, J.P., Tiwari, A. (eds.) HSCC 2006. LNCS, vol. 3927, pp. 272–286. Springer, Heidelberg (2006)

[KKMS03] Kapinski, J., Krogh, B.H., Maler, O., Stursberg, O.: On systematic simulation of open continuous systems. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, pp. 283–297. Springer, Heidelberg (2003)

[2] See `http://www-verimag.imag.fr/DIST-TOOLS/TEMPO/AMT/content.html`

[3] See `http://www.public.asu.edu/~gfaineko/taliro.html`