ISSN 2186-7437

NII Shonan Meeting Report

No. 2012-2

Hybrid Systems
Theory and Practice, *Seriously*

Ichiro Hasuo
Takuro Kutsuna
Toshimitsu Ushio

April 22–26, 2012

NII 湘南会議 SHONAN MEETING

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Hybrid Systems
Theory and Practice, *Seriously*

Organizers:
Ichiro Hasuo (University of Tokyo)
Takuro Kutsuna (Toyota Central R&D Labs.)
Toshimitsu Ushio (Osaka University)

April 22–26, 2012

Hybrid Systems Hybrid systems—those which exhibit both continuous "flow" and discrete "jump" dynamics—are everywhere in the modern world, with cars, airplanes and all others controlled by computers. Their failure can therefore have an immense impact on human lives and infrastructures, posing the problem of their quality assurance—getting hybrid systems right—as a pressing one.

Need of Hybrid Research Community for Hybrid Systems The name *hybrid system* itself manifests a research challenge: due to the heterogeneity of its dynamics as well as the diversity of its applications, no research effort is comprehensive if it stays within the realm of a single, already established, research discipline. Currently there are two theoretical "camps" aiming at hybrid applications:

- *control theory*, originally specialized in flow dynamics with the tool of differential equations; and
- theory of *system verification*, with its original goal to tame the astronomically complex jump dynamics of computer systems.

Obviously the two communities should join forces and bring their knowledge together, towards the goal of establishing a solid theoretical ground for hybrid systems.

One should not forget about another very important group: practitioners in the industry who are struggling with vastly complex hybrid systems and concrete quality assurance criteria.

Informal and Fruitful Mixture of Industry and Academia This SHONAN Meeting aimed to serve as a meeting point of these three camps—two theoretical ones that have been developed rather separately (up to now, to our regret), and the community of practitioners with whom theoreticians have not much contact. With a remarkably informal atmosphere—featuring four tutorials and many discussion sessions—the meeting established a common ground among participants. In particular, several participants from the industry brought their unique viewpoints to the attention of the participants from academia. What is valued in the industrial practice is certainly not the same as that in academia; and this poses a difficult but rewarding challenge to the academic research.

Overview of Talks

Tutorial on Control Theoretic Approach to Hybrid Systems

Ian Mitchell (U. British Columbia)

Hybrid systems: From industrial perspective (Position paper for tutorial)

Masataka Nishi (Hitachi Ltd.)

Mission-critical industrial systems are hybrid by design. Steadily growing size and complexity of the systems have highlighted that broad adoption of formal methods as integral part of design procedure can assure functional integrity of the systems. However, three brick walls reside in my sight.

The first one is a legal wall of defect liability, as the industry is reluctant to sharing proprietary design information, despite that they are indispensable as input to employ formal techniques. Theoreticians can sidestep the first wall by building ones own working example by following a standard system engineering practice, say IEC61508 or its relevant. I show a sanitized fault tolerant architecture of the industrial systems and show that major functional requirements, which include reliability, functional integrity, and correctness of fault handling mechanism, are automatically translated into formal verification problems.

The second wall is lack of formal language that bridges a gap between imperative nature of implementation and declarative nature of specification. Any challenges of inventing the formal language for hybrid systems should be capable of coherently integrating them, because the industry wants a formally verified product written in the imperative form, instead of a formally verified abstract model written in the declarative form. A language may fit in the need that states a reachability test based on hybrid state machine representation. Eventually, control logic design is continuous part of hybrid systems whose verification problem can be interpreted as reachability of state governed by nonlinear dynamics, while temporal logic design is discrete part of hybrid systems whose verification problem can be formulated as reachability of state governed by discrete transition logic.

The third wall is lack of scalable computational technique that is applicable to formal verification of the implementation of industrial systems. As a target of system validation is the implementation instead of the formally verified abstract model, the technique should be able to look into a sufficient detail in the implementation subject to affordable computational cost.

Tutorial on Stochastic Hybrid Models

Manuela Luminita Bujorianu (University of Manchester)

Tutorial on Formal Verification

Ichiro Hasuo (University of Tokyo)

“Formal verification” means giving a mathematical proof for the correctness of systems/programs. In this tutorial I explain the very basics of the theorem proving approach (also called the deductive approach) to formal verification, in as elementary terms as possible. Specifically the methodology described here is that of Hoare-style program logic; the emphasis is on how formal logic allows one to construct a correctness proof in a purely syntactic manner.

HydLa: A High-Level Language for Hybrid Systems

Kazunori Ueda (Waseda U.)

We have been working on the design and implementation of HydLa, a high-level modeling language for hybrid systems. Its objectives and features include a constraint-based (as opposed to automata-based) formalism, proper handling of uncertanties, and nondeterministic execution. The talk will describe an overview of the language with live demonstration of our prototype implementation.

Binary Decision Diagram meets Machine Learning

Takuro Kutsuna (Toyota Central R&D Labs.)

The binary decision diagram is a compressed representation of a Boolean formula and has been applied in many fields including model checking. In this talk, we will propose a novel approach to learn a one-class classifier that is build on binary decision diagram techniques.

Computing the viability kernel using maximal reachable sets

Ian Mitchell (U. British Columbia)

We present a connection between the viability kernel and maximal reachable sets. Current numerical schemes that compute the viability kernel suffer from a complexity that is exponential in the dimension of the state space. In contrast, extremely efficient and scalable techniques are available that compute maximal reachable sets. We show that under certain conditions these techniques can be used to conservatively approximate the viability kernel for possibly high-dimensional systems. We demonstrate the results on two practical examples, one of which is a seven-dimensional problem of safety in anesthesia.

The paper is available at: http://dx.doi.org/10.1145/2185632.2185644

Some Perspectives on Hybrid Systems Model Checking

Shaofa Yang (SIAT, Chinese Academy of Sciences)

This talk summarizes results in [1,2] on model checking the discrete-time behaviour of a class of hybrid systems, in which sensing and actuation incur bounded delays. It is shown in [1] that, even for very restricted dynamics ($dx_i/dt = ax_i + b$), simple model checking problems become undecidable. Nevertheless, for a restricted setting in which every continuous variable evolves either at possibly different constant rates ($dx_i/dt = b$) in all modes, or at possibly different exponential rates ($dx_i/dt = ax_i$) in all modes, the set of control state sequences can be represented by a finite state machine and consequently various model checking problems become decidable. In [2], for a network of hybrid systems in which every continuous variable evolves at possibly different constant rates ($dx_i/dt = b$) in all modes, a succinct representation of the global discrete-time behaviour is constructed, and it can serve as a basis for dealing with the state space explosion issue. This construction is based on the observation that in such a network, each component only senses a small subset of continuous variables, and updates the evolution rates of another small subset of continuous variables. The talk is concluded by a sketch of ongoing work, with S. Wang, N. Zhan, C. Zhou, towards verifying operating scenarios of train control systems, by exploiting the quantifier-elimination procedure for the first-order theory of the real field.

[1] (with M. Agrawal, F. Stephan, P.S. Thiagarajan) Behavioural approximations for restricted linear differential hybrid automata. In Proc. of Hybrid Systems: Computation and Control 2006.

[2] (with P.S. Thiagarajan) Modular discrete time approximations of distributed hybrid automata. Theoretical Comp. Sci. 2012.

Temporal Logic Testing for Hybrid Systems

Georgios Fainekos (Arizona State University)

One of the important challenges in Model Based Development of hybrid systems is the problem of verification of functional system properties. In its general form, the problem is undecidable. S-TaLiRo is a Matlab (TM) toolbox that searches for trajectories that falsify temporal logic properties of models of hybrid systems. At the heart of the tool, we use randomized testing based on stochastic optimization techniques including Monte-Carlo methods, Ant-Colony Optimization and the Cross Entropy method. 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 of control software. We present the architecture of S-TaLiRo and its working on application examples.

On model-based testing of hybrid systems

Thao Dang (VERIMAG)

Programming with Infinitesimals: A While-Language for Hybrid System Modeling

Kohei Suenaga (Kyoto U.)

We add, to the common combination of a WHILE-language and a Hoare-style program logic, a constant dt that represents an infinitesimal (i.e. infinitely small) value. The outcome is a framework for modeling and verification of hybrid systems: hybrid systems exhibit both continuous and discrete dynamics and getting them right is a pressing challenge. We rigorously define the semantics of programs in the language of nonstandard analysis, on the basis of which the program logic is shown to be sound and relatively complete. (Joint work with Ichiro Hasuo, U. Tokyo. The paper appeared in ICALP 2011)

Interrupt Race Condition Detection using Multiple Static Code Analysis Methods

Yutaka Inamori (Toyota Central R&D Labs.)

Interrupt handlers are used in vehicle control programs for high responsiveness but are a possible cause of data races. The present paper describes a detection method for interrupt race conditions that produces no false negatives and a smaller number of false positives. The proposed method is characterized by a mechanism whereby the masses of false positives are sifted through using five types of static code analysis methods that are free from false negatives.

Numerical stability analysis of floating-point computations using software model checking

Franjo Ivancic (NEC Labs America)