Continuous KAOS, ASM, and formal control system design across the continuous/discrete modeling interface: a simple train stopping application

Richard Banach¹, Huibiao Zhu², Wen Su² and Runlei Huang³

¹ School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK
² Software Engineering Institute, East China Normal University, 3663 Zhongshan Road North, Shanghai 200062, People's Republic of China. E-mail: hbzhu@sei.ecnu.edu.cn; wensu@sei.ecnu.edu.cn
³ Alcatel-Lucent Shanghai Bell, 388 Ningqiao Road, Pudong Jinqiao, Shanghai 201206, People's Republic of China. E-mail: runleihuang@alcatel-sbell.com.cn

Abstract. A very simple model for train stopping is used as a vehicle for investigating how the development of a control system, initially designed in the continuous domain and subsequently discretized, can be captured within a formal development process compatible with standard model based refinement methodologies. Starting with a formalized requirements analysis using KAOS, an abstract model of the continuous system is created in the ASM formalism. This requires extensions of the KAOS and ASM formalisms, capable of dealing with quantities evolving continuously over real time, which are developed. After considering how the continuous system, described as a continuous control system in the state space framework, can be discretized, a discrete control system is created in the state space framework. This is re-expressed in the ASM formalism. The rigorous results on the relationship between continuous and discrete control system models that are needed to establish provable properties of the discretization, then become the ingredients of a retrenchment between continuous and discrete ASM models, and are thus fully integrated into the formal development. The discrete ASM model can then be further refined towards implementation.

Keywords: Continuous KAOS, Continuous ASM, Control systems, Rigorous design, Refinement, Retrenchment, Continuous modeling, Discrete modeling, Train control

Correspondence and offprint requests to: R. Banach, E-mail:banach@cs.man.ac.uk

The majority of the work reported in this paper was done while Richard Banach was a visiting researcher at the Software Engineering Institute at East China Normal University. The support of ECNU is gratefully acknowledged.

Huibiao Zhu is supported by National Basic Research Program of China (No. 2011CB302904), National High Technology Research and Development Program of China (No. 2011AA010101 and No. 2012AA011205), National Natural Science Foundation of China (No. 61061130541 and No. 61021004).
1. Introduction

Conventional model based formal refinement technologies (see for example [dRE98, DB01, SS98, Abr96, PST96, WD96, Abr10]) are based on purely discrete mathematical and logical concepts. These turn out to be ill suited to modeling—still less, to formally developing—applications whose usual models are best expressed using continuous mathematics. Nevertheless, many such applications, control systems in particular, are these days implemented using digital techniques, despite being designed in the continuous domain. To the extent that such systems can be high consequence (e.g. avionics systems, nuclear power control systems, weapons command and control systems, automated public transport systems, medical instrument systems), the dependability to be gained by utilizing formal techniques during their development is of course a highly desirable addition to the development process [BH99a, BH99b, Hal90, BH95, Hal07]. This raises a dilemma: the conflict between the acceptance of the intrinsic desirability of using formal techniques, and the recognition that there are significant technical obstacles to their direct application in the natural problem domain. In practice this usually results in some sort of “avoiding the problem” for the majority of cases.

In the vast majority of scenarios, engineering rules of thumb are used to guide the implementation of a continuous control design by a discrete controller (i.e. one which reacts to inputs and determines outputs at regularly occurring instants corresponding to the sampling frequency, instead of continuously). Such implementations are evaluated principally by testing [Bro10], the more so when the application is not viewed as being of high criticality. Even in relatively critical applications, such as cruise control for vehicles, present day energy saving considerations place pressure on the design to reduce the sampling frequency to the minimum that will suffice for the application, thus forcing the sampling frequency to vary according to real-time system parameters [But]. Again, the evaluation of such designs is empirical and heuristic.

In more critical applications, what usually happens is that even if formal development techniques are utilized, their deployment takes place only after the design and modeling process has crossed the continuous to discrete watershed [Mey]. This is, of course, better than nothing, but it is hardly ideal.

In the light of the above considerations, and in view of increasing pressure to optimise design parameters including sampling frequency, it would clearly be of benefit to the development of this kind of system, if the rigour of formal techniques could be applied also to the continuous to discrete design transformation, supplementing the informal techniques routinely used.

In this paper, we tackle the mismatch between continuous modeling and discrete development techniques, head on. Although the traditional discrete development technique of choice, model based refinement (in one or other of its guises), is too exacting as regards how close a system model has to be to its successor in the development process for refinement to have the capacity to straddle the continuous to discrete demarcation line (at least in the most general case), a judicious weakening of it, retrenchment, proves to be adaptable enough to do the job.

Retrenchment, being an intrinsically weaker notion than refinement, thus possessing weaker generic properties, is best only used where refinement is inapplicable, or where the application of refinement would be so unnatural that it would risk derailing the development strategy. Accordingly, the ideal development methodology (from our perspective) is to combine refinement steps (where these can be made to work convincingly), with retrenchment steps (in those parts of the development where refinement will not work well). The whole process needs to be consistent of course. This consistency is handled by the Tower theorems [BJ, Jes05], which show how retrenchment and refinement steps can coexist in a consistent whole.

In this paper we tackle the continuous to discrete issue, by taking a simple running example, one that can be solved fully by analytic means in both the continuous and discretized domains, and tracing it through a full scale formal development process. This not only shows how the discretization problem may be tackled per se, but also exhibits the entire development process end-to-end, showing how the various technical ingredients fit together.

Thus we start with requirements analysis. In the context of our simple example, this entails eliciting the requirements in the continuous domain. Proceeding from top level requirements, we engage in a requirements refinement activity, albeit a rather simple one suited to the context of our example. So the top level requirements get refined to a model in the form of a continuous control problem. At this point our treatment of discretization kicks in. We remodel the continuous control problem to a discretization of it as a discrete control problem, and derive a description of the discretization step via a suitable retrenchment. Now firmly in the discrete world, we finally illustrate how a more complex development in this style would go, by refining further the discrete control problem towards implementation.

Discretizations in general are very challenging as regards the obtaining of precise and reliable quality metrics for their behaviour. This explains the overwhelming preponderance of informal and heuristic approaches to their