Safe, Automated and Formal Synthesis of Digital Controllers for Continuous Plants

Alessandro Abate
University of Oxford

Iury Bessa
Federal University of Amazonas, Brazil

Dario Cattaruzza
University of Oxford

Lucas Cordeiro
University of Manchester

Cristina David
University of Cambridge

Pascal Kesseli
University of Oxford

Daniel Kroening
University of Oxford

Elizabeth Polgreen
University of Oxford

We present a sound and automated approach to synthesizing safe, digital controllers for physical plants represented as linear, time-invariant models. The synthesis accounts for errors caused by the digitization effects introduced by digital controllers operating in either fixed- or floating-point arithmetic. Our approach uses counterexample-guided inductive synthesis (CEGIS): in the first phase an inductive generalisation engine produces a possible solution that is safe for some possible initial conditions but may not be safe for all initial conditions. Safety for all initial conditions is then verified in a second phase either via bounded model checking or abstract acceleration; if the verification step fails, a counterexample is provided to the inductive generalisation and the process iterates until a safe controller is obtained. We implemented our approach in a tool named DSSynth (Digital-System Synthesizer) and demonstrate its practical value by automatically synthesizing safe controllers for physical plant models from the digital control literature.

1 Introduction

Embedded control systems using fixed and floating-point arithmetic have become widespread as the availability of low-cost devices that can perform highly non-trivial control tasks has increased. Correct synthesis of control software for such platforms is non-trivial due to the digital representation of continuous quantities used by the controller. This representation introduces errors due to finite-precision arithmetic, time discretization and A/D - D/A conversions. Given an LTI model of a physical (continuous) plant, we present two automated approaches for generating correct-by-construction digital controllers that address all these challenges and satisfy a safety property for the physical plant. Both approaches make use of CounterExample-Guided Inductive Synthesis (CEGIS) [4, 6]. CEGIS is an iterative process, where each iteration performs inductive generalisation based on counterexamples provided by a verification module. Our two instantiations of CEGIS are described next.

The first approach starts by devising a digital controller that stabilizes the system's model, while remaining safe for a pre-selected time horizon and a single initial state; then, it verifies unbounded-time safety by unfolding the dynamics of the LTI model, considering the full set of initial states, and checking a completeness threshold [5].*The second approach* employs *abstract acceleration* [3] to evaluate all possible progressions of the LTI model simultaneously. This approach uses *abstraction refinement*, enabling us to start with a simple description regardless of the dynamics complexity, and only expand to more complex models when a solution cannot be found.

We provide experimental results showing that both our approaches are able to synthesize safe controllers for a set of physical plant models taken from digital control literature.

This abstract contains material published in [1, 2].

Submitted to:
SYNT 2018

References

[1] Alessandro Abate, Iury Bessa, Dario Cattaruzza, Lennon Chaves, Lucas C. Cordeiro, Cristina David, Pascal Kesseli, Daniel Kroening & Elizabeth Polgreen (2017): *DSSynth: an Automated Digital Controller Synthesis Tool for Physical Plants*. In: *International Conference on Automated Software Engineering (ASE)*, IEEE Computer Society, pp. 919–924, doi:10.1109/ASE.2017.8115705.

[2] Alessandro Abate, Iury Bessa, Dario Cattaruzza, Lucas C. Cordeiro, Cristina David, Pascal Kesseli, Daniel Kroening & Elizabeth Polgreen (2017): *Automated Formal Synthesis of Digital Controllers for State-Space Physical Plants*. In: *Computer Aided Verification (CAV)*, *LNCS* 10426, Springer, pp. 462–482, doi:10.1007/978-3-319-63387-9_23. Available at `https://doi.org/10.1007/978-3-319-63387-9_23`.

[3] Dario Cattaruzza, Alessandro Abate, Peter Schrammel & Daniel Kroening (2015): *Unbounded-Time Analysis of Guarded LTI Systems with Inputs by Abstract Acceleration*. In: *International Symposium on Static Analysis (SAS)*, *LNCS* 9291, Springer, pp. 312–331.

[4] Susmit Jha, Sumit Gulwani, Sanjit A. Seshia & Ashish Tiwari (2010): *Oracle-guided component-based program synthesis*. In: *International Conference on Software Engineering – Volume 1 (ICSE)*, ACM, pp. 215–224, doi:10.1145/1806799.1806833. Available at `http://doi.acm.org/10.1145/1806799.1806833`.

[5] Daniel Kroening & Ofer Strichman (2003): *Efficient Computation of Recurrence Diameters*. In: *Verification, Model Checking, and Abstract Interpretation (VMCAI)*, *LNCS* 2575, Springer, pp. 298–309.

[6] Armando Solar-Lezama, Liviu Tancau, Rastislav Bodík, Sanjit A. Seshia & Vijay A. Saraswat (2006): *Combinatorial sketching for finite programs*. In: *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, ACM, pp. 404–415.