Bounded Model Checking of State-Space Digital Systems

The impact of finite word-length effects on the implementation of fixed-point digital controllers based on state-space modeling

Felipe R. Monteiro
Federal University of Amazonas
Manaus, Amazonas, Brazil
felipemonteiro@ufam.edu.br

ABSTRACT

The extensive use of digital controllers demands a growing effort to prevent design errors that appear due to finite-word length (FWL) effects. However, there is still a gap, regarding verification tools and methodologies to check implementation aspects of embedded systems. Thus, the present paper describes an approach, which employs bounded model checking (BMC) techniques, to verify fixed-point digital controllers represented by state-space equations. The experimental results demonstrate the sensitivity of such systems to FWL effects and the reliability of the proposed approach to detect them. To the best of my knowledge, this is the first report tackling formal verification through BMC of fixed-point state-space digital controllers.

CCS Concepts

•**Computer systems organization** → **Real-time systems;** *Embedded systems;* •**Software and its engineering** → **Model checking;** Formal methods; •**Theory of computation** → *Verification by model checking;*

Keywords

Real-time Systems; Model Checking; State-Space; Formal Verification; Digital Controllers.

1. MOTIVATION

In real-time systems, digital controllers are algorithms that manipulate digital signals, in order to influence the behavior of a system [1]; it can be mathematically expressed as difference equations, transfer functions, or state-space equations. In the last decades, such systems has been broadly applied to a variety of applications, but, most importantly, it is core to manage computing systems and networks [2]. Indeed, digital-control systems present many advantages, if compared with analog ones, such as more flexibility, scalability, adaptability, and lower implementation cost. However, such systems are vulnerable to finite word-length (FWL) effects [3, 4], which can cause several quantization problems, such as truncation or round-off errors.

In addition, digital systems can also be implemented in several platforms, such as field programmable gate arrays (FPGA) devices [5] and digital signal processors [6], among other kinds of microprocessors. Importantly, each one of these platforms can manipulate and represent numbers using different formats and arithmetics (*e.g.*, number of bits, fixed- or floating-point arithmetic), which can directly affect the performance and precision of the digital-control system [7]. Additionally, fixed-point processors present high processing speed with reduced cost, which makes them a valuable choice for designing digital controllers; nonetheless, such an approach might lead to more nonlinearities, round-off errors, and overflows. Therefore, control engineers invest a huge amount of time and effort in the design phase of such systems, in order to avoid complications caused by FWL effects. In this context, the following issue can be raised:

- *How can control engineers detect problems on fixed-point state-space digital controllers caused by finite word-length effects?*

In order to answer that research question, this paper proposes a verification methodology based on bounded model checking (BMC) techniques [8], which verifies properties on state-space digital controllers, by means of a verification tool named as Digital-Systems Verifier (DSVerifier). It is worth noting that the approach described in this paper extends a previous work [7, 9, 10, 11, 12], which was only able to check properties in digital systems represented by transfer function; however, such an approach is extremely limited. Indeed, the support for state-space systems allows a better insight about the internal system behavior, which enables the verification of new properties (*e.g.*, controllability and observability) and considers initial conditions for system analysis [13]. In addition, DSVerifier now supports two efficient model-checking tools as back-end: ESBMC [14, 15] (previously supported) and CBMC [16, 17].

2. BACKGROUND AND RELATED WORK

In order to deal with FWL effects on digital systems, some approaches suggest special metrics, search algorithms or methodologies to achieve an optimal word-length and avoid FWL effects [18, 19, 20, 21, 22, 23]. There are also simulation tools (*e.g.*, LabVIEW [24] and MATLAB [25]), which are traditionally used by control engineers. However, such approaches depend on input stimulation to evaluate the

FSE '16, Seattle, WA, USA

DOI: 10.475/123_4

state-space of a system, which might not exploit all possible conditions that a system can exhibit. In contrast, Alur *et al.* [26, 27] proposed the prior automated verification approaches, regarding model checking, which inspired the development of other verifiers for cyber-physical systems and hybrid automata (*e.g.*, Maellan [28], Open-Kronos [29], and UPPAAL [30]). Nonetheless, differently from the work presented here, such approaches do not tackle system robustness related to implementation aspects [7, 9, 10].

3. METHODOLOGY

DSVerifier works as front-end for BMC tools (with support to full ANSI-C verification), in order to verify state-space digital systems. As one can see in Figure 1, the verification methodology proposed in this paper is split into two main stages as follows: user and DSVerifier procedures. In the former, the control engineer manually performs steps 1 to 3. Step 1 is related to the design process of a digital system, while step 2 to its implementation details, *i.e.*, numerical representation $< I, Q >$, where I is the number of bits for the integer part, and Q is the number of bits for the fractional part. Then, in step 3 the user chooses a property ϕ to be verified (*e.g.*, *quantization_error*), a maximum verification time, a bound k, and a BMC tool. Importantly, all specifications from the previous steps are detailed in an input file using the same syntax as MATLAB code standard.

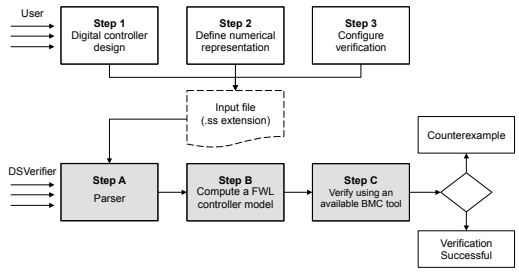

Figure 1: Verification methodology.

After that, DSVerifier receives the respective input file and then performs the verification of the desired property ϕ; it is worth noting that steps A to C are completely automatic. In step A, DSVerifier builds an intermediate ANSI-C code for the digital system implementation and then, in Step B, it formulates a FWL model to be used as input for the underlying model-checker. Finally in the step C, the translation of the resulted ANSI-C code into SAT or SMT formulae is completed, by a highly efficient bounded model-checking tool (*e.g.*, ESBMC or CBMC) [14, 16]. Here, DSVerifier symbolically checks a given property ϕ w.r.t. digital systems. If any violation is found, then DSVerifier reports a counterexample, which contains system inputs that lead to a failure. A successful verification result is reported if the system is safe w.r.t. ϕ up to a bound k.

As aforementioned, DSVerifier supports the verification of the following properties regarding quantized digital system:
Quantization error: it checks whether the output quantization is inside a tolerable bound.
Stability: it checks digital-system stability using the Eigen Library [31];
Controllability: it checks whether a digital system M is controllable, based on the rank of its controllability matrix.
Observability: it checks whether a digital system M is observable, based on the rank of its observability matrix.

It is worth noting that all numerical operations are performed through fixed-point arithmetic, according to a certain precision set by the user, and all properties are sound and complete. In addition, all aforementioned verifications can be performed in a closed-loop configuration.

4. PRELIMINARY RESULTS

For the following evaluation, an automatic test-suite was developed, with 25 digital systems[1] [2, 32]. In particular, this study employs CBMC $v5.4$, with the SAT solver MiniSAT $v2.2.0$ [33]. All systems are checked against four properties, as described in Section 3, using a 32-bits microcontroller hardware configuration with three precisions (8, 16, and 32-bits), which results in 300 verifications.

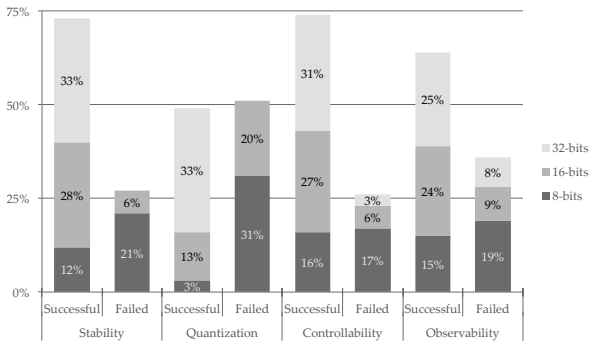

Figure 2: Experimental results.

Indeed, all components of the test-suite are stable, controllable, and observable; however, based on the experimental results shown in Figure 2, one may noticed that *(i)* the properties of a digital system might not be held, once quantization errors affect its representation, and *(ii)* the lower the precision, the higher its sensibility to FWL effects. In addition, all 300 verifications were performed in approximately 7 hours. Finally, the failed cases were validated with Simulink [34], using the respective counterexample.

Contributions. Particularly, this work makes four major contributions: **(i)** support for state-space representations, **(ii)** verification of quantization error for single-input and single-output (SISO) systems [1], **(iii)** stability (for state-space systems), controllability and observability verifications for SISO and multi-input and multi-output systems [1], and **(iv)** closed-loop verification for the aforementioned properties. To the best of my knowledge, this is the first report addressing formal verification through BMC of fixed-point digital controllers, based on the state-space representation. In future, other properties and BMC tools will be integrated into DSVerifier, in addition to support for systems with uncertainties.

5. ACKNOWLEDGMENTS

Special thanks to my supervisors (and colleagues) Lucas C. Cordeiro, Iury V. Bessa, Hussama I. Ismail, and Eddie B. de Lima Filho for their valuable support.

[1]DSVerifier, all benchmarks, and a detailed test evaluation are available at www.dsverifier.org

6. REFERENCES

[1] K. Ogata. *Modern Control Engineering*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 4^{th} edition, 2001.

[2] T. Abdelzaher, Y. Diao, J. L. Hellerstein, C. Lu, and X. Zhu. *Introduction to Control Theory And Its Application to Computing Systems*, pages 185–215. Springer US, Boston, MA, 2008.

[3] Y. Guang-Hong, G. Xiang-Gui, C. Wei-Wei, and G. Wei. *Linear Systems: Non-Fragile Control and Filtering*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2013.

[4] R. Istepanian and J. F. Whidborne. *Digital Controller Implementation and Fragility: A Modern Perspective*. Springer-Verlag London, London, UK, 1st edition, 2001.

[5] E. Monmasson and M. N. Cirstea. Fpga design methodology for industrial control systems: A review. *IEEE Transactions on Industrial Electronics*, 54(4):1824–1842, Aug 2007.

[6] M. K. Masten and I. Panahi. Digital signal processors for modern control systems. *Control Engineering Practice*, 5(4):449 – 458, 1997.

[7] I. V. Bessa, H. I. Ismail, L. C. Cordeiro, and J. E. C. Filho. Verification of fixed-point digital controllers using direct and delta forms realizations. *Design Automation for Embedded Systems*, 20(2):95–126, 2016.

[8] E. M. Clarke, E. A. Emerson, and J. Sifakis. Model checking: Algorithmic verification and debugging. *Commun. ACM*, 52(11):74–84, November 2009.

[9] H. I. Ismail, I. V. Bessa, L. C. Cordeiro, E. B. de Lima Filho, and J. E. Chaves Filho. *DSVerifier: A Bounded Model Checking Tool for Digital Systems*, pages 126–131. Springer International Publishing, Cham, 2015.

[10] B. R. Abreu, Y. M. R. Gadelha, C. L. Cordeiro, B. E. de Lima Filho, and S. W. da Silva. Bounded model checking for fixed-point digital filters. *Journal of the Brazilian Computer Society*, 22(1):1–20, 2016.

[11] I. V. d. Bessa, H. I. Ismail, L. C. Cordeiro, and J. E. C. Filho. Verification of delta form realization in fixed-point digital controllers using bounded model checking. In *2014 Brazilian Symposium on Computing Systems Engineering*, pages 49–54, Nov 2014.

[12] I. Bessa, R. Abreu, J. E. Filho, and L. Cordeiro. Smt-based bounded model checking of fixed-point digital controllers. In *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pages 295–301, Oct 2014.

[13] F. W. Fairman. *Linear Control Theory: The State Space Approach*. Wiley, 1998.

[14] L. Cordeiro, B. Fischer, and J. Marques-Silva. Smt-based bounded model checking for embedded ansi-c software. *IEEE Transactions on Software Engineering*, 38(4):957–974, 2012.

[15] J. Morse, M. Ramalho, L. Cordeiro, D. Nicole, and B. Fischer. *ESBMC 1.22*, pages 405–407. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[16] D. Kroening and M. Tautschnig. *CBMC – C Bounded Model Checker*, pages 389–391. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[17] E. Clarke, D. Kroening, and F. Lerda. *A Tool for Checking ANSI-C Programs*, pages 168–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[18] R. Middleton and G. Goodwin. Improved finite word length characteristics in digital control using delta operators. *IEEE Transactions on Automatic Control*, 31(11):1015–1021, Nov 1986.

[19] L. Harnefors. Implementation of resonant controllers and filters in fixed-point arithmetic. *IEEE Transactions on Industrial Electronics*, 56(4):1273–1281, April 2009.

[20] J. Carletta, R. Veillette, F. Krach, and Z. Fang. Determining appropriate precisions for signals in fixed-point iir filters. In *Design Automation Conference, 2003. Proceedings*, pages 656–661, June 2003.

[21] R. S. H. Istepanian and J. F. Whidborne. Multi-objective design of finite word-length controller structures. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1, page 68 Vol. 1, 1999.

[22] V. Mohta. The title of the work. Master's thesis, Finite worldlength effects in fixed-point implementations of linear systems, Massachusetts Institute of Technology, 1998.

[23] W. Sung and K. Kum. Simulation-based word-length optimization method for fixed-point digital signal processing systems. *IEEE Transactions on Signal Processing*, 43(12):3087–3090, Dec 1995.

[24] G. W. Johnson. *LabVIEW Graphical Programming: Practical Applications in Instrumentation and Control*. McGraw-Hill School Education Group, 2nd edition, 1997.

[25] Kermit Sigmon. *MATLAB Primer*. CRC Press, 5th edition, 1998.

[26] R. Alur, C. Courcoubetis, and D. Dill. Model-checking for real-time systems. In *Logic in Computer Science, 1990. LICS '90, Proceedings., Fifth Annual IEEE Symposium on e*, pages 414–425, Jun 1990.

[27] R. Alur, C. Courcoubetis, and D. Dill. Model-checking in dense real-time. *Inf. Comput.*, 104(1):2–34, May 1993.

[28] Synopsys. Hybrid rtl formal verification, 2006.

[29] S. Tripakis, Sergio Yovine, and Ahmed Bouajjani. Checking timed büchi automata emptiness efficiently. *Formal Methods in System Design*, 26(3):267–292, 2005.

[30] G. Behrmann, A. David, and K. G. Larsen. *A Tutorial on Uppaal*, pages 200–236. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[31] G. Guennebaud and B. Jacob. Eigen v3. http://eigen.tuxfamily.org, 2010.

[32] B. C. Kuo. *Digital Control Systems*. Oxford University Press, Inc., New York, NY, USA, 2nd edition, 1992.

[33] A. Cimatti, A. Griggio, B. J. Schaafsma, and R. Sebastiani. *The MathSAT5 SMT Solver*, pages 93–107. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[34] D. Xue and Y. Q. Chen. *System Simulation Techniques with MATLAB and Simulink*. No Longer used. Wiley, 2013.