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 control 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 effectiveness of the proposed approach to detect them. To the best of my knowledge, this is the first contribution 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 [29]; it can be mathematically expressed as difference equations, transfer functions, or state-space equations. In this particular work, the focus is on state-space models, which represent the behavior of a system through a state evolution equation $\dot{x}(n+1)$ and an instantaneous output equation $y(n)$, as follows:

$$\begin{aligned} \dot{x}(n+1) &= Ax(n) + Bu(n) \\ y(n) &= Cx(n) + Du(n), \end{aligned} \tag{1}$$

where A, B, C, and D are matrices that fully specify a digital system. Such models can be translated into algorithms and implemented in several kinds of microprocessors (*e.g.*, field programmable gate arrays (FPGA) devices [27] and digital signal processors [24]). 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]. In fact, such systems are vulnerable to finite word-length (FWL) effects [15, 19], which can cause several quantization problems, such as truncation or round-off errors. Particularly, in such circumstances, the precision of each element from matrices A, B, C, and D will be affected by FWL effects, which can compromise the system's properties (*e.g.*, stability). 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.

In order to tackle such problem, this paper proposes a verification methodology based on bounded model checking (BMC) techniques [11], 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 this paper extends a previous work [7, 18, 2, 13, 6]. In particular, the major improvement of the DSVerifier version described here relies on the support for state-space models, which allows a better insight about the internal system behavior, enables the verification of new properties (*e.g.*, controllability and observability), and considers initial conditions for system analysis [14]. In addition, DSVerifier now supports two efficient model-checking tools as back-end: ESBMC [12, 28] (previously supported) and CBMC [22, 10].

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 [25, 17, 8, 20, 26, 31]. There are also simulation tools (*e.g.*, LabVIEW [21] and MATLAB [30]), which are traditionally used by control engineers. However, such approaches depend on input stimulation to evaluate the

state-space of a system, which might not exploit all possible conditions that a system can exhibit. In contrast, Alur *et al.* [3, 4] 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 [32], Open-Kronos [33], and UPPAAL [5]). Nonetheless, differently from the work presented here, such approaches do not tackle system robustness related to implementation aspects [7, 18, 2].

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: manual (user) and automated (DSVerifier) procedures. In the former, the software 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 $< I, F >$, where I is the number of bits for the integer part, and F 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.

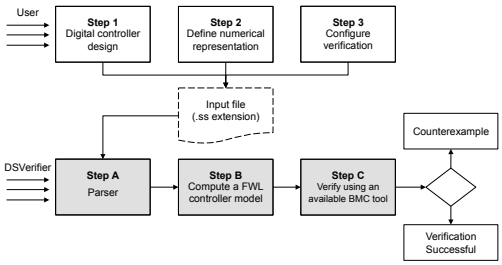

Figure 1: Verification methodology.

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. Then, in Step B, it formulates a FWL model using a function $FWL[\cdot] : \mathbb{R} \rightarrow Q[\mathbb{R}]$, which applies the FWL effects to a state-space digital system, where $Q[\mathbb{R}]$ represents the quantized set of representable real numbers in the chosen implementation format. Finally in the step C, the translation of the resulted ANSI-C code (*i.e.*, the respective quantized state-space digital system) into SAT or SMT formulae is completed, by a highly efficient bounded model-checking tool (*e.g.*, ESBMC or CBMC) [12, 22]. 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 [16]; **Controllability -** it checks whether a digital system M is controllable, based on the rank of its controllability matrix; and **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] extracted from literature [1, 23]. In particular, this study employs CBMC $v5.4$, with the SAT solver MiniSAT $v2.2.0$ [9]. All systems are checked against four properties, as described in Section 3, using a 32-bits micro-controller hardware configuration with three precisions (8, 16, and 32-bits), which results in 300 verifications.

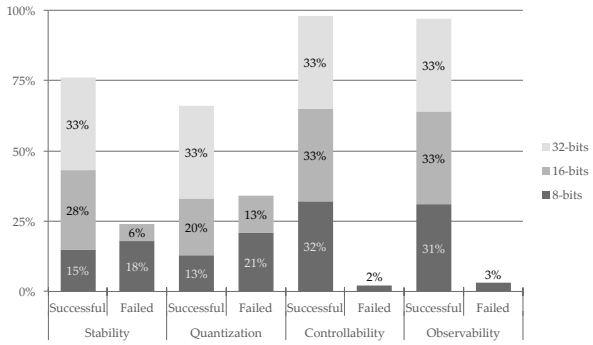

Figure 2: Experimental results.

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, *(ii)* the lower the precision, the higher its sensibility to FWL effects, and *(iii)* controllability and observability are less sensitive to FWL effects, once they only rely on the system's coefficients. 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 [29], **(iii)** stability (for state-space systems), controllability and observability verifications for SISO and multi-input and multi-output (MIMO) systems [29], 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.

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

5. REFERENCES

[1] 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.

[2] 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.

[3] 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.

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

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

[6] 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.

[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] 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.

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

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

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

[12] 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.

[13] 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.

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

[15] 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.

[16] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

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

[18] 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.

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

[20] 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.

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

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

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

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

[25] 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.

[26] 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.

[27] 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.

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

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

[30] K. Sigmon. *MATLAB Primer*. CRC Press, 5th edition, 1998.

[31] W. Sung and K.-I. 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.

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

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

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