Using Schedulability Analysis Techniques to Derive the Average-Case Behavior with a Monte-Carlo Simulation

Steffen Kollmann, Victor Pollex and Frank Slomka
Institute of Embedded Systems/Real-Time Systems
Ulm University
{firstname.lastname}@uni-ulm.de

I. Motivation

Evaluating the time-behavior of distributed embedded real-time systems can be performed in many ways. Mainly three different approaches can be used as shown in figure 1. The first one is to test the implemented system, the second one is to perform a simulation and the third one is to conduct a real-time analysis. Each of them has its advantages and disadvantages.

Testing the implemented system is certainly the most realistic approach concerning the results, because the final system is used. But correcting any failures discovered at this late design stage can lead to enormous costs. To prevent this situation other approaches have to be used that are suitable for early design stages where mistakes are less expensive to fix.

Such methods are the simulation or the analysis of a system. These can also be used in late design phases. For example a tool like chronSIM [1], which allows simulating a system, can provide information on what a system is actually doing especially in the average case. With a simulation however the border cases like the best-case and worst-case time behavior cannot be determined. Due to the coverage problem that simulations have, it is unknown whether such a border case has been simulated or not. Analytical approaches like those based on Tindell and Clark [2] construct the border cases and calculate guaranteed bounds for the time behavior. In [3] an automotive case study was conducted where both approaches - simulation and analysis - are compared. The conclusion of the paper is that both approaches are needed for the design process

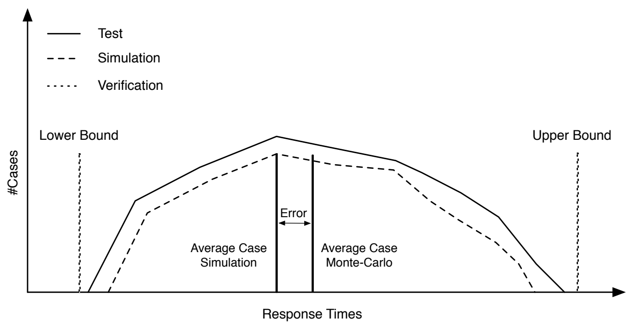

Figure 1. Different evaluation techniques

of embedded hard real-time systems. The focus in the paper is to determine the worst-case time behavior. But to consider only the border cases during a system design is often not sufficient to construct a reliable system.

An example for this can be found in the domain of control systems. To design a reliable controller on the one hand the border cases are required to verify the stability of the control loop. On the other hand the most important part in the design is the knowledge of the average-case time behavior, because a controller should perform best in the average case. Therefore both values are needed to design a reliable controller. In [4] the impact of the time behavior on a control system is discussed more in detail.

As mentioned with a simulation the average case can be obtained, because it will most likely describe the actual system behavior. However the more complex a system is, the longer will be the runtime of the simulation. A good impression about the costs required to perform a simulation can be obtained by considering the results in [3]. The runtime can easily be in the magnitude of days to get significant data regarding the average case due to it depending on the history of the simulation. Therefore such an approach cannot be parallelized easily. It is possible to start various simulation runs simultaneously, but it is not clear how long each run has to be executed in order to obtain the desired data. So the question arises whether other methods can be used to determine the average case while needing less time than a simulation.

Analytical approaches currently only construct the border cases and calculate guaranteed bounds for them. Naturally the average case must be between the calculated best case and worst case. The question is if an analytical approach like the SymTA/S approach [5] or the real-time calculus [6] can be modified or extended in such a way that the average case or at least an approximation of the average case can be obtained more quickly than with a simulation. In [7] an extension of the real-time calculus is proposed where probabilistic arrival and service curves are considered. But it is an open question if the approach is able to construct the average-case behavior, because with the presented model it seems improbable that the average case of the system can be found.

We now sketch a proposal on how an analytical approach can be modified to possibly obtain the average case. The idea is based on a Monte-Carlo simulation [8] and to the best of our knowledge there is no successful work which describes how schedulability analysis techniques can be used with a Monte-Carlo simulation to compute the average-case behavior of a system.

In [9] a technique is presented which is based on a similar idea. A Monte-Carlo simulation is used to determine the time-behavior of a system, but it is not clear which method is used for the Monte-Carlo simulation. It is stated that the network-calculus is not used, because the system behavior cannot be modelled adequately.

II. Proposal

We now present our proposal and start our discussion with the simulation approach, because it is reasonable to understand why it is able to find the average case. The advantage of such a method is that the different correlations/dependencies between the tasks are covered inherently. But dependencies within a system are an issue that analytical approaches have. Ignoring them in the analysis results in more pessimistic bounds, therefore as many dependencies as possible must be considered to obtain an average case. Many approaches have been developed to consider data dependencies [6] or task dependencies [10]. It is however unclear whether the existing work regarding the dependencies is sufficient to be able to calculate the average-case behavior.

The question is now how the average case can be determined efficiently. We propose to use a Monte-Carlo simulation based on a schedulability analysis by varying parameters like stimulation, execution time, etc. The stimulation may only be varied within its bound as well as the execution time may only be varied between its best-case and worst-case execution time. To be able to freely vary the stimulation the periodic model with jitter is insufficient, instead more expressive models have to be used like the event streams [11] or the arrival curves used by the real-time calculus [6]. Some questions which have to be answered to obtain the average-case behavior are:

- Is a Monte-Carlo simulation based on a schedulability analysis even suitable to obtain the average-case behavior?
- How must the parameters be randomized?
- Which dependencies must be modeled in order to obtain acceptable results?
- How many samples have to be gathered in order to be able to approximate the average case?
- Is it possible to bound the error of the approximation?

The first question addresses the fact that analytical approaches always construct the corner cases. Is it therefore possible to obtain the average case with a Monte-Carlo simulation? The next question considers the issue of how the parameters for the Monte-Carlo simulation have to be randomized. Can we use the same method as used by simulation approaches or do we have to consider anything else? To approximate the behavior of a simulation, task dependencies have to be considered during an analysis. So which are the important dependencies that influence the time behavior? The last two questions relate to the results of the Monte-Carlo simulation, where the quality of the results has to be determined and a threshold has to be set for which the quality is deemed acceptable.

References

[1] chronSIM, http://www.inchron.com.

[2] K. Tindell and J. Clark, “Holistic schedulability analysis for distributed hard real-time systems,” *Microprocessing and Microprogramming*, vol. 40, pp. 117–134, April 1994.

[3] S. Kollmann, V. Pollex, K. Kempf, F. Slomka, M. Traub, T. Bone, and J. Becker, “Comparative application of real-time verification methods to an automotive architecture,” in *Proceedings of the 18th International Conference on Real-Time and Network Systems*, 2010.

[4] T. Bund, S. Moser, S. Kollmann, and F. Slomka, “Guaranteed bounds for the control performance evaluation in distributed system architectures,” in *Proceedings of the International Conference on Real-Time and Embedded Systems (RTES 2010)*, Singapore, Sep. 2010.

[5] K. Richter, “Compositional scheduling analysis using standard event models - the symta/s approach,” Ph.D. dissertation, University of Braunschweig, 2005.

[6] E. Wandeler, “Modular performance analysis and interface-based design for embedded real-time systems,” Ph.D. dissertation, ETH Zurich, September 2006.

[7] L. Santinelli and L. Cucu-Grosjean, “Toward probabilistic real-time calculus,” *SIGBED Rev.*, vol. 8, pp. 54–61, March 2011.

[8] D. Kroese, T. Taimre, and Z. Botev, *Handbook of Monte Carlo Methods*. Wiley, 2011, vol. 706.

[9] C. Mauclair, “Analysis of Real-Time Networks with Monte Carlo Methods,” http://sites.onera.fr/journeesdestheses-tis/sites/sites.onera.fr.journeesdestheses-tis/files/articles/JDT-TIS-2011_Mauclair_Cedric.pdf, 2011.

[10] J. C. Palencia and M. G. Harbour, “Schedulability analysis for tasks with static and dynamic offsets,” in *RTSS*, 1998, p. 26 ff.

[11] K. Gresser, “An event model for deadline verification of hard real-time systems,” in *Proceedings of the 5th Euromicro Workshop on Real-Time Systems*, 1993, pp. 118–123.