Exploiting Execution Dynamics in Timing Analysis Using Job Sequences

Leonie Ahrendts, Sophie Quinton, Rolf Ernst

Exploiting Execution Dynamics in Timing Analysis Using Job Sequences

Leonie Ahrendts, Sophie Quinton, and Rolf Ernst

Abstract—Worst case design as needed for critical systems usually resorts to established methods for worst case response time analysis which rely on the worst case execution time of tasks and the minimum temporal distance between task activations. The result is often very pessimistic when compared to the real worst case load. Many feasible designs are therefore rejected under such analyses. Using worst case models based on job sequences rather than single jobs leads to less pessimistic results and makes worst case design more practical. This paper outlines existing modeling and analysis techniques which are based on job sequences and refers to several examples from automotive design where great benefits were demonstrated.

Index Terms—Embedded and cyber-physical systems, Real-time systems, Weakly hard real-time systems, Automotive systems, Safety, Timing analysis, Constraint specification

1 Introduction

Deriving tight bounds on the timing behavior of a real-time computing system is known to be a challenging verification problem. Verification itself is difficult, but another major problem is that of identifying a precise yet analyzable system model for which safe (i.e., possibly approximate but always correct) parameter values can be obtained in practice through measurements or formal approaches.

A real-time computing system consists of a set of software tasks which compete for processing and communication resources and are served according to a scheduling algorithm. A task is executed repeatedly and each of its instances is called a job. A task can therefore be seen as an infinite sequence of jobs over time. The creation of a job is triggered by an *activation event*, and the amount of service requested by a job is called its *workload*. Jobs may access shared resources such as memory during execution.

For performance verification, a task is modeled using bounds on its timing parameters. To derive such bounds, it is common to (1) characterize the best case/worst case parameters that can be observed for a single job of this task, and then (2) attribute these extreme parameters to every job of the considered task. Characteristic parameters of a job include its execution time, access times to shared resources, communication delays, as well as the temporal distance to the activation event of the subsequent job (simply called job distance in the sequel). This procedure leads to a safe but pessimistic timing model of a task.

A similar approach is commonly chosen for specifying the constraints imposed on a real-time computing system: The hardest timing constraint that applies to one job of a given task is adopted for all jobs of that task. One such example is a task deadline which must be met for every job.

In this paper, we advocate the use of *job sequences* to describe the best case/worst case timing parameters and constraints of a task: These should be formulated for sequences of n consecutive jobs (also called n-sequences in the following). By this means, execution dynamics and therefore variability in task behavior can be taken into account. For instance, in an n-sequence of jobs of the same task, the temporal distance between the first and the last activation event is guaranteed to be larger than n times the minimum job distance. Similarly, n consecutive jobs of a task have a maximum cumulative workload that is smaller than n times the worst case execution time. On the constraint side, it may be tolerable for some jobs in a given n-sequence to miss their deadline.

In various works on real-time computing systems, specific problems have been successfully solved by considering sequences of jobs for modeling and/or constraint specification. We believe that a rigorous and consistent use of job sequences for task modeling and constraint specification could represent an important step towards tighter bounds on system timing behavior. In addition, the effort required to model timing parameters and to derive constraints for potentially any $n \in \mathbb{N}$ can be reduced with appropriate mathematical methods.

In the rest of this paper, we first survey and discuss existing work based on job sequences for either modeling or constraint specification. We then show as an example how Typical Worst-Case Analysis (TWCA) achieves substantial improvements in accuracy by systematically using job sequences for both modeling and constraint specification. We illustrate its practical significance by industrial case studies.

2 Using Job Sequences for Task Modeling and Constraint Specification

In this section, we discuss seminal research contributions which exploit properties of job sequences for modeling

- *L. Ahrendts and R. Ernst are with the Institute of Computer and Network Engineering, TU Braunschweig, Hans-Sommer-Strasse 66, 38106 Braunschweig, Germany. Email: {ahrendts, ernst}@ida.ing.tu-bs.de*
- *S. Quinton is with Inria Grenoble – Rhône-Alpes, 655 Avenue de l'Europe – Montbonnot, 38334 St Ismier Cedex, France. Email: sophie.quinton@inria.fr*

or constraint specification of real-time computing systems. Note that, although these powerful abstractions for the description of job sequences exist, they are often not used to their full potential in practice.

2.1 Task Modeling Using Timing Parameters Based on Job Sequences

An activation event may be caused by a periodic timer interrupt, or by a measured variable falling below or exceeding a threshold value, an alarm indicating a specific incident like a timer overflow or a fault, etc. Many activation events thus have an aperiodic nature, and their timing depends on the dynamics of the system environment. The execution time of a task, on the other hand, may vary due to several reasons: data-dependent control flow, variable resource usage or access times as in e.g. memory accesses.

Using job sequences for modeling the arrival of activation events and workload can greatly improve the accuracy of the model when the timing of these parameters is subject to high variability. This important observation is at the core of Network Calculus [1], and has been exploited by a host of work in communication theory. Network Calculus was later adapted and proposed as a method for real-time system design under the name of Real-time Calculus [2] [3]. Both Network Calculus and Real-time Calculus use as fundamental modeling concepts *event arrival curves* and *workload curves*, which describe best case/worst case task parameters for job sequences. For instance, the upper event arrival curve $\alpha_i^+(\Delta t)$ of a task τ_i bounds from above the number of activation events that may occur in any time interval Δt. The relation to job sequences is even more obvious if one considers the pseudo inverse $\delta_i^-(n)$ of $\alpha_i^+(\Delta t)$, which we call the *distance function*: $\delta_i^-(n)$ returns the minimum temporal distance between the first and the last activation event in any sequence of n consecutive jobs of task τ_i. Similarly, the upper workload curve $\gamma_i^+(n)$ of a task τ_i bounds from above the workload requested by any n consecutive jobs.

Event arrival and workload curves provide an expressive task modeling approach which can yield more accurate analysis results. Figure 1 illustrates the striking difference between event arrival and workload curves obtained based on (1) worst case parameter values for a sequence of jobs and (2) linear extrapolations of worst case parameter values of a single job. We mean by linear extrapolation w.r.t. event arrival that the minimum inter-arrival time of any two jobs is used as period. Linear extrapolation w.r.t. requested workload is the weighting of the worst case execution time with the number of activation events. In contrast, the non-linear, job sequence-based worst case models represent tighter upper bounds $\alpha^+(\Delta t)$ and $\gamma^+(n)$ since they are based on the observation of more than one job. The shaded area between the linear models and the non-linear sequence models in Figure 1 illustrates the gain in accuracy.

Let us now shortly discuss options for deriving in practice such tight and expressive event arrival and workload curves over job sequences. Event arrival curves which are formally derived are tight if the behavior of the event source is either analytically known or enforced. Periodic event arrivals with jitter fall, for instance, in the first category.

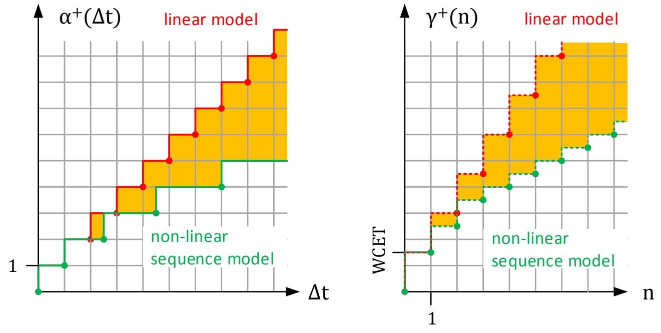


Fig. 1: **Bounding event arrival and workload.**

Shaped events streams fall in the second category. The workload of a job sequence can be formally bounded using e.g. a multiframe task model [3] [4] whenever knowledge about the task structure and functional behavior is available. Besides, it is not a problem if upper event arrival curves or upper workload curves are only known up to a specific n, as the concept of sub-additive extension can be applied [5].

The tightness of formal bounds relies on detailed information on the software and the hardware platform in use. Such information is not always available and then formally derived bounds are pessimistic. When platforms with complex performance-enhancing and power-saving features are used, this pessimism is so large that the practical usability of these formal bounds is disputable. A complementary approach is thus to derive bounds on event arrival and workload by measurements over execution traces. An execution trace of a task is a list of observed activation instants and execution times of an actual job sequence. Although it has the character of experiments and an uncertainty thus formally remains, trace recording is a widely used and accepted technique in industrial practice.

2.2 Constraint Specification Based on Job Sequences

The classical timing constraint for real-time systems is the deadline of a task, specifying the maximum allowed response time of any job of this task. Satisfying this constraint guarantees a maximum reaction time which fits the time constants of the system and its environment.

For systems with control or imaging applications, it has been demonstrated that deadline misses can actually be tolerated without any impact on their functional correctness cf. [6] [7] [8] as long as the pattern of deadline misses is precisely known. Such robust systems are called *weakly-hard* real-time systems. A tolerable pattern of deadline misses is usually defined as an (m, k) constraint, where at most m deadline misses in k consecutive task executions are allowed. This implies that for weakly-hard systems a response time constraint is a function of the past system behavior. (m, k) constraints thus capture variability in timing constraints over a sequence of k jobs. Weakly-hard systems are usually verified as if they had hard real-time constraints. Specifying (m, k) constraints for them rather than a single deadline clearly increases their likelihood to be successfully verified. Interestingly, from the guaranteed satisfaction of a given (m, k) constraint one can infer satisfaction of constraints for other values of k [9].

2.3 Discussion

We have seen so far that several standalone approaches exist which exploit execution dynamics in the timing analysis of real-time computing systems: On the one hand, the consideration of job sequences allows refined modeling of event arrival and workload. On the other hand, weakly-hard constraints improve accuracy in constraint specification by introducing requirements over a sequence of jobs, taking into account the inherent robustness of systems towards occasional deadline misses.

Furthermore, the presented concepts – task modeling based on job sequences and weakly-hard constraints – share the mathematical property that they describe or constrain job sequences in a *cumulative* manner. Cumulative functions do not preserve knowledge about the individual timing behavior of each job in the considered sequence, but summarize the timing characteristics of the sequence. This approach is mathematically elegant, because it contains all required information for performance verification but condenses at the same time several equivalent worst cases in one description. The introduced event arrival curves and workload curves are cumulative since they describe worst case aspects of task behavior with regard to a time interval Δt (event arrival curve) or a sequence of n jobs (workload curve). Likewise weakly-hard constraints define a budget of deadline misses for a sequence of n jobs, which generally includes several allowed patterns of jobs with missed deadlines.

In this paper, we argue for systematically applying sequence-based approaches in both modeling and constraining. As will be demonstrated in the following, this is an important step to significantly reduce pessimism of formal timing analysis results and make the verification of highly loaded, industrial real-time computing systems possible.

3 Verifying Highly Loaded Systems

Highly loaded real-time computing systems, which actually work in industrial practice, are often rejected by formal timing analysis. The discrepancy between measurements and formal analysis can be considerably reduced, if tighter upper bounds on event arrival and workload are applied as described in Section 2.1. This standalone approach is, however, often not sufficient.

System feasibility observed in practice suggests that event arrival and workload demand of tasks must be most of the time below the obtained upper bounds. In the transient overload situations, which may happen, there is experimental evidence that many systems tolerate a limited number of deadline misses. The functional robustness towards m deadlines misses in a sequence of k consecutive jobs can even be proven [7] [8]. It seems therefore reasonable to combine sequence-based modeling with weakly-hard constraints introduced in Section 2.2.

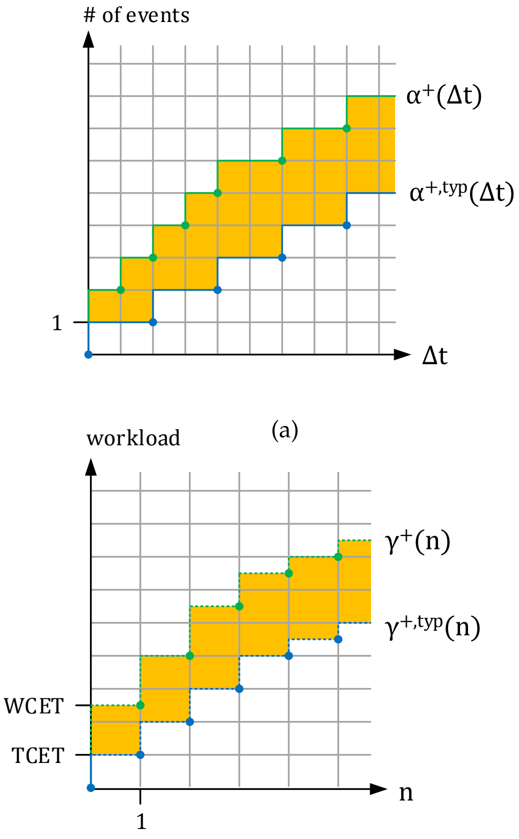
One key issue is how to formally provide (m, k) guarantees, considering schedulable and unschedulable phases of system behavior. The verification method Typical Worst Case Analysis (TWCA) [10] proposes a possible solution. First, event arrival curves and workload curves for each task are derived, which are true upper bounds for most of the run time. Such event arrival curves and workload curves are called typical, because they capture the predominant timing behavior of tasks (for example the periodic workload but not the rare sporadic workload). Those typical curves describe a less service-demanding job behavior than the worst case curves: Figure 2a shows a typical event arrival curve $\alpha^{+,typ}(\Delta t)$ and a worst case event arrival curve $\alpha^{+}(\Delta t)$ for a given task, where by definition we have $\alpha^{+,typ}(\Delta t) \leq \alpha^{+}(\Delta t)$. In the example, the typical event pattern is periodic, while in the worst case additional sporadic activation events occur. Figure 2b illustrates a typical workload curve $\gamma^{+,typ}(n)$ and a worst case workload curve $\gamma^{+}(n)$ for a given task, where again $\gamma^{+,typ}(n) \leq \gamma^{+}(n)$. In a phase of typical system behavior, a certain maximum typical execution time $TCET$ is never exceeded, while execution times larger than $TCET$ may occur in the worst case.

The difference between the worst case curve and the typical curve is monotonically increasing, both for events and workload. In contrast to the approaches presented in the previous section, however, important differences are not only obtained for longer job sequences but also for a single job: On the one hand, the typical event arrival curve does not assume the minimum inter-arrival time even for a single job. On the other hand, the typical workload curve does not attribute the worst case execution time $WCET$ to a single job but the maximum typical execution time $TCET$. On the basis of typical event arrival curves and workload curves, highly loaded real-time systems can be proven schedulable in phases of typical behavior.

To verify the worst case behavior, TWCA quantifies the maximum distance between the typical and the worst case curves: The additional activation events contained in the worst case event arrival curve but not in the typical event arrival curve can be considered as cause for transient overload in the system. It is actually possible to bound the occurrence of these overload events in Δt by the event arrival curve $\alpha^{+,over}(\Delta t)$. Similarly, jobs which exceed the typical execution time are a potential source of overload. The maximum number of jobs which exceed the typical execution time in Δt can be bounded by an event arrival curve $\alpha^{+,over}_{TCET}(\Delta t)$. From the comparison of the typical and worst case workload curves follows, moreover, that the amount of additional workload in a sequence of n consecutive jobs cannot be not larger than $\gamma^{+,over}_{TCET}(n)$. TWCA now derives the maximum number of missed deadlines in a job sequence of given length k as a function of the number of overload events and the amount of additional workload. As we will see in the following section, highly loaded real-time systems with weakly-hard constraints have been successfully verified using the TWCA method.

4 Case Studies

The significance of TWCA results for industrial practice has been demonstrated by several major use cases.



(a)

(b)

Fig. 2: **Typical and worst case curves for event arrival and workload**

4.1 Automotive Communication Networks

In [11], the timing behavior of automotive CAN buses has been investigated. Automotive CAN buses have seen a massive increase in utilization in recent years due to larger message sizes and a rapidly growing number of messages. CAN messages are time-triggered and/or event-triggered. If the minimum inter-arrival time is used for the modeling of event arrival in a system which is dominated by event triggering, the theoretical worst-case utilization of working systems exceeds 100% and may reach 500%. Tight non-linear event arrival curves allow for a much more accurate performance analysis. Yet such an improved WCRT analysis still discards many systems, which have proven functional in extensive simulation. The reason is that the occasional loss of messages can actually be tolerated, and the modeling of response times constraints in form of (m, k)-guarantees better represents the actual system requirements.

The work presented in [11] therefore applies sequence models and subsequently TWCA to the CAN case study. Firstly tight upper event arrival curves are derived based on specified and measured timing of message dispatch. Then sporadic dispatch events are identified which can be interpreted as overload events potentially causing deadline misses during transient workload peaks. Based on TWCA, for each message an (m, k)-guarantee is obtained. In the case study it could be shown that in at most 15 % of 10 000 executions, a CAN message transmission takes longer than in the overload-free case. For many of the 212 messages the percentage is significantly below 15 %. This experimental result was the first to formally show that an increase of the classical CAN bus load is actually tolerable.

4.2 Automotive Software: Engine Management

Automotive software applications integrate a large number of inter-dependent functions. The engine management ranks among the most complex software applications and is composed of about 20 container tasks including around 1500 functions which are scheduled by an OSEK-compliant operating system. These container tasks are a source of strong but well understood execution time variations. The average system utilization is usually above 90%, while the worst-case utilization in those systems exceeds easily 100%. Despite this evidence of overload, extensive simulation often suggests functional correctness of the investigated software systems. This discrepancy can be attributed to inaccurate utilization analysis which does not take into account variability of execution times.

The system-level timing feasibility test proposed in [12] for an engine control application shows that with a workload curve $\gamma^+(n)$ which describes the execution demand of job sequences, significantly tighter WCRTs can be derived. While with linear workload modeling 5 out of 20 tasks are found to be infeasible in formal performance analysis, workload modeling w.r.t. job sequences improved the accuracy of results such that only 2 out of 20 tasks are bound to complete after their deadline. Since the involved control applications are inherently robust towards occasional deadline misses, (m, k)-guarantees for the 2 unschedulable tasks are derived. TWCA is applied to this problem of computing the (m, k)-guarantees in [12], yet the overload is not caused by additional sporadic activations in this use case. In contrast, it is caused by execution times of tasks which are occasionally longer than the $TCET$. In the case study, the $100ms$ task and the $200ms$ task could each tolerate 3 deadline misses in 20 consecutive executions, and as few as 1 deadline miss in 20 executions could actually be guaranteed by TWCA.

5 Conclusion

Modern real-time computing systems with performance-enhancing features have high variability in event arrival and workload. At the same time system requirements with regard to job completion are not static but often depend on system history. For instance, there may be a precisely defined budget for deadline misses of jobs.

An approach to deal with these dynamic system characteristics is to model and constrain sequences of jobs rather than focusing on the behavior of a single job in isolation. Event arrival curves, workload curves and weakly-hard constraints are existing abstractions which allow to make worst case statements about sequences of jobs. The systematic and rigorous use of this more detailed modeling and constraint formulation allows designing systems with formal worst case guarantees where established methods for formal performance analysis are not applicable due to their pessimism. Since the approach is compatible to existing engineering methods of measuring and trace recording it provides an opportunity to improve design verification and optimization where current design practice has to live with unsafe simulation and prototyping. This paper has presented as an example the TWCA method which is based on an analysis of the impact of transient overload. The gained accuracy narrows significantly the gap between verification results of formal performance analysis and simulations that are currently used for validation in industrial practice.

Acknowledgments

This work has received funding from the German Research Foundation (DFG) under the contract number TWCA ER168/30-1. This work has also been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

References

[1] J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. Springer Science & Business Media, 2001, vol. 2050.

[2] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 4. IEEE, 2000, pp. 101–104.

[3] E. Wandeler, A. Maxiaguine, and L. Thiele, "Quantitative characterization of event streams in analysis of hard real-time applications," *Real-Time Systems*, vol. 29, no. 2-3, pp. 205–225, 2005.

[4] S. Baruah, D. Chen, S. Gorinsky, and A. Mok, "Generalized multiframe tasks," *Real-Time Systems*, vol. 17, no. 1, pp. 5–22, 1999.

[5] D. E. Wrege and J. Liebherr, "Video traffic characterization for multimedia networks with a deterministic service," in *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, vol. 2. IEEE, 1996, pp. 537–544.

[6] G. Bernat, A. Burns, and A. Liamosi, "Weakly hard real-time systems," *IEEE transactions on Computers*, vol. 50, no. 4, pp. 308–321, 2001.

[7] G. Frehse, A. Hamann, S. Quinton, and M. Woehrle, "Formal analysis of timing effects on closed-loop properties of control software," in *Real-Time Systems Symposium (RTSS), 2014 IEEE*. IEEE, 2014, pp. 53–62.

[8] R. Blind and F. Allgöwer, "Towards networked control systems with guaranteed stability: Using weakly hard real-time constraints to model the loss process," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 7510–7515.

[9] S. Quinton and R. Ernst, "Generalized weakly-hard constraints," in *International Symposium On Leveraging Applications of Formal Methods, Verification and Validation*. Springer, 2012, pp. 96–110.

[10] W. Xu, Z. A. Hammadeh, A. Kröller, R. Ernst, and S. Quinton, "Improved deadline miss models for real-time systems using typical worst-case analysis," in *Real-Time Systems (ECRTS), 2015 27th Euromicro Conference on*. IEEE, 2015, pp. 247–256.

[11] S. Quinton, T. T. Bone, J. Hennig, M. Neukirchner, M. Negrean, and R. Ernst, "Typical worst case response-time analysis and its use in automotive network design," in *Proceedings of the 51st Annual Design Automation Conference*. ACM, 2014, pp. 1–6.

[12] S. Tobuschat, R. Ernst, A. Hamann, and D. Ziegenbein, "System-level timing feasibility test for cyber-physical automotive systems," in *Industrial Embedded Systems (SIES), 2016 11th IEEE Symposium on*. IEEE, 2016, pp. 1–10.

Leonie Ahrendts is a Ph.D. student in the Embedded System Design Automation group of the Institute of Computer and Network Engineering at TU Braunschweig. She received her Master degree in electrical engineering from TU Braunschweig in 2015, and was fellow of the German Academic Scholarship Foundation. Her current research concerns the analysis of real-time systems with an emphasis on weakly-hard real-time systems. Other areas of interest are fault-tolerant real-time computing systems and communication networks.

Sophie Quinton is a researcher at Inria Grenoble Rhône-Alpes in France. She received her Ph.D. degree from the University of Grenoble, in 2011. She was a graduate research assistant at the VERIMAG laboratory and a postdoc at the Institute of Computer and Network Engineering at TU Braunschweig. Her research focus is mostly on real-time schedulability analysis and contract-based design and verification of embedded systems.

Rolf Ernst is a professor at the Technische Universität Braunschweig where he chairs the Institute of Computer and Network Engineering covering embedded systems research from real-time systems theory to challenging automotive and aerospace applications. He has a Ph.D. in EEIT from the University of Erlangen. He is an IEEE Fellow.