

Estimation of Worst Case Response Time Boundaries in Multi-Core Real-Time Systems

Matthias Mucha*, Jürgen Mottok* and Stefan Krämer*

* Friedrich-Alexander Universität Erlangen-Nürnberg

Fakultät fuer Informatik 3, Erlangen, Germany

† Research Professor for Safe and Secure Systems of Systems

Laboratory for Safe and Secure Systems (LaS³), Zentrum Digitalisierung Bayern (ZD.B)

Ostbayerische Technische Hochschule (OTH) Regensburg, 93025 Regensburg, Germany

‡ Laboratory for Safe and Secure Systems (LaS³), Zentrum Digitalisierung Bayern (ZD.B)

Ostbayerische Technische Hochschule (OTH) Regensburg, 93025 Regensburg, Germany

Abstract—We address a novel probabilistic approach to estimate the Worst Case Response Time boundaries of tasks. Multi-core real-time systems process tasks in parallel on two or more cores. Tasks in our contribution may preempt other tasks, block tasks with semaphores to access global shared resources, or migrate to another core. The depicted task behavior is random. The shape of collected response times of a task within a processing time is multimodal. Extreme Value approaches need unimodal response time distributions to estimate the Worst Case Response Time of tasks. The new proposed method derives a set of three task set shapes from the source task set. It is used to minimize the uncertainty of random task behavior by maximizing the coverage of possible Worst Case Response Times. The case study evaluates the new proposed estimation method by the use of dynamically generated random tasks with varying task properties.

I. INTRODUCTION

Multi-core real-time systems are state of the art embedded systems. The architecture of such systems allows parallel data processing, which means to distribute tasks on several cores. Load balancing, for example, is applied to reduce the calculation overhead and heat dissipation that appears in high-utilized powertrain single-core real-time systems. Such hard real-time systems have to meet deadlines to avoid system failures. The *Worst Case Response Time* (WCRT) as a task metric describes whether the task violates its deadline. *Static Timing Analysis* (STA) is widely used in literature and by practitioners to determine the WCRT analytically for single-core real-time systems. It assumes the most pessimistic WCRT and approximates asymptotically against the real WCRT. In comparison, probabilistic approaches assume the most optimistic WCRT at the beginning of the analysis. By nature, tasks in multi-core real-time systems are able to access global resources by the use of semaphores. Tasks that want to access the resources at the same time are blocked. Furthermore, multi-core real-time systems allow *Inter-Process Activation* (IPA) where a task can activate several other tasks at runtime. Automotive powertrain systems use tasks with periodic, engine-triggered and sporadic activation patterns. Precondition to apply an extreme value distribution is a unimodal distributed gapless sample shape. In our work the

shapes of the samples are multimodal where gaps in between the modes can occur. Mucha et al. first have distinguish the shape behaviour in [1]. Therefore, a new approach is required to estimate the WCRT's with probabilistic methods.

II. CONTRIBUTION AND OUTLINE

We present a novel probabilistic approach to estimate WCRT boundaries of tasks. Tasks can have different time bases, preempt other tasks, block tasks with semaphores to access global shared resources, or migrate to another core. We assume in our work, that tasks are scheduled in an automotive multi-core powertrain system. Aim one of the approach is to estimate the maximum, average and minimum WCRT boundary of the task with *Derived Task Set Shapes* (DTSS). DTSS are task set models derived from the *source task set* τ_s . The uncertainty, which is part of probabilistic approaches has to be reduced to an acceptable minimum. Therefore, aim two is to reduce the uncertainty of the estimated WCRT boundaries. We apply the *Margin of Error* (ME) to select the sample size n for the WCRT estimation. It expresses the maximal allowed difference between the estimated WCRT and the real WCRT. We use a cluster framework to execute N Monte Carlo simulations for each DTSS. The contribution is structured as follows: Section III gives an overview about related work. Section IV introduces the application model, the applied simulation approach and the notation that is used throughout the paper. Section IV introduces the proposed estimation approach, which is evaluated in section VI. Section VII summarizes the contribution and describes future work.

III. RELATED WORK

A lot of works regarding real-time analysis have been proposed the past years. Joseph and Pandya [2] proposed the first exact *Response Time Analysis* (RTA) for real-time systems with fixed-priority scheduling in 1986. In 2001, Edgar and Burns [3] were the first to introduce a probabilistic timing analysis approach. They estimated the *Worst Case Execution Time* (WCET) of tasks by using the *Gumbel max* extreme value

Legend: ■ execution ■ preemption ■ blocking

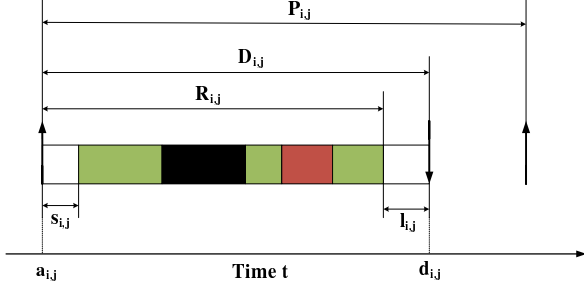


Fig. 1. Timing metrics of a job $J_{i,j}$. Job activation is denoted as $a_{i,j}$, start delay as $s_{i,j}$, lateness as $l_{i,j}$, and response time as $R_{i,j}$. $D_{i,j}$ denotes the relative deadline and $d_{i,j}$ its absolute deadline. $P_{i,j}$ is denoted as period.

distribution. Bril et al. [4] discussed WCRT analysis methods for *Fixed-Priority Scheduling with Deferred Preemption (FPDS)*. They proposed pessimistic WCRT analysis variants that are uniform for all tasks. In 2010, Griffin and Burns [5] presented an approach to estimate the WCET with *independent and identically distributed (iid)* samples. In 2013, Liu et al. [6] applied the *Generalized Pareto Distribution (GPD)* on iid samples to estimate the WCRT for tasks in complex real-time systems.

IV. SIMULATION APPROACH

Our paper assumes a *Symmetric Multiprocessor System (SMP)* with m identical processors on the hardware architecture level. All m processors share one memory. A task set τ is defined on the software level as a set of tasks T_i . Every task has timing properties such as deadline D_i , recurrence R_i , or start delay s_i . A task can be activated by one of the following time bases: periodic (time-triggered), engine-triggered, or sporadic. Each task T_i owns an amount of instructions. It corresponds to the amount of cycles which the processor needs as calculation effort for the task. The amount of instructions can be constant or variable within a minimum and maximum boundary. Job $J_{i,j}$ is defined in this contribution as the j^{th} task instance of task T_i within the task set τ . Jobs with higher priority can preempt jobs with lower priority. A job can read or write resources in the shared address space with exclusive semaphore access. This means if job $J_{i,j}$ granted exclusive semaphore access to write a resource, all other job accesses on this resource are blocked until job $J_{i,j}$ releases its semaphore. Figure 1 depicts the timing metrics of a job with execution, preemption, and blocking parts. Table I summarizes the metric and sample notation that is used throughout our contribution.

Monte Carlo task set simulation is done in this paper with a discrete event simulator. The simulator includes statistical sampling mechanisms to model tasks with variable job execution times. The simulation model consists of a hardware model, an operating system model, and a software model. Processing unit model and memory topology model are part of the hardware model. Processing units can be modeled with different

TABLE I
NOTATION

Symbol	Denotation
τ	task set
T_i	task i of task set τ
C_i	computational time of task T_i
D_i	relative deadline of task T_i
P_i	period of task T_i
$J_{i,j}$	job j of task T_i
$D_{i,j}$	relative deadline of job $J_{i,j}$
$R_{i,j}$	response time of job $J_{i,j}$
$P_{i,j}$	period of job $J_{i,j}$
α	significance level
$VAI_{i,j}$	variable amount of instructions of job $J_{i,j}$
S^Γ	set of Derived Task Set Shapes (DTSS) $\tau_u^\Gamma, \tau_{\min}^\Gamma$ and τ_{\max}^Γ
$WCRT_{\max,i}$	maximum WCRT bound of task T_i in τ_s
$WCRT_{\min,i}$	minimum WCRT bound of task T_i in τ_s
U_τ	amount of uncertainty of taskset τ
I	amount of instructions (processor cycles)

clock rates.

The operating system model describes the scheduling of the task set. The global, or core-local scheduler manages the assignment of jobs to processing units in the hardware model. The software model describes the task set and its workload. A call graph is defined for each task. It comprises the amount of instructions and blocking / non-blocking access to hardware elements, e.g., access via semaphores. Both task activation and the amount of instructions are described by probability distributions. According to the probability function, sampling mechanisms generate varying amount of instructions during the simulation.

V. WCRT ESTIMATION APPROACH

Uncertainty is the ignorance of a poorly known task behavior within its execution in the simulation, or a poorly known model parameter in the software model description. The response time of job $J_{i,j}$ depends on the metrics described in Sect. IV and its task configuration. Epistemic uncertainty is the lack of model parameter knowledge. A task set model is the abstraction of the real task set. The simplified model may have incorrect parameter values. A task set model with detailed information about task configuration minimizes the epistemic uncertainty. Model parameters such as the minimum, average and maximum amount of processor cycles that a task needs to execute are measured values from hardware traces. The quality of the measured values depends on the execution time of the hardware trace. A trace of 10s has a higher epistemic uncertainty compared to a trace of 100s.

An aleatory uncertainty arises during the simulation if a job $J_{i,j}$ of a task T_i is executed. For example, the injection function in a diesel powertrain system has an engine-triggered time base and depends on the angle position of the crankshaft. The time when the crankshaft is at the angel position is due to the nature

TABLE II
ALEATORY AND EPISTEMIC UNCERTAINTY IN A TASK SET
SIMULATION THAT INFLUENCES THE RESPONSE TIME OF JOBS.

Type of Uncertainty	Uncertainty Factors
Aleatory Uncertainty	<ul style="list-style-type: none"> • Varying job activations (Engine-triggered or sporadic time base) • Inter-Process Activation • Semaphore access (blocking) • Preemption • Start delay
Epistemic Uncertainty	<ul style="list-style-type: none"> • Probability distribution that describes the amount of processor cycles • Min, avg, and max amount of processor cycles measured from a hardware trace • Task offset

of the system being not exactly identifiable. This irregular behavior pattern activates jobs at different times. Each of the jobs may have a different amount of preemptions and blocking times so that the response time of each job varies. Further aleatory uncertainties are semaphore accesses to read or write resources, job preemptions and Inter-Process Activations. Inter-Process Activation means, that a job $J_{i,j}$ can activate other jobs during its execution. For example, job $J_{i,j}$ interrupts, activates job $J_{a,b}$, and finishes its execution if job $J_{a,b}$ is finished. Table II identifies uncertainty factors that influence the response time of jobs in a task set simulation. Both types of uncertainty may influence the sample distribution of collected response times.

Definition 1: The uncertainty U_τ in a task set simulation is denoted as the sum of the epistemic uncertainty U_e and the aleatory uncertainty U_a in which the aleatory uncertainty for the amount of processor cycles depends on the probability distribution for the amount of processor cycles in the model description (epistemic uncertainty). U_τ can be expressed as

$$U_\tau = \sum U_{e,k} + \sum U_{a,l} \quad (1)$$

$\sum U_{e,k}$ summarizes all epistemic uncertainty parts and $\sum U_{a,l}$ summarizes all aleatory uncertainty parts.

The aleatory uncertainty becomes large if the boundaries for the amount of processor cycles are undefined or set without the availability of hardware trace information. The WCRT can be estimated within a confidence interval if U_τ becomes minimal.

A. Minimization of the Epistemic Uncertainty

The goodness of the statistically estimated WCRT strongly depends on the sample quality. Therefore knowledge about the real-time system and the task set model are needed to minimize the epistemic uncertainty. System parameters such as the number of cores, clock rates in MHz, the memory model and scheduler are provided by a vendor of a real-time system in a data sheet. Each system parameter influences the

WCRT of a task. They have to be considered in the modeling process with the same confidence as the task set modeling. A pessimistic WCRT estimation arises if the system model is considered an abstract model or as a black box with unknown parameters. Expert systems may help to gather system information of different vendors to provide system models that are needed to estimate tight WCRT boundaries. The amount of processor cycles is described in abstract task set models and theoretical model descriptions as constant computational time C_i for each task T_i . As described in [7], C_i is a dynamic parameter that may change within a task set simulation. In general, the amount of processor cycles varies for each job $J_{i,j}$ of task T_i within a minimum and maximum boundary. It is defined in our contribution as *Variable Amount of Instructions* ($VAI_{i,j}$) and can be expressed with I representing the amount of instructions as thus:

$$VAI_{i,j} : VAI_{i,j} \in [I_{\min}, \dots, I_{\max}], I \in \mathbb{Z}_{\geq 0} \quad (2)$$

The $VAI_{i,j}$ is the netto job execution time of job $J_{i,j}$ without preemptions and blocking parts. With (2), the epistemic error for the amount of processor cycles is bounded between a minimum and maximum amount of processor cycles. The upper section of Fig. 3 depicts the modeling process with epistemic uncertainty. Next subsection introduces the generalized task set models that are applied to minimize the aleatory uncertainty.

B. Derived Task Set Shapes

The proposed Derived Task Set Shapes (DTSS) have different computation time probability functions. They are derived from a source task set model τ_s . All other model parameters are kept equally, in order to guarantee comparable task set models. In our contribution jobs of tasks of τ_s have Weibull-distributed computation times. The Weibull distribution is positive infinity. It is bounded so that the sampling mechanisms can select random $VAI_{i,j}$ in which the selection process depends on the probability of the Weibull distribution. The bounded Weibull distribution is implemented in the simulation approach with the minimum, average, and maximum amount of instructions as parameters. The $VAI_{i,j}$ is chosen with the highest probability at the average point of a Weibull distribution and with the lowest probability at its maximum. Therefore, the sampling mechanisms that select the $VAI_{i,j}$ may miss amounts of instructions I with a low probability during the task set simulation.

DTSS 1 (τ_u^Γ) consists of tasks with uniform distributed job computation times. The minimum and maximum boundary of the uniform distribution equals the minimum and maximum boundary of the Weibull distribution. Every $VAI_{i,j}$ is chosen with the same probability so that all possible amounts of instructions I can be chosen between the minimum and maximum boundary of the uniform distribution. The probability function can be expressed as:

$$\text{unif}(I_{\min}, I_{\max}) = \begin{cases} \frac{1}{I_{\max} - I_{\min}} & \text{VAI}_{i,j} \in [I_{\min}, I_{\max}] \\ 0 & \text{otherwise} \end{cases}. \quad (3)$$

τ_{\min}^{Γ} consists of jobs with minimal constant computation time. τ_{\max}^{Γ} consists of jobs with maximal constant computation time. All jobs of a task have equal computation times. Sampling is done on a constant amount of instructions.

Next subsection describes the Monte Carlo task set simulation method to minimize the aleatory uncertainty.

C. Minimization of the Aleatory Uncertainty

Both, τ_{\min}^{Γ} and τ_{\max}^{Γ} , are applied to cover the WCRT boundaries of tasks of τ_s . τ_u^{Γ} additionally covers Worst Case Response Times for unpredictable job computation delays within the WCRT boundaries. For example in situations involving job preemptions or job blocking times. Such unpredictable job computation delays may produce lower or higher worst case response times than with τ_{\min}^{Γ} and τ_{\max}^{Γ} . The Monte Carlo simulation is conducted with the described set of DTSS's. The proposed set minimizes the variance of the estimated minimum and maximum WCRT boundary for each task. The minimization of the variance minimizes the aleatory uncertainty of the WCRT boundaries.

Assertion 1: The set of DTSS $S^{\Gamma} = \tau_u^{\Gamma}, \tau_{\min}^{\Gamma}, \tau_{\max}^{\Gamma}$ minimizes the variance of the estimated minimum and maximum WCRT boundary.

The coverage of potential WCRTs per task is maximized with N simulations per task set. Figure 2 depicts the approach with DTSS in detail. Each Monte Carlo simulation uses a unique seed as start value and may produce a different WCRT per task. The number N of simulations per taskset is determined with the Margin of Error as shown in (4). The Margin of Error expresses the maximum possible difference of the estimated WCRT to the real WCRT for an assumed *confidence level* α . We assume a confidence level of $\alpha = 0.05$. It corresponds a confidence interval of 95%. Parameter z describes the z-score and is set to 1.96 for $\alpha = 0.05$.

$$N = 0.25z^2/\text{ME}^2 \quad (4)$$

D. WCRT Estimation with DTSS

Step four describes the estimation approach to estimate the minimum and maximum WCRT boundary for each task T_i of a task set τ as depicted in Fig. 3.

- 1) Task set τ consists of tasks with Weibull distributed job computation times to model real-time task behavior. It is the source task set model τ_s . The set of DTSS S^{Γ} is derived from τ_s .
- 2) The number of simulations N is determined with the Margin of Error and a confidence level $\alpha = 0.05$.
- 3) Each DTSS z in S^{Γ} is simulated N times. The WCRT of task T_i in S_z^{Γ} for one simulation X

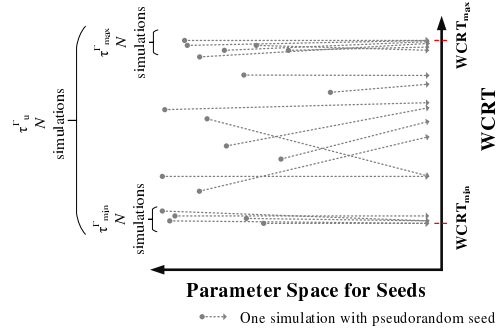


Fig. 2. WCRT Coverage with DTSS to estimate the minimum and maximum WCRT boundaries with minimized aleatory uncertainty. The x-axis represents the parameter space for the seeds. They are used as starting points for the Monte Carlo Simulation. The y-axis represents the estimated WCRT's. Each simulation has a unique starting point (gray circle) and produces a WCRT (grey arrow). Our approach has a high parameter space coverage around WCRT_{\min} and WCRT_{\max} .

is the maximum response time of collected job response times $R_{i,j}$ in S_z^{Γ} . It can be expressed with $S_{z,X,i,j}^R$ as sample of collected job response times for on simulation and n as its sample size as

$$\text{WCRT}_{z,X,i} = \max_n S_{z,X,i,j}^R. \quad (5)$$

The sample sizes n of the samples $S_{z,X,i,j}^R$ are equalized so that the estimated task WCRT boundaries are statistically comparable among each other.

- 4) The maximum WCRT of task T_i for S_z^{Γ} is defined as

$$\text{WCRT}_{i,z} = \max_N \text{WCRT}_{z,X,i}. \quad (6)$$

With the maximal WCRTs $\text{WCRT}_{i,z}$ of each DTSS we now can derive the minimal and maximal WCRT boundary of T_i for τ_s .

- 5) The minimum WCRT boundary $\text{WCRT}_{\min,i}$ of task T_i is defined in (7) as the minimum WCRT of τ_u^{Γ} and τ_{\min}^{Γ} .

$$\text{WCRT}_{\min,i} = \min S_{\min}, \quad (7)$$

$$S_{\min} = \{\text{WCRT}_{i,1}, \text{WCRT}_{i,2}\}.$$

The maximum WCRT boundary $\text{WCRT}_{\max,i}$ for each task is the maximum WCRT of $\Gamma_{1,i}$ and $\Gamma_{3,i}$.

$$\text{WCRT}_{\max,i} = \max S_{\max}, \quad (8)$$

$$S_{\max} = \{\text{WCRT}_{i,1}, \text{WCRT}_{i,3}\}.$$

VI. SIMULATION EXPERIMENTS

The case study evaluates the described WCRT estimation method. The hardware model of the simulation model consists of a dual core processor with 270 MHz, with each core possessing an OSEK scheduler. OSEK is an open standard for distributed control units in the Automotive industry. The OSEK scheduler uses the

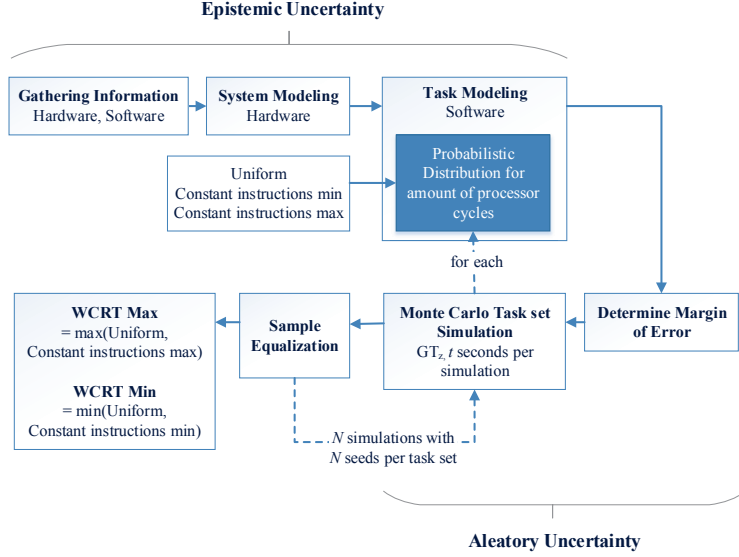


Fig. 3. Flow diagram that depicts the WCRT Estimation method for multimodal data records.

OSEK *Priority Ceiling Protocol* (PCP). It is applied in practical automotive powertrain systems. The software model consists of dynamically generated tasks with semaphore accesses to model a random task set behavior. Dynamically generated tasks have randomized task properties. For example, the amount of minimum and maximum instructions varies to model different task utilizations.

Table III represents the configuration for the case study to dynamically generate tasks for the source task set. Each task is assigned to a different OSEK task group to model task preemptions. The case study assumes a Margin of Error of 1%. It corresponds a sample size $N = 9604$. The simulation of the task sets is performed on the Cluster Computing framework OpenNebula with eight workers. Each task set is simulated 10s. Worst Case Response Times of tasks are normalized to their deadlines, so that every WCRT is in a range between $[0, 1]$. A *normalized WCRT* is defined as nWCRT. All tasks have a recurrence of 1ms to generate a large amount of job preemptions and blocking times. It simulates a high aleatory uncertainty.

We can observe following results in our Case Study:

- 1) The WCRTs of the source task set are within the estimated minimum and maximum WCRT boundaries (the WCRT confidence intervals) as shown in Fig. 4. Its range depends on the task utilization. Task T_5 has the highest task utilization in the source task set. As shown in Fig. 4, it has the lowest confidence interval of $[0.03, 0.08]$. The largest WCRT confidence interval shows T_6 with $[0.35, 0.74]$. Its range is 0.38.
- 2) We can detect extremal WCRTs and most likely WCRTs of the source task set. For it we apply a horizontal jitter on the WCRTs in Fig. 4. The

extremal WCRTs are depicted as scattered points in the upper part of the N Weibull simulations. The most likely WCRTs are depicted as fused points in the middle and lower part.

- 3) We are able to detect tasks with buffer to the maximum WCRT boundary. More load can be added to T_1, T_2, T_3, T_4, T_6 and T_7 .

VII. CONCLUSION

The contribution proposes a new probabilistic method to estimate WCRTs of tasks with different time bases in multi-core systems. The minimum and maximum WCRT boundary of each task in the source task set is estimated by the use of a set of three derived task set shapes $S^\Gamma = \tau_u^\Gamma, \tau_{\min}^\Gamma, \tau_{\max}^\Gamma$. τ_u^Γ consists of jobs with uniform distributed job computation time, τ_{\min}^Γ with minimal constant computation time and τ_{\max}^Γ with maximal job computation time. The minimum WCRT boundary is the minimum WCRT of τ_u^Γ and τ_{\min}^Γ . The maximum WCRT boundary is estimated by the use of τ_u^Γ and τ_{\max}^Γ . The aleatory uncertainty of the task set simulation is reduced with N Monte Carlo simulations. It maximizes the coverage of all possible WCRTs for each task in the taskset. Each Monte Carlo simulation is started with a unique seed to estimate different WCRTs. The result of the simulation experiments confirms the proposed WCRT estimation method. It shows that the estimation method is applicable for task sets where tasks can have different time bases. A further work is to proof the probabilistic WCRT estimation method with generated response time samples from hardware traces. Furthermore, a new estimation approach will be developed so that the WCRT can be estimated with one probability distribution.

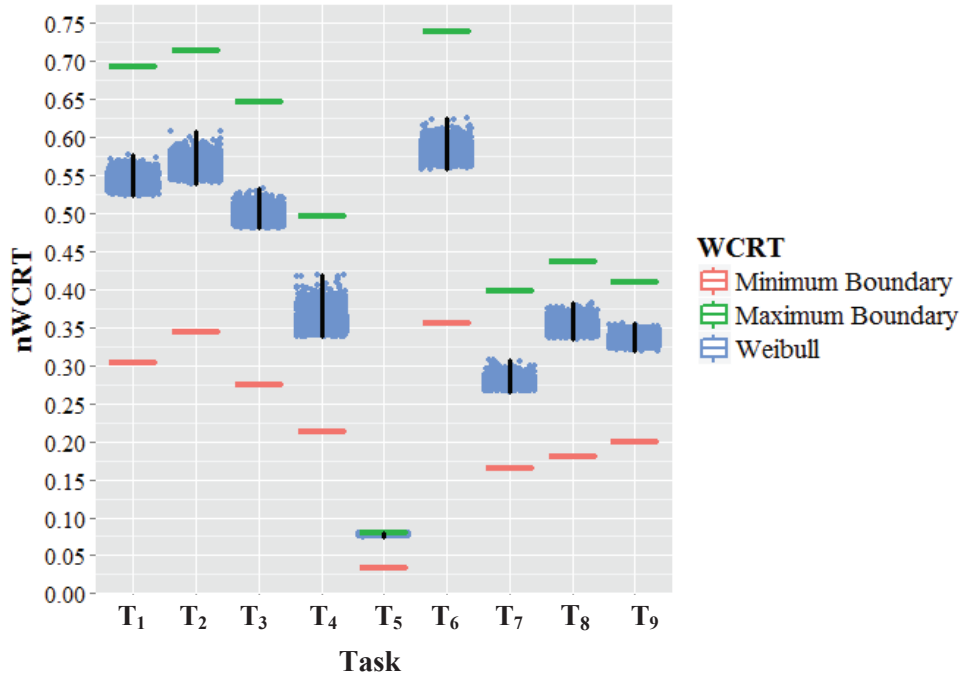


Fig. 4. Case Study: WCRT boundaries of a task set with dynamically generated tasks that are estimated with the proposed method. Each task has a periodic activation and a recurrence of 1 ms to generate a large amount of job preemptions. The minimum and maximum WCRT boundary are the confidence interval for the N WCRT values of the applied source task set with Weibull distributed job execution times. We applied a horizontal jitter on the WCRTs. With it we can detect extremal WCRTs and most likely WCRTs.

TABLE III
CONFIGURATION TO DYNAMICALLY GENERATE TASKS WITH VARIABLE PROPERTIES FOR THE SOURCE TASK SET.

Periodic Activation		Computational Time		Semaphores		Signals		
Period (ms)	Min = 0	Instructions per Block	Min = 100,000	Semaphore Count	Min = 1	Signal Count	Min = 1	
	Avg = 10		Avg = 120,000				Avg = 2	
	Max = 200		Max = 150,000		Max = 2		Max = 8	
Offset (μ s)	Min = 0	Blocks per Call Graph	Min = 1	Access Type	Read	Bit Size	Min = 8	
	Avg = 5		Max = 5				Write	Max = 256
	Max = 50							

ACKNOWLEDGMENT

This paper is supported by the Bayerische Forschungsstiftung, grant FORMUS³IC "Multi-Core safe and softwareintensive Systems Improvement Community", grant identity AZ-1165-15. The universities OTH Amberg-Weiden, Friedrich-Alexander-Universität Erlangen-Nürnberg, TH Ingolstadt, HS München, TH Nürnberg and OTH Regensburg are the network scientific partners of the Laboratory for Safe and Secure Systems (LaS³) community. Industrial partners are Airbus, Audi, Continental, Elektrobit, Infineon, iNTENCE, Timing Architects, TÜV Süd and XKRUG.

REFERENCES

[1] M. Mucha, J. Mottok, and M. Deubzer, "Probabilistic worst case response time estimation for multi-core real-time systems," in *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, June 2015, pp. 31–36.

[2] M. Joseph and P. Pandya, "Finding response times in a real-time system," *The Computer Journal*, vol. 29, no. 5, pp. 390–395, 1986.

[3] S. Edgar and A. Burns, "Statistical analysis of wcet for scheduling," in *Proceedings IEEE RTSS 2001*, 2001.

[4] R. Bril, J. Lukkien, and W. Verhaegh, "Worst-case response time analysis of real-time tasks under fixed-priority scheduling with deferred preemption," *Real-Time Systems*, vol. 42, no. 1-3, pp. 63–119, 2009.

[5] D. Griffin and A. Burns, "Realism in Statistical Analysis of Worst Case Execution Times," in *10th International Workshop on Worst-Case Execution Time Analysis (WCET 2010)*, ser. OpenAccess Series in Informatics (OASICS), B. Lisper, Ed., vol. 15. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2010, pp. 44–53.

[6] M. Liu, M. Behnam, and T. Nolte, "Applying the peak over thresholds method on worst-case response time analysis of complex real-time systems," in *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2013 IEEE 19th International Conference on*, Aug 2013, pp. 22–31.

[7] S. F. Edgar, "Estimation of worst-case execution time using statistical analysis," Ph.D. dissertation, University of York, 2002.