Study of the Reliability of Statistical Timing Analysis for Real-Time Systems

Dorin Maxim[1], Frank Soboczenski[2], Iain Bate[2], and Eduardo Tovar[1]

[1]CISTER/ISEP, Portugal , dorin@isep.ipp.pt, emt@isep.ipp.pt

[2]University of York, UK , frank.soboczenski@york.ac.uk, iain.bate@york.ac.uk

ABSTRACT

Probabilistic and statistical temporal analyses have been developed as a means of determining the worst-case execution and response times of real-time software for decades. A number of such methods have been proposed in the literature, of which the majority claim to be able to provide worst-case timing scenarios with respect to a given likelihood of a certain value being exceeded. Further, such claims are based on either some estimates associated with a probability, or probability distributions with a certain level of confidence. However, the validity of the claims are very much dependent on a number of factors, such as the achieved samples and the adopted distributions for analysis. This paper is the first one that puts side by side existing state of the art statistical and probabilistic analysis techniques, using the probabilistic analysis as the ground truth in order to asses the applicability and performance of the statistical technique. The evaluation clearly shows that for the experiments performed the approach can identify clear differences between a range of techniques and that these differences can be considered valid based on the trends expected from the academic theory.

1. INTRODUCTION

Edgar [13] produced the first work that uses statistical analysis methods, in the form of Extreme Value Analysis (EVA), to understand the worst-case behaviour of software. In the case of Edgar's work, execution times were used to form a Gumbel distribution. The distribution allowed a Worst-Case Execution Time (WCET) to be selected at a given exceedance threshold level, i.e. how likely it is for a certain value of WCET to be exceeded. The application of EVA-based techniques to the WCET analysis problem was extended to combine measurements with static analysis to form hybrid analysis [8] and the use of copulas to allow statistical analyses results to account for possible dependencies between different behaviours of the system [9]. Statistical analysis techniques, other than fitting a Gumbel distribution, have been applied to improve the results obtained including block maxima-based approaches [17,26] and theoretical analyses [34]. In addition to WCET, EVA-based techniques have been applied in other areas of real-time systems research including for Worst-Case Response Times (WCRT) of tasks [33] and messages [37].

In this paper we are interested in assessing the correctness and efficiency of statistical analyses in deriving exceedance probabilities for response times that an arbitrary task in the system may exhibit. This is based on some observed response times of that particular task. Correctness is when the technique predicts a value X is exceeded with a probability of Y then the actual probability is first and foremost less than Y (i.e. the approach tends to be pessimistic), and secondly the most correct approach is the one for which the actual probability is as close as possible to Y. Efficiency is a complementary property that builds an understanding of which approach is most correct given a finite amount of effort. For example, given Z observations which approach is both pessimistic and has the closest actual probability to Y.

Recently, this problem was also highlighted in the *6th Real-Time Scheduling Open Problems Seminar* [23] under the premises that probabilistic worst case response time (pWCRT) distributions would best be characterized using Gumbel distributions. In the present paper, not only do we assess how well the Gumbel distribution characterized pWCRTs, but also how well other distributions perform in comparison. It is hoped that the information obtained from this work can be used in certification cases as part of the confidence attributed to the claims made, concerning the timing requirements being met, and then real-time systems can be designed to compensate (e.g. through fault tolerance) for the fallibility of the techniques used [24].

The contributions of this work are as follows:

1. a methodology is proposed that when combined with a task set simulator allows a controlled experiment to be performed,
2. a discussion about the correctness of Measurement Based Techniques for deriving response time probability distributions based on traces collected at run-time, and
3. experimental results are presented, showing the ability of different statistical techniques to accurately predict the Worst-Case Response Times (WCRT) of tasks

The structure of the paper is as follows. Firstly, in section 2, a survey of related work is performed to better understand the analyses that have been performed in the past. This is not intended to be a comprehensive survey of probabilistic and statistical analysis methods applied to the domain of real-time systems. Instead it is to understand the broad categories of such techniques that have previously been used. That said, we believe the main pieces of work have been covered. In Section 3 we present the system model and terminology that we use throughout the paper, and then in Section 4 we present a summary of the tools that we use in our investigation. Next, in Section 5, the experimental approach is outlined together with the questions that we seek answers for, which is then followed by the results of our evaluation in Section 6. Finally Section 7 con-

RTNS 2015, November 04-06, 2015, Lille, France

DOI: http://dx.doi.org/10.1145/2834848.2834878

cludes the paper.

2. RELATED WORK

Looking at the development of temporal analysis for real-time embedded systems using probabilistic (sometimes called stochastic) methods and statistics using measurements, the related work mainly lies in the areas of WCET (well established in the state of the art) and WCRT (since very recently) analysis. The main difference between statistical and probabilistic methods lies in the fact that statistical methods use samples to form distributions for analysis whereas probabilistic methods use predicted distributions within analysis. In this paper we focus on the second area of applicability for probabilistic and statistical analyses, namely, we validate statistical analysis for WCRT by using state of the art probabilistic analysis. More details about probabilistic and statistical timing analyses are given below. In the rest of the paper we will be using the terms *sample* and *trace* interchangeably to represent a single value (e.g. response time) taken from the simulation or execution of the system.

Considering WCET analysis, it should be mentioned that, early work on Static Probabilistic Timing Analysis (SPTA) [18] [31] (for deterministic systems) relied on knowing the probabilities that different paths would be taken. SPTA was later derived for systems with a random replacement caches [14]. This analysis was based on the reuse distance of memory accesses. It was superseded by analysis for the more effective evict-on miss policy (which dominates the evict-on-access approach assumed by previous papers) [19]. In 2014, a more effective SPTA [4] was introduced based on cache contention, and a precise analysis of a limited set of focussed cache blocks. This method was further improved using cache simulation techniques [3]. Comparisons between MBPTA [15] and the SPTA methods based on reuse distance were shown in [1].

2.1 Probabilistic Timing Analysis

Probabilistic timing analysis and the associated methods model the target parameters (e.g. the task's WCET, Minimum Inter-arrival Time (MIT)) by using random variables. For WCET analysis multiple avenues have been explored, starting in 1995 when [41] introduced an analysis for tasks that have periodic releases but variable execution requirements. The algorithm, called Probabilistic Time Demand Analysis (PTDA) computes the worst case probability that a task misses its deadline by bounding the total amount of processor time demanded by all higher priority tasks. Since the algorithm is based on a bound of the processor demand of higher priority tasks, it is highly pessimistic. The next step towards an exact probabilistic analysis was made by [22] with the introduction of the Stochastic Time Demand Analysis (STDA) for tasks that have probabilistic execution times, computing a lower bound on the probability that jobs of each task will meet their respective deadlines. Later on, [21], [29], [32] refined STDA into an exact analysis for real-time systems that have random execution times. The execution time is represented as a general random variable and the priorities may be job-level or task-level. The analysis is proven to be bounded in time and exact for both cases when the system utilisation is lower or greater than one. Due to the resource costs (computation time and memory) of convolution, the proposed analysis can only be applied for small task systems - this problem was later studied in [40] and [35] with the introduction and refinement, respectively, of re-sampling techniques meant to decrease the size of the distributions that are convolved while introducing minimal pessimism.

Another branch of probabilistic analysis for real-time system considers inter-arrival times as being described by random variables. Most notable is the work of Broster et al. [11] and [12] where jobs with random arrival times are generated by various sources, e.g. external interrupts, network interfaces, etc.. In [15], the authors introduce a framework for computing response time distributions in the case when in the system there are tasks that have random arrivals, given as independent discrete random variables. The rest of the parameters of the tasks are deterministic. The output of the analysis is the response time probability distribution of the first release of an analysed task, considering that all tasks are released synchronously. Both frameworks mentioned have assumptions on the system model that are different to 'real' systems. For example Broster assumes Poisson arrivals whereas [15] only assumes the MIT is random and that the analysis stops when the deadline is reached.

Very few contributions exist that take into consideration multiple sources of variability in the tasks' parameters. Lehoczky [30] first extended queuing theory with real-time hypotheses, which was later improved by Zhu [44]. Task instances have arrivals described by a Poisson process and exponentially distributed execution times. The main problem with these methods is using the same probability distributions to model the execution times and inter-arrival times of all tasks is not realistic. In 2013 Maxim and Cucu [34] extended the probabilistic timing analysis of Diaz et al. [21] to the case where tasks are allowed to have multiple sources of variability in their parameters, that is, not just the execution time is given as a random variable but the inter-arrival times and deadlines as well, the analysis being able to cope with multiple probabilistic parameters given as random variables and returns a probability distribution representing the possible response times of the task if it were to be instantiated at the same time as all higher priority tasks, i.e. the synchronous case. This distribution is proven to be safe, since the synchronous case is an upper-bound for all possible cases that might arise in the system, i.e. in the non-synchronous cases there is less interference from higher-priority tasks and hence the response times can not be higher than in the synchronous case.

2.2 Statistical Timing Analysis

In recent years, Edgar and Hansen applied the Extreme Value Theory (EVT), which is used to handle extreme scenarios of events, to the problem of WCET analysis. Specifically, Edgar [13] presents the initial work on using EVT for WCET estimation, by firstly fitting the raw measured execution time sample to the Gumbel-Max distribution based upon an unbiased estimator. A WCET estimate is then calculated using an excess distribution function. Hansen [26] improves the work by using a block maxima approach for the estimation of the probabilistic WCET, rather than fitting the raw execution time sample to the Gumbel-Max distribution directly.

For WCRT analysis, Zeng [43] has presented a statistical method, which computes probability distributions of message response times in CAN by using a mixed model of the Gamma distribution and the degenerate distribution. Lu [33] has addressed limitations of EVT-based analysis based on a systematic analysis framework. Further, the analysis method generally can be applied on different systems, since the target system is considered as a black box without requiring estimates of parameters such as WCETs of tasks.

The state of practice in industry is that many companies developing their real-time embedded systems have no means for timing analysis and are forced to rely on testing to find timing-related problems by using measurements [24, 28]. Often, the maximum observed execution and response-times (referred to as High-Water-Mark (HWM) execution and response times) have a safety margin based on experts' experience. However for the purposes of certification, the need to add such a margin undermines the integrity of the analysis in the first place and the size of the margin is hard to justify as the inaccuracy in the analysis is not known. That said, there are recognized safety margins for existing systems however these tend to be for simple processors (i.e. without caches and simple pipelines) and it would be difficult to justify revised safety margins if more modern processors are deployed [6,7].

3. SYSTEM MODEL AND TERMINOLOGY

A random variable $\mathcal{X}$ has a probability function (PF) $f_{\mathcal{X}}(\cdot)$ with $f_{\mathcal{X}}(x) = P(\mathcal{X} = x)$. The possible values of $\mathcal{X}_i$ belong to the interval $[X^{\min}, X^{\max}]$. In this paper we associate the probabilities with the possible values of a random variable using the following notation:

$$\mathcal{X} = \begin{pmatrix} X^0 = X^{min} & X^1 & \cdots & X^k = X^{max} \\ f_{\mathcal{X}}(X^{min}) & f_{\mathcal{X}}(X^1) & \cdots & f_{\mathcal{X}}(X^{max}) \end{pmatrix} \quad (1)$$

where $\sum_{j=0}^{k_i} f_{\mathcal{X}}(X^j) = 1$. A random variable may also be specified using its cumulative distribution function (CDF) $F_{\mathcal{X}}(x) = \sum_{z=X^{min}}^{x} f_{\mathcal{X}}(z)$.

Definition 1. *Two random variables $\mathcal{X}$ and $\mathcal{Y}$ are (probabilistically)* **independent** *if they describe two events such that the outcome of one event does not have any impact on the outcome of the other.*

Definition 2. *[32] Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be two random variables. We say that $\mathcal{X}_1$ is* **greater than** *$\mathcal{X}_2$ if $F_{\mathcal{X}_1}(x) \leq F_{\mathcal{X}_2}(x)$, $\forall x$, and denote it by $\mathcal{X}_1 \succeq \mathcal{X}_2$.*

For example, in Figure 1 $F_{\mathcal{C}_1}(x)$ never goes below $F_{\mathcal{C}_2}(x)$, meaning that $\mathcal{C}_1 \succeq \mathcal{C}_2$. Note that $\mathcal{C}_2$, $\mathcal{C}_4$ and $\mathcal{C}_4$ are not comparable with each-other.

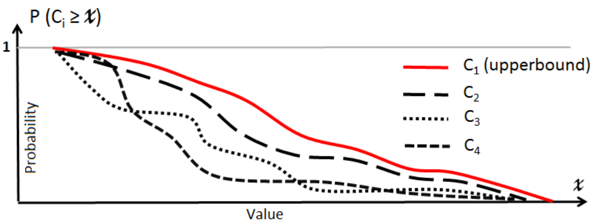

Figure 1: Possible relations between the 1-CDFs of various random variables

We consider a system of n synchronous tasks $\{\tau_1, \tau_2, \ldots, \tau_n\}$ to be scheduled on one processor according to a preemptive fixed-priority task-level scheduling policy. Without loss of generality, we consider that τ_i has a higher priority than τ_j for $i < j$. We denote by $hp(i)$ the set of tasks' indexes with higher priority than τ_i. By synchronous tasks we understand that all tasks are released simultaneously the first time at $t = 0$.

Each task τ_i generates an infinite number of successive jobs $\tau_{i,j}$, with $j = 1, \ldots, \infty$. All jobs are assumed to be independent of other jobs of the same task and those of other tasks.

Each task τ_i is a generalized sporadic task [36] and it is represented by a probabilistic worst case execution time (pWCET) denoted by $\mathcal{C}_i$ (In this paper, we use calligraphic typeface to denote random variables) and by a probabilistic minimal inter-arrival time (pMIT) denoted by $\mathcal{T}_i$. These notions are defined as follows.

Definition 3. *The probabilistic execution time (pET) of a job of a task describes the probability that the execution time of the job is equal to a given value.*

Definition 4. *The probabilistic worst case execution time (pWCET) of a task describes the probability that the worst case execution time of that task is equal to a given value.*

A safe pWCET $\mathcal{C}_i$ is an upper bound on the pETs $\mathcal{C}_i^j$, $\forall j$ and it may be described by the relation $\succeq$ as $\mathcal{C}_i \succeq \mathcal{C}_i^j$, $\forall j$. Graphically this means that the Complementary CDF (1-CDF) of $\mathcal{C}_i$ is always above the 1-CDF of $\mathcal{C}_i^j$, $\forall j$. The 1-CDF describes how often a particular variable is above the exceedance level.

Following the same reasoning the probabilistic minimal inter-arrival time (pMIT) denoted by $\mathcal{T}_i$ describes the probabilistic minimal inter-arrival times of all jobs.

Definition 5. *The probabilistic inter-arrival time (pIT) of a job of a task describes the probability that the job's arrival time occurs at a given value.*

Definition 6. *The probabilistic minimal inter-arrival time (pMIT) of a task describes the probability that the minimal inter-arrival time of that task is equal to a given value.*

A safe pMIT $\mathcal{T}_i$ is a bound on the pITs $\mathcal{T}_i^j$, $\forall j$ and it may be described by the relation $\succeq$ as $\mathcal{T}_i^j \succeq \mathcal{T}_i, \forall j$. Graphically this means that the complementary CDF (1-CDF) of $\mathcal{C}_i$ is always above the 1-CDF of $\mathcal{C}_i^j$, $\forall j$.

Hence, a task τ_i is represented by a tuple $(\mathcal{C}_i, \mathcal{T}_i)$. A job of a task must finish its execution before the arrival of the next job of the same task, i.e., the arrival of a new job represents the deadline of the current job[1]. Thus, the task's deadline may also be represented by a random variable $\mathcal{D}_i$ which has the same distribution as its pMIT, $\mathcal{T}_i$. Alternatively, we can consider the deadline described by a distribution different from the distribution of its pMIT if the system under consideration calls for such model [2,38], or the simpler case when the deadline of a task is given as one value. The latter case is probably the most frequent in practice.

As stated in [16], since we consider probabilistic worst case values (for MIT and WCET), then the random variables are (probabilistically) independent.

Definition 7 (Job deadline miss probability)**.** *For a job $\tau_{i,j}$ the deadline miss probability $\mathsf{DMP}_{i,j}$ is the probability that the j^{th} job of task τ_i misses its deadline and it is equal to:*

$$\mathsf{DMP}_{i,j} = P(\mathcal{R}_{i,j} > D_i). \quad (2)$$

where $\mathcal{R}_{i,j}$ is the response time distribution of the j^{th} job of task τ_i.

It was shown in [34] that the case when tasks are simultaneously released yields the greatest response time distribution for each task respectively. Here, greatest is defined with respect to the relation $\succeq$ and it indicates that the response time distribution of the first job upper bounds the response time distribution of any other job of that task. Since we are considering synchronous tasks, calculating the response time distribution of the first job of a task provides the worst case response time distribution of the task and, implicitly, its worst case DMP.

4. DETAILS OF THE ANALYSIS TOOLS

In order to help us in our endeavour of quantifying the applicability of statistical measurement based techniques to real-time system we have developed a suite of tools including (i) probabilistic task-set generators, (ii) simulator for the generated task-sets, (iii)

[1]In the analysis of GMF tasks this is known as the frame separation constraint.

probabilistic analysis tool for the generated tasks and (iv) statistical analysis tool to analyse the traces collected during simulations. These tools are freely available by email request to the authors. We describe the tools in detail below.

4.1 Task-set Generator and Simulator

The purpose of the simulator is to allow a sufficiently complex set of tasks to be simulated and a ground truth to be established. A decision was taken to create a bespoke simulator rather than use an existing simulator. The reason is existing simulators do not have probabilistic task sets as inputs, i.e. featuring tasks that have their WCET and/or their MIT given as random variables. The simulator used has the following characteristics, which are chosen to be suitably complex and relatively realistic. It was judged that absolute realism is not critical. What is important is not introducing experimental bias and whether the analysis approaches are able to deal with whatever data was fed into them.

1. Task sets consist of a number *N* of sporadic tasks. Depending on the experiment performed and the properties that we are looking to analyse, the number of tasks in the system is varied. If not stated otherwise, the default number of tasks is $N = 5$.
2. All parameters of each task can be given as random variables. For the sake of simplicity, only the execution times and inter-arrival times are described by distributions (with a defined profile between a minimum and maximum value), and are called pWCET and pMIT.
3. The deadline is given as a single value taken from the theoretical distribution at the required exceedance threshold, i.e. if we require a deadline that should only ever be exceeded 1 in a million times then the response time equating to 10^{-6} is taken from the Complimentary Distribution Function (1-CDF). The 1-CDF gives the likelihood that a given value is ever exceeded.
4. The number of values in each distribution can be given as input to the random task set generator. If the number is 1 then the parameter has a single value (the worst case value) and it is called a deterministic parameter. The simulator and analyser employed in the present work can cope with any combination of probabilistic and deterministic parameters of tasks. If not stated otherwise, each random variable has 5 values.
5. The only parameters of each task are: pWCET, pMIT and Deadline (D)
6. The task sets are scheduled according to a Fixed Priority Preemptive Scheduling (FPPS) policy.
7. The task set generator generates two deterministic task sets according to the Uunifast [10] algorithm, one of the sets having an utilisation equal to $U_{min} \leq 1$ and the other set having an utilisation equal to $U_{max} \geq 1$. Both U_{min} and U_{max} are given as input to the task set generator. For sake of simplicity, in this work, we have fixed $U_{min} = 0.1$ and $U_{max} = 4$. The tasks in the two sets are then paired two by two and each pair gives the minimum and maximum values of the pWCET and pMIT of a probabilistic task.
8. The pWCET distribution is monotonically non-increasing, i.e. an instance (job) of the task is more likely to have a lower execution time than a large one.
9. The pMIT distribution is monotonically non-decreasing, i.e. two instances (jobs) of the task are more likely to be separated by a large inter-arrival time than a small one.
10. As the pWCET distributions are decreasing and the pMIT distributions are increasing and the fact that the minimum and maximum utilisation of the probabilistic task set can be changed as necessary, task sets are generated with expected utilisation less than 1, so that deadline misses are extremely rare events. In this work, it is interesting to also understand how the analysis copes with these cases.
11. When the simulator starts, all tasks in the system are instantiated, hence, the first job is in a critical instant situation. Subsequent jobs have random arrivals as described by their pMIT distributions and the inter-arrival time between any two consecutive jobs is no less than the smallest value in the tasks' pMIT distribution. Further critical instants may occur during the run-time of the system.

The simulator represents the temporal operation of a real system without being slowed down by executing any of the functionality. The execution of these tasks using FPPS should be suitably complex, although not necessarily representative of real systems.

4.2 Probabilistic Analysis Tool

The technique presented in [34] is implemented in a probabilistic analysis tool and, in parallel to the simulator, is used to calculate the probabilistic WCRT (pWCRT), R_i, of each generated task. The response time distribution thus computed is used as a ground truth in our experiments. The analysis is summarized in the following recurrence relation:

$$\mathcal{R}_n^{i,j} = (\mathcal{R}_n^{i-1,head} \oplus (\mathcal{R}_n^{i-1,tail} \otimes \mathcal{C}_m^{pr})) \otimes \mathcal{P}_{pr} \quad (3)$$

where:

- n is the index of the task under analysis;
- i is the current step of the iteration;
- j represents the index of the current value taken into consideration from the pMIT distribution of the preempting task;
- $\mathcal{R}_n^{i-1,head}$ is the part of the distribution that is not affected by the current preemption under consideration;
- $\mathcal{R}_n^{i-1,tail}$ is the part of the distribution that may be affected by the current preemption under consideration;
- m is the index of the higher priority task that is currently taken into account as a preempting task;
- $\mathcal{C}_m^{pr}$ is the execution time distribution of the currently preempting task; and
- $\mathcal{P}_{pr}$ is a fake random variable used to scale the j^{th} copy of the response time with the probability of the current value i from the pMIT distribution of the preempting task. This variable has one unique value equal to 0 and its associated probability is equal to the i^{th} probability in the pMIT distribution of the preempting job.

For each value $v_{m,i}^j$ in $\mathcal{T}_{(m,j)}$ for which there exists at least one value $v_{n,i}$ in $\mathcal{R}_n^{i-1}$ so that $v_n^i > v_{m,i}^j$, the distribution $\mathcal{R}_n^{i-1}$ is split in two parts:

- $\mathcal{R}_n^{i-1,head}$ which contains all values $v_{n,i}^-$ of $\mathcal{R}_n^{i-1}$ that are less or equal than $v_{m,i}^j$, i.e., $v_{n,i}^- \leq v_{m,i}^j$, and
- $\mathcal{R}_n^{i-1,tail}$ which contains all values $v_{n,i}^+$ of $\mathcal{R}_n^{i-1}$ that are greater than $v_{m,i}^j$, i.e., $v_{n,i}^+ > v_{m,i}^j$.

The iterations end when there are no more arrival values $v_{m,i}^j$ of any job i of any higher priority task τ_m that is smaller than any value of the response time distribution at the current step. A stopping condition may be explicitly placed in order to stop the analysis after a desired response time accuracy has been reached. For example, the analysis can be terminated once an accuracy of 10^{-9} has been reached for the response time. Further details about the Probabilistic Analysis Tool can be found in [34].

4.3 Statistical Analysis Tool

Based on the literature survey in Section 2, the following statistical analysis techniques are used:

1. fitting a Weibull (W) distribution;
2. fitting a Gumbel (G) distribution;
3. fitting a Normal (N) distribution; and
4. applying the Block-Maxima (BM) method presented in [33]

The first three of these analyses are achieved using a standard function in R, *samlmu* (part of the lmom package) [27], which was chosen as other fitting functions, e.g. *fitdistr*, were found not to be robust (i.e. the analysis software often exited with unrecoverable errors) due to careful selection of initial fitting parameters being needed. The advantage of *samlmu* is that it did not require initial parameters. The Block-Maxima analysis can be summarized by the following steps with full details being available in [33]. All the values chosen in the algorithm are based on standard statistical test tables [20] that have subsequently been validated through trials.

1. Firstly, a set RT of training data is randomly sampled from the overall set of available data. Specifically, each sub-training data (RT) is taken such that an i.i.d. assumption can be made, and such that there are sufficient samples allowing for appropriate tuning of the final analysis. Again the *samlmu* package is used for to fit a Gumbel distribution to the set RT. The number of samples is progressively decreased (by dividing the number of samples by 2) from the maximum, 191, until a suitable fit is achieved, i.e. a Goodness of Fit (GoF) better than 0.95. The minimum number of samples in set RT is 6. If a sufficient GoF is not achieved then the best achieved is taken.
2. Then, a posterior statistical correction process is performed to decompose the reliability target for the WCRT of tasks into a number of probabilities to be used in the statistical analysis, i.e. the GoF of 0.95, the statistical confidence associated with the limited sampling, again 0.95, and the exceedance threshold, ET_N used in the final step of the analysis.
3. Given an appropriate task RT sub-training data, the statistical analysis in the posterior statistical correction process is tuned such that the maximum of a probability distribution (in this case the Gumbel Max distribution) generated with a given set of parameters is a sufficiently close match, at the required confidence level, to the maximum of the actual distribution of the sub-training data. At this step, a calibrated and tight PDF histogram of the task WCRT consisting of the estimates of the maximum of each task RT sub-training data, $WCRT_{BM}$, is obtained at a given exceedance threshold, RT_{BM}.
4. The final stage is to fit a normal distribution, again using the *samlmu* package, to the set of $WCRT_{BM}$ values and then read off the final WCRT at the chosen exceedance threshold, i.e. RT_N which is taken as 0.95.

For each of the approaches, the following parameters are employed.

1. Number of samples used: 10000.
2. Confidence interval chosen: 99.9999%.

5. EXPERIMENTAL APPROACH

This section will first introduce and define the term *Ground Truth* in the context of this work. It will then highlight possible confounds of the chosen simulation approach in terms of validity leading to the subsequent presentation of the objectives and research questions.

5.1 Establishing the Ground Truth

An important part of this work is knowing the ground truth of the techniques that are compared. The ground truth is essentially the accuracy of the approaches. In this work, a task set generator combined with a task-level simulator is preferred over using *real* software. The reasons for this decision are as follows:

1. *Controllability* - If *real* software is to be used then it would be very difficult to control particular execution times of individual tasks, patterns of release of tasks etc. This would make it difficult to perform evaluations that looked at how key parameters, e.g. task set utilization and number of samples, affect the efficacy of the techniques. It is assumed that the samples are (randomly) chosen from the set that are available following the simulation of a task set. The strategy proposed allows the key parameters to be decided a priori to each experiment and if needed specific experiments to be repeated.
2. *Comparison to the Ground Truth* - With *real* software it is widely recognized [42] that except for the simplest of software the exact WCET cannot be established. For a real system the same can be argued for the WCRT, principally because establishing the actual WCRT is dependent on exact WCETs being known. Also real systems have overheads which are hard to analyse, and most published pieces of work do not account for them [39]. This means there is no way to evaluate the efficacy of techniques. Having a ground truth is the singularly most important factor in the decision process and the whole purpose of the presented work is the ability to provide this form of evaluation. The need to compare against a ground truth, i.e. an exact result, means that a sufficient and necessary schedulability test is needed. The work in this paper is performed using a uniprocessor system.
3. *Realism* - It could be argued that the proposed approach is not realistic as it doesn't have the characteristics of *real* software. However in practice any evaluation not performed on the *actual* system is not realistic. In addition it is recognised that many benchmarks used in real-time systems are not realistic. For instance, the benchmarks used for most WCET research are those supported by Mälardalen University (www.mrtc.mdh.se/projects/wcet/benchmarks.html) and if *real* software were to be used then they would be the likely choice. In fact the maintainers of these benchmarks recognize that they are not realistic [25] even though they are very widely used. In contrast the proposed approach can be made as realistic as required. For example execution times can be drawn from any form of distribution including that of the *actual* system with the significant advantage that large-scale evaluations can be performed at comparatively low cost, and can also be performed early on in the development lifecycle, to gain confidence in any proposed approach.

Given the choice of a simulation-based approach, it is important to consider the threats to validity that come from the use of a simulator, and argue why these are acceptable. This information can also be used to reduce the size of the threat. The main threats are as follows:

1. *Unrealistic WCETs* - A task set simulator will not normally execute real software but instead each task will have release conditions (e.g. periods, jitter and arrival rates) and execution times. The key issue is where a random function is used to control a particular parameter (e.g. execution time, jitter or arrival rate) the distribution is not real, however it will be controllable such that the chosen distribution of samples is attained.
2. *Unrealistic Overheads* - In a similar fashion to *Unrealistic WCETs* the simulator will not execute a real-time operating system, however overheads can be introduced into the system and controlled.

3. *Unrealistic Dependencies* - Real systems have two types of dependencies, implicit and explicit. Explicit dependencies are considered here to be transactional requirements, e.g. precedence constraints between tasks. Implicit dependencies are those that come from the tasks sharing dependencies such as caches and execution dependencies [33]. Explicit dependencies can easily be built into task set simulators. Some implicit dependencies would be represented within the way tasks' execution times are generated but these can also be modeled. Its questionable to what extent these need to be modeled as they can be subsumed into the other randomized parameters, e.g. the tasks' execution times.

The reason these threats do not affect the validity of the results is the fact that the threats can be incorporated into the simulators according to a profile of our choosing, i.e. we can control the precise profile of the system that is to be examined. Importantly, using a simulator allows for the same data to be used across all the statistical techniques that we investigate as well as allowing ground truths to be found.

Out of the two approaches that can be considered sound (Gumbel and Block-Maxima), Gumbel appears to give the tightest results. However, as noted in the previous research question, the Gumbel distribution can be unsafe at times and would require post-processing correction. As such correction has not be considered in this evaluation, it is therefore not possible to definitively conclude that the Gumbel technique is superior to the Block-Maxima technique. Regarding the Block-Maxima technique, further research needs to be carried out to determine the level of pessimism that it introduces, and if this level of pessimism will result in a false positive rate (where it incorrectly classifies schedulable task sets as unschedulable) which is too high.

5.2 Experimental Objectives

The objectives of this paper are to develop an experimental method for understanding how well a number of statistical techniques predict the WCRT of tasks and how various parameters affect this ability, and present the results of applying the method to show the forms of insight that are obtained. The following are the objectives that are used to judge their relative abilities. For each of these objectives specific research questions are presented (in italics).

1. *RQ1 - Do any of the statistical analysis approaches provide sound results?* This is assessed by inspecting the 1-CDF for the analysis approach and judging whether it always gives a value greater than or equal to the theoretical analysis [34].
2. *RQ2 - Which of the sound analysis approaches gives the tightest bound?* This is assessed by comparing the WCRT obtained by the analysis $method$, $WCET_E^{method}T$ with that from the theoretical analysis, $WCRT_R^{maxim}T$, at a given exceedance threshold, RT, i.e.
$tightness(RT) = \frac{WCRT_R^{method}T}{WCRT_R^{maxim}T}$, where $method$ can be Normal, Weibull, Gumbel or Block-Maxima.
It is expected the tightness of the analysis $method$ is dependent on RT.

6. EVALUATION RESULTS

The starting point of this investigation is the hypothesis that the exact response time probability distribution of a task (like its execution time probability distribution) can be approximated by taking measurements at run-time and fitting them to distributions, such as Gumbel-Max, Weibull, etc.

In order to asses this claim we have randomly generated, simulated and analysed several thousands of task-sets using the tools and procedures outlined in Section 4. Due to lack of space we present here only small batches of task-sets, which we think are representative for how well different statistical techniques perform.

For clarity we will present each statistical technique separately, side by side with the theoretical analysis to see its performance.

For this section we have generated 100 task-sets, each with 5 tasks and each random variable (pWCET and pMIT) of each task having 5 values. We have collected 10000 traces for the least prioritary task in each set, i.e. task on priority level 5 in this case. Five tasks per task set and five values per random variable is enough to have significant results without the need to use approximation techniques such as re-sampling [35] to speed up the probabilistic analysis. Indeed, the probabilistic analysis is highly time consuming and a way to make it tractable is to use techniques such as re-sampling, but this kind of techniques introduces inaccuracy in the obtained results, and we are interested in having an exact theoretical response time distribution that matches the empirical distribution of response times observed during simulation.

The minimum utilizations of the 100 analysed task-sets are between 0.1 and 0.3, the maximum utilizations between 0.8 and 4, and the expected utilizations are between 0.2 and 0.8. With such large worst case utilisations, if the task-sets were to be analysed using existing state of the art (deterministic) response time analysis, they would mostly be deemed unschedulable, but since the expected utilisation is much less than the worst case one, then, the probability that any job actually misses its deadline is vanishingly small, the average DMP among all the 100 task-sets being 0.04 with a maximum of 0.71, with more than half of the analysed task-sets having DMPs smaller than 10^{-6}. We have chosen a wide range of utilizations and failure probabilities in order to see how the statistical techniques perform for different types of task-sets.

6.1 Applicability and Performance of the Gumble-Max Distribution

The first statistical technique that we take a closer look at is fitting the collected response time traces to the Gumbel distribution. Figure 2 depicts the DMPs of the 100 task-sets, as they are computed by the theoretical analysis and derived by the Gumbel statistical technique. We can see that the two curves overlap almost completely, indicating that Gumbel does a very good job of approximating the theoretical probability of deadline miss (for each task).

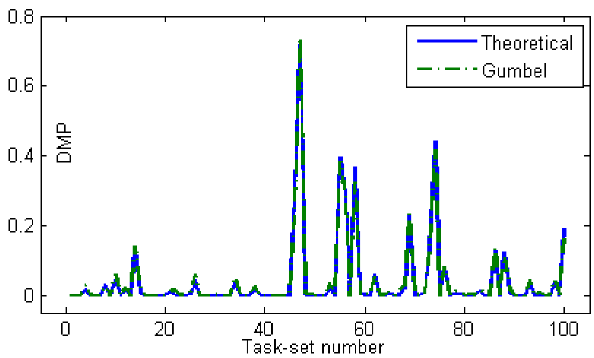

Figure 2: DMPs of 100 task-sets as derived by Gumbel and computed theoretically

A better way of seeing how close the two sets of results are is to plot the Gumbel ones as a ratio of the theoretical ones, as described in *RQ2* in Section 5.2. That is, for each task, the DMP derived by

Gumbel is divided by the DMP computed theoretically. In this way we see how many times larger (or smaller) the statistical results are compared to the theoretical ones. This relative representation can be seen in Figure 3, which has a logarithmic Y-axis. The theoretical DMPs are also divided by themselves, resulting in a straight line corresponding to 1 on the Y-axis and it serves as the base-line for the statistical results. That is, if the DMP derived by Gumbel for a task is above the base-line, then Gumbel provides a DMP larger than the theoretical one, and this pessimism means that the result can be treated as sound. The size of the Gumbel bar with respect to the base-line represents the accuracy of the statistical technique, i.e., the larger the bar, the less accurate the statistical result is. Similarly, if a bar is below the base-line, the statistical technique has produced an optimistic result, i.e., smaller than the exact DMP computed by the theoretical analysis.

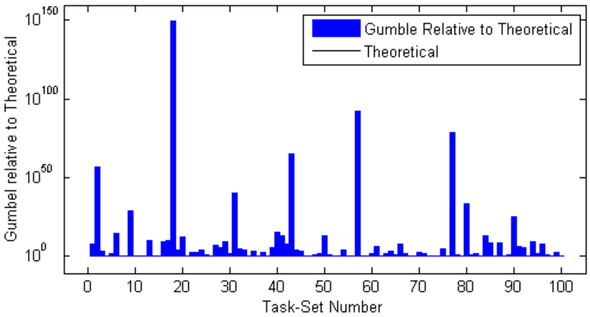

Figure 3: DMPs of 100 task-sets as derived by Gumbel relative to the DMPs computed theoretically

Concerning the tightness of the statistical technique we note that the smaller the theoretical DMP gets, the less tight the statistical results are. This is explained by the fact that Gumbel is a continuous function, having a constant slope, while the response time probability distribution has abrupt decreases every time there is a potential preemption and hence it gets further and further away from the Gumbel function (this phenomenon can be observed in Figure 12 of Section 6.5). For example in Figure 3 the result returned by Gumbel for task-set 18 seems to be very pessimistic, but this is not necessarily true. The theoretical DMP of the task-set is $5.9843e10^{-173}$ but the DMP derived by Gumbel is $2.925e10^{-23}$ and, while it does have a considerable pessimism, it is still accurate enough to correctly qualify the task-set as schedulable according to any certification standard in place to date (for instance, according to the aerospace standards, a task at the highest level of criticality must not have a failure rate larger than 10^{-9} per hour of operation [5]). This is similar for task-sets $2, 31, 43, 57$ and 78, which all have extremely small theoretical DMPs (between 10^{-50} and 10^{-110}), and so the statistical analysis will still classify them as schedulable even with a large amount of pessimism.

In order to see for how many task-sets the statistical analysis derives optimistic results and the degree of this optimism, we need to take a closer look at the lower part of Figure 3. Figure 4 provides a zoomed in view around the baseline of Figure 3. We can see that 11 out of the 100 analysed task-sets have been optimistically analysed, but the degree of inaccuracy is very small. For example, the most optimism can be observed for task-set 59, but even in this case the difference between the statistical result ($2.93e10^{-3}$) and the exact one ($4.98e10^{-3}$) is very small, both results being in the range of 10^{-3}. The other 10 optimistical results are even closer to the exact ones, so we can consider them as tight approximations. Note that the average theoretical DMP of the 100 task-sets is $3.9368x10^{2}$ and the average statistical DMP is $4.0602x10^{-2}$.

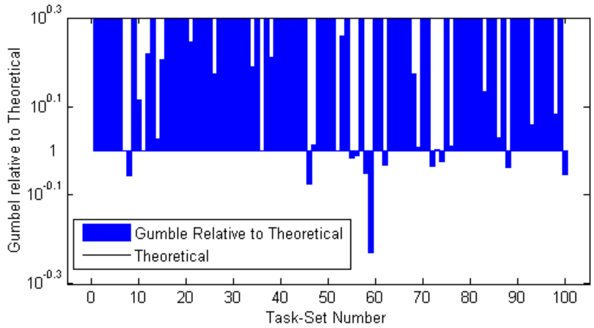

Figure 4: Close-up around the baseline of the relative DMPs derived with the Gumbel technique for 100 tasks

For all these reasons we consider that the Gumbel distribution is adequate to be used in deriving response time distributions based on execution traces, providing tight approximations of the theoretical deadline miss probabilities even in the cases where it might be optimistic.

6.2 Applicability and Performance of the Block-Maxima Technique

We have done the same investigation for the Block-Maxima technique in order to asses its applicability for real-time system analysis. In Figure 5 the DMPs of the 100 task-sets are plotted, as derived by the statistical and the theoretical analysis. It is easy to see that Block-Maxima (labelled as Lu in all figures) is not as accurate as Gumbel-Max, the amount of pessimism introduced by only considering the subset of the largest observed response times being considerable. A visual representation of this pessimism is depicted in Figure 6.

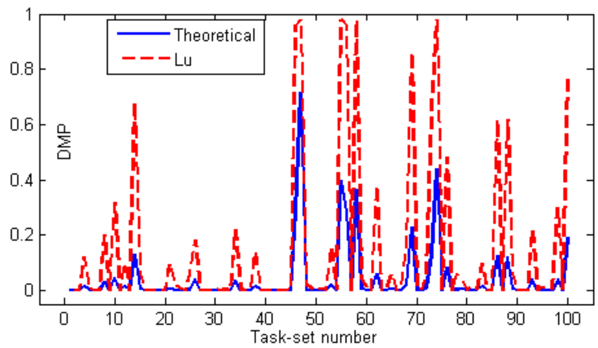

Figure 5: DMPs of 100 task-sets as derived by Block-Maxima (labelled as Lu) and computed theoretically

The close look on the baseline for the Block-Maxima results (Figure 7) shows that there is no task-set for which the technique provides optimistic results, as all the statistical results are above the baseline given by the theoretical analysis.

6.3 Applicability and Performance of the Normal and Weibull Distributions

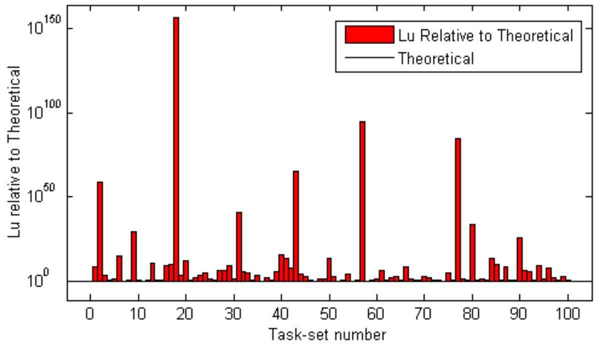

Figure 6: DMPs of 100 task-sets as derived by Block-Maxima relative to the DMPs computed theoretically

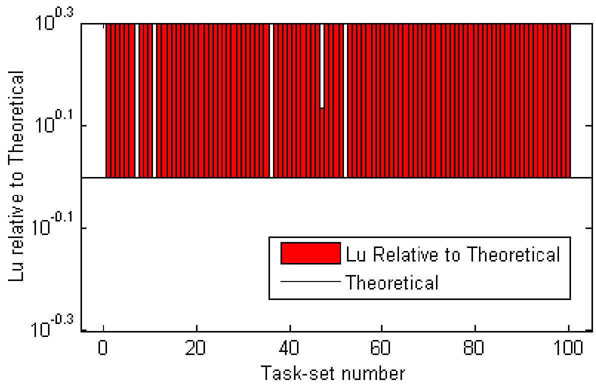

Figure 7: Close-up around the baseline of the relative DMPs derived with the Block-Maxima technique for 100 tasks

Figure 8 presents the DMPs derived using the Normal distribution compared to the exact DMPs. We see that the two curves are almost overlapping, indicating a good performance of the normal distribution for deriving response-time distributions for real-time tasks, but this is not necessarily the case. In Figure 9, presenting the ratios of the statistical DMPs relative to the exact DMPs, it is clear that for almost all the task-sets the Normal distribution provides optimistic results, unusable for validating real-time systems. In some cases this optimism leads to statistical DMPs of up to 50 orders of magnitude smaller than the exact DMP, potentially declaring as schedulable a task-set which is in fact unschedulable. For these reasons we consider that the Normal distributions is inadequate to be used in deriving response time distributions based on execution traces of real-time systems.

Similar to the Normal distribution, the Weibull distribution provides mostly optimistic results. This can be seen in Figure 10, where the statistical curve is below the theoretical one, and in Figure 11 where most relative DMPs are below the baseline. For these reasons we consider that the Weibull distribution should not be used in deriving response time distributions for real-time systems.

6.4 Comparison of the four Techniques

After seeing each of the statistical techniques separately, we can now take a look at all of them together and provide answers to the questions raised in Section 5.2.

1. *RQ1 - Do any of the statistical analysis approaches provide sound results?* We have seen that the Block-Maxima approach always provides safe results, though at the price of an increased amount of pessimism caused by the fact that only a subset of the larger response time traces are used in the analysis, shifting in this way the resulting distribution towards larger values. The Gumbel technique can also be considered to provide sound and safe results, even though at times it may be slightly optimistic, but this optimism can be countered with some post-processing correction, i.e., by introducing on purpose some extra pessimism as a measure of safety. It is evident that the Normal and Weibull distributions are not adequate for use in the statistical analysis of real-time

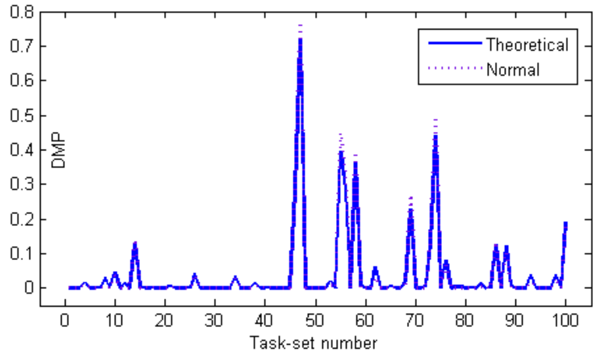

Figure 8: DMPs of 100 task-sets as derived using the Normal distribution and computed theoretically

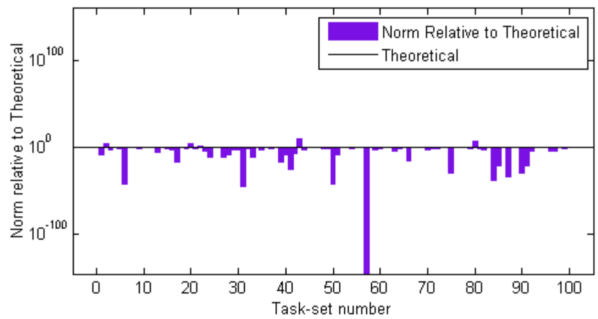

Figure 9: DMPs of 100 task-sets as derived using the Normal distribution relative to the DMPs computed theoretically

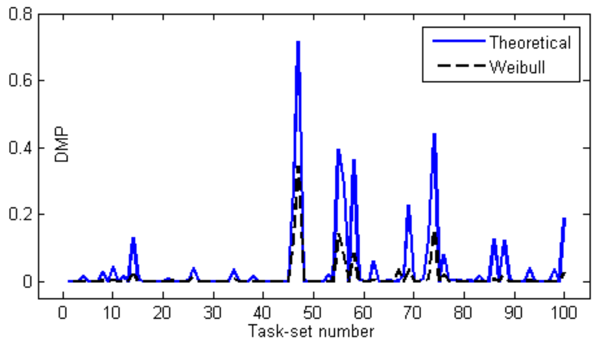

Figure 10: TDMPs of 100 task-sets as derived using the Weibull distribution and computed theoretically

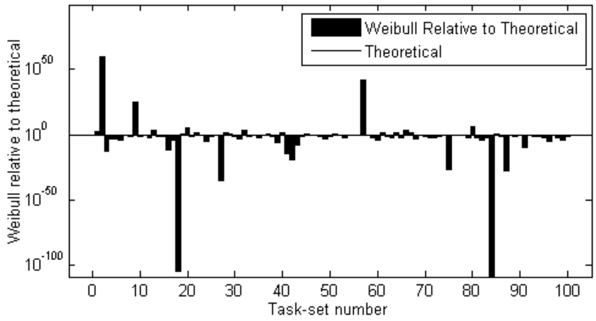

Figure 11: DMPs of 100 task-sets as derived using the Weibull distribution relative to the DMPs computed theoretically

systems.

2. *RQ2 - Which of the sound analysis approaches gives the tightest bound?* Out of the two approaches that can be considered sound (Gumbel and Block-Maxima), Gumbel appears to give the tightest results. However, as noted in the previous research question, the Gumbel distribution can be unsafe at times and would require post-processing correction. As such correction has not be considered in this evaluation, it is therefore not possible to definitively conclude that the Gumbel technique is superior to the Block-Maxima technique. Regarding the Block-Maxima technique, further research needs to be carried out to determine the level of pessimism that it introduces, and if this level of pessimism will result in a false positive rate (where it incorrectly classifies schedulable task sets as unschedulable) which is too high.

6.5 A closer look at a single task-set

In order to intuitively understand the presented results, we think that is is useful to take a look at the response time distributions of a single task-set. In Figure 12, a task-set of 5 tasks and 5 values per distribution is randomly generated and analysed (both probabilistically and statistically). Of the statistical distributions we present here only Gumbel and Block-Maxima since they are the relevant ones. Note that the Y-axis of the figure is logarithmic in order to emphasize the differences between the distributions. We can see that the Gumbel distributions does not always upper-bound the exact response time distribution, but instead it may intersect it for small values of response times, i.e. on the left side of the distribution. On the other hand, the Block-Maxima distribution is a strict upper-bound on the exact response time distribution. We can see the difference in accuracy for a deadline of 216 (the tasks minimum inter-arrival time), which increases even further for larger deadlines. A question that may be raised is if Block-Maxima could be made tighter by better choosing its parameters. It is not clear if there is an universal set of parameters which would make the technique tight for all possible task-sets, since the parameters may be system dependent, and different types of systems may require different settings. Of course this requires knowing additional details about the system under analysis. This is a further direction in which our investigation may advance and it is part of our future work.

7. CONCLUSIONS

We have addressed the problem of statistically deriving response time distributions and deadline miss probabilities of tasks, only by taking measurements of the systems behaviour (response time traces) at run-time. The system is treated like a black-box, meaning that no other information (such as arrival frequencies of execution times) are required in order to apply the statistical analysis. This is an important and practical problem, as it greatly reduces the analysis effort when designing a system or modifying it (for example adding new functionalities), since the validation of the system can be made by simply running it for an amount of time and passing the measured response times through the statistical analysis. We have investigated four statistical analysis techniques that we have compared against the state of the art probabilistic analysis which computes the exact worst case response time of the task-set. Our investigation revealed that the analysis technique based on Block-Maxima produces the safest results, though it may be too pessimistic to be effective in practice. Alternatively, the technique based on the Gumbel distribution produces much more accurate results even though it may be slightly optimistic at times (making it potentially still unsafe), but this optimism can be counter-balanced with post processing correction.

The results obtained are as expected based on the academic literature. Hence the framework developed provides valuable empirical evidence that is useful to industrial users who may wish to adopt the techniques and need evidence (e.g. as part of certification) that the techniques are fit for purpose, and for academic researchers who may also want to understand the correctness and efficiency of their techniques. That said, it is acknowledged the trials presented are limited and therefore more evaluation is needed to show the framework also provides valid insight for the context relevant to potential users.

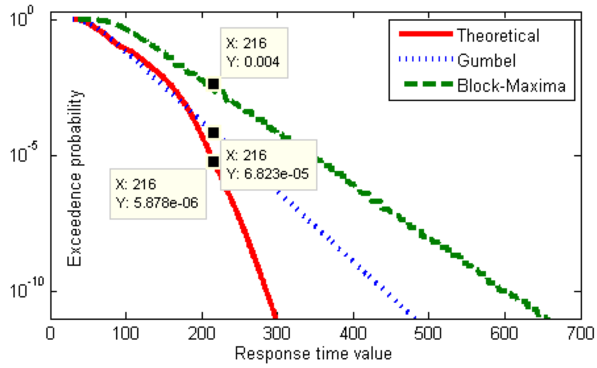

Figure 12: The theoretical, Gumbel and Block-Maxima response time distribution of a single task-set, as 1-CDF on a logarithmic Y-axis

Acknowledgments

This work was partially supported by FCT/MEC (Portuguese Foundation for Science and Technology), the EU ARTEMIS JU within projects ARTEMIS/0003/2012 - JU grant nr. 333053 (CONCERTO) and the EU FP7 Integrated Project PROXIMA (611085). The authors would also like to thank the anonymous reviewers and Dr Robert Davis (University of York) for their useful comments.

8. REFERENCES

[1] J. Abella, D. Hardy, I. Puaut, E. Quinones, and F. J. Cazorla. On the comparison of deterministic and probabilistic wcet estimation techniques. In *Real-Time Systems (ECRTS), 2014 26th Euromicro Conference on*, pages 266–275. IEEE, 2014.

[2] L. Abeni and G. Buttazzo. QoS guarantee using probabilistic deadlines. In *the 11th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 242–249, 1999.

[3] S. Altmeyer, L. Cucu-Grosjean, and R. I. Davis. Static probabilistic timing analysis for real-time systems using random replacement caches. *Real-Time Systems*, 51(1):77–123, 2015.

[4] S. Altmeyer, R. Davis, et al. On the correctness, optimality and precision of static probabilistic timing analysis. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pages 1–6. IEEE, 2014.

[5] ARINC653. An avionics standard for safe, partitioned systels. In *Wind River Systems/IEEE Seminar*, 2008.

[6] I. Bate, P. Conmy, T. Kelly, and J. McDermid. Use of modern processors in safety-critical applications. *The Computer Journal*, 44(6):531–543, 2001.

[7] I. Bate, P. Conmy, and J. A. McDermid. Generating evidence for certification of modern processors for use in safety-critical systems. In *Proceedings of the 5th International High Assurance Systems Engineering Symposium*, pages 125–134, 2000.

[8] G. Bernat, A. Colin, and S. Petters. WCET analysis of probabilistic hard real-time systems. In *Proceedings of the 23rd Real-Time Systems Symposium*, pages 279–288, 2002.

[9] G. Bernat, M. Newby, and A. Burns. Probabilistic timing analysis: an approach using copulas. *Journal of Embedded Computing*, 1(2):179–194, 2005.

[10] E. Bini and G. Buttazzo. Measuring the performance of schedulability tests. *Real-Time Systems Journal*, 30:129–154, 2005.

[11] I. Broster and A. Burns. Applying random arrival models to fixed priority analysis. In *the Proceedings of the Work-In-Progress of the 25th IEEE Real-Time Systems Symposium (RTSS04)*, 2004.

[12] I. Broster and A. Burns. Random arrivals in fixed priority analysis. In *1st International Workshop on Probabilistic Analysis Techniques for Real-time and Embedded Systems (PARTES2004)*, 2004.

[13] A. Burns and S. Edgar. Predicting computation time for advanced processor architectures. In *Proceedings 12th EUROMICRO Conference on Real-Time Systems*, pages 89–96, 2000.

[14] F. J. Cazorla, E. Quiñones, T. Vardanega, L. Cucu, B. Triquet, G. Bernat, E. Berger, J. Abella, F. Wartel, M. Houston, et al. Proartis: Probabilistically analyzable real-time systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 12(2s):94, 2013.

[15] L. Cucu and E. Tovar. A framework for response time analysis of fixed-priority tasks with stochastic inter-arrival times. *ACM SIGBED Review*, 3(1), 2006.

[16] L. Cucu-Grosjean. Independence - a misunderstood property of and for (probabilistic) real-time systems. March 14th, 2013.

[17] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Quinones, and F. Cazorla. Measurement-based probabilistic timing analysis for multi-path programs. In *Proceedings of the Euromicro Conference on Real-Time Systems*, pages 91–101, 2012.

[18] L. David and I. Puaut. Static determination of probabilistic execution times. In *Real-Time Systems, 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on*, pages 223–230, June 2004.

[19] R. Davis, L. Santinelli, S. Altmeyer, C. Maiza, L. Cucu-Grosjean, et al. Analysis of probabilistic cache related pre-emption delays. In *Real-Time Systems (ECRTS), 2013 25th Euromicro Conference on*, pages 168–179. IEEE, 2013.

[20] How to Determine a Sample Size. extension.psu.edu/evaluation/pdf/TS60.pdf.

[21] J. L. Díaz, D. F. García, K. Kim, C.-G. Lee, L. Lo Bello, J. M. López, S. L. Min, and O. Mirabella. Stochastic analysis of periodic real-time systems. In *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS 2002)*, page 289, 2002.

[22] M. Gardner and J. Lui. Analyzing stochastic fixed-priority real-time systems. In *5th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 1999.

[23] A. Gogonel and L. Cucu-Grosjean. How do we prove that probabilistic worst case response time is a gumbel? In *6th Real-Time Scheduling Open Problems Seminar (RTSOPS2015)*, 2015.

[24] P. Graydon and I. Bate. Realistic safety cases for the timing of systems. *The Computer Journal*, 57(5):759–774, 2014.

[25] J. Gustafsson, A. Betts, A. E., and B. Lisper. The mälardalen wcet benchmarks - past, present and future. In *Proceedings of the 10th International Workshop on Worst-Case Execution Time Analysis*, 2010.

[26] J. Hansen, S. Hissam, and G. Moreno. Statistical-based wcet estimation and validation. In *Proceedings of the 9th International Workshop on Worst-Case Execution Time Analysis*, pages 123–133, 2009.

[27] J. R. M. Hosking. Some theory and practical uses of trimmed L-moments. *Journal of Statistical Planning and Inference*, 137:3024–3039, 2007.

[28] S. Hutchesson and N. Hayes. Technology transfer and certification issues in safety critical real-time systems. In *Digest of the IEE Colloquium on Real-Time Systems*, number 98/306, April 1998.

[29] K. Kim, J. L. Diaz, L. Lo Bello, J. M. Lopez, C.-G. Lee, and S. L. Min. An exact stochastic analysis of priority-driven periodic real-time systems and its approximations. *IEEE Trans. Comput.*, 54(11):1460–1466, 2005.

[30] J. Lehoczky. Real-time queueing theory. In *Proceedings of the 10th IEEE Real-Time Systems Symposium*, pages 186–195, 1996.

[31] Y. Liang and T. Mitra. Cache modeling in probabilistic execution time analysis. In *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, pages 319–324, June 2008.

[32] J. Lopez, J. L. Diaz, J. E., and D. Garcia. Stochastic analysis of real-time systems under preemptive priority-driven scheduling. *Real-time Systems*, 40(2):180–207, 2008.

[33] Y. Lu, T. Nolte, I. Bate, and L. Cucu-Grosjean. A statistical response-time analysis of real-time embedded systems. In *Proceedings of the 33rd Real-Time Systems Symposium*, pages 351–362, 2012.

[34] D. Maxim and L. Cucu-Grosjean. Response time analysis for fixed-priority tasks with multiple probabilistic parameters. In *proceeding of the* 34^{th} *IEEE Real Time Systems Symposium, 2013. RTSS'2013.*, december 2013.

[35] D. Maxim, M. Houston, L. Santinelli, G. Bernat, R. I. Davis, and L. Cucu-Grosjean. Re-sampling for statistical timing analysis of real-time systems. In *Proceedings of the 20th International Conference on Real-Time and Network Systems*, pages 111–120. ACM, 2012.

[36] A. K.-L. Mok. *Fundamental Design Problems of Distributed Systems for the Hard-Real-Time Environment*. PhD thesis, Laboratory for Computer Science, Massachusetts Institute of Technology, 1983.

[37] T. Nolte, H. Hansson, and C. Norstrom. Probabilistic worst-case response-time analysis for the Controller Area Network. In *Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 200–207, 2003.

[38] L. Palopoli, D. Fontanelli, N. Manica, and L. Abeni. An analytical bound for probabilistic deadlines. In *Real-Time Systems (ECRTS), 2012 24th Euromicro Conference on*, pages 179–188, 2012.

[39] L. Phan, M. Xu, J. Lee, I. Lee, and O. Sokolsky. Overhead-aware compositional analysis of real-time systems. In *Proceedings of the Real-Time and Embedded Technology and Applications Symposium*, pages 237–246, 2013.

[40] K. S. Refaat and P.-E. Hladik. Efficient stochastic analysis of real-time systems via random sampling. In *IEEE Euromicro Conference on Real-Time Systems (ECRTS 2010)*, pages 175–183, 2010.

[41] T. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L. Wu, and J. Liu. Probabilistic performance guarantee for real-time tasks with varying computation times. In *IEEE Real-Time and Embedded Technology and Applications Symposium (ETFA 1995)*, 1995.

[42] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, and P. Stenström. The worst-case execution-time problem—overview of methods and survey of tools. *Transactions on Embedded Computing Systems*, 7(3):1–53, 2008.

[43] H. Zeng, M. D. Natale, P. Giusto, , and A. Sangiovanni-Vincentelli. Using statistical methods to compute the probability distribution of message response time in controller area network. *IEEE Transactions on Industrial Informatics*, 6(4):678–691, Nov. 2010.

[44] H. Zhu, J. Hansen, J. Lehoczky, and R. Rajkumar. Optimal partitioning for quantized edf scheduling. In *Proceedings of the Real-time Systems Symposium*, pages 202–213, 2002.