Technical Report
Systems and Automation Department (DAS)
Federal University of Santa Catarina (UFSC)
Florianopolis, Santa Catarina, Brazil

A Case Study on Probabilistic Estimation of the Worst-Case Execution Time of Consecutive Jobs

Rômulo Silva de Oliveira, Bruno Dourado Miranda,
Andreu Carminati, Luis Almeida

October 2022

Abstract

In this technical report we describe a case study on using Measurement-Based Probabilistic Timing Analysis in order to obtain estimates for the worst-case execution time of several consecutive jobs of the same task.

1 Introduction

The classic task model used for real-time schedulability analysis includes a parameter C that represents an upper bound on the task Worst-Case Execution Time (WCET). It is assumed that it can occur at every single task job.

There are works that extended this model by considering the cumulative execution time of several consecutive jobs of the same task [12] [16] [17]. These works defined functions that limit the total execution time of a sequence of jobs from each task.

One example is the workload curve $\gamma(k)$ that gives an upper bound on the execution time of any k consecutive jobs of a given task τ and which is tighter than $k \times C$. Thus, replacing $k \times C$ with $\gamma(k)$ in classic Response-Time Analysis (RTA) also reduces the resulting Worst-Case Response Time [12] [16].

Nevertheless, RTA seldom uses $\gamma(k)$ due to the difficulty of obtaining such values. The number of combinations of hardware state, application state and input data patterns across jobs explodes when one considers the execution of several jobs of a task.

In a work-in-progress paper [9], the authors proposed, for the first time, using Measurement-Based Probabilistic Timing Analysis (MBPTA) [7] based on Extreme Value Theory (EVT) [6] to obtain $\gamma(k)$ for $k > 1$, opening this path to lower pessimism in schedulability analysis. In this technical report we extend that preliminary work by presenting a concrete proof-of-concept example of how EVT-based MBPTA can be used to derive $\gamma(k)$.

MBPTA employs statistical tools to estimate probabilistic worst-case execution times (pWCET) associated to arbitrarily low exceedance probabilities, which are defined in accordance to the specification of the system, e.g., 10^{-8}. There are still several open questions regarding the applicability of EVT in the context of Real-Time Systems. In this technical report we do not address the specific issues of MBPTA representativeness and reliability.

The experiments show that using EVT to obtain $\gamma(k)$ is feasible in some scenarios. It is also possible to approximate $\gamma(k)$ for large values of k by adding several $\gamma(l), l < k$.

2 Extreme Value Theory

Measurement-Based Probabilistic Timing Analysis is an approach to generate probabilistic estimates of the Worst-Case Execution Time of a task. This is needed because of the limited observations that practical experiments allow. In the general case, it is not possible to guarantee that the WCET will be observed in a limited-size sample. MBPTA applies Extreme Value Theory to execution time measurements of a task running on its target environment. The distribution of execution times is adjusted to an asymptotic distribution of extreme values that will allow computing a pWCET with a corresponding probability of exceedance (see the surveys in [4] and [8]).

EVT is a framework of statistical inference with the objective of assessing the probability of rare events. It is mostly used to analyse natural phenomena, but also applied to financial and insurance markets. A basic premise for the applicability of EVT is that the observations must be produced by a random stationary process and be described as maximal independent and identically distributed (iid) random variables. The distribution of the measurements must also fall within the domain of attraction for one of the EVT distributions ([18], [4], [8]).

In this experiment we test the basic premise for the applicability of EVT to MBPTA and particularly to the estimation of worst-case sequences of execution times. We use the following techniques [1] [15] [19] [20]:

- **WW:** The Wald-Wolfowitz test of independence.
- **TP:** The Turning Point test of randomness.
- **LB2:** The Ljung-Box test of absence of correlation between observed values with a lag of 2.

- **LB10:** The Ljung-Box test of absence of correlation between observed values with a lag of 10.
- **KS:** The Kolmogorov-Smirnov test of identical distribution.
- **AD1:** The Anderson-Darling test of identical distribution, version 1 (adjusts for possibly different sample sizes).
- **AD2:** The Anderson-Darling test of identical distribution, version 2 (focuses on tail differences).

These tests are expected to produce so-called *p-values* uniformly distributed in $[0, 1)$. These values are usually presented in box and whiskers plots highlighting the 0%, 5%, 50%, 95% and 100% quantiles.

The application of EVT to MBPTA requires the following steps:

1. Measuring the execution time of several jobs.
2. Selecting a sample of values at the tail of the frequency distribution. We use the Block Maxima technique according to which we first break the full sample in blocks and then use the maximum value of each block, only.
3. Providing evidence that the method can be applied, i.e., measurements are maximal independent and identically distributed.
4. Fitting a Generalised Extreme Value (GEV) distribution to the maxima by estimating the parameters *Location*, *Scale* and *Shape*, e.g., the *Lmoments* method.
5. Checking the goodness of fit between the measurements and the fitted GEV, e.g., using quantile-quantile (Q-Q) plots.
6. Obtaining the execution time value with the desired exceedance probability (or vice-versa).

The quantile-quantile (Q-Q) plot is a graphical technique for determining whether two data sets come from populations with a common distribution. Quantiles are cut points dividing the observations in a sample (the execution-time measurements in our case) into intervals with equal probabilities. Quantiles can also divide the range of a probability distribution (the fitted GEV in our case) into continuous intervals also with equal probabilities. The quantiles of the measurements are plotted on the y-axis, against the quantiles of the fitted GEV on the x-axis. In our case the data sets can have the same size, so the Q-Q plot is essentially a plot of one sorted data set against the other sorted data set.

A 45-degree (1:1) reference line is also plotted. If the two sets come from populations with the same distribution, the Q-Q points should fall approximately along this reference line, on the line or randomly distributed around and close to it. The greater the departure from the 1:1 line, the greater the evidence that the two data sets have come from populations with different distributions.

We are mainly concerned with high values of the execution time, i.e, the fitting of the right tail of the probability function, which is described by the top right region of the Q-Q plot. It means that the sampled values are compatible with those expected when the fitted distribution properly models the observed data.

2.1 Workload Measurement and Auto-correlation

We are interested in measuring the cumulative execution time of any sequence of k consecutive jobs of task τ. Each specific sequence defines one measurement of $\gamma(k)$. To generate all possible measurements, we use the moving sum of k consecutive observed execution times over all jobs of τ in the sample.

However, using the moving sum, each single job of τ contributes to k different sequences, thus leading to a high auto-correlation in the $\gamma(k)$ measurements. The recurrent execution of task τ_i generates a sequence of execution times that can be accumulated in sub-sequences of k consecutive executions. Each of these sub-sequences defines one measurement of $\gamma_i(k)$.

This is specially true when using a moving sum of size k over all executions of τ_i. On the other hand, the moving sum does an exhaustive inspection of all possible sub-sequences of k consecutive jobs. An alternative way to obtain measurements of $\gamma(k)$ while avoiding such high auto-correlation is to use some arbitrary rule to split the whole sample of execution times of τ in non-overlapping sequences of size k. However, such partitioning can introduce bias in the measurements of $\gamma(k)$. For example, the splitting rule could destroy the sequences of k jobs that would generate longer cumulative execution times.

It is also possible to use random under-sampling, i.e., to randomly and uniformly choose a sub-set of the measurements of $\gamma(k)$ obtained with a moving sum. But, again, important samples could be discarded. There are other under-sampling methods used in machine learning, but they also can miss relevant measurements while focusing on reducing auto-correlation, a phenomenon that is actually present in the real world.

We also use the moving sum to generate all possible $\gamma(k)$ measurements, but we apply the Block Maxima approach that divides the whole sample in blocks and retains the maximum value of each block, only. By using an appropriate block size this method can strongly reduce auto-correlation without eliminating relevant measurements, allowing block maxima to be iid.

3 Proof-of-Concept Experiment

The increasing sofistication of Supervisory Control and Data Acquisition (SCADA) and automation systems in general in the era of Internet-of-Things and Industry 4.0 demands the use of interconnection and interoperability technologies such as XML encoding and SHA-256 cryptographic hash functions [5], [13], [22].

Therefore, we developed a proof-of-concept experiment based on the execution of those functions on the STMicroelectronics NUCLEO-F446RE [21]

Micro-Controller Unit (MCU), which is a commonly used platform for embedded systems. This MCU has a single core ARM Cortex-M4 microprocessor [14], which has floating point support, a 3-stage pipeline and a vector of 240 interruptions. The test frequency was fixed at 170MHz, and the maximum execution frequency supported by the microprocessor is 180MHz.

We used the Amazon FreeRTOS Real Time Operating System v.10.2.1 [3] [11] to perform the tests of a task running on the STMicrolectronics MCU. This OS is open source and written in the C language and assembly for more than 40 types of MCUs. Supporting queues, semaphores, event groups and software timers, it is one of the most popular real-time operating systems in the world.

The test task τ was implemented using the C language and it computes the SHA-256 hash for XML files of varying sizes. Initially a set of 20,000 files were randomly generated, using a normal distribution with mean 3000 bytes and standard deviation 500 bytes for the size of each file. During the measurements, at each activation of the task, one of the 20,000 files was selected following an uniform distribution. The time to compute the SHA-256 hash of that file was measured.

The task execution times were measured using the MCU's GPIO added to an 8-channel USB Logic Analyzer, with a resolution of 0.125 microseconds and using Amazon FreeRTOS kernel macros. The overall computer load does not vary during the experiment as it is usual with embedded systems.

We collected the execution time of 200,000 jobs (step 1, Section 2), which corresponds to measurements for $k = 1$. We then use a moving sum of size k to obtain measurements for k consecutive task executions. Actually, a little more than 200,000 individual jobs were measured so we could obtain 200,000 measurements for all considered values of k.

For each k, the corresponding set with 200,000 measurements defines the validation sample, used to test the reliability of the model fitting. We use a subset of 10,000 measurements as modeling sample, used to estimate $\gamma(k)$. Fig. 1 shows the distribution of the observed execution times of task τ in the modeling sample used to estimate $\gamma(1)$.

We used an exceedance probability of 10^{-8} to obtain an estimate of $\gamma(k)$ for each value of k. These estimates were obtained by fitting the measurements to a GEV distribution using *Lmoments* through the Block Maxima approach with block size of 100. Only the maximum of each block is used by the fitting method (step 2). We applied EVT using the R statistical software with the extRemes [10] package.

3.1 Applicability Tests

In order to test whether this experiment meets the basic premise of EVT we must test the measurements and check if their block maxima exhibit independence and identical distribution (step 3, Section 2).

Table 1 presents the *p-value* numerical result for each iid test described in Section 2. Those tests were applied to the block maxima considering different values of k. We considered the *p-values* yielded by the referred tests acceptable.

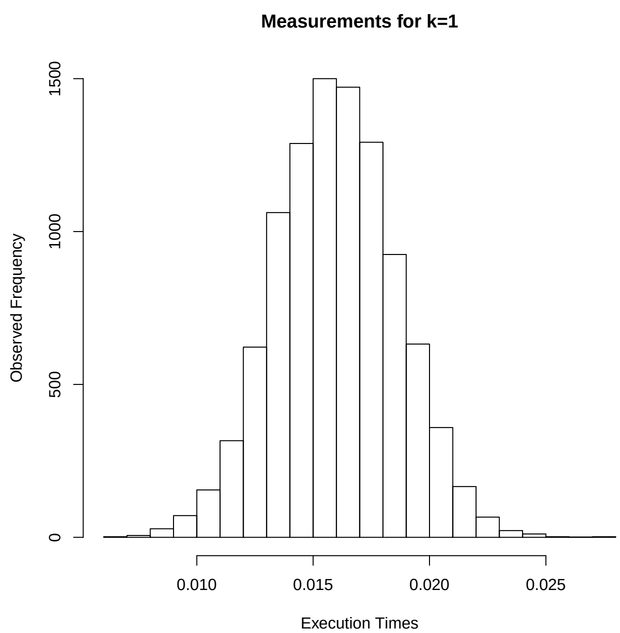

Figure 1: Observed values for $k = 1$.

But for measurements of larger values of k the auto-correlation increases and larger blocks may be necessary.

Table 1: IID p-values for each k.

k	**WW**	**TP**	**LB2**	**LB10**	**KS**	**AD1**	**AD2**
1	0.568	0.494	0.448	0.070	0.711	0.744	0.747
2	0.775	0.305	0.264	0.829	0.396	0.175	0.174
3	0.568	0.172	0.014	0.030	0.179	0.277	0.281
4	0.022	0.494	0.819	0.357	0.068	0.068	0.067
5	0.568	0.494	0.236	0.467	0.967	0.923	0.925
6	0.253	0.494	0.664	0.787	0.717	0.582	0.590
7	0.775	0.172	0.619	0.914	0.068	0.011	0.010
8	0.253	0.733	0.447	0.844	0.717	0.587	0.585
9	1.000	1.000	0.888	0.728	0.112	0.066	0.067
10	0.775	0.305	0.912	0.099	0.717	0.673	0.678
20	0.022	0.494	0.261	0.184	0.396	0.236	0.237
30	1.000	0.494	0.978	0.531	0.869	0.703	0.700
40	0.775	0.733	0.387	0.729	0.869	0.726	0.730
50	0.045	0.040	0.159	0.206	0.967	0.891	0.886

We fitted the block maxima to a GEV distribution (step 4, Section 2) and the resulting Q-Q plot (Fig. 2) was deemed acceptable (step 5). Good fitting in Q-Q plots is evidenced by the dots being disposed over, or randomly distributed around and close to, the 1:1 line. The discretization effect observed results from the fact that several jobs of the task process files of the same size, what results in very close or equal execution times.

We also fitted the measurements for $k = 10$ to a GEV distribution using *Lmoments*. The respective Q-Q plot in Fig. 3 shows adherence of the empirical data to the fitted distribution. Fig. 4 shows the same for $k = 50$. We consider the GEV model applicable in this case and proceed to compute $\gamma(k)$ for this experiment.

We applied the reliability test described in [2] to the pWCET values obtained in this work. First we used 10,000 measurements to make the fitting of a GEV function for each value of k considered. Since we have a validation sample of 200,000 measurements, we generated the value of $pWCET(\epsilon)$ for each value of k, where $\epsilon = 1/200{,}000$ represents the exceedance probability.

The number of exceedances in the validation sample were counted for each value of k. Finally, we computed the *p-value* of the reliability test as described in [2], i.e., the probability of at least the observed number of exceedances being observed purely by chance, in a sample of 200,000 measurements, under the assumption that the exceedance probability is 1/200,000.

Table 2 presents, for each value of k considered, the maximum value (HWM) observed in the validation sample, the $pWCET(1/200{,}000)$, the number of ex-

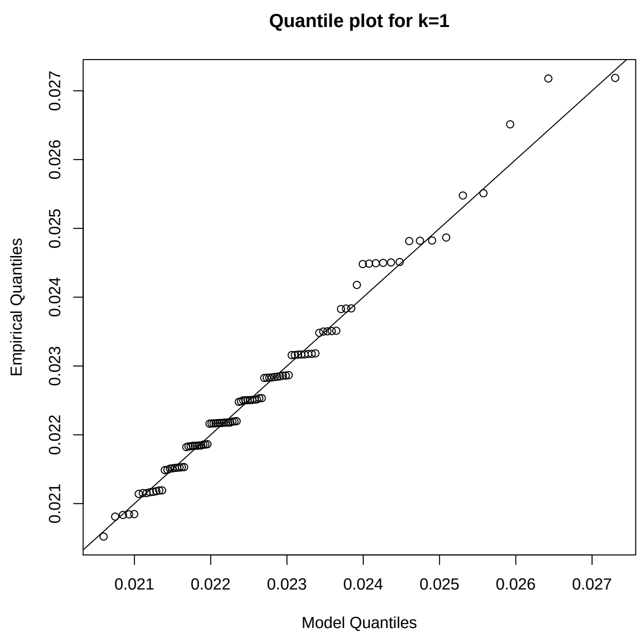

Figure 2: Quantile (Q-Q) plot for $k = 1$.

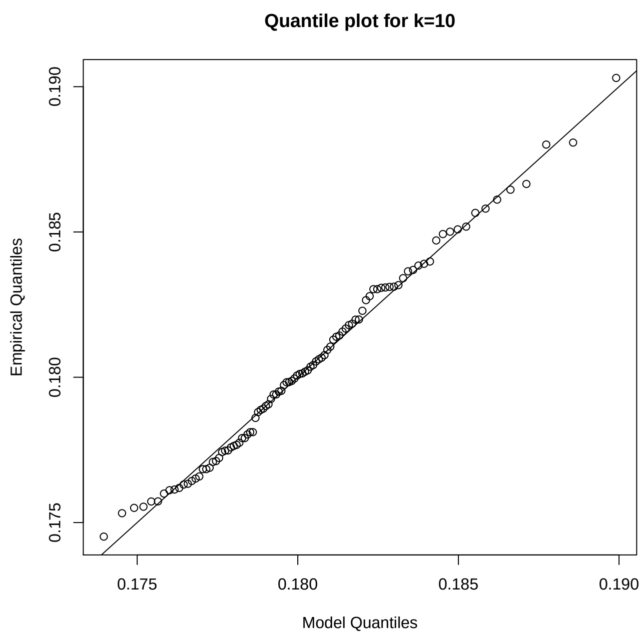

Figure 3: Quantile (Q-Q) plot for $k = 10$.

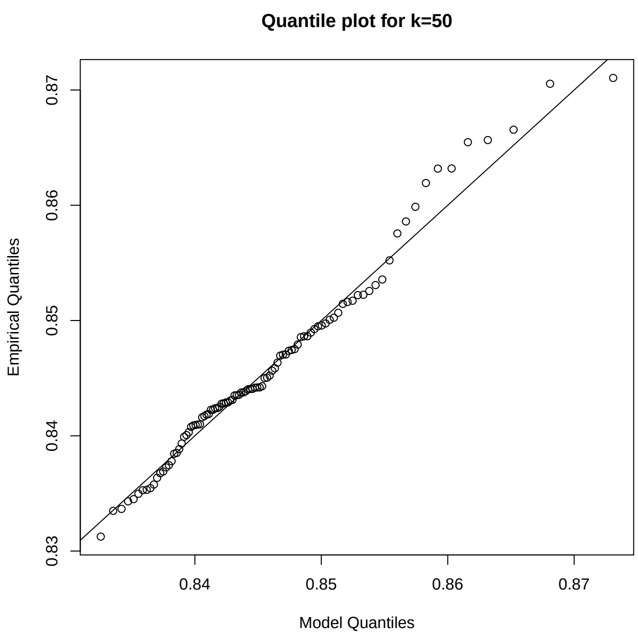

Figure 4: Quantile (Q-Q) plot for $k = 50$.

ceedances in the validation sample, and the *p-value* of the reliability test. We observed in our experiments the same variability previously described in [2], so several fittings were tried, using different sets of 10,000 measurements, until satisfactory reliability was achieved.

Table 2: Reliability test with a validation sample of 200,000 measurements.

k	$HWM(k)$	$\gamma(k)$	$Exceeded(k)$	p-$value(k)$
1	0.027	0.039	0	1
2	0.049	0.049	0	1
3	0.069	0.082	0	1
4	0.088	0.094	0	1
5	0.106	0.115	0	1
6	0.124	0.130	0	1
7	0.143	0.151	0	1
8	0.160	0.164	0	1
9	0.178	0.181	0	1
10	0.196	0.200	0	1
20	0.367	0.378	0	1
30	0.545	0.601	0	1
40	0.722	0.736	0	1
50	0.890	0.935	0	1

4 Results

After having determined the applicability of EVT, we used an exceedance probability of 10^{-8} to obtain an estimate for each value of $\gamma(k)$ (step 6, Section 2). Table 3 shows, for some values of k, the estimated value for $\gamma(k)$, the commonly used $k \times \gamma(1)$ and the Gamma Factor function $GF(k)$. For this scenario, overall, we have an observed average execution time $\mu = 0.07993$ and a maximum observed execution time $\gamma(1) = 0.09946$, which results in a Gamma Factor $GF = 0.8036$.

Since all values of $\gamma(k)$ are statistically estimated, they are always approximations. Notwithstanding, the values presented in Table 3 respect the property in (1).

$$k = \sum_{l=1}^{m} k_l \implies \gamma_i(k) \leq \sum_{l=1}^{m} \gamma_i(k_l) \leq k \times \gamma_i(1) \tag{1}$$

For instance, we can use $\gamma(20)+\gamma(30)$ instead of $\gamma(50)$, although the resulting $GF(50)$ would be smaller, i.e., the advantage of using $\gamma(k)$ in RTA would be lesser.

Table 3: Results for several values of k and an exceedance probability of 10^{-8} ($\gamma(k)$ values in seconds).

k	$\mu(k)$	$HWM(k)$	$\gamma(k)$	$k \times \gamma(1)$	$GF(k)$
1	0.016	0.027	0.053	0.053	1.000
2	0.032	0.045	0.051	0.107	0.477
3	0.048	0.067	0.097	0.160	0.604
4	0.064	0.084	0.100	0.213	0.470
5	0.080	0.104	0.123	0.267	0.461
6	0.096	0.121	0.136	0.320	0.424
7	0.112	0.136	0.160	0.373	0.429
8	0.128	0.158	0.167	0.426	0.391
9	0.144	0.174	0.184	0.480	0.383
10	0.160	0.190	0.203	0.533	0.382
20	0.320	0.363	0.386	1.066	0.362
30	0.480	0.540	0.672	1.599	0.420
40	0.640	0.702	0.762	2.132	0.357
50	0.800	0.871	0.997	2.665	0.374

Table 4 shows, for some values of k, the parameters *Location*, *Scale* and *Shape* of the GEV distribution used to estimate $\gamma(k)$, together with the Gamma Factor $GF(k)$. We can observe that *Shape* varies in a way apparently uncorrelated with k as opposed to the other parameters.

Overall, this proof-of-concept experiment showed that it is feasible, at least in some scenarios, to use MBPTA based on EVT to obtain estimates of $\gamma(k)$. It included a typical micro-controller platform and a well known real-time operating system. The Block Maxima approach successfully removed the natural correlation that appears when observing sequences of k consecutive jobs.

In the particular scenario of this experiment, $\gamma(10)$ is about 5% less than the classic value $C \times 10$. And $\gamma(20)$ is about 10% less than $C \times 20$. Considering these values in a Response-Time Analysis, the difference may be enough to find schedulable a system that would be considered unschedulable otherwise.

5 Conclusion

This technical report described a case study on using MBPTA based on EVT to determine the worst-case cumulative execution time of sequences of k consecutive jobs, namely $\gamma(k)$, as previously proposed in [9].

The relevance of this work stems from the observation that using EVT might be the only practical way to derive $\gamma(k)$. Then, we showed how to check EVT applicability and how to apply it, particularly with the Block Maxima approach that eliminates the auto-correlation naturally present in the execution time measurements of job sequences. A concrete proof-of-concept example related to a

Table 4: GEV Parameters for each k.

k	**Location**	**Scale**	**Shape**	$GF(k)$
1	0.022	0.001	0.053	1.000
2	0.040	0.001	−0.096	0.477
3	0.058	0.002	0.023	0.604
4	0.076	0.002	−0.037	0.470
5	0.093	0.002	−0.036	0.461
6	0.111	0.002	−0.064	0.424
7	0.128	0.002	−0.026	0.429
8	0.145	0.003	−0.125	0.391
9	0.162	0.003	−0.137	0.383
10	0.179	0.003	−0.105	0.382
20	0.347	0.004	−0.071	0.362
30	0.512	0.005	0.050	0.420
40	0.677	0.005	−0.015	0.357
50	0.842	0.006	0.030	0.374

commonly used function executing on a typical MCU and RTOS platform confirmed the EVT applicability and the potential of the method to tighten the interference a task may cause in lower priority tasks.

In this technical report we did not address several open problems regarding MBPTA based on EVT. For instance, the problem of representativity remains open.

References

[1] J. Abella, E. Quiñones, F. Wartel, T. Vardanega, and F. J. Cazorla. Heart of gold: Making the improbable happen to increase confidence in mbpta. In *2014 26th Euromicro Conference on Real-Time Systems*, pages 255–265, July 2014. doi: `10.1109/ECRTS.2014.33`.

[2] L. F. Arcaro, K. P. Silva, R. S. de Oliveira, and L. Almeida. Reliability test based on a binomial experiment for probabilistic worst-case execution times. In *2020 IEEE Real-Time Systems Symposium (RTSS)*, pages 1–12, Dec 2020.

[3] R. Barry. Mastering the freertos real time kernel, 2019. https://www.freertos.org.

[4] F. J. Cazorla, L. Kosmidis, E. Mezzetti, C. Hernandez, J. Abella, and T. Vardanega. Probabilistic worst-case timing analysis: Taxonomy and comprehensive survey. *ACM Comput. Surv.*, 52(1), February 2019. doi: `10.1145/3301283`.

[5] C.-M. Chituc. Xml interoperability standards for seamless communication: An analysis of industry-neutral and domain-specific initiatives. *Computers in Industry*, 92-93:118 – 136, 2017. doi: 10.1016/j.compind.2017.06.010.

[6] S. Coles. *An introduction to statistical modeling of extreme values*. Springer Series in Statistics. Springer-Verlag, London, 2001.

[7] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Quiñones, and F. J. Cazorla. Measurement-based probabilistic timing analysis for multi-path programs. In *2012 24th Euromicro Conference on Real-Time Systems*, pages 91–101, July 2012. doi: 10.1109/ECRTS.2012.31.

[8] R. Davis and L. Cucu-Grosjean. A survey of probabilistic timing analysis techniques for real-time systems. *Leibniz Transactions on Embedded Systems*, 6(1):60, 2019. doi: 10.4230/LITES-v006-i001-a003.

[9] R. de Oliveira and L. Almeida. Measurement-based estimation of the worst-case execution time of consecutive jobs. In *IEEE Real-Time and Embedded Technology and Applications Symposium (Work-in-Progress)*, 2020.

[10] E. Gilleland and R. Katz. extremes 2.0: An extreme value analysis package in r. *Journal of Statistical Software, Articles*, 72(8):1–39, 2016. doi: 10.18637/jss.v072.i08.

[11] F. Guan, L. Peng, L. Perneel, and M. Timmerman. Open source freertos as a case study in real-time operating system evolution. *J. Syst. Softw.*, 118(C):19–35, August 2016. doi: 10.1016/j.jss.2016.04.063.

[12] M. Jersak, R. Henia, and R. Ernst. Context-aware performance analysis for efficient embedded system design. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, volume 2, pages 1046–1051 Vol.2, Feb 2004. doi: 10.1109/DATE.2004.1269031.

[13] R. Kyusakov, H. Mäkitaavola, J. Delsing, and J. Eliasson. Efficient xml interchange in factory automation systems. In *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pages 4478–4483, 2011. doi: 10.1109/IECON.2011.6120046.

[14] Arm Limited. Cortex-m4, 2020.

[15] G. M. Ljung and G. E. P. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978. doi: 10.1093/biomet/65.2.297.

[16] A. Maxiaguine, S. Kunzli, and L. Thiele. Workload characterization model for tasks with variable execution demand. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, volume 2, pages 1040–1045 Vol.2, Feb 2004. doi: 10.1109/DATE.2004.1269030.

[17] S. Quinton, M. Hanke, and R. Ernst. Formal analysis of sporadic overload in real-time systems. In *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 515–520, March 2012. doi: `10.1109/DATE.2012.6176523`.

[18] L. Santinelli, F. Guet, and J. Morio. Revising measurement-based probabilistic timing analysis. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 199–208, April 2017. doi: `10.1109/RTAS.2017.16`.

[19] L. Santinelli, J. Morio, G. Dufour, and D. Jacquemart. On the sustainability of the extreme value theory for wcet estimation. In *WCET*, volume 39, 07 2014. doi: `10.4230/OASIcs.WCET.2014.21`.

[20] F. W. Scholz and M. A. Stephens. K-sample anderson-darling tests. *Journal of the American Statistical Association*, 82(399):918–924, 1987. doi: `http://links.jstor.org/sici?sici=0162-1459%28198709%2982%3A399%3C918%3AKAT%3E2.0.CO%3B2-9`.

[21] STMicroelectronics. Stmicroelectronics nucleo-f446re, 2019.

[22] A. Tidrea, A. Korodi, and I. Silea. Cryptographic considerations for automation and scada systems using trusted platform modules. *Sensors*, 19(19), 2019. doi: `10.3390/s19194191`.