Fault-Aware Sensitivity Analysis for Probabilistic Real-Time Systems

Luca Santinelli, Zhishan Guo, Laurent George

Fault-Aware Sensitivity Analysis for Probabilistic Real-Time Systems

Luca Santinelli[1], Zhishan Guo[2] and Laurent George[3]
[1]UFTMIP ONERA Toulouse, luca.santinelli@onera.fr
[2]Missouri University of Science and Technology, zsguo@cs.unc.edu
[3]University of Paris Est / LIGM - ESIEE Paris, laurent.george@esiee.fr

***Abstract*—In probabilistic real-time modeling, diverse task execution conditions can be characterized with probabilistic distributions, where multiple execution time thresholds are represented, each with an exceeding probability. Comparing to traditional deterministic real-time, probabilistic approaches provide more flexibility in system behavior modeling, which may result in more precise schedulability analysis. With this work, we combine sensitivity analysis and probabilistic models of fault effects on task execution behaviors. The goal is to develop probabilistic schedulability analysis that is applicable to both faulty and non-faulty execution conditions. While the probabilities accurately characterize faults and faults effects on worst-case execution times, the probabilistic schedulability analysis both qualifies and quantifies faults impacts on system schedulability.**

I. Introduction

Since last decade, real-time systems are mostly implemented with multi-core and many-core commercial-off-the-shelf architectures. The numerous interferences within such complex systems may end up into large variability of the execution time of tasks. For example, the state of the cache or the availability of a shared resource at runtime could largely change the time a task takes to execute.

The worst-case execution time (WCET) of a task represents an upper bound to the time needed to finish execution. In presence of faults, the execution time increases from the non-faulty conditions, due to recovery and restore procedures that tasks subdue to. Then, in order to be an upper-bound, the WCET has to account for any condition, including the highly improbable pathological cases such as faults. This could lead to extremely pessimistic tasks worst-case execution models.

Probabilistic worst case analysis associates to WCETs the probabilities of happening of execution conditions, including those with vanishingly small probabilities. This may lead to important reduction of the computational resource over-provisioning.

First papers on probabilistic approaches describe tasks worst-case execution times by random variables, either discrete [1] or continuous [2] distributions. Since Edgar and Burns [3], several papers have worked on obtaining safe and reliable probabilistic Worst-Case Execution Time (pWCET) estimates [4], [5], [6]. Unfortunately, pWCETs have been rarely applied for characterizing faults and faults impact on task executions.

The probabilistic worst-case abstraction applies into schedulability analysis with probabilities. Todays probabilistic schedulability relies on many restrictive assumptions in order to be applied. Examples are the independence constraint which is imposed, [7], [8] or the consideration of only the worst-case interactions between tasks [9], [10].

Finally, the sensitivity analysis applied to schedulability studies how parameters/input uncertainty affects system schedulability, [11], [12]. In its probabilistic version, the sensitivity analysis quantifies the impact of the different possible probabilistic models on system schedulability [13].

Fault management enlaces with both timing and schedulability analysis. For example, in [14] backups are executed for fault tolerance and recovering form task errors caused by hardware or software faults. In [15] instead, an algorithm to abort a restart a task in case of conflicts in shared resources accesses is proposed. Actual faults models are not included neither in the probabilistic timing analysis nor in the probabilistic schedulability analysis.

Contribution: In this work we investigate the impact of faults on the schedulability of real-time systems. We formalize the scheduling problem with probabilistic task models embedding both faulty and non-faulty execution conditions. Furthermore, we make use of newly developed C-Space+ for representing and studying the probabilistic fault models. The pWCETs, obtained in presence and absence of faults, are used together with the sensitivity analysis and C-Space+ for quantifying faults effects on the system schedulability.

Organization of the paper: In Section II we outline some backgrounding notions for probabilistic real-time. Section III details task modeling with probabilities and multiple execution scenarios. Section IV is for defining the probabilistic task modeling with faults effects, while Section V describes the C-Space+ and the possible sensitivity analysis on it. Finally, Section VI is for conclusions and future work.

II. Background: Real-Time with Probabilities

A real-time task consists of an infinite sequence of recurring jobs; each job must be executed before a given deadline.

In a periodic task system, a task τ_i is described with the tuple (O_i, T_i, D_i, C_i), where O_i is the offset or starting time that specifies the time instant at which the first job of τ_i is released; T_i is the period representing the temporal separation between two successive jobs; D_i is the deadline which defines the time interval in which task execution has to terminate; and C_i is the worst-case execution time defining the processing requirements for each job. The scheduling policy decides the task execution ordering, possibly with preemption or migration, while schedulability analysis guarantees the respect of the timing constraints.

Due to interferences from the system elements and the environment, tasks can exhibit multiple durations at runtime. It is then reasonable to describe task execution time as random

processes[1].

The probabilistic worst-case execution time $\mathcal{C}_i$ of a task τ_i generalizes the deterministic WCET. It is defined as the worst-case distribution that upper-bounds any possible execution time the task can exhibit. In its abstract interpretation, $\mathcal{C}_i$ would includes multiple values, each with the probability of being the worst-case execution time of the task[2].

Assuming the pWCETs as discrete distributions, the probability distribution function (pdf) representation of $\mathcal{C}_i$ is:

$$\mathsf{pdf}_{\mathcal{C}_i} = \begin{pmatrix} C_{i,1} & \dots & C_{i,k_i} \\ P(\mathcal{C}_i = C_{i,1}) & \dots & P(\mathcal{C}_i = C_{i,k_i}) \end{pmatrix}, \quad (1)$$

with $\mathsf{pdf}_{\mathcal{C}_i}(C_{i,r}) = P(\mathcal{C}_i = C_{i,r})$, $\sum_{r=1}^{k_i} \mathsf{pdf}_{\mathcal{C}_i}(C_{i,r}) = 1$, and k_i is the number of elements composing the pWCET distribution. $P(\mathcal{C}_i \leq C_i) = 1$, and for all k $1 \leq k \leq k_i$, $C_{i,k} \leq C_i$.

$\mathsf{cdf}_{\mathcal{C}_i}$ denotes the cumulative distribution function (cdf) representation of $\mathcal{C}_i$, $\mathsf{cdf}_{\mathcal{C}_i}(C) = P(\mathcal{C}_i \leq C) = \sum_{x=1}^{C} \mathsf{pdf}_{\mathcal{C}_i}(x)$ with discrete distributions. The inverse cumulative distribution function (icdf) $\mathsf{icdf}_{\mathcal{C}_i}(C)$ outlines the exceedence thresholds. $\mathsf{icdf}_{\mathcal{C}_i}(C) = P(\mathcal{C}_i \geq C)$ is the probability of having execution time larger than C, $\mathsf{icdf}_{\mathcal{C}_i}(C) = 1 - \sum_{x=1}^{C} \mathsf{pdf}_{\mathcal{C}_i}(x)$.

a) WCET Thresholds: From $\mathcal{C}_i$ it is possible to define *WCET thresholds* $\langle C_{i,j}, p_{i,j} \rangle$ such that the value $C_{i,j}$ is associated to the probability $p_{i,j}$ of being the WCET for τ_i. $p_{i,j} \stackrel{def}{=} \mathsf{icdf}_{\mathcal{C}_i}(C_{i,j})$ quantifies the *confidence* on $C_{i,j}$ of being the task worst-case execution time; $1 - p_{i,j}$ is the probability of respecting $C_{i,j}$ and $p_{i,j}$ is the probability of passing it. WCET thresholds are derived from pWCET estimates and are complementary representations to the worst-case. They are single value WCET with associated a probability.

With finite support pWCETs, it exist a WCET threshold C_i^* such that $\mathsf{cdf}_{\mathcal{C}_i}(C_i^*) = 1$. C_i^* would be the deterministic WCET estimation upper-bounding any WCET threshold.

Recent works have investigated how to derive continuous pWCETs estimates from execution time measurements in different system conditions, [5]. Discrete and finite support pWCETs can always be derived as approximations, at relatively low probabilities, of such continuous pWCET estimates [16].

b) Worst-Case Independence: Assuming $\mathcal{C}_i$ to be the probabilistic worst-case distribution of τ_i, it means that in $\mathcal{C}_i$ there have already been included all the possible interferences and their effects [17]. This is from the definition of statistical independences, $\mathsf{cdf}_{\mathcal{C}_i,\mathcal{C}_j} = \mathsf{cdf}_{\mathcal{C}_i} \cdot \mathsf{cdf}_{\mathcal{C}_j}$, which creates independence between tasks. *By assuming the pWCET $\mathcal{C}_i$, tasks become independent* at the cost of having embedded dependence effects; thus more pessimistic worst-case models.

In this work we consider a set on n periodic probabilistic tasks $\Gamma = \{\tau_1, \dots, \tau_n\}$, with $\tau_i = (O_i, T_i, D_i, \mathcal{C}_i)$. The hyperperiod H is the last common multiple of all the periods, $H = lcm(T_1, \dots, T_n)$. Γ is scheduled according to the preemptive Earliest Deadline First (EDF) paradigm, [18].

III. Probabilistic Worst-Case Execution Time Modeling

A Measurement Based Probabilistic Timing Analysis (MBPTA) approach applies the Extreme Value Theory (EVT) for computing pWCET estimates out of measured execution times, [6]. Figure 1 describes the general idea of MBPTA: i) measurements of task execution times are taken from a system configuration and specific execution conditions; ii) the trace of measurements is transformed into pWCET estimates with the extreme value theory (red line distribution). The EVT guarantees that if certain hypotheses are verified, from the actual measured execution behavior it is possible to infer rare events, where the worst-case execution time should lie. The EVT produces a continuous distribution which is a safe estimation of the worst-case behavior of the task: the pWCET estimate $\mathcal{C}_i$. By safe estimation we mean a *pWCET which is larger than or equal to any possible task execution time*.

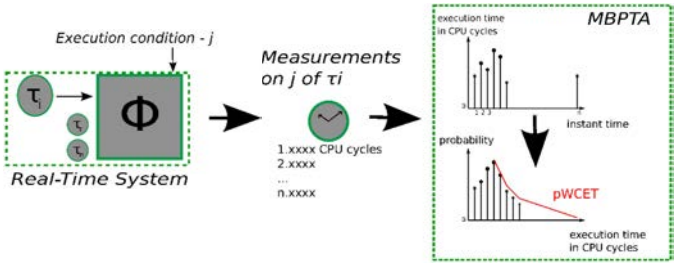


Fig. 1. Measurement-based probabilistic timing analysis toolchain. From the measurements the pWCET is inferred with the extreme value theory.

In the rest of the paper we apply both $\mathcal{C}_i$ and C_i, the WCET threshold $\langle C_i, 10^{-9} \rangle$, for modeling task behaviors and formalizing the sensitivity analysis. We consider $\mathcal{C}_i$ and C_i from different task execution conditions.

A trace of measurements accounts for some of the interfering conditions and the inputs which happens at runtime. The resulting pWCET estimate $\mathcal{C}_i$ embeds those system conditions and others which have not been measured. Therefore, the EVT is able to infer some of the unknown from the known measurements.

A. Worst-Case Guarantees

The pWCET from the MBPTA could be considered the worst-case for only the execution conditions measured - pWCET relative to the conditions measured. But what if the execution scenario/inputs heavily changes e.g., operational mode change, different environment conditions or faults appears, and they have not been included into the measurements? In that case, the EVT would not be able to correctly estimate absolute rare events anymore.

Given a scenario j among the set of possible system execution scenarios[3] J, the worst-case profile for j is $\mathcal{C}_i^j$ and it comes from the measurements taken under condition j. The way for guaranteeing safe pWCET estimates $\mathcal{C}_i$ for J is to account for all the scenarios within J.

c) Worst-Case Profiling: The worst-case estimate $\mathcal{C}_i$ could be defined as an envelope among the possible probabilistic profiles: $\mathcal{C}_i \stackrel{def}{=} max_{j \in J}\{\mathcal{C}_i^j\}$. With the 1-CDFs it is:

$$\mathsf{icdf}_{\mathcal{C}_i}(C) \stackrel{def}{=} max_{j \in J}\{\mathsf{icdf}_{\mathcal{M}_i^j}(C)\}. \quad (2)$$

d) Worst-Case Sets: Alternative to Equation (2), it is the solution where the scenarios are kept separated. All of them are needed but they are not merged in a single model. It is:

$$\overline{\mathcal{C}}_i \stackrel{def}{=} (\mathcal{C}_i^1, \mathcal{C}_i^2, \dots, \mathcal{C}_i^{|J|}) \quad (3)$$

[1]A random process is a sequence of random variables describing a process whose outcomes do not follow a deterministic pattern, but follow probability distributions.

[2]Calligraphic letters are used to represent distributions while non calligraphic letters are for scalars.

[3]The scenario is an abstract concept which could embed system inputs, system configurations, etc.. Defined a scenario, the measurements under it are the execution times of the effects enclosed by the scenario.

the set of pWCET estimates, and

$$\overline{C}_i \stackrel{def}{=} (C_i^1, C_i^2, \ldots, C_i^{|J|}) \quad (4)$$

the set of WCET thresholds such that $\langle C_i^j, 10^{-9} \rangle \ \forall j \in J$.

The worst-case sets representation i.e. Equation (3) and Equation (4) is what we apply in this work. Figure 2 gives an example of worst-case representations for a task multi-scenario. In particular there are illustrated two scenarios with the histogram, the pdf and the icdf representations of the measurements.

IV. Faults Effects on Task Execution Time

Real-time systems are prone to faults, either transient or permanent faults, which could affect task executions [19]. For example, a cache fault could end up in a temporal or permanent impossibility of using a cache block [20]; this increases the task execution time due to the larger latencies for retrieving the information in memory.

The abort and restart model [15] can be used to model the occurrence of a fault i.e. to restart the faulty task, thus increasing its WCET. There exist other fault models with different impacts on the task execution time. In this work we could exploit them all by investigating the resulting task execution times.

It would be possible to measure execution times under faulty conditions i.e. *fault* scenario. The *fault* scenario assumes that the fault happens at each task instance. The measurements would account for the penalties on the task execution time due to faults, such as the delay due to the restoring of the system status and the re-execution of the task. Traces of measurements under *fault* would have execution times, each impacted by the constant presence of faults. $\mathcal{C}_i^{fault}$ would be the pWCET for that scenario.

Instead, by considering non-faulty conditions i.e. fault never happening - *no-fault* scenario, it is possible to have the pWCET $\mathcal{C}_i^{no-fault}$ specific to that scenario. Here, the execution times observed exploit only the task functional behavior. The trace of measurements would have execution times due to the absence of faults. A collection of execution times measurements is a trace of measurements.

Intuitively, the fault scenario is the worst-case scenario (among the two considered) since the penalties/largest execution times are measured from it. Thus $\mathcal{C}_i^{fault}$ could be a more pessimistic safe pWCET for both *no-fault* and *fault*. Nonetheless, it would happen that small execution times values in the *no-fault* scenario have larger probabilities than the *fault* scenario. In order to have a complete task execution time model, both *no-fault* and *fault* scenarios have to be collected.

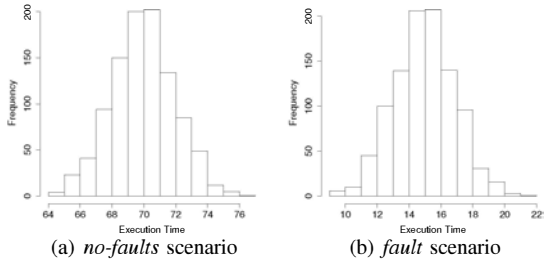
In this paper we consider faults and faults impact abstractly modeled through traces of execution times. This allows us to develop a modeling and analysis framework general enough to approach any possible fault. The examples presented are abstract case studies that we apply to validate the framework.

Example 1. *We consider a task with two execution conditions i.e. two scenarios happening:* no-faults *and* fault.
In case of no-faults *the task executes in its normal way, no faults happening. The execution time variability is given by functional and systemic effects from the normal task behavior.* no-faults *is the baseline for the other cases.*
In case of fault *the task executes always under the most critical fault condition.*

The impacts of the 2 scenarios on the task execution times are represented in both Figure 2 and Figure 3(a). The execution times differ according to the conditions experienced, and the differences quantifies the fault effects. Figure 3(b) depicts the pWCET resulting from the measured execution time. The 2 pWCETs reflect the faults impact on the task worst-case behavior.

In this example we assume that the measurements of execution times follow a Gaussian distribution in both scenarios. In particular, with the mean $\mu = 15$ and the standard deviation $\sigma = 2$ for the no-fault*, and with $\mu = 70$ and $\sigma = 2$ for the* fault *scenario. The* fault *scenario is more than doubling the* no-faults *condition. This could happen due to fault models where the task is asked to re-execute, or a recovery procedure is called for restoring the task normal behavior. In this example the faults affects task executions on average with an increase of 4.6 times. This is an abstract example where we chose to emphasize fault effects, even though in real cases faults could have a smaller impact on the average task execution.*

The different distributions are illustrated in Figure 2, and the execution times are expressed in CPU ticks.



(a) *no-faults* scenario (b) *fault* scenario

Fig. 2. Different execution scenarios in the histogram representation. Empiric distributions from 1000 measurements following Gaussian law.

Figure 2 illustrates the two scenarios; the two traces of execution times are depicted with the histogram, pdf and icdf representations.

A. Task Modeling with Faults

Having different execution scenarios directly affects the safety of pWCET estimations, intended as the capability of upper-bounding the assumed scenarios.

Supposing to have the set of possible scenarios $J = \{no - fault, fault\}$, by considering only the *no-fault* scenario it is not safe enough for characterizing the task worst-case behavior among J. The resulting pWCET estimation $\mathcal{C}_i^{no-fault}$ cannot be assumed as safe worst-case estimation for the task because the fault effects are not included.

The *fault* scenario, by embedding the faults effects in the measurements (fault penalties that increases the execution times) is an upper-bound to non-faulty conditions also. Hence, the pWCET $\mathcal{C}_i^{fault}$ could be considered a safe estimation of the worst-case task behavior resulting from the worst-case execution conditions within J.

We stress the fact that the worst-case is intended as the worst scenario among the possible known conditions J. *The conditions are modeled by measurements which makes the pWCET ony relative to the conditions accounted.* Instead, the non considered scenarios $j \notin J$, cannot be modeled by any MBPTA approach, thus *it is not possible to provide an absolute guarantee on the worst-case.* To note that $fault$ is an extreme scenario, since the fault is always active in it e.g., the task is always forced to abort and restart. We make use of it to enforce the task worst-case execution conditions. Nonetheless, both the

modeling and the analysis we propose can face any scenario, including more realistic fault models.

- From $\mathcal{C}_i^{fault}$ it would be possible to extract $\langle C_i(\text{HI}), 10^{-9}\rangle$; we name $C_i(\text{HI})$ the HI-safe WCET as it results from the safest pWCET model among the possible scenarios, $\mathcal{C}_i^{fault} \equiv \mathcal{C}_i^{\text{HI}}$.
- From $\mathcal{C}_i^{no-fault}$ it would be possible to extract $\langle C_i(\text{LO}), 10^{-9}\rangle$; we name $C_i(\text{LO})$ the LO-safe WCET. as it results from the less safe pWCET model $\mathcal{C}_i^{no-fault} \equiv \mathcal{C}_i^{\text{LO}}$.

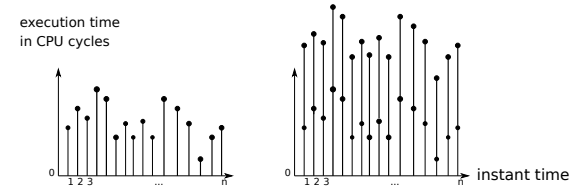
The difference that exist between *fault* and *no-fault* execution conditions results into $\mathcal{C}_i^{\text{HI}}$ greater than or equal to $\mathcal{C}_i^{\text{LO}}$. This is due to the larger execution time and eventually larger variability of the execution times in case of faulty conditions. The partial ordering between distributions defined as in [9]. Also $C_i(\text{HI}) \geq C_i(\text{LO})$ as they are both taken at $p = 10^{-9}$ from respectively $\mathcal{C}_i^{\text{HI}}$ and $\mathcal{C}_i^{\text{LO}}$. How much they differ depends on the relationship between faults and no faults and the impact that faults have on the task execution time.

Generalizing, there could exist sets of pWCETs $(\mathcal{C}_i^{\text{LO}}, \mathcal{C}_i^1, \ldots, \mathcal{C}_i^{\text{HI}})$ and sets of WCET thresholds $(C_i(\text{LO}), C_i(1), \ldots, C_i(\text{HI}))$ for a task. The intermediate safety levels would model execution conditions in between the worst-case and the *no-fault* case or different faults models/fault impacts. The probabilistic analysis would benefit from such a more fine grained modeling with better schedulability results. In this work we consider only two safety levels, thus $\tau_i = ((\mathcal{C}_i^{\text{LO}}, \mathcal{C}_i^{\text{HI}}), T_i, D_i)$ with the pWCET, or as $\tau_i = ((C_i(\text{LO}), C_i(\text{HI})), T_i, D_i)$ if only the WCET thresholds are applied.

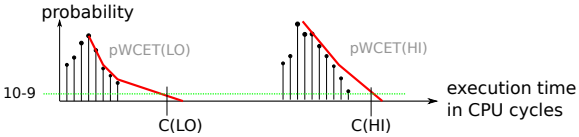
$$\tau_i \stackrel{def}{=} ([(\mathcal{C}_i^{\text{LO}}, \mathcal{C}_i^{\text{HI}}), (C_i(\text{LO}), C_i(\text{HI}))], T_i, D_i) \tag{5}$$

The enhanced task model from Equation (5) is essentially asserting that depending on the conditions for the timing analysis applied, it is possible to have more or less guarantees on the pWCET and the WCET thresholds estimates. Only by considering all the scenarios it would be possible to have safe worst-case models. Equation (5) defines the fault model to tasks.

Equation (5) reminds of the mixed-criticality task modeling with high and low criticality modes, [16]. Indeed it is a very similar notion, although here is more general as it applies to faults modeling, execution scenarios and confidence on the timing analysis tools. It could apply also to the mixed-criticality problem and the possible operational modes the task can have.



(a) Different execution time scenarios and the resulting traces of execution times



(b) Different execution time scenarios and the resulting worst-case estimates

Fig. 3. Different execution scenarios with different worst-case estimates.

Example 2. *From Example 1, the WCET thresholds resulting of the traces in faulty and non-faulty conditions are* $C(\text{HI}) = 80$ *(*$\mu = 70$*) and* $C(\text{LO}) = 20$ *(*$\mu = 15$*), respectively for the* fault *and* no-fault *scenarios.* fault *is the worst-case scenario, thus* $C(\text{HI})$ *and* $\mathcal{C}^{\text{HI}}$ *from it are safe models,* HI*-safe models. Instead,* no-fault *is the least safe scenario, thus* $C(\text{LO})$ *and* $\mathcal{C}^{\text{LO}}$ *from it are* LO*-safe models.*

While the WCET thresholds are taken at probability 10^{-9}*, other thresholds at* 10^{-3}*,* 10^{-5} *and* 10^{-6} *can be considered for increasing the flexibility and the representative of the probabilistic task representation. Then, it could exist* $\langle C_{i,1}^{\text{LO}}, 10^{-3}\rangle$*,* $\langle C_{i,2}^{\text{LO}}, 10^{-5}\rangle$*,* $\langle C_{i,3}^{\text{LO}}, 10^{-6}\rangle$ *and* $\langle C_{i,4}^{\text{LO}}, 10^{-9}\rangle$*. Figure 3(a) shows traces of execution time measurements in case of* no-fault *and* fault *conditions. Figure 3(b) depicts pWCETs in the two scenarios from the different measured execution times.*

V. The C-Space+

The C-Space+ is the extension of C-Space we develop for pWCETs appliying the new task model of Equation (5) for different execution scenarios and different safety levels. We make use of the C-Space+ for formalizing the sensitivity sensitivity and the schedulability analysis with probabilities and safety levels.

A. The C-Space

The C-Space is a space built on tasks WCETs C_i, [11], [12]. Within it, it is possible to represent feasibility regions where each point $\overline{c} = (C_1, C_2, \ldots, C_n)$ inside the region represents a schedulable task set. A point $\overline{c}$ outside the feasibility region is a task set not schedulable. Each task τ_i assumes WCETs from $\overline{c}$, $C_i \in \overline{c}$.

In case of EDF, the feasibility region is derived from the comparison between the demand bound function dbf and the available resource. The task schedulability is assured iff

$$\forall t \in D, \quad \mathsf{dbf}(t) \leq t, \tag{6}$$

with D the set of Γ deadlines within the hyperperiod H, [18].

Task τ_i demand bound functions dbf_i is such that $\mathsf{dbf}_i(t, \cdot) \stackrel{def}{=} \lfloor \frac{t-D_i}{T_i} + 1 \rfloor \times C_i$, while the demand bound function dbf of the set of tasks is the combination of tasks demand bound functions $\mathsf{dbf}(t, \cdot) \stackrel{def}{=} \sum_i \mathsf{dbf}_i(t, \cdot)$. The WCETs C_i are the variables for building the C-Space. The feasibility region defined according to Equation (6) defines bounds on the C_is.

B. C-Space+ Representativeness

The C-Space+ accounts for the fault models with the discrete and finite possible execution conditions per task, Equation (5). C-Space+ is a space built on tasks WCET thresholds of Equation (4), where it is possible to represent schedulable task set, safety levels and confidence probabilities. A point $\overline{c}$ outside the feasibility region is a task set not schedulable.

The task model $\tau_i = ((C_i(\text{LO}), C_i(\text{HI})), T_i, D_i)$ results into the set of dbfs:

$$\mathsf{dbf}_i(t, C_i(\text{HI})) \stackrel{def}{=} \lfloor \frac{t - D_i}{T_i} + 1 \rfloor \times C_i(\text{HI}) \text{ and}$$
$$\mathsf{dbf}_i(t, C_i(\text{LO})) \stackrel{def}{=} \lfloor \frac{t - D_i}{T_i} + 1 \rfloor \times C_i(\text{LO}),$$

with $\mathsf{dbf}_i(t, C_i(\text{HI})) \geq \mathsf{dbf}_i(t, C_i(\text{LO}))$. Some tasks combinations accounting for the safety levels are:

$$\begin{aligned} \overline{c}^{\text{LO}} &= (C_1(\text{LO}), C_2(\text{LO}), \ldots, C_n(\text{LO})), \\ \overline{c}^{1} &= (C_1(\text{LO}), C_2(\text{LO}), \ldots C_n(\text{HI})), \\ &\ldots \ , \\ \overline{c}^{\text{HI}} &= (C_1(\text{HI}), C_2(\text{HI}), \ldots C_n(\text{HI})). \end{aligned}$$

As a result, there exist multiple possible demand bound functions for the set of tasks depending on the WCET thresholds chosen:

$$\begin{aligned} \mathsf{dbf}(t, \overline{c}^{\text{LO}}) &= \sum_{i}^{n} \mathsf{dbf}_i(t, C_i(\text{LO})), \\ \mathsf{dbf}(t, \overline{c}^{1}) &= \sum_{i=1}^{n-1} \mathsf{dbf}_i(t, C_i(\text{LO})) + \mathsf{dbf}_n(t, C_n(\text{HI})), \\ &\ldots \ , \\ \mathsf{dbf}(t, \overline{c}^{\text{HI}}) &= \sum_{i=1}^{n} \mathsf{dbf}_i(t, C_i(\text{HI})). \end{aligned}$$

We remind that the combinations and dbfs here before are not all the possible combinations. They are a subset where each element dominates other configurations i.e. upper-bounds other configurations.

The different WCET thresholds per task allow enriching the C-Space+ representation with more informations about the safety or not of the points $\overline{c}$. Moreover, by considering the probability associated to the C_is, it is possible to add probabilities to the C-Space+. Indeed, with $\langle C_{i,j}, p_{i,j} \rangle$, the resulting demand bound function $\mathsf{dbf}_i(t, C_{i,j}) = \lfloor \frac{t - D_i}{T_i} + 1 \rfloor \times C_{i,j}$ has $p_{i,j}$ associated which is the confidence on the demand bound function upper-bounding task behaviors. With probability $p_{i,j}$, τ_i demands more computation time than $\mathsf{dbf}_i(t, C_{i,j})$. For a point $\overline{c}$, the probability is given by the product of the WCET thresholds composing $\overline{c}$ due to the pWCET independence. The HI and LO safety levels and the probabilities remain separated.

Figure 4 gives an example of the C-Space+ with points $\overline{c}$ and probabilities represented. The probabilities are the exceeding thresholds associated to the points, equivalently the confidences of not passing the execution times $\overline{c}$ at runtime.

Example 3. *Given the probabilistic set of tasks* (τ_1, τ_2, τ_3) *such that:* $\tau_1 = ((C_1(\text{LO}) = 10, C_1(\text{HI}) = 14), 40, 40)$, $\tau_2 = ((C_2(\text{LO}) = 8, C_2(\text{HI}) = 15), 50, 50)$ *and* $\tau_3 = ((C_3(\text{LO}) = 15, C_3(\text{HI}) = 20), 40, 40)$*; time is expressed in CPU ticks. To the tasks model of Equation (5) we add two more thresholds per task at probability* 10^{-5}*. In particular, it is for* τ_1 $\langle C_{1,1}^{\text{LO}} = 8, 10^{-5} \rangle$, $\langle C_{1,2}^{\text{LO}} = 10, 10^{-9} \rangle$, $\langle C_{1,1}^{\text{HI}} = 12, 10^{-5} \rangle$ *and* $\langle C_{1,2}^{\text{HI}} = 14, 10^{-9} \rangle$*; for* τ_2 $\langle C_{2,1}^{\text{LO}} = 5, 10^{-5} \rangle$, $\langle C_{2,2}^{\text{LO}} = 8, 10^{-9} \rangle$, $\langle C_{2,1}^{\text{HI}} = 11, 10^{-5} \rangle$ *and* $\langle C_{2,2}^{\text{HI}} = 15, 10^{-9} \rangle$*; for* τ_3 $\langle C_{3,1}^{\text{LO}} = 10, 10^{-5} \rangle$, $\langle C_{3,2}^{\text{LO}} = 15, 10^{-9} \rangle$, $\langle C_{3,1}^{\text{HI}} = 16, 10^{-5} \rangle$ *and* $\langle C_{3,2}^{\text{HI}} = 20, 10^{-9} \rangle$*. In Figure 5(a) all the possible WCET thresholds combinations* $\overline{c}$ *from the input pWCETs are plotted. Schedulable* $\overline{c}$ *are distinguished from non schedulable ones. Horizontal planes are to distinguish safety levels (*HI*and* LO*) and WCET confidences (at* 10^{-9} *and* 10^{-5} *per safety level). Figure 5(b) shows the feasibility region for the EDF schedulability.*

Figure 6 gives a better insight of the C-Space+ with 2D representations. Only the plane (τ_1, τ_2) *is represented at two values of* C_3*, respectively* $C_3 = 15$ *and* $C_3 = 20$.

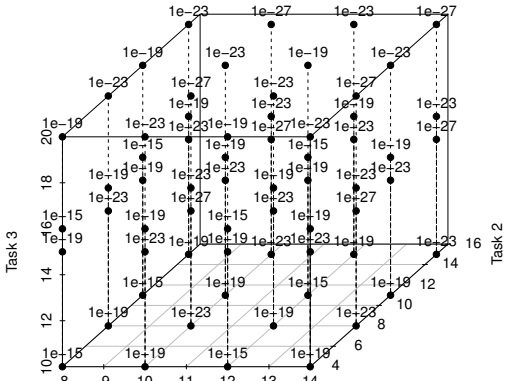
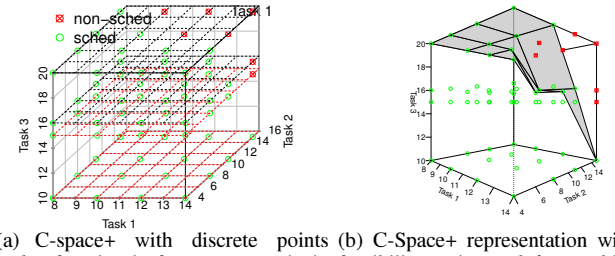


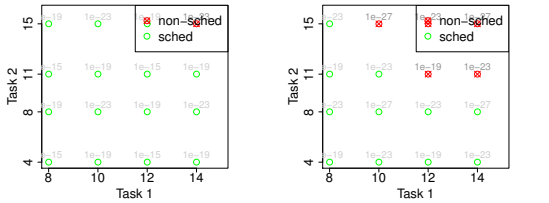
Fig. 4. Discrete points represented with probabilities associated to each point.



(a) C-space+ with discrete points and safety levels for τ_3, respectively $C_3(\text{LO}) = 15$ and $C_3(\text{HI}) = 20$

(b) C-Space+ representation with the feasibility region and the combination points $\overline{c}$

Fig. 5. C-space+ representations.



(a) 2D plane (τ_1, τ_2), $C_3 = 15$, thus $C_3(\text{LO})$

(b) 2D plane (τ_1, τ_2), $C_3 = 20$, thus $C_3(\text{HI})$

Fig. 6. 2D representation of the C-Space+.

The probabilities describe confidences on the $\overline{c}$*s: the tasks WCET configuration has that probability of being passed during execution. For example, for* $\overline{c} = (12, 11, 20)$*, it is that* τ_1 *and* τ_3 *have their* HI*-safe WCET thresholds in particular,* τ_1 *with the threshold at* $p = 10^{-5}$ *and* τ_3 *with the threshold at* $p = 10^{-9}$*. Instead,* τ_2 *has its* LO*-safe WCET threshold with the threshold at* $p = 10^{-9}$*.*

The probability associated to $\overline{c} = (12, 11, 20)$ *is* $p_{\overline{c}} = 10^{-5} * 10^{-9} * 10^{-9} = 10^{-23}$*, due to the independence between the tasks.* $p_{\overline{c}}$ *is the probability that the application* $\{\tau_1, \tau_2, \tau_3\}$ *passes the WCET threshold configuration* $\overline{c}$*; it is the confidence on* $\overline{c}$*. As a reminder, the probability is strictly associated to the safety level.* $p = 10^{-9}$ *for* τ_2 *is the probability that* $C_2(\text{LO})$ *is passed while remaining in the* LO*-safe characterization of the task. Thus* $p_{\overline{c}} = 10^{-23}$ *is the confidence on* $\overline{c} = (12, 11, 20)$ *where* τ_1 *and* τ_3*, although changing WCETs, they remain in their* HI*-safe characterization and* τ_2 *remains in its* LO*-safe characterization.*

C. Sensitivity Analysis for the C-Space+

The sensitivity analysis on top of the C-Space+ takes into account the different and possible WCET thresholds, the HI as well as LO safety levels and the probabilities.

With respect to faults, the sensitivity analysis allows i) quantifying the effects of faults on the application schedulability and ii) quantifying the maximum confidence it is possible guaranteeing for a specific configurations.

We describe the contributions of the sensitivity analysis via an example. Future work will consider a more detailed

investigation of the sensitivity analysis for the C-Space+.

Example 4. *Given the task set of Example 3, Figure 5(b) represents the feasibility region. Figure 7 depicts the C-Space+ with the distinction between* HI*-safe and* LO*-safe WCETs for* τ_3 *represented with the two horizontal planes. Figure 8 provides two 2D planes where both probabilities and safety levels are represented for the couple* (τ_1, τ_2) *with both* LO*-safety and* HI*-safety WCET for* τ_3.

We first observe how the non-schedulability relates to HI*-safety WCETs, especially with* $C_3 = 20$. *There are couples of non-schedulabilities, one with the* LO*-safety* τ_3, $C_3 = 15$, $\overline{c} = (14, 15, 15)$ *and one with the* HI*-safety* τ_3, $C_3 = 16$ *with probability* 10^{-5}, $\overline{c} = (14, 15, 16)$.

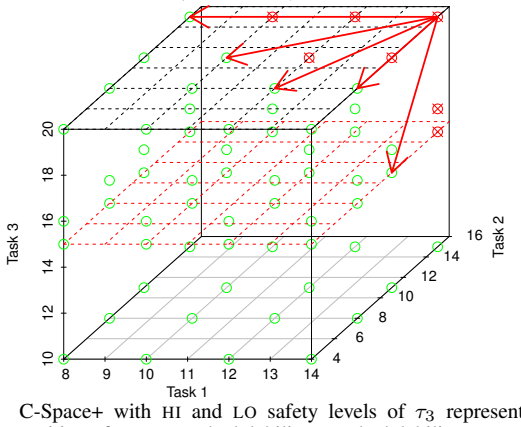
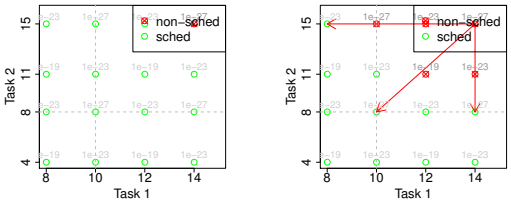


Fig. 7. C-Space+ with HI and LO safety levels of τ_3 represented. Some possible transitions from non-schedulability to schedulability are represented.



(a) 2D plane (τ_1, τ_2), $C_3 = 15$ (b) 2D plane (τ_1, τ_2), $C_2 = 20$

Fig. 8. 2D representations with safety levels HI and LO to distinguish effects on schedulability. Some possible transitions from non-schedulability to schedulability are represented.

From Figure 7 and Figure 8(b) we show how the sensitivity analysis work in quantifying the costs for a schedulable task set. For example, to pass from the non-schedulability $\overline{c} = (14, 15, 20)$ *to the schedulability, some possibilities are: i)* $\overline{c}^I = (14, 10, 15)$, *meaning that both* τ_2 *and* τ_3 *cannot be guaranteed at their* HI*-safety WCET; ii)* $\overline{c}^{II} = (8, 15, 20)$, *meaning that* τ_1 *can be only with its* LO*-safety WCET at* $p = 10^{-5}$*; iii)* $\overline{c}^{III} = (12, 8, 20)$, *meaning that* τ_1 *can be guaranteed up to* $C_1 = 12$ HI*-safety but at* $p = 10^{-5}$ *and* τ_2 *can be guaranteed up to* $C_2 = 8$ LO*-safety but at* $p = 10^{-9}$.

From Figure 8(b) we can see that, in order to restore task set schedulability and leaving τ_3 *unchanged, either: both* τ_1 *and* τ_2 *change to the* LO*-safety level, or only* τ_1 *changes to the* LO*-safety level at probability* 10^{-5}, *or* τ_2 *has to change to the* LO*-safety level. Each of the solutions has a different cost. For example, the first would have mean that for guaranteeing* τ_3 *schedulability with faults, both* τ_1 *and* τ_2 *can be guaranteed only without faults. With the second solution, it would have been possible guarantee* τ_1 *without faults and with a confidence of* 10^{-5} *instead of* 10^{-5}. *The third solution instead, would have implied that only* τ_2 *would have been guaranteed without faults but with a confidence of* 10^{-9}.

In terms of confidence of the schedulable points, both the first solution and the third solution have associated a confidence of 10^{-27}. *The second solution instead has a confidence associated of* 10^{-23}.

VI. Conclusion

The work proposed embeds faults and fault impacts into real-time schedulability analysis. First, the probabilistic task model accounts for both faulty conditions and non-faulty execution conditions. Then, the probabilistic task model is applied with the sensitivity analysis for evaluating the task set schedulability. The C-Space+ allows to both qualify and quantify faults impacts on the schedulability through the sensitivity analysis.

Future works will include the development of schedulability strategies for leveraging the probabilistic fault models and the safety levels. Furthermore, the sensitivity analysis developed will be applied to the mixed-critical scheduling problem.

References

[1] 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*, 1995.

[2] J. Lehoczky, "Real-time queueing theory," in *10th of the IEEE Real-Time Systems Symposium (RTSS96)*, 1996.

[3] S. Edgar and A. Burns, "Statistical analysis of WCET for scheduling," in *22nd of the IEEE Real-Time Systems Symposium*, 2001.

[4] J. Hansen, S. Hissam, , and G. Moreno, "Statistical- based wcet estimation and validation," in *the 9th International Workshop on Worst-Case Execution Time*, 2009.

[5] 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 *ECRTS*, 2012.

[6] F. Guet, L. Santinelli, and J. Morio, "On the reliability of the probabilistic worst-case execution time estimates," in *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, 2016.

[7] D. Maxim and L. Cucu-Grosjean, "Response time analysis for fixed-priority tasks with multiple probabilistic parameters," in *IEEE Real-Time Systems Symposium*, 2013.

[8] G. Kaczynski, L. Lo Bello, and T. Nolte, "Deriving exact stochastic response times of periodic tasks in hybrid priority-driven soft real-time systems," *12th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'07), Greece*, 2007.

[9] J. Díaz, D. Garcia, K. Kim, C. Lee, L. Bello, L. J.M., and O. Mirabella, "Stochastic analysis of periodic real-time systems," in *23rd of the IEEE Real-Time Systems Symposium (RTSS02)*, 2002, pp. 289–300.

[10] L. Santinelli and L. Cucu-Grosjean, "A probabilistic calculus for probabilistic real-time systems," *ACM Trans. Embedded Comput. Syst.*, vol. 14, no. 3, 2015.

[11] L. George and J. Hermant, "Characterization of the space of feasible worst-case execution times for earliest-deadline-first scheduling," *Journal of Aerospace Computing, Information and Communication (JACIC)*, vol. 6, no. 11, 2009.

[12] E. Bini, M. D. Natale, and G. C. Buttazzo, "Sensitivity analysis for fixed-priority real-time systems," *Real-Time Systems*, vol. 39, no. 1-3, 2008.

[13] L. Santinelli and L. George, "Probabilities and mixed-criticalities: the probabilistic c-space," in *3rd International Workshop on Mixed Criticality Systems (WMC) at RTSS*, 2016.

[14] R. M. Pathan, "Fault-tolerant and real-time scheduling for mixed-criticality systems," *Real-Time Systems*, vol. 50, 2014.

[15] H. C. Wong and A. Burns, "Schedulability analysis for the abort-and-restart (ar) model," in *Proceedings of the 22Nd International Conference on Real-Time Networks and Systems (RTNS)*, 2014.

[16] Z. Guo, L. Santinelli, and K. Yang, "Edf schedulability analysis on mixed-criticality systems with permitted failure probability," in *21th IEEE International Conference on Embedded and Real-Time Computing System and Applications*, 2015.

[17] L. Cucu-Grosjean, "Independence - a misunderstood property of and for (probabilistic) real-time systems," in *the 60th Anniversary of A. Burns, York*, 2013.

[18] S. K. Baruah, L. E. Rosier, and R. R. Howell, "Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor," *Real-Time Systems*, vol. 2, no. 4, pp. 301–324, 1990.

[19] M. Slijepcevic, L. Kosmidis, J. Abella, E. Q. Nones, and F. J. Cazorla, "Dtm: Degraded test mode for fault-aware probabilistic timing analysis," in *the 25th Euromicro Conference on Real-Time Systems (ECRTS)*, 2013.

[20] C. Chen, L. Santinelli, J. Hugues, and G. Beltrame, "Static probabilistic timing analysis in presence of faults," in *11th IEEE International Symposium on Industrial Embedded Systems, SIES*, 2016.