Probabilistic Schedulability Analysis for Fixed Priority Mixed Criticality Real-Time Systems

Yasmina Abdeddaïm, Maxim Dorin

Probabilistic Schedulability Analysis for Fixed Priority Mixed Criticality Real-Time Systems

Yasmina Abdeddaïm
Université Paris-Est, LIGM, ESIEE Paris, France

Dorin Maxim
University of Lorraine, LORIA/INRIA, France

***Abstract*—In this paper we present a probabilistic response time analysis for mixed criticality real-time systems running on a single processor according to a fixed priority pre-emptive scheduling policy. The analysis extends the existing state of the art probabilistic analysis to the case of mixed criticalities, taking into account both the level of assurance at which each task needs to be certified, as well as the possible criticalities at which the system may execute. The proposed analysis is formally presented as well as explained with the aid of an illustrative example.**

I. Introduction

In this paper we propose a probabilistic analysis for Mixed Criticality Real-Time Systems (MCRTS). A MCRTS incorporates several functionalities of different criticalities on the same architecture. The highest criticality functionalities are usually related to safety critical applications and need to fulfill strict certification requirements. It is important to note that lower critical functionalities are still relevant for the good functioning of the system [1] and this is a key observation for our approach.

We introduce a probabilistic mixed criticality model where every task is of a certain criticality and the worst-case execution time (WCET) of a task is a discrete random variable. The set of possible execution times of every task are grouped into sets of WCETs of different criticalities depending on their probability of occurrence. Larger execution time values are less likely to occur, but if they do occur they may be indicative of an erroneous event and so the criticality of the system changed to a higher level. When the citicality of the system is high, the guarantees required from lesser criticality tasks may be decreased so that higher criticality tasks may fulfill their requirements. The main difference between our work and the state of the art is the fact that in our work tasks are not evicted from the system, but instead the timing constraints imposed on them are gradually decreased (with the increase of the systems criticality) so that they still provide a minimal quality of service.

The first work dealing with real-time scheduling for mixed criticality systems is that of Vestal [2]. The model presented in [2] is based on the conjecture that the higher is the degree of criticality at which a task is designed, the more conservative is its worst-case execution time (WCET). This means that tasks have several WCET estimates, one per level of criticality of the system, and a mixed criticalty system is correct if every task respects its timing guarantees when all the WCET values that can impact on the schedulability of that task do not exceed their WCET estimates at the level of criticality of that task. Since then, many papers dealing with the mixed criticality scheduling problem have been published. For more details on the mixed criticality scheduling problem a complete review can be found in [3]. Further-on we present related work from the Probabilistic Analysis domain and work that is at the intersection of mixed criticality and probabilistic analysis.

Probabilistic timing analysis models task parameters as random variables (e.g. the task's WCET, Minimum Inter-arrival Time (MIT)). For WCET analysis multiple paths have been explored, starting in 1995 when [4] introduced an analysis for tasks that have periodic releases but variable execution requirements. The algorithm called Probabilistic Time Demand Analysis (PTDA) is based on a bound of the processor demand of higher priority tasks and hence it is highly pessimistic. The next step towards an exact probabilistic analysis was made by [5] 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, [6] refined STDA into an exact analysis for real-time systems that have random execution times, represented as general random variables. In this work we extend the probabilistic response time analysis of [6] to the case of MCRTS.

Existing probabilistic analyses for real-time systems are not well suited for the analyses of MCRTSs, as they do not take into account the possible criticalities of the system and the fact that permitted failure proabilities might change when the criticality of the system changes.

A first step towards a probabilistic analysis for MCRTSs is [7], where the authors consider a dual criticality model consisting of implicit-deadline sporadic independent tasks where a probability that no job exceeds its low criticality WCET estimate is assigned to every high criticality task. In this model a system is probabilistically schedulable in the strong sense if the probability of missing a deadline (for any task) does not exceed a given threshold, while it is probabilistically schedulable in the weak sense if the set of high criticality tasks does not exceed their respective deadlines. A schdeulability analysis of an EDF-based (earliest deadline first) algorithm is proposed. Using this analysis a system that is deemed unschedulable using classical mixed criticality analysis can be deemed schedulable in a probabilistic sense. In [8] the authors introduce the notion of probabilistic C-Space and give some intuitions concerning the way it can be used for the

probabilistic sensitivity analysis of a mixed criticality system.

Our approach has some similarities with the analysis of [7] in the fact that it extends the probabilistic analysis to the case of mixed criticalities, however the analysis we propose is applicable for fixed priority pre-emptive systems (rather then EDF) and for systems with more than two criticality levels. Another way in which our contribution differs from that of [7] is that in our model jobs are not dropped to increase the schedulability of other tasks, but all tasks need to be verified to be within certain thresholds of failure probabilities depending on their criticalities and the criticality of the system.

Organization of the paper: We continue the paper with the description of the system model in Section II and the problem description in Section III. Then in Section IV we present the main contribution of our work, which is a theoretical response time analysis for mixed criticality real-time systems. To show how the analysis can be applied on a task-set we present an ilustrative example in Section V. Finally we conclude the paper in Section VI.

II. Model and Terminology

We model the mixed criticality real-time system to be analysed as a set of tasks $\Gamma = \{\tau_1, \ldots, \tau_n\}$ sorted in an increasing order of priority, from the lowest priority (τ_1) to the highest priority (τ_n), scheduled according to a pre-emptive fixed priority policy on a single processor.

In adition to the set of tasks the system is also characterized by a set of criticalities $\Omega = \{L_1, \ldots, L_m\}$. We consider Ω to be sorted from lowest criticality L_1 to highest criticality L_m. To every criticality level a maximum-probability of failure threshold is attributed by a deciding entity (e.g. system designer, certification authority). We denote this threshold by p^{L_i} for criticality L_i. Intuitively a higher criticality implies a more stringent probability threshold.

Each task τ_i is characterized by a tuple $\tau_i = (\chi_i, \mathcal{C}_i, T_i, D_i)$ such that $\chi_i \in \Omega$ represents its criticality, $\mathcal{C}_i$[1] represents its probabilistic worst-case execution time (pWCET), T_i represents its minimum inter arrival time and D_i represents its relative deadline. The criticality of each task is defined by the system designer at design-time.

The pWCET $\mathcal{C}_i$ of a task τ_i is given as a discrete random variable with a sample space $S_{\mathcal{C}_i}$ and a probability mass function (pmf) $p_{\mathcal{C}_i}$ where $p_{\mathcal{C}_i}(c)$ is equal to $P\{\mathcal{C}_i = c\}$, the probability that $\mathcal{C}_i$ is equal to c. The probability mass function of $\mathcal{C}_i$ is represented as:

$$p_{\mathcal{C}_i} = \begin{pmatrix} \mathcal{C}_{i,1} & \ldots & \mathcal{C}_{i,|S_{\mathcal{C}_i}|} \\ P\{\mathcal{C}_i = C_{i,1}\} & \ldots & P\{\mathcal{C}_i = C_{i,|S_{\mathcal{C}_i}|}\} \end{pmatrix} \quad (1)$$

We consider pWCET distributions to be known and their calculation is beyond the scope of this paper. The interested reader may refer to [9] and [10] for further reading on this topic.

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. As stated in [9], since we consider probabilistic worst-case values (for WCETs), the random variables are (probabilistically) independent.

Let $cdf(c) = P\{\mathcal{C}_i \leq c\}$ be the cumulative distribution function of the pWCET random variable. A WCET outcome $C_{i,k}$ of the *pWCET $\mathcal{C}_i$ is of criticality* L_j if and only if the probability that the pWCET exceeds $C_{i,k}$ ($P\{\mathcal{C}_i > C_{i,k}\} = 1 - cdf(C_{i,k})$) does not exceed the probability failure threshold p^{L_j}. The subset of WCETs of criticality L_j of task τ_i is denoted by $S_{\mathcal{C}_i}^{L_j}$ and is given by the following Equations:

$$S_{\mathcal{C}_i}^{L_j} = \{C_{i,k} \in S_{\mathcal{C}_i} / p^{L_{j+1}} < 1 - cdf(C_{i,k}) \leq p^{L_j}\} \quad (2)$$

if $1 \leq j < m$ and

$$S_{\mathcal{C}_i}^{L_m} = \{C_{i,k} \in S_{\mathcal{C}_i} / 1 - cdf(C_{i,k}) \leq p^{L_m}\}. \quad (3)$$

Intuitively, Equations 2 and 3 describe the fact that the pWCET distribution of a task can be split into m pieces (i.e. partial distributions), one piece for each criticality level. One or several such partial distributions may be void. The boundary of each piece is given by the minimal and maximal WCET values that the task can exhibit in a certain criticality level of the system, i.e. larger values would push the system into a higher criticality mode.

The representative WCET of a task τ_i in criticality level L_j, noted $C_i(L_j)$, is the largest WCET outcome of criticality less than or equal to L_j. The representative WCET $C_i(L_j)$ is equal to zero if all the WCET outcomes of task τ_i are of higher criticality than L_j.

We define the *criticality mode of the system*[2] as the smallest criticality such that no task τ_i executes for more than its representative WCET estimate at this level of criticality.

We note by:

$$p_{\mathcal{C}_i}^{\leq L_h} = \begin{pmatrix} \mathcal{C}_{i,1} & \ldots & \mathcal{C}_i(L_h) \\ p_{\mathcal{C}_i}(\mathcal{C}_{i,1}) & \ldots & p_{\mathcal{C}_i}(\mathcal{C}_i(L_h)) \end{pmatrix} \quad (4)$$

to be the partial mass function of $p_{\mathcal{C}_i}$ restricted to WCET outcomes of criticality less than or equal to the criticality L_h. If the set of WCET oucomes of critiality less than or equal to L_h is empty, then $p_{\mathcal{C}_i}^{\leq L_h}(r) = 0, \forall r \in \mathbb{N}$.

Each task τ_i generates an infinite number of jobs noted $\tau_{i,k}$. All jobs are assumed to be independent of other jobs of the same task and those of other tasks. The execution time of a job $\tau_{i,k}$ is denoted by $c_{i,k}$.We use $hp(i)$ to be the indexes of tasks of higher or equal priority than τ_i.

The response time analysis for probabilistic real-time systems makes use of the convolution operator, which is a way of summing up two independent random variables and it is formally defined as follows:

Definition 1. *The probability mass function $p_{\mathcal{Z}}$ of the sum $\mathcal{Z}$ of two (probabilistically) independent random variables $\mathcal{X}$ and $\mathcal{Y}$ is the convolution $p_{\mathcal{X}} \otimes p_{\mathcal{Y}}$ where $P\{\mathcal{Z} = z\} = \sum_{k=-\infty}^{k=+\infty} P\{\mathcal{X} = k\} P\{\mathcal{Y} = z - k\}$.*

[1] Throughout the paper we use calligraphic typeface to denote random variables

[2] In the rest of the paper, criticality **level** denoted by $L \in \{L_1, \ldots, L_m\}$ is used for tasks' criticality, and criticality **mode** is used for system criticality

A probabilistic analysis for (non-mixed criticality) real-time tasks where the WCET is a discrete random variable has been proposed in [6]. In this analysis, the worst-case response time probability mass function $p_{\mathcal{R}_i}$ of task τ_i is computed using the following equation (for more details please refer to [6]):

$$p_{\mathcal{R}_i} = \mathcal{B}_i \otimes p_{\mathcal{C}_i} \otimes \mathcal{I}_i \tag{5}$$

where $\mathcal{B}_i = \left(\bigotimes_{j \in hp(i)} p_{\mathcal{C}_j}\right)$ is the accumulated execution time requirements of all higher priority tasks that are instantiated at the same time as the task under analysis and $\mathcal{I}_i$ is the iterative convolution of jobs that may preempt the task under analysis. Note that Equation 5 is computed as if jobs would still continue to execute past their deadlines since this is an upper-bound over the case when jobs are stopped at their deadline. The exact analysis for the case when jobs are dropped at deadline is still an open problem, but the *no-drop* analysis is a tight over-approximation for the *drop* analysis and its tightness is related to the probabilities of missing deadlines as this is the difference between the two analyses.

III. Problem Description

In this paper we are interested in the pre-emptive fixed priority scheduling problem of probabilistic mixed criticality real-time systems running on a single processor. Under this mixed criticality probabilistic model introduced in Section II a schedule is considered feasible if and only if: when the criticality mode of the system is not larger than the criticality of a task, each task respects the probability of failure threshold specified for its criticality level. The idea is that all the tasks are certified at their own criticality level, but the lowest criticality tasks should not disturb the highest criticality ones when the criticality mode of the system increases. That is, when the criticality of the system increases, lower criticality tasks are no longer constrained to respect a stringent probability failure threshold, but instead the threshold is replaced with a more relaxed one so to ensure minimal quality of service while not hindering the timely execution of higher criticality tasks.

We consider that a task exhibits a failure if one of its jobs misses it deadline, consequently, the probability failure of a task is equal to its probability of deadline miss. In this work we consider that: (1) if a job exceeds the maximum WCET estimate corresponding to its criticality level it is not stopped and (2) a job is evacuated from the system if and only if it misses its deadline. The problem addressed in this paper is the computation of tasks' deadline miss probabilities (DMP) in order to verify that failure thresholds are respected and the system can be declared schedulable. To this extend we propose an analysis to compute response time distributions for tasks. Out of these response time distributions we can then extract deadline miss probabilities.

IV. PMC Schedulability Analysis

A. Response Time Analysis

In this section we introduce our probabilistic analysis for MCRTS, which we will denote by PMC (Probabilistic Mixed Criticality) in the rest of the paper.

The Probabilistic Mixed Criticality (PMC) response time analysis computes the partial probability mass functions $p_{\mathcal{R}_i}^{L_h}$ of the worst-case response time of task τ_i when the criticality mode of the system is L_h.

Given $p_{\mathcal{R}_i}$, the probability mass function of $\mathcal{R}_i$, one can decompose the worst-case response time probability mass function into partial functions as follows,

$$p_{\mathcal{R}_i} = \left(\begin{array}{c|c|c} p_{\mathcal{R}_i}^{L_1} & \dots & p_{\mathcal{R}_i}^{L_m} \end{array} \right)$$

where

$$p_{\mathcal{R}_i}^{L_h} = \begin{pmatrix} r \\ \dots \quad p_{\mathcal{R}_i}^{L_h}(r) \quad \dots \end{pmatrix}$$

is a partial function of $p_{\mathcal{R}_i}$ with $p_{\mathcal{R}_i}^{L_h}(r)$ is the probability that the worst-case response time of τ_i is equal to r when,

1) none of the jobs that are active in the interval $[0, r[$ executes more than $C_i(L_h)$, i.e. their respective representative WCET in the criticality mode L_h and,
2) if $h > 1$, there exists at least an active job in the interval $[0, r[$ that executes more than $C_i(L_{h-1})$, i.e. its representative WCET at the criticality mode L_{h-1}.

More formally,

$$\begin{aligned} p_{\mathcal{R}_i}^{L_h}(r) &= P\{\mathcal{R}_i = r \text{ and } \forall \tau_{j,k} \in [0, r[, c_{j,k} \leq C_j(L_h) \\ &\quad \text{and if } h > 1, \exists \tau_{j,k} \in [0, r[\text{ s.t. } c_{j,k} > C_j(L_{h-1})\}. \\ &= P\{\mathcal{R}_i = r \text{ and } L = L_h\} \\ &\quad \text{with } L \text{ the criticality mode of the system} \end{aligned}$$

Note that,

$$\sum_{L_h \in L} \sum_{r \in S_{\mathcal{R}_i}^{L_h}} p_{\mathcal{R}_i}^{L_h}(r) = 1$$

where $S_{\mathcal{R}_i}^{L_h}$ is the subset of outcomes of $\mathcal{R}_i$ in the case where the criticality mode of the system is L_h.

We emphasize that this partitioning of the response time distribution of task τ_i according to the different criticality modes is not a simple slicing of the response time distribution into m parts, but there can exist overlapping among these parts and each part is computed by the analysis. The coalescing of all the parts forms the complete response time distribution of the task regardless of the criticality mode of the system. The coalesced distribution is also the result returned by the analysis of [6] which does not take into account the different criticalities of tasks nor of the system. The computation of the partial response-times probability mass functions $p_{\mathcal{R}_i}^{L_h}$ of a task τ_i is formalised in Algorithm 1. In step 1 the algorithm computes for every criticality L_h the partial response time probability mass function when the criticaliy mode is *less than or equal to* L_h, for each L_h. Then in step 2 the partial response time probability mass functions for each criticality mode are computed. Steps 1 and 2 are detailed below.

Step 1: *Compute the partial probability mass function* $p_{\mathcal{R}_i}^{\leq L_h}$ *using Equation 6 below:*

$$p_{\mathcal{R}_i}^{\leq L_h} = \bigotimes_{j \in hp(i)} p_{\mathcal{C}_j}^{\leq L_h} \otimes p_{\mathcal{C}_i}^{< L_h} \otimes \mathcal{I}_i^{\leq L_h} \tag{6}$$

Algorithm 1 Computes for task τ_i the distributions $p_{\mathcal{R}_i}^{L_h}, \forall L_h \in \Omega$

Require: τ_i, $hp(i)$, $\{\tau_j, \forall j \in hp(i)\}, \Omega$
Ensure: $p_{\mathcal{R}_i}^{L_h}, \forall L_h \in \Omega$
  $h \leftarrow |\Omega|$
  while $h \neq 0$ **do**
    Compute $p_{\mathcal{R}_i}^{\leq L_h}$ using equation 6 (see step 1)
    $h \leftarrow h - 1$
  end while
  $p_{\mathcal{R}_i}^{L_1} = p_{\mathcal{R}_i}^{\leq L_1}$
  $h \leftarrow m$
  while $h \neq 1$ **do**
    Compute $p_{\mathcal{R}_i}^{L_h}$ using equation 7 (see step 2)
    $h \leftarrow h - 1$
  end while

Equation 6 is similar to Equation 5 where partial probability mass functions $p_{\mathcal{C}_i}^{\leq L_h}$ are used rather than the complete probability mass functions.

Step 2: *Compute the partial probability mass function* $p_{\mathcal{R}_i}^{L_h}$.

In this step we compute $p_{\mathcal{R}_i}^{L_h}$ the partial probability mass function of $p_{\mathcal{R}_i}$ when the criticality mode is equal to L_h using equation 7:

$$p_{\mathcal{R}_i}^{L_h} = \begin{cases} p_{\mathcal{R}_i}^{\leq L_1} \text{ if } h = 1 \\ p_{\mathcal{R}_i}^{\leq L_h} \ominus p_{\mathcal{R}_i}^{\leq L_{h-1}} \text{ if } h > 1 \end{cases} \tag{7}$$

where $f \ominus g(x) = f(x) - g(x)\ \forall x \in \mathbb{N}$.

Note that the $\ominus$ operator is well defined in this case, as the (partial)distribution $p_{\mathcal{R}_i}^{\leq L_{h-1}}$ to be subtracted is a subset of the (partial)distribution $p_{\mathcal{R}_i}^{\leq L_h}$, i.e. the sample space of $p_{\mathcal{R}_i}^{\leq L_{h-1}}$ is included in the sample space of $p_{\mathcal{R}_i}^{\leq L_h}$ as a result of the partial distributions used in Step 1. Below is an example of how this operators works:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0.4 & 0.2 & 0.1 & 0.1 \end{pmatrix} \ominus \begin{pmatrix} 1 & 2 & 3 \\ 0.4 & 0.1 & 0.1 \end{pmatrix} = \begin{pmatrix} 2 & 4 \\ 0.1 & 0.1 \end{pmatrix}$$

Proposition 1. *The probability that "the worst-case response time of task τ_i is equal to r and the criticality mode of the system is equal to L_h" is equal to p if and only if $p_{\mathcal{R}_i}^{L_h}(r) = p$ where $p_{\mathcal{R}_i}^{L_h}(r)$ is computed using Equation 7.*

Proof. By definition, $p_{\mathcal{R}_i}^{\leq L_h}(r)$ is the probability that the response time of task τ_i is equal to r and the criticality of the system is less than or equal to L_h. This probability is equal to the sum of the probabilities that the response time is equal to r computed in every criticality mode of the system less than or equal to L_h, thus

$$p_{\mathcal{R}_i}^{\leq L_h}(r) = \sum_{j=1\ldots h} P\{\mathcal{R}_i = r \text{ and } L = L_j\}$$

and similarly,

$$p_{\mathcal{R}_i}^{\leq L_{h-1}}(r) = \sum_{j=1\ldots h-1} P\{\mathcal{R}_i = r \text{ and } L = L_j\}$$

The partial distribution $p_{\mathcal{R}_i}^{\leq L_{h-1}}(r)$ is a subset of $p_{\mathcal{R}_i}^{\leq L_h}(r)$.

Using Equation 7, we have $p_{\mathcal{R}_i}^{L_h}(r) = p_{\mathcal{R}_i}^{\leq L_h}(r) - p_{\mathcal{R}_i}^{\leq L_{h-1}}(r) = \sum_{j=1\ldots h} P\{\mathcal{R}_i = r \text{ and } L = L_j\} - \sum_{j=1\ldots h-1} P\{\mathcal{R}_i = r \text{ and } L = L_j\}$ and thus,

$$p_{\mathcal{R}_i}^{L_h}(r) = P\{\mathcal{R}_i = r \text{ and } L = L_h\}$$

that is equal to the probability that the worst-case response time of task τ_i is equal to r and the criticality mode of the system is equal to L_h. □

B. PMC Sufficient Schedulability Test

Along with the PMC response-time analysis presented in the previous section we also propose a sufficient schedulability test which we present in this section. In the PMC feasibility test, for each criticality L_h, the deadline miss probability of a task τ_i needs to be less than or equal to the tasks' allowed failure probability threshold in criticality mode L_h of the system.

Definition 2. *(Task Deadline Miss Probability Per Criticality Mode). The deadline miss probability of a task τ_i in mode L, noted DMP_i^L, is the probability that task τ_i misses its deadline when the system is in criticality mode L. This probability is equal to:*

$$DMP_i^L = P\{\mathcal{R}_i^L > D_i\} \tag{8}$$

where $\mathcal{R}_i^L$, as computed using Equation 7, is the random variable $\mathcal{R}_i$ restricted to outcomes in the case of a criticality mode of the system equal to L.

The tasks deadline miss probability constraint to a certain criticlity mode L_h can be extracted from the partial response time mass functions of the task in the respective mode, using Equation 9 below.

$$DMP_i^{L_h} = \sum_{r > D_i} p_{\mathcal{R}_i}^{L_h}(r) \tag{9}$$

A sufficient feasibility test for PMC can be derived using Equation 9. If in each criticality mode L the deadline miss probability of a task τ_i is less than or equal to the probability threshold imposed to the task in mode L, then the task is considered schedulable.

When the criticality mode of the system is greater than the criticality level of the task, the deadline miss probability threshold imposed on the task can be increased to 1 and in this case the task is allowed to miss all of its deadlines. Alternatively, the threshold can be increased to an intermediate value smaller than 1, signifying that even though the task is less critical than the current mode of the system, it should still provide a certain quality of service, i.e. the probability of missing deadlines may be larger, but the task is not completely evicted from the system. The thresholds can be given by a standard or by a certification authority, or, alternatively, they can be decided upon by the system designer. An example of such schedulability thresholds is presented in Table III which also presents the degradation of tasks' failure thresholds as the criticality of the system increases. For example, a task of criticality 1 is allowed to have a DMP of 0.1 in system

	Criticality of the task		
System's Criticality	L_1	L_2	L_3
L_1	0.1	0.01	0.001
L_2	0.5	0.01	0.001
L_3	1	0.1	0.001

Table III
PERMITTED DEADLINE MISS PROBABILITY THRESHOLDS FOR THE TASK-SET IN TABLE I

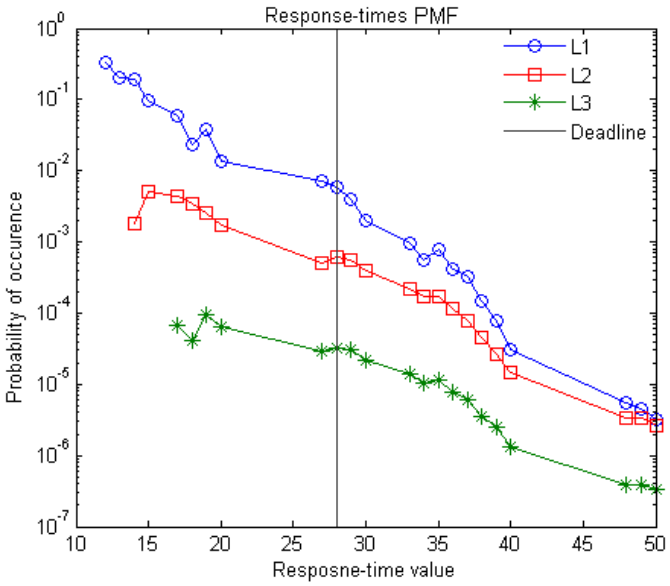

Figure 1. Response time partial distributions of task τ_5 in different criticality modes of the system.

criticality mode L_1, a DMP of 0.5 in mode L_2 and a DMP of 1 in mode L_3, i.e. in L_3 a task of criticality L_1 is not required to provide any service.

A note on complexity: it is well known that probabilistic analyses are computationally intensive [11] and the analysis we propose in this paper makes no exception. Nevertheless there are efficient solutions in the literature to go around this problem, such as re-sampling [11]. We do not go into details about complexity and ways of reducing it, as we rather use simple task-sets to exemplify our technique and provide a proof of concept.

V. ILLUSTRATIVE EXAMPLE

In order to provide an intuition of how the proposed analysis works we apply it on a simple tasks-set and show the results obtained. The task-set presented in Table I is composed of five tasks grouped in three levels of criticality L_1, L_2 and L_3. The pWCET distribution of each task has 6 values. The probability thresholds for each criticality level are $p^{L_1} = 0.1$, $p^{L_2} = 0.01$ and $p^{L_3} = 0.001$ and they further degrade as shown in Table III. These threshold together with Equation 2 and Equation 3 are used to split the pWCET distribution of each task into the (various sized) partial distributions that are depicted in Table II.

For this example we will be analyzing only task τ_5 as the analysis procedure is the same for all the other tasks. The system is scheduled according to deadline monotonic fixed priority preemptive scheduling policy, hence task τ_5 is the lowest priority in the set. Also, task τ_5 is considered to be of criticality L_2, an intermediate criticality in the system. From Table III we see that the schedulability constraints imposed to a task of criticality L_2 are as follows: if the system is in mode L_1 or L_2, which are normal modes from the point of view of a task of criticality L_2, i.e. neither it, nor any other task exceeded their largest WCET estimated for criticality L_2, then the task needs to function within a maximum allowed deadline miss probability of 0.01. On the other hand if the system switches to criticality mode L_3 - which is an error mode from the point of view of a task of criticality L_2 meaning that one of the tasks in the system exceeded their largest WCET estimated for mode L_2 - its threshold is modified to 0.1. Intuitively, a lower criticality task would have larger allowed DMPs (i.e. thresholds) making it possible for the task to be placed at a lower priority level, and, consequently, more critical tasks may have higher priorities.

The response time partial distributions of task τ_5 computed using our analysis are presented in Figure 1. We note that all curves were truncated at the value 50 as otherwise the plot would be too large and difficult to read and we mention only that maximal response time values of the task, in any mode, are infinitely large. That is, according to any deterministic analysis, task τ_5 would never finish its execution in the worst case, meaning that the system would be deemed unschedulable (even for the lowest criticality mode). It is easy to see why this is the case, as the (deterministic) utilisations of the system are too large for it to be schedulable. For example the maximal utilisation, computed using the largest values of each (complete) pWCET distribution, is equal to 2.19. Even the minimal utilisation of the system, computed using the smallest values of each pWCET distribution, is equal to 0.56, close to the schedulability limit of a fixed priority preemptive system.

We emphasize that the curves in Figure 1 are probability mass functions (PMFs) and not exceedence curves (i.e. 1-CDFs), and they show the probability that a specific response time value (read on the X-axis) is observed during the execution of task τ_5. These curves are decreasing exponentially (note the logarithmic Y-axis) as a result of the fact that the pWCETs of the example task-set are decreasing and so the larger response time values have exceedingly smaller probabilities of appearing. The decreasing pWCET distributions are characteristic of tasks of real systems, which are presumed to follow Gumbel distribution [12].

The deadline miss probability of task τ_5 in each criticality mode of the system are computed by adding together the probability mass of all values (from the respective curve) that are larger than the deadline. In this example task τ_5 has a deadline of 28. The probability that this deadline is missed is equal to 0.00935 (≤ 0.01) in criticality mode L_1 of the system, to 0.00177 (≤ 0.01) in mode L_2 and 0.00011 (≤ 0.1) in mode L_3. The reason that τ_5 has a smaller DMP in mode L_3 than in modes L_2 and L_1, even though it is allowed to have a larger DMP, is once more because the pWCET distributions are decreasing, making it highly unlikely for large response times to even be present in the system. If, on the contrary, we

ID	pWCET	T	D	χ
τ_1	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 0.8 & 0.1 & 0.099 & 0.0009 & 0.00009 & 0.00001 \end{pmatrix}$	10	10	L_3
τ_2	$\begin{pmatrix} 1 & 3 & 4 & 7 & 8 & 10 \\ 0.9 & 0.09 & 0.0099 & 0.00009 & 0.000009 & 0.000001 \end{pmatrix}$	15	15	L_2
τ_3	$\begin{pmatrix} 2 & 3 & 5 & 6 & 8 & 9 \\ 0.7 & 0.199 & 0.01 & 0.05 & 0.04099 & 0.00001 \end{pmatrix}$	20	20	L_1
τ_4	$\begin{pmatrix} 3 & 4 & 5 & 7 & 8 & 11 \\ 0.9 & 0.09 & 0.005 & 0.00399 & 0.001 & 0.00001 \end{pmatrix}$	20	20	L_3
τ_5	$\begin{pmatrix} 4 & 6 & 7 & 9 & 10 & 12 \\ 0.9 & 0.09 & 0.009 & 0.0009 & 0.00009 & 0.00001 \end{pmatrix}$	28	28	L_2

Table I
EXAMPLE OF A TASK-SET WITH PROBABILISTIC WORST CASE EXECUTION TIMES DISTRIBUTIONS.

ID	pWCET(L1)	pWCET(L2)	pWCET(L3)
τ_1	$\begin{pmatrix} 1 & 2 & 3 \\ 0.8 & 0.1 & 0.099 \end{pmatrix}$	$\begin{pmatrix} 4 \\ 0.0009 \end{pmatrix}$	$\begin{pmatrix} 5 & 6 \\ 0.00009 & 0.00001 \end{pmatrix}$
τ_2	$\begin{pmatrix} 1 & 3 & 4 \\ 0.9 & 0.09 & 0.0099 \end{pmatrix}$		$\begin{pmatrix} 7 & 8 & 10 \\ 0.00009 & 0.000009 & 0.000001 \end{pmatrix}$
τ_3	$\begin{pmatrix} 2 & 3 & 5 & 6 & 8 \\ 0.7 & 0.199 & 0.01 & 0.05 & 0.04099 \end{pmatrix}$		$\begin{pmatrix} 9 \\ 0.00001 \end{pmatrix}$
τ_4	$\begin{pmatrix} 3 & 4 \\ 0.9 & 0.09 \end{pmatrix}$	$\begin{pmatrix} 5 & 7 & 8 \\ 0.005 & 0.00399 & 0.001 \end{pmatrix}$	$\begin{pmatrix} 11 \\ 0.00001 \end{pmatrix}$
τ_5	$\begin{pmatrix} 4 & 6 \\ 0.9 & 0.09 \end{pmatrix}$	$\begin{pmatrix} 7 & 9 \\ 0.009 & 0.0009 \end{pmatrix}$	$\begin{pmatrix} 10 & 12 \\ 0.00009 & 0.00001 \end{pmatrix}$

Table II
THE SPLITTING OF PWCETS INTO PARTIAL DISTRIBUTIONS ACCORDING TO THE CRITICALITY THRESHOLDS.

would have chosen increasing pWCETs for our example, then the curve representing mode L_3 would have been above the other two curves and also the tasks' DMP in mode L_3 would be much larger than in the other modes.

If we coalesce these three partial distributions we would obtain the complete response time distribution of the task independent of the functioning mode of the system. This would be the same distribution that the analysis of [6] would return. According to this analysis, the DMP of the task (irrespective of the systems criticality) would be 0.01124 which is larger than the threshold of 0.01 that is imposed on the task, hence the task would be deemed unschedulable at this priority level.

VI. CONCLUSION

In this paper we proposed mixed criticality real-time system model and a probabilistic schedulability analysis for MCRTSs running on a single processor according to a fixed priority preemptive scheduling policy. The analysis extends the existing state of the art probabilistic analysis to the case of mixed criticalities, taking into account both the level of assurance at which each task needs to be certified, as well as the possible criticalities at which the system may execute.The proposed analysis is formally presented as well as explained with the aid of an illustrative example. As future work we plan to provide a formal proof of correctness. Intuitively the analysis is safe as it is a direct extension of an existing state of the art analysis which we further refine to decompose its results according to various functioning modes of the system.

ACKNOWLEDGEMENTS

This work was partially supported by the EU funded FP7 Integrated Project PROXIMA (611085), the FR BGLE funded Departs project (O16526-405635), the FR LEOC Capacites project, and the FR FUI Waruna project. The authors would like to thank Dagstuhl Seminar 15121 from which this work has emerged.

REFERENCES

[1] A. Burn, "Mixed criticality - a personal view," S. K. Baruah, L. Cucu-Grosjean, R. I. Davis, and C. Maiza, Eds., vol. 5, no. 3. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.

[2] S. Vestal, "Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance," in *the 28th IEEE Real-Time Systems Symposium RTSS 2007*, 2007.

[3] A. Burns and R. I. Davis, "Mixed criticality systems - a review [online]."

[4] 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.

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

[6] J. Diaz, D. Garcia, K. Kim, C.-G. Lee, L. Lo Bello, J. Lopez, S. L. Min, and O. Mirabella, "Stochastic analysis of periodic real-time systems," in *the 23rd IEEE Real-Time Systems Symposium RTSS 2002*, 2002.

[7] Z. Guo, L. Santinelli, and K. Yang, "EDF schedulability analysis on mixed-criticality systems with permitted failure probability," in *the 21st IEEE International Conference on Embedded and Real-Time Computing Systems and Applications RTCSA 2015*, 2015.

[8] L. Santinelli and L. George, "Probabilities and mixed-criticalities: the probabilistic c-space," in *the 3rd Workshop on Mixed Criticality Systems WMC 2015*, 2015.

[9] L. Cucu-Grosjean, "Independence - a misunderstood property of and for (probabilistic) real-time systems," in *"Real-Time Systems: the past, the present, and the future" conference organized in celebration of Professor Alan Burns sixtieth birthday*, March 14th, 2013.

[10] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. QuiÃśones, and F. J. Cazorla, "Measurement-based probabilistic timing analysis for multi-path programs," in *the 24th Euromicro Conference on Real-Time Systems*.

[11] D. Maxim and al., "Re-sampling for statistical timing analysis of real-time systems," in *Proceedings of the 20th International Conference on Real-Time and Network Systems*, 2012.

[12] A. G. Gogonel and L. Cucu-Grosjean, "How do we prove that probabilistic worst case response time is a Gumbel?" the 6th Real-Time Scheduling Open Problems Seminar, 2015.