This is a repository copy of *A Survey of Probabilistic Schedulability Analysis Techniques for Real-Time Systems*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/146181/

Version: Published Version

Article:

Davis, Robert Ian orcid.org/0000-0002-5772-0928 and Cucu-Grosjean, Liliana (2019) A Survey of Probabilistic Schedulability Analysis Techniques for Real-Time Systems. LITES: Leibniz Transactions on Embedded Systems. ISSN: 2199-2002

https://doi.org/10.4230/LITES-v006-i001-a004

Reuse

Other licence.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

A Survey of Probabilistic Schedulability Analysis Techniques for Real-Time Systems

Robert I. Davis
University of York, UK and Inria, France
rob.davis@york.ac.uk

Liliana Cucu-Grosjean
Inria, France
liliana.cucu@inria.fr

Abstract

This survey covers schedulability analysis techniques for probabilistic real-time systems. It reviews the key results in the field from its origins in the late 1980s to the latest research published up to the end of August 2018. The survey outlines fundamental concepts and highlights key issues. It provides a taxonomy of the different methods used, and a classification of existing research. A detailed review is provided covering the main subject areas as well as research on supporting techniques. The survey concludes by identifying open issues, key challenges and possible directions for future research.

2012 ACM Subject Classification Software and its engineering → Software organization and properties, Software and its engineering → Software functional properties, Software and its engineering → Real-time schedulability, Computer systems organization → Real-time systems

Keywords and Phrases Probabilistic, real-time, schedulability analysis, scheduling

Digital Object Identifier 10.4230/LITES-v006-i001-a004

Received 2018-01-04 **Accepted** 2019-02-26 **Published** 2019-05-14

1 Introduction

Systems are characterised as *real-time* if, as well as meeting functional requirements, they are required to meet timing requirements. Real-time systems may be further classified as *hard* real-time, where failure to meet their timing requirements constitutes a failure of the system, or *soft* real-time, where failure to meet timing requirements leads only to a degraded quality of service. Today, both hard and soft real-time systems are found in many diverse application areas including: automotive, aerospace, medical systems, robotics, and consumer electronics.

Real-time systems are typically implemented via a set of programs, also referred to as tasks, which are executed on a recurring basis. Verifying the timing correctness of a real-time system is typically framed as a two step process:

- *Timing Analysis* seeks to characterise the amount of time which each task can take to execute on the hardware platform. Typically, this is done by estimating, as a single value, an upper bound on the *Worst-Case Execution Time* (WCET) of the task.
- *Schedulability Analysis* seeks to characterise the end-to-end response time of functionality involving one or more tasks, taking into account the way in which the tasks are scheduled and any interference between them. An upper bound on the Worst-Case Response Time (WCRT) can then be compared to the deadline for that function to determine if end-to-end timing requirements can be guaranteed.

The concept of *probabilistic real-time systems* departs in two main ways from the classical model described above. Firstly, it recognises that the execution times of tasks may exhibit significant variability due to hardware effects (e.g. caches, branch prediction, pipelines, and other hardware acceleration features) as well as due to different input values and paths taken through the code.

© Robert I. Davis and Liliana Cucu-Grosjean;
licensed under Creative Commons Attribution 3.0 Germany (CC BY 3.0 DE)

Leibniz Transactions on Embedded Systems
LITES Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Thus the WCET may be substantially larger than typical execution times and may rarely occur. Much of the work on the analysis of probabilistic real-time systems therefore models the execution time of each task using a probability distribution with distinct probabilities associated with each possible value of execution time. Secondly, the timing requirements are such that deadlines are no longer considered to require absolute guarantees, but rather the probability of the deadline being exceeded must be below some specified threshold. (In practice, there may also be constraints on the number of consecutive deadlines that can be missed and hence on the probability that such a black-out period exceeds a certain length).

The concept of probabilistic real-time systems spans both the traditional classifications of hard and soft real time systems. In the case of a hard real-time system, failure to meet a deadline may constitute a failure of the system; however, provided that the probability of such a failure occurring is sufficiently small, for example leading to a *failure rate* of no more than 10^{-7}, 10^{-8}, or 10^{-9} per hour of operation, then it may still be acceptable to the system designers. This stems from the observation that the mechanical and electrical components of systems are typically designed with similar failure rates in mind (measured in terms of the number of failures per hour or billion hours). Thus engineering the timing behaviour of a system function to ensure a much smaller failure rate would have little or no impact on overall reliability and availability, while it could potentially require the provision of much more costly hardware. Acceptable failure rates depend on the criticality level assigned to the system function, as an example in the automotive standard ISO-26262 each Automotive Safety Integrity Level (ASIL) is associated with an observable failure rate. These are 10^{-9} per hour for ASIL D, 10^{-8} for ASIL C and B, and 10^{-7} for ASIL A. (Note, the relationship between failure rates per hour of operation and appropriate thresholds on probabilities of failure for individual tasks depend on various factors considered in fault tree analysis, including any mitigations and recovery mechanisms that may be applied in the event of a timing failure [62]).

We note that traditional analysis techniques can still be applied in cases where a very low level of deadline misses are permissible; however, since the reasoning used can only argue that either all deadlines will be met or not, then the results produced can in some cases be very conservative, thus requiring substantial hardware over-provision compared to the results of *probabilistic schedulability analysis*[1] expressed in terms of probabilities or probability distributions.

In the case of soft real-time systems, failure to meet a deadline and the consequent late response represents a degradation in the quality of service provided or the utility of the results produced. Here, probabilistic schedulability analysis can be used to characterise the expected quality of service that the system will provide.

This survey classifies and reviews probabilistic schedulability analysis techniques for real-time systems, where one or more task parameters are modelled as random variables, i.e. via probability distributions. Probabilistic schedulability analysis aims to determine for a set of tasks with parameters described by probability distributions, scheduled according to a given policy (for example fixed priority preemptive scheduling or Earliest Deadline First (EDF)) if those tasks can be guaranteed to meet their timing requirements, described in terms of *probabilistic deadlines* (i.e. deadlines with associated thresholds on the maximum acceptable probability of a deadline miss), or to simply compute the probability of deadline failure.

Much of the literature on probabilistic schedulability analysis models task execution times via probability distributions, while other works consider probabilistic Worst-Case Execution Time (pWCET) distributions. The latter can be derived via probabilistic timing analysis, us-

[1] In this survey, we adopt the widely used term "probabilistic schedulability analysis" noting that it can easily be misinterpreted. To clarify, while the results produced are expressed in terms of probabilities or probability distributions, the analysis methods themselves are deterministic in the sense of always producing the same results from the same inputs, unlike, for example, randomised search techniques.

ing *analytical methods* referred to as *Static Probabilistic Timing Analysis (SPTA)* [35, 53, 11, 10, 85, 84] or *statistical methods* referred to as *Measurement-Based Probabilistic Timing Analysis (MBPTA)* [32, 58, 66, 44, 150, 135, 133, 88, 87]. Probabilistic timing analysis techniques are reviewed in detail in a companion survey [52].

Research into probabilistic schedulability analysis can be classified into eight main categories. This classification forms the basis for the main sections of this survey. Note, for ease of reference we have numbered the categories to match the sections, starting at 3.

3. *Probabilistic Response Time Analysis*: these methods compute the probability distribution of the response time of each task, or the jobs of each task. These response time distributions can then be compared to the deadlines to determine if the timing requirements are met.
4. *Probabilistic Analysis assuming Servers*: these methods assume that each task is allocated a proportion of the processor bandwidth via a server. This has the advantage of isolating the tasks from interfering with one another, which simplifies the schedulability analysis.
5. *Real-Time Queuing Theory*: these methods estimate the fraction of deadlines that will be met from queue length dependent *lead-time profiles*. These profiles describe the time to go to the deadline and the distribution of queue lengths obtained via queueing theory.
6. *Probabilities from Faults*: works in this category assume additional execution time or load on the system from fault recovery operations as a consequence of faults that occur according to some probability distribution.
7. *Statistical Analysis of Response Times*: works in this category differ from those above in that they use statistical techniques based on observations of response times to predict the probability of deadline failure.
8. *Probabilistic Analysis of Mixed Criticality Systems*: these methods analyse mixed criticality systems using a richer representation (e.g. execution times described by probability distributions) rather than discrete WCET estimates for different levels of criticality.
9. *Miscellaneous*: research in this category explores different aspects of scheduling probabilistic real-time systems, including tasks with precedence constraints, the imprecise computational model, randomised job dropping, priority assignment policies, component based scheduling, and multiprocessor scheduling.
10. *Addressing Issues of Intractability*: the methods reviewed in this section aim to significantly reduce the runtime of probabilistic schedulability analysis techniques.

The research in these categories is summarised by authors and citations in Table 1. Note the sub-categories correspond to the subsections of this survey.

It is interesting to note how research in the different categories has progressed over time. Figure 1 illustrates the number of papers reviewed in each of the main categories covered by this survey that were published in 3-year time intervals from 1988 to 2018. (This figure is best viewed online in colour). A number of observations can be drawn from Figure 1. Firstly, the volume of research into probabilistic schedulability analysis has greatly increased since its origins in the late 1980s / early 1990s. The number of publications on the main theme of probabilistic response time analysis (Section 3) has steadily increased during this time. Work on server-based analysis and real-time queueing theory (Sections 4 and 5) have produced a small number of papers over most of the time period. By contrast, work on probabilities derived from faults (Section 6) began in 1999, with a peak in 2003 - 2006, and fewer publications in recent years, Categories of more recent interest (i.e. publications since 2007) include statistical analysis (Section 7) and addressing issues of tractability (Section 10). However, the hot topic in recent years, albeit with mainly preliminary publications, is work on probabilistic approaches to mixed criticality systems (Section 8). This area has shown a rapid expansion in the number of publications since 2015.

Before moving to the sections of this survey which review the literature, we first discuss (in Section 2) fundamental concepts and issues pertaining to probabilistic schedulability analysis.

Table 1 Summary of publications from different authors in the categories described in the main sections and subsections of this survey.

3 Probabilistic Response Time Analysis
3.1 Analysis for Periodic Tasks with No Backlog Woodbury and Shin [152], Tia et al. [143], Atlas and Bestavros [13], Gardner et al. [60], Tanasa et al. [141]
3.2 Analysis for Periodic Tasks with Backlog Diaz et al. [54, 55], Lopez et al. [93], Kim et al. [76], Ivers and Ernst [70], Tanasa et al. [140]
3.3 Analysis for More Complex Task Models Cucu-Grosjean and Tovar [41], Cucu-Grosjean [40], Maxim et al. [105], Maxim and Cucu-Grosjean [106], Maxim and Bertout [104], Santinelli and Cucu-Grosjean [130, 131], Santinelli et al. [136], Khan et al. [74], Santinelli [129], Carnevali et al. [34], Ben-Amor et al. [23], Markovic et al. [103]
4 Probabilistic Analysis assuming Servers
4.1 Analysis for Server-based Systems Abeni and Buttazzo [2, 3, 4] , Kaczynski et al. [72], Manica et al. [99], Abeni et al. [6, 5], Palopoli et al. [122, 120, 121], Frias et al. [59, 146]
5 Real-Time Queueing Theory
5.1 Analysis based on Real-Time Queuing Theory Barrer [20], Panwar et al. [123], Lehoczky [83], Doytchinov et al. [56], Hansen et al. [67], Zhu et al. [154], Gromoll and Kruk [63], Kruk at al. [79]
6 Probabilities from Faults
6.1 Analysis of Fault Recovery on Processors Burns et al. [33, 30], Broster and Burns [27, 26], Kim and Kim [75], Aysan et al. [18], Short and Proenza [138], Santinelli et al. [134]
6.2 Analysis of Fault Recovery on CAN Navet et al. [117], Broster et al. [28, 29], Aysan et al. [19] Axer et al. [17] , Davis and Burns [48], Nolte et al. [119], Zeng et al. [153]
7 Statistical Analysis of Response Times
7.1 Statistical Estimation Navet et al.[116], Lu et al. [96, 97, 94, 95], Liu et al. [91], Maxim et al. [111]
8 Probabilistic Analysis of Mixed Criticality Systems
8.1 Analysis for Mixed Criticality Systems Santinelli and George [132], Guo et al. [64], Maxim et al. [107, 108], Alahmad and Gopalakrishnan [9, 8], Draskovic et al. [57], Abdeddaim and Maxim [1], Kuttler et al. [80]
9 Miscellaneous
9.1 Task Graphs and Precedence Constraints Manolache et al. [100, 101, 102], Hua et al. [69]
9.2 Multiprocessor Analysis Nissanke et al. [118], Leulseged and Nissanke [86], Mills and Anderson [113, 112], Liu et al.[92], Wang et al. [149, 148] , Ren et al. [128]
9.3 Miscellaneous Models and Techniques Hu et al. [68], Hamann et al. [65], Kim et al. [77], Gopalakrishnan [61]
9.4 Position Papers Quinton et al. [125], Cucu-Grosjean [42, 43]
10 Addressing Issues of Intractability
10.1 Re-sampling Refaat et al. [127], Maxim et al. [110, 109], Milutinovic et al. [114]
10.2 Analytical Methods and Other Techniques Chen and Chen [36] von der Bruggen et al. [147], Chen et al. [37]

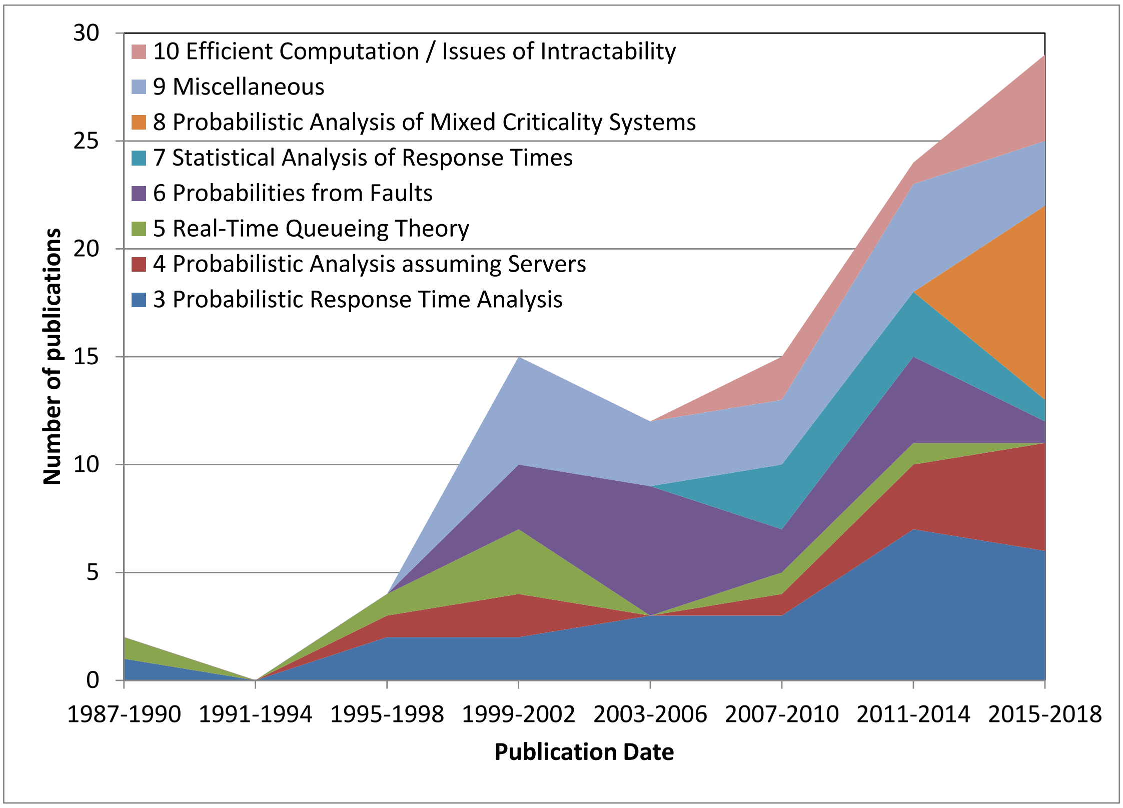

Figure 1 Intensity of research in the different categories corresponding to Sections 3 to 10 of this survey.

Note that conventional schedulability analysis techniques aimed at systems where task parameters are specified by single values rather than probability distributions are outside of the scope of this survey, they are reviewed in detail in a number of prior works [137, 49, 45].

2 Fundamentals and Key Issues

The term *probabilistic real-time systems* is used to refer to real-time systems where one or more of the task parameters (e.g. execution time, period, etc.) are described by random variables. Although a parameter, such as the execution time of a task, is described i.e. *modelled* by a random variable, this does not necessarily mean that the actual parameter itself exhibits random behaviour or that there is necessarily any underlying random element to the system that determines its behaviour. The actual behaviour of the parameter may depend on complex and unknown or uncertain behaviours of the overall system. As an example, the outcome of a coin toss can be modelled as a random variable with heads and tails each having a probability of 0.5 of occurring, assuming that the coin is fair. However, the actual process of tossing a coin does not actually have a random element to it. The outcome could in theory be predicted to a high degree of accuracy if there were sufficiently precise information available about the initial state and the complex behaviour and evolution of the overall physical processes involved. There are, however, many useful results that can be obtained by modelling the outcome of a coin toss as a random variable. The same is true in the analysis of probabilistic real-time systems.

In this survey we use calligraphic characters to denote random variables. As an example, the discrete probability distribution of the execution time $\mathcal{X}$ of a task may be described by a *Probability Mass Function* (PMF)

$$\mathcal{X} = \begin{pmatrix} 1 & 2 & 4 \\ 0.85 & 0.1 & 0.05 \end{pmatrix} \tag{1}$$

indicating that there is a probability of 0.85 that the execution time will be 1, a probability of 0.1 that it will be 2, and a probability of 0.05 that it will be 4. Thus the *Cumulative Distribution Function* (CDF) is given by:

$$F_{\mathcal{X}}(x) = P(\mathcal{X} \leq x) = \left\{ \begin{array}{ll} 0 & \text{if } x = 0 \\ 0.85 & \text{if } x = 1 \\ 0.95 & \text{if } x = 2 \\ 0.95 & \text{if } x = 3 \\ 1 & \text{otherwise} \end{array} \right) \tag{2}$$

Further, the *Complementary Cumulative Distribution Function* (1-CDF) or *Exceedance Function* is given by:

$$\bar{F}_{\mathcal{X}}(x) = P(\mathcal{X} > x) \left\{ \begin{array}{ll} 1 & \text{if } x = 0 \\ 0.15 & \text{if } x = 1 \\ 0.05 & \text{if } x = 2 \\ 0.05 & \text{if } x = 3 \\ 0 & \text{otherwise} \end{array} \right) \tag{3}$$

Timing requirements in probabilistic real-time systems are typically specified in terms of *probabilistic deadlines*. In contrast to conventional timing requirements where only a relative deadline is specified, probabilistic timing requirements also prescribe a limit or threshold on the maximum acceptable probability that the deadline will be missed. Extending the simple example above, let us assume that the task has a period of 10 and a deadline of 3, and is the only task in the system. In that case it would have a *Deadline Miss Probability* of 0.05 (which can be obtained directly from the 1 - CDF) and would therefore be viewed as schedulable if the threshold on the probability of deadline failure were specified as 0.1.

In the above example, with a single task, schedulability analysis deriving the probabilistic response time distribution is trivial, since the response time distribution is the same as the execution time distribution. However, once multiple tasks are considered, then execution time distributions need to be combined in some way to form a probabilistic response time distribution, and this gives rise to issues relating to *independence*.

2.1 Independence

▶ **Definition 1.** Two random variables $\mathcal{X}$ and $\mathcal{Y}$ are *probabilistically independent* if they describe two events such that knowledge of whether one event did or did not occur does not change the probability that the other event occurs. Stated otherwise, the joint probability is equal to the product of their probabilities $P(\{\mathcal{X} = x\} \cap \{\mathcal{Y} = y\}) = P(\mathcal{X} = x) \cdot P(\mathcal{Y} = y)$.

In our context this means that the execution times of two jobs of the same or different tasks are *independent* if the *event* that the execution time of the first job takes a particular value x has no effect on the probability that the execution time of the second job will take some value y. If this is not the case, then the execution times of the two jobs are said to be *dependent*.

Much of the literature on probabilistic schedulability analysis assumes that the random variables describing the execution times of jobs of the same or different tasks are independent. When the assumption of independence holds, then the *basic convolution* operator $\otimes$ can be used to determine the sum $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y}$ of the independent random variables $\mathcal{X}$ and $\mathcal{Y}$ where:

$$P\{\mathcal{Z} = z\} = \sum_{k=-\infty}^{k=+\infty} P\{\mathcal{X} = k\}P\{\mathcal{Y} = z - k\} \tag{4}$$

For example, if both $\mathcal{X}$ and $\mathcal{Y}$ are given by $\begin{pmatrix} 2 & 10 \\ 0.6 & 0.4 \end{pmatrix}$, then we have:

$$\begin{pmatrix} 2 & 10 \\ 0.6 & 0.4 \end{pmatrix} \otimes \begin{pmatrix} 2 & 10 \\ 0.6 & 0.4 \end{pmatrix} = \begin{pmatrix} 4 & 12 & 20 \\ 0.36 & 0.48 & 0.16 \end{pmatrix} \tag{5}$$

Issues of dependence are of great importance in probabilistic schedulability analysis, since an assumption of independence when it does not in fact exist can easily result in the computation of unsound[2] i.e. optimistic probabilities of a deadline miss. As an example, adapted from the work of Ivers and Ernst [70], consider how the execution time distributions $\mathcal{X}$ and $\mathcal{Y}$ for the jobs of two tasks may be combined to obtain a response time distribution. Here, we assume that task τ_X has the higher priority and runs first and we are interested in the response time of task τ_Y which is released at the same time as task τ_X, but runs once task τ_X has completed. If the execution times of the two jobs are independent, then the response time distribution may be obtained via convolution as given in (5) above. Thus assuming a deadline of 10 the probability of a deadline miss is 0.64, whereas with a deadline of 15 the probability of a deadline miss would be 0.16. If the execution times are instead perfectly positively correlated, then whenever task τ_X executes for 2 time units then so does task τ_Y, and similarly if τ_X executes for 10 time units, then so does task τ_Y. In this case, the response time can only take two values: 4 and 20, with probabilities of 0.6 and 0.4 respectively. Thus, compared to the independent case, the probability of missing a deadline of 15 is increased to 0.4, whereas the probability of missing a deadline of 10 is decreased to 0.6. If instead the execution times are perfectly negatively correlated, then whenever task τ_X executes for 2 time units then task τ_Y executes for 10 time units, and vice-versa. In this case the response time is always 12. Interestingly, while the probability of missing a deadline of 15 is reduced to zero, missing a deadline of 10 is now certain; it has a probability of 1, which is higher than in the case of either independence or positive correlation. This simple example serves to illustrate that the correctness of probabilistic schedulability analysis, and the response time distributions produced, crucially hinges upon the dependences between task execution times. The difficulty is that in practice the actual execution times of jobs of the same or different tasks are unlikely to be independent, rather they may exhibit correlation emanating from history dependent software states (e.g. static local variables) or history dependent hardware states (e.g. shared cache lines within the memory hierarchy), and also from dependences due to common or correlated input values that change only slowly over time.

2.2 pWCET Distributions

Some recent works on probabilistic schedulability analysis assume that the execution time behaviour of tasks is characterised by a probabilistic Worst-Case Execution Time (pWCET) distribution.

[2] In this survey, we use the adjective *sound* to indicate a description, an analysis, or a probability distribution that provides information about the system that is not optimistic with respect to its timing behaviour. Thus the information provided may be precise, or it may be pessimistic.

This term has been used widely in the literature, with a number of different definitions given. Below, we provide an overarching definition of the pWCET distribution.

▶ **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET)* distribution for a task is the least upper bound, in the sense of the greater than or equal to operator $\succeq$ (defined below), on the execution time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur.

▶ **Definition 3.** (From Diaz et al. [55]) The probability distribution of a random variable $\mathcal{X}$ is *greater than or equal to* (i.e. upper bounds) that of another random variable $\mathcal{Y}$ (denoted by $\mathcal{X} \succeq \mathcal{Y}$) if the Cumulative Distribution Function (CDF) of $\mathcal{X}$ is never above that of $\mathcal{Y}$, or alternatively, the 1-CDF of $\mathcal{X}$ is never below that of $\mathcal{Y}$. Similarly, the probability distribution of a random variable $\mathcal{X}$ is *less than or equal to* (i.e. lower bounds) that of another random variable $\mathcal{Y}$ (denoted by $\mathcal{X} \preceq \mathcal{Y}$) if the Cumulative Distribution Function (CDF) of $\mathcal{X}$ is never below that of $\mathcal{Y}$, or alternatively, the 1-CDF of $\mathcal{X}$ is never above that of $\mathcal{Y}$.

Graphically, Definition 2 means that the 1 - CDF of the pWCET distribution is never below that of the execution time distribution for any specific scenario of operation. Hence the 1 - CDF or *exceedance function* of the pWCET distribution may be used to determine an upper bound on the probability p that the execution time of a randomly selected job of the task will exceed an execution time budget x, for any chosen value of x. This upper bound is valid for any feasible scenario of operation.

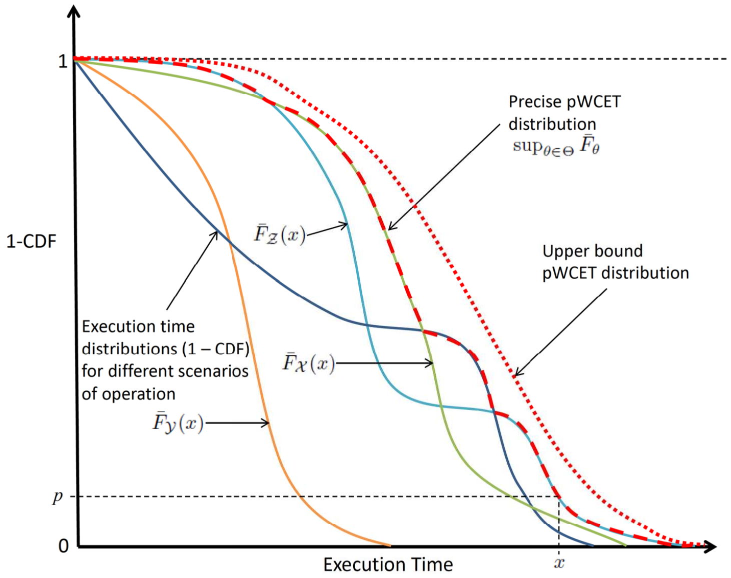

Figure 2 Exceedance function or 1-CDF for the pWCET distribution of a task, and also execution time distributions for specific scenarios of operation.

Figure 2 illustrates the execution time distributions for a number of different scenarios of operation (solid lines), the precise pWCET distribution (red dashed line) which is the least upper bound (i.e. the point-wise maxima of the 1 - CDF) for all of these distributions, and also some arbitrary upper bound pWCET distribution (red dotted line) which is a pessimistic estimate of the precise pWCET. Also shown (on the y-axis) is an upper bound p on the probability that any randomly selected job of the task will have an execution time that exceeds x (on the x-axis). The value x is referred to as the pWCET estimate at a probability of exceedance of p. (More formally, the precise upper bound pWCET distribution is given by $\sup_{\theta \in \Theta} \bar{F}_\theta$ where $\bar{F}_\theta$ is the 1 - CDF for scenario of operation θ, and Θ is the space of all valid scenarios of operation).

Note that the greater than or equal to relation $\succeq$ between two random variables does not provide a total order, i.e. for two random variables $\mathcal{X}$ and $\mathcal{Z}$ it is possible that $\mathcal{X} \not\succeq \mathcal{Z}$ and $\mathcal{Z} \not\succeq X$. Hence the precise pWCET distribution may not correspond to the execution time distribution for any specific scenario. This can be seen in Figure 2, considering the execution time distributions $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$. It is the case that $\mathcal{X} \succeq \mathcal{Y}$, but $\mathcal{X} \not\succeq \mathcal{Z}$ and $\mathcal{Z} \not\succeq X$.

We note that the term pWCET is open to misinterpretation and is often misunderstood. To clarify, it does *not* refer to the probability distribution of the worst-case execution time, since the WCET is a single value. Rather informally, following Definition 2, the pWCET may be thought of as the “worst-case” (in the sense of upper bound) probability distribution of the execution time for any scenario of operation.

It is interesting to consider the use and interpretation of the pWCET distribution for a task. Let us assume that the task will be run repeatedly a potentially unbounded number of times, and that a fixed execution time budget of x applies to each run. Further, we assume that this budget is enforced by the operating system, and therefore that any job of the task that has not completed within an execution time of x is terminated and assumed to have failed. The pWCET distribution provides the following information, by reading off the probability of exceedance p associated with the execution time budget x (see Figure 2):

(i) An upper bound p on the probability (with a long-run frequency interpretation) equating to the number of jobs expected to exceed the execution time budget divided by the total number of jobs in a long (tending to infinite) time interval.

(ii) An upper bound p on the probability that the execution time budget will be exceeded by a randomly selected job. (This is broadly equivalent to the above long-run frequency interpretation).

Contrast this with the binary information provided by the WCET. If x is greater than or equal to the WCET, then we can expect the budget to never be exceeded. If x is less than the WCET, then we expect the budget to be exceeded on some runs, but we have no information on how frequently this may occur.

We note that some researchers have interpreted the pWCET distribution as giving the probability or confidence $(1-p)$ that the WCET does not exceed some value x. This interpretation can be confusing, since the meaning of the WCET is normally taken to be the largest possible execution time that could be realised on any single job of the task. Instead, in line with Definition 2 and (i) above, we view the 1 - CDF of the pWCET distribution as providing, for any chosen value x for the execution time budget, an associated upper bound probability p (with a long-run frequency interpretation) equating to the number of jobs of the task expected to exceed the execution time budget x, divided by the total number of jobs of the task executing in a long time interval.

2.3 pWCET distributions and dependences

There are two main ways of obtaining pWCET distributions: via *Static Probabilistic Timing Analysis (SPTA)* or via *Measurement-Based Probabilistic Timing Analysis (MBPTA)*. (A detailed discussion and review of these methods is given in the companion survey [52] on probabilistic timing analysis).

The aim of SPTA methods is to *construct* an upper bound on the pWCET distribution of a task by applying static analysis techniques to the code, supplemented by information about input values, along with an abstract model of the hardware behaviour. For static analysis to produce a non-degenerate[3] pWCET distribution there typically has to be some part of the system or its environment that contributes random or probabilistic timing behaviour. SPTA methods for tasks running on time-randomised hardware (e.g. with a random replacement cache) effectively consider each path through the code. For each path, these methods construct a pWCET distribution that upper bounds the probability distribution of the execution time for that path, considering all possible initial hardware states and all possible input states that cause execution of the path. The upper bound pWCET distributions for every path are then combined using an envelope function (taking the point-wise maximum over the 1-CDFs) to determine an upper bound on the pWCET distribution for the task that is valid *independent* of the path taken. (More sophisticated SPTA methods analyse sub-paths and use appropriate join operations at path convergence to compute tighter upper bounds on the pWCET distribution of the task).

The upper bound pWCET distribution for a task derived by SPTA (as described above) upper bounds the pWCET distributions for every path through the code. Similarly, the pWCET distribution for each path upper bounds the execution time distribution for every input state that drives that path, and every initial hardware state that could occur at the start of that path. Hence, by *construction* the pWCET distribution derived by SPTA is *probabilistically independent* of the input state chosen or the path taken[4]. This has implications for the use that can be made of the pWCET distribution. Firstly, it can be used to bound the behaviour of any randomly selected job of the task. Secondly, since the pWCET distribution is independent of the input state, which may have strong dependences and correlations with the input states for previous jobs, it can be composed using basic convolution to upper bound the execution time behaviour of multiple jobs in a sequence. It can therefore be used to derive probabilistic worst-case response time (pWCRT) distributions, which can then be compared to deadlines to determine the probability of a deadline miss.

We note that the actual execution times for a sequence of jobs of a task, which exercise the same or different paths, may well show strong correlations and dependences. It is the *modelling* of the execution times via an appropriate pWCET distribution which enables probabilistic independence to be assumed. (This is similar to the conventional case of a single WCET which can similarly be used in this way, even though the actual execution times of different jobs have strong dependences).

The aim of Measurement-Based Probabilistic Timing Analysis (MBPTA) methods is to make a *statistical estimate* of the pWCET distribution of a task. This estimate is derived from a sample of execution time observations obtained by executing the task on the actual hardware or on a cycle-accurate simulator[5] according to an appropriate *measurement protocol.* The measurement protocol executes the task multiple times according to some sequence(s) of feasible input states and initial hardware states, thus sampling one or more possible scenarios of operation.

Provided that the sample of execution time observations passes appropriate statistical tests, then *Extreme Value Theory* (EVT) [39] can be used to derive a statistically valid estimate of the probability distribution of the *extreme values* of the execution time distribution, i.e. to estimate the pWCET distribution. By modelling the shape of the distribution of the extreme execution times, EVT is able to predict the probability of occurrence of execution time values that exceed

[3] A degenerate distribution has only a single possible value.

[4] This holds provided that the random values generated, for example by the random number generator within a random replacement cache, are also independent.

[5] A cycle-accurate simulator provides the same timing behaviour as the actual hardware, accurate to a single processor clock cycle.

any that have been observed. (For a basic introduction to the use of EVT in this context, see the companion survey [52] on probabilistic timing analysis. More detailed information about EVT can be found in Stuart Coles' textbook on the subject [39]).

▶ **Definition 4.** A sample of input states and initial hardware states used for analysis is *representative* of the population of states that may occur during a future scenario of operation if the property of interest (i.e. the pWCET distribution) derived from the sample of states used for analysis matches or upper bounds the property that would be obtained from the population of states that occur during the entire scenario of operation.

▶ **Definition 5.** As determined by MBPTA, the estimated pWCET distribution for a task (or path through a task) is a statistical estimate of the probability distribution of the *extreme values* of the execution time of that task (or path), valid for any future scenarios of operation that are properly represented by the sample of input states and initial hardware states used in the analysis.

Ideally, MBPTA would provide a pWCET distribution that is valid for any of the many possible future scenarios of operation; however, an important issue here is that there may not be one single distribution of input states and hardware states that is representative of all possible future scenarios of operation. This issue of *representativity* is a key open problem in research on the practical use of MBPTA, see Section 2.3 of the companion survey [52] on probabilistic timing analysis for a further discussion of this issue.

There are two main ways of applying MBPTA, referred to as *per-path* and *per-program*:

1. *Per-path:* MBPTA is applied at the level of paths. A measurement protocol is used to exercise all feasible paths through the program, then the execution time observations are divided into separate samples according to the path that was executed. EVT is then used to estimate the pWCET distribution for each path. The pWCET distribution for the program as a whole is then estimated by taking an upper bound (an envelope or point wise maxima on the 1 - CDFs) over the set of pWCET estimates for all paths.
2. *Per-program (or task):* MBPTA is applied at the level of the program i.e. the task. A measurement protocol is again used to exercise all paths. In this case, all of the execution time observations are grouped together into a single sample. EVT is then applied to that sample, thus estimating directly the pWCET distribution for the entire program.

With MBPTA, if the per-path approach is used, and it is known that the execution times for each path vary *only* due to random elements in the hardware, and do not otherwise vary across different input states that are in the same equivalence class (i.e. that drive the same path), then probabilistic independence of the pWCET distribution is assured. (Recall that with the per-path approach, the pWCET estimate for the task is an upper bound over the pWCET distributions for all of its paths). Probabilistic independence of the pWCET distribution means that it can be used to characterise the behaviour of any randomly selected job of the task, and also composed using basic convolution to upper bound the interference from multiple jobs in probabilistic schedulability analysis.

In other cases, the execution times obtained for a path may be dependent on the particular input states used to drive it. These input states may themselves exhibit dependences and correlations. Further, with the per-program approach, there may be dependences and correlations between the paths taken on consecutive jobs of the task, as happens with a software state machine. For systems with such dependences, then, assuming that the set of input states used during analysis is representative of those occurring during operation, the estimated pWCET distribution will still be valid in terms of characterising the extreme execution time behaviour of a *randomly* chosen single job of the task. However, it will typically *not* be valid to compose the pWCET distributions using basic convolution. The reason is that the pWCET distribution is not necessarily a valid estimate for the extreme execution time values of a job *conditional* on specific execution times having occurred for previous jobs.

As an example, consider a program E that implements a software state machine with four states and hence four paths that runs on time-randomised hardware. Here the main factor which affects the execution time of the program is the path taken, which is determined by the value of a software state variable. For this program, all valid scenarios of operation involve the software state variable cycling through its four possible values in order, and hence the four possible paths executing in order on any four consecutive runs of the program. Further, assume that there is some variability in the execution time of each path due to an independent random element in the hardware (e.g. a random number generator), which contributes either 5 or 10 time units to the execution time for the path, with a probability of 0.5 in each case. The execution times of the four paths are (i) 10 or 15, (ii) 20 or 25, (iii) 30 or 35, and (iv) 40 or 45. Note, each path has a probability of 0.25 of being taken in a randomly chosen execution of the program.

The precise pWCET distribution valid for any scenario of operation is:

$$pWCET_{per-program} = \begin{pmatrix} 0 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 \\ 1 & 0.875 & 0.75 & 0.625 & 0.5 & 0.375 & 0.25 & 0.125 & 0 \end{pmatrix}$$

Using the per-program approach, MBPTA may tightly upper bound this distribution. While using the per-path approach, the pWCET distribution obtained would tightly upper bound the following distribution for the longest path.

$$pWCET_{per-path} = \begin{pmatrix} 0 & 40 & 45 \\ 1 & 0.5 & 0 \end{pmatrix}$$

We note that while the $pWCET_{per-program}$ distribution is valid to describe the execution time behaviour of a randomly chosen job of the task, convolution of this distribution is not valid to describe the overall execution time of two or more jobs. This is because the distribution is not valid conditional on the execution times observed for previous jobs. For example, if an execution time of 30 or 35 is observed for a specific job, then the execution time of the next job will necessarily be either 40 or 45 (each with a probability of 0.5) due to the behaviour of the software state machine. The probability that the total execution time of two consecutive jobs exceeds 70 is therefore 0.1875, whereas the value computed by applying convolution to the $pWCET_{per-program}$ distribution is 0.15625, which is optimistic due to an incorrect assumption of independence. In this simple example optimism can be avoided by using the $pWCET_{per-path}$ distribution, which would also be obtained by SPTA; however, this discards information and so gives a pessimistic result, indicating that the probability of the total execution time of two consecutive jobs exceeding 70 is 1.

2.4 Probabilistic Inter-arrival Times

As well as probabilistic execution times and pWCETs, a few works on probabilistic schedulability analysis have also considered tasks with inter-arrival times characterised by random variables (see Section 4.1) or probabilistic Minimum Inter-arrival Times (pMIT) (see Section 3.3).

▶ **Definition 6.** The *probabilistic Minimum Inter-arrival Time (pMIT) distribution* for a task is the greatest lower bound, in the sense of the less than or equal to operator $\preceq$ (see Definition 3), on the inter-arrival time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of job arrivals that characterise a feasible way in which recurrent execution of the task may occur.

For example, the pMIT of a task τ_i may be described by $\mathcal{T}_i = \begin{pmatrix} 5 & 10 \\ 0.2 & 0.8 \end{pmatrix}$ meaning that the probabilistic minimum inter-arrival time is lower bounded such that a job of τ_i has a probability of 0.8 of arriving 10 or more time units after the previous job, and a probability of 1.0 of arriving 5 or more time units after the previous job.

The pMIT distribution provides the following information, by reading off the probability of exceedance p (from the 1 - CDF) associated with a limit t on the inter-arrival time:

(i) An upper bound p on the probability (with a long-run frequency interpretation) equating to the number of jobs that are expected to have an inter-arrival time of less than t, divided by the total number of jobs in a long (tending to infinite) time interval.

(ii) An upper bound p on the probability that the inter-arrival time of a randomly selected job will be less than t. (This is broadly equivalent to the above long-run frequency interpretation).

This contrasts with the sporadic task model which simply assumes that the minimum inter-arrival time is bounded by some value T_i (which would be 5 in the above example), but gives no information about how often longer inter-arrival times may occur.

The majority of works surveyed that model inter-arrival times as random variables assume that the inter-arrival times between jobs of the same task and between consecutive jobs of one task and consecutive jobs of another task are *independent*. For example, one practical application described by Maxim and Cucu-Grosjean [106] is vehicle reverse parking systems. Here, the inter-arrival time of the sensing task is randomised to avoid the possibility of systematic and repeated interference between the parking sensors of two vehicles that are reversing towards each other.

We note, however, that dependences are easily possible as a consequence of particular implementations. For example, the inter-arrival times of a task could follow a fixed pattern due to the operation of a software state machine which sets the next arrival time. This would give a fixed but repeating sequence, such as $(5, 10, 5, 10, \ldots)$. Further, two tasks could have inter-arrival times that are generated by a simple algorithm from the output values of a random number generator. They would have pMITs that are easily found from the properties of the random number generator and the algorithm used. In this case, the inter-arrival times of jobs of the same task would be independent, but the inter-arrival times between consecutive jobs of the two tasks would be in lock step i.e. correlated and dependent.

As with execution times and pWCETs, work on probabilistic schedulability analysis that considers tasks with inter-arrival times or pMITs characterised by random variables must take great care to either ensure that arrival times are independent or to handle dependences correctly.

2.5 Probabilistic Real-Time Constraints

In classical real-time systems, timing constraints are typically specified in terms of task deadlines. The relative deadline D_i of a task τ_i is inherited by all of it's jobs. The deadline D_i specifies the maximum elapsed time that is permitted from the release of a job of the task until that job completes its execution. The time from the release to the completion of a job is referred to as its *response time*. The worst-case response time R_i of a task τ_i is the longest possible response time of any of its jobs. If it can be proven, via schedulability analysis, that all jobs of a task will always complete by their deadlines, i.e. that $R_i \leq D_i$, then the task is said to be *schedulable*. If all the tasks in a system are schedulable, then the system itself is said to be schedulable.

Building upon these concepts, probabilistic real-time constraints are typically expressed in the form of *probabilistic deadlines* defined as follows:

▶ **Definition 7.** The *probabilistic deadline* of a task τ_i is given by the combination of a relative deadline D_i with a single (deterministic) value and a *threshold* ρ_i specifying the maximum acceptable probability that the deadline may be exceeded.

The concept of a probabilistic deadline has, in a few works, been extended to relative deadlines that may take a number of different values each with an associated probability, expressed as a discrete random variable $\mathcal{D}_i$. Again, a *threshold* ρ_i specifies the maximum acceptable probability that the deadline may be exceeded. In this survey, we use the term *probabilistic deadline* to refer to the simpler form defined above with a single value for D_i, and make a clear distinction when describing work that considers deadlines as random variables.

In the classical view of hard real-time systems any deadline miss is sufficient to make a task, and hence the system unschedulable, thus the pattern and probability of deadline misses are not considered relevant. By contrast, in probabilistic real-time systems there are a number of different ways in which the probability of a deadline miss can be considered and interpreted:

1. As a probability with a long-run frequency interpretation equating to the expected number of missed deadlines divided by the total number of deadlines in a long (tending to infinite) time interval.
2. As the probability that a randomly selected job will miss its deadline, which is broadly equivalent to the long-run frequency interpretation.
3. As a bound on the probability that any specific job will miss its deadline.

These interpretations can be made considering the jobs belonging to (i) a specific task, (ii) a group of tasks that comprise a given application, or (iii) the system as a whole (i.e. all tasks). In the latter cases, there would be a threshold on the probability of deadline misses for each application, or a single threshold for the system as a whole; rather than thresholds for each task.

In the literature, the term *Deadline Miss Probability* (DMP) is often used to refer to the long-run frequency interpretation. This quantity is typically computed for task sets that are strictly periodic and thus have a behaviour which repeats after the hyperperiod or Least Common Multiple (LCM) of the task periods. In contrast, the term *Worst-Case Deadline Failure Probability* (WCDFP) is used to mean a bound on the probability that any specific job of a task will miss its deadline. This quantity is typically computed by reference to the probabilistic Worst-Case Response Time (pWCRT) of a task.

▶ **Definition 8.** The *probabilistic Response Time (pRT)* of a job $\tau_{i,j}$ of task τ_i, denoted by $\mathcal{R}_{i,j}$, describes the probability distribution of the response time of the j-th job of that task, indexed from the start of the hyperperiod.

▶ **Definition 9.** The *probabilistic Worst-Case Response Time (pWCRT)* of a task τ_i, denoted by $\mathcal{R}_i$, is an upper bound on the worst-case response time distribution for *any* job of the task.

Note the pWCRT of a task can be computed for both periodic and sporadic tasks by taking into account the worst-case arrival pattern.

If the tasks are strictly periodic (with a single value for their periods), then the deadline miss probability for a task can be computed by taking the average of the deadline miss probabilities of all of its jobs activated during a hyperperiod as follows:

▶ **Definition 10.** The *Deadline Miss Probability DMP_i* for a task τ_i is given by :

$$DMP_i = \frac{1}{n_{LCM}} \sum_{j=1}^{n_{LCM}} P(\mathcal{R}_{i,j} > D_i) \tag{6}$$

where n_{LCM} is the number of jobs of task τ_i activated during the hyperperiod.

▶ **Definition 11.** The *Worst-Case Deadline Failure Probability* $WCDFP_i$ for task τ_i is an upper bound on the probability that any single job of the task misses its deadline, computed directly from the pWCRT distribution $\mathcal{R}_i$ and the deadline D_i of the task as follows:

$$WCDFP_i = P(\mathcal{R}_i > D_i) \tag{7}$$

There are advantages and disadvantages to using the DMP and the WCDFP formulations. For strictly periodic task sets, it is possible to compute the DMP over the hyperperiod. (Note, this calculation becomes more complex if the task model permits a *backlog* of outstanding execution at the end of the hyperperiod, see Section 3.2). For sporadic task sets the DMP formulation is not viable. This is because the deadline miss probability for the periodic case does not provide an upper bound on that for the sporadic case. As an example, consider a system with two tasks with minimum inter-arrival times $T_1 = 10$ and $T_2 = 15$. Sporadic behaviour of task τ_1 which aligns every release of that task with a job of τ_2 may result in a larger deadline miss probability than strictly periodic behaviour with a period of 10. This can be seen by considering the degenerate case where both tasks have an execution time and a deadline of 1 with task τ_1 having the higher priority. With all releases synchronised, i.e. both periods set to 15, task τ_2 never meets its deadline, while if task τ_1 is released more frequently with a period of 10, and task τ_2 has a period of 15, then the deadline miss probability for task τ_2 becomes 0.5. Attempting to account for all possible phasings of jobs of sporadic tasks leads to problems of intractability, hence a different approach is needed.

The WCDFP formulation potentially introduces some pessimism, since the relationship between task periods means that not all jobs of a task may be subject to the maximum interference from other tasks; however, it provides a valid upper bound on the probability of deadline misses for systems where not all tasks have strictly periodic releases (i.e. for the sporadic task model).

Hard real-time systems in many application domains can in practice tolerate a small number of consecutive deadline misses, but cannot tolerate long black-out periods when no deadlines are met. In the literature on deterministic schedulability analysis these issues have been investigated in work on *weakly-hard real-time systems* [24, 25], *m-k firm deadlines* [126], *skip-over* [78] techniques, and *typical worst-case analysis* [7]. The problem of reconciling requirements on the length of potential black-out periods (i.e. the number of consecutive missed deadlines) and a probabilistic treatment of deadline misses has, as far as we are aware, received little attention in the literature.

There are a number of issues here, which reflect long-run versus short-run viewpoints and the underlying issue of *independence*. We illustrate these problems via a simple example. Assume that we have two periodic task systems A and B, both with a hyperperiod that equates to four jobs of the task that we are interested in analysing. Further, let the probability that each of these jobs misses its deadline be as follows: system A $(0.75, 0.75, 0.25, 0.25)$ and system B $(0.5, 0.5, 0.5, 0.5)$. For both systems, the DMP for the task is 0.5; however, for system A, the WCDFP is 0.75, whereas for system B it is 0.5. Further, in order to reason about the probability of a black-out period equating to two consecutive deadline misses, we need to know if the response time distributions for the jobs, and hence the probabilities of deadline misses are independent or if they are correlated in some way. The probability that the first two jobs in the hyperperiod of system A both miss their deadlines would be 0.5625 if independence is assumed; however, it could be as high as 0.75 if the response times of the two jobs are positively correlated. We note that the response times of consecutive jobs of the same task may fail to be independent as a direct result of their execution times being dependent (e.g. due to dependences on shared software or hardware states, shared or correlated input variables etc.) or as an indirect result of interference from jobs of another task that exhibit execution time dependences.

2.6 Summary

In this section we described the fundamental building blocks used in probabilistic schedulability analysis, namely execution time distributions, response time distributions, and probabilistic deadlines. We also discussed the key underlying assumption in much of the literature on this topic: the independence of task execution times or pWCET distributions. We return to this issue in the conclusions. The next eight sections review the existing literature on probabilistic schedulability analysis.

3 Probabilistic Response Time Analysis

In this section we review work on probabilistic response time analysis. Comparing the probability distribution of the response time of a job of a task to its deadline enables the probability of a deadline miss for the job to be determined. For a task set with periodic release times, considering all these values for each job of the task over the hyperperiod (Least Common Multiple of task periods) enables the deadline miss probability for the task to be computed. Alternatively, some works derive the probabilistic worst-case response time distribution with respect to any job of the task, thus directly providing an upper bound on the worst-case deadline failure probability valid for any job. This latter approach works for both periodic and sporadic task sets.

In the following subsections, we classify and review papers according to the underlying task models that they support.

3.1 Analysis for Periodic Tasks with No Backlog

Initial work on probabilistic response time analysis focused on a simple task model where: (i) all tasks are periodic, (ii) the execution times of the jobs of each task are independent of those of other jobs of the same or different tasks, (iii) there is no *backlog* at the end of the hyperperiod, i.e. the worst-case processor utilisation is ≤ 1 and so the processor is guaranteed to be idle at the end of the hyperperiod.

In 1988, Woodbury and Shin [152] introduced analysis that computes the deadline miss probability for periodic tasks. The analysis assumes that tasks can be described in terms of multiple paths, each of which has a single deterministic execution time and a probability of occurrence. Thus each job of a task effectively has a probability distribution for its execution time, composed from the information about the paths. The analysis iterates over the hyperperiod in steps equating to the Greatest Common Denominator of the task periods. It computes the execution time remaining from higher priority jobs at the start of each step, and the joint distribution of the execution time of the jobs released at the start of the step. This information is used to derive the response time distribution for each job in the hyperperiod, and hence the probability of deadline failure for each task over the entire hyperperiod (i.e. considering all of its jobs). The technique is suitable for scheduling policies where the priority of a job does not change between releases (i.e. fixed priority or EDF). We note that it is arguable whether meaningful analysis can be obtained by assuming that the probability of taking a given path is known, and that the paths taken by different jobs are independent. In practice, the path taken is typically determined by the inputs, which are unlikely to be independent, for example sensor values may change only slowly over time. Further, formulating the analysis on the basis of paths can lead to tractability issues, since the number of paths could in practice be very large.

Probabilistic Time-Demand Analysis (PTDA) for fixed priority preemptive scheduling was proposed by Tia et al. [143] in 1995, based on the time-demand analysis technique given for the simpler case of WCETs by Lehoczky et al. [81]. With PTDA, at each scheduling point the cumulative probability distribution is computed (via convolution) for all job releases up to that point. This enables a bound to be computed on the probability that the task can meet its deadline.

The authors also propose a *transform-task* method of scheduling where part of the execution time distribution of each task (up to some specified value) is treated as a simple periodic task and guaranteed by time-demand analysis [81]. The remaining part of the distribution is considered as an additional sporadic task and runs under a sporadic server [139]. An alternative approach using EDF and slack stealing is also considered.

Statistical Rate-Monotonic Scheduling (SRMS) was introduced in 1998 by Atlas and Bestavros [13]. This method assumes that when a job is released then its execution time becomes known. SRMS consists of two parts, (i) admission control and (ii) a fixed priority (rate-monotonic) scheduling algorithm. The basic idea is to aggregate the capacity available to execute jobs of a task over multiple periods, equating to the so called *superperiod* given by the period of the next lower priority task. Jobs of a task are admitted if they can be executed within the remaining budget for the superperiod, and there is time available to do so before the deadline of the job, taking into account higher priority interference. Admitted jobs are then scheduled according to fixed priorities with rate-monotonic priority assignment. The authors show how the probability of admission can be computed for each period of a task within its superperiod. This allows quality of service guarantees to be computed in terms of the probability that a randomly selected job of a task will meet its deadline over an arbitrarily long time interval. Aside from the independence assumptions, the main drawback of SRMS is the requirement that job execution times become known upon release. In practice this is unlikely to be the case, however, in some cases execution times could be estimated from the input values.

Stochastic Time-Demand Analysis (STDA) for fixed priority scheduling, was developed in 1999 by Gardner et al. [60]. They note an issue with the prior work by Tia et al. on PTDA [143] in that it is only valid if there is no backlog at the deadline of a task. The problem is similar to the classical case of fixed priority scheduling with arbitrary deadlines [82]. All jobs in a priority level-i busy period need to be considered, since the first job is not necessarily the one that exhibits the worst-case behaviour. The authors provide an analysis based on the priority level-i busy period and the backlog present at subsequent releases of each job. They remark that the worst-case pattern of releases might not necessarily be following synchronous release of all tasks. They explore this aspect by considering different phasing in simulation, but did not find any cases where the probability of deadline failure was higher for random phasing than in the synchronous case. The convolution operations required as part of STDA are also discussed. Here, the authors indicate that the Fourier transform may be used to reduce complexity from $O(N^2)$ to $O(NlogN)$ where N is the number of points in each distribution. To use a fast Fourier transform, the distributions must have the same number of points and the same sampling rate.

In 2015, Tanasa et al. [141] studied the problem of determining probabilistic worst-case response time (pWCRT) distributions and hence deadline miss probabilities for a set of periodic tasks with execution times described by random variables. This work differs from earlier publications in that it describes the distributions via continuous variables and tightly approximates them with polynomial functions. This enables an analytical approach to be taken to their integration. The authors provide methods to compute the response time distribution for each job in the hyperperiod. They also introduce a method to compute the joint distribution between two jobs i.e. the probability that the response time of job $k_1 \leq r_1$ AND the response time of job $k_2 \leq r_2$. The methods used have exponential complexity and are thus limited in their application. In the evaluation, the maximum utilisation is limited to 85% for the experiments examining the univariate distributions and to no more than 40% for the joint distributions, using small hyperperiods of 25 or so jobs. The authors highlight an interesting observation, that even when the execution time distributions are continuous, the response time distributions may be disjoint, containing gaps where there are values that cannot be obtained. This occurs as a result of an additional preemption from a higher priority task that has a minimum as well as a maximum execution time.

3.2 Analysis for Periodic Tasks with Backlog

In contrast to classical task models, task sets containing a number of tasks with execution times described by random variables can usefully have a total worst-case processor utilisation that exceeds 1. This means that there is a *backlog*, meaning outstanding task execution with a finite probability of occurrence, at the end of each hyperperiod. This backlog makes the analysis of probabilistic response times for each job in the hyperperiod much more complex. Diaz et al. addressed this problem in two seminal papers published in 2002 [54] and 2004 [55]. This work has since been built upon by a number of other authors. Note the papers surveyed in this section also assume independence between the execution times of jobs of the same and different tasks, with the exception of the work of Ivers and Ernst [70] which addresses the important issue of dependences.

In 2002, Diaz et al. [54] noted that prior work on PTDA [143] and SDTA [60] assumes that the worst-case occurs for a job in the first busy period following synchronous release; however, this is not necessarily correct when the worst-case processor utilisation exceeds 1, and may lead to an under estimation of the response time distributions. They show that the *backlog* at the start of each hyperperiod is stationary provided that the average utilisation is less than 1. As the backlog at the end of one hyperperiod depends only on the backlog at the start of the previous hyperperiod, it can be modelled as a Markov chain. The authors show how to compute the stationary backlog, and use this backlog via a method of convolution (adding the execution of preempting jobs) and *shrinking* (shifting the distribution left and then accumulating the probabilities for all negative values at zero) to compute the response time distributions for each job of a given task over the hyperperiod. (A useful illustrative example is given in later work by the same authors [55]). By taking the average of these distributions, the response time distribution for the task can be computed and hence a bound on its deadline miss probability determined. We note that the method requires that the release times of jobs are fixed (i.e. periodic rather than sporadic), and when the hyperperiod is large then finding the stationary backlog can be computationally very expensive.

Subsequent research into the same problem by Diaz et al. [55] in 2004 discussed the concept of pessimism in probabilistic analysis, and introduced the concept of *greater than or equal to* between random variables (See Section 2, Definition 3). The authors note that any approximations in the analysis, for example of response time distributions, must result in distributions that are *greater than or equal to* the precise distribution in order to ensure soundness. They prove various properties with respect to their analysis, including that approximating the backlog and execution time distributions with distributions that are greater than or equal to ($\succeq$) the precise distributions produces sound results, i.e. there is no under-approximation of response time distributions. We note that this concept is similar to the one of *sustainability* introduced later for deterministic schedulability analysis by Baruah and Burns [21].

Diaz et al. [55] highlighted and addressed three issues with their previous work [54]:

(i) When the maximum utilisation exceeds 1, then the steady state backlog has an infinitely long tail.

(ii) Starting with zero backlog and iterating over a number of hyperperiods, the backlog becomes closer and closer to the steady state backlog; however, the estimation remains optimistic.

(iii) Probability distributions with a large number of points require a very large amount of memory and processing resource.

The problem of the infinite tail is solved via truncation and accumulation of the probabilities for larger values at ∞. A solution to the problem of a large number of points is suggested, accumulating values to the right, effectively a sound form of re-sampling (see Section 10.1 for later work in this area). The authors also show how blocking due to shared resource accesses can be accommodated as an extension to the task model by taking a *supremum* over all of the distributions for the execution time of relevant resource accesses. They also provide a sketch proof that the priority assignment algorithm of Audsley [15], which is optimal in the deterministic case, remains optimal when execution times are described by probability distributions and schedulability is defined as meeting the probabilistic deadline for each task. This was later confirmed by the work of Maxim et al. [105]. In 2008, Lopez et al. [93] extended the work of Diaz et al. [54, 55] providing: (i) a set of transformations that can be made to the parameters of a system which are guaranteed to result in a response time distribution greater than or equal to ($\succeq$) that for the original system; (ii) addressing issues related to the use of finite precision arithmetic; (iii) handling release jitter, by assuming the deterministic worst-case and its effect on the release times of jobs.

In 2005, Kim et al. [76] built upon the analysis framework set out by Diaz et al. [54, 55]. They discuss three solutions to obtaining the stationary backlog, an exact solution for the Markov matrix which has a high computational cost, and two approximate solutions. The first approximate solution truncates the matrix; however, the way in which truncation affects accuracy is unresolved. The second approximation involves iteratively computing the backlog over a number of hyperperiods, and examining its convergence. The number of iterations needed to provide a given level of accuracy is however unknown in advance. The evaluation shows that the deadline miss probability computed over the hyperperiod may be considerably smaller than that computed for jobs in the busy period following a critical instant, e.g. via SDTA [60]. The complexity of backlog computation is shown to be $O(j^2m^2)$ *per* job, where j is the number of the job and m is the number of points in the execution time distribution. In their conclusions the authors claim that the method could be indirectly applied to sporadic task systems by modeling them as a periodic system with periods equal to minimum inter-arrival times, and that this would give a safe approximation. We note that this assertion is not correct, as shown by the example given in Section 2.5.

The important problem of *dependences* between the execution times of jobs of the same task and jobs of different tasks was first addressed by Ivers and Ernst [70] in 2009. They presented an analysis that accounts for the effect of unknown *dependences* between the execution times of jobs of tasks in a system using fixed priority preemptive scheduling. The authors give a motivating example which illustrates how dependences can have a marked effect on the response time distribution. Their simple example, which we presented as an exemplar in Section 2, shows that considering a worst-case convolution (i.e. matching the largest values from each distribution) is in general not sufficient to determine the worst-case deadline miss probability. The authors introduce a method based on probability boxes which soundly bounds the response time distribution in the presence of unknown dependences. The approach uses the concept of *copulas* to model the relationship between marginal distributions and the joint distribution of two random variables, and Frechet bounds that give upper and lower bounds on the relationship between the marginal distributions and the joint distribution. Probabilistic bounds are then given on the *sum* of the two random variables. These bounds have been shown to be sound and point-wise tight [151]. The method of Diaz et al. [54, 55] is then lifted to probability boxes, enabling sound and tight bounds on the response time distributions to be computed for systems with unknown dependences between job execution times. Worked examples show that assuming independence when it does not exist can easily result in substantial optimism in the results.

In 2013, Tanasa et al. [140] provided a probabilistic analysis of the response times of messages transmitted via the dynamic segment of FlexRay. The system model assumes that messages are queued for transmission periodically, but are subject to variable queuing jitter, stemming from the task that queues them, and that this jitter may be modelled via some probability distribution. The method determines the deadline miss ratio for each message over the hyperperiod of message periods and the FlexRay cycle. To achieve this it first computes the possible backlog vectors at the end of the time interval by constructing a transition graph linking the backlog vector at the start of the interval to those backlog vectors that can possible stem from it at the end of the interval. We note that this transition graph may be very large depending on the length of the hyperperiod, the number of messages, and the number of possibilities in terms of different values of jitter. The authors use an MILP formulation to compute the transition graph and the vectors. As a second step, the method computes the probability that message instances miss their deadlines on each transition. In the third step, the deadline miss probability is computed by iterating over a number of hyperperiods. This provides an under approximation as the backlog converges towards a stationary value (see the earlier work of Diaz et al. [54]). The method is computationally expensive and the authors implement part of it on a TeslaM2050GPU with 448 cores to help speed up processing.

3.3 Analysis for More Complex Task Models

Following initial work on periodic tasks, researchers have explored more complex task models. The majority of the research published in this area emanates from Cucu-Grosjean and co-authors. This includes work [41] on sporadic task models where inter-arrival times are described by random variables, and subsequently the development by Maxim and Cucu-Grosjean of analysis [106] for tasks with execution times, inter-arrival times and deadlines that are described by random variables. Further work by Maxim et al. [105] explored issues of priority assignment. Note all of the research described in this section assumes that task parameters such as execution times and inter-arrival time are independent.

The first work in this area, by Cucu-Grosjean and Tovar [41] in 2006, considered a model where tasks have constant execution times, but their inter-arrival times are modelled by random variables and have known probability distributions. The authors introduce a method of computing the probabilistic worst-case response time distribution for tasks under fixed priority preemptive scheduling. This model is useful for message streams where the message length is fixed, or varies very little, but the inter-arrival times may vary widely. In 2008, Cucu-Grosjean [40] applied the method to the problem of deriving response time distributions for CAN messages where message arrivals are assumed to be independent and described by probability distributions in a range between some minimum and maximum values. The analysis assumes that the maximum utilisation of the system is less than 1, thus avoiding issues relating to backlog.

In 2011, Maxim et al. [105] investigated three related problems of priority assignment in fixed priority preemptive systems where task execution times are described by random variables. They define the deadline miss probability for a job of a task, and the deadline miss probability[6] for a task, which is effectively the expected deadline miss probability over all of the jobs in some long interval of time, for example the hyperperiod. The three priority assignment problems involve:

(i) Finding a priority assignment that results in the deadline miss probability of each task being below the threshold specified for that task.

(ii) Finding a priority assignment that minimises the maximum deadline miss probability over all tasks.

(iii) Finding a priority assignment that minimises the average deadline miss probability over all tasks.

The authors give an example showing that rate-monotonic priority assignment is not optimal for problem (i) in the case of implicit deadline tasks. Optimal solutions to problems (i) and (ii) are

[6] Referred to as the deadline miss ratio in [105].

given using a greedy approach similar to the optimal priority assignment algorithm of Audsley [15]. A greedy approach is shown to be non-optimal for problem (iii), and a tree-based search is proposed as a potential solution.

In 2013, Maxim and Cucu-Grosjean [106] introduced *exact* probabilistic response time analysis for tasks which may have their worst-case execution times, inter-arrival times and deadlines described by independent random variables. The scheduler is assumed to be fixed priority preemptive. Deadlines are assumed to be constrained, i.e. not greater than the inter-arrival time to the next job of the same task. The authors show that for the task model considered, where incomplete tasks are aborted at their deadline, the worst-case response time distribution of any job of a task occurs for the first job in the synchronous busy period. The method presented is exponential in the number of tasks and the size of the random variables. They note that as the problem is a superset of the non-cyclic Generalised Multi-Frame (GMF) task model [115] there cannot be a pseudo-polynomial time exact test. Re-sampling techniques [109] are shown to be effective in reducing the complexity of the method in practice (see Section 10). The method iteratively computes the response time distribution by considering each preemption following synchronous release. For each preempting job, it convolves the pWCET distribution of that job with the tail of the current response time distribution from the preemption point onwards. The result is scaled by the probability of the job preempting at that time. This is repeated for all possible preemption times for the job, and the resulting distributions coalesced. Iteration then moves on to the next preempting job. Iteration ends when the next possible preemption is later than the final point in the current response time distribution, or the deadline of the task under analysis is exceeded. The probability of a deadline miss can then be found by subtracting the deadline distribution from the final response time distribution, with the probability mass strictly greater than zero giving the probability of a deadline miss. The analysis has been implemented in a publicly available simulator and analysis tool called PanSim [104]. Note, motivation for variable inter-arrival times comes from an automotive application where ultra-sound parking sensors randomise their sampling frequency to avoid the reduced effectiveness that would occur if the same sampling frequency were used by two vehicles reversing back-to-back.

In 2016 Ben-Amor et al. [23] derived probabilistic schedulability analysis for tasks with precedence constraints with execution times described by random variables, scheduled under EDF. They built upon the work by Chetto et al. [38] for tasks with deterministic parameters, deriving an equivalent schedulability analysis for the probabilistic case. This includes proposing a new comparison operator between distributions for probabilistic response times (or execution times) and deadlines that are also described by independent random variables. This comparison is equivalent to the method of subtracting the deadline distribution used by Maxim et al. [106]. The authors note that the greater than or equal to operator ($\succeq$) of Diaz et al. [55] cannot be used for such comparisons, since it would give optimistic results. For example, consider the two distributions $\mathcal{R} = \begin{pmatrix} 1 & 3 \\ 0.9 & 0.1 \end{pmatrix}$ and $\mathcal{D} = \begin{pmatrix} 2 & 4 \\ 0.8 & 0.2 \end{pmatrix}$. Even though $\mathcal{D} \succeq \mathcal{R}$, there is still a non-zero probability that the deadline is missed. This occurs when the response time has a value of 3 and the deadline has a value of 2. Assuming independence, then such a combination has a probability of 0.08 of occurring.

In 2018, Markovic et al. [103] presented a probabilistic schedulability analysis for the limited preemption model where each task is scheduled using fixed priorities and preemption is only permitted at fixed preemption points. A key aspect of this problem is the selection of preemption points for each task from a larger set of possible preemption points. The system model used assumes that each task is divided into sub-tasks, separated by possible preemption points. Further, the execution time of each sub-task and the preemption overhead of allowing preemption at a particular point are represented by pWCET distributions. The authors utilise the approach of Diaz et al. [54], adapted along the lines of existing deterministic analysis for limited preemptive systems, to form a probabilistic schedulability analysis for limited preemption scheduling with fixed preemption points. Further, they propose an algorithm for preemption point selection, with the aim of minimising the deadline failure probability. This algorithm iterates over a number of confidence levels, starting at 1.0 and decrementing by a small step. The confidence level is used to reduce the pWCET distributions for sub-tasks and preemption point overheads to scalar values. This is done by taking the minimum value that does not have a probability of being exceeded that is more than the confidence level. This reduction of the random variables to scalar values enables an existing deterministic method to be used for preemption point selection. Once a set of preemption points have been selected then probabilistic schedulability analysis is used to determine the corresponding probability of deadline failure. The final preemption point selection is the one with the smallest probability of deadline failure found in any of the iterations. We note that by starting with a confidence level of 1.0, the algorithm is guaranteed to always find feasible solutions when the equivalent deterministic approach would do so.

Probabilistic Real-Time Calculus (an extension of Real-Time Calculus [142]) was proposed in 2011 by Santinelli and Cucu-Grosjean [130] with an extended journal version published in 2015 [131]. The authors study a task model where both execution times and inter-arrival times are described by independent random variables. A component model is used, with composability according to an *assume-guarantee* abstraction providing the basis for schedulability analysis. The model describes the resource demand of a task via a probabilistic upper bounding function (or curve). This curve upper bounds the cumulative demand as a function of the interval length t such that the probability that the actual demand exceeds the bound is less than a given probability threshold. Similarly, the resource supply (or service) is described by a lower bounding function (or curve) such that the probability that the actual supply is less than the bound is less than a given probability threshold. Different request and service curves may be obtained for different probability thresholds. The authors provide an algebra using Real-Time Calculus that facilitates the composition of systems from components and associated probabilistic schedulability analysis. The early work from 2011 [130] provides analysis for fixed priority scheduling, with extensions to EDF and hierarchical scheduling given later in 2011 by Santinelli et al. [136]. The journal extension [131] published in 2015 provides detailed proofs for the previous schedulability results. The same component-based formulation was also considered by Khan et al. [74] in 2012 in the context of multiprocessor scheduling, with identical cores, an interconnect (e.g. a TDMA bus), and a fixed partitioning of tasks between cores. They propose a mapping between the level of assurance (ASIL) needed for the safety of a function and its components and the probability threshold used. In 2016, Santinelli [129] proposed a component-based formulation for networks, similar to that previously proposed for tasks and processors [130, 131]. Here, the author considers the probability distribution for message payloads and inter-arrival times. We note that possible dependences between these distributions are not considered. The performance of the network is calculated for the case of AFDX switches implementing FIFO queues.

In 2014, Carnevali et al. [34] proposed computing the probability of deadline misses within a time interval t, based on modelling tasks using Stochastic Timed Petri Nets. They consider fixed-priority non-preemptive scheduling of periodic tasks. Task execution times are modelled via Erlang distributions, the parameters of which are selected to tightly upper bound the pWCET distribution for the task obtained via EVT. This enables regenerative transient analysis based on stochastic state classes to be used to compute the probability that each task misses a deadline within a time t.

3.4 Summary and Perspectives

Probabilistic response time analysis aims to compute either the response time distribution for each job over a relevant interval (such as the hyperperiod for a set of periodic tasks) or the worst-case response time distribution obtained for the worst-case scenario in terms of job releases (for sporadic tasks). Deadline miss probabilities and hence schedulability can then be determined via comparison between response time distributions and deadlines. We highlight three important threads of research in this area:

1. Diaz et al. [54] showed how to compute the probabilistic response time distributions for a set of jobs of periodic tasks taking into account the potential backlog accruing at the end of the hyperperiod. They also proved key properties required to ensure that the results obtained from probabilistic response time analysis are sustainable over-approximations [55].
2. Ivers and Ernst [70] showed that considering a worst-case convolution (i.e. matching the largest values from each distribution) is *not sufficient* to obtain a sound response time distribution when there are dependences between task execution times. They extended the approach of Diaz et al. [54, 55] to tasks with execution times that are related via unknown dependences.
3. Maxim and Cucu-Grosjean [106] proved that for systems with constrained deadlines where incomplete tasks are aborted at their deadline then the worst-case response time distribution is assumed by the first job of each task following synchronous release. Using this result, they derived probabilistic response time analysis for systems where execution times, inter-arrival times, and deadlines are all described by independent random variables.

In a further thread of research, Santinelli and various co-authors [130, 136, 131, 129] explored a probabilistic extension to Real-Time Calculus. We note that this approach relies heavily on the assumption that both the execution times and inter-arrival times of consecutive jobs are independent.

Two key problems that remain with probabilistic response time analysis are the tractability of the analysis for task sets of practical sizes (see Section 10 for initial work in this area), and issues relating to dependences between execution times.

4 Probabilistic Analysis assuming Servers

In this section, we review probabilistic schedulability analysis for tasks that are run within servers. Servers provide a partitioning of the available processor time and thus isolate the execution of each task from interference due to other tasks running on the same processor. This is useful in simplifying the analysis and in removing concerns regarding dependences between the execution times of jobs of different tasks.

4.1 Analysis for Server-based Systems

The majority of the research in this area has been published by Abeni and co-authors, including the seminal initial work [2, 3] from the late 1990s that also introduced the Constant Bandwidth Server, subsequent research that considers execution times [4, 121] and also inter-arrival times [6, 99, 122] described by independent random variables, and more recent work [59, 5] that considers dependences between the execution times of jobs of the same task.

In 1998 and 1999, Abeni and Buttazzo [2, 3] provided analysis for soft real-time tasks that have (i) variable execution times according to some probability distribution and fixed inter-arrival times, or (ii) fixed execution times and variable inter-arrival times according to some probability distribution. Jobs of a task are executed under a Constant Bandwidth Server (also introduced in the paper). This ensures independence between tasks, since each task can only utilise the capacity of its own server. The authors provide statistical guarantees determining the probability that jobs of a task will meet their deadlines. This is achieved by modelling the Constant Bandwidth Server as a queue and determining the stationary solution of the Markov chain for those cases where the queue is stable. Conditions for stability of the queue comprise ensuring that the average utilisation does not exceed the server bandwidth. The output is the probability distribution of the response time by which the jobs will finish. This can then be used to determine the probability of meeting deadlines and hence the quality of service of the soft real-time tasks. Subsequently in 2001, Abeni and Buttazzo [4] extended their earlier work [2] on variable execution times, by relaxing the assumption that the server period must exactly divide the task period. They show how to handle inter-arrival times that are expressed as a probability distribution by approximating this distribution in accordance with the value of the server period. With this approximation, larger server periods generally introduce more pessimism, but have the advantage of resulting in a smaller number of context switches.

The more complex model where tasks have both execution times and arrival times modelled via random variables with known probability distributions was addressed by Kaczynski et al. [72] in 2007. They assume that there is no minimum time between arrivals for an aperiodic task and thus motivate the use of servers to prevent unbounded interference on other tasks. The authors extend the method introduced by Diaz et al. [54] (see Section 3.2) to this model, using a server for each task.

In 2012, Abeni et al. [6] considered tasks with both execution times and inter-arrival times described by probability distributions. Jobs of these tasks are scheduled via resource reservations where each task has its own server. The authors present an efficient algorithm to compute a bound on the probability of deadline failure. As part of the method, the inter-arrival time distributions are soundly approximated by simpler distributions limited to multiples of the server periods. Further, incomplete execution time distributions can be handled, provided that the remaining probability for values over some limit is known. The method provides a valid bound provided that the expected utilisation of each task does not exceed the capacity of its server. Evaluation shows that the bound produced is sound and that the over-approximation is small compared to exact analysis, which takes over 1000 times longer in some cases. In further works in 2012, Manica et al. [99] and Palopoli et al. [122] also considered tasks with both execution times and inter-arrival times described by probability distributions. Manica et al. [99] derived a model for tasks scheduled by a Constant Bandwidth Server. They show that the model takes the form of a Quasi-Birth-Death Process that can be solved using a numerical algorithm to determine a bound on the probability of a deadline miss. Palopoli et al. [122] used a conservative model for the evolution of a periodic task scheduled via a periodic server to construct a closed-form bound on the probability of a deadline miss. The model again takes the form of a Quasi-Birth-Death Process which enables efficient numerical computation of the probabilities required. For tasks with implicit deadlines, the model is further simplified allowing analytical calculation of a lower bound on the probability of meeting the deadline. This bound enables server bandwidth to be appropriately sized in order to meet requirements on the maximum permitted probability of deadline failure. Evaluation shows that the approximation error in the closed-form solution is high when the server bandwidth is close to the expected utilisation of the task, but reduces to acceptable levels as the bandwidth increases. In the former case, the probability of a deadline miss is in any case high ($> 60\%$), and so the cases where the approximation error is large are unlikely to be those that are of practical interest. An important limitation of the methods proposed by Manica et al. [99] and Palopoli et al. [122] relates to their pessimism due to the fact that the model used neglects the server budget that may be shared between consecutive jobs of a task.

In 2016, Palopoli et al. [121] considered periodic tasks with execution times described by probability distributions. Jobs of these tasks are scheduled via resource reservations where each task has its own Constant Bandwidth Server. They model the evolution of a task scheduled via a resource reservation as a Discrete-Time Markov Chain that takes the form of a Quasi-Birth Death Process. The outcome of the analysis is an expression for the steady state probability of meeting the deadline. This is used to construct a numerical algorithm and also an analytical bound that can be used to determine the probability of meeting the deadline. The evaluation compares the performance of the analytical bound, the numerical algorithm, and the authors' previous method [6]. The results show that the analytical bound has a runtime that can be orders of magnitude faster, and provides a high degree of accuracy in the cases where the bandwidth provided by the server is sufficient such that the probability of meeting the deadline is high ($> 90\%$). Further, the performance of the numerical algorithm can be tuned using a scaling factor, trading-off between runtime and accuracy. The effectiveness of the approach is demonstrated using a case study involving the playback of two video streams.

In 2017, Frias et al. [59] noted that many real-time applications exhibit a wide variation in execution times and are resilient to occasional deadline misses. Such systems have previously been afforded probabilistic schedulability guarantees based on the use of a resource reservation scheduler and the assumption that execution times are independent and identically distributed (i.i.d.). They consider robotic applications where the execution time of the vision algorithms depends on the complexity of the scene while also exhibiting a random behaviour due to a random element of the search. Thus the i.i.d. assumption does not hold. Instead execution times may be described via a Hidden Markov Model (HMM). The authors show how to identify the different modes of execution, and for each mode the distribution of execution times, which are independent within the mode. The Markov Computation Time Model (MCTM) used enables an accurate estimation of the probability of missing deadlines. The effectiveness of the approach is evaluated on a robot vision case study (a lane detection algorithm for a robotic car) where it is shown to provide accurate results. By comparison analysis based on an i.i.d. assumption leads to significant and optimistic under-estimation of the probability of missing deadlines. In a further short paper in 2017, Abeni et al. [5] show how the MCTM can be derived from a set of execution time measurements using the theory of HMM. Here, a task is modelled as having a number of states (modes) with probabilities for transitions between them. In each state the execution time is described by a different probability distribution, while within a state, the execution times are i.i.d. The method is able to estimate the number of modes from the raw execution time data. Assuming the number of states is known, then existing techniques (Baum-Welch algorithm) can be used to estimate the transition probability matrix and the probability distributions for the different modes. The authors therefore propose a method of finding the correct number of states via a gradient-like approach based on cross validation of the likelihood. The approach is validated on the robot vision case study (discussed above). It is also shown to recover a single state, as expected, for standard i.i.d. processes.

A software tool called PROSIT that supports the design and analysis of real-time systems based on the idea of probabilistic deadlines was initially described by Palopoli et al. [120] in 2015, with a more extensive description of a later version of the tool given by Frias et al. in 2018 [146]. PROSIT provides analysis for fixed priority preemptive scheduling of periodic tasks with execution times described by random variables, based on the analysis of Diaz et al. [54, 55]. It also provides an analysis of server-based scheduling of periodic and aperiodic tasks with execution times and potentially also inter-arrival times described by random variables. Here, the execution times can either be i.i.d. or a Markov process. The tool implements the various analyses proposed by Palopoli et al. [122] and Frias et al. [59] . In the case of server-based scheduling, PROSIT can be used to optimise the reservations (server budgets) for periodic tasks so as to optimise the global quality of service.

4.2 Summary and Perspectives

In this section, we reviewed work on probabilistic schedulability analysis for tasks that are run within servers that provide a partitioning of the available processor time. The use of servers has the advantage that the execution of each task is isolated from interference by other tasks; however, it has the disadvantage that jobs requiring a long execution time cannot directly make use of spare capacity freed up by jobs of other tasks that have a shorter than expected execution time. Arguably, this removes one of the main advantages of probabilistic schedulability analysis, which can otherwise take advantage of that fact that it is unlikely that jobs of multiple tasks will simultaneously require their worst-case or near to worst-case execution times.

We highlight the important thread of research initiated by Abeni et al. [2, 3] on probabilistic schedulability analysis for tasks that are run within servers, and the subsequent extensions of the analysis by Abeni et al. [6], Manica et al. [99], and Palopoli et al. [122] to tasks with both execution times and inter-arrival times described by independent random variables. While the use of servers removes issues of dependences between tasks, the issue of dependences between jobs of the same task remain. In practice, it is often the case that jobs of the same task exhibit dependences between their execution times, due to input variables that change only slowly over time, as well as execution starting from similar software and hardware states. Only in the recent work of Frias et al. [59] has this issue of dependences begun to be investigated. The analysis derived by Palopoli et al. [122] and Frias et al. [59] has recently been implemented in a software tool called PROSIT [146].

5 Real-Time Queueing Theory

In this section, we review research based on Queuing Theory, which is the mathematical study of queues with the aim of deriving information about queue lengths and waiting times. Problems in queuing theory are typically described in Kendall's notation [73] $A/B/C$ where A is the distribution of arrival times, B is the distribution of service (execution) times, and C is the number of servers. For example M/G/1 implies a Poisson or Markovian (M) arrival process, a General (G) execution time distribution, and a single (1) server.

5.1 Analysis based on Real-Time Queuing Theory

The majority of the research in this area has been published by Lehoczky and co-authors, including the seminal paper [83] on Real-Time Queuing Theory from 1996, and its later mathematical formalisation [56]. Subsequent work has investigated the impact of quantisation on EDF queues [67, 154], and extended the analysis to systems where jobs can be reneged, i.e. the remaining work of a job is discarded if its deadline is reached before it has completed execution [79].

The origins of work in this area can be traced back to the 1950s. In 1957, Barrer [20] considered the problem of queues with jobs that arrive according to a Poisson process, have execution times that follow an exponential distribution, i.e. an M/M/1 queue in Kendall's notation [73], and have a single fixed deadline. This work computes the ratio of the rate at which jobs miss their deadlines and are discarded to the arrival rate. In 1988, Panwar et al. [123] considered the problem of single-server queues with deadlines on jobs which become known on their arrival. The aim here is to schedule the jobs so that the fraction of jobs served within their deadline is maximised. The authors show that the Shortest Time to Extinction (STE) policy is optimal for this problem, for a class of non-preemptive M/G/1 queues that are work-conserving, i.e. do not permit inserted idle time, and where execution times are i.i.d random variables. The STE policy schedules the ready job with the earliest deadline that is still in the future. Jobs with expired deadlines are discarded.

They also showed that when inserted idle time is permitted, then policies from the STEI class are the best possible. (An STEI policy either schedules the job with the earliest unexpired deadline, or inserts idle time).

Real-Time Queueing Theory was introduced in 1996 by Lehoczky [83] as a means of analysing soft real-time systems where tasks have execution times that exhibit a large amount of variation. In such systems, using deterministic upper bounds on response times would lead to the system being significantly under-utilised on average. The method extends heavy traffic queueing theory utilizing the fact that the number of tasks in the queue behaves like reflected Brownian motion under heavy traffic. The theory can estimate the *lead-time* profile (meaning the time to go until the deadline) for the tasks in the queue, based on the distributions of arrival time, execution time, and deadline, and the queueing policy (EDF and processor sharing policies were considered). The fraction of missed deadlines can be derived from the queue-length dependent lead-time profiles and the distribution of queue lengths obtained via queueing theory. We note that the heavy traffic phenomenon occurs only for processor utilisations close to 1, which implies long queues and which may imply large latencies. Real-Time Queueing Theory was subsequently placed on firm mathematical foundations by Doytchinov et al. [56] in 2001. The heavy traffic constraint was subsequently relaxed by Zhu et al. [154]. Thus the main limitation in applying real-time queueing theory is that the probability distributions for all tasks must belong to the same family.

In 2002, Hansen et al. [67] examined the effect that quantisation has on EDF queues with a stochastic traffic model (arrival times, execution times and deadlines). Instead of using precise relative deadlines, this method quantises them, assigning tasks the next smaller deadline from a small set. The authors found via simulation that using just 3 bits (8 quantisation bins) was sufficient to obtain performance for Q-EDF close to that of EDF. They found that a log bin quantisation was more effective than a uniform one, since the log bins provide better granularity for small deadlines. The small number of bits required is important for network scheduling where this information takes up bandwidth. Also in 2002, following on from the work of Hansen et al. [67], Zhu et al. [154] aimed to optimise the set of quantisation bins used by Q-EDF. Using Real-Time Queueing Theory [83] they proved properties of a deadline distribution that would result in the worst-case behaviour for Q-EDF. It has tasks with deadlines that are at either end of the quantisation bins, leading to the maximum priority inversion. Hence they found that to minimise the maximum number of deadline misses for an arbitrary distribution of deadlines with the same minimum, maximum, and mean, one should create the bins by dividing the deadline range in a uniform way. Further they showed that with uniform bins 3-bits (8 bins) were sufficient for the performance of Q-EDF to converge to that of EDF in practical cases.

An alternative scheduling policy to EDF was examined by Gromoll and Kruk [63] in 2007. They considered processor sharing where all active jobs in the queue receive an equal share of the available processor time, and used heavy traffic queueing theory to obtain approximations for the lead-time profile and the profile of times in the queue.

The analysis given by Lehoczky [83] and Doytchinov et al. [56] assumes that late jobs continue to execute. Complementary work by Kruk et al. [79] in 2011 presented a heavy-traffic analysis of the behaviour of a single queue under an EDF scheduling policy adapted so that when the deadline of a job is reached, if the job has not yet completed, then its remaining work is discarded (referred to as *reneged*). The performance metric used is the fraction of work that is lost (reneged) due to missed deadlines. The authors show that this metric is minimised by using EDF scheduling. The evolution of the lead-time distribution of jobs in the queue is described by a measure-valued process. The heavy traffic limit of this process is shown to be a deterministic function of the limit of the scaled workload process which in turn is identified to be a doubly reflected Brownian motion. The fraction of reneged work in a heavily loaded system and the fraction of late work in the corresponding system without reneging are compared using formulas based on the heavy traffic approximations. These formulas closely match simulation results, which show that the amount of work lost (i.e. reneged or late) due to deadline misses is reduced by a factor of one to two orders of magnitude if remaining work is discarded at the deadline.

5.2 Summary and Perspectives

In this section, we reviewed work on real-time queuing theory. The analysis used requires that the traffic intensity / processor utilisation is close to 1, where the queues are long, and the latencies may be high, which might not be acceptable in real systems. We note that although the theory does not characterise system behaviour at lower traffic intensities, the probability of deadline misses decreases as the traffic intensity decreases, all other parameters being equal. Thus the results for heavy traffic can serve as bounds on the behaviour at lower intensities.

We highlight the important thread of research initiated by Lehoczky [83] and Doytchinov et al. [56] on real-time queueing theory and its subsequent extension by Kruk at al. [79] to an EDF scheduler that discards any incomplete jobs when their deadlines expire. The main limitation of real-time queuing theory is that it requires job execution times, arrival times, and deadlines to be independent, which may not be the case in real systems.

6 Probabilities from Faults

In this section we review work on probabilistic schedulability analysis where random variables are used to represent the occurrence of some form of fault. Initial research focused on faults occurring in tasks running on a processor and the impact of fault recovery operations; however, the main thread of research in this area concerns Controller Area Network (CAN), a broadcast bus that is widely used in the automotive industry for in-vehicle networks.

6.1 Analysis of Fault Recovery on Processors

Research into probabilistic schedulability analysis for systems with faults was initiated by Burns and co-authors [33, 30, 27, 26], who considered the impact of fault recovery operations on tasks running on a processor.

In 1999, Burns et al. [33] addressed fault tolerant hard real-time systems. They introduced the concept of a probabilistic guarantee on schedulability for a fixed priority preemptive system. This is achieved by incorporating the cost of fault recovery (e.g. re-running a task, recovery block, executing since the last check point) into the analysis, along with the minimum time between faults. It is assumed that a single fault causes a detectable error in a single task. Sensitivity analysis is then used to determine a threshold on the minimum time between faults that can be tolerated by the system, while still meeting all deadlines. This threshold and the lifetime of the system are then used as parameters in a fault model which determines the probability that no two faults will occur closer together than the threshold during the lifetime of the system. Faults are modelled as a *homogeneous Poisson process* which enables this probability to be analytically computed, or upper and lower bounded using a simpler approach. We note that the approach is potentially somewhat pessimistic, since the schedulability analysis considers only a critical instant corresponding to synchronous release, whereas over much of the lifetime of the system the phasing of tasks may be such that a higher fault rate could be tolerated. Nevertheless, the method provides a valid lower bound on the probability that all jobs will meet their deadlines over the lifetime of the system.

Further work by Burns et al. [30] in 2003 looked at how conventional response time analysis [71, 14] could be extended to provide probabilistic guarantees. They remarked on the limitations of a deterministic approach: (i) in fault tolerant systems, fault arrivals are inherently stochastic, (ii) more complex applications have widely varying execution times, (iii) even for simple applications, advanced processor architectures (with cache, pipelines etc.) lead to wide variability in execution times. The authors present two methods of accounting for fault arrivals in response time analysis. The first method computes the maximum schedulable periodic fault rate from modified response time equations using sensitivity analysis [124]. An upper bound can then be derived on the probability of failure due to faults arriving closer together than the maximum tolerated rate. The second method computes the response time assuming no faults and then determines the maximum number of faults that could occur in that time with a probability higher than a specified threshold on the probability of failure. This number of faults is then added and the extended response time computed. This process iterates until it converges or the deadline is exceeded.

In 2004, Broster and Burns [27, 26] presented a simple method for computing probabilistic worst-case response times when there is just one task that has arrivals described by a probability distribution. This method is based on the approach taken to probabilistic modelling of fault arrivals on CAN [28] (see Section 6.2). The method works by computing the response times R_i^0 to R_i^m for task τ_i assuming 0 to m random arrivals of the higher priority task with random behaviour. It then determines the probability that each response time occurs. For example to obtain response time R^2, there must be exactly two arrivals within that interval, excluding the case where there are no arrivals in R_i^0 or exactly one arrival in R_i^1, since the latter cases result in a smaller response time. Once the probability of each response time has been computed, then the probability of deadline failure can be determined.

Also in 2004, Kim and Kim [75] presented a method of probabilistic schedulability analysis for harmonic task systems under Dual-Modular Temporal Redundancy (DMTR). DMTR utilises two processors to simultaneously run duplicates of each task, with execution split into sub-tasks that execute in defined time slots. At the end of each time slot, comparisons are made between the outputs of the sub-task duplicates. If they are the same, then processing proceeds with the next sub-task, otherwise the duplicates are re-executed and comparisons made between the four outputs (two new and two previous). If two or three matching outputs are found, then processing can proceed, otherwise the duplicate sub-tasks are executed again. A maximum of three time slots can be used for each sub-task, corresponding to three separate temporary data areas available to the processor. The authors derive a probabilistic schedulability analysis for the DMTR model, assuming transient faults that occur according to a Markov model with arrivals according to a Poisson process and an exponentially distributed duration. The approach is formulated using state transition probability matrices. The authors use their formulation to determine the optimal number of sub-slots to use given a fixed checkpointing overhead. They note that there are issues with the complexity of the analysis and restrict their examples and evaluation to three tasks.

In 2011, Aysan et al. [18] provided a probabilistic schedulability analysis for tasks, running under fixed priority preemptive scheduling, that are subject to faults in the form of error bursts. The model of recovery is that errors detected at the end of a task lead to the subsequent execution of an alternate, or the re-running of the original task. The error model analysed consists of bursts of errors occurring with a minimum inter-arrival time (T_E), with each burst having a duration described by a probability distribution and a number of errors within it separated by an intra-burst minimum inter-arrival time. Sensitivity analysis is used to compute the minimum value for T_E for each burst length in the distribution such that the task set remains schedulable if error bursts of that duration are separated by at least T_E. For each burst length and associated minimum value of T_E, the probability of the system remaining schedulable over its lifetime is computed, based on a Poisson distribution of events (burst arrivals). This information is then used to compose the overall probability of the system remaining schedulable for its lifetime. The analysis takes account of the fact that errors may impact both the task of interest (whose schedulability is being checked) and higher priority tasks that may execute within its response time.

Error occurrence described by a two-state discrete Markov model was considered by Short and Proenza in 2013 [138]. This model specifies two states G and B (Good and Bad) and the probabilities for the transitions between them. Further, associated with each state is the probability of error occurrence while in that state. This model is a general one that can describe bursts of errors with probabilistic inter-arrival times between the start of each consecutive burst, and probabilistic inter-arrival times between the errors within a burst. The authors derive an efficient closed-form bound on the maximum number of errors occurring in an arbitrary time interval t according to this model, subject to a required confidence level or probability r. This bound can be used to provide a function giving the number of errors that must be tolerated as a function of t for the system to operate with a probability of failure that is no more than $1 - r$. The authors show how this function can be incorporated into standard schedulability analysis for EDF providing a "one-shot" analysis that can determine if a simple fault tolerant EDF-scheduled system can operate with a probability of deadline failure that is lower than $1 - r$. We note that a similar approach could be applied to fixed priority scheduling by integrating the bound into response time analysis.

In 2016, Santinelli et al. [134] discussed the idea of a C-Space (the space of task execution times that lead to a feasible, i.e. schedulable system) and how it can be adapted to a probabilistic model of execution times. The authors consider separate pWCET distributions resulting from (i) no-fault (or *LO-safe*) and (ii) fault (or *HI-safe*) behaviour for each task. These pWCET distributions are used to determine discrete execution time budgets with a probability of being exceeded of, for example, 10^{-9} for each task, assuming (i) no-fault and (ii) fault behaviour. The C-Space can then be used to indicate the probability of each combination of execution time budgets being exceeded (assuming i.i.d. execution times), and hence whether a particular set of execution time budgets can be considered feasible for a given combination of behaviours that have to be accommodated (for example task τ_1 considering LO-safe behaviour with no faults, and task τ_2 considering HI-safe behaviour with faults).

6.2 Analysis of Fault Recovery on CAN

Controller Area Network (CAN) is a broadcast bus used for in-vehicle networks. Messages sent on CAN have a bounded length and are transmitted according to a fixed priority non-preemptive scheduling policy, with the message ID also representing the message priority (see Davis et al. [50] for full details of the protocol and its analysis). CAN has strong error checking mechanisms that can detect faults that result in bit-errors on the bus and so cause message corruption. The protocol ensures that any message that fails to be transmitted correctly will be later re-sent. Hence faults result in an additional load on the bus due to re-transmissions, which can potentially result in deadline misses. Even though CAN is a deterministic protocol and the messages have bounded lengths, the random occurrence of faults means that analysis techniques are needed that can determine the probability of messages failing to meet their deadlines. A significant thread of research in this area began with the work of Navet et al. [117] in 2000. This was built upon by Broster et al. [28, 29], and Davis and Burns [48], and later adapted to more complex message arrival functions by Axer et al. [17].

In 2000, Navet et al. [117] proposed a fault model for messages on CAN based on random arrivals, where faults are assumed to occur according to a Poisson distribution. They introduce the idea of a *tolerable error threshold*, corresponding to the maximum number of errors that a message can tolerate before it becomes unable to meet its deadline. This threshold is then used in a calculation of the worst-case deadline failure probability (WCDFP).

Subsequently, in 2002, Broster et al. [28, 29] extended the work of Navet et al. [117], correcting and improving upon the WCDFP analysis. (We note that this work is based on early analysis of CAN that contains a number of flaws, but the method could easily use the correct equations that were derived by Davis et al. [51] in 2007). The analysis works by deriving the probability $p(R_m|K)$ that a worst-case response time of $R_m|K$ occurs, where $R_m|K$ corresponds to the worst-case response time for message m when exactly K faults occur between it being queued for transmission and transmission completing, i.e. within the response time of the message. The calculation of $R_m|K$ assumes the worst-case scenario, i.e. the maximum delay due to blocking, maximum interference from higher priority messages, and maximum bit stuffing. The probability that K faults occur in a given time interval is obtained from the Poisson distribution of faults. The probability $p(R_m|K)$ is computed for values of K from zero to the maximum number of faults that the message can tolerate without missing its deadline. The sum of these probabilities lower bounds the probability that the message will be successfully transmitted by its deadline, hence subtracting this sum from 1 gives an upper bound on the worst-case deadline failure probability. We note that due to the worst-case assumptions in the response time calculation, this WCDFP may be significantly larger than the actual probability of deadline failure averaged over a large number of instances of the message. In 2012, Axer et al. [17] extended the approach of Broster et al. [28] to more complex arrival functions for messages, including the case where messages have arbitrary deadlines.

The work of Broster et al. [28, 29] was improved upon by Axer and Ernst [16] in 2013. They considered the probabilistic schedulability analysis of messages on CAN assuming a Poisson distribution of faults. They present a method of probabilistic response time analysis based on the use of probability distributions representing queueing delays, busy period lengths, and response times. The key idea is to represent the worst-case total transmission time for each message including its re-transmission due to faults as a probability distribution, based on the probability of k faults occurring *within* transmission of that specific message. This is possible since the Poisson fault model is memoryless. The schedulability analysis derives from the deterministic response time analysis for CAN [51] adapted to consider probability distributions. The authors show how to compute the longest priority level-i busy period that can occur with a probability that is above some small threshold of interest. The response time distributions for each message of priority i in the busy period are then computed using a process of convolution and splitting. An upper bound on the probability that the response time of any message of priority i in the busy period will exceed an arbitrary time t (e.g. its deadline) can then be obtained from these distributions. Evaluation shows that the results given by the analysis are very close to the empirical distribution obtained via Monte Carlo simulation. Further, the probability of failure using typical fault rates for CAN is approximately one order of magnitude better than can be obtained via the analysis of Broster et al. [28, 29]. The reason being that the latter approach pessimistically assumes that the largest possible re-transmission time (for any higher priority message) occurs on every fault.

In 2009, Davis and Burns [48] introduced algorithms that determine Robust Priority Assignments (RPA) [47] for messages sent over CAN. They also investigated a probabilistic variant of this problem. This work builds on prior analysis of the worst-case deadline failure probability (WCDFP) for CAN messages in the presence of faults by Navet et al. [117] and Broster et al. [28, 29]. The authors derive a Probabilistic RPA algorithm which determines a robust and optimal priority ordering, in the sense that it returns a priority ordering which minimises the maximum WCDFP over all messages, provided that a schedulable priority ordering exists. The algorithm builds on the optimal priority assignment algorithm of Audsley [15]. A case study example shows that using the Probabilistic RPA algorithm can result in a worst-case failure rate that is orders of magnitude better than that obtained using deadline monotonic priority assignment (e.g. a failure rate of 1 in 28,500 compared to values in the range 1 in 500 to 1 in 1000).

With CAN, the physical layer employs *bit stuffing* to ensure that there are enough transitions in polarity to maintain synchronisation between the nodes on the network. Thus within each message transmission, a bit of opposite polarity is inserted after every five consecutive bits of the same polarity. This increases the transmission time of the messages. In 2003, Nolte et al. [119] provided a probabilistic worst-case response time analysis for messages on CAN. They used distributions of the number of stuff bits, as opposed to worst-case values, to calculate probabilistic response times based on the critical instant. The authors assume independence between the distributions of the number of stuff bits for different messages and between those numbers for instances of the same message. We note that it is unlikely that such independence would exist in practice. For example consider the situation at start up, when the vehicle is not moving. Many of the signals in the CAN messages may be at their default values e.g. zero, which incur a large number of stuff bits. Further many values (temperatures, pressures etc.) change only slowly over time, thus it is reasonable to expect a strong correlation between the values in one instance of a message and the next, and hence a strong correlation between the numbers of stuff bits.

A probabilistic analysis for CAN messages and end-to-end latencies in an automotive system was presented by Zeng et al. [153] in 2009. They build upon the basic approach of Diaz et al. [54, 55] (see Section 3.2), with a number of approximations and adaptations for distributed systems connected via CAN. Task execution times are assumed to be independent and described by probability distributions. Similarly, the transmission times of CAN messages are also assumed to be independent and described by probability distributions, in this case accounting for the varying levels of bit-stuffing assuming variable message contents. The key approximation introduced for CAN involves handling the lack of synchronisation between messages sent by different ECUs. (The entire system is not simply periodic). This is done by approximating all messages sent by a remote ECU via a single *characteristic message* that has a probability distribution and values for its transmission time equating to the number of messages of priority i or higher that may be released at the same time. For example if there are two messages with periods of 10 and 40 sent by the ECU, then the characteristic message will have a period of 10 (the greatest common denominator) and a probability distribution indicating 1 message with a probability of 0.75 and 2 messages with a probability of 0.25. The authors note that this introduces some inaccuracy into the analysis and it may be quite pessimistic for long intervals of time. Further, there is also the potential for optimistic (i.e. unsound) results. The analysis is, however, intended as an approximation to be used in design space exploration rather than as an upper bound. The characteristic message is given an offset and random jitter to account for the lack of synchronisation between messages transmitted by different ECUs. Messages sent by the same ECU as the message under analysis are treated as individual messages, since their phasing with respect to the message of interest is known. Blocking due to lower priority messages is also accounted for by assuming that the probability of such messages being transmitted is uniform over the hyperperiod. Again, the authors note this is a potential source of inaccuracy. The analysis method follows that of Diaz et al. [54, 55], first the stationary backlog is computed at the start of the hyperperiod (of messages on the ECU that transmits the message under analysis), then the backlog at the release time of each message instance is computed, and finally, the response time distribution of each message instance within the hyperperiod. These are averaged to obtain the response time distribution of the message. The evaluation considers a 69 message case-study based on an experimental vehicle. The results show that the stochastic analysis provides results that are close to those obtained via simulation averaged over 10^8 different relative phasings. The analysis is subsequently extended to end-to-end latency (i.e. tasks communicating across multiple ECUs and two CAN buses). Again, the analysis results are a close fit to those obtained via extensive system level simulation. The key advantage of the stochastic analysis over simulation is its speed e.g. 8 seconds of analysis versus 20 hours of simulation. This means that the analysis is much better suited for use in design space exploration.

Building on their prior work on probabilistic schedulability analysis for tasks [18], in 2012 Aysan et al. [19] derived probabilistic schedulability analysis for CAN under a general fault model. This model considers fault bursts of a duration described by a probability distribution. The analysis proceeds in three steps: (i) For each potential burst duration in the distribution, sensitivity analysis is used to determine the minimum inter-arrival time T_E^{burst} of errors within that burst such that the system remains schedulable. (ii) An upper bound is computed on the probability of a smaller inter-arrival time than T_E^{burst} occurring within the burst for each duration. These bounds are then used to determine an upper bound on the probability that the minimum tolerable inter-arrivals times for errors are violated for all potential fault durations over the mission duration. (iii) Finally, the probability of unschedulability is computed from the probability of two bursts occurring too close together and the probability of the minimum tolerable inter-arrival time of errors within a burst being violated.

6.3 Summary and Perspectives

In this section we reviewed work on probabilistic schedulability analysis where random variables are used to represent the occurrence of some form of fault. Beginning with initial work by Burns et al. [33] in 1999, we can trace an important thread of research providing probabilistic schedulability analysis for fixed priority systems, in particular Controller Area Network (CAN), under an assumed fault model [117, 28, 29, 48, 16, 17]. This work derives effective estimates of the worst-case probability of deadline failure, and provides the tools needed to assign message priorities in such a way as to make the system as robust as possible to the occurrence of faults. The main issue with this line of research is whether the fault models used reflect reality; however, the method is sufficiently flexible to incorporate any reasonable fault model where the probability of some number of faults is monotonically non-decreasing in the length of the time interval considered.

7 Statistical Analysis of Response Times

Previous sections reviewed work on probabilistic schedulability analysis based on analytical models of the system. By contrast, in this section we review work that takes a statistical approach, treating the system as a "black box" and making observations of response times from which an estimation of the response time distribution and hence deadline miss probabilities can be derived.

One of the key methods used in this thread of research is Extreme Value Theory (EVT). An overview of the use of EVT in measurement-based probabilistic timing analysis can be found in the companion survey [52], with more detailed information given in Stuart Coles' textbook on the subject [39]. Here, we provide a brief synopsis, focusing on the Extreme Value Theorem (or Fisher–Tippett–Gnedenko theorem). This theorem states that if the normalised maximum of a sequence of i.i.d. random variables converges, then the limit distribution belongs to either the Gumbel, Frechet, or reversed Weibull family of distributions. In practice, EVT may be applied using the *Block Maxima* method as follows: (i) obtain a representative sample of observations (e.g. response times), (ii) check using appropriate statistical tests that the sample of observations collected is analysable using EVT, (iii) divide the sample into blocks of observations of a fixed size, and take the maxima for each block, (iv) fit a *Generalised Extreme Value* (GEV) distribution (i.e. reversed Weibull, Gumbel, or Frechet distribution, depending on the shape parameter) to the distribution of the maxima, (v) check the goodness of fit between the distribution of the maxima and the fitted GEV distribution. The GEV distribution so obtained then approximates (estimates) the distribution of the *extreme values* of the sampled distribution. We note that if the underlying (measured) distribution is badly behaved, then the normalised maximum may not converge to any of the limit distributions, in which case the method is not applicable. This can be determined by appropriate statistical tests.

7.1 Statistical Estimation

A number of authors have sought to apply statistical methods to estimate response time distributions and deadline miss probabilities. The main thread of research in this area comes from Lu, Nolte, and their co-authors [96, 97, 94, 95] with a recent investigation into the soundness of such approaches by Maxim et al. [111].

The theory of *Large Deviations* [144] was applied by Navet et al. [116] in 2007 to the problem of estimating the mean or the sum of the response times of a series of aperiodic jobs. This method makes use of frequency histograms of response times that are obtained via measurement. It assumes that the response times of jobs of an aperiodic task are i.i.d. The authors note that this is not the case with the response times of periodic tasks, since the interference from other tasks follows a pattern over the hyperperiod due to the release times of higher priority tasks. The method provides an estimation of the probability that the mean response time of a sequence of n jobs of the task will exceed some value x, for all values of x.

In 2010, Lu et al. [96] introduced a method of estimating probabilistic worst-case response times (pWCRT) using Extreme Value Theory (EVT). They record observations of response times, obtained from simulation. EVT is then applied to these observations, using the Block Maxima method, with the distribution of the maxima fitted to a Gumbel distribution using a χ-squared test. The authors present an algorithm which searches for an appropriate block size to use, while enforcing a minimum of 30 blocks. The results are compared to those obtained via a Monte-Carlo search (i.e. keeping the largest response time found from a set of randomised simulations) and also via meta-heuristic search applied on top of Monte-Carlo simulation. The evaluation considers three system models (M1-M3) indicative of those used in robotic control systems. Here, tasks exhibit strong dependences through asynchronous message passing, shared global variables, and runtime changes to task periods and priorities. The system models used in the evaluation vary in complexity. The validation model based on M1 is amenable to conventional response time analysis techniques, and so an exact worst-case response time could be determined. By contrast, M3 has intricate dependences via message passing, global shared variables, and changes in task periods and priorities. The evaluation results show that the EVT-based approach requires far fewer simulation runs (approx. 6% as many) to produce meaningful results compared to the Monte-Carlo and search-based methods. Following on from this work, Lu et al. [97, 94, 95] refined the method, using a form of simple random sampling to break dependences between observations. They also sought to ensure that the pWCRT value returned by their tool (RapidRT) for a given probability of exceedance is an upper bound with an appropriate level of confidence. This is done by repeating the process of obtaining observations and applying EVT n times to produce n pWCRT distributions. The set of values at a probability of exceedance of 10^{-9} from each of these distributions is then checked to see if it complies with a normal distribution. If so, the pWCRT value returned is the one that corresponds to the desired level of confidence (e.g. $3\sigma \approx 99.7\%$). In their final work in this area, Lu et al. [95] evaluated their method using a case study based on an industrial robotic control system with the results compared against a state of the art method based on using meta-heuristic search to guide Monte Carlo simulation to determine parameters that will lead to long response times. Note such simulation requires a detailed model of the system. Four levels of system complexity were explored containing from 40-60 tasks, 7-12 queues, and in the case of the most complex system, run-time priority and period changes, unbounded message passing, and task offsets. The proposed method was shown to bound the estimates obtained via meta-heuristic search and Monte Carlo simulation, with no more than 15% pessimism.

Subsequent work in 2013 by Liu et al. [91] applied EVT to the problem of estimating the worst-case response times of messages on a CAN bus. Due to the scheduling policy used, the distribution of observations and their maxima show multiple peaks. Such distributions are difficult to analyse, since they cannot be fitted to the known EVT distributions. To address this problem the authors use a filtering method which aims to reduce the distributions to single peaks by discarding observations below a threshold. (This threshold is set such that the mean value of observations above the threshold is greater than or equal to their median value). Evaluation shows that the method provides results that are only a few percent pessimistic compared to response time analysis for CAN [50] using computed worst-case values.

The soundness and precision of applying statistical techniques to determine the probabilistic worst-case response time (pWCRT) distribution of tasks was investigated by Maxim et al. [111] in 2015. They noted that to obtain meaningful results, a ground truth is required. In other words the pWCRT must be known. This is far from simple, and may not be possible for tasks in a real system. Therefore they constructed a simulation of task behaviour based on pWCET distributions, which could potentially be obtained from a real system. The approach obtains the ground truth via probabilistic worst-case response time analysis using the method given by Maxim et al. [106] (see Section 3.3), which determines precise pWCRT distributions from the input pWCETs. The ground truth is compared to a number of statistical approaches. These include fitting to Normal, reversed Weibull, and Gumbel distributions, and an EVT-based approach using the Block Maxima method. The evaluation shows that fitting to a Normal or reversed Weibull distribution is unsound with approximately half of the pWCRTs under-estimating the probability of a deadline failure. Fitting to a Gumbel distribution produced better results in this respect with about 10% unsound results. Using the EVT-based approach, none of the results were unsound; however, there was an increase in pessimism compared to directly fitting a Gumbel distribution.

7.2 Summary and Perspectives

In this section we reviewed research that takes a statistical approach to estimating response time distributions. Of particular note is the work by Lu et al. [96, 97, 94, 95] and Maxim et al. [111]. The former showing that EVT can provide meaningful predictions of the tail of response time distributions even in the case of systems with intricate dependences, and the latter showing that the results from EVT are sound compared to the ground truth, while those from directly fitting a distribution to the observations are not.

All of the work reviewed in this area has focused on single processor systems. For tasks running on COTS multi-core platforms there are significant difficulties involved in obtaining precise worst-case response times via analytical methods due to issues of cross-core contention. The application of statistical methods to directly predict the extreme values of the response time distributions for tasks in such systems could potentially provide some solutions to this problem. This is an interesting area for future research.

8 Probabilistic Analysis of Mixed Criticality Systems

The term Mixed Criticality System (MCS) is used to describe real-time systems where applications with different *criticality levels* (meaning different levels of assurance required against failure) are integrated onto the same hardware platform. This integration gives rise to research questions in terms of how to reconcile the conflicting requirements of sharing for efficient resource usage and separation for reasons of assurance [31]. In 2007, Vestal [145] described a mixed criticality task model whereby LO-criticality tasks have a single worst-case execution time estimate $C(LO)$, and HI-criticality tasks have two estimates $C(LO)$ and $C(HI)$, with the latter, larger estimate obtained via methods that give a higher level of confidence / assurance that it will not be exceeded. (For example $C(LO)$ might be an upper bound on the longest execution time observed during testing, while $C(HI)$ may be a conservative value obtained via detailed static timing analysis). The timing constraints placed on the system require that all tasks meet their deadlines provided the $C(LO)$ budgets are not exceeded; however, if a HI-criticality task exceeds its $C(LO)$ budget, then it is *only* required that the HI-criticality tasks meet their deadlines, assuming that they execute for at most their $C(HI)$ budgets. This required behaviour reflects the different failure rates that may be acceptable at different criticality levels (see the discussion in Section 1). For more information on research into scheduling mixed criticality systems, see the survey by Burns and Davis [31].

In this section we review recent research on probabilistic schedulability analysis for MCS. These methods typically use a richer representation based on execution time or pWCET distributions, rather than the discrete execution time budgets $C(LO)$ and $C(HI)$ at different criticality levels assumed by Vestal's model [145]. Here, one may consider the $C(LO)$ and $C(HI)$ budgets from Vestal's model as two points on the x-axis of the 1 - CDF of a pWCET distribution (see Figure 2 in Section 2), each with an associated probability of exceedance (i.e. the corresponding y-axis value).

8.1 Analysis for Mixed Criticality Systems

Research into probabilistic schedulability analysis for mixed criticality systems is in its infancy with a small number of papers published from 2015 onwards. The majority of these works are short papers that have appeared in workshops. A necessarily brief review of them is given below.

In 2015, Santinelli and George [132] presented preliminary work on probabilistic schedulability analysis for MCS scheduled using EDF. They investigated how schedulability varies with task execution times, referred to as the probabilistic C-space. Later the same year, Guo et al. [64] extended the mixed criticality task model with a single exceedance probability value for the low assurance budget of each HI-criticality task, and used probabilistic analysis to improve schedulability.

In 2016, Maxim et al. [107] adapted probabilistic response time analysis from [106] (see Section 3.3) to fixed priority preemptive scheduling of MCS using the Adaptive Mixed Criticality (AMC) and Static Mixed Criticality (SMC) schemes [22]. They compared this analysis to the equivalent deterministic methods, highlighting the performance gains that can be obtained by utilising more detailed information about worst-case execution time estimates described in terms of probability distributions. This work was extended by Maxim et al. [108] to provide a more precise analysis, and also to examine by how much the execution time budgets of LO-criticality tasks can be increased by employing probabilistic rather than deterministic schedulability analysis methods.

In 2016, Alahmad and Gopalakrishnan [9, 8] studied the problem of scheduling mixed criticality job sets with execution times described by random variables. The aim of this work is to compute implementable scheduling policies that meet the probabilistic timing constraints. The problem is modelled as a Constrained Markov Decision Process (CMDP), with feasible policies obtained using a linear program.

In 2016, Draskovic et al. [57] examined fixed priority preemptive scheduling of MCS of periodic tasks with execution times described by random variables. They employed the method of Diaz et al. [54] (see Section 3.2) to compute the probability of a deadline miss for every job in the hyperperiod, and from that the overall deadline failure rate. They also computed the expected time before a change to HI-criticality mode, and showed that this expected time depends on the LO-criticality execution time budget allocated to HI-criticality tasks. A smaller budget results in a lower probability of deadline failure, but a shorter expected time before a transition to HI-criticality mode.

In 2017, Abdeddaim and Maxim [1] derived probabilistic response time analysis for mixed criticality tasks under fixed priority preemptive scheduling, adapting the techniques of Maxim and

Cucu-Grosjean [106] (see Section 3.3) to the MCS model. The analysis computes the probability of deadline misses for each task in each criticality mode. This work does not assume any monitoring, hence lower criticality tasks are assumed to continue executing in higher criticality modes.

In 2017, Kuttler et al. [80] introduced an algorithmic approach to probabilistic schedulability analysis called *symbolic scheduling*. They considered an extension of AMC [22] where the priorities of LO-criticality tasks are reduced (to below that of any HI-criticality task) when the system switches to HI-criticality mode. Assuming this behaviour, they calculate the probability of each job of a LO-criticality task meeting its deadline. The method applies to periodic tasks. Conceptually, it considers every possible combination of execution times, forming a tree where each path from root to leaf represents a possible behaviour of the system. The disadvantage of this naive approach is that the tree quickly becomes very large. Symbolic scheduling is therefore used, whereby paths that may have different execution times but agree on the order of jobs and their success or otherwise in meeting deadlines are combined. The evaluation shows that considering only the probabilistic worst-case response time (i.e. the behaviour at the critical instant) can be pessimistic in its estimate of the probability of LO-criticality jobs missing their deadlines.

8.2 Summary and Perspectives

In this section we reviewed research on probabilistic schedulability analysis for Mixed Criticality Systems (MCS). These methods consider MCS described using execution time or pWCET distributions rather than the conventional $C(LO)$ and $C(HI)$ WCET estimates / execution time budgets of Vestal's model [145]. This additional information provides the potential for improvements in schedulability and in the size of the budgets that can be afforded to different tasks, see for example the work of Maxim et al. [108]. MCS are a hot topic of real-time systems research. A probabilistic view of MCS would appear to provide an excellent match to requirements that are specified in terms of levels of assurance and failure rates. We note, however, that research in this area is currently in its infancy with a small number of works starting in 2015, the majority of which are workshop papers or other short publications. (For a comprehensive review of other research into MCS see the survey by Burns and Davis [31]).

9 Miscellaneous

In this section, we review research that explores miscellaneous aspects of scheduling and schedulability analysis for probabilistic real-time systems, including task graphs and precedence constraints, analysis for multiprocessor systems, miscellaneous models and techniques, and position papers.

9.1 Task Graphs and Precedence Constraints

The majority of the research in this area was published by Manolache et al. in a series of papers [100, 101, 102] from 2001 to 2008.

In 2001, Manolache et al. [100] presented a method of analysing systems with precedence relations between tasks described by task graphs, and task execution times described by probability distributions. They assume that the tasks are periodic with a reasonably small hyperperiod. The method is applicable only to non-preemptive scheduling algorithms such as fixed priority and EDF that do not alter job priorities between scheduling points (i.e. task release times and deadlines). It is assumed that if a job misses its deadline, then it is aborted. The first step in the analysis is to divide the hyperperiod into so called *Priority Monotonic Intervals* (PMI) de-marked by job release times and deadlines. The stochastic process is then constructed and analysed at the same time, thus reducing memory requirements. A stochastic process state consists of the index of the currently running job, the start time of the job, and the indices of the ready jobs. The number of next states depends on the number of possible execution times of the job. As this number can be very large, states are grouped together, while still preserving the Markovian property. States are processed in order, by PMI first, and then within a PMI by highest priority ready job. The method thus determines the expected deadline miss probability for each task. Evaluation shows that the method is effective for tasks sets of cardinality up to 20 and hyperperiods from 360 to 5040. Subsequently, in 2004, Manolache et al. [101] extended their earlier work [100] to the case where tasks may continue to execute beyond their deadlines. Such overruns are restricted by limiting the maximum number of jobs of the same task that can exist in the system at any given time. On the release of a task, if this limit would be exceeded, then there are two options: discard the oldest job of the task, or reject the new job. Evaluation shows that rejecting the new job leads to much greater complexity, since a bound is removed on the number of successor states. The authors also discuss possible extensions to preemptive scheduling, but note that the complexity of the method would be greatly increased. Later, in 2008, Manolache et al. [102] proposed a solution to the problem of task priority assignment and mapping in a multiprocessor system. The task model is the same as in their previous works [100, 101]. The method is based on a Tabu search, with various approximations used to reduce the complexity of computing estimates of the deadline miss probability and thus the fitness function used in the search.

In 2003, Hua et al. [69] proposed a method of using probabilistic descriptions of task execution times to optimise other parameters of interest in multimedia systems, such as energy consumption. The model considered is a task graph, where the tasks in the graph are executed in a fixed order, and must be completed by a given deadline. The application i.e. the task graph is executed periodically. The system must meet a *completion ratio* condition, effectively a threshold on the expected proportion of a large number of instances of the task graph that will meet their deadlines. A simple formula is given for computing the completion ratio based on the execution time distributions. This formula has exponential complexity, since it effectively considers all combinations of possible task execution times from the distributions. An approximation is therefore used that starts with each task assigned its WCET, and then while the deadline is not met, it removes the largest value from one of the task execution time distributions. This lowers the overall execution time, but decreases the completion ratio. Eventually, either the task graph is deemed schedulable with an acceptable completion ratio, or the completion ratio becomes too small. The value to remove is chosen in a greedy way, by selecting the one that gives the largest reduction in overall execution time, weighted by how much the completion ratio is reduced. The authors also describe an offline/online algorithm for minimising energy consumption via dynamic voltage and frequency scaling (DVFS). The aim here is to either drop jobs or to extend their execution to reduce energy consumption, while meeting the specified threshold on the completion ratio. We note that it is implicitly assumed that the execution time of a job becomes known at the point when it is released, which may not be possible to achieve in practice.

9.2 Multiprocessor Analysis

Below, we cover the few works on probabilistic schedulability analysis for multiprocessor systems. The relative absence of work in this area contrast strongly with the wealth of research into conventional schedulability analysis techniques for multiprocessor systems, (see Davis and Burns [49] for a survey).

In 2002, Nissanke et al. [118] and Leulseged and Nissanke [86] described a probabilistic framework for investigating the schedulability of tasks on a multiprocessor, with execution times and deadlines modelled via probability distributions. Each task has a fixed period, and its behaviour is represented by points on a cartesian graph of remaining execution time versus laxity.

A set of non-zero probabilities characterise the task as arriving with a certain execution time and laxity. Between arrival times, remaining computation times and laxities are also described by probabilities, but evolve according to the scheduling algorithm, reaching either negative laxity indicating a missed deadline or zero remaining computation time indicating completion. By knowing the probability of realising each scenario, and the competition between tasks at the same priority, the probability of a task being executed is computed. This enables calculation of the execution patterns of all tasks over the hyperperiod, enabling the various properties of interest to be derived. The movement of tasks through this scheduling domain depends on the probability of m processors being assigned tasks to execute that are in a particular state (specified by laxity and execution time), and also on the arrival rate. Overall, the framework can be used to determine performance indicators such as expected deadline failure rate, success rate, number of tasks executing, number of tasks at a particular point etc. The authors propose that a probabilistic scheduling policy could be determined by prescribing a probability to executing tasks based on their location in the scheduling domain. These probabilities could be obtained by solving an optimisation problem with the aim of maximising some performance indicator of interest, such as the expected success rate.

In 2010, Mills and Anderson [112] extended prior work on tardiness bounds for global EDF (GEDF) scheduling to tasks with execution times described by i.i.d. random variables. For such systems, they derived a bound on expected mean tardiness for all tasks. Subsequently in 2011, Mills and Anderson [113] generalised their previous work to address tasks with stochastic execution times (specified via mean and variance) scheduled via sporadic servers under GEDF or any other global scheduling algorithm with bounded tardiness. They proved a worst-case tardiness bound when the system has a worst-case utilisation that is bounded by the number of processors, and an expected or average-case tardiness bound when the average-case utilisation is bounded by the number of processors. This latter bound does not require knowledge of the task's WCET, or even for the WCET to be bounded. Hence the average-case tardiness bound can be computed on the basis of the mean and variance obtained from representative execution time observations. An example shows that the computed tardiness bounds are much tighter that those derived previously [112].

In 2014, Liu et al. [92] considered how to deal with dependencies between the execution times of jobs of a task. They build upon the work of Mills and Anderson [113], thus assuming that tasks are scheduled via sporadic servers under GEDF. The key idea is to represent the stochastic execution times of the task via two components: (i) a fixed threshold, and (ii) an excess over that threshold. The idea is that by tuning the threshold to an appropriate level, the non-zero excesses over the threshold become independent. (This notion is similar to the one of extremal independence, where extreme execution time values are sufficiently rare and far apart as to be independent). Using an *independence threshold* for each task enables the system designer to balance the need for a tractable probabilistic analysis based on modeling execution times as independent random variables, and the need to avoid a pessimistic provision based on deterministic worst-case reasoning. The authors integrate the concept of independence thresholds into the prior approach of Mills and Anderson [113]. They present a measurement process based on statistical tests of independence that is able to find the smallest threshold such that dependences are effectively eliminated. Finally, they show via an MPEG video decoding case study that the overall approach is highly effective, on average achieving a two-fold reduction in the required server execution time budgets compared to deterministic worst-case provisioning.

In 2015 and a later journal extension in 2017, Wang et al. [149, 148] proposed a task partitioning algorithm for fixed priority preemptive scheduling of tasks with execution times described by independent random variables on a homogeneous multi-core platform. They explored four different heuristics that quantify the degree of harmonicity among the tasks assigned to a processor. These are mean-based, variance-based, cumulative distribution-based, and distribution sum-based. The evaluation shows that these heuristics significantly outperform existing (deterministic) methods in terms of the number of cores required to ensure that probabilistic timing guarantees are met (i.e. that all tasks meet their deadlines with an acceptable probability). Later work by Ren et al. [128] built upon the above work addressing some of its drawbacks. In particular, the partitioning approach employed by Wang et al. [149, 148] can suffer from fragmentation. Ren et al. address this issue by combining a consideration of both harmonicity and probabilistic workload. Their approach first orders the tasks by decreasing expected utilisation and selects the highest expected utilisation task as a reference task. It then selects tasks to add to the subset containing the reference task based on harmonicity. Tasks are added until no further task can be added without the subset failing the probabilistic schedulability test for a single processor. The selected subset of tasks is then assigned to one processor and the method repeats for the remaining unassigned tasks. Evaluation using synthetic task sets shows that this approach is both more effective and has a shorter average runtime than the previous partitioning approach of Wang et al. [149, 148].

9.3 Miscellaneous Models and Techniques

In this subsection we review work relating to miscellaneous models and techniques such as randomised job dropping and imprecise computation.

In 2001, Hu et al. [68] studied fixed priority preemptive systems with task execution times described by random variables. They argue that finding the probability of each task missing its deadline does not give the full picture for a system and can be misleading. Instead, they propose that the probability of failures is assessed over *state cycles* corresponding to the intervals between job releases for periodic tasks. They reason that a *state* is only feasible if all of the jobs that are ready in that state meet their deadlines. (A job with a long deadline may thus affect the feasibility of multiple states). The overall probability of system feasibility is assessed by determining the expected number of feasible states over the total number of states in the hyperperiod, or as an approximation a shorter interval such as the task period. The focus on states ensures that correlations between different jobs missing their deadlines are captured. We note that this method does not consider the backlog at the end of the hyperperiod and thus is only applicable if the worst-case processor utilisation does not exceed 1. Constrained deadline periodic tasks are assumed. The method is also extended to EDF. The complexity of the method is exponential in the number of values in the execution time distributions, with the exponent being the number of releases of the task within the hyperperiod. It therefore seems unlikely that the method is viable for systems that do not have both a small number of tasks and a short hyperperiod.

Also in 2001, Hamann et al. [65] integrated a probabilistic description of execution times with the imprecise computation model based on *mandatory* and *optional* components [90]. They assume that each task is composed of a single mandatory part that must be guaranteed to complete by its deadline and multiple optional parts for which only soft (probabilistic) guarantees are required. It is assumed that the execution time distribution is provided for each part of a task, and that the WCET is known for the mandatory part. A simple analysis is given that determines the size of the reservation required to guarantee the mandatory part and to provide the desired probabilistic guarantee that a required percentage of the optional parts complete. This is achieved by convolving the execution time distributions, hence their independence is assumed. The approach is motivated by multimedia examples involving the decoding of MPEG frames, with I and P frames mandatory and B frames optional.

In 2002, Kim et al. [77] proposed an alternative to task-level or job-level isolation based on *randomised dropping*. The motivation for this approach is that isolation does not allow sharing of processing resources when a job executes in less time than reserved for it. In the model addressed in this work, periodic tasks are assumed to have independent execution times with known probability distributions, and are assigned an execution time budget that they can use without triggering the dropping mechanism. Typically this corresponds to the expected or average execution time. Job scheduling is via EDF; however, each job also has one or more trigger values on the execution time that it uses. When one of these trigger values is reached, there is an associated probability that the job will be dropped. By tuning these values and the probabilities of dropping, the interference on other jobs can be limited in a probabilistic way. The authors propose a stochastic analysis for their model, based on a Markov process. They derive the Markov chain over multiple hyperperiods, and compute the stationary backlog distribution. This is then used to determine the response time distribution and deadline miss probability for each job, and hence also for each task. The randomised dropping is modelled as an adjustment to the execution time distribution for each task. Evaluation shows that the method is successful in achieving a high probability of deadlines being met in an otherwise overloaded system.

In 2009, Gopalakrishnan [61] explored the idea of sharp utilisation thresholds in fixed priority preemptive scheduling of periodic tasks. They show that for task sets chosen uniformly at random, there is a transition around some utilisation U where the probability of an implicit deadline task set being schedulable under rate-monotonic scheduling changes from 1 to 0. The width of this transition depends on the cardinality of the task set and tends to a sharp threshold (i.e. an interval of zero width) as the number of tasks tends to infinity. A similar result giving a sharp synthetic utilisation (or density) threshold was obtained for aperiodic tasks, where a task's density is given by its execution time divided by its relative deadline. This work provides a highly efficient means of admitting task sets at runtime based on a simple utilisation-based test, while ensuring that there is a high probability that the task sets will be schedulable. For soft real-time systems this approach could be much more effective than using hard utilisation bounds [89], below which there are no task sets that are unschedulable.

9.4 Position Papers

The following works discuss some requirements for probabilistic schedulability analysis to be useful in practice, as well as issues relating to independence.

In 2012, Quinton et al. [125] set out four requirements (or conditions) that must be satisfied by probabilistic analysis for it to be useful: (i) it must be efficient enough to scale to real systems, (ii) it must provide meaningful results for system designers, (iii) the model used must be practical (i.e. simple enough to be provided by the designer) or automatically generated, (iv) any assumptions made by the analysis must be formally described so they can be validated. They note that most existing probabilistic approaches determine the distribution of response times, but can say nothing about the behaviour of the system in a short and bounded time window. In other words, they cannot answer the question, "can deadline misses occur in a burst?" (See the discussion in Section 2.5).

In 2013, Cucu-Grosjean [42] considered different types of *independence* in the context of probabilistic real-time systems. A key aspect of this work is the discussion covering the definition of, and the differences between, probabilistic execution time distributions (pET) of jobs and probabilistic worst-case execution time distributions (pWCET) of tasks. The author notes that since the pWCET distribution upper bounds all the pET distributions for the jobs of a task (in the sense of the greater than or equal to operator $\succeq$ on random variables defined in [55]), then the pWCET distribution of a task is *by definition* probabilistically independent with respect to jobs of the same or different tasks. They also highlight the differences between independence between tasks (as required by the Liu and Layland model [89]), probabilistic independence between random variables (needed so that basic convolution may be used to determine probabilistic response times), and statistical independence between observations of execution times. In a paper accompanying an invited talk at the ETR summer school in 2013, Cucu-Grosjean [43] discusses different types of independence (probabilistic, statistical) [42] and also recaps the work on probabilistic schedulability analysis [54] and probabilistic WCET analysis [44] for real-time systems, re-sampling techniques [109], and priority assignment policies [105].

9.5 Summary and Perspectives

In this section, we reviewed works that explore different aspects of scheduling and schedulability analysis for probabilistic real-time systems. Here, we highlight the work of Liu et al. [92] that provides a means of dealing with dependencies between the execution times of jobs of a task via an *independence threshold*, and the work of Gopalakrishnan [61] that provides a simple but highly effective probabilistic admission test for soft real-time task sets. We note that while there is substantial literature on conventional schedulability analysis for multiprocessor systems (see Davis and Burns [49] for a survey), with a handful of exceptions, research into probabilistic schedulability analysis has focused on uniprocessor systems. Further, there are now a large number of papers focused on providing conventional schedulability analysis for tasks running on multi-core platforms, taking into account the effects of contention for shared hardware resources (interconnect, memory hierarchy, I/O system etc.) between tasks running in parallel on different cores. There appears to be little if any published work on probabilistic schedulability analysis for such systems. This is an important area that could benefit from future research.

10 Addressing Issues of Intractability

Much of the research into probabilistic schedulability analysis relies on combining execution time distributions via basic convolution. A naive assessment of basic convolution would assume that it has exponential complexity $O(m^n)$ where m is the number of points in each distribution and n is the number of distributions convolved. While this is correct in terms of the theoretical worst-case, in practice the range of values in each distribution is such that the overall complexity is far lower. For example, assuming that the largest (integer) value in an execution time distribution is N, then the number of points in the intermediate distribution after m convolutions is at most mN, and hence the overall complexity of m convolutions is $O(Nm^2)$. Nevertheless, probabilistic response time analysis involving realistic numbers of tasks with a spread of periods of a few orders of magnitude can involve significant computation. In this section, we review works that seek to reduce the amount of computation required, while in some cases trading faster calculation for pessimism in the results.

10.1 Re-sampling

Re-sampling reduces the number of values present in a discrete probability distribution, and therefore reduces the amount of computation required in convolution operations involving that distribution. This improvement in efficiency comes at a cost of reduced accuracy or pessimism in the results.

In 2010, Refaat et al. [127] presented a method of reducing the complexity of the analysis of Diaz et al. [54, 55] and Kim et al. [76] by re-sampling the execution time distributions used. Their method involves taking k random samples from the probability distribution and assigning the probability for all excluded points to the worst-case value from the distribution, which is always kept. This method provides a sound but pessimistic approximation [55]. Subsequently, later in 2010, Maxim et al. [110] presented an alternative approach to re-sampling execution time distributions that improves on the method introduced by Refaat et al. [127]. This approach keeps the $k-1$ values with the largest probability as well as the largest value. It then reassigns the probability mass for each removed point to the retained point with the next larger value. This is a sound approximation, with less pessimism than assigning the probability mass of excluded points to the worst-case value.

Building on the earlier work in this area, in 2012, Maxim et al. [109] considered the need for re-sampling probabilistic worst-case execution time (pWCET) distributions when computing probabilistic worst-case response time (pWCRT) distributions. This computation involves repeated use of the convolution operator. The runtime of convolution of *arbitrary* distributions can grow rapidly, as the number of points (distinct execution times) in the existing distribution can in the worst case be multiplied by the number of points in each pWCET distribution that is convolved onto it. In theory, this leads to an exponential growth in the runtime. In practice, the number of points in a discrete distribution cannot exceed max-et – min-et, where max-et and min-et are the maximum and minimum values in the distribution. The authors address the complexity issues with convolution by introducing sound ways of re-sampling the distributions created. A sound re-sampling is one that does not move probability values right-to-left, thus never allocating them to a smaller execution time. Although moving some values left-to-right introduces pessimism, it ensures that the pWCRT produced over-approximates that which would be obtained without re-sampling. The work considers *uniform spacing*, a re-sampling technique that is widely used in other contexts. This method selects sample points that are uniformly spaced in terms of probabilities, i.e. at equally spaced percentiles throughout the distribution. It provides a good fit in terms of overall pessimism; however, the shape of the tail, which is important in pWCRT calculation, can be heavily compromised. Two new re-sampling techniques are introduced: *reduced pessimism* re-sampling, which seeks to minimise the probability mass moved to larger execution time values, and *domain quantisation*, which re-samples at equally spaced points in the execution time domain. Domain quantisation has the desirable property that it greatly reduces the number of points in the distributions produced *after* convolution, and also provides a good fit to the tail of the pWCET distributions. Evaluation shows that it gives the best compromise between runtime and accuracy.

In 2015, Milutinovic et al. [114] examined methods of speeding up the discrete convolution operations that are used a large number of times in SPTA. They consider methods that are precision preserving, such as power operations used when the same distribution needs to be convolved multiple times, and precision non-preserving methods such as re-sampling as proposed by Maxim et al. [109]. They found that the precision preserving techniques speeded up convolution by approximately a factor of two, while the precision non-preserving techniques traded off a minimal amount of over-approximation ($< 3\%$) for an order of magnitude increase in speed.

10.2 Analytical Methods and Other Techniques

While re-sampling, which reduces the number of values present in a discrete probability distribution, makes a trade-off between efficiency and precision in probabilistic schedulability analysis calculations, analytic methods can result in greater improvements in runtime while retaining better precision. In some cases, full precision can be retained while still making substantial improvements in runtime efficiency.

In 2017, Chen and Chen [36] considered the problem of probabilistic response time analysis and the computational complexity involved in repeated use of the convolution operator. They proposed a more efficient approach to computing the probability of deadline misses, based on using the *moment generating function* (mgf) of random variables, and Chernoff bounds for the probability that the sum of a number of random variables (e.g. the execution times of multiple jobs) exceeds some bound (e.g. the deadline). The authors demonstrate how this approach can be applied to probability distributions consisting of two values representing normal operation, and abnormal operation in the event of a soft error. They also show how the approach can be extended to distributions with more values, and to give the probability of l consecutive deadline misses. The evaluation compares the proposed method to exact analysis [106] and to exact analysis with re-sampling [109] applied. The results show that the proposed method is able to efficiently determine slightly pessimistic bounds on the probability of deadline misses without the need to derive the whole response time distribution, which can be inefficient. Exact analysis was not suitable for more than 10 tasks in the experiments considered, while re-sampling, limiting the distributions to a maximum of 100 values, resulted in highly pessimistic deadline miss probabilities (e.g 1) for task set utilisation values $> 70\%$.

In 2018, von der Bruggen et al. [147] considered the problem of determining the worst-case probability of deadline misses for tasks under fixed priority preemptive scheduling. They note that traditional convolution-based approaches to computing pWCRT distributions quickly become intractable as the number of jobs within the deadline of a lower priority task that is being analysed increases. To address this problem, they present a novel approach based on using multinomial distributions, which they further improve via the use of a pruning technique. This method retains full precision and is viable for much larger task sets than previous approaches. In the evaluation, the technique is used for systems of up to 35 tasks and is shown to be viable for up to 100 tasks. These task sets have a range of periods spanning two orders of magnitude. Hence there can be up to 100 jobs of each higher priority task within the response time of the lower priority task under analysis. The authors also present two methods based on analytical upper bounds based on Hoeffding's inequality and Bernstein's inequality. These methods offer further substantial improvements in runtime (two orders of magnitude faster than the precise multinomial-based approach), but trade off some precision in the results.

Later in 2018, Chen et al. [37] studied the problem of determining the *expected* deadline miss rate for tasks under fixed priority preemptive scheduling. The task model used assumes that each task has a normal execution time and a longer abnormal execution time that occurs when dealing with fault conditions. As the faults are assumed to be independent, the execution time distribution for each task is i.i.d. Further, the probabilities for the different execution times reflect the probability of fault occurrence. The authors make the realistic assumption that job execution continues even if a deadline is missed. (By contrast, some other works assume that jobs are aborted on reaching their deadline). The authors show that this difference in behaviour can have a substantial effect in increasing the expected deadline miss rate, since the overrun of a job affects the probability of the next job meeting its deadline. They derive an upper bound on the expected deadline miss rate. This is done by considering the probability that a task has j consecutive deadline misses within the same busy period, for all values of j up to some limit. The method leverages the authors' prior work [36, 147] to determine the probability of j consecutive deadline misses. Using the convolution-based approach [147] results in bounds that are tighter with respect to the simulation results, compared to using the analytical bound [36]. The trade-off is however a significantly greater runtime.

10.3 Summary and Perspectives

Issues of tractability were once considered a substantial roadblock to the use of probabilistic schedulability analysis on practical systems. This issue has been addressed, first by methods based on re-sampling [109] that can reduce the amount of computation required to perform convolution (the basic operation used repeatedly in many probabilistic schedulability analyses) while trading off additional pessimism in the results. More recently, analytical approaches have been developed that have a much shorter runtime, but still trade off pessimism in the results [36]. Finally, the work of von der Bruggen et al. [147] in 2018, provides an approach which promises an efficient method of computing deadline miss probabilities for large task sets without a significant trade-off in precision.

11 Conclusions

In this survey, we reviewed research into schedulability analysis for probabilistic real-time systems. We covered the main subject areas including probabilistic response time analysis, probabilistic analysis assuming execution time servers, real-time queuing theory, probabilities emanating from fault models, statistical analysis of the response times, and probabilistic analysis of mixed criticality systems; as well as reviewing supporting mechanisms and analyses that address issues of intractability.

We now conclude by identifying open issues, key challenges and possible directions for future research. We present these as a series of questions and statements.

- How to determine the (worst-case) execution time distribution for a task? This is the subject of probabilistic timing analysis, see the companion survey [52] for a detailed discussion. We note that in some cases the variation of the execution times over time may be such that using a single valid distribution may be too pessimistic (e.g. when the system exhibits different modes of behaviour).
- How to handle issues relating to dependences between the execution times of jobs of (i) the same task, and (ii) jobs of different tasks? The impact of these dependences may vary based on how strong they are. Appropriate statistical studies are needed to investigate the types of dependences and their impact on probabilistic schedulability analysis. Analyses are needed that can address dependencies.
- How to reconcile requirements on the maximum length of black-out periods (number of consecutive missed deadlines) with a probabilistic treatment of deadline failures? This problem relates to dependences between response times that may occur due to dependences between the execution times of jobs of the task considered, and due to dependences in the amount of interference from other tasks.
- How to provide probabilistic schedulability analysis based on probabilistic Worst-Case Execution Time (pWCET) distributions when there are dependences between execution times of consecutive jobs? This is particularly problematic in the case of pWCET distributions derived via MBPTA techniques (see the discussion in Section 2.3).
- How to provide appropriate solutions for multiprocessor schedulability analysis? While there is a wealth of research into conventional schedulability analysis for multiprocessor systems, research into probabilistic schedulability analysis has, with only a few exceptions, focussed on uniprocessor systems.
- How to adapt probabilistic models, using the richer description given by pWCET distributions, in the context of Mixed Criticality Systems. Although expanding rapidly (see Figure 1), work in this area is still in its infancy.
- How to adapt the current statistical approaches such as Extreme Value Theory in the context of response time analysis? The use of EVT has shown some promise in this area, but has not been explored in detail.

- How to ensure that probabilistic schedulability analysis methods are viable for use in practical systems? Issues here include validation of the methods, and ensuring that they can be applied to problems of a practical size.

Since the initial work in the late 1980s and 1990s, significant progress has been made in the development of probabilistic schedulability analysis techniques. However, there are still important unanswered questions and open issues to be resolved, as well as a number of interesting areas for future research that are only beginning to be explored. We end this survey with a brief discussion of an important direction for future real-time systems research which probabilistic analysis techniques may be able contribute to.

There is a continuing trend in industry sectors including avionics, automotive electronics, consumer electronics, and robotics away from development and deployment on single-core processors towards using significantly more powerful and complex Common-Off-The-Shelf (COTS) multi-core and many-core hardware platforms. This trend is driven by requirements on size, weight and power consumption, increasing cost pressures and the demand for more complex and capable functionality delivered through software. The use of COTS multi-core hardware poses significant challenges in terms of verifying timing behaviour and ensuring that real-time constraints are met. These challenges stem from the complexity of the architecture and the way in which hardware resources such as the interconnect and the memory hierarchy are shared between different processing cores. Some researchers are seeking to address these problems through approaches based on partitioning and separation (e.g. single-core equivalence [98]), while others aim for solutions based on considering the explicit interference on each hardware resource from co-running programs and how this demand can be served by the available resource supply [12, 46]. There is the potential for probabilistic schedulability analysis and probabilistic timing analysis techniques (reviewed in a companion survey [52]) to play a role in the timing verification of such complex real-time systems.

Work on probabilistic timing analysis and probabilistic schedulability analysis for multi-core and many-core systems is in its infancy with opportunities for significant advances addressing this important research challenge.

Acknowledgements. The research that went into writing this survey was funded, in part, by the Inria International Chair program and the ESPRC grant MCCps (EP/P003664/1). EPSRC Research Data Management: No new primary data was created during this study.

References

1 Y. Abdeddaim and D. Maxim. Probabilistic Schedulability Analysis for Fixed Priority Mixed Criticality Real-Time Systems. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, 2017.

2 L. Abeni and G. Buttazzo. Integrating multimedia applications in hard real-time systems. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 4–13, December 1998. doi:10.1109/REAL.1998.739726.

3 L. Abeni and G. Buttazzo. QoS guarantee using probabilistic deadlines. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 242–249, 1999. doi:10.1109/EMRTS.1999.777471.

4 L. Abeni and G. Buttazzo. Stochastic analysis of a reservation based system. In *Proceedings 15th International Parallel and Distributed Processing Symposium. IPDPS 2001*, pages 946–952, April 2001. doi:10.1109/IPDPS.2001.925049.

5 L. Abeni, D. Fontanelli, L. Palopoli, and B. Villalba Frías. A Markovian model for the computation time of real-time applications. In *Proceedings of IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, May 2017. doi:10.1109/I2MTC.2017.7969878.

6 L. Abeni, N. Manica, and L. Palopoli. Efficient and Robust Probabilistic Guarantees for Real-time Tasks. *J. Syst. Softw.*, 85(5):1147–1156, May 2012. doi:10.1016/j.jss.2011.12.042.

7 Z. Alabedin, H. Hammadeh, S. Quinton, and R. Ernst. Extending typical worst-case analysis using response-time dependencies to bound deadline misses. In *Proceedings of the IEEE & ACM International Conference on Embedded Software*

(EMSOFT), pages 10:1–10:10, 2014. doi:10.1145/2656045.2656059.

8 B. Alahmad and S. Gopalakrishnan. A Risk-Constrained Markov Decision Process Approach to Scheduling Mixed-Criticality Job Sets. In *Proceedings of Workshop on Mixed Criticality (WMC)*, 2016.

9 B. Alahmad and S. Gopalakrishnan. Risk-aware scheduling of dual criticality job systems using demand distributions. Leibniz Transactions on Embedded Systems. In *Leibniz Transactions on Embedded Systems (LITES)*, 2016.

10 S. Altmeyer, L. Cucu-Grosjean, and R. I. Davis. Static probabilistic timing analysis for real-time systems using random replacement caches. *Springer Real-Time Systems*, 51(1):77–123, 2015. doi:10.1007/s11241-014-9218-4.

11 S. Altmeyer and R. I. Davis. On the Correctness, Optimality and Precision of Static Probabilistic Timing Analysis. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 26:1–26:6, 2014. URL: http://dl.acm.org/citation.cfm?id=2616606.2616638.

12 S. Altmeyer, R. I. Davis, L. Indrusiak, C. Maiza, V. Nelis, and J. Reineke. A Generic and Compositional Framework for Multicore Response Time Analysis. In *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, pages 129–138, 2015. doi:10.1145/2834848.2834862.

13 A. Atlas and A. Bestavros. Statistical rate monotonic scheduling. In *Real-Time Systems Symposium, 1998. Proceedings., The 19th IEEE*, pages 123–132, December 1998. doi:10.1109/REAL.1998.739737.

14 N. Audsley, A. Burns, M. Richardson, K. Tindell, and A. J. Wellings. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal*, 8(5):284–292, 1993.

15 N.C. Audsley. On priority assignment in fixed priority scheduling. *Information Processing Letters*, 79(1):39–44, 2001. doi:10.1016/S0020-0190(00)00165-4.

16 P. Axer and R. Ernst. Stochastic response-time guarantee for non-preemptive fixed-priority scheduling under errors. In *Proceedings of the Design Automation Conference (DAC)*, pages 1–7, May 2013. doi:10.1145/2463209.2488946.

17 P. Axer, M. Sebastian, and R. Ernst. Probabilistic response time bound for CAN messages with arbitrary deadlines. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 1114–1117, March 2012. doi:10.1109/DATE.2012.6176662.

18 H. Aysan, R. Dobrin, S. Punnekkat, and R. Johansson. Probabilistic Schedulability Guarantees for Dependable Real-Time Systems under Error Bursts. In *Proceedings of IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pages 1154–1163, November 2011. doi:10.1109/TrustCom.2011.157.

19 H. Aysan, R. Dobrin, S. Punnekkat, and J. Proenza. Probabilistic scheduling guarantees in distributed real-time systems under error bursts. In *Proceedings of the IEEE Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–9, September 2012. doi:10.1109/ETFA.2012.6489644.

20 D. Y. Barrer. Queuing with Impatient Customers and Ordered Service. *Operations Research*, 5(5):650–656, 1957. doi:10.1287/opre.5.5.650.

21 S. Baruah and A. Burns. Sustainable Scheduling Analysis. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 159–168, 2006. doi:10.1109/RTSS.2006.47.

22 S. K. Baruah, A. Burns, and R. I. Davis. Response-time analysis for mixed criticality systems. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 34–43. IEEE, 2011.

23 S. Ben-Amor, D. Maxim, and L. Cucu-Grosjean. Schedulability Analysis of Dependent Probabilistic Real-time Tasks. In *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, pages 99–107. ACM, 2016. doi:10.1145/2997465.2997499.

24 G. Bernat, A Burns, and A Liamosi. Weakly hard real-time systems. *IEEE Transactions on Computers*, 50(4):308–321, April 2001. doi:10.1109/12.919277.

25 G. Bernat and R. Cayssials. Guaranteed on-line weakly-hard real-time systems. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 25–35, December 2001. doi:10.1109/REAL.2001.990593.

26 I. Broster and A. Burns. 1st International Workshop on Probabilistic Analysis Techniques for Real-Time and Embedded Systems (PARTES). In *Random Arrivals in Fixed Priority Analysis*, 2004.

27 I. Broster and A. Burns. Work-in-Progress of the IEEE Real-Time Systems Symposium. In *Applying Random Arrival Models to Fixed Priority Analysis*, December 2004.

28 I. Broster, A. Burns, and G. Rodriguez-Navas. Probabilistic analysis of CAN with faults. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 269–278, 2002. doi:10.1109/REAL.2002.1181581.

29 I. Broster, A. Burns, and G. RodrÍguez-Navas. Timing Analysis of Real-Time Communication Under Electromagnetic Interference. *Springer Real-Time Systems*, 30(1):55–81, 2005. doi:10.1007/s11241-005-0504-z.

30 A. Burns, G. Bernat, and I. Broster. *A Probabilistic Framework for Schedulability Analysis*, pages 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. doi:10.1007/978-3-540-45212-6_1.

31 A. Burns and R. I. Davis. A Survey of Research into Mixed Criticality Systems. *ACM Comput. Surv.*, 50(6):82:1–82:37, November 2017. doi:10.1145/3131347.

32 A. Burns and S. Edgar. Predicting computation time for advanced processor architectures. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 89–96, 2000. doi:10.1109/EMRTS.2000.853996.

33 A. Burns, S. Punnekkat, L. Strigini, and D. R. Wright. Probabilistic scheduling guarantees for

fault-tolerant real-time systems. In *Dependable Computing for Critical Applications 7, 1999*, pages 361–378, November 1999. doi:10.1109/DCFTS.1999.814306.

34 L. Carnevali, A. Melani, L. Santinelli, and G. Lipari. Probabilistic Deadline Miss Analysis of Real-Time Systems Using Regenerative Transient Analysis. In *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, pages 299:299–299:308, 2014. doi:10.1145/2659787.2659823.

35 F. J. Cazorla, E. Quiñones, T. Vardanega, L. Cucu, B. Triquet, G. Bernat, E. Berger, J. Abella, F. Wartel, M. Houston, L. Santinelli, L. Kosmidis, C. Lo, and D. Maxim. PROARTIS: Probabilistically Analyzable Real-Time Systems. *ACM Transactions on Embedded Computing Systems*, 12(2s):94:1–94:26, May 2013. doi:10.1145/2465787.2465796.

36 K. H. Chen and J. J. Chen. Probabilistic schedulability tests for uniprocessor fixed-priority scheduling under soft errors. In *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 1–8, June 2017. doi:10.1109/SIES.2017.7993392.

37 Kuan-Hsun Chen, Georg von der Bruggen, and Jian-Jia Chen. Analysis of Deadline Miss Rates for Uniprocessor Fixed-Priority Scheduling. In *Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, August 2018.

38 H. Chetto, M. Silly, and T. Bouchentouf. Dynamic Scheduling of Real-time Tasks Under Precedence Constraints. *Springer Real-Time Systems*, 2(3):181–194, September 1990. doi:10.1007/BF00365326.

39 S. Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer, 2001. doi:10.1007/978-1-4471-3675-0.

40 L. Cucu. Preliminary results for introducing dependent random variables in stochastic feasibility analysis on CAN. In *Proceedings of IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 271–274, Dresden, Germany, May 2008. IEEE. doi:10.1109/WFCS.2008.4638759.

41 L. Cucu and E. Tovar. A Framework for the Response Time Analysis of Fixed-priority Tasks with Stochastic Inter-arrival Times. *SIGBED Rev.*, 3(1):7–12, January 2006. doi:10.1145/1279711.1279714.

42 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*, pages 29–37, 2013.

43 L. Cucu-Grosjean. Probabilistic real-time scheduling. In *ETR 2013-Ecole d'été temps réel*, 2013.

44 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 *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 91–101, July 2012. doi:10.1109/ECRTS.2012.31.

45 R. I. Davis. A review of fixed priority and EDF scheduling for hard real-time uniprocessor systems. *ACM SIGBED Review*, 11(1):8–19, 2014.

46 R. I. Davis, S. Altmeyer, L. S. Indrusiak, C. Maiza, V. Nelis, and J. Reineke. An extensible framework for multicore response time analysis. *Springer Real-Time Systems*, 54(3):607–661, July 2018. doi:10.1007/s11241-017-9285-4.

47 R. I. Davis and A. Burns. Robust Priority Assignment for Fixed Priority Real-Time Systems. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 3–14, December 2007. doi:10.1109/RTSS.2007.11.

48 R. I. Davis and A. Burns. Robust priority assignment for messages on Controller Area Network (CAN). *Springer Real-Time Systems*, 41(2):152–180, 2009. doi:10.1007/s11241-008-9065-2.

49 R. I Davis and A. Burns. A survey of hard real-time scheduling for multiprocessor systems. *ACM Computing Surveys (CSUR)*, 43(4):35, 2011.

50 R. I Davis, A. Burns, R. J. Bril, and J. J. Lukkien. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Springer Real-Time Systems*, 35(3):239–272, 2007.

51 R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3):239–272, 2007. doi:10.1007/s11241-007-9012-7.

52 R. I. Davis and L. Cucu-Grosjean. A Survey of Probabilistic Timing Analysis Techniques for Hard Real-Time Systems. *Leibniz Transactions on Embedded Systems (LITES)*, 6(1):03:1–03:60, May 2019. doi:10.4230/LITES-v006-i001-a003.

53 R. I. Davis, L. Santinelli, S. Altmeyer, C. Maiza, and L. Cucu-Grosjean. Analysis of Probabilistic Cache Related Pre-emption Delays. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 168–179, July 2013. doi:10.1109/ECRTS.2013.27.

54 J. L. Diaz, D. F. Garcia, K. Kim, C-G. Lee, L. Lo Bello, J. M. Lopez, S. L. Min, and O. Mirabella. Stochastic analysis of periodic real-time systems. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 289–300, 2002. doi:10.1109/REAL.2002.1181583.

55 J. L. Diaz, J. M. Lopez, M. Garcia, A. M. Campos, Kanghee Kim, and L. L. Bello. Pessimism in the stochastic analysis of real-time systems: concept and applications. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 197–207, December 2004. doi:10.1109/REAL.2004.41.

56 B. Doytchinov, J. Lehoczky, and S. Shreve. Real-time queues in heavy traffic with earliest-deadline-first queue discipline. *The Annals of Applied Probability*, 11(2):332–378, 2001. doi:10.1214/aoap/1015345295.

57 S. Draskovic, P. Huang, and L. Thiele. On the Safety of Mixed-Criticality Scheduling. In *Proceedings of Workshop on Mixed Criticality (WMC)*, 2016.

58 S. Edgar and A. Burns. Statistical analysis of WCET for scheduling. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 215–224, December 2001. doi:10.1109/REAL.2001.990614.

59 B. Frias, L. Palopoli, L. Abeni, and D. Fontanelli. Probabilistic Real-Time Guarantees: There is Life Beyond the i.i.d. Assumption. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2017.

60 M. K. Gardner and J. W. S. Liu. *Analyzing Stochastic Fixed-Priority Real-Time Systems*, pages 44–58. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. doi:10.1007/3-540-49059-0_4.

61 S. Gopalakrishnan. Sharp utilization thresholds for some real-time scheduling problems. *CoRR*, abs/0912.3852, 2009. URL: http://arxiv.org/abs/0912.3852.

62 D. Griffin, I. Bate, B. Lesage, and F. Soboczenski. Evaluating Mixed Criticality Scheduling Algorithms with Realistic Workloads. In *Proceedings of Workshop on Mixed Criticality (WMC)*, 2015.

63 H. Christian Gromoll and Łukasz Kruk. Heavy traffic limit for a processor sharing queue with soft deadlines. *The Annals of Applied Probability*, 17(3):1049–1101, June 2007. doi:10.1214/105051607000000014.

64 Z. Guo, L. Santinalli, and K. Yang. EDF Schedulability Analysis on Mixed-Criticality Systems with Permitted Failure Probability. In *Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2015.

65 C. J. Hamann, J. Loser, L. Reuther, S. Schonberg, J. Wolter, and H. Hartig. Quality-assuring scheduling-using stochastic behavior to improve resource utilization. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 119–128, December 2001. doi:10.1109/REAL.2001.990603.

66 J. Hansen, S. A. Hissam, and G. A. Moreno. Statistical-based WCET estimation and validation . In *Proceedings of the Workshop on Worst-Case Execution Time Analysis (WCET)*, volume 252, 2009.

67 J. P. Hansen, J. P. Lehoczky, H. Zhu, and R. Rajkumar. Quantized EDF Scheduling in a Stochastic Environment. In *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, IPDPS '02, pages 279–, Washington, DC, USA, 2002. IEEE Computer Society. URL: http://dl.acm.org/citation.cfm?id=645610.660905.

68 X. S. Hu, Tao Zhou, and E. H. M. Sha. Estimating probabilistic timing performance for real-time embedded systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(6):833–844, December 2001. doi:10.1109/92.974897.

69 S. Hua, G. Qu, and S. S. Bhattacharyya. Exploring the probabilistic design space of multimedia systems. In *in IEEE International Workshop on Rapid System Prototyping, 2003*, pages 233–240, 2003.

70 M. Ivers and R. Ernst. *Probabilistic Network Loads with Dependencies and the Effect on Queue Sojourn Times*, pages 280–296. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-10625-5_18.

71 M. Joseph and P. Pandya. Finding response times in a real-time system. *The Computer Journal*, 29(5):390–395, 1986.

72 G. A. Kaczynski, L. Lo Bello, and T. Nolte. Deriving exact stochastic response times of periodic tasks in hybrid priority-driven soft real-time systems. In *Proceedings of the IEEE Conference on Emerging Technologies Factory Automation (ETFA)*, pages 101–110, September 2007. doi:10.1109/EFTA.2007.4416759.

73 D. G. Kendall. Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain. *The Annals of Mathematical Statistics*, 24(3):338–354, September 1953. doi:10.1214/aoms/1177728975.

74 D. A. Khan, L. Santinelli, and L. Cucu-Grosjean. Modeling uncertainties in safety-critical real-time systems: A probabilistic component-based analysis. In *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 166–175, June 2012. doi:10.1109/SIES.2012.6356582.

75 J. K. Kim and B. K. Kim. Probabilistic Schedulability Analysis of Harmonic Multi-Task Systems with Dual-Modular Temporal Redundancy. *Springer Real-Time Systems*, 26(2):199–222, March 2004. doi:10.1023/B:TIME.0000016130.91111.75.

76 K. Kim, J. L. Diaz, L. Lo Bello, J. M. Lopez, C-G. Lee, and S. L. Min. An Exact Stochastic Analysis of Priority-Driven Periodic Real-Time Systems and Its Approximations. *IEEE Trans. Comput.*, 54(11):1460–1466, November 2005. doi:10.1109/TC.2005.174.

77 K. Kim, L. Lo Bello, S. L. Min, and O. Mirabella. On Relaxing Task Isolation in Overrun Handling to Provide Probabilistic Guarantees to Soft Real-Time Tasks with Varying Execution Times. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 193–, Washington, DC, USA, 2002. IEEE Computer Society. URL: http://dl.acm.org/citation.cfm?id=787256.787354.

78 G. Koren and D. Shasha. Skip-Over: algorithms and complexity for overloaded systems that allow skips. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 110–117, December 1995. doi:10.1109/REAL.1995.495201.

79 L. Kruk, J. Lehoczky, K. Ramanan, and S. Shreve. Heavy traffic analysis for EDF queues with reneging. *The Annals of Applied Probability*, 21(2):484–545, 2011. doi:10.1214/10-AAP681.

80 M. Kuttler, M. Roitzsch, C-J Hamann, and Marcus Volp. Probabilistic Analysis of Low-Criticality Execution. In *Proceedings of Workshop on Mixed Criticality (WMC)*, 2017.

81 J. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: exact characterization and average case behavior. In *Pro-*

ceedings of the IEEE Real-Time Systems Symposium (RTSS), pages 166–171, December 1989. doi:10.1109/REAL.1989.63567.

82 J. P. Lehoczky. Fixed priority scheduling of periodic task sets with arbitrary deadlines. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 201–209, December 1990. doi:10.1109/REAL.1990.128748.

83 J. P. Lehoczky. Real-time queueing theory. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 186–195, December 1996. doi:10.1109/REAL.1996.563715.

84 B. Lesage, D. Griffin, S. Altmeyer, L. Cucu-Grosjean, and R. I. Davis. On the analysis of random replacement caches using static probabilistic timing methods for multi-path programs. *Real-Time Systems*, December 2017. doi:10.1007/s11241-017-9295-2.

85 B. Lesage, D. Griffin, S. Altmeyer, and R. I. Davis. Static Probabilistic Timing Analysis for Multi-path Programs. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 361–372, December 2015. doi:10.1109/RTSS.2015.41.

86 A. Leulseged and N. Nissanke. Probabilistic Analysis of Multi-processor Scheduling of Tasks with Uncertain Parameters. In *Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 103–122, 2003. doi:10.1007/978-3-540-24686-2_7.

87 G. Lima and I. Bate. Valid Application of EVT in Timing Analysis by Randomising Execution Time Measurements. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2017.

88 G. Lima, D. Dias, and E. Barros. Extreme Value Theory for Estimating Task Execution Time Bounds: A Careful Look. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, July 2016.

89 C. L. Liu and J. W. Layland. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *J. ACM*, 20(1):46–61, January 1973. doi:10.1145/321738.321743.

90 Jane W. S. W. Liu. *Real-Time Systems*. Prentice Hall, 1st edition, 2000.

91 M. Liu, M. Behnam, and T. Nolte. An EVT-based worst-case Response Time Analysis of complex real-time systems. In *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 249–258, June 2013. doi:10.1109/SIES.2013.6601498.

92 R. Liu, A. F. Mills, and J. H. Anderson. Independence Thresholds: Balancing Tractability and Practicality in Soft Real-Time Stochastic Analysis. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 314–323, December 2014. doi:10.1109/RTSS.2014.38.

93 J. M. López, J. L. Díaz, J. Entrialgo, and D. García. Stochastic analysis of real-time systems under preemptive priority-driven scheduling. *Springer Real-Time Systems*, 40(2):180–207, 2008. doi:10.1007/s11241-008-9053-6.

94 Y. Lu, T. Nolte, I. Bate, and L. Cucu-Grosjean. A statistical response-time analysis of complex real-time embedded systems by using timing traces. In *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 43–46, June 2011. doi:10.1109/SIES.2011.5953676.

95 Y. Lu, T. Nolte, I. Bate, and L. Cucu-Grosjean. A Statistical Response-Time Analysis of Real-Time Embedded Systems. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 351–362, December 2012. doi:10.1109/RTSS.2012.85.

96 Y. Lu, T. Nolte, J. Kraft, and C. Norstrom. Statistical-Based Response-Time Analysis of Systems with Execution Dependencies between Tasks. In *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 169–179, March 2010. doi:10.1109/ICECCS.2010.55.

97 Y. Lu, T. Nolte, J. Kraft, and C. Norström. A Statistical Approach to Response-Time Analysis of Complex Embedded Real-Time Systems. In *Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 153–160, August 2010.

98 R. Mancuso, R. Pellizzoni, M. Caccamo, L. Sha, and H. Yun. WCET(m) Estimation in Multi-core Systems Using Single Core Equivalence. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 174–183, July 2015. doi:10.1109/ECRTS.2015.23.

99 N. Manica, L. Palopoli, and L. Abeni. Numerically efficient probabilistic guarantees for resource reservations. In *Proceedings of the IEEE Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–8, September 2012. doi:10.1109/ETFA.2012.6489566.

100 S. Manolache, P. Eles, and Z. Peng. Memory and Time-Efficient Schedulability Analysis of Task Sets with Stochastic Execution Time. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 19–, Washington, DC, USA, 2001. IEEE Computer Society. URL: http://dl.acm.org/citation.cfm?id=871910.871936.

101 S. Manolache, P. Eles, and Z. Peng. Schedulability Analysis of Applications with Stochastic Task Execution Times. *ACM Transactions on Embedded Computing Systems*, 3(4):706–735, November 2004. doi:10.1145/1027794.1027797.

102 S. Manolache, P. Eles, and Z. Peng. Task Mapping and Priority Assignment for Soft Real-time Applications Under Deadline Miss Ratio Constraints. *ACM Transactions on Embedded Computing Systems*, 7(2):19:1–19:35, January 2008. doi:10.1145/1331331.1331343.

103 F. Markovic, J. Carlson, R. Dobrin, B. Lisper, and A. Thekkilakattil. Probabilistic Response Time Analysis for Fixed Preemption Point Selection. In *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 1–10, June 2018. doi:10.1109/SIES.2018.8442099.

104 D. Maxim and A. Bertout. Analysis and Simulation Tools for Probabilistic Real-Time Systems. In *Proceedings of International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS)*, 2017.

105 D. Maxim, O. Buffet, L. Santinelli, L. Cucu-Grosjean, and R. I. Davis. Optimal Priority Assignment Algorithms for Probabilistic Real-Time Systems. In *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, pages 129–138, 2011.

106 D. Maxim and L. Cucu-Grosjean. Response Time Analysis for Fixed-Priority Tasks with Multiple Probabilistic Parameters. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 224–235, December 2013. doi:10.1109/RTSS.2013.30.

107 D. Maxim, R. I. Davis, L. Cucu-Grosjean, and A. Easwaran. Probabilistic Analysis for Mixed Criticality Scheduling with SMC and AMC. In *Proceedings of Workshop on Mixed Criticality (WMC)*. York, 2016.

108 D. Maxim, R. I. Davis, L. Cucu-Grosjean, and A. Easwaran. Probabilistic Analysis for Mixed Criticality Systems using Fixed Priority Preemptive Scheduling. In *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, 2017.

109 D. Maxim, M. Houston, L. Santinelli, G. Bernat, R. I. Davis, and L. Cucu-Grosjean. Re-sampling for Statistical Timing Analysis of Real-time Systems. In *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, pages 111–120, 2012. doi:10.1145/2392987.2393001.

110 D. Maxim, L. Santinelli, and L. Cucu-Grosjean. Improved sampling for statistical timing analysis of real-time systems. In *the 4th Junior Researcher Workshop on Real-Time Computing*, Toulouse, France, November 2010. URL: https://hal.inria.fr/inria-00544651.

111 D. Maxim, F. Soboczenski, I. Bate, and E. Tovar. Study of the Reliability of Statistical Timing Analysis for Real-time Systems. In *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, pages 55–64, 2015. doi:10.1145/2834848.2834878.

112 A. F. Mills and J. H. Anderson. A Stochastic Framework for Multiprocessor Soft Real-Time Scheduling. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 311–320, April 2010. doi:10.1109/RTAS.2010.33.

113 A. F. Mills and J. H. Anderson. A Multiprocessor Server-Based Scheduler for Soft Real-Time Tasks with Stochastic Execution Demand. In *Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, volume 1, pages 207–217, August 2011. doi:10.1109/RTCSA.2011.30.

114 S. Milutinovic, J. Abella, D. Hardy, E. Quiñones, I. Puaut, and F. J. Cazorla. Speeding up Static Probabilistic Timing Analysis. In *Proceedings of the International Conference on the Architecture of Computing Systems (ARCS)*, pages 236–247, March 2015. doi:10.1007/978-3-319-16086-3_19.

115 N. Tchidjo Moyo, E. Nicollet, F. Lafaye, and C. Moy. On Schedulability Analysis of Non-cyclic Generalized Multiframe Tasks. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 271–278, July 2010. doi:10.1109/ECRTS.2010.24.

116 N. Navet, L. Cucu, and R. Schott. Probabilistic Estimation of Response Times Through Large Deviations. In *Work-in Progress of the 28th IEEE Real-Time Systems Symposium (RTSS'2007 WiP)*, Tucson, United States, December 2007. URL: https://hal.inria.fr/inria-00191163.

117 N. Navet, Y.-Q. Song, and F. Simonot. Worst-case Deadline Failure Probability in Real-time Applications Distributed over Controller Area Network. *J. Syst. Archit.*, 46(7):607–617, April 2000. doi:10.1016/S1383-7621(99)00016-8.

118 N. Nissanke, A. Leulseged, and S. Chillara. Probabilistic performance analysis in multiprocessor scheduling. *Computing Control Engineering Journal*, 13(4):171–179, August 2002. doi:10.1049/cce:20020403.

119 T. Nolte, H. Hansson, and C. Norstrom. Probabilistic worst-case response-time analysis for the controller area network. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 200–207, May 2003. doi:10.1109/RTTAS.2003.1203052.

120 L. Palopoli, L. Abeni, and D. Fontanelli. A tool for the optimal design of soft real–time systems. In *Proceedings of International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS)*, pages 31–36, 2014.

121 L. Palopoli, D. Fontanelli, L. Abeni, and B. V. Frías. An Analytical Solution for Probabilistic Guarantees of Reservation Based Soft Real-Time Systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(3):640–653, March 2016. doi:10.1109/TPDS.2015.2416732.

122 L. Palopoli, D. Fontanelli, N. Manica, and L. Abeni. An Analytical Bound for Probabilistic Deadlines. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 179–188, July 2012. doi:10.1109/ECRTS.2012.19.

123 S. S. Panwar, D. Towsley, and J. K. Wolf. Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service. *J. ACM*, 35(4):832–844, October 1988. doi:10.1145/48014.48019.

124 S. Punnekkat, R. I. Davis, and A. Burns. Sensitivity analysis of real-time task sets. In *Annual Asian Computing Science Conference*, pages 72–82. Springer Berlin Heidelberg, 1997.

125 S. Quinton, R. Ernst, D. Bertrand, and P. Meumeu Yomsi. Challenges and new trends in probabilistic timing analysis. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 810–815, March 2012. doi:10.1109/DATE.2012.6176605.

126 P. Ramanathan and M. Hamdaoui. A Dynamic Priority Assignment Technique for Streams with

(M, K)-Firm Deadlines. *IEEE Transactions on Computers*, 44(12):1443–1451, December 1995. doi:10.1109/12.477249.

127 K. S. Refaat and P. E. Hladik. Efficient Stochastic Analysis of Real-Time Systems via Random Sampling. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 175–183, July 2010. doi:10.1109/ECRTS.2010.29.

128 J. Ren, R. Bi, X. Su, Q. Liu, G. Wu, and G. Tan. Workload-aware harmonic partitioned scheduling for probabilistic real-time systems. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 213–218, March 2018. doi:10.23919/DATE.2018.8342005.

129 L. Santinelli. Probabilistic Component-based Analysis for Networks: Invited Paper. *SIGBED Rev.*, 13(3):65–72, August 2016. doi:10.1145/2983185.2983197.

130 L. Santinelli and L. Cucu-Grosjean. Toward Probabilistic Real-time Calculus. *SIGBED Rev.*, 8(1):54–61, March 2011. doi:10.1145/1967021.1967028.

131 L. Santinelli and L. Cucu-Grosjean. A Probabilistic Calculus for Probabilistic Real-Time Systems. *ACM Transactions on Embedded Computing Systems*, 14(3):52:1–52:30, April 2015. doi:10.1145/2717113.

132 L. Santinelli and L. George. Probabilities and Mixed-Criticalities: the Probabilistic C-Space. In *Proceedings of Workshop on Mixed Criticality (WMC)*, 2015.

133 L. Santinelli, F. Guet, and J. Morio. Revising Measurement-Based Probabilistic Timing Analysis. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2017.

134 L. Santinelli, Z. Guo, and L. George. Fault-aware sensitivity analysis for probabilistic real-time systems. In *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pages 69–74, September 2016. doi:10.1109/DFT.2016.7684072.

135 L. Santinelli, J. Morio, G. Dufour, and D. Jacquemart. On the Sustainability of the Extreme Value Theory for WCET Estimation. In *Proceedings of the Workshop on Worst-Case Execution Time Analysis (WCET)*, pages 21–30, 2014. doi:10.4230/OASIcs.WCET.2014.21.

136 L. Santinelli, P. M. Yomsi, D. Maxim, and L. Cucu-Grosjean. A component-based framework for modeling and analyzing probabilistic real-time systems. In *Proceedings of the IEEE Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–8, September 2011. doi:10.1109/ETFA.2011.6059013.

137 L. Sha, T. Abdelzaher, K-E. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A. K. Mok. Real time scheduling theory: A historical perspective. *RTSJ*, 28(2-3):101–155, 2004.

138 M. Short and J. Proenza. Towards Efficient Probabilistic Scheduling Guarantees for Real-Time Systems Subject to Random Errors and Random Bursts of Errors. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 259–268, July 2013. doi:10.1109/ECRTS.2013.35.

139 B. Sprunt, L. Sha, and J. Lehoczky. Aperiodic task scheduling for Hard-Real-Time systems. *Springer Real-Time Systems*, 1(1):27–60, 1989. doi:10.1007/BF02341920.

140 B. Tanasa, U. D. Bordoloi, P. Eles, and Z. Peng. Probabilistic Timing Analysis for the Dynamic Segment of FlexRay. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 135–144, July 2013. doi:10.1109/ECRTS.2013.24.

141 B. Tanasa, U. D. Bordoloi, P. Eles, and Z. Peng. Probabilistic Response Time and Joint Analysis of Periodic Tasks. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 235–246, July 2015. doi:10.1109/ECRTS.2015.28.

142 L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. In *IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century*, volume 4, pages 101–104 vol.4, May 2000. doi:10.1109/ISCAS.2000.858698.

143 T. S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L. C. Wu, and J. W. S. Liu. Probabilistic performance guarantee for real-time tasks with varying computation times. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 164–173, May 1995. doi:10.1109/RTTAS.1995.516213.

144 S. R. S. Varadhan. Large deviations. *The Annals of Probability*, 36(2):397–419, March 2008. doi:10.1214/07-AOP348.

145 S. Vestal. Preemptive Scheduling of Multi-criticality Systems with Varying Degrees of Execution Time Assurance. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 239–243, 2007. doi:10.1109/RTSS.2007.47.

146 B. Villalba Frías, L. Palopoli, L. Abeni, and D. Fontanelli. The PROSIT tool: Toward the optimal design of probabilistic soft real-time systems. *Software: Practice and Experience*, 0(0), 2018. doi:10.1002/spe.2604.

147 G. von der Brüggen, N. Piatkowski, K-H. Chen, J. J. Chen, and K. Morik. Efficiently Approximating the Probability of Deadline Misses in Real-Time Systems. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, volume 106, pages 6:1–6:22, 2018. doi:10.4230/LIPIcs.ECRTS.2018.6.

148 T. Wang, S. Homsi, L. Nui, S. Ren, O. Bai, G. Quan, and M. Qiu. Harmonicity Aware Task Partitioning for Fixed Priority Scheduling of Probabilistic Real-Time Tasks on Multi-Core Platforms. *ACM Transactions on Embedded Computing Systems*, 2016.

149 T. Wang, L. Niu, S. Ren, and G. Quan. Multi-core fixed-priority scheduling of real-time tasks with statistical deadline guarantee. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pages 1335–1340, March 2015.

150 F. Wartel, L. Kosmidis, C. Lo, B. Triquet, E. Quiñones, J. Abella, A. Gogonel, A. Baldovin, E. Mezzetti, L. Cucu, T. Vardanega, and F. J. Cazorla. Measurement-based probabilistic timing analysis: Lessons from an integrated-modular avionics case study. In *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 241–248, June 2013. doi:10.1109/SIES.2013.6601497.

151 R. C. Williamson and T. Downs. Probabilistic arithmetic. I. Numerical methods for calculating convolutions and dependency bounds. *International Journal of Approximate Reasoning*, 4(2):89–158, 1990. doi:10.1016/0888-613X(90)90022-T.

152 M. H. Woodbury and K. G. Shin. Evaluation of the probability of dynamic failure and processor utilization for real-time systems. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 222–231, December 1988. doi:10.1109/REAL.1988.51117.

153 H. Zeng, M. Di Natale, P. Giusto, and A. Sangiovanni-Vincentelli. Stochastic Analysis of CAN-Based Real-Time Automotive Systems. *IEEE Transactions on Industrial Informatics*, 5(4):388–401, November 2009. doi:10.1109/TII.2009.2032067.

154 H. Zhu, J. P. Hansen, J. P. Lehoczky, and R. Rajkumar. Optimal partitioning for quantized EDF scheduling. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 212–222, 2002. doi:10.1109/REAL.2002.1181576.