CABS: Bounding the Consecutive Deadline-Failure Probability of Periodic Tasks via Backlog Sampling

Oleksandr Marmaliuk* Mario Günzel* Bite Ye* Filip Marković† Björn B. Brandenburg*

*Max Planck Institute for Software Systems, Saarland Informatics Campus, Germany
†University of Southampton, United Kingdom

***Abstract*—Stochastic schedulability analysis is quickly gaining relevance due to several converging trends: the inability to obtain sound WCET bounds on modern hardware platforms, the prevalence of systems that can tolerate infrequent deadline misses, and a desire to avoid provisioning on a worst-case basis for efficiency and cost reasons. However, all available stochastic schedulability analyses come with at least one of the following limitations: they (i) assume that all execution times are independent (which is unrealistic), (ii) require that tardy jobs are aborted at their deadline (which is uncommon and difficult to support in practice), (iii) are limited to constrained or implicit deadlines, (iv) support only a specific scheduling policy, or (v) are limited to the analysis of single deadline misses.**

Aiming to address all of these limitations, this paper introduces the first *correlation-aware analysis* for bounding the probability of *multiple consecutive deadline misses* in periodic real-time systems with *arbitrary deadlines*, while allowing tardy jobs to complete (*i.e.*, *without job abortion*), under uniprocessor *fixed-priority* (FP), *earliest-deadline first* (EDF), and *first-in/first-out* (FIFO) scheduling. The key technique enabling the proposed analysis is the use of *sampled backlog bounds* that account for the pent-up demand of both tardy and regular carry-in jobs, and which can be statistically inferred via non-parametric bootstrapping.

A simulator-based evaluation with synthetic workloads shows that the proposed analysis improves on the only prior correlation-aware analysis (CAA), which is specific to FP scheduling with job abortion. The paper further explores the proposed analysis in settings without job abortion, under all considered scheduling policies, for diverse task counts, and with varying relative deadlines. In the industrially relevant case of multiple consecutive deadline misses, the analysis is shown to yield less pessimistic bounds than a single-miss analysis in settings with few tasks.

I. Introduction

Many, if not most, real-time systems found in practice today are *soft*, in the sense that they can tolerate occasional or bounded deadline misses [2]. Additionally, many real-time workloads are deployed on hardware that renders effective *worst-case execution time* (WCET) analysis impractical, in particular when they require fast multicore processors or GPUs.

For such systems, traditional schedulability analysis techniques rooted in WCET-based reasoning are a poor fit: they require workload parameters that cannot be soundly obtained, and, due to their focus on worst-case scenarios, inherently yield pessimistic bounds. Such pessimism, however, ultimately results in over-provisioning of costly system resources, which is not only unappealing from an efficiency perspective, but often also not economically viable due to market pressures.

The research community has responded to this mismatch by developing a stochastic perspective on timeliness [13, 14]. Treating execution times in a real-time system as random variables offers several advantages: it is possible to *quantify* the likelihood that a system will violate its timing requirements, measurement procedures can be made *statistically rigorous*, and the resulting *residual uncertainty* can be precisely characterized, analyzed, and ultimately bounded. From a practical point of view, stochastic analysis methods hold the promise of allowing systems to be provisioned below worst-case needs in a formally justifiable manner, based on informed and rigorously evaluated cost-timeliness tradeoff decisions. However, despite much progress in recent years (see Sec. IX), the state of the art still does not fully support this vision, as currently available methods have limitations that pose major practical challenges.

Early work on stochastic real-time systems generally assumed independence between execution times of jobs from the same and different tasks (*e.g.*, [15, 16, 23, 48, 52]), which clearly do not hold in practice (*e.g.*, due to shared caches, shared program state, and producer-consumer relationships in which one task processes the output of another). While these independence assumptions have been relaxed in more recent work, either by requiring dependence-masking *probabilistic WCET* (pWCET) distributions [5] as analysis inputs or by accounting for dependencies explicitly [41, 42], most such analyses assume that tardy jobs are immediately aborted and removed from the system upon missing their deadlines. This is difficult to support in practice (*e.g.*, due to the risk of incomplete and hence inconsistent updates of shared state), and most operating systems do not offer such a capability.

However, when tardy jobs continue executing after their deadlines, unfinished work can accumulate as a *backlog*, which can in turn delay subsequent jobs. The precise analysis of such a backlog is challenging: current approaches either assume mutually independent execution times [15, 16, 28, 34, 53, 54], require exact and identical execution-time distributions for each job [27], or focus on a single, isolated task [1, 18–22].

Additionally, most existing analyses are limited to tasks with *implicit* or *constrained deadlines* (*e.g.*, [11, 26, 41, 42, 47]). Another common restriction is that, with few exceptions [26, 28, 50], most analyses consider only *fixed-priority* (FP) scheduling, with alternative policies such as *earliest-deadline first* (EDF) or *first-in/first-out* (FIFO) receiving comparatively little attention.

Finally, most existing analyses focus on the occurrence of a single deadline miss as the sole event of interest. Many systems encountered in practice, however, can tolerate a small number of *consecutive* deadline misses [2, Question 15]. For example,

control applications are frequently designed to be robust to two or more consecutive deadline misses [24, 38, 46, 51]. More specifically, specifications that permit some number of consecutive deadline misses are a special case of *weakly-hard* timing requirements [3] for which robust controllers can be designed [38, 46], and are thus of particular practical interest.

This paper. Taking a step towards increased practicality, we explore a new approach for bounding the deadline-failure probability of periodic tasks that addresses all of the above limitations in a single analysis. Our method—*correlation-aware backlog sampling* (CABS)—differs from prior analyses in that it *samples* the backlog *directly* (by means of non-parametric bootstrapping). By avoiding the need to derive a backlog bound from task parameters, CABS circumvents a number of analytical hurdles that limit prior approaches.

In particular, as the backlog accounts for both tardy and regular carry-in jobs alike, our analysis does *not* require tardy jobs to be aborted. Additionally, our general analysis setup naturally supports *arbitrary deadlines* (*i.e.*, multiple pending activations per task) and extends to all three major uniprocessor scheduling policies (FP, EDF, and FIFO). Finally, our bound is parametric in the number of consecutive misses under analysis, and hence seamlessly supports bounding the probability of *multiple consecutive deadline failures*, which is of particular industrial relevance, as already discussed above.

In our work, we make the following contributions:

- Secs. III and IV introduce a ground-truth system model with an explicit definition of *per-job backlog* and the concept of a *horizon-exceedance probability* that generalizes the notion of consecutive deadline misses.
- Sec. V derives the proposed CABS approach as Theorem 3, which bounds the probability that all jobs in any sequence of m consecutive jobs of a given task miss their deadlines.
- Sec. VI explains how all parameters required to apply Theorem 3 can be obtained via statistical inference.
- Sec. VII reports on a simulator-based evaluation of CABS with synthetic workloads and establishes two findings: 1) in a restricted setting where a prior analysis for FP scheduling with job abortion [42] is applicable, CABS achieves comparable or better bounds, which demonstrates that using sampled backlog bounds can reduce pessimism, and 2) factors such as whether relative deadlines exceed periods, the number of permissible consecutive deadline misses, and the choice of scheduling policy all impact the final bound and thus should receive further attention.

We begin by laying the necessary mathematical foundations.

II. Background and Definitions

Consider a *probability space* $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the set of all possible outcomes, $\mathcal{F} \subseteq 2^\Omega$ is the event space, and $\mathbb{P}\colon \mathcal{F} \to [0,1]$ is a probability function. A *random variable* X on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a function $X\colon \Omega \to \mathbb{R}$ with $\{\omega \in \Omega \mid X(\omega) = x\} \in \mathcal{F}$ for all $x \in \mathbb{R}$. We use $\mathbb{P}[X = x]$ to denote the probability of the random variable X taking the value x, i.e., $\mathbb{P}[X = x] = \mathbb{P}[\{\omega \in \Omega \mid X(\omega) = x\}]$. Similarly, we use $\mathbb{P}[X > x]$ and $\mathbb{P}[X < x]$ to denote the probability of X being greater than or less than x, respectively.

Given a random variable X, its *expected value* $\mathbb{E}[X] \triangleq \int_{\omega\in\Omega} X(\omega)\,\mathrm{d}\mathbb{P}$ is a measure of the central tendency of its possible outcomes. The *covariance* $\mathrm{Cov}[X,Y] \triangleq \mathbb{E}[(X - \mathbb{E}[X]) \cdot (Y - \mathbb{E}[Y])]$ of two random variables X and Y measures the degree to which X and Y fluctuate together. Intuitively, when $\mathrm{Cov}[X,Y] > 0$, then an *increase* of X is associated with an *increase* of Y. Conversely, if $\mathrm{Cov}[X,Y] < 0$, then an *increase* of X predicts a *decrease* of Y. Finally, a random variable's *variance* $\mathrm{Var}[X] \triangleq \mathrm{Cov}[X,X]$ reflects the spread of its possible outcomes.

We say that two random variables X and Y are *correlated* if $\mathrm{Cov}[X,Y] \neq 0$; correlated random variables are necessarily *dependent*. By the *linearity of expectation* (*e.g.*, [25, p. 40]), the expected value of a sum of possibly dependent random variables is simply the sum of their individual expectations: $\mathbb{E}[a \cdot X + b \cdot Y] = a \cdot \mathbb{E}[X] + b \cdot \mathbb{E}[Y]$.

In contrast, this is not generally the case when considering the variance of a sum of random variables, where possible dependence cannot be ignored. To bound the probability that a sum of possibly dependent random variables (*e.g.*, execution times) exceeds a given threshold (*e.g.*, a deadline), we therefore rely on a correlation-aware concentration bound derived from *Cantelli's inequality* [7] by Marković et al. [42].

Fact 1 ([42, Theorem 3])**.** *Let $\{X_i\}_{i=1}^n$ be a finite set of possibly dependent random variables with $\mathrm{Cov}[X_i, X_j] < \infty$ for all $i, j \in \{1, \ldots, n\}$. If $\hat{e} \geq \sum_{i=1}^n \mathbb{E}[X_i]$ and $\hat{c} \geq \sum_{i,j=1}^n \mathrm{Cov}[X_i, X_j]$, then, for any $t > \hat{e}$,*

$$\mathbb{P}\left[\sum_{i=1}^{n} X_i \geq t\right] \leq \frac{\hat{c}}{\hat{c} + (t - \hat{e})^2}\,.$$

Following Marković et al. [42], we use *non-parametric bootstrapping* [17] to infer the statistical parameters of interest for a given workload, such as the expected values, variances, and covariances of execution costs. Non-parametric bootstrapping works as follows. Given n observations $O = \{o_1, o_2, \ldots, o_n\}$ of a random variable X with an unknown distribution, let $T\colon \mathbb{R}^n \to \mathbb{R}$ denote a statistic of interest (*e.g.*, the sample mean, variance, or covariance). The bootstrap procedure constructs *pseudo-samples* $O_1^*, O_2^*, O_3^*, \ldots$, each of size n, by sampling with replacement from O. For each pseudo-sample O_k^*, the statistic of interest $\theta_k^* = T(O_k^*)$ is computed. Repeating this process B times yields an empirical *bootstrap distribution* $\{\theta_1^*, \theta_2^*, \ldots, \theta_B^*\}$, which approximates the true sampling distribution of $\theta = T(X)$.

From the bootstrap distribution, one can obtain a *confidence interval* $[\theta^-, \theta^+]$ that contains the true statistic θ with a chosen confidence level γ. For example, for a confidence level of 99.9% (*i.e.*, $\gamma = 0.999$), the bootstrapping procedure is expected to fail (*i.e.*, to yield an interval $[\theta^-, \theta^+]$ that does *not* contain the true statistic θ) only once in 1,000 applications.

With the mathematical foundation in place, we next introduce a stochastic ground-truth model of the system under analysis.

III. Ground-Truth System

We consider a set of n periodic tasks $\tau = \{\tau_1, \tau_2, \ldots, \tau_n\}$ scheduled on a preemptive uniprocessor, where each task τ_i is characterized by its *relative deadline* D_i, *period* T_i, and *offset* Φ_i. We make no assumption about how D_i, T_i, and Φ_i relate (*i.e.*, we allow for arbitrary deadlines and offsets). While we assume a uniprocessor system for simplicity, our analysis can be applied without modifications to each core of a partitioned multiprocessor (where each core is scheduled individually).

Following prior work [5, 15, 41], we interpret the set of all possible outcomes Ω to represent the set of potentially observable *execution traces* (or *system evolutions*) of task system τ. The set of all possible system evolutions Ω induces the underlying probability space that we analyze.

The j-th job of task τ_i, denoted $J_{i,j}$, *arrives* at time $a_{i,j} = \Phi_i + (j-1)T_i$ and has its *absolute deadline* at time $d_{i,j} \triangleq a_{i,j} + D_i$. As we focus on periodic tasks, each job arrives at the same time in all system evolutions. Job execution costs, however, may vary across system evolutions. Each job's *execution cost* $c_{i,j}(\omega)$ is hence a distinct random variable; we say that $J_{i,j}$ has cost $c_{i,j}(\omega)$ in system evolution $\omega \in \Omega$. We assume that time is discrete: all arrival times, execution times, and deadlines are integer-valued (*e.g.*, processor cycles).

A job is *tardy* if it does not finish execution by its absolute deadline. For the sake of generality, we assume that tardy jobs are *not* aborted (*i.e.*, all jobs run to completion, if necessary after their deadlines), but our analysis remains valid if (some) jobs are aborted upon missing their deadlines.

Table I provides a summary of our notation.

A. Schedule and Service

We consider three classic work-conserving scheduling policies: *earliest-deadline first* (EDF), *fixed-priority* (FP), and *first-in-first-out* (FIFO) scheduling. To abstract over the specific policy in use, we assume a priority relation $\preceq^\pi$ for $\pi \in \{\text{EDF}, \text{FP}, \text{FIFO}\}$, where $J_{k,\ell} \preceq^\pi J_{i,j}$ means that $J_{k,\ell}$'s priority is higher than or equal to $J_{i,j}$'s priority under policy π.

Def. 1 (priority relation)**.** For any two jobs $J_{k,\ell}$ and $J_{i,j}$:

$$J_{k,\ell} \preceq^{\text{EDF}} J_{i,j} \Rightarrow d_{k,\ell} \leq d_{i,j}$$
$$J_{k,\ell} \preceq^{\text{FP}} J_{i,j} \Rightarrow k \leq i$$
$$J_{k,\ell} \preceq^{\text{FIFO}} J_{i,j} \Rightarrow a_{k,\ell} \leq a_{i,j}$$

We require $\preceq^\pi$ to be a total order that reflects that tasks are *sequential* (*i.e.*, jobs of the same task necessarily execute in arrival order). Ties in priority among jobs of distinct tasks are broken arbitrarily but consistently (*e.g.*, by task index); we assume $\preceq^\pi$ to already reflect all such tie-breaks. Under the FP policy, we assume that tasks are indexed by decreasing priority (*i.e.*, τ_1 holds the highest priority). We further assume that all scheduling and preemption overheads are accounted for implicitly (*i.e.*, included in each job's execution cost $c_{i,j}(\omega)$).

The system always executes the pending job (if any) with the highest priority according to $\preceq^\pi$. The resulting *schedule* is a mapping $\sigma : \mathbb{N} \times \Omega \to \mathbb{J} \cup \{\bot\}$, where $\mathbb{J}$ is the set of all jobs and $\bot$ indicates that the processor is idle; $\sigma(t, \omega)$ indicates the scheduled job (or $\bot$) at time t in evolution ω.

A job's *cumulative service* received prior to time $t \in \mathbb{N}$ is given by $S_{i,j}(t, \omega) \triangleq \sum_{u=0}^{t-1} \mathbb{I}_{i,j}(u, \omega)$, where $\mathbb{I}_{i,j}(u, \omega) = 1$ if $\sigma(u, \omega) = J_{i,j}$ and $\mathbb{I}_{i,j}(u, \omega) = 0$ otherwise. The *remaining processing time* of job $J_{i,j}$ at time t is thus given by $\rho_{i,j}(t, \omega) \triangleq c_{i,j}(\omega) - S_{i,j}(t, \omega)$. A job is *pending* at time t if it has arrived ($t \geq a_{i,j}$) and remains incomplete ($\rho_{i,j}(t, \omega) > 0$). We will repeatedly use the following trivial observation: if $t \leq a_{i,j}$, then $\rho_{i,j}(t, \omega) = c_{i,j}(\omega)$, that is, before a job arrives, its remaining processing time is simply its full execution requirement.

For notational convenience, we let $\oplus_t^\omega[\, S \,]$ denote the *cumulative remaining processing time* of a set of jobs S in evolution ω at time t:

$$\oplus_t^\omega[\, S \,] \triangleq \sum_{J_{i,j} \in S} \rho_{i,j}(t, \omega). \tag{1}$$

By definition, $\oplus_t^\omega[\, S \,]$ is trivially additive for disjoint sets: For any two sets of jobs A and B with $A \cap B = \varnothing$, and any t:

$$\oplus_t^\omega[\, A \cup B \,] = \oplus_t^\omega[\, A \,] + \oplus_t^\omega[\, B \,]. \tag{2}$$

B. Backlog and Interference

When considering a given job, we distinguish two sources of competing workload: 1) the *backlog* that already exists *when the job arrives* (also known as *carry-in work*), and 2) any work that arrives *with* or *after* the job, up to a given *horizon of interest*.

Def. 2 (backlog jobs)**.** For a reference job $J_{k,\ell}$, the set of jobs potentially contributing to the backlog at $J_{k,\ell}$'s arrival is

$$\mathcal{B}(J_{k,\ell}) \triangleq \{\, J_{i,j} \mid a_{i,j} < a_{k,\ell} \ \wedge \ J_{i,j} \preceq^\pi J_{k,\ell} \,\}.$$

Def. 3 (horizon-bounded interfering jobs)**.** Given a reference job $J_{k,\ell}$ and a relative horizon $h > 0$, the set of higher-or-equal-priority jobs arriving during $[a_{k,\ell},\ a_{k,\ell} + h)$ is

$$\mathcal{I}(J_{k,\ell}, h) \triangleq \{\, J_{i,j} \mid a_{k,\ell} \leq a_{i,j} < a_{k,\ell} + h \ \wedge \ J_{i,j} \preceq^\pi J_{k,\ell} \,\}.$$

Note that $\mathcal{I}(J_{k,\ell}, h)$ includes $J_{k,\ell}$ itself if $h > 0$, which is convenient when we compute the cumulative remaining processing time of jobs in $\mathcal{B}(J_{k,\ell}) \cup \mathcal{I}(J_{k,\ell}, h)$, as done next.

C. Finish and Response Time

Given a job $J_{k,\ell}$, let $W(h) \triangleq \oplus_{a_{k,\ell}}^\omega[\, \mathcal{B}(J_{k,\ell}) \cup \mathcal{I}(J_{k,\ell}, h) \,]$. Under the considered work-conserving, priority-driven scheduling policies, if $c_{k,\ell}(\omega) > 0$, $J_{k,\ell}$'s *finish time* is

$$f_{k,\ell}(\omega) \triangleq \inf\{\, F > a_{k,\ell} \mid F \geq a_{k,\ell} + W(F - a_{k,\ell}) \,\}.$$

Otherwise, we set $f_{k,\ell}(\omega) \triangleq a_{k,\ell}$. Work conservation ensures that the processor does not idle while higher-or-equal-priority work is pending. The *ground-truth response time* of job $J_{k,\ell}$ is thus $r_{k,\ell}(\omega) \triangleq f_{k,\ell}(\omega) - a_{k,\ell}$.

IV. Horizon Exceedance Probability

Ultimately, we seek to bound the probability of one or more jobs missing their respective deadlines. More generally, given a response-time threshold (or relative *horizon*) for each job in a sequence of consecutive jobs, we are interested in

TABLE I
SUMMARY OF NOTATION

Symbol	Meaning
τ	the analyzed task set
τ_i	the i-th task in τ
T_i	period of τ_i
D_i	relative deadline of τ_i
Φ_i	offset of τ_i
$J_{i,j}$	the j-th job (or activation) of task τ_i
$a_{i,j}$	arrival time of job $J_{i,j}$
$d_{i,j}$	absolute deadline of job $J_{i,j}$
$\preceq^{\pi}$	priority relation among jobs
$\mathcal{B}(J_{i,j})$	set of jobs potentially contributing to the backlog of $J_{i,j}$
$\mathcal{I}(J_{i,j}, h)$	horizon-bounded set of jobs interfering with $J_{i,j}$
Ω	set of all potentially observable execution traces
ω	one execution trace (or, equivalently, system evolution)
$c_{i,j}(\omega)$	execution cost of job $J_{i,j}$ in system evolution ω
$\sigma(t, \omega)$	the job scheduled at time t (if any) or $\perp$ in evolution ω
$S_{i,j}(t, \omega)$	cumulative processor service received by $J_{i,j}$ before time t
$\rho_{i,j}(t, \omega)$	remaining processing time of job $J_{i,j}$ at time t
$\oplus_t^{\omega}[\mathcal{S}]$	cumulative remaining processing time of all jobs in set $\mathcal{S}$
$r_{i,j}(\omega)$	ground-truth response time of job $J_{i,j}$ in execution trace ω
$\vec{H}$	horizon vector
$\mathrm{HEP}_{k,\ell}^{m}(\vec{H})$	joint horizon-exceedance probability of m consecutive jobs
$\mathrm{DFP}_k(m)$	maximum probability of m consecutive deadline misses

the probability that all jobs in the sequence finish after their respective horizon points. We next formalize these events and introduce notation to express them concisely.

A. Definition of the Horizon Exceedance Probability

Consider a sequence of m jobs $J_{k,\ell}, J_{k,\ell+1}, \ldots, J_{k,\ell+m-1}$ and a given *horizon vector* $\vec{H} = \langle H_0, H_1, \ldots, H_{m-1} \rangle$. Our goal is to bound the probability that each job in the sequence exhibits a response time in excess of its corresponding horizon.

Def. 4. For $m \geq 1$, the *joint horizon-exceedance probability* is

$$\mathrm{HEP}_{k,\ell}^{m}(\vec{H}) \triangleq \mathbb{P}\left[\left\{\omega \in \Omega \;\middle|\; \bigwedge_{\ell \leq q < \ell+m} r_{k,q}(\omega) > H_{q-\ell}\right\}\right].$$

In the following, we assume that the absolute horizon time points are monotone: $a_{k,\ell+i} + H_i \leq a_{k,\ell+i+1} + H_{i+1}$ for all $i \in \{0, \ldots, m-2\}$. Equivalently, since consecutive jobs of task τ_k are separated by T_k, this condition can be written as

$$H_i \leq H_{i+1} + T_k \qquad \text{for all } i \in \{0, \ldots, m-2\}. \tag{3}$$

In the special case where $\vec{H} = \langle D_k, \ldots, D_k \rangle \in \mathbb{N}^m$, this yields the probability of τ_k experiencing at least m consecutive deadline misses starting with a specific job $J_{k,\ell}$. The maximum over all possible starting points (*i.e.*, $\ell \geq 1$) thus gives us the worst-case probability that a job of task τ_k starts a sequence of m consecutive deadline misses.

Def. 5. Task τ_k's *consecutive deadline-failure probability* is

$$\mathrm{DFP}_k(m) \triangleq \max_{\ell \geq 1} \mathrm{HEP}_{k,\ell}^{m}(\vec{H})$$

with $\vec{H} = \langle D_k, \ldots, D_k \rangle \in \mathbb{N}^m$. For $m = 1$, $\mathrm{DFP}_k(m)$ reduces to the classical (single-job) *deadline-failure probability* (DFP) studied in prior work, e.g., [41, 42].

B. An Upper Bound on the Horizon Exceedance Probability

As we seek to upper-bound $\mathrm{HEP}_{k,\ell}^{m}(\vec{H})$, we begin by establishing a necessary condition for the event to occur.

Lemma 1. *Consider m jobs $J_{k,\ell}$, $J_{k,\ell+1}$, $\ldots$, $J_{k,\ell+m-1}$ and a corresponding horizon vector $\vec{H} = \langle H_0, H_1, \ldots, H_{m-1} \rangle$. If the joint horizon-exceedance condition $\bigwedge_{\ell \leq q < \ell+m} (r_{k,q}(\omega) > H_{q-\ell})$ holds, then necessarily*

$$\oplus_{a_{k,\ell}}^{\omega}\left[\bigcup_{\ell \leq q < \ell+m} \big(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell})\big)\right] > \left|\bigcup_{\ell \leq q < \ell+m} \big[a_{k,q},\, a_{k,q} + H_{q-\ell}\big)\right|,$$

where $|\cdot|$ denotes the total length of the union of time intervals.

Proof. First consider a single job $J_{k,q}$, where $\ell \leq q < \ell + m$. Since by assumption $r_{k,q}(\omega) > H_{q-\ell}$, job $J_{k,q}$ remains pending throughout $[a_{k,q}, a_{k,q} + H_{q-\ell})$. The use of a preemptive, priority-driven scheduler rules out the execution of lower-priority jobs during this interval, while work conservation excludes idle time. Thus, the processor executes only higher-or-equal-priority jobs (w.r.t. $J_{k,q}$) during the interval $[a_{k,q}, a_{k,q} + H_{q-\ell})$. These jobs arrive either before or during the interval, that is, they come from the set $\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell})$. We conclude that only jobs from the set $\mathcal{J} \triangleq \bigcup_{\ell \leq q < \ell+m} (\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell}))$ are executed during the joint window $\mathcal{W} \triangleq \bigcup_{\ell \leq q < \ell+m} [a_{k,q}, a_{k,q} + H_{q-\ell})$.

By monotonicity of the absolute horizon points, $t_e \triangleq a_{k,\ell+m-1} + H_{m-1}$ is the last such point. As only jobs from $\mathcal{J}$ are executed during $\mathcal{W}$, the remaining cumulative processing time of $\mathcal{J}$ at time t_e has decreased by at least $|\mathcal{W}|$ time units compared to time $a_{k,\ell}$: $\oplus_{a_{k,\ell}}^{\omega}[\mathcal{J}] - \oplus_{t_e}^{\omega}[\mathcal{J}] \geq |\mathcal{W}|$.

Additionally, since by assumption $r_{k,\ell+m-1}(\omega) > H_{m-1}$, the job $J_{k,\ell+m-1} \in \mathcal{J}$ is still incomplete at time t_e.

$$\rho_{k,\ell+m-1}(t_e,\, \omega) > 0$$

Hence, $\oplus_{t_e}^{\omega}[\mathcal{J}] > 0$, and therefore $\oplus_{a_{k,\ell}}^{\omega}[\mathcal{J}] > |\mathcal{W}|$, which yields the claimed bound after substituting $\mathcal{J}$ and $\mathcal{W}$. □

We next simplify the total interval length.

Lemma 2. *Under the assumption of Eq.* (3),

$$\left|\bigcup_{\ell \leq q < \ell+m} \big[a_{k,q},\, a_{k,q} + H_{q-\ell}\big)\right| = \left(\sum_{i=0}^{m-2} \min(H_i,\, T_k)\right) + H_{m-1}.$$

Proof. By Eq. (3), $a_{k,q} + H_{q-\ell} \leq a_{k,q+1} + H_{q+1-\ell}$ holds for any $q \in \{\ell, \ldots, \ell + m - 2\}$. Hence, $\bigcup_{\ell \leq q < \ell+m} [a_{k,q}, a_{k,q} + H_{q-\ell})$ can be restated equivalently as:

$$\bigcup_{\ell \leq q < \ell+m-1} \big[a_{k,q}, \min(a_{k,q} + H_{q-\ell}, a_{k,q+1})\big) \cup [a_{k,\ell+m-1}, a_{k,\ell+m-1} + H_{m-1})$$

Since the intervals in this union are disjoint by construction, its total length is the sum of the individual interval lengths: $\sum_{\ell \leq q < \ell+m-1} \big|\big[a_{k,q}, \min(a_{k,q} + H_{q-\ell}, a_{k,q+1})\big)\big| + |[a_{k,\ell+m-1}, a_{k,\ell+m-1} + H_{m-1})|$. For the latter summand, we have $|[a_{k,\ell+m-1}, a_{k,\ell+m-1} + H_{m-1})| = H_{m-1}$. Furthermore, for $q \in \{\ell, \ldots, \ell + m - 2\}$, we have $a_{k,q+1} -$

$a_{k,q} = T_k$ since jobs arrive periodically, and therefore $\left|\left[a_{k,q}, \min(a_{k,q} + H_{q-\ell}, a_{k,q+1})\right)\right| = \min(H_{q-\ell}, a_{k,q+1} - a_{k,q}) = \min(H_{q-\ell}, T_k)$. The claim follows since $i = q-\ell$. □

Combining Lemmas 1 and 2, we obtain an upper bound on the joint horizon-exceedance probability.

Lemma 3. $\mathrm{HEP}_{k,\ell}^{m}(\vec{H}) \leq$

$$\mathbb{P}\left[\begin{array}{r} \oplus_{a_{k,\ell}}^{\omega}\Big[\bigcup_{\ell \leq q < \ell+m}\big(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell})\big)\Big] \\ > \Big(\sum_{i=0}^{m-2} \min(H_i,\, T_k)\Big) + H_{m-1} \end{array}\right]$$

Proof. The bound follows directly from Lemmas 1 and 2:

$$\begin{aligned}
&\mathrm{HEP}_{k,\ell}^{m}(\vec{H}) \\
&\overset{\text{(i)}}{=} \mathbb{P}\left[\left\{\omega \in \Omega \;\middle|\; \textstyle\bigwedge_{\ell \leq q < \ell+m} r_{k,q}(\omega) > H_{q-\ell}\right\}\right] \\
&\overset{\text{(ii)}}{\leq} \mathbb{P}\left[\begin{array}{r} \oplus_{a_{k,\ell}}^{\omega}\Big[\bigcup_{\ell \leq q < \ell+m}\big(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell})\big)\Big] \\ > \Big|\bigcup_{\ell \leq q < \ell+m}\big[a_{k,q},\, a_{k,q} + H_{q-\ell}\big)\Big| \end{array}\right] \\
&\overset{\text{(iii)}}{=} \mathbb{P}\left[\begin{array}{r} \oplus_{a_{k,\ell}}^{\omega}\Big[\bigcup_{\ell \leq q < \ell+m}\big(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell})\big)\Big] \\ > \Big(\sum_{i=0}^{m-2} \min(H_i,\, T_k)\Big) + H_{m-1} \end{array}\right],
\end{aligned}$$

where (i) unfolds Def. 4, (ii) follows from Lemma 1, and (iii) rewrites the total interval length using Lemma 2. □

Lemma 3 provides a starting point for our analysis, but still makes use of dynamic job-level parameters, namely each job's remaining processing time in Eq. (1), and hence is not directly applicable in *a priori* analysis. We therefore next turn our attention to upper-bounding the probability term in Lemma 3 using statically obtainable task-level parameters.

V. Correlation-Aware Backlog Analysis

The term $\oplus_{a_{k,\ell}}^{\omega}\Big[\bigcup_{\ell \leq q < \ell+m}\big(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell})\big)\Big]$ in Lemma 3, which we refer to below as the *workload of the considered job sequence*, is a sum of random variables. To apply Cantelli's inequality to this sum (via Fact 1), we need to bound the corresponding covariance and mean terms (*i.e.*, $\hat{c}$ and $\hat{e}$ in Fact 1) using task-level parameters.

To this end, we proceed in four steps: first, we introduce the task-level parameters that we rely on in Sec. V-A; second, we decompose the workload variable in Sec. V-B; then we derive upper bounds on the covariance and mean values in Sec. V-C; and finally, we combine Lemma 3 with these building blocks to obtain a *correlation-aware backlog analysis* in Sec. V-D.

A. Inputs to the Static Analysis

We base our bound on the same task-level *analysis inputs* as previously used by Marković et al. [42]. For each task τ_i:

Input 1: $\widehat{e}_i$ upper-bounds the mean execution time of any job of τ_i: $\forall j,\ \mathbb{E}\left[c_{i,j}(\omega)\right] \leq \widehat{e}_i$.

Input 2: $\widehat{s}_i^c$ upper-bounds the execution time variance of any job of τ_i: $\forall j,\ \mathrm{Var}\left[c_{i,j}(\omega)\right] \leq \widehat{s}_i^c$.

Input 3: $\widehat{v}_{i,i}$ upper-bounds the *intra-task* covariance of the execution times of any two distinct jobs of τ_i: $\forall j \neq j',\ \mathrm{Cov}\left[c_{i,j}(\omega), c_{i,j'}(\omega)\right] \leq \widehat{v}_{i,i}$.

Furthermore, for any two distinct tasks $\tau_i \neq \tau_k$:

Input 4: $\widehat{v}_{i,k}$ upper-bounds the *inter-task* covariance of the execution times of any two jobs of τ_i and τ_k: $\forall j, \ell,\ \mathrm{Cov}\left[c_{i,j}(\omega), c_{k,\ell}(\omega)\right] \leq \widehat{v}_{i,k}$.

In addition to Analysis Inputs 1–4, we further require the following parameters related to the backlog, which were not used in prior work. For each task τ_i:

Input 5: $\widehat{b}_i$ upper-bounds the mean of the backlog at the arrival of any job of τ_i: $\forall \ell,\ \mathbb{E}\left[\oplus_{a_{i,\ell}}^{\omega}[\,\mathcal{B}(J_{i,\ell})\,]\right] \leq \widehat{b}_i$.

Input 6: $\widehat{s}_i^b$ upper-bounds the variance of the backlog at the arrival of any job of τ_i: $\forall \ell,\ \mathrm{Var}\left[\oplus_{a_{i,\ell}}^{\omega}[\,\mathcal{B}(J_{i,\ell})\,]\right] \leq \widehat{s}_i^b$.

Finally, for any two tasks τ_i, τ_k:

Input 7: $\widehat{v}_{i,k}^b$ upper-bounds the covariance between the backlog at the arrival of any job of τ_i and the job execution time of any job of τ_k: $\forall j, \ell,\ \mathrm{Cov}\left[\oplus_{a_{i,j}}^{\omega}[\,\mathcal{B}(J_{i,j})\,], c_{k,\ell}(\omega)\right] \leq \widehat{v}_{i,k}^b$.

All analysis inputs are assumed to be finite, non-negative constants. Sec. VI explains how to derive the required analysis inputs via statistical inference for a given system.

B. Workload Decomposition

Before we can apply the just-introduced parameters, we must decompose the workload into random variables corresponding to either the backlog or individual job execution times. For this, we rely on three simple properties that follow directly from Defs. 2 and 3 and the considered scheduling policies. For any $\ell \leq q \leq r < \ell + m$, we have:

$$\mathcal{B}(J_{k,q}) \subseteq \mathcal{B}(J_{k,r}) \tag{4}$$

$$\mathcal{I}(J_{k,q}, H_{q-\ell}) \subseteq \mathcal{B}(J_{k,r}) \cup \mathcal{I}(J_{k,r}, H_{r-\ell}) \tag{5}$$

$$\mathcal{B}(J_{k,q}) \cap \mathcal{I}(J_{k,q}, H_{q-\ell}) = \varnothing \tag{6}$$

With these properties, the workload can be rewritten as follows.

Lemma 4. *For any $\ell \geq 1$ and $m \geq 1$:*

$$\begin{aligned}
&\textstyle\bigcup_{\ell \leq q < \ell+m}\big(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell})\big) \\
&\qquad = \mathcal{B}(J_{k,\ell+m-1}) \cup \mathcal{I}(J_{k,\ell+m-1}, H_{m-1})
\end{aligned}$$

Proof. We establish that both sides are subsets of each other. The direction '$\supseteq$' trivially holds since $\mathcal{B}(J_{k,\ell+m-1}) \cup \mathcal{I}(J_{k,\ell+m-1}, H_{m-1})$ is part of the union with $q = \ell + m - 1$. Regarding the reverse direction '$\subseteq$', for each $q \in \{\ell, \ldots, \ell + m - 1\}$, the set $\mathcal{B}(J_{k,q})$ is a subset of $\mathcal{B}(J_{k,\ell+m-1}) \cup \mathcal{I}(J_{k,\ell+m-1}, H_{m-1})$ by Eq. (4), and the set $\mathcal{I}(J_{k,q}, H_{q-\ell})$ is a subset of $\mathcal{B}(J_{k,\ell+m-1}) \cup \mathcal{I}(J_{k,\ell+m-1}, H_{m-1})$ by Eq. (5). □

In the next step, we isolate the backlog $\oplus_{a_{k,\ell}}^{\omega}[\,\mathcal{B}(J_{k,\ell})\,]$, as this is the random variable that Analysis Inputs 5–7 provide bounds on. We collect all other jobs contributing to the workload of the considered job sequence in the set

$$\mathcal{S} \triangleq \left(\mathcal{B}(J_{k,\ell+m-1}) \setminus \mathcal{B}(J_{k,\ell})\right) \cup \mathcal{I}(J_{k,\ell+m-1}, H_{m-1})\,. \tag{7}$$

By construction, Eq. (7) satisfies the following properties.

Lemma 5. *Every job $J_{i,j} \in \mathcal{S}$ **(i)** has a higher-or-equal priority w.r.t. $J_{k,\ell+m-1}$, **(ii)** arrives before $a_{k,\ell+m-1} + H_{m-1}$ (i.e., $a_{i,j} < a_{k,\ell+m-1} + H_{m-1}$), and either **(iii-a)** arrives no earlier than $a_{k,\ell}$ (i.e., $a_{i,j} \geq a_{k,\ell}$) or **(iii-b)** has lower priority than $J_{k,\ell}$.*

Proof. Consider a job $J_{i,j} \in \mathcal{S}$. Then $J_{i,j} \in \mathcal{B}(J_{k,\ell+m-1}) \cup \mathcal{I}(J_{k,\ell+m-1}, H_{m-1})$, and hence (i) and (ii) follow directly from Defs. 2 and 3. Additionally, $J_{i,j} \notin \mathcal{B}(J_{k,\ell})$, which, by Def. 2, implies that either (iii-a) or (iii-b) must hold. □

By combining Eqs. (2) and (7), we can decompose the workload into the backlog and job execution times as follows.

Lemma 6. *The workload of the considered job sequence is upper-bounded by the backlog plus a sum of job execution times.*

$$\oplus^{\omega}_{a_{k,\ell}} \left[\bigcup_{\ell \leq q < \ell+m} \left(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell}) \right) \right] \leq \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,] + \textstyle\sum_{J_{i,j} \in \mathcal{S}} c_{i,j}(\omega).$$

Proof. Consider the following sequence of (in)equalities:

$$\begin{aligned}
&\oplus^{\omega}_{a_{k,\ell}} \left[\bigcup_{\ell \leq q < \ell+m} \left(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell}) \right) \right] \\
&\quad \overset{\text{(i)}}{=} \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell+m-1}) \cup \mathcal{I}(J_{k,\ell+m-1}, H_{m-1}) \,] \\
&\quad \overset{\text{(ii)}}{=} \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \cup \mathcal{S} \,] \\
&\quad \overset{\text{(iii)}}{=} \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,] + \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{S} \,] \\
&\quad \overset{\text{(iv)}}{=} \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,] + \textstyle\sum_{J_{i,j} \in \mathcal{S}} \rho_{i,j}(a_{k,\ell}, \omega) \\
&\quad \overset{\text{(v)}}{\leq} \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,] + \textstyle\sum_{J_{i,j} \in \mathcal{S}} c_{i,j}(\omega),
\end{aligned}$$

where (i) applies Lemma 4, (ii) restates the workload using Eq. (7), (iii) applies Eq. (2) since, by construction, $\mathcal{B}(J_{k,\ell}) \cap \mathcal{S} = \varnothing$, and (iv) unfolds Eq. (1). Finally, (v) holds because $\rho_{i,j}(t, \omega) \leq c_{i,j}(\omega)$ for any job $J_{i,j}$ and any time t. □

The random variables that appear in the final decomposition of the workload in Lemma 6 are exactly those that are bounded *individually* by the analysis inputs listed in Sec. V-A. However, the mean and covariance of their sum remain to be bounded.

C. Bounds on Mean and Covariance

Consider the set of all random variables appearing in the decomposition of the workload in Lemma 6:

$$RV \triangleq \{ \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,] \} \cup \{ c_{i,j} \mid J_{i,j} \in \mathcal{S} \}. \tag{8}$$

Clearly, $\sum_{X \in RV} X = \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,] + \sum_{J_{i,j} \in \mathcal{S}} c_{i,j}(\omega)$. To apply Fact 1 to this sum of random variables, we thus require bounds on $\sum_{X \in RV} \mathbb{E}\,[X]$ and $\sum_{X,X' \in RV} \mathrm{Cov}\,[X, X']$.

By the linearity of expectation and Analysis Inputs 1 and 5, an upper bound on $\sum_{X \in RV} \mathbb{E}\,[X]$ is directly given by

$$\textstyle\sum_{X \in RV} \mathbb{E}\,[X] \leq \widehat{b}_k + \sum_{J_{i,j} \in \mathcal{S}} \widehat{e}_i. \tag{9}$$

Obtaining an upper bound on $\sum_{X,X' \in RV} \mathrm{Cov}\,[X, X']$ requires more careful investigation. We start by splitting the sum into four different categories of summands.

Lemma 7. $\sum_{X,X' \in RV} \mathrm{Cov}\,[X, X'] = A + B + C + D$, *where:*

$$A \triangleq \mathrm{Var}\left[\oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,] \right] + \textstyle\sum_{J_{i,j} \in \mathcal{S}} \mathrm{Var}\,[c_{i,j}(\omega)] \tag{10}$$

$$B \triangleq \sum_{i=1}^{n} \sum_{J_{i,j} \in \mathcal{S}} \sum_{\substack{J_{i,g} \in \mathcal{S} \\ g \neq j}} \mathrm{Cov}\,[c_{i,j}(\omega), c_{i,g}(\omega)] \tag{11}$$

$$C \triangleq \sum_{i=1}^{n} \sum_{\substack{h=1 \\ h \neq i}}^{n} \sum_{J_{i,j} \in \mathcal{S}} \sum_{J_{h,g} \in \mathcal{S}} \mathrm{Cov}\,[c_{i,j}(\omega), c_{h,g}(\omega)] \tag{12}$$

$$D \triangleq 2 \cdot \textstyle\sum_{J_{i,j} \in \mathcal{S}} \mathrm{Cov}\left[\oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,], c_{i,j}(\omega) \right] \tag{13}$$

Proof. We proceed by case analysis, sorting the summands as follows. First, for summands $\mathrm{Cov}\,[X, X']$ with $X = X' \in RV$, by the definition of variance, $\mathrm{Cov}\,[X, X'] = \mathrm{Var}\,[X]$. These summands are collected in A.

Second, the remaining summands all involve pairs of distinct random variables. The summands $\mathrm{Cov}\,[X, X']$ where $X \neq X'$ and $X = \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,]$ or $X' = \oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,]$ are collected in D. The factor 2 in Eq. (13) arises from the symmetry $\mathrm{Cov}\,[X, X'] = \mathrm{Cov}\,[X', X]$.

We are left with the summands $\mathrm{Cov}\,[X, X']$ where $X \neq X'$ and neither X nor X' equal $\oplus^{\omega}_{a_{k,\ell}} [\, \mathcal{B}(J_{k,\ell}) \,]$. By the definition of RV, both X and X' are then execution variables of jobs in $\mathcal{S}$. There are two possible sub-cases: If both X and X' stem from jobs of the *same task*, they are collected in B, whereas if they stem from jobs of *different tasks*, they are collected in C. This covers all possible cases, which establishes the claim. □

Since Eqs. (10)–(13) can be bounded using Analysis Inputs 2–4, 6 and 7, what remains to be done is to restate the set $\mathcal{S}$ defined in Eq. (7) in a statically computable form. As this is necessarily policy-specific (recall Def. 1), we introduce a bound x^{π}_i on the number of jobs of τ_i that occur in $\mathcal{S}$ under scheduling algorithm $\pi \in \{\text{EDF}, \text{FP}, \text{FIFO}\}$: $x^{\pi}_i \geq |\mathcal{S} \cap \{J_{i,j} | j \in \mathbb{N}\}|$.

Lemma 8. *The following policy-specific bounds x^{π}_i hold:*

$$x^{\text{FP}}_i \triangleq \begin{cases} \left\lceil \frac{(m-1)T_k + H_{m-1}}{T_i} \right\rceil & \text{if } i \leq k \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$x^{\text{FIFO}}_i \triangleq \left\lceil \frac{(m-1)T_k + 1}{T_i} \right\rceil \tag{15}$$

$$x^{\text{EDF}}_i \triangleq \begin{cases} \left\lceil \frac{\max(D_k - D_i + 1, 0)}{T_i} \right\rceil & \text{if } m = 1 \\ \left\lceil \frac{(m-1)T_k + \max(D_k - D_i, 0) + 1}{T_i} \right\rceil & \text{otherwise} \end{cases} \tag{16}$$

Proof. Under FP scheduling, jobs of task τ_i with $i > k$ have lower priority than $J_{k,\ell+m-1}$. Hence, $x^{\text{FP}}_i = 0$ for $i > k$ by Lemma 5(i). For τ_i with $i \leq k$, all jobs have priority no lower than that of $J_{k,\ell+m-1}$, so Lemma 5(iii-b) does not apply. Hence, by Lemma 5(ii) and Lemma 5(iii-a), $\mathcal{S}$ only contains jobs that arrive during the half-open interval $[a_{k,\ell}, a_{k,\ell+m-1} + H_{m-1})$, which has a length of $a_{k,\ell+m-1} + H_{m-1} - a_{k,\ell} = (m-1)T_k + H_{m-1}$. The number of jobs of task τ_i with $i \leq k$ that arrive in the relevant interval is thus bounded by $\left\lceil \frac{(m-1)T_k + H_{m-1}}{T_i} \right\rceil$.

Under FIFO scheduling, jobs that arrive after $a_{k,\ell+m-1}$ have lower priority than $J_{k,\ell+m-1}$, and jobs that arrive before $a_{k,\ell}$

have higher priority than $J_{k,\ell}$, so Lemma 5(iii-b) does not apply. Thus, all jobs of task τ_i satisfying Lemma 5(i) and Lemma 5(iii-a) arrive during the closed interval $[a_{k,\ell}, a_{k,\ell+m-1}]$, which has length $a_{k,\ell+m-1} - a_{k,\ell} + 1 = (m-1)T_k + 1$. Thus, there are at most $\left\lceil \frac{(m-1)T_k+1}{T_i} \right\rceil$ such jobs.

Under EDF scheduling, jobs with an absolute deadline after $a_{k,\ell+m-1} + D_k$ have lower priority than $J_{k,\ell+m-1}$. Thus, to satisfy Lemma 5(i), a job of task τ_i cannot arrive after time $a_{k,\ell+m-1} + D_k - D_i$. To also satisfy Lemma 5(iii), it cannot arrive before $a_{k,\ell}$ under (iii-a) or before $a_{k,\ell} + D_k - D_i$ under (iii-b). We consider two cases: $m = 1$ and $m > 1$.

For $m = 1$, $J_{k,\ell} = J_{k,\ell+m-1}$, and thus Lemma 5(i) and Lemma 5(iii-b) cannot hold simultaneously. Hence, a job $J_{i,j} \in \mathcal{S}$ must satisfy Lemma 5(i) and Lemma 5(iii-a), which implies $a_{k,\ell} \le a_{i,j} \le a_{k,\ell} + D_k - D_i$. There are at most $\left\lceil \frac{\max(D_k - D_i + 1, 0)}{T_i} \right\rceil$ such jobs of τ_i.

For $m > 1$, a job $J_{i,j} \in \mathcal{S}$ must satisfy Lemma 5(i) and Lemma 5(iii) and thus arrive during the closed interval $[\min(a_{k,\ell},\, a_{k,\ell} + D_k - D_i),\, a_{k,\ell+m-1} + D_k - D_i]$. If $D_i \le D_k$, this interval has a length of $a_{k,\ell+m-1} + D_k - D_i - a_{k,\ell} + 1 = (m-1)T_k + D_k - D_i + 1$. Otherwise, this interval has a length of $a_{k,\ell+m-1} + D_k - D_i - (a_{k,\ell} + D_k - D_i) + 1 = (m-1)T_k + 1$. Combining the two cases, the interval length is bounded by $(m-1)T_k + \max(D_k - D_i, 0) + 1$, so there are at most $\left\lceil \frac{(m-1)T_k + \max(D_k - D_i, 0) + 1}{T_i} \right\rceil$ jobs of τ_i in $\mathcal{S}$. □

Combining the above lemma with Eq. (9) and Lemma 7, we arrive at the following bounds on the sums of the expectations and covariances of the random variables in RV.

Lemma 9. *Given RV as defined in Eq. (8), we have:*

$$\sum_{X \in RV} \mathbb{E}[X] \le \widehat{E}^{\pi}(\vec{H})$$
$$\sum_{X, X' \in RV} \mathrm{Cov}[X, X'] \le \widehat{C}^{\pi}(\vec{H})$$

where

$$\widehat{E}^{\pi}(\vec{H}) \triangleq \widehat{b}_k + \sum_{i=1}^{n} x_i^{\pi} \cdot \widehat{e}_i, \text{ and} \tag{17}$$

$$\begin{aligned} \widehat{C}^{\pi}(\vec{H}) \triangleq\ & \widehat{s}_k^b + \sum_{i=1}^{n} x_i^{\pi} \cdot \widehat{s}_i^c \\ & + \sum_{i=1}^{n} x_i^{\pi} \cdot (x_i^{\pi} - 1) \cdot \widehat{v}_{i,i} \\ & + 2 \sum_{h=1}^{n} \sum_{i=1}^{h-1} x_i^{\pi} \cdot x_h^{\pi} \cdot \widehat{v}_{i,h} \\ & + 2 \sum_{i=1}^{n} x_i^{\pi} \widehat{v}_{k,i}^b. \end{aligned} \tag{18}$$

Proof. Eq. (17) follows directly from Eq. (9) since by Lemma 8 $\mathcal{S}$ contains at most x_i^{π} jobs of τ_i for each $i \in \{1, \ldots, n\}$, that is, $\sum_{J_{i,j} \in \mathcal{S}} \widehat{e}_i \le \sum_{i=1}^{n} x_i^{\pi} \cdot \widehat{e}_i$. We derive Eq. (18) from Lemma 7 by considering each of the parts A, B, C, and D individually.

***A*:** By Analysis Input 6, $\mathrm{Var}\left[\oplus_{a_{k,\ell}}^{\omega}[\mathcal{B}(J_{k,\ell})]\right] \le \widehat{s}_k^b$. Additionally, $\sum_{J_{i,j} \in \mathcal{S}} \mathrm{Var}[c_{i,j}(\omega)] \le \sum_{i=1}^{n} x_i^{\pi} \cdot \widehat{s}_i^c$, since by Lemma 8 there are at most x_i^{π} jobs of τ_i in $\mathcal{S}$, and by Analysis Input 2 each $\mathrm{Var}[c_{i,j}(\omega)]$ is bounded by $\widehat{s}_i^c$. Hence, $A \le \widehat{s}_k^b + \sum_{i=1}^{n} x_i^{\pi} \cdot \widehat{s}_i^c$.

***B*:** Observe that, for each task τ_i, there is one summand in B for each ordered pair of distinct jobs of τ_i in $\mathcal{S}$. It follows from Lemma 8 that there are at most $x_i^{\pi} \cdot (x_i^{\pi} - 1)$ such pairs. By Analysis Input 3, each summand $\mathrm{Cov}[c_{i,j}(\omega), c_{i,g}(\omega)]$ is bounded by $\widehat{v}_{i,i}$. Hence, $B \le \sum_{i=1}^{n} x_i^{\pi} \cdot (x_i^{\pi} - 1) \cdot \widehat{v}_{i,i}$.

***C*:** For every ordered pair of distinct tasks τ_i and τ_h, the sum C contains one summand $\mathrm{Cov}[c_{i,j}(\omega), c_{h,g}(\omega)]$ for each corresponding pair of jobs of τ_i and τ_h. By Lemma 8, there are at most $x_i^{\pi} \cdot x_h^{\pi}$ such pairs of jobs in $\mathcal{S}$. Additionally, by first combining symmetric covariance terms ($\mathrm{Cov}[c_{i,j}(\omega), c_{h,g}(\omega)] = \mathrm{Cov}[c_{h,g}(\omega), c_{i,j}(\omega)]$) to pull out a factor of two and then applying Analysis Input 4, we obtain $C \le 2 \sum_{h=1}^{n} \sum_{i=1}^{h-1} x_i^{\pi} \cdot x_h^{\pi} \cdot \widehat{v}_{i,h}$.

***D*:** For each task τ_i, there is a summand in D of the form $\mathrm{Cov}\left[\oplus_{a_{k,\ell}}^{\omega}[\mathcal{B}(J_{k,\ell})], c_{i,j}(\omega)\right]$ for each job of τ_i in $\mathcal{S}$. By Lemma 8, there are no more than x_i^{π} such jobs. By Analysis Input 7, $\mathrm{Cov}\left[\oplus_{a_{k,\ell}}^{\omega}[\mathcal{B}(J_{k,\ell})], c_{i,j}(\omega)\right]$ is bounded by $\widehat{v}_{k,i}^b$. Hence, we obtain $D \le 2 \sum_{i=1}^{n} x_i^{\pi} \widehat{v}_{k,i}^b$. □

With Lemma 9, we have all necessary elements in place.

D. Correlation-Aware Backlog Analysis

We now wrap up our analysis and put all the building blocks together, starting with the horizon-exceedance probability.

Theorem 1. *For any scheduling policy $\pi \in \{$EDF, FP, FIFO$\}$, any horizon vector $\vec{H}$ of length $m \ge 1$, and any task $\tau_k \in \tau$, if $\Delta(\vec{H}) > \widehat{E}^{\pi}(\vec{H})$, then the joint horizon-exceedance probability $\mathrm{HEP}_{k,\ell}^{m}(\vec{H})$ for any job $J_{k,\ell}$ of τ_k is bounded as follows:*

$$\mathrm{HEP}_{k,\ell}^{m}(\vec{H}) \le \frac{\widehat{C}^{\pi}(\vec{H})}{\widehat{C}^{\pi}(\vec{H}) + \left(\Delta(\vec{H}) - \widehat{E}^{\pi}(\vec{H})\right)^2},$$

where $\widehat{E}^{\pi}(\vec{H})$ and $\widehat{C}^{\pi}(\vec{H})$ are given by Eqs. (17) and (18), respectively, and

$$\Delta(\vec{H}) \triangleq \left(\sum_{i=0}^{m-2} \min(H_i,\, T_k)\right) + H_{m-1}. \tag{19}$$

Proof. From Lemmas 3 and 6, we obtain:

$$\begin{aligned} & \mathrm{HEP}_{k,\ell}^{m}(\vec{H}) \\ & \overset{\text{(i)}}{\le} \mathbb{P}\left[\oplus_{a_{k,\ell}}^{\omega}\left[\bigcup_{\ell \le q < \ell+m} \left(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell})\right)\right] > \Delta(\vec{H})\right] \\ & \overset{\text{(ii)}}{\le} \mathbb{P}\left[\oplus_{a_{k,\ell}}^{\omega}[\mathcal{B}(J_{k,\ell})] + \sum_{J_{i,j} \in \mathcal{S}} c_{i,j}(\omega) > \Delta(\vec{H})\right] \\ & \overset{\text{(iii)}}{=} \mathbb{P}\left[\sum_{X \in RV} X > \Delta(\vec{H})\right] \overset{\text{(iv)}}{\le} \mathbb{P}\left[\sum_{X \in RV} X \ge \Delta(\vec{H})\right] \\ & \overset{\text{(v)}}{\le} \frac{\widehat{C}^{\pi}(\vec{H})}{\widehat{C}^{\pi}(\vec{H}) + \left(\Delta(\vec{H}) - \widehat{E}^{\pi}(\vec{H})\right)^2} \end{aligned}$$

Here, (i) is Lemma 3 after substituting Eq. (19). Let $A \triangleq \left\{\omega \in \Omega \mid \oplus_{a_{k,\ell}}^{\omega}\left[\bigcup_{\ell \le q < \ell+m} \left(\mathcal{B}(J_{k,q}) \cup \mathcal{I}(J_{k,q}, H_{q-\ell})\right)\right] > \Delta(\vec{H})\right\}$ and $B \triangleq \left\{\omega \in \Omega \mid \oplus_{a_{k,\ell}}^{\omega}[\mathcal{B}(J_{k,\ell})] + \sum_{J_{i,j} \in \mathcal{S}} c_{i,j}(\omega) > \Delta(\vec{H})\right\}$. By Lemma 6, we have $A \subseteq B$, and so (ii) follows from monotonicity of the probability measure $\mathbb{P}$. Step (iii) then restates the sum of random variables using Eq. (8), (iv) relaxes the inequality since $>$ implies $\ge$, and finally (v) applies Fact 1 with $\hat{e} = \widehat{E}^{\pi}(\vec{H})$ and $\hat{c} = \widehat{C}^{\pi}(\vec{H})$ as justified by Lemma 9 and the assumption that $\Delta(\vec{H}) > \widehat{E}^{\pi}(\vec{H})$. □

From Theorem 1, we obtain a bound on $\mathrm{DFP}_k(m)$. Let

$$\mathcal{H}_k^m \triangleq \{\langle H_0, \ldots, H_{m-1}\rangle \mid 1 \le H_i \le D_k,\, 0 \le i < m\} \quad (20)$$

be the set of all horizon vectors of length m with positive components not exceeding the relative deadline D_k.

Theorem 2. *For any scheduling policy $\pi \in \{EDF, FP, FIFO\}$, any $m \ge 1$, any task $\tau_k \in \tau$, and any horizon vector $\vec{H} \in \mathcal{H}_k^m$, if $\Delta(\vec{H}) > \widehat{E}^\pi(\vec{H})$, then*

$$\mathrm{DFP}_k(m) \;\le\; \frac{\widehat{C}^\pi(\vec{H})}{\widehat{C}^\pi(\vec{H}) + \left(\Delta(\vec{H}) - \widehat{E}^\pi(\vec{H})\right)^2}.$$

Proof. From Theorem 1 and Eq. (20), we have:

$$\begin{aligned}
\mathrm{DFP}_k(m) &\overset{\text{(i)}}{=} \max\nolimits_{\ell \ge 1} \mathrm{HEP}_{k,\ell}^m(\langle D_k, \ldots, D_k\rangle) \\
&\overset{\text{(ii)}}{\le} \max\nolimits_{\ell \ge 1} \mathrm{HEP}_{k,\ell}^m(\vec{H}) \\
&\overset{\text{(iii)}}{\le} \max\nolimits_{\ell \ge 1} \frac{\widehat{C}^\pi(\vec{H})}{\widehat{C}^\pi(\vec{H}) + \left(\Delta(\vec{H}) - \widehat{E}^\pi(\vec{H})\right)^2} \\
&\overset{\text{(iv)}}{=} \frac{\widehat{C}^\pi(\vec{H})}{\widehat{C}^\pi(\vec{H}) + \left(\Delta(\vec{H}) - \widehat{E}^\pi(\vec{H})\right)^2},
\end{aligned}$$

where (i) is Def. 5, (ii) follows since exceeding the horizon vector $\langle D_k, \ldots, D_k\rangle$ also implies exceeding any vector in $\mathcal{H}_k^m$, (iii) applies Theorem 1, and (iv) drops the $\max$ operator since ℓ does not occur in the expression being maximized. □

Theorem 2 bounds $\mathrm{DFP}_k(m)$ for any admissible horizon vector $\vec{H}$. To obtain a practical analysis, we search through the space of such vectors and report the least resulting bound.

Given the structure of Eqs. (17)–(19), the search space of relevant horizon vectors can be restricted without loss of accuracy. Under FP scheduling, only horizon vectors whose first $m-1$ components are D_k are relevant, that is, vectors of the form $\langle D_k, D_k, \ldots, D_k, H_{m-1}\rangle$ with $1 \le H_{m-1} \le D_k$. Additionally, the search space can be limited to $H_{m-1} = D_k$ and values of H_{m-1} at which Eq. (14) "takes a step," that is, values of H_{m-1} such that $\left\lceil \frac{(m-1)T_k + H_{m-1}}{T_i} \right\rceil \ne \left\lceil \frac{(m-1)T_k + H_{m-1} + 1}{T_i} \right\rceil$ for some $i \le k$.

Under FIFO and EDF scheduling, x_i^π does not depend on $\vec{H}$ at all (recall Eqs. (15) and (16)), so it suffices to consider only the horizon vector $\langle D_k, \ldots, D_k\rangle$ as used in Def. 5.

We summarize these observations with our final theorem.

Theorem 3 (CABS analysis)**.** *For any scheduling policy $\pi \in \{EDF, FP, FIFO\}$, any $m \ge 1$, and any task $\tau_k \in \tau$, we have*

$$\mathrm{DFP}_k(m) \;\le\; \inf_{\substack{\vec{H} \in Q^\pi \\ \Delta(\vec{H}) > \widehat{E}^\pi(\vec{H})}} \frac{\widehat{C}^\pi(\vec{H})}{\widehat{C}^\pi(\vec{H}) + \left(\Delta(\vec{H}) - \widehat{E}^\pi(\vec{H})\right)^2},$$

where $Q^{EDF} = Q^{FIFO} \triangleq \{\langle D_k, \ldots, D_k\rangle\}$ and $Q^{FP} \triangleq \{\langle D_k, \ldots, D_k\rangle\} \cup \left\{ \langle D_k, \ldots, D_k, H_{m-1}\rangle \,\middle|\, H_{m-1} \le D_k \wedge \exists\, i \le k,\, \left\lceil \frac{(m-1)T_k + H_{m-1}}{T_i} \right\rceil \ne \left\lceil \frac{(m-1)T_k + H_{m-1} + 1}{T_i} \right\rceil \right\}$.

Proof. If no $\vec{H} \in Q^\pi$ satisfies $\Delta(\vec{H}) > \widehat{E}^\pi(\vec{H})$, then $\inf$ yields ∞ and the claim is trivially true. Otherwise, the bound follows directly from Theorem 2 since, by construction, $Q^\pi \subset \mathcal{H}_k^m$. □

We next explain how to obtain Analysis Inputs 1–7.

VI. Statistical Parameter Estimation

Following Marković et al. [42], we rely on non-parametric bootstrapping, reviewed in Sec. II, to statistically estimate Analysis Inputs 1–7 with a configurable high degree of confidence.

To apply bootstrapping to our system of periodic tasks, we first set an *observation limit* $\mathcal{L}$ that determines the number of observed jobs. This can be, for example, a multiple of the task set's hyperperiod. Given $\mathcal{L}$, we record N independent execution traces $\mathcal{D} = \{s_1, \ldots, s_N\}$ of the system (*e.g.*, by rebooting the system in between observations). Each trace contains information on all jobs released prior to $\mathcal{L}$ (*i.e.*, each job $J_{i,j}$ such that $a_{i,j} < \mathcal{L}$) and all scheduling decisions. When $\mathcal{L}$ is reached, the system is allowed to continue executing normally until all jobs released before $\mathcal{L}$ are run to completion. Thus, the same set of completed jobs is observed in all traces.

The information contained in each trace allows a complete reconstruction of the schedule until all jobs released prior to $\mathcal{L}$ are complete. Thus, we can compute the total execution time of each job, the service received by each job up to any given time, the remaining processing time of unfinished jobs at any time, and the overall backlog at each job's arrival time.

We assume that the measurement setup defines a valid random experiment. That is, all traces are generated under the same fixed configuration that ensures a representative execution environment, and the variation between traces is due exclusively to the random execution-time model represented by Ω. Consequently, each trace provides one observation of every job in the observed common release prefix up to $\mathcal{L}$. The trace set $\mathcal{D}$ forms the basis for bootstrapping.

Analysis Inputs 1 and 2: We first derive execution-time observations from the trace set. Let $\mathbb{J}_i$ denote the set of jobs of task τ_i released until $\mathcal{L}$. For each job $J_{i,j} \in \mathbb{J}_i$ and trace $s \in \mathcal{D}$, let $c_{i,j}(s)$ denote the realized execution time of $J_{i,j}$ in that trace. For each $J_{i,j} \in \mathbb{J}_i$, we apply the *non-parametric bootstrap* procedure from Sec. II to the set of observations $O = \{o_1, \ldots, o_N\} = \{c_{i,j}(s_1), \ldots, c_{i,j}(s_N)\}$ to obtain confidence intervals for the mean and variance of each job's execution time. We denote the respective upper bounds of these confidence intervals by $\widehat{\mathbb{E}}\,[c_{i,j}]$ and $\widehat{\mathrm{Var}}\,[c_{i,j}]$, respectively. Analysis Inputs 1 and 2 are then derived as follows:

$$\widehat{e}_i \triangleq \max_{J_{i,j} \in \mathbb{J}_i} \widehat{\mathbb{E}}\,[c_{i,j}] \quad \text{and} \quad \widehat{s}_i^c \triangleq \max_{J_{i,j} \in \mathbb{J}_i} \widehat{\mathrm{Var}}\,[c_{i,j}]\,. \quad (21)$$

Analysis Inputs 3 and 4: For every pair of distinct jobs $J_{i,j} \in \mathbb{J}_i$ and $J_{k,\ell} \in \mathbb{J}_k$, we apply the non-parametric bootstrap procedure to the set of paired observations $O = \left\{\left(c_{i,j}(s_1), c_{k,\ell}(s_1)\right), \ldots, \left(c_{i,j}(s_N), c_{k,\ell}(s_N)\right)\right\}$. We denote the upper bound of the resulting confidence interval as $\widehat{\mathrm{Cov}}\,[c_{i,j}, c_{k,\ell}]$. To derive $\widehat{v}_{i,i}$, we take the maximum over all distinct job pairs of the same task, and to derive $\widehat{v}_{i,k}$, we take the maximum over all pairs of jobs from distinct tasks, while ensuring that the parameter estimates are non-negative:

$$\widehat{v}_{i,i} \triangleq \max\left(0,\, \max_{\substack{J_{i,j}, J_{i,\ell} \in \mathbb{J}_i \\ j \ne \ell}} \widehat{\mathrm{Cov}}\,[c_{i,j}, c_{i,\ell}]\right),$$

$$\widehat{v}_{i,k} \triangleq \max\Big(0,\, \max_{\substack{J_{i,j}\in\mathbb{J}_i \\ J_{k,\ell}\in\mathbb{J}_k}} \widehat{\mathrm{Cov}}\left[c_{i,j}, c_{k,\ell}\right]\Big), \quad i \neq k.$$

Analysis Inputs 5 and 6: For a given task $\tau_i \in \tau$, each trace $s \in \mathcal{D}$ provides a complete record of the observed schedule involving all jobs released before time $\mathcal{L}$. This allows us to compute, for each job $J_{i,j} \in \mathbb{J}_i$, the *actual backlog* $B_{i,j}(s)$ faced at the time of its release, by evaluating Eq. (1) and Def. 2 with respect to the observed schedule and execution times. For each job, this yields a set of observations $O = \{B_{i,j}(s_1), \ldots, B_{i,j}(s_N)\}$. We obtain the task parameters $\widehat{b}_i$ and $\widehat{s}_i^b$ analogously to Eq. (21), as the maxima over the respective confidence interval bounds.

Analysis Input 7: For every pair of jobs $J_{i,j} \in \mathbb{J}_i$ and $J_{k,\ell} \in \mathbb{J}_k$, to bound the covariance between the actual backlog encountered by $J_{i,j}$ upon arrival and $J_{k,\ell}$'s execution time, we consider the set of paired observations $O = \big\{\big(B_{i,j}(s_1), c_{k,\ell}(s_1)\big), \ldots, \big(B_{i,j}(s_N), c_{k,\ell}(s_N)\big)\big\}$. For each such set, we again obtain a confidence interval for the covariance via bootstrapping and let $\widehat{\mathrm{Cov}}\left[B_{i,j}, c_{k,\ell}\right]$ denote its upper bound. By taking the maximum over all jobs of tasks τ_i and τ_k released prior to $\mathcal{L}$, we obtain

$$\widehat{v}_{i,k}^b \triangleq \max\Big(0,\, \max_{\substack{J_{i,j}\in\mathbb{J}_i \\ J_{k,\ell}\in\mathbb{J}_k}} \widehat{\mathrm{Cov}}\left[B_{i,j}, c_{k,\ell}\right]\Big).$$

This concludes the proposed statistical parameter-inference procedure. Sec. VIII reflects on its assumptions and limitations.

VII. Empirical Evaluation

We evaluated the proposed analysis using synthetically generated workloads in a simulated setting to ensure precise control over execution-time correlations. For brevity, we refer to Theorem 3 as "CABS" throughout this section.

As the closest existing baseline, we chose Marković et al.'s *correlation-aware analysis* (CAA) [42]. We first compared CABS with CAA in the restricted setting supported by CAA, which assumes FP scheduling *with* job abortion and bounds the single-job deadline-miss probability $\mathrm{DFP}_k(1)$. In a second set of experiments, we evaluated CABS in a more general setting *without* job abortion under FP, EDF, and FIFO scheduling and examined its sensitivity to the number of consecutive deadline misses m, number of tasks n, and choice of relative deadlines.

Workloads. To control the degree of intra- and inter-task correlation, we generated synthetic workloads as follows. Each task τ_i was specified by its period T_i, offset Φ_i, and the number of *work items* c_i processed by each of its jobs. Each such work item represents a small amount of randomly varying execution time. We let $\overline{w}$ denote the expected execution time per work item, and hence $\overline{u}_i = \frac{c_i \cdot \overline{w}}{T_i}$ is τ_i's *expected utilization*. Tasks had implicit deadlines ($D_i = T_i$) unless stated otherwise.

Each job of τ_i processed c_i work items with randomly drawn execution costs (as specified below). Intra- and inter-task correlation was introduced by *sharing* work items among multiple jobs. To this end, we introduced *pair-shared pools* and a *global pool* of work items that were refreshed over time, and configured the jobs to obtain a specified fraction of their items from each category, as discussed further below.

We generated task sets in a two-step process: first, we selected a *period configuration* (*i.e.*, a vector of periods $T_1, \ldots, T_n$), which was then paired with multiple *work-item configurations* (*i.e.*, vectors of work-item counts $c_1, \ldots, c_n$). Each resulting pairing yielded a separate task set.

To obtain one period configuration for a given number of tasks n, we selected n periods $T_1, \ldots, T_n$ uniformly at random from $\{2, 5, 10, 20, 50, 100, 200\}$ ms, which commonly feature in automotive workloads [29]. To ensure a suitable starting point for the subsequent generation of work-item configurations, we checked that the chosen periods satisfy $\sum_{i=1}^{n} \overline{w}/T_i < \theta$, where $\theta \in \{0.05, 0.6, 0.99\}$ is an experiment-specific minimum feasible utilization threshold. If a set of periods failed the check, then it was discarded and redrawn (up to 1000 times, after which smaller periods were simply replaced by larger ones until the condition was satisfied). For each resulting period T_i, we selected a corresponding offset Φ_i uniformly from $[0, T_i)$. Under FP scheduling, we assigned rate-monotonic priorities.

To generate a work-item configuration for a given target total number of work items $c_\Sigma = \sum_{i=1}^{n} c_i \geq n$ and a period configuration $T_1, \ldots, T_n$, we first assigned $1 + \lfloor (c_\Sigma - n) T_i / \sum_j T_j \rfloor$ work items to each task τ_i to ensure a roughly proportional amount of work per task. Any remaining work items were then assigned one by one in order of decreasing remainder $((c_\Sigma - n) T_i / \sum_j T_j) - \lfloor (c_\Sigma - n) T_i / \sum_j T_j \rfloor$.

For each period configuration, we generated work-item configurations as described above by varying $c_\Sigma \in \{n, n+1, n+2, \ldots\}$ until reaching the upper utilization threshold $\sum_i \overline{u}_i \geq 0.99$; this final configuration was discarded.

Simulation. For each generated task set, we simulated $N = 100$ traces up to the observation limit $\mathcal{L} = 0.7\,\mathrm{s}$. Each job's execution cost was the sum of the execution costs of the work items that it processed. Specifically, each job of a task τ_i was configured to generate $\lfloor 0.9 \cdot c_i \rfloor$ *private* (*i.e.*, unshared) work items and to draw, uniformly at random and with replacement, $\lfloor 0.05 \cdot c_i \rfloor$ work items from pair-shared pools and $\lfloor 0.05 \cdot c_i \rfloor$ work items from the global pool, with any remaining work items assigned to the categories in order of decreasing remainder.

The pools were configured as follows. For any two tasks τ_i and τ_j, with $1 \leq i < j \leq n$, there was one pair-shared pool containing 10 work items. Every $\min\{T_i, T_j\}$ time units, the oldest work item was replaced by a newly sampled one. For each task, the pair-shared pools accessed by all its jobs were selected uniformly at random. The global pool also contained 10 work items, with the oldest work item being replaced by a newly sampled one every $\min_i\{T_i\}$ time units.

Unless noted otherwise, whenever a new work item was required—when initially filling a pool, replacing an item, or generating a job's private items—its associated execution cost was sampled uniformly from $[200, 300]\,\mu$s; thus, $\overline{w} = 250\,\mu$s.

For each experiment discussed below, we generated 40 period vectors for $n = 5$ tasks, unless noted otherwise. From the $N = 100$ simulated traces of each considered task set, we inferred Analysis Inputs 1–7 using $B = 1000$ bootstrap resamples at confidence level $\gamma = 0.99$. We averaged the mean execution times of τ_i's jobs to obtain $\bar{C}_i$ and computed the task's mean

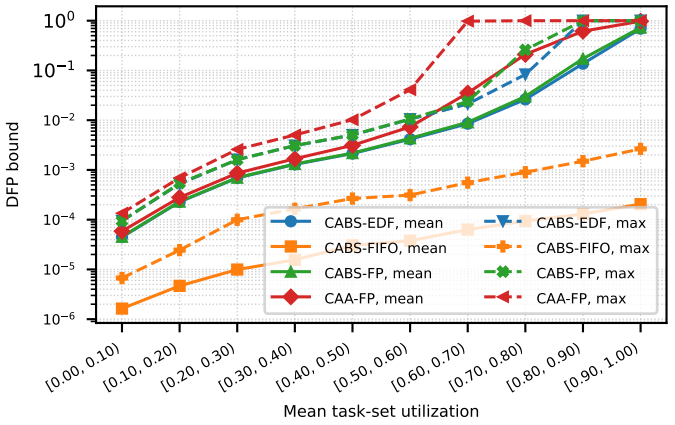


Fig. 1. Mean (solid lines) and maximum (dashed lines) DFP bounds for $m = 1$ and with job abortion. CAA supports only FP scheduling.

utilization as $U = \sum_i \bar{C}_i/T_i$. For visualization purposes, we grouped task sets by U into bins of width 0.1.

A. Baseline Comparison with Job Abortion

In the first experiment, we compared CABS and CAA in the common setting of FP scheduling with job abortion. This is not the setting for which CABS was developed, but our analysis also supports it. Although CABS does not model job abortion explicitly, the effect of aborting tardy jobs is captured by the backlog bounds (Analysis Input 5), which are lower when tardy jobs do not contribute past their deadlines. We also report CABS bounds for EDF and FIFO scheduling. Fig. 1 shows the mean and maximum $\mathrm{DFP}_k(1)$ bounds as a function of U for the task with the lowest priority under FP scheduling.

CABS matches or outperforms CAA in the FP case: where there is a noticeable difference, CABS consistently yields lower $\mathrm{DFP}_k(1)$ bounds than CAA. The reason is that CAA conservatively over-approximates carry-in interference by accounting for one additional job [42]. In contrast, CABS uses measured backlog bounds and therefore incurs less pessimism.

The comparison between scheduling policies shows that the CABS bounds for FP and EDF virtually coincide in this experiment. This is consistent with the policy-specific job-count bounds in Lemma 8. Since the analyzed task is the lowest-priority task under FP, Eq. (14) includes all tasks; with implicit deadlines, Eq. (16) yields nearly the same arrival interval. Thus, the resulting x_i^π values are almost identical for FP and EDF. In contrast, FIFO uses the shorter interval in Eq. (15), leading to lower job counts and hence also to a lower CABS bound.

In summary, CABS performs favorably compared to the prior state-of-the-art correlation-aware analysis for FP scheduling, while also providing analogous bounds for EDF and FIFO scheduling, for which there are no comparable prior analyses.

B. Sensitivity Analysis without Job Abortion

Next, we explored the sensitivity of CABS in the general setting without job abortion (for which no prior correlation-aware baselines exist). To this end, we varied the following parameters while keeping other settings fixed: **(i)** the task count n, **(ii)** the ratio of a task's relative deadline to its period, **(iii)** the task under analysis, **(iv)** the number of consecutive deadline misses m, and **(v)** the range of per-work-item execution costs.

Task count. First, we varied the number of tasks n. Fig. 2(a) shows the results for five mean-utilization bins. Overall, the results clearly indicate that the number of tasks has only a limited effect on the $\mathrm{DFP}_k(1)$ bound in the considered setting. The dominant factor is instead the total mean utilization U, which is consistent with the trends in Fig. 1.

Deadlines. Next, we varied the ratio of the relative deadline of the task under analysis to its period (D_i/T_i). Fig. 2(b) presents the resulting trends for three mean-utilization bins. At all utilization levels, the bound increases as the normalized deadline decreases (*i.e.*, as the relative deadline becomes tighter). This increase is particularly steep for constrained deadlines ($D_i \leq T_i$). Conversely, CABS can yield lower bounds when the deadline exceeds the period, with the effect being most clearly visible at the highest utilization level. This demonstrates the benefit of supporting arbitrary deadlines in the analysis.

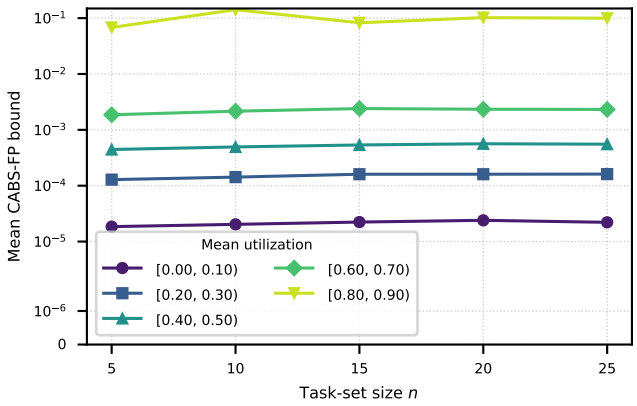
Analyzed task. We additionally explored how tasks with different period lengths (and hence priorities under FP) differ in their $\mathrm{DFP}_k(1)$ bounds. Fig. 3 shows CABS bounds for all three considered scheduling policies and for two different tasks in each task set: the shortest-period (or highest-priority) task (HI) and the longest-period (or lowest-priority) task (LO). For the LO task, FP and EDF behave nearly identically, whereas FIFO is qualitatively different. For the HI task, FP yields lower mean bounds than EDF, but follows a broadly similar trend, whereas FIFO again exhibits a markedly different trend.

Under FP and EDF, the bound is lower for the HI task than for the LO task. Under FP, the highest-priority task suffers no higher-priority interference. Hence, the response time of a job of the HI task is determined only by its own execution time and any tardy jobs of the same task. Under EDF, since we assign implicit deadlines in this experiment, the task with the shortest period is similarly shielded from most interference, but to a lesser degree because the two policies behave differently under transient overload [6]. Conversely, the LO task is subject to interference from all tasks under both FP and EDF.

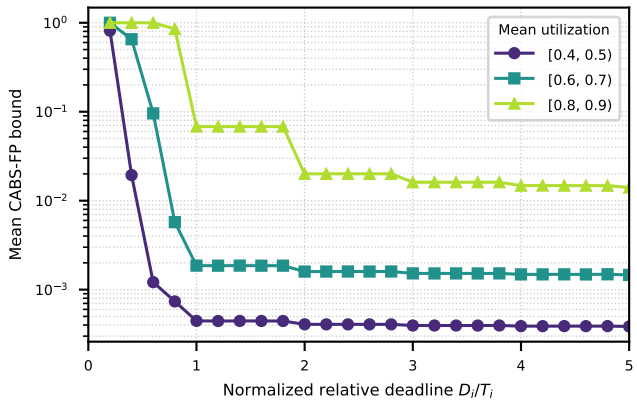
FIFO shows the opposite behavior. Since FIFO is oblivious to task parameters, jobs of the HI task are not given preferential treatment. They can thus be delayed by earlier jobs of all tasks. This disproportionately impacts the HI task because of its short period (and hence deadline). In fact, the CABS bound is clipped at 1, indicating that CABS cannot provide a nontrivial DFP bound for this task. For the LO task, FIFO yields a lower bound than FP and EDF for the same reason as in Fig. 1.

Consecutive misses. Finally, we varied the number of consecutive deadline misses m, as shown in Fig. 4 for $n = 2$ tasks. In this setting, the largest reduction in the mean and maximum $\mathrm{DFP}_k(m)$ bounds occurs when m increases from 1 to 2. The step from $m = 2$ to $m = 3$ is still clearly visible, but the jump to $m = 5$ offers only diminishing returns.

Upon further investigation, we observed that the CABS $\mathrm{DFP}_k(m)$ bound for $m > 1$ is highly sensitive to the number of tasks n. This is clearly visible in Fig. 5, which shows the ratio



(a) Impact of the task-set size n on $\mathrm{DFP}_k(1)$ for five utilization bins.



(b) Normalized deadline of the analyzed task for three utilization intervals.

Fig. 2. Sensitivity of the mean CABS bound for FP scheduling without job abortion.

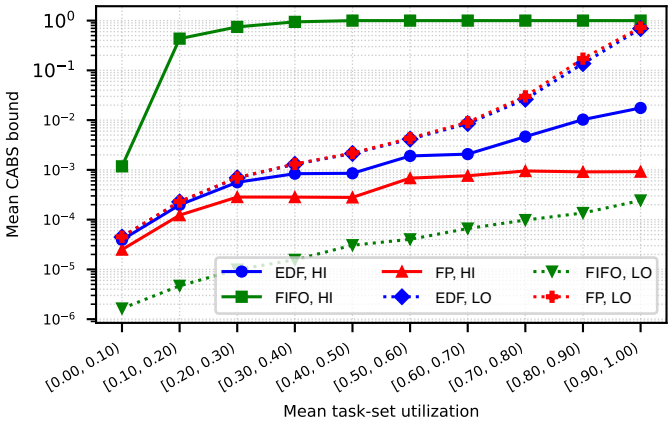


Fig. 3. Mean $\mathrm{DFP}_k(1)$ bounds for the tasks with the shortest and longest periods under FP, EDF, and FIFO scheduling without job abortion.

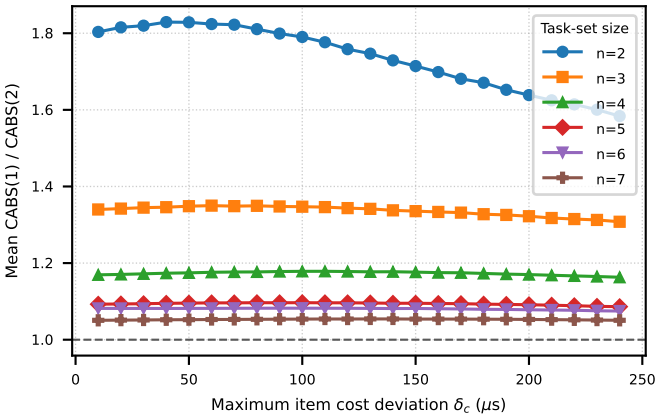


Fig. 5. Mean improvement factor $\mathrm{DFP}_k(1)/\mathrm{DFP}_k(2)$ for $2 \leq n \leq 7$ as a function of the maximum work-item cost deviation δ_c.

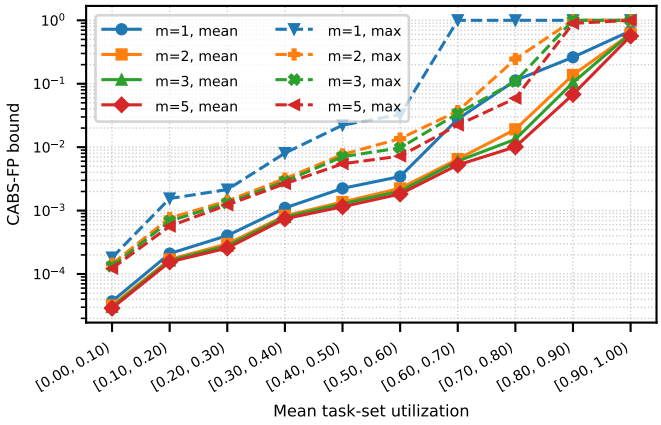


Fig. 4. Mean (solid lines) and maximum (dashed lines) $\mathrm{DFP}_k(m)$ bounds for $m \in \{1, 2, 3, 5\}$ and $n = 2$ under FP scheduling without job abortion.

$\mathrm{DFP}_k(1)/\mathrm{DFP}_k(2)$ for $2 \leq n \leq 7$ as the range of possible work-item costs changes. Specifically, as a proxy for execution-time variance, the cost of each work item was drawn uniformly from $[250 - \delta_c, 250 + \delta_c]$ μs as δ_c was varied from 0 to 250 μs. Fig. 5 demonstrates that CABS is sensitive to such changes in execution-cost variance, and that $\mathrm{DFP}_k(2)$ approaches $\mathrm{DFP}_k(1)$ as n increases (*i.e.*, the benefit of tolerating deadline misses diminishes for larger task sets). We believe this effect to be an artifact of the interplay of Eq. (18) and Fact 1. This observation calls for further research into alternative ways of deriving correlation-aware $\mathrm{DFP}_k(m)$ bounds.

In summary, where prior correlation-aware analysis applies, CABS yields $\mathrm{DFP}_k(1)$ bounds that match or improve upon those yielded by CAA. In the more general setting, CABS extends coverage to workloads previously unsupported by the state of the art and shows that relaxing timing requirements (by increasing D_i or m) can lead to more favorable bounds.

VIII. Threats to Validity and Limitations

Our work rests on a number of premises that, if violated, may undermine the presented results and conclusions. Most importantly, the sampling-based parameter-inference procedure proposed in Sec. VI cannot by itself guarantee that the estimated analysis inputs are valid, for the following reasons.

Theorem 3 requires Analysis Inputs 1–7 to be finite upper bounds. Even if all jobs released before the observation limit $\mathcal{L}$ complete in every trace, a finite sample cannot rule out an underlying distribution with infinite variance (*e.g.*, rare but arbitrarily large execution costs may remain unobserved). Sustained overload may also cause the mean or variance of the backlog to grow without bound over successive job arrivals.

Observations of jobs released before $\mathcal{L}$ cannot establish that the inferred bounds remain valid for later-arriving jobs without additional assumptions about system behavior. Additionally, the measured traces must be representative of the stochastic process encountered after deployment. This requires covering all relevant operating conditions (also known as *scenarios of operation* [5, 13, 14]) that may arise at runtime. If there are distinct scenarios of operation (*e.g.*, due to differences in environment or configuration), they must be analyzed separately. Determining whether all relevant operating conditions have been considered is difficult in general and beyond our scope.

Additionally, Sec. VI treats the collected traces as independent observations. Resetting the system between repetitions (*e.g.*, by rebooting) can help, but may not suffice because of thermal or other persistent environmental effects. Instrumentation can also perturb execution and scheduling through probe effects. Unless these probe effects are conservatively accounted for or shown to be negligible, the resulting CABS bound applies only to the instrumented configuration.

Finally, Sec. VII considers only simulations with synthetic workloads that do not represent any particular real-world workload. The presented results therefore merely suggest the potential of CABS and require further validation in practice.

IX. Related Work

Stochastic real-time systems have attracted considerable interest in recent years. For a broad overview, see the comprehensive surveys by Davis and Cucu-Grosjean [13, 14]. We focus here on the most closely related work to CABS.

Statistical approaches. Extreme Value Theory (EVT) has been applied to analyze measured execution times [8, 12, 31, 32] and observed response times [35–37], but, for dependent tasks, requires the observation sequence to be extremally independent [45] or stationary [30]. Bozhko et al. [4] applied Monte Carlo sampling to estimate response-time distributions.

Dependencies. Hidden Markov Models (HMMs) are a natural way to capture execution-time dependencies induced by a task's internal state. Existing HMM-based analyses [1, 18–21], however, address different problems, typically without inter-task dependencies and with long-run deadline-miss probabilities rather than the deadline-failure probability studied here. The problem of unknown dependencies was studied by Ivers and Ernst [27] for FP scheduling under the assumption that probability distributions are known and identical for each job. Further approaches addressing restricted notions of dependencies study average-case tardiness when dependence is limited to bounded time intervals [44] or propose independence thresholds [33] to split the execution cost into dependent and independent parts. The analysis by von der Brüggen et al. [50] accounts for dependencies among a bounded number of consecutive jobs when bounding the DFP under EDF.

Backlog. Bounding the backlog is essential when tardy jobs are not aborted. Early stochastic analyses assuming independent per-job execution times [15, 16, 34] use Markov chains to model the accumulation of backlog and rely on the calculation of a stationary backlog distribution. As this can become computationally expensive for large hyperperiods, alternative methods for calculating the stationary backlog have been proposed by Kim et al. [28]. More recently, Zagalo et al. [54] analyzed transient and steady-state response-time distributions under FP scheduling, assuming mutually independent job execution times [53, 54]. Prior work modeling dependent execution costs with HMMs focuses on a single task executing in an isolated reservation [18, 21, 22]. While Ivers and Ernst [27] have extended the results by Díaz et al. [15, 16] to allow for dependencies, the resulting analyses still require the execution-time distribution to be identical for each job.

DFP. Prior single-miss DFP bounds mostly rely on strong independence assumptions, often by requiring suitable pWCET distributions [5] as inputs. Maxim and Cucu-Grosjean [43] proposed a method for preemptive FP scheduling that bounds the DFP by convolving discrete pWCET distributions of individual jobs. Marković et al. [39] improved the approach with optimal resampling and more efficient circular convolution, and von der Brüggen et al. [49] proposed task-level convolution as a more scalable alternative. Additionally, approaches using analytical bounds such as the Chernoff bound [9, 10], the Hoeffding and Bernstein inequalities [49], and the Berry-Esseen theorem [40] avoid convolution altogether. Notably, Chen et al. [11] identified a flaw in DFP bounds built on the critical-instant pattern [9, 43] and proposed alternative solutions. Most recently, Sun et al. [47] generalized the Chernoff bound to systems with stochastic arrivals and Guan et al. [26] presented an improved analysis for EDF-scheduled systems.

X. Conclusion

We have explored a new approach to analyzing stochastic real-time systems based on directly sampling the backlog when jobs arrive. Our method, CABS, provides the first correlation-aware bound on the probability of deadline misses in periodic uniprocessor real-time systems that simultaneously 1) does not require per-job execution times to be independent or identically distributed, 2) allows tardy jobs to continue executing past their deadlines, 3) supports arbitrary deadlines, 4) applies under FP, EDF, and FIFO scheduling, and 5) generalizes to multiple consecutive deadline misses. CABS also applies to each core individually under partitioned multicore scheduling.

In a simulation-based evaluation with synthetic workloads, CABS matched or outperformed CAA [42] in the restricted setting supported by CAA. In more general settings, our results show that relaxing timing requirements by either extending relative deadlines beyond periods or permitting a small number of consecutive deadline misses can lead to more favorable bounds, which also depend on the scheduling policy. These factors warrant further attention.

In future work, an evaluation of CABS with representative workloads on real hardware is needed to validate our findings from simulation. Additionally, it would be interesting to develop and evaluate rigorous measurement protocols that incorporate statistical hypothesis testing, and to extend multicore support beyond partitioned scheduling to migratory scheduling policies.

Acknowledgments

This work has been funded in part by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) – Project No. 569077889 (PEACH).

References

[1] L. Abeni, D. Fontanelli, L. Palopoli, and B. V. Frias, "A Markovian model for the computation time of real-time applications," in *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2017, pp. 1–6.

[2] B. Akesson, M. Nasri, G. Nelissen, S. Altmeyer, and R. I. Davis, "A comprehensive survey of industry practice in real-time systems," *Real-Time Systems*, vol. 58, no. 3, pp. 358–398, 2022.

[3] G. Bernat, A. Burns, and A. Llamosí, "Weakly hard real-time systems," *IEEE Transactions on Computers*, vol. 50, no. 4, pp. 308–321, 2001.

[4] S. Bozhko, G. von der Brüggen, and B. B. Brandenburg, "Monte Carlo response-time analysis," in *Proceedings of the 42nd IEEE Real-Time Systems Symposium (RTSS)*, 2021, pp. 342–355.

[5] S. Bozhko, F. Marković, G. von der Brüggen, and B. B. Brandenburg, "What really is pWCET? A rigorous axiomatic proposal," in *Proceedings of the 44th IEEE Real-Time Systems Symposium (RTSS)*, 2023, pp. 13–26.

[6] G. C. Buttazzo, "Rate monotonic vs. EDF: judgment day," *Real-Time Systems*, vol. 29, no. 1, pp. 5–26, 2005.

[7] F. P. Cantelli, "Sui confini della probabilita," in *Atti del Congresso Internazionale dei Matematici: Bologna del 3 al 10 de settembre di 1928*, 1929, pp. 47–60.

[8] 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 22nd International Conference on Real-Time Networks and Systems (RTNS)*, 2014, p. 299.

[9] K. Chen and J. Chen, "Probabilistic schedulability tests for uniprocessor fixed-priority scheduling under soft errors," in *Proceedings of the 12th IEEE International Symposium on Industrial Embedded Systems (SIES)*, 2017, pp. 1–8.

[10] K. Chen, N. Ueter, G. von der Brüggen, and J. Chen, "Efficient computation of deadline-miss probability and potential pitfalls," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, pp. 896–901.

[11] K. Chen, M. Günzel, G. von der Brüggen, and J. Chen, "Critical instant for probabilistic timing guarantees: Refuted and revisited," in *Proceedings of the 43rd IEEE Real-Time Systems Symposium (RTSS)*, 2022, pp. 145–157.

[12] 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 24th Euromicro Conference on Real-Time Systems (ECRTS)*, 2012, pp. 91–101.

[13] R. I. Davis and L. Cucu-Grosjean, "A survey of probabilistic timing analysis techniques for real-time systems," *Leibniz Transactions on Embedded Systems*, vol. 6, no. 1, pp. 03:1–03:60, 2019.

[14] ——, "A survey of probabilistic schedulability analysis techniques for real-time systems," *Leibniz Transactions on Embedded Systems*, vol. 6, no. 1, pp. 04:1–04:53, 2019.

[15] J. L. Díaz, D. F. García, K. Kim, C. Lee, L. L. Bello, J. M. López, S. L. Min, and O. Mirabella, "Stochastic analysis of periodic real-time systems," in *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS)*, 2002, pp. 289–300.

[16] J. L. Díaz, J. M. López, M. G. Vazquez, A. M. Campos, K. Kim, and L. L. Bello, "Pessimism in the stochastic analysis of real-time systems: Concept and applications," in *Proceedings of the 25th IEEE Real-Time Systems Symposium (RTSS)*, 2004, pp. 197–207.

[17] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. Springer, 1993.

[18] B. V. 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 23rd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2017, pp. 175–186.

[19] A. Friebe, A. V. Papadopoulos, and T. Nolte, "Identification and validation of Markov models with continuous emission distributions for execution times," in *Proceedings of the 26th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2020, pp. 1–10.

[20] A. Friebe, F. Marković, A. V. Papadopoulos, and T. Nolte, "Adaptive runtime estimate of task execution times using Bayesian modeling," in *Proceedings of the 27th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2021, pp. 1–10.

[21] ——, "Continuous-emission Markov models for real-time applications: Bounding deadline miss probabilities," in *Proceedings of the 29th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2023, pp. 14–26.

[22] ——, "Efficiently bounding deadline miss probabilities of Markov chain real-time tasks," *Real-Time Systems*, vol. 60, no. 3, pp. 443–490, 2024.

[23] M. K. Gardner and J. W.-S. Liu, "Analyzing stochastic fixed-priority real-time systems," in *Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, 1999, pp. 44–58.

[24] W. Geelen, D. Antunes, J. P. Voeten, R. R. Schiffelers, and W. Heemels, "The impact of deadline misses on the control performance of high-end motion control systems," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1218–1229, 2015.

[25] G. Grimmett and D. J. Welsh, *Probability: an introduction*. Oxford University Press, 2014.

[26] F. Guan, X. Jiang, W. Jing, and N. Guan, "Reducing worst-case deadline failure probability for EDF scheduling," in *Proceedings of the 46th IEEE Real-Time Systems Symposium (RTSS)*, 2025, pp. 16–28.

[27] M. Ivers and R. Ernst, "Probabilistic network loads with dependencies and the effect on queue sojourn times," in *Proceedings of the 6th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE)*, 2009, pp. 280–296.

[28] K. Kim, J. L. Díaz, L. L. Bello, J. M. López, C. Lee, and S. L. Min, "An exact stochastic analysis of priority-driven periodic real-time systems and its approximations," *IEEE Transactions on Computers*, vol. 54, no. 11, pp. 1460–1466, 2005.

[29] S. Kramer, D. Ziegenbein, and A. Hamann, "Real world automotive benchmarks for free," in *Proceedings of the 6th International Workshop on Analysis Tools and Methodologies for Embedded and Real-Time Systems (WATERS)*, 2015.

[30] M. Leadbetter, G. Lindgren, and H. Rootzén, "Conditions for the convergence in distribution of maxima of stationary normal processes," *Stochastic Processes and their Applications*, vol. 8, no. 2, pp. 131–139, 1978.

[31] G. Lima and I. Bate, "Valid application of EVT in timing analysis by randomising execution time measurements," in *Proceedings of the 23rd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2017, pp. 187–198.

[32] G. Lima, D. Dias, and E. Barros, "Extreme value theory for estimating task execution time bounds: A careful look," in *Proceedings of the 28th Euromicro Conference on Real-Time Systems (ECRTS)*, 2016, pp. 200–211.

[33] 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 35th IEEE Real-Time Systems Symposium (RTSS)*, 2014, pp. 314–323.

[34] J. M. López, J. L. Díaz, J. Entrialgo, and D. F. García, "Stochastic analysis of real-time systems under preemptive priority-driven scheduling," *Real-Time Systems*, vol. 40, no. 2, pp. 180–207, 2008.

[35] Y. Lu, T. Nolte, J. Kraft, and C. Norström, "Statistical-based response-time analysis of systems with execution dependencies between tasks," in *Proceedings of the 15th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, 2010, pp. 169–179.

[36] ——, "A statistical approach to response-time analysis of complex embedded real-time systems," in *Proceedings of the 16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2010, pp. 153–160.

[37] Y. Lu, T. Nolte, I. Bate, and L. Cucu-Grosjean, "A statistical response-time analysis of real-time embedded systems," in *Proceedings of the 33rd IEEE Real-Time Systems Symposium (RTSS)*, 2012, pp. 351–362.

[38] M. Maggio, A. Hamann, E. Mayer-John, and D. Ziegenbein, "Control-system stability under consecutive deadline misses constraints," in *Proceedings of the 32nd Euromicro Conference on Real-Time Systems (ECRTS)*, 2020, pp. 21:1–21:24.

[39] F. Marković, A. V. Papadopoulos, and T. Nolte, "On the convolution efficiency for probabilistic analysis of real-time systems," in *Proceedings of the 33rd Euromicro Conference on Real-Time Systems (ECRTS)*, 2021, pp. 16:1–16:22.

[40] F. Marković, T. Nolte, and A. V. Papadopoulos, "Analytical approximations in probabilistic analysis of real-time systems," in *Proceedings of the 43rd IEEE Real-Time Systems Symposium (RTSS)*, 2022, pp. 158–171.

[41] F. Marković, P. Roux, S. Bozhko, A. V. Papadopoulos, and B. B. Brandenburg, "CTA: A correlation-tolerant analysis of the deadline-failure probability of dependent tasks," in *Proceedings of the 44th IEEE Real-Time Systems Symposium (RTSS)*, 2023, pp. 317–330.

[42] F. Marković, G. von der Brüggen, M. Günzel, J. Chen, and B. B. Brandenburg, "A distribution-agnostic and correlation-aware analysis of periodic tasks," in *Proceedings of the 45th IEEE Real-Time Systems Symposium (RTSS)*, 2024, pp. 215–228.

[43] D. Maxim and L. Cucu-Grosjean, "Response time analysis for fixed-priority tasks with multiple probabilistic parameters," in *Proceedings of the 34th IEEE Real-Time Systems Symposium (RTSS)*, 2013, pp. 224–235.

[44] 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 17th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2011, pp. 207–217.

[45] L. Santinelli, J. Morio, G. Dufour, and D. Jacquemart, "On the sustainability of the extreme value theory for WCET estimation," in *Proceedings of the 14th International Workshop on Worst-Case Execution Time Analysis (WCET)*, 2014, pp. 21–30.

[46] M. Seidel, M. Maggio, and F. Allgöwer, "A controller synthesis framework for weakly-hard control systems," in *Proceedings of the 32nd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2026, pp. 41–54.

[47] S. Sun, C. Yu, X. Jiang, Q. Deng, and N. Guan, "WCDFP analysis for real-time tasks with stochastic release patterns using Chernoff bound," in *Proceedings of the 46th IEEE Real-Time Systems Symposium (RTSS)*, 2025, pp. 553–565.

[48] T.-S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.-C. Wu, and J.-S. Liu, "Probabilistic performance guarantee for real-time tasks with varying computation times," in *Proceedings of the 1st IEEE Real-Time Technology and Applications Symposium (RTAS)*, 1995, pp. 164–173.

[49] G. von der Brüggen, N. Piatkowski, K. Chen, J. Chen, and K. Morik, "Efficiently approximating the probability of deadline misses in real-time systems," in *Proceedings of the 30th Euromicro Conference on Real-Time Systems (ECRTS)*, 2018, pp. 6:1–6:22.

[50] G. von der Brüggen, N. Piatkowski, K. Chen, J. Chen, K. Morik, and B. B. Brandenburg, "Efficiently approximating the worst-case deadline failure probability under EDF," in *Proceedings of the 42nd IEEE Real-Time Systems Symposium (RTSS)*, 2021, pp. 214–226.

[51] N. Vreman and M. Maggio, "Stochastic analysis of control systems subject to communication and computation faults," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 5s, pp. 1–25, 2023.

[52] 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 9th IEEE Real-Time Systems Symposium (RTSS)*, 1988, pp. 222–231.

[53] K. Zagalo, "Stochastic analysis of stationary real-time systems," Ph.D. dissertation, Sorbonne University, 2023.

[54] K. Zagalo, Y. Abdeddaïm, A. Bar-Hen, and L. Cucu-Grosjean, "Response time stochastic analysis for fixed-priority stable real-time systems," *IEEE Transactions on Computers*, vol. 72, no. 1, pp. 3–14, 2023.