Toward Probabilistic Real-Time Calculus

Luca Santinelli
Scuola Superiore Sant'anna, Pisa, luca.santinelli@sssup.it
INRIA Nancy-Grand Est, luca.santinelli@inria.fr

Liliana Cucu-Grosjean
INRIA Nancy-Grand Est, liliana.cucu@inria.fr

***Abstract*—A challenging research issue in analyzing probabilistic real-time systems is to model the tasks composing the system and the resource provided to the system. In this paper we propose a solution based on a probabilistic component-based model that abstracts the functional and non-functional requirements of real-time components. The obtained interfaces encode timing requirements and probability information of the component in a probabilistic version of the real-time calculus.**

Besides, it has been derived probabilistic scheduling and compositional guarantees to provide real-time analyses of probabilistic real-time systems. Finally, a test case illustrates the potentialities of the proposed model and its applicability in a large variety of problems within the probabilistic real-time scenario.

I. Introduction

The performances of the real-time systems depend on the correctness of those systems and mostly from the perspective of time. Nowadays these systems have become more complex and they are composed by different elements exploiting functional aspects of the system.

Recent trends depict real-time systems as component-based systems where the applications or dedicated HW/SW components provide information about the processing semantics that are used to execute the various applications [1]–[3]. Component-based design provides a means for decomposing a real-time system into components reducing a single complex problem into multiple simpler design problems. A real-time system is then composed through component interfaces that abstract the internal complexity of the components and encode the timing requirements of real-time components. In particular, hierarchical scheduling frameworks [4]–[6] provide a way to compose large and complex real-time systems from independent sub-systems.

Furthermore, abstraction frameworks are applied with the purpose of analyzing complex real-time systems and their timing requirements, [7]–[9]. Those frameworks work with different methods. Among those techniques we are interested in the real-time calculus (RTC) [10], which is derived from network calculus [11]. The real-time calculus is a worst-case analysis framework for real-time systems based on deterministic bounds by which model the system behavior. The RTC allows event occurrences to be related to the passage of quantitative deterministic time: non-deterministic decisions can be taken throughout bounding curves.

The timing analysis of real-time systems has been extensively studied by considering worst-case values. Such analyses be they deterministic or non-deterministic provide overly pessimistic results, and not all real-time systems could afford this pessimism. For these cases other approaches could be used and for this reason in this paper we deal with probabilistic approaches. A probabilistic approach, allows probabilistic choices to be defined, rather than the simple non-deterministic choices of standard processes.

Papers related to our work had equally used the words *stochastic analysis* [12], [13], *probabilistic analysis* [14], *statistical analysis* [15] and *real-time queuing theory* [16]. Since the paper of Diaz et al. [17], the term *stochastic analysis* of real-time systems has been used regularly by the community regardless of the approach (probabilistic or statistical). In order to avoid confusion, in this paper we will not use the word *stochastic*, that is often associated with unpredicted behavior, but the word *probabilistic* in order to indicate that the work is based on the theory of probability. Moreover by *probabilistic real-time system* we mean a real-time system with at least one parameter defined by a random variable. For these systems, there is the need to extend the analysis abstractions and the classical analysis methods in terms of probabilistic bounds, i.e., functions which expresses the service given to a task flow can be modeled in terms of a probabilistic bound. The stochastic network calculus [18], [19] is a new methodology for performance evaluation of networked systems (backlog and delay analysis) that can account for probabilistic description of the arrivals and probabilistic service guarantees. Although, the stochastic network calculus does not provide information for real-time analyses.

To the best of our knowledge there is no extension of the real-time calculus toward a probabilistic version. Within this paper we propose a probabilistic framework that extends the RTC applicability to probabilistic scenarios.

Contribution of the paper. The aim of this paper is to propose a schedulability analysis of probabilistic real-time systems through the development of a model for probabilistic real-time calculus. First, we propose probabilistic bounds of the resource provisioning and resource demand to cope with generic probabilistic task models. This allows to define a probabilistic interface for the general real-time component. Second, we translate the schedulability and composability conditions for (deterministic) real-time systems into schedulability and composability conditions for probabilistic real-time systems.

Organization of the paper. We present the problem in Section II together with the notations associated to the RTC algebra. Section III proposes a component-based view of real-time systems with the definition of composability for components. Section IV extends the real-time calculus notation to the probabilistic case. Constrained probabilistic curves are defined to form the probabilistic interface of real-time components. In Section V we consider the probabilistic real-time calculus with discrete random variables. Section VI shows a representative example for applying the probabilistic real-time interfaces and doing real-time analysis. Finally, the conclusions and the future works are outlined

II. Problem Statement and Real-Time System Modeling

In a compositional real-time scheduling framework, the main problem is to define a scheduling interface in order to specify the collective real-time requirements of a component. In this work we define an interface for probabilistic real-time systems. A task is characterized by an offset O_i, a relative deadline D_i and a probability of meeting the deadline p_i. We denote by $\mathcal{C}_i$[1] the random variable providing the possible values for worst-case execution time of task τ_i, $\mathcal{C}_i = \begin{pmatrix} C_{i,k} \\ P(C = C_{i,k}) \end{pmatrix}_{k\in\{1,\cdots,k_{C_i}\}}$, where $C_{i,k} \in [C_i^{min}, C_i^{max}]$ and $k_{C_i} \in \mathbb{N}^*$ is the number of values that the random variable $\mathcal{C}_i$ has. Moreover, we denote by $\mathcal{T}_i$ the random variable providing the possible values for the task inter-arrival time, $\mathcal{T}_i = \begin{pmatrix} T_{i,k} \\ P(T = T_{i,k}) \end{pmatrix}_{k\in\{1,\cdots,k_{T_i}\}}$, where $T_{i,k} \in [T_i^{min}, T_i^{max}]$ and $k_{T_i} \in \mathbb{N}^*$ is the number of values that the random variable $\mathcal{T}_i$ has. We denote a task τ_i by $(O_i, \mathcal{C}_i, \mathcal{T}_i, D_i, p_i)$, and it is assumed that all random variables are independent.

In the rest of this section we introduced the basic modeling techniques that will be used for the analysis we are proposing. In the remaining of the paper we extend those modeling in order to cope with probabilistic real-time systems.

A. Real-Time Calculus

We base our proposal on the framework for Modular Performance Analysis with real-time calculus [10], which is a compositional framework for system-level performance analysis of distributed real-time systems. It analyzes the flow of task streams through a network of processing and communication resources in order to compute worst-case backlogs, end-to-end delays, etc.

A General Event Stream Model: Considering a task, its activations can be described as a flow of events a. A flow is said to have a (deterministic) arrival curve $\alpha(t)$ if

$$a(t_2) - a(t_1) \leq \alpha(t_2 - t_1), \forall\ 0 \leq t_1 \leq t_2.$$

The curve $\alpha(t)$ provides an upper bound to the number of events that arrive in *any* time interval of length t. In RTC arrival curves are tuple described by upper and lower bounds to the number of events. In this paper we make use only of the upper bounds for the event stream as the arrival curve. Arrival curves substantially generalize traditional task arrival models such as periodic, periodic with jitter, and sporadic. Event-based arrival curves can be converted into workload-based arrival curves by scaling with the best-case/worst-case execution demand of the events. In this paper, we make use of the workload-based interpretation and assume that each event has a fixed execution demand. More general concepts for characterization of these units are discussed in [20].

A General Resource Model: Processing and communication resources are also represented abstractly. Considering system elements providing resources, and $b(t_i)$ that denotes the amount of workload units the resource makes available up to a time instant t_i. A system element providing the resource is said to provide the resource with a (deterministic) service curve $\beta(t)$, if

$$b(t_2) - b(t_1) \geq \beta(t_2 - t_1), \forall\ 0 \leq t_1 \leq t_2.$$

The resource availability is described by $\beta(t)$ which provides a lower bound on the available service in *any* time interval of length t. Although in RTC the service curve is described by upper and lower bounding curves, in this papers we refer to service curve as the one lower bounding the resource provisioning. The service is expressed in a suitable workload unit that matches the one of the arrival curve, i.e., the number of cycles for computing resources or bits for communication resources. The service curve abstraction allows to model the possible resource supply functions in a generic form of the service.

A system element receives inputs (α, β) and provides outputs as the task execution α' and the residual service β'. For details about how to derive the outputs from the inputs please refer to [7], [10].

The RTC makes use of the convolution and deconvolution operators of the min-plus and max-plus algebra to compute the RTC curves. From the min-plus algebra we have $\oslash$: $\beta \oslash \alpha(t) = sup_{\lambda\geq 0}\{\beta(t+\lambda) - \alpha(\lambda)\}$ and $\otimes$: $\beta \otimes \alpha(t) = inf_{0\leq\lambda\leq t}\{\beta(t) + \alpha(t-\lambda)\}$. From the max-plus algebra we have $\overline{\oslash}$: $\beta\overline{\oslash}\alpha(t) = inf_{\lambda\geq 0}\{\beta(t+\lambda) - \alpha(\lambda)\}$ and $\overline{\otimes}$: $\beta\overline{\otimes}\alpha(t) = sup_{0\leq\lambda\leq t}\{\beta(t) + \alpha(t-\lambda)\}$.

In the RTC the real-time element behavior is characterized in a deterministic manner by the tuple $(\alpha, \beta, \alpha', \beta')$, where α, β are the input ports, and α', β' the output ports composing the interface of a real-time component. Figure 1 shows examples of real-time representations for real-time system elements.

B. Real-Time Schedulability

The RTC applies the bounds in order to guarantee the timing requirements of the tasks. α that upper bounds the

[1] In this paper we utilise calligraphic letters to denote random variables

event stream and β lower bound the service provisioning are sufficient to guarantee schedulability of the system.

To verify the schedulability of a system under a fixed priority (FP) policy, the workload curve is used [21]. A task set Γ is schedulable with a resource provisioning $\beta(t)$ if

$$\forall\ i \quad \exists t_0 \text{ such that } \omega_i(t_0) \leq \beta(t_0), \tag{1}$$

In this case t_0 is searched among a reduced set $schedP_i$ of intervals and the schedulability can be checked as in [22]. The level-i workload ω_i is defined as the i-th task arrival curve and the interference from its higher priority task. $\omega_i(t) = \alpha_i(t) + \sum_{\tau_j \ \in \ hp(\tau_i)} \alpha_j(t)$ is then the cumulative workload of the higher priority tasks than τ_i, $hp(\tau_i)$. For space reasons in this paper we consider the case of fixed priority scheduling, but the same reasoning can be applied to dynamic priority scheduling paradigms.

III. Component-Based Real-Time Systems

A component-based view of real-time systems is defined such that each system element can be modeled as a component [4], [8]. The component interface describes how the component relates to the other components and the environment in terms of inputs/outputs. In particular, real-time interfaces codes the timing requirements of the component [23], [24]. Figure 1 shows a generic real-time interfaces and its assume-guarantee version.

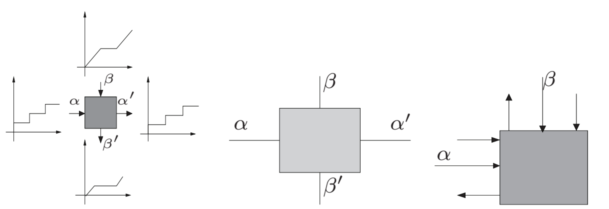


Figure 1. A real-time component: its representation, the RTC interface and the assume-guarantee version.

Henzinger et al. [25] propose assume-guarantee interfaces which are particular instances of real-time interfaces and consider a) the requirements of a component in terms of resource or expected arrivals in order to work properly, and b) the resource or arrivals a component provides. At this stage of our analysis we are interested in the resource composability, but also in the arrival curve in order to have the full composability among components.

By comparing the resource request and the resource availability, it is possible to conclude on the composability of components, which is equivalent to the classical schedulability criteria: *the resource provided to a component by another component has to be enough to satisfy the timing requirements of the component itself.*

1) Composability: According to the assume-guarantee abstraction, in a real-time component-based system there is a component requesting for the computational resource and another component providing such resource [26], [27].

For example, in case of a scheduling component i which schedules an application Γ_i, it *assumes* a minimum amount of resource in order to work properly. A resource provisioning component j *guarantees* a minimum amount of resource. The assume-guarantee interface defines the bounds for the computational resource, while β is the exact resource amount that i receives and that j provides.

The composability criteria [27] compares the assumed and the guaranteed bounds saying that two components are composable if the assumed resource by the first component is less than or equal to the resource guaranteed by the second component. If that is guaranteed, β would be enough to schedule Γ, and the composability is guaranteed.

The composability of real-time components is affected by the scheduling policy which defines the resource distribution among the components. In case of fixed priority scheduling the priority describes the composition order among the tasks. Figure 2 depicts a FP scheduling for n tasks each of them modeled as a component and the assume-guarantee interface, where the β_is are the resources passed among the components.

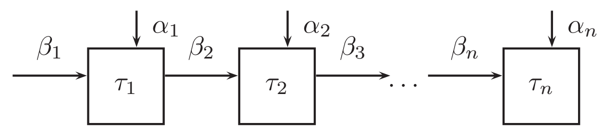


Figure 2. An example for fixed-priority scheduling. The computational resource is passed according the priority assignment: form the highest priority task to the lowest priority taks.

The composability criteria in case of FP scheduling policies applied can be derived in a compositional manner. In [28] it has been derived the resource demand of a FP scheduling component assuming, without loss of generality, the task set ordered by decreasing priority. Therefore, to guarantee the satisfaction of timing constraint for task τ_n the service provided to task τ_n must be at least $\beta_n^A(t) = \alpha_n(t - D_n)$ which is the assumed resource amount the n-th task expect to guarantee its deadlines. In FP tasks are ordered according their priority, meaning that the service for task τ_n is provided by task τ_{n-1}. This also implies that the remaining service curve after have served $\tau_1, \tau_2, \ldots, \tau_{n-1}$ must be at least $\beta_n^A(t)$. The resource requirement of the whole FP scheduling component is the service bound of the most priority task composing the component $\beta_1^A(t)$. To derive that, it has to be sequentially computed the service bounds $\beta_n^A(t), \beta_{n-1}^A(t), \ldots, \beta_2^A(t)$ that each element expects to work properly. The resource requirement of a generic task k is given as

$$\beta_{k-1}^A(t) = \max\{\beta_{k-1}^{\sharp}(t), \alpha_{k-1}^u(\Delta - D_{k-1})\}^2. \tag{2}$$

with $\beta_{k-1}^{\sharp}(t) = \beta_k^A(t-\lambda) + \alpha_{k-1}(t-\lambda)$ where $\lambda = sup\{\psi : \beta_k^A(t-\psi) = \beta_k\}$, that guarantees the remaining service to

[2] It differs from the results of the Theorem 3.4 of [29] because of the real-time constraints that have to be guaranteed in the RTC

be passed to the k-th component to be no less than what the k-th component requires, $\beta_k^A(t)$. Furthermore, to guarantee the timing constraint of τ_{k-1}, the service provided to the $k-1$-th element $\beta_{k-1}^A(t)$ must be no less than $\alpha_{k-1}(t - D_{k-1})$. Applying Equation (2) for $k = n-1, n-2, \dots, 2$ we guarantee the tasks timing constraints by computing the resource required. $\beta_1^A(t)$ is the resource amount that has to be provided in order to satisfy the timing constraints of all the tasks scheduled: the assumed resource requirements.

Theorem III.1 (FP Composability)**.** *A FP component is composable with a resource provisioning component that guarantees β amount of resource if*

$$\forall\ t \quad \beta_1^A(t) \le \beta(t), \tag{3}$$

with β_1^A computed following Equation (2).

The composability obtained with the former assumed bound $\beta_1^A(t)$ guarantee also the schedulability of the FP scheduling.

IV. Probabilistic Real-Time Components

In this section we propose a probabilistic interface model for real-time components. Our contribution is inspired by the probabilistic network calculus [19], [29].

Let G denote the set of non-negative wide-sense increasing curves, where for each function $\alpha(t)$, there holds $G = \{\alpha(\cdot) \ : \ \forall\ 0 \le x \le y, 0 \le \alpha(x) \le \alpha(y)\}$. With $\overline{G}$ we denote the set of non-negative wide-sense decreasing functions where for each function $f(x)$, there holds $\overline{G} = \{f(\cdot) \ : \ 0 \le x \le y, 0 \le f(y) \le f(x)\}$.

The process a is upper bounded by the arrival curve $\alpha(t) - x$, then for any time instant $0 \le t_1 \le t_2$ and $t = t_2 - t_1$ there holds $a(t_2) - a(t_1) \le \alpha(t) - x$ which means that $\alpha(t_2 - t_1) - [a(t_2) - a(t_1)] \ge x$. In [19] the definition of the probabilistic arrival curve as follows.

Definition IV.1 (Probabilistic Arrival Curve)**.** *A task is said to have a probabilistic arrival curve (pac) $\alpha \in G$ with bounding function $f \in \overline{G}$, denoted by $\langle f, \alpha \rangle$, if for all $t_2 \ge t_1 \ge 0$ and all $x \ge 0$, there holds*

$$P\{sup_{0 \le t_1 \le t_2}\{\alpha(t_2 - t_1) - [a(t_2) - a(t_1)]\} > x\} \le f(x).$$

With the same reasoning it is also possible to define the probabilistic service curve. The service provisioning b is lower bounded by the curve $\beta(t) + x$, then for any $0 \le t_2 \le t_1$ there holds $b(t_2) - b(t_1) \ge \beta(t_2 - t_1) + x$ which means $[b(t_2) - b(t_1)] - \beta(t_2 - t_1) \ge x$. We define the probability service curve as follows.

Definition IV.2 (Probabilistic Service Curve)**.** *A system is said to provide the arrival with a probabilistic service curve (psc) $\beta \in G$ with bounding function $g \in \overline{G}$, denoted by $\langle g, \beta \rangle$, if for all $t_2 \ge t_1 \ge 0$ and $x \ge 0$, there holds*

$$P\{sup_{0 \le t_1 \le t_2}\{[b(t_2) - b(t_1)] - \beta(t_2 - t_1)\} > x\} \le g(x).$$

The probabilistic output curves can be inferred by applying the relationship among the arrival and service inputs. From [29], the probabilistic output arrival curve is characterized as follows.

Definition IV.3 (Output Characterization)**.** *Consider a system component providing an probabilistic service curve $\langle \beta, g \rangle$, with $\beta \in G$ and bounding function $g \in \overline{G}$; consider the component arrival which has a probabilistic arrival curve $\langle \alpha, f \rangle$, with $\alpha \in G$ and the arrival bounding function $f \in \overline{G}$. The departure flow has a probabilistic output curve (poc) $\langle \alpha', f' \rangle$ with $\alpha' = \alpha \overline{\oslash} \beta(t_2 - t_1)$ and the bounding function $f' = f * g \in \overline{G}^3$, bounded as*

$$P\{sup_{0 \le t_1 \le t_2}\{\alpha \overline{\oslash} \beta(t_2 - t_1) - [a'(t_2) - a'(t_1)]\} \ge x\} \\ \le f * g(x).$$

We propose the probabilistic version of the residual curve.

Theorem IV.4 (Residual Service Curve)**.** *Consider a system component providing an probabilistic service curve $\langle \beta, g \rangle$ with $\beta \in G$ and its bounding function $g \in \overline{G}$, and the arrival which has a probabilistic arrival curve $\langle \alpha, f \rangle$ with $\alpha \in G$ and the bounding function $f \in \overline{G}$. Let $b'(t_2) = b(t_2) - a(t_2)$ be the system residual process and β' its bounded residual service curve, then b' is bounded by*

$$P\{[b'(t_2) - b'(t_1)] - \beta \oslash \alpha(0) > x\} \le f * g(x). \tag{4}$$

$\langle \beta', g' \rangle$ is the probabilistic residual curve (prc) with $\beta' = \beta \oslash \alpha(0)^4$, and $g'(x) = f * g(x)$.

Proof: For any $t_2 \ge t_1 \ge 0$, by the definition of residual service $b'(t_2) = b(t_2) - a(t_2)$, there holds

$$\begin{aligned} b'(t_2) - b'(t_1) &= b(t_2) - a(t_2) - b(t_1) + a(t_1) \\ &= [b(t_2) - b(t_1)] - [a(t_2) - a(t_1)] \\ &= [b(t_2) - b(t_1)] - \beta(t_2 - t_1) \\ &\quad + \beta(t_2 - t_1) - \alpha(t_2 - t_1) \\ &\quad + \alpha(t_2 - t_1) - [a(t_2) - a(t_1)] \\ &\le \sup_{0 \le t_1 \le t_2} \{[b(t_2) - b(t_1)] - \beta(t_2 - t_1)\} \\ &\quad + \sup_{0 \le t_1 \le t_2} \{\beta(t_2 - t_1) - \alpha(t_2 - t_1)\} \\ &\quad + \sup_{0 \le t_1 \le t_2} \{\alpha(t_2 - t_1) - [a(t_2) - a(t_1)]\} \end{aligned}$$

$$\begin{aligned} b'(t_2) - b'(t_1) - \sup_{0 \le k \le t_2} \{\beta(k) - \alpha(k)\} = \\ = \sup_{0 \le t_1 \le t_2} \{[b(t_2) - b(t_1)] - \beta(t_2 - t_1)\} \\ + \sup_{0 \le t_1 \le t_2} \{\alpha(t_2 - t_1) - [a(t_2) - a(t_1)]\}. \end{aligned}$$

[3] Among bounding functions it is the convolution $*$ with the classical algebra; with the RTC curves is the min-plus convolution $\otimes$.

[4] As defined in the classical RTC [10], and by the notion of min-plus deconvolution

where the right-hand side of the equation implies a sufficient condition to obtain $P\{b'(t_2) - b'(t_1) - \sup_{0 \le k \le t_2}\{\beta(k) - \alpha(k)\} < x\}$, which are that $P\{sup_{0 \le t_1 \le t_2}\{\alpha(t_2 - t_1) - [a(t_2) - a(t_1)]\} > x\}$ and $P\{sup_{0 \le t_1 \le t_2}\{[b(t_2) - b(t_1)] - \beta(t_2 - t_1)\} > x\}$ are known. To ensure the system stable, assume there yields

$$lim_{k \to \infty} \frac{1}{k}[\beta(k) - \alpha(k)] \le 0.$$

From the lemma in [29] and $\sup_{k \ge 0}\{\beta(k) - \alpha(k)\} = \beta \oslash \alpha(0)$ we conclude that

$$P\{b'(t_2) - b'(t_1) - \beta \oslash \alpha(0) \ge x\} \le f * g(x)$$

■

We have derived a probabilistic interpretation of the RTC curves where each curve has a probability function associated. With the inputs and the outputs of a generic real-time systems component and the algebra for probabilistic curves we have defined probabilistic real-time interfaces for real-time components by which start reasoning in terms of probabilistic real-time calculus. The interface is given $(\langle \alpha, f \rangle, \langle \beta, g \rangle, \langle \alpha', f' \rangle, \langle \beta', g' \rangle)$.

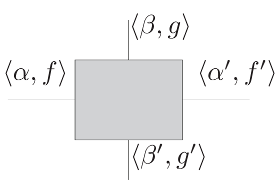


Figure 3. Probabilistic real-time component with it probabilistic real-time interface representation.

A. Probabilistic Schedulability and Composability

We propose a schedulability conditions for the tasks that that have the worst-case execution times and the inter-arrival times given by random variables as modeled in Section II. The tasks are scheduled under a FP policy.

The schedulability conditions comes from the the arrival and service curves. The main interest of these conditions is that they involves tasks with parameters given by random variables. Considering a task set Γ of n tasks $\tau_i = (O_i, \mathcal{C}_i, \mathcal{T}_i, D_i, p_i)$ with τ_1 the highest priority task and τ_n the lowest. In this case all the arrivals are upper bounded by $\alpha_i^u(t) = \lceil \frac{t}{T_i^{min}} \rceil C_i^{max}$ and lower bounded by $\alpha_i^l(t) = \lceil \frac{t}{T_i^{max}} \rceil C_i^{min}$.

Theorem IV.5 (Probabilistic FP Schedulability)**.** *A task set Γ is schedulable under FP with a probabilistic resource provisioning $\langle \beta, g \rangle$ if*

$$\forall i \quad \exists t_0 \in schedP_i \quad \omega_i(t_0) \le \beta(t_0), \tag{5}$$

with the probabilistic level-i workload $\langle \omega_i, h_i \rangle$ defined as $\omega_i(t) = \sum_{\tau_j \in hp(\tau_i)} \alpha_j^u(t)$, and $h_i(x) = \sum_{\tau_j \in hp(\tau_i)} f_i(x)$.

As for the schedulability, the composability can be extended to the case of probabilistic service curves.

Theorem IV.6 (Probabilistic Resource Composability)**.** *A component i asking for resource $\langle \beta_i, g_i \rangle$ is composable to a component j providing resource $\langle \beta_j, g_j \rangle$ if*

$$\forall \ t \quad \beta_i(t) \le \beta_j(t) \tag{6}$$

The former two theorems applies the classical schedulability conditions although with probabilistic curves, thus do not need a demonstration. In future works the probability function will be accounted for in the schedulability criteria resulting in more flexible conditions.

V. Discrete Probability Bounds

In this section we apply the probabilistic real-time calculus formerly introduced, to real cases with discrete probability distributions. In particular, we couple the RTC representation, and its defined probabilistic version with the known probabilistic real-time modeling.

The probabilistic task model tells that each task instance can have a certain inter-arrival interval and computation time, and the probability associated to them tells the probability to have that arrival instant and computation time for the next job instance.

In case of probabilistic tasks, specifically the case where the period of τ is described by a random variable, the upper bound to the arrivals is obtained from the least possible period for the task, $\alpha^u(t) \stackrel{def}{=} \lceil \frac{t}{T^{min}} \rceil \cdot C$. The lower bound instead comes form the largest possible period for that task, $\alpha^l(t) \stackrel{def}{=} \lfloor \frac{t}{T^{max}} \rfloor \cdot C$. Those two curves bounds any possible instance of the task period within $[T^{min}, T^{max}]$.

Example V.1. *Given a task τ_i*

$$\tau_i = (0, 1, \begin{pmatrix} 2 & 3 & 4 \\ 0.5 & 0.3 & 0.2 \end{pmatrix}, 2),$$

with the period described by a random variable, while the rest of its parameter are deterministic. All the possible arrivals of τ in the interval domain along all of its instances are upper bounded by the curve $\alpha_i^u(t) = \lceil \frac{t}{2} \rceil \cdot 1$. The lower bound is described by $\alpha_i^l(t) = \lfloor \frac{t}{4} \rfloor \cdot 1$. See Figure 4.

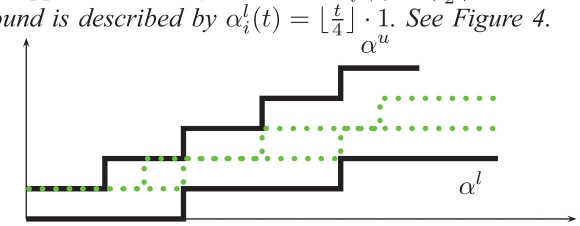
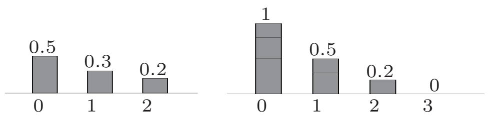


Figure 4. Upper and lower bound of the arrivals for τ_i and all the possible arrivals in between in the interval domain.

The rest of the period distribution derives a set of arrival curves which are not bounding the whole arrivals of the probabilistic task. The curve $\alpha_{i,0}(t) = \lceil \frac{t}{3} \rceil \cdot 1$ is obtained assuming all the job instances with arrivals every 3 time



(a) Distribution (b) Cumulative distribution

Figure 5. The example distribution of the samples with the relative probabilities and a cumulative version.

units. Such a curve cannot bound 100% *of the job instance arrivals; instead it can bound the case where all the instances have arrivals either every* 3 *time unit, or every* 4 *or a mixture among the two According to the distribution provided, the cases bounded by* $\alpha_{i,1}$ *represents the* 50% *of the total, where* $50\% = 30\% + 20\%$*. The remaining* 50% *is the probability of having instances with arrivals every* 2 *time units. It is enough just one instance with such a period (*$T = 2$*) that the task arrivals cannot be bounded by* $\alpha_{i,1}$ *anymore. This means that* $\alpha_{i,1}$ *does not bound the remaining* 50% *of the cases. Finally,* $\alpha_{i,2}(t) = \lceil \frac{t}{4} \rceil \cdot 1$ *bounds only* 20% *of the cases, those that have only arrivals every* 4 *time units. Figure 5 shows the distribution of the values and the cumulative function obtained. The indexes are relative to the distribution values.*

We consider the task model $\tau_i = (O, C_i, \begin{pmatrix} T_i^{min} \equiv T_{i,0} & T_{i,1} & \dots & T_{i,k_{T_i}} \equiv T_i^{max} \\ P_{i,0} & P_{i,1} & \dots & P_{i,k_{T_i}} \end{pmatrix}, D_i)$.

Given a discrete distribution, it is easy to derive its cumulative probability function $f_i(x)$ such as

$$f_i(x) = \begin{cases} P_{i,0} + P_{i,1} + \dots P_{i,k_{T_i}} = 1 & \text{if } x \leq T_{i,0} \\ P_{i,1} + P_{i,2} + \dots P_{i,k_{T_i}} & \text{if } x = T_{i,1} \\ \dots & \\ P_{i,k_{T_i}} & \text{if } x = T_{i,k_{T_i}} \\ 0 & \text{if } x > T_{i,k_{T_i}} \end{cases}$$

The cumulative function $f_i(x)$ is applied for the upper bound analysis, as intuitively expressed by Example V.2. $f_i(x)$ is the bounding function for the discrete arrival curve which is defined as

$$\alpha_i(t, x) = \begin{cases} \alpha^u_{i,0}(t) & \text{if } x \leq T_i^{min} \equiv T_{i,0} \\ \alpha^u_{i,1}(t) & \text{if } x = T_{i,1} \\ \dots & \\ \alpha^u_{i,k_{T_i}}(t) & \text{if } x = T_{i,k_{T_i}}, \\ \alpha^l_i(t) & \text{if } x > T_{i,k_{T_i}}, \end{cases}$$

That curve depends on x and it bounds the application arrivals for any interval t. α_i and $f_i(x)$ defines the probabilistic arrival curve $\langle \alpha_i, f_i \rangle$ such that

$$P\{\sup_{0 \leq t_1 \leq t_2} \{\alpha_i(t_2 - t_1, x) - [a_i(t_2) - a_i(t_1)]\} > 0\} \geq f_i(x).$$

The same idea can be applied for tasks with a computation time described by a random variable. An order among the samples and cumulative functions of the distribution can be created.

Example V.2. *Given a task* τ_i

$$\tau_i = (0, \begin{pmatrix} 1 & 2 & 3 \\ 0.5 & 0.3 & 0.2 \end{pmatrix}, 10, 10),$$

with the computation time described by a random variable, while the rest of its parameter are deterministic. All the possible arrival of τ_i *in the interval domain along all its instances are upper bounded by the curve* $\alpha^u_i(t) = \lceil \frac{t}{10} \rceil \cdot 3$ *upper bounds all the possible arrivals of* τ_i *because given by the least possible period for arrivals of* τ_i*. The lower bound is described by* $\alpha^l_i(t) = \lfloor \frac{t}{10} \rfloor \cdot 1$*. See Figure 6.*

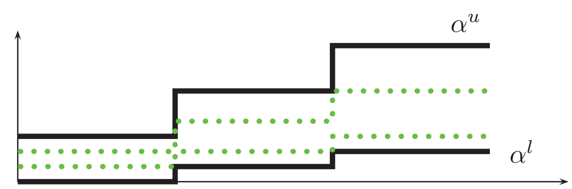


Figure 6. Upper and lower bound of the arrivals for τ_i and all the possible arrivals in between in the interval domain.

The rest of the distribution derives a set of arrival curves which are not bounding the whole arrivals of the probabilistic task. The curve $\alpha_{i,0}(t) = \lceil \frac{t}{10} \rceil \cdot 2$ *is obtained assuming all the job instances with a computation time of* 2 *time units. Such a curve cannot bound* 100% *of the job instances, while it can bound the case where all the instances have computation time of* 2 *or* 2*. Even a mixture among the two cases is still bounded by* $\alpha_{i,0}$*. According to the sample distribution, the cases bounded by* $\alpha_{i,1}$ *represents the* 50% *of the cases, which comes form the summation of* 30% *and* 20%*. The remaining* 50% *is represented by the probability of having instances with computation time larger than* 3*:* $\alpha_{i,1}$ *does not bound* 50% *of the cases.* $\alpha_{i,2}(t) = \lceil \frac{t}{10} \rceil \cdot 1$ *bounds only* 20% *of the cases, those that have computation time of* 4*. Since the probability distribution is the same as the former example, Figure 5 depicts this case as well.*

The probabilistic computation time case and the probabilistic period one can be combined obtaining the most general case for a task $\tau_i = (O_i, \mathcal{C}_i, \mathcal{T}_i, D_i)$. Its probabilistic arrival curve is $\langle \alpha_i, f_i \rangle$ follows the former definitions.

The probabilistic interface for a real-time component is defined by the tuple $(\langle \alpha, f \rangle, \langle \beta, g \rangle, \langle \alpha', f' \rangle, \langle \beta', g \rangle)$, and with that it is possible to verify the schedulability and the composability of probabilistic real-time components as exploited in the previous section. With the probabilistic curves model it is possible to define more flexible schedulability conditions by taking into account the bound and the accuracy level we want to consider.

The real-time calculus abstraction allows to model both the random inter-arrival time and the random computation time of a task. This unifies the probabilistic real-time analysis.

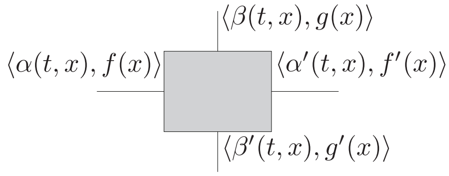


Figure 7. Probabilistic real-time component with its probabilistic real-time interface representation. Discrete random variable case.

VI. Test Case

We consider a peculiar case where to apply the probabilistic framework we have developed. We refer our test-case to a basic hierarchical scheduling system, with two components representing two tasks τ_1, τ_2 that are scheduled according a fixed-priority scheduling policy.

The hierarchy among the components is defined through priority of the tasks. The most priority one receives the computational resource first. The resource amount it does not apply to execute is the passed to the second one which can execute only when the first is not executing. We assume τ_1 and τ_2 as probabilistic tasks and τ_1 with an higher priority than τ_2. The two tasks are defined as

$$\tau_1 = (0, \begin{pmatrix} 1 & 2 \\ 0.3 & 0.7 \end{pmatrix}, \begin{pmatrix} 5 & 6 \\ 0.6 & 0.4 \end{pmatrix}, 10)$$

$$\tau_2 = (0, \begin{pmatrix} 1 & 2 \\ 0.3 & 0.7 \end{pmatrix}, 10, 10)$$

where τ_1 has inter-arrival time and computation time described by random variable, while τ_2 has just the computation time as a random variable.

$$f_1(x) = \begin{cases} 0.42+0.28+0.18+0.12 = 1 & \text{if } x \le 0 \\ 0.28+0.18+0.12 = 0.58 & \text{if } x = 1 \\ 0.18+0.12 = 0.3 & \text{if } x = 2 \\ 0.12 & \text{if } x = 3 \\ 0 & \text{if } x \ge 4 \end{cases}$$

$$\alpha_1(t,x) = \begin{cases} \lceil \frac{t}{5} \rceil \cdot 2 & \text{if } x \le 0 \\ \lceil \frac{t}{6} \rceil \cdot 2 & \text{if } x = 1 \\ \lceil \frac{t}{5} \rceil \cdot 1 & \text{if } x = 2 \\ \lceil \frac{t}{6} \rceil \cdot 1 & \text{if } x = 3 \\ \lfloor \frac{t}{6} \rfloor \cdot 1 & \text{if } x \ge 4 \end{cases}$$

$$f_2(x) = \begin{cases} 0.4+0.6 = 1 & \text{if } x \le 0 \\ 0.6 & \text{if } x = 1 \\ 0 & \text{if } x \ge 2 \end{cases}$$

$$\alpha_2(t,x) = \begin{cases} \lceil \frac{t}{10} \rceil \cdot 2 & \text{if } x \le 0 \\ \lceil \frac{t}{10} \rceil \cdot 1 & \text{if } x = 1 \\ \lfloor \frac{t}{10} \rfloor \cdot 1 & \text{if } x \ge 2 \end{cases}$$

We have represented the probabilistic arrival curves of the two task, $\langle\alpha_1, f_1\rangle$, $\langle\alpha_2, f_2\rangle$. See Figures 9 for details about α_1.

Because of the composition of the two tasks, to the high priority task τ_1 it has to be guaranteed a probabilistic service curve $\langle\beta_1, g_1\rangle$ where $\beta_1 = max\{\beta_1^\sharp, \alpha_1(t - D_1)\}$ with $\beta_1^\sharp(t) = \beta_2(t-\lambda) + \alpha_1(t-\lambda)$ and $\lambda = sup\{\psi : \beta_2(t-\psi) = \beta_2\}$, according to Equation (2). And $g_1(x) = max\{g_2 \overline{*} f_1(x), f_1(x)\}$, which comes from the implicit task hierarchy due to the fixed priority scheduling. The deconvolution $\overline{*}$, as the inverse of the convolution operator in Equation 4, derives the bounding function of high priority service curves from lower priority ones. It has been inferred from similar reasoning as the one applied for Equation 4, but a more formal proof will be provided in future works. Furthermore, $g_2 = f_2$ since $\beta_2 = \alpha_2(t - D_2)$ in order to have the τ_2 scheduled on time.

$$g_2(x) = f_2(x) = \begin{cases} 0.4+0.6 = 1 & \text{if } x \le 0 \\ 0.6 & \text{if } x = 1 \\ 0 & \text{if } x \ge 2 \end{cases}$$

$\beta_1^\sharp$ and β_1 can be easily computed with the operators defined in the modular performance analysis framework in [30].

$\langle\beta_1, g_1\rangle$ defines the probabilistic function as the minimum resource requirement of the fixed-priority composition of the two tasks that guarantees the composability and then schedulability of the FP macro component.

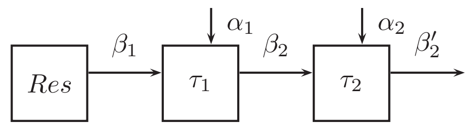


Figure 8. Two task components with a fixed-priority resource scheduling.

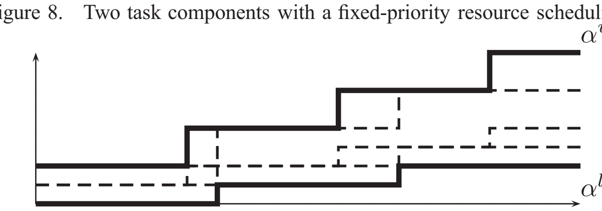


Figure 9. Upper and lower bound of the arrivals for τ_1 and all the possible arrivals in between in the interval domain.

VII. Conclusions

We introduce in this paper the problem of timing analysis of probabilistic real-time systems making use of abstraction frameworks. We have extended the real-time calculus to the probabilistic case, and the RTC algebra by taking into account probability functions. In this way we develop flexible abstractions able to cope with different grades of real-time schedulability and quality of service.

We would like to extend, within future work, the model developed here toward a complete probabilistic real-time calculus. The flexible schedulability and composability conditions, that we obtained, require a deeper investigation in order to outline their full potential.

Acknowledgment

This work was partially founded by the European Community's Seventh Framework Programme FP7/2009-2013 under grant agreement no. 249100.

References

[1] I. Shin and I. Lee, "Compositional real-time scheduling framework with periodic model," *ACM Trans. Embed. Comput. Syst.*, vol. 7, no. 3, pp. 1–39, 2008.

[2] J. Regehr and J. A. Stankovic, "Hls: A framework for composing soft real-time schedulers," in *IEEE Real-Time Systems Symposium*, 2001, pp. 3–14.

[3] S. P. Marimuthu and S. C. Chakraborty, "A framework for compositional and hierarchical real-time scheduling," *Real-Time Computing Systems and Applications, International Workshop on*, vol. 0, pp. 91–96, 2006.

[4] J. L. Lorente, G. Lipari, and E. Bini, "A hierarchical scheduling model for component-based real-time systems," in *Proc. of IPDPS'06*, 2006.

[5] R. I. Davis and A. Burns, "Resource sharing in hierarchical fixed priority pre-emptive systems," in *Proc. of the 27th IEEE International Real-Time Systems Symposium (RTSS'06)*, 2006.

[6] A. K. Mok and A. K. Feng, "A model of hierarchical real-time virtual resources," in *RTSS'02, IEEE Computer Society*, pp. 26–35.

[7] S. Chakraborty, S. Kunzli, and S. Thiele, "A general framework for analysing system properties in platform-based embedded system design," in *Proc. of the DATE'03*, 2003.

[8] I. Shin and I. Lee, "Periodic resource model for compositional real-time guarantees," in *RTSS '03: Proceedings of the 24th IEEE International Real-Time Systems Symposium*. Washington, DC, USA: IEEE Computer Society, 2003, p. 2.

[9] ——, "Compositional real-time scheduling framework," in *25th IEEE International Real-Time System Symposium*, 2004, pp. 57–67.

[10] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," in *Proc. ISCAS'00*, vol. 4, 2000, pp. 101–104.

[11] J. L. Boudec and P. Thiran, *Netweork Calculus - A Theory of Deterministic Queuing Systems for the Internet*, S. Verlag, Ed. Springer Verlag, 2001.

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

[13] G. A. Kaczynski, L. L. Bello, and T. Nolte, "Towards stochastic response-time of hierarchically scheduled real-time tasks," in *Proceedings of 11th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2006*, 2006, pp. 453–456.

[14] T. Tia, Z. Z. Deng, M. Shankar, M. Storch, J. Sun, L. Wu, and J. Liu, "Probabilistic performance guarantee for real-time tasks with varying computation times." in *IEEE Real-Time and Embedded Technology and Applications Symposium,*, 1995.

[15] A. Atlas and A. Bestavros, "Statistical rate monotonic scheduling," in *19th IEEE Real-Time Systems Symposium*, 1998.

[16] J. Lehoczky, "Real-time queueing theory," in *10th of the IEEE Real-Time Systems Symposium*, 1990.

[17] J. L. Díaz, D. F. García, K. Kim, C.-G. Lee, L. L. Bello, J. M. López, S. L. Min, and O. Mirabella, "Stochastic analysis of periodic real-time systems," in *RTSS '02: Proceedings of the 23rd IEEE Real-Time Systems Symposium*. Washington, DC, USA: IEEE Computer Society, 2002, p. 289.

[18] D. Kadhim, S. Jobbar, W. Liu, and W. Cheng, *The Stochastic Network Calculus Methodology*. Springer Berlin / Heidelberg, 2009, vol. 208.

[19] J. Xie and Y. Jiang, "Stochastic network calculus models under max-plus algebra," in *Proceedings of the Global Communications Conference, 2009. GLOBECOM 2009, Honolulu, Hawaii, USA, 30 November - 4 December 2009*, 2009, pp. 1–6.

[20] A. Maxiaguine, S. Künzli, and L. Thiele, "Workload characterization model for tasks with variable execution demand," in *DATE '04: Proceedings of the conference on Design, automation and test in Europe*. Washington, DC, USA: IEEE Computer Society, 2004.

[21] J. P. Lehoczky, L. Sha, and Y. Ding, "The rate monotonic scheduling algorithm: Exact characterization and average case behavior," in *IEEE Real-Time Systems Symposium*, 1989, pp. 166–171.

[22] E. Bini and G. C. Buttazzo, "Schedulability analysis of periodic fixed priority systems," *IEEE Transactions on Computers*, vol. 53, no. 11, pp. 1462–1473, Nov. 2004.

[23] L. de Alfaro and T. Henzinger, "Interface theories for component-base design," in *In EMSOFT'01: Embedded Software, Lecture notes in Computer Science 2211*. Springer Verilag, 2001, pp. 148–165.

[24] A. Chakrabarti, L. de Alfaro, T. Henzinger, and M. Stoeling, "Resource interfaces," in *EMSOFT'03*, vol. 2855, 2003, pp. 117–133.

[25] T. A. Henzinger and S. Matic, "An interface algebra for real-time components," in *Proc. of the twelfth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06)*, 2006.

[26] E. Wandeler and L. Thiele, "Real-time interfaces for interface-based design of real-time systems with fixed priority scheduling," in *EMSOFT'05, ACM*, 2005.

[27] L. Thiele, E. Wandeler, and N. Stoimenov, "Real-time interfaces for composing real-time systems," in *EMSOFT*, 2006, pp. 34–43.

[28] K. Huang, L. Santinelli, J.-J. Chen, L. Thiele, and G. C. Buttazzio, "Periodic power management schemes for real-time event streams." in *the 48th IEEE Conf. on Decision and Control (CDC)*, Shanghai, China, 2009, pp. 6224–6231.

[29] Y. Jiang, "A basic stochastic network calculus," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 123–134, 2006.

[30] "Modular performance analysis," http://www.mpa.ethz.ch/Rtctoolbox/Overview.