JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 6, November 2009

Characterization of the Space of Feasible Worst-Case Execution Times for Earliest-Deadline-First Scheduling

Laurent George*
University of Paris 12, 94400 Vitry sur Seine, France

and

Jean-François Hermant†
ECE, LACSC, Paris, France

DOI: 10.2514/1.44721

This paper presents a sensitivity analysis for the dimensioning of real-time systems in which sporadic tasks are executed according to the preemptive earliest deadline first scheduling policy. The timeliness constraints of the tasks are expressed in terms of late termination deadlines. New results for earliest deadline first are shown, which enable us to determine the space of feasible worst-case execution times, denoted the C-space, valid for any configuration of worst-case execution times. The C-space is such that any task set with its worst-case execution times in the C-space domain is feasible with earliest deadline first. We show in a first approach that the C-space domain is convex, a property that can be used to reduce the number of inequalities characterizing the C-space domain. We propose in another approach to reduce the number of inequalities based on the concept of worst-case busy period for worst-case execution times in the C-space. This approach can be used for busy periods meeting a given property. We apply the two approaches on an example, and we compare the C-space obtained with earliest deadline first scheduling to the C-space obtained with deadline monotonic scheduling.

I. Introduction

THIS paper considers the problem of correctly dimensioning real-time systems. The correct dimensioning of a real-time system strongly depends on the determination of the worst-case execution times (WCETs) of the tasks. Based on the WCETs, a feasibility condition (FC) [1–3] can be established to ensure that the timeliness constraints of all the tasks are always met, regardless of their release times, when they are scheduled with either a fixed or dynamic priority-driven preemptive scheduling algorithm. The timeliness constraints are expressed in terms of late termination deadlines imposed on the completion times of the tasks. The task model is the classical sporadic model. A sporadic task set $\tau = \{\tau_1, \dots, \tau_n\}$ is composed of n sporadic tasks, where a sporadic task τ_i is defined by:

- x_i: its WCET;
- T_i: its minimum inter-arrival time (also called, by extension, the period);
- D_i: its relative deadline (a task released at time t must be executed by its absolute deadline $t + D_i$).

In the sequel, we assume the general case where deadlines and periods are independent.

A recent research area called sensitivity analysis aims at providing interesting information on the validity of FC when changing task WCETs [4], task periods [4], or task deadlines [5]. This permits, for example, finding a feasible

Received 4 April 2009; accepted for publication 4 April 2009.

* Associate Professor, LISSI, University of Paris 12, 120 rue Paul Armangot, 94400 Vitry sur Seine, France, lgeorge@ieee.org
† Associate Professor, ECE, LACSC, 37 quai de Grenelle, Paris, France, hermant@ieee.org

task set, if the current one is not feasible, by modifying the task parameters (WCETs, periods, or deadlines) or determining the impact of an architecture change on the feasibility of a task set (WCET change). In this paper, we are interested in the sensitivity of WCETs. We want to determine the C-space feasibility domain as defined in [6] when tasks are scheduled with preemptive earliest deadline first (EDF). The C-space is a region of n dimensions where each dimension denotes the WCET of a task such that for any vector $X = \{x_1, \ldots, x_n\}$ in the C-space, task set τ is feasible.

In the case of fixed-priority (FP) scheduling, when deadlines are less than or equal to periods, Bini et al. [4] have shown how to compute the maximum expansion factor α applied to all the WCETs of the tasks to remain in the C-space at a reasonable cost (see Sec. III), such that $\forall i \in \{1, \ldots, n\}$, $\tau_i \in \tau$, the WCET of τ_i is $\alpha\, x_i$. They finally propose a parametric equation of the C-space, detailed in Sec. III. In the general case, when deadlines and periods are independent, α can be computed by successive iterations, where each iteration requires a pseudo-polynomial time complexity.

Characterizing the space of feasible WCETs can help proposing new services for the temporal robustness of real-time systems. This corresponds to the current trends in real-time systems for which a temporal protection service is proposed to handle WCET overruns. If a WCET overrun occurs, an interrupt is raised and a service is called to handle the problem. The C-space could be used to determine on-line if the WCET overrun can be tolerated or if a correction should be undertaken (stop task or put it in background for example). Among such systems, we can cite the AUTOSAR specification used for automotive applications [7], the real-time specification for Java (RTSJ) [8,9] which specify such services (even if no solution is currently provided in real systems).

The rest of the paper is organized as follows. Section II reviews classical concepts for uniprocessor scheduling. Section III presents the state-of-the-art in real-time scheduling with a focus on FP scheduling sensitivity analysis and EDF scheduling. In Sec. IV, we present new results on EDF scheduling that are used for a sensitivity analysis of the WCETs. We show how to derive from an analysis of EDF in time interval $[\min(D_1, \ldots, D_n), \mathrm{lcm}(T_1, \ldots, T_n))$ the C-space region parametric equations (lcm: least common multiple) and show that it is convex. We then show how to reduce the number of times characterizing the C-space by two approaches, the first one is based on a linear programming approach used to remove the non-significant constraints defining the C-space. We use the simplex algorithm to determine if a constraint can be removed. The second approach is based on the computation of the maximum busy period obtained for any WCET in the C-space. In Sec. V, we show in an example how to determine the C-space domain and how to reduce its complexity. We compare the C-space obtained for EDF to the one obtained for FP scheduling using the deadline monotonic (DM) algorithm. Finally, we give some conclusion.

II. Concepts and Notations

We recall classical results in the uniprocessor context for real-time scheduling.

- A task set is said to have constrained deadlines if $\forall i \in \{1, \ldots, n\}$, $D_i \leqslant T_i$.
- A task set is said to have arbitrary deadlines if no constraint is imposed between the deadlines and the periods of the tasks.

The C-space characterization given in this paper for EDF is valid for arbitrary deadlines. Nevertheless, the reduction of the number of times characterizing the C-space is more efficient for tasks with constrained deadlines.

- A task is said to be non-concrete if its request time is not known in advance. In this paper, we only consider non-concrete request times, since the activation request times are supposed to be unpredictable.
- Given a non-concrete task set, the synchronous scenario corresponds to the scenario where all the tasks are released at the same time, at time 0.
- EDF is the preemptive version of earliest deadline first non-idling scheduling. EDF schedules tasks according to their absolute deadlines: the task with the shortest absolute deadline has the highest priority. Ties are broken arbitrarily.
- FP is a preemptive fixed-priority scheduling according to an arbitrary priority assignment. We assume that tasks are indexed by decreasing priorities.
- For FP, $\mathrm{hp}(i)$ denotes the subset of all the tasks, except τ_i, with a priority higher than or equal to that of τ_i.
- DM is a preemptive FP scheduling where the highest priority is given to the task with the shortest relative deadline. Ties are broken arbitrarily.

- A task set is said to be valid with a given scheduling policy if and only if no task occurrence ever misses its absolute deadline with this scheduling policy.
- $U = \sum_{i=1}^{n} x_i/T_i$ is the processor utilization factor, i.e., the fraction of processor time spent in the execution of the task set [10]. If $U > 1$, then no scheduling algorithm can meet the task deadlines.
- $W_i(t) = x_i + \sum_{\tau_j \in \mathrm{hp}(i)} \lceil t/T_j \rceil x_j$. $W_i(t)$ is the cumulative workload of all the tasks in the synchronous scenario, including the first request of τ_i at time 0 and all the tasks in hp(i) whose release times are in time interval $[0, t)$.
- $W(t) = \sum_{j=1}^{n} \lceil t/T_j \rceil x_j$. $W(t)$ is the cumulative workload of all the tasks in the synchronous scenario whose release times are in time interval $[0, t)$. The length of the first busy period is solution of $t = W(t)$.
- The processor demand bound function (DBF) $h(t)$ is the amount of processing time requested by all tasks, whose release times and absolute deadlines are in time interval $[0, t]$ in the synchronous scenario [1], where $\lfloor x \rfloor$ returns the integer part of x. We have for a given task set

$$\tau : h(t) = \sum_{j=1}^{n} h_j(t) x_j, \quad \text{where } h_j(t) = \mathrm{Max}\left\{0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\}$$

- $D_{\min}$ is the minimum deadline ($D_{\min} = \mathrm{Min}\{D_1, \ldots, D_n\}$).
- P is the least common multiple of the task periods ($P = \mathrm{lcm}\{T_1, \ldots, T_n\}$).

III. State-of-the-Art

For FP scheduling, necessary and sufficient FCs have been proposed, based on the computation of the task worst-case response times [2,7]. The worst-case response time is obtained in the worst-case synchronous scenario and is computed by successive iterations. A task set is then declared feasible if the worst-case response time of any task in the synchronous scenario is less than or equal to its deadline.

In the case of tasks set with constrained deadlines, the worst-case response time r_i of a task τ_i is obtained in the synchronous scenario for the first release of τ_i at time 0 and is solution of equation [11] $r_i = W_i(r_i)$. r_i is computed by successive iterations and the number of iterations is bounded by $1 + \sum_{\tau_j \in \mathrm{hp}(i)} \lfloor D_i/T_j \rfloor$. This FC has the drawback of being recursive. This FC has been revisited by [12] that proposes a non recursive FC for tasks scheduled with DM in the case of constrained deadlines. This FC is described in the following theorem.

Theorem III.1 [12]: Let $\tau = \{\tau_1, \ldots, \tau_n\}$ be a sporadic task set with constrained deadlines scheduled with DM. A task τ_i is feasible if and only if:

$$X_i = \min_{t \in S_i} \left\{ \frac{W_i(t)}{t} \right\} \leqslant 1, \quad \text{where } S_i = \{D_i\} \cup \left\{ kT_j : j \in \mathrm{hp}(i), k = 1, \ldots, \left\lfloor \frac{D_i}{T_j} \right\rfloor \right\}$$

Task set τ is schedulable with DM if and only if: $\max_{i=1,\ldots,n}\{X_i\} \leqslant 1$.

For any task τ_i, the times to check correspond to time D_i union the arrival times of the tasks of higher priority than τ_i in time interval $[0, D_i]$.

The feasibility condition given in Theorem 1 has been significantly improved by Manabe and Aoyagi [13] that shows how to reduce the times to consider as follows. For a task τ_i, the times to consider are D_i and a set of times corresponding to the request times of the tasks for tasks in hp(i) indexed by increasing deadlines, τ_1 to τ_{i-1}. For task τ_{i-1}, at $t_{i-1} = k\,T_i$, $k \in \mathbb{N}$, such that $t_{i-1} \leqslant D_i \leqslant t_{i-1} + D_{i-1}$. For task τ_{i-2}, at two times t_{i-2} corresponding to times D_i and t_{i-1}, $t_{i-2} \leqslant D_i \leqslant t_{i-2} + D_{i-2}$, and $t_{i-2} \leqslant t_{i-1} \leqslant t_{i-2} + D_{i-2}$, and so on till task τ_1, leading to at most 2^{i-1} times instead of $1 + \sum_{\tau_j \in \mathrm{hp}(i)} \lfloor D_i/T_j \rfloor$ times.

This improvement has been also proposed by [6] with a recursive equation characterizing the times to consider, for the C-space computation. This result can be used to determine the C-space (n dimensions) feasibility region for the WCETs of a sporadic task set such that any vector $X = \{x_1, \ldots, x_n\}$ of WCETs in the C-space region leads to a feasible task set. The C-space region is then defined as follows.

Theorem III.2 [6]: Let $\tau = \{\tau_1, \ldots, \tau_n\}$ be a set of periodic tasks with constrained deadlines, indexed by decreasing priorities. The C-space region is defined as the region such that $\forall X = \{x_1, \ldots, x_n\} \in \mathbb{R}^{+n}$:

$$\forall i \in \{1, \ldots, n\},\ \exists t \in \mathcal{P}_{i-1}(D_i),\ t = x_i + \sum_{j=1}^{i-1} \left\lceil \frac{t}{T_j} \right\rceil x_j$$

where $\mathcal{P}_i(t)$ is defined by recursive equation:

$$\begin{cases} \mathcal{P}_0(t) = t \\ \mathcal{P}_i(t) = \mathcal{P}_{i-1}\left(\left\lfloor \frac{t}{T_i} \right\rfloor T_i\right) \cup \mathcal{P}_{i-1}(t) \end{cases}$$

When deadlines and periods are independent, Tindell et al. [2] showed that the worst-case response time of a sporadic task τ_i is not necessarily obtained for the first activation request of τ_i at time 0. The number of activations to consider is $1 + \lfloor L_i/T_i \rfloor$, where L_i is the length of the worst-case level-τ_i busy period defined by Lehoczky [14] as the longest period of processor activity running tasks of priority higher than or equal to that of τ_i in the synchronous scenario. It can be shown, when task are indexed by decreasing priorities, that $L_i = \sum_{j=1}^{i} \lceil L_i/T_j \rceil x_j$. From its definition, L_i is bounded by $\sum_{j=1}^{i}(x_j/T_j) \times \text{lcm}(T_1, \ldots, T_i)$ [3].

In that case, the complexity depends on L_i, leading to a pseudo-polynomial time complexity. In such a context, the characterization of the C-space might be very costly. α is computed by iterations, but the computation becomes more and more costly. Indeed, when α increases, the length of the level-τ_i busy period tends toward $\text{lcm}(T_1, \ldots, T_i)$ as the load utilization tends toward 1. As a conclusion for FP scheduling, sensitivity analysis can be proposed in the case of constrained deadlines but not for arbitrary deadlines, because increasing the task WCETs requires recomputing the lengths of the level-τ_i busy periods, $\forall i \in [1, n]$, which tends toward $\text{lcm}(T_1, \ldots, T_i)$ of potentially exponential length.

For EDF scheduling, Baruah et al. [1] showed that a necessary and sufficient feasibility condition is $\forall t \in [0, L), h(t) \leqslant t$, where L is the length of the first busy period in the synchronous scenario. When $U \leqslant 1$, L can be computed by successive iterations and is a solution of $L = W(L)$. With this feasibility test, we have the same drawback as with FP in the general case of independent periods and deadlines, as the value of L increases and tends toward P when we compute α by increasing iterations.

We note that, in both approaches, the dimensioning strongly depends on the values of the WCETs. We now introduce new results for EDF to determine the C-space feasibility domain.

IV. Sensitivity Analysis for EDF

This section is divided into three subsections. In Sec. IV.A, we revisit the classical feasibility condition for EDF based on the processor DBF and establish new results for the feasibility of a sporadic task set scheduled with EDF. In Sec. IV.B, we show how to determine the C-space feasibility domain. The C-space region is expressed with parametric equations. We also show that the C-space is convex. In Sec. IV.C, we show how to reduce the number of constraints characterizing the C-space by two approaches, one based on a linear programming approach, the second one based on the computation of the worst-case busy period valid for any WCETs in the C-space.

A. Revisiting the Feasibility Condition for EDF

The following lemma is an adaptation of [1].

Lemma 1: Let τ be a sporadic task set.

$$\tau \text{ feasible with preemptive EDF} \iff \operatorname*{Sup}_{t \in \mathbb{R}^{+*}} \left\{ \frac{h(t)}{t} \right\} \leqslant 1$$

Proof: The necessary and sufficient feasibility condition for EDF is as follows: task set τ is feasible with preemptive EDF if and only if, $\forall t \in \mathbb{R}^+$, $h(t) \leqslant t$, which is equivalent to: $\mathrm{Sup}_{t\in\mathbb{R}^{+*}}\{h(t)/t\} \leqslant 1$. The condition $U \leqslant 1$ is clearly necessary as $\mathrm{Sup}_{t\in\mathbb{R}^{+*}}\{h(t)/t\} \leqslant 1 \Rightarrow U \leqslant 1$. Indeed, $\lim_{t\to\infty}(\mathrm{Sup}_{t\in\mathbb{R}^{+*}}\{h(t)/t\}) = U$. ■

We now prove the following theorem, which allows us to compute $\mathrm{Sup}_{t\in\mathbb{R}^{+*}}\{h(t)/t\}$.

Theorem 1:

$$\operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\} = \mathrm{Max}\left\{U,\ \operatorname*{Sup}_{t\in[D_{\min},P)}\left\{\frac{h(t)}{t}\right\}\right\}.$$

Proof: First, we show that

$$\mathrm{Max}\left\{U,\ \operatorname*{Sup}_{t\in[D_{\min},P)}\left\{\frac{h(t)}{t}\right\}\right\} \leqslant \operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\}$$

By definition, we have

$$\lim_{t\to+\infty}\left\{\frac{h(t)}{t}\right\} \leqslant \operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\},\ \text{i.e., } U \leqslant \operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\}$$

Furthermore, we have

$$\operatorname*{Sup}_{t\in[D_{\min},P)}\left\{\frac{h(t)}{t}\right\} \leqslant \operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\}$$

It follows that

$$\mathrm{Max}\left\{U,\ \operatorname*{Sup}_{t\in[D_{\min},P)}\left\{\frac{h(t)}{t}\right\}\right\} \leqslant \operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\}$$

Second, we show that

$$\operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\} \leqslant \mathrm{Max}\left\{U,\ \operatorname*{Sup}_{t\in[D_{\min},P)}\left\{\frac{h(t)}{t}\right\}\right\}$$

Given that $h(t)$ returns 0 for all $t \in [0, D_{\min})$, we have

$$\forall t \in [0, P),\quad h(t) \leqslant \operatorname*{Sup}_{t\in[D_{\min},P)}\left\{\frac{h(t)}{t}\right\} t$$

Furthermore, we have

$$\forall t_1 \in \mathbb{R}^+,\quad \forall t_2 \in \mathbb{R}^+,\ t_2 \geqslant t_1,\ h(t_2) - h(t_1) \leqslant W(t_2 - t_1)$$

Consequently, we have

$$\forall t \in [0, P),\ \forall k \in \mathbb{N},\quad \frac{h(t+kP)}{t+kP} \leqslant \frac{h(t) + W(kP)}{t+kP}$$

Hence, we have

$$\frac{h(t+kP)}{t+kP} \leqslant \frac{\mathrm{Sup}_{t\in[D_{\min},P)}\left\{\frac{h(t)}{t}\right\} t}{t+kP} + \frac{U\,kP}{t+kP} \quad \text{and} \quad \frac{h(t+kP)}{t+kP} \leqslant \mathrm{Max}\left\{U,\ \operatorname*{Sup}_{t\in[D_{\min},P)}\left\{\frac{h(t)}{t}\right\}\right\}$$

It follows that

$$\operatorname{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\} \leqslant \operatorname{Max}\left\{U,\ \operatorname{Sup}_{t\in[D_{\min},P)}\left\{\frac{h(t)}{t}\right\}\right\} \qquad \blacksquare$$

We therefore have the following theorem.

Theorem 2: A sporadic task set τ is feasible with preemptive EDF $\Leftrightarrow \operatorname{Sup}_{t\in\mathbb{R}^{+*}}\{h(t)/t\} = \operatorname{Max}\{U, \operatorname{Sup}_{t\in\mathcal{M}}\{h(t)//t\}\} \leqslant 1$, where

$$\mathcal{M} = \bigcup_{j=1}^{n}\left\{D_j + k_j T_j,\ 0 \leqslant k_j \leqslant \left\lceil \frac{P-D_j}{T_j}\right\rceil - 1\right\}$$

Proof: The proof is straightforward from Lemma 1 and Theorem 3.

Set $\mathcal{M} = \bigcup_{j=1}^{n}\{D_j + k_j T_j,\ 0 \leqslant k_j \leqslant \lceil (P-D_j)/T_j\rceil - 1\}$ corresponds to the deadlines of the tasks in time interval $[D_{\min}, P)$ where function $h(t)$ varies. $\blacksquare$

Note that this test is valid for any WCET configuration and will be used to characterize the C-space domain in Sec. IV.B.

We now show in the following two examples that function $h(t)/t$ is not necessarily maximum in the first busy period of the synchronous scenario (i.e., in $[0, L)$). Consider a task set $\tau = \{\tau_1, \tau_2, \tau_3\}$, composed of three tasks.

Example 1:

- $\tau_1 : \{x_1 = 10; T_1 = 50; D_1 = 50\}$;
- $\tau_2 : \{x_2 = 20; T_2 = 100; D_2 = 60\}$;
- $\tau_3 : \{x_3 = 30; T_3 = 200; D_3 = 80\}$.

For this task set, we have $D_{\min} = 50$, $P = 200$, $L = W(L) = 70$ and $U = 0.55$. Figure 1 shows the variations of function $h(t)/t$ w.r.t. time t.

We observe that the maximum values of $h(t)/t$ in time intervals $[D_{\min}, L)$ and $[D_{\min}, P)$ are, respectively, 0.5 and 0.75. This proves that the maximum value of $h(t)/t$ is not necessarily obtained in $[0, L)$ and that the computation of $h(t)/t$ must be done in $[0, P)$. Furthermore, in this example, we see that $h(t)/t$ tends toward U for values of t higher than P. Hence, in this example, we have $\operatorname{Sup}_{t\in\mathbb{R}^{+*}}\{h(t)/t\} = \operatorname{Sup}_{t\in[D_{\min},P)}\{h(t)/t\}$.

Example 2:

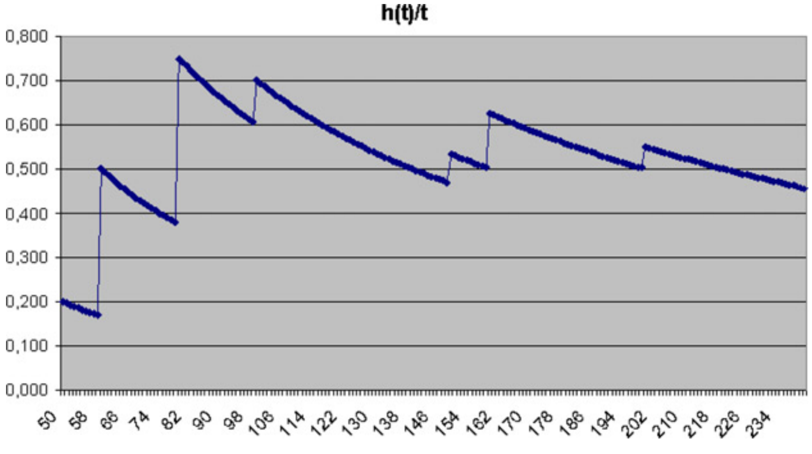

Fig. 1 Variations of $h(t)/t$ w.r.t. time t.

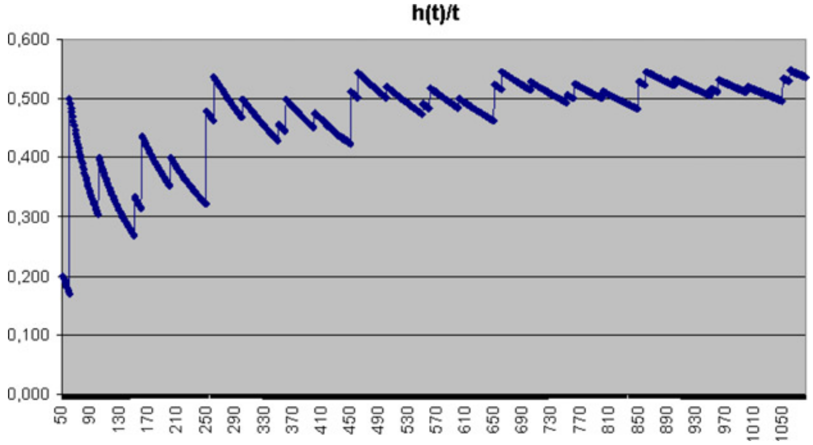

Fig. 2 Variation of $h(t)/t$ w.r.t. time t.

- $\tau_1 : \{x_1 = 10;\ T_1 = 50;\ D_1 = 50\}$;
- $\tau_2 : \{x_2 = 20;\ T_2 = 100;\ D_2 = 120\}$;
- $\tau_3 : \{x_3 = 30;\ T_3 = 200;\ D_3 = 250\}$.

For this task set, we have: $D_{\min} = 50$, $P = 200$, $L = W(L) = 70$ and $U = 0.55$. Figure 2 shows the variations of function $h(t)/t$ w.r.t. time t.

We observe that the maximum values of $h(t)/t$ in time intervals $[D_{\min}, L)$ and $[D_{\min}, P)$ are, respectively, 0.5 and 0.5, which is less than $U = 0.55$. We see again that $h(t)/t$ tends toward U for values of t higher than P. Hence, in this example, we have $\mathrm{Sup}_{t \in \mathbb{R}^{+*}}\{h(t)/t\} = U$.

B. The C-space Feasibility Domain for EDF

Theorem 3: The C-space feasibility domain $\mathcal{D}^{\mathrm{EDF}}(\tau) \subset \mathbb{R}^{+n}$ of $X = (x_1, \ldots, x_n)$ is defined as the subset of $X \in \mathbb{R}^{+n}$ such that:

$$\operatorname*{Sup}_{t \in \mathbb{R}^{+*}} \left\{ \frac{1}{t} \sum_{j=1}^{n} h_j(t) x_j \right\} \leqslant 1$$

This defines all the task sets feasible with EDF. We have

$$\mathcal{D}^{\mathrm{EDF}}(\tau) = \left\{ X \in \mathbb{R}^{+n}, \mathrm{Max} \left(\operatorname*{Sup}_{t \in \mathcal{M}} \left\{ \frac{1}{t} \sum_{j=1}^{n} h_j(t) x_j \right\}, \sum_{j=1}^{n} \frac{x_j}{T_j} \right) \leqslant 1 \right\}$$

where:

$$\mathcal{M} = \bigcup_{j=1}^{n} \left\{ D_j + k_j T_j,\ 0 \leqslant k_j \leqslant \left\lceil \frac{P - D_j}{T_j} \right\rceil - 1 \right\}$$

Proof: Straightforward from Theorem 3. ∎

Set $\mathcal{M}$ denotes the set of points of discontinuity of processor DBF $h(t)$ in time interval $[D_{\min}, P)$, and m denotes the cardinal of $\mathcal{M}$. From Theorem 4, the C-space feasibility domain $\mathcal{D}^{\mathrm{EDF}}(\tau)$ is defined by a set of $m + 1$ constraints. The first m constraints are derived from the set of times in $\mathcal{M}$, while the $(m + 1)$th constraint is derived from the load utilization. We now show how to reduce the times we need to consider in $\mathcal{M}$, i.e., how to extract, from the first m constraints, the subset of times in $\mathcal{M}$ representing the most constrained inequalities, i.e., times where

$\mathrm{Sup}_{t \in \mathbb{R}^{+*}}\{h(t)/t\}$ is obtained. For any time t_i, starting from time t_m down to t_1, we show how to determine if time t_i should be considered or can be removed from $\mathcal{M}$. We formalize the problem as a linear programming problem LP_i. For any time t_i, we try to maximize the objective function $\sum_{j=1}^{n} h_j(t_i)\, x_j$ taking into account the $m-1$ constraints, $k \neq i$, $\sum_{j=1}^{n} h_j(t_k)\, x_j \leqslant t_k$. We then check if, for time t_i, $\sum_{j=1}^{n} h_j(t_i)\, x_j < t_i$. If this is the case, then adding the constraint $\sum_{j=1}^{n} h_j(t_i)\, x_j \leqslant t_i$ for time t_i will bring the same result, i.e., $\sum_{j=1}^{n} h_j(t_i)\, x_j < t_i$. Hence t_i can then be removed from $\mathcal{M}$. Otherwise, time t_i must be kept, indeed, $\sum_{j=1}^{n} h_j(t_i)\, x_j \geqslant t_i$. The constraint $\sum_{j=1}^{n} h_j(t_i)\, x_j \leqslant t_i$ must be taken into account. We use the *simplex* algorithm to solve, for any time t_i, the maximization problem LP_i. The simplex algorithm must be applied on convex regions. We can therefore apply it step by step on the times of $\mathcal{M}$ provided that the C-space region obtained for any time t_i is convex (we show this property in this section).

1. *C-space Domain Convexity*

In Corollary 1, we show that the C-space feasibility domain $\mathcal{D}^{\mathrm{EDF}}(\tau)$ is the intersection of a finite number of convex regions in $\mathbb{R}^{+n}$ (Lemmas 2 and 3). It is therefore a convex region in $\mathbb{R}^{+n}$. In Corollary 2, we show that it is also a convex polyhedra in $\mathbb{R}^{+n}$.

Let $\mathcal{E}_i \subset \mathbb{R}^{+n}$ be the closed region of $X = (x_1, \ldots, x_n)$ meeting the following property:

$$\frac{1}{t_i} \sum_{j=1}^{n} \mathrm{Max}\left\{0, 1 + \left\lfloor \frac{t_i - D_j}{T_j} \right\rfloor\right\} x_j \leqslant 1$$

Hence, we have

$$\mathcal{E}_i = \left\{ X \in \mathbb{R}^{+n}, \frac{1}{t_i} \sum_{j=1}^{n} \mathrm{Max}\left\{0, 1 + \left\lfloor \frac{t_i - D_j}{T_j} \right\rfloor\right\} x_j \leqslant 1 \right\}, \quad t_i \in \mathcal{M}$$

Lemma 2: Set $\mathcal{E}_i \in \mathbb{R}^{+n}$ is *convex. That is,*

$$\forall (X, X') \in \mathcal{E}_i^2, \quad \forall \lambda \in [0,1], \quad \lambda\, X + (1-\lambda)\, X' \in \mathcal{E}_i$$

Proof: By definition, we have

$$X \in \mathcal{E}_i \quad \Leftrightarrow \quad \sum_{j=1}^{n} \mathrm{Max}\left\{0, 1 + \left\lfloor \frac{t_i - D_j}{T_j} \right\rfloor\right\} x_j \leqslant t_i$$

$$X' \in \mathcal{E}_i \quad \Leftrightarrow \quad \sum_{j=1}^{n} \mathrm{Max}\left\{0, 1 + \left\lfloor \frac{t_i - D_j}{T_j} \right\rfloor\right\} x'_j \leqslant t_i$$

Furthermore, we have $\lambda \in [0,1]$. It follows that

$$\lambda \sum_{j=1}^{n} \mathrm{Max}\left\{0, 1 + \left\lfloor \frac{t_i - D_j}{T_j} \right\rfloor\right\} x_j + (1-\lambda) \sum_{j=1}^{n} \mathrm{Max}\left\{0, 1 + \left\lfloor \frac{t_i - D_j}{T_j} \right\rfloor\right\} x'_j \leqslant t_i$$

Hence, we have

$$\sum_{j=1}^{n} \mathrm{Max}\left\{0, 1 + \left\lfloor \frac{t_i - D_j}{T_j} \right\rfloor\right\} (\lambda\, x_j + (1-\lambda)\, x'_j) \leqslant t_i$$

Finally, we have

$$\lambda\, X + (1-\lambda)\, X' \in \mathcal{E}_i \qquad \blacksquare$$

The C-space region $\mathcal{E} \subset \mathbb{R}^{+n}$ of $X = (x_1, \ldots, x_n)$ meeting the following property:

$$\operatorname*{Lim}_{t \to +\infty} \left\{ \frac{1}{t} \sum_{j=1}^{n} \operatorname{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\} \leqslant 1$$

is denoted as

$$\mathcal{E} = \left\{ X \in \mathbb{R}^{+n},\ \operatorname*{Lim}_{t \to +\infty} \left\{ \frac{1}{t} \sum_{j=1}^{n} \operatorname{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\} \leqslant 1 \right\}$$

Lemma 3: The C-space region $\mathcal{E} \in \mathbb{R}^{+n}$ is *convex. That is,*

$$\forall (X, X') \in \mathcal{E}^2, \quad \forall \lambda \in [0, 1], \quad \lambda\, X + (1-\lambda)\, X' \in \mathcal{E}$$

Proof: By definition, we have

$$\operatorname*{Lim}_{t \to +\infty} \left\{ \frac{1}{t} \sum_{j=1}^{n} \operatorname{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\} = \sum_{j=1}^{n} \frac{x_j}{T_j}$$

Furthermore, we have $X \in \mathcal{E} \Leftrightarrow \sum_{j=1}^{n} x_j / T_j \leqslant 1$ and $X' \in \mathcal{E} \Leftrightarrow \sum_{j=1}^{n} x'_j / T_j \leqslant 1$.
By assumption, we have $\lambda \in [0, 1]$. It follows that $\lambda\ \sum_{j=1}^{n} x_j / T_j + (1-\lambda)\ \sum_{j=1}^{n} x'_j / T_j \leqslant 1$.
Hence, we have

$$\sum_{j=1}^{n} \frac{\lambda\, x_j + (1-\lambda)\, x'_j}{T_j} \leqslant 1$$

Finally, we have

$$\lambda\, X + (1-\lambda)\, X' \in \mathcal{E} \qquad \blacksquare$$

The C-space region $\mathcal{D}^{\mathrm{EDF}}(\tau) \subset \mathbb{R}^{+n}$ of $X = (x_1, \ldots, x_n)$ meeting the following property:

$$\operatorname*{Sup}_{t \in \mathbb{R}^{+*}} \left\{ \frac{1}{t} \sum_{j=1}^{n} \operatorname{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\} \leqslant 1$$

is denoted as

$$\mathcal{D}^{\mathrm{EDF}}(\tau) = \left\{ X \in \mathbb{R}^{+n},\ \operatorname*{Sup}_{t \in \mathbb{R}^{+*}} \left\{ \frac{1}{t} \sum_{j=1}^{n} \operatorname{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\} \leqslant 1 \right\}$$

Corollary 1: The C-space region $\mathcal{D}^{\mathrm{EDF}}(\tau) \in \mathbb{R}^{+n}$ is *convex*. That is,

$$\forall (X, X') \in \mathcal{D}^{\mathrm{EDF}}(\tau) \times \mathcal{D}^{\mathrm{EDF}}(\tau), \quad \forall \lambda \in [0, 1], \quad \lambda\, X + (1 - \lambda)\, X' \in \mathcal{D}^{\mathrm{EDF}}(\tau).$$

Proof: From Theorem 5, we have

$$\begin{aligned}\operatorname*{Sup}_{t \in \mathbb{R}^{+*}} \left\{ \frac{1}{t} \sum_{j=1}^{n} \mathrm{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\} &= \operatorname*{Sup}_{t \in [D_{\min}, P[\cup\{+\infty\}} \left\{ \frac{1}{t} \sum_{j=1}^{n} \mathrm{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\} \\ &= \mathrm{Max} \left\{ \operatorname*{Sup}_{t \in \mathcal{M}} \left\{ \frac{1}{t} \sum_{j=1}^{n} \mathrm{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\}, \sum_{j=1}^{n} \frac{x_j}{T_j} \right\}\end{aligned}$$

By definition, we have

$$\mathcal{D}^{\mathrm{EDF}}(\tau) = \left\{ X \in \mathbb{R}^{+n}, \operatorname*{Sup}_{t \in \mathbb{R}^{+*}} \left\{ \frac{1}{t} \sum_{j=1}^{n} \mathrm{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\} \leqslant 1 \right\}$$

It follows that

$$\mathcal{D}^{\mathrm{EDF}}(\tau) = \left\{ X \in \mathbb{R}^{+n}, \operatorname*{Sup}_{t \in \mathcal{M}} \left\{ \frac{1}{t} \sum_{j=1}^{n} \mathrm{Max} \left\{ 0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor \right\} x_j \right\} \leqslant 1 \wedge \sum_{j=1}^{n} \frac{x_j}{T_j} \leqslant 1 \right\}$$

Therefore, we have

$$\mathcal{D}^{\mathrm{EDF}}(\tau) = \left(\bigcap_{i=1}^{m} \mathcal{E}_i \right) \bigcap \mathcal{E}$$

The intersection of a finite number of convex regions in $\mathbb{R}^{+n}$ is a convex region in $\mathbb{R}^{+n}$. ■

Corollary 2: The C-space feasibility domain $\mathcal{D}^{\mathrm{EDF}}(\tau)$ is the intersection of a finite number of convex and closed regions. It is therefore a *convex polytope* in $\mathbb{R}^{+n}$.

Furthermore, the C-space feasibility domain $\mathcal{D}^{\mathrm{EDF}}(\tau)$ is a closed *convex polytope*. It is therefore a *convex polyhedra* in $\mathbb{R}^{+n}$.

C. Reducing the Number of Constraints Characterizing the C-space

As shown in Sec. IV.B, the set of constraints in $\mathcal{M}$ enables us to characterize the EDF C-space where $\mathcal{M}$ is given by $\mathcal{M} = \bigcup_{j=1}^{n} \{D_j + k_j\, T_j,\ 0 \leqslant k_j \leqslant \lceil (P - D_j)/T_j \rceil - 1\}$.

The number of times in $\mathcal{M}$ can potentially be high. In order to reduce the number of times to consider for the C-space, we propose two methods.

- The first method consists in formalizing the problem of reducing $\mathcal{M}$ as a linear programming problem. As the EDF C-space is convex, we can apply the simplex algorithm to reduce the number of non-pertinent times in $\mathcal{M}$.
- The second method consists in reducing $[D_{\min}, \mathrm{lcm}(T_1, \ldots, T_n))$ to $[D_{\min}, \lambda^{\max})$, where $\lambda^{\max}$ is the maximum busy period valid for any WCET configuration in the C-space. We show that this reduction is possible if a given property is met. We then propose an algorithm to compute $\lambda^{\max}$ by iterations. This significant reduction is confirmed by an example in Sec. V.

1. Linear Programming Problem LP$_i$

We now express as a linear programming problem, the problem of determining if a time t_i in $\mathcal{M}$ should be kept or not. To solve this linear programming problem, we maximize step by step every function $h(t_i) = \sum_{j=1}^{n} h_j(t_i)\, x_j$, representing an objective function under the following linear constraints:

$$\bigcup_{\substack{k=1 \\ k\neq i}}^{m} \{h(t_k) \leqslant t_k\}.$$

The linear programming problem can be expressed by means of a matrix of m lines and n rows where the value at line p and row q is $h_p(t_q)$ except for line $p = i$ where it is equal to 0. We multiply this matrix with a times vector $X = \{x_1, \ldots, x_n\}$ and check if the result is less than a time vector such that any line $j \neq i$, equals to t_j and line i equals to 0 (the constraint for line i is always met). The linear programming problem LP$_i$ associated to time t_i is as follows:

$$(\mathrm{LP}_i)\begin{cases} \text{Maximize } \sum_{j=1}^{n} h_j(t_i)\, x_j, \\ \begin{bmatrix} h_1(t_1) & h_2(t_1) & \cdots & h_n(t_1) \\ \vdots & \vdots & \vdots & \vdots \\ h_1(t_{i-1}) & h_2(t_{i-1}) & \cdots & h_n(t_{i-1}) \\ 0 & 0 & 0 & 0 \\ h_1(t_{i+1}) & h_2(t_{i+1}) & \cdots & h_n(t_{i+1}) \\ \vdots & \vdots & \vdots & \vdots \\ h_1(t_m) & h_2(t_m) & \cdots & h_n(t_m) \end{bmatrix}_{m,n} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}_n \leqslant \begin{pmatrix} t_1 \\ \vdots \\ t_{i-1} \\ 0 \\ t_{i+1} \\ \vdots \\ t_m \end{pmatrix}_m \\ x_1 \geqslant 0, x_2 \geqslant 0, \ldots, x_n \geqslant 0. \end{cases}$$

Linear Programming problem LP$_i$ associated to time t_i.

2. The $\lambda^{\max}$ Approach

Definition 1: $\lambda^{\max}$ is the maximum busy period valid for any WCET configuration under the constraint that the WCETs are in the C-space. Hence $\lambda^{\max}$ is the solution of:

$$\lambda^{\max} = \max_{(x_1,\ldots,x_n)\in C-\text{space}} W(\lambda^{\max}). \tag{1}$$

We now show that if $W(\lambda^{\max})$ meets a given property detailed in Lemma 4 then it is possible to reduce the set of times from $\mathcal{M}$ to the absolute deadlines of the tasks in the synchronous scenario in time interval $[D_{\min}, \lambda^{\max})$.

Lemma 4: Let $\tau = \{\tau_1, \ldots, \tau_n\}$ be a sporadic task set, composed of n sporadic tasks τ_j. Let $\mathcal{M}$ denote the set of points of discontinuity of processor DBF $h(t)$ in time interval $[D_{\min}, P)$, and m denote the cardinal of $\mathcal{M}$.

$$\mathcal{M} = \bigcup_{j=1}^{n} \left\{ D_j + k_j\, T_j, k_j \in \left\{ 0, \ldots, \frac{P}{T_j} - 1 \right\} \right\}$$

If there exists a time $t_c \in [D_{\min}, P)$, a linear combination of the t_i's, i.e., $t_c = \sum_{i=1}^{m} p_i\, t_i$, with $p_i \in \mathbb{N}$ and $t_i \in \mathcal{M}$, such that:

$$W(t_c) = \sum_{j=1}^{n} \left\lceil \frac{t_c}{T_j} \right\rceil C_j = \sum_{i=1}^{m} p_i\, h(t_i)$$

Then, we have

$$\operatorname*{Sup}_{t \in \mathbb{R}^{+*}} \left\{ \frac{h(t)}{t} \right\} = \operatorname*{Sup}_{t \in [D_{\min}, t_c)} \left\{ \frac{h(t)}{t} \right\}$$

Proof: Let us define θ_k, for all $k \in \mathbb{N}$, as follows:

$$\theta_k = \operatorname*{Sup}_{t \in [k t_c, (k+1) t_c), t \neq 0} \left\{ \frac{h(t)}{t} \right\}$$

We have

$$\theta_0 = \operatorname*{Sup}_{t \in [D_{\min}, t_c)} \left\{ \frac{h(t)}{t} \right\}$$

And

$$\forall k \in \mathbb{N}^*, \quad \theta_k = \operatorname*{Sup}_{t \in [k t_c, (k+1) t_c)} \left\{ \frac{h(t)}{t} \right\}$$

By definition, we have

$$\forall t \in \mathbb{R}^+, \quad h(t + t_c) - h(t) \leqslant W(t_c)$$

And

$$W(t_c) = \sum_{j=1}^{n} \left\lceil \frac{t_c}{T_j} \right\rceil C_j = \sum_{i=1}^{m} p_i h(t_i)$$

That is,

$$\forall t \in \mathbb{R}^+, \quad h(t + t_c) \leqslant h(t) + \sum_{i=1}^{m} p_i h(t_i)$$

$$\forall t \in \mathbb{R}^+, \quad \frac{h(t + t_c)}{t + t_c} \leqslant \frac{h(t) + \sum_{i=1}^{m} p_i h(t_i)}{t + \sum_{i=1}^{m} p_i t_i}$$

By definition, we have

$$\theta_{k+1} = \operatorname*{Sup}_{t\in[(k+1)t_c,(k+2)t_c)}\left\{\frac{h(t)}{t}\right\} = \operatorname*{Sup}_{t\in[kt_c,(k+1)t_c)}\left\{\frac{h(t+t_c)}{t+t_c}\right\}$$

$$\theta_{k+1} \leqslant \operatorname*{Sup}_{t\in[kt_c,(k+1)t_c)}\left\{\frac{h(t+t_c)}{t+t_c}\right\} \leqslant \operatorname*{Sup}_{t\in[kt_c,(k+1)t_c)}\left\{\frac{h(t)+\sum_{i=1}^{m} p_i h(t_i)}{t+\sum_{i=1}^{m} p_i t_i}\right\}$$

$$\theta_{k+1} \leqslant \operatorname*{Sup}_{t\in[kt_c,(k+1)t_c)}\left\{\frac{h(t+t_c)}{t+t_c}\right\} \leqslant \operatorname*{Sup}_{t\in[kt_c,(k+1)t_c)}\left\{\frac{\theta_k t+\sum_{i=1}^{m} p_i \theta_k t_i}{t+\sum_{i=1}^{m} p_i t_i}\right\}$$

$$\theta_{k+1} \leqslant \theta_k$$

In other words, we have

$$\theta_k \leqslant \theta_0.$$

Therefore, we have

$$\operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\} = \operatorname*{Max}_{k\in\mathbb{N}}\{\theta_k\} = \theta_0 = \operatorname*{Sup}_{t\in[D_{\min},t_c)}\left\{\frac{h(t)}{t}\right\}$$ ■

In all the experiments we have done, we have been able to find a value of $W(\lambda^{\max})$ meeting Lemma 4 for task sets having constrained deadlines for which $\mathrm{Sup}_{t\in\mathbb{R}^{+*}}\{h(t)/t\} = \mathrm{Sup}_{t\in[D_{\min},P)}\{h(t)/t\}$. We conjecture that such a time always exists for task set having constrained deadlines but we have not yet been able to prove it. Section V shows an example meeting Lemma 4.

We nevertheless show in Lemma 5 that it is always possible to find two absolute deadlines t_k and t_p in the synchronous scenario such that $W(\lambda^{\max}) \leqslant h(t_k) + h(t_p)$.

Lemma 5: Let τ be a sporadic task set, and λ be the length of the first synchronous busy period for τ, with an arbitrary WCETs configuration. Then $\exists t_k$ or t_p, absolute deadlines in the synchronous scenario such that $W(\lambda) \leqslant h(t_k) + h(t_p)$.

Proof: Let t_l be the last absolute deadline in $[0, \lambda]$. Let t_f be the first request time of a task in $[0, \lambda]$ having an absolute deadline after λ. $W(\lambda) = h(t_l) + Q$ where $h(t_l)$ is equal by its definition to the workload of all the tasks having their deadlines in $[D_{\min}, t_l]$ and Q is the workload of the tasks requested in $[t_f, \lambda]$ having their deadlines after λ. We denote τ^* the later task subset of τ. Let $t_{\max}$ be the last deadline of the tasks in τ^*, taken into account in Q. Consider now a scenario starting at time t_f where all the tasks in τ^* having their request time in the time interval $[t_f, \infty)$ are left shifted, to be first released at time t_f. The resulting workload in $[t_f, \lambda]$ cannot decrease. Let $t'_{\max}$ be the new maximum deadline resulting from the transformation. We have $t'_{\max} \leqslant t_{\max}$. The workload obtained for tasks in τ^* is then bounded by the workload of tasks in the synchronous scenario starting at time 0 having their deadlines in $[0, t'_{\max} - t_f]$ equal to $h(t'_{\max} - t_f)$. By their definitions, t_l and $t'_{\max} - t_f$ are absolute deadlines in the synchronous scenario. By setting $t_k = t_l$ and $t_p = t'_{\max} - t_f$, we therefore have $W(\lambda) \leqslant h(t_k) + h(t_p)$. ■

We now propose in Algorithm 1 a function to compute $\lambda^{\max}$. This function considers step by step the ordered set of times t_i in $\mathcal{M} \cup \{P\}$ composed of all the absolute deadlines of the tasks in the synchronous scenario union time $P = \mathrm{lcm}(T_1, \ldots, T_n)$. We propose an iterative computation of $\lambda^{\max}$ starting from the minimum absolute deadline t_1. For any time t_i, we consider the function $N(x_1, \ldots, x_n) = \max(U, h(t_1)/t_1, \ldots, h(t_i)/t_i)$, restriction of the $\mathrm{Sup}(h(t)/t)$ to all times from t_1 to t_i. To reduce the number of possible WCETs, we consider all possible values of the WCETs of task $\tau_1, \ldots, \tau_n$ denoted $x_1, \ldots, x_n$ where each x_i is constrained by $h(D_i) \leqslant D_i$, a constraint that must be met by x_i belonging to the C-space. For any configuration of $x_1, \ldots, x_n$, such that the $\max(U, \sup_{t\in[t_1,\ldots,t_i]}(h(t)/t))$ is less than or equal to 1, we compute the busy period λ solution of $\lambda = W(\lambda)$, by successive iterations as long as either:

- $W(\lambda) = \lambda$. In this case, the iterative equation computing λ has converged and $\lambda \leqslant t_{i+1}$. The busy period λ obtained must be considered for the computation of $\lambda^{\max}$.

- $\lambda > t_{i+1}$. In this case λ exceeds t_{i+1}. The computation of λ can be stopped as t_{i+1} is a constraint that must be considered in the computation of $\mathrm{Sup}(h(t)/t)$. We need to start again the computation of λ to consider times from t_1 to t_{i+1}.

Algorithm 1 Computation of $\lambda^{\max}$.

```
Function Compute-lambda-max($\tau$, $S_i$) : Real
    $\tau$ : Task Set;
    $S_i$ : Set of times in $\mathcal{M} \cup P$; // Ordered set of times in $\mathcal{M} \cup \{P\}$
    temp, $\lambda$, $\lambda^{max}$ : Real;
    $N(x_1, \ldots, x_n)$ : Function;
    // For all times $t_i$ in $\mathcal{M} \cup P$, starting from $t_1$
    For (Each $t_i \in S_i$, starting from $t_1$) do
        // Compute the max between U and the Sup of h(t)/t composed of the subset of times from $t_1$ to $t_i$
        $N(x_1, \ldots, x_n) = \max(U, h(t_1)/t_1, \ldots, h(t_i)/t_i)$; $\lambda^{\max} = 0$;
        // For all possible WCETs, with the constraint $h(t) \leqslant t$,
        for $t \in \{D_1, D_2, \ldots, D_n\}$
        For (All $(x_1, \ldots, x_n)$ with $x_1 \leqslant D_1, x_1 + x_2 \leqslant D_2, \ldots, x_1 + \cdots + x_n \leqslant D_n$) do
            $\lambda = temp = t_i$;
            // If the approximation of $\sup(h(t)/t)$ is less than or equal to 1, we compute $\lambda$
            If ($N(x_1, \ldots, x_n) \leqslant 1$) then
                $temp = W(\lambda)$;
                // As long as we have not converged and not exceeded the next time $t_{i+1}$
                While ($temp \neq \lambda$ AND $\lambda \leqslant t_{i+1}$) do
                    $\lambda = temp$; $temp = W(\lambda)$;
                done
            end If
            // If we have exceeded $t_{i+1}$, we stop and restart for the next time $t_{i+1}$
            If ($\lambda > t_{i+1}$) then
                $\lambda^{\max} = -1$; Exit-For-All;
            end If
            $\lambda^{\max} = \max(\lambda^{\max}, \lambda)$;
        end For
        // if we have converged before the next time in $S_i$, $\lambda^{max}$ has been found
        If ($\lambda^{\max} \leqslant t_{i+1}$ AND $\lambda^{\max} \neq -1$) then
            Exit-For;
        end If
    end For
    return $\lambda^{\max}$
End
```

V. Numerical Applications

In this section, we consider a sporadic task set $\tau = \{\tau_1, \tau_2, \tau_3\}$, composed of three sporadic tasks τ_i, where, for any task τ_i, T_i, and D_i are fixed, and $x_i \in \mathbb{R}^+$, the WCET of task τ_i, is variable.

- $\tau_1 : (x_1, T_1, D_1) = (x_1, 7, 5)$;
- $\tau_2 : (x_2, T_2, D_2) = (x_2, 11, 7)$;
- $\tau_3 : (x_3, T_3, D_3) = (x_3, 13, 10)$.

First, we show how to determine the C-space parametric equations for EDF, using the linear programming approach detailed in Sec. IV.C. We show that the simplex algorithm enables us to remove a very significant number of constraints characterizing the C-space. Then, we compare the C-space obtained with EDF scheduling to the C-space obtained with DM scheduling, following the approach given in Theorem 2. Finally, we study the $\lambda^{\max}$ approach.

In this example, we have: $D_{\min} = 5$ and $P = 1001$. From Theorem 4, we have to consider the set $\mathcal{M}$ of times (absolute deadlines) for the computation of $h(t)/t$ in time interval [5, 1001), where $\mathcal{M}$ is given by:

$$\begin{aligned}\mathcal{M} = &\{5 + 7\,k_1,\ k_1 \in \{0, \ldots, 142\}\}\\ &\cup \{7 + 11\,k_2,\ k_2 \in \{0, \ldots, 90\}\}\\ &\cup \{10 + 13\,k_3,\ k_3 \in \{0, \ldots, 76\}\}\end{aligned}$$

In this example, the cardinal m of $\mathcal{M}$ is equal to 281.

We recall that the C-space feasibility domain $\mathcal{D}^{\mathrm{EDF}}(\tau)$ for EDF is defined by a set of $m + 1$ linear constraints:

$$\mathcal{D}^{\mathrm{EDF}}(\tau) = \left\{ X \in \mathbb{R}^{+n},\quad \operatorname*{Sup}_{t \in \mathcal{M}} \left\{ \frac{1}{t} \sum_{j=1}^{n} \mathrm{Max}\left\{0, 1 + \left\lfloor \frac{t - D_j}{T_j} \right\rfloor\right\} x_j \right\} \leqslant 1 \wedge \sum_{j=1}^{n} \frac{x_j}{T_j} \leqslant 1 \right\}$$

A. Applying the Linear Programming Approach

The simplex algorithm is applied on the Linear Programming problem LP$_i$, for any time $t_i \in \mathcal{M}$, starting from time t_m down to time t_1 (to optimize the computation). We obtain the following subset $\mathcal{S}_1$ of times in $\mathcal{M}$ maximizing $h(t)/t$ for any time vector $X = \{x_1, \ldots, x_n\} \in \mathbb{R}^{+n}$. We solve this problem with the classical simplex algorithm implemented in the Maple 11 computer algebra system.

We obtain the following subset of $\mathcal{M}$:

$$\mathcal{S}_1 = \{5, 7, 10, 12, 19, 40, 62\} \subseteq \mathcal{M}.$$

Therefore, we have

$$\begin{aligned}\operatorname*{Sup}_{t \in \mathcal{M}} \left\{\frac{h(t)}{t}\right\} &= \operatorname*{Sup}_{t \in \mathcal{S}_1} \left\{\frac{h(t)}{t}\right\}\\ &= \mathrm{Max}\left\{\frac{x_1}{5}, \frac{x_1 + x_2}{7}, \frac{x_1 + x_2 + x_3}{10}, \frac{2\,x_1 + x_2 + x_3}{12},\right.\\ &\qquad \left.\frac{3\,x_1 + 2\,x_2 + x_3}{19}, \frac{6\,x_1 + 4\,x_2 + 3\,x_3}{40}, \frac{9\,x_1 + 6\,x_2 + 5\,x_3}{62}\right\}.\end{aligned}$$

Since

$$\begin{cases} x_1 + x_2 \leqslant 7 \\ 2\,x_1 + x_2 + x_3 \leqslant 12 \end{cases} \Longrightarrow 3\,x_1 + 2\,x_2 + x_3 \leqslant 19$$

and

$$\begin{cases} x_1 + x_2 + x_3 \leqslant 10 \\ 2\,x_1 + x_2 + x_3 \leqslant 12 \\ 6\,x_1 + 4\,x_2 + 3\,x_3 \leqslant 40 \end{cases} \Longrightarrow 9\,x_1 + 6\,x_2 + 5\,x_3 \leqslant 62$$

We can still reduce the set of times to check to

$$\mathcal{S}_2 = \{5, 7, 10, 12, 40\} \subseteq \mathcal{S}_1.$$

Hence,

$$\begin{aligned}\operatorname*{Sup}_{t\in\mathcal{M}}\left\{\frac{h(t)}{t}\right\} &= \operatorname*{Sup}_{t\in\mathcal{S}_2}\left\{\frac{h(t)}{t}\right\}\\ &= \text{Max}\left\{\frac{x_1}{5}, \frac{x_1+x_2}{7}, \frac{x_1+x_2+x_3}{10}, \frac{2\,x_1+x_2+x_3}{12}, \frac{6\,x_1+4\,x_2+3\,x_3}{40}\right\}.\end{aligned}$$

We have

$$\begin{aligned}\operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\} &= \text{Max}\left\{U, \operatorname*{Sup}_{t\in\mathcal{M}}\left\{\frac{h(t)}{t}\right\}\right\}\\ &= \text{Max}\left\{\frac{x_1}{7}+\frac{x_2}{11}+\frac{x_3}{13}, \frac{x_1}{5}, \frac{x_1+x_2}{7}, \frac{x_1+x_2+x_3}{10}, \frac{2\,x_1+x_2+x_3}{12}, \frac{6\,x_1+4\,x_2+3\,x_3}{40}\right\}\end{aligned}$$

Since

$$\begin{cases}11\,(x_1+x_2+x_3)\leqslant 110\\ 24\,(2\,x_1+x_2+x_3)\leqslant 288\\ 14\,(6\,x_1+4\,x_2+3\,x_3)\leqslant 560\end{cases}$$

$$\begin{aligned}&\Longrightarrow 143\,x_1+91\,x_2+77\,x_3\leqslant 958\\ &\Longrightarrow 143\,x_1+91\,x_2+77\,x_3\leqslant 1001\\ &\Longrightarrow \frac{x_1}{7}+\frac{x_2}{11}+\frac{x_3}{13}\leqslant 1\\ &\Longrightarrow U\leqslant 1\end{aligned}$$

We have

$$\begin{aligned}\operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t)}{t}\right\} &= \text{Max}\left\{U, \operatorname*{Sup}_{t\in\mathcal{M}}\left\{\frac{h(t)}{t}\right\}\right\}\\ &= \text{Max}\left\{\frac{x_1}{5}, \frac{x_1+x_2}{7}, \frac{x_1+x_2+x_3}{10}, \frac{2\,x_1+x_2+x_3}{12}, \frac{6\,x_1+4\,x_2+3\,x_3}{40}\right\}.\end{aligned}$$

Hence

$$\mathcal{D}^{\text{EDF}}(\tau) = \left\{X\in\mathbb{R}^3,\ \operatorname*{Sup}_{t\in\mathbb{R}^{+*}}\left\{\frac{h(t\,;\,\tau)}{t}\right\}\leqslant 1\right\},$$

That is, the C-space obtained with EDF scheduling is given by:

$$\begin{cases}\dfrac{x_1}{5}\leqslant 1\\ \dfrac{x_1+x_2}{7}\leqslant 1\\ \dfrac{x_1+x_2+x_3}{10}\leqslant 1\\ \dfrac{2x_1+x_2+x_3}{12}\leqslant 1\\ \dfrac{6x_1+4x_2+3x_3}{40}\leqslant 1\end{cases}$$

In our example, the C-space is obtained for x_1, x_2 and x_3 satisfying:

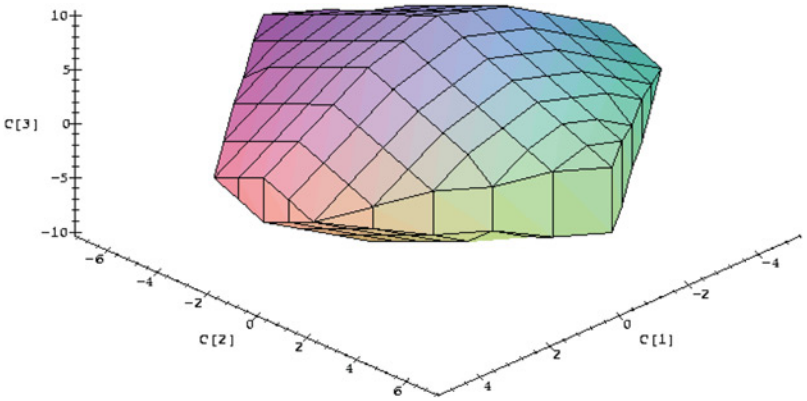

Fig. 3 The C-space feasibility domain $\mathcal{D}^{\mathrm{EDF}}(\tau)$.

- $0 \leqslant |x_1| \leqslant 5$;
- $0 \leqslant |x_2| \leqslant 7 - |x_1|$;
- $0 \leqslant |x_3| \leqslant \mathrm{Min}\left\{10 - |x_1| - |x_2|, 12 - 2\,|x_1| - |x_2|, \frac{40}{3} - 2\,|x_1| - \frac{4}{3}\,|x_2|)\right\}$.

In Fig. 3, we show a graphical representation of the C-space obtained with EDF scheduling.

Now, if we compare the C-space obtained with EDF scheduling to the C-space obtained with DM scheduling, using Theorem 2, we get the set of times $P_{i-1}(D_i)$ to consider for each task τ_i given in Table 1.

We recall that the C-space feasibility domain $\mathcal{D}^{\mathrm{DM}}(\tau)$ for DM is defined by a set of linear constraints:

$$\mathcal{D}^{\mathrm{DM}}(\tau) = \left\{X \in \mathbb{R}^{+^n},\ \underset{j \in \{1,\ldots,n\}}{\mathrm{Max}} \left\{\frac{r_j}{D_j}\right\} \leqslant 1 \wedge \sum_{j=1}^{n} \frac{x_j}{T_j} \leqslant 1\right\}$$

In our example, we have the following conditions:

- $((r_1 = x_1) \wedge (r_1 \leqslant 5))$;
- $((r_2 = x_1 + x_2) \wedge (r_2 \leqslant 7))$;
- $((r_3 = x_1 + x_2 + x_3) \wedge (r_3 \leqslant 7)) \vee ((r_3 = 2x_1 + x_2 + x_3) \wedge (r_3 \leqslant 10))$.

In other words, we have

$$\mathrm{Max}\left\{\frac{x_1}{5}, \frac{x_1 + x_2}{7}, \mathrm{Min}\left\{\frac{x_1 + x_2 + x_3}{7}, \frac{2x_1 + x_2 + x_3}{10}\right\}\right\} \leqslant 1.$$

Let us show that:

$$\mathrm{Max}\left\{\frac{x_1}{5}, \frac{x_1 + x_2}{7}, \mathrm{Min}\left\{\frac{x_1 + x_2 + x_3}{7}, \frac{2x_1 + x_2 + x_3}{10}\right\}\right\} \leqslant 1 \Longrightarrow \frac{x_1}{7} + \frac{x_2}{11} + \frac{x_3}{13} \leqslant 1.$$

Table 1 Task set τ and times to consider for the C-space with DM scheduling

	T_i	D_i	$P_{i-1}(D_i)$
τ_1	7	5	{5}
τ_2	11	7	{7}
τ_3	13	10	{7, 10}

We need to distinguish two cases.
Case 1:

- $x_1 \leqslant 5$;
- $x_1 + x_2 \leqslant 7$;
- $x_1 + x_2 + x_3 \leqslant 7$.

Case 2:

- $x_1 \leqslant 5$;
- $x_1 + x_2 \leqslant 7$;
- $2x_1 + x_2 + x_3 \leqslant 10$.

Case 1: Let us show that $U \leqslant 1$.
Since:

$$\begin{cases} 52\,x_1 \leqslant 260 \\ 14\,(x_1 + x_2) \leqslant 98 \\ 77\,(x_1 + x_2 + x_3) \leqslant 539 \end{cases}$$

$$\begin{aligned} &\Longrightarrow 143\,x_1 + 91\,x_2 + 77\,x_3 \leqslant 897 \\ &\Longrightarrow 143\,x_1 + 91\,x_2 + 77\,x_3 \leqslant 1001 \\ &\Longrightarrow \frac{x_1}{7} + \frac{x_2}{11} + \frac{x_3}{13} \leqslant 1 \\ &\Longrightarrow U \leqslant 1 \end{aligned}$$

Case 2: Let us show that $U \leqslant 1$.
Since

$$\begin{cases} 14\,(x_1 + x_2) \leqslant 98 \\ 77\,(2\,x_1 + x_2 + x_3) \leqslant 770 \end{cases}$$

$$\begin{aligned} &\Longrightarrow 168\,x_1 + 91\,x_2 + 77\,x_3 \leqslant 868 \\ &\Longrightarrow 143\,x_1 + 91\,x_2 + 77\,x_3 \leqslant 1001 \\ &\Longrightarrow \frac{x_1}{7} + \frac{x_2}{11} + \frac{x_3}{13} \leqslant 1 \\ &\Longrightarrow U \leqslant 1 \end{aligned}$$

In both cases, $U \leqslant 1$.
Therefore, we have

$$\mathcal{D}^{\mathrm{DM}}(\tau) = \left\{ X \in \mathbb{R}^{+n},\ \mathrm{Max}\left\{ \frac{x_1}{5}, \frac{x_1 + x_2}{7}, \mathrm{Min}\left\{ \frac{x_1 + x_2 + x_3}{7}, \frac{2x_1 + x_2 + x_3}{10} \right\} \right\} \leqslant 1 \right\}.$$

The C-space feasibility domain $\mathcal{D}^{\mathrm{DM}}(\tau)$ obtained with DM scheduling is given below. All inequalities are in conjunction except the third and the fourth ones which are in disjunction (at least one of the two must be met).

$$\begin{cases} x_1 \leqslant 5 \\ x_1 + x_2 \leqslant 7 \\ \left.\begin{array}{l} x_1 + x_2 + x_3 \leqslant 7 \\ 2x_1 + x_2 + x_3 \leqslant 10 \end{array}\right\} \\ x_1/7 + x_2/11 + x_3/13 \leqslant 1 \end{cases}$$

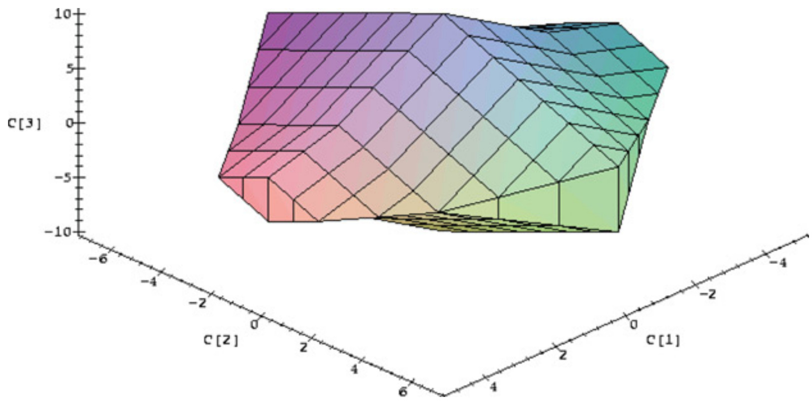

Fig. 4 The C-space feasibility domain $\mathcal{D}^{\text{DM}}(\tau)$.

In our example, the C-space is obtained for x_1, x_2 and x_3 satisfying:
- $0 \leqslant |x_1| \leqslant 5$;
- $0 \leqslant |x_2| \leqslant 7 - |x_1|$;
- $0 \leqslant |x_3| \leqslant \text{Max}\{7 - |x_1| - |x_2|, 10 - 2\,|x_1| - |x_2|)\}$.

In Fig. 4, we show a graphical representation of the C-space obtained with DM scheduling.
The volume of the C-space of EDF is computed as follows:

$$\int_0^5 \int_0^{7-x_1} \int_0^{\min(10-x_1-x_2,\, 12-2x_1-x_2,\, 40/3-2x_1-4x_2/3)} \mathrm{d}x_3\,\mathrm{d}x_2\,\mathrm{d}x_1 = \frac{439}{4}$$

The volume of the C-space of DM is computed as follows:

$$\int_0^5 \int_0^{7-x_1} \int_0^{\max(7-x_1-x_2,\, 10-2x_1-x_2)} \mathrm{d}x_3\,\mathrm{d}x_2\,\mathrm{d}x_1 = \frac{497}{6}$$

Hence, we obtain a volume ratio between the C-space of EDF and the C-space of DM equal to $1317/994 \approx 1.325$.

B. Applying the $\lambda^{\max}$ Approach

Applying Algorithm 1 on task set τ, we find: $\lambda^{\max} = 52$.
We now show that Lemma 4 is valid at time 52 for our example. Let us prove that:

$$\operatorname*{Sup}_{t\in\mathbb{R}^{+*}} \left\{\frac{h(t)}{t}\right\} = \operatorname*{Sup}_{t\in[5,52)} \left\{\frac{h(t)}{t}\right\}.$$

There exists a critical time $t_c = 52 = 12 + 40$, such that:

$$W(52) = \left\lceil\frac{52}{T_1}\right\rceil x_1 + \left\lceil\frac{52}{T_2}\right\rceil x_2 + \left\lceil\frac{52}{T_3}\right\rceil x_3 = \left\lceil\frac{52}{7}\right\rceil x_1 + \left\lceil\frac{52}{11}\right\rceil x_2 + \left\lceil\frac{52}{13}\right\rceil x_3.$$

And

$$W(52) = 8x_1 + 5x_2 + 4x_3 = h(12) + h(40).$$

From Lemma 4, it follows that the constraints characterizing the C-space of EDF are restrained to the absolute deadlines of the tasks in the synchronous scenario, i.e., in $[D_{\min}, \lambda^{\max}) \cap \mathcal{M}$:

$$\operatorname*{Sup}_{t\in\mathbb{R}^{+*}} \left\{\frac{h(t)}{t}\right\} = \operatorname*{Sup}_{t\in[5,52)\cap\mathcal{M}} \left\{\frac{h(t)}{t}\right\}.$$

If Lemma 4 is satisfied, then the best solution to reduce the times in $\mathcal{M}$ is probably to compute first $\lambda^{\max}$ and then to apply the simplex algorithm using the linear programming approach. We end this section with a conjecture that there always exists a $\lambda^{\max}$ satisfying Lemma 4 for task set with constrained deadlines.

VI. Conclusion

In this paper, we have presented new results for a sensitivity analysis of preemptive EDF. We have considered sporadic tasks with independent periods and deadlines. Our goal was to characterize the space of feasible WCETs for EDF, also called the C-space, with parametric equations. We have shown that the C-space can be obtained from an analysis of EDF in a time interval of a duration bounded by the least common multiple of the task periods. From this analysis, we have proposed two approaches to reduce the number of times to consider. One based on a linear programming problem solved with the simplex algorithm. The other based on the computation of the worst-case busy period valid for any WCET in the C-space. The second approach is valid for busy periods meeting a given property. Both approaches can be used together. We have compared the two approaches in an example. We conjecture that the second approach can always be used (i.e., that the given property is always valid) and leave this as an open problem.

References

[1] Baruah, S., Howell, R., and Rosier, L., "Algorithms and Complexity Concerning the Preemptive Scheduling of Periodic Real-Time Tasks on One Processor," *Real-Time Systems*, Vol. 2, 1990, pp. 301–324.
doi: 10.1007/BF01995675

[2] Tindell, K., Burns, A., and Wellings, A. J., "Analysis of Hard Real-Time Communications," *Real-Time Systems*, Vol. 9, 1995, pp. 147–171.
doi: 10.1007/BF01088855

[3] George, L., Rivierre, N., and Spuri, M., "Preemptive and Non-preemptive Real-Time Uniprocessor scheduling," INRIA Research Report No. 2966, Sept. 1996.

[4] Bini, E., Di Natale, M., and Buttazzo, G., "Sensitivity Analysis for Fixed-Priority Real-Time Systems," *Proceedings of the 18th Euromicro Conference on Real-Time Systems* (*ECRTS'06*), Dresden, Germany, 5–7 July 2006.

[5] Balbastre, P., Ripoll, I., and Crespo, A., "Optimal Deadline Assignment for Periodic Real-Time Tasks in Dynamic Priority Systems," *Proceedings of the 18th Euromicro Conference on Real-Time Systems* (*ECRTS '06*), Dresden, Germany, 5–7 July 2006.

[6] Bini, E., and Buttazzo, G., "Schedulability Analysis of Periodic Fixed Priority Systems," *IEEE Transactions on Computers*, Vol. 53, No. 11, Nov. 2004, 12 pages.
doi: 10.1109/TC.2004.103

[7] Hladik, P., Déplanche, A., Faucou, S., and Trinquet, Y., "Adequacy between AUTOSAR OS Specification and Real-Time Scheduling Theory," *IEEE Second International Symposium on Industrial Embedded Systems* (*SIES'07*), Lisbon, Portugal, 4–6 July 2007.

[8] Bollella, G., Brosgol, B., Gosling, J., Dibble, P., Furr, S., and Turnbull, M., *The Real-Time Specification for Java*, 1st ed., Addison Wesley Longman, 2000.

[9] Bougueroua, L., George, L., and Midonnet, S., "Dealing with Execution-Overruns to Improve the Temporal Robustness of Real-Time Systems Scheduled FP and EDF," *The Second International Conference on Systems* (*ICONS'07*), 22–28 Apr. 2007.

[10] Liu, L. C., and Layland, W., "Scheduling Algorithms for Multi-Programming in a Hard Real-Time Environment," *Journal of the ACM*, Vol. 20, No. 1, Jan. 1973, pp. 46–61.
doi: 10.1145/321738.321743

[11] Joseph, M., and Pandya, P., "Finding Response Times in a Real-Time System," *The Computer Journal*, Vol. 29, No. 5, 1986, pp. 390–395.
doi: 10.1093/comjnl/29.5.390

[12] Lehoczky, J. P., Sha, L., Strosnider, J. K., and Tokuda, H., "Fixed Priority Scheduling Theory for Hard Real-Time Systems," *Foundations of Real-Time Computing, Scheduling and Resource Management*, edited by A. M. Tilborg and G. M. Koob, Chap. 1, Kluwer Academic Publishers, 1991.

[13] Manabe, Y., and Aoyagi, S., "A Feasibility Decision Algorithm for Rate Monotonic and Deadline Monotonic Scheduling," *Real-Time Systems*, Vol. 14, No. 2, 1998, pp. 171–181.
doi: 10.1023/A:1007964900035

[14] Lehoczky, J., "Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines," *Proceedings of the 11th IEEE Real-Time Systems Symposium*, Lake Buena Vista, FL, USA, Dec. 1990, pp. 201–209.