

Non-Schedulability Conditions for Off-line Scheduling of Real-Time Systems Subject to Precedence and Strict Periodicity Constraints

Patrick MEUMEU YOMSI
INRIA Rocquencourt
Domaine de Voluceau BP 105
78153 Le Chesnay Cedex - France
Email: patrick.meumeu@inria.fr

Yves SOREL
INRIA Rocquencourt
Domaine de Voluceau BP 105
78153 Le Chesnay Cedex - France
Email: yves.sorel@inria.fr

Abstract

Classical off-line approaches based on preemption such as RM (Rate Monotonic), DM (Deadline Monotonic), EDF (Earliest Deadline First), LLF (Least Laxity First), etc, give schedulability conditions but most of the time assuming on the one hand that all the tasks are independent, and on the other hand, that the first instances of all tasks are released at the same time. We are interested in hard real-time systems subject to precedence and strict periodicity constraints, i.e. such that for all instances of each task, the release time and start time are equal. For such systems, it is mandatory to satisfy these constraints. In this paper we give non-schedulability conditions in order to restrict the study field of all systems of tasks to only potentially schedulable systems.

1. Introduction

We address here hard real-time applications found in the domain of automobile, avionic, mobile robotic, telecommunication, etc, where it is mandatory to satisfy the real-time constraints otherwise dramatic consequences occur [1]. Such applications, based on automatic control and/or signal processing algorithms, are usually specified with block-diagrams. They are composed of functions producing and consuming data, and each function has a strict period. Consequently, in this paper we study the problem of scheduling tasks, onto a single computing resource, i.e. a monoprocessor, each of them corresponding to a function, while satisfying a successor/predecessor relation in addition to their strict periods. Furthermore, since we aim hard real-time, off-line scheduling approaches are better suited because they guarantee a deterministic behavior for the applications.

Classical off-line approaches based on preemption, such as RM (Rate Monotonic) [2], DM (Deadline Monotonic) [3], EDF (Earliest Deadline First) [4], LLF (Least Laxity First) [5], etc, use the Liu & Layland model [6]. This model allows to give schedulability conditions but

assuming on the one hand that all the tasks are independent in that the requests for a certain task do not depend on the initiation or the completion of requests of other tasks, and on the other hand, that the first instances of all tasks are released at the same time. With this model, the release time and the start time of any instance may not be equal for each task during its execution. Actually, because we are in the monoprocessor case, and because we are interested in systems subject to precedence and strict periodicity constraints, two tasks cannot use the processor at the same time. Indeed, strict periodicity constraint imposes for all instances of each task the release time and the start time to be equal. Clearly, this is not the case for the classical model. Furthermore, the start time of any instance of any task should not occur while the processor is occupied by an already scheduled task.

In this paper, we give non-schedulability conditions in order to restrict the study field of all systems of tasks to only potentially schedulable systems. To achieve this goal we use the extension, with preemption, of our previous model [7] that is well suited to the applications we are interested in.

Afterwards, to clearly distinguish the specification level and its associated model we are interested in, we will use the term *operation* instead of the commonly used “*task*” [8] which is too closely related to the implementation level.

The paper is structured as follows: Section 2 describes the model and gives notations used throughout this paper. Section 3 gives non-schedulability conditions to restrict the study field of all systems of tasks to only potentially schedulable systems when seeking a valid schedule thanks on the one hand to properties on the strict periods, and on the other hand to properties on the worst case execution time (WCET) of the tasks. We conclude and propose future works in section 4.

2. Model

The model depicted in figure 1 is an extension, with preemption, of our previous model [7] for systems with

precedence and strict periodicity constraints executed on a single processor.

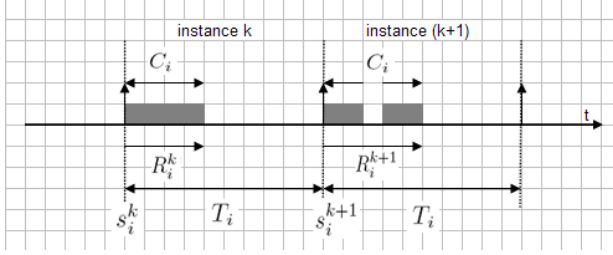


Figure 1. Model

Here are the notations used in this model assuming all temporal values are multiple of the period of the processor clock:

- τ_i : An operation
- C_i : WCET of operation τ_i , $C_i \leq T_i$
- T_i : Period of operation τ_i
- s_i^0 : Start time of the first instance of operation τ_i
- $s_i^k = s_i^0 + (k-1)T_i$: Start time of the k^{th} instance of operation τ_i
- R_i^k : Response time of the k^{th} instance of operation τ_i
- R_i : Worst-case response time of operation τ_i
- $T_i \wedge T_j$: The greatest common divisor of T_i and T_j , when $T_i \wedge T_j = 1$, T_i and T_j are co-prime
- $\tau_i < \tau_j$: $\tau_i \rightarrow \tau_j$, τ_i is a predecessor of τ_j

We assume a single processor, and denote V the set of systems of operations. Each system consists in a given number of operations, subject to precedence and strict periodicity constraints. Each operation τ_i of a system in V consists in a pair (C_i, T_i) : C_i its WCET and T_i its period. The precedence constraints are given by a partial order on the execution of the operations: $\tau_i < \tau_j$ means τ_j cannot start before τ_i is completed. The strict periodicity constraint means that two successive instances of an operation are **exactly** separated by its period: $s_i^{k+1} - s_i^k = T_i \quad \forall k \in \mathbb{N} \quad \forall i \in \{1, \dots, n\}$. In this model the start time is always equal to the release time, contrary to the Liu & Layland's classical model, and for each operation, each instance should be completed before the next one starts.

It is fundamental to note that, because of the strict periodicity constraint and the fact that we are in the mono-processor case, any two instances of two operations of the system cannot start their executions at the same time.

3 Study field restriction

Firstly, we eliminate all the systems such that start times of any two instances of any two operations are identical. This will be achieved thanks only to properties on the strict periods of the operations, using the *Bezout theorem*. This is formally expressed through both theorems given in section 3.1. Secondly, we eliminate all the systems such that the start time of any instance of an operation occurs while the processor is occupied by an already scheduled operation thanks to properties on WCET

of the operations. This is formally expressed through the theorem given in section 3.2. These theorems give non-schedulability conditions in order to restrict the study field of all systems of tasks to only potentially schedulable systems.

3.1 Strict periodicity restriction

Theorem 1

Given a system of n operations in V , if there are two operations $\tau_i = (C_i, T_i)$ and $\tau_j = (C_j, T_j)$ with $(\tau_i < \tau_j)$ starting their executions respectively at the dates s_i^0 and s_j^0 such that

$$T_i \wedge T_j = 1 \quad (1)$$

then the system is not schedulable. Moreover, any additional assumption (for example preemption and idle times) on the system intending to satisfy all the constraints does not have any interest in this case.

Proof

Let $\tau_i = (C_i, T_i)$ and $\tau_j = (C_j, T_j)$ be two operations of a system in V starting their executions respectively at the dates s_i^0 and s_j^0 such that: $T_i \wedge T_j = 1$.

Notice that $s_j^0 \geq s_i^0$ because $\tau_i < \tau_j$.

Let:

$$\gamma = s_j^0 - s_i^0 \quad (2)$$

If $\gamma = 0$, the result is immediate because the first start times of the two operations are then identical. This is not possible because several operations cannot use the processor at the same time.

Let us suppose from now that $\gamma \neq 0$, then:

$$\exists(\alpha, \beta) \in \mathbb{Z}^2 \quad \text{such that} \quad \alpha T_i = \beta T_j + \gamma \quad (3)$$

Indeed, $T_i \wedge T_j = 1$ implies:

$$\exists(m, n) \in \mathbb{Z}^2 \quad \text{such that} \quad m T_i - n T_j = 1 \quad (\text{Cf. Bezout})$$

i.e.,

$$\exists(m, n) \in \mathbb{Z}^2 \quad \text{such that} \quad (m\gamma) \cdot T_i - (n\gamma) \cdot T_j = \gamma \quad (4)$$

Let:

$$\begin{cases} \alpha_0 = (m\gamma) \\ \beta_0 = (n\gamma) \end{cases}$$

It is clear that (α_0, β_0) is a particular solution of (3).

The expressions (3) and (4) give:

$$\begin{cases} (\alpha - \alpha_0) \cdot T_i - (\beta - \beta_0) \cdot T_j = 0 & (*) \\ \alpha T_i - \beta T_j = \gamma & (**) \end{cases}$$

(*) can be rewritten as : $(\alpha - \alpha_0) \cdot T_i = (\beta - \beta_0) \cdot T_j$

Thus we have:

$$\begin{cases} T_i \mid (\beta - \beta_0) \cdot T_j \\ T_i \wedge T_j = 1 \end{cases}$$

Hence $T_i \mid (\beta - \beta_0)$ i.e.,

$$\exists k \in \mathbb{N} \quad \text{such that} \quad \beta - \beta_0 = k T_i \quad (5)$$

Now, replacing $\beta - \beta_0$ by its value in (*), we obtain:

$$\alpha - \alpha_0 = kT_j.$$

Then:

$$(\alpha, \beta) = (\alpha_0 + kT_j, \beta_0 + kT_i) \quad (6)$$

is the general form of all the solutions of (3).

Given any $k \in \mathbb{N}$, at all the following dates $t_{i,j}^k$ the start times of instances of operations τ_i and τ_j are identical:

$$\begin{aligned} t_{i,j}^k &= s_i^0 + (\alpha_0 + kT_j) \cdot T_i \\ &= s_i^0 + (m(s_j^0 - s_i^0) + kT_j) \cdot T_i \\ &= s_j^0 + (\beta_0 + kT_i) \cdot T_j \\ &= s_j^0 + (n(s_j^0 - s_i^0) + kT_i) \cdot T_j \end{aligned}$$

and, thus the strict periodicity constraint is violated. Consequently the system is not schedulable.

Since it is not possible to prevent this situation, any other additional assumption on the system intending to satisfy all the constraints is useless. ▪

N.B.

In the expression (3), the choice of the integers α and β can be made in \mathbb{N} to meet the fact that time is positive.

In fact, if (3) is satisfied and α or β is negative, we just have to consider $h_0 \in \mathbb{N}$ large enough such that:

$$\begin{cases} \alpha_1 = \alpha + h_0T_j \in \mathbb{N} \\ \beta_1 = \beta + h_0T_i \in \mathbb{N} \end{cases}$$

and thus:

$$(3) \iff \alpha_1 T_i - \beta_1 T_j = \gamma$$

In the remainder of this paper, we will no longer care about this detail because it is easy to be solved.

Example 1

Let A and B be two operations of a system in V with the characteristics defined in table 1 below:

Table 1. Characteristics of example 1

	C_i	T_i	s_i^0
A	1	4	0
B	1	7	2

We notice that $4 \wedge 7 = 1$ and conclude thanks to theorem 1 that this system is not schedulable. Indeed, at the date $t = 16$ in figure 2 the two operations A and B start their executions simultaneously. Consequently we cannot schedule them without violating the strict periodicity constraint of one of the two operations.

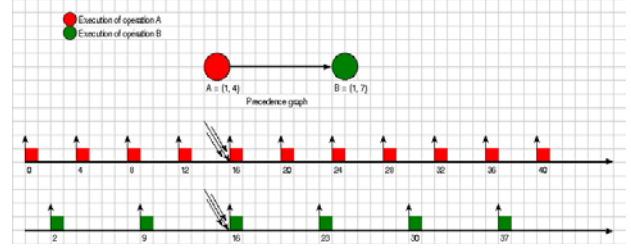


Figure 2. System with 2 operations such that: $T_i \wedge T_j = 1$

Theorem 2

Given a system of n operations in V , if there are two operations $\tau_i = (C_i, T_i)$ and $\tau_j = (C_j, T_j)$ with $(\tau_i \prec \tau_j)$ starting their executions respectively at the dates s_i^0 and s_j^0 such that

$$T_i \wedge T_j \mid (s_j^0 - s_i^0) \quad (7)$$

then the system is not schedulable. Moreover any additional assumption on the system intending to satisfy all the constraints does not have any interest in this case.

Proof

Let $\tau_i = (C_i, T_i)$ and $\tau_j = (C_j, T_j)$ be two operations of a system in V starting their executions respectively at the dates s_i^0 and s_j^0 such that: $T_i \wedge T_j \mid (s_j^0 - s_i^0)$.

Notice once more that $s_j^0 \geq s_i^0$ because $\tau_i \prec \tau_j$.

Let:

$$\gamma = s_j^0 - s_i^0$$

We assume $\gamma \neq 0$: Cf. proof of Theo.1, then:

$$\exists(\alpha, \beta) \in \mathbb{Z}^2 \text{ such that } \alpha T_i = \beta T_j + \gamma \quad (8)$$

Indeed,

let us assume $T_i \wedge T_j = \lambda$ ($\lambda \in \mathbb{N}^*$), then it exists $T_i^1, T_j^2 \in \mathbb{N}^*$ such that:

$$\begin{cases} T_i = \lambda T_i^1 \\ T_j = \lambda T_j^2 \\ T_i^1 \wedge T_j^2 = 1 \end{cases}$$

$T_i^1 \wedge T_j^2 = 1$ implies:

$$\exists(m, n) \in \mathbb{Z}^2 \text{ such that } mT_i^1 - nT_j^2 = 1 \text{ (Cf. Bezout)}. \quad (9)$$

By multiplying each side of (9) by λ , we obtain:

$$\begin{aligned} m \cdot (\lambda T_i^1) - n \cdot (\lambda T_j^2) &= \lambda \\ \text{i.e., } mT_i - nT_j &= \lambda \end{aligned} \quad (10)$$

Thus, (8) and (10) give:

$$(\alpha - m)T_i - (\beta - n)T_j = \gamma - \lambda \quad (11)$$

However, we assumed $\lambda \mid \gamma$, i.e.,

$$\exists p \in \mathbb{N}^* \text{ such that } \gamma = p\lambda$$

We then have:

$$\begin{aligned} (\alpha - m)T_i - (\beta - n)T_j &= (p - 1)\lambda \\ \text{i.e., } (\alpha - m)T_i^1 - (\beta - n)T_j^2 &= p - 1 \end{aligned} \quad (12)$$

By multiplying each side of (9) by $(p - 1)$, we obtain:

$$\exists(m, n) \in \mathbb{Z}^2 \text{ such that } m(p - 1)T_i^1 - n(p - 1)T_j^2 = p - 1 \quad (13)$$

The expressions (12) and (13) give:

$$\exists(m, n) \in \mathbb{Z}^2 \text{ such that } (\alpha - mp)T_i^1 = (\beta - np)T_j^2 \quad (14)$$

In this last relation, we have:

$$\begin{cases} T_i^1 \mid (\beta - np) \cdot T_j^2 \\ T_i^1 \wedge T_j^2 = 1 \end{cases}$$

Thus $T_i^1 \mid (\beta - np)$ i.e.,

$$\exists k \in \mathbb{N}^* \text{ such that } \beta - np = kT_i^1 \quad (15)$$

By replacing this expression in the relation (13), we obtain:

$$\alpha - mp = kT_j^2 \quad (16)$$

Consequently, the general form of the solutions of (8) is given by:

$$(\alpha, \beta) = \left(m \frac{s_j^0 - s_i^0}{T_i \wedge T_j} + k \frac{T_j}{T_i \wedge T_j}, n \frac{s_j^0 - s_i^0}{T_i \wedge T_j} + k \frac{T_i}{T_i \wedge T_j} \right) \quad (17)$$

Thus, given any $k \in \mathbb{N}$, at all the following dates $t_{i,j}^k$ the start times of instances of operations τ_i and τ_j are identical:

$$t_{i,j}^k = s_0^i + [m(s_j^0 - s_i^0) + kT_j] \frac{T_i}{T_i \wedge T_j} \quad (18)$$

$$= s_0^j + [n(s_j^0 - s_i^0) + kT_i] \frac{T_j}{T_i \wedge T_j} \quad (19)$$

and, thus the strict periodicity constraint is violated. Consequently the system is not schedulable.

Since it is not possible to prevent this situation, any other additional assumption on the system intending to satisfy all the constraints is useless.

Example 2

Let A and B be two operations of a system in V with the characteristics defined in table 2:

Table 2. Characteristics of example 2

	C_i	T_i	s_i^0
A	1	4	0
B	1	6	6

We notice that $4 \wedge 6 = 2$ and $2 \mid 6$. We can conclude thanks to theorem 2 that this system is not schedulable.

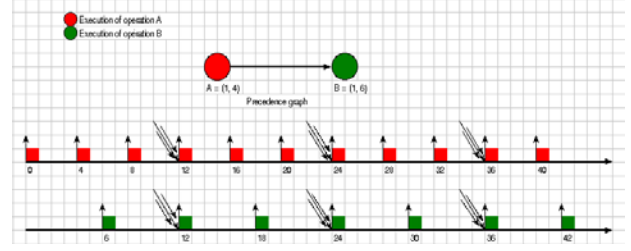


Figure 3. System with 2 operations such that: $T_i \wedge T_j \mid (s_j^0 - s_i^0)$

Indeed, at the dates $t = 12, t = 24, \dots$ in figure 3 the two operations A and B start their executions simultaneously. Consequently we cannot schedule them without violating the strict periodicity constraint of one of the two operations.

Theorems 1 and 2 give conditions of non schedulability for systems subject to strict periodicity constraints when both previous relations on the strict periods hold. Moreover, any additional assumption on the system would be useless because of the identical start times of two instances of at least two operations.

We denote by Ω_λ the sub-set of V excluding the cases where the strict periods of the operations verify both previous relations, see figure 4.

$$\Omega_\lambda = \{ \{(C_i, T_i)\}_{1 \leq i \leq n} \in V / \forall i, j \in \{1, \dots, n\} \exists \lambda > 1, T_i \wedge T_j = \lambda \text{ and } \lambda \nmid (s_j^0 - s_i^0) \}$$

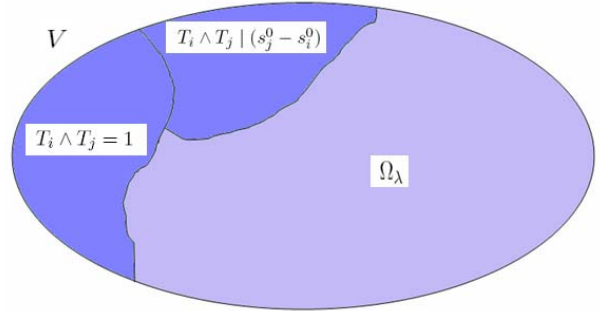


Figure 4. Ω_λ : Restriction of the study field

3.2 WCET restriction

Because of the precedence constraints, it has been shown in [9] that given two operations $\tau_i = (C_i, T_i)$ and $\tau_j = (C_j, T_j)$:

$$\tau_i \prec \tau_j \implies T_i \leq T_j$$

Thus, the operations must be scheduled in an increasing order of their periods.

Now, we assume any operation of the system may only be preempted by those already scheduled [1], [2] relatively to \prec .

We call U_p the p^{th} temporary load factor of the processor ($1 \leq p \leq n$) for a system of n operations $\{\tau_i = (C_i, T_i)\}_{1 \leq i \leq n}$ in Ω_λ .

$$U_p = \sum_{i=1}^p \frac{C_i}{T_i}$$

That system will be said to be *potentially schedulable* if and only if:

$$U_n \leq 1 \quad (20)$$

From now we assume (20) is always satisfied. A *potential schedule* \mathcal{S} of a system is given by a list of the start times of the first instance of all the operations:

$$\mathcal{S} = \{(s_1^0, s_2^0, \dots, s_n^0)\} \quad (21)$$

The start times s_i^k ($k \geq 1, 1 \leq i \leq n$) of the other instances of operation τ_i are directly deduced from the first one. The *response time* R_i^k of the k^{th} instance of operation $\tau_i = (C_i, T_i)$ is the time elapsed between its start time s_i^{k-1} and its end time. This latter takes into account the preemption thus,

$$R_i^k \geq C_i \quad \forall k.$$

We call R_i the *worst response time* of operation τ_i , defined as the maximum of the response times of all its instances.

These definitions enable us to say that, to satisfy the strict periodicity, any operation $\tau_i = (C_i, T_i)$ of a potentially schedulable system in Ω_λ must satisfy:

$$R_i \leq T_i \quad \forall i \in \{1, \dots, n\} \quad (22)$$

We say that a system in Ω_λ has one *overlapping* when the start time of any instance of a given operation occurs while the processor is occupied by an already scheduled operation. Such systems are not schedulable, and we express that through the following theorem.

Theorem 3

Given a system of n operations in Ω_λ , if there are two operations $\tau_i = (C_i, T_i)$ and $\tau_j = (C_j, T_j)$ with $(\tau_i \prec \tau_j)$ starting their executions respectively at the dates s_i^0 and s_j^0 such that

$$\exists \alpha, \beta \in \mathbb{N} \text{ and } 0 \leq (s_j^0 + \beta T_j) - (s_i^0 + \alpha T_i) < R_i^k, \quad k \geq 1 \quad (23)$$

then the system is not schedulable. Moreover any additional assumption on the system intending to satisfy all the constraints does not have any interest in this case.

Proof

The proof of this theorem derives directly from the assumption that an operation may only be preempted by those already scheduled. An example is given below (see figure 5).

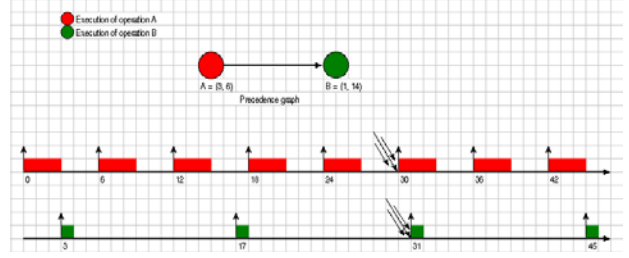


Figure 5. System with an overlapping

Now we can partition Ω_λ in the three following sub-sets: the sub-set V_c of systems with overlappings which are not schedulable thanks to theorem 3, the sub-set V_d of systems with regular operations, i.e. where the periods of all operations constitute a geometric sequence¹, and the sub-set V_i of systems with irregular operations which both remain potentially schedulable (see figure 6).

$$V_c = \{ \{(C_i, T_i)\}_{1 \leq i \leq n} \in \Omega_\lambda / \forall i \in \{1, \dots, n-1\}, \exists j \in \{i+1, \dots, n\} \text{ and } 0 \leq (s_j^0 + \beta T_j) - (s_i^0 + \alpha T_i) < R_i^k, \quad k \geq 1 \}$$

$$V_d = \{ \{(C_i, T_i)\}_{1 \leq i \leq n} \in \Omega_\lambda / T_1 \mid T_2 \mid \dots \mid T_n \}$$

$$V_i = \Omega_\lambda \setminus (V_c \cup V_d)$$

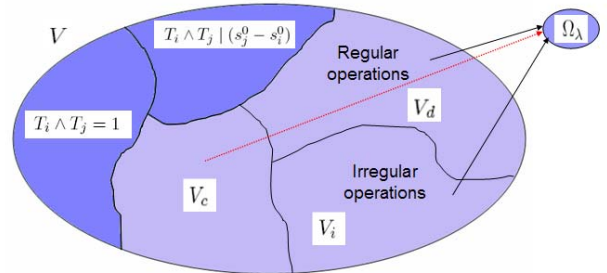


Figure 6. Ω_λ -partitioning

At this point, we have given non schedulability conditions for systems subject to precedence and strict periodicity constraints. Thus, thanks to the partition of V obtained in figure 6, once the previous cases discussed above are excluded, the remaining systems of operations corresponding to $V_d \cup V_i$ are potentially schedulable.

4 Conclusion and future works

We are interested in hard real-time systems subject to precedence and strict periodicity constraints where it is mandatory to satisfy these constraints. We are also interested in preemption which offers great advantages when seeking valid schedules. Because the classical approaches are not well suited for the applications we are interested in, we propose a new approach consisting firstly in building

¹A sequence $(a_i)_{1 \leq i \leq n}$ is geometric if and only if it exists q such that $a_{i+1} = qa_i \quad \forall i \geq 1$.

a set V of all systems of operations, secondly in restricting V by providing non-schedulability conditions thanks to properties on the strict periods and the WCET of all operations. At the end, these non-schedulability conditions lead to the subset $V_d \cup V_i$ of systems with both regular and irregular operations as presented in section 3.2 which consists in potentially schedulable systems of operations. The proposed approach restricts drastically the domain of the possible schedules and will lead to a simpler scheduling analysis.

Presently we are seeking schedulability conditions and valid schedules for systems in $V_d \cup V_i$. Afterwards, we plan to extend our study to the problem with several computing resources.

References

- [1] Joseph Y.-T. Leung and M. L. Merrill. A note on preemptive scheduling of periodic, real-time tasks. *Information Processing Letters*, 1980.
- [2] J.P. Lehoczky, L. Sha, and Y Ding. The rate monotonic scheduling algorithm: exact characterization and average case behavior. *Proceedings of the IEEE Real-Time Systems Symposium*, 1989.
- [3] N.C. Audsley, Burns A., M.F. Richardson, and A.J. Wellings. Hard real-time scheduling : The deadline-monotonic approach. *Proceedings 8th IEEE Workshop on Real-Time Operating Systems and Software*, 1991.
- [4] Jukka Mäki-Turja, Gerhard Fohler, and Kristian Sandström. Towards efficient analysis of interrupts in real-time systems. *11th EUROMICRO Conference on Real-Time Systems*, 1999.
- [5] Patchrawat Uthaisombut. The optimal online algorithms for minimizing maximum lateness. *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory (SWAT)*, 2003.
- [6] C.L. Liu and J.W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 1973.
- [7] L. Cucu, R. Kocik, and Y. Sorel. Real-time scheduling for systems with precedence, periodicity and latency constraints. In *Proceedings of 10th Real-Time Systems Conference, RTS'02*, Paris, France, March 2002.
- [8] J.H.M. Korst, E.H.L. Aarts, and J.K. Lenstra. Scheduling periodic tasks. *INFORMS Journal on Computing* 8, 1996.
- [9] L. Cucu and Y. Sorel. Schedulability condition for systems with precedence and periodicity constraints without preemption. In *Proceedings of 11th Real-Time Systems Conference, RTS'03*, Paris, March 2003.