Characterization and analysis of tasks with offsets: monotonic transactions

Karim Traore[1], Emmanuel Grolleau[2] and francis Cottet[3]
LISI / ENSMA - BP 40109 - 1,avenue Clment ADER
86961 Futuroscope Chasseneuil Cedex
Telephone: 00 335 49 49 80 63
Fax : 00 335 49 49 80 64
Email: karim.traore@ensma.fr[1], grolleau@ensma.fr[2] and cottet@ensma.fr[3]

Abstract

This article introduces the concept of monotonic transactions. A monotonic transaction is a particular case of transactions for which the load arrival pattern is (or can be by rotation) localized at the beginning of the transaction. In the general context of tasks with offsets(general transactions) only exponential methods are known to calculate the worst-case response time. The pseudo-polynomial methods known give an upper bound of the Worst-case response time. The method of analysis suggested in this article gives the real worst-case response time; moreover, this method has a complexity lower than that of the existing methods of approximation. There are two main steps in the application of this method: grouping the tasks of the transaction in a normal form and seeking a monotonic pattern.

1 Introduction

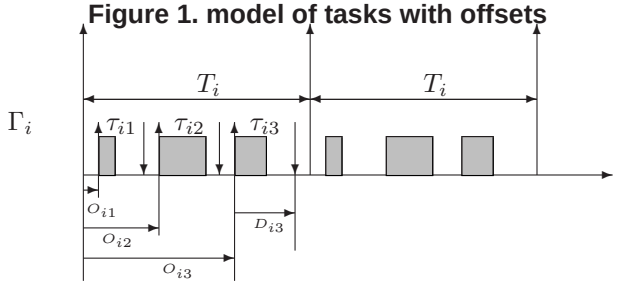
The last step of the development of a hard real-time application consists in modeling the tasks in order to prove the temporal correctness of the application. This validation process consists in proving that, whatever happens, the scheduling policy guarantees that all the temporal constraints are met. In worst-case analysis, the most used task model is an extension of the model of Liu and Layland [1] (methods RMA based). The schedulability conditions obtained with this model are however too pessimistic for certain kinds of pattern of tasks. Thus some articles suggested many other models of tasks:the multiframe model [2] [13], the generalized Multiframe[3], the model of tasks with self-suspension[10] [11] [12], the model of tasks with offsets(transaction) [4] [5] [6] [7]; the models of serial transactions and reverse transactions [9] which appear as particular instances of the model of tasks with offset. Tindell [4] suggested the model of tasks with offsets; Palencia and Harbour[5] extended and formalized the Tindell's work. Then, Turja and Nolin [6] improved the schedulability conditions by introducing the concept of "imposed interference" different from the "released for execution interference" which is the method of calculation of interference inherited from the model of Liu and Layland. In a context of tasks with offsets, all the tasks bound by relations of offsets form a transaction; and in a configuration of tasks, we can have several transactions. In spite of the interest of a lot of researchers for this model, until now, the method of determination of the real worst-case response time remains exponential. For this reason, methods of approximation giving more or less pessimistic schedulability conditions have been proposed. In any case, the concept of approximation leads to the acceptation of some pessimism. This paper is a complementary contribution for analyzing tasks with offsets. We show that, in certain cases, it is possible to propose an exact method of calculation of the real worst-case response time having a lower complexity than the one of all the existing methods of approximation. The structure of the article is as follows: in section 2, we present the model of tasks with offsets. Section 3 presents the normalization process of the transactions. Section 4 presents the concept of monotonic transaction. In section 5, we present an exact method of calculation of the worst-case response time for monotonic transactions. Lastly, section 6 presents the application of the new method.

2 Model of tasks with offsets

2.1 Generality

The model of tasks with offsets was proposed by Tindell [4] in order to reduce existing pessimism of the schedulability analysis where the critical instant for a task occurs when it is released at the same time as all the higher priority tasks. Indeed, certain tasks can for example have the same period and be bound by relations of offsets i.e. they can never be

Figure 1. model of tasks with offsets



released at the same time. A set of tasks of the same period bound by offset is called a transaction. The release of a transaction is bound to an external event(the transactions themselves are offsets free), whose worst-case period of occurrence is the period of the transaction. A task system Γ is compound of a set of transactions Γ_i. [5][6]:

$$\Gamma := \{\Gamma_1, \Gamma_2, .., \Gamma_k\}$$

A transaction (see Figure 1) contains $|\Gamma_i|$ tasks of the same period (with $|E|$ is the cardinal of set E) :

$$\Gamma_i :=< \{\tau_{i1}, \tau_{i2}, ..., \tau_{i|\Gamma_i|}\}, T_i >$$

A task is defined by

$$\tau_{ij} :=< C_{ij}, \Phi_{ij}, D_{ij}, J_{ij}, B_{ij}, P_{ij} >$$

where C_{ij} is the worst-case execution time (WCET), Φ_{ij} is the offset (minimal time between the release of the transaction and the release of the task),D_{ij} is the relative deadline, J_{ij} the maximum jitter (giving t_0 the release date of an instance of the transaction Γ_i , then the task τ_{ij} is released between $t_0+\Phi_{ij}$ and $t_0+\Phi_{ij}+J_{ij}$), B_{ij} maximum blocking due to lower priority tasks, and P_{ij} the priority. It has been shown in [5] that it is equivalent , regarding to the worst-case response time analysis to consider $O_{ij} = \Phi_{ij}\%T_i$. Without loss of generality, we consider that the tasks are ordered by increasing offsets O_{ij} ; in our case, we define the response time as being the time between the release of the task and the completion of this task. Let us note also $hp_i(\tau_{ua})$ the set of indices of the tasks of Γ_i with a priority higher than the priority of a task under analysis τ_{ua} i.e. $j \in hp_i(\tau_{ua})$ if and only if $P_{ij} > P_{ua}$. (assuming that the priorities of the tasks are unique).

In order to validate the system, the Real-Time Analysis (RTA) [14] method is to be applied on each task of the transactions. The task under analysis is usually noted τ_{ua}. Tindell showed that the critical instant of τ_{ua} is a particular instant when it is released at the same time as at least one task of higher priority in each transaction Γ_i. The main difficulty is to determine what is the critical instant candidate τ_{ic} of a transaction Γ_i that initiates the critical instant of τ_{ua}. An exact calculation method would require to evaluate the response time obtained by carrying out all the possible combinations of the tasks of priority higher in each transaction and to choose the task in each transaction that leads to the worst-case response time. This exhaustive method has an exponential complexity and is intractable for realistic task systems; several approximation methods giving an upper bound of the worst-case response time have been proposed. The best known approximation method is the upper bound method based on the "imposed interference".

2.2 Upper bound method based on the "imposed interference"

The "imposed interference" method has been proposed in [6]. This method removes the unnecessary overestimation taken into account in the classic computation of the interference imposed by a task τ_{ij} on a lower priority task τ_{ua}. This overestimation does not have any impact in the case of tasks without offset but has a considerable effect in the approximation of the worst-case response time when we are in the presence of tasks with offsets. This method consists in calculating the interference effectively imposed by a task τ_{ij} on a task τ_{ua} with a lower priority during a time interval of length t; the idea is that the interference cannot exceed the interval of time t. In order to calculate this "imposed interference", [6] substracts a parameter x (see Figure 2) from the original interference formula; let us note $W_{ic}(\tau_{ua}, t)$ the interference that Γ_i imposes effectively on the response time of τ_{ua} during a time interval of length t when τ_{ic} is released at the same instant as τ_{ua} [6]. In a first study of transactions, we will focus on cases with no jitter(i.e $J_{ij} = 0$).

$$W_{ic}(\tau_{ua}, t) = \sum_{j \in hp_i(\tau_{ua})} \left(\left(\left\lfloor \frac{t^*}{T_i} \right\rfloor + 1 \right) * C_{ij} - x_{ijc}(t) \right)$$

$$t^* = t - phase(\tau_{ij}, \tau_{ic})$$

$$phase(\tau_{ij}, \tau_{ic}) = (T_i + (O_{ij} - O_{ic})) \ \% \ T_i$$

$$x_{ijc}(t) = \begin{cases} 0 & \text{for } t^* < 0 \\ \max(0, C_{ij} - (t^*\%T_i)) & \text{otherwise} \end{cases}$$

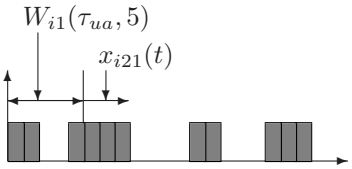
$x_{ijc}(t)$ corresponds to the part of the task τ_{ij} that cannot be executed in the time interval of length t; since this interference is not effectively imposed in this interval, it is not taken into account(See an example on Figure 2).

In order to determine the upper bound of the response-time, [6] uses this function :

$$W_i(\tau_{ua}, t) = \max_{c \in hp_i \tau_{ua}} (W_{ic}(\tau_{ua}, t))$$

With the value of each $W_i(\tau_{ua}, t)$, the upper bound of response-time R_{ua} of τ_{ua} can be calculated: R_{ua} is found

Figure 2. "Imposed interference" method on a transaction of 4 tasks



$\Gamma_i =< \{\tau_{i1}, \tau_{i2}, \tau_{i3}, \tau_{i4}\}, 50 >$
$W_{i1}(\tau_{ua}, 5) = (2-0) + (4-3) + (0-0) + (0-0) = 3$

by iterative fix-point lookup.

$$R_{ua}^0 = C_{ua}$$

$$R_{ua}^{(n+1)} = C_{ua} + \sum_{\Gamma_i \in \Gamma} (W_i(\tau_{ua}, R_{ua}^n))$$

The "imposed interference" method is less pessimistic than the others methods of approximation but its application needs the evaluation of the value of $x_{ijc}(t)$ for each iteration and for each task. Moreover, the application of these methods of approximations for some tasks in a concrete real-time application is sometimes unnecessary. Indeed, in certain cases there is a tractable method for determining the real worst-case response time; this method is less complex than all known approximation methods.

3 Transactions in normal form

Let Γ_i be a transaction and τ_{ua} a task under analysis; without loss of generality, we will consider that all the tasks of Γ_i are higher priority tasks for τ_{ua}. Moreover, we assume that the load of the configuration is less than 1.

3.1 Generalities

Definition : The transaction Γ_i is in normal form if $O_{ij} + C_{ij} < O_{i(j+1)}$ for $1 \leq j < |\Gamma_i|$ and $O_{i|\Gamma_i|} + C_{i|\Gamma_i|} < T_i + O_{i1}$
For example the transactions Γ_1,Γ_2 of Figure 3 and the transaction Γ_i of Figure 4 are in normal form. In opposite, the transaction Γ_i of Figure 7 is not in normal form; indeed, we have for example $O_{i3} + C_{i3} > O_{i4}$.
Let us suppose that there is a task τ_{ij} such as $O_{ij} + C_{ij} \geq O_{i(j+1)}$ in a transaction Γ_i; according to theorem 1 of [4], the busy period starting at O_{ij} contains the busy period starting at $O_{i(j+1)}$. Consequently, the task $\tau_{i(j+1)}$ cannot initiate the critical instant for the task τ_{ua}; therefore it is useless to evaluate $W_{i(j+1)}(\tau_{ua}, t)$ in the process of calculation of the worst-case response time. For this reason, if a transaction Γ_i is not in normal form, we group the tasks of Γ_i in order to obtain a normal form before starting the iterative lookup of the fix-point.

3.2 Grouping in normal form

The method of grouping in normal form is close to the method of merging presented in [8]. Let Γ_i^* be the normal form of Γ_i. Γ_i^* is obtained as follows :
Γ_i^* is first initialized with the value of Γ_i :

$$\Gamma_i^* :=< \{\tau_{i1}^*, \tau_{i2}^*, ..., \tau_{i|\Gamma_i|}^*\}, T_i >$$

$$with \ \tau_{ij}^* = \tau_{ij} \quad for \quad 1 \leq j \leq |\Gamma_i|$$

Process of normalization :

- **Step 1** : for $1 \leq j < |\Gamma_i^*|$, if $O_{ij}^* + C_{ij}^* \geq O_{i(j+1)}^*$ then merge $\tau_{i(j+1)}^*$ into τ_{ij}^*. These two tasks form one task starting at O_{ij}^* with a WCET equal to $C_{ij}^* + C_{i(j+1)}^*$. Renumber the tasks of the transaction in increasing order of O_{ij}^* because $\tau_{i(j+1)}^*$ is deleted
- **Step 2** :
  - if $O_{i|\Gamma_i^*|}^* + C_{i|\Gamma_i^*|}^* \geq T_i + O_{i1}^*$ then merge τ_{i1}^* into $\tau_{i|\Gamma_i^*|}^*$. $C_{i|\Gamma_i^*|}^* = C_{i|\Gamma_i^*|}^* + C_{i1}^*$. Renumber the tasks of the transaction and start again the step 2
  - otherwise it is the end of the process

This process converges if the load of the system is less than 1. The transaction of the figure 8 is the normal form of the transaction of figure 7.

4 Monotonic transactions

4.1 Definition:

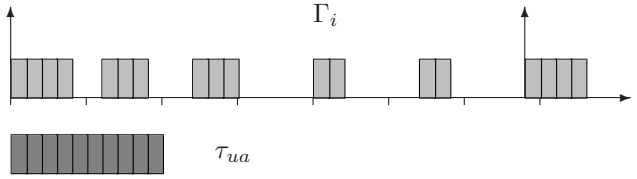
Let $\Gamma_i =< \{\tau_{i1}, \tau_{i2}, ..., \tau_{i|\Gamma_i|}\}, T_i >$ be a transaction and τ_{ua} a task under analysis. Without loss of generality, we consider that all the tasks of Γ_i have a higher priority than the one of τ_{ua}. Let $\Gamma_i^* =< \{\tau_{i1}^*, \tau_{i2}^*, ..., \tau_{i|\Gamma_i^*|}^*\}, T_i >$ be the normal form of the transaction Γ_i. Let us note:

- $\alpha_{ij} = O_{i(j+1)}^* - (O_{ij}^* + C_{ij}^*) \qquad for \ 1 \leq j < |\Gamma_i^*|$
- $\alpha_{i|\Gamma_i^*|} = (T_i + O_{i1}^*) - O_{i|\Gamma_i^*|}$

Note that $\alpha_{ij} > 0$ since Γ_i^* is in normal form.
Γ_i is a monotonic transaction for the task τ_{ua} if the WCET of Γ_i^* have decreasing values while the phases α_{ij} have increasing values i.e:

- $C_{i(p+1)}^* \leq C_{ip}^* \qquad for \ all \ 1 \leq p < |\Gamma_i^*|$
- $\alpha_{ip} \leq \alpha_{i(p+1)} \qquad for \ all \ 1 \leq p < |\Gamma_i^*|$

Figure 3. monotonic transaction



$\Gamma_i =< \tau_{i1}, \tau_{i2}, \tau_{i3}, \tau_{i4}, \tau_{i5}, 34 >$
$\tau_{i1} =< 4, 0, 6, 0, 0, 1 > \ \tau_{i2} =< 3, 6, 5, 0, 0, 2 >$
$\tau_{i3} =< 3, 12, 4, 0, 0, 3 > \tau_{i5} =< 2, 27, 3, 0, 0, 5 >$
$\tau_{ua} =< 0, 10, 30, 0 >$

Example of monotonic transaction : (See Figure 4) in this example, the task τ_{ua} is a lower priority task than all the tasks of Γ_i; moreover, Γ_i is already in normal form: $\Gamma_i = \Gamma_i^*$. we have $C_{i1} \geq C_{i2} \geq C_{i3} \geq C_{i4} \geq C_{i5}$ and $\alpha_{ip} \leq \alpha_{i(p+1)} \quad for\ all\ 1 \leq p < |\Gamma_i^*|$. Therefore, according to the definition of monotonic transaction, Γ_i is monotonic for the task τ_{ua}.

4.2 Looking for monotonic pattern

For the transaction $\Gamma_i^* :=< \{\tau_{i1}^*, \tau_{i2}^*, ..., \tau_{i|\Gamma_i|^*}^*\}, T_i >$, there is no difference, regarding the longest busy period, to consider that:

$$\Gamma_i^* :=< \{\tau_{i2}^*, \tau_{i3}^*, ..., \tau_{i|\Gamma_i|^*}^*, \tau_{i1}^*\}, T_i > \qquad or$$

$$\Gamma_i^* :=< \{\tau_{ik}^*, \tau_{i(k+1)}^*, ..., \tau_{i|\Gamma_i|^*}^*, \tau_{i1}^*, \tau_{i2}^*, ..., \tau_{i(k-1)}^*\}, T_i >$$

We can rotate the tasks of the transaction Γ_i^* without modifying the interference imposed by Γ_i^* on the tasks having a lower priority. For this reason, we consider that Γ_i is monotonic if we can find a monotonic pattern in Γ_i^* by rotating the tasks of Γ_i^*. We know that for a monotonic pattern the first task has the highest WCET. In order to look for a monotonic pattern, we start by inventorying all the tasks with maximum WCET. Then, we consider alternatively each of these tasks τ_{ik}^* as the first task of the transaction Γ_i^* by rotating the tasks of Γ_i^*; and we verify if the conditions of monotony (on C_{ij}^* and α_{ij}) are respected; if so, Γ_i is monotonic and τ_{ik}^* become the first task of Γ_i^*. Then,

$$\Gamma_i^* :=< \{\tau_{ik}^*, \tau_{i(k+1)}^*, ..., \tau_{i|\Gamma_i|^*}^*, \tau_{i1}^*, \tau_{i2}^*, ..., \tau_{i(k-1)}^*\}, T_i >$$

For example in the figure 8, there is a monotonic pattern starting from the task τ_{i2}^*; thus, the transaction Γ_i of the figure 7 is monotonic (the transaction of the figure 8 is its normal form).

5 Presentation of the method for monotonic transaction

In this section, we present the method of calculating the worst-case response time when the transaction Γ_i is monotonic for a task τ_{ua}.

Theorem 1: Let $\Gamma_i =< \{\tau_{i1}, \tau_{i1}, ..., \tau_{i|\Gamma_i|}\}, T_i >$ be a transaction and τ_{ua} a task under analysis. Let Γ_i^* be the normal form of transaction Γ_i. If Γ_i is monotonic for the task τ_{ua}, then the critical instant of τ_{ua} occurs when it is released at the same time as the first task of Γ_i^*.

Proof : To simplify the writings, without loss of generality, we consider that all the tasks of Γ_i have a higher priority than the priority of τ_{ua} and $\Gamma_i = \Gamma_i^*$.
Let τ_{ip} be a task of Γ_i. Let $W_{i1}(\tau_{ua}, t)$ be the interference imposed ("imposed interference") on the task τ_{ua} by the transaction Γ_i in a time interval of length t when τ_{ua} is released at the same time as τ_{i1}. In the same way, $W_{ip}(\tau_{ua}, t)$ is the interference imposed when τ_{ua} is released at the same time as τ_{ip}.
To show that the critical instant of τ_{ua} always coincides with the release of τ_{i1}, we calculate
$W_{i1}(\tau_{ua}, t) - W_{ip}(\tau_{ua}, t)$ and we prove that this value is always ≥ 0 for all $p \in [1..|\Gamma_i|]$ for any time interval of length t.
For $t \geq 0$, we know that there is an integer k such as $t = k * T_i + t\%T_i$. According to theorem 2 of [8], we have:
$W_{i1}(\tau_{ua}, t) = W_{i1}(\tau_{ua}, k * T_i) + W_{i1}(\tau_{ua}, t\%T_i)$ and
$W_{ip}(\tau_{ua}, t) = W_{ip}(\tau_{ua}, k * T_i) + W_{ip}(\tau_{ua}, t\%T_i)$; Moreover, the interference imposed by Γ_i on the task τ_{ua} in a time interval of length T_i is the same whatever the task candidate for the critical instant is. This value is:

$$W_{ip}(\tau_{ua}, T_i) = \sum_{j=1}^{|\Gamma_i|} C_{ij} \ \Rightarrow \ W_{ip}(\tau_{ua}, k \cdot T_i) = k \cdot \sum_{j=1}^{|\Gamma_i|} C_{ij}$$

Consequently,

$$W_{i1}(\tau_{ua}, t) - W_{ip}(\tau_{ua}, t) = W_{i1}(\tau_{ua}, t\%T_i) - W_{ip}(\tau_{ua}, t\%T_i)$$

Thus, we can reduce the problem to $0 \leq t < T_i$.
Moreover, for $t \leq C_{i1}$, we have $W_{i1}(\tau_{ua}, t) = t$; by definition of "imposed interference"

$$W_{ic}(\tau_{ua}, t) \leq W_i(\tau_{ua}, t) \leq t$$

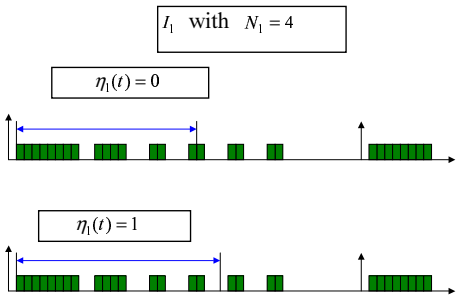
then for $t \leq C_{i1}$,

$$W_{i1}(\tau_{ua}, t) \geq W_{ip}(\tau_{ua}, t)$$

so, we can reduce the problem to:

$$C_{i1} < t < T_i$$

Figure 4. Illustration of the phase and N_1 value



Let us note I_1 the interval from O_{i1} *to* $O_{i1} + t$:

$$I_1 = [O_{i1}; O_{i1} + t]$$

and N_1 the number of tasks activated in the interval I_1 :

$$N_1 = |\{\tau_{ij} \in \Gamma_i \ / O_{ij} < O_{i1} + t\}|$$

Let η_1 be the characteristic function of the interference imposed in the last task of I_1 (see figure 4):

$$\eta_1(t) = \begin{cases} 1 & \text{if } (O_{iN_1} + C_{iN_1}) \leq (O_{i1} + t) \\ 0 & \text{otherwise} \end{cases}$$

η_1 indicates if the time interval of length t goes beyond the last task which is taken into account.

Thus we have :

$$\sum_{j=1}^{N_1-1} C_{ij} < W_{i1}(\tau_{ua}, t) \leq \sum_{j=1}^{N_1} C_{ij} \qquad (1)$$

$$t - \sum_{j=1}^{N_1} \alpha_{ij} \leq W_{i1}(\tau_{ua}, t) \leq t - \sum_{j=1}^{N_1-1} \alpha_{ij} \qquad (2)$$

As for the calculation of $W_{i1}(\tau_{ua}, t)$, let us note I_p the interval from O_{ip} *to* $O_{ip} + t$:

$$I_p = [O_{ip}; O_{ip} + t]$$

and N_p the number of tasks activated in the interval I_p :

$$N_p = \begin{array}{l} |\{\tau_{ij} \in \Gamma_i \ / O_{ip} \leq O_{ij} < O_{i1} + t \ or \\ (O_{ij} < O_{ip} \ and \ O_{ip} + t > O_{ij} + T_i)\}| \end{array}$$

case1: $N_p < N_1$ (See Figure 5)

Let us note $< \tau_{ik_1}, \tau_{ik_2}, ..., \tau_{ik_{N_p}} >$ the tasks activated in the interval I_p by WCET decreasing i.e $C_{ik_1} \geq C_{ik_2} \geq ..... \geq C_{ik_{N_p}}$ Thus, we have :

$$\sum_{j=1}^{N_p-1} C_{ik_j} < W_{ip}(\tau_{ua}, t) \leq \sum_{j=1}^{N_p} C_{ik_j} \qquad (3)$$

Figure 5. $N_1 = 2$ *and* $N_p = 1$ *with* $t = 13$

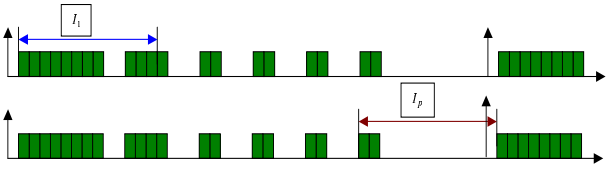
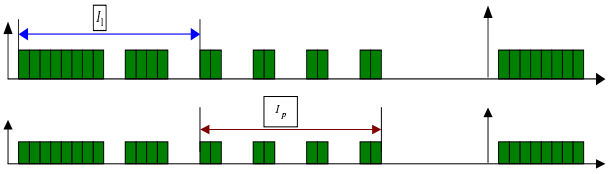


Figure 6. $N_1 = 2$ *and* $N_p = 4$ *with* $t = 17$



Moreover, according to the assumption of the theorem, we have inevitably:
$C_{ik_1} \leq C_{i1}$, $C_{ik_2} \leq C_{i2}$, ..., $C_{ik_{N_p}} \leq C_{iN_p}$ By adding member to member, we obtain

$$\sum_{j=1}^{N_p} C_{ik_j} \leq \sum_{j=1}^{N_p} C_{ij}$$

Since by hypothesis $N_p < N_1$ (Thus $N_p \leq N_1 - 1$), then

$$\sum_{j=1}^{N_p} C_{ik_j} \leq \sum_{j=1}^{N_1-1} C_{ij} \qquad (4)$$

Consequently, according to (1),(3) and (4), we have

$$W_{ip}(\tau_{ua}, t) \leq \sum_{j=1}^{N_p} C_{ik_j} \leq \sum_{j=1}^{N_1-1} C_{ij} \leq W_{i1}(\tau_{ua}, t)$$

case 2: $N_p > N_1$ (See Figure 6)

Let us note $< \tau_{ik_1}, \tau_{ik_2}, ..., \tau_{ik_{N_p}} >$ the tasks activated in the interval I_p by phases increasing i.e $\alpha_{ik_1} \leq \alpha_{ik_2} \leq ..... \leq \alpha_{ik_{N_p}}$ By analogy with inequality (2), we have :

$$t - \sum_{j=1}^{N_p} \alpha_{ik_j} \leq W_{ip}(\tau_{ua}, t) \leq t - \sum_{j=1}^{N_p-1} \alpha_{ik_j} \qquad (5)$$

According to the assumption of the theorem, we have inevitably :
$\alpha_{ik_1} \geq \alpha_{i1}$, $\alpha_{ik_2} \geq \alpha_{i2}$,....,$\alpha_{ik_{N_1}} \geq \alpha_{iN_1}$. By adding

member to member, we obtain

$$\sum_{j=1}^{N_1} \alpha_{ik_j} \geq \sum_{j=1}^{N_1} \alpha_{ij}$$

Since $N_p > N_1$ then

$$\sum_{j=1}^{N_p-1} \alpha_{ik_j} \geq \sum_{j=1}^{N_1} \alpha_{ik_j}$$

from which,we have :

$$t - \sum_{j=1}^{N_p-1} \alpha_{ik_j} \leq t - \sum_{j=1}^{N_1} \alpha_{ik_j} \quad (6)$$

According to (2), (5) and (6) we have:

$$W_{ip}(\tau_{ua}, t) \leq t - \sum_{j=1}^{N_p-1} \alpha_{ik_j} \leq t - \sum_{j=1}^{N_1} \alpha_{ij} \leq W_{i1}(\tau_{ua}, t)$$

case 3:$N_p = N_1$

We have two possibilities : $\eta_1(t) = 0$ or $\eta_1(t) = 1$
If $\eta_1(t) = 0$ then there are as many phases as tasks in the interval I_1. Therefore, the last task taken into account for the calculation of $W_{i1}(\tau_{ua}, t)$ is entirely in the interval I_1. Thus we use the same reasoning as in case 1.
If $\eta_1(t) = 1$ then There are $N_1 - 1$ phases in the interval I_1. Therefore, the task τ_{iN_1} is not inevitably entirely in the interval I_1. We use the same reasoning as in case 2.

6 Applications of the method

In this section we apply the method of monotonic transaction on an example. Let

$$\Gamma_i = \{< \tau_{i1}, \tau_{i2}, \tau_{i3}, \tau_{i4}, \tau_{i5}, \tau_{i6}, \tau_{i7}, \tau_{i8} >, 50\}$$

be a transaction. The tasks of Γ_i are :(Figure 7):
$\tau_{i1} =< 2, 1, 10, 0, 0, 11 >$ $\tau_{i2} =< 5, 9, 10, 0, 0, 12 >$
$\tau_{i3} =< 5, 19, 10, 0, 0, 13 >$ $\tau_{i4} =< 7, 23, 10, 0, 0, 14 >$
$\tau_{i5} =< 1, 34, 10, 0, 0, 15 >$ $\tau_{i6} =< 8, 35, 10, 0, 0, 18 >$
$\tau_{i7} =< 5, 47, 10, 0, 0, 17 >$ $\tau_{i8} =< 1, 48, 10, 0, 0, 18 >$
Let τ_{ua} be a task under analysis with a WCET $C_{ua} = 8$ and a lower priority than all the tasks of Γ_i.

Steps of the application of the method:

Step 1: We group the tasks of Γ_i in order to obtain a normal form and we obtain the transaction of Figure 8:
$\Gamma_i^* = \{< \tau_{i1}^*, \tau_{i2}^*, \tau_{i3}^*, \tau_{i4}^* > 50\}$
$\tau_{i1}^* =< 5, 9, x, 0, 0, x >$ $\tau_{i2}^* =< 12, 19, x, 0, 0, x >$
$\tau_{i3}^* =< 9, 34, x, 0, 0, x >$ $\tau_{i4}^* =< 8, 47, x, 0, 0, x >$
(see Figure 8)
Step 2: Looking for a monotonic pattern We have :

$$C_{i2}^* \geq C_{i3}^* \geq C_{i4}^* \geq C_{i1}^* \quad and \quad \alpha_{i2}^* \leq \alpha_{i3}^* \leq \alpha_{i4}^* \leq \alpha_{i1}^*$$

A monotonic pattern starts from task τ_{i2}^*. Consequently, the critical instant of the task τ_{ua} coincides with the release of the task τ_{i2}^*. We apply the iterative fix-point lookup with the method presented in this article(see Table 1).

Table 1 : New method

Iter #	I_{12}	R_{ua}
0		8
1	12	20
2	21	29
3	29	37
4	29	37

Let us note that $I_{i2}(\tau_{ua}, t)$ is the value obtained with classical RTA method.

$$I_{i2}(\tau_{ua}, t) = \sum_{j=1}^{|\Gamma_i^*|} \left(\left\lceil \frac{t^*}{T_i} \right\rceil \cdot C_{ij} \right)$$

With the new method, it is sufficient to calculate only $I_{i2}(\tau_{ua}, t)$ at each iteration instead of calculating eight values of $W_{ij}(\tau_{ua}, t)$ at each iteration. Moreover, for the calculation of each $W_{ij}(\tau_{ua}, t)$ it is necessary to evaluate $|\Gamma_i|$ times the value of $x_{ijc}(t)$. This evaluation is no longer necessary with the new method. The number of steps in the fix-point lookup is significantly lower. Finally, let us note that RTA analysis is exact. A concrete example of application of monotonic transaction can be found in [9](intermediate priority tasks of a serial transaction).

7 Conclusion

In a general context of tasks with offsets, the RTA methods are intractable because they are exponential in time. This article defines a specific class of tasks with offsets: the monotonic transactions. For this class, we found an exact but simple RTA method which requires less steps (the method is pseudo-polynomial) than the known approximation methods for the general case. This method consists in grouping at first the tasks of the transaction in a normal form. If the normal form presents a monotonic pattern, we showed that the critical instant occurs when the task under analysis is released at the same time as the first task of the pattern; then, we applied the method presented in this article. It is important to note that in a task system, some tasks may be faced to some monotonic transactions, and some transactions which are not. Thus, in order to find the worst-case response time of such a task, the method would con-

Figure 7. Transaction Γ_i

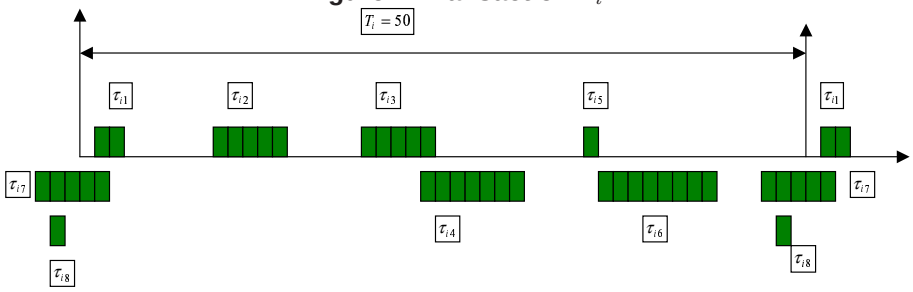
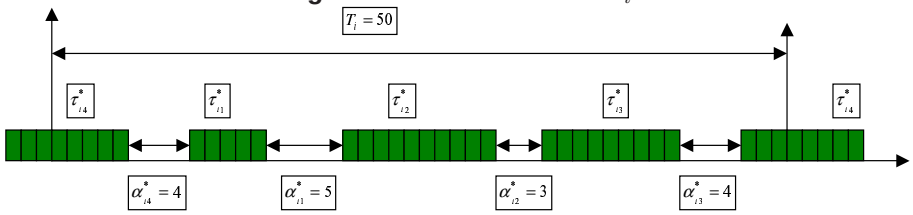


Figure 8. Normal Form of Γ_i



sist in analyzing the transactions looking for monotonic patterns, then consider them as classic tasks in the system, with a constant worst-case interference, and then to use the best approximation method [8] for the non-monotonic transactions. The two methods (ours and [8]) are thus complementary. In our future work on tasks with offsets, we will investigate new classes in order to find less pessimistic schedulability conditions with a lower complexity. Moreover, we will try to extend this method to transactions with jitters.

References

[1] C.L. Liu and J.W. Layland, Scheduling algorithms for mutltiprogramming in real-time environnement, Journal of the ACM 20(1) (1973), 46-61.

[2] A.K.Mok and D.Chen., A Multiframe Model for Real-Time Tasks. In IEEE Real-Time Systems Symposium, pages 22-29, Dec. 1996

[3] S.Baruah,D.Chen,S.Gorinsky and A.Mok, Generalized Multiframe Tasks, The International Journal of Time-Critical Computing Systems,17,5-22 (1999)

[4] K.Tindell, Addind Time-Offsets to Schedulability Analysis, Technical Report YCS 221, Dept of Computer Science, University of York, England, Jan 1994

[5] J.Palencia Gutierrez and M.Gonzalez Harbour. Schedulability Analysis for Tasks with Static and Dynamic Offsets. In Proc. 19th IEEE Real-Time System Symposium (RTSS), December 1998

[6] J.Mäki-Turja and M.Nolin. Tighter Response Time Analysis of Tasks with Offsets. In Proc. 10th International conference on Real-Time Computing and Applications (RTCSA'04), August 2004

[7] J.Mäki-Turja and M.Nolin. Faster Response Time Analysis of Tasks with Offsets. In Proc. 10th IEEE Real-Time Technology and Applications Symposium (RTAS), May 2004

[8] J.Mäki-Turja and M.Nolin. Fast and Tight Response-Times for Tasks with Offsets. In 17th EUROMICRO Conference on Real-Time Systems, p 10, IEEE, Palma de Mallorca Spain, July 2005

[9] K.Traore, E.Grolleau and F.Cottet, Efficient Schedulability Analysis of Serial Transactions, rapport de recherche nr 06001, Laboratoire d'Informatique Scientifique et Industrielle ENSMA, Janvier 2006

[10] F. Ridouard, P. Richard, and F. Cottet. Negative results for scheduling independent hard real-time tasks with self-suspensions. 25th IEEE International Real-Time Systems Symposium (RTSS04), 1, December 2004.

[11] F. Ridouard, P. Richard, and F. Cottet. Scheduling independent tasks wit selfsuspension. Proceedings of the 13rd RTS Embedded Systems (RTS05), 1, April 2005.

[12] Frédéric Ridouard, Pascal Richard, Francis Cottet and Karim Traore, Some results on scheduling tasks with self-suspensions, Journal of Embedded Computing, 2006

[13] C.-C. Han and H.y. Tyan, A Better Polynomial-Time Schedulability Test for Real-Time Fixed-Priority Scheduling Algorithms, Proc. IEEE Real-Time Systems Symp., pp. 36-45, Dec. 1997.

[14] M. Joseph, and P. Pandya, Finding Response Time in a Real-Time System,Computer journal, Vol. 29, No.5, pp.390-395, Oct.1986