On the Convergence of the Holistic Analysis for EDF Distributed Systems

Unai Díaz-de-Cerio[a], Juan P. Uribe[a], Michael González Harbour[b], J. Carlos Palencia[b]

[a]*Software Technologies, Ik4-Ikerlan Research Center, 20500, Mondragón, Spain*
[b]*Software Engineering and Real-Time Group, Universidad de Cantabria, 39005, Santander, Spain*

Abstract

Dynamic scheduling techniques, and EDF (Earliest Deadline First) in particular, have demonstrated their ability to increase the schedulability of real time systems compared to fixed-priority scheduling. In distributed systems, the scheduling policies of the processing nodes tend to be the same as in stand-alone systems and, although few EDF networks exist, it is foreseen that dynamic scheduling will gradually develop into real-time networks. There are some response time analysis techniques for EDF scheduled distributed systems, mostly derived from the holistic analysis developed by Spuri. The convergence of the holistic analysis in context of EDF distributed systems with shared resources had not been studied until now. There is a circular dependency between tasks' release jitter values, response times and preemption level ceilings of shared resources. In this paper we present an extension of Spuri's algorithm and we demonstrate that its iterative formulas are non-decreasing, even in the presence of shared resources. This result enables us to assert that the new algorithm converges towards a solution for the response times of the tasks and messages in a distributed system.

Keywords: Real-Time, Schedulability Analysis, Response Time Analysis, Holistic Analysis, EDF, Distributed Systems

1. Introduction

In the domain of on-line real-time scheduling and analysis there are two main lines of work: fixed and dynamic priority techniques (in both cases we refer to task level priorities). Although fixed priority (FP) scheduling techniques are more used in the industrial field due to their simplicity and predictability under overload conditions, dynamic priority scheduling techniques, and concretely EDF (Liu and Layland, 1973), have been demonstrated to achieve more efficient exploitation of the computation resources (Buttazzo, 2005). This advantage is very important in embedded systems with cost restrictions.

Real-time systems can be deployed on different kind of topologies: single processor, distributed or multiprocessor systems. Multiprocessor system scheduling can be differentiated in two types (Carpenter et al., 2004; Baker, 2005): partitioned or global. In the partitioned systems the tasks are preallocated in a specific processor during the design phase and each processor has its own scheduler. Even though in distributed systems communication overhead is bigger than in partitioned systems, from the point of view of the real-time analysis, a partitioned system and a distributed system where tasks are also statically allocated to processors have similar behaviour. On the contrary, in global systems tasks are dynamically assigned to a processor at execution time by a single scheduler. In distributed and partitioned systems, the scheduling policies of the processing nodes tend to be the same as in single processors and therefore there is a trend towards dynamic priorities, which allow making better use of the available resources. Although few EDF networks exist, it is foreseen that dynamic scheduling will gradually develop into real-time networks.

There are different techniques to analyse the schedulability of a real-time system, i.e., the ability of the system to meet all its timing requirements. These techniques can be classified in different groups: utilization bound tests, demand bound tests and response time analysis among others. Response time analysis techniques calculate upper

Email addresses: udiazcerio@ikerlan.es (Unai Díaz-de-Cerio), jpuribe@ikerlan.es (Juan P. Uribe), mgh@unican.es (Michael González Harbour), palencij@unican.es (J. Carlos Palencia)

bounds on the worst case response times of all the tasks in the systems and stipulate that the system is schedulable if all of them meet their deadlines. Many authors (Audsley et al., 1993; Harter Jr, 1987; Joseph and Pandya, 1986; Palencia et al., 1997, 1998; Palencia and González Harbour, 2003, 2005, 1998, 1999; Pellizzoni and Lipari, 2007; Spuri, 1996a,b; Tindell and Clark, 1994; Tindell et al., 1994b) have published different response time analysis techniques for different system models and scheduling policies. The results of response time analysis provide a very intuitive way of reasoning about the worst-case that can happen in a real-time system, allowing us to know whether the system is more or less loaded and where the bottlenecks are. Since these techniques obtain as a result worst-case response time of all the tasks, system analysts can know which tasks are overloaded or near to be overloaded. This is a big advantage over other techniques, such as some utilization bound tests, which just provide a yes/no answer to the schedulability of the system.

This work focuses on response time analysis for distributed systems and partitioned multiprocessor/multi-core systems. In these systems the increasing need to make better use of the available resources is causing a shift towards the use of dynamic priority scheduling policies, EDF in particular. Most work on the response time analysis for EDF systems is based on Spuri's analysis techniques (Spuri, 1996a,b). The first of these techniques (Spuri, 1996a) is focused on EDF scheduling over single processors. The second technique (Spuri, 1996b), extends the response time analysis to distributed or partitioned multi-processor systems, using the holistic analysis approach by Tindell et al (Tindell and Clark, 1994). Holistic analysis calculates the response times of the tasks with an iterative process that bases its convergence on the non-decreasing nature of the iterative equations. There is a circular dependence of the response times of tasks in one resource with the response times in the other resources through the task's input jitter effect. The set of input jitters of the tasks, calculated from the response times obtained in the previous iteration, are the inputs for each iteration. If an increase of a task jitter would result in a decrease of the response time of this task or another task, the convergence of the iterative method would be jeopardized. Spuri's work (Spuri, 1996b) was published as a technical report and does not include a demonstration of the correctness of its convergence. The first contribution of this paper is to demonstrate that an increase of the input jitter of a task cannot lead to decreasing the response times, and thus proves the correctness of the method.

Real-time applications usually need to share resources among their tasks and therefore a resource sharing protocol is needed. Most resource sharing protocols rely on mutual exclusion for consistency and use different techniques to provide short predictable access times. In the case of single processor systems, we find, among others, non-preemptive sections (Mok, 1983), Priority Inheritance Protocol (PIP) (Sha et al., 1990), Priority Ceiling Protocol (PCP) (Sha et al., 1990; Chen and Lin, 1990) and its simpler emulation called Immediate Priority Ceiling (IPC) (Lampson and Redell, 1980), Stack Resource Policy (SRP) (Baker, 1990, 1991), and Deadline-Floor Inheritance Protocol (DFP) (Gutierrez et al., 2014). These protocols are also usable in partitioned multiprocessors and distributed systems when the resource is local, i.e., it is only shared among tasks of the same processing node. For global resources shared among tasks allocated to different processing nodes it is necessary to consider protocols that are capable of avoiding remote blocking (Rajkumar et al., 1988). For this purpose, it is possible to increase the priority ceilings in the case of PCP (Rajkumar et al., 1988) and IPC or, similarly, the preemption levels for SRP, avoiding this remote blocking effect and keeping the access time to the shared resources short and predictable. For global resources in multiprocessors under global scheduling new resource sharing protocols are required (Davis and Burns, 2011).

Spuri (Spuri, 1996a) based its single-processor EDF analysis of tasks with local resource sharing on the PCP and SRP protocols. In the case of distributed or partitioned multiprocessor systems there is a problem that is not mentioned in Spuri's extended technique. The preemption levels on which the SRP technique is based should be assigned as a function of the deadline minus the input jitter (Baker, 1991). The ceilings in shared resources scheduled with dynamic PCP also depend on jitter. However, as the estimations of maximum jitter terms increase during the iterative analysis (with the successive increases of response times) there is the possibility that preemption levels might need to change. Since response times depend on preemption levels and jitter depends on response times, there is a new circular dependency in the analysis in the presence of shared resources that could jeopardize the convergence of the method. The second contribution of this paper is the study of this effect, the proposal of an extension to Spuri's algorithm dealing with the calculation of preemption levels, and the demonstration of the convergence conditions of the new holistic algorithm for EDF in systems with shared resources.

Although there are techniques that increase the schedulability of the distributed system scheduled by FP (Gutiérrez and González Harbour, 1996; Sun and Liu, 1996) or EDF (Serreli et al., 2010), by reducing or eliminating jitter of the system, they need the modification of the usual scheduling rules. This can be difficult to implement in commercial operating systems and even more in standard network protocols. Consequently, in this paper we will use the technique with jitters.

Response-time analysis techniques in distributed and partitioned systems are not only applicable to systems with a monolithic scheduling policy, but can also be used in heterogeneous systems in which there is a mixture of different scheduling policies (Rivas et al., 2011). For instance, a plausible implementation might use EDF in the processing nodes and a CAN bus under fixed priorities as a communications network. The results of this paper are, therefore, of direct applicability in actual distributed systems, as we will show through an example based on an industrial system.

The paper is organized as follows. In Section 2 we review the analysis technique derived by Spuri for periodic tasks under EDF. Then, in Section 3 we demonstrate the correctness of Spuri's technique in relation to the convergence of the iterative method. In Section 4 we present an example of an industrial system to which we have applied Spuri's method in order to calculate the response times of the tasks of the system. Finally, in Section 5 we give our conclusions.

2. Spuri's Analysis

2.1. System Model

The system considered in this analysis is a partitioned multiprocessor or distributed system (as mentioned before both have similar behaviour from the point of view of the analysis) composed by n tasks in k processors. The tasks are statically allocated and can communicate with other tasks in the same or other processor making up an end-to-end flow. As can be seen in Fig. 1, an end-to-end (e2e) flow is a sequence of tasks and messages that are executed one after the other. Each e2e flow, and therefore its tasks and messages, is composed by an infinite number of jobs or instances that are released periodically or sporadically. The nominal release times of the jobs of the first task of e2e flow i are separated by a minimum time or period T_i. For intermediate tasks and messages in the e2e flow the release time is the completion time of the previous task or message. This release time can suffer a variation from job to job that is called release jitter J_i. The maximum release jitter that a task/message τ_i can suffer is propagated from the response time of the previous task/message $\tau_{i-1}$and is calculated subtracting the best case response time to the worst case response time ($J_i = R_{i-1} - R^b_{i-1}$). For notational convenience, we do not distinguish between tasks and messages in the analysis, and we treat messages in EDF networks as if they were tasks, making the transmission and execution time concepts equivalent.

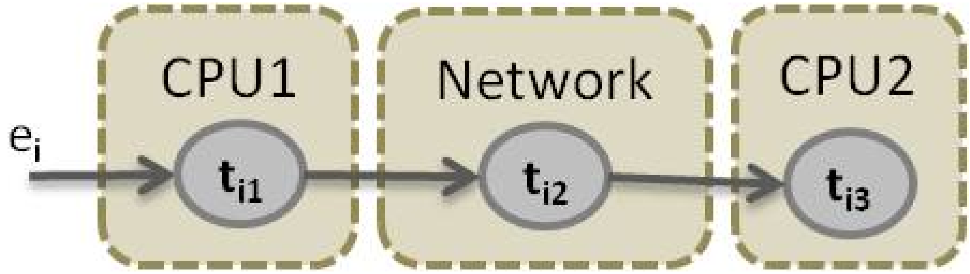


Figure 1: End-to-end flow diagram.

Each task will execute a bounded time of computation, with the bound called worst case execution time, C_i. This computation should be completed by a relative deadline D_i after its best-case arrival time. The ready task queue will be ordered by the EDF (Liu and Layland, 1973) preemptive dispatching algorithm.

Tasks may locally share resources, locking and unlocking critical sections CS_i according to a protocol like the Stack Resource Policy (Baker, 1991) or Priority Ceiling (Sha et al., 1990). Those critical sections generate blocking time B_j in other tasks.

The network response time analysis depends on the protocol used in the network. Spuri in (Spuri, 1996b) proposed a Time Token MAC protocol for the network messages, but any other protocol that allows response time analysis can be used. The results of this analysis can be added to the holistic analysis.

The target of the analysis is to obtain the worst-case response time r_i of the tasks (or an upper bound of the response times).

2.2. Response-time Analysis

Spuri in (Spuri, 1996b) proposed a holistic analysis for deadline scheduled real time distributed systems. This holistic analysis is based on non-decreasing iterative equations which decouple the analysis of the system in k different subsystems corresponding to each processing node or communications network. We now summarize Spuri's technique.

$$\begin{cases} R_1^{(m+1)} = R_{edf}\left(J_1^{(m)}\right) \\ \dots \\ R_k^{(m+1)} = R_{edf}\left(J_k^{(m)}\right) \end{cases} \quad \begin{cases} J_1^{(m+1)} = P_1\left(R_1^{(m+1)}, ..., R_k^{(m+1)}\right) \\ \dots \\ J_k^{(m+1)} = P_k\left(R_1^{(m+1)}, ..., R_k^{(m+1)}\right) \end{cases} \tag{1}$$

where m is the iteration index, R_{edf} is the response time for EDF obtained for one node in one iteration, J_i^m is the vector of release jitters of the tasks in node i in iteration m, R_i^m is the vector of the response times of the tasks in node i obtained in iteration m, and P_i is the function that obtains the release jitters from the response times in the predecessor tasks: for each task i, $J_i = R_{i-1} - R_{i-1}^b$, where task τ_{i-1} is the predecessor task of τ_i in the e2e flow. The first task in the e2e flow has no predecessor and thus its release jitter is zero.

As we can see, the response time of the tasks is calculated with the jitter got from the previous iteration. For the first iteration, release jitter values are assumed to be zero. With the holistic algorithm, each subsystem can be analysed independently using the single-processor EDF response time algorithm developed by Spuri in (Spuri, 1996a) and summarized in (Spuri, 1996b). In this paper, we are only concerned about the analysis of EDF resources of the system. If there are networks or any other resources scheduled with another protocol different from EDF, the analysis of heterogeneous systems introduced in (Rivas et al., 2011) could be used. This technique allows analysing each processor separately with the appropriate technique for the scheduling policy used in each processor and then propagating the results to the tasks in other processors through the jitter.

For the single-processor EDF response time analysis, according to Lemma 4.1 in (Spuri, 1996a) it is sufficient to study the schedule of the most demanding arrival pattern in the first busy period. The length L of the busy period can be computed by means of a simple iterative formula:

$$\begin{cases} L^{(0)} = \sum_{i=1}^{n} C_i \\ L^{(m+1)} = W(L^{(m)}), \end{cases} \tag{2}$$

where $W(t)$ is the cumulative workload at time t, i.e. the sum of the computation times of the task instances arrived at or before time t:

$$W(t) = \sum_{i=1}^{n} \left\lceil \frac{t + J_i}{T_i} \right\rceil C_i \tag{3}$$

The worst-case response time computation is also based on Lemma 4.1 in (Spuri, 1996a) that says that the worst-case response time of a task i is found in a busy period in which all other tasks are released synchronously at the beginning of the period and then at their maximum rate and at the earliest possible time inside the busy period, according to their maximum release jitter. Notice that the task under analysis is not necessarily released at the start of the busy period. We set $t = 0$ at the start of the busy period, and we call a the time at which a job of the task under analysis that is inside the busy period is released. The next formula calculates the interaction caused by all the other tasks at time t on the job of task i released at a. Notice that only the tasks with an absolute deadline earlier that that of the task job under analysis can cause interference.

$$W_i(a,t) = \sum_{j \neq i, D_j \leq a + D_i + J_i} \min\left\{ \left\lceil \frac{t + J_i}{T_j} \right\rceil, 1 + \left\lfloor \frac{a + D_i + J_j - D_j}{T_j} \right\rfloor \right\} C_j \tag{4}$$

Then, the length $L_i(a)$ of the resulting busy period related to the activation at a can be computed with the next iterative formula that takes into account the interaction from other task jobs with earlier deadlines, the own execution of the jobs of task i that fall inside the busy period starting at or before a, and $B_{k(a+D_i)}$ which is the worst blocking time caused on the task jobs inside the busy period by sharing resources with other task jobs with larger deadlines, i.e., $D_k - J_k \leq a + D_i$:

$$\begin{cases} L_i^{(0)}(a) & = \sum_{j \neq i} C_j, \\ L_i^{(m+1)}(a) & = W_i\left(a, L_i^{(m)}(a)\right) + \left(1 + \left\lfloor \frac{a+J_i}{T_i} \right\rfloor\right) C_i + B_{k(a+D_i)}. \end{cases} \tag{5}$$

According to the SRP and PCP protocols, the blocking time B_i that a task τ_i can suffer is limited to a unique critical section cs. Hence, blocking is calculated with the maximum of the critical sections cs of the tasks that have a preemption level π_j lower than or equal to the task τ_i's preemption level π_i and the ceiling of the shared resource $\lceil s \rceil$ is bigger than or equal to the preemption level π_i:

$$B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \; and \; \pi_j \leq \pi_i\right\} \tag{6}$$

where the preemption order is assigned ordered by effective deadline (defined as the difference between the relative deadline and the release jitter: $D_i - J_i$) and the ceiling of the shared resources s is the maximum of the preemption levels of the tasks (π_i) that may lock that resource:

$$\begin{gathered} \pi_i > \pi_j \Longleftrightarrow D_i - J_i < D_j - J_j \\ \lceil s \rceil = \max\{\pi_i : i \; may \; lock \; s\} \end{gathered} \tag{7}$$

Once the value of $L_i(a)$ has been determined, the worst-case response time relative to a is:

$$r_i(a) = \max\{J_i + C_i + B_i, L_i(a) - a\} \tag{8}$$

Finally, according to Lemma 4.1 in (Spuri, 1996a), the worst-case response time of task i is:

$$r_i = \max_{a \geq -J_i}\{r_i(a)\} \tag{9}$$

Fortunately, there is no need to calculate $r_i(a)$ for all $a \geq -J_i$. L, the length of the first busy period, is the maximum of the lengths of all busy periods and thus the significant values of a are in the interval $[-J_i, L - J_i - C_i - B_i]$. Moreover, we only need to check the values of a that make a job of task i start at $t = 0$ or those values that make the absolute deadline of the analysed job of task i coincide with the absolute deadline of another task.

3. Demonstration of Spuri's Analysis Convergence

We will now demonstrate the convergence of Spuri's iterative method to calculate an upper bound on the worst-case response times of EDF tasks in a distributed real time system. Spuri already discussed the convergence of this iterative method based on the fact that its iterative equations were non-decreasing: as the iteration progresses, estimations on worst-case response times increase, and so do the estimated release jitters. However, this claim was not formally proved. Moreover, in systems with shared resources there is a new circular dependence among release jitters and preemption levels or priority ceilings. The convergence of the holistic analysis for EDF systems also needs to be proven in the presence of shared resources. In this section we will first discuss the convergence conditions of the EDF holistic analysis in a system without resource sharing. Then, we will do the same for systems with shared resources.

3.1. System without Resource Sharing

In this subsection, we will demonstrate the correctness of the analysis method for systems without shared resources. To simplify the presentation and provide some intuition on the proof mechanism, first we will analyse a system with three tasks and then, we will generalise it to a system with n tasks.

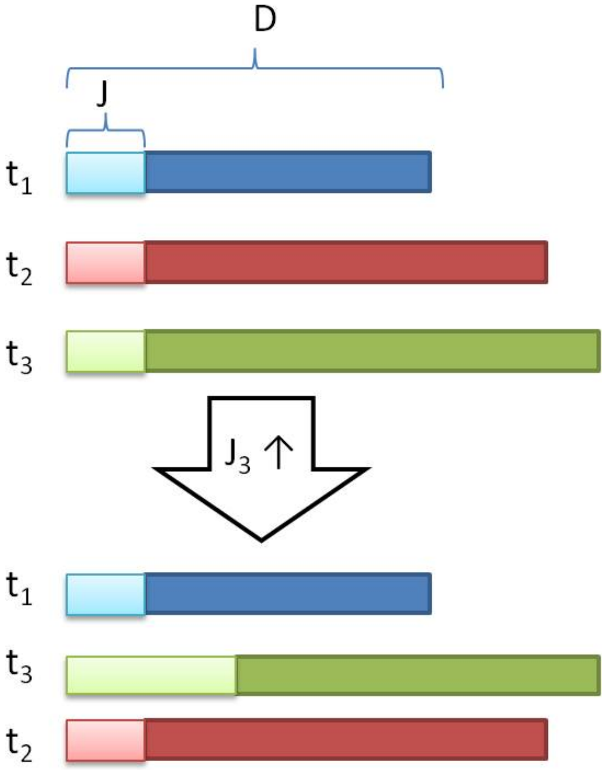


Figure 2: Preemption order change because of jitter increase.

3.1.1. Three Tasks System

In this subsection, we will assume three tasks without shared resources. The tasks are numbered in increasing effective deadline, so $D_1 - J_1 \leq D_2 - J_2 \leq D_3 - J_3$. To symplify the demonstration, first, we will assume infinite periods for the tasks. We will demonstrate that the response time of these tasks cannot decrease because of the increase of the release jitter. To that end, we will analyse the response time of one task in two scenarios. The first scenario corresponds to the situation before increasing the jitter. Then, we will create the second scenario by increasing the release jitter of task τ_3, up to a value J'_3, so that its effective deadline lies between tasks 1 and 2: $D_1 - J_1 \leq D_3 - J'_3 \leq D_2 - J_2$ (see Fig. 2). This change to the effective priority would apparently make τ_3's response time smaller, but we will see that this is not true. To this end we will analyse the response time of the same task with the new value of jitter. Finally we will demonstrate that the response time in the second scenario can never be lower than in the first one. We have to repeat this process for the three tasks.

Fig. 2 allows us to visually compare the intervals of release jitter (light-shaded boxes) and scheduling deadline (dark-shaded boxes). Initially we have chosen an infinite period to simplify the analysis. In this way, each task will only have one job to be analysed. Later, in subsection 3.3 we will analyse what happens when the periods are smaller.

Lemma 3.1. *Let τ be a group of three tasks with infinite periods and without shared resources, scheduled under EDF. The response time of any of the tasks in τ cannot decrease by the effect of increasing the release jitter of any of the tasks in τ.*

Proof. In the first analysis scenario, before increasing the release jitter, we have the following task attributes:

$$\begin{aligned} &\tau_1 : C_1, D_1, J_1, T_1 = \infty, \\ &\tau_2 : C_2, D_2, J_2, T_2 = \infty, \\ &\tau_3 : C_3, D_3, J_3, T_3 = \infty, \\ &\textit{where } D_3 - J_3 \geq D_2 - J_2 \geq D_1 - J_1 \end{aligned} \tag{10}$$

For the second analysis scenario we will increase the release jitter of task τ_3 (see Fig. 2) and the task attributes will be:

$$\begin{aligned} &\tau_1 : C_1, D_1, J_1, T_1 = \infty, \\ &\tau_2 : C_2, D_2, J_2, T_2 = \infty, \\ &\tau_3 : C_3, D_3, J_3', T_3 = \infty, \\ &\textit{where } D_2 - J_2 \geq D_3 - J_3' \geq D_1 - J_1 \end{aligned} \tag{11}$$

We start analysing task τ_1, then we continue with τ_2 and finish with τ_3. According to Spuri's analysis we need to analyse two different situations, corresponding to different activation times in the first busy period. In the first situation all the tasks are activated synchronously and with their maximum jitter ($a = -J_i$, see Fig. 3a). The second situation is created when the deadline of τ_1 coincides with some other task's deadline ($a = D_2 - J_2 - D_1$ and $a = D_3 - J_3 - D_1$, see Fig. 3b and Fig. 3c, respectively).

If we apply Spuri's analysis to these situations we obtain the response time of τ_1:

$$r_1(-J_1) = C_1 - (-J_1) \tag{12}$$

$$r_1(D_2 - J_2 - D_1) = C_2 + C_1 - (D_2 - J_2 - D_1) \tag{13}$$

$$r_1(D_3 - J_3 - D_1) = C_3 + C_2 + C_1 - (D_3 - J_3 - D_1) \tag{14}$$

Consequently, the worst case response time of the task τ_1 is:

$$\begin{aligned} r_1 &= \max\{r_1(-J_1), r_1(D_2 - J_2 - D_1), r_1(D_3 - J_3 - D_1)\} \\ &= \max\{C_1 - (-J_1), C_2 + C_1 - (D_2 - J_2 - D_1), C_3 + C_2 + C_1 - (D_3 - J_3 - D_1)\} \end{aligned} \tag{15}$$

After increasing the release jitter of task τ_3 we need to calculate the response time for three activation times ($a = -J_1$, $a = D_2 - J_2 - D_1$ and $a = D_3 - J_3 - D_1$, see Fig. 4a, Fig. 4b and Fig. 4c, respectively). In this situation, we get these response times:

$$r_1'(-J_1) = C_1 - (-J_1) \tag{16}$$

$$r_1'(D_2 - J_2 - D_1) = C_2 + C_3 + C_1 - (D_2 - J_2 - D_1) \tag{17}$$

$$r_1'(D_3 - J_3' - D_1) = C_3 + C_1 - (D_3 - J_3' - D_1) \tag{18}$$

Consequently, the worst case response time of task τ_1 is:

$$\begin{aligned} r_1' &= \max\left\{r_1(-J_1), r_1(D_2 - J_2 - D_1), r_1(D_3 - J_3' - D_1)\right\} \\ &= \max\left\{C_1 - (-J_1), C_2 + C_3 + C_1 - (D_2 - J_2 - D_1), C_3 + C_1 - (D_3 - J_3' - D_1)\right\} \end{aligned} \tag{19}$$

To conclude with task τ_1, we need to demonstrate that $r_1' \geq r_1$. Using equations (15) and (19), we need to prove:

$$\begin{aligned} &\max\left\{C_1 - (-J_1), C_2 + C_3 + C_1 - (D_2 - J_2 - D_1), C_3 + C_1 - (D_3 - J_3' - D_1)\right\} \geq \\ &\max\{C_1 - (-J_1), C_2 + C_1 - (D_2 - J_2 - D_1), C_3 + C_2 + C_1 - (D_3 - J_3 - D_1)\} \end{aligned} \tag{20}$$

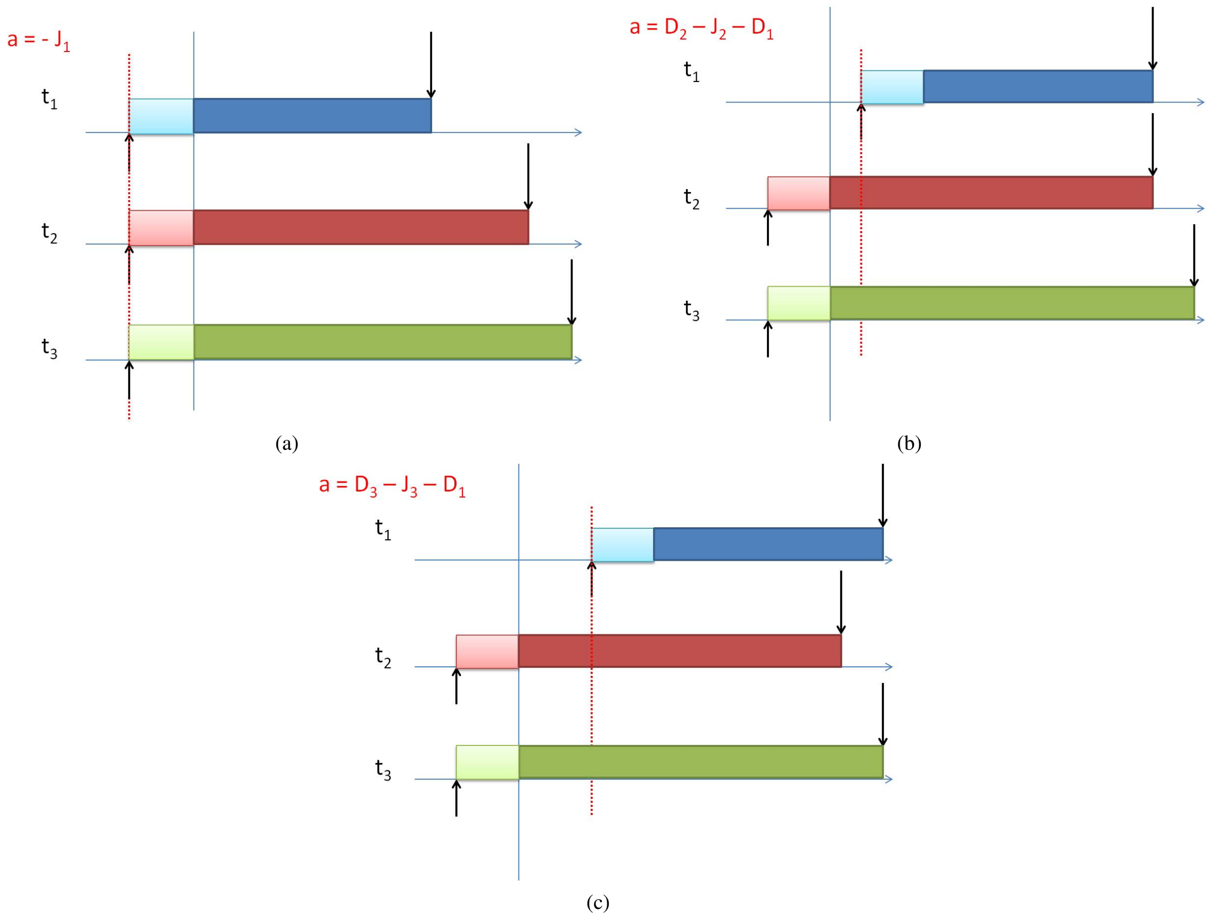


Figure 3: First analysis scenario of task τ_1. (a) Synchronous activation. (b) Synchronous deadline with task τ_2. (c) Synchronous deadline with task τ_3.

Therefore, we need to demonstrate that all terms of the max function in the right side of the inequality have a greater one in the left term of the inequality:

$$C_1 - (-J_1) \geq C_1 - (-J_1) \tag{21}$$

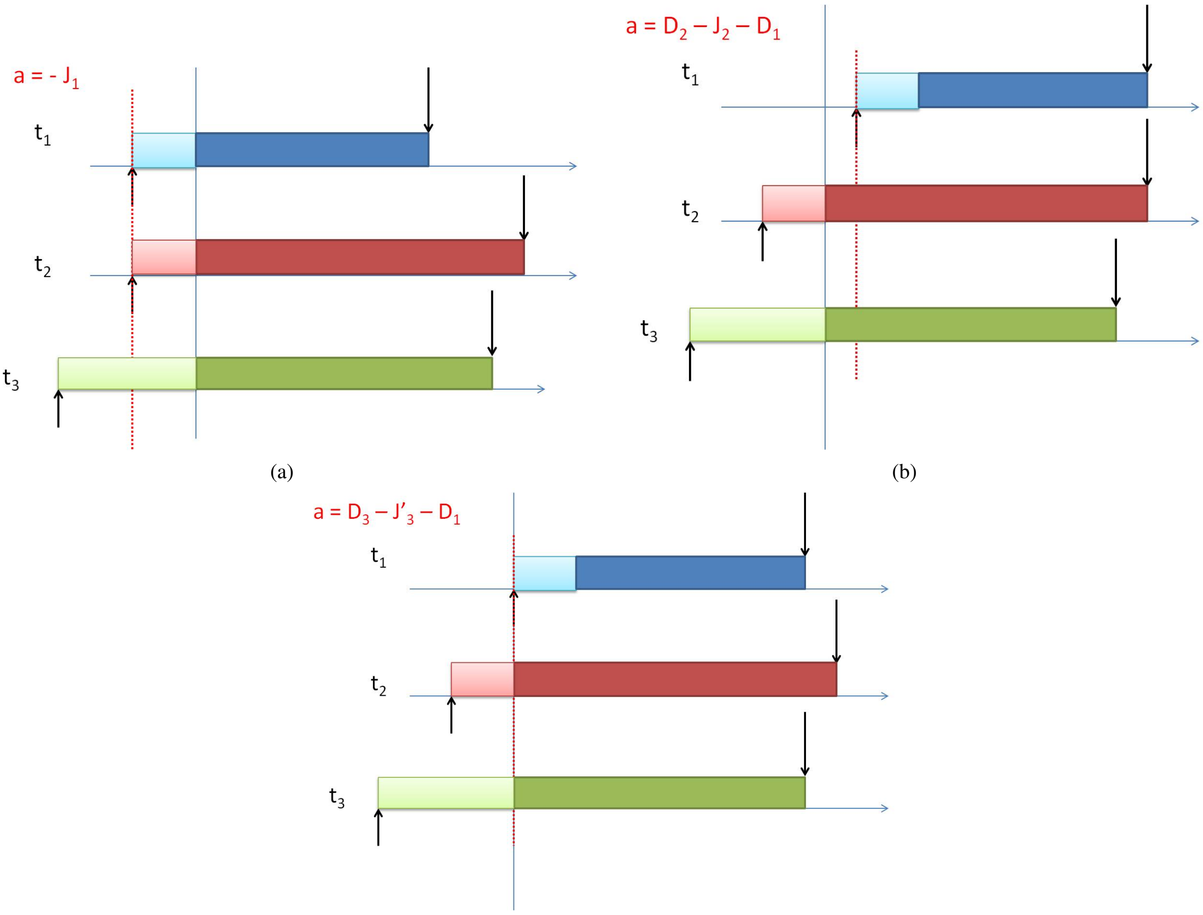
$$C_2 + C_3 + C_1 - (D_2 - J_2 - D_1) \geq C_2 + C_1 - (D_2 - J_2 - D_1) \tag{22}$$

$$C_2 + C_3 + C_1 - (D_2 - J_2 - D_1) \geq C_3 + C_2 + C_1 - (D_3 - J_3 - D_1) \tag{23}$$

Inequality (21) is trivially true because both sides of the inequality are equal. Inequality (22) is also trivially true because C_3 is always positive. Resolving inequality (23), we obtain that $D_2 - J_2 \leq D_3 - J_3$. Because of the condition in equation (10), this inequality is true.

At this point, we have demonstrated that task τ_1 cannot decrease its worst case response time because of the increase of the release jitter of τ_3. Now we will complete the same process with task τ_2. First, we will analyse the response time of task τ_2 in the first scenario. In this case, we only need to analyse two activations, when all the tasks are activated synchronously with the maximum jitter ($a = -J_2$, see Fig. 5a) and when the deadline of τ_2 coincides with some other task's deadline, in this case, τ_3 ($a = D_3 - J_3 - D_2$, see Fig. 5b). In this case, we don't need to analyse the response time when the deadline of task τ_2 coincides with the deadline of task τ_1. The reason is that it is impossible to delay task τ_2 to match its deadline with the deadline of task τ_1. As we have selected an infinite period for task τ_1, it has only one job, and its deadline is previous to τ_2's job.

If we apply Spuri's analysis, explained in section 2.2, to these activations we obtain these response times for τ_2:



(a)

(b)

(c)

Figure 4: Second analysis scenario of task τ_1. (a) Synchronous activation. (b) Synchronous deadline with task τ_2. (c) Synchronous deadline with task τ_3.

$$r_2(-J_2) = C_1 + C_2 - (-J_2) \tag{24}$$

$$r_2(D_3 - J_3 - D_2) = C_1 + C_2 + C_3 - (D_3 - J_3 - D_2) \tag{25}$$

$$r_2 = \max\{r_2(-J_2), r_2(D_3 - J_3 - D_2)\} \tag{26}$$

$$r_2 = \max\{C_1 + C_2 - (-J_2), C_1 + C_2 + C_3 - (D_3 - J_3 - D_2)\} \tag{27}$$

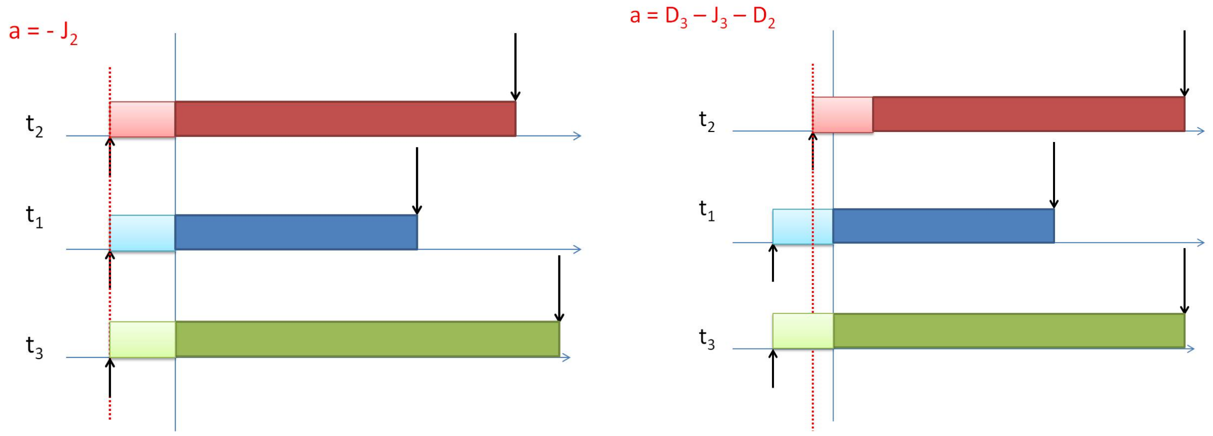
Now we analyse the response time of task τ_2 in the second scenario, after increasing the jitter of task τ_3. In this case we only need to analyse one situation: when all tasks are activated synchronously and with maximum jitter ($a = -J_2$, see Fig. 6). The reason is that it is impossible to delay task τ_2 to match its deadline with other task's deadlines.

The worst case response time that we obtain is:

$$r_2' = r_2'(-J_2) = C_1 + C_3 + C_2 - (-J_2) \tag{28}$$

To finish with task τ_2, we must demonstrate that $r_2' \geq r_2$. From equations (27) and (28), we find that we must prove:

$$C_1 + C_3 + C_2 - (-J_2) \geq \max\{C_1 + C_2 - (-J_2), C_1 + C_2 + C_3 - (D_3 - J_3 - D_2)\} \tag{29}$$



(a) (b)

Figure 5: First analysis scenario of task τ_2. (a) Synchronous activation. (b) Synchronous deadline with task τ_3.

To demonstrate this, we need to check that the term in the left side of the inequality is greater than or equal to all the terms in the right side:

$$C_1 + C_3 + C_2 - (-J_2) \geq C_1 + C_2 - (-J_2) \tag{30}$$

$$C_1 + C_3 + C_2 - (-J_2) \geq C_1 + C_2 + C_3 - (D_3 - J_3 - D_2) \tag{31}$$

Inequality (30) is trivially true because C_3 is always positive. Solving inequality (31) we obtain that $D_2 - J_2 \leq D_3 - J_3$. As mentioned before, the condition in inequality (10) makes this inequality true.

At this point, we have demonstrated that tasks τ_1 and τ_2 cannot decrease their worst case response times because of the increase of the release jitter of τ_3. Now we will finish by analysing task τ_3. First, we will analyse the response time of task τ_3 in the first scenario. In this case, we only need to analyse one activation, when all the tasks are activated synchronously with the maximum jitter ($a = -J_3$, see Fig. 7). In this case, we don't need to analyse the response time when the deadline of task τ_3 coincides with the deadline of some other task. The reason is that it is impossible to delay task τ_3 to match its deadline with the others.

If we apply Spuri's analysis to these activations we obtain a response time for τ_3:

$$r_3 = r_3(-J_3) = C_1 + C_2 + C_3 - (-J_3) \tag{32}$$

Now we analyse the response time of task τ_3 in the second scenario, after increasing its jitter. In this case we need to analyse two situations: when all tasks are activated synchronously and with maximum jitter ($a = -J'_3$, see Fig. 8a) and when the deadline of task τ_3 matches task τ_2's deadline ($a = D_2 - J_2 - D_3$, see Fig. 8b).

If we analyse these situations, the worst case response time that we obtain is:

$$\begin{aligned} r'_3 &= \max\left\{r'_3(-J'_3), r'_3(D_2 - J_2 - D_3)\right\} \\ &= \max\left\{C_1 + C_3 - (-J'_3), C_1 + C_2 + C_3 - (D_2 - J_2 - D_3)\right\} \end{aligned} \tag{33}$$

To finish with task τ_3, we must demonstrate that $r'_3 \geq r_3$. From equations (32) and (33), we find that we must prove:

$$\max\left\{C_1 + C_3 - (-J'_3), C_1 + C_2 + C_3 - (D_2 - J_2 - D_3)\right\} \geq C_1 + C_2 + C_3 - (-J_3) \tag{34}$$

To demonstrate this, we need to check that there is a term in the left side of the inequality that is greater than or equal to the term in the right side:

$$C_1 + C_2 + C_3 - (D_2 - J_2 - D_3) \geq C_1 + C_2 + C_3 - (-J_3) \tag{35}$$

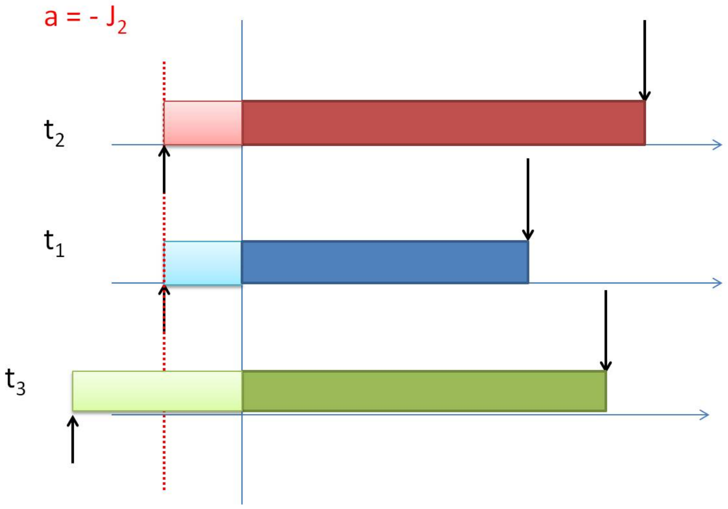


Figure 6: Second analysis scenario of task τ_2, synchronous activation.

Solving inequality (35) we obtain that $D_2 - J_2 \leq D_3 - J_3$. As mentioned before, the condition in inequality (10) makes this inequality true.

Consequently, we have evidenced that if the release jitter of task τ_3, J_3, is increased, the response times of the tasks do not decrease their value. We can extend this statement to the increase of J_1 or J_2. Tasks that are initially below the task that changes its jitter in the effective deadline order will keep the same effective deadline order after the jitter increase and thus would not be affected by it. As an example suppose the previous task set but adding a fourth task τ_4, so that $D_3 - J_3 \leq D_4 - J_4$. If we apply Spuri's holistic response time algorithm before and after the increase of J_3, we obtain the same response time for τ_4, that is $r_4 = r'_4$.

At this point, we have demonstrated that three tasks with infinite periods and without shared resources cannot decrease their worst case response time because of the increase of their release jitters. □

3.1.2. System with Any Number of Tasks

Now, we will generalise the analysis of the previous section to cover systems with any number of tasks.

Lemma 3.2. *Let τ be a group of n tasks with infinite periods and without shared resources, scheduled under EDF. The response time of any of the tasks in τ cannot be decreased by the effect of increasing the release jitter of any of the tasks in τ.*

Let us take a system with $n + m + k$ tasks divided in three groups that would be affected in different ways by a task that would increase its release jitter (see Fig. 9). In Fig. 9a, it can be seen that the group of tasks is ordered by effective deadline:

$$D_i - J_i \leq D_{i+1} - J_{i+1} \tag{36}$$

The tasks are divided in (possibly empty) different groups: Group 1 (tasks from τ_1 to τ_n) have effective deadlines that are always less than those of the tasks from the other groups; Group 2 (tasks from τ_{n+1} to τ_{n+m}) have effective deadlines that are initially less than those of other groups except Group 1, but become greater than the effective deadline of the task that increases its release jitter in scenario 2; Group 3(tasks from τ_{n+m+2} to τ_{n+m+k}) have effective deadlines that are always larger than those of the other groups; finally, task τ_{n+m+1} is the one that changes its release jitter. In Fig. 9b, we can see how task τ_{n+m+1}'s jitter has increased and the order of the effective deadlines of the tasks has changed. Now the order is:

$$D_z - J_z \leq D_{n+m+1} - J_{n+m+1} \leq D_y - J_y \leq D_x - J_x, where\ 1 \leq z \leq n, n+1 \leq y \leq n+m, n+m+2 \leq x \leq n+m+k \tag{37}$$

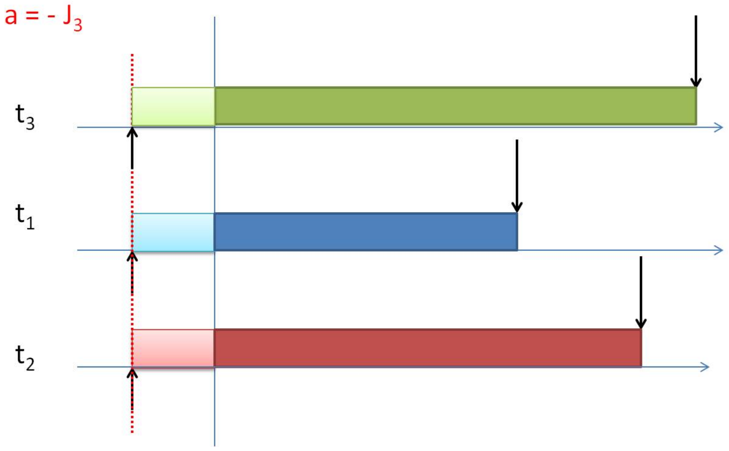
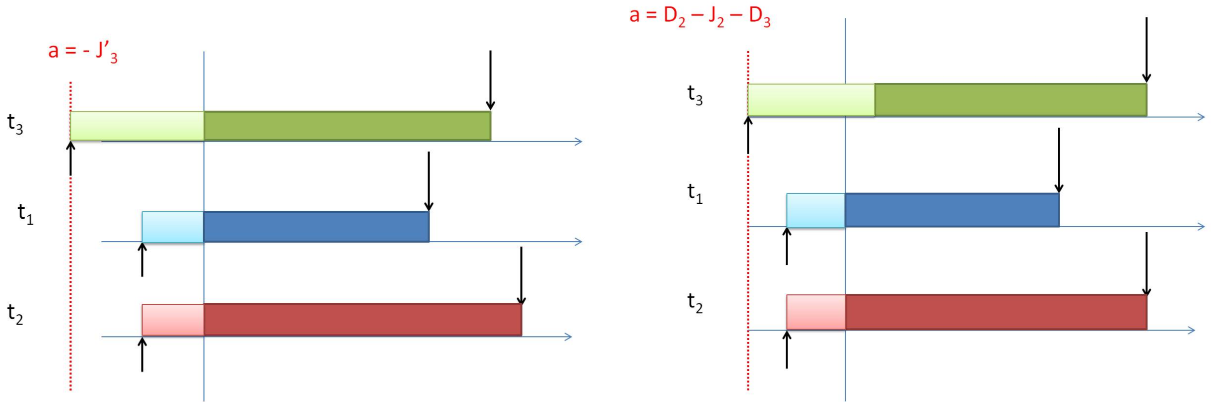


Figure 7: First analysis scenario of task τ_3, synchronous activation.



(a)

(b)

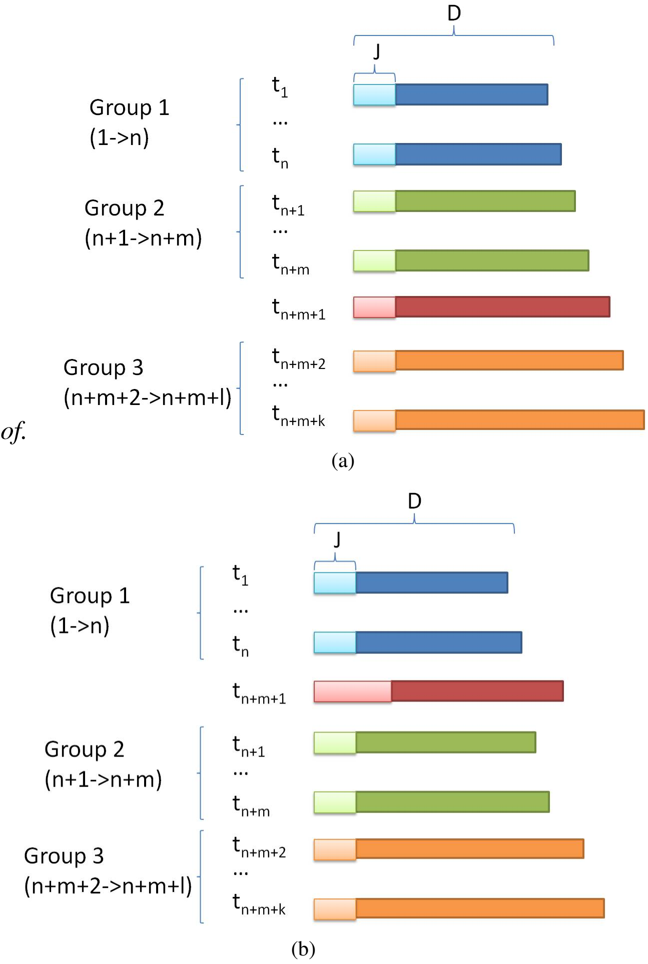
Figure 8: Second analysis scenario of task τ_3. (a) Synchronous activation. (b) Synchronous deadline with task τ_2.

In summary, task τ_{n+m+1} is below Group 2 in Fig. 9a and is then positioned between groups 1 and 2 in Fig. 9b.

In order to demonstrate the lemma we will prove that any task in these groups cannot decrease its response time when the system changes from the scenario shown in Fig. 9a to the one shown in Fig. 9b, that is, when the jitter of task τ_{n+m+1} is increased. When the jitter of task τ_{n+m+1} is increased the preemption order of the task changes and this could induce a reduction in the response time of this task or other tasks. We will prove that this cannot happen. To fulfil this objective we will analyse the groups separately, starting with Group 1 and finishing with Group 3. We also have to check the own task that changes its release jitter.

To start with Group 1, let $\tau_\alpha \in (1 \leq \alpha \leq n)$ be the task under analysis. First, we will analyse the response time of the tasks in the first scenario (see Fig. 9a). Then, we will increase the release jitter of task τ_{n+m+1} and we will do the analysis of the second scenario (see Fig. 9b).Finally we will compare the results and demonstrate that the response time cannot decrease. In the first scenario, the activations that have to be analysed occur when all the tasks are released synchronously with their maximum jitter and when task τ_α's deadline matches the deadline of some other task with longer effective deadline. In the synchronous case the task under analysis, τ_α, is released at the beginning of the busy period having experienced its maximum jitter. The set of the task activations that we have to check, including the synchronous case, is: $\Psi = \cup(D_i - J_i - D_\alpha) \vee i \in n + m + k \; and \; D_i - J_i \geq D_\alpha - J_\alpha$. We are going to separate Ψ in four cases depending on which group does τ_i belong: $A1$ when $\alpha \leq i \leq n$, $A2$ when $n+1 \leq i \leq n+m$, $A3$ when $i = n+m+1$

Proof.



(a)

(b)

Figure 9: System with any number of tasks. (a) System before increasing the release jitter of task τ_{n+m+1}. (b) System after increasing the release jitter of task τ_{n+m+1}.

and $A4$ when $n+m+2 \leq i \leq n+m+k$. For each case, let us define the activation time a as $a(i) = D_i - J_i - D_\alpha$. Then, if we apply Spuri's holistic algorithm, explained in section 2.2, to these activation cases, we obtain these response times:

$$A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{38}$$

$$A2 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{39}$$

$$A3 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) \tag{40}$$

$$A4 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{41}$$

In the second situation, after increasing the jitter of task τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), the activation cases to be analysed are the same as in the first situation. The reason is that the position of group 1 in the effective deadline order remains the same. Now we will use the letter B to differentiate the cases: $B1$ when $\alpha \leq i \leq n$, $B2$ when $n + 1 \leq i \leq n + m$, $B3$ when $i = n + m + 1$ and $B4$ when $n + m + 2 \leq i \leq n + m + k$. The response times are:

$$B1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{42}$$

$$B2 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{43}$$

$$B3 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} - \left(D_{n+m+1} - J'_{n+m+1} - D_\alpha\right) \tag{44}$$

$$B4 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{45}$$

Now we need to demonstrate that the response time of a task τ_α in Group 1 cannot be larger in the first scenario than in the second. Therefore, the next inequality has to be true:

$$\max\{r(A1) \cup r(A2) \cup r(A3) \cup r(A4)\} \leq \max\{r(B1) \cup r(B2) \cup r(B3) \cup r(B4)\}. \tag{46}$$

where $r(X)$ is the set of response times in X.

It is sufficient to prove that each case in the left side of the inequality has a greater or equal one in the right side. In that way, we can break the inequality down in four:

$$\begin{aligned}
&r(B1) \geq r(A1)\\
&\left(\sum_{j=1}^{i} C_j\right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n) \geq \left(\sum_{j=1}^{i} C_j\right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n)
\end{aligned} \tag{47}$$

$$\begin{aligned}
&r(B2) \geq r(A2)\\
&\left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha), (n \leq i \leq n+m) \geq\\
&\left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha), (n \leq i \leq n+m)
\end{aligned} \tag{48}$$

$$\begin{aligned}
&r(B2) \geq r(A3)\\
&\left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha), (n \leq i \leq n+m) \geq\\
&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha)
\end{aligned} \tag{49}$$

$$\begin{aligned}
&r(B4) \geq r(A4)\\
&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha), (n+m+2 \leq i \leq n+m+k) \geq\\
&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha), (n+m+2 \leq i \leq n+m+k)
\end{aligned} \tag{50}$$

Inequalities (47) and (50) are trivially true because both sides have the same terms. From inequality (48), we obtain $C_{n+m+1} \geq 0$ which is always true because execution times are positive. And finally, in inequality (49), if we take $i = n + m$, we can obtain that $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$. This is always true because of the condition in inequality (36).

In second place, we repeat this analysis with group 2. In the first scenario, the activation cases that have to be analysed are: $A1$ when $\alpha \leq i \leq n + m$, $A2$ when $i = n + m + 1$ and $A3$ when $n + m + 2 \leq i \leq n + m + k$. Thus, if we apply Spuri's holistic algorithm, described in section 2.2, to these activation groups, we obtain these response times:

$$A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{51}$$

$$A2 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) \tag{52}$$

$$A3 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_{n+m+1} - J_{n+m+1} - D_\alpha) \tag{53}$$

In the second scenario, after increasing the jitter of task τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), the activation cases to be analysed are only two. The reason is that the position of group 2 in the effective deadline order has changed. Now we will use letter B to differentiate them: $B1$ when $\alpha \leq i \leq n + m$ and $B2$ when $n + m + 2 \leq i \leq n + m + k$. The response times are:

$$B1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{54}$$

$$B2 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j\right) + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{55}$$

Now we need to demonstrate that the response time of a task τ_α with $(n + 1 \leq \alpha \leq n + m)$ cannot be larger in the first scenario than in the second. Therefore, the following inequality has to be true:

$$\max\{r(A1) \cup r(A2) \cup r(A3)\} \leq \max\{r(B1) \cup r(B2)\}. \tag{56}$$

It is sufficient to prove that each case in the left side of the inequality has a greater or equal one in the right side. In that way, we can break the inequality down in three:

$$\begin{aligned} & r(B1) \geq r(A1) \\ & \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n + m) \geq \\ & \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n + m) \end{aligned} \tag{57}$$

$$\begin{aligned} & r(B1) \geq r(A2) \\ & \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n + m) \geq \\ & \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) \end{aligned} \tag{58}$$

$r(B2) \geq r(A3)$

$$\begin{aligned}&\left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j\right) + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha)\,, (n+m+2 \leq i \leq n+m+k) \geq \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha)\,, (n+m+2 \leq i \leq n+m+k)\end{aligned} \tag{59}$$

Inequality (59) is trivially true because both sides have the same terms. From inequality (57), we obtain $C_{n+m+1} \geq 0$ which is always true. And finally, in inequality (58), if we take $i = n + m$, we can obtain that $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$. This is always true because of the condition in equation (36).

In third place, we will repeat this analysis with task τ_{n+m+1}. In the first scenario, the activation cases that have to be analysed are: $A1$ when $i = n + m + 1$ and $A2$ when $n + m + 2 \leq i \leq n + m + k$. Thus, if we apply Spuri's holistic algorithm, described in section 2.2, to these activation cases, we obtain these response times:

$$A1 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (-J_{n+m+1}) \tag{60}$$

$$A2 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{n+m+k} C_j\right) - (D_i - J_i - D_{n+m+1}) \tag{61}$$

In the second scenario, after increasing the jitter of task τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), the activation cases to be analysed are three. The reason is that the position of task τ_{n+m+1} in the effective deadline order has changed. Now we will use the letter B to differentiate the activation cases: $B1$ when $n \leq i \leq n + m$, $B2$ when $i = n + m + 1$ and $B3$ when $n + m + 2 \leq i \leq n + m + k$. The response times are:

$$B1 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) + C_{n+m+1} - (D_i - J_i - D_{n+m+1}) \tag{62}$$

$$B2 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} - \left(-J'_{n+m+1}\right) \tag{63}$$

$$B3 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_{n+m+1}) \tag{64}$$

Thus, we need to demonstrate that the response time of task τ_α with $(n + 1 \leq a \leq n + m)$ cannot be larger in the first scenario than in the second. Therefore, the next inequality has to be true:

$$\max\left\{r(A1) \cup r(A2)\right\} \leq \max\left\{r(B1) \cup r(B2) \cup r(B3)\right\}. \tag{65}$$

It is sufficient to prove that each group in the left side of the inequality has a greater or equal one in the right side. In that way, we can break the inequality down in two:

$r(B1) \geq r(A1)$

$$\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) + C_{n+m+1} - (D_i - J_i - D_{n+m+1})\,, (n \leq i \leq n+m) \geq \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (-J_{n+m+1}) \tag{66}$$

$r(B3) \geq r(A2)$

$$\begin{aligned}&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_{n+m+1})\,, (n+m+1 \leq i \leq n+m+k) \geq \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_{n+m+1})\,, (n+m+1 \leq i \leq n+m+k)\end{aligned} \tag{67}$$

Inequality (67) is trivially true because both sides have the same terms. From inequality (66), if we take $i = n + m$, we can obtain that $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$. This is always true because of the condition in equation (36).

Finally, we will repeat this analysis with group 3. In the first scenario, we only need to analyse one set of activations: $A1$ when $n + m + 2 \leq i \leq n + m + k$. Thus, if we apply Spuri's holistic algorithm to this activation group, we obtain this response time:

$$A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{68}$$

In the second scenario, after increasing the jitter of task τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), there is also only one activation case to be analysed. The reason is that the position of group 3 in the effective deadline order has not changed. Now we will use letter B to differentiate them: $B1$ when $n + m + 2 \leq i \leq n + m + k$. The response times are:

$$B1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) \tag{69}$$

Now we need to demonstrate that the response time of task τ_α, $\alpha \in (n + 1 \leq \alpha \leq n + m)$, cannot be bigger in the first scenario than in the second. Therefore, the next inequality has to be true:

$$\max\{r(A1)\} \leq \max\{r(B1)\}. \tag{70}$$

It is sufficient to prove that the response times in case B1 are greater than or equal to those in A1. In that way, we need to prove the validity of the inequality:

$$\begin{aligned}
&r(B1) \geq r(A1)\\
&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha), (n + m + 2 \leq i \leq n + m + k) \geq\\
&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha), (n + m + 2 \leq i \leq n + m + k)
\end{aligned} \tag{71}$$

Inequality (71) is trivially true because both sides of the inequality have the same terms.

Consequently, we have evidenced that if the release jitter of task τ_{n+m+1}, J_{n+m+1}, is increased, the response times of the tasks do not decrease their value. This statement can be extended to the increase of the release jitter of any task or set of tasks. Through the task groups 1, 2 and 3 the demonstration takes into account all the roles that may occur in relation with one task that changes its jitter. If there are several tasks that change their jitter we can make successive changes with one task at a time, and observe that the response times never decrease.

At this point, we have demonstrated that a system with any number of tasks with infinite periods and without shared resources cannot decrease their worst case response time because of the increase of their release jitters.

□

3.2. System with Resource Sharing

In this subsection, we will extend the analysis done in the previous section to include the use of shared resources. Shared resources introduce a new dimension to the problem. Spuri's proposal manages resources according to a protocol like the Stack Resource Policy (Baker, 1991) or Priority Ceiling (Sha et al., 1990) and assigns the task's preemption levels π_n and shared resource's priority ceilings $\lceil s \rceil$ according to:

$$\begin{gathered}
\pi_i > \pi_j \Longleftrightarrow D_i - J_i < D_j - J_j\\
\lceil s \rceil = \max\{\pi_i : i\ may\ lock\ s\}
\end{gathered} \tag{72}$$

So the task's preemption levels and shared resource's priority ceilings are ordered with the effective deadlines and therefore depend on the release jitters. This fact can cause that an increase of the release jitter of one task could change the ordering of the preemption levels (and/or ceilings). This effect would change the blocking times and thus the response times of the tasks could decrease. Consequently, it appears that this effect could cause the non-convergence of the iterative EDF response time analysis. In order to demonstrate that this potential problem can never occur, we will assume a system with any number of tasks, like the one presented in Fig. 9, but in this case, the tasks can use any number of shared resources. As in the previous section we will demonstrate that the response times of each task cannot decrease because of the increase of the release jitter.

Lemma 3.3. *Let τ be a group of tasks with infinite periods scheduled under EDF and using shared resources under the SRP or PCP protocol. The response time of any of the tasks in τ cannot decrease by the effect of increasing the release jitter of any of the tasks in τ.*

Proof. To prove the lemma, we will follow the same analysis schema as in subsection 3.1.2. We have separated the tasks in three groups and a task that will increase its jitter: Group 1 (tasks from τ_1 to τ_n), Group 2 (tasks from τ_{n+1} to τ_{n+m}), Group 3(tasks from τ_{n+m+2} to τ_{n+m+k}), and the task that increases its jitter is τ_{n+m+1}. We start from an initial scenario with specific values of release jitter and we then move to a second scenario where the release jitter of task τ_{n+m+1} has been incremented (see Fig. 9b). Each group will be analysed to demonstrate that in the second scenario the response times are always greater than in the first scenario (see Fig. 9a). To reach this target we will analyse the groups separately, starting with group 1, then doing group 2, the task that changes its jitter, and then finishing with group 3.

To start with group 1, let τ_α with $(1 \leq \alpha \leq n)$ be the task under analysis. First, we will analyse the response time of the tasks in the first scenario (see Fig. 9a). Then, we will increase the jitter of task τ_{n+m+1} and we will do the analysis for the second scenario (see Fig. 9b). Finally we will compare them and demonstrate that the response time cannot decrease.

In the first scenario, the activations that have to be analysed occur when all the tasks are released synchronously after having experienced their maximum jitter or when task τ_α's deadline matches another task's deadline. The set of the task activations that we have to check, including the synchronous case, is: $\Psi = \{\, i \mid D_i - J_i \geq D_\alpha - J_\alpha\}$. We are going to separate Ψ in four cases depending on which group does τ_i belong to: $A1$ when $\alpha \leq i \leq n$, $A2$ when $n+1 \leq i \leq n+m$, $A3$ when $i = n+m+1$ and $A4$ when $n+m+2 \leq i \leq n+m+k$. If we apply Spuri's holistic algorithm to these activation cases, we obtain these response times:

$$A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, where\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\} \tag{73}$$

$$A2 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, where\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\} \tag{74}$$

$$\begin{gathered} A3 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) + B_{n+m+1}, \\ where\ B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ n+m+2 \leq j \leq n+m+k\right\} \end{gathered} \tag{75}$$

$$\begin{gathered} A4 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ where\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\} \end{gathered} \tag{76}$$

As we can see the difference with the analysis in section 3.1.2 is the blocking time terms generated by the critical sections using resources of higher preemption level than the task under analysis and belonging to tasks that cannot preempt the task job under analysis.

In the second scenario, after increasing the jitter of task τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), the activation cases to be analysed are the same as in the first scenario. The reason is that the position of group 1 in the effective deadline order remains the same. Now we will use letter B to differentiate them: $B1$ when $\alpha \leq i \leq n$, $B2$ when $n+1 \leq i \leq n+m$, $B3$ when $i = n+m+1$ and $B4$ when $n+m+2 \leq i \leq n+m+k$. The response times are:

$$B1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \textit{where } B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } i+1 \leq j \leq n+m+k\right\} \tag{77}$$

$$\begin{aligned} B2 : r_\alpha(a(i)) &= \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ \textit{where } B_i &= \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } i+1 \leq j \leq n+m+k \textit{ and } j \neq n+m+1\right\} \end{aligned} \tag{78}$$

$$\begin{aligned} B3 : r_\alpha(a(i)) &= \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} - \left(D_{n+m+1} - J'_{n+m+1} - D_\alpha\right) + B_i, \\ \textit{where } B_i &= \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } n+1 \leq j \leq n+m+k \textit{ and } j \neq n+m+1\right\} \end{aligned} \tag{79}$$

$$\begin{aligned} B4 : r_\alpha(a(i)) &= \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ \textit{where } B_i &= \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{80}$$

We can see that the blocking times in B1 are the same as in A1 because the preemption level change only affects tasks that are always below the levels of the tasks in group 1. Similarly, blocking times in B4 are the same as in A4 because the preemption levels of tasks in group 4 do not change. However, the blocking times may have changed for the activations cases B2 and B3 in scenario 2, in relation with A2 and A3 in scenario 1. The blocking times for the tasks in B2 may be smaller than in A2 because they are no longer influenced by the critical sections of τ_{n+m+1}. Oppositely, the blocking time of the task in B3 may be larger than in A3 because it may now be affected by the critical sections of the tasks in group 2.

Now we need to demonstrate that the response time of a task $\tau_\alpha, (1 \leq \alpha \leq n)$ cannot be bigger in the first scenario than in the second. Therefore, the next inequality has to be true:

$$\max\left\{r(A1) \cup r(A2) \cup r(A3) \cup r(A4)\right\} \leq \max\left\{r(B1) \cup r(B2) \cup r(B3) \cup r(B4)\right\}. \tag{81}$$

It is sufficient to prove that each group in the left side of the inequality has a greater or equal one in the right side. In that way, we can break the inequality down in four:

$$\begin{aligned} & r(B1) \geq r(A1) \\ & \left(\sum_{j=1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \textit{ where } (\alpha \leq i \leq n) \textit{ and } B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } i+1 \leq j \leq n+m+k\right\} \geq \\ & \left(\sum_{j=1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \textit{ where } (\alpha \leq i \leq n) \textit{ and } B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{82}$$

$$\begin{aligned} & r(B2) \geq r(A2) \\ & \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \textit{ where } (n \leq i \leq n+m) \textit{ and} \\ & B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } i+1 \leq j \leq n+m+k \textit{ and } j \neq n+m+1\right\} \geq \\ & \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \textit{ where } (n \leq i \leq n+m) \textit{ and} \\ & B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{83}$$

$$
\begin{aligned}
&r(B2) \geq r(A3)\\
&\left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i,\ where\ (n \leq i \leq n+m)\ and\\
&B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\ and\ j \neq n+m+1\right\} \geq\\
&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) + B_{n+m+1},\\
&where\ B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ n+m+2 \leq j \leq n+m+k\right\}
\end{aligned} \tag{84}
$$

$$
\begin{aligned}
&r(B4) \geq r(A4)\\
&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i,\\
&where\ (n+m+2 \leq i \leq n+m+k)\ and\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\} \geq\\
&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i,\\
&where\ (n+m+2 \leq i \leq n+m+k)\ and\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\}
\end{aligned} \tag{85}
$$

Inequalities (82) and (85) are trivially true because both sides have the same terms. From inequality (83), we obtain:

$$
\begin{aligned}
&C_{n+m+1} + B_i,\ where\ (n \leq i \leq n+m)\ and\\
&B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\ and\ j \neq n+m+1\right\} \geq\\
&B_i,\ where\ (n \leq i \leq n+m)\ and\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\}
\end{aligned} \tag{86}
$$

The blocking time (B_i) in the right part of the inequality can only be bigger than the blocking time in the left part when $j = n + m + 1$. In this case, B_i would be cs_{n+m+1} which can never be bigger than C_{n+m+1}. Then, inequality (83) is correct.

Finally, in inequality (84) it is sufficient to find an activation in B2 that it is greater than those in A3. Therefore, we will take $i = n + m$ to obtain:

$$
\begin{aligned}
&-(D_{n+m} - J_{n+m}) + B_{n+m},\ where\ B_{n+m} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ n+m+1 \leq j \leq n+m+k\ and\ j \neq n+m+1\right\} \geq\\
&-(D_{n+m+1} - J_{n+m+1}) + B_{n+m+1},\ where\ B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ n+m+2 \leq j \leq n+m+k\right\}
\end{aligned} \tag{87}
$$

Analysing the resulting inequality, we can say that the blocking time B_{n+m+1} before increasing the jitter of the task τ_{n+m+1} is the same as the blocking time B_{n+m} after increasing the jitter. The reason is that tasks $\{n + m + 1 \leq j \leq n + m + k\ and\ j \neq n + m + 1\}$ are the same as $\{n + m + 2 \leq j \leq n + m + k\}$. Then, the resulting inequality is $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$, which is true because of the condition in equation (36). Then it is demonstrated that the response time of tasks in the first group cannot be decreased by a jitter increase.

In second place, we will repeat this analysis with group 2: the task under analysis is τ_α with $\alpha \in [n + 1..n + m]$. In the first scenario the activation cases that have to be analysed are: $A1$ when $\alpha \leq i \leq n + m$, $A2$ when $i = n + m + 1$ and $A3$ when $n + m + 2 \leq i \leq n + m + k$. There are only three cases (not four, like in the previous case) because the task τ_α cannot be delayed to coincide its deadline with the deadlines of the tasks in Group 1. Thus, if we apply Spuri's holistic algorithm to these activation cases, we obtain these response times:

$$
A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i,\ where\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\} \tag{88}
$$

$$\begin{aligned} &A2 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) + B_{n+m+1}, \\ &\textit{where } B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } n+m+2 \leq j \leq n+m+k\right\} \end{aligned} \tag{89}$$

$$\begin{aligned} &A3 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_{n+m+1} - J_{n+m+1} - D_\alpha) + B_i, \\ &\textit{where } B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{90}$$

In the second scenario, after increasing the jitter of task τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), the activation cases to be analysed are only two. The reason is that the position of group 2 in the effective deadline order has changed. Now we will use the letter B to differentiate them: $B1$ when $\alpha \leq i \leq n+m$ and $B2$ when $n+m+2 \leq i \leq n+m+k$. The response times are:

$$\begin{aligned} &B1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ &\textit{where } B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } n+1 \leq j \leq n+m+k \textit{ and } j \neq n+m+1\right\} \end{aligned} \tag{91}$$

$$\begin{aligned} &B2 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j\right) + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ &\textit{where } B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } n+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{92}$$

Now we need to demonstrate that the response time of a task τ_α, $n+1 \leq \alpha \leq n+m$, cannot be bigger in the first situation than in the second. Therefore, the next inequality has to be true:

$$\max\left\{r(A1) \cup r(A2) \cup r(A3)\right\} \leq \max\left\{r(B1) \cup r(B2)\right\}. \tag{93}$$

It is sufficient to prove that each group in the left side of the inequality has a greater or equal one in the right side. In that way, we can break the inequality down in three:

$$\begin{aligned} &r(B1) \geq r(A1) \\ &\left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ &\textit{where } (\alpha \leq i \leq n+m) \textit{ and } B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } n+1 \leq j \leq n+m+k \textit{ and } j \neq n+m+1\right\} \geq \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) - (D_i - J_i - D_\alpha), \textit{ where } (\alpha \leq i \leq n+m) \\ &\textit{and } B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{94}$$

$$\begin{aligned} &r(B2) \geq r(A2) \\ &\left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j\right) + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ &\textit{where } (n+m+2 \leq i \leq n+m+k) \textit{ and } B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } n+1 \leq j \leq n+m+k\right\} \geq \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) + B_{n+m+1}, \\ &\textit{where } B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i \textit{ and } n+m+2 \leq j \leq n+m+k\right\} \end{aligned} \tag{95}$$

$$
\begin{aligned}
&r(B2) \geq r(A3)\\
&\left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j\right) + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i,\\
&\mathit{where}\ \ (n+m+2 \leq i \leq n+m+k)\ \ \mathit{and}\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ n+1 \leq j \leq n+m+k\right\} \geq \\
&\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i,\\
&\mathit{where}\ \ (n+m+2 \leq i \leq n+m+k)\ \ \mathit{and}\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ i+1 \leq j \leq n+m+k\right\}
\end{aligned}
\tag{96}
$$

Inequality (96) is trivially true because both sides have the same terms. Simplifying inequality (94), we obtain the same as in inequality (86). Therefore, it can be solved in the same way. And finally, from inequality (95), we can get:

$$
\begin{aligned}
&\left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i) + B_i, \mathit{where}\ \ (n+m+2 \leq i \leq n+m+k)\ \ \mathit{and}\\
&B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ n+1 \leq j \leq n+m+k\right\} \geq\\
&-(D_{n+m+1} - J_{n+m+1}) + B_{n+m+1},\ \mathit{where}\ B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ n+m+2 \leq j \leq n+m+k\right\}
\end{aligned}
\tag{97}
$$

We can divide this inequality in two new ones:

$$
(D_i - J_i)\,, \mathit{where}\ \ (n+m+2 \leq i \leq n+m+k) \leq (D_{n+m+1} - J_{n+m+1}) \tag{98}
$$

$$
\begin{aligned}
&\left(\sum_{j=n+m+2}^{i} C_j\right) + B_i, \mathit{where}\ \ (n+m+2 \leq i \leq n+m+k)\ \ \mathit{and}\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ n+1 \leq j \leq n+m+k\right\} \geq\\
&B_{n+m+1},\ \mathit{where}\ B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ n+m+2 \leq j \leq n+m+k\right\}
\end{aligned}
\tag{99}
$$

Inequality (98) is true for all the cases of i because of the condition in inequality (36). In inequality (99), it is enough to find a value of i that ensures the correctness of the inequality. In this case, we will use $i = n + m + k$. In that way, any critical section cs_j of task j that can block task τ_{n+m+1} in the right part of the inequality will have its respective execution time C_j in the summation of the left part of the inequality. As C_j is always greater than cs_j, inequality (99) is correct. Therefore it is demonstrated that the response time of tasks in the second group can only be increased when a jitter increase.

In third place, we will repeat this analysis with task τ_{n+m+1}. In the first scenario, the activation cases that have to be analysed are: $A1$ when $i = n + m + 1$ and $A2$ when $n + m + 2 \leq i \leq n + m + k$. There are only two cases (not four, like in the first case) because the task τ_α cannot be delayed to coincide its deadline with the deadlines of the tasks in Group 1 and Group 2. Now if we apply the analysis to these activation cases, we obtain these response times:

$$
\begin{aligned}
&A1 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (-J_{n+m+1}) + B_{n+m+1},\\
&\mathit{where}\ B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ n+m+2 \leq j \leq n+m+k\right\}
\end{aligned}
\tag{100}
$$

$$
\begin{aligned}
&A2 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{n+m+k} C_j\right) - (D_i - J_i - D_{n+m+1}) + B_i,\\
&\mathit{where}\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ i+1 \leq j \leq n+m+k\right\}
\end{aligned}
\tag{101}
$$

In the second scenario, after increasing the jitter of task τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), the activation cases to be analysed are three. The reason is that the position of task τ_{n+m+1} in the effective deadline order has changed. Now we will use the letter B to differentiate them: $B1$ when $n \leq i \leq n+m$, $B2$ when $i = n+m+1$ and $B3$ when $n+m+2 \leq i \leq n+m+k$. The response times are:

$$\begin{aligned} &B1 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) + C_{n+m+1} - (D_i - J_i - D_{n+m+1}) + B_i, \\ &where\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\ and\ j \neq n+m+1\right\} \end{aligned} \tag{102}$$

$$\begin{aligned} &B2 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + C_{n+m+1} - \left(-J'_{n+m+1}\right) + B_i, \\ &where\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ n+1 \leq j \leq n+m+k\ and\ j \neq n+m+1\right\} \end{aligned} \tag{103}$$

$$\begin{aligned} &B3 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_{n+m+1}) + B_i, \\ &where\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{104}$$

Now we need to demonstrate that the response time of task τ_{n+m+1} cannot be bigger in the first scenario than in the second. Therefore, the next inequality has to be true:

$$\max\{r(A1) \cup r(A2)\} \leq \max\{r(B1) \cup r(B2) \cup r(B3)\}. \tag{105}$$

It is sufficient to prove that each group in the left side of the inequality has a greater or equal one in the right side. In that way, we can break the equation down in two equations:

$$\begin{aligned} &r(B1) \geq r(A1) \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{i} C_j\right) + C_{n+m+1} - (D_i - J_i - D_{n+m+1}) + B_i, \\ &where\ (n \leq i \leq n+m)\ and\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\ and\ j \neq n+m+1\right\} \geq \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} - (-J_{n+m+1}) + B_{n+m+1}, \\ &where\ B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ n+m+2 \leq j \leq n+m+k\right\} \end{aligned} \tag{106}$$

$$\begin{aligned} &r(B3) \geq r(A2) \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_{n+m+1}) + B_i, \\ &where\ (n+m+1 \leq i \leq n+m+k)\ and\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\} \geq \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_{n+m+1}) + B_i, \\ &where\ (n+m+1 \leq i \leq n+m+k)\ and\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{107}$$

Inequality (107) is trivially true because both sides have the same terms. In inequality (106) it is enough to find a value for i that demonstrates that the left-hand side is greater than the right-hand side. If we take $i = n+m$ and simplify the equation, we can obtain:

$$\begin{aligned} &-(D_i - J_i - D_{n+m+1}) + B_{n+m}, where\ B_{n+m} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ n+m+2 \leq j \leq n+m+k\right\}, \geq \\ &-(-J_{n+m+1}) + B_{n+m+1},\ where\ B_{n+m+1} = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ and\ n+m+2 \leq j \leq n+m+k\right\} \end{aligned} \tag{108}$$

Analysing inequality (108), we can say that the blocking time B_{n+m+1} before increasing the jitter of the task τ_{n+m+1} is the same than the blocking time B_{n+m} after increasing the jitter because the tasks with higher effective deadline that may affect them are the same. Then we can simplify the equation to obtain $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$, which is always true because of the condition in inequality (36). Then it is demonstrated that the response time of task τ_{n+m+1} cannot be decreased by its jitter increase.

Finally, we will repeat this analysis with group 3: the task under analysis is τ_α, $n+m+2 \leq \alpha \leq n+m+k$. In the first situation, we only need to analyse one set of activation cases: $A1$ when $n+m+2 \leq i \leq n+m+k$. There is only one cases (not four, like in the first case) because the task τ_α cannot be delayed to coincide its deadline with the deadlines of the tasks in Group 1 and Group 2 and the task τ_{n+m+1}. Thus, if we apply Spuri's holistic algorithm to this activation case, we obtain this response time:

$$\begin{aligned} &A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ &\mathit{where}\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{109}$$

In the second scenario, after increasing the jitter of task τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), there is also one activation case to be analysed. The reason is that the position of group 3 in the effective deadline order has not changed. Now we will use the letter B to differentiate this case: $B1$ when $n+m+2 \leq i \leq n+m+k$. The response times are:

$$\begin{aligned} &B1 : r_\alpha(a(i)) = \left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ &\mathit{where}\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{110}$$

Now we need to demonstrate that the response time of a task τ_α, $n+m+2 \leq \alpha \leq n+m+k$, cannot be bigger in the first situation than in the second. Therefore, the next inequality has to be true:

$$\max\{r(A1)\} \leq \max\{r(B1)\}. \tag{111}$$

It is sufficient to prove that the maximum response time in case B1 is greater than or equal to that of A1. In that way, we need to prove the validity of the inequality:

$$\begin{aligned} &r(B1) \geq r(A1) \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ &\mathit{where}\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ i+1 \leq j \leq n+m+k\right\} \geq \\ &\left(\sum_{j=1}^{n} C_j\right) + \left(\sum_{j=n+1}^{n+m} C_j\right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{i} C_j\right) - (D_i - J_i - D_\alpha) + B_i, \\ &\mathit{where}\ B_i = \max\left\{cs_j(s) : \lceil s \rceil \geq \pi_i\ \mathit{and}\ i+1 \leq j \leq n+m+k\right\} \end{aligned} \tag{112}$$

Inequality (112) is trivially true because both sides have the same terms.

Recapitulating, we have evidenced that if the release jitter of task τ_{n+m+1}, J_{n+m+1}, is increased, the response times of the tasks do not decrease their values. This statement can be extended to the increase of the release jitter of any task or set of tasks. Through the task groups 1, 2 and 3, the demonstration takes into account all the roles that may occur in relation with one task that changes its jitter. If there are several tasks that change their jitter we can make successive changes with one task at a time, and observe that the response times never decrease.

At this point, we have demonstrated that a system with any number of tasks with infinite periods and with shared resources cannot decrease their worst case response time because of the increase of their release jitters. □

3.3. System with Finite Periods

In previous sections, we have analysed sets of tasks with their periods restricted to an infinite value. In this section, we will explain why this restriction can be removed without invalidating the above demonstrations.

Theorem 3.1. *Let τ be a group of tasks scheduled under EDF and possibly using shared resources under the SRP or PCP protocol. The response time of any of the tasks in τ cannot decrease by the effect of increasing the release jitter of any of the tasks in τ.*

Proof. Let us start from a system like τ but with infinite periods. According to Lemma 3 the response times can only increase when moving from an initial scenario with specific release jitter values to a second scenario with increased jitters. Now let us gradually reduce the period of one of the tasks so that its second job preempts the task under analysis in scenario 2. If the response time of that particular task is larger in scenario 2 than in scenario 1, the new job brought into the busy period will enlarge it, thus causing the response time of the task under analysis in scenario 2 to become even larger than it was in scenario 1. If the response times of that particular task are the same in scenario 2 as in scenario 1, the new job brought into the busy period will enlarge it, causing the same effect on the response times of the task under both scenarios. Therefore, there is no decrease when moving from scenario 1 to 2. We can now continue reducing the period of the task introducing new jobs in the busy period with the same effect. We can do the same with other tasks. Therefore, the response times never decrease. □

Theorem 1 allows us to conclude that the holistic analysis defined in (Spuri, 1996b) does not suffer from convergence problems due to the use of shared resources, as long as the preemption levels of the tasks and the ceilings of the shared resources are always kept in a coherent order in relation to the varying effective deadlines of the tasks. This latter requirement does not appear explicitly in (Spuri, 1996b) and leads to an extension of the response-time algorithm in which the preemption levels are recalculated at each step (Algorithm 1).

```
begin
    Initialize jitters;
    repeat
        calculate preemption levels and resource ceilings;
        for each task, calculate worst-case response times;
        for each task, calculate jitter=worst-case response - best-case response;
    until stable solution reached or deadlines are not met;
end
```

Algorithm 1: Extended holistic EDF worst-case response time analysis algorithm

4. Industrial Example

In this section we will analyse an industrial example with the technique described in the previous sections. The chosen task model is part of an elevator system, concretely the door operator subsystem. This module is responsible for the opening and closing of the elevator's doors. Furthermore, it has to comply with some temporal restrictions. The most important restriction occurs when the system is closing the doors and an obstacle (usually a person) gets in the way of the door. The system must complete the action of stopping the door closing operation and start opening the door in a specified interval of time, in order not to cause damage to a person standing in the way of the door.

As we can see in the task deployment model (see Fig. 10), it is a real-time distributed system integrated by two processors (operator and maneuver) connected by a CAN bus. The operator processor manages the door opening/closing mechanical tasks, while the maneuver processor takes the decisions about all the system. We can also see the seven transactions or end-to-end flows, ten tasks and two messages that compound the door control and whose parameters are described in Table 1, Table 2 and Table 3, respectively. As we can see, there is a sporadic end-to-end flow called obstacle. For the purpose of worst-case analysis, it has been modelled as a periodic end-to-end flow with a period equal to its minimum inter-arrival time (MIT). The scheduling policy applied in both processors is EDF and the technique described in (Spuri, 1996a), extended with Algorithm 1 described in this paper, has been used to calculate response

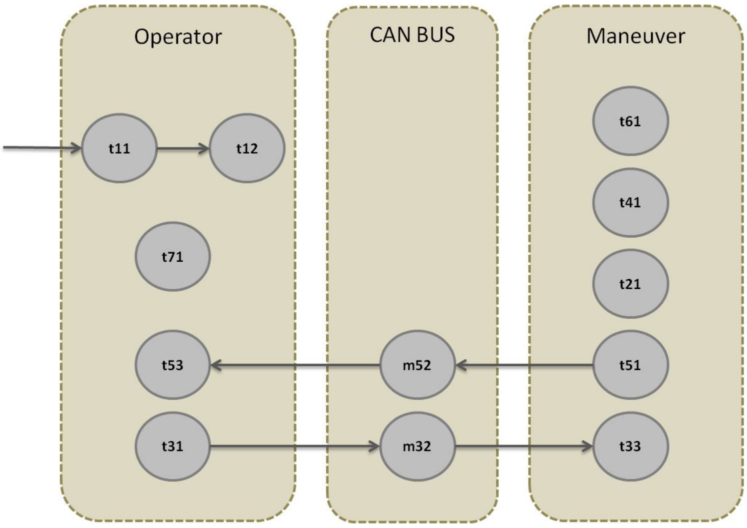


Figure 10: Door operator system's task model

times of the tasks. To calculate the response times of the messages in the CAN bus we have used the non-preemptive fixed priority technique introduced in (Tindell et al., 1994a) and revised in (Davis et al., 2007). As the scheduling policy is different in the network and the processors, the analysis technique for heterogeneous systems analysis has been used (Rivas et al., 2011).

Table 1: End-to-end flows of the system.

e2e flow	Name	Task type	T / MIT (ms)
e2eF1	obstacle	sporadic	1000.00
e2eF2	door_power	periodic	1000.00
e2eF3	op_vss	periodic	200.00
e2eF4	man_vss	periodic	100.00
e2eF5	pdo_transmission	periodic	100.00
e2eF6	man_reports	periodic	500.00
e2eF7	op_reports	periodic	500.00

The system also has shared resources represented in Table 4. The use of these resources by the tasks can be seen in Table 5.

All these elements give us a complete model to be analysed with the holistic analysis technique. After applying this technique we get the results collected in tables 7 and 6. In Table 7 we can see the upper bounds for the worst case response time bound obtained for each task. It can be noted that all the tasks have a worst case response time that is less than the corresponding deadline. Therefore, we can conclude that the system is schedulable. In addition, Table 6 provides an overview of the processor utilizations. The low utilization in the CAN bus is due to the fact that the number of messages sent through the network is very low. The utilizations of the Operator and Maneuver processors (62% and 52% respectively) reveal that the system could afford more processing load.

The results obtained in this analysis allow us to show that the holistic analysis technique can be used to analyse real industrial systems and that we can apply tools that implement it.

Table 2: Tasks of the system.

Task	Name	C (ms)	D (ms)	proc
t11	op_obstacle_isr	0.01	0.10	op
t12	op_obstacle_task	1.00	10.00	op
t21	man_powersafe_timer	1.00	10.00	man
t31	op_vss_machine_task	120.00	200.00	op
t33	man_comm_task	1.0	200.00	man
t41	man_vss_machine_task	50.00	100.00	man
t51	man_periodic_send_task	1.00	100.00	man
t53	op_comm_task	1.00	100.00	op
t61	man_reporter_task	5.00	500.00	man
t71	op_reporter_task	5.00	500.00	op

Table 3: Messages of the system.

Message	Name	C (ms)	Priority	Network
m32	msj_sdo_server	0.40	2	can
m52	msj_pdo_server	0.40	1	can

Table 4: Shared resources in the system.

Resource	Name
R1	man_events_fifo
R2	man_reports_fifo
R3	op_events_fifo
R4	op_reports_fifo

Table 5: Shared resources utilization by the tasks.

Task	R1 (ms)	R2 (ms)	R3 (ms)	R4 (ms)
t11	-	-	-	-
t12	-	-	0.07	-
t21	0.07	-	-	-
t31	-	-	0.06	1.0
t33	0.07	-	-	-
t41	0.06	1.00	-	-
t51	-	-	-	-
t53	-	-	0.07	-
t61	-	0.01	-	-
t71	-	-	-	0.01

5. Conclusions

In this paper we have presented an analysis of Spuri's holistic technique for schedulability analysis of real-time distributed systems with EDF scheduling. We have proven the correctness of the analysis by demonstrating that its iterative formulas are non-decreasing with and without the use of shared resources. In addition, we have applied this technique to the door opening/closing management subsystem of an industrial elevator. The results obtained assure the schedulability of the system and the fulfilment of the time restrictions.

One interesting conclusion of this work is that since the preemption levels of shared resources depend on release jitter terms that are only known after the analysis has been completed, the preemption levels need to change during the analysis and their final values are a result of the analysis process. This represents an extension of the original algorithm presented by Spuri.

Prototype tools implementing the techniques used in this paper have been developed to analyse the system presented in the case study section. The holistic analysis is implemented in the MAST tool suite (MAST, 2013), but it

Table 6: Processors utilization.

processor	U(%)
Operator	62.10
CAN bus	0.60
Maneuver	52.60

Table 7: Task response time bounds.

Task/Message	Name	D (ms)	R (ms)
t11	op_obstacle_isr	0.10	0.08
t12	op_obstacle_task	10.00	1.16
t21	man_powersafe_timer	10.00	1.06
t31	op_vss_machine_task	200.00	123.00
m32	msj_sdo_server	200.00	123.82
t33	man_comm_task	200.00	153.06
t41	man_vss_machine_task	100.00	53.01
t51	man_periodic_send_task	100.00	54.00
m52	msj_pdo_server	100.00	54.80
t53	op_comm_task	100.00	56.87
t61	man_reporter_task	500.00	58.00
t71	op_reporter_task	500.00	128.00

is restricted to not using shared resources because before the studies presented in this paper the applicability of the holistic technique to these systems was questioned. In this article, the correctness of the technique including the use of shared resources has been demonstrated. Thus, in the future, MAST will be extended to cover it. The extended holistic algorithm presented here will have an impact on the architecture of the MAST analysis tools, which used to have preemption levels as an input.

References

Audsley, N., Burns, A., Richardson, M., Tindell, K., Wellings, A. J., 1993. Applying new scheduling theory to static priority pre-emptive scheduling. Software Engineering Journal 8 (5), 284–292.

Baker, T. P., 1990. A stack-based resource allocation policy for realtime processes. In: Proceedings of the 11th Real-Time Systems Symposium. IEEE, pp. 191–200.

Baker, T. P., 1991. Stack-based scheduling of realtime processes. Real-Time Systems 3 (1), 67–99.

Baker, T. P., 2005. A comparison of global and partitioned edf schedulability tests for multiprocessors. In: Proceedings of International Conf. on Real-Time and Network Systems. Citeseer, pp. 119–130.

Buttazzo, G. C., 2005. Rate monotonic vs. edf: judgment day. Real-Time Systems 29 (1), 5–26.

Carpenter, J., Funk, S., Holman, P., Srinivasan, A., Anderson, J., Baruah, S., 2004. A categorization of real-time multiprocessor scheduling problems and algorithms. Handbook on Scheduling Algorithms, Methods, and Models, pages, 30.1–30.19.

Chen, M.-I., Lin, K.-J., 1990. Dynamic priority ceilings: A concurrency control protocol for real-time systems. Real-Time Systems 2 (4), 325–346. URL `http://dx.doi.org/10.1007/BF01995676`

Davis, R. I., Burns, A., Oct. 2011. A survey of hard real-time scheduling for multiprocessor systems. ACM Comput. Surv. 43 (4), 35:1–35:44. URL `http://doi.acm.org/10.1145/1978802.1978814`

Davis, R. I., Burns, A., Bril, R. J., Lukkien, J. J., 2007. Controller area network (can) schedulability analysis: Refuted, revisited and revised. Real-Time Systems 35, 239–272.

Gutierrez, M., Rivas, M. A., Burns, A., Harbour, M. G., 2014. A deadline-floor inheritance protocol for edf scheduled embedded real-time systems with resource sharing. IEEE Transactions on Computers 99 (PrePrints), 1.

Gutiérrez, J. J., González Harbour, M., 1996. Minimizing the effects of jitter in distributed hard real-time systems. Journal of systems architecture 42 (6), 431–447.

Harter Jr, P. K., 1987. Response times in level-structured systems. ACM Transactions on Computer Systems (TOCS) 5 (3), 232–248.

Joseph, M., Pandya, P., 1986. Finding response times in a real-time system. The Computer Journal 29 (5), 390–395.

Lampson, B. W., Redell, D. D., Feb. 1980. Experience with processes and monitors in mesa. Commun. ACM 23 (2), 105–117. URL `http://doi.acm.org/10.1145/358818.358824`

Liu, C. L., Layland, J. W., 1973. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM (JACM) 20 (1), 46–61.

MAST, 2013. Mast: modelling and analysis suite for real-time systems. URL `http://mast.unican.es`

Mok, A. K., 1983. Fundamental design problems of distributed systems for the hard real-time environment. Ph.D. thesis, MIT, Cambridge, Massachusetts.

Palencia, J. C., González Harbour, M., 1998. Schedulability analysis for tasks with static and dynamic offsets. In: Proceedings of the 19th Real-Time Systems Symposium. IEEE, pp. 26–37.

Palencia, J. C., González Harbour, M., 1999. Exploiting precedence relations in the schedulability analysis of distributed real-time systems. In: Proceedings of the 20th Real-Time Systems Symposium. IEEE, pp. 328–339.

Palencia, J. C., González Harbour, M., 2003. Offset-based response time analysis of distributed systems scheduled under edf. In: Proceedings of the 15th Euromicro Conference Real-Time Systems. Vol. 0. pp. 3–3.

Palencia, J. C., González Harbour, M., 2005. Response time analysis of edf distributed real-time systems. J. Embedded Comput. 1 (2), 225–237.

Palencia, J. C., Gutiérrez, J. J., González Harbour, M., 1997. On the schedulability analysis for distributed hard real-time systems. In: Proceedings of the 11th Real-Time Systems Symposium. IEEE, pp. 136–143.

Palencia, J. C., Gutiérrez, J. J., González Harbour, M., 1998. Best-case analysis for improving the worst-case schedulability test for distributed hard real-time systems. In: Proceedings of the 10th Euromicro Workshop Real-Time Systems. IEEE, pp. 35–44.

Pellizzoni, R., Lipari, G., 2007. Holistic analysis of asynchronous real-time transactions with earliest deadline scheduling. Journal of Computer and System Sciences 73 (2), 186–206.

Rajkumar, R., Sha, L., Lehoczky, J., 1988. Real-time synchronization protocols for multiprocessors. In: Proceedings of Real-Time Systems Symposium. pp. 259–269.

Rivas, J. M., Gutiérrez, J. J., Palencia, J. C., González Harbour, M., 2011. Schedulability analysis and optimization of heterogeneous edf and fp distributed real-time systems. In: Proceedings of the 23rd Euromicro Conference on Real-Time Systems. ECRTS '11. IEEE Computer Society, Washington, DC, USA, pp. 195–204.
URL `http://dx.doi.org/10.1109/ECRTS.2011.26`

Serreli, N., Lipari, G., Bini, E., 2010. The distributed deadline synchronization protocol for real-time systems scheduled by edf. In: Proceedings of Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, pp. 1–8.

Sha, L., Rajkumar, R., Lehoczky, J. P., 1990. Priority inheritance protocols: An approach to real-time synchronization. Computers, IEEE Transactions on 39 (9), 1175–1185.

Spuri, M., 1996a. Analysis of deadline scheduled real-time systems. Tech. rep., Institut national de recherche en informatique et en automatique.

Spuri, M., 1996b. Holistic analysis for deadline scheduled real-time distributed systems. Tech. rep., Institut national de recherche en informatique et en automatique, projet REFLECS RR-2873.

Sun, J., Liu, J., 1996. Synchronization protocols in distributed real-time systems. In: Proceedings of the 16th International Conference on Distributed Computing Systems. IEEE, pp. 38–45.

Tindell, K., Clark, J., 1994. Holistic schedulability analysis for distributed hard real-time systems. Microprocessing and microprogramming 40 (2), 117–134.

Tindell, K., Hansson, H., Wellings, A. J., 1994a. Analysing real-time communications: controller area network (can). In: Proceedings of Real-Time Systems Symposium. IEEE, pp. 259–263.

Tindell, K. W., Burns, A., Wellings, A. J., 1994b. An extendible approach for analyzing fixed priority hard real-time tasks. Real-Time Systems 6 (2), 133–151.