2016 IEEE Real-Time Systems Symposium

EDF-VD Scheduling of Mixed-Criticality Systems with Degraded Quality Guarantees

Di Liu[1], Jelena Spasic[1], Nan Guan[2], Gang Chen[3], Songran Liu[3], Todor Stefanov[1], Wang Yi[3, 4]
[1] Leiden University, The Netherlands
[2] Hong Kong Polytechnic University, Hong Kong
[3] Northeastern University, China
[4] Uppsala University, Sweden

***Abstract*—This paper studies real-time scheduling of mixed-criticality systems where low-criticality tasks are still guaranteed some service in the high-criticality mode, with reduced execution budgets. First, we present a utilization-based schedulability test for such systems under EDF-VD scheduling. Second, we quantify the suboptimality of EDF-VD (with our test condition) in terms of speedup factors. In general, the speedup factor is a function with respect to the ratio between the amount of resource required by different types of tasks in different criticality modes, and reaches 4/3 in the worst case. Furthermore, we show that the proposed utilization-based schedulability test and speedup factor results apply to the elastic mixed-criticality model as well. Experiments show effectiveness of our proposed method and confirm the theoretical suboptimality results.**

I. Introduction

An important trend in real-time embedded systems is to integrate applications with different criticality levels into a shared platform in order to reduce resource cost and energy consumption. To ensure the correctness of a *mixed-criticality* (MC) system, highly critical tasks are subject to certification by Certification Authorities under extremely rigorous and pessimistic assumptions [1]. This generally causes large worst-case execution time (WCET) estimation for high-criticality tasks. On the other hand, the system designer needs to consider the timing requirement of the entire system, but under less conservative assumptions. The challenge in scheduling MC systems is to simultaneously guarantee the timing correctness of (1) only high-criticality tasks under very pessimistic assumptions, and (2) all tasks, including low-critical ones, under less pessimistic assumptions.

The scheduling problem of MC systems has been intensively studied in recent years (see Section II for a brief review). In most of previous works, the timing correctness of high-criticality tasks are guaranteed in the worst case scenario at the expense of low-criticality tasks. More specifically, when any high criticality task executes for longer than its low-criticality WCET (and thus the system enters the high-criticality mode), all low-criticality tasks will be completely discarded and the resource are all dedicated for meeting the timing constraints of high-criticality tasks [2]–[5]. However, such an approach seriously disturbs the service of low-criticality tasks. This is not acceptable in many practical problems, especially for control systems where the performance of controllers mainly depends on the execution frequency of control tasks [6].

To overcome this problem, Burns and Baruah in [7] introduced an MC task model where low-criticality tasks reduce their execution budgets such that low-criticality tasks are guaranteed to be scheduled in high-criticality mode with regular execution frequency (i.e., the same period) but degraded quality[1]. Since the idea of reducing execution budgets to keep tasks running is conceptually similar to the *imprecise computation model* [8] [9], in this paper we call an MC system with possibly reduced execution budgets of low criticality tasks an *imprecise mixed criticality* (IMC) system.

In [7], the authors consider preemptive fixed-priority scheduling for the IMC system model and extend the adaptive mixed criticality (AMC) [3] approach to provide a schedulability test for the IMC model. In this paper, we study the EDF-VD scheduling of IMC systems. EDF-VD is designed for the classical MC system model, in which EDF algorithm is enhanced by deadline adjustment mechanisms to compromise the resource requirement on different criticality levels. EDF-VD has shown strong competence by both theoretical and empirical evaluations [2], [4], [5]. For example, [2] proves that EDF-VD is a speedup-optimal MC scheduling and in [4], [5] experimental evaluations show that EDF-VD outperforms other MC scheduling algorithms in terms of acceptance ratio. The main technical contributions of this paper include

- We propose a sufficient test for the IMC model under EDF-VD, - see Theorem 3 in Section IV;
- For the IMC model under EDF-VD, we derive a speedup factor function with respect to the utilization ratios of high criticality tasks and low criticality tasks - see Theorem 4 in Section V. The derived speedup factor function enables us to quantify the suboptimality of EDF-VD and evaluate the impact of the utilization ratios on the speedup factor. We also compute the maximum value $4/3$ of the speedup factor function, which is equal to the speedup factor bound for the classical MC model [2].
- With extensive experiments, we show that for the IMC model, by using our proposed sufficient test, in most cases EDF-VD outperforms AMC [7] in terms of the number of schedulable task sets. Moreover, the experimental results validate the observations we obtained for speedup factor.

[1]In [8] [9], the output quality of a task is related to its execution time. The longer a task executes, the better quality results it produces.

978-1-5090-5303-2/16 $31.00 © 2016 IEEE
DOI 10.1109/RTSS.2016.10

Moreover, the schedulability test and speedup factor results of this paper also apply to the *elastic mixed-criticality* (EMC) model proposed in [6], where the periods of low-criticality tasks are scaled up in high-criticality mode, see in Section VI.

The remainder of this paper is organized as follows: Section II discusses the related work. Section III gives the preliminaries and describes the IMC task model and its execution semantics. Section IV presents our sufficient test for the IMC model and Section V derives the speedup factor function for the IMC under EDF-VD. Section VI extends the proposed sufficient test to the EMC model. Finally, Section VII shows our experimental results and Section VIII concludes this paper.

II. Related Work

Burns and Davis in [10] give a comprehensive review of work on real-time scheduling for MC systems. Many of these literatures, e.g., [2] [4] [5], consider the classical MC model in which all low criticality tasks are discarded if the system switches to the high criticality mode. In [7], Burns and Baruah discuss three approaches to keep some low criticality tasks running in *high*-criticality mode. The first approach is to change the priority of low criticality tasks. However, for fixed-priority scheduling, deprioritizing low criticality tasks cannot guarantee the execution of the low criticality tasks with a short deadline after the mode switches. [7]. Similarly, for EDF, lowering priority of low criticality tasks leads to a degraded service [11]. In this paper, we consider the IMC model which improves the schedulability of low criticality tasks in *high*-criticality mode by reducing their execution time. The IMC model can guarantee the regular service of a system by trading off the quality of the produced results. For some applications given in [8] [9] [12], such trade-off is preferred. The second approach in [7] is to extend the periods of low criticality tasks when the system mode changes to *high*-criticality mode such that the low criticality tasks execute less frequently to ensure their schedulability. Su *et al.* [6] [13] and Jan *et al.* [14] both consider this model. However, some applications might prefer an on-time result with a degraded quality rather than a delayed result with a perfect quality. Some example applications can be seen in [15] [8] [9]. Then, the approach of extending periods is less useful for this kind of applications. The last approach proposed in [7] is to reduce the execution budget of low criticality tasks when the system mode switches, i.e., the use of the IMC model studied in this paper. In [7], the authors extend the AMC [3] approach to test the schedulability of an IMC task set under fixed-priority scheduling. However, the schedulability problem for an IMC task set under EDF-VD [2], has not yet been addressed. Therefore, in this paper, we study the schedulability of the IMC task model under EDF-VD and propose a sufficient test for it.

III. Preliminaries

This section first introduces the IMC task model and its execution semantics. Then, we give a brief explanation for EDF-VD scheduling [2] and an example to illustrate the execution semantics of the IMC model under EDF-VD scheduling.

A. Imprecise Mixed-Criticality Task Model

We use the *implicit-deadline sporadic* task model given in [7] where a task set γ includes n tasks which are scheduled on a uniprocessor. Without loss of generality, all tasks in γ are assumed to start at time 0. Each task τ_i in γ generates an infinite sequence of jobs $\{J_i^1, J_i^2...\}$ and is characterized by $\tau_i = \{T_i, D_i, L_i, \mathcal{C}_i\}$:

- T_i is the period or the minimal separation interval between two consecutive jobs;
- D_i denotes the relative task deadline, where $D_i = T_i$;
- $L_i \in \{LO, HI\}$ denotes the criticality (*low or high*) of a task. In this paper, like in many previous research works [6] [11] [2] [4] [5], we consider a dual-criticality MC model. Then, we split tasks into two task sets, $\gamma_{LO} = \{\tau_i | L_i = LO\}$ and $\gamma_{HI} = \{\tau_i | L_i = HI\}$;
- $\mathcal{C}_i = \{C_i^{LO}, C_i^{HI}\}$ is a list of WCETs, where C_i^{LO} and C_i^{HI} represent the WCET in *low*-criticality mode and the WCET in *high*-criticality mode, respectively. For a *high*-criticality task, it has $C_i^{LO} \leq C_i^{HI}$, whereas $C_i^{LO} \geq C_i^{HI}$ *for a low-criticality task, i.e., low-criticality task* τ_i *has a reduced WCET in high-criticality mode*.

Then each job J_i is characterized by $J_i = \{a_i, d_i, L_i, \mathcal{C}_i\}$, where a_i is the absolute release time and d_i is the absolute deadline. Note that if *low*-criticality task τ_i has $C_i^{HI} = 0$, it will be immediately discarded at the time of the switch to *high*-criticality mode. In this case, the IMC model behaves like the classical MC model.

The utilization of a task is used to denote the ratio between its WCET and its period. We define the following utilizations for an IMC task set γ:

- For every task τ_i, it has $u_i^{LO} = \frac{C_i^{LO}}{T_i}$, $u_i^{HI} = \frac{C_i^{HI}}{T_i}$;
- For all *low*-criticality tasks, we have total utilizations
$$U_{LO}^{LO} = \sum_{\forall \tau_i \in \gamma_{LO}} u_i^{LO}, \;\; U_{LO}^{HI} = \sum_{\forall \tau_i \in \gamma_{LO}} u_i^{HI}$$
- For all *high*-criticality tasks, we have total utilizations
$$U_{HI}^{LO} = \sum_{\forall \tau_i \in \gamma_{HI}} u_i^{LO}, \;\; U_{HI}^{HI} = \sum_{\forall \tau_i \in \gamma_{HI}} u_i^{HI}$$
- For an IMC task set, we have
$$U^{LO} = U_{LO}^{LO} + U_{HI}^{LO}, \quad U^{HI} = U_{LO}^{HI} + U_{HI}^{HI}$$

B. Execution Semantics of the IMC Model

The execution semantics of the IMC model are similar to those of the classical MC model. The **major difference** occurs after a system switches to *high*-criticality mode. *Instead of discarding all low-criticality tasks, as it is done in the classical MC model, the IMC model tries to schedule low-criticality tasks with their reduced execution times* C_i^{HI}. The execution semantics of the IMC model are summarized as follows:

- The system starts in *low*-criticality mode, and remains in this mode as long as no *high*-criticality job overruns its *low*-criticality WCET C_i^{LO}. If any job of a *low*-criticality task tries to execute beyond its C_i^{LO}, the system will suspend it and launch a new job at the next period;

Task	L	C_i^{LO}	C_i^{HI}	T_i	$\hat{D}_i$
τ_1	LO	4	2	9	
τ_2	HI	4	8	10	7

Table I: Illustrative example

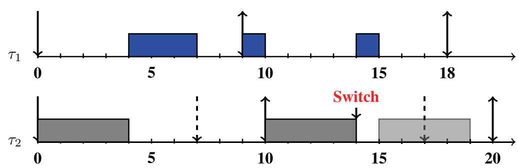


Figure 1: Scheduling of Example I

- If any job of *high*-criticality task executes for its C_i^{LO} time units without signaling completion, the system immediately switches to *high*-criticality mode;
- As the system switches to *high*-criticality mode, *if jobs of low-criticality tasks have completed execution for more than their C_i^{HI} but less than their C_i^{LO}, the jobs will be suspended till the tasks release new jobs for the next period. However, if jobs of low-criticality tasks have not completed their C_i^{HI} ($\leq C_i^{LO}$) by the switch time instant, the jobs will complete the left execution to C_i^{HI} after the switch time instant and before their deadlines. Hereafter, all jobs are scheduled using C_i^{HI}.* For *high*-criticality tasks, if their jobs have not completed their C_i^{LO} ($\leq C_i^{HI}$) by the switch time instant, all jobs will continue to be scheduled to complete C_i^{HI}. After that, all jobs are scheduled using C_i^{HI}.

Santy *et al.* [16] have shown that the system can switch back from the *high*-criticality mode to the *low*-criticality mode when there is an idle period and no *high*-criticality job awaits for execution. For the IMC model, we can use the same scenario to trigger the switch-back. In this paper, we focus on the switch from *low*-criticality mode to *high*-criticality mode.

C. EDF-VD Scheduling

The challenge to schedule MC tasks with EDF scheduling algorithm [17] is to deal with the overrun of *high*-criticality tasks when the system switches from *low*-criticality mode to *high*-criticality mode. Baruah *et al.* in [2] proposed to artificially tighten deadlines of jobs of *high*-criticality tasks in *low*-criticality mode such that the system can preserve execution budgets for the *high*-criticality tasks across mode switches. This approach is called *EDF with virtual deadlines* (EDF-VD).

D. An Illustrative Example

Here, we give a simple example to illustrate the execution semantics of the IMC model under EDF-VD. Table I gives two tasks, one *low*-criticality task τ_1 and one *high*-criticality task τ_2, where $\hat{D}_i$ is the virtual deadline. Figure 1 depicts the scheduling of the given IMC task set, where we assume that the mode switch occurs in the second period of τ_2. When the system switches to *high*-criticality mode, τ_2 will be scheduled by its original deadline 10 instead of its virtual deadline 7. Hence, τ_1 preempts τ_2 at the switch time instant. Since in *high*-criticality mode τ_1 only has execution budget of 2 , i.e., C_1^{HI}, τ_1 executes one unit and suspends. Then, τ_2 completes its left execution 4 ($C_2^{HI} - C_2^{LO}$) before its deadline.

IV. Schedulability Analysis

In [7], an AMC-based schedulability test for the IMC model under fixed priority scheduling has been proposed. However, to date, a schedulability test for the IMC model under EDF-VD has not been addressed yet. Therefore, inspired by the work in [2] for the classical MC model, we propose a sufficient test for the IMC model under EDF-VD.

A. Low Criticality Mode

We first ensure the schedulability of tasks when they are in *low*-criticality mode. As the task model is in *low*-criticality mode, the tasks can be considered as traditional real-time tasks scheduled by EDF algorithm with virtual deadlines (VD). The following theorem is given in [2] for tasks scheduled in *low*-criticality mode.

Theorem 1 (Theorem 1 from [2])**.** *The following condition is sufficient for ensuring that EDF-VD successfully schedules all tasks in low-criticality mode:*

$$1 \geq \frac{U_{HI}^{LO}}{x} + U_{LO}^{LO} \tag{1}$$

where $x \in (0, 1)$ is used to uniformly modify the relative deadline of high-criticality tasks.

Since the IMC model behaves as the classical MC model in *low*-criticality mode, Theorem 1 holds for the IMC model as well.

B. High Criticality Mode

For *high*-criticality mode, the classical MC model discards all *low*-criticality jobs after the switch to *high*-criticality mode. In contrast, the IMC model keeps *low*-criticality jobs running but with degraded quality, i.e., a shorter execution time. So the schedulability condition in [2] does not work for the IMC model in the *high*-criticality mode. Thus, we need a new test for the IMC model in *high*-criticality mode.

To derive the sufficient test in *high*-criticality mode, suppose that there is a time interval $[0, t_2]$, where a first deadline miss occurs at t_2 and t_1 denotes the time instant of the switch to *high*-criticality mode in the time interval, where $t_1 < t_2$. Assume that $\mathcal{J}$ is the minimal set of jobs generated from task set γ which leads to the first deadline miss at t_2. The minimality of $\mathcal{J}$ means that removing any job in $\mathcal{J}$ guarantees the schedulability of the rest of $\mathcal{J}$. Here, we introduce some notations for our later interpretation. Let variable η_i denote the cumulative execution time of task τ_i in the interval $[0, t_2]$. J_1 denotes a special *high*-criticality job which has switch time instant t_1 within its period (a_1, d_1), i.e, $a_1 < t_1 < d_1$. Furthermore, J_1 is the job with the earliest release time amongst all *high*-criticality jobs in $\mathcal{J}$ which execute in $[t_1, t_2)$. Moreover, we define a special type of job for *low*-criticality tasks which is useful for our later proofs.

Definition 1. *A job J_i from low-criticality task τ_i is a carry-over job, if its absolute release time a_i is before and its* *absolute deadline d_i is after the switch time instant, i.e., $a_i < t_1 < d_i$.*

With the notations introduced above, we have the following propositions,

Proposition 1 (Fact 1 from [2])**.** *All jobs in $\mathcal{J}$ that execute in $[t_1, t_2)$ have deadline $\leq t_2$.*

It is easy to observe that only jobs which have deadlines $\leq t_2$ are possible to cause a deadline miss at t_2. If a job has its deadline $> t_2$ and is still in set $\mathcal{J}$, it will contradict the minimality of $\mathcal{J}$.

Proposition 2. *The switch time instant t_1 has*

$$t_1 < (a_1 + x(t_2 - a_1)) \tag{2}$$

Proof: Let us consider a time instant $(a_1 + x(d_1 - a_1))$ which is the virtual deadline of job J_1. Since J_1 executes in time interval $[t_1, t_2)$, its virtual deadline $(a_1 + x(d_1 - a_1))$ must be greater than the switch time instant t_1. Otherwise, it should have completed its *low*-criticality execution before t_1, and this contradicts that it executes in $[t_1, t_2)$. Thus, it has

$$\begin{aligned} &t_1 < (a_1 + x(d_1 - a_1)) \\ \Rightarrow &t_1 < (a_1 + x(t_2 - a_1)) \quad (\text{since } d_1 \leq t_2) \end{aligned}$$

■

Proposition 3. *If a carry-over job J_i has its cumulative execution equal to $(d_i - a_i)u_i^{LO}$ and $u_i^{LO} > u_i^{HI}$, its deadline d_i is $\leq (a_1 + x(t_2 - a_1))$.*

Proof: For a carry-over job J_i, if it has its cumulative execution equal to $(d_i - a_i)u_i^{LO}$ and $u_i^{LO} > u_i^{HI}$, it should complete its C_i^{LO} execution before t_1. Otherwise, if job J_i has executed time units $C_i \in [C_i^{HI}, C_i^{LO})$ at time instant t_1, it will be suspended and will not execute after t_1.

Now, we will show that when job J_i completes its C_i^{LO} execution, its deadline is $d_i \leq (a_1 + x(t_2 - a_1))$. We prove this by contradiction. First, we suppose that J_i has its deadline $d_i > (a_1 + x(t_2 - a_1))$ and release time a_i. As shown above, job J_i completes its C_i^{LO} execution before t_1. Let us assume a time instant t^* as the latest time instant at which this carry-over job J_i starts to execute before t_1. This means that at this time instant all jobs in $\mathcal{J}$ with deadline $\leq (a_1 + x(t_2 - a_1))$ have finished their executions. This indicates that these jobs will not have any execution within interval $[t^*, t_2]$. Therefore, jobs in $\mathcal{J}$ with release time at or after time instant t^* can form a smaller job set which causes a deadline miss at t_2. Then, it contradicts the minimality of $\mathcal{J}$. Thus, carry-over job J_i with its cumulative execution time equal to $(d_i - a_i)u_i^{LO}$ and $u_i^{LO} > u_i^{HI}$ has its deadline $d_i \leq (a_1 + x(t_2 - a_1))$. ■

With the propositions and notations given above, we derive an upper bound of the cumulative execution time η_i of *low*-criticality task τ_i.

Lemma 1. *For any low-criticality task τ_i, it has*

$$\eta_i \leq (a_1 + x(t_2 - a_1))u_i^{LO} + (1 - x)(t_2 - a_1)u_i^{HI} \tag{3}$$

Proof: If $u_i^{LO} = u_i^{HI}$, it is trivial to see that Lemma 1 holds. Below we focus on the case when $u_i^{LO} > u_i^{HI}$. If a system switches to *high*-criticality mode at t_1, then we know that *low*-criticality tasks are scheduled using C_i^{LO} before t_1 and using C_i^{HI} after t_1. To prove this lemma, we need to consider two cases, where τ_i releases a job within interval $(a_1, t_2]$ or it does not. We prove the two cases separately.

Case A (task τ_i releases a job within interval $(a_1, t_2]$): There are two sub-cases to be considered.

- **Sub-case 1 (No carry-over job):** The deadline of a job of *low*-criticality task τ_i coincides with switch time instant t_1. The cumulative execution time of *low*-criticality task τ_i within time interval $[0, t_2]$ can be bounded as follows,

  $$\eta_i \leq (t_1 - 0) \cdot u_i^{LO} + (t_2 - t_1) \cdot u_i^{HI}$$

  Since $t_1 < (a_1 + x(t_2 - a_1))$ according to Proposition 2 and for *low*-criticality task τ_i it has $u_i^{LO} > u_i^{HI}$, then

  $$\begin{aligned} &\eta_i < \big(a_1 + x(t_2 - a_1)\big)u_i^{LO} + \big(t_2 - \big(a_1 + x(t_2 - a_1)\big)\big)u_i^{HI} \\ \Leftrightarrow &\eta_i < (a_1 + x(t_2 - a_1))u_i^{LO} + (1 - x)(t_2 - a_1)u_i^{HI} \end{aligned}$$

- **Sub-case 2 (with carry-over job):** In this case, before the carry-over job, jobs of τ_i are scheduled with its C_i^{LO}. After the carry-over job, jobs of τ_i are scheduled with its C_i^{HI}. It is trivial to observe that for a carry-over job its maximum cumulative execution time can be obtained when it completes its C_i^{LO} within its period $[a_i, d_i]$, i.e., $(d_i - a_i)u_i^{LO}$. Considering the maximum cumulative execution for the carry-over job, we then have for *low*-criticality task τ_i,

  $$\begin{aligned} &\eta_i \leq (a_i - 0)u_i^{LO} + (d_i - a_i)u_i^{LO} + (t_2 - d_i)u_i^{HI} \\ \Leftrightarrow &\eta_i \leq d_i u_i^{LO} + (t_2 - d_i)u_i^{HI} \end{aligned}$$

  Proposition 3 shows as J_i has its cumulative execution equal to $(d_i - a_i) \cdot u_i^{LO}$, it has $d_i \leq (a_1 + x(t_2 - a_1))$. Given $u_i^{LO} > u_i^{HI}$ for *low*-criticality task, we have

  $$\begin{aligned} &\eta_i \leq d_i u_i^{LO} + (t_2 - d_i)u_i^{HI} \\ \Rightarrow &\eta_i \leq \big(a_1 + x(t_2 - a_1)\big)u_i^{LO} + \big(t_2 - \big(a_1 + x(t_2 - a_1)\big)\big)u_i^{HI} \\ \Leftrightarrow &\eta_i \leq (a_1 + x(t_2 - a_1))u_i^{LO} + (1 - x)(t_2 - a_1)u_i^{HI} \end{aligned}$$

Case B (task τ_i does not release a job within interval $(a_1, t_2]$): In this case, let J_i denote the last release job of task τ_i before a_1 and a_i and d_i are its absolute release time and absolute deadline, respectively. If $d_i \leq t_1$, we have

$$\eta_i = (a_i - 0)u_i^{LO} + (d_i - a_i) \cdot u_i^{LO} = d_i u_i^{LO}$$

If $d_i > t_1$, J_i is a carry-over job. As we discussed above, the maximum cumulative execution time of carry-over job J_i is $(d_i - a_i)u_i^{LO}$, so we have

$$\eta_i \leq (a_i - 0)u_i^{LO} + (d_i - a_i) \cdot u_i^{LO} \Leftrightarrow \eta_i \leq d_i u_i^{LO}$$

Similarly, according to Proposition 3, we obtain,

$$\begin{aligned} &\eta_i \leq d_i \cdot u_i^{LO} \leq (a_1 + x(t_2 - a_1))u_i^{LO} \\ \Rightarrow &\eta_i < (a_1 + x(t_2 - a_1))u_i^{LO} + \big(t_2 - \big(a_1 + x(t_2 - a_1)\big)\big)u_i^{HI} \\ \Leftrightarrow &\eta_i < (a_1 + x(t_2 - a_1))u_i^{LO} + (1 - x)(t_2 - a_1)u_i^{HI} \end{aligned}$$

■

Lemma 1 gives the upper bound of the cumulative execution time of a *low*-criticality task in *high*-criticality mode. In order to derive the sufficient test for the IMC model in *high*-criticality mode, we need to upper bound the cumulative execution time of *high*-criticality tasks.

Proposition 4 (Fact 3 from [2])**.** *For any high-criticality task* τ_i*, it holds that*

$$\eta_i \leq \frac{a_1}{x} u_i^{LO} + (t_2 - a_1) u_i^{HI} \tag{4}$$

Proposition 4 is used to bound the cumulative execution of the *high*-criticality tasks. Since in the IMC model the *high*-criticality tasks are scheduled as in the classical MC model, Proposition 4 holds for the IMC model as well. With Lemma 1 and Proposition 4, we can derive the sufficient test for the IMC model in *high*-criticality mode.

Theorem 2. *The following condition is sufficient for ensuring that EDF-VD successfully schedules all tasks in high-criticality mode:*

$$xU_{LO}^{LO} + (1-x)U_{LO}^{HI} + U_{HI}^{HI} \leq 1 \tag{5}$$

Proof: Let N denote the cumulative execution time of all tasks in $\gamma = \gamma_{LO} \cup \gamma_{HI}$ over interval $[0, t_2]$. We have

$$N = \sum_{\forall \tau_i \in \gamma_{LO}} \eta_i + \sum_{\forall \tau_i \in \gamma_{HI}} \eta_i$$

By using Lemma 1 and Proposition 4, N is bounded as follows

$$\begin{aligned} N \leq & \sum_{\forall \tau_i \in \gamma_{LO}} \Big(\big(a_1 + x(t_2 - a_1)\big) u_i^{LO} + (1-x)(t_2 - a_1) u_i^{HI} \Big) \\ & + \sum_{\forall \tau_i \in \gamma_{HI}} \left(\frac{a_1}{x} u_i^{LO} + (t_2 - a_1) u_i^{HI} \right) \\ \Leftrightarrow & N \leq (a_1 + x(t_2 - a_1)) U_{LO}^{LO} + (1-x)(t_2 - a_1) U_{LO}^{HI} \\ & + \frac{a_1}{x} U_{HI}^{LO} + (t_2 - a_1) U_{HI}^{HI} \\ \Leftrightarrow & N \leq a_1 (U_{LO}^{LO} + \frac{U_{HI}^{LO}}{x}) + x(t_2 - a_1) U_{LO}^{LO} \\ & + (1-x)(t_2 - a_1) U_{LO}^{HI} + (t_2 - a_1) U_{HI}^{HI} \end{aligned} \tag{6}$$

Since the tasks must be schedulable in *low*-criticality mode, the condition given in Theorem 1 holds and we have $1 \geq (U_{LO}^{LO} + \frac{U_{HI}^{LO}}{x})$. Hence,

$$\begin{aligned} N \leq & a_1 + x(t_2 - a_1) U_{LO}^{LO} \\ & + (1-x)(t_2 - a_1) U_{LO}^{HI} + (t_2 - a_1) U_{HI}^{HI} \end{aligned} \tag{7}$$

Since time instant t_2 is the first deadline miss, it means that there is no idle time instant within interval $[0, t_2]$. Note that if there is an idle instant, jobs from set $\mathcal{J}$ which have release time at or after the latest idle instant can form a smaller job set causing deadline miss at t_2 which contradicts the minimality of $\mathcal{J}$. Then, we obtain

$$\begin{aligned} & N = \left(\sum_{\forall \tau_i \in \gamma_{LO}} \eta_i + \sum_{\forall \tau_i \in \gamma_{HI}} \eta_i \right) > t_2 \\ \Rightarrow & a_1 + x(t_2 - a_1) U_{LO}^{LO} + (1-x)(t_2 - a_1) U_{LO}^{HI} + (t_2 - a_1) U_{HI}^{HI} \\ & > t_2 \\ \Leftrightarrow & x(t_2 - a_1) U_{LO}^{LO} + (1-x)(t_2 - a_1) U_{LO}^{HI} + (t_2 - a_1) U_{HI}^{HI} \\ & > t_2 - a_1 \\ \Leftrightarrow & xU_{LO}^{LO} + (1-x) U_{LO}^{HI} + U_{HI}^{HI} > 1 \end{aligned}$$

By taking the contrapositive, we derive the sufficient test for the IMC model when it is in *high*-criticality mode:

$$xU_{LO}^{LO} + (1-x)U_{LO}^{HI} + U_{HI}^{HI} \leq 1$$

■

Note that if $U_{LO}^{HI} = 0$, i.e., no *low*-criticality tasks are scheduled after the system switches to *high*-criticality mode, our Theorem 2 is the same as the sufficient test (Theorem 2 in [2]) for the classical MC model in *high*-criticality mode. Hence, our Theorem 2 actually is a generalized schedulability condition for (I)MC tasks under EDF-VD.

By combining Theorem 1 (see Section IV-A) and our Theorem 2, we prove the following theorem,

Theorem 3. *Given an IMC task set, if*

$$U_{HI}^{HI} + U_{LO}^{LO} \leq 1 \tag{8}$$

then the IMC task set is schedulable by EDF; otherwise, if

$$\frac{U_{HI}^{LO}}{1 - U_{LO}^{LO}} \leq \frac{1 - (U_{HI}^{HI} + U_{LO}^{HI})}{U_{LO}^{LO} - U_{LO}^{HI}} \tag{9}$$

where

$$U_{HI}^{HI} + U_{LO}^{HI} < 1 \text{ and } U_{LO}^{LO} < 1 \text{ and } U_{LO}^{LO} > U_{LO}^{HI} \tag{10}$$

then this IMC task set can be scheduled by EDF-VD with a deadline scaling factor x *arbitrarily chosen in the following range*

$$x \in \left[\frac{U_{HI}^{LO}}{1 - U_{LO}^{LO}}, \ \frac{1 - (U_{HI}^{HI} + U_{LO}^{HI})}{U_{LO}^{LO} - U_{LO}^{HI}} \right]$$

Proof: Total utilization $U \leq 1$ is the exact test for EDF on a uniprocessor system. If the condition in (8) is met, the given task set is *worst-case reservation* [2] schedulable under EDF, i.e., the task set can be scheduled by EDF without deadline scaling for *high*-criticality tasks and execution budget reduction for *low*-criticality tasks. Now, we prove the second condition given by (9). From Theorem 1, we have,

$$x \geq \frac{U_{HI}^{LO}}{1 - U_{LO}^{LO}}$$

From Theorem 2, we have

$$\begin{aligned} & xU_{LO}^{LO} + (1-x)U_{LO}^{HI} + U_{HI}^{HI} \leq 1 \\ \Leftrightarrow & x \leq \frac{1 - (U_{HI}^{HI} + U_{LO}^{HI})}{U_{LO}^{LO} - U_{LO}^{HI}} \end{aligned}$$

Therefore, if $\frac{U_{HI}^{LO}}{1 - U_{LO}^{LO}} \leq \frac{1 - (U_{HI}^{HI} + U_{LO}^{HI})}{U_{LO}^{LO} - U_{LO}^{HI}}$, the schedulability conditions of both Theorem 1 and 2 are satisfied. Thus, the IMC tasks are schedulable under EDF-VD. ■

V. Speedup Factor

The speedup factor bound is a useful metric to compare the worst-case performance of different MC scheduling algorithms. The speedup factor bound for the classical MC model under EDF-VD [2] has been shown to be $4/3$. The following is the definition of the speedup factor for an MC scheduling algorithm.

Definition 2 (from [2])**.** *The speedup factor of an algorithm $\mathcal{A}$ for scheduling MC systems is the smallest real number $f \geq 1$ such that any task system that is schedulable by a hypothetical optimal clairvoyant scheduling algorithm[2] on a unit-speed processor is correctly scheduled by algorithm $\mathcal{A}$ on a speed-f processor.*

For notational simplicity, we define

$$U_{HI}^{HI} = c, \quad U_{HI}^{LO} = \alpha \times c$$
$$U_{LO}^{LO} = b, \quad U_{LO}^{HI} = \lambda \times b$$

where $\alpha \in (0, 1]$ and $\lambda \in [0, 1]$. α denotes the utilization ratio between U_{HI}^{LO} and U_{HI}^{HI}, while λ denotes the utilization ratio between U_{LO}^{HI} and U_{LO}^{LO}.

First, let us analyze the speedup factor of two corner cases. When $\alpha = 1$, i.e., $U_{HI}^{LO} = U_{HI}^{HI}$, this means that there is no mode-switch. Therefore, the task set is scheduled by the traditional EDF, i.e., the task set is schedulable if $U_{LO}^{LO} + U_{HI}^{LO} \leq 1$. Since EDF is the optimal scheduling algorithm on a uniprocessor system, the speedup factor is 1. When $\lambda = 1$, i.e., $U_{LO}^{LO} = U_{LO}^{HI}$, if the task set is schedulable in *high*-criticality mode, it must hold $U_{HI}^{HI} + U_{LO}^{LO} \leq 1$ by Theorem 2. Then it is scheduled by the traditional EDF and thus the speedup factor is 1 as well.

In this paper, instead of generating a single speedup factor bound, we derive a speedup factor function with respect to (α, λ). This speedup factor function enables us to quantify the suboptimality of EDF-VD for the IMC model in terms of speedup factor (by our proposed sufficient test) and evaluate the impact of the utilization ratio on the schedulability of an IMC task set under EDF-VD.

First, we strive to find a minimum speed s ($\leq$1) for a clairvoyant optimal MC scheduling algorithm such that any implicit-deadline IMC task set which is schedulable by the clairvoyant optimal MC scheduling algorithm on a speed-s processor can satisfy the schedulability test given in Theorem 3, i.e., schedulable under EDF-VD on a unit-speed processor.

Lemma 2. *Given $b, c \in [0, 1]$, $\alpha \in (0, 1)$, $\lambda \in [0, 1)$, and*

$$\max\{b + \alpha c, \lambda b + c\} \leq S(\alpha, \lambda) \tag{11}$$

where

$$S(\alpha, \lambda) = \frac{(1 - \alpha\lambda)((2 - \alpha\lambda - \alpha) + (\lambda - 1)\sqrt{4\alpha - 3\alpha^2})}{2(1 - \alpha)(\alpha\lambda - \alpha\lambda^2 - \alpha + 1)}$$

then it guarantees

$$\frac{\alpha c}{1 - b} \leq \frac{1 - (c + \lambda b)}{b - \lambda b} \tag{12}$$

[2] A 'clairvoyant' scheduling algorithm knows all run-time information, e.g., when the mode switch will occur, prior to run-time.

Proof: Suppose that λ and α are constants and we have a real number $s \leq 1$, where $\max\{b+\alpha c, \lambda b+c\} \leq s$. We need to find the minimum of s which guarantees that *any* $b, c \in [0, 1]$ ensure (12). First, $\max\{b + \alpha c, \lambda b + c\} \leq s$ implies

$$b + \alpha c \leq s \tag{13}$$

$$\lambda b + c \leq s \tag{14}$$

Then, condition (12) can be written as follows,

$$\lambda b^2 + (\alpha\lambda - \alpha + 1)bc - (\lambda + 1)b - c + 1 \geq 0 \tag{15}$$

Inequalities (13)(14)(15) define a feasible space in three-dimension space, respectively. In Figure 2(a), the space above the plane is a feasible space satisfying (13), where the plane corresponds to $b + \alpha c = s$. For (14), $\lambda b + c = s$ draws a plane and the feasible space is above the plane shown in Figure 2(b). Similarly, when (15) makes its right-hand-side equal to the left-hand-side, we draw a vertical curved surface seen in Figure 2(c) and the space inside the vertical surface is the feasible space (the opposite of the arrow direction). We need to find the *minimum* of s in the feasible space (above the two planes and inside the vertical surface) such that *any* b and c that meet (11) satisfy (12). Since $\max\{b + \alpha c, \lambda b + c\} = s$ is strictly increasing, to ensure (11)(12) hold for *any* b and c, we somehow *maximize* s in the feasible space. This problem can be transformed into another form, where, instead of maximizing s inside of the vertical surface, we can find the *minimum* value of s in the space outside the vertical surface[3] which is defined by

$$\lambda b^2 + (\alpha\lambda - \alpha + 1)bc - (\lambda + 1)b - c + 1 \leq 0 \tag{16}$$

Then, the minimization problem is formulated as follows,

$$
\begin{aligned}
\text{minimize} \quad & s && (17)\\
\text{subject to} \quad & b + \alpha c \leq s && (18)\\
& \lambda b + c \leq s && (19)\\
& \lambda b^2 + (\alpha\lambda - \alpha + 1)bc - (\lambda + 1)b - c + 1 \leq 0 && (20)\\
& 0 \leq b \leq 1, \quad 0 \leq c \leq 1 && (21)
\end{aligned}
$$

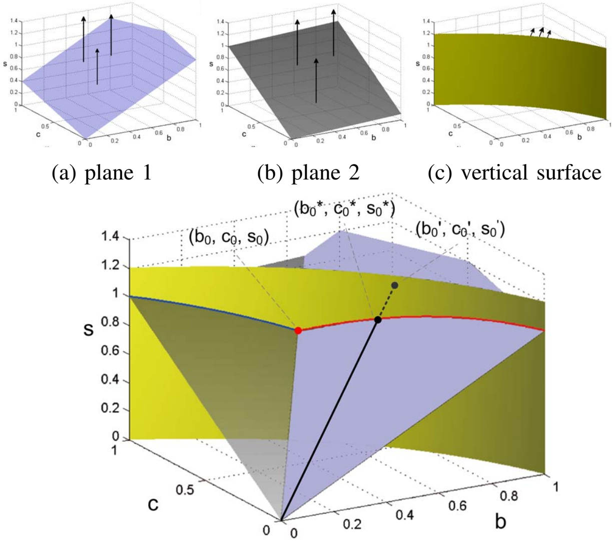
where α and λ are constant and s, b, c are variables. If $S(\alpha, \lambda)$ is the optimal solution of the optimization problem (17), then Lemma 2 is proved.

Below, we prove that $S(\alpha, \lambda)$ is the optimal solution of the optimization problem (17)[4].

As stated before, the feasible solutions subject to these three constraints must be above both planes and outside the vertical curved surface. First assume that we have a point (b_0', c_0', s_0') which satisfies all constraints but is not on the vertical surface. If we connect the origin $(0, 0, 0)$ and (b_0', c_0', s_0'), this line must have an intersection point (b_0^*, c_0^*, s_0^*) with the vertical surface. It is easy to observe that $s_0^* < s_0'$ - see in Figure 2(d). This means that any point which is not on the vertical surface can find a point with smaller value of s on the vertical surface

[3] As the arrows direct

[4] This optimization problem is a non-convex problem and thus we cannot use general convex optimization techniques such as the Karush-Kuhn-Tucker (KKT) approach [18] to solve it.



(a) plane 1 (b) plane 2 (c) vertical surface

(d) Feasible solution space

Figure 2: 3D space of optimization problem (17)

which satisfies all constraints. Therefore, the point with the minimum s must be on the vertical surface. Similarly, the minimum s must be on one of the two planes. Otherwise, if it is not on any plane, we always can find a projected point on one plane which has a smaller value of s.

We have shown above that to obtain the minimum value of s the point must be on the vertical surface and one plane. Then, the two planes have an intersection line and this line intersects with the vertical surface at a point denoted by (b_0, c_0, s_0). By taking constraints (18)(19) and (20), we formulate a piece-wise function of s with respect to b as follows.

$$s(b) = \begin{cases} \frac{(\alpha\lambda^2-\alpha\lambda)b^2+b-1}{(\alpha\lambda-\alpha+1)b-1} & 0 < b \le b_0 \\ \frac{(1-\alpha)b^2+(\alpha\lambda+\alpha-1)b-\alpha}{(\alpha\lambda-\alpha+1)b-1} & b_0 < b \le 1 \end{cases} \tag{22}$$

This function covers all points which are on the vertical surface and one plane and at same time satisfy all constraints. By doing some calculus, we know that Eq. (22) is monotonically decreasing in $(0, b_0]$ and monotonically increasing in $(b_0, 1]$. Therefore, the minimum value of Eq. (22) can be obtained at (b_0, c_0, s_0). The complete proof is given by Lemma 4 in Appendix I. It means that we can obtain the optimal solution of problem (17) by solving the following equation system.

$$\begin{cases} b_0 + xc_0 = s_0 \\ \lambda b_0 + c_0 = s_0 \\ \lambda b_0^2 + (\alpha\lambda-\alpha+1)b_0c_0 - (\lambda+1)b_0 - c_0 + 1 = 0 \end{cases} \tag{23}$$

By joining the first two equations we have $c_0 = \frac{1-\lambda}{1-\alpha} \times b_0$, and applying it to the last equation in (23) gives

$$(-\alpha\lambda^2 + \alpha\lambda - \alpha + 1)b_0^2 + (\alpha\lambda + \alpha - 2)b_0 + (1-\alpha) = 0$$

By the well-known Quadratic Formula we get the two roots of the above quadratic equation.

$$b_0^1 = \frac{(2-\alpha\lambda-\alpha) + (1-\lambda)\sqrt{-3\alpha^2+4\alpha}}{2(-\alpha\lambda^2+\alpha\lambda-\alpha+1)} \tag{24}$$

$$b_0^2 = \frac{(2-\alpha\lambda-\alpha) - (1-\lambda)\sqrt{-3\alpha^2+4\alpha}}{2(-\alpha\lambda^2+\alpha\lambda-\alpha+1)} \tag{25}$$

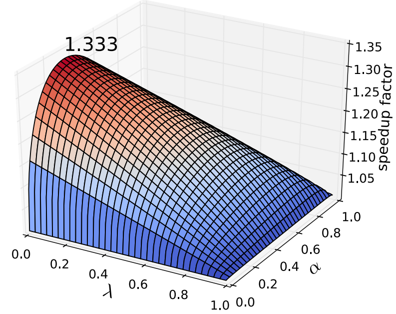


Figure 3: 3D image of speedup factor w.r.t α and λ

λ \ α	0.1	0.3	1/3	0.5	0.7	0.9	1
0	1.254	1.332	**1.333**	1.309	1.227	1.091	1
0.1	1.231	1.308	1.310	1.293	1.219	1.090	1
0.3	1.183	1.256	1.259	1.254	1.201	1.087	1
0.5	1.134	1.195	1.200	1.206	1.174	1.083	1
0.7	1.082	1.126	1.130	1.143	1.133	1.074	1
0.9	1.028	1.046	1.048	1.056	1.061	1.048	1
1	1	1	1	1	1	1	1

Table II: Speedup factor w.r.t α and λ

We can prove that b_0^2 is larger than 1 and thus should be dropped (since we require $0 \le b \le 1$), while b_0^1 is in the range of $[0, 1]$. The detailed proof is given by Lemma 5 in Appendix I. As a result, we obtain the optimal solution $(b_0^1, \frac{1-\alpha}{1-\lambda}b_0^1, \frac{1-\alpha\lambda}{1-\lambda}b_0^1)$ for Eq. (23). Thus, we have

$$\begin{aligned} S(\alpha, \lambda) &= \frac{1-\alpha\lambda}{1-\lambda}b_0^1 \\ &= \frac{(1-\alpha\lambda)((2-\alpha\lambda-\alpha)+(\lambda-1)\sqrt{4\alpha-3\alpha^2})}{2(1-\alpha)(\alpha\lambda-\alpha\lambda^2-\alpha+1)} \end{aligned}$$

Therefore, Lemma 2 is proved. ■

Lemma 2 shows that any IMC task set that is schedulable by an optimal clairvoyant MC scheduling algorithm on a speed-$S(\alpha, \lambda)$ is schedulable by EDF-VD on a unit-speed processor. Therefore, we can compute the speedup factor of EDF-VD by $1/S(\alpha, \lambda)$.

Theorem 4. *The speedup factor of EDF-VD with IMC task sets is*

$$f = \frac{2(1-\alpha)(\alpha\lambda-\alpha\lambda^2-\alpha+1)}{(1-\alpha\lambda)((2-\alpha\lambda-\alpha)+(\lambda-1)\sqrt{4\alpha-3\alpha^2})}$$

The speedup factor is shown to be a function with respect to α and λ. Figure 3 plots the 3D image of this function and Table II lists some of the values with different α and λ. By doing some calculus, we obtain the maximum value 1.333 (4/3) of the speedup factor function when $\lambda = 0$ and $\alpha = \frac{1}{3}$, which is highlighted in Figure 3 and Table II. We see that the speedup factor bound is achieved when the task set is a classical MC task set. From Figure 3 and Table II, we observe different trends for the speedup factor with respect to α and λ.

- First, given a fixed λ, the speedup factor is not a monotonic function with respect to α. The relation between α and the speedup factor draws a downward parabola.

Therefore, a straightforward conclusion regarding the impact of α on the speedup factor cannot be drawn.

- Given a fixed α, the speedup factor is a monotonic decreasing function with respect to increasing λ. It is seen that increasing λ leads to a smaller value of the speedup factor. *This means that a larger λ brings a positive effect on the schedulability of an IMC task set.*

VI. Extension to Elastic Mixed-Criticality Model

Su and Zhu in [6] introduced an *Elastic Mixed-Criticality* (EMC) task model, where the *elastic model* [19] is used to model low criticality tasks. When the MC system switches to high criticality mode, low-criticality tasks scale up their original period to a larger period such that low-criticality tasks continue to be scheduled with a degraded service (less frequently). Although the EMC model has been studied by [6] [13] [14], there is not a utilization-based sufficient test for the EMC model. Therefore, in this section, we prove that the theories proposed in Section IV apply to the EMC model [6] as well. Here, we use $T_i^{max}(\geq T_i)$ to denote the extended period of a *low*-criticality task τ_i. Since, in the EMC model, the WCETs of a *low*-criticality tasks are the same in two modes, the utilization of *low*-criticality task τ_i in *high*-criticality mode is computed as $u_i^{HI} = C_i^{LO}/T_i^{max}$.

Proposition 5 (Lemma 1 from [6]). *A set of EMC tasks is EMC schedulable under EDF-VD if $U_{HI}^{HI} + U_{LO}^{HI} \leq 1$.*

Here, in order to keep the consistence, we use U_{LO}^{HI} to denote $U(L, min)$ in [6]. Proposition 5 is provided in [6] to check the schedulability of an EMC task set on a uniprocessor. However, Proposition 5 is a necessary test. This means that even if a given task set satisfies the condition presented in Proposition 5, it is still possible that the task set is unschedulable under EDF-VD. Below, we prove that the theories proposed in Section IV can apply to the EMC model.

First, in *low*-criticality mode, since the EMC model just behaves like the classical MC model, Theorem 1 holds for the EMC model. Then we discuss the schedulability of the EMC model in *high*-criticality mode. We have the following definition for the carry-over job of a low criticality task in the EMC model:

Definition 3. *In the EMC model, carry-over job J_i of low criticality task τ_i has its release time $a_i < t_1$ and **original** deadline $d_i > t_1$.*

Then, we prove the following proposition for a carry-over job.

Proposition 6. *For an EMC carry-over job J_i, if it completes its execution before switch time instant t_1, then its **original** deadline d_i is $\leq (a_1 + x(t_2 - a_1))$.*

Proof: Consider that carry-over job J_i completes its execution before switch time instant t_1. Suppose that J_i has its **original** deadline $d_i > (a_1 + x(t_2 - a_1))$. Let t^* denote the latest time instant at which J_i starts to execute before t_1. At time instant t^*, all jobs in $\mathcal{J}$ with deadlines $\leq (a_1 + x(t_2 - a_1))$ then have finished their execution. Therefore, these jobs do not have any execution within interval $[t^*, t_2]$. This implies that jobs in $\mathcal{J}$ which are released at or after t^* can form a smaller job set and this smaller job set is sufficient to cause deadline miss at t_2. This contradicts the minimality of $\mathcal{J}$. Therefore, in this case we have $d_i \leq (a_1 + x(t_2 - a_1))$ ∎

Lemma 3. *Lemma 1 still holds for low-criticality tasks of the EMC model in high-criticality mode.*

Proof: We can prove this lemma by doing some modifications on the proof of Lemma 1. Here, we mainly focus on the modified part. The proof uses the same notations explained in Section IV.

For the EMC model, we need to consider a special case when carry-over job J_i of *low*-criticality task τ_i has its extended deadline $d_i^{max} > t_2$. Since t_2 is a deadline miss, a job with deadline $> t_2$ will not be scheduled within $[t_1, t_2)$ -see Proposition 1. If $d_i^{max} > t_2$, job J_i will not be executed after the switch time instant t_1 and the maximum cumulative execution time of τ_i can be obtained as job J_i completes its C_i^{LO} before t_1. Hence, the cumulative execution of task τ_i can be bounded by,

$$\eta_i \leq a_i \cdot u_i^{LO} + (d_i - a_i)u_i^{LO} = d_i \cdot u_i^{LO} \tag{26}$$

By Proposition 6, we replace d_i with $(a_1 + x(t_2 - a_1))$ in Eq. (33)

$$\begin{aligned} &\eta_i \leq (a_1 + x(t_2 - a_1))u_i^{LO} + (t_2 - (a_1 + x(t_2 - a_1)))u_i^{HI} \\ \Leftrightarrow &\eta_i \leq (a_1 + x(t_2 - a_1))u_i^{LO} + (1 - x)(t_2 - a_1)u_i^{HI} \end{aligned} \tag{27}$$

The rest of the proof can follow the proof of Lemma 1. A complete proof can be found in Appendix II. ∎

Lemma 3 shows that Lemma 1 can still bound the cumulative execution time of *low*-criticality tasks of the EMC model in *high*-criticality mode. Moreover, since there is no difference how the *high*-criticality tasks are scheduled in the EMC model or in the classical MC model, Proposition 4 still holds for the *high*-criticality tasks in the EMC model. As a result, Theorem 2 holds for the EMC model as well. Then, we can directly obtain the following theorem,

Theorem 5. *Theorem 3 is a sufficient test for the EMC model under EDF-VD.*

Since Theorem 3 is a sufficient test for the EMC model under EDF-VD, the speedup factor results we obtained in Section V also apply to the EMC model, i.e., the speedup factor bound of the EMC model under EDF-VD is also $4/3$ by using our proposed sufficient test.

VII. Experimental Evaluation

In this section, we conduct experiments to evaluate the effectiveness of the proposed sufficient test for the IMC model in terms of schedulable task sets (acceptance ratio). Moreover, we conduct experiments to verify the observations obtained in

Section V regarding the impact of α and λ on the average acceptance ratio. Our experiments are based on randomly generated MC tasks. We use a task generation approach, similar to that used in [5] [4], to randomly generate IMC task sets to evaluate the proposed sufficient test. Each task τ_i is generated based on the following procedure,

- pCriticality is the probability that the generated task is a *high*-criticality task; pCriticality$\in [0, 1]$.
- Period T_i is randomly selected from range $[100, 1000]$.
- In order to have sufficient number of tasks in a task set, utilization u_i is randomly drawn from the range$[0.05, 0.2]$.
- For any task τ_i, $C_i^{LO} = u_i * T_i$.
- $R \geq 1$ denotes the ratio C_i^{HI}/C_i^{LO} for every *high*-criticality task. If $L_i = HI$, we set $C_i^{HI} = R * C_i^{LO}$. It is easy to see that α used in the speedup factor function is equal to $\frac{1}{R}$;
- $\lambda \in (0, 1]$ denotes the ratio C_i^{HI}/C_i^{LO} for every *low*-criticality task. If $L_i = LO$, we set $C_i^{HI} = \lambda * C_i^{LO}$.

In the experiment, we generate IMC task sets with different target utilization U. Each task set is generated as follows. Given a target utilization U, we first initialize an empty task set. Then, we generate task τ_i according to the task generation procedure introduced above and add the generated task to the task set. The task set generation stops as we have

$$U - 0.05 \leq U_{avg} \leq U + 0.05$$

where

$$U_{avg} = \frac{U^{LO} + U^{HI}}{2}$$

is the average total utilization of the generated task set. If adding a new task makes $U_{avg} > U + 0.05$, then the added task will be removed and a new task will be generated and added to the task set till the condition is met.

A. Comparison with AMC [7]

To date, the modified AMC given in [7] is the only related work considering the schedulability of the IMC model under fixed-priority scheduling. Therefore, in the first experiment, we compare EDF-VD by using our proposed test to the AMC approach in [7] in terms of average acceptance ratio. In this experiment, R is randomly selected from a uniform distribution $[1.5, 2.5]$. With different λ and pCriticality settings, we vary U_{avg} from 0.4 to 0.95 with step of 0.05, to evaluate the effectiveness of the proposed sufficient test in terms of the average acceptance ratios. We generate 10,000 task sets for each given U_{avg}. Due to space limitations, we only present the experimental results when pCriticality$= 0.5$. Results with different pCriticality settings can be found in Appendix III. The results are shown in Figure 4, where the x-axis denotes the varying U_{avg} and the y-axis denotes the acceptance ratio. In the figures, let EDF-VD and AMC denote our proposed schedulability test and the one proposed in [7], respectively. In most cases, EDF-VD outperforms AMC in terms of acceptance ratio. We observe the following trends:

1) When $U_{avg} \in [0.5, 0.8]$, EDF-VD always outperforms AMC in terms of acceptance ratio. However, if $U_{avg} > 0.8$ and $\lambda = 0.3$ or 0.5, AMC performs better than EDF-VD. The same trend is also found for the classical MC model under EDF-VD and AMC, see in [4].
2) By comparing sub-figures in Figure 4, we see that the average acceptance ratio improves when λ increases. This confirms the observation for the speedup factor we obtained in Section V. The increasing λ leads to a smaller speedup factor. As a result, it provides a better schedulability. We need to notice that when λ increases, not only EDF-VD improves its acceptance ratio but the acceptance ratio of AMC [7] also improves.

B. Impact of α and λ

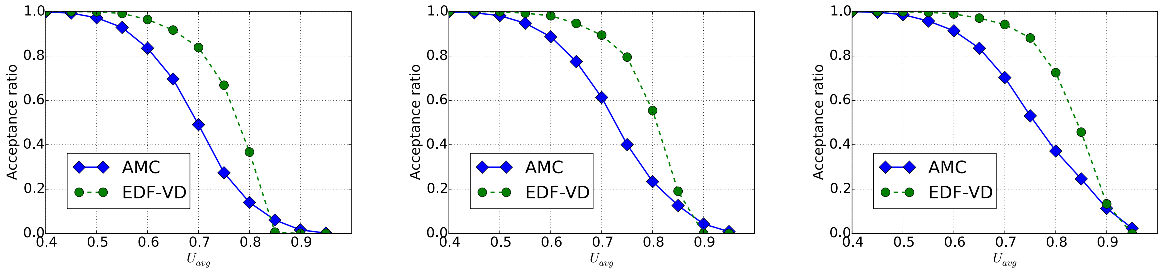
Above, we compare our proposed sufficient test to the existing AMC approach. In this section, we conduct experiments to further evaluate the impact of λ and α $(1/R)$ on the acceptance ratio. In this experiment, we select $U_{avg} = \{0.65, 0.7, 0.75, 0.8, 0.85\}$ to conduct experiments. We fix U_{avg} to a certain utilization and vary λ and α to evaluate the impact.

We first show the results for λ. The results are depicted in Figure 5, where the x-axis denotes the value of λ from 0.2 to 0.9 with step of 0.1 and the y-axis denotes the average acceptance ratio. R is randomly selected from a uniform distribution $[1.5, 2.5]$ and pCriticality$= 0.5$. Similarly, 10,000 task sets are generated for each point in the figures. A clear trend can be observed that the acceptance ratio increases as λ increases. This trend confirms the positive impact of increasing λ on the schedulability which we have observed in Section V.

Next we conduct experiments to evaluate the impact of α on the schedulability. Similarly, we fix U_{avg} and vary α to carry out the experiments. Due to $\alpha = \frac{1}{R}$, if α is given, we compute the corresponding R to generate task sets. The results are depicted in Figure 6, where $\lambda = 0.5$. The x-axis denotes the varying α from 0.1 to 0.9 with step of 0.1. while the y-axis denotes the average acceptance ratio. First, from Table II, we see that with increasing α the speedup factor first increases till a point. This means within this range the scheduling performance of EDF-VD gradually decreases. After that point, the speedup factor decreases which means the scheduling performance of EDF-VD gradually improves. The experimental results confirm what we have observed for α in Section V. The acceptance ratio gradually decreases till a point and then it increases.

VIII. Conclusions

In this paper, the *imprecise mixed-criticality* (IMC) model from [7] is investigated. A sufficient test for the IMC model under EDF-VD is proposed and the proposed sufficient test later applies to the EMC model as well. Based on the proposed sufficient test, we derive a speedup factor function with respect to the utilization ratio α of all *high*-criticality tasks and the utilization ratio λ of all *low*-criticality tasks. This speedup factor function provides a good insight to observe the impact of α and λ on the speedup factor and enables us to quantify



(a) $\lambda = 0.3$

(b) $\lambda = 0.5$

(c) $\lambda = 0.7$

Figure 4: Varying U_{avg} with different λ and pcriticality=0.5

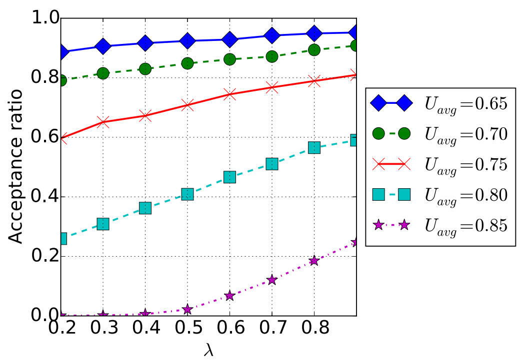


Figure 5: Impact of λ

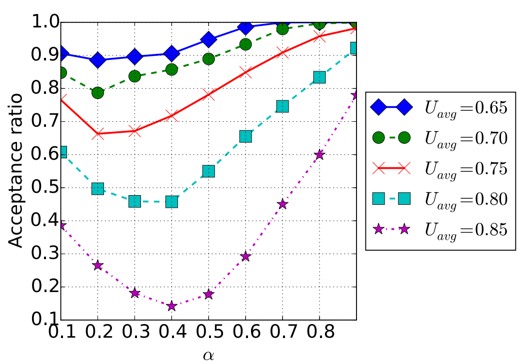


Figure 6: Impact of α

the suboptimality of EDF-VD for the IMC/EMC model in terms of speedup factor. Our experimental results show that our proposed sufficient test outperforms the AMC approach in terms of acceptance ratio. Moreover, the extensive experiments also confirm the observations we obtained for the speedup factor.

Acknowledgement

This article was partially supported by National Natural Science Foundation of China (61370076 and 61532007), Collaborative Innovation Center of Major Machine Manufacturing in Liaoning, and State Key Laboratory of Synthetical Automation for Process Industries (PAL-N201503).

References

[1] S. Baruah, H. Li, and L. Stougie, "Towards the design of certifiable mixed-criticality systems," in *Proceedings of the 16th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2010.

[2] S. Baruah, V. Bonifaci, G. D'Angelo, H. Li, A. Marchetti-Spaccamela, S. V. der Ster, and L. Stougie, "The preemptive uniprocessor scheduling of mixed-criticality implicit-deadline sporadic task systems," in *Proceedings of the 24th Euromicro Conference on Real-Time Systems*, 2012.

[3] S. K. Baruah, A. Burns, and R. I. Davis, "Response-time analysis for mixed criticality systems," in *Proceedings of the 32nd IEEE Real-Time Systems Symposium*, 2011.

[4] P. Ekberg and W. Yi, "Bounding and shaping the demand of generalized mixed-criticality sporadic task systems," *Real-time systems*, vol. 50, no. 1, 2014.

[5] A. Easwaran, "Demand-based scheduling of mixed-criticality sporadic tasks on one processor," in *Proceedings of the 2013 IEEE 34th Real-Time Systems Symposium*, 2013.

[6] H. Su and D. Zhu, "An elastic mixed-criticality task model and its scheduling algorithm," in *Design, Automation Test in Europe Conference Exhibition*, 2013.

[7] A. Burns and S. Baruah, "Towards a more practical model for mixed criticality systems," in *Proceedings of Workshop on Mixed Criticality, IEEE Real-Time Systems Symposium*, 2013.

[8] J. W. Liu, K.-J. Lin, W.-K. Shih, A. C.-s. Yu, J.-Y. Chung, and W. Zhao, "Algorithms for scheduling imprecise computations," *Computer*, vol. 24, no. 5, 1991.

[9] J. W. Liu, W.-K. Shih, K.-J. Lin, R. Bettati, and J.-Y. Chung, "Imprecise computations," *Proceedings of the IEEE*, vol. 82, no. 1, 1994.

[10] A. Burns and R. Davis, "Mixed criticality systems-a review," *University of York, Tech. Rep*, 2015.

[11] P. Huang, G. Giannopoulou, N. Stoimenov, and L. Thiele, "Service adaptions for mixed-criticality systems," in *Proceedings of the 19th Asia and South Pacific Design Automation Conference*, Jan 2014.

[12] R. Ravindran, C. M. Krishna, I. Koren, and Z. Koren, "Scheduling imprecise task graphs for real-time applications," *International Journal of Embedded Systems (IJES)*, vol. 6, 2014.

[13] H. Su, N. Guan, and D. Zhu, "Service guarantee exploration for mixed-criticality systems," in *Proceedings of IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*, 2014.

[14] M. Jan, L. Zaourar, and M. Pitel, "Maximizing the execution rate of low-criticality tasks in mixed criticality system," *Proceedings of Workshop on Mixed-Criticality, IEEE Real-Time Systems Symposium*, 2013.

[15] J.-Y. Chung, J. W. S. Liu, and K.-J. Lin, "Scheduling periodic jobs that allow imprecise results," *IEEE Trans. Comput.*, vol. 39, no. 9, 1990.

[16] F. Santy, L. George, P. Thierry, and J. Goossens, "Relaxing mixed-criticality scheduling strictness for task sets scheduled with fp," in *Proceedings of the 24th Euromicro Conference on Real-Time Systems*, July 2012.

[17] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM (JACM)*, 1973.

[18] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, Calif.: University of California Press, 1951.

[19] G. C. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni, "Elastic scheduling for flexible workload management," *IEEE Trans. Comput.*, vol. 51, no. 3, 2002.

APPENDIX I

Lemma 4. *The minimum value of piece-wise function (22) given in Section V is obtained when $b = b_0$.*

$$s(b) = \begin{cases} \frac{(\alpha\lambda^2-\alpha\lambda)b^2+b-1}{(\alpha\lambda-\alpha+1)b-1} & 0 < b \le b_0 \\ \frac{(1-\alpha)b^2+(\alpha\lambda+\alpha-1)b-\alpha}{(\alpha\lambda-\alpha+1)b-1} & b_0 < b \le 1 \end{cases} \tag{28}$$

Proof: For case of $0 < b \le b_0$, its derivative is

$$s'(b) = \frac{\alpha(\lambda-1)(\lambda(\alpha y-\alpha+1)b^2-2\lambda b+1)}{((\alpha\lambda-\alpha+1)b-1)^2}$$

The denominator is obviously positive. For the numerator, since the discriminant of $\lambda(\alpha\lambda-\alpha+1)b^2-2\lambda b+1=0$ is $(2\lambda)^2-4\lambda(\alpha\lambda-\lambda+1)$, which is negative since $0<\lambda<1$, so we know $\lambda(\alpha\lambda-\alpha+1)b^2-2\lambda b+1>0$. Moreover, we have $\lambda-1<0$, so putting them together we know the numerator is negative. In summary, $s'(b)$ is negative and thus $s(b)$ is monotonically decreasing with respect to b in the range $b \in (0, b_0]$.

For case of $b_0 < b \le 1$, we can compute the derivative of $s(b)$ by

$$s'(b) = \frac{(1-\lambda)((\lambda y-x+1)b^2-2b-(\lambda y-x-1))}{((\lambda y-x+1)b-1)^2}$$

The denominator is obviously positive. For the numerator, we focus on $(x\lambda-x+1)b^2-2b-(x\lambda-x-1)$ part. The following equation

$$(x\lambda-x+1)b^2-2b-(x\lambda-x-1)=0$$

has two roots $b_1=1$ and $b_2=\frac{1+(x-x\lambda)}{1-(x-x\lambda)}$, which is greater than 1, so we know $(x\lambda-x+1)b^2-2b-(x\lambda-x-1)$ is either always positive or always negative in the range of $b\in(b_0,1)$. Since we can construct $(x\lambda-x+1)b^2-2b-(x\lambda-x-1)>0$ with $x=\lambda=b=0.5$, so we know $(x\lambda-x+1)b^2-2b-(x\lambda-x-1)$ is always positive. Moreover, since $1-x>0$, the numerator of $s'(b)$ is positive, so overall $s'(b)$ is positive, and thus $s(b)$ is monotonically increasing with respect to b in the range of $b\in(b_0,1]$.

In summary, we have proved $s(b)$ is monotonically decreasing in $(0,b_0]$, and monotonically increasing in $(b_0,1]$, both with respect to b, so the smallest value of $s(b)$ must occur at b_0. ∎

Lemma 5. *If $0<\alpha<1$ and $0\le\lambda<1$, then*

$$b_0^1 = \frac{(2-\alpha\lambda-\alpha)+(1-\lambda)\sqrt{-3\alpha^2+4\alpha}}{2(-\alpha\lambda^2+\alpha\lambda-\alpha+1)} > 1 \tag{29}$$

$$b_0^2 = \frac{(2-\alpha\lambda-\alpha)-(1-\lambda)\sqrt{-3\alpha^2+4\alpha}}{2(-\alpha\lambda^2+\alpha\lambda-\alpha+1)} \in [0,1] \tag{30}$$

Proof: We start with proving $b_0^1>1$. We first prove $b_0^1\ge 0$ by showing both the numerator and dominator are positive. For simplicity, we use N_1 and M_1 to denote the numerator and denominator of b_0^1 in (29), and N_2 and M_2 the numerator and denominator of b_0^2 in (30). Note that the following reasoning relies on that $\alpha\in(0,1),\lambda\in[0,1)$.

1) $N_1>0$. First, we have

$$\begin{aligned} & N_1 \times N_2 \\ &= (2-\alpha\lambda-\alpha)^2-(1-\lambda)^2(-3\alpha^2+4\alpha) \\ &= 4\alpha\lambda(1-\lambda)(1-\alpha)+4(1-\alpha)^2 \\ &> 0 \end{aligned}$$

Moreover, it is easy to see $N_2>0$. Therefore, we can conclude that N_1 is also positive.

2) $M_1>0$. $2(-\alpha\lambda^2+\alpha\lambda-\alpha+1)=2(\alpha\lambda(1-\lambda)+(1-\alpha))$, which is positive.

In summary, both the numerator and the denominator of b_0^1 in (29) are positive, so $b_0^1\ge0$. Next we prove $b_0^1\le1$ by showing $N_1-M_1\le0$:

$$\begin{aligned} & N_1-M_1 \\ &= (\lambda-1)(\sqrt{-3\alpha^2+4\alpha}+\alpha(2\lambda-1)) \end{aligned}$$

which is negative if $\lambda\ge0.5$ (since $\lambda-1<0$ and $\sqrt{-3\alpha^2+4\alpha}+\alpha(2\lambda-1)\ge0$). So in the following we focus on the case of $\lambda<0.5$. Since $\lambda<0.5$, we know $\alpha(2\lambda-1)$ is negative, so we define two positive number A and B as follows

$$A=\sqrt{-3\alpha^2+4\alpha} \tag{31}$$

$$B=\alpha(1-2\lambda) \tag{32}$$

so $N_1-M_1=(\lambda-1)(A-B)$. Since $\lambda-1<0$, we only need to prove $A-B>0$, which is equivalent to proving $A^2-B^2>0$ (as both A and B are positive): $A^2-B^2>0$, which is done as follows:

$$\begin{aligned} A^2-B^2 &= -3\alpha^2+4\alpha-\alpha^2(2\lambda-1)^2 \\ &= 4\alpha(1-\alpha)+4\alpha^2\lambda(1-\lambda) \\ &> 0 \end{aligned}$$

so we have $A-B>0$ and thus $N_1-M_1=(\lambda-1)(A-B)<0$. In summary, we have proved $N_1-M_1<0$ for the cases of both $\lambda\ge0.5$ and $\lambda<0.5$, so we know $b_0^1\in[0,1]$.

Next we prove $b_0^2>1$, by showing $N_2-M_2>0$

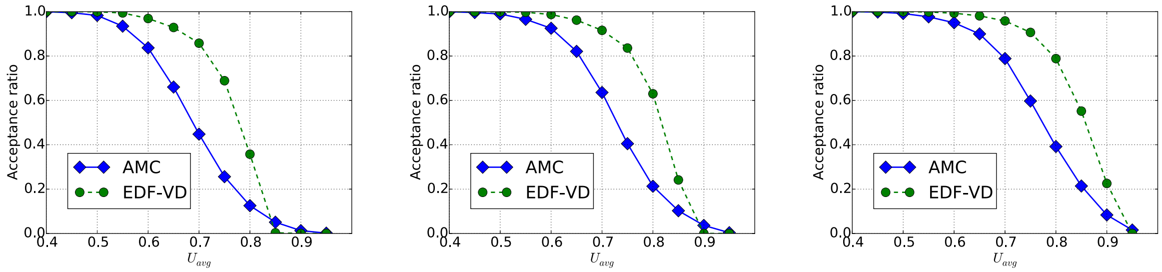
$$\begin{aligned} & N_2-M_2 \\ &= (1-\lambda)(\sqrt{-3\alpha^2+4\alpha}-\alpha(2\lambda-1)) \end{aligned}$$

If $\lambda\le0.5$, then $\sqrt{-3\alpha^2+4\alpha}-\alpha(2\lambda-1)>0$, and since $1-\lambda>0$ we have $N_2-M_2>0$. If $\lambda>0.5$, we let $C=\alpha(2\lambda-1)>0$ and also use A as defined above, $N_2-M_2=(1-\lambda)(A-C)$. To prove $A-C>0$, it suffices to prove $A^2-C^2>0$, as shown in the following:

$$\begin{aligned} A^2-C^2 &= -3\alpha^2+4\alpha-\alpha^2(2\lambda-1)^2 \\ &= 4\alpha-(3+(2\lambda-1)^2)\alpha^2 \\ &> 4\alpha-4\alpha^2 \quad (\lambda<1 \text{ ,so } 2\lambda-1<1) \\ &> 0 \end{aligned}$$

By now we have proved N_2-M_2 for both cases of $\lambda\le0.5$ and $\lambda>0.5$, so we known $b_0^2>1$.

∎



(a) $\lambda = 0.3$ (b) $\lambda = 0.5$ (c) $\lambda = 0.7$

Figure 7: Varying U_B with different λ and pcriticality=0.3

Appendix II

The following is the complete proof of Lemma 3.

Proof: We use the same notations explained in Section IV. When $u_i^{LO} = u_i^{HI}$, it is trivial to see that Lemma 1 holds for the EMC model. Now we focus on the case when $u_i^{LO} > u_i^{HI}$ To prove this case, we need to consider two cases where *low*-criticality task τ_i releases a job within interval $(a_1, t_2]$ or it does not.

- **Case 1** (τ_i releases a job within interval $(a_1, t_2]$): If there is no carry-over job , the proof is the same as we have proved for the IMC model (see the proof of Sub-case 1 in Lemma 1). Here, we focus on the case that there is a carry-over job. Let J_i denote the carry-over job with absolute release time a_i, original deadline d_i, and maximum deadline d_i^{max}. Here, we consider two cases, $d_i^{max} > t_2$ and $d_i^{max} \le t_2$.
  - $d_i^{max} > t_2$: since t_2 is a deadline miss, a job with deadline $> t_2$ will not be scheduled within $[t_1, t_2)$ -see Proposition 1. If $d_i^{max} > t_2$, job J_i will not be executed after the switch time instant t_1 and the maximum cumulative execution time of τ_i can be obtained as job J_i completes its C_i^{LO} before t_1. Hence, the cumulative execution of task τ_i can be bounded by,

$$\eta_i \le a_i \cdot u_i^{LO} + (d_i - a_i)u_i^{LO} = d_i \cdot u_i^{LO} \qquad (33)$$

    By Proposition 6, we replace d_i with $(a_1 + x(t_2 - a_1))$ in Eq. (33)

$$\begin{aligned} &\eta_i \le (a_1 + x(t_2 - a_1))u_i^{LO} + (t_2 - (a_1 + x(t_2 - a_1)))u_i^{HI} \\ \Leftrightarrow &\eta_i \le (a_1 + x(t_2 - a_1))u_i^{LO} + (1 - x)(t_2 - a_1)u_i^{HI} \end{aligned} \qquad (34)$$

  - $d_i^{max} \le t_2$: in this case, the cumulative execution of *low*-criticality task τ_i can be bounded as follows:

$$\begin{aligned} &\eta_i \le a_i u_i^{LO} + (t_2 - a_i)u_i^{HI} \\ &(\text{Since } a_i < t_1 \\ &\text{ and } t_1 < (a_1 + x(t_2 - a_1)) \text{ from Proposition 2}) \\ \Rightarrow &\eta_i \le \big(a_1 + x(t_2 - a_1)\big)u_i^{LO} + \big(t_2 - \big(a_1 + x(t_2 - a_1)\big)\big)u_i^{HI} \\ \Leftrightarrow &\eta_i \le (a_1 + x(t_2 - a_1))u_i^{LO} + (1 - x)(t_2 - a_1)u_i^{HI} \end{aligned} \qquad (35)$$

- **Case 2** (τ_i does not release a job within interval $(a_1, t_2]$): For *low*-criticality task τ_i, let J_i denote the last release job before a_1, where $a_i (< a_1)$ and d_i are the absolute release time and deadline of J_i, respectively. Moreover, let $d_i^{max} (> d_i)$ denote the new absolute deadline of job J_i as the system switches to *high*-criticality mode. Here, there are two cases, $d_i \le t_1$ and $d_i > t_1$. For $d_i \le t_1$, the cumulative execution of task τ_i can be computed as follows:

$$\eta_i = d_i \cdot u_i^{LO} \qquad (36)$$

  For $d_i > t_1$, if $d_i^{max} \le t_2$, then the maximum cumulative execution can be bounded as follows:

$$\begin{aligned} &\eta_i \le a_i \cdot u_i^{LO} + (d_i^{max} - a_i)u_i^{HI} \\ \Rightarrow &\eta_i \le a_i \cdot u_i^{LO} + (t_2 - a_i)u_i^{HI} \quad (\text{since } d_i^{max} \le t_2) \end{aligned}$$

  Since $a_i < t_1 \le (a_1 + x(t_2 - a_1))$ by Proposition 2, we obtain

$$\begin{aligned} &\eta_i \le (a_1 + x(t_2 - a_1))u_i^{LO} + (t_2 - (a_1 + x(t_2 - a_1)))u_i^{HI} \\ \Leftrightarrow &\eta_i \le (a_1 + x(t_2 - a_1))u_i^{LO} + (1 - x)(t_2 - a_1)u_i^{HI} \end{aligned} \qquad (37)$$

  If $d_i^{max} > t_2$, our reasoning is similar to the case discussed in Case 1. The maximum cumulative execution happens to that J_i completes its execution before t_1. Similarly, in this case, its cumulative execution can be upper bounded by

$$\eta_i \le a_i u_i^{LO} + (d_i - a_i)u_i^{LO}$$

  By Proposition 6, we obtain

$$\begin{aligned} &\eta_i \le (a_1 + x(t_2 - a_1))u_i^{LO} + (t_2 - (a_1 + x(t_2 - a_1)))u_i^{HI} \\ \Leftrightarrow &\eta_i \le (a_1 + x(t_2 - a_1))u_i^{LO} + (1 - x)(t_2 - a_1)u_i^{HI} \end{aligned}$$

The above discussion proves that Lemma 1 still holds for *low*-criticality tasks of the EMC model. ■

Appendix III

Experimental results between EDF-VD and AMC is depicted in Figure 7, where pCriticality= 0.3.