Schedulability Analysis for Real-time Task Set on Resource with Performance Degradation and Periodic Rejuvenation

Xiayu Hua, Chunhui Guo, Hao Wu, Douglas Lautner†, Shangping Ren *
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616, USA
{xhua, cguo13, hwu28, dlautner}@hawk.iit.edu, ren@iit.edu

***Abstract*—Most schedulability analyses in the literature assume that the performance of computing resource does not change over time. However, due to ever increased complexity of computer systems, software aging issues become more difficult, if not impossible, to eradicate. Hence, the assumption that computing resource has a constant performance in its entire lifetime does not hold in real world long-standing systems. In this paper, we study real-time task schedulability under a resource model that the resource's performance degrades with a known degradation function and the resource is periodically rejuvenated. The resource model is referred to as P^2-resource model for performance degradation and periodic rejuvenation. We address three real-task schedulability related questions under the P^2-resource model, i.e., (1) resource supply bounds of the P^2-resource; (2) task set utilization bounds under Earliest Deadline First (EDF) and Rate Monotonic (RM) scheduling policies, respectively; and (3) experimentally study the tightness of the bounds developed, and the impact of resource degradation rate, rejuvenation period, and rejuvenation cost on the bounds.**

I. Introduction

Since the publication of the seminal paper by Liu and Layland [1] in 1973, the problem of real-time task scheduling under different resource models has been studied intensively. However, most of the studies rely on a strong assumption that the computing resource's performance does not change during its lifetime. Unfortunately, for many long-standing real-time applications, such as data acquisition systems (DAQ) [2], [3], deep-space exploration programs [4], [5] and SCADA systems for power, water and other national infrastructures [6], [7], the performance of computational resources decrease notably after a long and continuous execution period.

Over a twenty-day period we collected CPU and memory usage data using monitoring software [8] deployed on a Fermilab control system. As shown in Fig. 1, both CPU and memory consumptions increase with time. As the data monitoring software is the only application deployed on the computer, under normal operation, both CPU and memory consumption used by the software would remain at a constant level. Therefore, the increase of resource demand indicates that the amount of computational power provided by the system in a unit time keep decreasing when the system is running.

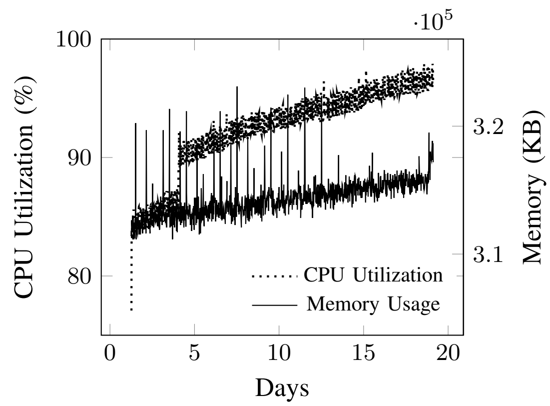


Fig. 1. Aging Effect on Fermi Monitoring System

The root cause of this phenomenon is software error accumulation and memory leak, which is also referred to as software aging problem [9]. Software bugs generally exist in any software, especially large and complex software systems. It is impractical, if not impossible, to determine and fix all of the bugs in software [9]. Due to the software aging problem, from application's perspective, the resources' performance continuously decreases while the application is running and hence the execution of the application keeps slowing down. If the running time of a system is sufficiently long, the software errors could consume all of the resources and eventually the application would stop working.

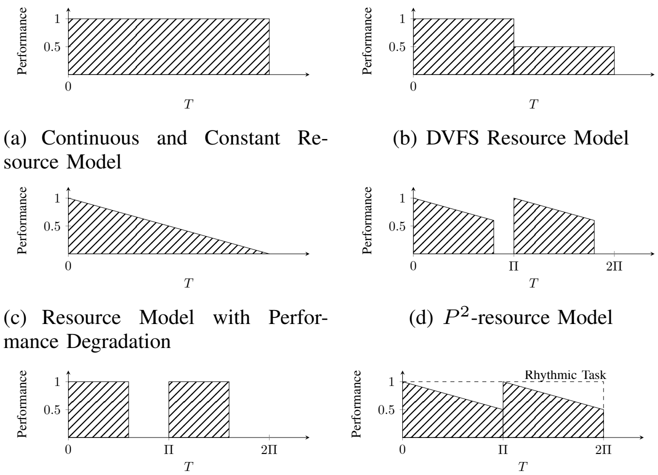
Apparently, the traditional resource models which assume the resource performance does not change are not accurate for real world scenarios where the software aging problem is ubiquitous and the resource performance degradation is unavoidable. To keep the long-standing system functional, software rejuvenation [10], [11] techniques are introduced as the countermeasure to recover the resource performance. However, software rejuvenation also introduces extra overhead.

†Douglas Lautner is a part-time Ph.D student. He is also the Director of Software Engineering at Motorola Mobility.

*This work is supported by NSF CAREER 381211; NSF CNS 381367; and NSF CPS 381396.

Specifically, when resources are performing software rejuvenation, they are not available to the applications. Since fast rejuvenation methods are introduced in the literature , such as application-level rejuvenation [12], [13], the unavailable time of resource is tolerable for tasks with relatively long period, such as the status monitor system used in FermiLab [8]. From application's perspective, such resources are only periodically available and even within their available time, the performance of the resources are continuously decreasing. To reflect these characteristics, we introduce a new resource model, the P^2-resource, to feature resources with $\underline{p}$erformance degradation and $\underline{p}$eriodic rejuvenation. Under the P^2-resource model, we study (1) resource supply bound provided by the P^2-resource; (2) task set utilization bounds under Earliest Deadline First (EDF) and Rate Monotonic (RM) scheduling policies, respectively; and (3) experimentally study the tightness of the bounds developed, and the impact of resource degradation rate, rejuvenation period, and rejuvenation cost on the bounds.

The rest of the paper is organized as follows: we discuss related work in Section II. The P^2-resource model is formally defined in Section III. The resource supply bound and linear supply bound of a P^2-resource are studied in Section IV. The utilization bounds for a task set under the EDF and RM scheduling policies on a P^2-resource are presented in Section V and Section VI, respectively. We further study the tightness of the two bounds and the impacts of different resource factors on the two bounds with experiments in Section VII. We conclude our work in Section VIII.



(a) Continuous and Constant Resource Model

(b) DVFS Resource Model

(c) Resource Model with Performance Degradation

(d) P^2-resource Model

(e) Constant and Periodic Resource Model

(f) Resource With Rhythmic Task

Fig. 2. Resource Models

II. Related Work

In 1973, Liu and Layland first introduced the Earliest Deadline First (EDF) and the Rate Monotonic (RM) scheduling policies for real-time systems and provided the utilization bounds for both EDF and RM scheduling policies [1]. In the following four decades, the real-time scheduling problem has been extensively studied. The main research focus is on developing new scheduling algorithms for real-time scheduling problems [14], [15] and improving utilization bounds for both EDF and RM scheduling policies on single [16] and multiple processors [17] under different constraints (preemptive [18] vs non-preemptive [19]) and for different task models (harmonic task set [14], mixed-criticality task set [20], etc.). However, most of the aforementioned work is based on a *continuous* and *constant* resource model, i.e., the resource is always available to applications and its performance does not change as illustrated in Fig. 2(a).

One exception is the research on the resource with Dynamic Voltage and Frequency Scaling (DVFS) ability. To reduce energy consumption of task execution, the speed of modern processors can be lowered via (DVFS) technology [21], [22]. Hence, in a DVFS-available system, the resource model is changed from the continuous and constant resource model to a continuous resource model with performance variations, as illustrated in Fig. 2(b). The schedulability analysis based on the DVFS resource model is studied intensively by the research community [21], [22], [23]. In the literature, the task schedulability study under the DVFS resource model makes two major assumptions: (1) resource can be switched between different performance levels and once it is switched to one level, the performance is stablized until the next switch takes place [24], [22], and (2) the performance change via DVFS is controllable and voluntary [22], [23].

On the other hand, as pointed in [25], [26], software aging problem is essentially error accumulation and memory leaking caused by software defects which are difficult to eradicate if at all. As shown in Fig. 2(c), the computing resource is inconsistent since it is unavailable to the tasks during rejuvenation. Therefore, schedulability analysis on the continuous and constant resource models can not be used on a P^2-resource.

For DVFS, at a high level, it can be considered as controlled aging (if we continuously scale down the frequency), and the switching overhead can also be mapped as rejuvenation overhead. However, as between two consecutive switches, the frequency remains the same which makes it different from P^2-resource resource model that the resource is decreasing within each rejuvenation period. Also, dislike DVFS resources, the performance degradation of a P^2-resource caused by software aging is progressive and involuntary. Therefore, the DVFS resource model is also insufficient for addressing a P^2-resource.

In another aspect, due to the resource performance degradation, the application will eventually stop working, which is unacceptable by any system. Hence, software rejuvenation technology is introduced as a countermeasure [9], [10], [27]. Through periodically rejuvenations, systems keep regaining their original performances. However, the computing resource becomes unavailable to applications when the rejuvenation is in progress. Therefore, from an application's point of view, resource with rejuvenation can be characterized as a P^2-resource that is periodically available and with performance degradation as illustrated in Fig. 2(d).

From Fig. 2(d), it is not difficult to see that the P^2-

resource is a periodic resource. The concept of the constant periodic resource was first introduced by Shigero *et al.* in 1999 [28]. Mok *et al.* [29] and Feng *et al.* [30] extended Shigero's original periodic resource model to the fixed-pattern periodic resource model and provided theoretical analysis on the schedulability of real-time task set under this model. Later, Shin *et al.* further extended the fixed-pattern periodic resource model to the dynamic pattern periodic resource model and provided formal analysis under both EDF and RM scheduling policies [31], [32]. However, all the literature work on periodic resources are based on one general assumption, i.e., when resources are available to applications, their computational power do not change. Hence, none of the existing theoretic results obtained under constant performance periodic resources can be directly applied under the P^2-resource model.

If considering the resource performance degradation as a special task, we can transform a resource with periodical performance degradation to a regular continuous and constant resource with a hidden task that has periodically increasing resource consumption running upon. Fig. 2(f) depicts this scenario. A similar case is studied in [33] where the authors call this special task a rhythmic task. In their work, they define the rhythmic task as a task with decreasing period and hence increasing utilization. The authors studied the schedulability when the system has one rhythmic task and one or multiple regular tasks. Their results are based on the assumption that the period of the rhythmic task is smaller than any of the regular tasks. However, for the problem we intend to solve, the rejuvenation period is often much larger than any of application tasks' periods due to the slow progress of aging effect [34]. Hence, the method of considering resource degradation as a rhythmic task can not be directly applied.

In this paper, we focus on the schedulability analysis for the P^2-resource model under both EDF and RM scheduling policies. We believe that the P^2-resource model is a more generalized resource model that can be easily transformed to the continuous and constant resource model [1] and constant periodic model [32]. In the next section, we formally define the P^2-resource model and formulate the problems to be studied in the paper.

III. Models and Problem Formulation

A. Resource model and assumptions

As illustrated in Fig. 2(d), we consider both the performance degradation and the rejuvenation time cost and model the resource. Note that by resource performance, we mean the computation cycles per unit time provided by the resource to applications. In the following section, we first provide the definitions and models used in this paper and then give the formal formulation of the problems we are to solve.

Resource performance degradation function

We use function $f(t)$ to denote the resource performance at time t. We assume that $f(t)$ is continuous, non-increasing and $f(0) = 1$.

Resource Rejuvenation

We assume that the resource is repeatedly rejuvenated with period π and that the resource performance never decreases to zero, i.e., $f^{-1}(\pi) > 0$, where $f^{-1}(\pi)$ is the time duration that the resource decreases from fully performing to not working. We also assume that the rejuvenation process is atomic and each rejuvenation takes ϕ time to complete. After each rejuvenation, the resource performance is reset to $f(0)$, i.e., $f(k\pi) = f(0)$ where $k \in \mathbb{N}^+$.

P^2-resource Model

A P^2-resource R is characterized by a triple $(f(t), \pi, \phi)$, where $f(t)$ is the *performance degradation function*, π is the resource *rejuvennation period*, and ϕ is the *rejuvenation time cost*. We assume the resource starts at time zero.

Task Model

The task model considered in this paper is similar to the one defined by Liu and Layland [1]. A task set $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$ has n independent periodic tasks that are all released at time 0. Each task $\tau_i \in \Gamma$ is a 2-tuple (P_i, e_i), where P_i is the *inter-arrival time* between any two consecutive jobs of τ_i (also called *period*), and e_i is the *task execution time* calibrated under maximum performance $f(0) = 1$ of a given P^2-resource. The *utilization* of the task set Γ is denoted as U_Γ, where

$$U_\Gamma = \sum_{\tau_i \in \Gamma} e_i / P_i$$

We use H to denote the hyper-period of Γ where H is the least common multiple of P_i for all $\tau_i \in \Gamma$.

We use P_{min} to denote the minimum task period of the task set Γ, i.e.,

$$P_{\text{min}} = \min\{P_i | \forall \tau_i \in \Gamma\}.$$

If $P_{\text{min}} \leq \phi$, a task set is not schedulable in the worst case. Hence, we assume $P_{\text{min}} > \phi$.

B. Problem formulation

This paper meant to study real-time task schedulability under the P^2-resource. We take two steps to address the problem. First, we analyze the minimal resource supply of a P^2-resource in a time interval with a given length. Second, we present the sufficient utilization bounds (UB) under both EDF and RM scheduling policies for a task set on a P^2-resource. The two problems are as follows.

Problem 1. *Given a P^2-resource $R(f(t), \phi, \pi)$, determine its supply bound function and linear supply bound function.*

Problem 2. *Given a P^2-resource $R(f(t), \phi, \pi)$ and a task set Γ, determine the utilization bounds of task set Γ on R under EDF and RM scheduling policies, respectively.*

As for any P^2-resource, the strategy to solve the problems are the same. To simplify mathematical transformations and deviations, and focus more on analysis strategies, in the following sections we assume that the resource performance function is a linear decreasing function, i.e.,

$$f(t) = 1 - a \cdot t \tag{1}$$

where a is a constant and $0 \leq a < 1$.

IV. P^2-RESOURCE SUPPLY BOUND ANALYSIS

To analyze task schedulability on P^2-resources, we first need to analyze the resource's supply bound. In this section, we present the supply bound function (SBF) and the linear supply bound function (LSBF) of a P^2-resource.

We use θ to denote the total computational cycles provided by a P^2-resource within one rejuvenation period (π), which is given by the following equation.

$$\theta = \int_0^{\pi-\phi} f(t)dt \tag{2}$$

In the next step, we derive the minimal supply bound function of a P^2-resource R.

Lemma 1. *Given a P^2-resource $R(f(t), \phi, \pi)$, its minimal supply function (msf) in a time interval with length $t (t \leq \pi)$ is*

$$\mathit{msf}(t, \pi, \phi) = \begin{cases} \int_{\pi-t}^{\pi-\phi} f(x)dx & \text{if } \phi < t \leq \pi \\ 0 & \text{if } 0 \leq t \leq \phi \end{cases} \tag{3}$$

Proof. We prove the lemma in the following two complementary cases separately.

Case 1: $0 \leq t < \phi$

As resource R is not available during its rejuvenation cost ϕ, hence the worst case is that the entire time interval is in the rejuvenation period. Therefore, like time interval t_1 in Fig. 3, the minimal resource supply is 0 in this case.

Case 2: $\phi \leq t \leq \pi$

We assumed in Section III that the resource's performance function $f(x)$ is a non-increasing function. Hence, the minimal resource supply when $0 < t \leq \pi$ is $\int_{\pi-t}^{\pi-\phi} f(x)dx$. Time interval t_2 in Fig. 3 is an example of this case. ☐

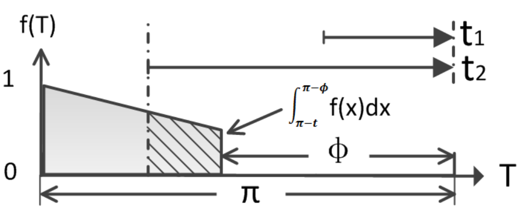


Fig. 3. Minimal Supply Function

We now extend the time interval length to an arbitrary value and give the P^2-resource's supply bound function and linear supply bound function in the following theorems.

Theorem 1. *Given a P^2-resource $R(f(t), \phi, \pi)$, its supply bound function (SBF) is*

$$\mathit{sbf}(t) = \left\lfloor \frac{t}{\pi} \right\rfloor \theta + \mathit{msf}(t \bmod \pi, \pi, \phi). \tag{4}$$

Proof. For a given time interval t, it contains $\lfloor \frac{t}{\pi} \rfloor$ entire periods that provide $\lfloor \frac{t}{\pi} \rfloor \theta$ amount of resource. For the remaining part of the time interval, its length is $(t \bmod \pi)$ and its minimal resource supply is $\mathit{msf}(t \bmod \pi, \pi, \phi)$. Hence, the supply bound function is calculated as Eq.(4). ☐

Theorem 2. *Given a P^2-resource $R(f(t), \phi, \pi)$, we have:*

(1) the resource's linear supply bound function lsbf(t) is the lower bound of the sbf(t), i.e.,

$$\forall t : \mathit{lsbf}(t) \leq \mathit{sbf}(t) \tag{5}$$

where

$$\mathit{lsbf}(t) = \frac{\theta}{\pi}(t - Tp) + \mathit{msf}(Tp, \pi, \phi) \tag{6}$$

and $Tp = \max\{\pi - \frac{\pi-\theta}{a\pi}, \phi\}$,

(2) the lsbf(t) a tight bound of the sbf(t), i.e.,

$$\exists t : \mathit{lsbf}(t) = \mathit{sbf}(t) \tag{7}$$

Proof. To prove the theorem, we consider when $n\pi \leq t < n\pi + \phi$ and when $n\pi + \phi \leq t < (n+1)\pi$ separately, for all $n \in \mathbb{N}$.

1) When $n\pi \leq t < n\pi + \phi$:
Based on Eq.(4), we have $\mathit{sbf}(t) = n\theta$. As $\mathit{lsbf}(x)$ is a monotonically increasing function, we have $\mathit{lsbf}(t) < \mathit{lsbf}(n\pi + \phi)$. Now, we need to prove $\mathit{lsbf}(n\pi + \phi) \leq n\theta$. We consider the following two complementary cases based on the two different values of Tp:

- **Case 1:** $\mathrm{Tp} = \phi$. Since $\mathit{msf}(\phi, \pi, \phi) = 0$ and $\mathit{lsbf}(x)$ is a non-decreasing function, we have

$$\mathit{lsbf}(n\pi+\phi) = \frac{\theta}{\pi}(n\pi+\phi-\mathrm{Tp})+\mathit{msf}(\mathrm{Tp}, \pi, \phi) = n\theta \tag{8}$$

- **Case 2:** $\mathrm{Tp} = \pi - \frac{\pi-\theta}{a\pi} \geq \phi$. Since $f(x)$ is a non-increasing function, we have $f(\pi - \phi) \leq f(\pi - (\pi - \frac{\pi-\theta}{a\pi})) = \frac{\theta}{\pi}$ and

$$\mathit{msf}(t, \pi, \phi) = \int_{\pi-\phi}^{\pi-x} f(t)dt \leq \frac{\theta}{\pi}(t - \phi)$$

Therefore, $\mathit{msf}(\mathrm{Tp}, \pi, \phi) \leq \frac{\theta}{\pi}(\mathrm{Tp} - \phi)$ and

$$\mathit{lsbf}(n\pi+\phi) \leq \frac{\theta}{\pi}(n\pi+\phi-\mathrm{Tp})+\mathit{msf}(\mathrm{Tp}, \pi, \phi) \leq n\theta$$

Since for both cases, $\mathit{lsbf}(n\pi + \phi) \leq n\theta$, we have $\mathit{lsbf}(t) \leq \mathit{sbf}(t)$ when $n\pi \leq t < n\pi + \phi$.

2) When $n\pi + \phi \leq t < (n+1)\pi$:
For this scenario, we want to prove $\mathit{sbf}(t) - \mathit{lsbf}(t) \geq 0$. To simplify the notation, we let $t' = t \bmod \pi$.

- **Case 1:** $\mathrm{Tp} = \phi$. We first do the following transformation.

$$\mathit{sbf}(t) - \mathit{lsbf}(t) =$$

$$n\theta + \mathit{msf}(t', \pi, \phi) - \frac{\theta}{\pi}(t - \phi) - \mathit{msf}(\phi, \pi, \phi)$$

$$= \mathit{msf}(t', \pi, \phi) - \frac{\theta}{\pi}(t' - \phi)$$

Since $\text{Tp} = \phi$, which indicates $\pi - \frac{\pi-\theta}{a\pi} \leq \phi$, we have $f(\pi - \phi) \geq f(\pi - (\pi - \frac{\pi-\theta}{a\pi})) = \frac{\theta}{\pi}$. Therefore,

$$msf(t, \pi, \phi) = \int_{\pi-\phi}^{\pi-x} f(t)dt \geq \frac{\theta}{\pi}(t - \phi)$$

and hence $msf(t', \pi, \phi) \geq \frac{\theta}{\pi}(t' - \phi)$, which indicates $sbf(t) - lsbf(t) \geq 0$.

Specially, when $t' = \text{Tp} = \phi$, $sbf(t) = lsbf(t)$.

- **Case 2:** $\text{Tp} = \pi - \frac{\pi-\theta}{a\pi}$. Similar to the proof in Case 1, we do the following transformation:

$$\begin{aligned} & sbf(t) - lsbf(t) = \\ & = n\theta + msf(t', \pi, \phi) - \frac{\theta}{\pi}(n\pi + t' - \text{Tp}) \\ & - msf(\text{Tp}, \pi, \phi) \\ & = msf(t', \pi, \phi) - \frac{\theta \cdot t'}{\pi} - msf(\text{Tp}, \pi, \phi) + \frac{\theta \cdot \text{Tp}}{\pi} \end{aligned}$$

Let $S(t) = msf(t, \pi, \phi) - \frac{\theta \cdot t}{\pi}$, then $S'(\text{Tp}) = 0$ which indicates function $S(t)$ has an extreme value when $t = \text{Tp}$. Moreover, as $\forall t \in [n\pi + \phi, (n+1)\pi)$, $S''(\text{Tp}) > 0$, then when $t = \text{Tp}$, function $S(t)$ has the minimal value. Therefore,

$$sbf(t) - lsbf(t) = S(t') - S(\text{Tp}) \geq 0$$

Specially, when $t' = \text{Tp}$, $sbf(t) = lsbf(t)$.

Combine both scenarios together, we have $\forall t, sbf(t) - lsbf(t) \geq 0$. Furthermore, when $t = \text{Tp}$, $sbf(t) = lsbf(t)$. Hence, Theorem 2 always holds.

Intuitively, $lsbf(t)$ is the tangent line of $sbf(t)$ and Tp is their first tangent point. Fig.4 depicts the relationship of $lsbf(t)$ and $sbf(t)$. □

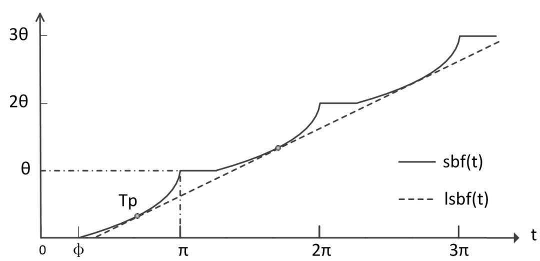


Fig. 4. SBF and LSBF

V. Task Set Utilization Bound on P^2-resource under the EDF Scheduling Policy

Based on the supply bound analysis of P^2-resource given in Section IV, we derive the sufficient utilization bound under EDF scheduling policy in this section.

Since the task model we use in this paper is the same as the task model used in [32], the linear demand bound function (LDBF) for a task set and the schedulability condition under EDF scheduling policy are the same.

Definition 1. *[32] Given a task set Γ, its linear demand bound function (LDBF) under EDF scheduling policy is defined as*

$$ldbf_{EDF}(t) = U_\Gamma \cdot t. \tag{9}$$

Theorem 3. *[32] Given a task set Γ and a P^2-resource $R(f(t), \phi, \pi)$, Γ is schedulable on R under EDF scheduling policy if*

$$\forall t \in [0, H] : dbf_{EDF}(t) \leq sbf(t) \tag{10}$$

where H is the hyper-period of Γ and $dbf_{EDF}(t) = \sum_{\tau_i \in \Gamma} \left\lfloor \frac{t}{\pi_i} \right\rfloor \cdot e_i$.

To simplify the calculation, in the following corollary, we further reduce the range of time interval length t that is needed to be checked and replace both the demand bound and the supply bound with their linear bounds, respectively.

Corollary 1. *Given a task set Γ and a P^2-resource $R(f(t), \phi, \pi)$, Γ is schedulable on R under EDF scheduling policy if*

$$\forall t \in [P_{min}, H] : ldbf_{EDF}(t) \leq lsbf(t) \tag{11}$$

where H is the hyper-period of Γ.

Proof. For a time interval length t that $t \in [0, P_{\text{min}})$, $dbf_{\text{EDF}}(t) = 0$. Since $sbf(t) \geq 0$, $dbf_{\text{EDF}}(t) \leq sbf(t)$ always holds over $t \in [0, P_{\text{min}})$. Hence, we only need to check time interval length $t \in [P_{\text{min}}, H]$ when determining the schedulability.

Moreover, according to Theorem 2, we have $\forall t : lsbf(t) \leq sbf(t)$. With $\forall t : \text{ldbf}_{\text{EDF}}(t) \geq dbf_{\text{EDF}}(t)$, we then have

$$\forall t : ldbf_{EDF}(t) \leq lsbf(t) \rightarrow dbf_{\text{EDF}}(t) \leq sbf(t)$$

Therefore, Γ is schedulable on R under the EDF policy if Eq.(11) holds. □

In the next step, we analyze the relationship between the LSBF and the LDBF and derive the utilization bound under the EDF scheduling policy.

For a P^2-resource and a task set, we first give the definition of the *critical time interval length*.

Definition 2. *Given a P^2-resource $R(f(t), \phi, \pi)$ and a schedulable task set Γ, the critical time interval length Tc under the EDF scheduling policy is defined as*

$$Tc = \frac{\frac{\theta}{\pi} \cdot Tp - msf(Tp, \pi, \phi)}{\frac{\theta}{\pi} - U_\Gamma} \tag{12}$$

In fact, the *critical time interval length* Tc is derived from equation $lsbf(t) = ldbf_{EDF}(t)$. In the following lemma, we prove that if a task set is schedulable, then $\text{Tc} > 0$. Also, if a time interval's length is equal or longer than Tc, we further prove that within this time interval, the minimal resource supply from the P^2-resource is assuredly equal or larger than the maximal resource demand of a task set under the EDF policy.

Lemma 2. *Given a P^2-resource $R(f(t), \phi, \pi)$ and a schedulable task set Γ, their critical time interval length Tc under EDF satisfies the following conditions:*

$$\begin{cases} Tc > 0 \\ \forall t \geq Tc : lsbf(t) \geq ldbf_{EDF}(t). \end{cases} \tag{13}$$

Proof. Since Γ is schedulable on R, we have $U_\Gamma < \frac{\theta}{\pi}$, which indicates $ldbf'_{EDF}(t) < lsbf'(t)$. With $ldbf_{EDF}(0) = 0$ and $lsbf(0) < 0$, we then have $\text{Tc} > 0$.

Moreover, with $lsbf(\text{Tc}) = ldbf_{EDF}(\text{Tc})$ and $ldbf'_{EDF}(t) < lsbf'(t)$, we have $\forall t > \text{Tc} : lsbf(t) > ldbf_{EDF}(t)$.

Therefore, conditions in Eq.(13) hold. □

With the critical time interval length and the schedulability condition, we derive the utilization bound for a task set on a P^2-resource under the EDF policy.

Theorem 4. *Given a task set Γ and a P^2-resource $R(f(t), \phi, \pi)$, the sufficient utilization bound under EDF scheduling policy is*

$$UB_{EDF}(P_{min}, a, \pi, \phi) = \frac{\theta}{\pi} - \frac{\frac{\theta}{\pi} \cdot Tp - msf(Tp, \pi, \phi)}{P_{min}} \tag{14}$$

where $Tp = \max\{\pi - \frac{\pi-\theta}{a\pi}, \phi\}$.

Proof. According to Corollary 1, the task set Γ is schedulable on the resource R if $\forall t \in [P_{\min}, H] : ldbf_{EDF}(t) \leq lsbf(t)$. Based on Lemma 2, we have $\forall t \geq \text{Tc} : ldbf_{EDF}(t) \leq lsbf(t)$. Hence, the task set Γ is guaranteed to be schedulable on resource R if $P_{\min} \geq \text{Tc}$. By solving the formula $P_{\min} = \text{Tc}$, we derive the utilization bound UB_{EDF} as below:

$$UB_{\text{EDF}}(P_{\min}, a, \pi, \phi) = \frac{\theta}{\pi} - \frac{\frac{\theta}{\pi} \cdot \text{Tp} - msf(\text{Tp}, \pi, \phi)}{P_{\min}}$$

□

The proposed P^2-resource model is a generalized model. Suppose a resource has no performance degradation, then the rejuvenation process is unnecessary, i.e., $f(t) = 1$ and $\phi = 0$. In this case, P^2-resource is de-generalized to a continuous and constant resource and the utilization bound under EDF policy $UB_{\text{EDF}}(P_{\min}, a, \pi, \phi)$ becomes the utilization bound given by Liu and Layland [1], i.e., $UB_{\text{EDF}}(P_{\min}, a, \pi, \phi) = 1$.

Corollary 2. *Given a task set Γ and a P^2-resource $R(1, \pi, 0)$, the task utilization bound under the EDF scheduling policy is $UB_{EDF}(P_{min}, a, \pi, \phi) = 1$*

Proof. For the given resource $R(f(t), \phi, \pi)$, let $\theta = \int_0^{\pi-\phi} f(t)dt = \pi$. As $f(t) = 1$ and $\text{Tp} \geq \phi$, $lsbf(\text{Tp}) = \text{Tp} - \phi = \text{Tp}$. Based on Lemma 2, Tp always exists. Hence,

$$\begin{aligned} UB_{\text{EDF}}(P_{\min}, a, \pi, \phi) &= \frac{\theta}{\pi} - \frac{\frac{\theta}{\pi} \cdot \text{Tp} - msf(\text{Tp}, \pi, \phi)}{P_{\min}} \\ &= \frac{\pi}{\pi} - \frac{\frac{\pi}{\pi} \cdot \text{Tp} - \text{Tp}}{P_{\min}} = 1. \end{aligned}$$

□

VI. Task Set Utilization Bound on P^2-resource under the RM Scheduling Policy

In this section, we analyze the sufficient utilization bound for a task set on a P^2-resource under the RM scheduling policy. First, we use a theorem and a corollary to derive the utilization bound and the de-generalized utilization bound, respectively, for a real-time task set on a P^2-resource under RM scheduling policy. We then give the formal proof of the utilization bound theorem.

Theorem 5. *Given a P^2-resource $R(f(t), \phi, \pi)$ and a task set Γ with task number n and minimal task period P_{min}. The utilization bound of the task set Γ on R under RM scheduling policy is:*

$$\begin{aligned} &UB_{RM}(P_{min}, n, a, \pi, \phi) \\ &= \frac{\theta}{\pi} \cdot n[\left(1 + \frac{k\pi + \frac{\pi}{\theta} msf(\phi + \delta, \pi, \phi)}{k\pi + \phi + \delta}\right)^{1/n} - 1] \end{aligned} \tag{15}$$

where $k = \lfloor \frac{P_{min}}{\pi} \rfloor$, $\delta = \max\{\min\{\lambda, \pi - \phi\}, 0\}$ and

$$\begin{aligned} \lambda = &-a(\phi + k\pi) + ((a\phi + ak\pi)^2 \\ &- \min\{2a((1 + a\phi - a\pi)(\phi + k\pi) + k\theta), (a\phi + ak\pi)^2\})^{\frac{1}{2}} \end{aligned}$$

Similar to the utilization bound under EDF scheduling policy, UB_{RM} can also be de-generalized to the utilization bound for RM policy given in [1] when P^2-resource is de-generalized to the continuous and constant resource.

Corollary 3. *Given a task set Γ and a P^2-resource $R(1, \pi, 0)$, the task utilization bound under RM scheduling policy is*

$$UB_{RM}(P_{min}, n, a, \pi, \phi) = n(2^{1/n} - 1)$$

Proof. Since $f(t) = 1$ and $\phi = 0$, we have $\frac{\theta}{\pi} = 1$ and $msf(\phi + \delta, \pi, \phi) = \phi + \delta$. Therefore,

$$\frac{k\pi + \frac{\pi}{\theta} msf(\phi + \delta, \pi, \phi)}{k\pi + \phi + \delta} = \frac{k\pi + \phi + \delta}{k\pi + \phi + \delta} = 1$$

and hence $UB_{\text{RM}}(P_{\min}, n, a, \pi, \phi) = n(2^{1/n} - 1)$. □

The following parts of this section are dedicated to prove Theorem 5. To do so, we first determine the utilization bound of a P^2-resource under the RM scheduling policy with the restriction that the ratio between any two tasks' period in Γ is less than two. We then remove the restriction for arbitrary task sets.

In our proof, we derive the utilization bound based on a schedulable task set that has lowest utilization and fully utilizes the resource, i.e., decreasing the period or increasing the execution time of any task in this task set makes the task set un-schedulable.

For a given resource R and a schedulable task set Γ that fully utilizes R, we take three steps to derive the utilization bound: (1) we first prove that if U_Γ equals to the utilization bound, the sum of task execution times of Γ is equal to $sbf(P_{\min})$; (2) we then calculate the $P_{\min}$ value for Γ that minimizes U_Γ; and (3) we derive the utilization bound based on the found $P_{\min}$ value.

Lemma 3. *For a real-time task set $\Gamma = \{\tau_1, ..., \tau_n\}$ and a P^2-resource $R(f(t), \phi, \pi)$, under the restriction that the ratio between any two task periods of Γ is less than 2, if Γ fully utilizes R under the RM scheduling policy with the smallest possible U_Γ, then it follows that*

$$\sum_{\tau_i \in \Gamma} e_i = sbf_R(P_{min}) \tag{16}$$

where P_{min} is the smallest task period of Γ.

Proof. Although the resource models are different, but the proof strategy of this lemma is similar with the proofs of Theorem 4 in [1] and Lemma 9.1 in [32]. We provide the detailed proof in the technical report [35]. □

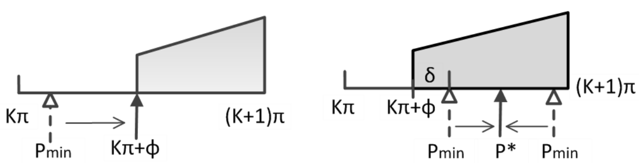
Lemma 4. *Given a real-time task set $\Gamma = \{\tau_1, ..., \tau_n\}$ and a P^2-resource $R(f(t), \phi, \pi)$, let $k = \lfloor \frac{P_{min}}{\pi} \rfloor$ and let P_{min} denote the smallest task period of Γ. Under the restriction that the ratio between any two task periods of Γ is less than two, if Γ fully utilizes R under RM scheduling policy, then U_Γ is minimized when*

$$P_{min} = k\pi + \phi + \max\{\min\{\lambda, \pi - \phi\}, 0\} \tag{17}$$

for all $P_{min} \in [k\pi, (k+1)\pi)$ where

$$\lambda = -(a\phi + ak\pi) + ((a\phi + ak\pi)^2 - \min\{2a((1 + a\phi - a\pi)(\phi + k\pi) + k\theta), (a\phi + ak\pi)^2\})^{\frac{1}{2}}$$

Proof. For all task sets of which $P_{\min} \in [k\pi, (k+1)\pi)$, let P^* denote the minimal task periods of the task set Γ^* which fully utilizes the resource R with *minimal* utilization. In the following parts, we first prove that $P^* \in [k\pi + \phi, (k+1)\pi)$ and then calculate the value of P^*.



(a) $P_{\min} \in [k\pi, k\pi + \phi)$ (b) $P_{\min} \in [k\pi + \phi, (k+1)\pi)$

Fig. 5. Transformation of $P_{\min}$

To prove $P^* \in [k\pi + \phi, (k+1)\pi)$, we consider the task sets of which $P_{\min} \in [k\pi, k\pi + \phi)$. Let Γ denote one of such task sets. As illustrated in Fig. 5(a), we transform $\Gamma = \{\tau_i\}$ to $\Gamma' = \{\tau_i'\}$ such that:

$$\tau_i' = \begin{cases} \tau_i(e_i, P_i), \text{ if } (P_i \geq k\pi + \phi) \\ \tau_i(e_i, k\pi + \phi), \text{ otherwise,} \end{cases} \tag{18}$$

Since the resource is under rejuvenation during the interval $[k\pi, k\pi + \phi)$ in the worst case, we have $sbf(P_{\min}) = k\theta$ for all $P_{\min} \in [k\pi, k\pi + \phi]$. Therefore, Γ' is still schedulable on R after the transformation.

Also, since the transformation increases the periods of some tasks, $U_{\Gamma'} < U_\Gamma$. Therefore, all task sets with $P_{\min} \in [k\pi, k\pi + \phi]$ can be transformed to Γ' with lower utilization. In other words, $P^* \in [k\pi + \phi, (k+1)\pi)$.

In the next step, we consider the task sets with $P_{\min} \in [k\pi + \phi, (k+1)\pi)$. We let Γ denote one of such task sets and transform $\Gamma = \{\tau_i\}$ to $\Gamma'' = \{\tau_i''\}$ such that

$$\begin{cases} e_i'' = q \cdot e_i \\ P_i'' = q \cdot P_i \end{cases} \tag{19}$$

where $q = \frac{P^*}{P_{\min}}$. Fig. 5(b) illustrates the transformation. After the transformation, $U_\Gamma = U_{\Gamma''}$ and $P''_{\min} = P^*$.

We first make an assumption that the transformed task set Γ'' is no longer schedulable. Based on this assumption, some e_i'' in Γ'' need to be decreased in order to make Γ'' schedulable. Let Γ''' denote the new schedulable task set, then $U_{\Gamma'''} < U_\Gamma$. Therefore, for a task set Γ with $P_{\min} \neq P^*$, we can always find a task set Γ''' that fully utilizes R with $U_{\Gamma'''} < U_\Gamma$. Moreover, as $P'''_{\min} = P''_{\min} = P^*$, we can then come to our conclusion that P^* is the minimal task period of the task set which fully utilizes R and has the minimal utilization.

In the following part, we derive the value of P^* by guaranteeing that the assumption is always true, i.e., Γ'' is not schedulable.

According to Lemma 3, Γ'' is not schedulable indicates

$$\sum_{\tau_i'' \in \Gamma''} e_i'' > sbf(P^*)$$

which can be further transformed into

$$\sum_{\tau_i'' \in \Gamma''} e_i'' = q \cdot \sum_{\tau_i \in \Gamma} e_i = q \cdot sbf(P_{\min}) > sbf(P^*)$$

and then

$$\frac{sbf(P^*)}{P^*} - \frac{sbf(P_{\min})}{P_{\min}} < 0 \tag{20}$$

As shown in Fig. 5(b), $P_{\min}$ can be represented as $k\pi + \phi + \delta$ over $\delta \in [0, \pi - \phi)$. To simplify the notation, we define function $\mathrm{F}(\delta)$ over $\delta \in [0, \pi - \phi)$ as

$$\mathrm{F}(\delta) = \frac{sbf(k\pi + \phi + \delta)}{k\pi + \phi + \delta} = \frac{a\delta^2/2 + (1 + a\phi - a\pi)\delta + k\theta}{k\pi + \phi + \delta}$$

With function $\mathrm{F}(\delta)$, Eq.(20) becomes $\mathrm{F}(\delta^*) - \mathrm{F}(\delta) < 0$ where $\delta^* = P^* - (k\pi + \phi)$.

Now, we derive the value of δ^* by the following condition:

$$\forall \delta \in [0, \pi - \phi), \mathrm{F}(\delta^*) \leq \mathrm{F}(\delta)$$

This can be done by solving the function $\mathrm{F}'(\delta) = 0$. If this function has real number solution, we let λ denote the solution, i.e.

$$\lambda = -a(\phi + k\pi) + ((a\phi + ak\pi)^2 - 2a((1 + a\phi - a\pi)(k\pi + \phi) - k\theta))^{\frac{1}{2}}$$

Since $\mathrm{F}''(\delta) > 0$, we have:

$$\delta^* = \begin{cases} 0, \ \textit{if } \lambda \leq 0 \\ \pi - \theta, \ \textit{if } \lambda > \pi - \theta \\ \lambda, \ \textit{otherwise} \end{cases} \tag{21}$$

If $\mathrm{F}'(\delta) = 0$ has no solution, it indicates $(1 + a\phi - a\pi)(k\pi + \phi) - k\theta > 0$, which guarantees $\mathrm{F}'(0) > 0$. Therefore, $\mathrm{F}(\delta)$ is monotonically increasing over $\delta \in [0, \pi - \phi)$. In this case, we have $\delta^* = 0$. To simplify the expression, we let $\lambda = -a(\phi + k\pi)$.

For both cases that $\mathrm{F}'(\delta) = 0$ has or has no solution, we calculate δ^* based on Eq.(21) as

$$\delta^* = \max\{\min\{\lambda, \pi - \phi\}, 0\}$$

where

$$\lambda = -a(\phi + k\pi) + ((a\phi + ak\pi)^2 - \min\{2a((1 + a\phi - a\pi)(\phi + k\pi) + k\theta), (a\phi + ak\pi)^2\})^{\frac{1}{2}}$$

Since F(δ^*) has the minimal value of function F(δ) over $\delta \in [0, \pi-\phi)$, when $P^* = \delta^* + k\pi + \phi$, Eq.(20) always holds, which further assures the assumption that Γ'' is not schedulable is true. Hence, we come to our conclusion that $P^* = \delta^* + k\pi + \phi$ is the minimal task period of the task set which fully utilizes R and has the minimal utilization. □

With Lemma 3 and Lemma 4, we derive the utilization bound UB_{RM} for a task set Γ under the RM scheduling policy with the restrictions that the ratio between any two task periods of Γ is less than 2.

Lemma 5. *Given a P^2-resource $R(f(t), \phi, \pi)$ and a task set Γ with minimal task period P_{min} and task number n. Under the restriction that the ratio between any two task periods of Γ is less than 2, the utilization bound of the task set Γ on R is:*

$$UB_{RM}(P_{min}, a, \pi, \phi) = \frac{\theta}{\pi} \cdot n[\left(1 + \frac{k\pi + \frac{\pi}{\theta} msf(\phi + \lambda, \pi, \phi)}{P^*}\right)^{1/n} - 1] \tag{22}$$

where $k = \lfloor \frac{P_{min}}{\pi} \rfloor$, $P^* = k\pi + \phi + \max\{\min\{\lambda, \pi - \phi\}, 0\}$ *and*

$$\lambda = -a(\phi + k\pi) + ((a\phi + ak\pi)^2 - \min\{2a((1 + a\phi - a\pi)(\phi + k\pi) + k\theta), (a\phi + ak\pi)^2\})^{\frac{1}{2}}$$

Proof. Without loss of generality, we assume that for the tasks in Γ, $P_1 < P_2 < ... < P_n$. Under the condition that Γ is schedulable and Γ fully utilizes R, let U^*_Γ denote the least schedulable utilization bound for Γ and let $e^*_1, e^*_2, .., e^*_n$ be the execution times of the tasks $\tau_1, \tau_2, ..., \tau_n$ that determine U^*_Γ. Then, according to Lemma 3, the execution times $e^*_1, e^*_2, ..., e^*_n$ is determined as follow:

$$e^*_1 = sbf(P_2) - sbf(P^*), ..., e^*_{n-1} = sbf(P_n) - sbf(P_{n-1})$$

Specially, $e^*_n = sbf(P_1) - sbf(0) - (sbf(P_n) - sbf(P_1))$.

According to Lemma 4, to find the minimal value of U^*_Γ, we let $P_1 = P^*$ where $P^* = k\pi + \phi + \max\{\min\{\lambda, \pi - \phi\}, \phi\}$, hence $e^*_n = k\theta + msf(\phi + \lambda, \pi, \phi) - sbf(P_n) + sbf(P^*)$. Then, U^*_Γ can be represented as:

$$\begin{aligned} U^*_\Gamma &= \frac{e^*_1}{P^*} + ... + \frac{e^*_{n-1}}{P_{n-1}} + ... + \frac{e^*_n}{P_n} \\ &= \frac{sbf(P_2) - sbf(P^*)}{P^*} + ... + \frac{sbf(P_n) - sbf(P_{n-1})}{P_{n-1}} \\ &+ \frac{k\theta + msf(\phi + \lambda, \pi, \phi) - sbf(P_n) + sbf(P^*)}{P_n} \end{aligned} \tag{23}$$

Furthermore, we replace $sbf(t)$ by $lsbf(t)$ and rewrite Eq.(23) as follows:

$$\begin{aligned} U^*_\Gamma &= \frac{lsbf(P_2) - lsbf(P^*)}{P^*} + ... + \frac{lsbf(P_n) - lsbf(P_{n-1})}{P_{n-1}} \\ &+ \frac{(lsbf(P^*) - lsbf(P_n)) + k\theta + msf(\phi + \lambda, \pi, \phi)}{P_n} \\ &= \frac{\theta}{\pi}(\frac{P_2}{P^*} + ... + \frac{P_n}{P_{n-1}} \\ &+ \frac{k\pi + \frac{\pi}{\theta} msf(\phi + \lambda, \pi, \phi) + P^*}{P_n} - n) \end{aligned} \tag{24}$$

Then, we calculate the extreme value of U^*_Γ by setting the first derivative of U^*_Γ with respect to each P_is equal to zero and by solving the resultant difference equations:

$$\partial U^*_\Gamma / \partial P_i = \frac{P_i^2 - P_{i-1} * P_{i+1}}{P_{i-1} \cdot P_i^2} = 0, i \in [2, n] \tag{25}$$

Since the second partial derivative is always larger than zero, the solution of Eq. (25) makes U^*_Γ minimal.

In the next step, we adopt the definition $P_{n+1} = (k\pi + \frac{\pi}{\theta} msf(\phi + \lambda, \pi, \phi) + P^*)$ for convenience. Then, Eq.(25) implies that $\forall i \in [2, n], \frac{P_i}{P_{i-1}} = \frac{P_{i+1}}{P_i}$ which means the sequence $\{P^*, P_2, ..., P_n\}$ is a geometric sequence. Therefore, the solution for Eq.(25) is

$$P_i = P^* * \left(1 + \frac{k\pi + \frac{\pi}{\theta} msf(\phi + \lambda, \pi, \phi)}{P^*}\right)^{-\frac{i-1}{n}} \tag{26}$$

With the solutions of P_is for U^*_Γ, we can then derive $UB_{\text{RM}}(P_{\min}, a, \pi, \phi)$ from Eq.(24) as:

$$UB_{\text{RM}}(P_{\min}, n, a, \pi, \phi) = \frac{\theta}{\pi} \cdot n[\left(1 + \frac{k\pi + \frac{\pi}{\theta} msf(\phi + \lambda, \pi, \phi)}{P^*}\right)^{1/n} - 1] \tag{27}$$

□

In Lemma 5, the restriction that the largest ratio between task period is less than 2 can be removed through the method introduced in the proof of Theorem 5 in [1]. Therefore, we have the closed form of the utilization bound in Theorem 5.

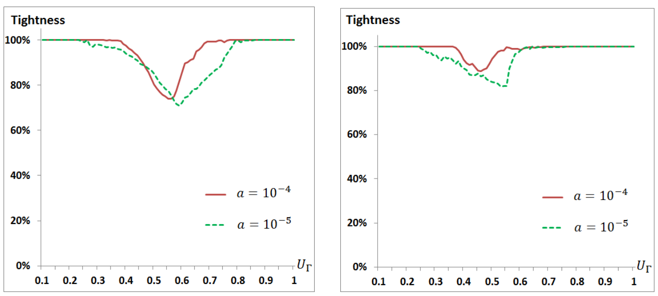
VII. Simulation Analysis

Section V and VI give the analytical utilization bound for real-time task sets on a P^2-resource under EDF and RM scheduling policies, respectively. In this section, we further study their tightnesses and the impacts of different factors on them through simulations.

A. Bound Tightness

Both UB_{EDF} and UB_{RM} are sufficient schedulability bounds of periodic task sets on P^2-resources. Therefore, it is possible that a task set with utilization higher than the bound is still schedulable. If a utilization bound is too conservative, many schedulable task sets will be measured as un-schedulable and thus the practical value of the utilization bound is low. To evaluate how conservative a bound is, we define an evaluatioin criteria, *Tightness*, as $\mathit{Tightness} = N_{same}/N_{total}$, where N_{total} is the total number of task sets that are tested and N_{same} is the number of task sets of which the schedulability determined by the utilization bound is the same as the schedulability determined by the corresponding scheduling policy.

In the following experiments, we measure the tightnesses of both $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ with different resource degradation rate a and task set utilization U_Γ. We use the UUnifast algorithm [36] to randomly generate 1000 task sets with utilizations ranging from 0.1 to 1.0. Each task set contains 4 tasks with periods ranging from 50 to 100. For the P^2-resource, we set its rejuvenation cost $\phi = 50$. As aging progress is slow [34], we set $a = 10^{-4}$ and $a = 10^{-5}$ for the two experiments, respectively, and set rejuvenation period $\pi = 1000$.



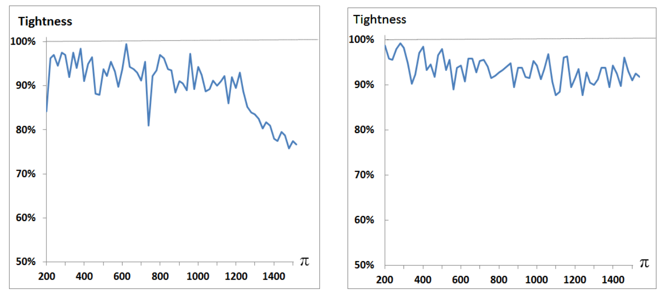
(a) $\mathrm{UB}_{\mathrm{EDF}}$ (b) $\mathrm{UB}_{\mathrm{RM}}$

Fig. 6. Tightness of the utilization bounds with different U_Γ and a values

As shown in the Fig. 6, with U_Γ increasing from 0.1 to 1.0, the tightnesses of both $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ share a similar changing pattern. For example, in the case of $a = 10^{-4}$ under EDF, before U_Γ increases to 0.4, the tightness of $\mathrm{UB}_{\mathrm{EDF}}$ stays at one, which means the schedulability determined by the bound is the same as the schedulability determined by the EDF policy for all of the 1000 task sets. When $0.4 \leq U_\Gamma < 0.8$, the tightness of $\mathrm{UB}_{\mathrm{EDF}}$ decreases first and then increases to one again. When $U_\Gamma > 0.8$, the tightness again stays at one. One possible explanation of this pattern is that when U_Γ is low, the utilization bounds are relatively high, hence most of the task sets are determined as schedulable by both bound and scheduling policy. On the contrary, when U_Γ is sufficiently high, most of the task sets are determined as un-schedulable by both bound and scheduling policy. Therefore, in both cases, the tightnesses are high. However, if U_Γ is in a certain range, such as $[0.4, 0.8]$ for the case $a = 0.4$ under EDF policy, a schedulable task set is more likely to be determined as un-schedulable. Therefore, when U_Γ is not sufficiently low or high, the tightness is relatively low.

Another interesting observation is that, when a value increases, both $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ becomes tighter. In addition, $\mathrm{UB}_{\mathrm{RM}}$ is tighter than $\mathrm{UB}_{\mathrm{EDF}}$ in the main trend.

Next, we evaluate the tightnesses of the both bounds with different π values. We use the same configuration of task set in the previous experiment but set $U_\Gamma = 0.5$ and performance degradation rate $a = 10^{-5}$. We then measure the tightnesses for both $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ with different π values ranging from 200 to 1500. As are depicted in Fig. 7, when π value increases, the tightnesses of both $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ decrease in general.



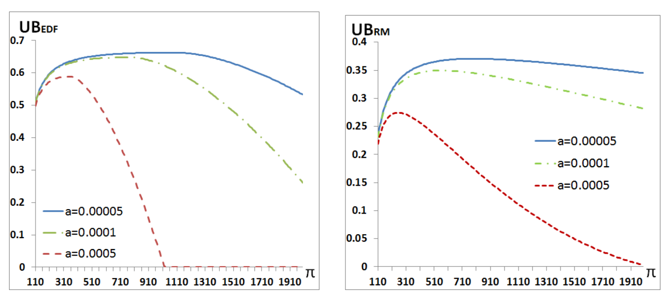
(a) $\mathrm{UB}_{\mathrm{EDF}}$ (b) $\mathrm{UB}_{\mathrm{RM}}$

Fig. 7. Tightness of the utilization bounds with different π values

The theoretical analysis of the bound's tightness is beyond the scope of this paper, we will continue analyzing the phenomenons illustrated above in our future work.

B. Impacts of π and a values on the utilization bounds

As aforementioned, $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ are determined by multiple factors. The impact of a factor can be evaluated by calculating the first derivative of of the factor in $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ formula. However, for factor a and π, the calculation of their first derivatives are complicated, hence we evaluate their impacts on both bounds by simulations instead.



(a) $\mathrm{UB}_{\mathrm{EDF}}$ (b) $\mathrm{UB}_{\mathrm{RM}}$

Fig. 8. Impact of a and π

We set $\phi = 50$, $P_{\min} = 100$, $n = 4$ and calculate $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ under different π and a values. As shown in Fig. 8, both bounds decrease when a increases, which matches the intuition that resources with faster performance degradation can only support task sets with lower utilizations.

In addition, as π increases, both utilization bounds show the pattern of growing up first, reaching its maximum and then decreasing. This observation raises a question that under what π value, the utilization bound reaches its maximum. For a P^2-resource, changing the performance degradation rate or rejuvenation cost is difficult, if not impossible, since they are determined by the software and hardware infrastructure. However, the rejuvenation period is configurable. Therefore, how to determine the rejuvenation period π is critical to the performance of a P^2-resource in a real-time system. Our future research will focus on how to determine the rejuvenation period to maximize the utilization bound for a P^2-resource under both EDF and RM policies.

VIII. CONCLUSION

In this paper, we have three major contributions: 1) Defined the P^2-resource model and provided its supply bound analysis; 2) provided the closed form of the utilization bounds for a task set on a P^2-resource under both EDF and RM scheduling policies, respectively; and 3) studied the tightnesses of the two utilization bounds and the impacts of different factors on the two bounds as well by simulations.

In order to simplify the study, we assume the performance degradation function of a P^2-resource is linear, which is not always held in the real world systems. In our future works, we will remove this assumption and study the P^2-resources with non-linear performance degradation functions. Meanwhile, as we mentioned before, we will theoretically analyze the tightnesses of both bounds. Also, we will study the scenario that during the resource rejuvenation, instead of stoping providing service, the resource is still available to the tasks but in the minimum performance.

REFERENCES

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

[2] Z. Kunpeng, S. Zhiyi, The measurement and data acquisition system for the aging test of the composite insulators, in: Power System Technology, 2002. Proceedings. PowerCon 2002. International Conference on, Vol. 3, IEEE, 2002, pp. 1863–1866.

[3] A. Norman, The nova data acquisition system, Fermilab, 2012. URL http://www-nova.fnal.gov/index.html,https://indico.cern.ch/event/149557/session/2/contribution/529/material/slides/1.pdf

[4] A. Tai, L. Alkalai, S. Chau, On-board preventive maintenance for long-life deep-space missions: a model-based analysis, in: Computer Performance and Dependability Symposium, 1998. IPDS '98. Proceedings. IEEE International, 1998, pp. 196–205. doi:10.1109/IPDS.1998.707722.

[5] A. Tai, L. Alkalai, On-board maintenance for long-life systems, in: Application-Specific Software Engineering Technology, 1998. ASSET-98. Proceedings. 1998 IEEE Workshop on, 1998, pp. 69–74. doi:10.1109/ASSET.1998.688236.

[6] K. Tomsovic, D. E. Bakken, V. Venkatasubramanian, A. Bose, Designing the next generation of real-time control, communication, and computations for large power systems, Proceedings of the IEEE 93 (5) (2005) 965–979.

[7] J. Davidson, F.-C. Bouchart, Adjusting nodal demands in scada constrained real-time water distribution network models, Journal of Hydraulic Engineering 132 (1) (2006) 102–110.

[8] http://www.ni.com/labview/.

[9] Y. Huang, C. Kintala, N. Kolettis, N. D. Fulton, Software rejuvenation: Analysis, module and applications, in: Fault-Tolerant Computing, 1995. FTCS-25. Digest of Papers., Twenty-Fifth International Symposium on, IEEE, 1995, pp. 381–390.

[10] K. S. Trivedi, K. Vaidyanathan, K. Goseva-Popstojanova, Modeling and analysis of software aging and rejuvenation, in: Simulation Symposium, 2000.(SS 2000) Proceedings. 33rd Annual, IEEE, 2000, pp. 270–279.

[11] C. Guo, H. Wu, X. Hua, S. Ren, J. Nogiec, Maximize system reliability for long lasting and continuous applications 353 (2015) 603–612.

[12] A. Bovenzi, J. Alonso, H. Yamada, S. Russo, K. S. Trivedi, Towards fast os rejuvenation: An experimental evaluation of fast os reboot techniques, in: Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium on, IEEE, 2013, pp. 61–70.

[13] J. Alonso, R. Matias, E. Vicente, A. Maria, K. S. Trivedi, A comparative experimental study of software rejuvenation overhead, Performance Evaluation 70 (3) (2013) 231–250.

[14] M. Fan, G. Quan, Harmonic semi-partitioned scheduling for fixed-priority real-time tasks on multi-core platform, in: Proceedings of the Conference on Design, Automation and Test in Europe, EDA Consortium, 2012, pp. 503–508.

[15] L. Niu, G. Quan, A hybrid static/dynamic dvs scheduling for real-time systems with (m, k)-guarantee, in: Real-Time Systems Symposium, 2005. RTSS 2005. 26th IEEE International, IEEE, 2005, pp. 10–pp.

[16] J. A. Stankovic, Deadline scheduling for real-time systems: EDF and related algorithms, Springer Science & Business Media, 1998.

[17] O. U. P. Zapata, P. M. Alvarez, Edf and rm multiprocessor scheduling algorithms: Survey and performance evaluation, Seccion de Computacion Av. IPN 2508.

[18] S. K. Baruah, L. E. Rosier, R. R. Howell, Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor, Real-Time Systems 2 (4) (1990) 301–324.

[19] K. Jeffay, D. F. Stanat, C. U. Martel, On non-preemptive scheduling of period and sporadic tasks, in: Real-Time Systems Symposium, 1991. Proceedings., Twelfth, IEEE, 1991, pp. 129–139.

[20] J. Y. Leung, Handbook of scheduling: algorithms, models, and performance analysis, CRC Press, 2004.

[21] S. Saha, B. Ravindran, An experimental evaluation of real-time dvfs scheduling algorithms, in: Proceedings of the 5th Annual International Systems and Storage Conference, ACM, 2012, p. 7.

[22] G. Von Laszewski, L. Wang, A. J. Younge, X. He, Power-aware scheduling of virtual machines in dvfs-enabled clusters, in: Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on, IEEE, 2009, pp. 1–10.

[23] E. Bini, G. Buttazzo, G. Lipari, Minimizing cpu energy in real-time systems with discrete speed management, ACM Transactions on Embedded Computing Systems (TECS) 8 (4) (2009) 31.

[24] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, M. L. Scott, Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling, in: High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on, IEEE, 2002, pp. 29–40.

[25] A. B. Nagarajan, F. Mueller, C. Engelmann, S. L. Scott, Proactive fault tolerance for hpc with xen virtualization, in: Proceedings of the 21st annual international conference on Supercomputing, ACM, 2007, pp. 23–32.

[26] D. L. Parnas, Software aging, in: Proceedings of the 16th international conference on Software engineering, IEEE Computer Society Press, 1994, pp. 279–287.

[27] K. Vaidyanathan, K. S. Trivedi, A comprehensive model for software rejuvenation, Dependable and Secure Computing, IEEE Transactions on 2 (2) (2005) 124–137.

[28] S. Shirero, M. Takashi, H. Kei, On the schedulability conditions on partial time slots, in: Real-Time Computing Systems and Applications, 1999. RTCSA'99. Sixth International Conference on, IEEE, 1999, pp. 166–173.

[29] A. K. Mok, X. Feng, D. Chen, Resource partition for real-time systems, in: Real-Time Technology and Applications Symposium, 2001. Proceedings. Seventh IEEE, IEEE, 2001, pp. 75–84.

[30] X. Feng, Design of real-time virtual resource architecture for large-scale embedded systems.

[31] A. Easwaran, M. Anand, I. Lee, Compositional analysis framework using edp resource models, in: Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International, IEEE, 2007, pp. 129–138.

[32] I. Shin, I. Lee, Compositional real-time scheduling framework with periodic model, ACM Transactions on Embedded Computing Systems (TECS) 7 (3) (2008) 30.

[33] J. Kim, K. Lakshmanan, R. R. Rajkumar, Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems, in: Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems, IEEE Computer Society, 2012, pp. 55–64.

[34] L. I. Millett, M. Thomas, D. Jackson, et al., Software for Dependable Systems:: Sufficient Evidence?, National Academies Press, 2007.

[35] X. Hua, H. W. Chunhui Guo, S. R. Douglas Lautner, Tech report: Schedulability analysis for real-time task set on resource with performance degradation and periodic rejuvenation, http://gauss.cs.iit.edu/~code/techreports/DecayResource.pdf/.

[36] E. Bini, G. C. Buttazzo, Biasing effects in schedulability measures, in: Real-Time Systems, 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on, IEEE, 2004, pp. 196–203.