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

Xiayu Hua, Chunhui Guo, Hao Wu
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616, USA
{xhua, hwu28, cguo}@hawk.iit.edu

Douglas Lautner
Motorola
Chicago, IL 60654, USA
dlautner@hawk.iit.edu

Shangping Ren
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616, USA
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 system, 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 analysis under a resource model when 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 performace degradation and periodic rejuvenation. We address three real-task schedulability related questions under the P^2-resource model, i.e., (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.**

I. Introduction

Since the publication of the seminal paper by Liu and Layland [1] in 1973, real-time task scheduling problem 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 system(DAQ) [2], [3], [4], [5], deep-space exploration programs [6], and SCADA systems for power, water and other national infrastructures [7], [8], [9], the performance of computational resources decrease notably after a long continuous execution period.

We also collected over a twenty-day period the CPU and memory usages consumed by a data monitoring software deployed on a Fermilab control system. It is depicted in Fig. 1. As shown in the figure, both the CPU consumption and memory consumption by the data monitoring software increase with time. As the data monitoring software is the only application deployed on the computer, under normal operation, both the CPU consumption and memory consumption used by the software would remain at a constant level. However, as the figure indicates, when system's continuous operation time increases, the application consumes more resources. In other words, the amount of computational power provided by system in a unit time keep decreasing when the system is running.

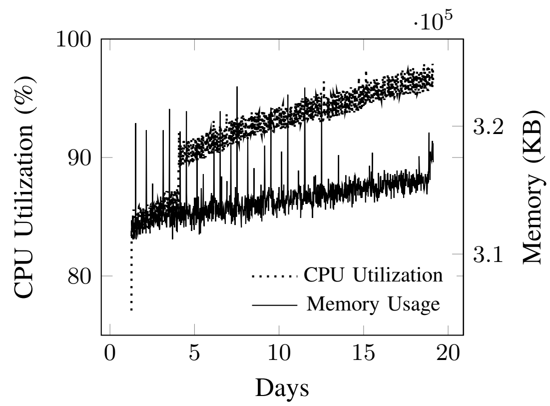


Fig. 1. Aging Effect on Fermi Monitoring System

The root cause of this phenomenon is software error accumulation and memory leak, which is also referred to as software aging problem [10]. 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 [10]. 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 the application stops working eventually.

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 [11] techniques are introduced as the countermeasure to recover the resource performance. However, software rejuvenation also introduces extra overhead. In fact, when resources are performing software rejuvenation, they are not available to the applications. Hence, from application's perspective, such resources are only periodically available and even within their available time, the resource performances are continuously decreasing. To reflect these characteristics, we introduce a new resource model, the P^2-resource, to characterize resources with performance degradation and periodic 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 EDF and RM scheduling policies on a P^2-resource are presented in Section V and Section VI, respectively. We further experimentally study the tightness of the two bounds and the impacts of different resource factors on the two bounds in Section VII. We conclude our work in Section VIII.

II. Related Work

In 1973, Liu and Layland [1] 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. 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 problem [12], [13] and improving utilization bounds for both EDF and RM scheduling policies on single [14], [15] and multiple processors [16], [17], [18] under different constraints (preemptive[19] vs non-preemptive [20]) and for different task models (synchronous vs asynchronous tasks, harmonic task set [12], mixed-criticality task set [21],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 [22], [23]. Hence, in a DVFS-available system, the resource model is changed from the continues and constant resource model to a continues 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 [22], [23], [24]. In the literature, task schedulability study under the DVFS resource model makes two major assumptions: (1) computing resource performance can switch between a finite number of levels and once it is switched to a level, it remains at the level until the DVFS switch it to another level [25], [26], and (2) the performance change via DVFS is controllable and voluntary.

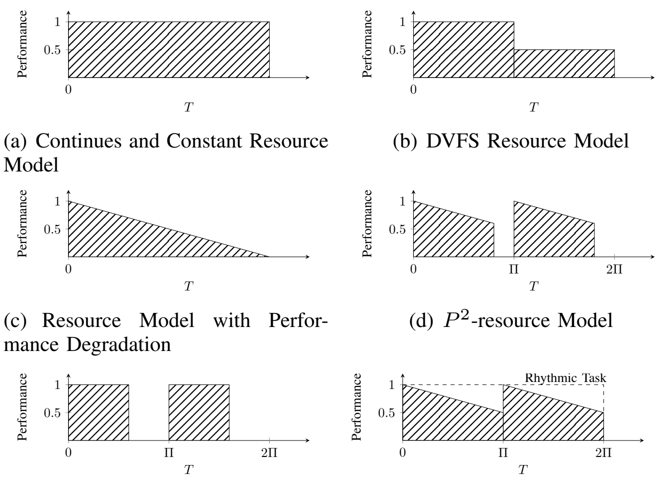
On the other hand, as pointed in [27], [28], software aging problem is essentially error accumulation and memory leaking caused by software defects which are difficult to eradicate if at all. The computing resource performance degradation caused by software aging is progressive and involuntary as shown in Fig. 2(c). Therefore, neither the continuous and constant resource model nor the DVFS resource model is sufficient for abstract the resource with performance degradation caused by software aging process.

Due to the resource performance degradation, the application will eventually stop working, which is unacceptable by any systems. Hence, software rejuvenation technology is introduced as a countermeasure [10], [11], [29]. Through rejuvenation, systems regain 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-resourceis a periodic resource. The concept of the constant periodic resource was first introduced by Shigero *et al.* in 1999 [30]. Mok *et al.* [31] and Feng *et al.* [32] 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 [33], [34], [35]. 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 (we refer this resource model as CP-resource model) can be directly applied under the P^2-resource model.

If considering the resource performance degradation as a special periodic task with increased utilization, we can transform a resource with performance degradation as a periodic resource with constant performance and with a hidden task that has increased resource consumption running on the resource. Fig. 2(f) depicts this scenario. A similar case is studied in [36] where the author call this special task as 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 [37]. 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 easily transformed to the continuous and constant resource model [1] and constant periodic model [34]. In next section, we formally define the P^2-resourcemodel and formulate the problems to be studied in the paper.



(a) Continues 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

III. Models and Problem Formulation

A. Resource model and assumptions

As illustrated in Fig.2(c), we consider both the performance degradation and the rejuvenation time cost to 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 of this 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 that $f(0) = 1$.

Resource Rejuvenation

We assume that the resource is repeatly rejuvenated with period π and that the resource performance never decreases to zero, i.e., $f^{-1}(\pi) > 0$. 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 resource *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. When the resource performance is $f(t)$ at time t, the task's actual worst-case execution time is $e_i/f(t)$. 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 may not be schedulable no matter how small its utilization is, as ϕ is the resource unavailable time due to rejuvenation for each rejuvenation period. Hence, we assume $P_{\text{min}} > \phi$.

B. Problem formulation

The paper is 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 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 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 is the same. To simplify mathematical transformations and deviations, and focus more on analysis strategies, in 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*

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

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 ϕ time, 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 resource's minimal supply is 0 in this case.
Case 2: $\phi \leq t \leq \pi$

As 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. □

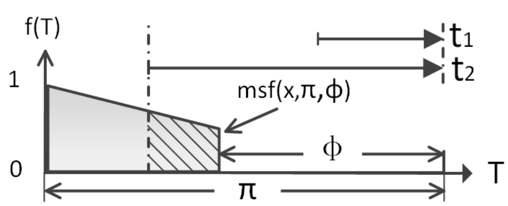


Fig. 3. Minimal Supply Function

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*

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

Proof. For a given time interval t, it contains $\lfloor \frac{t}{\pi} \rfloor$ entire periods that supply $\lfloor \frac{t}{\pi} \rfloor \theta$ amount of computational cycles resource. For the remaining part of the time interval, its length is (t mod π) and its minimal supply is $msf(t \bmod \pi, \pi, \phi)$. 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) satisfies

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

where $lsbf(t) = \frac{\theta}{\pi}(t - Tp) + msf(Tp, \pi, \phi)$, and $Tp = \max\{\pi - \frac{\pi-\theta}{a\pi}, \phi\}$, i.e., the lsbf$(t)$ is the lower bound of the sbf(t), and

(2) $\exists t : lsbf(t) = sbf(t)$, i.e., the lsbf$(t)$ a tight bound of the sbf(t).

Proof. To prove the theorem, we consider when $n\pi \leq t < n\pi + \phi$ and $n\pi + \phi \leq t < (n+1)\pi$ separately, where $n \in I$.

1) When $n\pi \leq t < n\pi + \phi$

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

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

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

- **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

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

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

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

Since for both cases, $lsbf(n\pi + \phi) \leq n\theta$, we have $lsbf(t) \leq sbf(t)$ 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 $sbf(t) - lsbf(t) \geq 0$. To simplify the notation, we let $t' = t \bmod \pi$.

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

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

Since $\mathrm{Tp} = \phi$, 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 $t' = \mathrm{Tp} = \phi$, $sbf(t) = lsbf(t)$.

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

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

Let $S(t) = \mathit{msf}(t, \pi, \phi) - \frac{\theta \cdot t}{\pi}$, 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,

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

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

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

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

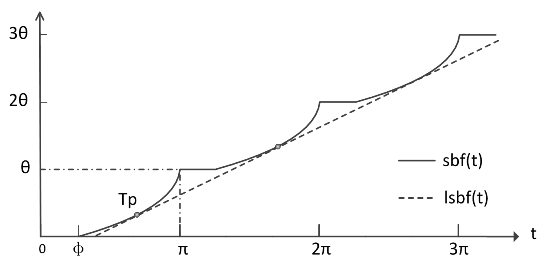


Fig. 4. SBF and LSBF

V. Task Set Utilization Bound on P^2-resource under 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 the paper is the same as the task model used in [34], the linear demand bound function (LDBF) for a task set and the schedulability condition under EDF scheduling policy are the same. We provide the tasks' linear demand bound function and the schedulability condition in the following definition and theorem, respectively.

Definition 1. *[34] Given a task set Γ, its linear demand bound function (LDBF) under EDF scheduling policy is defined as*

$$\mathit{ldbf}_{EDF}(t) = U_\Gamma \cdot t. \tag{7}$$

Theorem 3. *[34] 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*

$$\forall t \in [0, H] : \mathit{dbf}_{EDF}(t) \leq \mathit{sbf}(t) \tag{8}$$

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

To simplify the calculation, in the following corollary, we further reduce the range of time interval length t that 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*

$$\forall t \in [P_{min}, H] : \mathit{ldbf}_{EDF}(t) \leq \mathit{lsbf}(t) \tag{9}$$

where H is the hyper-period of Γ.

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

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

$$\forall t : \text{ldbf}_{\text{EDF}}(t) \leq \mathit{lsbf}(t) \rightarrow \mathit{dbf}_{\text{EDF}}(t) \leq \mathit{sbf}(t)$$

Therefore, Γ is schedulable on R under EDF policy if Eq.(9) holds. □

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

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

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 EDF scheduling policy is defined as*

$$Tc = \frac{\frac{\theta}{\pi} \cdot Tp - \mathit{msf}(Tp, \pi, \phi)}{\frac{\theta}{\pi} - U_\Gamma} \tag{10}$$

In fact, the *critical time interval length* Tc is derived from equation $\mathit{lsbf}(t) = \mathit{sbf}(t)$. 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 in 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 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:*

$$\begin{cases} Tc > 0 \\ \forall t \geq Tc : \mathit{lsbf}(t) \geq \mathit{ldbf}_{EDF}(t). \end{cases} \tag{11}$$

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

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

Therefore, conditions in Eq.(11) hold. □

With the critical time interval length and the schedulability condition given in Corollary 1, we derive the utilization bound for a task set on a P^2-resource under 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*

$$UB_{EDF}(P_{min}, a, \pi, \phi) = \frac{\theta}{\pi} - \frac{\frac{\theta}{\pi} \cdot Tp - msf(Tp, \pi, \phi)}{P_{min}} \tag{12}$$

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 $\forall t \in [P_{\min}, H] : \mathrm{ldbf}_{\mathrm{EDF}}(t) \leq lsbf(t)$. Based on Lemma 2, we have $\forall t \geq \mathrm{Tc} : \mathrm{ldbf}_{\mathrm{EDF}}(t) \leq lsbf(t)$. Hence, the task set Γ is guaranteed schedulable on the resource R if $P_{\min} \geq \mathrm{Tc}$. By solving the formula $P_{\min} = \mathrm{Tc}$, we derive the utilization bound $\mathrm{UB}_{\mathrm{EDF}}$ as below:

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

□

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 $UB_{\mathrm{EDF}}(P_{\min}, a, \pi, \phi)$ becomes the utilization bound given by Liu and Layland [1], i.e., $UB_{\mathrm{EDF}}(P_{\min}, a, \pi, \phi) = 1$.

Corollary 2. *Given a task set* Γ *and a* P^2*-resource* $R(1, \pi, 0)$*, the task utilization bound under 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 $\mathrm{Tp} \geq \phi$, $lsbf(\mathrm{Tp}) = \mathrm{Tp} - \phi = \mathrm{Tp}$. Based on Lemma 2, Tp always exists. Hence,

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

□

VI. Task Set Utilization Bound on P^2-resource under 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. We first 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 of* n *and minimal task period* P_{min}*. The utilization bound of the task set* Γ *on* R *under RM scheduling policy is:*

$$\begin{aligned} &UB_{RM}(P_{min}, n, a, \pi, \phi) \\ &= \frac{\theta}{\pi} \cdot n\Big[\Big(1 + \frac{k\pi + \frac{\pi}{\theta} msf(\phi + \delta, \pi, \phi)}{k\pi + \phi + \delta}\Big)^{1/n} - 1\Big] \end{aligned} \tag{13}$$

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

The following parts of this section are dedicated to proving 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, then 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 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*

$$\sum_{\tau_i \in \Gamma} e_i = sbf_R(P_{min}) \tag{14}$$

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

Proof. We adapt the proof of Lemma 9.1 in [34] to prove this lemma. 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, we first need to show that:

$$e^*_1 = \mathit{sbf}(P_1, P_2)$$

where we define $\mathit{sbf}(P_1, P_2) = \mathit{sbf}(P_2) - \mathit{sbf}(P_2)$ for notational simplicity. We then prove this by contradiction.

Assume that

$$e^*_1 = \mathit{sbf}(P_1, P_2) + \Delta \tag{15}$$

Let $e'_1 = \mathit{sbf}(P_1, P_2)$, $e'_2 = e^*_2 + \Delta$, $e'_3 = e^*_3$, ... $e'_n = e^*_n$. Given that $e^*_1, e^*_2, ..., e^*_n$ guarantee the schedulability of task set Γ under RM and that any increase in e^*_i will make Γ unschedulable, it is clear that a workload set with $e'_1, ..., e'_n$ is schedulable over R and that any increase in e'_i will violate the schedulability of the task set over R. Let U'_Γ denote the corresponding utilization, we have

$$U^*_\Gamma - U'_\Gamma = (\Delta/P_1) - (\Delta/P_2) > 0$$

Hence, this assumption given in Eq. (15) is false when $\Delta > 0$. Alternatively, suppose that

$$e^*_1 = \mathit{sbf}(P_1, P_2) - \Delta, \Delta > 0 \tag{16}$$

Let $e''_1 = \mathit{sbf}(P_1, P_2)$, $e''_2 = e^*_2 - 2\Delta$, $e''_3 = e^*_3$, ..., $e''_n = e^*_n$. Then, a workload set with $e''_1, e''_2, ..., e''_n$ is also schedulable over R and any ncrease in e''_i will violate the schedulability of the task set. Let U'' denote the corresponding utilization. we have

$$U^*_\Gamma - U'_\Gamma = -(\Delta/P_1) + 2(\Delta/P_2) > 0$$

Hence, this assumption given in Eq. (16) is also false, when $\Delta > 0$. Therefore, if indeed U^*_Γ is the least upper bound of the workload utilization, then $e^*_1 = \mathit{sbf}(P_1, P_2)$. In a similar way, we can show that $e^*_2 = \mathit{sbf}(P_2, P_3)$, $e^*_3 = \mathit{sbf}(P_3, P_4)$,..., $e^*_{n-1} = \mathit{sbf}(P_{n-1}, P_n)$. Consequently,

$$\begin{aligned} e^*_n &= \mathit{sbf}(0, P_n) - 2(e^*_1 + e^*_2 + ... + e^*_{n-1}) \\ &= \mathit{sbf}(0, P_1) - \mathit{sbf}(P_1, P_n) \end{aligned}$$

Finally, we have $\sum_{\tau_i \in \Gamma} e_i = \mathit{sbf}_R(P_{\min})$.

□

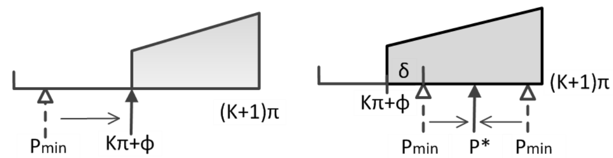
Lemma 4. *Given a real-time task set $\Gamma = \{\tau_1, ..., \tau_n\}$ and a resource $R(f(t), \phi, \pi)$, let $k = \lfloor \frac{P_{min}}{\pi} \rfloor$ and let P_{min} denote the smallest task period of Γ. Under the restriction that the ratio between any two task periods of Γ is less than two, if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

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

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

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

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 $\mathit{sbf}(P_{\min}) = k\theta$ for all $P_{\min} \in [k\pi, k\pi + \phi]$. Therefore, Γ' is still schedulable on R and fully utilize R after the transformation.

On the other hand, 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 $q = \frac{P^*}{P_{\min}}$. After the transformation, $U_\Gamma = U_{\Gamma''}$ and $P''_{\min} = P^*$.

We first make a 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 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 $\mathrm{F}(\delta)$ for $\delta\in[0,\pi-\phi)$ 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:

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

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

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

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

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

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 not solution, we calculate δ^* based on Eq.(21) as

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

where

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

Since $\mathrm{F}(\delta^*)$ has the minimal value of function $\mathrm{F}(\delta)$ 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 $\mathrm{UB_{RM}}$ for a task set Γ under the RM scheduling policy under 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:*

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

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

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

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,

$$\begin{aligned}e^*_n&=sbf(P^*)-sbf(0)-(sbf(P_n)-sbf(P^*))\\&=k\theta+msf(\phi+\lambda,\pi,\phi)-sbf(P_n)+sbf(P^*)\end{aligned}$$

Therefore, U^*_Γ can be represented as:

$$\begin{aligned}U^*_\Gamma&=\frac{e^*_1}{P_1}+...+\frac{e^*_{n-1}}{P_{n-1}}+...+\frac{e^*_n}{P_n}\\&=\frac{sbf(P_2)-sbf(P_1)}{P_1}+...+\frac{sbf(P_n)-sbf(P_{n-1})}{P_{n-1}}\\&+\frac{k\theta+msf(\phi+\lambda,\pi,\phi)-sbf(P_n)+sbf(P^*)}{P_n}\end{aligned} \tag{23}$$

According to Lemma 4, to find the minimal value of U^*_Γ, we let $P_1=P^*$ where $P^*=k\pi+\phi+\max\{\min\{\lambda,\pi-\phi\},\phi\}$. Furthermore, we replace $sbf(t)$ by $lsbf(t)$ and rewrite Eq.(23) as follows:

$$\begin{aligned}U^*_\Gamma&=\frac{lsbf(P_2)-lsbf(P^*)}{P^*}+...\\&+\frac{lsbf(P_n)-lsbf(P_{n-1})}{P_{n-1}}+\frac{2lsbf(P^*)-lsbf(P_n)}{P_n}\\&=\frac{\theta}{\pi}(\frac{P_2}{P^*}+...+\frac{P_n}{P_{n-1}}\\&+\frac{k\pi+\frac{\pi}{\theta}msf(\phi+\lambda,\pi,\phi)+P^*}{P_n}-n)\end{aligned} \tag{24}$$

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

$$\partial U_\Gamma^*/\partial P_i = \frac{P_i^2 - P_{i-1} * P_{i+1}}{P_{i-1} \cdot P_i^2} = 0, i \in [2, n] \quad (25)$$

The definition $P_{n+1} = (k\pi + \frac{\pi}{\theta} msf(\phi + \lambda, \pi, \phi) + P^*)$ has been adopted for convenience. 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

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

With the solutions of P_is for U_Γ^*, we can then derive $\mathit{UB}_{\mathrm{RM}}(P_{\min}, a, \pi, \phi)$ from Eq.(24) as:

$$\begin{aligned} &\mathit{UB}_{\mathrm{RM}}(P_{\min}, a, \pi, \phi) \\ &= \frac{\theta}{\pi} \cdot n[\left(1 + \frac{k\pi + \frac{\pi}{\theta} msf(\phi + \lambda, \pi, \phi)}{P^*}\right)^{1/n} - 1] \end{aligned} \quad (27)$$

□

The restriction that the largest ratio between task period is less than 2 in Lemma 5 can be removed through 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.

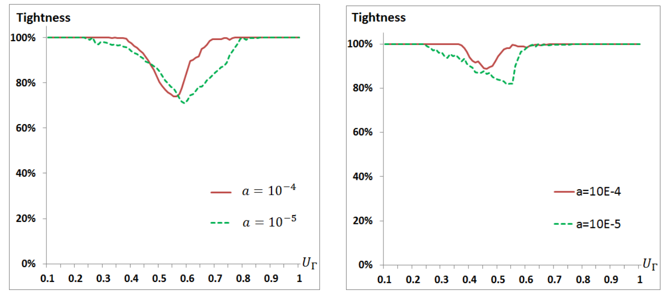
A. Bound Tightness

Both $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ 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 below:

$$\mathit{Tightness} = \frac{N_{same}}{N_{total}}$$

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 UUnifast algorithm [38] 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 [37], hence, 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 EDF scheduling algorithm for all of the 1000 task sets. When $0.4 \le U_\Gamma < 0.8$, the tightness of $\mathrm{UB}_{\mathrm{EDF}}$ decreases first and then increase 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 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, schedulable task sets are 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.

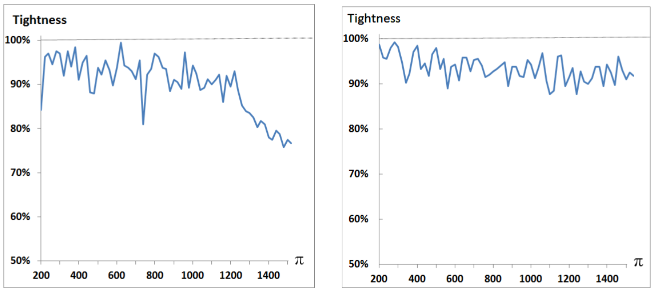
Next, we evaluate the tightnesses of the both bounds with different π values. We use the same configuration of task set in the previous experiment but set $U_\Gamma = 0.5$, set performance degradation rate $a = 10^{-5}$ and measure the tightnesses for both $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ 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.

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 futhre work.

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

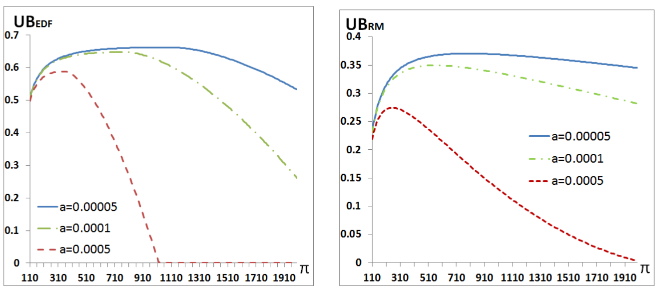
As aforementioned, $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ are determined by multiple factors. The impact of a factor can be evaluated by calculating the first derivative of of the factor in $\mathrm{UB}_{\mathrm{EDF}}$ and $\mathrm{UB}_{\mathrm{RM}}$ formula. However, for factor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

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,



(a) UB_{EDF} (b) UB_{RM}

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



(a) UB_{EDF} (b) UB_{RM}

Fig. 8. Impact of a and π

both bounds decrease when a increases, which matches the intuition that resource 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 rises 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. We will also focus on the research issue of finding the optimal rejuvenation period for a P^2-resource to maximize its utilization bounds under EDF and RM policies, respectively.

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] X. Hua, H. Wu, S. Ren, Enhancing throughput of hadoop distributed file system for interaction-intensive tasks, in: Proceedings of the 2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, IEEE Computer Society, 2014, pp. 508–511.

[4] 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

[5] X. Hua, H. Wu, Z. Li, S. Ren, Enhancing throughput of the hadoop distributed file system for interaction-intensive tasks, Journal of Parallel and Distributed Computing 74 (8) (2014) 2770–2779.

[6] 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.

[7] 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.

[8] 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.

[9] S. A. Boyer, SCADA: supervisory control and data acquisition, International Society of Automation, 2009.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] E. Bini, G. Buttazzo, A hyperbolic bound for the rate monotonic algorithm, in: Real-Time Systems, 13th Euromicro Conference on, 2001., IEEE, 2001, pp. 59–66.

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

[16] J. M. López, M. García, J. L. Diaz, D. F. Garcia, Worst-case utilization bound for edf scheduling on real-time multiprocessor systems, in: Real-Time Systems, 2000. Euromicro RTS 2000. 12th Euromicro Conference on, IEEE, 2000, pp. 25–33.

[17] J. M. López, J. L. Díaz, D. F. García, Utilization bounds for edf scheduling on real-time multiprocessor systems, Real-Time Systems 28 (1) (2004) 39–68.

[18] X. Hua, Z. Li, H. Wu, S. Ren, Scheduling periodic tasks on multiple periodic resources, in: INFOCOMP 2014, The Fourth International Conference on Advanced Communications and Computation, 2014, pp. 35–40.

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] S. Eyerman, L. Eeckhout, Fine-grained dvfs using on-chip regulators, ACM Transactions on Architecture and Code Optimization (TACO) 8 (1) (2011) 1.

[25] T. D. Burd, T. A. Pering, A. J. Stratakos, R. W. Brodersen, A dynamic voltage scaled microprocessor system, Solid-State Circuits, IEEE Journal of 35 (11) (2000) 1571–1580.

[26] 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.

[27] 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.

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

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

[30] 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.

[31] 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.

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

[33] 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.

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

[35] N. Fisher, F. Dewan, A bandwidth allocation scheme for compositional real-time systems with periodic resources, Real-Time Systems 48 (3) (2012) 223–263.

[36] 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.

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

[38] 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.