Schedulability Analysis for Real-Time Task Set on Resource with Performance Degradation and Dual-Level Periodic Rejuvenations

Xiayu Hua, Chunhui Guo, Hao Wu, Douglas Lautner, and Shangping Ren

Abstract—Researches in real-time scheduling often assume that the performance of a computing resource does not change over time. However, as system softwares and system architectures become increasingly complex, resource performance degradation over time becomes more evident. In this paper, we study the schedulability of a hard real-time task set on a resource which has performance degradation over time with a known pattern and use both cold and warm periodic rejuvenations as countermeasures. Such resource model is referred to as P^2D-resource model for performance degradation and periodic rejuvenation with dual-levels. In this paper, we study (1) the formal specification of the P^2D-resource model, (2) P^2D-resource supply analysis, and (3) task set utilization bounds of a P^2D-resource under Earliest Deadline First (EDF) and Rate Monotonic (RM) scheduling policies.

Index Terms—Real-time system, real-time scheduling, software aging, software rejuvenation, periodic resource

1 Introduction

Since the publication of the seminal paper by Liu and Layland in 1973 [1], the problem of real-time task scheduling under different resource models has been studied intensively. However, many of the studies rely on a strong assumption that the performance of a computing resource does not change during its lifetime. Unfortunately, for many long-running 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 decreases notably after a long and continuous execution period.

Over a 20-day period we collected the CPU and memory usage data of a monitoring software [8] deployed on Fermilab control system which has a stable application workload. As shown in Fig. 1, both CPU and memory consumptions increase with time, which indicates that the system performance, i.e., the amount of computational power provided by the system in a unit time, keeps decreasing when the system runs continuously.

The root cause of this phenomenon is complex and not yet fully understood, but often caused by software error accumulation and memory leak which are also referred to as software aging problem [9]. Due to software aging problem, from an application's perspective, the system keeps slowing down as it continuously operates and eventually causes execution failures. As an effective countermeasure, software rejuvenation [9], [10] techniques are introduced to recover the system performance. However, almost all types of software rejuvenation approaches have a common side-effect, i.e., when systems perform software rejuvenation, they are not available to execute applications. The trade-off between rejuvenation overhead and system performance gain has drawn large efforts from the community to optimize different system objectives, such as availability [11], [12], reliability [13], [14] and capacity [15]. At the same time, studies on different rejuvenation methods, from the simplest reboot (i.e., cold rejuvenation) [10], [16] to more complex multilevel rejuvenation [17], [18], [19], and different strategies of rejuvenation, such as the mixed or fine-grained rejuvenations [17], [20], have also been introduced in the literature to reduce the overhead and maximize the recovery of system performance.

Through rejuvenation, systems are able to operate at a desired performance level for long period of time. However, from an application's perspective, resources with repeated rejuvenations are only periodically available to them. Moreover, due to performance degradation, even within the time interval when the resource is available, its performance keeps decreasing. These features are not considered by traditional resource models, such as the model used by Liu [1]. To capture these new properties, we introduced a resource model, the P^2-resource, to feature resources with performance degradation and periodic rejuvenation in our earlier work [21]. We also studied the schedulability issues on a P^2-resource with only one level (cold) of rejuvenation.

In this paper, we extend the study to a P^2-resource model with dual-level (both cold and warm) rejuvenations, i.e., the P^2D-resource model. Based on the new resource model, we then study the resource's worst case supply and provide both the resource supply bound and the linear supply bound. With the supply bound analyses, we further derive the task set utilization bounds to determine the schedulability of a given real-time task set under EDF or RM scheduling policies. As the new resource model was emerged from observations of real engineering systems, we believe the research results presented in this paper can benefit many long-standing hard real-time systems in practice. One example is the deep-space exploration system [4], [5] which consists of a master resource and a back-up resource. The back-up resource remains idle until the master resource is under rejuvenation. With the schedulability analysis for a P^2D-resource in this work, we can improve the system's utilization by scheduling more real-time tasks on both resources without causing deadline miss while using rejuvenations to combat performance degradation, or aging, problem.

The rest of the paper is organized as follows: we discuss related work in Section 2. The P^2D-resource model is formally defined in Section 3. The resource supply bound and linear supply bound of a P^2D-resource are studied in Section 4. The utilization bounds for a task set under the EDF and RM scheduling policies on a P^2D-resource are presented in Section 5. We conclude our work in Section 6.

2 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 policies [1]. Since then, the real-time scheduling problem has been extensively studied. Significant amount of work has been done in developing new scheduling algorithms under different assumptions [22], [23] and improving utilization bounds for both EDF and RM policies on single processor [24], multiple processors [25] and distributed system [26], [27] under different constraints (preemptive [28] versus non-preemptive [29]) with various task models (harmonic task set [22], mixed-criticality task set [30], to name a few). However, most of the existing works are based on the assumption that computing resource is always available to applications and its performance does not change (the *regular resource model*), as illustrated in Fig. 2a.

• *The authors are with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616.*
E-mail: {xhua, cguo13, hwu28, dlautner}@hawk.iit.edu, ren@iit.edu.

Manuscript received 4 Dec. 2015; revised 14 Aug. 2016; accepted 16 Aug. 2016. Date of publication 24 Aug. 2016; date of current version 16 Feb. 2017.
Recommended for acceptance by J.D. Bruguera.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2016.2602833

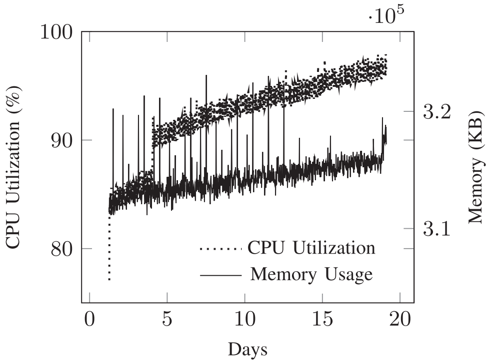


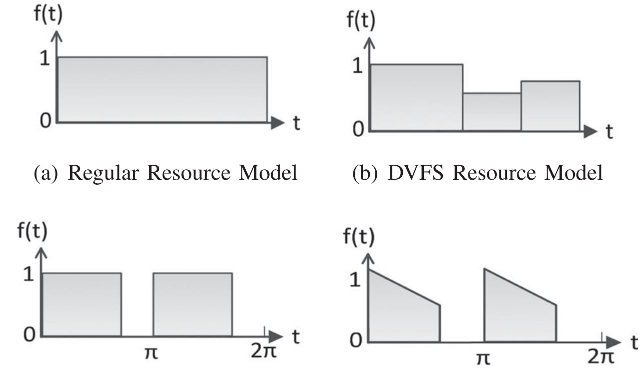
Fig. 1. Aging effect on Fermi monitoring system.

As technology advances, Dynamic Voltage and Frequency Scaling (DVFS) becomes available. With this ability, computing resource can on purposely lower its frequency [31], [32] to reduce power consumption. Therefore, for a DVFS-enabled system, its resource model is no longer the regular model but becomes a continuous model with performance variations, as illustrated in Fig. 2b. The schedulability analysis based on the DVFS resource model is studied intensively by the research community [31], [32], [33]. 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 stabilized until the next switch takes place [32] though the time between two consecutive switches can be infinitely small, i.e., continuous frequency, and (2) the performance change via DVFS is controllable and voluntary [32], [33].

We can use DVFS to simulate the behavior of aging-caused performance degradation, but the DVFS model itself which only specifies either available discrete frequencies, or continuous frequencies, is not sufficient to capture the properties of resource with performance degradation and periodic rejuvenation.

If we only consider the rejuvenation overhead, then a P^2D-resource becomes a periodic resource. The concept of the periodic resource was first introduced by Shigero et al. in 1999 [34]. Mok et al. [35] and Feng et al. [36] extended Shigero's original periodic resource model to the fixed-pattern periodic resource model as illustrated in Fig. 2c. They further provided theoretical analysis on the schedulability of a 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 [37], [38]. However, the work on periodic resources are based on one general assumption, i.e., when a periodic resource is available to applications, its computational capacity does not change. The periodic resource model does not capture the property of performance degradation. The P^2-resource explicitly models both performance degradation and rejuvenation as illustrated in Fig. 2d.

Due to the unavoidable resource performance degradation, applications running on the resource will eventually fail to work if no action is taken to countermeasure the degradation. Software rejuvenation techniques are hence introduced [9], [10]. Various rejuvenation methods and strategies with different optimization purposes have been developed in the literature [11], [12], [13], [14], [15]. Rejuvenation brings back resource performance, but also introduces downtime to applications running on the resource. To balance the tradeoffs, mixed rejuvenation techniques are developed [17], [39], [40]. Hong et al. studied two-level closed-loop rejuvenation techniques and proposed an approach to minimize the average rejuvenation cost [39]. Koutras et al. introduced a two-level software rejuvenation model enhanced with the concept of preventive rejuvenations at the robust state and the rejuvenation failure possibility [40]. Candea et al. introduced the *Microreboot* rejuvenation approach [17] which conducts rejuvenations on fine-grained application components and adjusts the granularity dynamically based on the rejuvenation performance. Researchers also modeled the two-level rejuvenation with Semi-Markov processes and analyzed the optimal rejuvenation policy to maximize the system availability [41], [42].



(a) Regular Resource Model (b) DVFS Resource Model

(c) Periodic Resource Model (d) P^2-resource Model

Fig. 2. Resource models.

In this paper, we focus on the supply bound analysis of a P^2-resource with dual rejuvenation levels, i.e., the P^2D-resource and the schedulability analysis of a real-time task set running on it. We believe that the P^2D-resource model is a more generalized resource model that can be easily transformed to either the regular resource model [1] or the periodic model [38], [43]. In the next section, we formally define the P^2D-resource model and formulate the problems addressed by this paper.

3 Models and Problem Formulation

3.1 Resource Model and Assumptions

As illustrated in Fig. 2d, we model the resource by considering both performance degradation and rejuvenation time cost. Note that by resource performance, we mean the computation cycles provided by the resource to applications in a unit time.

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 $f(0) = 1$.

To simplify mathematical transformations and deviations, and focus more on analysis strategies, we assume in this paper 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 $0 \leq a < 1$.

Resource Rejuvenation. We consider a rejuvenation that can fully recover the system performance as a *cold rejuvenation*, it often refers to system reboot [10], [16]. Likewise, we call a rejuvenation which can only partially recover the system performance, such like application refresh or system component restart [17], [44], as a *warm rejuvenation*. For warm rejuvenations, we assume they recover p percent of the performance from their previous rejuvenation. Both types of rejuvenations are considered as atomic procedures, i.e., they cannot be interrupted.

The P^2D-Resource Model. We model P^2D-resource R with a sextuple $R(f(t), \pi, \phi_c, \phi_w, p, m)$. $f(t)$ is the *performance degradation function*. π is the time interval between the starting times of two consecutive rejuvenations. ϕ_c and ϕ_w are the cold and warm rejuvenation time costs, respectively, and all the overheads of rejuvenations, such as the status change between rejuvenation and execution, are included in the time costs. p is the percentage of performance recovered by a warm rejuvenation. m is the number of warm rejuvenations between two consecutive cold rejuvenations. The pattern of resource R repeats every $(m+1) \cdot \pi$ time units. We

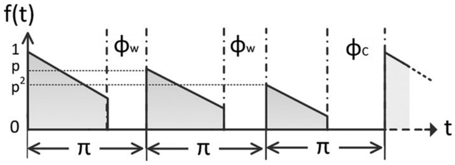


Fig. 3. P^2D-resource model.

further assume that the resource performance never decreases to zero, i.e., $f^{-1}(p^m) > \pi - \phi_c$.

Fig. 3 shows an example of P^2D-resource with $m = 2$.

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 two-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^2D-resource. All the running overheads of τ_i, such as the task lunching or finishing overheads, are included in e_i. 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 and $P_{\min}$ to denote the minimum task period of the task set Γ, i.e., $P_{\min} = \min\{P_i | \forall \tau_i \in \Gamma\}$. If $P_{\min} \leq \phi$, a task set is not schedulable in the worst case. Hence, we assume $P_{\min} > \phi$.

3.2 Problem Formulation

To analyze the schedulability of a task set on a P^2D-resource, we take two steps. First, we study the minimal resource supply of a P^2D-resource in a time interval with a given length. Then, we present the sufficient utilization bounds under both EDF and RM scheduling policies for a task set on a P^2D-resource. These two problems are formalized as follow.

Problem 1. *Given a P^2D-resource R, determine its supply bound function (SBF) and linear supply bound function (LSBF).*

Problem 2. *Given a P^2D-resource R and a task set Γ, determine the utilization bounds of task set Γ on R under EDF and RM scheduling policies, respectively.*

To solve the first problem, we must to prove that the resource amount given by both bound functions are no more than the resource amount provided by the given P^2D-resource in any time interval with given length. For the second problem, we must prove that any arbitrary task set is guaranteed to be schedulable if its utilization rate is no larger than the bound.

4 Supply Bound Analysis for P^2D-resource

To analyze task schedulability on P^2D-resources, we need to determine the resource's supply bound which describes the least amount of resource that can be provided in a time interval with given length.

To do so, we first consider a simple case where each cold rejuvenation is followed by a warm rejuvenation and vice versa, i.e., $m = 1$. We call it a P^2D-resource with *interleaving* rejuvenations, or *interleaving P^2D-resource*. We analyze its worst case resource supply and provide its supply bound functions. We then extend the results in two directions: 1) determine the sufficient condition for interleaving rejuvenation to be superior than cold-only rejuvenation from the schedulability perspective; and 2) generalize the supply analysis from a interleaving P^2D-resource to a regular P^2D-resource.

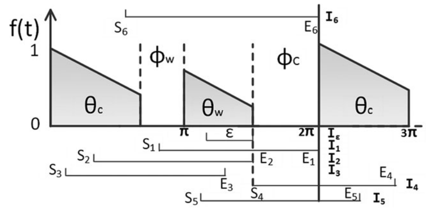


Fig. 4. Resource supply in time interval I.

4.1 Resource Supply Bounds for P^2D-Resource with Interleaving Rejuvenations

We denote the P^2D-resource with interleaving rejuvenation as $R_I(f(t), \pi, \phi_c, \phi_w, p, 1)$. To simplify the expressions, we introduce the following notations:

1) Define $\theta_c = \int_0^{\pi-\phi_w} f(t)dt$ as the amount of resource provided from the completion of a cold rejuvenation to the start of the next warm rejuvenation.
2) Define $\theta_w = \int_0^{\pi-\phi_c} [f(t) - (1-p)]dt$ as the amount of resource provided from the completion of a warm rejuvenation to the start of the next cold rejuvenation.
3) Define $\theta_I = \theta_w + \theta_c$ as the amount of resource provided in a resource period.

Fig. 4 shows an example of R_I.

Similar to the analysis of P^2-resource with only cold rejuvenation in [21], in Lemma 1 we first determine which time interval obtains the least amount of resource for a given time length. Then, we provide the minimal supply function (MSF) of R_I in Lemma 2 and the supply bound function in Lemma 3. With MSF and SBF, we derive the linear supply bound function for R_i in Theorem 1.

Lemma 1. *For a P^2D-resource with interleaving rejuvenation $R_I(f(t), \pi, \phi_c, \phi_w, p, 1)$, a time interval of length $t (t < 2\pi)$ obtains the minimal amount of resource if the interval ends at the completion point of a cold rejuvenation.*

Proof. Denote the time interval that ends at the completion time point of a cold rejuvenation by I. Denote I's start and end times as S_i and E_i, respectively. For I's length t, we consider the following four different cases separately:

1) $0 \leq t < \phi_c$
2) $\phi_c \leq t < \pi$
3) $\pi \leq t < \pi + \phi_w$
4) $\pi + \phi_w \leq t < 2\pi$

Case 1. $0 \leq t < \phi_c$. In this case, no resource is provided in I since R_I is under rejuvenation.

Case 2. $\phi_c \leq t < \pi$. In this case, R provides resource in time interval I. Let $t = \phi_c + \epsilon$ where $0 \leq \epsilon < \pi - \phi_c$. It is clear that for any time interval of length t in a resource period, the longest time without resource supply is ϕ_c and the shortest time that always has resource supply is ϵ. Moreover, during the entire resource period, the least and consistent resource supply provided by R in intervals with length ϵ is the interval that ends at the beginning of a cold rejuvenation. This scenario is illustrated as I_ϵ in Fig. 4. Since I_ϵ is also adjacent to ϕ_c, therefore the time interval with length t which ends at the completion time point of a cold rejuvenation obtains the minimal resource supply from R.

Case 3. $\pi \leq t < \pi + \phi_w$. We prove that time interval I_1 shown in Fig. 4 has the minimal resource supply of θ_w among all intervals with length t.

We first consider the scenario when a time interval I_i ends no later than the completion time point of a cold rejuvenation, i.e., $E_i \leq 2\pi$. If a time interval I_i with length t starts during a warm rejuvenation, i.e., $\pi - \phi_w \leq S_i \leq \pi$, and ends within a cold rejuvenation, i.e., $2\pi - \phi_c \leq E_i < 2\pi$, then this interval obtain θ_w resource, which is equal to the resource obtained by I_1. As $\phi_c \geq \phi_w$, it is possible for an interval, such as I_2 in Fig. 4, to start before the beginning of a warm rejuvenation and end during a cold rejuvenation, i.e., $S_i < \pi - \phi_w$ and $E_i \geq 2\pi - \phi_c$. In this case, the interval obtains resource more than θ_w. If an interval, such as I_3 illustrated in Fig. 4, starts in $[0, \pi - \phi_w)$ and end in $[\pi, 2\pi - \phi_c)$, it obtains resource from both θ_c and θ_w. Comparing interval I_3 and I_2 where $E_2 = 2\pi - \phi_c$, I_3 obtains $\int_{S_3}^{S_2} f(t)dt - \int_{E_3}^{E_2}[f(t) - (1-p)]dt$ more resource, which is always larger than zero, than I_2 does, hence I_1 still obtains the minimal resource supply.

Next, we consider the scenario that for a time interval I_i, $E_i \geq 2\pi$. It is easy to see that if $E_i \geq 4\pi$, I_i repeats the case of $E_i \leq 2\pi$ in another resource period. For the case that $4\pi > E_i \geq 3\pi$, the time interval obtains at lest θ_c resource which is more than θ_w. For the case that the $3\pi \geq E_i \geq 2\pi$, if $2\pi \geq S_i \geq 2\pi - \phi_c$, then it is obvious that resource provided in I_i is no less than I_4 where $S_4 = 2\pi - \phi_c$. If $2\pi - \phi_c \geq S_i \geq 2\pi$, like I_5 in Fig. 4, the more resource provided in I_5 than in I_4 can be calculated as $\int_{S_4}^{S_5} f(t)dt - \int_{E_4}^{E_5}[f(t) - (1-p)]dt > 0$, which indicates I_5 gains more resource than I_4. Similar to the comparison between I_3 and I_2, we have I_4 obtains more resource than I_1.

In either scenario, I obtains the least resource.

Case 4. $\pi + \phi_w \leq t < 2\pi$. With the similar approach used in Case 3, we can prove that time interval I_6 in Fig. 4 obtains the least resource in this case.

We have proved that in all four cases, if the time interval ends up with the beginning of a cold rejuvenation, then the resource provided in this time interval is minimal compared to all intervals with the same length. □

Lemma 2. *For a P^2D-resource with interleaving rejuvenation $R_I(f(t), \pi, \phi_c, \phi_w, p, 1)$, its minimal supply function $msf_I(t)$ $(t < 2\pi)$, i.e., the minimal amount resource that can be provided in a time interval of length t, is*

$$msf_I(t) = \begin{cases} 0, & \text{if } 0 \leq t < \phi_c \\ \int_{\pi-t}^{\pi-\phi_c} pf(x)dx, & \text{if } \phi_c \leq t < \pi \\ \theta_w, & \text{if } \pi \leq t < \pi + \phi_w \\ \int_{2\pi-t}^{\pi-\phi_w} f(x)dx + \theta_w, & \text{if } \pi + \phi_w \leq t < 2\pi. \end{cases} \tag{2}$$

Proof. As Eq. (2) is a step function, we analyze its four cases separately. Based on Lemma 1, we only consider the time interval that ends at the completion time point of a cold rejuvenation, which is denoted as I. To simplify the presentation, we denote the right end of I, i.e., the completion of a cold rejuvenation, as time 0, and the left end as time t.

Case 1. $\phi_c > t \geq 0$. I gains no resource since R_I is under rejuvenation.

Case 2. $\pi > t \geq \phi_c$. We split I into two time intervals $I_1 = [0, \phi_c)$, and $I_2 = [\phi_c, t]$. As discusses in Case 1, I_1 gains no resources. For I_2, like I_ϵ in Fig. 4 where $\epsilon = t - \phi_c$, the amount of resource it obtains is $\int_{\pi-t}^{\pi-\phi_c}[f(x) - 1 + p]dx$.

Case 3. $\pi - \phi_w > t \geq \pi$. We split I into $I_1 = [0, \pi)$ and $I_2 = [\pi, t]$. For I_1, it obtains θ_w resource; for I_2, it obtains no resource.

Case 4. $2\pi > t \geq \pi - \phi_w$. We split I into $I_1 = [0, \pi - \phi_w)$ and $I_2 = [\pi - \phi_w, w\pi]$. For I_1, it obtains θ_w resource; for I_2, it obtains $\int_{2\pi-t}^{\pi-\phi_w} f(x)dx$ resource. □

Based on the minimal supply function, we derive the supply bound function of a P^2D-resource in the following lemma.

Lemma 3. *For a P^2D-resource $R(f(t), \pi, \phi_c, \phi_w, p, 1)$ with interleaving rejuvenation, its supply bound function is calculated as below:*

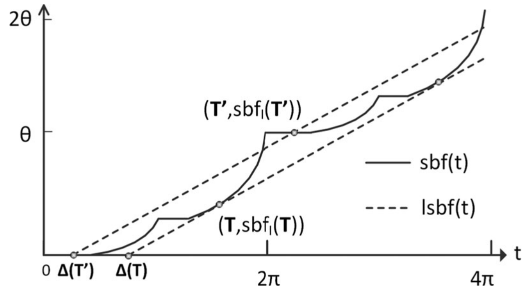


Fig. 5. Intuition of $\Delta(t)$.

$$sbf_I(t) = \left\lfloor \frac{t}{2\pi} \right\rfloor \theta_I + msf_I(t \bmod 2\pi). \tag{3}$$

Proof. Denote a time interval as I, and divide I into two parts: the first part (if $t \geq 2\pi$) that contains several integral resource period and the second part is the remaining time interval that is shorter than 2π. For the first part, it obtains $\lfloor \frac{t}{2\pi} \rfloor \theta_I$ amount of resource. For the second part, we only need to consider the worst case, i.e., the time interval that ends with a cold rejuvenation according to Lemma 1. In this part, the time interval obtains at least $msf_I(t \bmod 2\pi)$ amount of resource. □

To analyze the schedulability of a task set on R_I, we further calculate its linear supply bound function $lsbf_I(t)$ which is the lower tangent line of the supply bound function. Obviously, the slope of $lsbf_I(t)$ is $\frac{\theta_I}{2\pi}$ and the challenge is to calculate the tangent point. To simplify the expressions, we define the tangent point discriminant $\Delta(t)$ as follow:

$$\Delta(t) = t - \frac{2\pi \cdot sbf_I(t)}{\theta}. \tag{4}$$

Geometrically, for a linear function $L(t)$ which has slope $\frac{\theta_I}{2\pi}$ and passes through point (T, $sbf_I(T)$), $\Delta(T)$ is the value of $L^{-1}(0)$. Obviously, if $(T, sbf_I(T))$ is the tangent point of $lsbf_I(t)$ and $sbf_I(t)$, then $\forall T' \in \mathbb{R},\ \Delta(T) \geq \Delta(T')$, which is demonstrated in Fig. 5.

Theorem 1. *For a P^2D-resource with interleaving rejuvenation $R(f(t), \pi, \phi_c, \phi_w, p, 1)$, its linear supply bound function $lsbf_I(t)$ is calculated as below:*

$$lsbf_I(t) = \frac{\theta_I}{2\pi}(t - Tp) + msf_I(Tp), \tag{5}$$

where

$$Tp = \begin{cases} Tp_1, & \text{if } \Delta(Tp_1) \geq \Delta(Tp_2) \\ Tp_2, & \text{otherwise,} \end{cases} \tag{6}$$

and $Tp_1 = \max(\pi + \phi_w, 2\pi - \frac{2\pi - \theta_I}{2\pi a}), Tp_2 = \max(\phi_c, \pi - \frac{2\pi p - a\theta_I}{2\pi a})$.

Proof. To prove $lsbf_I(t)$ is the lower tangent line of $sbf_I(t)$, we need to prove that the following two conditions always hold:

$$\begin{cases} \forall t : lsbf(t) \leq sbf(t) \\ \exists t : lsbf(t) = sbf(t), \end{cases} \tag{7}$$

In the first step, we consider the scenario that $\mathrm{Tp} = \mathrm{Tp}_1$, which indicates $\Delta(\mathrm{Tp}_1) \geq \Delta(\mathrm{Tp}_2)$. As $sbf_I(t)$ is a step function (due to $msf_I(t)$ is a step function) which contains four cases, we consider the following four cases separately:

I) $2n\pi \leq t < 2n\pi + \phi_c$
II) $2n\pi + \phi_c \leq t < (2n+1)\pi$
III) $(2n+1)\pi \leq t < (2n+1)\pi + \phi_w$
IV) $(2n+1)\pi + \phi_w \leq t < (2n+2)\pi$

for all $n \in \mathbb{N}$.

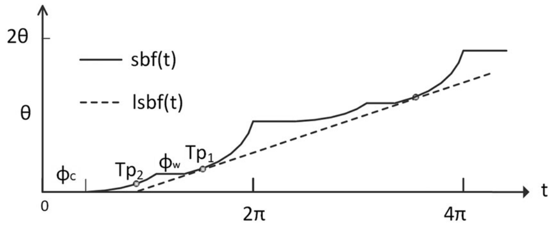


Fig. 6. SBF and LSBF.

Case 1. when $2n\pi \le t < 2n\pi + \phi_c$. In this case $sbf(t) = n\theta_I$ by definition. On the other hand, we also know that $lsbf(\Delta(\mathrm{Tp}_1)) = 0$ and $\Delta(\mathrm{Tp}_1) \ge \phi_c$ according to Eq. (4). As $lsbf_I(t)$ increases monotonically with t, we further have $lsbf_I(\phi_c + 2n\pi) \le n\theta_I \Rightarrow lsbf_I(t) \le sbf(t)$.

Case 2. when $2n\pi + \phi_c \le t < (2n+1)\pi$, we have $n = \lfloor \frac{t}{2\pi} \rfloor$. Let $t' = t \bmod 2\pi$ then

$$\begin{aligned} sbf_I(t) - lsbf_I(t) = {} & n\theta_I + msf_I(t') \\ & - \frac{\theta_I}{2\pi}(2n\pi + t' - \mathrm{Tp}_1) - msf_I(\mathrm{Tp}_1), \end{aligned} \tag{8}$$

Define function $S(x)$ as $S(x) = msf_I(x) - \frac{\theta_I}{2\pi}x$, then Eq. (8) is transformed to $sbf_I(t) - lsbf_I(t) = S(t') - S(\mathrm{Tp}_1)$. As $S'(\mathrm{Tp}_2) = 0$ and $S''(\mathrm{Tp}_2) > 0$ which indicates $S(x)$ has minimal value when $x = \mathrm{Tp}_2$. Also, since $\Delta(\mathrm{Tp}_1) \ge \Delta(\mathrm{Tp}_2)$ and $sbf_I(\mathrm{Tp}_1) = lsbf_I(\mathrm{Tp}_1)$, we have $S(t') - S(\mathrm{Tp}_2) > 0$, which further indicates $sbf_I(t) - lsbf_I(t) > 0$ always holds.

Case 3. when $(2n+1)\pi \le t < (2n+1)\pi + \phi_w$. It is obvious that $sbf_I(t) = (n+1)\pi$. Since $lsbf_I(t)$ is monotonically increase with t and $lsbf_I((2n+1)\pi + \phi_w) \le (n+1)\pi$, we have $sbf_I(t) > lsbf_I(t)$ always holds.

Case 4. when $(2n+1)\pi + \phi_w \le t < (2n+2)\pi$. Following the same prove approach given in Case 2, we have $S'(\mathrm{Tp}_1) = 0$ and $S''(\mathrm{Tp}_1) > 0$. Plus $sbf_I(\mathrm{Tp}_1) = lsbf_I(\mathrm{Tp}_1)$, we now have $sbf_I(t) - lsbf_I(t) > 0$ always holds. In particular, when $t = \mathrm{Tp}_1$, $sbf_I(t) = lsbf_I(t)$.

In the four scenarios, we proved that the first condition in Eq. (7) always holds. Moreover, we further proved that the second condition also holds since $sbf_I(\mathrm{Tp}_1) = lsbf_I(\mathrm{Tp}_1)$.

With the same approach, we can also derive the same result from the second scenario that $\mathrm{Tp} = \mathrm{Tp}_2$. □

An example of SBF and LSBF with $\mathrm{Tp} = \mathrm{Tp}_1$ is shown in Fig. 6.

Based on Theorem 1, we extend this result in two directions in the following two sections: 1) provide the sufficient condition when the interleaving rejuvenation is superior than code rejuvenation, i.e., $m = 0$; and 2) extend the result to generic mixed rejuvenations, i.e., $m \ge 1$.

4.2 Interleaving Rejuvenation versus Cold Rejuvenation

We define the criteria that resource R_2's schedulability is *better* than R_1's when 1) all task sets that are schedulable on R_1 are also schedulable on R_2, and 2) there exists a task set that is schedulable on R_2 but is unschedulable on R_1.

Next, we develop a sufficient condition that a P^2D-resource with interleaving rejuvenation is better than with only cold rejuvenation.

Corollary 1. *Given resources $R_a(f(t), \pi, \phi_c, \phi_w, p, 0)$ with cold rejuvenation and $R_b(f(t), \pi, \phi_c, \phi_w, p, 1)$ with interleaving rejuvenation. Let θ_a and θ_b denotes the amount of resource provided in one resource period of R_a and R_b, respectively, then $\theta_a = \theta_c$ and $\theta_b = \theta_c + \theta_w$. R_b has better schedulability than R_a if the following conditions are satisfied*

$$\begin{cases} \Delta(\mathrm{Tp}) < \max((\Delta(\mathrm{Tp}_1), \Delta(\mathrm{Tp}_2)) \\ \pi \le \frac{\phi_c + \phi_w + 1}{2} - \frac{(1-p)(\pi - \phi_c)}{ap(\phi_c - \phi_w)}, \end{cases} \tag{9}$$

where $\mathrm{Tp} = \max\{\pi - \frac{\pi - \theta_a}{a\pi}, \phi_1\}$, $\mathrm{Tp}_1 = \max(\pi + \phi_w, 2\pi - \frac{2\pi - \theta_b}{2\pi a})$ *and* $\mathrm{Tp}_2 = \max(\phi_c, \pi - \frac{2\pi p - \theta_b}{2\pi a p})$.

Proof. As mentioned in [38], the condition of schedulability guarantee of a task set is that the task set's demand bound (for EDF) or each task's resource demand plus the interferences from higher priority tasks (for RM) is always less than the supply bound in any given time interval. Therefore, to guarantee that R_2 has better schedulability, we consider the two conditions based on the linear supply bound function:

1) the longest time of R_2 that provides no resource, i.e., blackout time, is shorter than R_1's longest blackout time.
2) the average resource providing rate of R_2 is higher than the rate of R_1.

Based on these two conditions, we derive the condition for R_2 to be better than R_1.

Based on Eq. (5) in Theorem 2 and the definition of $\Delta(t)$ in Eq. (4), we derive the blackout time of R_1 and R_2 as $\Delta(\mathrm{Tp})$ and $\max(\Delta(\mathrm{Tp}_1), \Delta(\mathrm{Tp}_2))$, respectively. Therefore, condition (1) is satisfied when $\Delta(\mathrm{Tp}) < \max((\Delta(\mathrm{Tp}_1), \Delta(\mathrm{Tp}_2))$ holds.

For the second condition, we have $\frac{\theta_a}{\pi} \ge \frac{\theta_b}{2\pi}$ which can be further transformed to the second condition in Eq. (9). □

4.3 Linear Supply Bound Function for P^2D-Resource

To find the linear supply bound function $lsbf_M(t)$ of a P^2D-resource with cold and warm rejuvenations, we take the similar approach adopted for interleaving P^2D-resource. First, we provide the supply bound function $sbf_I(t)$ in Lemma 4, then we provide $lsbf_M(t)$ in Theorem 2. To simplify the expression, we introduce the following notations for $R(f(t), \pi, \phi_c, \phi_w, p, m)$.

1) Define $\Pi = (m+1)\pi$ as the resource period.
2) Define $\phi_0 = \phi_c$ and $\forall i \in [1, m] : \phi_i = \phi_w$.
3) Define $\theta = \int_0^{\pi - \phi_c} f(t)dt + \sum_{j=1}^{m} p^j \int_0^{\pi - \phi_w} f(t)dt$ as the amount of resource provided in one resource period Π.
4) Define $\theta_i = p^i \int_0^{\pi - \phi_i} f(t)dt$ as the resource provided in time interval $[(i-1)\pi, i\pi]$ where $i > 0$ and $\theta_0 = 0$.
5) Define

$$\Delta_M(t) = t - \frac{\Pi \cdot sbf_M(t)}{\theta}, \tag{10}$$

as the tangent point discriminant where $sbf_M(t)$ is defined in Lemma 4.

Lemma 4. *For a P^2D-resource $R(f(t), \pi, \phi_c, \phi_w, p, m)$, its supply bound function $sbf_M(t)$ is calculated as below:*

$$sbf_M(t) = \left\lfloor \frac{t}{\Pi} \right\rfloor \theta + \sum_{j=0}^{i-1} \theta_j + msf_M(T), \tag{11}$$

where $i = \lfloor \frac{t \bmod \Pi}{\pi} \rfloor$, $T = t \bmod \pi$ *and*

$$msf_M(t) = \begin{cases} p^i \int_{\pi - T}^{\pi - \phi_i} f(t)dt, & \text{if } \pi - T > \phi_i \\ 0, & \text{otherwise}. \end{cases} \tag{12}$$

Proof. Similar to the proof of Lemma 2, we focus on the time interval, denoted by I, that ends at the completion time point of a cold rejuvenation. We then divide the time interval into three segments and calculate the resource separately: segment I_1 that contains integral resource periods $(n+1)\pi$; segment I_2 that contains integral rejuvenation periods π; and the remaining segment, denoted as I_3, which is shorter than π.

For I_1, it has $\lfloor \frac{t}{\Pi} \rfloor$ integral resource periods, therefore it obtains $\lfloor \frac{t}{\Pi} \rfloor \cdot \theta$ amount of resource.

For I_2, it has i integral rejuvenation periods π, hence it contains all the available resource provided from the end of the ith warm rejuvenation to the next cold rejuvenation. On the other hand, for the time interval between the end of jth rejuvenation and the start of the $(j+1)$th rejuvenation, the resource amount is calculated as $p^j \int_0^{\pi-\phi_w} f(t)dt$. Therefore, I_2's total resource amount is calculated as $\sum_{j=0}^{i-1} \theta_j$.

For I_3, we take the similar method of calculating of $msf_I(t)$ in Lemma 2 and calculate the amount of resource provided in this time interval, i.e., $msf_M(t)$, in Eq. (12).

Combine these three part together, we have the $sbf_M(t)$ calculated in Eq. (13). $\square$

Theorem 2. *For a P^2D-resource $R(f(t), \pi, \phi_c, \phi_w, p, m)$, let $i = \lfloor \frac{t \bmod \Pi}{\pi} \rfloor$ and $\mathrm{Tp}_i = \max(i\pi + \phi_i, (i+1)\pi - \frac{\Pi p^i - \theta}{a \Pi p^i})$. R's linear supply bound function $lsbf_M(t)$ is*

$$lsbf_M(t) = \frac{\theta}{\Pi}(t - Tp) + \sum_{j=0}^{i-1} \theta_j + msf_M(Tp), \tag{13}$$

where $Tp \in \{Tp_1, \ldots, Tp_{m+1}\}$ and $\forall Tp_x \in \{Tp_1, \ldots, Tp_{m+1}\}$: $\Delta_M(Tp) \geq \Delta_M(Tp_x)$.

Proof. With the same approach used to prove Theorem 1, we prove this theorem in technical report [45]. $\square$

5 Utilization Bound of a P^2D-resource under EDF and RM Scheduling Policies

With the supply bound analysis of P^2D-resource given in Section 4, we are now able to derive the sufficient utilization bound under different scheduling policies for a real-time task set. To simplify the expression, we keep using the notations Π, ϕ_i, θ and θ_i defined in Section 4.3.

Theorem 3. *Given a task set Γ and a P^2D-resource $R(f(t), \pi, \phi_c, \phi_w, p, m)$, let $i = \left\lfloor \frac{t \bmod \Pi}{\pi} \right\rfloor$ and let Tp The sufficient utilization bound under EDF scheduling policy is*

$$UB_{\mathrm{EDF}}(\Gamma, R) = \frac{\theta}{\Pi} - \frac{\frac{\theta}{\Pi} \cdot Tp - \sum_{j=0}^{i-1} \theta_j - msf_M(Tp)}{P_{\min}}, \tag{14}$$

where $Tp \in \{Tp_1, \ldots, Tp_{m+1}\}$ satisfies $\forall Tp_x \in \{Tp_1, \ldots, Tp_{m+1}\}$: $\Delta_M(Tp) \geq \Delta_M(Tp_x)$ and $Tp_x = \max(x\pi + \phi_x, (x+1)\pi - \frac{\Pi p^x - \theta}{a \Pi p^x})$.

Proof. Detailed proof is given in technical report [45]. $\square$

It is worth noting that the presented P^2D-resource model is more general. For instance, we can de-generalized the P^2D-resource model to a continuous and constant resource by setting $f(t) = 1$ and $\phi = 0$. In this case, the utilization bound of a P^2D-resource under EDF policy $UB_{\mathrm{EDF}}(\Gamma, R)$ becomes the utilization bound given by Liu and Layland [1], i.e., $UB_{\mathrm{EDF}}(\Gamma, R) = 1$.

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

Proof. For the given resource R, since $f(t) = 1$ and $\phi_w = \phi_c = 0$, we have $\theta = \Pi$ and $msf_M(\mathrm{Tp}) = \mathrm{Tp}$. Therefore,

$$UB_{\mathrm{EDF}}(\Gamma, R) = 1 - \frac{\mathrm{T_p} - \mathrm{T_p}}{P_{\min}} = 1.$$

$\square$

In the next step, we introduce the utilization bound for RM scheduling policy.

Theorem 4. *Given a P^2D-resource $R(f(t), \pi, \phi_c, \phi_w, p, m)$ 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*

$$UB_{\mathrm{RM}}(\Gamma, R) = \frac{n\theta}{\Pi}\left[\left(1 + \frac{w\Pi + \frac{\Pi}{\theta}(\sum_{j=0}^{i-1} \theta_j + msf_M(\phi_i + \lambda))}{k\pi + \phi_i + \delta}\right)^{\frac{1}{n}} - 1\right],$$

where a is the slope of $f(t)$, $k = \left\lfloor \frac{P_{\min}}{\pi} \right\rfloor$, $w = \lfloor \frac{P_{\min}}{\Pi} \rfloor$, $i = \left\lfloor \frac{P_{\min} \bmod \Pi}{\pi} \right\rfloor$, $\delta = \max\{\min\{\lambda, \pi - \phi_i\}, 0\}$ and

$$\lambda = -a(\phi_i + k\pi) + ((a\phi_i + ak\pi)^2 - \min\{2a((1 + a\phi_i - a\pi) \cdot (\phi_i + k\pi) - (w\theta + \sum_{j=1}^{i-1} \theta_j)/p^i), (a\phi_i + ak\pi)^2\})^{\frac{1}{2}}.$$

Proof. Detailed proof is provided in technical report [45]. $\square$

The $\mathrm{UB_{RM}}$ comes the RM utilization bound given in [1] for traditional continuous and constant resources.

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

$$UB_{\mathrm{RM}}(\Gamma, R) = n(2^{1/n} - 1).$$

Proof. Since $f(t) = 1$ and $\phi_c = \phi_w = 0$, we have $\frac{\theta}{\Pi} = 1$, $w\Pi + \sum_{j=0}^{i-1} \theta_j = k\pi$, $\delta = \lambda$ and $msf_M(\phi_i + \lambda) = \lambda$. Therefore,

$$\frac{w\Pi + \sum_{j=0}^{i-1} \theta_j + \frac{\Pi}{\theta} msf_M(\phi_i + \lambda)}{k\pi + \phi_i + \delta} = \frac{k\pi + \lambda}{k\pi + \lambda} = 1,$$

and hence $UB_{\mathrm{RM}}(\Gamma, R) = n(2^{1/n} - 1)$. $\square$

6 Conclusion and Future Work

In this paper, we have formally defined resources with performance degradation and dual periodic rejuvenations, provided supply bound analysis of a P^2D-resource, and derived the closed forms of utilization bounds for a task set on a P^2D-resource with both EDF and RM scheduling policies, respectively. With these two bounds, the schedulability of a given task set can be easily determined. Comparing to the existing works on real-time scheduling where the resource is assumed to be continuous and with constant performance, we captured the system aging effect and rejuvenation cost in our resource model, therefore we believe our work is more generic and closer to the systems in reality, especially for long-running real-time embedded applications.

However, to simplify the study, we have assumed the performance degradation function of a P^2D-resource is linear, which may not always hold. Our next step is to remove this assumption and study the P^2D-resources with non-linear performance degradation functions and also consider to extend from dual-level rejuvenation to multiple level rejuvenation. More practical factors, such as system running overheads, and more research methods, such as empirical study, will also be taken into consideration. Meanwhile, as multi-core, multi-processor and multi-host structure are becoming the mainstream, we will also study the schedulability of a task set on multiple P^2D-resources and design task assignment and scheduling algorithms. More complex computing systems, such like HPC-based cluster or Cloud system, and different types of resources, such as storage and network, will also be included in our future research scope.

ACKNOWLEDGMENTS

This work is supported by US National Science Foundation CPS 1545008.

REFERENCES

[1] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[2] Z. Kunpeng and S. Zhiyi, "The measurement and data acquisition system for the aging test of the composite insulators," in *Proc. IEEE Int. Conf. Power Syst. Technol.*, 2002, vol. 3, pp. 1863–1866.

[3] A. Norman, "The nova data acquisition system," Fermilab, 2012. [Online]. Available: 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, and S. Chau, "On-board preventive maintenance for long-life deep-space missions: A model-based analysis," in *Proc. IEEE Int. Comput. Performance Dependability Symp.*, Sep. 1998, pp. 196–205.

[5] A. Tai and L. Alkalai, "On-board maintenance for long-life systems," in *Proc. IEEE Workshop Appl.-Specific Softw. Eng. Technol.*., Mar. 1998, pp. 69–74.

[6] K. Tomsovic, D. E. Bakken, V. Venkatasubramanian, and A. Bose, "Designing the next generation of real-time control, communication, and computations for large power systems," *Proc. IEEE*, vol. 93, no. 5, pp. 965–979, May 2005.

[7] J. Davidson and F.-C. Bouchart, "Adjusting nodal demands in SCADA constrained real-time water distribution network models," *J. Hydraulic Eng.*, vol. 132, no. 1, pp. 102–110, 2006.

[8] LabVIEW System Design Software. National Instruments. Available: http://www.ni.com/labview/, 2016.

[9] Y. Huang, C. Kintala, N. Kolettis, and N. D. Fulton, "Software rejuvenation: Analysis, module and applications," in *Proc. IEEE 25th Int. Symp. Fault-Tolerant Comput. Digest Papers*, 1995, pp. 381–390.

[10] K. S. Trivedi, K. Vaidyanathan, and K. Goseva-Popstojanova, "Modeling and analysis of software aging and rejuvenation," in *Proc. IEEE 33rd Annu. Simulation Symp.*, 2000, pp. 270–279.

[11] T. Dohi, K. Goševa-Popstojanova, and K. Trivedi, "Estimating software rejuvenation schedules in high-assurance systems," *Comput. J.*, vol. 44, no. 6, pp. 473–485, 2001.

[12] A. Pfening, S. Garg, A. Puliafito, M. Telek, and K. S. Trivedi, "Optimal software rejuvenation for tolerating soft failures," *Performance Evaluation*, vol. 27, pp. 491–506, 1996.

[13] Y. Bao, X. Sun, and K. S. Trivedi, "A workload-based analysis of software aging, and rejuvenation," *IEEE Trans. Rel.*, vol. 54, no. 3, pp. 541–548, Sep. 2005.

[14] Y. Bao, X. Sun, and K. S. Trivedi, "Adaptive software rejuvenation: Degradation model and rejuvenation scheme," in *Proc. Int. Conf. Dependable Syst. Netw.*, 2003, pp. 241–248.

[15] S. Garg, Y. Huang, C. Kintala, and K. S. Trivedi, "Time and load based software rejuvenation: Policy, evaluation and optimality," in *Proc. 1st Fault-Tolerant Symp.*, 1995, pp. 22–25.

[16] M. Grottke and K. S. Trivedi, "Fighting bugs: Remove, retry, replicate, and rejuvenate," *IEEE Comput.*, vol. 40, no. 2, pp. 107–109, Feb. 2007.

[17] G. Candea, S. Kawamoto, Y. Fujiki, G. Friedman, and A. Fox, "Microreboot-a technique for cheap recovery," in *Proc. 6th Conf. Symp. Opearting Syst. Des. Implementation*, 2004, vol. 4, pp. 31–44.

[18] M. Accetta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian, and M. Young, "Mach: A new kernel foundation for UNIX development," in *Proc. USENIX Assoc. Conf.*, 1986, pp. 93–112.

[19] M. M. Swift, B. N. Bershad, and H. M. Levy, "Improving the reliability of commodity operating systems," *ACM SIGOPS Operating Syst. Rev.*, vol. 37, no. 5, pp. 207–222, 2003.

[20] W. Xie, Y. Hong, and K. Trivedi, "Analysis of a two-level software rejuvenation policy," *Rel. Eng. Syst. Safety*, vol. 87, no. 1, pp. 13–22, 2005.

[21] X. Hua, Z. Li, H. Wu, and S. Ren, "Scheduling periodic tasks on multiple periodic resources," in *Proc. 4th Int. Conf. Adv. Commun. Comput.*, 2014, pp. 35–40.

[22] M. Fan and G. Quan, "Harmonic semi-partitioned scheduling for fixed-priority real-time tasks on multi-core platform," in *Proc. Conf. Des. Autom. Test Eur.*, 2012, pp. 503–508.

[23] L. Niu and G. Quan, "A hybrid static/dynamic DVS scheduling for real-time systems with (m, k)-guarantee," in *Proc. 26th IEEE Int. Real-Time Syst. Symp.*, 2005, pp. 356–365.

[24] J. A. Stankovic, *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms*. New York, New York. USA: Springer, 1998.

[25] O. U. P. Zapata and P. M. Alvarez, "EDF and RM multiprocessor scheduling algorithms: Survey and performance evaluation," CINVESTAV-IPN, Departmento de Ingenieria Electrica, Seccion de Computacion, Ciudad de México, D.F., Mexico, Rept. no. CINVESTAV-CS-RTG-02, 2005.

[26] X. Qin and H. Jiang, "A dynamic and reliability-driven scheduling algorithm for parallel real-time jobs executing on heterogeneous clusters," *J. Parallel Distrib. Comput.*, vol. 65, no. 8, pp. 885–900, 2005.

[27] R. Santhosh and T. Ravichandran, "Pre-emptive scheduling of on-line real time services with task migration for cloud computing," in *Proc. Int. Conf. Pattern Recognition Informat. Mobile Eng*, 2013, pp. 271–276.

[28] S. K. Baruah, L. E. Rosier, and R. R. Howell, "Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor," *Real-Time Syst.*, vol. 2, no. 4, pp. 301–324, 1990.

[29] K. Jeffay, D. F. Stanat, and C. U. Martel, "On non-preemptive scheduling of period and sporadic tasks," in *Proc. 12th Real-Time Syst. Symp.*, 1991, pp. 129–139.

[30] J. Y. Leung, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Boca Raton, FL, USA: CRC Press, 2004.

[31] S. Saha and B. Ravindran, "An experimental evaluation of real-time DVFS scheduling algorithms," in *Proc. 5th Annu. Int. Syst. Storage Conf.*, 2012, Art. no. 7.

[32] G. Von Laszewski, L. Wang, A. J. Younge, and X. He, "Power-aware scheduling of virtual machines in DVFS-enabled clusters," in *Proc. IEEE Int. Conf. Cluster Comput. Workshops*, 2009, pp. 1–10.

[33] E. Bini, G. Buttazzo, and G. Lipari, "Minimizing CPU energy in real-time systems with discrete speed management," *ACM Trans. Embedded Comput. Syst.*, vol. 8, no. 4, 2009, Art. no. 31.

[34] S. Shirero, M. Takashi, and H. Kei, "On the schedulability conditions on partial time slots," in *Proc. 6th Int. Conf. Real-Time Comput. Syst. Appl.*, 1999, pp. 166–173.

[35] A. K. Mok, X. Feng, and D. Chen, "Resource partition for real-time systems," in *Proc. 7th IEEE Real-Time Technol. Appl. Symp.*, 2001, pp. 75–84.

[36] X. Feng, "Design of real-time virtual resource architecture for large-scale embedded systems," Ph.D. dissertation, Dept. Comput. Sci., Univ. Texas, Austin, 2004.

[37] A. Easwaran, M. Anand, and I. Lee, "Compositional analysis framework using EDP resource models," in *Proc. IEEE Int. Real-Time Syst. Symp.*, 2007, pp. 129–138.

[38] I. Shin and I. Lee, "Compositional real-time scheduling framework with periodic model," *ACM Trans. Embedded Comput. Syst.*, vol. 7, no. 3, 2008, Art. no. 30.

[39] Y. Hong, D. Chen, L. Li, and K. S. Trivedi, "Closed loop design for software rejuvenation," presented at Workshop on Self-Healing, Adaptive, and Self-Managed Systems, New York, NY, Jun. 2002.

[40] V. Koutras and A. Platis, "Applying partial and full rejuvenation in different degradation levels," in *Proc. IEEE 3rd Int. Workshop Softw. Aging Rejuvenation*, Nov. 2011, pp. 20–25.

[41] V. Koutras and A. Platis, "Semi-Markov availability modeling of a redundant system with partial and full rejuvenation actions," in *Proc. 3rd Int. Conf. Dependability Comput. Syst.*, Jun. 2008, pp. 127–134.

[42] V. Koutras, "Two-level software rejuvenation model with increasing failure rate degradation," in *Dependable Computer Systems*. Berlin Germany: Springer, 2011, vol. 97, pp. 101–115.

[43] I. Shin and I. Lee, "Periodic resource model for compositional real-time guarantees," in *Proc. IEEE 24th Real-Time Syst. Symp.*, 2003, pp. 2–13.

[44] C. Guo, H. Wu, X. Hua, D. Lautner, and S. Ren, "Use two-level rejuvenation to combat software aging and maximize average resource performance," in *Proc. IEEE 17th Int. Conf. High Performance Comput. Commun.*, 2015, pp. 1160–1165.

[45] X. Hua, H. W. Chunhui Guo, and S. R. Douglas Lautner, "Tech report: Schedulability analysis for real-time task set on resource with performance degradation and dual-level periodic rejuvenation," 2016. [Online]. Available: http://gauss.cs.iit.edu/~code/publication/dualp2.pdf, https://drive.google.com/open?id=0B06WDTAMREJOVVJxMVRDWGNETFE