amcs

A NEW SUFFICIENT SCHEDULABILITY ANALYSIS FOR HYBRID SCHEDULING

FENGXIANG ZHANG [a,*], YANFENG ZHAI [a], JIANWEI LIAO [a]

[a] School of Computer and Information Science
Southwest University, Beibei District, Chongqing, China
e-mail: `zhangfx@swu.edu.cn`

Earliest deadline first (EDF) and fixed priority (FP) are the most commonly used and studied scheduling algorithms for real-time systems. This paper focuses on combining the EDF and FP strategies in one system. We provide a new sufficient schedulability analysis for real-time hybrid task systems which are scheduled by EDF and FP. The proposed analysis has a polynomial time complexity and no restrictions on task parameters, where the relative deadline of each task could be less than, equal to, or greater than its period. By extensive experiments, we show that our proposed analysis significantly improves the acceptance ratio compared with the existing results of the sufficient schedulability test for hybrid scheduling systems.

Keywords: scheduling algorithms, real-time systems, schedulability analysis, preemptive scheduling, earliest deadline first, fixed priority.

1. Introduction

Schedulability analysis is significant for a hard real-time system since any task which misses its deadline may result in a catastrophic consequence. Research on schedulability analysis for real-time systems mainly concentrates on two types of scheduling policies: the dynamic priority assignment and the fixed priority (FP) assignment. Earliest deadline first (EDF) (Liu and Layland, 1973) is the most commonly studied dynamic scheduling algorithm, while rate monotonic (RM) (Liu and Layland, 1973) and deadline monotonic (DM) (Leung and Whitehead, 1982) are the most widely used FP scheduling algorithms. It is well known that EDF is optimal among all scheduling algorithms on a uniprocessor (Dertouzos, 1974), RM is optimal for an FP task set when each task's relative deadline is equal to its period, and DM is optimal among the FP policies when each task's relative deadline is less than or equal to its period.

EDF has the advantage of optimality—the tasks scheduled by EDF would miss their deadlines with the minimum possibility, and if any given task set cannot be scheduled by EDF then this task set cannot be scheduled

by any priority assignment policy; FP has the advantage of predictability—one failure task can only affect the schedulability of the tasks with lower priorities, and the priorities of the tasks can reflect the importance of each task. Combining EDF and FP can take advantages of both algorithms to meet the requirements of system design. If high critical tasks run under FP with higher priorities, the majority of tasks in the system will be scheduled by EDF. As a result of this scheme, any overrun of the EDF tasks will not affect the most critical ones scheduled by FP. In industrial design and implementation, efficient schedulability analysis with a polynomial time complexity is needed to verify the schedulability of such systems.

In this paper, we concentrate on sufficient analysis of hybrid scheduling systems. We propose a new efficient analysis for the situation when there is more than one high critical FP task in a system, and there is no restriction on the task parameters. Through extensive experiments, we show that the proposed schedulability analysis significantly improves the acceptance ratio of the existing results.

The rest of this paper is organized as follows. Section 2 describes the system model and notation. Section 3 presents the existing results on sufficient schedulability analysis for real-time systems. In Section 4,

---

*Corresponding author

we propose a new sufficient schedulability analysis for systems scheduled by EDF under the interference of higher priority FP tasks. Experimental results are reported in Section 5. A conclusion is given in Section 6.

2. System model and notation

A hybrid real-time task system or task set is comprised of two task subsets: $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$ and $\Psi = \{\varphi_1, \varphi_2, \cdots, \varphi_m\}$, where $n \geq 1$ and $m \geq 1$. The task subset Γ is scheduled by the EDF algorithm, and the task subset Ψ is scheduled by an FP policy such as RM or DM. In this paper, we assume that all the tasks of Ψ have higher priorities than any task of Γ, i.e., if there is a task φ_i ($1 \leq i \leq m$) of Ψ ready to be executed, all the tasks of Γ have to wait until φ_i finishes its execution; at any time instance, an arrived task of Ψ can preempt the execution of any Γ task. The execution sequences of the Ψ tasks are determined by their priorities. Each task in this hybrid-scheduled hard real-time system consists of a finite or infinite stream of jobs or requests which must be completed before their timing deadlines.

Any task τ_i of Γ is characterized by four non-negative numbers, C_i: the worst case execution, D_i: the relative deadline, T_i: the period, and ϕ_i: the start time of τ_i. When the k-th job of τ_i arrives at time $\phi_i + (k-1) \times T_i$, then this job's absolute deadline is given by $\phi_i + (k-1) \times T_i + D_i$, and this job is schedulable iff the processor can allocate C_i time units to τ_i in the time interval

$$[\phi_i + (k-1) \times T_i, \phi_i + (k-1) \times T_i + D_i].$$

If all jobs of every task in the sub task set Γ are schedulable, then we say Γ is schedulable. We assume that there are no restrictions between D_i and T_i, i.e., $D_i \leq T_i$ or $D_i > T_i$. The subset Γ consists of synchronous periodic tasks or sporadic tasks, i.e., for any $1 \leq i \leq n$, $\phi_i = 0$. The utilization of τ_i is characterized by the equation $U(\tau_i) = C_i/T_i$, and the total utilization of Γ is calculated by $U(\Gamma) = \sum_{i=1,\dots,n} U(\tau_i)$. According to the EDF scheduling policy, an arrived job with an earlier absolute deadline can preempt the execution of a job with a later absolute deadline, and when a job completes its execution, the scheduler chooses the pending job with the earliest absolute deadline to execute.

Any task ϕ_i of Ψ is characterized by a three-tuple $(C_i^{fp}, T_i^{fp}, D_i^{fp})$, where $C_i^{fp} > 0$, $T_i^{fp} > 0$ and $D_i^{fp} > 0$, which are the task's worst-case execution time, period and relative deadline, respectively. We also assume that there are no restrictions between D_i^{fp} and T_i^{fp}. We denote $U(\varphi_i) = C_i^{fp}/T_i^{fp}$ to be the utilization of task φ_i, and the total utilization of task set Ψ to be $U(\Psi) = \sum_{i=1,\dots,m} U(\varphi_i)$. At any time, an arrived job with a higher priority can preempt the execution of a lower

priority job, and when a job completes its execution, the system chooses the pending job with the highest priority to execute.

In summary, the following notation is used through the paper:

C_i: the worst case execution time of τ_i,

D_i: the relative deadline of τ_i,

T_i: the period of τ_i,

n: the number of tasks in the task subset Γ,

C_i^{fp}: the worst-case execution time of φ_i,

T_i^{fp}: the period of φ_i,

D_i^{fp}: the relative deadline of φ_i,

m: the number of tasks in the task subset Ψ.

3. Related work

This section reviews some existing results on sufficient schedulability analysis for hard real-time systems. In 1973, Liu and Layland proved that a set of periodic tasks scheduled by EDF are schedulable if and only if $U \leq 1$ under the assumption that each task's relative deadline is equal to its period. Liu (2000) reported that the density of a task set given by $\Delta = \sum_{i=1}^{n} C_i / \min\{D_i, T_i\} \leq 1$ is a sufficient schedulability condition for general EDF systems. With the assumption that the tasks are sorted by non-decreasing relative deadlines, Devi (2003) presented a sufficient schedulability analysis for preemptable, asynchronous, and periodic EDF systems with arbitrary relative deadlines. The author also proved that this test is more accurate than the commonly used density schedulability condition of Liu (2000).

Theorem 1. (Devi, 2003) *A system which consists of preemptable, asynchronous, periodic tasks, with arbitrary relative deadlines, arranged in the order of non-decreasing relative deadlines, is schedulable using an optimal scheduling algorithm if the following condition is satisfied:*

$$\sum_{i=1}^{k} \frac{C_i}{T_i} + \frac{1}{D_k} \sum_{i=1}^{k} \left(\frac{T_i - \min(T_i, D_i)}{T_i} \right) C_i \leq 1,$$

where

$$\forall k: \ 1 \leq k \leq n.$$

There are also a number of papers (e.g., Albers and Slomka, 2004; Chakraborty *et al.*, 2002) that provide an approximation schedulability analysis for EDF scheduling which allows a small error in the output results.

For the tasks scheduled by an FP algorithm, Liu and Layland (1973) presented a sufficient schedulability condition for RM under the assumption that all tasks' relative deadlines are equal to their periods; they showed that, for a set of n periodic tasks with a fixed priority order, the task set is schedulable if

$$\sum_{i=1}^{n} U_i = n \left(2^{\frac{1}{n}} - 1 \right).$$

The achievable processor utilization of the task set under the RM algorithm decreases with the number of tasks, for large $t_2' \geq t_2$:

$$\lim_{n \to +\infty} n \left(2^{\frac{1}{n}} - 1 \right) = \ln 2 \approx 0.69.$$

Bini *et al.* (2001) proposed another sufficient schedulability test for RM under the same assumptions of Liu and Layland (1973), and the test has the same complexity but is less pessimistic compared with the results of Liu and Layland (1973).

In the general case when the relative deadlines could be less than the periods, the tasks can be scheduled by the deadline monotonic (DM) (Leung and Whitehead, 1982) algorithm, where DM and RM are scheduled in the same way—it is just the assignment to priorities that is different. DM assigns the task priorities based on their relative deadlines: each task's priority is inversely proportional to its relative deadline. The schedulability of the task set can be guaranteed by reducing the tasks' periods to relative deadlines in the RM schedulability test:

$$\sum_{i=1}^{n} \frac{C_i}{D_i} \leq n \left(2^{\frac{1}{n}} - 1 \right).$$

The scheduling problem of combining EDF and FP can be solved by the analysis of hierarchical systems (Kuo and Li, 1999; Zhang and Burns, 2007). In these approaches, the servers (e.g., Ghazalie and Baker, 1995) are required to build a system; therefore, additional behavior needs to be carried out to implement the system. Harbour and Palencia (2003) presented an exact response time analysis for a more general framework according to which the scheduler handles several priority levels. However, all these schedulability analyses including the hierarchical approaches run in pseudo-polynomial time. Since most embedded real-time systems have limited resources and urgent timing requirement, the computational time for online decision control must be bounded; therefore, schedulability tests with polynomial time complexity are required for the hybrid scheduling system. Santos *et al.* (2013) simplified the system model of Harbour and Palencia (2003) and provided a new schedulability analysis for combining EDF and FP systems under the assumption that there is only one high

priority FP task, and this analysis can only apply to implicit-deadline systems, i.e., for each task, its relative deadline must be equal to its period. Under these severe constraints, the following results were provided.

Theorem 2. (Santos *et al.*, 2013) *Let* $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$ *be a set of tasks scheduled by EDF with implicit deadlines, and let* τ_0 *be the only FP task with the highest priority. Then there is no deadline missing if the following condition holds:*

$$\max_{\tau_i \in \Gamma} \left(\left\lceil \frac{T_i}{T_0} \right\rceil \frac{T_0}{T_i} \right) U(\tau_0) + U(\Gamma) \leq 1, \qquad (1)$$

where $U(\tau_0)$ *and* T_0 *are the utilization and period of* τ_0, *respectively.*

Theorem 3. (Santos *et al.*, 2013) *Let* $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$ *be a set of tasks scheduled by EDF with implicit deadlines, and let* τ_0 *be the only FP task with the highest priority. Then there is no deadline missing if the following condition holds:*

$$\max_{\tau_i \in \Gamma} \left(\frac{T_i}{\left\lceil \frac{1-U(\Gamma)}{U(\tau_0)} \frac{T_i}{T_0} \right\rceil} \right) \frac{1}{T_0} \leq 1. \qquad (2)$$

Since Theorems 2 and 3 do not dominate each other, and there is no assumption on the relationship between T_0 and each T_i, we will compare our proposed schedulability analysis with these theorems in Section 5.1.

There are some other research works (e.g., Alcorta-Garciaa *et al.*, 2014; Hwang *et al.*, 2010; Samy *et al.*, 2011) that improve system reliability and the effectiveness of process control by fault detection and isolation systems. López-Estrada *et al.* (2015) addressed the design of a state estimation and sensor fault detection for descriptor-linear parameter varying (D-LPV) systems by considering additional disturbance rejection to improve state and fault estimation.

4. Schedulability analysis for hybrid scheduling systems

In this section, first we consider the schedulable condition for EDF scheduling under the interference of a single high FP task, and then we consider the situation when there is more than one FP task and the tasks can suffer release jitters. The following definition is given.

Definition 1. The *load of tasks or jobs in an interval* $[t_1, t_2]$ is the task execution time that must be completed at or before t_2, i.e., the total execution time of the jobs which arrive before t_2, with the absolute deadlines less than or equal to t_2 (when they are scheduled by EDF), or with higher priorities (when they are scheduled by FP) than the studied tasks, where t_1 and t_2 are any positive numbers and $t_1 < t_2$.

Theorem 4. *Let* $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$ *be a task set which comprises n periodic or sporadic tasks scheduled by EDF, with an extra high priority task τ_0 with its execution time C_0 and period T_0. The system is schedulable if the following inequality holds for all $k \in \{1, 2, \ldots, n\}$:*

$$\frac{C_0}{T_0} + \frac{C_0}{D_k} + \sum_{\substack{i=1 \\ D_i \leq D_k}}^{n} \frac{C_i}{T_i}\left(1 + \frac{T_i - \min(T_i, D_i)}{D_k}\right) \leq 1. \quad (3)$$

Proof. This theorem is proved by contradiction, i.e., by showing that, if a system is not schedulable, then the inequality (3) must not hold. Let d^Δ be a failure time point under any tasks' arrival pattern, and let t_0 be the latest instant before d^Δ such that there are no pending jobs with absolute deadlines $d_i \leq d^\Delta - t_0$ before d^Δ (i.e., no jobs arriving before t_0 contribute their loads to the interval $[t_0, d^\Delta]$). Then we change the tasks' arrival pattern, we move "left" each task's first job which arrives after t_0, to let all the tasks arrive simultaneously at t_0. By the definition of t_0, the total task load in the interval $[t_0, d^\Delta]$ can only be increased, therefore there will still be a failure that occurs at d^Δ. Under this tasks arrival pattern, the processor demand from the EDF tasks in the interval $[t_0, d^\Delta]$ is given by

$$\sum_{\substack{i=1 \\ D_i \leq d^\Delta - t_0}}^{n} \left(\left\lfloor \frac{d^\Delta - t_0 - D_i}{T_i} \right\rfloor + 1\right) C_i$$

and the task load of τ_0 in $[t_0, d^\Delta]$ is given by

$$\left\lceil \frac{d^\Delta - t_0}{T_0} \right\rceil C_0.$$

Therefore, the total task load in $[t_0, d^\Delta]$ is calculated by

$$\left\lceil \frac{d^\Delta - t_0}{T_0} \right\rceil C_0$$
$$+ \sum_{\substack{i=1 \\ D_i \leq d^\Delta - t_0}}^{n} \left(\left\lfloor \frac{d^\Delta - t_0 - D_i}{T_i} \right\rfloor + 1\right) C_i. \quad (4)$$

Since d^Δ is a failure time point, the total task load given by expression (4) must be greater than the length of this interval $d^\Delta - t_0$, so we have

$$d^\Delta - t_0$$
$$< \left\lceil \frac{d^\Delta - t_0}{T_0} \right\rceil C_0$$

$$+ \sum_{\substack{i=1 \\ D_i \leq d^\Delta - t_0}}^{n} \left(\left\lfloor \frac{d^\Delta - t_0 - D_i}{T_i} \right\rfloor + 1\right) C_i$$

$$\leq \left(\frac{d^\Delta - t_0}{T_0} + 1\right) C_0$$

$$+ \sum_{\substack{i=1 \\ D_i \leq d^\Delta - t_0}}^{n} \left(\left(\frac{d^\Delta - t_0 - D_i}{T_i}\right) + 1\right) C_i$$

$$\leq \left(\frac{d^\Delta - t_0}{T_0} + 1\right) C_0$$

$$+ \sum_{\substack{i=1 \\ D_i \leq d^\Delta - t_0}}^{n} \left(\left(\frac{d^\Delta - t_0 - \min(T_i, D_i)}{T_i}\right) + 1\right) C_i$$

$$= \left(\frac{d^\Delta - t_0}{T_0} + 1\right) C_0$$

$$+ \sum_{\substack{i=1 \\ D_i \leq d^\Delta - t_0}}^{n} \frac{C_i}{T_i}(d^\Delta - t_0 + T_i - \min(T_i, D_i))$$

$$= \frac{d^\Delta - t_0}{T_0}C_0 + C_0$$

$$+ \sum_{\substack{i=1 \\ D_i \leq d^\Delta - t_0}}^{n} \frac{C_i}{T_i}(d^\Delta - t_0 + T_i - \min(T_i, D_i)).$$

Therefore, we have

$$d^\Delta - t_0$$
$$< \frac{d^\Delta - t_0}{T_0}C_0 + C_0$$
$$+ \sum_{\substack{i=1 \\ D_i \leq d^\Delta - t_0}}^{n} \frac{C_i}{T_i}(d^\Delta - t_0 + T_i - \min(T_i, D_i)). \quad (5)$$

Dividing both the sides of the above inequality (5) by $d^\Delta - t_0$, we get

$$1 < \frac{C_0}{T_0} + \frac{C_0}{d^\Delta - t_0}$$
$$+ \sum_{\substack{i=1 \\ D_i \leq d^\Delta - t_0}}^{n} \frac{C_i}{T_i}\left(1 + \frac{T_i - \min(T_i, D_i)}{d^\Delta - t_0}\right). \quad (6)$$

Let D_k be the largest $D_i \leq d^\Delta - t_0$, i.e., $D_k = \max\{D_i | D_i \leq d^\Delta - t_0, i = 1, 2, \ldots, n\}$. Then the inequality (6) becomes

$$1 < \frac{C_0}{T_0} + \frac{C_0}{D_k} + \sum_{\substack{i=1 \\ D_i \leq D_k}}^{n} \frac{C_i}{T_i}\left(1 + \frac{T_i - \min(T_i, D_i)}{D_k}\right). \quad (7)$$

Since the length of $[t_0, d^\Delta]$ is uncertain, D_k could be any value in $\{D_i | i = 1, 2, \ldots, n\}$. If the system is

unschedulable, the inequality (7) must be true at a value of $D_k \in \{D_i | i = 1, 2, \ldots, n\}$. Therefore, if $\forall D_k \in \{D_i | i = 1, 2, \ldots, n\}$, the inequality (3) holds, and so the system is schedulable. ∎

In realistic situations, the tasks can suffer from release jitters (Burns and Wellings, 2009; Buttazzo, 2005); when a job of task τ_i arrives at time t with its absolute deadline $t + D_i$, it will be released for execution at the latest time $t + J_i$ (the actual release time could be anywhere in the interval $[t, t + J_i]$). This could happen when a task arrives at a time when the capacity of its server has been exhausted in hierarchical scheduling systems (Davis and Burns, 2005; Zhang and Burns, 2007), or if task and clock periods are incompatible, or when an arrived task is waiting for a signal from another task in a distributed system (Tindell and Clark, 1994).

The result of Theorem 4 can be extended to the situation when there is more than one FP task, and the tasks can suffer from their release jitters. The schedulability analysis is given by the following theorem.

Theorem 5. *Let* $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$ *be a set of tasks scheduled by EDF, under the interference of a set of tasks* $\Psi = \{\varphi_1, \varphi_2, \ldots, \varphi_m\}$ *scheduled by an FP policy. The task set* Γ *is schedulable if* $\forall k \in \{1, 2, \ldots, n\}$:

$$
\sum_{x=1}^{m} \frac{1 + J_x^{fp}}{T_x^{fp}} C_x^{fp} + \frac{1}{D_k - J_k} \sum_{x=1}^{m} C_x^{fp}
$$
$$
+ \sum_{\substack{i=1 \\ D_i - J_i \leq D_k - J_k}}^{n} \frac{C_i}{T_i} \left(1 + \frac{T_i + J_i - \min(T_i, D_i)}{D_k - J_k} \right)
$$
$$
\leq 1, \quad (8)
$$

where J_x^{fp} *is the maximum release jitter of* φ_x, *and* J_i *is the maximum release jitter of* τ_i.

Proof. This theorem can be proven by a similar principle as in the discussions of Theorem 4. Let d^Δ be a failure time point under ANY tasks' arrival pattern, and let t_0 be the latest instance before d^Δ such that there are no pending jobs being released before d^Δ with their absolute deadlines $d_i \leq d^\Delta - t_0$, i.e., there are no pending jobs being released before t_0 contributing their loads to the interval $[t_0, d^\Delta]$. Note that from the definition of a release jitter, there could be jobs (with $d_i \leq d^\Delta - t_0$) arriving before t_0, but released after t_0, with their arrivals at the earliest time $t_0 + 1/\infty - J_i$ when they suffered their maximum release jitters, where $1/\infty$ is an infinitesimally small positive number, as shown in Fig. 1.

Then we change the tasks' arrival pattern; we move each task's first released job "left" after t_0, to let all tasks be released simultaneously at t_0 after having experienced their maximum release jitters, and then the subsequent jobs of each task are released at their maximum rate
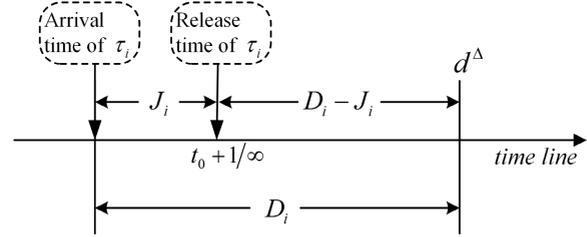
Fig. 1. Arrival and release time of a τ_i's job when it suffered from a maximum release jitter J_i.

(without jitter). By the definition of t_0, the total task load in the interval $[t_0, d^\Delta]$ can only be increased by this behavior; therefore, there will still be a failure that occurs at d^Δ. Since any task with $D_i - J_i \leq d^\Delta - t_0$ could contribute its load to $[t_0, d^\Delta]$ under this task's arrival pattern, the processor time demanded by the EDF tasks in the interval $[t_0, d^\Delta]$ is given by

$$
\sum_{\substack{i=1 \\ D_i - J_i \leq d^\Delta - t_0}}^{n} \left(\left\lfloor \frac{d^\Delta - t_0 - D_i}{T_i} \right\rfloor + 1 \right) C_i,
$$

and the total load from the tasks of Ψ in $[t_0, d^\Delta]$ is given by

$$
\sum_{x=1}^{m} \left\lceil \frac{d^\Delta - t_0 + J_x^{fp}}{T_x^{fp}} \right\rceil C_x^{fp}.
$$

Therefore, the total task load of the whole task set in $[t_0, d^\Delta]$ is calculated by

$$
\sum_{x=1}^{m} \left\lceil \frac{d^\Delta - t_0 + J_x^{fp}}{T_x^{fp}} \right\rceil C_x^{fp}
$$
$$
+ \sum_{\substack{i=1 \\ D_i - J_i \leq d^\Delta - t_0}}^{n} \left(\left\lfloor \frac{d^\Delta - t_0 + J_i - D_i}{T_i} \right\rfloor + 1 \right) C_i. \quad (9)
$$

Since d^Δ is a failure time point, the total task load calculated by the expression (9) must be greater than the length of this interval $d^\Delta - t_0$, so we have

$$
d^\Delta - t_0
$$
$$
< \sum_{x=1}^{m} \left\lceil \frac{d^\Delta - t_0 + J_x^{fp}}{T_x^{fp}} \right\rceil C_x^{fp}
$$
$$
+ \sum_{\substack{i=1 \\ D_i - J_i \leq d^\Delta - t_0}}^{n} \left(\left\lfloor \frac{d^\Delta - t_0 + J_i - D_i}{T_i} \right\rfloor + 1 \right) C_i
$$

$$\leq \sum_{x=1}^{m} \left(\frac{d^\Delta - t_0 + J_x^{fp}}{T_x^{fp}} + 1 \right) C_x^{fp}$$
$$+ \sum_{\substack{i=1 \\ D_i - J_i \leq d^\Delta - t_0}}^{n} \left(\left(\frac{d^\Delta - t_0 + J_i - D_i}{T_i} \right) + 1 \right) C_i$$

$$\leq \sum_{x=1}^{m} \left(\frac{d^\Delta - t_0 + J_x^{fp}}{T_x^{fp}} + 1 \right) C_x^{fp}$$
$$+ \sum_{\substack{i=1 \\ D_i - J_i \leq d^\Delta - t_0}}^{n} \left(\left(\frac{d^\Delta - t_0 + J_i - \min(T_i, D_i)}{T_i} \right) \right.$$
$$\left. + 1 \right) C_i$$

$$= \sum_{x=1}^{m} \left(\frac{d^\Delta - t_0 + J_x^{fp}}{T_x^{fp}} + 1 \right) C_x^{fp}$$
$$+ \sum_{\substack{i=1 \\ D_i - J_i \leq d^\Delta - t_0}}^{n} \frac{C_i}{T_i} \left(d^\Delta - t_0 + J_i \right.$$
$$\left. - \min(T_i, D_i) + T_i \right)$$

$$= \sum_{x=1}^{m} \frac{d^\Delta - t_0 + J_x^{fp}}{T_x^{fp}} C_x^{fp} + \sum_{x=1}^{m} C_x^{fp}$$
$$+ \sum_{\substack{i=1 \\ D_i - J_i \leq d^\Delta - t_0}}^{n} \frac{C_i}{T_i} \left(d^\Delta - t_0 + T_i \right.$$
$$\left. + J_i - \min(T_i, D_i) \right).$$

Therefore, we have

$$d^\Delta - t_0$$
$$< \sum_{x=1}^{m} \frac{d^\Delta - t_0 + J_x^{fp}}{T_x^{fp}} C_x^{fp} + \sum_{x=1}^{m} C_x^{fp}$$
$$+ \sum_{\substack{i=1 \\ D_i - J_i \leq d^\Delta - t_0}}^{n} \frac{C_i}{T_i} \left(d^\Delta - t_0 + T_i \right.$$
$$\left. + J_i - \min(T_i, D_i) \right). \tag{10}$$

Dividing both the sides of the inequality (10) by $d^\Delta - t_0$, we have

$$1 < \sum_{x=1}^{m} \frac{1 + J_x^{fp}}{T_x^{fp}} C_x^{fp} + \frac{1}{d^\Delta - t_0} \sum_{x=1}^{m} C_x^{fp}$$
$$+ \sum_{\substack{i=1 \\ D_i - J_i \leq d^\Delta - t_0}}^{n} \frac{C_i}{T_i} \Big(1 \tag{11}$$
$$+ \frac{T_i + J_i - \min(T_i, D_i)}{d^\Delta - t_0} \Big).$$

Let $D_k - J_k$ be the largest $D_i - J_i \leq d^\Delta - t_0$, i.e., $D_k - J_k = \max \{ D_i | D_i - J_i \leq d^\Delta - t_0, i = 1, 2, \ldots, n \}$.

Then the inequality (11) becomes

$$1 < \sum_{x=1}^{m} \frac{1 + J_x^{fp}}{T_x^{fp}} C_x^{fp} + \frac{1}{D_k - J_k} \sum_{x=1}^{m} C_x^{fp}$$
$$+ \sum_{\substack{i=1 \\ D_i - J_i \leq D_k - J_k}}^{n} \frac{C_i}{T_i} \Big(1 \tag{12}$$
$$+ \frac{T_i + J_i - \min(T_i, D_i)}{D_k - J_k} \Big).$$

As the length of $[t_0, d^\Delta]$ is an uncertain value, $D_k - J_k$ could be any value in $\{D_1 - J_1, D_2 - J_2, \ldots, D_n - J_n\}$. If the system is unschedulable, the inequality (12) must be true at a value of $D_k - J_k \in \{D_i - J_i | i = 1, 2, \ldots, n\}$. Therefore, if $\forall D_k - J_k \in \{D_i - J_i | i = 1, 2, \ldots, n\}$, the inequality (8) is true, and thus the system is schedulable. ∎

Complexity of schedulability analysis. The inequality (8) or (3) has a complexity of $O(m + n)$, since the inequality is required to be checked at every $k \in \{1, 2, \ldots n\}$ to complete a schedulability test, the proposed analysis has a complexity of $O((m + n)n)$.

Schedulability tests of the FP tasks of Ψ. Since we assume that the FP tasks of Ψ have higher priorities than the EDF tasks of Γ, schedulability analysis of the FP tasks can be carried out by the existing results for the traditional FP systems. In consequence, when there is more than one FP task in a hybrid-scheduled system, the schedulability of the Ψ tasks can be tested by worst-case response time analysis (Audsley *et al.*, 1993; Joseph and Pandya, 1986), where each task's relative deadline could be less than or equal to its period. For arbitrary deadline FP systems in which tasks' relative deadlines could be greater than their periods, extended response time analysis (Tindell *et al.*, 1994) could be employed to test the schedulability of the FP tasks. The sufficient analysis reviewed in Section 2 can also be deployed to test the FP tasks' schedulability.

5. Experimental evaluations

In this section, we evaluate the performance of our proposed schedulability analysis through extensive experiments on a large number of randomly generated task sets. The tasks' utilizations in the experiments are generated according to the policies described by Bini and Buttazzo (2005) to ensure the values of utilizations are uniformly distributed in $(0, 1)$. The task periods are generated according to an exponential distribution similar to the policies described by Davis *et al.* (2008) as well as Zhang and Burns (2009) when the task periods value range is determined. Each point on the x-axis in the experimental results is based on 10,000 randomly generated task sets.

In the experiments of Section 5.1, we will compare the acceptance ratio of our proposed analysis with the existing schedulability tests of Theorems 2 and 3. The acceptance ratio of these schedulability tests is compared by using the same task sets which are randomly generated as described above; this ensures the fairness of the comparisons. Let τ_0 be the only FP task in Ψ with its worst-case execution time C_0 and the period T_0. Theorems 2 and 3 are compared since there are no assumptions that $T_0 < \min_{\tau_i \in \Gamma}(T_i)$, and they do not dominate each other (Santos *et al.*, 2013). As the RM priority assignment policy does not apply to the situation when all the tasks are scheduled by EDF excepting just one FP task in a system, the relationships between T_0 and the periods of the EDF tasks are varied in different experiments to investigate the performance of the analyses.

All parameters of each task set, including the EDF and FP tasks, are randomly generated according to the default policies, e.g., if there are 50 EDF tasks with an extra FP task in a task set, then 51 tasks' parameters are randomly generated, and one of the 51 tasks is chosen to be the FP task. The criterion of the FP task selection is described in each experiment. Since the EDF and FP tasks are generated together in each task set, in this section, we denote τ_i to be any task in the subset Γ or Ψ e.g., $\max_{\tau_i \in \Gamma \cup \Psi}(T_i)$ represents the maximum task period in a task set, including both Γ and Ψ.

5.1. Experimental comparisons.
Although our proposed analysis has no restrictions on the number of FP tasks and parameters, in order to compare the existing sufficient schedulability analysis, in this section we assume that there is only one FP task in the subset Ψ, and we impose the restriction that each task's relative deadline is equal to its period in Experiments (A)–(E). The acceptance ratio of our proposed analysis when there is more than one task in Ψ is reported in Section 5.2.

Experiment A. The task sets are randomly generated according to the policy described above. For each task set, we let the total number of tasks be 51, so there are 50 EDF tasks with an extra FP task τ_0. We chose the task with a median value of period between $\min_{\tau_i \in \Gamma \cup \Psi}(T_i)$ and $\max_{\tau_i \in \Gamma \cup \Psi}(T_i)$ as the FP task. Write $T_{\min} = \min_{\tau_i \in \Gamma \cup \Psi}(T_i)$, $T_{\max} = \max_{\tau_i \in \Gamma \cup \Psi}(T_i)$, and let the maximum value of $T_{\max}/T_{\min}$ for each task set be 1000. We investigate the influence on the performance when the value of the total utilization $U(\Gamma) + U(\tau_0)$ is changed from 0.66 to 0.99. The results of this experiment are illustrated in Fig. 2.

Experiment B. This experiment investigates the performance of the analyses when the number of tasks of each task set is varied. Note that in Fig. 3, when there are four EDF tasks on the x-axis, this means that
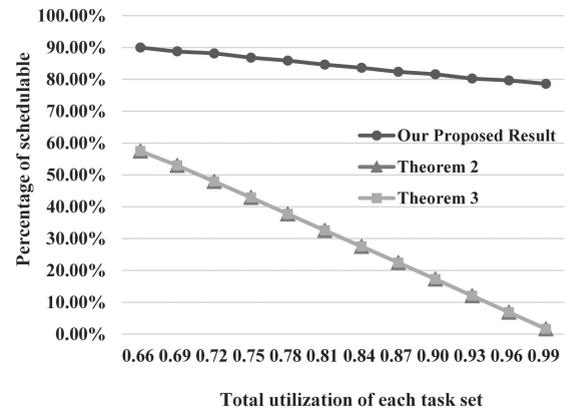
Fig. 2. Impact of the utilization of a task set.

totally five tasks are randomly generated for each task set, and one of the five tasks is chosen to be the FP task. In this experiment, the task with a median period's value between $\min_{\tau_i \in \Gamma \cup \Psi}(T_i)$ and $\max_{\tau_i \in \Gamma \cup \Psi}(T_i)$ is chosen to be the FP task. The total utilization $U(\Gamma) + U(\tau_0)$ of each task set is set to be 0.9, and the maximum value of $\max_{\tau_i \in \Gamma \cup \Psi}(T_i)/\min_{\tau_i \in \Gamma \cup \Psi}(T_i)$ is set to be 1000. The results of this experiment are illustrated in Fig. 3.
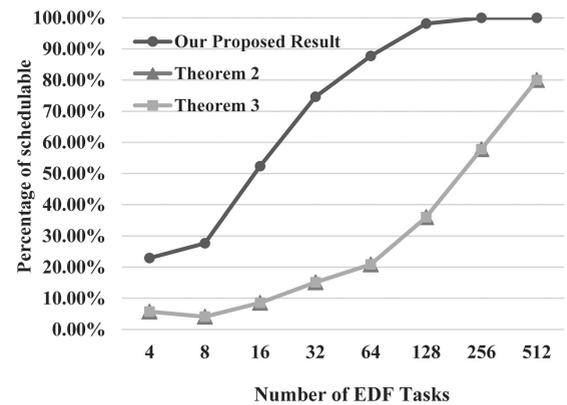
Fig. 3. Impact of the number of tasks.

Experiment C. Write $T_{\min} = \min_{\tau_i \in \Gamma \cup \Psi}(T_i)$ and $T_{\max} = \max_{\tau_i \in \Gamma \cup \Psi}(T_i)$, where this experiment investigates the performance of the results when the maximum value of $T_{\max}/T_{\min}$ is changed from 10 to 10^6. For each task set, the total utilization $U(\Gamma) + U(\tau_0)$ is set to be 0.9, the number of tasks is set to be 51, and the task with an intermediate period's value between $T_{\min}$ and $T_{\max}$ is chosen to be τ_0. The results of this experiment are illustrated in Fig. 4.

Experiment D. In this experiment, we investigate the results when the task with the minimum period of each task set is chosen as the FP task τ_0, i.e.,
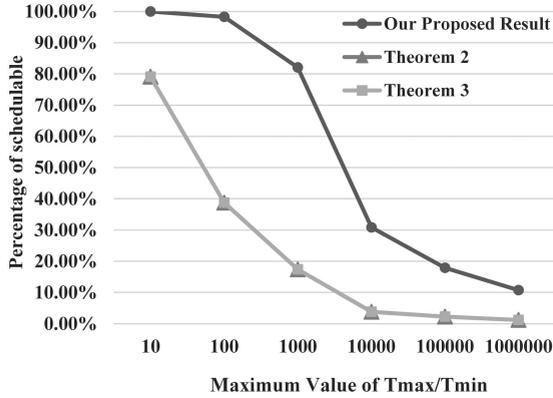
Fig. 4. Impact of the task periods range.

$T_0 \leq \min_{\tau_i \in \Gamma}(T_i)$. For each task set, we still let the total number of tasks be 51, so there are 50 EDF tasks with one FP task. The maximum value of $\max_{\tau_i \in \Gamma \cup \Psi}(T_i) / \min_{\tau_i \in \Gamma \cup \Psi}(T_i)$ is 1000. The value of the total utilization $U(\Gamma) + U(\tau_0)$ is changed from 0.78 to 0.99. The results of this experiment are shown in Fig. 5.
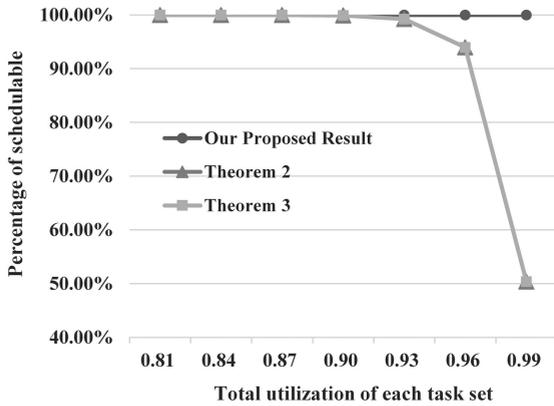
Fig. 5. Impact of the utilization of a task set.

Experiment E. In this experiment, we investigate the influence of the FP task choice. We let the FP task τ_0 be selected from each task set $\Omega = \{\text{task\_1, task\_2, \ldots,} \text{task\_51}\}$, where 51 randomly generated tasks have been sorted in the increasing order of their periods, i.e. for any task_i with a period T_i^{Ω} and a task_$(i + 1)$ with a period T_{i+1}^{Ω} belonging to Ω, $T_i^{\Omega} \leq T_{i+1}^{\Omega}$, where $1 \leq i \leq 50$. When task_i is chosen to be τ_0, let $T_0 = T_i^{\Omega}$ and $C_0 = C_i^{\Omega}$. There are always 50 EDF tasks with an extra FP task in each task set. Note in Fig. 6, when the value of the x-axis is "task_21", this means for each task set that task_21 from Ω is always chosen as the FP task. Let the total utilization of each task set be 0.9, and let the maximum value of $\max_{\tau_i \in \Gamma \cup \Psi}(T_i) / \min_{\tau_i \in \Gamma \cup \Psi}(T_i)$ be 1000. The results of this experiment are illustrated in Fig. 6.
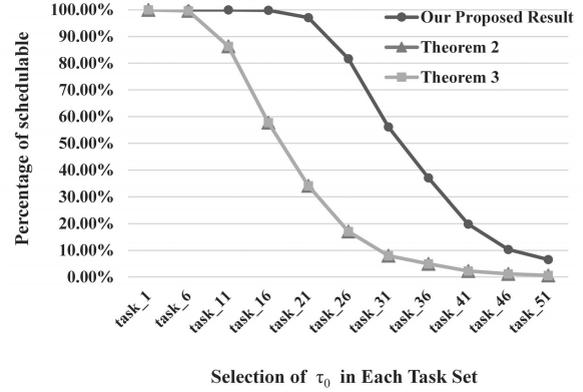
Fig. 6. Impact of selection of τ_0 in each task set when 51 randomly generated tasks are sorted by the increasing order of their periods.

5.2. Experimental results of more than one FP task.
This section investigates the performance of our proposed schedulability analysis when there is more than one FP task in Ψ. Since the existing results on sufficient schedulability analysis of this hybrid scheduling model only apply to the situation where there is only one FP task in a system, there are no comparisons in the experiments of this section. There are two lines in each experiment diagram; "EDF task sets" represents the acceptance ratio of the proposed analysis of Theorem 5, and "Whole task sets" represents the acceptance ratio of the whole task sets including the EDF and FP tasks. The worst-case response time analysis (Audsley *et al.*, 1993; Joseph and Pandya, 1986) is employed to test the schedulability of the FP tasks of Ψ. If both the task subsets Γ and Ψ are schedulable, then this task set is schedulable.

The parameters of each task set are randomly generated according to the default policy described at the beginning of this section. For each task set, 50 tasks' parameters are generated, and 10 tasks with the minimum relative deadlines are chosen to be the FP tasks; therefore, there are 40 EDF tasks and 10 FP tasks in each task set. The maximum value of $\max_{\tau_i \in \Gamma \cup \Psi}(T_i) / \min_{\tau_i \in \Gamma \cup \Psi}(T_i)$ is set to be 1000.

Experiment F. In this experiment, we let each task's relative deadline to be equal to its period. When $U(\Gamma) + U(\Psi) \leq 0.99$, no matter what the value of the total utilization is, the EDF task subsets and the whole task sets are almost 100% schedulable. When $U(\Gamma) + U(\Psi) > 0.99$, the experimental results are illustrated in Fig. 7. Note that the two lines are superposed in the graph.

Experiment G. In this experiment, we remove the restriction that each task's relative deadline equals its period, to allow the relative deadline of each task in the task set (including Γ and Ψ) to be randomly generated between its worst-case execution time C_i and its period T_i. The experimental results are illustrated in Fig. 8.
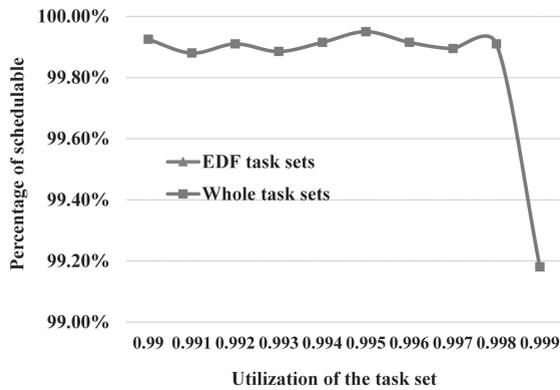
Fig. 7. Impact of the total utilization of each task set when each $D_i = T_i$.
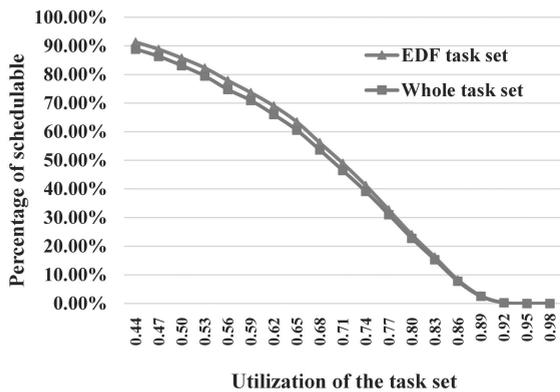
Fig. 8. Impact of the total utilization of each task set when each D_i is randomly generated in $[C_i, T_i]$.

5.3. Summary of the experiments. This section compares the proposed result with the existing tests which also run in polynomial time. From the experiments of Section 5.1, our proposed analysis significantly improves the percentage of the task sets which are judged to be schedulable, and it has approximately a 300% improvement on the average across all experiments of this section. When the task with a median value of period between $\min_{\tau_i \in \Gamma \cup \Psi}(T_i)$ and $\max_{\tau_i \in \Gamma \cup \Psi}(T_i)$ of each task set is chosen as the FP task, the existing results have a low acceptance rate in most cases, and our proposed analysis has a 400% improvement on a typical situation. When the task with the minimum period is chosen as the FP task, the acceptance ratio is very close to 100% for all the analyses when the total utilization is less than 0.9, but the existing results decrease sharply when the total utilization is greater than 0.93. From the experiments of Section 5.2, when there is more than one FP task in each task set, the proposed analysis has almost a 100% acceptance ratio for the implicit-deadline task sets when the total utilization is less than 0.998. For arbitrary relative deadline systems when there is more than one task in Ψ, the acceptance ratio is affected greatly by the total utilization.

6. Conclusion

In this paper, we have presented a new sufficient schedulability analysis for hybrid scheduling real-time systems where both EDF and FP are employed in the same system. The proposed analysis has a polynomial time complexity $O\left(n(n+m)\right)$. The proposed analysis can be applied to the situation when there is more than one FP task. The tasks could suffer release jitters, and there are no restrictions on the parameters of both the EDF and FP tasks, i.e., the relative deadline of each task can be less than, equal to, or greater than its period. We have compared our proposed schedulability analysis with the existing tests with polynomial time complexity through the use of extensive experiments, and we have shown that the proposed result exhibits a significant improvement: it has approximately a 300% improvement on the acceptance ratio across all experiments of this paper (i.e., more than 400,000 randomly generated task sets). We also observed that when there is more than one FP task in each task set our proposed analysis has almost a 100% acceptance ratio for the implicit-deadline task sets when the total utilization is not very close to one.

References

Albers, K. and Slomka, F. (2004). An event stream driven approximation for the analysis of real-time systems, *Proceedings of the 16th Euromicro Conference on Real-Time Systems, Catania, Sicily, Italy*, pp. 187–195.

Alcorta-Garcia, E., Saucedo-Flores, S. and Diaz-Romero, D.A. (2014). Intelligent fault diagnosis in nonlinear systems, *Intelligent Automation and Soft Computing* **20**(2): 201–212.

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

Bini, E. and Buttazzo, G.C. (2005). Measuring the performance of schedulability tests, *Real-Time Systems* **30**(1–2): 129–154.

Bini, E., Buttazzo, G.C. and Buttazzo, G.M. (2001). A hyperbolic bound for the rate monotonic algorithm, *Proceedings of the 13th Euromicro Conference on Real-Time Systems, Delft, The Netherlands*, pp. 59–66.

Burns, A. and Wellings, A.J. (2009). *Real-Time Systems and Programming Languages*, 4th Edn., Addison Wesley, Boston, MA.

Buttazzo, G. (2011). *Hard Real-Time Computing Systems*, Springer US, New York, NY.

Chakraborty, S., Kunzli, S. and Thiele, L. (2002). Approximate schedulability analysis, *23rd IEEE Real-Time Systems Symposium, Austin, TX, USA*, pp. 159–168.

Davis, R.I. and Burns, A. (2005). Hierarchical fixed priority preemptive scheduling, *26th IEEE Real-Time Systems Symposium, Miami, FL, USA*, pp. 389–398.

Davis, R.I., Zabos, A. and Burns, A. (2008). Efficient exact schedulability tests for fixed priority real-time systems, *IEEE Transactions on Computers* 57(9): 1261–1276.

Dertouzos, M.L. (1974). Control robotics: The procedural control of physical processes, *IFIP Congress, Stockholm, Sweden*, pp. 807–813.

Devi, M. (2003). An improved schedulability test for uniprocessor periodic task systems, *Proceedings of the 15th Euromicro Conference on Real-Time Systems, Porto, Portugal*, pp. 23–30.

Ghazalie, T.M. and Baker, T.P. (1995). Aperiodic servers in a deadline scheduling environment, *Real-Time Systems* 9(1): 31–67.

Harbour, M.G. and Palencia, J.C. (2003). Response time analysis for tasks scheduled under EDF within fixed priorities, *24th IEEE Real-Time Systems Symposium, Cancun, Mexico*, pp. 200–209.

Hwang, I., Kim, S., Kim, Y. and Seah, C.E. (2010). A survey of fault detection, isolation, and reconfiguration methods, *IEEE Transactions on Control Systems Technology* 18(3): 636–653.

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

Kuo, T.-W. and Li, C.-H. (1999). A fixed priority driven open environment for real-time applications, *20th IEEE Real-Time Systems Symposium, Phoenix, AZ, USA*, pp. 256–267.

López-Estrada, F.-R., Ponsart, J.-C., Theilliol, D., Astorga-Zaragoza, C.-M. and Camas-Anzueto, J.-L. (2015). Robust sensor fault estimation for descriptor-LPV systems with unmeasurable gain scheduling functions: Application to an anaerobic bioreactor, *International Journal of Applied Mathematics and Computer Science* 25(2): 233–244, DOI: 10.1515/amcs-2015-0018.

Leung, J. and Whitehead, J.W. (1982). On the complexity of fixed priority scheduling of periodic real-time tasks, *Performance Evaluation* 2(4): 237–250.

Liu, C.L. and Layland, J.W. (1973). Scheduling algorithm for multiprogramming in a hard real-time environment, *Journal of the ACM* 20(1): 40–61.

Liu, J.W.S. (2000). *Real-Time Systems*, Prentice-Hall, Upper Saddle River, NJ.

Samy, I., Postlethwaite, I. and Gu, D.-W. (2011). Survey and application of sensor fault detection and isolation schemes, *Control Engineering Practice* 19(7): 658–674.

Santos, J.A., Jr., Lima, G. and Bletsas, K. (2013). Efficient schedulability tests for real-time embedded systems with urgent routines, *Design Automation for Embedded Systems* 18(1–2): 19–38.

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

Tindell, K. and Clark, J. (1994). Holistic schedulability analysis for distributed hard real-time systems, *Microprocessors and Microprogramming* 40(2–3): 117–134.

Zhang, F. and Burns, A. (2007). Analysis of hierarchical EDF pre-emptive scheduling, *28th IEEE Real-Time Systems Symposium (RTSS), Tucson, AZ, USA*, pp. 423–434.

Zhang, F. and Burns, A. (2009). Schedulability analysis for real-time systems with EDF scheduling, *IEEE Transactions on Computers* 58(9): 1250–1258.

Fengxiang Zhang received his Ph.D. degree in computer science from the University of York (UK) in 2009. He is currently an associate professor in the School of Computer and Information Science, Southwest University, China. His main research interests are in the areas of scheduling analysis of real-time and embedded systems.

Yanfeng Zhai is a Master's degree student in the School of Computer and Information Science, Southwest University, China. His research interests are in the area of real-time scheduling.

Jianwei Liao received his Ph.D. degree in computer science from the University of Tokyo, Japan, in 2012. He is currently an associate professor in the School of Computer and Information Science, Southwest University, China. His research interests are high performance storage systems for distributed computing environments and real-time processing.