

# A Safe Stochastic Analysis with Relaxed Limitations on the Periodic Task Model

Kanghee Kim and Chang-Gun Lee, *Member, IEEE*

**Abstract**—This paper proposes a safe stochastic analysis for fixed-priority scheduling, which is applicable to a broader spectrum of periodic tasks than the ones analyzable by any of existing techniques. The proposed analysis can find a safe upper bound of deadline miss probability for periodic tasks with (1) arbitrary execution time distributions, (2) varying interrelease times with the period as the minimum, and (3) the maximum utilization factor  $U^{max}$  that can be greater than 1. One challenge for this is that the release times of tasks are not known a priori because we are not limiting the interrelease times of each task to a constant, i.e., the period. In such a situation, the relative phases of task instances at run time can be arbitrary. Thus, we need to consider all possible phase combinations among jobs to find the worst case deadline miss probability, which is not tractable. To handle this difficulty, we first derive the worst case phase combination for harmonic task sets. Then, we present a safe way to transform a non-harmonic task set to a harmonic task set such that the deadline miss probabilities obtained with the worst case phase combination for the transformed harmonic task set are guaranteed to be worse than those for the original non-harmonic task set with all possible phase combinations. Therefore, the worst case deadline miss probabilities of the transformed harmonic tasks can be used as safe upper bounds of deadline miss probabilities of the original non-harmonic tasks. Through experiments, we show that the safe upper bound computed by the proposed analysis is tight enough for practical uses.

**Index Terms**—Real-time and embedded systems, scheduling, stochastic analysis, worst-case analysis, periodic task model.



## 1 INTRODUCTION

Traditionally, the existing studies on schedulability analysis of real-time systems have tried to deterministically guarantee that the deadlines of all tasks are met assuming their worst case execution times [11], [14], [17]. However, such deterministic guarantee is too harsh for many practical applications where the application can tolerate occasional deadline misses as long as the ratio of deadline misses is less than a given threshold [4], [19], [23]. For example, a VOD (video on demand) server should serve multiple concurrent video sessions by periodically reading movie frames, transcoding, and delivering them to the customers. Due to the dynamics of the movie contents, the size of each movie frame may differ resulting in varying execution times for different frames. For this application, it is too pessimistic to assume the worst case execution time for every frame for the deterministic deadline guarantee, which will significantly limit the number of sessions that the VOD server can concurrently serve. Practically, the VOD server can serve a much larger number of sessions without any observable quality degradation, as long as the deadline miss probability is less than a given threshold. Even certain control applications, such as multi-target tracking radar systems, require such a probabilistic guarantee [4], [19] in order to maintain the probability of control loss or track loss under a threshold.

For such applications, the traditional deterministic guarantee results in an over-pessimistic decision. More importantly, if the execution times vary widely, it may severely underutilize the system resources since the assumed worst case execution times can be far from the averages. To rectify this problem, recent studies have proposed various stochastic analysis methods [1], [6], [8], [12], [13], [15], [18], [22] to calculate the deadline miss probability of each task to be compared with the required threshold. However, they work only for task sets with limited conditions. In the following, we categorize them according to their limitations:

- *K. Kim is with Mobile Communication Division, Samsung Electronics Co., Ltd., Suwon, 443-742 Korea.  
E-mail: kang.hee.kim@samsung.com*
- *C.-G. Lee is with School of Computer Science and Engineering, Seoul National University, Seoul, 151-742 Korea.  
E-mail: cglee@snu.ac.kr*

*Manuscript received January 15, 2008; revised June 9, 2008; accepted August 14, 2008.*

- **Limited execution times and interrelease times:** Real-Time Queueing Theory [12], [13] works only for tasks with Poisson arrivals and exponential execution times. Although such a task model may be acceptable in a sufficiently populated system, the main problem of the method is that it is not safe, that is, it cannot provide either the exact deadline miss probability or an upper bound of it due to approximation with no safeness enforcement.
- **Limited interrelease times:** The analysis methods proposed in [8], [15], [18] can calculate accurate deadline miss probabilities for tasks with arbitrary execution time distributions. However, they work only when the interrelease times of each task are constant. It is obvious that this assumption of constant interrelease times does not hold in a real system, due to the scheduling effect of the operating system, which introduces jitters into task release times.
- **Limited maximum utilization factor:** The analysis methods proposed in [6], [22] can calculate a safe upper bound of deadline miss probability even when the execution time distributions are arbitrary and the interrelease

times vary with the period as the minimum. They can analyze fixed-priority systems such as Rate Monotonic [17] and Deadline Monotonic [16] using stochastic extensions of the Time Demand Analysis (TDA) [14]. However, they can analyze the system only when the maximum utilization factor is less than or equal to 1. This is a serious limitation because widely varying execution times of stochastic tasks commonly make the maximum utilization factor greater than 1 while keeping the average utilization factor below 1, which gives a high possibility of meeting the required deadline miss probability.

- **Limited scheduling model:** The analysis proposed in [1] makes the stochastic analysis tractable by using a reservation-based scheduling model. However, it does not work for the generic fixed-priority scheduling model, which is most common in commercial real-time operating systems [20].

In this paper, we propose a new stochastic analysis method that overcomes all the above limitations. That is, the proposed method can calculate a safe upper bound of the deadline miss probability even when (1) the execution time distribution of each task is arbitrarily given, (2) the interrelease times vary with the given period as the minimum, (3) the maximum utilization factor can be greater than 1, and (4) the generic fixed-priority scheduling model is used. For this, we base the new analysis on our previous exact stochastic analysis [8] that can calculate the exact response time distribution for each task. Although the previous analysis overcomes three of the aforementioned limitations, it still suffers from one serious limitation, i.e., constant interrelease times. In a realistic situation where the interrelease times can vary with the given period as the minimum, the relative phases among jobs can be arbitrary. It is worth noting that, even if we add the simple minded *in-phase* condition — jobs of all the tasks are released at the same time — to our previous analysis, it does not give the worst case in the stochastic sense as we will show later. Moreover, it will be explained that Lehoczky’s approach [11], which picks the worst case out of all the jobs in a busy interval assuming the in-phase condition at the start of the interval and constant interrelease times afterward, neither can give the real worst case in the stochastic sense. Intuitively, in order to find the worst case deadline miss probability, we need to consider all possible phase combinations among jobs, which is not tractable. In order to safely cover all possible phase combinations without explicitly enumerating all of them, we take a two-step approach. First, we prove that when the task periods, i.e., minimum interrelease times, are *harmonic*—each task period is an integer multiple of every shorter period, the response time distribution of each task obtained assuming the in-phase condition is the worst among the ones obtained assuming all possible phase combinations. This observation is important in that the safeness of the in-phase condition, which is already known for  $U^{max} \leq 1$  [6], [22], can be extended to the case of  $U^{max} > 1$ . Note, however, that this extension is valid only for harmonic task sets. Second, we extend our analysis to a system of non-harmonic tasks by transforming the non-harmonic tasks to harmonic tasks. We prove that there

always exists a safe transformation such that the response time distributions of the transformed harmonic tasks are worse than those of the original non-harmonic tasks. As a consequence, the response time distribution of each transformed harmonic task assuming in-phase is worse than that of the original non-harmonic task considering all possible phase combinations. Therefore, by comparing the response time distributions of the transformed in-phase harmonic tasks with the given relative deadlines, we can find safe upper bounds of the deadline miss probabilities of the original non-harmonic tasks.

This paper is organized as follows. In Section 2, we explain our system model and summarize our previous exact stochastic analysis for periodic tasks with constant interrelease times. In Section 3, we describe our proposed analysis that can be applied even for varying interrelease times. Section 4 gives experimental results to evaluate the analysis accuracy and Section 5 describes a theoretical study on the analysis accuracy. Finally, in Section 6, we conclude the paper and give directions for future research.

## 2 PRELIMINARIES

### 2.1 System model

In our stochastic system, each task  $\tau_i$  ( $i = 1, \dots, n$ ) is described by a tuple  $(T_i, C_i, D_i)$ , where  $T_i$  is the period of the task, equal to the minimum interrelease time,  $C_i$  its execution time, and  $D_i$  its relative deadline. The execution time  $C_i$  is a discrete random variable whose probability mass function we know a priori, denoted by  $f_{C_i}(t) = \mathbf{P}\{C_i = t\}$ . The execution time distribution can be given by a measurement-based analysis such as automatic tracing analysis [21] or an analytical program analysis such as probabilistic worst case execution time analysis [2]. The period or the minimum interrelease time  $T_i$  and the relative deadline  $D_i$  are constant values given by application characteristics. The relative deadline  $D_i$  can be smaller than, equal to, or greater than the period  $T_i$ .

In the task model, each task is assumed to give rise to an infinite sequence of instances, called jobs. The  $j$ th job of task  $\tau_i$  is denoted by  $J_{i,j}$ . The  $J_{i,j}$ ’s release time denoted by  $\lambda_{i,j}$  is separated at least by the minimum interrelease time  $T_i$  from the previous release time  $\lambda_{i,j-1}$ , that is,  $\lambda_{i,j} \geq \lambda_{i,j-1} + T_i$ . Once released,  $J_{i,j}$  requires  $C_{i,j}$  execution time, following the given distribution  $f_{C_i}(t)$  for  $\tau_i$ . The execution time of a job is assumed to be independent of other jobs of the same task and those of other tasks. For  $J_{i,j}$ , its absolute deadline denoted by  $d_{i,j}$  is given by its release time  $\lambda_{i,j}$  plus the relative deadline  $D_i$ , i.e.,  $d_{i,j} = \lambda_{i,j} + D_i$ . Thus, each job  $J_{i,j}$  is described by a tuple  $(\lambda_{i,j}, C_{i,j}, d_{i,j})$ .

The response time of  $J_{i,j}$  is denoted by a random variable  $R_{i,j}$  and its probability mass function by  $f_{R_{i,j}}(t)$ . From  $f_{R_{i,j}}(t)$ , the deadline miss probability  $DMP_{i,j}$  of  $J_{i,j}$  is computed as follows:

$$DMP_{i,j} = \mathbf{P}\{R_{i,j} > D_i\} = \sum_{t=D_i+1}^{\infty} f_{R_{i,j}}(t). \quad (1)$$

Our goal is to find a safe upper bound  $DMP_i^{bound}$  of the deadline miss probability for each task  $\tau_i$  which is guaranteed

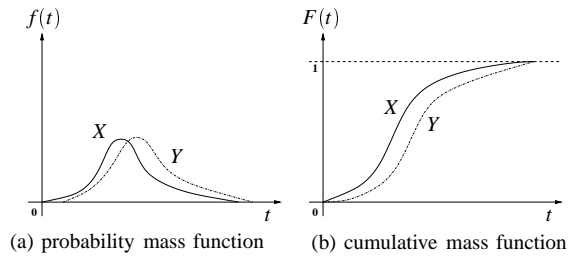


Fig. 1. Bounding distributions

to be larger than or equal to the maximum of  $DMP_{i,j}$ 's of task  $\tau_i$  for all  $j$ . To obtain a safe upper bound of the deadline miss probability of each task, we need to find a response time distribution that is guaranteed to be “worse than” all possible response time distributions of the task. For this, we formally define the *worse-than* relation between two response time distributions as follows, which is a stochastic extension of the *worse-than* relation between two scalar values of response times [5]. Figure 1 depicts two response time distributions, one of which is worse than the other.

**Definition 1.** For two random variables  $X$  and  $Y$  describing response times, if the following inequality holds for any  $s$ , we say that  $Y$  is worse than  $X$  ( $f_y$  upper-bounds  $f_x$ ) and that  $X$  is better than  $Y$  ( $f_x$  lower-bounds  $f_y$ ).

$$\sum_{t=0}^s f_x(t) \geq \sum_{t=0}^s f_y(t) \quad \text{for any } s.$$

The scheduling algorithms we consider are fixed-priority algorithms where tasks' priorities are given in any order, including Rate Monotonic and Deadline Monotonic. We denote the priority of a task  $\tau_i$  by a priority value  $p_i$  that is the same as the task index  $i$ . A smaller priority value (or a smaller task index) means a higher priority— $\tau_1$  has the highest priority,  $\tau_2$  the second highest, and so on. The priority of every job  $J_{i,j}$  belonging to the task  $\tau_i$  is denoted by  $p_{i,j}$  and it is the same as  $p_i$  in the fixed-priority system. Thus, if multiple jobs of the same task are pending, they are served in a FCFS (First Come First Served) manner.

For being self-contained, we list the following definitions that are well known in the real-time literature.

**Definition 2.** A job  $J_{i,j}$  of a task  $\tau_i$  is said to be “in-phase” if its release time is synchronized with that of a job from every higher priority task.

**Definition 3.** A task set  $\{\tau_1, \tau_2, \dots, \tau_n\}$  is said to be “harmonic” if  $T_i$  is an integer multiple of  $T_k$  for any pair of  $(i, k)$  where  $T_i \geq T_k$ .

**Definition 4.** A “hyperperiod” of a task set  $\{\tau_1, \tau_2, \dots, \tau_n\}$  is defined as the least common multiple of the periods of all the tasks in the set. Similarly, a “level- $i$  hyperperiod” is defined as the least common multiple of the periods of  $\tau_1, \tau_2, \dots$ , and  $\tau_i$ .

**Definition 5.** The “system utilization” of a task set  $\{\tau_1, \tau_2, \dots, \tau_n\}$  is defined as the sum of utilizations of all the tasks in the set. The “level- $i$  utilization” is defined as the sum

of utilizations of tasks  $\tau_1, \tau_2, \dots$ , and  $\tau_i$ . Since the execution times of the tasks are variable, the average level- $i$  utilization  $\bar{U}_i$  and the maximum level- $i$  utilization  $U_i^{\max}$  are defined as follows:

$$\bar{U}_i = \sum_{k=1}^i \frac{\bar{C}_k}{T_k} \quad \text{and} \quad U_i^{\max} = \sum_{k=1}^i \frac{C_k^{\max}}{T_k}$$

where  $\bar{C}_k$  and  $C_k^{\max}$  are the average and maximum values of the random variable  $C_k$ .

Table 1 summarizes the above notations that will be used throughout the paper.

notation	description
$\tau_i$	$i$ th priority task
$T_i$	period (i.e. minimum interrelease time) of $\tau_i$ —constant
$D_i$	relative deadline of $\tau_i$ —constant
$C_i$	execution time of $\tau_i$ —random variable with minimum $C_i^{\min}$ , maximum $C_i^{\max}$ , and average $\bar{C}_i$
$\bar{U}_i$	average level- $i$ utilization ( $\bar{U}_i = \sum_{k=1}^i \frac{\bar{C}_k}{T_k}$ )
$U_i^{\max}$	maximum level- $i$ utilization ( $U_i^{\max} = \sum_{k=1}^i \frac{C_k^{\max}}{T_k}$ )
$DMP_{i,j}^{\text{bound}}$	a safe upper bound of deadline miss probability for $\tau_i$
$dmp_{i,j}^{\text{bound}}$	a safe lower bound of deadline meet probability for $\tau_i$
$J_{i,j}$	$j$ th job of $\tau_i$
$\lambda_{i,j}$	release time of job $J_{i,j}$ ( $\lambda_{i,j} \geq \lambda_{i,j-1} + T_i$ )
$d_{i,j}$	absolute deadline of job $J_{i,j}$ ( $d_{i,j} = \lambda_{i,j} + D_i$ )
$C_{i,j}$	execution time of job $J_{i,j}$ —random variable with the same probability distribution as $C_i$
$R_{i,j}$	response time of job $J_{i,j}$ —random variable whose probability distribution needs to be computed
$DMP_{i,j}$	deadline miss probability of $J_{i,j}$
$dmp_{i,j}$	deadline meet probability of $J_{i,j}$

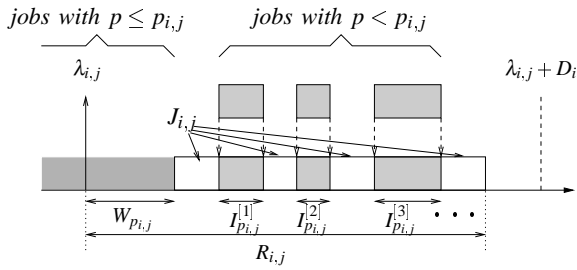
 TABLE 1  
Notations

## 2.2 Exact analysis for constant interrelease times

Since our new stochastic analysis is based on our previous exact stochastic analysis [8], we give its brief overview in this subsection. The exact stochastic analysis has shown that it is possible to compute the exact deadline miss probability for every task if all task phases are known a priori and the interrelease times of each task are constantly equal to its period. Under this assumption, the deadline miss probability of each task  $\tau_i$  can be calculated by averaging the deadline miss probabilities of all the jobs  $J_{i,j}$  from  $\tau_i$  in a steady-state hyperperiod. A steady-state hyperperiod means that in the hyperperiod, the system is in steady state, thus the response time distribution of each job  $J_{i,j}$  is stationary. The stationary response time distribution of  $J_{i,j}$  is defined as follows:

**Definition 6.** Let  $J_{i,j}^{(h)}$  ( $j = 1, 2, \dots, m_i$ ) be the  $j$ th job of task  $\tau_i$  in the  $h$ th level- $i$  hyperperiod and  $f_{R_{i,j}^{(h)}}(t)$  its response time distribution ( $m_i$ : the number of jobs of  $\tau_i$  released in a level- $i$  hyperperiod). Then the stationary response time distribution of  $J_{i,j}$  is defined as the limiting distribution  $f_{R_{i,j}}(t) = \lim_{h \rightarrow \infty} f_{R_{i,j}^{(h)}}(t)$ .

Thus, to compute the response time distribution of each task  $\tau_i$ , the exact analysis tries to determine the response time distributions of all the jobs  $J_{i,j}$  in a hyperperiod  $h$  assuming that  $h \rightarrow \infty$ .


 Fig. 2. Factors affecting the response time of a job  $J_{i,j}$ 

The computation of the stationary response time distributions consists of two steps. The first step is to determine the stationary distribution of the backlog that  $J_{i,j}$  may encounter at the release time  $\lambda_{i,j}$ . The backlog, called  $p_{i,j}$ -backlog, consists of jobs with a priority higher than or equal to that of  $J_{i,j}$ . The second step is to construct the stationary response time distribution of  $J_{i,j}$  from the stationary  $p_{i,j}$ -backlog distribution by considering the execution time distributions of  $J_{i,j}$  and higher priority jobs that may interfere with  $J_{i,j}$ , which are released later than  $J_{i,j}$ . The interference of the higher priority jobs is called  $p_{i,j}$ -interference. In summary, the response time distribution of a job can be determined based on the following equation:

$$R_{i,j} = W_{p_{i,j}} + C_{i,j} + I_{p_{i,j}} \quad (2)$$

where  $W_{p_{i,j}}$  is the  $p_{i,j}$ -backlog observed at time  $\lambda_{i,j}$ ,  $C_{i,j}$  the execution time of  $J_{i,j}$ , and  $I_{p_{i,j}}$  the  $p_{i,j}$ -interferences occurring after time  $\lambda_{i,j}$ . Figure 2 explains the above equation more clearly.

In the exact analysis, the first step to compute the stationary  $p_{i,j}$ -backlog distribution is particularly important. In the case where the maximum level- $i$  utilization  $U_i^{max} > 1$ , the backlog that each job in a hyperperiod encounters at the release time may be affected by jobs from the previous hyperperiods. Thus, the exact analysis models the evolution process of the backlog over hyperperiods as a stochastic process. It has been shown that the stochastic process is a Markov process, and that the stationary  $p_{i,j}$ -backlog distribution can be computed with a set of equilibrium equations derived from a steady-state hyperperiod, as long as the average level- $i$  utilization  $\bar{U}_i$  is less than 1.

Once the stationary  $p_{i,j}$ -backlog distribution is determined for  $J_{i,j}$ , it is easy to construct the stationary response time distribution by using the techniques introduced in the Stochastic Time Demand Analysis [6]. Figure 3 shows an example of the construction assuming that the stationary  $p_j$ -backlog distribution  $f_W(t)$  is given for  $J_j$  and that  $J_{j+1}$  and  $J_{j+2}$  may interfere with  $J_j$  with the same execution time distribution  $f_{C_j}(t) = f_{C_{j+1}}(t) = f_{C_{j+2}}(t)$  as in the top-right box. In this example, first, the stationary response time distribution assuming no interference is computed by convolving the stationary  $p_j$ -backlog distribution  $f_W(t)$  and the execution time distribution  $f_{C_j}(t)$ . This convolution process can be understood as follows: When  $p_j$ -backlog is 1 and  $J_j$ 's execution time is 1, then the response time assuming no interference is 2. Thus, the probability of the response time being 2 is  $1/4$

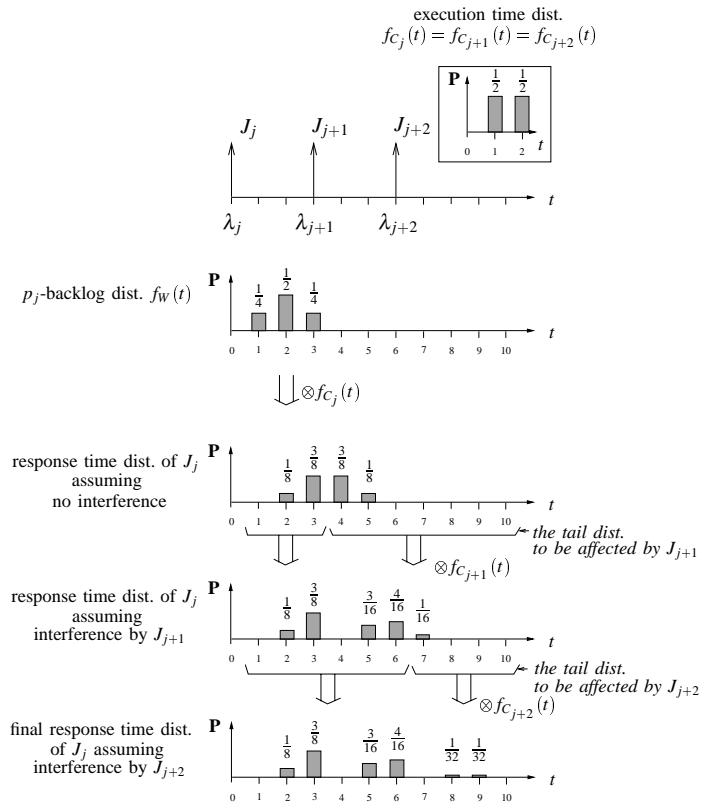


Fig. 3. An example of response time distribution construction

multiplied by  $1/2$ , i.e.,  $1/8$ . Similarly, the response time is 3 when  $p_j$ -backlog is 1 and  $J_j$ 's execution time is 2 or when  $p_j$ -backlog is 2 and  $J_j$ 's execution time is 1. Thus, its probability is  $1/4 \times 1/2 + 1/2 \times 1/2 = 3/8$ . The probabilities of the response time being 4 and 5 can be computed similarly. With the probability that the resulting response time is less than or equal to 3, i.e., the release time of  $J_{j+1}$ ,  $J_j$  can be finished without being affected by  $J_{j+1}$ . However, with the probability that the response time is greater than 3, the response time is further affected by  $J_{j+1}$ . Thus, the stationary response time distribution reflecting the interference from  $J_{j+1}$  is computed by convolving (1) the tail distribution defined in the range  $[3 + 1, \infty)$  of the response time distribution of  $J_j$  calculated so far and (2) the execution time distribution  $f_{C_{j+1}}(t)$ . Finally, the complete stationary response time distribution, reflecting the interference from  $J_{j+2}$ , is computed in the same manner, i.e., by convolving (1) the tail distribution defined in the range  $[6 + 1, \infty)$  of the response time distribution of  $J_j$  calculated so far and (2) the execution time distribution  $f_{C_{j+2}}(t)$ . In this example, if the relative deadline  $D_j$  of  $J_j$  is 7, the deadline miss probability of the job  $J_j$  will be  $\mathbf{P}\{R_j > D_j\} = 1/32 + 1/32 = 1/16$ . For more details on the exact analysis, refer to [8].

### 3 SAFE ANALYSIS FOR ARBITRARY INTERRELEASE TIMES

When the interrelease times of tasks vary, a job  $J_{i,j}$  of task  $\tau_i$  can have arbitrary phase relations with jobs of higher priority

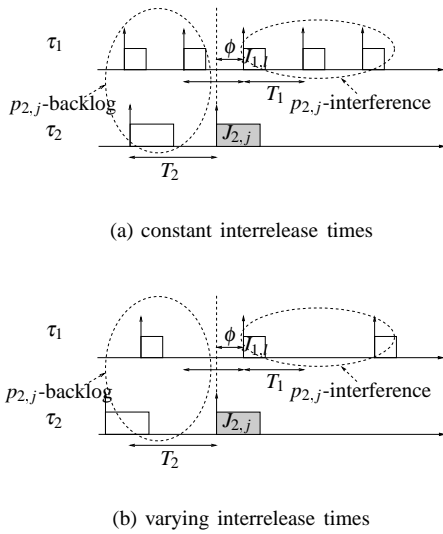


Fig. 4. Effect of constant and varying releases on the response time distribution

tasks. In this case, the exact stochastic analysis cannot find a safe upper bound of the response time distribution of  $\tau_i$  since it considers only a limited set of phase combinations that arise in a level- $i$  hyperperiod assuming fixed initial phases and constant interrelease times of the tasks.

Fortunately, the problem of enumerating all possible job-level phase combinations for varying interrelease times can be simplified to the problem of enumerating all possible task-level initial phase combinations (i.e.,  $\phi_1 = \lambda_{1,1}$ ,  $\phi_2 = \lambda_{2,1}$ ,  $\dots$ ,  $\phi_i = \lambda_{i,1}$ ) while fixing the interrelease times of each task as the minimum, i.e., the period. This is possible due to the following intuitive observation. From the perspective of a single job  $J_{i,j}$ , when its phase relation with other jobs of higher priority tasks is given, the response time distribution of  $J_{i,j}$  assuming the constant interrelease times of periods is worse than all other possible ones obtained for varying interrelease times. This can easily be understood by an example of Figures 4(a) and (b), which show the cases of constant interrelease times and varying interrelease times, respectively, when the job-level phase relation between  $J_{1,i}$  and  $J_{2,j}$  is fixed as  $\phi$ . For the job  $J_{2,j}$ , its response time distribution is determined by  $p_{2,j}$ -backlog and  $p_{2,j}$ -interference as explained in Equation (2). The  $p_{2,j}$ -backlog is maximized when all jobs of  $\tau_1$  and  $\tau_2$  released before  $J_{2,j}$  are maximally packed toward the release time of  $J_{2,j}$  with the minimum interrelease times  $T_1$  and  $T_2$ . Similarly, the  $p_{2,j}$ -interference is maximized when all jobs of  $\tau_1$  released after  $J_{2,j}$  are maximally packed toward the release time of  $J_{2,j}$  with the minimum interrelease time  $T_1$ . This means that without loss of generality, any job-level phase combination observed at a certain time can be represented by a phase combination  $(\phi_1, \phi_2, \dots, \phi_i)$  of the first jobs  $J_{k,1}^{(h)}$  of all the tasks  $\tau_k$  ( $k = 1, \dots, i$ ) in a steady-state hyperperiod  $h$ . Thus, from now on, we focus on finding a safe upper bound of the response time distribution for the first job  $J_{i,1}^{(h)}$  of task  $\tau_i$  in a steady-state hyperperiod  $h$  by enumerating all the

$\tau_i$	$T_i$	$C_i^{\min}$	$C_i^{\max}$	$\bar{U}$	$U^{\max}$
$\tau_1$	70	1	26	0.5079	0.9914
$\tau_2$	100	1	62		

TABLE 2  
A task set example

possible initial phase combinations  $(\phi_1, \phi_2, \dots, \phi_i)$  assuming the constant interrelease times of periods. For the same reason, we describe each task  $\tau_i$  by a tuple  $(T_i, \phi_i, C_i, D_i)$  assuming the constant interrelease time that is equal to the period  $T_i$ . Henceforth, each job  $J_{i,j}$ 's release time  $\lambda_{i,j}$  is given by  $\phi_i + (j-1)T_i$ . Without loss of generality, the phase  $\phi_i$  of each task  $\tau_i$ , i.e., the release time of the first instance of  $\tau_i$ , is assumed to be smaller than  $T_i$  ( $0 \leq \phi_i < T_i$ ).

Although we simplify the problem to finding the worst case initial phase combination  $(\phi_1, \phi_2, \dots, \phi_i)$ , it is not guaranteed that there exists such a phase combination that the resulting response time distribution of task  $\tau_i$  can upper-bound those of  $\tau_i$  obtained with all the other phase combinations. To explain this, let us consider the task set used by Lehoczky in [11]. We add one assumption to the task set that each task has a uniform execution time distribution varying from 1 to the maximum value originally specified in the paper. This task set is shown by Table 2.

The task set consists of two tasks  $\tau_1$  and  $\tau_2$  with uniform execution time distributions [1..26] and [1..62], respectively. Within a level-2 hyperperiod, we encounter seven jobs  $J_{2,1}$ ,  $J_{2,2}$ , ..., and  $J_{2,7}$  of  $\tau_2$  that represent seven different phase combinations. In the deterministic domain, assuming the in-phase condition and the worst case execution times 26 and 62 of  $\tau_1$  and  $\tau_2$ , respectively, the worst response time of  $\tau_2$  can be found by General Time Demand Analysis [11]. That is, in the in-phase level-2 busy interval, which comprises all the seven jobs, their response times are calculated as 114, 102, 116, 104, 118, 106, and 94, respectively. Out of them, the worst response time 118 is found in  $J_{2,5}$ .<sup>1</sup>

In the stochastic domain, however, the worst response time distribution cannot easily be found. Figure 5 compares the response time distributions of  $J_{2,1}$  and  $J_{2,5}$  computed by our previous exact analysis. In Figure 5(a), the response times  $R_{2,1}$  and  $R_{2,5}$  are expressed as cumulative mass functions where the solid one is for  $J_{2,1}$  and the dashed one is for  $J_{2,5}$ . According to Definition 1, if there exists such a time  $s$  that the cumulative probability  $\sum_{t=0}^s f_{R_{2,5}}(t)$  is greater than the cumulative probability  $\sum_{t=0}^s f_{R_{2,1}}(t)$ , then  $R_{2,5}$  cannot upper-bound  $R_{2,1}$  in the stochastic sense. The figure clearly shows such time points in the ranges of [0, 40] and [70, 100]. This means that the deterministic worst case  $J_{2,5}$  is no longer the stochastic worst case. Also  $J_{2,1}$  is not the stochastic worst case either, since there exists a time  $s$  in the range of [115, 120] where the cumulative probability of  $R_{2,1}$  is greater than that of  $R_{2,5}$ . This is not clear in Figure 5(a) but it becomes clear if we see "1 - cumulative probability" in Figure 5(b). In fact, for this task set example, none of the phase combinations represented

1. Remember that  $J_{2,1}$  is not the worst case in general considering the case where the relative deadline is larger than the period.

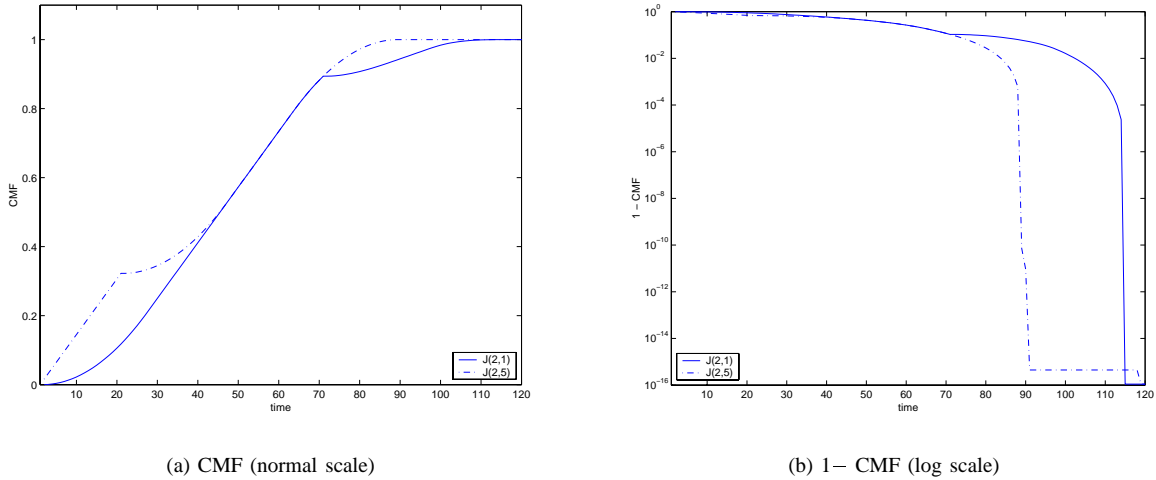


Fig. 5. Comparison between  $J_{2,1}$  and  $J_{2,5}$  of the example task set

by  $J_{2,1}$ ,  $J_{2,2}$ , ..., and  $J_{2,7}$  can give a safe upper bound in the stochastic sense.

Since we could not find the worst case phase combination  $(\phi_1, \phi_2, \dots, \phi_i)$  in general, we take a detour. That is, we first address harmonic task sets. For any harmonic task set, we prove that the response time distribution of  $\tau_i$  obtained assuming the in-phase condition (i.e.,  $\phi_1 = \phi_2 = \dots = \phi_i$ ) safely upper-bounds the response time distributions of  $\tau_i$  obtained for all the other phase combinations. Second, to address general non-harmonic task sets, we transform a given non-harmonic task set  $\{\tau_1, \tau_2, \dots, \tau_i\}$  to a harmonic task set  $\{\tau'_1, \tau'_2, \dots, \tau'_i\}$  such that the response time distribution of  $\tau'_i$  upper-bounds that of  $\tau_i$  with any phase combination. As a result, the response time distribution of  $\tau_i$  with any phase combination in the original task set can be upper-bounded by that of  $\tau'_i$  in the transformed harmonic task set and, in turn, upper-bounded by that of  $\tau'_i$  obtained assuming the in-phase condition.

### 3.1 Harmonic task sets

In a harmonic task set, there exists only one instance  $J_{i,1}$  of task  $\tau_i$  in every level- $i$  hyperperiod and it has the same phase combination with respect to higher priority jobs in every level- $i$  hyperperiod. From this property, in this subsection, we try to prove that the response time distribution of  $J_{i,1}^{(h)}$  in a steady-state hyperperiod  $h$  assuming the in-phase condition, which can be calculated by our previous exact stochastic analysis, upper-bounds those assuming all the other possible phase combinations.

For this proof, we use the concept of a *level- $i$  busy interval*, which was mentioned in Lehoczy's example described above. The formal definition is an interval during which the processor is busy all the time executing jobs of task  $\tau_i$  and higher priority tasks  $\tau_1, \dots, \tau_{i-1}$  and has no backlog of the tasks both at the start and at the end of the interval. As seen in the example, the worst response time of task  $\tau_i$  is found in one of the jobs  $J_{i,j}$  executing in the in-phase level- $i$  busy interval. We extend this theorem of the deterministic domain, which only considers the

case of the worst case execution times, to the case of varying execution times as follows.

**Lemma 1.** *Suppose an in-phase level- $i$  busy interval where each job  $J_{k,l}$  of every task  $\tau_k$  ( $1 \leq k \leq i$ ) has an "arbitrary" execution time  $C_{k,l}$  ( $C_k^{\min} \leq C_{k,l} \leq C_k^{\max}$ ). Then the response time of each job  $J_{i,j}$  from task  $\tau_i$  executed in the in-phase level- $i$  busy interval is never less than that of  $J_{i,j}$  executed in a "not-in-phase" level- $i$  busy interval.*

*Proof:* To understand the effect of task phases on the response time of each job  $J_{i,j}$  of task  $\tau_i$  executed in a level- $i$  busy interval, we formulate the response time of  $J_{i,j}$  as a function of the phase combination  $(\phi_1, \phi_2, \dots, \phi_i)$ . For the sake of brevity, we take the smallest  $\phi_{\min}$  of all the phases of the tasks and according to  $\phi_{\min}$ , adjust all the phase values so that  $\phi_k := \phi_k - \phi_{\min}$  ( $1 \leq k \leq i$ ). Then we take as the time origin the release time of the first job from the task whose phase value becomes 0 due to the adjustment. To formulate the response time of  $J_{i,j}$ , we have to find the completion time  $t_f$  of  $J_{i,j}$ , which should be large enough for serving the processor demand of (1) all the jobs  $J_{i,1}, J_{i,2}, \dots, J_{i,j}$  of task  $\tau_i$  and (2) all the jobs  $J_{k,l}$  of the higher priority tasks  $\tau_k$  ( $1 \leq k \leq i-1$ ) released in the interval  $[0, t_f)$ . So, the completion time  $t_f$  is the smallest value of  $t$  that satisfies the following equation.

$$t = \sum_{l=1}^j C_{i,l} + \sum_{k=1}^{i-1} \left( \sum_{l=1}^{\left\lceil \frac{t - \phi_k}{T_k} \right\rceil} C_{k,l} \right) \quad (3)$$

Since the release time of  $J_{i,j}$  is  $\phi_i + (j-1)T_i$ , the response time  $R_{i,j}$  of  $J_{i,j}$  is formulated as follows:

$$\begin{aligned} R_{i,j}(\phi_1, \phi_2, \dots, \phi_i) &= t_f - \phi_i - (j-1)T_i \\ &= \sum_{l=1}^j C_{i,l} + \sum_{k=1}^{i-1} \left( \sum_{l=1}^{\left\lceil \frac{t_f - \phi_k}{T_k} \right\rceil} C_{k,l} \right) - \phi_i \\ &\quad - (j-1)T_i \end{aligned} \quad (4)$$

In the above equation, the completion time  $t_f$  of  $J_{i,j}$  and all the phase values  $\phi_1, \phi_2, \dots, \phi_i$  are variables while all the other terms are constants, including all the given execution times  $C_{k,l}$ . If  $\phi_k \rightarrow 0$ , the release time of  $J_{i,j}$  is shifted to the starting instant of the level- $i$  busy interval, so the response time  $R_{i,j}$  is increased while the completion time  $t_f$  does not change. If  $\phi_k \rightarrow 0$  ( $1 \leq k \leq i-1$ ), there are chances that a job of  $\tau_k$  originally released after the end of the level- $i$  busy interval would join the interval, thus resulting in an increase in the completion time  $t_f$  by the execution time of the joining job, in turn an increase in the response time  $R_{i,j}$ . From this, we can see that the maximum value of  $R_{i,j}$  is obtained for any given execution times  $C_{k,l}$  of the jobs when  $\phi_i = 0$  and  $\phi_k = 0$  for all  $k = 1, 2, \dots, i-1$ . Thus, the above lemma holds.  $\square$

Using this lemma and the harmonic property that there exists only one job  $J_{i,1}$  of task  $\tau_i$  in every level- $i$  hyperperiod, we can now prove the following theorem.

**Theorem 1.** *For a harmonic task set, the stationary response time distribution of job  $J_{i,1}$  from task  $\tau_i$  obtained for the in-phase combination in a steady-state hyperperiod upper-bounds that of  $J_{i,1}$  obtained for any other phase combination.*

*Proof:* First, consider the case where a level- $i$  busy interval starts at the earliest release time out of the first jobs of  $\tau_1, \tau_2, \dots, \tau_i$  within a level- $i$  hyperperiod  $h$ . According to the definition of a busy interval, there is no backlog at the starting instant of the interval. In this case,  $J_{i,1}$  in an in-phase level- $i$  busy interval has the largest interferences from higher priority jobs and hence the worst response time according to Lemma 1.

Next, consider the case where a level- $i$  busy interval spans over two consecutive level- $i$  hyperperiods, say  $(h-1)$ th level- $i$  hyperperiod and  $h$ th level- $i$  hyperperiod. In this case, since there is a backlog left from the  $(h-1)$ th hyperperiod for  $J_{i,1}^{(h)}$ , the response time of  $J_{i,1}^{(h)}$  is further affected by this backlog. Nevertheless, the in-phase condition gives a worse response time of  $J_{i,1}^{(h)}$  than any other phase combinations for arbitrary execution times. This can be explained by the harmonic property. That is, if the release time of  $J_{i,1}^{(h)}$  is synchronized with those of the first jobs of  $\tau_1, \tau_2, \dots, \tau_i$  (i.e., in-phase) in the  $h$ th level- $i$  hyperperiod,  $J_{i,1}^{(h-1)}$  is as well in the  $(h-1)$ th level- $i$  hyperperiod. Since the in-phase condition in the  $(h-1)$ th level- $i$  hyperperiod gives the worse response time to  $J_{i,1}^{(h-1)}$  than any other phase combinations due to Lemma 1, it also leaves the larger backlog to  $J_{i,1}^{(h)}$  in the  $h$ th hyperperiod than any other phase combinations. Also, the interference by a higher priority task to  $J_{i,1}^{(h)}$  is largest in the in-phase condition according to Lemma 1. Thus, the in-phase condition in the  $(h-1)$ th level- $i$  hyperperiod (in the  $h$ th level- $i$  hyperperiod as well due to the harmonic property) gives a worse response time of  $J_{i,1}^{(h)}$  than any other combinations.

This observation is valid even when extending the starting instant of the level- $i$  busy interval to the  $(h-2)$ th, the  $(h-3)$ th, and finally the first hyperperiod. Thus, when  $h \rightarrow \infty$ , the above theorem holds.  $\square$

### 3.2 Non-harmonic task sets

For a non-harmonic task set, since it is not straightforward to find the worst case phase combination for  $J_{i,1}$  as explained before, we transform the non-harmonic task set  $\{\tau_1, \tau_2, \dots, \tau_i\}$  to a harmonic task set  $\{\tau'_1, \tau'_2, \dots, \tau'_i\}$ . In this transformation, we try to guarantee that the stationary response time distribution of  $J_{i,1}$  in the original task set is upper-bounded by that of  $J'_{i,1}$  in the transformed harmonic task set. For this, we introduce the following conditions that should be ensured when transforming a non-harmonic task set  $\mathbf{T} = \{\tau_k = (T_k, \phi_k, C_k, D_k) \mid 1 \leq k \leq i\}$  to a harmonic task set  $\mathbf{T}' = \{\tau'_k = (T'_k, \phi'_k, C_k, D_k) \mid 1 \leq k \leq i, T'_l | T'_{l+1}, 1 \leq l \leq i-1\}$  (Note that  $a|b$  means that  $a$  evenly divides  $b$ ).

**Condition 1.**  $T'_k \leq T_k$

**Condition 2.**  $\phi'_k = \begin{cases} \phi_k & 0 \leq \phi_k < T'_k \\ \max(T'_k - \chi_k, 0) & T'_k \leq \phi_k < T_k \end{cases}$

where  $\chi_k = T_k - \phi_k$ .

If we make the transformation ensuring the above two conditions, we can prove the following lemma.

**Lemma 2.** *If a non-harmonic task set  $\mathbf{T}$  is transformed to a harmonic task set  $\mathbf{T}'$  with Conditions 1 and 2, the stationary response time distribution of job  $J_{i,1}$  of task  $\tau_i$  in the original task set  $\mathbf{T}$  with the phase combination  $(\phi_1, \phi_2, \dots, \phi_i)$  is upper-bounded by that of  $J'_{i,1}$  of task  $\tau'_i$  in the transformed task set  $\mathbf{T}'$  with the phase combination  $(\phi'_1, \phi'_2, \dots, \phi'_i)$ .*

*Proof:* First, consider for a task  $\tau_k$  the case where  $0 \leq \phi_k < T'_k$  ( $1 \leq k \leq i$ ), and let  $\phi'_k = \phi_k$  according to Condition 2 (See Figure 6(a)). In this case, we can investigate the effect of the transformed harmonic task  $\tau'_k$  on the response time distribution of  $J_{i,1}$ , which is released at the starting instant of a level- $i$  hyperperiod, say  $t_0$  (Note that we always can define the starting instant of a level- $i$  hyperperiod as the release time of  $J_{i,1}$  without loss of generality). Then the effect can be formulated into two separate terms:  $p_{i,1}$ -backlog and  $p_{i,1}$ -interference. We can easily see that the transformation increases the  $p_{i,1}$ -backlog of  $J_{i,1}$  with the increased number of jobs from the higher priority task  $\tau'_k$ , released closer to the release time of  $J_{i,1}$  than in the original task set, during the time interval  $[0, t_0]$ . It also increases the  $p_{i,1}$ -interference of  $J_{i,1}$ , i.e.,  $I_{p_{i,1}}^{[1]}, I_{p_{i,1}}^{[2]}, I_{p_{i,1}}^{[3]}, \dots$  with the increased number of jobs from  $\tau'_k$  released earlier than those in the original task set. Thus, the resulting stationary response time distribution of  $J'_{i,1}$  in the transformed task set  $\mathbf{T}'$  upper-bounds that of  $J_{i,1}$  in the original task set  $\mathbf{T}$ . Note that this still holds even if the period  $T_i$  of task  $\tau_i$  is reduced due to the transformation.

Next, consider for a task  $\tau_k$  the case where  $T'_k \leq \phi_k < T_k$  ( $1 \leq k \leq i$ ), and if  $\chi_k = T_k - \phi_k \leq T'_k$ , let  $\phi'_k = T'_k - \chi_k$  according to Condition 2 (See Figure 6(b)). Similar to the above case, it is trivial to show that the transformed harmonic task  $\tau'_k$  increases both the  $p_{i,1}$ -backlog and the  $p_{i,1}$ -interference of job  $J_{i,1}$ . Thus, the resulting stationary response time distribution of  $J'_{i,1}$  upper-bounds that of  $J_{i,1}$ . If  $\chi_k = T_k - \phi_k > T'_k$ , let  $\phi'_k = 0$  according to Condition 2. Even in this case, it is trivial to show the same property. Therefore, if every transformed harmonic task satisfies Conditions 1 and 2, the resulting stationary response

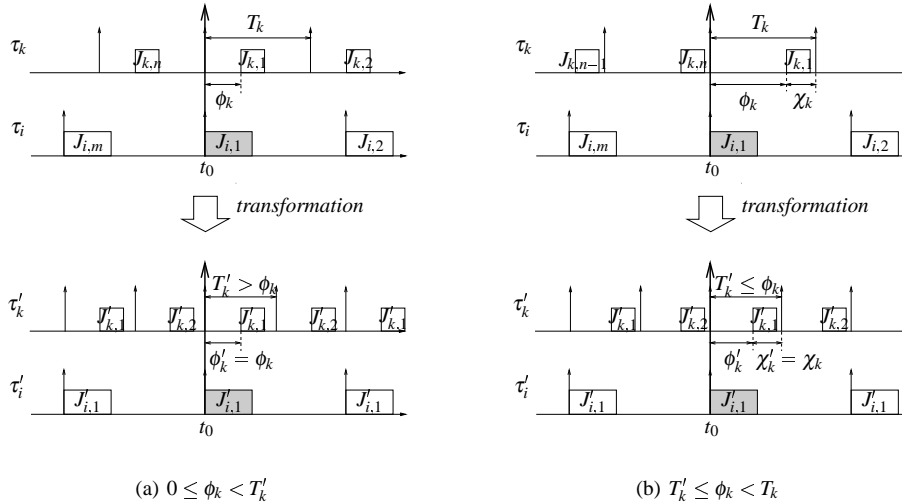


Fig. 6. Examples of harmonic task transformation ( $t_0$  = the starting instant of a hyperperiod)

time distribution of  $J'_{i,1}$  upper-bounds that of  $J_{i,1}$ . Thus, the above lemma holds.  $\square$

Then, we finally can prove the following theorem, which allows us to calculate an upper-bounding response time distribution for non-harmonic tasks with varying interrelease times.

**Theorem 2.** *For a given non-harmonic task set  $\mathbf{T} = \{\tau_1, \tau_2, \dots, \tau_i\}$ , if we transform it to a harmonic task set  $\mathbf{T}' = \{\tau'_1, \tau'_2, \dots, \tau'_i\}$  such that  $T'_k \leq T_k$  for all  $k = 1, 2, \dots, i$ , the stationary response time distribution of  $J'_{i,1}$  obtained assuming the in-phase condition is a safe upper bound for the stationary response time distribution of any job  $J_{i,j}$  from  $\tau_i$  for any possible phase combination of the original tasks  $\tau_1, \tau_2, \dots$ , and  $\tau_i$  with varying interrelease times.*

*Proof:* As mentioned before, any job  $J_{i,j}$  from task  $\tau_i$  (having varying interrelease times) with a phase combination with higher priority tasks (having varying interrelease times) can be represented by the first job  $J_{i,1}$  of task  $\tau_i$  in a level- $i$  hyperperiod of the tasks (having the minimum interrelease times) with an initial phase combination of  $(\phi_1, \phi_2, \dots, \phi_i)$  — See Figure 4. Whatever initial phase combination is given, the stationary response time distribution of  $J_{i,1}$  in the original task set  $\mathbf{T}$  can be upper-bounded by that of  $J'_{i,1}$  in the transformed harmonic task set  $\mathbf{T}'$  by Lemma 2. The latter again is upper-bounded by the stationary response time distribution of  $J'_{i,1}$  obtained assuming the in-phase condition by Theorem 1. Thus, the above theorem holds.  $\square$

Note that there are many ways to transform a non-harmonic task set to a harmonic task set satisfying Condition 1 (In the actual transformation for analysis of a given task set, Condition 2 is not needed, since its purpose was only to prove the above theorem). In this paper, we use the *DCT* algorithm [7] since the resulting increase of the system utilization is known to be smaller than that of the *Sr* [7]. A brief summary of the *DCT* algorithm is given in the following. First, it chooses the period  $T_f$  of a task  $\tau_f$  as a base and transforms the periods  $T_k$  of all the other tasks to  $T'_k$  such that  $T'_k \leq T_k$  for all  $k = 1, \dots, i$ ,

and  $T'_k$  evenly divides  $T'_{k+1}$  for all  $k = 1, \dots, i-1$ . In this transformation, in order to reduce the resulting increase of the system utilization, for each  $k > f$ ,  $T'_k$  is selected as the largest integral multiple of  $T'_{k-1}$  that is less than or equal to  $T_k$ . For each  $k < f$ ,  $T'_k$  is selected as the largest divisor of  $T'_{k+1}$  that is less than or equal to  $T_k$ . To further reduce the utilization increase due to the transformation, in the second step, the algorithm repeats the above transformation for all  $f = 1, \dots, i$  and finally selects  $f$  that gives the smallest utilization increase. It is important to minimize the system utilization increase due to the harmonic task transformation, because we cannot obtain a meaningful response time distribution if the increase is too large.

## 4 EXPERIMENTAL RESULTS

In this section, we give experimental results to evaluate the accuracy of the proposed analysis. For the evaluation of accuracy, we present the deadline “meet” probability. The deadline miss probability is simply  $1 - \text{deadline meet probability}$ .

### 4.1 Accuracy of the proposed analysis

For general evaluation of accuracy of the proposed analysis, we consider randomly generated non-harmonic task sets of  $n$  tasks. For a task set, the period  $T_i$  of each task  $\tau_i$  ( $1 \leq i \leq n$ ) is randomly sampled over an integer array of 20, 30, 40, ..., and 200. The relative deadline  $D_i$  of task  $\tau_i$  is assumed to be equal to the period  $T_i$ .

For each task set generated, we consider the cases where the average system utilizations  $\bar{U}$  are equal to 0.50, 0.55, 0.60, 0.65, 0.70, and 0.75, respectively by determining the execution time distributions of all  $n$  tasks in the set such that the average utilizations of the individual tasks are all equal to  $\bar{U}/n$ . That is, the execution time distribution of each task  $\tau_i$  is a uniform distribution ranging from  $1$  to  $2 \times \frac{\bar{U}}{n} \times T_i - 1$ .

In order to randomly determine the initial phase  $\phi_i$  of each task  $\tau_i$ , we introduce a control factor  $\alpha$  ( $0 \leq \alpha \leq 1$ ). For a



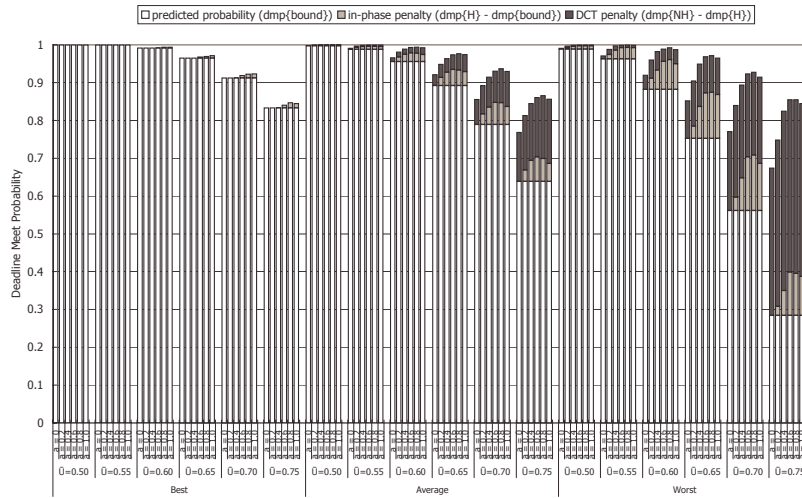


Fig. 7. Effect of  $\bar{U}$  on analysis accuracy ( $n = 3$ )

given  $\alpha$ , the phase  $\phi_i$  of every task  $\tau_i$  in the set is randomly sampled over the range  $[0, \alpha \times T_i]$ . Using  $\alpha$ , we effectively control phase combinations of the first jobs of tasks in the given set. The case where  $\alpha = 0$  leads to in-phase scenarios while the case where  $\alpha = 1$  leads to random phase scenarios.

For such a given task set, we investigate the accuracy of our computed the deadline meet probability  $dmp_n^{bound}$  for the lowest priority task  $\tau_n$ . Ideally, we have to compare the computed  $dmp_n^{bound}$  with the accurate deadline meet probability of  $\tau_n$  assuming varying interrelease times of the tasks  $\tau_1, \tau_2, \dots, \tau_n$ . However, the accurate deadline meet probability depends on the actual release scenario of jobs over the infinite time interval  $[0, \infty)$  and there are an infinite number of different release scenarios where the interrelease times of each task are greater than the period. Thus, as a reference, we consider only one release scenario where the task phases are fixed and the interrelease times are constant as given, that is, the release times of all the jobs are precisely known. For that scenario, the exact stochastic analysis [8] can compute the accurate deadline meet probability denoted by  $dmp_n^{NH}$  ('NH' means non-harmonic). Even if  $dmp_n^{NH}$  is not an accurate probability when the interrelease times are arbitrary, we use  $dmp_n^{NH}$  as a reference to evaluate the accuracy loss by the harmonic task transformation and in-phase assumption of our analysis. In other words, the gap between  $dmp_n^{NH}$  and  $dmp_n^{bound}$  gives us an idea on how much we lose the accuracy because of unknown release times. As another reference, we also present the deadline meet probability of the transformed harmonic task  $\tau_n^H$  denoted by  $dmp_n^H$  assuming the fixed task phases and constant interrelease times as determined by Conditions 1 and 2 for the transformed harmonic task set. This again is not a safe bound for the original non-harmonic task set with arbitrary interrelease times. Nevertheless, from the gap between  $dmp_n^H$  and  $dmp_n^{bound}$ , we can have an idea on how much we lose the accuracy due to the in-phase assumption for the transformed harmonic task set.

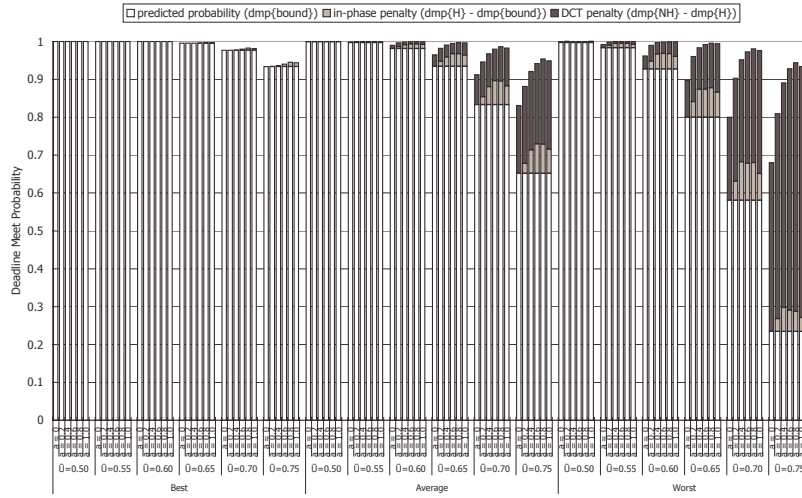
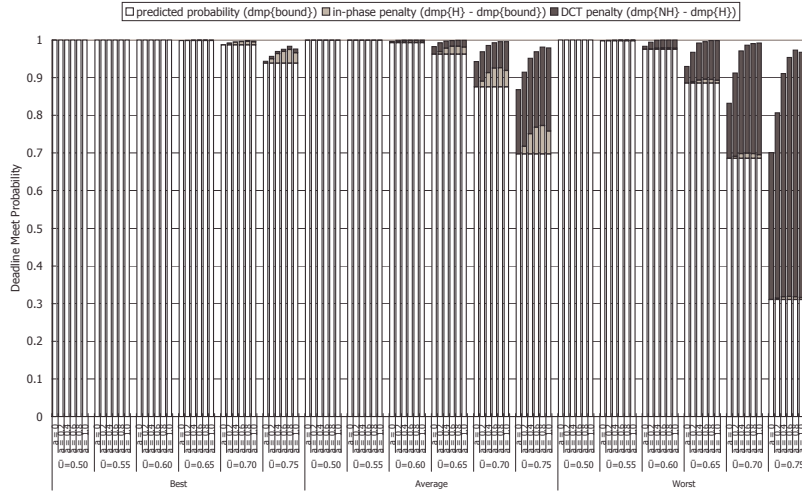
Figure 7 shows such three levels of deadline meet probability of task  $\tau_3$ , while varying the average system utilization  $\bar{U}$  and the control factor  $\alpha$ . For a particular value of  $\bar{U}$ ,

we consider 100 randomly generated task sets. The leftmost column (Best Case) shows the result for the task set (out of the 100 task sets) with the smallest increase of  $\bar{U}$  after DCT transformation. Each bar is the average over 100 different phase combinations randomly generated with a given value of  $\alpha$  for the best case task set. One trivial observation is that all the three levels of deadline meet probability decrease as the  $\bar{U}$  increases, which is natural. In the best case, the original task set happens to be harmonic for all values of  $\bar{U}$  and hence the penalty of DCT transformation is zero, i.e.,  $dmp_3^{NH} = dmp_3^H$ . Also, the penalty due to the in-phase assumption is negligibly small, especially when  $\alpha$  is close to zero, which makes the original phase combination in-phase from the beginning. As a result, our safe bound  $dmp_3^{bound}$  is very close to  $dmp_3^H$ .

The middle column (Average Case) shows the average for all the 100 task sets. Again, for each task set, we first average the results of 100 random phase combinations for a given value of  $\alpha$ . This average is averaged again over the 100 task sets. On average, our safe bound is reasonably tight up to  $\bar{U} = 65\%$ . When  $\bar{U} = 65\%$ , the increase of  $\bar{U}$  is 7.83%, and the difference between our safe lower bound  $dmp_3^{bound}$  and  $dmp_3^{NH}$  observed for various  $\alpha$ 's is less than 10%. For a non-zero  $\alpha$ , the penalty due to the in-phase assumption, i.e.,  $dmp_3^H - dmp_3^{bound}$  is also non-zero and varies, but it is not that significant compared to the penalty due to the DCT transformation, i.e.,  $dmp_3^{NH} - dmp_3^H$ . The penalty due to the DCT transformation becomes more significant as further increasing  $\bar{U}$  to 70% and 75%. Thus, we can conclude that the DCT transformation is the main source of the pessimism of our analysis.

This becomes clearer in the rightmost column (Worst Case) that shows the results for the task set (out of the 100 task sets) with the largest increase of  $\bar{U}$  after DCT transformation. In this worst case, when  $\bar{U} = 75\%$ , the  $\bar{U}$  increase is 19.44%, which makes our bound far from  $dmp_3^{NH}$ .

Figures 8 and 9 show similar graphs when the numbers of tasks, i.e.,  $n$ , are 5 and 7, respectively. Compared with Figure 7, we can see that for the same utilization value  $\bar{U}$  the deadline meet probability  $dmp_n^{NH}$  observed for the lowest priority task  $\tau_n$  tends to increase. This is because while the number of tasks

Fig. 8. Effect of  $\bar{U}$  on analysis accuracy ( $n = 5$ )Fig. 9. Effect of  $\bar{U}$  on analysis accuracy ( $n = 7$ )

$\bar{U}$ (%)	Best						Average						Worst					
	50	55	60	65	70	75	50	55	60	65	70	75	50	55	60	65	70	75
$n = 3$	0.0	0.0	0.0	0.0	0.0	0.0	6.02	6.26	7.24	7.83	8.43	9.05	12.96	14.16	15.56	16.85	18.05	19.44
$n = 5$	0.0	0.0	0.0	0.0	0.0	0.0	8.34	9.14	9.99	10.81	11.62	12.51	14.29	15.58	17.04	18.46	19.84	21.43
$n = 7$	1.67	2.01	2.19	2.40	2.50	2.81	8.99	9.86	10.78	11.70	12.61	13.47	14.32	16.01	17.21	18.75	20.00	21.51

TABLE 3  
Increases of the average system utilization  $\bar{U}$  (unit = %)

increases, the probability that the system becomes overloaded decreases, since the probability decreases that all the tasks in the system have the maximum execution times at the same time.

On the other hand, we can also observe that when the number of tasks increases, the penalty due to the DCT transformation becomes significant. This observation is reasonable, since for tasks with randomly generated periods it is unlikely that they happen to be close to being harmonic when the number of tasks increases. As an example, the best case for  $n = 7$  is not a harmonic task set, while the best cases where  $n = 3$  and  $n = 5$  are harmonic ones in our experiments. In short, the accuracy of our analysis depends on how close the

original task set was to being harmonic, that is, how much  $\bar{U}$  increases after DCT transformation.

Table 3 shows the  $\bar{U}$  increases after DCT transformation for Figures 7, 8, and 9. From the table, we can observe that the pessimism of our analysis has a strong correlation with the  $\bar{U}$  increase due to the DCT transformation.

From the above and other extensive experiments, which are not shown in the paper due to the page limitation, we can see that in the case of low system utilization (i.e.,  $\bar{U} \leq 0.50$ ), the  $\bar{U}$  increase due to the DCT transformation does not matter, since the resulting  $\bar{U}$  is still enough low to cause few deadline misses, thus the safe bound  $dmp_n^{bound}$  approaches to 100%. On the contrary, in the case of high system utilization (i.e.,

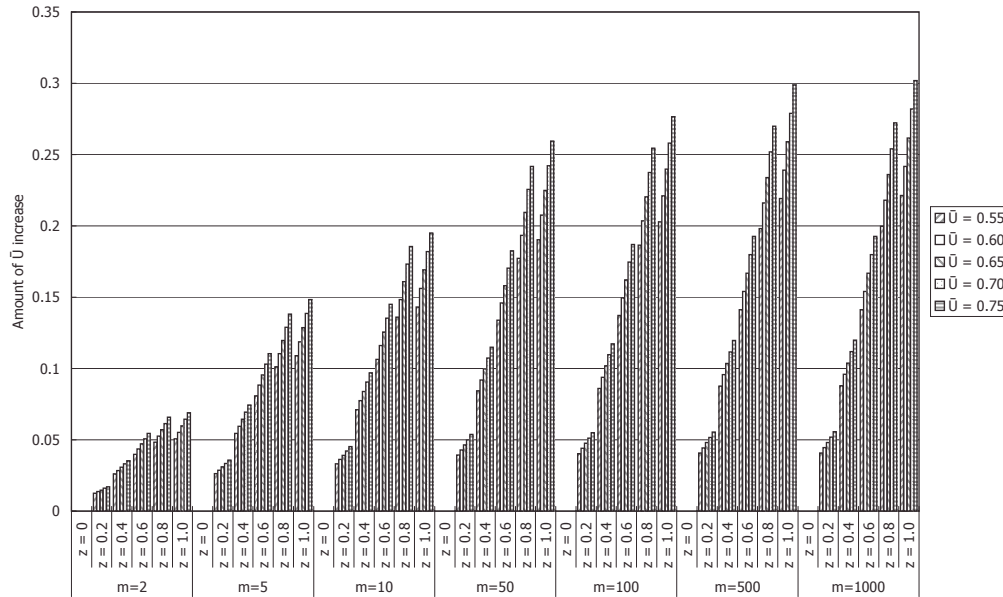


Fig. 10.  $\bar{U}$  increase due to DCT transformation according to  $z$

$\bar{U} \geq 0.75$ ), even a small increase in  $\bar{U}$  matters significantly, since it makes  $dm p_n^{bound}$  far from  $dm p_n^{NH}$ . Our rule of thumb is that when  $\bar{U}$  after DCT transformation is less than 80%, our analysis gives a reasonably tight bound of the deadline meet probability compared to  $dm p_n^{NH}$ .

In the next subsection, we will investigate under which circumstance  $\bar{U}$  after DCT transformation is managed under 80% and hence our analysis can be effectively used.

## 4.2 Effect of the harmonic task transformation

To investigate the effect of the DCT transformation on the system utilization of the transformed task set, we randomly generate non-harmonic task sets over period options. In many practical systems, it is a common practice for each task to choose a period from a limited set of period options rather than selecting arbitrary periods [10]. If such period options happen to be close to being harmonic, the DCT transformation causes only a marginal increase of  $\bar{U}$ . However, if the original period options are far from being harmonic, the transformation causes a significant increase of  $\bar{U}$ .

To see this effect, we control the closeness to being harmonic of the period options using  $z$ . More specifically,  $z$  determines the range of period options as  $[2^{16}..2^{16+z}]$  ( $0 \leq z \leq 1$ ), and over this range, we randomly pick  $m$  period options. When  $z$  is close to 0, the selected period options can easily be transformed to harmonic periods with a little increase of the system utilization (In particular, when  $z = 0$ , all selected period options are harmonic). On the contrary, when  $z$  approaches to 1, the period option set degenerates to truly random numbers, thus it becomes far from being harmonic.<sup>2</sup> The reason is

that for any sampled set of period options over the range of  $[2^{16}..2^{17}]$  to be close to being harmonic, the only chance is that all the sampled period options center around one single value between  $2^{16}$  and  $2^{17}$ , or that some of them are close to  $2^{16}$  while the others are close to  $2^{17}$ , either of which is very unlikely in random sampling when the number of sampled period options is large. For each particular  $z$  value, to see the effect of the number of period options on the system utilization, we vary the number of period options, i.e.,  $m$ , from 2 to 1000.

For generating a non-harmonic task set consisting of  $n$  tasks (usually,  $n \gg m$ ), each task randomly picks one from the  $m$  period options. Note that we do not care  $n$  since all tasks using the same period option can be merged as a single compound task and eventually form a task set with  $m$  compound tasks. For each task set generated, we consider the cases where the average system utilizations  $\bar{U}$  are equal to 0.55, 0.60, 0.65, 0.70, and 0.75, respectively.

Figure 10 shows how much  $\bar{U}$  increases due to the DCT transformation for various pairs of the number of period options, i.e.,  $m$  and harmonicity of period options, i.e.,  $z$ . When  $z$  is close to zero and the number of period options is small, the  $\bar{U}$  increase is only marginal. For example, when  $z = 0.2$  and  $m = 5$ , the  $\bar{U}$  increase is at most 4%. Therefore, for a task set with the original  $\bar{U}$  up to 75%, the increased  $\bar{U}$  can be managed under our rule of thumb number 80% and hence our analysis can be effectively used.

However, as  $z$  increases and  $m$  increases, the  $\bar{U}$  increase becomes significant. Nevertheless, we can make an interesting observation that the  $\bar{U}$  increase converges to a certain value without keeping increasing. Even when  $z = 1$  and  $m = 1000$ , which represents the case of truly arbitrary selection of task periods, the  $\bar{U}$  increase converges at about 30%. To confirm this converging trend, we conducted one more experiment with a period set of truly random numbers, i.e.,  $[2 \times 10^3..2 \times 10^5]$  and the result is given in Figure 11. This figure more clearly

2. This definition of the control factor  $z$  is similar to that of  $\zeta$  introduced in [3], which was defined as  $\max_{1 \leq i \leq n} X_i - \min_{1 \leq i \leq n} X_i$  where  $X_i = \log_2 T_i - \lfloor \log_2 T_i \rfloor$ . To quote the definition of  $\zeta$ , however, our  $z$  value is considered an upper bound of the values of  $\zeta$  of all the task sets generated over the period option range  $[2^k..2^{k+z}]$ , since the minimum and the maximum period value of each task set generated are not necessarily equal to  $2^k$  and  $2^{k+z}$ , respectively.

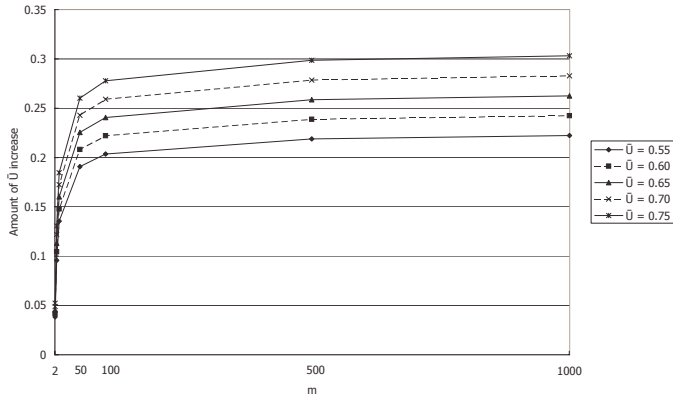


Fig. 11.  $\bar{U}$  increase due to DCT transformation in truly random numbers

shows that the  $\bar{U}$  increase converges to a certain value as the number of period options increases. The converged value is again about 30% when the original  $\bar{U}$  was 75%, which confirms the result of Figure 10. From this result, even for the most generic circumstances where a large number of tasks can use truly arbitrary period values, we can say that the proposed analysis can address non-harmonic task sets with  $\bar{U} < 55\%$  and  $U^{max} > 100\%$ , since the converged value of  $\bar{U}$  increase in the case of  $\bar{U} = 55\%$  is 22% and thus the resulting  $\bar{U}$  after DCT transformation will not exceed 80%.

## 5 THEORETICAL STUDY ON THE ACCURACY LOSS DUE TO HARMONIC TASK TRANSFORMATION

In the previous section, we see that the accuracy of the proposed analysis heavily depends on the original utilization  $\bar{U}$  and its increase  $\Delta\bar{U}$  due to the harmonic task transformation. This section presents a theoretical way to approximately estimate the loss of the deadline meet probability  $\Delta dmp$  as a function of  $\bar{U}$  and  $\Delta\bar{U}$ .

We explain the theoretical estimation focusing on the deadline meet probability  $dmp_i$  of  $\tau_i$ . Thus, we focus on the task  $\tau_i$  itself and its higher priority tasks  $\tau_1, \tau_2, \dots, \tau_{i-1}$ . The original utilization of this task set is denoted by  $\bar{U}_i$ . Our safe stochastic analysis transforms this original task set to a harmonic task set resulting in an utilization increase by  $\Delta\bar{U}_i$ . Our question is how to theoretically quantify the effect of  $\Delta\bar{U}_i$  on the loss of the deadline meet probability  $\Delta dmp_i$ .

The key idea of our theoretical estimation is to use the queueing theory to roughly estimate the deadline meet probability of  $\tau_i$  for two cases: (1) the deadline meet probability for the original utilization, i.e.,  $dmp_i^{theory}(\bar{U}_i)$  and (2) the deadline meet probability for the increased utilization, i.e.,  $dmp_i^{theory}(\bar{U}_i + \Delta\bar{U}_i)$ . The gap between the two, i.e.,  $dmp_i^{theory}(\bar{U}_i) - dmp_i^{theory}(\bar{U}_i + \Delta\bar{U}_i)$  is the accuracy loss due to  $\Delta\bar{U}_i$  increase by the harmonic task transformation.

For this theoretical estimation to be possible, we use two simplifying assumptions: (1) the execution time distribution  $f_{C_k}(t)$  of each individual task  $\tau_k$  is a normal distribution with mean  $\mu_k$  and standard deviation  $\sigma_k$  and (2) the relative deadline

$D_i$  of task  $\tau_i$  is equal to the period  $T_i$ . For the estimation of  $dmp_i^{theory}(\bar{U}_i)$  using the queueing theory, we aggregate  $\tau_i$  itself and its higher priority tasks  $\tau_1, \tau_2, \dots, \tau_{i-1}$  into a single representative task denoted by  $\tau(\bar{U}_i)$  such that its utilization is kept the same as  $\bar{U}_i$ . Specifically, for this aggregation we transform the original task set into a harmonic task set by the same DCT algorithm. However, the mean values  $\mu_k$  of the execution time distributions  $f_{C_k}(t)$  ( $k = 1, 2, \dots, i$ ) are reduced to compensate the period reductions by the DCT and hence we can keep the resulting utilization of the harmonic task set the same as the original one, i.e.,  $\bar{U}_i$ . We denote such reduced execution time distribution by  $f_{C_k}^{reduced}(t)$ . Then, the reduced execution time distributions of all the jobs in a level- $i$  hyperperiod of the harmonic task set are convolved into a single execution time distribution forming an aggregated workload during the level- $i$  hyperperiod, i.e., the transformed period of  $\tau_i$ . This convolution is represented by Equation (5).

$$f_{C_i}^{\bar{U}_i}(t) = (f_{C_1}^{reduced}(t))^{m_1} \otimes (f_{C_2}^{reduced}(t))^{m_2} \otimes \dots \otimes (f_{C_i}^{reduced}(t))^{m_i} \quad (5)$$

where  $m_k$  is the number of jobs of  $\tau_k$  in a level- $i$  hyperperiod and  $(f_{C_k}^{reduced}(t))^{m_k}$  is  $m_k - 1$  times convolution of  $f_{C_k}^{reduced}(t)$  with  $f_{C_k}^{reduced}(t)$ .

As a result, we can finally form a single representative task  $\tau(\bar{U}_i)$  with the period  $T$  equal to the level- $i$  hyperperiod and the execution time distribution  $f_{C_i}^{\bar{U}_i}(t)$  equal to the convolved one. The utilization of this single representative task is equal to the original utilization  $\bar{U}_i$ .

The single representative task  $\tau(\bar{U}_i)$  can be modeled as a D/G/1 queueing system where the interarrival times among jobs from the task are constant  $T$ , the execution time distribution is  $f_{C_i}^{\bar{U}_i}(t)$ , and the jobs are served by a single server. For the D/G/1 queueing system, the waiting time distribution of a job of  $\tau(\bar{U}_i)$  can be calculated using Kingman's theorem [9] as follows:

$$\begin{aligned} f_W^{\bar{U}_i}(t) &= \mathbf{P}[W \leq t] - \mathbf{P}[W \leq t+1] \\ &= e^{-s_0 t} - e^{-s_0(t+1)} \\ &= (1 - e^{-s_0})e^{-s_0 t} \end{aligned} \quad (6)$$

where  $s_0 = \max\{s : s > 0 \text{ and } e^{-sT} \int_{-\infty}^{\infty} e^{st} f_{C_i}^{\bar{U}_i}(t) dt \leq 1\}$ .

Then, the response time distribution of a job of  $\tau(\bar{U}_i)$  can be simply calculated by convolving the above waiting time distribution and its own execution time distribution as follows:

$$\begin{aligned} f_R^{\bar{U}_i}(t) &= f_W^{\bar{U}_i}(t) \otimes f_{C_i}^{\bar{U}_i}(t) \\ &= (1 - e^{-s_0})e^{-s_0 t} \otimes (f_{C_1}^{reduced}(t))^{m_1} \otimes \dots \\ &\quad \otimes (f_{C_i}^{reduced}(t))^{m_i} \end{aligned} \quad (7)$$

With this response time distribution, we can finally find a theoretical estimation of the deadline meet probability as follows:

$$dmp_i^{theory}(\bar{U}_i) = \sum_{t=1}^{D_i} f_R^{\bar{U}_i}(t) \quad (8)$$

The theoretical estimation of  $dmp_i^{theory}(\bar{U}_i + \Delta\bar{U}_i)$  can be similarly done. One difference is that we do not reduce the execution time distribution after the DCT transformation. Thus, only the periods are reduced by the DCT transformation while the execution time distributions are the same as the original ones and hence the resulting utilization of the single representative task, denoted by  $\tau(\bar{U}_i + \Delta\bar{U}_i)$ , becomes  $\bar{U}_i + \Delta\bar{U}_i$ . Applying the same D/G/1 analysis, we can theoretically estimate the deadline meet probability for the case of the increased utilization as follows:

$$dmp_i^{theory}(\bar{U}_i + \Delta\bar{U}_i) = \sum_{t=1}^{D_i} f_R^{\bar{U}_i + \Delta\bar{U}_i}(t) \quad (9)$$

where  $f_R^{\bar{U}_i + \Delta\bar{U}_i}(t)$  is computed by Equations (5), (6), and (7) with the original execution time distributions  $f_{C_k}(t)$  instead of  $f_{C_k^{reduced}}(t)$ .

Finally, we can find a theoretical estimation of the  $dmp_i$  loss as follows:

$$\Delta dmp_i = dmp_i^{theory}(\bar{U}_i) - dmp_i^{theory}(\bar{U}_i + \Delta\bar{U}_i)$$

In order to investigate how closely the theoretical estimation of  $\Delta dmp_i$  reflects the real impact of the  $\bar{U}_i$  increase, i.e.,  $\Delta\bar{U}_i$ , by the DCT transformation of our safe stochastic analysis, we test 100 randomly generated task sets each of which has three tasks. For each task set, we calculate the  $\Delta dmp_3$  using the above theoretical estimation. For the same task set, we also consider one release scenario where the task phases are fixed as in-phase and interrelease times are constants as given. For such a specific scenario where all the release times are precisely known, the exact stochastic analysis can find the accurate deadline meet probabilities for both cases of the original non-harmonic task set and its transformed harmonic task set. We use the gap between the two probabilities calculated for  $J_{3,1}$  and  $J'_{3,1}$  as the accurate reference of  $\Delta dmp_3$ . Figures 12 (a), (b), (c), and (d) compare the theoretical estimations of  $\Delta dmp_3$  with their accurate references for 100 task sets when the original average utilization  $\bar{U}$  is 0.4, 0.5, 0.6, and 0.7, respectively. Interestingly, the theoretical estimation of  $\Delta dmp_3$  is always higher than the accurate reference for the same  $\Delta\bar{U}$ . Also, the theoretical estimation of  $\Delta dmp_3$  increases along with  $\Delta\bar{U}$  in a comparable order of the increase of the accurate reference, especially when  $\bar{U}$  is high, say 0.7. Thus, using the theoretical estimation of  $\Delta dmp$ , we can get a rough (i.e., conservative) idea on the loss of deadline meet probability once the utilization increase  $\Delta\bar{U}$  by the harmonic task transformation is given.

## 6 CONCLUSIONS AND FUTURE WORK

This paper proposes a novel stochastic analysis for fixed-priority scheduling, which provides safe upper bounds of deadline miss probabilities for periodic tasks with (1) arbitrary execution time distributions, (2) varying interrelease times with the period as the minimum, and (3) the maximum utilization factor  $U^{max}$  that can be greater than 1. The proposed analysis is based on the observation that for harmonic task sets the safeness of the in-phase condition can be extended to the

case of  $U^{max} > 1$ . It is proven that in a harmonic task set, the response time distribution of each task obtained assuming that all the tasks are in-phase is the worst among the ones obtained assuming all possible phase combinations. Since this proof is valid only for harmonic task sets, we extended our analysis to a system of non-harmonic tasks by transforming the non-harmonic tasks to harmonic tasks. It is also proven that there always exists a safe transformation such that the response time distributions of the transformed harmonic tasks are worse than those of the original non-harmonic tasks. As a consequence, the response time distribution of each transformed harmonic task assuming in-phase is worse than that of the original non-harmonic task considering all possible phase combinations. Therefore, by comparing the response time distributions of the transformed in-phase harmonic tasks with the given relative deadlines, we can find safe upper bounds of deadline miss probabilities of the original non-harmonic tasks. Through experiments, it is shown that the in-phase condition brings a tight upper bound on the deadline miss probability for harmonic tasks, and that the harmonic task transformation for non-harmonic tasks does not introduce a significant overhead to the tightness when the resulting increase in the average system utilization is small.

One possible direction for future research is to directly address non-harmonic tasks without taking the detour of the harmonic task transformation, which is the major source of the pessimism of our analysis. Another research direction is to investigate the schedulable utilization bound based on the proposed analysis, which makes a stochastic schedulability check possible without massive computation of response time distributions.

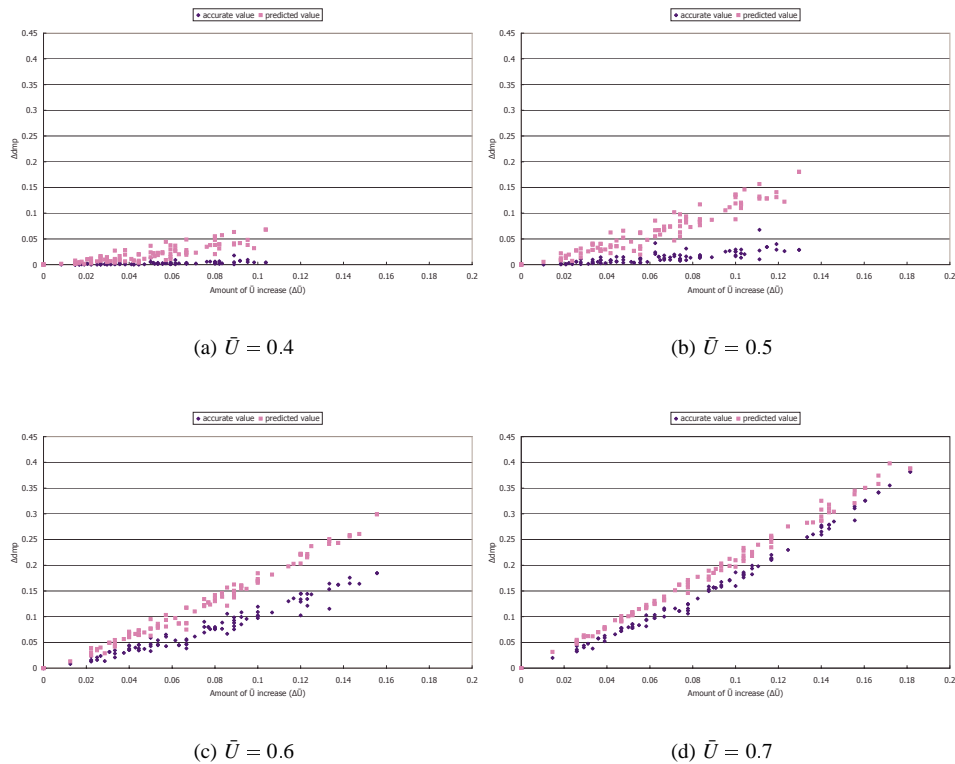
## ACKNOWLEDGMENTS

The authors of this paper are pleased to thank Prof. José Luis Díaz and Prof. José María López for the valuable comments and observations regarding the example task set given in Section 3. This work is supported by Seoul R&BD Program (NT070124). The corresponding author is Chang-Gun Lee.

## REFERENCES

- [1] L. Abeni and G. Buttazzo. Stochastic Analysis of a Reservation Based System. In *Proc. of the 9th International Workshop on Parallel and Distributed Real-Time Systems*, Apr. 2001.
- [2] G. Bernat, A. Colin, and S. Petters. WCET Analysis of Probabilistic Hard Real-Time Systems. In *Proc. of the 23rd IEEE Real-Time Systems Symposium*, Dec. 2002.
- [3] A. Burchard, J. Liebeherr, Y. Oh, and S. H. Son. New Strategies for Assigning Real-Time Tasks to Multiprocessor Systems. *IEEE Transactions on Computers*, 44(12):1429–1442, Dec. 1996.
- [4] A. Cervin. *Integrated Control and Real-Time Scheduling*. PhD thesis, Lund Institute of Technology, Lund, Sweden, 2003.
- [5] J. L. Díaz, J. M. López, M. García, A. M. Campos, K. Kim, and L. LoBello. Pessimism in the Stochastic Analysis of Real-Time Systems: Concept and Applications. In *Proc. of the 25th Real-Time Systems Symposium*, Dec. 2004.
- [6] M. K. Gardner. *Probabilistic Analysis and Scheduling of Critical Soft Real-Time Systems*. PhD thesis, School of Computer Science, University of Illinois, Urbana-Champaign, 1999.
- [7] C.-C. Han and H.-Y. Tyan. A Better Polynomial-Time Schedulability Test for Real-Time Fixed-Priority Scheduling Algorithms. In *Proc. of the 18th Real-Time Systems Symposium*, Dec. 1997.



Fig. 12.  $\Delta\bar{U}$  vs.  $\Delta dmp$ 

- [8] K. Kim, J. L. Díaz, L. LoBello, J. M. López, C.-G. Lee, and S. L. Min. An Exact Stochastic Analysis of Priority-Driven Periodic Real-Time Systems and Its Approximations. *IEEE Transactions on Computers*, 54(11):1460–1466, Nov. 2005.
- [9] J. F. C. Kingman. Inequalities in the Theory of Queues. *Journal of the Royal Statistical Society, Series B*, 32, pages 102–110, 1970.
- [10] C.-G. Lee, L. Sha, and A. Peddi. Enhanced Utilization Bounds for QoS Management. *IEEE Transactions on Computers*, 53(2):187–200, 2004.
- [11] J. P. Lehoczky. Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines. In *Proc. of the 11th IEEE Real-Time Systems Symposium*, pages 201–209, Dec. 1990.
- [12] J. P. Lehoczky. Real-Time Queueing Theory. In *Proc. of the 17th IEEE Real-Time Systems Symposium*, pages 186–195, Dec. 1996.
- [13] J. P. Lehoczky. Real-Time Queueing Network Theory. In *Proc. of the 18th IEEE Real-Time Systems Symposium*, pages 58–67, Dec. 1997.
- [14] J. P. Lehoczky, L. Sha, and Y. Ding. The Rate-Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behavior. In *Proc. of the 10th IEEE Real-Time Systems Symposium*, Dec. 1989.
- [15] A. Leulsegged and N. Nissanke. Probabilistic Analysis of Multiprocessor Scheduling of Tasks with Uncertain Parameter. In *Proc. of the 9th International Conference on Real-Time and Embedded Computing Systems and Applications*, Feb. 2003.
- [16] J. Leung and J. Whitehead. On the Complexity of Fixed Priority Scheduling of Periodic Real-Time Tasks. *Performance Evaluation*, 2(4):237–250, 1982.
- [17] L. Liu and J. Layland. Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment. *Journal of ACM*, 20(1):46–61, 1973.
- [18] S. Manolache, P. Eles, and Z. Peng. Memory and Time-Efficient Schedulability Analysis of Task Sets with Stochastic Execution Times. In *Proc. of the 13th Euromicro Conference on Real-Time Systems*, pages 19–26, Jun. 2001.
- [19] M.-Y. Nam, C.-G. Lee, K. Kim, and M. Caccamo. Time-Parameterized Sensing Task Model for Real-Time Tracking. In *Proc. of the Real-Time Systems Symposium*, pages 245–255, Miami, Florida, Dec. 2005.
- [20] K. M. Obenland. POSIX in Real-Time, 2001. [http://www.xtrj.org/collection/posix\\_rtos.htm](http://www.xtrj.org/collection/posix_rtos.htm).
- [21] A. Terrasa and G. Bernat. Extracting Temporal Properties from Real-Time Systems by Automatic Tracing Analysis. In *Proc. of the 9th International Conference on Real-Time and Embedded Computing Systems and Applications*, Feb. 2003.
- [22] T.-S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.-C. Wu, and J.-S. Liu. Probabilistic Performance Guarantee for Real-Time Tasks with Varying Computation Times. In *Proc. of the Real-Time Technology and Applications Symposium*, pages 164–173, Chicago, Illinois, May 1995.
- [23] W. Yuan and K. Nahrstedt. Energy-Efficient Soft Real-Time CPU Scheduling for Mobile Multimedia Systems. In *Proc. of the ACM Symposium on Operating Systems Principles*, pages 149–163, Bolton Landing, New York, Oct. 2003.

**Kanghee Kim** Kanghee Kim received the BS, MS, and PhD degrees in Computer Engineering from Seoul National University, Korea, in 1996, 1998, and 2004, respectively. He is currently a Senior Engineer in the Mobile Communication Division, Samsung Electronics Co., Ltd. His current research interests include real-time and embedded systems, operating systems, and mobile platforms.

**Chang-Gun Lee** Chang-Gun Lee received the BS, MS and Ph.D. degrees in Computer Engineering from Seoul National University, Korea, in 1991, 1993 and 1998, respectively. He is currently an Assistant Professor in the School of Computer Science and Engineering, Seoul National University, Korea. Previously, he was an Assistant Professor in the Department of Electrical and Computer Engineering, The Ohio State University, Columbus from 2002 to 2006, a Research Scientist in the Department of Computer Science, University of Illinois at Urbana-Champaign from 2000 to 2002, and a Research Engineer in the Advanced Telecomm. Research Lab., LG Information and Communications, Ltd. from 1998 to 2000. His current research interests include real-time systems, complex embedded systems, ubiquitous systems, QoS management, wireless ad-hoc networks, and flash memory systems. Dr. Lee is a member of the IEEE and the IEEE Computer Society.