国 防 科 技 大 学 学 报
第22卷 第3期 JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY Vol. 22 No. 3 2000

文章编号：1001-2486（2000）03-0090-06

随机弱实时系统作业调度概率保证分析*

彭良智，刘宗仕，戴金海

（国防科技大学航天与材料工程学院，湖南 长沙 410073）

摘 要：随机弱实时系统与强实时系统的显著区别是，它不要求实时任务的每一项作业都满足时限要求，只要保证作业的时限延误率低于某一阈值，其总体执行性能就是可以接受的。提出了概率时间需求分析法（PTDA），用于估计可剥夺静态优先级调度策略下周期性任务的作业满足时限约束的概率的下界，并通过一个具体的实例考察了下界的紧性。对实例系统的仿真结果表明，PTDA分析的误差小于10%，计算速度快，可以为弱实时应用的设计提供重要的参考依据。

关键词：随机弱实时系统；PTDA分析；下界估计

中图分类号：TP391.9 文献标识码：A

Probabilistic Scheduling Guarantee Analysis for Jobs in Stochastic Soft Real-Time Systems

PENG Liang-zhi, LIU Zhong-shi, DAI Jin-hai

(College of Aerospace and Material Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Stochastic soft real-time systems distinguish themselves from hard real-time systems in that there is no demand for every job in a task to meet its deadline. For these systems, provided that job deadline missing rate is below some threshold, total performance is acceptable. For a fixed-priority preemptive scheduled soft real-time system, we present probabilistic time demand analysis technique by which a lower bound on the probability that a job in a periodic task meets its deadline is estimated. We have also performed a simulation study of an example system and check the tightness of the lower bound. The simulation results shows PTDA has a high speed and an error below 10%. It can provide important reference for the design of soft real-time application.

Key words: stochastic soft real-time systems; PTDA analysis; lower bound estimation

实时系统是能够在确定的时间内执行计算或处理事务并对外部的异步事件做出响应的计算机系统[1]。它不但要保证任务计算结果的逻辑正确性，而且要在规定的时间内完成。这个规定的时间界限称为任务时限（Deadline），是任务从发布到完成所允许的最大时间间隔。如果任务没有在规定时限内完成，计算结果毫无意义，甚至引起严重后果，称之为强时限，相应的任务为强实时任务，至少包含一个强实时任务的实时系统称为强实时系统。如果任务未在规定时限内完成，计算结果的可用性逐渐减小，称之为弱时限，相应的任务为弱实时任务，所有任务均为弱实时任务的实时系统称为弱实时系统，我们称各任务对处理机资源的需求随机的弱实时系统为随机弱实时系统。这类系统在远程多媒体通信、信号处理、工业控制等领域广泛存在，最典型的如VBR视频传输，对MPEG视频流解码这一任务而言，I帧的解码时间明显长于P帧或B帧。随机弱实时系统与强实时系统的显著区别是，它不要求实时任务的每一项作业都满足时限要求，只要保证作业的时限延误率低于某一阈值，其总体执行性能就是可以接受的。

对实时系统的可调度性分析主要有确定性分析和统计性分析两种。前者如Liu和Layland提出的判断单机系统周期任务基于静态优先级和动态优先级的最优调度算法的可调度性的充分但非必要条件[2]，Lehoczky提出的用于精确分析任务最坏时间行为的时间需求分析法（Time Demand Analysis, TDA）[3]和忙周期分析法（Busy Period Analysis, BPA）[4]，以及Audsley等人提出的具有任意时限和起始抖动的突发性实时任务的可调度性分析算法[5]。后者如Tia等人提出的性能概率保证（Probabilistic

* 收稿日期：2000-01-03
作者简介：彭良智（1973-），男，博士生。

Performance Guarantee，PPG）分析[6]。PPG 分析假设所有任务的相对时限不超过周期，认定任务第一项作业满足时限的概率就是整个任务时限满足率的下界，这些假设在系统的平均资源利用率接近 1.0，最大资源利用率大于 1.0 时是不成立的。

1 问题的提出

假设随机弱实时系统由 n 个周期任务 T_1，T_2，…，T_n 组成，每个任务由五元组 T_i（P_i，D_i，R_i，I_i，$f_i(t)$）表征，P_i，D_i，R_i，I_i 分别表示任务的周期、相对时限、就绪时间、初始相位，$f_i(t)$是作业处理需求所对应的概率密度函数（Probabilistic Density Function，以下简称 PDF）。任务 T_i 的第j 个作业$J_{i,j}$的处理需求为$E_{i,j}$，它是从统计分布$f_i(t)$得到的随机样本。对给定的 i，$E_{i,j}$，$j=1$，2，…独立同分布，不同任务的作业执行时间之间也是相互独立的。$J_{i,j}$在$r_{i,j}$时刻就绪，必须在绝对时限 $d_{i,j}$之前完成，否则称之为延误时限。根据应用场合的不同，延误时限会造成不同程度的损失。就绪时间和绝对时限之间的时间间隔为任务的相对时限，即 $D_i=d_{i,j}-r_{i,j}$。跟踪实际的任务调度执行过程，记 $J_{i,j}$ 的完成时间为$C_{i,j}$，相应的作业响应时间为 $\rho_{i,j}=C_{i,j}-r_{i,j}$。

假定该实时系统采用可剥夺的静态优先级调度策略，记任务 T_i 的优先级为 ϕ_i，T_i 的作业流$J_{i,1}$，$J_{i,2}$，…的优先级也相应为 ϕ_i。为了讨论的方便，并不失一般性，本文假定任务的优先级各不相同，并按优先级降低的次序排列所有任务，对 $\forall i$，j，如果 $i<j$，则 T_i 的优先级高于 T_j 的优先级。E_i^-（E_i^+）表示任务 T_i 的所有作业中最小（最大）的处理需求，E_i 表示T_i 的平均处理需求。系统处理机利用率总和的平均值和最大值分别为 $U_{ave}=\sum_{i=1}^{n}E_i/P_i$ 和 $U_{max}=\sum_{i=1}^{n}E_i^+/P_i$。在单处理机（Uniprocessor）条件下，假设 $U_{max}>1.0$，$U_{ave}<1.0$，由经典的实时系统调度理论可知，即使 U_{ave}很小，平均情况下满足系统可调度性的充分条件，但由于 $U_{max}>1.0$，可能出现瞬间过载，我们无法保证系统总是可调度的，因此有必要估计出各任务的作业满足时限的概率的下界，据此判断是否满足具体的实时应用的要求。

2 概率时间需求分析

2.1 PTDA 的简单情形—任务间相互独立

定义 1 ϕ_i-级忙周期是一个时间区间$[a, b]$，在该区间内，处理机一直无空闲地处理优先级高于等于 ϕ_i 的作业。对$\forall \varepsilon>0$，在时间区间（$a-\varepsilon$，a）和（b，$b+\varepsilon$）上不处理任何优先级高于等于 ϕ_i 的作业。

定义 2 随机变量的条件映射 h：$x\rightarrow y$，记作 $y=h$（x|对 x 的约束条件）。若 x 对应的 PDF 为 $f_X(x)$，则随机变量 y 对应的 PDF 为：

$$f_Y(y)=\begin{cases} f_X(y)/A & y\in E \\ 0 & y\notin E \end{cases} \tag{1}$$

其中 E 是满足指定条件的x 的集合，$A=\int_{x\in E}f_X(x)\,\mathrm{d}x$。

为了确定作业时限延误率的上界，必须找出作业响应时间的最坏情形。由经典实时调度理论可知，任务 T_i 的所有作业的最长响应时间发生在ϕ_i-级忙周期内，这个忙周期起始于临界时刻，即所有任务的初始相位相同，它们的第一个作业同时发布并就绪。下面基于这种最坏情形进行讨论。

考虑任务 T_i 的执行。假设 $J_{i,j}$是ϕ_i-级忙周期中 T_i 发布的第j 个作业。为了简化讨论且不失一般性，不妨把这个 ϕ_i-级忙周期的起点当作时间原点。$J_{i,j}$ 的响应时间是所有能在（$r_{i,j}$，$C_{i,j}$）时间段运行的作业的执行时间的函数，为一随机变量。由于一个作业可能在其后续的作业发布之前仍未完成，故需考虑 ϕ_i-级忙周期中的所有作业。

ϕ_i-级忙周期的长度是一个随机变量，确定 ϕ_i-级忙周期何时结束是 PTDA 分析的关键。在每个优先级对应一个任务的情况下，如果任务 T_i 中的某个作业$J_{i,j}$在其后续作业$J_{i,j+1}$发布之前已经完成，

则 ϕ_i- 级忙周期终止于 $J_{i,j}$完成的时刻。若多个任务具有相同的优先级，则根据这些任务的发布时间先后计算其响应时间分布。上述讨论可归纳为：

定理1 随机弱实时系统周期任务集 T_1，T_2，…，T_n 按优先级降序排列，每个优先级对应一个任务，采用可剥夺的静态优先级调度算法。如果 $\exists j>0$，使得 $P\{w_{i,j}(r_{i,j+1})\leqslant r_{i,j+1}\}=1.0$，则 ϕ_i- 级忙周期在 $J_{i,j}$完成的时刻结束。

推论1 在定理1的前提但存在多个任务具有相同的优先级的情况下，ϕ_i- 级忙周期在 t 时刻结束，当且仅当：t 时刻之前发布的所有优先级高于等于 ϕ_i 的作业在 t 之前已经完成的概率为1.0。

下面计算 ϕ_i- 级忙周期内任务 T_i 的作业响应时间的概率分布。令 $w_{i,j}(t)$ 是需在时间区间 $(r_{i,j}, t]$ 上执行的所有作业的处理时间需求，$p_{i,j}(t)=P\{w_{i,j}(t)\leqslant t\}$ 表示 ϕ_i- 级忙周期尚未结束时处理机时间需求在 t 之前已被满足，$J_{i,j}$已经完成的概率。$p_{i,j}(t)$ 同时也是 $J_{i,j}$的响应时间不超过 t 的概率，因此 $J_{i,j}$满足时限要求的概率至少为 $p_{i,j}(d_{i,j})$。响应时间分布 $p_{i,j}(t)$ 与 $J_{i,j}$ 发布时是否存在优先级高于等于 ϕ_i 的作业积压有关。如果不存在这样的作业积压，ϕ_i- 级忙周期从 $J_{i,j}$发布的时刻开始，不妨将 $J_{i,j}$重新编号为 $J_{i,1}$，而且：

$$p_{i,1}(t)=P\{w_{i,1}(t)\leqslant t\} \tag{2}$$

否则 T_i 中后续作业的响应时间分布按照发布的先后顺序计算：

$$p_{i,j}(t)=P\{w_{i,j}(t)\leqslant t \mid w_{i,j-1}(r_{j,j})>r_{i,j}\} \tag{3}$$

对优先级最高的任务而言，ϕ_i- 级忙周期中第一个作业的响应时间分布等同于作业执行时间的分布，而其后续作业的响应时间分布可以通过计算条件积压作业的分布与作业执行时间的卷积得到。计算过程一直到 ϕ_i- 级忙周期结束。

具体计算 $w_{i,j}(t)$，$j>1$ 时，需综合考虑各种因素的影响。优先级高于等于 ϕ_i 的作业显然可以在 $(r_{i,j}, C_{i,j}]$ 区间执行；作业 $J_{i,1}$，$J_{i,2}$，…，$J_{i,j-1}$中在 $J_{i,j}$发布之后完成的也可在该区间执行；还必须考虑 $(r_{i,j}, C_{i,j}]$ 期间发布的优先级高于 ϕ_i 的作业。按优先级高于 ϕ_i 的作业发布的顺序，把 $(r_{i,j}, C_{i,j}]$ 分割成一系列的子区间，即：

$$(r_{i,j}, C_{i,j}]=(r_{i,j}, r_{k_1,l_1})\cup(r_{k_1,l_1}, r_{k_2,l_2})\cup\cdots\cup(r_{k_{n-1},l_{n-1}}, r_{k_n,l_n}] \tag{4}$$

其中 r_{k_i,l_i}，$i=1, 2, \ldots, n-1$ 依次递增，对应高优先级作业发布的时刻，$r_{k_n,l_n}=C_{i,j}$。若 $J_{i,j}$在时刻 $t\in(r_{i,j}, r_{k_1,l_1})$ 之前完成，对处理机总的时间需求为：

$$w_{i,j}(t)=E_{i,j}+B_{i,j-1}(r_{i,j}) \tag{5}$$

其中 $B_{i,j-1}(r_{i,j})$ 表示作业 $J_{i,j-1}$截至 $r_{i,j}$时刻尚未完成时的条件作业积压分布，即：

$$B_{i,j-1}(r_{i,j})=h(w_{i,j-1}(t)\mid t>r_{i,j}) \tag{6}$$

此时 $J_{i,j}$的响应时间分布为：

$$p_{i,j}(t)=P\{w_{i,j}(t)\leqslant t\mid w_{i,j-1}(r_{i,j})>r_{i,j}\} \tag{7}$$

若 $J_{i,j}$在时刻 $t\in(r_{k_i,l_i}, r_{k_{i+1},l_{i+1}}]$，$i=1, 2, \ldots, n-1$ 之前完成：

$$B_{i,j}(r_{k_i,l_i})=h(w_{i,j}(t)\mid t>r_{k_i,l_i}) \tag{8}$$

相应的时间需求和响应时间分布为：

$$w_{i,j}(t)=E_{k_i,l_i}+B_{i,j}(r_{k_i,l_i}) \tag{9}$$

$$p_{i,j}(t)=P\{w_{i,j}(t)\leqslant t\mid w_{i,j-1}(r_{i,j})>r_{i,j}, w_{i,j}(r_{k_i,l_i})>r_{k_i,l_i}\}\cdot P\{w_{i,j}(r_{k_i,l_i})>r_{k_i,l_i}\} \tag{10}$$

$p_{i,j}(t)$ 的表达式中乘积的后一项是对应该子区间的概率折合因子。综合 $w_{i,j}(t)$ 在各子区间的分布，得到作业 $J_{i,j}$在其时限之前完成的概率：

$$p=\sum_{i=1}^{M}p_{i,j}(r_{k_i,l_i}) \tag{11}$$

其中 $M=\max\{i\mid r_{k_i,l_i}\leqslant d_{i,j}\}$。对 ϕ_i- 级忙周期中任务 T_i 的所有作业，用前述方法计算它们满足时限的下限概率，所有这些概率的最小值即为任务 T_i 的作业满足时限的概率的下界。

2.2 PTDA 的复杂情形—对资源同步的处理

实际应用中的实时系统的任务之间一般非独立，例如，由于资源共享，任务之间存在间接的相互制约关系。资源同步协议的好坏直接影响到系统的可预测性和可调度性。以下的讨论基于临界区不剥夺（Non-Preemptable Critical Section，NPCS）协议。根据 NPCS 协议，如果某作业正处于临界区访问共享资源，其他作业不能剥夺它的执行，直到该作业退出临界区为止。Mok 在文［7］中已证明，作业的最大阻塞时间是所有能够阻塞它的低优先级作业中最长的临界区持续时间。

定理 2 ϕ_i－级忙周期中只有第一个作业可能被阻塞。

证明 当没有阻塞时，只有优先级高于等于 ϕ_i 的作业能在 ϕ_i－级忙周期内执行，因此对优先级低于 ϕ_i 的作业，除 ϕ_i－级忙周期开始的时刻外，它不可能进入临界区，因此只有 ϕ_i－级忙周期中的第一个作业可能被阻塞。

注意到作业临界区持续时间为一随机变量。令 $B_{i,j}^k$ 为作业 $J_{i,j}$ 单独执行时，它的第 k 个临界区的持续时间，b_i^{k+} 为随机变量 $B_{i,j}^k$ 的最大值，b_i^+ 为任务 T_i 的所有 k 个临界区的持续时间的最大值，即 $b_i^+=\max\limits_{1\leqslant m\leqslant k} b_i^{m+}$。由定理 2 可知，只需考虑阻塞时间对 ϕ_i－级忙周期中第一个作业的处理时间需求 $w'_{i,1}(t)$ 的影响，即：

$$w'_{i,1}(t)=w_{i,1}(t)+\max_{i\leqslant j\leqslant n} b_j^+ \tag{12}$$

其中 $w_{i,1}(t)$ 是任务相互独立时 $J_{i,1}$ 的处理时间需求。式（12）考虑了共享资源时任务 T_i 被阻塞的最坏情形。通常情况下，为提高系统的执行性能和响应速度，临界区的持续时间都很短，相互之间的差异也不大，因此由式（12）计算出来的处理时间需求不至于过分保守。

我们还可更准确地估计出任务阻塞时间的概率分布。假设系统由 n 个任务组成，每个任务包含一个临界区，临界区持续时间的概率分布已知。任务 T_i 的作业被来自任务 T_j 的作业阻塞而产生的延迟时间分布为 $P\{B_j^1(x)\leqslant x\}$，因此任务 T_i 的作业总的延迟时间分布为：

$$P\{B^1_{\{i+1,\ldots,n\}}(x)\leqslant x\}=\sum_{l=i+1}^{n}\partial_{i,l}P\{B_l^1(x)\leqslant x\} \tag{13}$$

其中 $\partial_{i,l}$ 是任务 T_i 中的作业被 T_l，$l=i+1,\ i+2,\ \ldots,\ n$ 的作业阻塞的概率，通常根据验前信息或仿真实验确定。如果任务对应的执行代码是直线型的，没有循环和条件分支，各任务的初始相位任意，则 $\partial_{i,l}$ 与作业发布的频度成正比，可由下式估计：

$$\partial_{i,l}=\lim_{t\to\infty}\frac{t}{pl}\Big/\sum_{i<l\leqslant n}\frac{t}{pl} \tag{14}$$

如果任务 T_l 有 m_l 个临界区，则：

$$P\{B_{\{i+1,\ldots,n\}}(x)\leqslant x\}=\sum_{l=i+1}^{n}\sum_{k=1}^{m_l}\partial_{i,l}^kP\{B_l^k(x)\leqslant x\} \tag{15}$$

式中 $\partial_{i,l}^k$ 是任务 T_i 的作业被 T_l 的作业的第 k 个临界区阻塞的概率。

3 PTDA 分析的数值实现及计算复杂度分析

当作业执行时间的概率分布无法用解析式表达时，必须用数值方法实现 PTDA 分析。PTDA 分析牵涉到的主要运算是两个代表作业执行时间的相互独立的随机变量的和。和的 PDF 等于两个随机变量 PDF 的卷积，即 $f(t)=g(t)\otimes h(t)$。直接实现卷积运算的方法是先离散化，再累加，即：

$$f_i=\sum_{j=0}^{N-1}g_jh_{i-1} \tag{16}$$

计算复杂度是 $O(N^2)$，其中 N 是对 $g(t)$ 和 $h(t)$ 离散化表示的点数。我们还可通过卷积定理和快速傅里叶变换 FFT 将计算复杂度降到 $O(N\log_2N)$。

下面估算任务相互独立时 PTDA 分析的计算复杂度。设 ϕ_i－级忙周期包含 M_i（$i=1,\ 2,\ \ldots,\ n$）个作业，记 $M=\max\{M_i|\ i=1,\ 2,\ \ldots,\ n\}$。在 $J_{i,j}$ 的响应区间（$r_{i,j}$，$C_{i,j}$）内发布的优先级高于 ϕ_i 的作业总数为：

$$m_{i,j} = \sum_{k=1}^{i-1}\left(\left\lceil \frac{C_{i,j}}{P_k} \right\rceil - \left\lceil \frac{r_{i,j}}{P_k} \right\rceil\right) \tag{17}$$

$\lceil x \rceil$ 表示大于等于 x 的最小整数。$(r_{i,j}, C_{i,j})$ 区间被分割成 $m_{i,j}+1$ 个子区间，计算 $w_{i,j}(t)$ 需进行 $m_{i,j}+1$ 次卷积运算，故整个 PTDA 分析的计算复杂度为：

$$\left[\sum_{i=1}^{n}\sum_{j=1}^{M_i}(m_{i,j}+1)\right] \cdot O(N\log_2 N) \tag{18}$$

如果优先级高于 ϕ_i 的作业在 $(r_{i,j}, C_{i,j})$ 内都只发布一次，则 PTDA 的计算复杂度小于下式。

$$M\sum_{i=1}^{n} i \cdot O(N\log_2 N) = \frac{n(n+1)M}{2} \cdot O(N\log_2 N) \tag{19}$$

在利用 FFT 对作业执行时间的 PDF 进行卷积时，需注意以下问题：第一，在对两个随机变量的 PDF 进行离散化表示时，采样率和离散点数必须相等；第二，离散化后的矢量长度必须是 2 的幂次，长度不够的在后面添 0；第三，对 PDF 离散化时，离散后的矢量长度要选择合适的大小。抽样点数越多，计算得到的作业响应时间的概率分布与实际的分布之间的绝对误差越小，计算时间相应增加。我们的实验表明，对计算复杂度和结果的精度折衷考虑，选取 $N=1024$ 比较合适，卷积结果的绝对误差在 0.005 以内。

4 实例分析与讨论

考虑一个简单的随机弱实时系统，它由两个相互独立的周期任务 T_1，T_2 组成，任务属性说明如表 1 所示。假定任务执行时间服从均匀分布。平均和最坏情形下资源利用率总和分别是 0.708 和 1.411，因此在单处理机条件下，可能会有一些作业延误时限。T_1 的优先级高且它的最大资源利用率小于 1.0，故其所有作业均能满足时限约束。我们只需对任务 T_2 进行分析。

表 1 任务属性

Tab. 1 Task attributes

T_i	P_i	D_i	E_i^-	E_i	E_i^+	U_i^-	U_i	U_i^+
$T1$	300	300	1	100	199	0.0033	0.333	0.663
$T2$	400	400	1	150	299	0.0025	0.375	0.748
总和						0.0058	0.708	1.411

为计算 $J_{2,1}$ 按时完成的概率，把区间 $[0, 400)$ 分割成两个子区间 $[0, 300)$ 和 $[300, 400)$。若 $J_{2,1}$ 在 $t \in (0, 300)$ 之前完成，处理时间需求 $W_{2,1}(t)$ 为 $J_{1,1}$ 和 $J_{2,1}$ 的执行时间之和，其 PDF 为：

$$f_z(z) = \begin{cases} \frac{1}{6*10^4}z & ,0 \leqslant z < 200 \\ \frac{1}{300} & ,200 \leqslant z < 300 \\ \frac{1}{6*10^4}(500-z) & ,300 \leqslant z < 500 \end{cases} \tag{20}$$

由此可得 $J_{2,1}$ 在 $t=300$ 之前完成的概率为 $P\{W_{2,1}(300) \leqslant 300\} = 0.668$。若 $J_{2,1}$ 在 $t \in [300, 400)$ 之前完成，时间需求 $W_{2,1}(t)$ 是 $J_{1,2}$ 的执行时间与截至 $t=300$ 时刻 $J_{2,1}$ 尚未完成时的条件作业积压分布 $B_{2,1}(300)$ 之和，$B_{2,1}(300)$ 是对 $W_{2,1}(t)$，$t \in (0, 300)$ 中 $(300, 498)$ 部分进行归一化处理后得到的条件映射分布，$W_{2,1}(t)$ 的 PDF 为：

$$f_z(z) = \begin{cases} -\frac{1}{8*10^6}(z^2 - 1000z + 2.3^* 10^5) & ,300 \leqslant z < 500 \\ \frac{1}{8*10^6}(z^2 - 1400z + 9^* 10^4), & 500 \leqslant z \leqslant 700 \end{cases} \tag{21}$$

因此当 $J_{2,1}$ 在 $t=300$ 之前未完成而在 $t=400$ 之前完成的条件概率为 0.209。综上所述，$J_{2,1}$ 在其时限

$t=400$ 之前完成的概率为:

$$P\{w_{2,1}(400)\leqslant 400\}=0.668+0.209^{*}(1-0.668)=0.738 \tag{22}$$

同理可分析出 $J_{2,2}$在其时限 $t=800$ 之前按时完成的概率为 0.994，$J_{2,3}$在其时限 $t=1200$ 之前按时完成的概率为1.0。ϕ_2- 级忙周期在 $J_{2,3}$完成时结束。起始于临界时刻的第一个 ϕ_2- 级忙周期对应着 T_2 的作业的响应时间的最坏情形，因此任务 T_2 的作业的时限延误率的上界为 $1-0.738=0.262$。

表 2　仿真结果与 PTDA 分析结果的对比

Tab. 2　Comparison of simulation results and PTDA analysis results

T_i	PTDA 结果	仿真实验结果		误差
		初相同步情形	初相随机情形	
T_1	100.0	100.0±0.0	100.0±0.0	0
T_2	73.8	78.4±0.1	80.2±0.1	-5.9%

对表 1 所示的弱实时系统作 1000 次仿真运行，每次运行执行 1000 项 T_2 的作业和 1333 项 T_1 的作业。对 T_1 和 T_2 的每一项作业，将 1000 次仿真实验得到的作业响应时间平均，得到任务 T_1 和 T_2 所有作业的响应时间直方图，进而可得作业响应时间的概率分布。表 2 列出了仿真结果与 PTDA 分析结果的对比，其中初相同步情形取 T_1 和 T_2 的初始相位为 0，初相随机情形则让 T_i，$i=1$，2 的初始相位服从 $(-P_i, P_i)$ 上的均匀分布。由表 2 可知，初相随机情形下作业满足时限要求的概率高于初相同步情形，因为初相同步时各任务的作业经常同时请求处理机资源；PTDA 得到的作业满足时限的概率比仿真结果低 5.9%，表明该方法是准确有效的。PTDA 对影响 ϕ_i- 级忙周期内的作业 $J_{i,j}$的响应时间的各种因素进行了全面合理的分析，但分析相对比较保守，它仅考虑了最坏情形即起始于临界时刻的第一个 ϕ_2- 级忙周期而并非任务 T_2 中所有作业满足时限约束的数学期望，因此通常情况下 PTDA 结果低于实际值。

参考文献:

[1] 毛羽刚. 分布强实时系统调度技术的研究 [D]. 国防科技大学工学博士学位论文，1999. 10.

[2] Liu C L, Layland J W. Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment [J]. Journal of ACM, 1973, 10 (1): 174-189.

[3] Lehoczky J, Sha L, Ding Y. The Rate Monotonic Scheduling Algorithms: Exact Characterization and Average Case Behaviour [C]. IEEE Real-Time Systems Symposium, 1989, 12: 166-171.

[4] Lehoczky J. Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Dealines [C}. IEEE Real-Time Systems Symposium, 1990, 12: 201-209.

[5] Audsley N, Burns A, Tindell K. Applying New Scheduling Theory to Static Priority Preemptive Scheduling [J]. Journal of Software Engineering, 1993, 8 (5).

[6] Tia T S, Deng Z, et al. Probabilistic Performance Guarantee for Real-Time Tasks with Varying Computation Times [C]. Proceedings of IEEE Real-Time Technology and Applications Symposium, 1995, 5: 164-173.

[7] Mok A K. Fundamental Design Problems of Distributed Systems for the Hard Real-Time Environment [D]. Ph. D Thesis, MIT, 1983.

[8] Peng Liangzhi, Dai Jinhai, Gui Xianzhou. QoS Guarantee for Stochastic Real-time Systems [C]. BICSC' 99, Beijing, China, 1999, 10: 494-499.

[9] 彭良智，戴金海，桂先洲，单懿. 强实时系统静态优先级调度的可调度性分析 [J]. 计算机工程与应用，1999，35 (12): 13-16.