

## 周期タスクの初期位相分布を考慮した 応答時間の確率的解析

石川 拓也<sup>†1</sup> 松原 豊<sup>†1</sup> 高田 広章<sup>†1</sup>

近年、タスクの実行時間を確率変数として扱い、リアルタイムシステムの応答時間を確率的に解析する手法が提案されている。一般的に、確率変数を用いて解析する場合、解析手法が複雑になり、解析時間が長くなってしまうことが問題とされる。そのため、複雑さを軽減し、短時間で解析可能な確率的解析手法が求められる。本論文では、タスクの実行時間に加えて、初期位相を確率変数として扱い、タスクの応答時間分布を解析する手法を提案する。提案する手法ではまず、タスクの実行時間と初期位相の確率分布を離散化して扱う。離散化にともなって生じる誤差については、解析結果の応答時間分布が悲観的になるように丸める。さらに、応答時間分布の終端部分に着目して解析することによって、解析する初期位相の組合せ、および、解析する応答時間分布の区間を限定し、解析時間を短縮する。

### Stochastic Analysis on Periodic Task Model with Initial Phase Distribution

TAKUYA ISHIKAWA,<sup>†1</sup> YUTAKA MATSUBARA<sup>†1</sup>  
and HIROAKI TAKADA<sup>†1</sup>

These days, a stochastic analysis method for response times of real-time systems is proposed using random execution times of tasks. In general, it becomes a problem that a stochastic analysis takes long time because of complexity. Therefore, techniques for stochastic analysis with reducing the complexity are required. In this paper, we propose a stochastic analysis method for response times of tasks. In the proposed method, initial phases of tasks are modelled as random values, in addition to execution times of tasks. One problem caused by using random initial phases of tasks is that analysis takes long time, because response time distribution is analyzed for all combinations of initial phases of tasks. To solve this problem, we first discretize both of execution time distribution and initial phase distribution of tasks. The discretization errors are rounded off so that response time distribution is analyzed pessimistically. Then, we focus only on analyzing the end of response time distribution.

### 1. はじめに

従来のリアルタイムシステムにおける応答時間解析では、タスクの最大実行時間など、一意的な値が用いられることが多い。このような解析手法は、短時間での解析が可能であり、文献 6) や文献 5) をはじめとして、現在までに多くの手法が提案されている。しかし、これらの手法で仮定されている、実行時間が最大となるような状況の発生頻度は非常に低く、これらの手法によって解析された結果はあまりに悲観的である<sup>7),9)</sup>。応答時間の解析結果が悲観的である場合、そのシステムに求められる性能を過大に見積もってしまい、開発コストの増加や開発期間の増加を起こしてしまう可能性がある。

そこで近年、タスクの実行時間を確率変数として扱い、リアルタイムシステムの応答時間を確率的に解析する手法が提案されている。文献 2), 4) では、単一のプロセッサ内において、タスクの実行時間のみを確率変数として扱い、プリエンティブな固定優先度ベースのスケジューリングのもとで、タスクの応答時間分布を数学的に解析する手法を提案している。そして、文献 8), 9) では、文献 2), 4) で提案されている手法をベースとして、分散リアルタイムシステムにおける端点間処理の応答時間分布を数学的に解析する手法を提案している。一方で、確率変数を用いて解析を行う場合、解析手法が複雑になり、解析時間が長くなってしまうことが問題とされる。そのため、複雑さを軽減し、短時間で解析可能な手法として、応答時間分布を悲観的に近似する手法が提案されている<sup>3),7)</sup>。しかしながら、これまでに提案されている解析手法では、対象とするタスクセットのモデルにおいて、周期タスクの初期位相が確率変数である場合が考慮されておらず、初期位相が定数であるタスクセットしか解析することができない。たとえば、エンジン制御のアプリケーションにおいて、周期が一定のタスクと、エンジンの回転数に同期して周期の変動するタスクとが混在する場合、タスク間の位相差がランダムな値となる。

本論文では、タスクの実行時間に加えて、初期位相を確率変数として扱い、タスクの応答時間分布を解析する手法について提案する。タスクの初期位相を確率変数として扱う場合、各タスクの初期位相がとりうる値の任意の組合せに対して、応答時間分布を解析する必要がある。解析時間が長くなると考えられる。そこで、本論文で提案する手法では、次に示す 2

<sup>†1</sup> 名古屋大学大学院情報科学研究科  
Graduate School of Information Science, Nagoya University

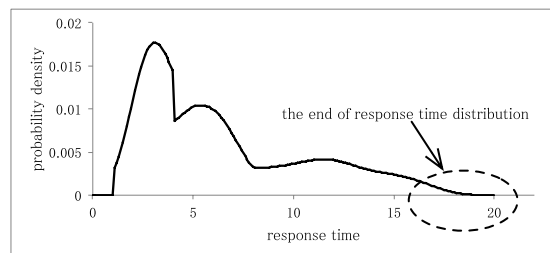


図 1 応答時間分布の終端部分

Fig. 1 The end of response time distribution.

つの手法により、応答時間分布の解析時間を短縮する。

- (1) タスクの実行時間分布と初期位相分布を離散化する。
- (2) 応答時間分布の終端部分に着目して数学的に解析する。

(1) については、まず、タスクの実行時間分布を離散化することにより、応答時間分布を数値解析で近似的に求めることができる。そして、初期位相分布を離散化することにより、応答時間分布を解析する初期位相の組合せを有限とすることができる。タスクの実行時間分布と初期位相分布の離散化については、離散化した確率分布を用いて解析した応答時間分布が、正確な応答時間分布よりも悲観的に近似されるように離散化誤差を丸める。

(2) については、応答時間分布の終端部分に着目して解析することによって、解析する初期位相の組合せ、および、解析する応答時間分布の区間を限定する。ここで、応答時間分布の終端部分とは、最大応答時間に近い区間を指す(図1)。リアルタイムシステムの開発では、応答時間がデッドラインを超えるかどうかを検証することが重要であり、応答時間が長い区間について、確率分布を求めることが重要であると考えられる。そして、応答時間分布の終端部分は、確率密度が低く、シミュレータなどにより分布を求める場合、解析時間が長くなってしまふ。そこで、応答時間分布の終端部分に着目して、数学的に解析する手法を提案することを考えた。

本論文の構成は次のとおりである。まず、2章で関連研究について紹介する。3章では、本論文で扱うシステムのモデルについて述べる。4章では、タスクの実行時間分布と初期位相分布の離散化について述べる。5章では、タスクの応答時間分布の解析について述べる。最後に、6章で本論文のまとめを行う。

## 2. 関連研究

文献 1) では、タスクの最大実行時間を用いて解析した結果、デッドラインミスが発生すると判定されたとしても、実際にシステムが稼働する際には、デッドラインミスが発生しない場合や、デッドラインミスが許容される場合があるとしている。そして、スケジューラビリティの判定を、デッドラインミスが発生する確率や回数を用いて行うようリアルタイムシステムとして、weakly hard real-time systems を提案している。

文献 2), 4) では、単一のプロセッサにおいて、互いに依存関係のない周期タスクからなるタスクセットを対象として、プリエンティブな固定優先度ベースのスケジューリングのもとで、タスクの応答時間を確率的に解析する手法を提案している。ここでは、タスクの実行時間のみを確率変数として扱っている。提案されている手法では、対象とするタスクについて、ハイパーピリオド内のすべてのインスタンスについて応答時間分布を数学的に解析し、それらの結果を平均化することによって、タスクの応答時間分布を解析している。

文献 8), 9) では、自動車の電子システムを対象として、分散システムにおける端点間処理の応答時間を確率的に解析する手法を提案している。分散システム内の各プロセッサにおけるタスクの応答時間分布の解析には、文献 2), 4) で提案されている手法をベースとして用いており、加えて、ノンプリエンティブなスケジューリングのもとで応答時間分布を解析する手法を提案している。また、プロセッサ間通信として CAN プロトコルによる通信を想定し、CAN メッセージの応答時間を近似する手法を提案している。そして、タスクの応答時間分布と CAN メッセージの応答時間分布を組み合わせることで、分散システムにおける端点間処理の応答時間分布を解析する手法を提案している。

文献 3), 7) では、文献 2), 4) で提案されている手法は、単純なモデルを扱う場合でのみ適用可能であると述べている。そして、資源の共有やリリースジッタなどを考慮した、複雑なモデルを扱う場合において、確率を悲観的に近似するための手法を提案している。さらに、文献 2), 4) のモデルを拡張し、資源共有によるブロッキング時間やリリースジッタを考慮したモデルを扱い、応答時間分布を悲観的に近似する手法を示している。

しかしながら、これまでに提案されている解析手法では、対象とするタスクセットのモデルにおいて、周期タスクの初期位相が確率変数である場合が考慮されておらず、初期位相が定数であるタスクセットしか解析することができない。

### 3. システムのモデル

本論文で扱うリアルタイムシステムは、単一のプロセッサのみで構成されるものとし、プロセッサ内には複数のタスクが存在するものとする。そして、タスクのスケジューリング方式は、プリエンプティブな固定優先度ベースのスケジューリングとする。また、タスク間の依存関係や割込みについては考慮しないものとする。

タスク  $\tau_i$  は、 $(P_i, T_i, E_i, O_i)$  というパラメータによって特徴づけられるものとする。 $P_i$ ,  $T_i$  は定数とし、それぞれ、 $\tau_i$  の優先度、 $\tau_i$  の起動周期を示す。ここで、 $P_i$  は正の整数であり、優先度が高いタスクほど、 $P_i$  の値は小さいものとする。 $E_i$ ,  $O_i$  は確率変数とし、それぞれ、 $\tau_i$  の実行時間分布、 $\tau_i$  の初期位相分布を示す。ここで、初期位相とは、システムの起動後に初めて  $\tau_i$  が起動する時刻である。 $E_i$ ,  $O_i$  は、それぞれ、区間  $[E_i^{min}, E_i^{max}]$ ,  $[0, T_i)$  においてのみ確率密度が正の値をとるものとし、この区間以外では、確率密度が 0 になるものとする。そして、 $O_i$  は区間  $[0, T_i)$  の一様分布に従うものとする。また、タスク  $\tau_i$  の最大応答時間  $R_i^{max}$  は、 $\tau_i$  の起動周期  $T_i$  以下であるとする。以降、確率変数  $X$  について、 $X$  の確率密度関数、累積確率分布関数をそれぞれ、 $f_X$ ,  $F_X$  と表すものとし、 $X$  がある区間  $I$  に属する確率を、 $P(X \in I)$  のように表すものとする。

### 4. 確率分布の離散化

リアルタイムシステムでは、タスクの応答時間が、デッドラインを満たすことが厳格に求められる。そのため、従来の決定的な応答時間解析では、タスクの最大実行時間などを用い、応答時間を悲観的に解析する。よって、応答時間を確率的に解析する場合においても、応答時間分布を悲観的に解析する必要がある<sup>3),7)</sup>。そこで、タスクの実行時間分布と初期位相分布の離散化では、離散化した確率分布を用いて解析した応答時間分布が、正確な応答時間分布よりも悲観的な結果となるように離散化誤差を丸める。本章ではまず、確率分布における悲観論について説明する。その後、タスクの実行時間分布と初期位相分布を離散化する手法について、それぞれ述べる。

#### 4.1 確率分布における悲観論

文献 3), 7) では、2 つの確率分布の間における悲観論を次のように定義している。

定義 4.1 ある 2 つの確率変数  $X$ ,  $Y$  について、任意の  $x$  に対し、 $F_X(x) \leq F_Y(x)$  が成立するならば、 $X$  は  $Y$  よりも悲観的である。

ここで、正確な応答時間分布を  $R$ ,  $R$  よりも悲観的な応答時間分布を  $R'$  とする。 $R'$  は、 $R$  よりも悲観的であるので、任意の  $t$  について、 $F_{R'}(t) \leq F_R(t)$  が成立し、 $1 - F_{R'}(t) \geq 1 - F_R(t)$  が成立する。 $t$  をデッドラインだと仮定すると、先述の式は、悲観的な応答時間分布から解析されるデッドラインミス率が、正確なデッドラインミス率よりも高いことを意味している。よって、定義 4.1 で定義されているように、悲観的に応答時間分布を解析すれば、デッドラインミス率を低く見積もることがなく、安全な解析結果が得られる。

4.2 実行時間分布の離散化

本節では、タスクの実行時間分布を離散化する手法について述べる。タスクの実行時間分布の離散化は、離散化後の確率分布を用いて解析した応答時間分布が、4.1 節で述べたように、正確な応答時間分布よりも悲観的になるように、離散化誤差を丸める。

文献 3) において、実行時間分布と応答時間分布の関係について、次の定理が示されている。

定理 4.1 あるタスクの 1 つのインスタンスのみ、実行時間が確率変数  $C$  であるとし、その他のパラメータはすべて定数であるとする。そして、 $C$  を用いて解析した応答時間分布を  $R$  とし、 $C$  よりも悲観的な確率分布  $C'$  を用いて解析した応答時間分布を  $R'$  とする。このとき、 $R'$  は  $R$  よりも悲観的である。

定理 4.1 より、タスクの実行時間分布として、正確な実行時間分布よりも悲観的な確率分布を用いて応答時間分布を解析した場合、解析結果は正確な応答時間分布よりも悲観的になることが分かる。よって、タスクの実行時間分布の離散化では、基となる確率分布よりも悲観的な確率分布となるように離散化誤差を丸める。

そこで、 $\tau_i$  の実行時間分布  $E_i$  を離散化した確率分布  $E_i^d$  について、次のように定義する。

定義 4.2 離散化の刻み幅を  $\Delta_e$  とし、実際の  $\tau_i$  の実行時間分布を  $E_i$ ,  $E_i$  を離散化した実行時間分布を  $E_i^d$  とする。任意の整数  $k$  について、 $f_{E_i^d}(k\Delta_e)$  を次の式で定義する。

$$f_{E_i^d}(k\Delta_e) = P((k-1)\Delta_e < E_i \leq k\Delta_e) \quad (1)$$

ここで、定義 4.2 において、任意の時間  $t$  に対する  $E_i^d$  の累積確率は、次の式を満たす。

$$F_{E_i^d}(t) = F_{E_i}\left(\left\lfloor \frac{t}{\Delta_e} \right\rfloor \Delta_e\right) \leq F_{E_i}(t) \quad (2)$$

定義 4.1 より、 $E_i^d$  は  $E_i$  よりも悲観的な分布である。よって、 $E_i^d$  を用いて解析した応答時間分布は、 $E_i$  を用いて解析した応答時間分布よりも悲観的になる。

#### 4.3 初期位相分布の離散化

本節では、タスクの初期位相分布を離散化する手法について述べる。タスクの初期位相分

布を離散化する際には、実行時間分布の離散化と同様に、離散化後の確率分布を用いて解析した応答時間分布が、正確な応答時間分布よりも悲観的になるように、離散化誤差を丸める。

本論文で提案する応答時間分布解析では、解析対象のタスク  $\tau_i$  と、 $\tau_i$  の実行を妨げるタスク  $\tau_j$  の初期位相の差が必要となる。そのため、初期位相分布の離散化は、 $\tau_i$  と  $\tau_j$  の初期位相の差の離散確率分布を求めることによって行う。 $\tau_i$  と  $\tau_j$  の初期位相の差  $O_i - O_j$  の離散確率分布  $O_{ij}^d$  について、次のように定義する。

定義 4.3 離散化の刻み幅を  $\Delta_o$  とする。そして、 $O_i$  を離散化した確率分布として、 $O_i^U$  と  $O_i^L$  を、それぞれ次の式で定義する。

$$f_{O_i^U}(l\Delta_o) = P((l-1)\Delta_o < O_i \leq l\Delta_o) \quad (3)$$

$$f_{O_i^L}(l\Delta_o) = P(l\Delta_o \leq O_i < (l+1)\Delta_o) \quad (4)$$

ここで、 $l$  は任意の整数とする。さらに、 $O_j$  を離散化した確率分布として、 $O_j^U$  と  $O_j^L$  を、それぞれ  $O_i^U$ 、 $O_i^L$  と同様に定義する。また、 $O_{ij}^d = k\Delta_o$  ( $k$  は任意の整数) のときの、 $\tau_i$  の解析対象のインスタンスの起動時刻を  $a_i$ 、 $a_i$  以前で、どのタスクも実行中でない最も遅い時刻を  $a_0$ 、 $a_0$  以降で最も早い  $\tau_j$  のインスタンスの起動時刻を  $a_j$  とする。

(1)  $a_i \neq a_0$  のとき、 $f_{O_{ij}^d}$  を次の式で定義する。

$$f_{O_{ij}^d}(k\Delta_o) = \begin{cases} \sum_{l \in Z} f_{O_i^L}(l\Delta_o) f_{O_j^U}((l-k)\Delta_o) & (a_j < a_i) \\ \sum_{l \in Z} f_{O_i^L}(l\Delta_o) f_{O_j^L}((l-k)\Delta_o) & (a_j > a_i) \\ \sum_{l \in Z} f_{O_i^L}(l\Delta_o) \left( f_{O_j^U}((l-k)\Delta_o) + f_{O_j^L}((l-k)\Delta_o) \right) & (a_j = a_i) \end{cases} \quad (5)$$

(2)  $a_i = a_0$  のとき、 $f_{O_{ij}^d}$  を次の式で定義する。

$$f_{O_{ij}^d}(k\Delta_o) = \begin{cases} \sum_{l \in Z} f_{O_i^U}(l\Delta_o) f_{O_j^L}((l-k)\Delta_o) & (a_j > a_i) \\ \sum_{l \in Z} \left( f_{O_i^U}(l\Delta_o) f_{O_j^U}((l-k)\Delta_o) + f_{O_i^U}(l\Delta_o) f_{O_j^L}((l-k)\Delta_o) + f_{O_i^L}(l\Delta_o) f_{O_j^U}((l-k)\Delta_o) \right) & (a_j = a_i) \end{cases} \quad (6)$$

ここで、定義 4.3 における、 $O_{ij}^d$  を用いて解析した応答時間分布が、正確な応答時間分布よりも悲観的であることを示す。

証明 (式 (5) について) まず、 $O_i$  の離散化について示す。式 (5) では、 $O_i$  を離散化した分布として、 $O_i^L$  を用いている。これは、 $a_i \leq a'_i$  を満たす  $a'_i$  を、離散化によって  $a_i$  に丸めることを示している。よって、 $\tau_i$  の初期位相を小さくしたとき、つまり、 $a'_i$  を  $a_i$  に近づけたときの応答時間分布が、初期位相を小さくする前の応答時間分布よりも悲観的になることを示せばよい。

まず、 $\tau_i$  の起動時刻が  $a_i$  である場合を考える。 $a_0$  から  $a_i$  の間に起動したタスクによって、 $\tau_i$  の実行が妨げられる時間の確率分布を  $W_i$  とする。そして、 $a_i$  以降で、 $\tau_i$  の実行を妨げるタスクの起動時刻を  $a_k$ 、実行時間分布を  $E_k$  とする。このときの、 $\tau_i$  の応答時間分布  $R_i$  について考える。 $R_i$  が、 $(a_k - a_i) < t$  を満たす任意の時間  $t$  以上となる確率は、

$$P(R_i \geq t) = \int_0^{a_k - a_i} \int_{(a_k - a_i) - x}^{\infty} f_{W_i}(x) f_{E_i}(y) P(t - y \leq E_k) dy dx + \int_{a_k - a_i}^{\infty} f_{W_i}(x) P(t - x \leq E_i + E_k) dx \quad (7)$$

となる。

次に、 $\tau_i$  の起動時刻が  $a'_i$  である場合を考える。 $a_0$  から  $a'_i$  の間に起動したタスクによって、 $\tau_i$  の実行が妨げられる時間の確率分布を  $W'_i$  とする。このときの、 $\tau_i$  の応答時間分布  $R'_i$  について考える。 $R'_i$  が、 $t$  以上となる確率は、

$$P(R'_i \geq t) = \int_0^{a_k - a'_i} \int_{(a_k - a'_i) - x}^{\infty} f_{W'_i}(x) f_{E_i}(y) P(t - y \leq E_k) dy dx + \int_{a_k - a'_i}^{\infty} f_{W'_i}(x) P(t - x \leq E_i + E_k) dx \quad (8)$$

となる。

ここで、 $W_i > (a'_i - a_i)$  のとき、 $f_{W_i}(x + a'_i - a_i) = f_{W'_i}(x)$  であること、および、5章で述べるように、提案手法では、対象タスクが最大回数実行を妨げられる場合のみ解析することより、式 (8) の第 1 項は、

$$\int_{a'_i - a_i}^{a_k - a_i} \int_{(a_k - a_i) - x}^{\infty} f_{W_i}(x) f_{E_i}(y) P(t - y \leq E_k) dy dx \quad (9)$$

となり、また、第 2 項も同様に、

$$\int_{a_k - a_i}^{\infty} f_{W_i}(x) P(t - (x - (a'_i - a_i)) \leq E_i + E_k) dx \quad (10)$$

となる．ここで，式 (9) が式 (7) の第 1 項よりも小さく，式 (10) が式 (7) の第 2 項よりも小さいことは明らかである．

よって， $P(R_i \geq t) \geq P(R'_i \geq t)$  が成立し， $R_i$  は  $R'_i$  よりも悲観的であるといえるため， $\tau_i$  の初期位相を小さくしたときの応答時間分布が，初期位相を小さくする前の応答時間分布よりも悲観的になるといえる．

次に， $O_j$  の離散化について，3 つの場合に分けて示す．

(a)  $a_j < a_i$  の場合

$O_j$  を離散化した分布として， $O_j^U$  を用いている．これは， $a_j \geq a'_j$  を満たす  $a'_j$  を，離散化によって  $a_j$  に丸めることを示している．よって， $\tau_j$  の初期位相を大きくしたとき，つまり， $a'_j$  を  $a_j$  に近づけたときの応答時間分布が，初期位相を大きくする前の応答時間分布よりも悲観的になることを示せばよい．

まず， $\tau_j$  の起動時刻が  $a_j$  である場合について考える． $a_0$  から  $a_j$  の間に起動したタスクによって， $\tau_j$  の実行が妨げられる時間の確率分布を  $W_j$  とする．このときの， $\tau_j$  が  $\tau_i$  の実行を妨げる時間の確率分布  $W_i$  について考える． $W_i$  が， $0 < t$  を満たす任意の時間  $t$  以上となる確率は，

$$P(W_i \geq t) = \int_0^\infty f_{W_j}(x)P(t - (x - (a_i - a_j))) \leq E_j)dx \quad (11)$$

となる．

次に， $\tau_j$  の起動時刻が  $a'_j$  である場合について考える． $a_0$  から  $a'_j$  の間に起動したタスクによって， $\tau_j$  の実行が妨げられる時間の確率分布を  $W'_j$  とする．このときの， $\tau_j$  が  $\tau_i$  の実行を妨げる時間の確率分布  $W'_i$  について考える． $W'_i$  が， $t$  以上となる確率は，

$$P(W'_i \geq t) = \int_0^\infty f_{W'_j}(x)P(t - (x - (a_i - a'_j))) \leq E_j)dx \quad (12)$$

となる．

ここで， $W_j > 0$  のとき， $f_{W_j}(x) = f_{W'_j}(x + (a_j - a'_j))$  であり，かつ， $W_j = 0$  のとき， $f_{W_j}(0) = \int_0^{a_j - a'_j} f_{W'_j}(x)dx$  であることより，式 (11) は，

$$\begin{aligned} & \int_0^{a_j - a'_j} f_{W'_j}(x)dx P(t + (a_i - a_j) \leq E_j) \\ & + \int_{a_j - a'_j}^\infty f_{W'_j}(x)P(t - (x - (a_i - a'_j))) \leq E_j)dx \end{aligned} \quad (13)$$

となる．そして， $0 \leq x \leq (a_j - a'_j)$  の範囲において， $t - (x - (a_i - a'_j)) \geq t + (a_i - a_j)$  であるため，式 (12) は式 (13) (式 (11)) 以下であることが分かる．

よって， $P(W'_i \geq t) \leq P(W_i \geq t)$  が成立し， $W_i$  は  $W'_i$  よりも悲観的であるといえるため， $W_i$  を用いて解析した応答時間分布は， $W'_i$  を用いて解析した応答時間分布よりも悲観的であるといえる．そして， $\tau_j$  の初期位相を大きくしたときの応答時間分布が，初期位相を大きくする前の応答時間分布よりも悲観的になるといえる．

(b)  $a_j > a_i$  の場合

$O_j$  を離散化した分布として， $O_j^L$  を用いている．これは， $a_j \leq a'_j$  を満たす  $a'_j$  を，離散化によって  $a_j$  に丸めることを示している．よって， $\tau_j$  の初期位相を小さくしたとき，つまり， $a'_j$  を  $a_j$  に近づけたときの応答時間分布が，初期位相を小さくする前の応答時間分布よりも悲観的になることを示せばよい．

まず， $\tau_j$  の起動時刻が  $a_j$  である場合について考える． $\tau_i$  の応答時間分布を  $R_i$  とし， $\tau_j$  に実行を妨げられない場合の， $\tau_i$  の応答時間分布を  $\hat{R}_i$  とする． $R_i$  が， $0 < t$  を満たす任意の時間  $t$  以下となる確率は，

$$P(R_i \leq t) = \begin{cases} \int_0^t f_{\hat{R}_i}(x)dx & (t \leq (a_j - a_i)) \\ \int_0^{a_j - a_i} f_{\hat{R}_i}(x)dx + \int_{a_j - a_i}^t \int_0^{x - (a_j - a_i)} f_{\hat{R}_i}(x - y)f_{E_j}(y)dydx & (t > (a_j - a_i)) \end{cases} \quad (14)$$

となる．

次に， $\tau_j$  の起動時刻が  $a'_j$  である場合について考える．このときの， $\tau_i$  の応答時間分布を  $R'_i$  とする． $R'_i$  が， $t$  以下となる確率は，

$$P(R'_i \leq t) = \begin{cases} \int_0^t f_{\hat{R}_i}(x)dx & (t \leq (a'_j - a_i)) \\ \int_0^{a'_j - a_i} f_{\hat{R}_i}(x)dx + \int_{a'_j - a_i}^t \int_0^{x - (a'_j - a_i)} f_{\hat{R}_i}(x - y)f_{E_j}(y)dydx & (t > (a'_j - a_i)) \end{cases} \quad (15)$$

となる．

ここで， $a'_j > a_j$  より， $(a'_j - a_i) > (a_j - a_i)$  であるため，式 (14) と式 (15) より， $P(R'_i \leq t)$  は， $P(R_i \leq t)$  以上であることが分かる．

よって， $P(R'_i \leq t) \geq P(R_i \leq t)$  が成立し， $R_i$  は  $R'_i$  よりも悲観的であるといえるため， $\tau_j$  の初期位相を小さくしたときの応答時間分布が，初期位相を小さくする前の応答時間分布よりも悲観的になるといえる．

(c)  $a_j = a_i$  の場合

$a_j = a_i$  の場合は， $a_j < a_i$  の場合と  $a_j > a_i$  の場合との境界となる場合であるため，

$a_j < a_i$  である  $a_j$  に離散化誤差を丸めた確率と,  $a_j > a_i$  である  $a_j$  に離散化誤差を丸めた確率の両者を含む. そのため,  $O_j$  を離散化した分布として,  $O_j^U + O_j^L$  を用いており, そして, この分布を用いて解析した応答時間分布が悲観的であることは, 先述の (a) と (b) の場合の証明より明らかである.  $\square$

証明 (式 (6) について) (d)  $a_j > a_i$  の場合と, (e)  $a_j = a_i$  の場合の 2 つに分けて示す.

(d)  $a_j > a_i$  の場合

$O_j$  を離散化した分布として,  $O_j^L$  を用いており, そして,  $O_i$  を離散化した分布として,  $O_i^U$  を用いている. これは,  $a_j \leq a'_j$  を満たす  $a'_j$  を, 離散化によって  $a_j$  に丸め, そして,  $a_i \geq a'_i$  を満たす  $a'_i$  を, 離散化によって  $a_i$  に丸めることを示している. よって,  $a'_j$  を  $a_j$  に近づけたとき, また,  $a'_i$  を  $a_i$  に近づけたときに, 応答時間分布が悲観的になることを示せばよい.

$a'_j$  を  $a_j$  に近づけたときについては, 式 (5) の証明における (b) の証明より, 応答時間分布が悲観的になることが分かる.

$a'_i$  を  $a_i$  に近づけたときについては,  $\tau_i$  から見た  $\tau_j$  の相対的な起動時刻を近づけると考えることができる. よって, 式 (5) の証明における (b) より, 応答時間分布が悲観的になることが分かる.

(e)  $a_j = a_i$  の場合

$a_j = a_i$  となる場合は次の 3 通りである.

- (e-1)  $a'_j \geq a_j$  を満たす  $a'_j$  を離散化によって  $a_j$  に丸めたとき, また,  $a'_i \leq a_i$  を満たす  $a'_i$  を離散化によって  $a_i$  に丸めたとき
- (e-2)  $a'_j \leq a_j$  を満たす  $a'_j$  を離散化によって  $a_j$  に丸めたとき, また,  $a'_i \geq a_i$  を満たす  $a'_i$  を離散化によって  $a_i$  に丸めたとき
- (e-3)  $a'_j \leq a_j$  を満たす  $a'_j$  を離散化によって  $a_j$  に丸めたとき, また,  $a'_i \leq a_i$  を満たす  $a'_i$  を離散化によって  $a_i$  に丸めたとき

これらの場合のそれぞれについて, 応答時間分布が悲観的になることを示す.

(e-1) の場合については, (d) の場合についての証明より, 応答時間分布が悲観的になることが分かる.

(e-2) の場合については, 式 (5) の証明における  $O_i$  の離散化と, (a) の場合についての証明より, 応答時間分布が悲観的になることが分かる.

(e-3) の場合について考える. まず,  $a'_j < a'_i$  のときに,  $a'_j$  を  $a'_i$  に近づけると, 式 (5) の証明における (a) の場合の証明より, 応答時間分布が悲観的になることが分かる. 次に,

$a'_j > a'_i$  のときに,  $a'_i$  を  $a'_j$  に近づけると, (d) の場合の証明より, 応答時間分布が悲観的になることが分かる. そして,  $a'_j = a'_i$  のときに,  $a'_j$  と  $a'_i$  を同時に  $a_j (= a_i)$  に近づけると,  $a_j$  以降に起動するタスクの相対的な起動時刻が  $a_j$  に近づくと見なすことができる. よって, (d) の証明より, 応答時間分布が悲観的になることが分かる.  $\square$

以上より, 定義 4.3 に従って初期位相分布を離散化した  $O_{i,j}^d$  を用い, 5 章で述べる手法で応答時間分布を解析した場合, 得られる応答時間分布は, 正確な応答時間分布よりも悲観的になるといえる.

## 5. タスクの応答時間解析

本章では, まず, タスクの初期位相を定数とした場合における, 応答時間の確率的な解析手法について述べる. 次に, タスクの初期位相を確率変数とした場合における, 応答時間の確率的な解析手法について述べる.

### 5.1 タスクの初期位相が定数の場合

タスクの初期位相を定数とした場合における, 応答時間の解析は, 文献 4) で提案されている手法に基づいて行う. タスクの応答時間分布の解析は, タスクの実行時間分布の和の分布を繰り返し計算することによって行う. ここで, 確率分布の和の分布の確率密度を計算するためには, 畳込み積分が用いられる.

タスクの応答時間を解析するために, まず, 対象とするタスク  $\tau_i$  のインスタンス  $\tau_{i,j}$  に対し,  $\tau_{i,j}$  の起動時刻  $A_{i,j}$  における, 優先度が  $P_i$  より高いタスクの残り実行時間の総和  $W_{A_{i,j}}^{P_i}$  を求める. ここで, 優先度が  $P_i$  より高いタスクの集合を  $hp(P_i)$  とする.

$W_{A_{i,j}}^{P_i}$  の解析は, 次のような手順で行う.

- (1) 時刻  $A_{i,j}$  以前で,  $hp(P_i)$  に属するどのタスクも実行中でない最も遅い時刻を  $t_0$  とする. ここで,  $W_{A_{i,j}}^{P_i}$  を 0 に初期化し,  $t$  を,  $t_0$  以降で,  $hp(P_i)$  に属するタスクが起動する最も早い時刻に初期化する. 確率変数  $W_{A_{i,j}}^{P_i}$  を 0 に初期化するとは,  $W_{A_{i,j}}^{P_i} = 0$  となる確率密度を 1, そうでない確率密度を 0 とすることを指す.
- (2)  $hp(P_i)$  に属するタスクのうち, 時刻  $t$  で起動するタスクの集合を  $At(t)$  とする. ここで,  $At(t)$  に属するタスクの実行時間を,  $W_{A_{i,j}}^{P_i}$  に加える. つまり,

$$W_{A_{i,j}}^{P_i} = W_{A_{i,j}}^{P_i} + \sum_{\forall \tau_k \in At(t)} E_k \quad (16)$$

となる.

- (3) 時刻  $t$  以降で,  $hp(P_i)$  に属するタスクが起動する最も早い時刻を  $t'$  とする.  $t'$  が  $A_{i,j}$

以上となるならば,  $t'$  を  $A_{i,j}$  とする. ここで,  $W_{A_{i,j}}^{P_i}$  の分布を, 負の方向に,  $t' - t$  だけ移動させる. このとき,  $W_{A_{i,j}}^{P_i}$  が負の値となる分布については, 0 の分布に集約する. つまり,  $W_{A_{i,j}}^{P_i}$  の確率密度関数  $f_{W_{A_{i,j}}^{P_i}}(w)$  は,

$$f_{W_{A_{i,j}}^{P_i}}(w) = \begin{cases} f_{W_{A_{i,j}}^{P_i}}(w + (t' - t)) & (w > 0) \\ \sum_{w'=-\infty}^{t'-t} f_{W_{A_{i,j}}^{P_i}}(w') & (w = 0) \\ 0 & (w < 0) \end{cases} \quad (17)$$

となる.

- (4)  $t'$  が  $A_{i,j}$  ならば解析を終了し, 現時点での  $W_{A_{i,j}}^{P_i}$  を解析結果とする. そうでなければ,  $t$  を  $t'$  に更新した後, (2) に戻る.

$W_{A_{i,j}}^{P_i}$  の解析が終了した後, その解析結果を用いて  $\tau_{i,j}$  の応答時間  $R_{i,j}$  を求める.

$R_{i,j}$  の解析は, 次のような手順で行う.

- (1)  $R_{i,j}$  を  $W_{A_{i,j}}^{P_i} + E_i$  に初期化する.  $A_{i,j}$  において,  $hp(P_i)$  に属するタスクが起動するならば,  $t$  を  $A_{i,j}$  に初期化し, そうでなければ,  $t$  を,  $A_{i,j}$  以降で,  $hp(P_i)$  に属するタスクが起動する最も早い時刻に初期化する.
- (2)  $R_{i,j}$  が  $\tau_i$  の最大応答時間  $R_i^{max}$  となる確率を確認し, その確率が 0 より大きいならば, 解析を終了する. そうでなければ, (3) に進む.
- (3)  $hp(P_i)$  に属するタスクのうち, 時刻  $t$  で起動するタスクの集合を  $At(t)$  とする. ここで,  $At(t)$  に属するタスクの実行時間を,  $R_{i,j}$  のうち,  $t$  を超える分布について加える. つまり,

$$R_{i,j} = \begin{cases} R_{i,j} + \sum_{\forall \tau_k \in At(t)} E_k & (R_{i,j} > t) \\ R_{i,j} & (R_{i,j} \leq t) \end{cases} \quad (18)$$

となる.

- (4) 時刻  $t$  以降で,  $hp(P_i)$  に属するタスクが起動する最も早い時刻を  $t'$  とする.  $t$  を  $t'$  に更新した後, (2) に戻る.

## 5.2 タスクの初期位相が確率変数の場合

タスクの初期位相を確率変数とした場合における応答時間の確率的な解析は, まず, 各タスクの初期位相をある値に固定して, タスクの応答時間分布を求める. その後, 固定した初

期位相の組合せとなる確率を, 求めたタスクの応答時間分布に掛け合わせ, その結果を足し合わせて平均化する. そのため, 各タスクがとりうる初期位相の値すべての組合せに対し, タスクの応答時間分布を解析する必要がある. 解析時間が長くなると考えられる. 短時間で解析を行うための手法として, 本論文では, タスクの応答時間分布の終端部分のみを解析する手法について述べる.

### 5.2.1 応答時間分布の解析

タスクの応答時間分布の終端部分を求めるために, 対象とするタスクが, 高優先度タスクに最大回数実行を妨げられる場合について解析を行う. まず,  $hp(P_i)$  に属するタスク  $\tau_j$  が,  $\tau_i$  の実行を妨げる最大回数  $C_j^{max}$  を求める.  $C_j^{max}$  は,  $\tau_i$  の最大応答時間  $R_i^{max}$  に対し, 次の式で表される.

$$C_j^{max} = \left\lceil \frac{R_i^{max}}{T_j} \right\rceil \quad (19)$$

あるタスクが, 高優先度タスクに最大回数実行を妨げられる場合の例を, 表 1 に示すタスクセットを用いて示す.  $\tau_3$  について考えると,  $\tau_3$  の最大応答時間は 20 であり,  $\tau_3$  が  $\tau_1$ ,  $\tau_2$  に実行を妨げられる最大回数は, それぞれ, 4 回, 2 回である. よって,  $\tau_3$  が  $\tau_1$ ,  $\tau_2$  に最大回数実行を妨げられる場合の 1 つは, 図 2 のように表すことができる.

次に,  $\tau_i$  の応答時間分布の終端部分  $R_i$  を求める. ここで,  $hp(P_i)$  に属するタスク  $\tau_1, \tau_2, \dots, \tau_{i-1}$  の初期位相の組  $(O_1, O_2, \dots, O_{i-1})$  を, 確率変数  $\Psi_i$  とし,  $\Psi_i$  がとりう

表 1 タスクセット 1  
Table 1 A task set No.1.

| task     | priority | period | $E_i^{max}$ |
|----------|----------|--------|-------------|
| $\tau_1$ | 1        | 5      | 2           |
| $\tau_2$ | 2        | 10     | 4           |
| $\tau_3$ | 3        | 20     | 4           |

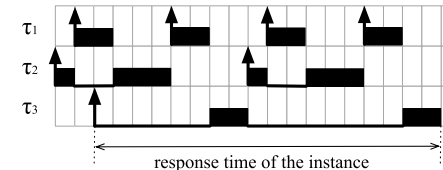


図 2  $\tau_3$  が最大回数実行を妨げられる場合

Fig. 2 A case that  $\tau_3$  is interfered by higher priority tasks at maximum times.

る値の集合を,  $\Omega_i$  とする.  $R_i$  を解析する手順は次のとおりである.

- (1)  $\Omega_i$  に属する任意の  $\psi$  に対し, 次の (2) と (3) の処理を行い,  $\tau_i$  の応答時間分布  $R_i^\psi$  を解析する.
- (2)  $hp(P_i)$  に属する任意の  $\tau_j$  に対し,  $\tau_j$  が,  $\tau_i$  の実行を妨げる回数が  $C_j^{max}$  ならば, (3) へ進む. そうでなければ,  $R_i^\psi = 0$  とする.
- (3)  $\tau_i$  の応答時間分布を 5.1 節で述べた手法により解析する. その解析結果を  $R_i^\psi$  とする.
- (4)  $\Omega_i$  に属する任意の  $\psi$  に対し,  $R_i^\psi$  の解析が終了したら,  $R_i$  を次の式により解析する.

$$f_{R_i}(r) = \sum_{\forall \psi \in \Omega_i} f_{\Psi_i}(\psi) f_{R_i^\psi}(r) \quad (20)$$

### 5.2.2 応答時間分布の信頼区間

提案手法により求めたタスクの応答時間分布の信頼区間について考える. ここで, ある確率変数  $X$  に対し,  $X$  の確率密度が正しいことが保証されており, かつ,  $X$  の確率密度が 0 でない区間を,  $X$  の信頼区間と定義する. すなわち, 実際のタスクの応答時間分布と, 提案手法により求めたタスクの応答時間分布が一致する, 確率密度が 0 でない区間である. 本論文では, 対象とするタスクが, 高優先度タスクに最大回数実行を妨げられる場合について応答時間分布を解析するため, タスクの応答時間分布の信頼区間は, 対象とするタスクが高優先度タスクによって最大回数実行を妨げられる場合の応答時間分布のみが存在する区間である. 提案手法により求めた  $R_i$  の信頼区間  $(R_i^{from}, R_i^{to}]$  について, 次の定理が成立する.

**定理 5.1** 提案手法により求めた  $R_i$  の信頼区間  $(R_i^{from}, R_i^{to}]$  について,  $R_i^{to}$  は,  $R_i^{max}$  であり,  $R_i^{from}$  は,  $R_i^{max} - \min_{\tau_j \in hp(P_i)} E_j^{max}$  である.

**証明** まず,  $R_i^{to}$  は,  $\tau_i$  が高優先度タスクに最大回数実行を妨げられる場合における,  $\tau_i$  の応答時間の最大値であり, それは,  $\tau_i$  の最大応答時間  $R_i^{max}$  である.

次に,  $R_i^{from}$  は,  $\tau_i$  が高優先度タスクに最大回数実行を妨げられる場合以外における,  $\tau_i$  の応答時間の最大値であり, それは,  $\tau_i$  の最大応答時間から,  $hp(P_i)$  に属するタスクを 1 回実行したときの最大実行時間を引いた値の最大値である. よって,  $R_i^{from}$  は,  $\max_{\tau_j \in hp(P_i)} (R_i^{max} - E_j^{max})$  と表されるため,  $R_i^{from}$  は,  $R_i^{max} - \min_{\tau_j \in hp(P_i)} E_j^{max}$  と表すことができる.  $\square$

## 6. 評価実験

本章では, 提案手法の有効性を評価するために行った評価実験について述べる. まず, 実験内容の概要, および, 実験環境について述べる. その後, 実験結果と, 実験結果に対する

考察を述べる.

### 6.1 実験の概要

5 章では, 応答時間分布を代数的に解析する手法について述べたが, 本提案手法は, 解析式が複雑であり, 代数的に解析することはできない. そこで, 提案手法の妥当性を評価するために, 提案手法による解析において, 離散化した確率分布を用い, 数値解析によって応答時間分布を求め, その結果をモンテカルロシミュレーションと比較することとした. 評価実験では, あるタスクセットを対象とし, そのタスクセット内で優先度が最低であるタスクの応答時間分布を解析した. タスクの応答時間分布解析は, モンテカルロシミュレーションによる解析と, 提案手法による解析をそれぞれ行い, その結果得られた応答時間分布と, 解析に要した時間を, それぞれ比較した. 評価に用いたプログラムは, すべて C 言語で実装し, 2.93 GHz のプロセッサ, 4 GB のメモリを搭載したマシンを用いて実行した. 解析に要した時間は, シミュレータ, 提案手法による解析のそれぞれを実装したプログラムの処理に要した時間を, time コマンドを用いて取得した値とした.

応答時間分布の比較には, 応答時間分布の累積確率分布を 1 から引いた値の常用対数を用いた. この値は, 対応する時間  $t$  を, 応答時間が超える累積確率を表しており,  $t$  をデッドラインと仮定すると, デッドラインミス率を表す.

本論文で述べた提案手法は, 既存手法<sup>4)</sup> を拡張したものであり, 前提条件が既存手法と同じ (初期位相が定数として与えられる) タスクセットに関しては, 既存手法を用いて応答時間分布を解析する. そのため, 既存手法で解析可能なタスクセットを提案手法により解析した場合, 解析時間は, 既存手法と提案手法でほぼ同じになる. そこで, 本評価実験では, タスクの初期位相が確率変数として与えられるタスクセットのみを対象とする.

また, 提案手法について, タスクの実行時間分布と初期位相分布を離散化することによる影響を評価するため, 実行時間分布の離散化の刻み幅 ( $\Delta_e$ ) を変動させた場合, および, 初期位相分布の離散化の刻み幅 ( $\Delta_o$ ) を変動させた場合の解析結果をそれぞれ比較した.

対象とするタスクセットには, タスクを 4 つから 7 つ含むタスクセットを, それぞれ 10 タスクセット (合計 40 タスクセット) 用いた. 各タスクの周期は, 乱数を用いて生成した, 10 から 100 の間の整数とした. そして, 各タスクの実行時間分布は, 区間  $[E_i^{min}, E_i^{max}]$  の切断正規分布に従うものとし, 基となる正規分布の平均と分散をそれぞれ,  $\mu_{E_i}$ ,  $\sigma_{E_i}^2$  とした. ここで, 切断正規分布とは, 確率変数の定義域が有限の確率分布である. 基となる正規分布が  $\hat{X}$  で, 区間  $[a, b]$  の切断正規分布に従う確率変数  $X$  の確率密度関数  $f_X$  は, 次の式で定義される.

表 2 評価対象のタスクセットの 1 つ  
Table 2 One of task sets for evaluation.

| task     | priority | period | $E_i$       |             |             |                |
|----------|----------|--------|-------------|-------------|-------------|----------------|
|          |          |        | $E_i^{max}$ | $E_i^{min}$ | $\mu_{E_i}$ | $\sigma_{E_i}$ |
| $\tau_1$ | 1        | 13     | 1.897       | 1.0         | 1.546       | 1.5            |
| $\tau_2$ | 2        | 38     | 6.355       | 1.0         | 4.689       | 1.5            |
| $\tau_3$ | 3        | 48     | 4.014       | 1.0         | 3.889       | 1.5            |
| $\tau_4$ | 4        | 49     | 3.439       | 1.0         | 2.694       | 1.5            |
| $\tau_5$ | 5        | 59     | 2.195       | 1.0         | 2.131       | 1.5            |
| $\tau_6$ | 6        | 71     | 10.42       | 1.0         | 8.205       | 1.5            |
| $\tau_7$ | 7        | 73     | 1.297       | 1.0         | 1.089       | 1.5            |

$$f_X(x) = \begin{cases} \frac{f_{\hat{X}}(x)}{P(a \leq \hat{X} \leq b)} & (a \leq x \leq b) \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

$E_i^{max}$  と  $\mu_{E_i}$  は、乱数を用いて生成した実数とし、 $E_i^{min}$  と  $\sigma_{E_i}$  は、それぞれ、1.0、1.5 とした。

本章では、評価実験を行ったタスクセットのうち、表 2 に示すタスクセットについて、詳細に結果を述べる。表 2 のタスクセットについては、 $\tau_4$ 、 $\tau_5$ 、 $\tau_6$ 、 $\tau_7$  の応答時間分布を解析した。

シミュレータは、入力として、タスクセットの情報、応答時間計測の対象とするタスク、シミュレーションの試行回数、出力する離散応答時間分布の刻み幅を与える。そして、モンテカルロシミュレーションにより、計測対象タスクの離散応答時間分布を解析し、結果として、確率分布、および、累積確率分布を出力する。本実験では、シミュレーションの試行回数は、 $10^9$  回とし、出力する離散応答時間分布の刻み幅は、0.1 とした。

## 6.2 解析時間の比較

表 2 のタスクセットについて、各タスクの応答時間分布解析に要した時間を、表 3 と表 4 に示す。

表 3 は、 $\Delta_o$  を 1.0 に固定し、 $\Delta_e$  を 0.1、0.2、0.4 とした場合の解析時間を、タスクセットに含まれるタスク数ごとにまとめたものである。そして、表 4 は、 $\Delta_e$  を 0.1 に固定し、 $\Delta_o$  を 1.0、2.0、4.0 とした場合の解析時間を、タスクセットに含まれるタスク数ごとにまとめたものである。ここで、表 3 と表 4 の一番下の行は、シミュレータによる解析時間を示している。そして、それぞれの表中における解析時間の単位は秒である。

表 3 と表 4 をみると、提案手法を用いて解析した場合、シミュレータを用いて解析した

表 3  $\Delta_e$  を変化させた場合の解析時間の比較  
Table 3 Analysis time ( $\Delta_o$  is constant).

|            |     | タスク数     |          |          |          |
|------------|-----|----------|----------|----------|----------|
|            |     | 4        | 5        | 6        | 7        |
| $\Delta_e$ | 0.1 | 0.151    | 1.207    | 702.416  | 8950.128 |
|            | 0.2 | 0.102    | 0.446    | 203.220  | 2605.243 |
|            | 0.4 | 0.008    | 0.212    | 64.910   | 871.049  |
| シミュレータ     |     | 3058.291 | 4046.404 | 5992.767 | 6993.226 |

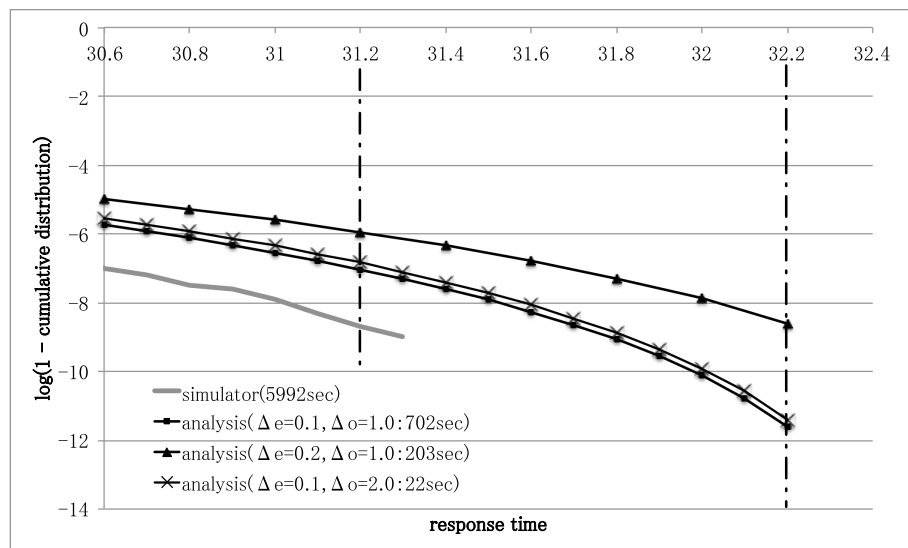
表 4  $\Delta_o$  を変化させた場合の解析時間の比較  
Table 4 Analysis time ( $\Delta_e$  is constant).

|            |     | タスク数     |          |          |          |
|------------|-----|----------|----------|----------|----------|
|            |     | 4        | 5        | 6        | 7        |
| $\Delta_o$ | 1.0 | 0.151    | 1.207    | 702.416  | 8950.128 |
|            | 2.0 | 0.111    | 0.157    | 22.268   | 130.637  |
|            | 4.0 | 0.093    | 0.086    | 1.042    | 3.747    |
| シミュレータ     |     | 3058.291 | 4046.404 | 5992.767 | 6993.226 |

場合よりも、解析時間を大きく短縮できていることが分かる。ここで、タスク数が 7 つであり、 $\Delta_e = 0.1$ 、 $\Delta_o = 1.0$  の場合の提案手法による解析時間は、シミュレータによる解析時間よりも長くなっているが、これは、シミュレータの試行回数が少ないことが原因である。タスク数が 7 つの場合の応答時間分布をみると、シミュレータによって得られた最大応答時間は、32.2 であり、そのときの累積確率は、 $10^{-8.699}$  であった。一方、提案手法によって得られた最大応答時間は、33.5 であり、そのときの累積確率は、 $10^{-12.741}$  であった。よって、タスク数が 7 つの場合に、シミュレータにより、応答時間分布の終端部分を解析するには、本実験の約 1,000 倍の試行回数が必要であると考えられるため、提案手法を用いることで、解析時間を大きく短縮できているといえる。

表 3 をみると、 $\Delta_e$  を大きくしたときの解析時間が短くなっていることが分かる。これは、 $\Delta_e$  が畳込み積分の計算量に影響していることが原因であると考えられる。 $\Delta_e$  を小さくすると、畳込み積分の対象となる 2 つの離散確率分布関数の刻み幅がそれぞれ小さくなり、計算する組合せが多くなる。具体的には、全体の計算量は、最大で、 $1/\Delta_e^2$  に比例して多くなると考えられる。よって、提案手法では、 $\Delta_e$  を大きくすることによって、解析に要する時間を短くできたと考えられる。

表 4 をみると、 $\Delta_o$  を大きくしたときの解析時間が短くなっていることが分かる。これは、 $\Delta_o$  が初期位相の組合せの数に影響していることが原因であると考えられる。 $\Delta_o$  を小さく

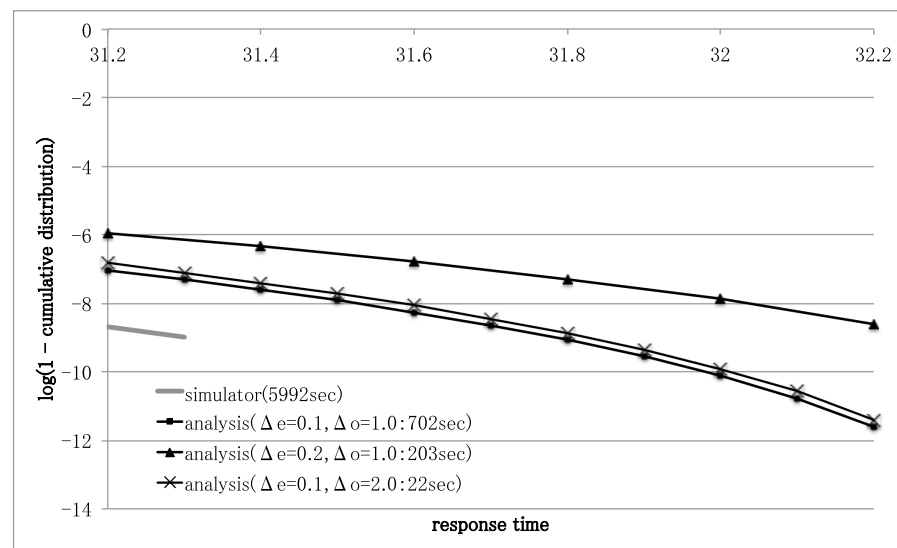
図3  $\tau_6$  の応答時間分布Fig. 3 The distribution of  $\tau_4$ 's response time.

すると、初期位相の組合せが多くなり、応答時間分布を解析する回数が増える。具体的には、全体の計算量は、最大で、解析の対象となるタスクの実行を妨げるタスク数  $N$  に対し、 $1/\Delta_o^N$  に比例して多くなると考えられる。よって、提案手法では、タスク数が増えるにつれて解析に要する時間が急激に長くなるが、 $\Delta_o$  を大きくすることによって、解析に要する時間を短くできたと考えられる。

すべての評価対象のタスクセットについて確認したところ、解析時間について、表2以外のタスクセットについても、表2のタスクセットの場合と同様の傾向がみられることが分かった。

### 6.3 応答時間分布の比較

図3と図4は、表2の $\tau_6$ について、得られた応答時間分布をまとめ、その終端部分を拡大したものである。ここで、図4は、提案手法によって得られる応答時間分布の信頼区間内における、応答時間分布の比較結果を示している。図3、図4のグラフの横軸はタスクの応答時間を、縦軸は応答時間分布の累積確率分布を1から引いた値の常用対数をそれぞれ表している。また、図3における2つの鎖線は、提案手法によって得られる応答時間分

図4 信頼区間における $\tau_6$ の応答時間分布Fig. 4 The distribution of  $\tau_4$ 's response time in the confidential interval.

布の信頼区間  $[31.2, 32.2]$  の上限と下限を表している。

提案手法によって得られた応答時間分布 (analysis) と、シミュレータによって得られた応答時間分布 (simulator) を比較すると、両者の分布が近似していることが分かる。ここで、信頼区間の範囲内をみると、analysis が simulator を上回っていることが分かる。これは、提案手法において、タスクの実行時間分布と初期位相分布を離散化した際に、得られる応答時間分布が悲観的になるように離散化誤差を丸めたためである。そして、信頼区間の範囲外をみると、信頼区間から遠ざかるにつれ、analysis と simulator が乖離していることが分かる。これは、信頼区間の範囲外では、 $\tau_6$  が他のタスクに最大回数実行を妨げられる場合以外の応答時間分布が含まれるためである。また、応答時間 31.3 付近をみると、simulator の分布が途切れていることが分かる。これは、simulator によって得られる確率の限界 ( $10^{-9}$ ) を超えてしまい、正確な応答時間分布を出力できなかったためである。

analysis について、 $\Delta_e$  と  $\Delta_o$  を変動させた場合をそれぞれ比較する。比較項目としては、応答時間分布に関する誤差の最大値と平均値を用いた。

表2のタスクセットについて、解析によって得られた各タスクの応答時間分布に関する

表 5  $\Delta_e$  を変化させた場合における応答時間分布の誤差の比較  
Table 5 Error in response time distribution ( $\Delta_o$  is constant).

|            |     | タスク数          |               |               |               |
|------------|-----|---------------|---------------|---------------|---------------|
|            |     | 4             | 5             | 6             | 7             |
| $\Delta_e$ | 0.1 | 2.105 / 1.283 | 1.535 / 1.333 | 1.691 / 1.675 | -             |
|            | 0.2 | 1.667 / 1.080 | 1.926 / 1.103 | 2.994 / 1.807 | 3.076 / 2.209 |
|            | 0.4 | 3.062 / 2.317 | 3.425 / 2.477 | 5.162 / 3.848 | 5.977 / 4.646 |

表 6  $\Delta_o$  を変化させた場合における応答時間分布の誤差の比較  
Table 6 Error in response time distribution ( $\Delta_e$  is constant).

|            |     | タスク数          |               |               |               |
|------------|-----|---------------|---------------|---------------|---------------|
|            |     | 4             | 5             | 6             | 7             |
| $\Delta_o$ | 1.0 | 2.105 / 1.283 | 1.535 / 1.333 | 1.691 / 1.675 | -             |
|            | 2.0 | 0.294 / 0.294 | 0.307 / 0.307 | 0.201 / 0.201 | 0.361 / 0.361 |
|            | 4.0 | 0.793 / 0.793 | 0.826 / 0.826 | 0.628 / 0.628 | 1.008 / 1.008 |

誤差の最大値と平均値を、表 5 と表 6 に示す。

表 5 は、 $\Delta_o$  を 1.0 に固定し、 $\Delta_e$  を 0.1, 0.2, 0.4 とした場合の実験結果を、タスクセットに含まれるタスク数ごとにまとめたものである。そして、表 6 は、 $\Delta_e$  を 0.1 に固定し、 $\Delta_o$  を 1.0, 2.0, 4.0 とした場合の実験結果を、タスクセットに含まれるタスク数ごとにまとめたものである。ここで、表 5 と表 6 において、応答時間分布に関する誤差は、常用対数軸上での差分を記載している。そして、それぞれの表中では、応答時間分布に関する誤差を、(最大値)/(平均値)と表記しており、誤差を求めることができなかった場合には、“-”と表記している。また、それぞれの表中において、 $\Delta_e = 0.1$ 、かつ、 $\Delta_o = 1.0$  のときの値は、シミュレータの出力した応答時間分布との誤差を示しており、それ以外のときの値は、提案手法を用いて、 $\Delta_e = 0.1$ 、かつ、 $\Delta_o = 1.0$  とした場合に解析された応答時間分布との誤差を示している。すべての提案手法について、シミュレータの出力した応答時間分布と比較しなかった理由は、シミュレータの出力した応答時間分布の終端部分について、分布が得られなかった部分があったためである。

$\Delta_e$  が小さい場合と大きい場合とを比較すると、 $\Delta_e$  が小さい場合よりも、 $\Delta_e$  が大きい場合の方が、応答時間分布が大きくなっている。また、すべてのタスクセットにおいて、信頼区間内では、提案手法による解析結果の応答時間分布が、シミュレーション結果の応答時間分布を上回っており、 $\Delta_e$  が小さい場合よりも、 $\Delta_e$  が大きい場合の方が、応答時間分布が

大きくなっていることを確認した。この結果から、提案手法では、解析結果の応答時間分布が悲観的になるように、実行時間分布を離散化できていることを確認でき、さらに、実行時間分布の離散化の刻み幅を大きくすることによって、解析結果の応答時間分布がより悲観的になることが分かった。

$\Delta_o$  が小さい場合と大きい場合とを比較すると、 $\Delta_o$  が小さい場合よりも、 $\Delta_o$  が大きい場合の方が、応答時間分布が大きくなっている。また、すべてのタスクセットにおいて、信頼区間内では、提案手法による解析結果の応答時間分布が、シミュレーション結果の応答時間分布を上回っており、 $\Delta_o$  が小さい場合よりも、 $\Delta_o$  が大きい場合の方が、応答時間分布が大きくなっていることを確認した。この結果から、提案手法では、解析結果の応答時間分布が悲観的になるように、初期位相分布を離散化できていることを確認でき、さらに、初期位相分布の離散化の刻み幅を大きくすることによって、解析結果の応答時間分布がより悲観的になることが分かった。

#### 6.4 終端部分のみを解析することの効果

提案手法において、応答時間分布の終端部分のみを解析することによる、解析の高速化の有効性を評価するために、比較実験を行った。表 2 の  $\tau_6$  について、提案手法によって、応答時間分布の終端部分のみを解析した結果と、提案手法から、5.2.1 項で述べた、解析を行う場合の限定条件を取り除き、応答時間分布全体を解析した場合の結果とを比較した。ここで、離散化の刻み幅については、 $\Delta_e = 0.2$ 、かつ、 $\Delta_o = 1.0$  として解析した。

その結果、解析時間については、応答時間分布の終端部分のみを解析した場合は表 3 のとおり、203.220 秒であった。一方、応答時間分布全体を解析した場合は、2553.309 秒であり、終端部分のみを解析した場合の約 13 倍であった。このように、応答時間分布の終端部分のみを解析することによって、解析時間を大きく短縮できていることが確認できた。また、2 つの解析手法によって得られた応答時間分布を比較した結果、信頼区間内において、両者の応答時間分布が一致していることを確認した。

以上の結果から、提案手法を用いることで、タスクの応答時間分布の終端部分について、短時間で、悲観的な解析ができることを示した。そして、提案手法では、タスク数が増えるにつれて解析に要する時間が急激に長くなるが、タスクの実行時間分布と初期位相分布の離散化の刻み幅を大きくし、悲観的に応答時間分布を解析することによって、解析に要する時間を短くできることを示した。

## 7. おわりに

本論文では、タスクの実行時間に加えて、初期位相を確率変数として扱い、タスクの応答時間分布を解析する手法を提案した。提案手法では、タスクの実行時間分布と初期位相分布を離散化して扱い、さらに、応答時間分布の終端部分に着目して数学的に解析することによって、解析時間の短縮を実現した。タスクの実行時間分布と初期位相分布の離散化について、離散化した確率分布を用いて解析した結果の応答時間分布が、正確な応答時間分布よりも悲観的に近似されるように離散化誤差を丸める手法を提案し、その手法により、応答時間分布が悲観的に近似されることを証明した。応答時間分布の終端部分に着目した解析について、解析を行う条件を提案し、応答時間分布の終端部分を解析する手法を提案した。そして、提案した解析手法によって得られる応答時間分布の信頼区間を示した。提案手法の有効性を評価するための実験を行った。その結果、提案手法を用いることで、タスクの応答時間分布の終端部分を短時間で、悲観的に解析できることを示した。そして、提案手法では、タスク数が増えるにつれて解析に要する時間が長くなるが、タスクの実行時間分布と初期位相分布の離散化の刻み幅を大きくし、解析結果の応答時間分布をより悲観的にすることによって、解析に要する時間を短くできることを示した。

今後の課題として、異なるタスクセットのモデルへの対応が考えられる。今回扱ったシステムは、周期タスクのみが存在するタスクセットを対象としており、プリエンティブな優先度ベースのスケジューリングのみを対象としている。そこで、タスク間の依存関係や割り込みを考慮する場合や、スケジューリングポリシーの異なる場合についての解析手法を提案し、より複雑なタスクセットにおける解析手法を提案することが必要であると考えられる。

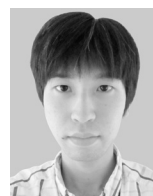
## 参考文献

- 1) Bernat, G., Burns, A. and Llamasí, A.: Weakly Hard Real-Time Systems, *IEEE Trans. Comput.*, Vol.50, No.4, pp.308–321 (2001).
- 2) Díaz, J.L., García, D.F., Kim, K., et al.: Stochastic Analysis of Periodic Real-Time Systems, *Proc. 23rd IEEE Real-Time Systems Symposium*, p.289 (2002).
- 3) Díaz, J.L., López, J.M., García, M., et al.: Pessimism in the Stochastic Analysis of Real-Time Systems: Concept and Applications, *Proc. 25th IEEE International Real-Time Systems Symposium*, pp.197–207 (2004).
- 4) Kim, K., Díaz, J.L., Bello, L.L., et al.: An Exact Stochastic Analysis of Priority-Driven Periodic Real-Time Systems and Its Approximations, *IEEE Trans. Comput.*, Vol.54, No.11, pp.1460–1466 (2005).

- 5) Lehoczky, J.P.: Fixed priority scheduling of periodic task sets with arbitrary deadlines, *Proc. 11th IEEE Real-Time Systems Symposium*, pp.201–209 (1990).
- 6) Liu, C.L. and Layland, J.W.: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *J. ACM*, Vol.20, No.1, pp.46–61 (1973).
- 7) López, J.M., Díaz, J.L., Entrialgo, J. and García, D.: Stochastic analysis of real-time systems under preemptive priority-driven scheduling, *Real-Time Systems*, Vol.40, No.2, pp.180–207 (2008).
- 8) Zeng, H.: Probabilistic Timing Analysis of Distributed Real-time Automotive Systems, Technical Report, University of California, Berkeley (2008).
- 9) Zeng, H., Natale, M.D., Giusto, P. and Sangiovanni-Vincentelli, A.: Stochastic Analysis of CAN-Based Real-Time Automotive Systems, *IEEE Trans. Industrial Informatics*, Vol.5, No.4, pp.388–401 (2009).

(平成 23 年 3 月 1 日受付)

(平成 23 年 9 月 12 日採録)



石川 拓也

2011 年名古屋大学大学院情報科学研究科博士課程前期課程修了。現在、同博士課程後期課程に在学。リアルタイムスケジューリング理論、リアルタイム OS、組み込みシステム向けコンポーネント技術の研究に従事。



松原 豊 (正会員)

名古屋大学大学院情報科学研究科附属組み込みシステム研究センター研究員。2006 年名古屋大学大学院情報科学研究科博士前期課程修了。2009 年同博士後期課程単位取得満期退学。2009 年 4 月より現職。リアルタイム OS、リアルタイムスケジューリング理論、組み込みシステム向けの安全分析手法に関する研究に従事。博士 (情報科学)。



高田 広章（正会員）

名古屋大学大学院情報科学研究科情報システム学専攻教授。1988年東京大学大学院理学系研究科情報科学専攻修士課程修了。同専攻助手，豊橋技術科学大学情報工学系助教授等を経て，2003年より現職。2006年より大学院情報科学研究科附属組込みシステム研究センター長を兼務。リアルタイムOS，リアルタイムスケジューリング理論，組み込みシステム開発技術等の研究に従事。オープンソースのITRON仕様OS等を開発するTOPPERSプロジェクトを主宰。博士（理学）。IEEE，ACM，電子情報通信学会，日本ソフトウェア科学会各会員。

---