

分散リアルタイムシステムの端点間処理における 応答時間の確率的解析

石川 拓也^{†1} 松原 豊^{†2} 高田 広章^{†1,†2}

分散リアルタイムシステムにおいて、システムに入力が与えられてから、最終的な処理結果を出力するまでの一連の処理を端点間処理と呼ぶ。分散リアルタイムシステムの開発では、端点間処理の応答時間が、与えられた時間制約を満たすかどうかを検証することが必要であり、端点間処理の応答時間解析は重要なプロセスである。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解析を確率的に行う手法が提案されている。一般的に、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。本論文では、タスクの実行時間と初期位相を確率変数として扱い、タスクの応答時間分布、および、端点間処理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、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、解析時間を大幅に削減する。

Stochastic Analysis of End-to-End Latency for Distributed Real-Time Systems

TAKUYA ISHIKAWA,^{†1} YUTAKA MATSUBARA^{†2}
and HIROAKI TAKADA^{†1,†2}

In distributed real-time systems, an end-to-end computation is a sequence of computations from getting inputs to outputting results. In developing distributed real-time systems, it is necessary to verify that latency of end-to-end computations meets timing constraints. Therefore, it is important to analyze latency of end-to-end computations. 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. In this paper, we propose a stochastic analysis method for both of response times of tasks and end-to-end latency, using random execution times and random initial phases of tasks. By using the method, analysis time is drastically reduced by focusing only on analyzing the end of the response time distribution.

1. はじめに

分散リアルタイムシステムとは、ネットワークやバスなどで接続された、複数のプロセッサによって構成されるリアルタイムシステムである。ここで、分散リアルタイムシステムにおいて、システムに入力が与えられてから、プロセッサ間でデータ通信を行いながら処理を行い、最終的な処理結果を出力するまでの一連の処理を端点間処理と呼ぶこととする。分散リアルタイムシステムの開発では、システムに入力が与えられてから、端点間処理が終了するまでの時間（応答時間）が、与えられた時間制約を満たすかどうかを検証する必要があるため、端点間処理の応答時間解析は重要なプロセスである。

従来のリアルタイムシステムにおける応答時間解析は、タスクの最大実行時間など、一意的で悲観的な値を用いて解析を行う。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 1) や文献 2) 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、実行時間が最大に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^{3),4)}。リアルタイムシステムに対する解析結果が悲観である場合、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。

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解析を確率的に行う手法が提案されている。一般的に、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、複雑さを軽減した、確率的解析手法が提案されている。文献 5), 6) 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、プリエンプ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解析する手法を提案している。そして、文献 4), 7) では、文献 5), 6) 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解析する手法を提案している。

本論文では、タスクの実行時間と初期位相を確率変数として扱い、タスクの応答時間分布、および、端点間処理の応答時間分布を解析する手法を提案する。タスクの初期位相を確

^{†1} 名古屋大学大学院情報科学研究科

Graduate School of Information Science, Nagoya University

^{†2} 名古屋大学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

Center for Embedded Computing Systems, Nagoya University

率変数として扱う場合、各タスクの初期位相が取り得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り、解析時間が長くなると考えられる。そのため、提案する手法では、解析の対象とするタスク、または、端点間処理の実行中において、タスクの起動回数が最大となる場合についてのみ解析を行い、応答時間分布の終端部分を解析する。このように、解析する場合を限定することによって、解析時間を短縮する。

本論文の構成は次の通りである。まず、2章で関連研究について紹介する。3章では、本論文で扱う分散リアルタイムシステムのモデルについて述べる。4章では、タスクの応答時間の確率的な解析について述べる。5章では、端点間処理の応答時間の確率的な解析について述べる。そして、6章では、提案手法についての評価を行う。最後に、7章で本論文のまとめを行う。

2. 関連研究

文献 8) 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な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 5), 6) 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、プリエンプ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 3), 9) では、文献 5), 6) 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、確率を悲観的に見積もるための手法を提案している。さらに、文献 5), 6) 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見積もる手法を示している。

文献 4), 7) 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 5), 6) で提案されている手法をベースとして

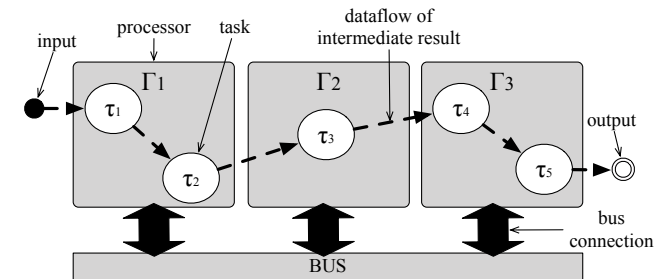


図 1 分散リアルタイムシステムのモデル
Fig. 1 A model of the distributed real-time system.

用いており、加えて、ノンプリエンプ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見積も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。

3. システムのモデル

本論文で扱う分散リアルタイムシステムは、図 1 に示されるように、複数のプロセッサで構成されており、バスによってプロセッサ間が接続されているシステムである。各プロセッサには、複数の周期タスクが存在するものとし、プロセッサ内におけるタスクセットは、ハーモニックタスクセットとする。ハーモニックタスクセットとは、タスクセットに含まれる任意の異なる 2 つのタスクに対し、一方のタスクの起動周期が、他方のタスクの起動周期の整数倍となるようなタスクセットである。また、プロセッサ内におけるタスクのスケジューリング方式は、Rate Monotonic Scheduling¹⁾ とする。

本システムでの端点間処理は、次のような流れで行われる。まず、システムに入力データが与えられる。そして、入力データがあるタスクが受信し、処理を施した後で中間結果を出力し、その結果を別のタスクが受信する。このような処理を繰り返し、最終的な結果を出力する。ここで、データの受信はタスクの実行開始時に行うものとし、中間結果の出力はタスクの終了時に起こるものとする。また、データの送受信にかかる時間は無視できるものとする。

データの流れについては、タスクの実行順序で表すこととし、実行順の早いタスクから順

に、 $\tau_1, \tau_2, \dots, \tau_n$ と表すものとする。ここで、任意の i, j に対し、 $i \neq j$ ならば、 τ_i と τ_j は異なるタスクであるものとし、端点間処理の中で同一のタスクが2回以上処理を行うことはないものとする。そして、任意の i に対し、 τ_i の所属するプロセッサと τ_{i+1} の所属するプロセッサが異なるならば、 i より大きい任意の j に対し、 τ_i の所属するプロセッサと τ_j の所属するプロセッサは異なるものとする。また、同一のプロセッサに所属する2つのタスク τ_i, τ_j に対し、 $i < j$ ならば、 τ_i の優先度は、 τ_j の優先度よりも高いものとし、プロセッサ内では、優先度の高いタスクから優先度の低いタスクにデータが流れるものとする。

タスク τ_i は、 $(N_i, P_i, T_i, E_i, O_i)$ というパラメータを持つものとする。 N_i, P_i, T_i は定数とし、それぞれ、 τ_i の所属するプロセッサの ID、 τ_i の優先度、 τ_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、 τ_i の実行時間、 τ_i の初期位相を示す。ここで、初期位相とは、システムの起動後に初めて τ_i が起動する時刻である。 E_i の分布は、区間 $[E_i^{min}, E_i^{max}]$ の切断正規分布に従うものとし、その確率密度関数を f_{E_i} とする。 O_i の分布は、区間 $[0, T_i)$ の一様分布に従うものとし、その確率密度関数を f_{O_i} とする。また、タスク τ_i の最大応答時間 R_i^{max} は、 τ_i の起動周期 T_i 以下であるとする。

4. 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。

4.1 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6)で提案されている手法に基づいて行う。タスクの応答時間を解析するために、まず、対象とするタスク τ_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、もしくは、 P_i と等し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、もしくは、 P_i と等しいタスクの集合を $hep(P_i)$ 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、 $hep(P_i)$ 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、 $hep(P_i)$ 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) $hep(P_i)$ に属するタスクの内、時刻 t で起動するタスクの集合を $At(t)$ とする。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、

$$W_{A_{i,j}}^{P_i} = W_{A_{i,j}}^{P_i} + \sum_{\forall \tau_k \in At(t)} E_k \quad (1)$$

となる。

- (3) 時刻 t 以降で、 $hep(P_i)$ に属するタスクが起動する最も早い時刻を t' とする。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、 $W_{A_{i,j}}^{P_i}$ の確率密度関数を $f_{W_{A_{i,j}}^{P_i}}(w)$ とすると、

$$f_{W_{A_{i,j}}^{P_i}}(w) = \begin{cases} f_{W_{A_{i,j}}^{P_i}}(w + (t' - t)) & (w > 0) \\ \sum_{w' = -\infty}^{t' - t} f_{W_{A_{i,j}}^{P_i}}(w') & (w = 0) \\ 0 & (w < 0) \end{cases} \quad (2)$$

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。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。

$R_{i,j}$ 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) τ_i の最大応答時間を R_i^{max} とする。ここで、 $R_{i,j}$ が R_i^{max} となる確率が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) $hp(P_i)$ に属するタスクの内、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、 $R_{i,j}$ の内、 t を超える分布について加える。つまり、

$$R_{i,j} = \begin{cases} R_{i,j} + \sum_{\forall \tau_k \in At(t)} E_k & (R_{i,j} > t) \\ R_{i,j} & (R_{i,j} \leq t) \end{cases} \quad (3)$$

表 1 タスクセット 1
Table 1 A task set No.1.

task	processor	priority	period	E_i^{max}
τ_1	Γ_1	1	5	2
τ_2	Γ_1	2	10	4
τ_3	Γ_1	3	20	4

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.

4.2 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, 固定した初期位相になる確率を, 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, 各タスクが取り得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, 解析時間が膨大になると考えられる. 短時間で解析を行うための手法として, ここでは, 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.

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 τ_j が, τ_i の実行を妨げる最大回数 C_j^{max} を求める. C_j^{max} は, τ_i の最大応答時間 R_i^{max} に対し, 次の式で表される.

$$C_j^{max} = \left\lceil \frac{R_i^{max}}{T_j} \right\rceil \quad (4)$$

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. τ_3 について考えると, τ_3 の最大応答時間は 20 であり, τ_3 が τ_1, τ_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, τ_3 が τ_1, τ_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, τ_i の応答時間分布 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 Ψ_i とし, Ψ_i が取り得る値の集合を, Ω_i とする. R_i を解析する手順は次のとおりである.

- (1) Ω_i に属する任意の ψ に対し, 次の (2) と (3) の処理を行い, τ_i の応答時間分布 R_i^ψ

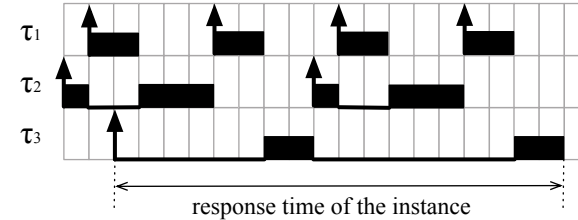


図 2 τ_3 が最大回数実行を妨げられる場合

Fig. 2 A case that τ_3 is interfered by higher priority tasks at maximum times.

を解析する.

- (2) $hp(P_i)$ に属する任意の τ_j に対し, τ_j が, τ_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) τ_i の応答時間分布を 4.1 節で述べた手法により解析する. その解析結果を R_i^ψ とする.
- (4) Ω_i に属する任意の ψ に対し, R_i^ψ の解析が終了したら, R_i を次の式により解析する.

$$f_{R_i}(r) = \sum_{\forall \psi \in \Omega_i} f_{\Psi_i}(\psi) f_{R_i^\psi}(r) \quad (5)$$

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, 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, タスク τ_i の応答時間分布 R_i の信頼区間 $(R_i^{f^{rom}}, R_i^{t^{to}}]$ について, 次の定理が成立する.

定理 1 提案手法により求めた, タスク τ_i の応答時間分布 R_i の信頼区間 $(R_i^{f^{rom}}, R_i^{t^{to}}]$ について, $R_i^{t^{to}}$ は, R_i^{max} であり, $R_i^{f^{rom}}$ は, $R_i^{max} - \min_{\tau_j \in hp(P_i)} E_j^{max}$ である.

証明 まず, $R_i^{t^{to}}$ は, τ_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, τ_i の応答時間の最大値であり, それは, τ_i の最大応答時間 R_i^{max} である.

次に, $R_i^{f^{rom}}$ は, τ_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, τ_i の応答時間の最大値であり, それは, τ_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。□

5. 端点間処理の応答時間解析

本章では、まず、プロセッサ内における端点間処理の応答時間に対する確率的な解析手法について述べる。次に、プロセッサ間を含めた端点間処理の応答時間に対する確率的な解析手法について述べる。

5.1 プロセッサ内における端点間処理

プロセッサ内における端点間処理の応答時間とは、対象とするプロセッサにおいて、他プロセッサから出力されたデータを受信するタスクが起動する時刻から、最後にデータを出力するタスクが終了する時刻までにかかる時間を意味する。プロセッサ内における端点間処理の応答時間の確率的な解析については、まず、プロセッサ内のすべてのタスクの初期位相をある値に固定して、データを送信するタスクが起動する時刻と、データを受信するタスクが起動する時刻の差をすべて加算する。そして、その加算結果に対し、プロセッサ内において、最後にデータを受信するタスクの応答時間を加算することにより、固定した初期位相に対する、端点間処理の応答時間分布を求める。その後、固定した初期位相になる確率を、求めた端点間処理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。また、4.2 節と同様、ここでも、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。

プロセッサ内における端点間処理の応答時間分布の終端部分を求めるために、次に示す 2 つの条件 Cond1 と Cond2 を同時に満たす場合について解析を行う。

Cond1. プロセッサ内において、最後にデータを受信するタスクが、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。

Cond2. データを送信するタスクが起動する時刻と、データを受信するタスクが起動する時刻の差が、それら 2 つのタスクの周期の差より大きい場合について解析を行う。

Cond1 と Cond2 を同時に満たす場合の例を、表 2 に示すタスクセットを用いて示す。Cond1 を満たす場合は、 τ_3 が τ_1 に 2 回、 τ_2 に 1 回実行を妨げられる場合である。Cond2 を満たす場合は、 τ_1 の起動時刻と τ_2 の起動時刻の差が 4 より大きく、また、 τ_2 の起動時刻と τ_3 の起動時刻の差が 8 より大きい場合である。よって、表 2 のタスクセットにおいて、Cond1 と Cond2 を同時に満たすような場合の 1 つは、図 3 のように表すことができる。

表 2 タスクセット 2
Table 2 A task set No.2.

task	processor	priority	period	E_i^{max}
τ_1	Γ_1	1	4	1
τ_2	Γ_1	2	8	3
τ_3	Γ_1	3	16	3

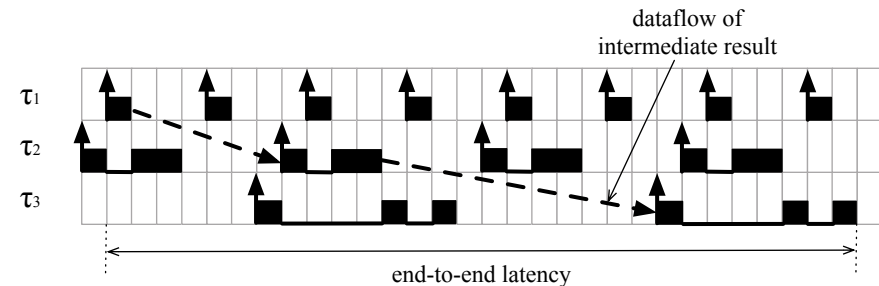


図 3 Cond1 と Cond2 を同時に満たす端点間処理
Fig. 3 A end-to-end computation satisfying Cond1 and Cond2.

Cond1 と Cond2 を同時に満たす場合について、プロセッサ Γ_i 内における端点間処理の応答時間分布 L_{Γ_i} を求める。ここで、プロセッサ Γ_i に属するタスク $\tau_{k_i+1}, \tau_{k_i+2}, \dots, \tau_{k_i+n_i}$ の初期位相の組 $(O_{k_i+1}, O_{k_i+2}, \dots, O_{k_i+n_i})$ を、確率変数 Ψ_{Γ_i} とし、 Ψ_{Γ_i} が取り得る値の集合を、 Ω_{Γ_i} とする。 L_{Γ_i} を解析する手順は次のとおりである。

- (1) Ω_{Γ_i} に属する任意の ψ に対し、次の (2) から (4) の処理を行い、プロセッサ Γ_i 内における端点間処理の応答時間分布 $L_{\Gamma_i}^{\psi}$ を解析する。
- (2) Cond1 と Cond2 を同時に満たすならば、(3) へ進む。そうでなければ、 $L_{\Gamma_i}^{\psi} = 0$ とする。
- (3) 4.1 節で述べた手法により、 Γ_i 内で最後にデータを受信するタスク $\tau_{k_i+n_i}$ の応答時間分布 $R_{k_i+n_i}^{\psi}$ を解析する。その後、(4) へ進む。
- (4) $R_{k_i+n_i}^{\psi}$ に対し、プロセッサ Γ_i におけるデータを送信するタスクが起動する時刻と、データを受信するタスクが起動する時刻の差をすべて足し合わせることで、 $L_{\Gamma_i}^{\psi}$ を求める。ここで、対象プロセッサ内で、 j 番目にデータを受信するタスク τ_{k_i+j} が起動する時刻を A_j^{ψ} とし、 $O_{k_i+j} = o_{k_i+j}^{\psi}$ とすると、 A_j^{ψ} は次の式で表すことができる。

$$\begin{cases} A_1^\psi &= o_{k_i+1}^\psi \\ A_j^\psi &= o_{k_i+j}^\psi + \alpha_j^\psi T_{k_i+j} \end{cases} \quad (6)$$

ここで、 α_j^ψ は、 $A_j^\psi \geq A_{j-1}^\psi$ を満たす、最小の整数である。

j 番目にデータを受信するタスク τ_{k_i+j} と、そのデータを出力するタスク τ_{k_i+j-1} の起動時刻の差 ($A_j^\psi - A_{j-1}^\psi$) を $d_{j-1,j}^\psi$ とすると、 $L_{\Gamma_i}^\psi$ を求める式は次のようになる。

$$f_{L_{\Gamma_i}^\psi}(l) = f_{R_{k_i+n_i}^\psi}(l - \sum_{j=2}^{n_i} d_{j-1,j}^\psi) \quad (7)$$

(5) Ω_{Γ_i} に属する任意の ψ に対し、 $L_{\Gamma_i}^\psi$ の解析が終了したら、 L_{Γ_i} を次の式により解析する。

$$f_{L_{\Gamma_i}}(l) = \sum_{\forall \psi \in \Omega_{\Gamma_i}} f_{\Psi_{\Gamma_i}}(\psi) f_{L_{\Gamma_i}^\psi}(l) \quad (8)$$

提案手法により求めた、プロセッサ内の端点間処理の応答時間分布の信頼区間について考える。ここでいう信頼区間とは、実際の端点間処理の応答時間分布と、提案手法により求めた端点間処理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、対象とする端点間処理が Cond1 と Cond2 を同時に満たす場合について応答時間分布を解析するため、対象とする端点間処理が Cond1 と Cond2 を同時に満たす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、プロセッサ Γ_i 内における端点間処理の応答時間分布 L_{Γ_i} の信頼区間 ($L_{\Gamma_i}^{from}, L_{\Gamma_i}^{to}$) について、次の定理が成立する。

定理 2 プロセッサ Γ_i に属するタスクを $\tau_{k_i+1}, \tau_{k_i+2}, \dots, \tau_{k_i+n_i}$ とし、4.2 節で述べた手法によって求めた $\tau_{k_i+n_i}$ の応答時間分布の信頼区間を ($R_{n_i}^{from}, R_{n_i}^{to}$) とする。このとき、提案手法により求めた、 Γ_i 内における端点間処理の応答時間分布 L_{Γ_i} の信頼区間を ($L_{\Gamma_i}^{from}, L_{\Gamma_i}^{to}$) とすると、 $L_{\Gamma_i}^{to}$ は、 $R_{n_i}^{to} + \sum_{j=2}^{n_i} T_{k_i+j}$ であり、 $L_{\Gamma_i}^{from}$ は、 $R_{n_i}^{from} + \sum_{j=2}^{n_i} T_{k_i+j}$ である。

証明 $L_{\Gamma_i}^{to}$ について考える。式 7 より、 L_{Γ_i} の信頼区間の最大値は、 $R_{n_i}^{to}$ と $\sum_{j=2}^{n_i} d_{j-1,j}^\psi$ の最大値の和であることが分かる。 $d_{j-1,j}^\psi$ の最大値は T_{k_i+j} であるため、 $\sum_{j=2}^{n_i} d_{j-1,j}^\psi$ の最大値は、 $\sum_{j=2}^{n_i} T_{k_i+j}$ である。よって、 $L_{\Gamma_i}^{to}$ は、 $R_{n_i}^{to} + \sum_{j=2}^{n_i} T_{k_i+j}$ である。次に、 $L_{\Gamma_i}^{from}$ について考える。式 7 より、 L_{Γ_i} の確率密度が保証されていない区間の最大値は、 $R_{n_i}^{from}$ と $\sum_{j=2}^{n_i} d_{j-1,j}^\psi$ の最大値の和である。よって、 $L_{\Gamma_i}^{from}$ は、 $R_{n_i}^{from} + \sum_{j=2}^{n_i} T_{k_i+j}$ である。□

5.2 プロセッサ間を含む端点間処理

プロセッサ間を含む端点間処理の応答時間とは、対象とするシステムに入力データが与えられた時刻から、最終結果が出力される時刻までにかかる時間を意味する。プロセッサ間を含む端点間処理の応答時間の確率的な解析については、まず、システムへの入力信号が入ってから、その入力信号を受信するタスクが起動するまでの時間の分布 D_0 を求める。そして、各プロセッサ内における端点間処理の応答時間分布 L_{Γ_i} 、および、各プロセッサ間において、データをプロセッサ Γ_j に送信するタスク τ_{k_j} の終了時刻と、プロセッサ Γ_i からデータを受信するタスク τ_{k_j+1} の起動時刻の差の分布 D_{Γ_i, Γ_j} をすべて求める。その後、これらの解析結果をすべて加算する。

ここで、それぞれの分布を求める方法について述べる。 D_0 は、 τ_1 の初期位相の分布 O_1 に等しい。 L_{Γ_i} は、5.1 節で述べた手法により求める。 D_{Γ_i, Γ_j} は、 τ_{k_j+1} の初期位相の分布 O_{k_j+1} と等しい。

上述の結果と、プロセッサ数 n を用いて、プロセッサ間を含む端点間処理の応答時間 L は、次の式で求めることができる。

$$L = D_0 + \sum_{i=1}^n L_{\Gamma_i} + \sum_{j=2}^n D_{\Gamma_{j-1}, \Gamma_j} \quad (9)$$

提案手法により求めた、端点間処理の応答時間分布 L の信頼区間 (L^{from}, L^{to}) について考えるために、まず次の補題を示す。

補題 3 ある 2 つの確率変数 X, Y に対し、 X の信頼区間を (X^{from}, X^{to})、 Y の信頼区間を (Y^{from}, Y^{to}) とし、 $P(X > X^{to}) = 0$ 、 $P(Y > Y^{to}) = 0$ であることが保証されているとする。このとき、 $X + Y$ の信頼区間 (Z^{from}, Z^{to}) は、($Z^{to} - \min(X^{to} - X^{from}, Y^{to} - Y^{from}), X^{to} + Y^{to}$) である。

証明 まず、 Z^{to} は、 $X + Y$ の信頼区間の最大値であるため、 X の信頼区間の最大値と、 Y の信頼区間の最大値の和であるといえる。よって、 $Z^{to} = X^{to} + Y^{to}$ が成立する。

次に、 Z^{from} は、 $X + Y$ の確率密度が正しいことが保証されていない区間の最大値である。ここで、 $X + Y$ の確率密度が正しいことが保証されていないための必要十分条件は、 $X + Y = x + y$ となる x, y に対し、次に示す 2 つの論理式の内、少なくとも一方が成立することである。

- $0 \leq x \leq X^{from}$ 、かつ、 $0 \leq y \leq Y^{to}$
- $0 \leq y \leq Y^{from}$ 、かつ、 $0 \leq x \leq X^{to}$

よって, Z^{from} は, $\max(X^{from} + Y^{to}, Y^{from} + X^{to})$ である. 以上より, $Z^{from} = Z^{to} - \min(X^{to} - X^{from}, Y^{to} - Y^{from})$ が成立する. $\square$

次に, 提案手法により求めた, 端点間処理の応答時間分布 L の信頼区間について, 次の定理が成立することを示す.

定理 4 対象とする端点間処理に含まれる任意のプロセッサ Γ_i に対し, 5.1 節で述べた手法により求めた, Γ_i 内における端点間処理の応答時間分布 L_{Γ_i} の信頼区間が $(L_{\Gamma_i}^{from}, L_{\Gamma_i}^{to}]$ であるとする. また, Γ_i で他プロセッサから出力されたデータを受信するタスクを, τ_{k_i+1} とする. このとき, 提案手法により求めた, 端点間処理の応答時間分布 L の信頼区間 $(L^{from}, L^{to}]$ について, L^{to} は, $\sum_{i=1}^n L_{\Gamma_i}^{to} + \sum_{j=1}^n T_{k_j+1}$ であり, L^{from} は, $L^{to} - \min_{\Gamma_i} (L_{\Gamma_i}^{to} - L_{\Gamma_i}^{from})$ である.

証明 L^{to} について考える. 式 9 より, L の信頼区間の最大値は, $L_{\Gamma_i}^{to}$ と T_{k_i+1} の総和であることが分かる. よって, L^{to} は, $\sum_{i=1}^n L_{\Gamma_i}^{to} + \sum_{j=1}^n T_{k_j+1}$ である. 次に, L^{from} について考える. 式 9 と, 補題 3 より, L の確率密度が保証されていない区間の最大値は, $(L_{\Gamma_i}^{to} - L_{\Gamma_i}^{from})$, および, T_{k_i+1} の内で, 最も小さい値を L^{to} から引いた値であることが分かる. ここで, 定理 1 と定理 2 より, $(L_{\Gamma_i}^{to} - L_{\Gamma_i}^{from})$ は, Γ_i に属するタスクの最大実行時間の最小値である. さらに, 前提条件より, すべてのタスクは, そのタスクの起動周期以内で実行を終了し, また, T_{k_i+1} は Γ_i に属するタスクの起動周期の最小値である. よって, 任意の Γ_i に対し, $(L_{\Gamma_i}^{to} - L_{\Gamma_i}^{from}) \leq T_{k_i+1}$ である. 以上より, L^{from} は, $L^{to} - \min_{\Gamma_i} (L_{\Gamma_i}^{to} - L_{\Gamma_i}^{from})$ である. $\square$

6. 評価実験

本章では, 本論文で提案する確率的解析手法の評価を行う. 評価は, 解析が終了するまでにかかる時間, および, 解析の結果得られる確率分布について, シミュレーションの結果と比較することによって行う. ここで, シミュレーションの試行回数は, 10^9 回である. また, 評価を行った環境としては, 2.93GHz のクアッドコアプロセッサ, 4GB のメモリを搭載したマシンを用いた.

評価対象とするタスクセットを表 3 に示す. 表 3 のタスクセットに対し, タスク τ_6 の応答時間分布と, 端点間処理 $\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6$ の応答時間分布を解析する.

まず, タスク τ_6 の応答時間分布を対象として評価を行った. ここで, 対象とするタスクの最大応答時間は 200 であり, 提案手法によって得られる応答時間分布の信頼区間は, $(160, 200]$ である. 提案手法によって得られた応答時間分布と, シミュレーションによって

表 3 評価対象のタスクセット
Table 3 A task set for evaluation.

task	processor	priority	period	E_i^{max}
τ_1	Γ_1	1	100	40
τ_2	Γ_1	2	300	80
τ_3	Γ_1	3	600	40
τ_4	Γ_2	1	100	40
τ_5	Γ_2	2	300	80
τ_6	Γ_2	3	600	40

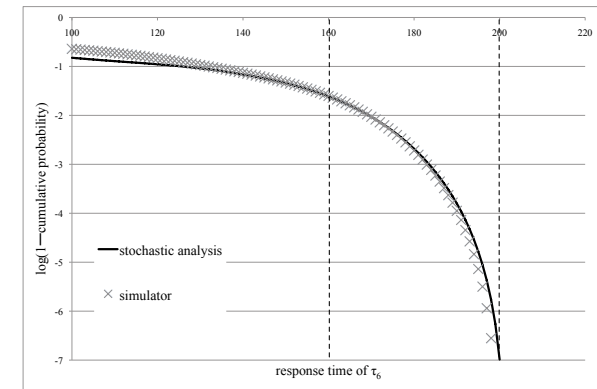


図 4 τ_6 の応答時間分布

Fig. 4 The distribution of τ_6 's response time

得られた応答時間分布を比較した結果は, 図 4 のようになった. 図 4 のグラフの横軸はタスクの応答時間を, 縦軸は対応する応答時間を超える確率の常用対数をそれぞれ表しており, また, 2つの破線は, 提案手法によって得られる, タスクの応答時間に対する信頼区間の境界を表している. 2つの応答時間分布を比較してみると, 提案手法によって得られる応答時間分布の信頼区間の中では, 2つの分布は非常に近いものとなっており, 信頼区間から離れるに従って, 2つの分布も遠ざかっていることが分かる. このような結果から, 解析によって得られる応答時間分布の妥当性, および, 信頼区間を求めることの意義を確認できたといえる. 一方, 応答時間分布を得るまでにかかった時間を比較してみると, 提案手法による解析時間が 4 秒, シミュレーション時間がおよそ 40 分であった. このような結果から, 提案手法の方が, 応答時間分布を短時間で解析できていることが分かる.

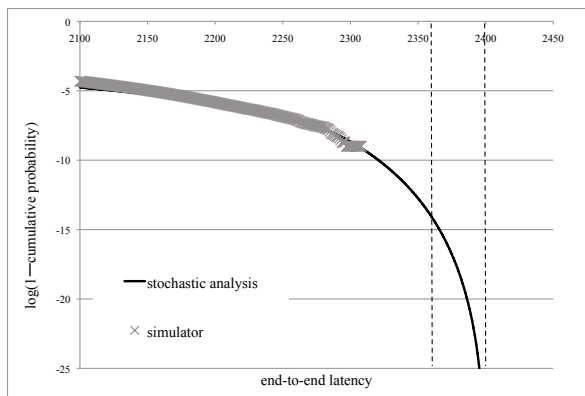


図 5 端点間処理の応答時間分布

Fig. 5 The distribution of the end-to-end latency.

次に、端点間処理 $\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6$ の応答時間分布を対象として評価を行った。ここで、対象とする端点間処理の最大応答時間は 2400 であり、提案手法によって得られる応答時間分布の信頼区間は、 $(2360, 2400]$ である。提案手法によって得られた応答時間分布と、シミュレーションによって得られた応答時間分布を比較した結果は、図 5 のようになった。図 5 のグラフの横軸は端点間処理の応答時間を、縦軸は対応する応答時間を超える確率の常用対数をそれぞれ表しており、また、2 つの破線は、提案手法によって得られる、端点間処理の応答時間に対する信頼区間の境界を表している。2 つの応答時間分布を比較してみると、2 つの分布は近くなっているが、提案手法によって得られる応答時間分布の信頼区間は、非常に確率が小さい区間であることが分かる。このような結果から、解析によって得られる応答時間分布の信頼区間を拡大し、現在よりも確率が大きい区間を解析できるようにすることが必要であると考えられる。一方、応答時間分布を得るまでにかかった時間を比較してみると、提案手法による解析時間が 2.5 秒、シミュレーション時間がおよそ 3 時間であった。このような結果から、解析時間については、タスクの応答時間分布の解析と同様に、提案手法の方が、応答時間分布を短時間で解析できていることが分かる。

7. おわりに

本論文では、タスクの実行時間と初期位相を確率変数として扱い、タスクの応答時間分布、および、端点間処理の応答時間分布について、分布の終端部分を解析する手法を提案した。

そして、提案手法により、解析時間の短い確率的応答時間解析を実現できることを示した。

今後の課題として、解析の結果得られる応答時間分布の信頼区間を拡大することが必要であると考えられる。現在の提案手法による解析では、信頼区間が短く、解析の結果得られる応答時間分布の信頼区間が、確率の非常に小さい区間に限定されてしまう可能性がある。よって、確率が現実的な値であるような区間の応答時間を解析できるように、提案手法を改善する必要があると考えられる。また、今回扱ったシステムでは、バスやネットワーク部分における通信負荷を考慮していなかった。このような通信負荷を考慮した応答時間分布を解析する手法の提案も、今後の課題であると考えられる。

参 考 文 献

- 1) Liu, C.L. and Layland, J.W.: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *Journal of the ACM*, Vol.20, No.1, pp.46–61 (1973).
- 2) Lehoczky, J.P.: Fixed priority scheduling of periodic task sets with arbitrary deadlines, *Proceedings of the 11th IEEE Real-Time Systems Symposium*, pp.201–209 (1990).
- 3) 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).
- 4) Zeng, H., Natale, M.D., Giusto, P. and Sangiovanni-Vincentelli, A.: Stochastic Analysis of CAN-Based Real-Time Automotive Systems, *IEEE Transactions on Industrial Informatics*, Vol.5, No.4, pp.388–401 (2009).
- 5) Díaz, J.L., García, D.F., Kim, K., Lee, C.-G., Bello, L.L., López, J.M., Min, S.L. and Mirabella, O.: Stochastic Analysis of Periodic Real-Time Systems, *RTSS '02: Proceedings of the 23rd IEEE Real-Time Systems Symposium*, p.289 (2002).
- 6) Kim, K., Díaz, J.L., Bello, L.L., López, J.M., Lee, C.-G. and Min, S.L.: An Exact Stochastic Analysis of Priority-Driven Periodic Real-Time Systems and Its Approximations, *IEEE Transactions on Computers*, Vol.54, No.11, pp.1460–1466 (2005).
- 7) Zeng, H.: Probabilistic Timing Analysis of Distributed Real-time Automotive Systems, Technical report, University of California, Berkeley (2008).
- 8) Bernat, G., Burns, A. and Llamas, A.: Weakly Hard Real-Time Systems, *IEEE Transactions on Computers*, Vol.50, No.4, pp.308–321 (2001).
- 9) Díaz, J.L., López, J.M., García, M., Campos, A.M., Kim, K. and Bello, L.L.: Pessimism in the Stochastic Analysis of Real-Time Systems: Concept and Applications, *RTSS '04: Proceedings of the 25th IEEE International Real-Time Systems Symposium*, pp.197–207 (2004).