未知処理量タスクに対する省電力化を目指した周波数制御リアルタイムスケジューリングアルゴリズムとその評価

渋谷 雄[†] 辻野嘉宏[†] 倉本到[†]
新保稔康[††] 中本幸一[†††]

携帯機器の普及にともない，低消費電力技術が重要となってきている．これまで，CPU の周波数を制御することにより，タスクのデッドラインを守り消費電力を削減する様々なリアルタイムスケジューリングアルゴリズムを提案してきた．本論文では，タスク到着時にタスクのデッドラインは既知だが処理量が未知の場合のリアルタイムスケジューリングアルゴリズムを提案し，シミュレーションによって消費電力削減効果を評価した．その結果，提案アルゴリズムは，広い条件下で，固定周波数の場合と比較して，CPU の消費電力を 50%より小さくできることを示した．

A Real Time Scheduling Algorithm for Tasks with Unknown Work Load to Minimize Power Consumption

Yu Shibuya,[†] Yoshihiro Tsujino,[†] Itaru Kuramoto,[†]
Toshiyasu Shimbo[††] and Yukikazu Nakamoto[†††]

The spread of portable devices leads an importance of low power consumption techniques. Several real-time scheduling algorithms have been proposed. They maximize utilized time of the secondary battery under preserving deadlines of tasks with arbitrary or discrete frequency control of CPU. In this paper, we introduce a scheduling algorithm for tasks with known deadline and unknown work load to minimize battery consumption, and illustrates the effectiveness of the algorithm by simulation. As the result, we found that the power consumption of the CPU with our proposed algorithm was reduced to less than 50% of that with fixed frequency.

1. はじめに

携帯機器の普及にともない，低消費電力技術が重要となってきている．通常，携帯機器は，CPU，ディスプレイ，メモリ，ハードディスク，2 次電池，通信ポート等から構成される．それらのうち，ディスプレイやハードディスクに対しては，様々な技法を使って消費電力を小さくする試みが行われてきている[1),3)~5),8),15),17)]．また，回路技術や実装技術の進歩によって CPU 自体の消費電力対策も行われている[3),5),8),15),17)]．しかし，依然として CPU による消費電力は顕著であり，さらなる低消費電力化にはソフトウェアによる粒度の細かい制御が必要である．

そこで，これまでに，リアルタイムタスクのデッドラインを守るという制約の下に，一定量の処理を行う場合に，1 回の充電あたりの 2 次電池の利用可能時間を最長にするリアルタイムスケジューリングアルゴリズムの提案を行った[10)~12),14)]．これらの提案手法では，CPU の周波数を制御することによって，2 次電池の利用可能時間を最長にしている．なお，これらの手法では，CPU の消費電力が周波数によって変化することを利用している．また，タスクの処理は CPU だけで完了することを前提としており，入出力デバイス，メモリ等での消費電力は考慮していない．

また，これらの手法は CPU の周波数をいくらでも大きくすることができると仮定しており，理論上デッドラインミスは生じない．一見この仮定は非現実的なものと考えらる．しかし，CPU の性能が年々指数的に向上している今日，一般的な使用においては，CPU の能力の一部しか利用していないと考えても問題はな

† 京都工芸繊維大学
Kyoto Institute of Technology
†† 三洋電機株式会社
SANYO Electric Co., Ltd.
††† 兵庫県立大学
University of Hyogo

いと思われる．つまり，希に必要となる計算能力のために，大きな能力の CPU を装備しているのであり，通常の使用状態では，周波数をいくらでも大きくできるとモデル化しても問題はないと考える．

本論文では，これまで提案してきたアルゴリズムにおける条件を緩和し，タスク到着時にタスクのデッドラインは既知であるが処理量が未知である場合のスケジューリングアルゴリズムを提案し，シミュレーションにより評価する．

2. 2 次電池の特性

本章では，2 次電池の特性，特に 2 次電池の残存電流量の減少量と周波数の関係について述べる．

定理 1 長さ T の時間を区間 $D_0, D_1, \ldots, D_{m-1}$ に分割する．時間幅 t_i である区間 D_i における CPU の周波数を f_i とするとき，時間 T における 2 次電池の残存電流量の減少量（以下,「消費電力量」と呼ぶ）E は以下のように表される[10),11)]．

$$E(t_0, f_0; \ldots; t_{m-1}, f_{m-1}) = C \cdot \sum_{i=0}^{m-1} t_i {f_i}^{\alpha} \quad (1)$$

ただし，C，α は定数で，$C > 0$，$\alpha > 1$ である．

ここで，本論文のように CPU の周波数が可変の環境下では，CPU の周波数に依存しないプログラムの処理量の尺度が必要である．そこで，本論文ではそのような尺度として「仕事量」を以下のように定義する．

定義 1 実行周波数 f で時間 T の間に処理できる CPU の仕事量 W を $W = fT$ と定義する．

このことは，CPU が 1 クロックで実行可能な処理は一定であるとの仮定に基づき，仕事量がその処理に必要なクロック数で表せることを示している．

また，定理 1 から，次の定理と系が成り立つ．

定理 2 ある仕事量を，時間幅 t_0，t_1 である区間 D_0，D_1 に分割して実行し，区間 D_0，D_1 における CPU の周波数をそれぞれ f_0，f_1 とするとき，消費電力量 $E(t_0, f_0; t_1, f_1)$ は $|f_0 - f_1|$ について単調増加である[10),11)]．

系 1 ある仕事量を，時間幅 $t_0, t_1, \ldots, t_{m-1} (T = \sum_{i=0}^{m-1} t_i)$ である区間 $D_0, D_1, \ldots, D_{m-1}$ に分割して実行するとせよ．このとき，各区間を同一周波数 f で実行したときに消費電力量（式 (1)）は最小になる[10),11)]．

したがって，ソフトウェアによって CPU の周波数を制御できるシステムにおいて，周波数の変化の差が小さいほど消費電力量は小さくなり，同一で可能な限り低い周波数で仕事量を実行した場合に消費電力量は最小になる．また，周波数にあわせて電圧も制御できる場合も式 (1) が成立し，電圧を下げることでさらに消費電力を削減できる．

3. 仕事量が既知の場合のスケジューリングアルゴリズム

本章では，文献 10) において提案した，タスク到着時にそのタスクの仕事量が分かると仮定した 2 次電池の最長スケジューリングアルゴリズムについて述べる．ここで扱う問題は，リアルタイムタスクのデッドラインを守るという制約の下に，2 次電池の 1 回の充電あたりに利用可能な時間を最長にするスケジュール更新問題である．

以降の説明のために，以下の用語を定義する．

定義 2 タスクにおいてデッドラインを守る性質を *dl-safe*（deadline-safe）と呼ぶ

また，簡単化のために以下の仮定を置く．

(1) 各タスクは独立である（他のタスクに依存せず実行することができる）．
(2) タスクの切替えやスケジューリングにともなうオーバヘッドは考慮しない．

本問題では，入力となるタスクは任意の時刻に到着する．ただし，到着タスクの仕事量 w，タスクの実行が完了しなければならないデッドライン時刻 d がタスク到着時に分かるものとする．

本問題は，以下の入力から出力を得るスケジュール更新問題である．

入力：以下の条件を満足する最適タスクスケジュールと新たに到着したタスク
- **条件 1**：各タスクは *dl-safe* であること．すなわち，タスクの到着からデッドライン以内にタスクの実行を完了させること．
- **条件 2**：単位仕事量あたりの消費電力量を最小にすること．

出力：到着したタスクを含め，上記 2 つの条件を満たすタスクスケジュール．

CPU の周波数を任意の値に設定することができ，周波数の変更はタスクの切替え時のみにできる場合について，この問題を解くアルゴリズム MINBAT[☆10)] の概要を以下に述べる．

動的スケジューリングアルゴリズムにおいて，デッドラインを守ることが可能ならば，EDF（Earliest Deadline First）を用いてデッドラインを守ったスケジューリングが可能であることが知られている．MINBAT

☆ algorithm for MINimizing BATtery consumption

では，このEDFに基づいて条件1を満たす．EDFはデッドライン順にタスクを並べた実行待ちタスクのリストで実現できる．さらに条件2を満たすために，可能な限り同一で低い周波数でタスクを実行させるようにする．そこで，同一周波数で実行するタスクを管理するタスクブロックを導入する．タスクブロックは同一周波数で連続して実行されるタスクのリストであり，以下の性質を持つ．

性質1 タスクブロック内のタスクはデッドライン順に並んでいる．

性質2 タスクブロックの最終タスクの実行終了時刻はそのタスクのデッドラインと等しい．

性質3 タスクブロック内のタスクは *dl-safe* である．

さらに，タスクブロックを実行開始時刻の昇順で並べたリストを実行待ちタスクリスト L と呼ぶ．

定義3 実行待ちタスクリストにおいて，隣接するタスクブロックの周波数が降順でない場合に，周波数逆転が生じているという．

なお，最適タスクスケジュールでは，周波数逆転は存在しない．周波数逆転が存在するスケジュールでは，周波数が降順でない隣接するタスクブロックを合併することにより，より消費電力の少ないスケジュールを構成することができる（定理2より）．あるタスクブロック TB_i（周波数：f_i）と，その直後のタスクブロック TB_{i+1}（周波数：f_{i+1}）との間で周波数逆転が起きている場合には，これらのタスクブロックを合併することによって，両者を合わせた仕事量を変化させることなく，ある一定の周波数（f_i'）で両タスクブロックを処理することができる．また，合併した場合，タスクブロック TB_i のタスクを時間的に前倒しで実行することになるので，合併前のスケジュールのタスクが *dl-safe* であれば，合併後のスケジュールでも *dl-safe* である．

タスクが到着したときのMINBATの動作概要を以下に示す．

(1) 到着したタスク τ をデッドライン順に並ぶように実行待ちタスクリスト L に挿入する．そして，タスクが挿入されたタスクブロックの実行周波数 f を，タスクが挿入され増えた仕事量を当該タスクブロックの実行時間で割った値に更新する．

(2) タスク τ が挿入されたタスクブロック TB は仕事量が増加するので，TB 内のタスクが *dl-safe* でなくなる可能性がある．この場合は TB の先頭のタスクから *dl-safe* でないタスクまでと，それ以降を TB_x と TB_y として2つのタスクブロックに分割し，実行周波数を更新する．これを *dl-safe* でないタスクがなくなるまで繰り返す．

(3) (2) のようにタスクブロックを分割した場合，分割されたタスクブロックの前後で周波数逆転が生じる場合がある．周波数逆転が生じているとき，その2つのタスクブロックを合併して同一周波数で実行することにより，*dl-safe* を保持したまま消費電力量を減らすことができる．

なお，実行待ちタスクリストからタスクを取り出すときには先頭タスクブロックの先頭タスクから順に取り出す．

表1 提案済み最適アルゴリズム
Table 1 Proposed algorithms.

	任意時刻	タスク切替え時
連続周波数	STEFC	MINBAT
離散周波数	STEDFC	（NP 完全）

MINBATを修正して任意時刻に周波数の切替えを行うことができる場合の最適アルゴリズムをSTEFC☆と呼ぶ．また，文献11)では，タスク切替え時のみに周波数を切り替え，定数個の値（離散周波数）しかとれない場合のスケジュール更新問題はNP完全であることを示すとともに，MINBATを基に周波数の切替えを任意時刻に行い，離散周波数に設定できる場合の最適アルゴリズムSTEDFC☆☆の提案を行っている．なお，これらの3アルゴリズムの最悪時間計算量は $O(n)$ である（n は実行待ちタスクリスト中のタスク数）[10),11)]．

表1 に各アルゴリズムとその条件を示す．

4. 未知処理量タスクに対するスケジューリングアルゴリズム

本章では，前章で述べた既知仕事量タスクに対するアルゴリズムを基に，未知処理量タスクに対するスケジューリングアルゴリズムを提案する．すなわち，本問題ではタスク到着時にタスクのデッドラインは分かっているのだが仕事量が分からないとする．そこで，到着したタスクの仕事量を定数（C_w）であると仮定し，前章で述べた各アルゴリズムを用いてスケジュールを更新することを提案する．

本問題は，以下の入力から出力を得るスケジュール

☆ Slowest Task Execution algorithm with Frequency Control

☆☆ Slowest Task Execution algorithm with Discrete Frequency Control

更新問題である．

入力：以下の条件を満足する最適タスクスケジュールと新たに到着した未知仕事量タスク

条件 1：各タスクは *dl-safe* であること．すなわち，タスクはデッドライン以内に実行を完了させること．

条件 2'：仮定した仕事量が実際の仕事量と等しい場合に，単位仕事量あたりの消費電力量を最小にすること．

出力：到着したタスクを含め，上記 2 つの条件を満たすタスクスケジュール．

本論文で提案するアルゴリズム SWMFC☆は，条件 2' を緩めて近似的にこの問題を解く．

4.1 仮定した仕事量と実際の仕事量

あるタスクに対して仮定した仕事量（C_w）と実際の仕事量（w）の大小関係には以下の 3 種類がある．

(1) 仮定した仕事量が実際の仕事量より大きい場合（$C_w > w$）

(2) 仮定した仕事量が実際の仕事量と等しい場合（$C_w = w$）

(3) 仮定した仕事量が実際の仕事量より小さい場合（$C_w < w$）

(1) の場合はスケジュール上のタスク終了時刻よりも早くタスクの処理が終了する．その時点でスケジュールを更新する．

(2) の場合はスケジュールどおりなので，スケジュールに従い次のタスクの実行を行う．また，実行可能タスクがない場合には，CPU は停止する．

(3) の場合にはスケジュール上の終了時刻にはタスクが完了しない．そこで，仮定した仕事量（C_w）を持つタスクが新たに到着したものと見なし，スケジュールを更新し，未完了タスクを引き続き実行する．これは，実際の仕事量（w）を持つ 1 つのタスクを，仮定した仕事量（C_w）を持つ複数のタスクに分割することに相当する．しかし，該当タスクのデッドラインが更新前スケジュール上の終了時刻と等しかった場合には，更新時点で新たなタスクが到着したと見なして処理を再開してもデッドラインを超えてしまう．そこで，本論文ではダミータスクを用いてデッドラインを守る方法を提案する．

4.2 ダミータスク

本アルゴリズムでは *dl-safe* を保持するためにダミータスクを導入する．ダミータスクとは以下のような特徴を持っている．

- 実際の仕事量は 0 である．スケジューリングする際には通常のタスクと同様に仮定した仕事量（C_w）を持つタスクとして以下のように扱う．
- ダミータスクは各タスクブロックの最終タスクの次に挿入する．ダミータスクのデッドラインはタスクブロックの実行終了時刻と等しい．つまり，各タスクブロックの最終タスクのデッドラインと等しい．したがって，ダミータスクの挿入により，ダミータスク以外のタスクのスケジュール上での終了予定時刻がデッドラインよりも前になる．
- ダミータスクがタスクリストの先頭タスクブロックの先頭となった場合には時間 0 で完了したものと見なし，次の処理へ進む．

ダミータスクをスケジューリング済みの各タスクブロックに挿入すると，タスクブロック内では仕事量が増え，周波数が上昇する．しかし，タスクブロック内の各タスクの終了時刻はダミータスクを挿入する前の終了時刻よりも挿入後は早くなり，*dl-safe* は保持される．

ところが，ダミータスク含むタスクブロックの間に周波数逆転が生じる可能性がある．従来のアルゴリズムでは，周波数逆転が生じたタスクブロックを合併していた．本アルゴリズムでは，ダミータスクは各タスクブロックの最後にだけ存在させるので，タスクブロックを合併した場合一方のタスクブロックのダミータスクは削除しなくてはならない．しかし，ダミータスクを削除するとタスクブロックの仕事量が減少し，周波数が下降する．結果として，*dl-safe* を破るタスクが存在する可能性が生じる．また，*dl-safe* を破るタスクが存在する場合にはタスクブロック TB の分割を行う必要がある．しかし，タスクブロック TB を TB_x，TB_y に分割を行うと TB_x にはダミータスクが存在しないので挿入する必要がある．ダミータスクの挿入により仕事量が増加し，周波数は上昇する．結果として，周波数逆転が生じる可能性がある．これらのことを繰り返すと，スケジュール更新にともなう計算量が爆発的に増加してしまう．

本研究では，ダミータスクを挿入したタスクブロックの周波数が上昇する際に，周波数逆転の可能性があるのはそのタスクブロックよりも終了時刻が早いタスクブロックであるという点に注目した．そこで，最終タスクブロックから順にダミータスクを挿入する（**図 1**）．

ダミータスクを挿入したタスクブロックとまだ挿入されていないタスクブロックの間で周波数逆転が生じた場合には，タスクブロックの合併を行う（**図 2**）．こ

☆ Scheduling algorithm for tasks with unknown Work load to Minimize battery consumption by Frequency Control

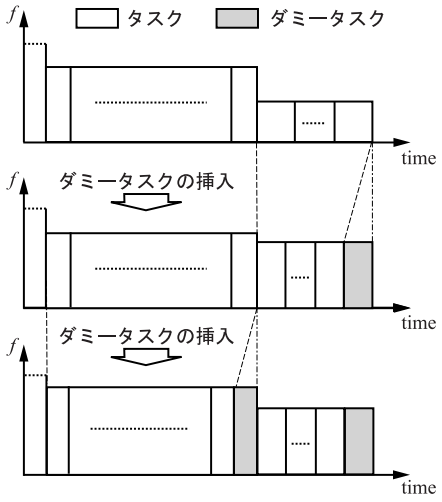

図 1 周波数逆転が生じない場合のダミータスクの挿入
Fig. 1 Inserting dummy tasks with simply decreasing frequency.

の合併では先行するタスクブロックにダミータスクがまだ挿入されていないので，ダミータスクを削除することによる周波数の低下が起こらず，*dl-safe* を破ることはない．これにより，一度タスクリストの最終タスクブロックから先頭のタスクブロックまで処理するだけでダミータスクを含む実行待ちタスクリスト L' を生成することができる．

提案アルゴリズム SWMFC は，以下の手順でスケジュールの更新を行う．

(1) 到着したタスクの仕事量を定数（C_w）であると仮定し，提案済みのアルゴリズム（3 章参照）を用いて最適なタスクリスト L を生成する．
(2) タスクリスト L の最終タスクブロックに対してダミータスクを挿入する．
(3) 挿入されたタスクブロックとその直前のタスクブロックの間で周波数逆転が起こった場合には，2 つのタスクブロックを合併する．この処理を周波数逆転が起こらなくなるまで繰り返す．
(4) ダミータスクの挿入を行っていない最後のタスクブロックに対してダミータスクを挿入する．
(5) (3), (4) をタスクリストの先頭のタスクブロックまで繰り返す．

本アルゴリズムでは，上記のようにダミータスクを用いるため，ダミータスクを挿入する前のタスクリスト L とダミータスクを挿入したタスクリスト L' を保持し，L に対してスケジューリングした後にダミータスクを挿入し L' を作成する．タスク実行の際にも L と L' 内のタスクの関係を保持する．

本アルゴリズムはダミータスクを挿入したことにより，条件 1 を満たすが条件 2' を近似的にしか満たしていない．すべてのタスクが仮定した仕事量と等しい場合に，ダミータスクの挿入にともなう実行周波数の増加はダミータスクを挿入しない場合に比べ最悪で 2 倍となり，消費電力が増加する．そこで，シミュレーションにより，本アルゴリズムの消費電力削減効果を評価する．

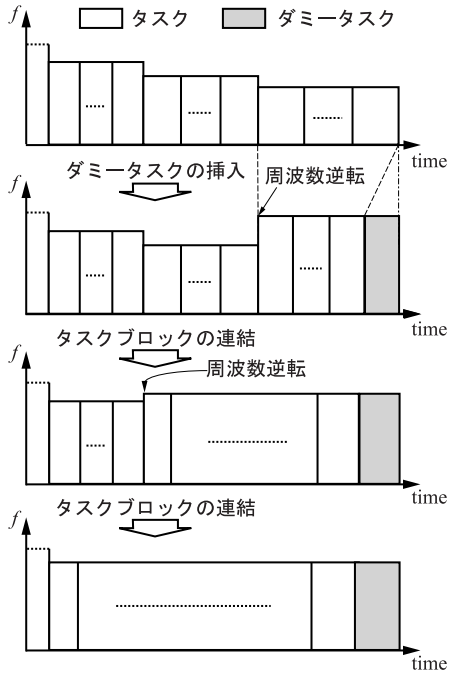

図 2 周波数逆転が生じる場合のダミータスクの挿入
Fig. 2 Inserting dummy tasks with non-decreasing frequency.

5. シミュレーション

本論文において提案した未知仕事量タスクに対するスケジューリングアルゴリズム SWMFC を用いた際，仮定した仕事量の値が消費電力に与える影響をシミュレーションにより評価する．ここで，タスク到着時に分かる仕事量の情報を以下のように分類する．

(1) 未知の場合
(2) 最悪仕事量が既知の場合

(1) の場合は仕事量が未知なので，実際の仕事量の分布（[2, 20] MHz·msec の一様分布と仮定）に対して，仮定仕事量を非常に小さい場合（2 MHz·msec）から非常に大きい場合（40 MHz·msec）まで変化させた場合それぞれについてシミュレーションを行い，仮定仕事量が消費電力に与える影響を評価する．

(2) の場合は最悪仕事量 WCWL（Worst Case Work Load）が既知であるので，仮定仕事量を最悪

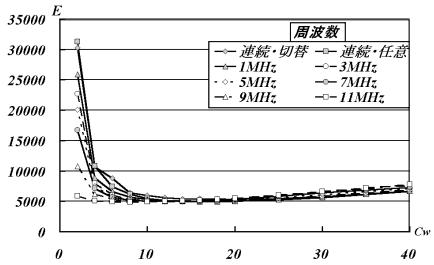

図 3 各仮定仕事量における消費電力比（仕事量：未知，周波数：連続あるいは刻み幅小）

Fig. 3 The power consumption ratio in each assumed task load. The task load is unknown. The frequency is variable continuously or discretely with small step.

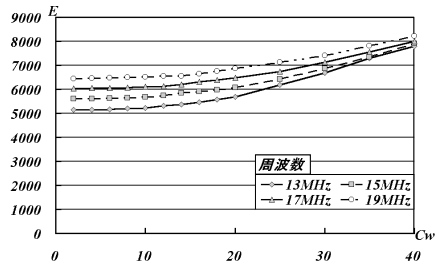

図 4 各仮定仕事量における消費電力比（仕事量：未知，周波数：刻み幅大）

Fig. 4 The power consumption ratio in each assumed task load (task load: unknown, frequency: variable discretely with large step).

仕事量とし，実際の仕事量の分布範囲を変化させることによる消費電力への影響をシミュレーションによって評価する．ここで，実際の仕事量の分布を $[aWCWL, WCWL]$ MHz·msec の一様分布とする．ただし，$0 < a < 1$ である．

以下にシミュレーションの条件を示す．

タスク到着 ：1.0 [msec] 等間隔

タスクのデッドライン：到着時刻+1.5〜8.5 [msec]：一様分布

タスクの数 ：100

離散周波数の刻み幅（とりうる周波数の間隔）：1, 2, . . . , 15 [MHz]

なお，以上のシミュレーション条件は，仮定する仕事量と消費電力との関係を見るためのもので，現実のタスク実行条件を反映したものではない．また，以降のシミュレーション結果の図において，消費電力比（E）とは 1 MHz で 1 msec の間 CPU を実行させたときの消費電力との比（100 回のシミュレーションの平均値）である．ここで消費電力の計算には式 (1) を用い，$C = 1$，$\alpha = 1.6$ とする．また，ダミータスク挿入前のタスクリスト生成のためには，周波数切替えのタイミングと設定できる周波数に応じて，表 1 に示した中から適切なアルゴリズムを用いる．

5.1 タスクの仕事量が未知の場合

図 3 は仮定した各仕事量と消費電力比の関係を示しており，周波数は連続周波数か，離散周波数の刻み幅が細かい場合である．この図が示すように，周波数切替えタイミングの違いにかかわらず仮定した仕事量が小さすぎると消費電力比が非常に大きくなる．一方，仮定した仕事量が大きくなるにつれて消費電力比が増加する傾向にあるが，その変化は緩やかである．

ここで，仮定した仕事量が実際の仕事量の $1/n$ であったとすると，n 回タスクの再スケジュールを行う必要があり，そのたびにデッドラインまでの時間は短くなるので周波数は上昇する．したがって n が大きい場合，つまり仮定した仕事量が小さい場合には周波数が非常に大きくなり，消費電力が増加する．

また，仮定した仕事量が実際の仕事量の m 倍であるとすると，タスクはスケジュール上の割り当てられた時間の $1/m$ の時間で完了する．これは周波数が実際の仕事量に対して m 倍になることを意味する．結果として，スケジュール上の終了時刻よりも早く完了する．そして，実行可能タスクがなくなった場合には停止する．この高い周波数での実行と停止を繰り返した結果，消費電力が増加する．

さらに，仮定した仕事量が実際の仕事量に近いほど，実行周波数はスケジュール上の周波数に近くなり，仕事量が既知の場合の最適アルゴリズムを用いたのと同様となり，消費電力は減少する．

図 4 は離散周波数の刻み幅が粗い場合の消費電力比と仮定した仕事量の関係を表している．この図から分かるように，周波数の刻み幅が粗いものは仮定した仕事量が非常に小さい場合でも消費電力比が顕著には増加していない．

この理由として以下のことがあげられる．まず，仮定した仕事量が小さく必要な周波数が低い場合でも，周波数の刻み幅が粗いので，高い周波数が割り当てられる．結果として，わずかな時間で仮定した仕事量を完了するのだが，実際にはタスクを完了していないので再びタスクが到着したと見なし，スケジュールを更新する．ここで，低い周波数で実行した場合よりもデッドラインまでの時間が長いので，再び低い周波数が必要となる．しかし，刻み幅が粗いのでほとんどの場合前に実行したときと同じ高い周波数が割り当てられる．このことを繰り返すので，周波数の異常な上昇が起こらず，消費電力の増加がほとんど起こらないのである．

また，仮定した仕事量が大きすぎる場合は図 3 と同

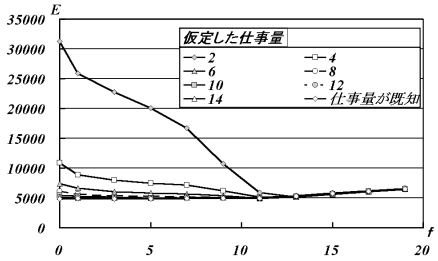

図 5 各周波数刻み幅における消費電力比（仮定仕事量：2〜14 MHz·msec および既知）
Fig. 5 The power consumption ratio in each frequency step (task load: 2–14 MHz·msec and known).

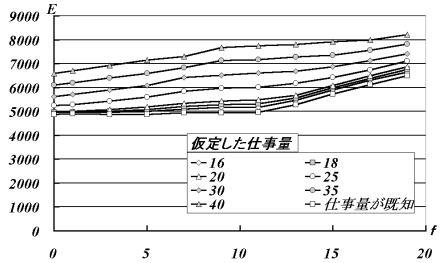

図 6 各周周波数刻み幅における消費電力比（仮定仕事量：16〜40 MHz·msec）
Fig. 6 The power consumption ratio in each frequency step (task load: 16–40 MHz·msec).

様に実行と停止を繰り返すことが多くなり消費電力が増加する．

図 5 は周波数の刻み幅と消費電力の関係を示している．この図は，仕事量が既知の場合と，仮定している仕事量が 2〜14 MHz·msec の場合である．この図から，周波数の刻み幅が粗くなるにつれ消費電力が減少しているが，11 MHz 刻みを超えてからは消費電力が増加傾向にあることが分かる．

周波数の刻み幅が非常に細かい場合には，仮定した仕事量に対して必要な周波数に近い値が設定できる．しかし，仮定した仕事量と実際の仕事量に差があるとスケジュールどおりにタスクは完了しない．その結果再スケジュールを行う機会が増え，周波数の上昇が生じるので消費電力が多くなる．

また，周波数の刻み幅が非常に粗い場合には仮定した仕事量に必要な周波数，さらには実際の仕事量に必要な周波数よりも高い周波数が割り当てられる．この場合，そのタスクは早く完了し，実行可能なタスクもすべて完了させてしまい，停止する時間が長くなる．結果として，実行と停止を繰り返し消費電力が増加してしまう．

ところが，周波数刻み幅が 11 MHz の付近では，仮定した仕事量にかかわらず，ダミータスクのために，必要な周波数よりも少し高い周波数が割り当てられる．その結果，仮定したタスクはデッドラインよりも早く終わるが，再スケジュールの際に急激には周波数が増加しないので，周波数の刻み幅が細かい場合や粗い場合に比べ消費電力が少ない．

図 6 は図 5 と同様のグラフであり，仮定している仕事量は 16〜40 MHz·msec である．このグラフでは周波数の刻み幅が粗くなるにつれ消費電力が増加している．

この場合は，仮定した仕事量が実際の仕事量よりも大きいということが頻繁に起こる．周波数の刻み幅が

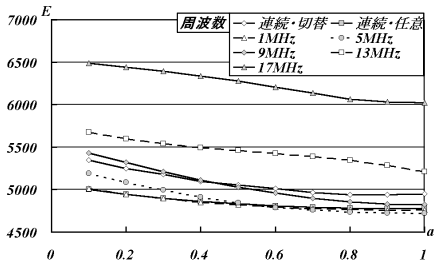

図 7 各仮定仕事量における消費電力比（最悪仕事量が既知の場合）
Fig. 7 The power consumption ratio in each assumed task load. The worst case task load is known.

細かい場合は，スケジュールよりもタスクが早く完了しても余る時間は少なく，実行可能なタスクが存在する場合が多いのでスケジュールから大きくはずれることはない．しかし，周波数の刻み幅が粗くなるにつれ，実行周波数は高くなり，タスクの早期完了にともなう余剰の時間は長くなる．結果として，実行可能なタスクがなくなり停止する時間も長くなる．このことが消費電力の増加につながる．

5.2 最悪仕事量のみ分かる場合

図 7 は最悪仕事量が既知で，仮定仕事量を最悪仕事量にした場合である．横軸は実際の仕事量の分布を示しており，値が大きいほど最悪仕事量に近い範囲に分布していることになる．この図から周波数切替えタイミングの違いにかかわらず分布が狭いほど消費電力が少ないことが分かる．これは，仮定した仕事量と実際の仕事量が近づくほど，タスクリストのスケジュールにタスクの実行と完了が近づくためである．

図 8 は最悪仕事量が既知で，仮定仕事量を最悪仕事量にした場合の離散周波数の刻み幅と消費電力の関係を示している．周波数の刻み幅が細かい場合は必要な周波数に近く設定できるために消費電力は少ない．特に仮定した仕事量が実際の仕事量と近い場合には，必要な周波数よりも少し上の周波数が与えられる方が新

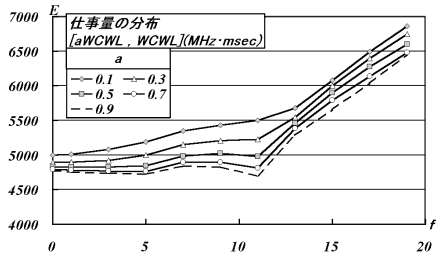

図 8 各周波数刻み幅における消費電力比（最悪仕事量が既知の場合）

Fig. 8 The power consumption ratio in each frequency step. The worst case task load is known.

たなタスクの到着にともなう仕事量の増加と周波数の上昇が抑えられる．しかし，周波数の刻み幅が粗いと必要以上の周波数で実行してしまうので，実行と停止を繰り返し消費電力が増加する．また周波数の刻み幅によって，仮定した仕事量が必要とする周波数に近い値が設定できない場合には必要以上の周波数で実行してしまい，消費電力が増加している．

5.3 仕事量が既知の場合との比較

シミュレーション全体を通して，提案アルゴリズムを用いた場合の消費電力は，仕事量が既知の場合のアルゴリズムに比べ，最悪の場合で約 6.4 倍であった．これは仮定した仕事量があまりにも小さかったために再スケジュールを繰り返した場合に起こった．また，消費電力が最も少ない場合には，仕事量が既知のアルゴリズムと同等の消費電力であった．これは，周波数の刻み幅がやや粗く，仮定した仕事量は平均仕事量よりも少ない場合に起こった．この組合せの場合，再スケジュールを数回行うことで，実際の仕事量に対して最適な周波数よりもやや高い周波数で実行し，デッドラインよりもわずかに早く処理を完了することで新たなタスクの到着時に周波数の急激な増加を抑えられたためだと考えられる．

また，固定周波数（20 MHz）で実行した場合の消費電力比に対して，本アルゴリズムでは最大 62%減少した．ここで，20 MHz はシミュレーションの条件下で新たなタスクが到着する前に現在のタスクを確実に完了できる周波数である．

6. 関 連 研 究

Lee ら[9] は高電圧高周波数と低電圧低周波数の 2 つの走行モードを持った CPU に対して，低周波数時のタスクの最悪実行時間が分かっていると仮定して，周期タスクを対象とした静的アルゴリズムと動的アルゴリズムを提案しているが，動的アルゴリズムはヒューリスティックに基づいている．Yao ら[16] は，各タスクの仕事量は分かっていると仮定し，最適な静的アルゴリズムと，ヒューリスティックを用いた動的アルゴリズムを提案している．そして，動的アルゴリズムの性能を最適な場合との比で評価している．

Krishna ら[7] は，最悪時を仮定して前もって静的にスケジュールしておき，動的にそれを修正しながらスケジュールするアルゴリズムを提案している．なお，タスクについての情報は前もって分かっていると仮定している．Hong ら[6] は，タスクのそれぞれに仕事量が割り当てられているときのヒューリステックアルゴリズムを扱っている．また，現実のプロセッサでは周波数と電圧を瞬時に変化させることは困難であるが，その点についても議論している．

Aydin ら[2] は，周期タスクを対象として，最悪仕事量が分かる場合の静的最適手法，それに実際の仕事量で修正を加える動的手法，そして平均仕事量を用いて適応的にスケジュールする動的手法を提案している．Pillai ら[13] は，最悪仕事量を用いた静的アルゴリズム，最悪時を仮定して余裕ができれば周波数を下げる動的手法，そして将来の必要仕事量を予測しスケジュールする手法の 3 種を提案している．しかし，いずれもヒューリスティックなアルゴリズムである．

本論文で提案する手法には，以下にあげるような長所がある．

- 仕事量が分かっている場合に最適性が証明されているアルゴリズムを，仕事量が未知の場合に応用するという理論的基礎を持っており，提案アルゴリズムが良いスケジュールを行っているかどうかを判定することができる．上記の関連研究であげられている動的アルゴリズムのすべてはヒューリステックに基づいており，提案アルゴリズムがどの程度良いスケジュールを行っているかを直接的には評価できない．
- 関連研究で提案されているアルゴリズムと同様に，前もって到着タスクの分からない動的アルゴリズムであり，応用範囲が広い．もちろん，すべてのタスクが最初に到着していると考えることで，静的アルゴリズムとしても利用可能である．
- Lee ら[9] の提案とは異なり，CPU のとりうる周波数が，何通りであっても（任意の値，すなわち無数であっても）対応できる．
- Aydin ら[2] の提案とは異なり，タスクは周期タスクでなくてもよい．

7. ま と め

本論文では，未知仕事量タスクに対するスケジューリングアルゴリズムを提案した．提案アルゴリズムは，各タスクの仕事量が分かっている場合に消費電力を最適にするアルゴリズムとして筆者らの研究グループが提案したものを，タスク到着時にそのタスクのデッドラインは分かっているが仕事量が分からない場合に適用したものである．その意味で最適性について理論的な基礎がある．

提案アルゴリズムをシミュレーションにより評価したところ，仮定した仕事量が実際のタスクの仕事量に比較してかなり小さく，かつ周波数の刻み幅が特に小さい場合を除いて，パラメータの値がほぼ何であっても低い消費電力を実現していることが分かった．特に，仮定した仕事量を実際のタスクの仕事量よりわずかに多めに見積もった場合に最適となり，さらに多めに見積もっても消費電力量はそれほど悪くはならなかった．この点から本提案手法の実用性は高いといえる．

また，本提案アルゴリズムでは固定周波数の場合と比較して 50%以上（最大 62%）の消費電力の低減を可能としたが，この値は，デッドラインを守ることができる最低の実行周波数での消費電力を 100%とした場合の値である．現実には，CPU が処理すべきタスクの仕事量にはばらつきがある．なかには多くの仕事量を持つタスクもあるため，固定周波数でデッドラインを守るためには，より高性能の CPU が必要となる．しかし，このような高性能の CPU を固定周波数で利用することは，タスクの仕事量が少ない場合に，電力の浪費となる．したがって，デッドラインを守りつつ低消費電力を実現している本アルゴリズムは，CPU の性能が高ければ高いほど，その有効性を増すものと考えられる．

参 考 文 献

1) 相原 達，関谷一雄，黒田昭裕：ノートブック PC におけるローパワーシステム技術，電子情報通信学会誌，Vol.80, No.4, pp.370–376 (1997).
2) Aydin, H., Melhem, R., Mosse, D. and Mejia-Alvarez, P.: ynamic and Aggressive Scheduling Techniques for Power-Aware Real-Time Systems, *Proc. 22nd IEEE Real-Time Systems Symposium*, pp.95–105 (2001).
3) 千葉耕司，卜部周二：自動車携帯電話システムにおける間欠受信による待ち受け時低消費電力化，信学論，Vol.J80-A, No.5, pp.728–734 (1997).
4) 藤田憲治，倉田雅弘，高橋秀和：特集ノート PC のジレンマ，日経バイト，No.185, pp.118–143 (1998).
5) 服部 武，生越重章，久保田周治，小林 聖：携帯機低消費電力化技術とその効果，信学論，Vol.J80-A, No.5, pp.735–745 (1997).
6) Hong, I., Qu, G., Potkonjak, M. and Srivastava, M.B.: Synthesis Techniques for Low-Power Hard Real-Time Systems on Variable Voltage Processors, *Proc. 19th IEEE Real-Time Systems Symposium*, pp.178–187 (1998).
7) Krishna, C.M. and Lee, Y.H.: Voltage-Clock-Scaling Adaptive Scheduling echniques for Low Power in Hard Real-Time Systems, *Proc. 6th IEEE Real Time Technology and Applications Symposium*, pp.156–165 (2000).
8) 黒田忠広，櫻井貴康：マルチメディア CMOS VLSI のための低電力回路設計技術，信学論，Vol.J80-A, No.5, pp.746–752 (1997).
9) Lee, Y.H. and Krishna, C.M.: Voltage-Clock Scaling for Low Energy Consumption in Real-time Embedded Systems, *Proc. 6th International Conference on Real-Time Embedded Systems and Application*, pp.272–279 (1999).
10) 中本幸一，辻野嘉宏，都倉信樹：携帯機器の 2 次電池の最長利用を目的とするリアルタイムスケジューリングアルゴリズム，信学論，Vol.J83-D-I, No.1, pp.125–133 (2000).
11) 中本幸一，辻野嘉宏，都倉信樹：離散周波数制御を利用した 2 次電池の最長利用のためのタスクスケジューリングアルゴリズム，信学論，Vol.J83-D-I, No.12, pp.1249–1259 (2000).
12) 岡本大介，渋谷 雄，辻野嘉宏，中本幸一：2 次電池の最長利用のための実行順序制約を用いたスケジューリングアルゴリズムについて，情報処理学会アルゴリズム研究会資料，Vol.76, No.6, pp.35–42 (2001).
13) Pillai, P. and Shin, K.G.: Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems, *Proc. 8th ACM Symposium on Operating Systems Principles*, pp.89–102 (2001).
14) 新保稔康，渋谷 雄，辻野嘉宏：消費電力の最小化を目指したリアルタイムスケジューリングアルゴリズムの効果，情報処理学会論文誌，Vol.43, No.8, pp.2836–2839 (2002).
15) 瀧 和男，李 副烈：パストランジスタ論理に基づく低消費電力回路方式と設計事例，信学論，Vol.J80-A, No.5, pp.753–764 (1997).
16) Yao, F., Demers, A. and Shenker, S.: A Scheduling Model for Reduced CPU Energy, *Proc. 36th Annual Symposium on Foundations of Computer Science*, pp.374–382 (1995).
17) 吉田幸弘，宋 宝玉，奥畑広之，尾上孝雄，白川功：組込み用プロセッサの低消費電力化に関する一手法，信学論，Vol.J80-A, No.5, pp.765–771

(1997).

(平成 16 年 3 月 26 日受付)
(平成 17 年 4 月 1 日採録)

渋谷　雄（正会員）

1990 年大阪大学大学院工学研究科通信工学専攻博士後期課程修了．工学博士．同年より京都工芸繊維大学工芸学部電子情報工学科助手，同講師，同助教授を経て，2000 年同大学大学院工芸科学研究科助教授（工芸学部兼務）．1997〜1998 年ドイツ，カッセル大学客員研究員．ヒューマンインタフェース，メディアコミュニケーション，アウェアネスに関する研究に従事．電子情報通信学会，ヒューマンインタフェース学会，システム制御情報学会，日本人間工学会，ACM 各会員．

辻野　嘉宏（正会員）

1979 年大阪大学基礎工学部情報工学科卒業．1984 年大阪大学大学院博士課程修了．工学博士．同年大阪大学基礎工学部助手．同大学講師，助教授を経て，1999 年より京都工芸繊維大学大学院工芸科学研究科教授（情報・生産科学専攻）．計算機言語，並列処理記述，HCI，ソフトウェア工学に関する研究に従事．IEEE-CS，ACM，電子情報通信学会，日本ソフトウェア科学会，ヒューマンインタフェース学会各会員．

倉本　到（正会員）

2001 年大阪大学大学院基礎工学研究科情報数理系専攻博士後期課程修了．博士（工学）．同年より京都工芸繊維大学工芸学部電子情報工学科助手．計算機上での協調作業およびエンタテインメントコンピューティングに関する研究に従事．日本ソフトウェア科学会，ヒューマンインタフェース学会各会員．

新保　稔康（正会員）

2003 年京都工芸繊維大学大学院工芸科学研究科博士前期課程修了．現在，三洋電機株式会社勤務．在学中にリアルタイムスケジューリングアルゴリズムの研究に従事．

中本　幸一（正会員）

1982 年大阪大学大学院基礎工学研究科前期課程修了．同年 NEC 入社．組込みソフトウェア，モバイルシステムソフトウェアの研究開発に従事．1990〜2001 年 Cornell 大学計算機科学科客員研究員．1997 年大阪大学大学院情報数理系専攻博士課程入学．2000 年単位取得退学．博士（工学）．2003〜2004 年より電気通信大学客員教授．2004 年より現職．電子情報通信学会，日本ソフトウェア科学会，IEEE Computer Society，ACM 各会員