Vol.2012-SLDM-155 No.13
Vol.2012-EMB-24 No.13
2012/3/2

組込みシステムにおける消費エネルギー削減のためのスラック時間の活用

三輪 遼平[†1]　高瀬 英希[†1,†2]
曾 剛[†1]　高田 広章[†1]

動的電圧・周波数制御技術の有効性を高めるためには，タスクスケジューリングの解析によって導出されるスラック時間を適切に活用することが重要である．本研究では，組込みシステムの消費エネルギー削減を目的としたスラック時間活用手法 LFST を提案する．提案手法は，導出されたスラック時間をただちに全て使用する設定のものと，各タスクの動作周波数が平準化できるような設定のものの 2 種類の動作周波数を算出する．これらの動作周波数を適切に選択し，各タスクにスラック時間を分配することで，システムのデッドライン制約を保証した上で消費エネルギーを削減する．実験により提案手法の有効性を評価したところ，スラック時間の分配率を固定したものと比較して，最大で 64 %の消費エネルギー削減を達成した．

A Distribution Method of the Slack Time for Reducing the Enegy Consumption of Embedded Systems

Ryohei Miwa,[†1] Hideki Takase,[†1,†2] Gang Zeng[†1]
and Hiroaki Takada[†1]

In order to improve the efficiency of DVFS, it is important to properly distribute the slack time among tasks which is estimated by analyzing the task scheduling. In this paper, we proposed a method for slack time distribution called LFST (Leveling Frequency with Slack Time) to reduce the energy consumption of embedded real-time systems. The method uses two kinds of frequencies. One is the lowest frequency for guaranteeing deadline, and the other is a leveled frequency for analyzed tasks. Then, it selects the higher one from them and distributes it to the dispatched task so that energy can be reduced while guaranteeing the deadline restriction. The effectiveness of the LFST has been validated via experiments, and the results showed that maximal 64 % energy can be reduced comparing with fixed distribution rate of slack time.

1. はじめに

組込みシステムの性能や計算精度を保証しつつ消費エネルギーを最小化することは，運用コストや製造コスト，システムの信頼性の面から，近年の非常に重要な課題となっている．組込みシステムの消費エネルギーを削減する手法として，プロセッサへの供給電圧および動作周波数を制御する動的電圧・周波数制御技術（Dynamic Voltage and Frequency Scaling. 以降，DVFS）がある．組込み向けプロセッサに採用される CMOS 回路の消費エネルギーは，動作周波数の 2 乗に比例すると近似できる[1)]．ゆえに，DVFS によってプロセッサの動作周波数を降下させることで，システムの消費エネルギー削減が実現できる．

高いリアルタイム性が要求される組込みシステムにおいては，デッドラインという定められた時刻までにタスクの処理を終えなければならないというデッドライン制約を持つ．このような組込みシステムでは，タスクがデッドラインを超えて実行されること（デッドラインミス）は許容されない．DVFS の適用時には，デッドライン制約を保証できる範囲内でのみ，タスクの実行時間を伸張させることができる．タスクの実行を伸張できるこの時間を，スラック時間という．文献 2) によれば，各タスクの動作周波数が平準化されるようにスラック時間を配分することで，システムの消費エネルギーが最適になることが分かっている．

本研究では，組込みシステムにおいて一般に採用される Rate-Monotonic Scheduling (RMS)[3)] によるタスクスケジューリング方式を適用対象とした，スラック時間活用手法 LFST（Leveling Frequency with Slack Time）を提案する．LFST では，デッドライン保証周波数および平準化周波数という 2 種類の動作周波数を用いる．前者は，タスクスケジューリングの解析により導出されたスラック時間を，ただちに全て使用するという設定により算出される．後者は，対象とするタスクの後続に実行されるタスク群の動作も考慮し，各タスクの動作周波数が平準化できるように算出される．タスクのディスパッチ時にこれらの動作周波数から適切なものを選択することで，各タスクにスラック時間を配分していく．これにより，導出されたスラック時間を効率良く活用し，システムのデッドライン制約を保証した上での消費エネルギー削減を達成できる．

さらに本研究では，上述のスラック時間活用手法の問題点を改善した拡張手法 LF-NTA

†1 名古屋大学
Nagoya University
†2 日本学術振興会
Japan Society for the Promotion of Science

Vol.2012-SLDM-155 No.13
Vol.2012-EMB-24 No.13
2012/3/2

（LFST and NTA）を提案する．本手法では，タスクのディスパッチ時の次に最初にリリースされるタスクのリリース時刻を利用する．これにより，LFST では動作周波数を正しく平準化できない状況を解消し，さらなる消費エネルギー削減効果を得る．

DVFS およびスラック時間に関する既存研究では，より正確なスラック時間を少ない計算時間で導出することに焦点が当てられていた．本研究のような，導出されたスラック時間を効率良く活用することで組込みシステムの消費エネルギー削減を狙う手法は，われわれの知る限り存在しない．ランダムタスクセットに対して DVFS を適用した評価実験によって，提案手法は，どのような特性を持つタスクセットに対してもシステムの消費エネルギーを最小化する動作周波数を決定できることを示す．

本論文の構成は，以下の通りである．まず，2 章にて関連研究について述べたのちに，3 章にて提案手法を着想するに至った経緯を具体例を交えて述べる．4 章にてスラック時間活用手法 LFST，および，拡張手法 LF-NTA の詳細を述べる．5 章にて評価実験の結果と考察を述べたのち，6 章にてまとめと今後の方針を述べる．

2. 関 連 研 究

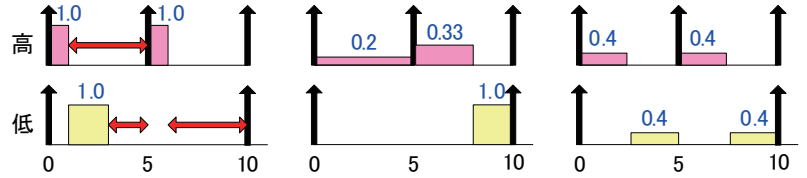
DVFS においてタスクの動作周波数を決定する手法およびタスクのスラック時間を導出する手法は，これまでに数多く提案されている．本章では，それらの手法の一部を紹介する．

文献 4) における手法では，デッドライン制約の保証の可否を定めるスケジューリング判定式を用いて．システム動作前に最大の動作周波数を決定する．文献 5) では，あるタスクが最悪実行時間よりも早く終了した場合，そのスラック時間を次に実行されるタスクの動作周波数の決定に用いる手法を提案している．文献 6) では，Stretching-to-NTA という手法が提案されている．この手法では，ディスパッチ対象となるタスク（対象タスク）のスラック時間の導出に，全タスクについて次にリリースされる時刻のうちの最小値（Next Time Arrival．以降，NTA）を用いる．文献 7) では，Stretching-to-NTA を拡張し，対象タスクのみならず起動されている他のタスクにもスラック時間を配分する手法を提案してる．

文献 8) では，RMS を採用したシステムにおける手法 Work-Demand Analysis（WDA）を提案している．この手法は，タスクのディスパッチ時に，限定した区間内のタスクスケジューリングを解析し，対象タスクのスラック時間を導出する．WDA は，文献 9) による評価のうちでは，固定優先度によるシステムにおいて最も有効な手法である．われわれは，文献 10) において，WDA にある問題点を改善した手法である Effective-WDA を提案した．具体的には，まず．対象タスクがより高優先度のタスク群から受ける実行時間の影響を求める計算式を詳細化した．さらに，より低優先度のタスク群に与える，対象タスクの実行時間の影響を求める計算式を単純化した．以上により，Effective-WDA は，WDA よりも正確なスラック時間を少ない計算時間で導出できる．

表 1 タスクセット 1

タスク	周期	最悪実行時間	実行時間
τ_1	5	1	1
τ_2	10	2	2



(a) DVFS を適用しない場合　(b) ただちに全て使用する場合　(c) 各タスクに配分する場合

図 1 タスクセット 1 でのスラック時間を用いた DVFS の適用

3. スラック時間を配分する必要性

本研究で提案するスラック時間活用手法の必要性について，例題を交えて解説する．例題として，**表 1** で示されるタスクセット 1 に対して DVFS を適用することを考える．**図 1** は，DVFS を適用した際の場合のタスクスケジューリングをあらわしている．ここで，最大周波数を 1.0 として，時刻 0 からタスクセットのハイパーピリオド（周期の最大公倍数時間）である時刻 10 までの消費エネルギーを考える．なお，文献 1)，8) における式より，消費エネルギー E は，動作周波数 f の 3 乗と実行時間 T の積によって求められることとする．つまり，DVFS を適用しない**図 1(a)** では，消費エネルギーは 4.0 と計算される．

図 1(a) に示す上側の高優先度のタスクは，時刻 1 においてスラック時間 4 を残して実行が終了する．ここで，タスクのディスパッチ時に発生するスラック時間を，関連研究における評価実験のようにただちに全て使用する場合（**図 1(b)**）を考える．このとき，時刻 0 では高優先度タスクの動作周波数が 0.2 に設定され，高優先度タスクの 2 回目のディスパッチ時にはスラック時間 2 を使い切る動作周波数 0.33 が設定される．結果として，全体の消費エネルギーは 1.15 となる．次に，各タスクの動作周波数が平準化されるようにスラック時間を配分する場合を考える．**図 1(c)**）では，低優先度タスクにもスラック時間 4 を配分で

Vol.2012-SLDM-155 No.13
Vol.2012-EMB-24 No.13
2012/3/2

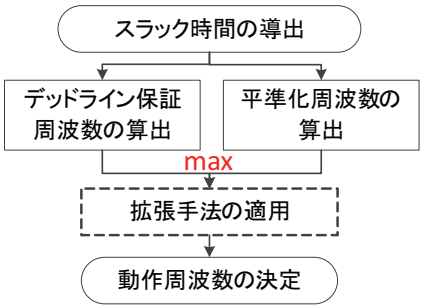


図 2 スラック時間活用手法の全体像

きる動作周波数 0.4 が設定され，全体の消費エネルギーは 0.64 となる．以上より，スラック時間をただちに全て使用するよりも，対象タスクには一部分だけ与えて後続のタスクにスラック時間を配分したほうが，消費エネルギーが小さくなることがわかる．図 1(b) の場合は，後続のタスクの動作周波数が高くなるため，全体の消費エネルギーが大きくなってしまっている．つまり，スラック時間は後続のタスク群も考慮して適切に活用したほうがよい．

Ishihara らは，図 1(c) のように，スラック時間を使い切って全てのタスクの動作周波数が同一となるよう DVFS を適用することが，消費エネルギー最小にできることを証明した[2]．本研究では，この動作周波数の設定方法に着目する．タスクスケジューリングの解析により，各タスクにスラック時間を配分して動作周波数を自動的に平準化できる手法を開発する．

4. スラック時間活用手法

本章では，RMS を採用した組込みシステムの消費エネルギー削減を目的としたスラック時間活用手法を解説する．提案手法は，2 種類の動作周波数を算出して用いるスラック時間活用手法 LFST およびその拡張手法 LF-NTA である．なお，RMS では，全てのタスクは周期的に起動され，その周期および優先度は静的に与えられる．また，本研究では，最悪実行時間と同様に，DVFS を行わない場合の各タスクの平均実行時間は既知であるとする[⋆1]．

4.1 全 体 像

図 2 に，提案手法の全体像を示す．提案するスラック時間活用手法は，スラック時間が導出される各タスクのディスパッチ時に動作し，デッドライン保証周波数算出部，平準化周波数算出部，および，拡張手法適用部によって構成される．

デッドライン保証周波数とは，全てのタスクのデッドライン制約を保証できる最小の動作周波数のことを指す．これの算出方法は，4.2 節で述べる．平準化周波数とは，各タスクの動作周波数が同一になるようにスラック時間を配分するとして算出される動作周波数のことを指す．これは，対象タスクとより低優先度のタスク群について，それぞれのタスクと後続にあるより高優先度のタスク群を解析対象として解析対象のタスクに割り当てられる動作周波数が平準化されるような値を求め，それらの最大値を取ることによって算出される．詳細については，4.3 節で述べる．

スラック時間活用手法 LFST は，算出された 2 つの動作周波数のうち大きいものを対象タスクに割り当てる．デッドライン保証周波数の方が大きい場合，平準化周波数を用いるとデッドライン制約を保証できないことを意味する．逆に，平準化周波数の方が大きい場合，平準化周波数はデッドライン制約を保証でき，かつ，消費エネルギー削減効果の高いものであることを意味する．LFST では，以上の算出と選択の処理により，デッドライン制約を保証し，かつ，消費エネルギー削減効果の高い動作周波数を決定することができる．

LFST においては，対象タスクがある条件を満たす時に，決定された動作周波数でもスラック時間を活用しきれないことがある．この問題点を改善した拡張手法として，LF-NTA（LFST and NTA）を提案する．本手法では，既存のスラック時間導出手法である Stretching-to-NTA[6] の考え方を用いて，スラック時間をより積極的に活用できる動作周波数を決定する．提案手法の拡張の詳細については，4.4 節で述べる．

4.2 デッドライン保証周波数の算出

デッドライン保証周波数 $f_i^{gd}(t)$ は，時刻 t においてディスパッチ時に導出されたスラック時間 $slack_i(t)$ を，ただちに全て対象タスクに割り当てるとしたものである．つまり，全タスクのデッドライン制約を保証可能な動作周波数の最小値となる．設定可能な動作周波数の最大値を f^{max} とすると，$f_i^{gd}(t)$ は以下の式で求められる．

$$f_i^{gd}(t) = \frac{\omega_i^{rem}}{\omega_i^{rem} + slack_i(t)} \cdot f^{max} \tag{1}$$

ここで，ω_i^{rem} は，文献 10) の 2.3.1 節で定義した対象タスクのディスパッチ時における残り最悪実行時間である．

4.3 平準化周波数の算出

平準化周波数算出部では，まず，対象タスクおよび低優先度のタスク群について，タスク毎平準化周波数 $f_\alpha^{lv-\tau}(t)$ を求める．これは，タスク τ_α とより高優先度のタスク群における

⋆1 組込みシステムは，タスクの実行時の動作や振る舞いを設計時に十分解析できるため，この前提は妥当である．

Vol.2012-SLDM-155 No.13
Vol.2012-EMB-24 No.13
2012/3/2

表 2 タスクセット 2

タスク	周期	平均実行時間
τ_1	5	1
τ_2	10	2
τ_2	15	0.5

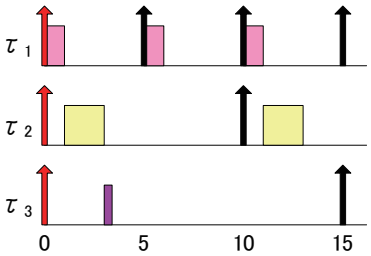


図 3 タスクセット 2 のスケジューリング

動作周波数が，平均実行時にスラック時間を発生することなく平準化されるよう算出されるものである．$f_\alpha^{lv\text{-}\tau}(t)$ は，平均実行時間での高優先度タスクからの干渉 H_α^{ave} およびタスクの残り平均実行時間 $acet_\alpha^{rem}(t)$ を用いて，以下の式で算出する．

$$f_\alpha^{lv\text{-}\tau}(t) = \frac{H_\alpha^{ave} + acet_\alpha^{rem}(t)}{ud(t) - t} \cdot f^{max} \tag{2}$$

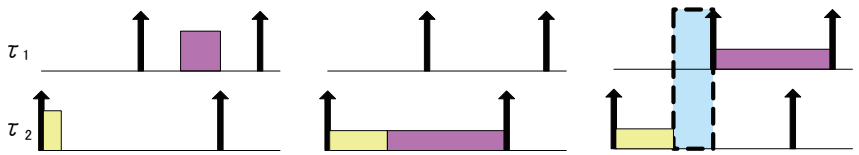
ここで，H_α^{ave} および $acet_\alpha^{rem}(t)$ は，Effective-WDA における高優先度タスクからの干渉 H_α および残り最悪実行時間 ω_α^{rem} において，最悪実行時間に関する変数および計算方法を，平均実行時間のものに置換することで求められる．これらの計算方法の詳細については，本論文の紙面の都合上，文献 10) を参照されたい．ただし，ω_α^{rem} と異なる点として，タスクの残り平均実行時間が 0 未満となる場合は，$acet_\alpha^{rem}(t) = 0$ として扱う．

対象タスクの平準化周波数 $f_i^{lv}(t)$ は，上記の方法により算出されたタスク毎平準化周波数の最大値を取ることで求める．すなわち，

$$f_i^{lv}(t) = \max_{\tau_\alpha \in T_i^L(t)} (f_\alpha^{lv\text{-}\tau}(t)) \quad \textbf{where} \;\; T_i^L(t) = \left\{ \tau_\alpha \mid \phi_\alpha \leq \phi_i \;\wedge\; \tau_\alpha \in T \right\} \tag{3}$$

となる．ここで，T はタスクの集合，ϕ_i は τ_i の優先度をあらわす．

具体例として，**表 2** および**図 3** に示すタスクセット 2 を考える．最大の動作周波数が 1.0 として，時刻 0 における τ_1 の平準化周波数 f_1^{lv} を算出する．まず，τ_1 の次のデッドライン $ud_1(0)$，平均実行時間での高優先度タスクからの干渉 H_1^{ave} およびタスクの残り平均実行時間 $acet_1^{rem}(0)$ は，図 3 からそれぞれ，5，0，1 と計算される．同様にして，$ud_2(0)$，H_2^{ave}，$acet_2^{rem}(0)$ は 10，2，2 に，$ud_3(0)$，H_3^{ave}，$acet_3^{rem}(0)$ は 15，7，0.5 になる．以上の値を式 (2) に代入し，タスク毎平準化周波数 $f_1^{lv\text{-}\tau}(0)$，$f_2^{lv\text{-}\tau}(0)$ および $f_3^{lv\text{-}\tau}(0)$ は，それぞれ，0.2，0.4 および 0.5 と算出される．これらの最大値を取り，f_1^{lv} は 0.5 と算出される．



(a) LFST 適用前　(b) 平準化周波数算出時の解析　(c) LFST の適用後

図 4 LFST において問題が発生するタスクセットのスケジューリング

平準化周波数は，タスク毎平準化周波数の最大値を取ることで，対象タスクと後続に実行されるタスクの動作周波数が可能な限り同一となるような値を得ることができる．また，ディスパッチ時に導出されるスラック時間を他ののタスクに配分するため，後続のタスク群の実行周波数が大きくなることを避けられる．ただし，平準化周波数は，タスクのデッドライン制約を必ずしも保証できるものではないことに注意が必要である．

4.4 拡張手法 LF-NTA

LFST において平準化周波数が選択された場合，全てのスラック時間を有効に活用できない状況が起きることがある．**図 4** によって問題を説明する．**図 4(a)** のタスクスケジューリングを解析した LFST では，時刻 0 における動作周波数は $f_2^{lv\text{-}\tau}(0)$ となる．このときの平準化周波数は，各タスクの平均実行時間を用いて，時刻 0 以降は**図 4(b)** のように実行されると見なして算出している．しかし，実際には**図 4(c)** に示すように，τ_2 の実行終了時刻から τ_1 の次の起動時刻までに活用できないスラック時間が生じてしまう．これは，式 (2) の解析区間は対象タスクのディスパッチ時から各タスクの次のデッドラインまでであり，また，平準化周波数を決定する際に各タスクの次に起動される時刻が考慮されないためである．

LF-NTA は，LFST における上記の問題を，既存手法 Stretching-to-NTA[6] の考え方を導入することにより改善した手法である．Stretching-to-NTA[6] では，対象タスクのディスパッチ時から NTA までの区間において，他のタスクが起動されることがなく，かつ，スラック時間が存在すれば，NTA まで対象タスクの実行時間を伸張させる．LF-NTA は，他のタスクが NTA まで起動されることがなく，かつ，LFST による動作周波数で実行される対象タスクが NTA までに終了できる場合に適用される．LF-NTA による動作周波数 $f_i^{lf\text{-}nta}(t)$ は，以下の計算式で算出される．

$$f_i^{lf\text{-}nta}(t) = \frac{\omega_i^{rem}}{NTA - t} \cdot f^{max} \tag{4}$$

Vol.2012-SLDM-155 No.13
Vol.2012-EMB-24 No.13
2012/3/2

LF-NTA の適用時は，NTA まで対象タスクの実行時間が他のタスクに影響することはないため，デッドライン制約は保証される．そして，平準化周波数が選択された場合に残るスラック時間を活用できるため，さらなる消費エネルギー削減効果が期待できる．

4.5 提案手法の計算量

タスクのディスパッチ時におけるスラック時間活用手法の計算量について述べる．デッドライン保証周波数算出部については，式 (1) の処理のみであり計算量は $\mathcal{O}(1)$ となる．平準化周波数算出部については，平均実行時間での高優先度タスクからの干渉 H_α^{ave} の計算量は，Effective-WDA における H_α と同じく，タスク数を n として $\mathcal{O}(n)$ となる[10]．式 (2) は $\mathcal{O}(1)$，式 (3) は $\mathcal{O}(n)$ であり，全体の計算量は $\mathcal{O}(n)$ である．拡張手法適用部については，Stretching to NTA の計算量が $\mathcal{O}(1)$[6]，式 (4) は一次式であり，全体の計算量は $\mathcal{O}(1)$ である．以上より，スラック時間活用手法 LFST および拡張手法 LF-NTA の計算量は，タスク数を n として $\mathcal{O}(n)$ に抑えられ，いずれの手法も，実用的な範囲であるといえる．

5. 評価実験

5.1 実験環境

提案手法の評価を行うため，タスクセットに対して DVFS を行って消費エネルギーを計算するシミュレータを作成した．シミュレータでは，タスクセットの情報を入力とし，システムの動作開始時からハイパーピリオド（すべてのタスクの周期の最小公倍数）までの区間で DVFS を行う．動作周波数の切り替えは，各タスクのディスパッチ時に行う．切り替え時の動作周波数は，文献 10) において提案したスラック時間導出手法 Effective-WDA および本研究で提案したスラック時間活用手法を適用して決定した．動作周波数は，最大値 f^{max} まで連続値で設定可能であるとした．タスクセットの消費エネルギー E は，文献 1) を参考に，タスクに割り当てられた動作周波数 f の 3 乗とタスクの実行時間 T の積をとり，各タスクの実行ごとの総和によって計算した．

評価対象は，自作したスクリプトによって生成されるランダムタスクセットとした．自作スクリプトに与える条件は，以下の通りである．

- 各タスクの周期は，10 から 100 の間でランダムに設定される．
- 各タスクの最悪実行時間は，1 からタスクの周期の間でランダムに設定される．

さらに，評価実験で用いたタスクセットは，上記の条件に対し，**表 3** で示す条件を付与して生成した．各設定で生成されるランダムタスクセットの情報は，次の通りである．

- 設定 1：タスク数が少ない場合

表 3 作成するタスクセットの条件

設定	タスク数	最悪実行時の CPU 使用率	最悪実行時間に対する平均実行時間の割合
1	2	0.5	0.5
2	10	0.5	0.5
3	5	0.1	0.5
4	5	0.9	0.5
5	5	0.5	0.1
6	5	0.5	0.9

- 設定 2：タスク数が多い場合
- 設定 3：最悪実行時の CPU 使用率が低い場合
- 設定 4：最悪実行時の CPU 使用率が高い場合
- 設定 5：最悪実行時間に対する平均実行時間の割合が小さい場合
- 設定 6：最悪実行時間に対する平均実行時間の割合が大きい場合

提案手法の評価のための比較対象として，導出されたスラック時間を対象タスクに配分する割合を 10 %から 100 %まで 10 %刻みで固定して動作周波数を決定する手法を導入した．本手法および提案手法の適用による消費エネルギーは，表 3 の設定によるランダムタスクセットを 100 個ずつ生成し，それらの平均をとった．なお，本評価では，DVFS 切り替えにかかるオーバヘッド，および，静的な消費エネルギーは無視できるものとした．

5.2 実験結果と考察

実験結果を**図 5** に示す．横軸は，表 3 の各設定およびすべての設定での平均値を，スラック時間活用手法ごとに示している．凡例は，それぞれ，“df-best” がスラック時間の配分の割合を固定した手法のうちで最も消費エネルギーが小さいもの，“LFST” が提案手法 LFST，“LF-NTA” が拡張手法 LF-NTA を示している．縦軸は，導出されるスラック時間をただちにすべて利用する手法，すなわち，スラック時間の配分の割合を 100 %で固定したものの消費エネルギーを 100 として正規化して示した．

提案手法は，いずれのタスクセットにおいても df-best に対して有効であり，平均で 17 %の消費エネルギーを削減した．提案手法は，どのような特性を持つタスクセットにも有効であり，各タスクの動作周波数を平準化して消費エネルギーが削減できた．さらに，CPU 使用率が低い設定 3 のタスクセットにおいて，df-best から最大で 64 %の消費エネルギーを削減できた．これは，CPU 使用率が低い場合ほど，スラック時間の総量が多くなり，提案手法により動作周波数の平均値を小さくできたためであると考えられる．

Vol.2012-SLDM-155 No.13
Vol.2012-EMB-24 No.13
2012/3/2

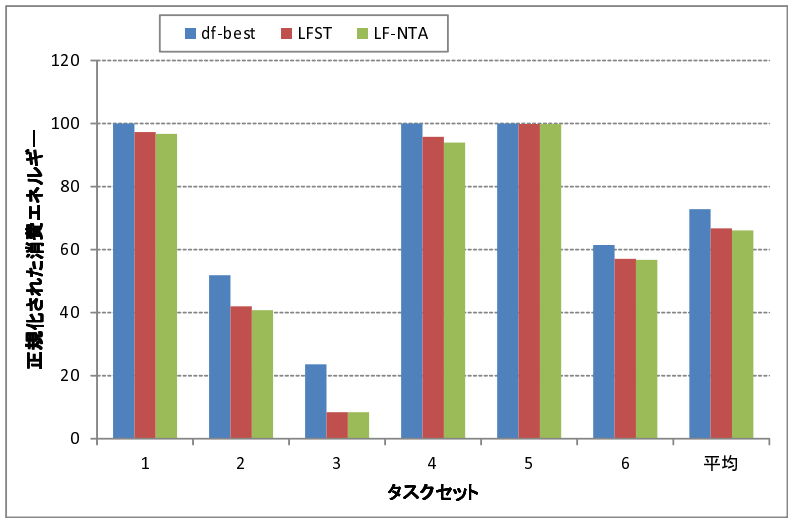


図 5 各設定における消費エネルギー

拡張手法 LF-NTA は，いずれのタスクセットに対しても LFST と比較して有効となり，特にタスク数が多いタスクセットにおいて，最大で 3 %の消費エネルギーを削減できた．これは，LF-NTA が，LFST では利用できないスラック時間を利用することができたためである．タスク数が多いタスクセットには，LFST によって最低優先度タスクの動作周波数が大きくなりやすい．このとき，LF-NTA は，LFST の問題点を改善できる場合が多くなる．

6. ま と め

本研究では，組込みシステムにおける動的電圧・周波数制御の消費エネルギー削減効果の向上を目的としたスラック時間活用手法 LFST（Leveling Frequency with Slack Time），および，その拡張手法 LF-NTA（LF and NTA）を提案した．

LFST は，デッドライン保証周波数と平準化周波数の 2 種類の動作周波数を用いる．前者は，デッドライン制約を保証できる最小の周波数であり，後者は，解析区間における各タスクの動作周波数が平準化できるように算出されたものである．各タスクのディスパッチ時にこれらの動作周波数から適切な方を選択することで，デッドライン制約を保証した上で，各タスクにスラック時間を配分し，消費エネルギーを削減する．LF-NTA は，ディスパッチ時に次のリリース時刻までは対象タスクしか実行されない場合において，LFST では利用できなかったスラック時間を利用できたためである．提案手法を DVFS に適用して評価したところ，スラック時間の分配率を固定するものと比較して最大で 64 %の消費エネルギー削減を達成した．

今後の方針としては，提案したスラック時間活用手法をリアルタイム OS に実装し，実用システムにて有効性を評価することが挙げられる．

謝辞 本研究の成果の一部は，独立行政法人科学技術振興事業団（JST）戦略的創造研究推進事業（CREST）「情報システムの超低消費電力化を目指した技術革新と統合化技術」の支援による．

参 考 文 献

1) Hu,S.X. and Gang,Q.: “Fundamental of power-aware scheduling,” in *Designing Embedded Processors: A Low Power Perspective*, chap. 9, Springer, pp. 219–230, (2007).
2) Ishihara,T. and Yasuura,H.: “Voltage scheduling problem for dynamically variable voltage processors,” in *Proc. of Int’l Sympo. on Low Power Electronics and Design*, pp.197–202, (1998).
3) Liu,L.C. and Layland,W.J.: “Scheduling algorithms for multiprogramming in a hard real-time environment,” *Journal of ACM*, Vol.20, No.1, pp.46–61, (1973).
4) Shin,Y., et al.: “Power optimization of real-time embedded systems on variable speed processors,” in *Proc. of the Int’l Conf. on Computer-Aided Design*, pp.365–368, (2000).
5) Aydin,H., et al.: “Dynamic and aggressive scheduling techniques for power-aware real-time systems,” in *Proc. of Real-Time Systems Sympo.*, pp.95–105, (2001).
6) Shin,Y. and Choi,K.: “Power conscious fixed priority scheduling for hard real-time systems,” in *Proc. of Design Automation Conf.*, pp.134–139, (1999).
7) Pillai,P. and Shin.K.G.: “Real-time dynamic voltage scaling for low-power embedded operating systems,” in *Proc. of ACM Sympo. on Operating Systems Principles*, pp.89–102, (2001)
8) Kim,W., et al.: “Dynamic voltage scaling algorithm for fixed-priority real-time systems using work-demand analysis,” in *Proc. of Int’l Sympo. on Low Power Electronics and Design*, pp.396–401, (2003).
9) Kim,W., et al.: “Performance evaluation of dynamic voltage scaling algorithms for hard real-time systems,” *Journal of Low Power Electronics*, Vol.1, pp.1-11, (2005).
10) 三輪遼平ほか：組込みシステムにおける低消費エネルギー志向の効率的なスラック時間の導出，情報処理学会研究報告, Vol.2010-EMB-18, No.11, (2010).