社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－SLDM－124 (15)
2006／3／17

リアルタイム性を考慮したフィードバック制御による動的周波数制御手法

中村 哲朗† 加藤 真平† 小林 秀典† 山崎 信行†

† 慶應義塾大学大学院理工学研究科開放環境科学専攻
〒 223-8522 横浜市港北区日吉 3-14-1
E-mail: †{tetsuro,shinpei,kobayashi,yamasaki}@ny.ics.keio.ac.jp

あらまし Dynamic Freqency Scaling (DFS) 及び Dynamic Voltage Scaling (DVS) は，動作周波数と動作電圧を制御することにより，エネルギー消費の削減を可能にするもので，組込システムにとって有用な技術である．しかし，既存のハードリアルタイムシステムを対象とした DFS 及び DVS 手法の多くは，タスクのコンテキストスイッチが起きる度に動作周波数を計算して利用するため，あまり実用的ではない．本論文では，ある程度のデッドラインミスは許容するソフトリアルタイムシステムを対象として，フィードバック制御により一定範囲内にデッドラインミス率を制御しながら，十分なエネルギー節約を実現する DFS 手法を提案する．また，この提案手法では，ある一定の周期で動作周波数を制御するため既存の手法と比べて実用的である．シミュレーションによる評価では，提案手法がデッドラインミス率を一定の範囲内に制御しながら，既存の手法に近いエネルギー節約を実現できたことを示す．
キーワード リアルタイムシステム，エネルギー消費，フィードバック制御，動的周波数制御

Real-Time Dynamic Frequency Scaling Technique by Feedback Control

Tetsuro NAKAMURA†, Shinpei KATO†, Hidenori KOBAYASHI†, and Nobuyuki YAMASAKI†

† Keio University
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama, Kanagawa 223-8522 Japan
E-mail: †{tetsuro,shinpei,kobayashi,yamasaki}@ny.ics.keio.ac.jp

Abstract Dynamic Freqency Scaling (DFS) and Dynamic Voltage Scaling (DVS) are promising techniques for embedded systems, which scale freqency and voltage, in order to reduce the energy consumption. However, most of existing DFS techniques for hard real-time system are not practical, since they compute the freqency scaling every time contexts are switched. In this paper, we propose a DFS technique for soft real-time systems, which is able to provide significant energy savings with controling the deadline miss ratio in a constant range. Also, the proposed DFS technique is more practical than the existing techniques, since it controls the freqency only at a certain period. The simulation experiments demonstrate that the proposed technique is able to save up nearly as much energy as the exsisting techniques with controling the deadline miss ratio in a constant range.
Key words Real-Time System, Energy Consumption, Feedback Control, Dynamic Frequency Scaling

1. はじめに

現在，我々の周りには多くのリアルタイムシステムが存在する．リアルタイムシステムでは，システム全体の正確さが計算結果の論理的な正確さだけでなく，その計算が定められた時間内に完了するかどうかという時間的な正確さにも依存する．

リアルタイム性は要求される時間制約の厳しさにより，ハードリアルタイム性とソフトリアルタイム性に分けられる．ハードリアルタイム性とはデッドラインがシステムに深刻なダメージを引き起こす場合の時間制約の性質のことをいう．それに対して，ソフトリアルタイム性とはデッドラインを満たすことがシステムの性能面では望まれるが，デッドラインミスが生じてもシステムに深刻なダメージを引き起こすことはない場合の時間制約の性質のことをいう．

近年急速に発達しているユビキタスコンピューティングやマルチメディア処理等は，ソフトリアルタイムシステムの典型的な例であり，その重要性は増している．例えば，ユビキタス空間内での Bluetooth や無線 LAN 等によるネットワーク通信や，PDA 等の携帯型端末でソフトリアルタイム性を実現し，デッドラインミスするかしないかではなく，その QoS を保証する

という研究が盛んに行なわれている．

こういったソフトリアルタイムシステムは主に組込システムであることが多い．例えば，ソフトリアルタイム通信や動画再生を行う携帯型端末等である．このようなシステムではエネルギー資源に限りがあり，バッテリーを長持ちさせるためにはそれらを効率良く使わなければならない．そのためには，プロセッサのエネルギー消費を下げることが重要となる．プロセッサのエネルギー消費に着目するのは，ほとんどのアプリケーションにおいて，プロセッサが最もエネルギー消費の高い要素であるためである．DFS (Dynamic Freqency Scaling) 及び DVS (Dynamic Voltage Scaling) は，動作周波数と動作電圧を制御することにより，プロセッサのエネルギー消費を削減することが可能であり，これらのシステムにとって有望な技術である．

そういったシステムではバッテリーの寿命を引き延ばすことは重要であるが，システムの性能が高まるにつれて，強力なプロセッサが必要となり，エネルギー消費の増加につながってしまう．現在このようなシステムに用いられているプロセッサの特徴として，まず最大の計算速度は，平均の計算速度よりもかなり高いということが挙げられる．つまり高いパフォーマンスが必要なのはある短い時間だけで，残りの時間は低いパフォーマンスで十分である．よって最大計算速度が必要でないときは，DFS によりプロセッサの動作周波数をただ下げればよい．次にこれらのプロセッサの多くは CMOS 回路により形成されている．CMOS 回路では動作周波数の最大値が動作電圧に依存するため，DFS により動作周波数を下げることで，プロセッサはより低い動作電圧で動作可能となる．つまり DFS と DVS は密接に結び付いている．以後，動作周波数を制御して，それと同時に動作電圧をその動作周波数で動作可能な最低値に制御することを，合わせて DFS と呼ぶことにする．また，CMOS 回路ではサイクルあたりに消費されるエネルギーは動作周波数に比例し，かつ動作電圧の 2 乗に比例するので，動作周波数と動作電圧を下げれば，大きなエネルギー節約が可能である．

しかし，時間制約があるソフトリアルタイムシステムでは，動作周波数を制御することによりタスクの実行時間に影響を与え，デッドラインミスが増加してしまうかもしれない．このようなバッテリーの寿命が重要であり，また時間制約を持つシステムにおいて DFSS を用いる際には，エネルギー消費とリアルタイム性のトレードオフを扱わなければならない．

本論文では，組込ソフトリアルタイムシステムを対象として，フィードバック制御によりエネルギー消費とリアルタイム性の QoS を実現する DFS 手法を提案する．

2. 関連研究

リアルタイムシステムにおいてフィードバック制御を用いた最初の研究は，Stankovic らによって提案された FC-EDF であった [1]．FC-EDF は，ソフトリアルタイムシステムを対象とした，PID フィードバックコントローラと EDF (Earliest Deadline First) [2] スケジューラを統合したリアルタイムスケジューリング手法である．FC-EDF はデッドラインミス率を PID フィードバック制御により制御しながら，適切なシステム CPU 利用率を求め，要求されたシステム CPU 利用率に従って，インプリサイス計算モデル [3] における予測実行時間のバージョンを変更する．そして，その予測実行時間を使って EDF によるスケジュールを行う．

DFS については，現在までに多くの研究が行われている．その中には単純なシステムを対象としているものもあれば，リアルタイムシステムを対象としているものもある．

Pillai らは，ハードリアルタイムシステムを対象とした DFS をいくつか提案した [4]．これらが対象とするリアルタイムタスクは周期タスクのみである．また，これらの手法は基盤とする OS のスケジューリングアルゴリズムやタスク管理システムと密接につながっている．彼らが提案した 1 つ目の手法は，EDF と DFS を統合した CC-DFS (Cycle-conserving DFS) である．この手法は，リアルタイムタスクが一般的に実際の実行時間が最悪実行時間より短いことを利用して，できるだけ動作周波数を下げる手法である．タスクのリリース時は，実際の実行時間は分からないので，最悪実行時間を用いてタスクセットの CPU 利用率を計算する．タスクの実行完了後は，そのタスクの次のリリース時まで実際の実行時間を用いて，タスクセットの CPU 利用率を計算する．そして，各タスクのリリース時に，動作周波数を，(最大動作周波数) × (計算した CPU 利用率) に制御する，という手法である．2 つ目の手法は，CC-EDF と同様に EDF と DFS を統合した LA-DFS (Look-ahead DFS) である．この手法は，コンテキストスイッチ時に，その時の最高優先度タスク，つまり最もデッドラインが近いタスクの動作周波数をできるだけ下げるという手法である．具体的には，コンテキストスイッチが起きて，あるタスクが実行を開始する時，各タスクの 1 つ先の周期の実行時間を予約し，先のタスクが実行可能であることを保証する．その上で，優先度が低いタスクから実行時間を予約していき，余りの時間を計算する．そして，動作周波数を，(最高優先度タスクの最悪実行時間) / (余りの時間) に制御するという手法である．シミュレーションによる評価では，LA-DFS が CC-DFS より，多くのエネルギー消費を削減できたことが示されている．

Zhu らは，ハードリアルタイムシステムを対象として，フィードバック制御を用いた FB-DFS (Feedback-DFS) を提案した [5]．Pillai らの手法と同様に，対象とするリアルタイムタスクは周期タスクのみであり，EDF を基盤としている．この手法は，タスクの実行時間がある規則的なパターンに従って動的に変動するようなシステムを対象としており，その実行時間の変動に対応するためにフィードバック制御を用いる．各コンキストスイッチ時に，フィードバック制御により予測した実行時間と計算した余りの時間を用いて，動作周波数を計算して，その値で制御するという手法である．もしタスクの実際の実行時間が予測した実行時間より長い場合には，予測実行時間分は計算した動作周波数で制御し，残りの時間は最大動作周波数で制御する．

上記のハードリアルタイムシステムを対象とした DFS は，いずれもコンテキストスイッチが起きる度に動作周波数を計算し，その値でを制御するという手法である．また，その計算に

要するオーバヘッドはタスクセットのタスク数を N とすると，$O(N)$ であり，計算量やオーバヘッドを考えると，あまり実用的であるとはいえない．

本論文では，ソフトリアルタイムシステムを対象としたDFSを提案する．提案手法では，ある一定の周期で動作周波数を計算し，そのオーバヘッドはタスクセットに関わらず $O(1)$ であり，上記の手法に比べて実用的である．また，フィードバック制御を用いることによりデッドラインミス率をある一定の範囲内に制御しながら，十分なエネルギー節約を実現する．

3. 設　　計

3.1 システムモデル

本論文で想定するシステムでは，周期タスクのみを扱う．システム内には n 個の周期タスクが存在し，各タスクは実行の途中でプリエンプト可能であり，またタスクは互いに独立しているものとする．各タスクの実際の実行時間と最悪実行時間は等しいものとし，相対デッドラインは周期と等しいものとする．

もしタスクがデッドラインミスを起こした場合は，その時点でそのタスクの実行を強制的に終了させ，次に優先度の高いタスクに実行を切り替えることにする．

3.2 アプローチ

関連研究で述べた，ハードリアルタイムシステムを対象とする既存のDFS手法の主な特徴として次の2点が挙げられる．

- コンテキストスイッチの度に最高優先度タスクの動作周波数を計算し，次のコンテキスイッチ時までその計算した値で制御
- 動作周波数の計算では，その他のタスクがデッドラインミスしないことを保証する上での，最高優先度タスクが利用できる余りの時間とそのタスクの最悪実行時間を利用

この手法では，コンテキストスイッチの度に動作周波数を計算するのでオーバヘッドは大きくなってしまい，またタスクの最悪実行時間等のタスクの情報を前もって知っておかなければならない．本論文では，既存の手法とは異なる次のようなアプローチをとる．

- ある一定のサンプリング周期で，次のサンプリング周期における動作周波数を計算し，そのサンプリング周期が終了するまでその計算した値で制御
- 動作周波数の計算では，前のサンプリング周期の情報を利用

ここでポイントとなるのが，動作周波数の計算の際に前のサンプリング周期のどのような情報を用いるかである．本論文の目的は，組込ソフトリアルタイムシステムにおいて，DFSを用いることにより，エネルギー消費とリアルタイム性のQoSを実現することである．多少のデッドラインミスを許容するソフトリアルタイムシステムでリアルタイム性を高めるということは，つまりデッドラインミスの割合を減少させることを意味する．よってデッドラインミスの割合をできるだけ低く抑えることが重要になる．動作周波数をできるだけ大きい値に保てばデッドラインミスの割合は低下するが，大きいエネルギー消費の削減は望めない．動作周波数を小さい値に保てば大きくエネルギー消費を削減できるが，デッドラインミスの割合は増加してしまう．そこでフィードバック制御を利用してデッドラインミスの割合を，ある一定の範囲内に制御するように，動作周波数を調節する．デッドラインミスの割合には，式(1)のようなデッドラインミス率を用いる．ただし，あるサンプリング周期においてリリースしたタスクの数を NR，デッドラインミスしたタスクの数を ND，デッドラインミス率を MR とする．

$$MR = \frac{ND}{NR} \tag{1}$$

3.3 フィードバック制御

フィードバック制御を用いるには，制御量，目標値，操作量を定義しなければならない．制御量とは，周期的に観測し，制御する値である．目標値とは，制御量の目標とする値である．操作量とは，制御量に影響を及ぼす値で，フィードバックコントローラによりこの値を調整する．また，周期的に観測する制御量と目標値との誤差を偏差と呼ぶ．

フィードバック制御ループの流れは次の通りである．

（1） 周期的に制御量を観測し，目標値と比較して偏差を計算

（2） コントローラは，偏差を基に新しい操作量を計算

（3） 制御対象のシステムを，この新しい操作量を用いて実行

（4） ステップ1に戻る

フィードバック制御は，以前の状態を基に適した状態を決定する手法であるといえる．このように，フィードバック制御を用いることで，状態に応じた柔軟な対処が可能となる．

3.4 FC-DFS-MR

本節では，FC-DFS-MR (Feedback Control DFS based on Deadline Miss Ratio) について述べる．FC-DFS-MRはEDFスケジューラとフィードバックコントローラを統合したシステムである．

フィードバックコントローラではデッドラインミス率を制御したいので，制御量にはデッドラインミス率を用いる．操作量には，デッドラインミス率に影響を及ぼす動作周波数を用いる．目標値は，この場合でいえばデッドラインミス率の目標とする値であり，リアルタイムシステムにおいてはその値は0である．しかし，目標値を0に設定した場合，もし前のサンプリング周期におけるデッドラインミス率が0であった場合は，偏差は0となり，仮にシステムに余裕がある場合でも動作周波数を下げることができなくなってしまう．そこで目標値には限りなく0に近い値として0.01を設定する．デッドラインミス率は，なるべく早く，かつ，スムーズに0に近づくように制御されることが望ましい．そこでフィードバックコントローラにはPID制御を用いる．デッドラインミス率の目標値を MR_s とすると，時刻 t におけるデッドラインミス率の偏差 $e(t)$ は，時刻 t におけるデッドラインミス率 $MR(t)$ を用いて式(2)で求められる．

$$e(t) = MR_s - MR(t) \tag{2}$$

偏差 $e(t)$ を用いて，フィードバックコントローラが算出する動作周波数の変化量 $\Delta FREQ$ は式(3)で表わされる．式(3)

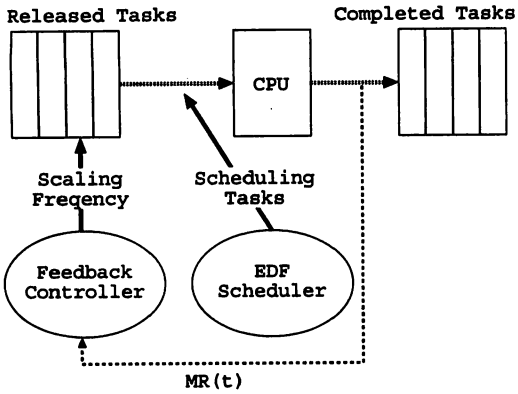

図 1 FC-DFS-MR システムの流れ

における，積分及び微分は，実際の処理では，式 (4) 及び式 (5) で表される．

$$\Delta FREQ = K_P\{e(t) + \frac{1}{T_I}\int e(t)dt + T_D\frac{d}{dt}e(t)\} \quad (3)$$

$$\int e(t)dt = \sum_{IP} e(t) \quad (4)$$

$$\frac{d}{dt}e(t) = \frac{e(t) - e(t-DP)}{DP} \quad (5)$$

ここで，K_P，T_I，T_D は，それぞれ PID 制御における調節可能なパラメータである．積分計算では，時間 IP の間の偏差のみ有効となり，微分計算では時間 DP の間の偏差のみ有効となる．

PID 制御で最も重要なのは，パラメータの決定である．コントローラが適切に設計されていても，パラメータの設定が適切でないと，制御量が目標値に収束せず，安定したシステムを提供することができない．本論文では，繰り返し実験を行い，その中で最も良い性能を示した値をパラメータとした．結果として，$K_P = -1.8$，$T_I = 1$，$T_D = 2$ に設定した．

FC-DFS-MR システムの流れは次の通りである．また，その様子を図 1 に示す．

（1） サンプリング周期毎にフィードバックコントローラによりデッドラインミス率を観測し，PID フィードバック制御式により動作周波数を計算して出力

（2） システムの動作周波数を出力された動作周波数に設定

（3） EDF スケジューラによりタスクをデッドラインが近い順にスケジューリングし，最もデッドラインが近いタスクを実行

4. シミュレーション

評価はシミュレーションを用いて行った．タスクの CPU 利用率は一様分布によって決定される．この際，CPU 利用率の最大値を 10%，最小値を 1% に設定する．CPU 利用率を決定した後は，タスクの周期を {100, 200, ..., 800} からランダムに選択する．タスクの CPU 利用率及び周期が求まれば，実行時間も自動的に求まる．CPU 利用率の合計が 20%, 30%, ..., 90% となる 8 つのタスクセットを生成し，それぞれのタスクセットに対する評価を行った．

表 1 動作周波数と動作電圧のレベル

動作周波数	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
動作電圧	0	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1

表 2 フィードバック (PID) 制御のパラメータ

目標値	サンプリング周期	K_P	T_I	T_D
0.01	800	-1.8	1.0	2.0

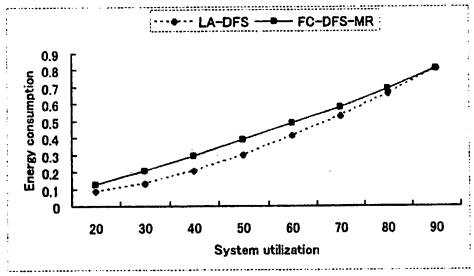

図 2 エネルギー消費の比較

エネルギー消費の評価では，実際のプロセッサは，動作周波数と動作電圧を連続的に変化させることは難しいので，表 1 の，11 レベルの値を仮定した．ここで，最大動作周波数は 1，最大動作電圧は 1 とし，各動作周波数で実行するときには，必ず対応する動作電圧で実行するものとする．FC-DFS-MR は，最初のサンプリング周期ではフィードバック制御を行うことができないため，動作周波数及び動作電圧の初期値は最大動作周波数及び最大動作電圧に設定した．章 1. で述べた通り，CMOS 回路では，エネルギー消費は動作周波数に比例し，かつ動作電圧に比例するので，表 1 の値を用いて，式 (6) により計算した．

$$\text{(エネルギー消費)} = \text{(動作周波数)} \times \text{(動作電圧)}^2 \quad (6)$$

なお，実行しているタスクが存在しない状態では，動作周波数，動作電圧共に 0 と仮定しているのでエネルギー消費も 0 となる．

また，FC-DFS-MR におけるフィードバック制御のパラメータを表 2 に示す．

4.1 エネルギー消費

まず，それぞれのタスクセットにおけるエネルギー消費に関して LA-DFS と比較を行った．CC-DFS は実際の実行時間が最悪実行時間よりも短かい場合にのみ動作周波数を下げることができ，本論文では実際の実行時間が最悪実行時間と等しいことを仮定しているので，評価対象としなかった．その評価結果を図 2 に示す．エネルギー消費の削減という点では，提案手法である FC-DFS-MR の性能は，LA-DFS に最大で約 10% ほど劣っていた．FC-DFS-MR がフィードバック制御によって徐々に動作周波数を落としていくのに対して，LA-DFS はタスクが切り替わる度に実行可能な全てのタスクに対して先読みし，デッドラインミスを起こさないように動作周波数の最小化を試

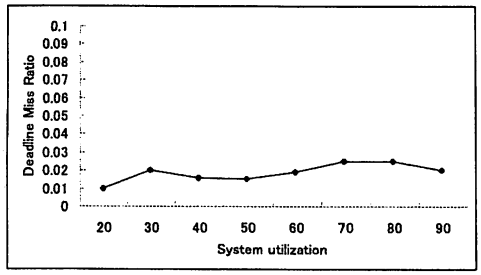

図3 デッドラインミス率

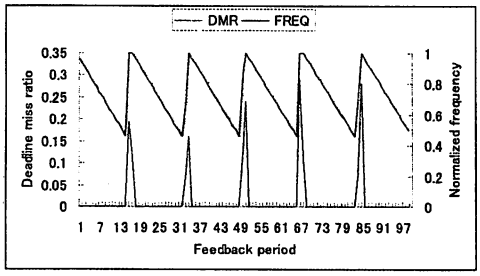

図4 デッドラインミス率と動作周波数の関係

みるため，多くの場合で LA-DFS の性能が FC-DFS-MR を上回ったのだと考えられる．しかしながら，タスクセットの CPU 利用率が増加するにつれて差は縮まっていき，最終的に CPU 利用率が 90% の時点では，その性能はほとんど同じであった．また，実行可能な全てのタスクを先読みするには，再帰的な処理が必要であり，その計算量はタスクセットのタスク数を N とすると $O(N)$ となる．タスクが切り替わる度に $O(N)$ の計算を行うのは，実際のシステムでは現実的ではないと考えられる．一方，FC-DFS-MR は，フィードバック周期ごとに単純な PID 計算を行うだけなので，実行時のオーバーヘッドはほとんど発生しない．よって，FC-DFS-MR は比較的良い性能を示し，かつ，実用性に優れた手法であるといえる．

4.2 デッドラインミス率

次に，図 2 の実験における FC-DFS-MR のデッドラインミス率を計測した．LA-DFS はハードリアルタイムシステムを対象とした手法であるため，デッドラインミスは発生しない．評価結果を図 3 に示す．全体を通して，デッドラインミス率は約 1% ～ 2.5% であった．目標値を 1% として実行したにもかかわらず，いずれのタスクセットにおいても 1% 以上だった原因として，デッドラインミス率は実際にデッドラインミスが発生した時に始めて算出されるので，1% に戻す場合の制御が遅れてしまうことが考えられる．この原因を踏まえれば，評価結果は妥当である．すなわち，FC-DFS-MR は，デッドラインミス率 1% ～ 2.5% を許容できるシステムであれば効果的であるといえる．

4.3 フィードバック制御の様子

最後に，CPU 利用率が 60% のタスクセットを例にとり，FC-DFS-MR のフィードバック制御によるデッドラインミス率と動作周波数の関係を観察した．計測結果を図 4 に示す．エネルギー消費を削減するためにデッドラインミス率が 1% 未満の場合には動作周波数は次第に下がっていく．デッドラインミス率が 1% 以上になった時点で，デッドラインミス率を 1% に戻すように動作周波数の値を調節している．そして，デッドラインミス率が 1% 未満になったら，再び動作周波数を下げていきエネルギー消費を削減している．この結果から，動作周波数を調節してデッドラインミス率が目標値になるように制御ができていることがわかる．

5. 結論及び今後の課題

本論文では，ソフトリアルタイムシステムを対象として，フィードバック制御によりデッドラインミス率を制御しながら，エネルギー消費を削減する FC-DFS-MR を設計し，シミュレーションにより有効性を評価した．評価では，エネルギー消費の削減という点では，提案手法である FC-DFS-MR の性能は，LA-DFS に最大で約 10% ほど劣っている結果となった．これは，フィードバック制御を用いる FC-DFS-MR ではサンプリング周期の度に動作周波数を徐々に下げるが，一方 LA-DFS はタスクのコンテキストスイッチの度に，他のタスクがデッドラインミスを起こさないことを保証した上で可能な限り動作周波数を下げるためである．しかし，LA-DFS が動作周波数の計算に要する計算量は $O(N)$ であり，その計算をコンテキストスイッチの度に行うので，オーバヘッドの面で実用的であるとはいえない．その点，FC-DFS-MR は，PID 制御式による単純な計算をサンプリング周期の度に行うので，LA-DFS より実用的である．また，フィードバック制御により動作周波数を制御して，デッドラインミス率が目標値になるように制御できたことを確認できた．

今後の課題として，PID フィードバックコントローラのパラメータの決定方法の改善が挙げられる．本論文では，繰り返し実験を行うことにより，最も良い性能を示した値をパラメータとした．しかし，この方法では，時間的コストが高くなってしまう．そこで，制御対象を制御モデルとして近似し，そこから計算によってパラメータを求めるという方法が考えられる．また，FC-DFS-MR のフィードバックコントローラでは，デッドラインミス率を制御量として用い，デッドラインミス率が 0 であった場合を考えて，デッドラインミス率の目標値を限りなく 0 に近い 0.01 に設定することで対応した．しかし，この方法ではデッドラインミスが生じる瞬間に近づいたということを検出できないため，デッドラインミスが生じた際のサンプリング周期では，デッドラインミス率がやや高い値となってしまうことが予想される．デッドラインミスが生じる瞬間に近づいたということを検出するためには，制御量としてタスクの最大遅れ時間を設定することが考えられる．遅れ時間とは，タスクの実行が完了した時刻が，デッドラインからどれだけ遅れているかという時間であり，デッドラインミスが起きていない時点では，その値は負の値となる．各サンプリング周期におけるタスクセット中の最大遅れ時間を制御量に設定し，目標値を 0 に近い

負の値に設定すれば，デッドラインミスが近付くにつれ，操作量である動作周波数の変化は緩やかになり，デッドラインミスが減少することが予想される．

謝辞 本論文の研究は，科学技術振興機構 CREST の支援による．

文　献

[1] J.A. Stankovic, C. Lu, S.H. Son and G. Tao: "The Case for Feedback Control Real-Time Scheduling", Proceedings of Euromicro Conference on Real-Time Systems, pp. 11-20 (1999)

[2] C.L. Liu and J.W. Layland: "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment", J. ACM, Vol. 20, pp. 46-61 (1973)

[3] K. Lin, S. Natarajan and J. Liu: "Imprecise Results: Utilizing Partial Computations in Real-Time Systems, Proceedings of IEEE Real-Time Systems Symposium, pp. 210-217 (1987)

[4] P. Pillai and K. Shin: "Real-time dynamic voltage scaling for low-power embedded operating systems", In Symposium on Operating Systems Principles (2001)

[5] Y. Zhu and F. Mueller, "Feedback EDF scheduling exploiting dynamic voltage scaling", Proceedings of the Real-time and Embedded Technology and Applications Symposium, pp. 84-93 (2004)