社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

QoSの公平性を考慮したマルチリソースの適応配分制御

原田　史子† 　潮　 俊光† 　中本　幸一††

† 大阪大学大学院基礎工学研究科　〒 560–8531 豊中市待兼山町 1–3
†† NEC システムプラットフォーム研究所　〒 224–3555 横浜市都築区池部町 4035
E-mail: †harada@hopf.sys.es.osaka-u.ac.jp, †ushio@sys.es.osaka-u.ac.jp, ††nakamoto@ct.jp.nec.com

あらまし　近年，リアルタイムシステムにおいて過負荷状態を避けるために，各タスクに割り当てるリソースを動的に変化させるフィードバック形の制御法が注目されている．筆者らはリソースが CPU しかない場合に，QoS の公平性を達成するようなリソース配分を行う問題について，フィードバック制御を用いたリソース配分制御を提案した．本報告ではこの手法を拡張し，リソースが複数あるシステムに対して QoS の公平性を達成するリソース配分制御を提案する．リソースが複数ある場合は QoS 要求に対する各リソースの要求量がリソース消費関数で決まっており，実際に実行に使われるリソース量が配分された量より小さくなる場合があるが，本手法によりこのようなリソース量をなくし，かつ最大の公平な QoS レベルを達成できる．シミュレーション実験により，リソース消費関数が線形であるようなタスク集合に対して，提案手法が有効であることを確認する．
キーワード　リアルタイムシステム，適応制御，QoS，公平性，フィードバック制御

Adaptive Allocation Control of Multiple Resources Based on Fairness of QoS

Fumiko HARADA†, Toshimitsu USHIO†, and Yukikazu NAKAMOTO††

† Graduate School of Engineering Science, Osaka University,　Machikaneyama 1–3, Toyonaka-shi, 560–8531, Japan
†† NEC System Platform Laboratories, 4035, Ikebe-cho, Tsuzuki-ku, Yokohama, 224–3555, Japan
E-mail: †harada@hopf.sys.es.osaka-u.ac.jp, †ushio@sys.es.osaka-u.ac.jp, ††nakamoto@ct.jp.nec.com

Abstract　Recently, in real-time systems, adaptive resource allocation methods based on feedback control have been proposed in order to avoid overload situations. We proposed an adaptive resource allocation control to achieve fair QoS and to avoid overload situation for CPU resource. In this report, we extend the control to an allocation problem of multiple resources. In multiple resources environment, allocated resrouces can be unused for execution of jobs. Our proposed method guarantees the fair and maximum QoS level without unused resources. We show effectiveness of this method for a task set with linear resource consumption functions by simulation experiment.
Key words　Real-time system, adaptive control, QoS, fairness, feedback control

1. まえがき

リアルタイムシステムにおいて，過負荷状態を避けるために，今までは，タスクの起動要求があったとき，その実行を許可するか否かというアドミッション制御を行ってきた [1]．最近では，タスクに割り当てられたサービス品質（Quality of Service, QoS）に応じて，タスクはリソース利用率を下げてジョブをリリースすることで，過負荷状態を避けるという研究が行われている [2].

一方，処理時に割り当てられるリソースが増加するにしたがって，提供するサービス品質 QoS が向上する，Flexible Application と呼ばれるソフトウェアが存在する [3, pp.394–419]. ここで QoS レベルは，リリースされたジョブの実行結果に対する利用者の満足度のレベルを表す．例えば，MPEG データの再生処理ではイメージ全体を有する I フレームと I フレームからの差分情報を有する B フレーム，P フレームを処理する必要がある．この場合，I フレームのみならず，P フレーム，B フレームの処理を行うタスクに CPU 時間とメモリなどのリソースを割り当てることにより，より精確な MPEG 画像を再生できる．しかし，各ソフトウェアが提供する QoS レベルを同時

に保つ，あるいは向上させようとすると，システムが過負荷状態になる可能性がある．このため，競合する QoS レベルを調停する必要が出てくる．

QoS レベルを調停する研究はこれまでいくつかなされてきた．システム過負荷時のタスクに対して受容可能な QoS レベルを予め記述し，システム過負荷時には，この記述をもとに各タスクとネゴシエーションを行うことにより，各タスクはリリースするジョブの QoS レベルを低下させ，過負荷状態に対処する研究がされている [2]．Buttazzo らは，ジョブの処理がばねのように伸縮するものとみなす弾性モデルを導入し，各タスクの弾性係数に比例して各ジョブの実行時間を短縮することで過負荷を避けるアルゴリズムを提案している [4]．Rajkumar らは，複数のリソースに同時にアクセスする場合に QoS に基づくリソース確保モデルを提案している [5]．またこのモデルに基づき，各アプリケーションソフトウェアの持つ最低限の QoS レベルの要求を満たすという制限下で，QoS レベルを最大化するアルゴリズムを提案している．追川らはリアルタイムソフトウェアの時間上の QoS レベルを保証するためにソフトウェアアーキテクチャとリソース予約を行う API を提案し，ポータブルなカーネルモジュールを開発している [6]．本カーネルモジュールではアドミッション制御機能を提供している．

一方，リアルタイムシステムに制御理論を適用する研究が近年注目されている．[7] では，リアルタイムシステムにおけるタスクの CPU 使用率を管理する手法に，制御理論が利用されている．この研究では，CPU の使用時間を制御するために PID（Propotional-Integral-Derivative）制御器が応用されている．また，タスクの CPU 利用率やデッドラインミス率を指定された値に維持するためにフィードバック制御理論を用いる研究がなされている．例えば [8] では，相対的デッドラインの目標値を与え，実際の相対的デッドラインとその目標値の差を基にして，CPU 時間の予約を PI 制御器によって行うことで，デッドラインミス率をある範囲内に押さえている．[9] では，CPU 利用率またはデッドラインミス率に対して目標値を与え，ゲインフィードバックによる制御法を提案している．いずれの場合にも目標値を前もって与える必要がある．しかしながら，目標値は処理されるべきタスクの集合に依存しており，時間とともにタスクの集合が変動するようなリアルタイムシステムでは，タスクの集合が変動すると目標値の再計算が必要となる．

ところで，過負荷状態を回避するために，同じ割合でタスクのリソース利用率を変化させたとき，QoS レベルの変化はタスクによって異なる．筆者らは既にシングルリソースの場合に QoS レベルの偏りをなくすために，タスクの QoS レベルを公平化するような QoS 適応制御手法を提案している [10]．この手法では前もって目標値を計算する必要がなく，目標値をオンラインで探索しながらその値に近づいていくという特徴をもっている．本報告では，この制御手法を複数リソースの配分問題に拡張する．シミュレーションによりその手法の有効性を確認する．

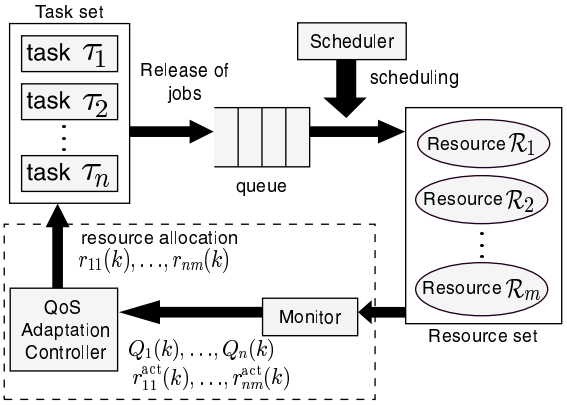

図 1　対象とするリアルタイムシステム

2. 対象システム

本報告では，図 1 のような独立タスク集合 $\{\tau_1, \tau_2, \ldots, \tau_n\}$，スケジューラ，QoS 適応制御器からなり，複数のリソースの集合 $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_m\}$ が存在するリアルタイムシステムを考える．各構成要素は以下の特徴を持つ．

- リソース: リソース $\mathcal{R}_i$ は総量 R_i で配分量を 0 から R_i までの連続値で定められるものとする．
- スケジューラ: 指定されたスケジューリングアルゴリズムに従ってリリースされたジョブのスケジューリングを行う．
- タスク集合 $\{\tau_1, \tau_2, \ldots, \tau_n\}$: タスク τ_i は 0 から 1 までの連続的な QoS レベルを持ち，配分されたリソースに応じた QoS レベルで実行することができる．一般性を失うことなく最小リソース要求を 0 とする．

– $r_{ij}(k)$: 時刻 $[t_k, t_{k+1})$ にリリースされた τ_i のジョブに配分されるリソース $\mathcal{R}_j$ の配分量．但し時刻 t_k は，k 回目に制御器によってリソース配分量が決定された時刻とする．

– $r_{ij}^{\mathrm{act}}(k)$: 時刻 $[t_k, t_{k+1})$ にリリースされた τ_i の実行時に実際に消費されたリソース $\mathcal{R}_j$ の量．例えば，リソース $\mathcal{R}_1$ の配分量 $r_{i1}(k)$ が R_j に近いのに対し，リソース $\mathcal{R}_2$ の配分量 $r_{i2}(k)$ に近い値しかない場合，$r_{i2}(k)$ に対応した QoS レベルでしかジョブを実行できない．このとき，実際に使われるリソース $\mathcal{R}_2$ の消費量は $r_{i2}(k)$ に等しいが，リソース $\mathcal{R}_1$ は実際の QoS レベルに対応した量しか消費されない．ゆえに $r_{i1}^{\mathrm{act}}(k) \leqq r_{i1}(k)$ になる．

– $Q_i(k)$: 時刻 $[t_k, t_{k+1})$ にリリースされた τ_i のジョブの (正規化された)QoS レベル [10]．$r_{ij}(k)$，$r_{ij}^{\mathrm{act}}(k)$，$Q_i(k)$ の関係はリソース消費関数 ϕ_{ij} で以下のようになる．

$$r_{ij}^{\mathrm{act}}(k) = \phi_{ij}(Q_i(k)) \tag{1}$$

$$Q_i(k) =: \min_j \phi_{ij}^{-1}(r_{ij}(k)) \tag{2}$$

$$\forall i = 1, 2, \ldots, n, \exists j, r_{ij}(k) = r_{ij}^{\mathrm{act}}(k) \tag{3}$$

ここでは ϕ_{ij} が微分同相な単調増加関数で次式を満たすとする．

$$0 < h_{ij} \leqq \frac{d\phi_{ij}}{dQ_i} \leqq H_{ij} < \infty \tag{4}$$

$$\phi_{ij}(0) = 0,\ \phi_{ij}(1) \leqq R_j \tag{5}$$

$m > 1$ のとき，QoS レベルが公平でかつ最大であるようなリソース配分は

$$\exists j_u, \sum_{i=1}^{n} r_{ij_u}^{\mathrm{act}}(k) = R_{j_u} \tag{6}$$

$$\sum_{i=1}^{n} r_{ij}^{\mathrm{act}}(k) \leqq R_j,\ j \neq j_u \tag{7}$$

を満たす．これはタスク集合の実行に少なくとも一種類のリソース（上式ではリソース $\mathcal{R}_{j_u}$）が全て使われることを示す．$r_{ij_u} \leqq r_{ij_u}^{\mathrm{act}}$ から

$$r_{ij_u}(k) = r_{ij_u}^{\mathrm{act}}(k),\ i = 1, 2, \ldots, n \tag{8}$$

が成り立っていることが分かる．$\mathcal{R}_{j_u}$ 以外のリソースに関しては式 (8) のような関係が成り立つとは限らないが，リソース利用の効率性からリソース配分量と実際の消費量が異なるのは好ましくない．ゆえに制御目標は

$$Q_1(k) = Q_2(k) = \cdots = Q_n(k) \tag{9}$$

$$\max_{1 \leqq j \leqq m} \left(\frac{\sum_{i=1}^{n} r_{ij}^{\mathrm{act}}(k)}{R_j} \right) = 1 \tag{10}$$

$$r_{ij}(k) = r_{ij}^{\mathrm{act}}(k) = \phi_{ij}(Q_i(k)),\ i = 1, \ldots, n, j = 1, \ldots, m \tag{11}$$

を満たすようなリソース配分を決定することになる．ϕ_{ij} の連続性と単調増加性から，このようなリソース配分は必ず一意に存在する．公平なリソース配分を達成する QoS 適応制御器の制御手法として，次式のような制御則を提案する．

$$\lambda(k) = 1 - \max_{1 \leqq j \leqq m} \left(\frac{\sum_{i=1}^{n} r_{ij}^{\mathrm{act}}(k)}{R_j} \right) \tag{12}$$

$$\underline{h}_j = \min_i h_{ij},\ \overline{Q}(k) = \frac{1}{n} \sum_{i=1}^{n} Q_i(k) \tag{13}$$

$$\begin{aligned} r_{ij}(k+1) = {} & \left\{ 1 + \alpha\big(1 - Q_i(k)\big)\lambda(k) \right\} r_{ij}^{\mathrm{act}}(k) \\ & + \beta \underline{h}_j \big(\overline{Q}(k) - Q_i(k) \big) \end{aligned} \tag{14}$$

この制御は $\overline{Q}(k)$ との偏差を参照することで $Q_1(k), \ldots, Q_n(k)$ のばらつきを小さくするだけではなく，現在最も利用率の高いリソースと現在の QoS レベルに応じてリソースの配分量を増やす．リソースの利用率が 1 のとき，すべてのリソースが配分されているのでこれ以上配分量を増やさない．そうでないとき，QoS の低いタスクほどリソース配分量の増加を大きくする．

3. スケジュール可能性の保証

式 (14) による制御では，スケジュール可能性とリソース制約を保証するように α, β を設定しなければならない．本制御則ではリソース $\mathcal{R}_i$ の総配分量が R_i に保たれるとは限らない．よって全ての時刻 k で

$$\sum_{i=1}^{n} r_{ij}(k) \leqq R_j \tag{15}$$

$$0 \leqq r_{ij}(k) \leqq R_j \tag{16}$$

を満たす必要がある．式 (15)，(16) を満足するための十分条件は次の補題で与えられる．

［補題 1］

$$\sum_{i=1}^{n} r_{ij}(0) \leqq R_j,\ j = 1, 2, \ldots, m \tag{17}$$

$$0 \leqq r_{ij}(0) \leqq R_j,\ i = 1, \ldots, n, j = 1, \ldots, m \tag{18}$$

のもとで

$$0 \leqq \alpha \leqq 1,\ 0 < \beta \leqq \frac{n}{n-1} \tag{19}$$

ならば任意の時刻 k で式 (15),(16) が成り立つ．

(証明) 時刻 k で式 (15),(16) が成り立っていると仮定し，帰納法を用いて示す．

$$\begin{aligned} \alpha \leqq 1 \leqq {} & \frac{R_j}{\sum_{i=1}^{n} r_{ij}^{\mathrm{act}}(k)} \times \frac{1 - \sum_{i=1}^{n} r_{ij}^{\mathrm{act}}(k)/R_j}{1 - \max_q \left(\sum_{p=1}^{n} r_{pq}^{\mathrm{act}}(k)/R_q \right)} \\ & \times \frac{1}{1 - Q_i(k)} \end{aligned}$$

より

$$\begin{aligned} & \big(1 - Q_i(k)\big) \left\{ 1 - \max_q \left(\sum_{p=1}^{n} r_{pq}^{\mathrm{act}}(k)/R_q \right) \right\} \alpha \\ & \quad \leqq \frac{R_j}{\sum_{i=1}^{n} r_{ij}^{\mathrm{act}}(k)} \left\{ 1 - \sum_{i=1}^{n} r_{ij}^{\mathrm{act}}(k)/R_j \right\} \\ \Leftrightarrow \quad & 1 + \big(1 - Q_i(k)\big) \left\{ 1 - \max_q \left(\sum_{p=1}^{n} r_{pq}^{\mathrm{act}}(k)/R_q \right) \right\} \alpha \\ & \quad \leqq \frac{R_j}{\sum_{i=1}^{n} r_{ij}} \\ \Leftrightarrow \quad & 1 + \alpha\big(1 - Q_i(k)\big)\lambda(k) \leqq \frac{R_j}{\sum_{i=1}^{n} r_{ij}^{\mathrm{act}}} \end{aligned}$$

が成り立つ．ゆえに

$$\begin{aligned} \sum_{i=1}^{n} r_{ij}(k+1) = {} & \sum_{i=1}^{n} \Big[\left\{ 1 + \alpha\big(1 - Q_i(k)\big)\lambda(k) \right\} r_{ij}^{\mathrm{act}}(k) \\ & + \beta\big(\overline{Q}(k) - Q_i(k) \big) \Big] \\ \leqq {} & \sum_{i=1}^{n} \frac{R_j}{\sum_{p=1}^{n} r_{pj}^{\mathrm{act}}(k)} r_{ij}^{\mathrm{act}}(k) \\ & + \beta \underline{h}_j \sum_{i=1}^{n} \big(\overline{Q}(k) - Q_i(k) \big) = R_j \end{aligned}$$

となり，式 (15) が満たされる．また，

$$
\begin{aligned}
r_{ij}(k+1) &= r_{ij}^{\mathrm{act}}(k) + \alpha\big(1-Q_i(k)\big)\lambda(k) r_{ij}^{\mathrm{act}}(k) \\
&\quad + \beta \underline{h}_j \big(\overline{Q}(k) - Q_i(k)\big) \\
&\geqq h_{ij}(k) Q_i(k) - \frac{(n-1)\beta \underline{h}_j}{n} Q_i(k) \\
&\quad + \alpha\big(1-Q_i(k)\big)\lambda(k) r_{ij}^{\mathrm{act}}(k) \\
&\quad + \frac{\beta \underline{h}_j}{n} \sum_{p=1, p\neq i}^{n} \big(\overline{Q}(k) - Q_i(k)\big) \\
&\geqq \left\{ \underline{h}_j - \frac{(n-1)\beta \underline{h}_j}{n} \right\} Q_i(k) \\
&\geqq 0,\ i = 1, 2, \ldots, n, j = 1, 2, \ldots, m
\end{aligned}
$$

より式 (16) が成り立つ. □

4. システムの安定性

本節では，式 (14) で決定されるリソース配分が QoS レベルを公平にするような α, β の条件について議論する．式 (14) による制御の平衡点に関して次の事実が成り立つ．

［事実 1］ 式 (1)，(12)，(13)，(14) によって決まる QoS 制御系の平衡点は，$r_{ij}(k) = 0, i = 1, 2, \ldots, n, j = 1, 2, \ldots, m$ なる点と式 (9)，(10)，(11) を満たす点のみである．

(証明) $Q_i(k) = 0, i = 1, 2, \ldots, n$ なる点が平衡点であることは自明である．それ以外の平衡点が式 (9)，(10)，(11) を満たす点だけであることを示す．システムの平衡点を

$$
Q_i(k) = Q_i^f,\ \overline{Q}(k) = \overline{Q}^f,\ r_{ij}(k) = r_{ij}^f,\ r_{ij}^{\mathrm{act}}(k) = r_{ij}^{\mathrm{act},f} \tag{20}
$$

とおくと，$r_{ij}^{\mathrm{act},f} = r_{ij}^f$ かつ $\overline{Q}^f - Q_i^f \geqq 0$ を満たす i, j に関して

$$
\begin{aligned}
r_{ij}^f = &\left\{ 1 + \alpha \left(1 - \max_{1\leqq q\leqq m} \left(\frac{\sum_{p=1}^n r_{pq}^{\mathrm{act},f}}{R_q} \right) \right) (1 - Q_i^f) \right\} r_{ij}^f \\
&+ \beta \underline{h}_j (\overline{Q}^f - Q_i^f) \geqq r_{ij}^f
\end{aligned} \tag{21}
$$

が成り立ち，

$$
\overline{Q}^f - Q_i^f = 0,\ 1 - \max_{1\leqq q\leqq m} \left(\frac{\sum_{p=1}^n r_{pq}^{\mathrm{act},f}}{R_q} \right) = 0 \tag{22}
$$

が得られる．ゆえに，すべての $1 \leqq p \leqq n, 1 \leqq q \leqq m$ について

$$
\overline{Q}^f - Q_p^f = 0,\ p = 1, 2, \ldots, n \Rightarrow Q_1^f = \cdots = Q_n^f \tag{23}
$$

$$
\max_{1\leqq q\leqq m} \left(\frac{\sum_{p=1}^n r_{pq}^{\mathrm{act},f}}{R_q} \right) = 1 \tag{24}
$$

$$
r_{pq}^f = r_{pq}^{\mathrm{act},f} \tag{25}
$$

になる． □

実用上，$r_{ij}(k) = 0, i = 1, 2, \ldots, n, j = 1, 2, \ldots, m$ なるリソース配分を行うことはないので，以下，式 (9)，(10)，(11) を満たす

$$
r_{ij}(k) = r_{ij}^e,\ Q_i(k) = Q^e \tag{26}
$$

の安定性について議論する．この点に対して状態変数を

$$
\hat{Q}_i(k) := Q_i(k) - Q^e,\ \hat{r}_{ij}(k) := r_{ij}(k) - r_{ij}^e \tag{27}
$$

$$
\hat{Q}(k) := \Big[\hat{Q}_1(k),\ \cdots,\ \hat{Q}_n(k) \Big]^T, \tag{28}
$$

$$
\begin{aligned}
\hat{r}(k) := \Big[&\hat{r}_{11}(k),\ \cdots,\ \hat{r}_{1m}(k),\ \hat{r}_{21}(k), \cdots, \\
&\hat{r}_{2m}(k),\ \cdots,\ \hat{r}_{n1}(k),\ \cdots,\ \hat{r}_{nm}(k) \Big]^T
\end{aligned} \tag{29}
$$

$$
\hat{\phi}_{ij}(\hat{Q}_i) := \phi_{ij}(Q_i) - r_{ij}^e \tag{30}
$$

とおく．また，添え字 j_i, j_u をそれぞれ

$$
\hat{r}_{ij_i} = \hat{r}_{ij_i}^{\mathrm{act}}(k) \tag{31}
$$

$$
j_u = \arg \left(\max_{1\leqq j\leqq m} \frac{\sum_{i=1}^n r_{ij}^{\mathrm{act}}(k)}{R_j} \right) \tag{32}
$$

を満たすリソースのインデックスとする．このとき式 (13)，(14) は次のように書き換えられる．

$$
\begin{aligned}
&\hat{r}_{ij}(k+1) = \\
&\left\{ 1 + \alpha \left(1 - \max_q \left(\frac{\sum_{p=1}^n r_{pq}^{\mathrm{act}}(k)}{R_q} \right) \right) \right. \\
&\left. \times \big(1 - Q^e - \hat{Q}_i(k)\big) \right\} \hat{r}_{ij}^{\mathrm{act}}(k) + \beta \underline{h}_j \left(\overline{\hat{Q}}(k) - \hat{Q}_i(k) \right) \\
&+ \alpha \left(1 - \max_q \left(\frac{\sum_{p=1}^n r_{pq}^{\mathrm{act}}(k)}{R_q} \right) \right) \big(1 - Q^e - \hat{Q}_i(k)\big) r_{ij}^e
\end{aligned} \tag{33}
$$

式 (33) を原点まわりで線形化したシステムは次のようになる．

$$
\hat{r}_{ij}(k+1) = A_{ij} \tilde{h}_{ij_i} \hat{r}_{ij_i}(k) + \sum_{p=1, p\neq i}^{n} B_{ij}^p \tilde{h}_{pj_p} \hat{r}_{pj_p}(k) \tag{34}
$$

但し，

$$
\begin{aligned}
A_{ij} =& \frac{d\hat{\phi}_{ij}(0)}{d\hat{Q}_i} - \frac{\beta \underline{h}_j (n-1)}{n} \\
&- \frac{\alpha}{R_{j_u}} (1 - Q^e) r_{ij}^e \frac{d\hat{\phi}_{ij_u}(0)}{d\hat{Q}_i} \\
B_{ij}^p =& \frac{\beta \underline{h}_j}{n} - \frac{\alpha}{R_{j_u}} (1 - Q^e) r_{ij}^e \frac{d\hat{\phi}_{pj_u}(0)}{d\hat{Q}_p} \\
\tilde{h}_{ij_i} =& \frac{d\hat{\phi}_{ij_i}^{-1}(0)}{d\hat{r}_{ij_i}}
\end{aligned}
$$

この線形化システムは固有値として 0 をもち，それ以外の固有値は，行列

$$
A' := \begin{bmatrix}
A_{1j_1} \tilde{h}_{1j_1} & B_{1j_1}^2 \tilde{h}_{1j_1} & \cdots & B_{1j_1}^n \tilde{h}_{1j_1} \\
B_{2j_2}^1 \tilde{h}_{2j_2} & A_{2j_2} \tilde{h}_{2j_2} & \cdots & B_{2j_2}^n \tilde{h}_{2j_2} \\
\vdots & \vdots & \ddots & \vdots \\
B_{nj_n}^1 \tilde{h}_{nj_n} & B_{nj_n}^2 \tilde{h}_{nj_n} & \cdots & A_{nj_n} \tilde{h}_{nj_n}
\end{bmatrix} \tag{35}
$$

の固有値に一致する [15]．したがって，行列 A' の全ての固有値が安定ならば式 (34) の原点は漸近安定となる．これに関して次の補題が成り立つ．

［補題 2］

$$0 < \alpha \cdot \max_{i,j} \frac{H_{ij}}{\underline{h}_j} \cdot \max_{i,j} \frac{H_{ij}}{R_j} \cdot \min_j \frac{nR_j}{\sum_{i=1}^n h_{ij}} \leqq \beta \leqq 1 \quad (36)$$

ならば行列 A' のすべての固有値は安定である．

(証明) Gershgorin の定理 [16] から，行列 A' の無限大ノルムが 1 より小さいことを示せば全ての固有値が安定であることが言える．まず，$\phi_{ij}(0) = 0$ かつ $H_{ij} \geqq d\phi_{ij}/dQ_i \geqq h_{ij}$ より

$$\begin{aligned} R_j \geqq \sum_{i=1}^n r_{ij}^e \geqq \sum_{i=1}^n h_{ij} Q^e \Rightarrow \min_q \frac{R_q}{\sum_{p=1}^n h_{pq}} \geqq Q^e \\ \Rightarrow \min_q \frac{R_q}{\sum_{p=1}^n h_{pq}} H_{ij} \geqq r_{ij}^e \end{aligned} \quad (37)$$

が成り立つ．また

$$\begin{aligned} A_{ij_i} &= \left\{ \frac{d\hat{\phi}_{ij_i}(0)}{d\hat{Q}_i} - \frac{\beta \underline{h}_{j_i}(n-1)}{n} - \alpha(1-Q^e) r_{ij_i}^e \frac{d\hat{\phi}_{ij_u}(0)/d\hat{Q}_i}{R_{j_u}} \right\} \\ &\geqq \left\{ \underline{h}_{j_i} - \frac{\beta \underline{h}_{j_i}(n-1)}{n} - \alpha H_{ij_i} \cdot \min_q \frac{R_q}{\sum_{p=1}^n h_{pq}} \cdot \max_{p,q} \frac{H_{pq}}{R_q} \right\} \\ &\qquad (\because \text{式 (37)}) \\ &= \left\{ \underline{h}_{j_i} - \frac{\beta \underline{h}_{j_i}(n-1)}{n} \right. \\ &\qquad \left. - \alpha \underline{h}_{j_i} \cdot \frac{H_{ij_i}}{\underline{h}_{j_i}} \cdot \min_q \frac{R_q}{\sum_{p=1}^n h_{pq}} \cdot \max_{p,q} \frac{H_{pq}}{R_q} \right\} \\ &\geqq \left\{ \underline{h}_{j_i} - \frac{\beta \underline{h}_{j_i}(n-1)}{n} \right. \\ &\qquad \left. - \alpha \underline{h}_{j_i} \cdot \max_{p,q} \frac{H_{pq}}{\underline{h}_q} \cdot \min_q \frac{R_q}{\sum_{p=1}^n h_{pq}} \cdot \max_{p,q} \frac{H_{pq}}{R_q} \right\} \\ &\geqq \left\{ \underline{h}_{j_i} - \frac{\beta \underline{h}_{j_i}(n-1)}{n} - \frac{\beta \underline{h}_{j_i}}{n} \right\} = \underline{h}_{j_i}(1-\beta) \geqq 0 \end{aligned}$$

$$\begin{aligned} B_{ij_i}^p &= \left\{ \frac{\beta \underline{h}_{j_i}}{n} - \alpha(1-Q^e) r_{ij_i}^e \frac{d\hat{\phi}_{pj_u}(0)/d\hat{Q}_p}{R_{j_u}} \right\} \\ &\geqq \left\{ \frac{\beta \underline{h}_{j_i}}{n} - \alpha \underline{h}_{j_i} \max_{p,q} \frac{H_{pq}}{\underline{h}_q} \min_q \frac{R_q}{\sum_{p=1}^n h_{pq}} \max_{p,q} \frac{H_{pq}}{R_q} \right\} \\ &\geqq \left\{ \frac{\beta \underline{h}_{j_i}}{n} - \frac{\beta \underline{h}_{j_i}}{n} \right\} = 0 \end{aligned}$$

より $A_{ij_i}, B_{ij_i}^p \geqq 0$ が成り立つ．このことから

$$\begin{aligned} \|A'\|_\infty &= \max_i \left[|A_{ij_i} \tilde{h}_{ij_i}| + \sum_{p=1, p \neq i}^n |B_{ij_i}^p \tilde{h}_{ij_i}| \right] \\ &= \max_i \left[\left| \frac{d\hat{\phi}_{ij_i}(0)}{d\hat{Q}_i} - \frac{\beta \underline{h}_{j_i}(n-1)}{n} \right. \right. \\ &\qquad \left. - \frac{\alpha}{R_{j_u}} (1-Q^e) r_{ij_i}^e \frac{d\hat{\phi}_{ij_u}(0)}{d\hat{Q}_i} \right| \frac{d\hat{\phi}_{ij_i}^{-1}(0)}{d\hat{r}_{ij_i}} \\ &\qquad \left. + \sum_{p \neq i} \left| \frac{\beta \underline{h}_{j_i}}{n} - \alpha(1-Q^e) r_{ij_i}^e \frac{d\hat{\phi}_{pj_u}/d\hat{Q}_p}{R_{j_u}} \right| \frac{d\hat{\phi}_{ij_i}^{-1}(0)}{d\hat{r}_{ij_i}} \right] \end{aligned}$$

表 1　シミュレーションにおける各パラメータ h_{ij} と R_j

	R_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}	h_{i6}
$\mathcal{R}_1$	1	0.63	0.98	0.35	0.94	0.62	0.76
$\mathcal{R}_2$	3	2.8	2.47	1.96	2.44	2.76	2.94
$\mathcal{R}_3$	2	1.85	0.78	1.47	1.30	1.95	1.19

各リソース消費関数は切片 0, 傾き h_{ij} の線形関数

$$\begin{aligned} &= 1 - \frac{\alpha}{R_{j_u}} (1-Q^e) r_{ij_i}^e \sum_{p=1}^n \frac{d\hat{\phi}_{pj_u}}{d\hat{Q}_p} \cdot \frac{d\hat{\phi}_{ij_i}^{-1}(0)}{d\hat{r}_{ij_i}} \\ &< 1 \end{aligned}$$

が導かれる．ゆえに A' の全ての固有値は単位円内になる． □

A' の全ての固有値が単位円内に存在すれば，システムの全ての固有値が単位円内に存在することになり，式 (36) によって局所的漸近安定性が保証できる．

補題 1 と補題 2 から，式 (19)，(36) が満たされていれば式 (9)，(15)，(16) が満足される．これをまとめると次の定理が導ける．

［定理 1］（QoS レベル公平化可能条件）

$$\sum_{i=1}^n r_{ij}(0) \leqq R_j, \ j = 1, 2, \ldots, m \quad (38)$$

$$0 \leqq r_{ij}(0) \leqq R_j, \ i = 1, 2, \ldots, n, j = 1, 2, \ldots, m \quad (39)$$

のもとで

$$0 < \alpha \cdot \max_{i,j} \frac{H_{ij}}{\underline{h}_j} \cdot \max_{i,j} \frac{H_{ij}}{R_j} \cdot \min_j \frac{nR_j}{\sum_{i=1}^n h_{ij}} \leqq \beta \leqq 1, \ \alpha \leqq 1 \quad (40)$$

ならば QoS レベルの公平化が可能である．

前述したように，システムは $r_{ij}(k) = 0, i = 1, 2, \ldots, n, j = 1, 2, \ldots, m$ も平衡点なので，公平なリソース配分への収束は局所漸近安定性で与えられる．しかしながら，$r_{ij}(k) \neq 0$ $(j = 1, 2, \ldots, m)$ となるタスク τ_i が一つでもあればこの点から離れていくことが期待できる．このようなリソース配分の不安定性を解析し，最大の公平なリソース配分の安定条件をより緩いものにするのが今後の課題である．

5. シミュレーション実験

本節では，シミュレーション実験により，提案手法の有効性を確認する．対象とするリソース集合 $\{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ に対して，タスク集合 $\{\tau_1, \ldots, \tau_6\}$ は線形のリソース消費関数を持つとする．リソース消費関数の傾き $h_{ij} = H_{ij}$ とリソースの総量 R_j は表 1 に示すとおりである．最適なリソース配分を数値的に解くと，公平化された QoS レベルはおよそ 0.195 になる．式 (40) を満たすように $\alpha = 0.312, \beta = 1$ に設定し，提案手法を適用したときの $Q_i(k)$ の変化を図 2 に示す．$Q_i(k)$ は数値的に解いた Q^e に収束していることがわかる．また，リソース $\mathcal{R}_2$ に対するリソースの総消費量の変化を図 3 に示す．過渡状態では総消費量 $\sum_{i=1}^n r_{i2}^{\mathrm{act}}(k)$ は $R_2 = 3$ より小さいが，定常状態では R_2 に収束し，リソース $\mathcal{R}_2$ が全て消費されていることが分かる．

他のリソース $\mathcal{R}_1$ に対して，$r_{41}(k)$ と $r_{41}^{\mathrm{act}}(k)$ の差の変動を図 4 に示す．これを見ると，過渡状態では制御器に配分されたリソース量と実際の消費量に差があるが， 定常状態では $r_{41}(k)$ と $r_{41}^{\mathrm{act}}(k)$ が等しくなり，実際に消費される量だけが配分されていることがわかる．この結果から，提案手法によって複数のリソースが存在する場合に QoS の公平化が可能であることと，実際に消費されないリソースの配分を回避できることが確認された．

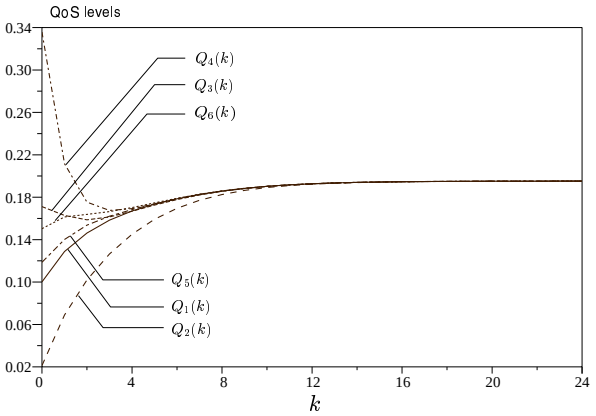

図 2　マルチリソース環境下での QoS レベルの変動

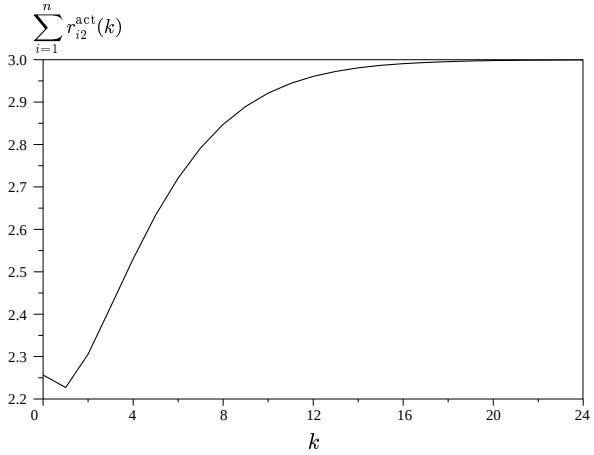

図 3　マルチリソース環境下でのリソース $\mathcal{R}_2$ の総配分量の変動

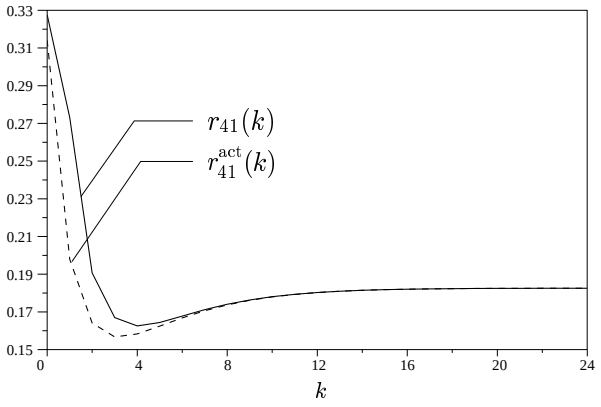

図 4　マルチリソース環境下での $r_{11}(k)$ と $r_{11}^{\mathrm{act}}(k)$ の変動

6. あとがき

本報告では，マルチリソースに対して QoS が公平となるような適応制御法を提案した．提案手法によってリソース配分が望ましい値に収束するための十分条件を求めた．しかし，この条件は，QoS 関数の最小傾き h_{ij} と最大傾き H_{ij} の差が大きくなるほど制御パラメータ α のとりうる範囲が小さくなり、条件が厳しくなる．α を小さく取ると収束が遅くなる．この条件を緩め，収束の速さを改善できるような条件を求めることが今後の課題である．さらに，本手法を複数の QoS 次元を持つリアルタイムシステムに拡張するのも今後の課題である．

文　献

[1] G. Buttazzo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, Kluwer Academic Publishers, Boston, 1997.

[2] T. Abdelzaher, E. Atkins, and K. Shin, “QoS negotiation in real-time systems and its application to automated flight control,” Proc. IEEE Real-Time Technology and Applications Symposium, pp.228–238, June 1997.

[3] J. Liu, Real-Time Systems, Prentice Hall, Upper Saddle River, NJ, 2000.

[4] G. Buttazzo and L. Abeni, “Adaptive workload management through elastic scheduling,” Real-Time Systems, vol.23, No. 1, pp.7–24, 2002.

[5] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, “A resource allocation model for QoS management,” Proc. 18th IEEE Real-Time Systems Symposium, pp.298–307, December 1997.

[6] S. Oikawa and R. Rajkumar, “Portable RK: A portable Resource Kernel for guaranteed and enforced timing behavior, ” Proc. IEEE Real-Time Technology and Applications Symposium, pp.111–120, June 1999.

[7] D. Steere, A. Goel, J. Gruenberg, D. McNamee, C. Pu and J. Walpole, “A feedback-driven proportion allocator for Real-Rate scheduling,” Proceedings of the 3rd Symposium on Operating Systems Design and Implementation, pp.145–158, 1999.

[8] L. Abeni, L. Palopoli, G. Lipari and J. Walpole, “Analysis of a Reservation-Based Feedback Scehduler,” Proc. 23rd IEEE Real-Time Systems Symposium, pp.71–80, December 2002.

[9] C. Lu, J. Stankovic, S. Son, and G. Tao, “Feedback control real-time scheduling: framework, modeling, and algorithms,” Real-Time Systems, vol.23, No.1, pp.85–126, 2002.

[10] 原田，潮，中本，“QoS レベル公平化に基づくリアルタイムシステムの QoS 適応制御，” 信学論，vol.J87-DI, no.3, 2004.

[11] 久保，潮，村田，大崎，“伝播遅延時間を考慮した ATM の PID 型ふくそう制御，” 信学論，vol.J85-B, no.3, pp.371–380, 2002.

[12] B. Li and K. Nahrstedt, “A control-based middleware framework for Quality-of-Service adaptations,” IEEE Journal on Selected Areas in Communications, vol.17, no.9, pp.1632–1650, September 1999.

[13] K. Åström and B. Wittenmark, Computer-Controlled Systems: Theory and Design, Third Edition, Prentice Hall, Upper Saddle River, NJ, 1997.

[14] 白川洋充，竹垣盛一，リアルタイムシステムとその応用，システム制御情報学会 (編)，朝倉書店，2001.

[15] 原田史子，リアルタイムシステムにおけるマルチリソース環境下での fair QoS 制御，大阪大学大学院基礎工学研究科修士論文，2004.

[16] G. Golub and C. Loan, Matrix Computations(3rd edition), Johns Hopkins University Press, Baltimore, 1996.