Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Application of A Consensus Problem to Fair Multi-resource Allocation in Real-time Systems

Naoki Hayashi and Toshimitsu Ushio

Abstract— **This paper provides an application of a consensus problem with nonlinear performance functions to resource allocation in soft real-time systems. In soft real-time systems, fairness of QoS (Quality of Service) levels plays an important role to guarantee appropriate service under timing and resource constraints. We propose fair QoS control with an agent-based controller where each agent manages an allocated resource and a QoS level of each task. We derive an adaptive resource allocation algorithm and sufficient conditions to achieve fair QoS levels by extending results of a consensus problem.**

I. Introduction

Coordinated behavior of multi-agent systems can be observed in diverse fields of engineering and applied sciences such as formation control of Unmanned Aerial Vehicles (UAVs), swarming of animals, and automated transportation systems [1]–[4]. Bertsekas *et al*. investigated asynchronous optimization algorithms for parallel and distributed computing systems [5], [6]. The theoretical framework of a consensus problem was introduced by Jadbabaie *et al*. [7]. They studied swarming behavior with nearest neighbor rules, and showed that heading directions of all agents converge to a common value if information topologies are jointly connected in a bounded time interval. Olfati-Saber and Murray investigated a consensus problem with switching information topologies [8]. They provided conditions for an average consensus with time-delayed communication among agents. Moreau considered a continuous-time nonlinear consensus algorithm for undirected communication networks using set-valued Lyapnov functions [9]. Ren and Beard studied an information consensus problem over time-varying interaction topologies [10]. They investigated a global stability on an information state of agents using a linear consensus algorithm. Xiao and Boyd provided an optimal center-free algorithm for distributed resource allocation with convex cost functions [11]. Xiao and Wang addressed a consensus problem where each agent has a high dimensional state [12]. They presented some necessary and sufficient conditions to reach consensus on a state of agents.

Real-time systems are computing systems whose behaviors depend not only on a logical result of computation but also on a timing constraint. In fact, real-time systems can be classified into two categories with respect to the timing constraint: hard real-time systems and soft real-time systems [13], [14]. Hard real-time systems may have a critical damage when an execution of a task can not be completed before its deadline. On the other hand, soft real-time systems cause performance degradation by deadline miss but may tolerate to continue an operation of the systems. Application of control theory to resource allocation in real-time systems has been widely examined [15], [16]. Marti *et al*. presented several resource management policies using a set of feedback controllers [17]. In real-time systems, performance of an executed task is evaluated as a QoS (Quality of Service) level. QoS levels generally depend on resources allocated to a task such as a CPU utilization, network bandwidth, and a memory size. This means that we need to allocate more resources to tasks so as to improve their QoS levels. However, improving QoS levels of all tasks may cause overload conditions. Rajkumar *et al*. proposed Q-RAM (QoS-based Resource Allocation Model) with multiple QoS dimensions under resource-constrained environment [18]. They also considered a mixed integer programming problem based on Q-RAM to maximize a system utility [19]. Buttazzo *et al*. proposed an adaptive resource management method using an elastic task model to prevent overload conditions [20]. By avoiding overload conditions, however, QoS levels have large variations depending on tasks and some tasks may have to run with unacceptably low QoS levels. Therefore fairness of QoS levels, which implies that tasks have the same QoS levels, plays an important role to guarantee appropriate service for all tasks. The control problem to achieve fair QoS levels avoiding overload conditions is called fair QoS control. Harada *et al*. proposed adaptive fair QoS control in soft real-time systems based on errors between the current QoS levels and their average [21].

This paper considers an application of a consensus problem with nonlinear performance functions to fair QoS control in soft real-time systems. We propose an adaptive resource allocation algorithm with an agent-based controller to achieve fair QoS levels. We model the agent-based controller as a multi-agent system where each agent has information of a resource and a QoS level of each task, and updates the resource according to a consensus protocol. We show sufficient conditions to achieve fair QoS levels by extending results of a consensus problem.

The rest of this paper is organized as follows. Section II reviews some notations of algebraic graph theory. In Section III, we introduce a performance consensus problem. Section IV considers an application of the performance consensus problem to fair QoS control in soft real-time systems. We

The authors are with the Division of Mathematical Science for Social Systems, Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, Japan. na-hayashi@hopf.sys.es.osaka-u.ac.jp ushio@sys.es.osaka-u.ac.jp

This work was supported by Grant-in-Aid for Scientific Research (No. 17360198).

conduct simulation with the EDF (Earliest Deadline First) scheduling in Section V. Finally, we conclude this paper in Section VI.

II. Preliminaries

This section reviews some fundamental facts of graph theory and matrix theory [22], [23].

A. Graph

A directed graph or a digraph $G(V, E)$ consists of a finite and nonempty node set $V = \{v_i \mid i \in \mathcal{I}\}$ and an edge set $E \subseteq V \times V$, where $\mathcal{I} = \{1, 2, \ldots, n\}$. A directed tree is a digraph whose nodes except the root have exactly one parent. A spanning tree of a digraph is a directed tree formed by directed edges that connect all nodes of the tree. A digraph is said to have a spanning tree if the digraph contains a spanning tree as a subgraph. Let $G_i(V, E_i)$ be a possible digraph with a node set V and edge sets E_i $(i = 1, 2, \ldots, M)$. The union of graphs $G = \bigcup_{i=1}^{M} G_i(V, E_i)$ is the digraph with the common node set V and the union of the edge sets $\bigcup_{i=1}^{M} E_i$.

B. Matrix

A vector p $(\in \mathbb{R}^n)$ is said to be positive or nonnegative, denoted as $p > 0$ or $p \geq 0$, if all components of p are positive or nonnegative, respectively. For vectors p $(\in \mathbb{R}^n)$ and q $(\in \mathbb{R}^n)$, $p > q$ and $p \geq q$ stand for $p - q > 0$ and $p - q \geq 0$, respectively. A matrix $P = [p_{ij}]$ $(\in \mathbb{R}^{n \times n})$ is said to be nonnegative, denoted as $P \geq 0$, if $p_{ij} \geq 0$ for all $i,\ j$ $(\in \mathcal{I})$. A (row) stochastic matrix P is a nonnegative matrix which satisfies $\sum_{j=1}^{n} p_{ij} = 1$ for all i.

III. Consensus Problem with Nonlinear Performance Functions

This section introduces a performance consensus problem [24]. In our model, each agent has a high dimensional state and a performance value. We show sufficient conditions for a group of agents to achieve consensus on their performance values.

A. Consensus Protocol

We consider a system with n agents each of which has an m-dimensional state and a performance value. The vector $x_i[k] = [x_{i_1}[k]\ x_{i_2}[k]\ \cdots\ x_{i_m}[k]]^T$ $(\in \mathbb{R}^m)$ represents the state of agent i at time k $(i \in \mathcal{I},\ k = 0, 1, \ldots)$. We assume that $x_{i_\ell}[k]$ $(\in \mathbb{R})$ is in the interval $[x_{i_\ell}^{\inf},\ x_{i_\ell}^{\sup}]$ $(\ell \in \mathcal{L} = \{1, 2, \ldots, m\})$. The performance value of agent i is denoted as $y_i[k]$ $(\in \mathbb{R})$ and characterized by the performance function $f_i(x_i) = f_i(x_{i_1}, x_{i_2}, \ldots, x_{i_m})$. We assume that each performance function $f_i(x_i)$ is strictly increasing and of class C^1 in $\prod_{\ell=1}^{m}[x_{i_\ell}^{\inf},\ x_{i_\ell}^{\sup}]$. We define performance consensus as follows:

Definition 1: A group of agents achieves performance consensus if (1) holds for any $i,\ j$ $(\in \mathcal{I})$ and for any initial state.

$$|y_i[k] - y_j[k]| \ \to\ 0 \quad \text{as} \quad k \ \to\ \infty. \tag{1}$$

To achieve performance consensus, we consider a consensus protocol given by the following equations for each agent i:

$$x_i[k+1] = x_i[k] + b_i[k] \sum_{j=1, j\neq i}^{n} \{p_{ij}[k] g_{ij}[k] (y_j[k] - y_i[k])\}, \tag{2}$$

$$y_i[k] = f_i(x_i[k]), \tag{3}$$

where $p_{ij}[k]$ $(\in \mathbb{R})$ is the positive time-varying weight on the communication from agent j to agent i. The vector $b_i[k]$ is the weight of agent i and defined as $b_i[k] = [b_{i_1}[k]\ b_{i_2}[k]\ \cdots\ b_{i_m}[k]]^T > 0$ $(\in \mathbb{R}^m)$. The variable $g_{ij}[k]$ is a time-varying Boolean value which indicates the existence of the communication from agent j to agent i:

$$g_{ij}[k] = \begin{cases} 1 & \text{if agent } i \text{ receives information from} \\ & \text{agent } j \text{ at time } k, \\ 0 & \text{otherwise,} \end{cases}$$

and $g_{ii}[k] = 0$ for any k. We assume that $p_{ij}[k]$ and $b_{i_\ell}[k]$ are lower and upper bounded.

B. Performance Consensus

In this section, we investigate sufficient conditions for performance consensus.

Let $d_i^{\text{in}}[k]$ be the number of incoming edges to node v_i $(i \in \mathcal{I})$. Note that $x_i[k+1] = x_i[k]$ for any $p_{ij}[k]$ and $b_i[k]$ if $d_i^{\text{in}}[k] = 0$. Thus, we assume that, for each k, there is at least one agent which receives information from others, that is, there exists i such that $d_i^{\text{in}}[k] > 0$ for each k. From (3) and the mean value theorem, there exists $c_{i_\ell}[k]$ such that

$$y_i[k+1] = y_i[k] + \sum_{\ell=1}^{m} \{c_{i_\ell}[k]\,(x_{i_\ell}[k+1] - x_{i_\ell}[k])\}. \tag{4}$$

The supremum $c_{i_\ell}[k]$ with respect to time k is given as follows:

$$c_{i_\ell}^{\sup} = \sup_k c_{i_\ell}[k] = \sup_{x_i} \frac{\partial}{\partial x_{i_\ell}} f_i(x_i). \tag{5}$$

From (4), we can rewrite (2) and (3) as follows:

$$x[k+1] \ = \ x[k] + B[k]P[k]y[k], \tag{6}$$

$$y[k+1] \ = \ y[k] + C[k](x[k+1] - x[k]), \tag{7}$$

where

$$x[k] \ = \ \left[x_1^T[k]\ x_2^T[k]\ \cdots\ x_n^T[k]\right]^T \ \in \mathbb{R}^{mn},$$

$$y[k] \ = \ [y_1[k]\ y_2[k]\ \cdots\ y_n[k]]^T \ \in \mathbb{R}^n,$$

$$P[k] \ = \ \tilde{P}[k] \otimes \mathbf{1}_m \ \in \mathbb{R}^{mn \times n},$$

$$\tilde{P}[k] \ = \ \begin{bmatrix} -\sum_{j=2}^{n} p_{1j}[k] g_{1j}[k] & \cdots & p_{1n}[k] g_{1n}[k] \\ \vdots & \ddots & \vdots \\ p_{n1}[k] g_{n1}[k] & \cdots & -\sum_{j=1}^{n-1} p_{nj}[k] g_{nj}[k] \end{bmatrix} \in \mathbb{R}^{n \times n},$$

$$B[k] \ = \ \text{diag}\,[b_{1_1}[k] \ \cdots\ b_{1_m}[k] \ \cdots\ b_{n_1}[k] \ \cdots b_{n_m}[k]] \in \mathbb{R}^{mn \times mn},$$

$$C[k] = \begin{bmatrix} c_{1_1}[k] & \dots & c_{1_m}[k] & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & c_{n_1}[k] & \dots & c_{n_m}[k] \end{bmatrix} \in \mathbb{R}^{n \times mn},$$

and $\otimes$ is the Kronecker product. From (6) and (7), we have

$$y[k+1] = y[k] + C[k]B[k]P[k]y[k] = W[k]y[k], \qquad (8)$$

where $W[k] = I_n + C[k]B[k]P[k]$ $(\in \mathbb{R}^{n \times n})$ and I_n is an $n \times n$ identity matrix.

In the same way as [24], we can prove the following lemma which guarantees that the matrix $W[k]$ in (8) is a stochastic matrix with positive diagonal entries.

Lemma 1: The matrix $W[k]$ is a stochastic matrix with positive diagonal entries if any edge weight $p_{ij}[k]$ with $d_i^{\text{in}}[k] > 0$ satisfies

$$0 < p_{ij}[k] < \left\{ \left(\sum_{\ell=1}^{m} c_{i_\ell}^{\sup} b_{i_\ell}[k] \right) d_i^{\text{in}}[k] \right\}^{-1}. \qquad (9)$$

The following theorem shows sufficient conditions to achieve performance consensus based on a result of [24].

Theorem 1: A set of agents achieves performance consensus with (2) and (3) if the following conditions hold:

(a) Any edge weight $p_{ij}[k]$ with $d_i^{\text{in}}[k] > 0$ satisfies (9).
(b) There exists an infinite sequence $0 = k_0 < k_1 < k_2 < \cdots$ satisfying conditions (i) and (ii):
  (i) the time interval $[0, \infty)$ is divided into non-overlapping subintervals $[k_0, k_1) \cup [k_1, k_2) \cup \cdots$,
  (ii) the union of interaction graphs in each subinterval $[k_s, k_{s+1})$ has a spanning tree $(s = 0, 1, 2, \dots)$.

IV. Real-time Systems and Fair QoS Control

This section formulates fair QoS control as a performance consensus problem. We propose an agent-based controller where each agent has information of an allocated resource and a QoS level of each task. Each agent in the agent-based controller updates a resource of each task based on a consensus protocol to achieve fair QoS levels.

A. Overall System

We consider a soft real-time system which consists of a set of n periodic tasks $\{\tau_1, \tau_2, \dots, \tau_n\}$ and m independent resources $\{X_1, X_2, \dots, X_m\}$. We assume that n tasks are scheduled on a single processor. Each instance of a task is called a job of the task. A relative deadline of a task is duration between its release time and the time by which its job has to be completed. We also assume that the relative deadline of each task D_i is equal to its period T_i.

B. Multi-agent Based Resource Management

We propose a multi-agent based resource management for fair QoS control as shown in Fig. 1. Each task τ_i periodically releases a job at its release time. The released jobs are put in a queue and a scheduler makes their scheduling. Information exchange of a QoS level is managed by an agent-based controller, where each agent i has information of the allocated resource $x_i[k]$ $(\in \mathbb{R}^m)$ and the monitored QoS level $y_i[k]$ $(\in \mathbb{R})$ of task τ_i. The procedure of agent i from monitoring of the QoS level $y_i[k]$ to resource allocation for task τ_i is illustrated in Fig. 2. Note that, in real-time systems, it is undesirable to change resource allocation of tasks whose jobs are being executed or waiting in a queue. Therefore, we assume that the only tasks whose jobs are neither being executed nor waiting in a queue can exchange their QoS levels. In this paper, we say that an agent is exchangeable at time k if a job of its task is neither being executed nor waiting in a queue at time k. Note also that there are not always other exchangeable agents when some agent exchanges its QoS level. This may fail to achieve fair QoS levels as shown in the following example.

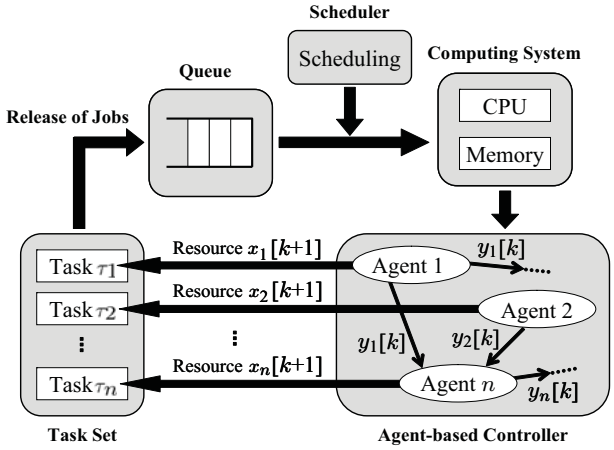

Fig. 1. Multi-agent based resource management for fair QoS control.

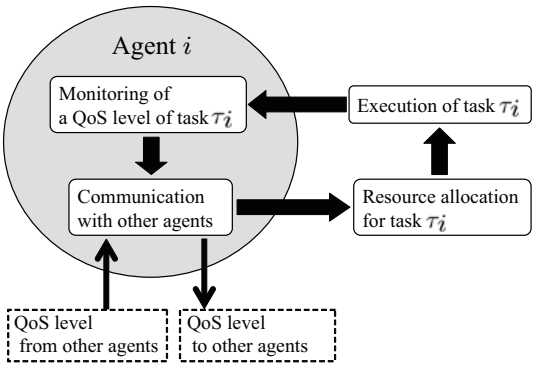

Fig. 2. The procedure of agent i in an agent-based controller.

Example 1: We consider 3 tasks which have the same periods T. We assume that relative deadlines of 3 tasks are equal to their periods. The Gantt chart [1] of these tasks is shown in Fig. 3. A vertical dotted line in the Gantt chart represents a relative deadline of a task. In this example, agent 3 is exchangeable at each time αT $(\alpha = 1, 2, \dots)$. However, agents 1 and 2 are not exchangeable at these times. Thus, agent 3 can never exchange its QoS level $y_3[k]$ and we fail

[1]The Gantt chart illustrates how a set of tasks is scheduled on a processor. The time line of each task in the Gantt chart shows an interval during which its job is executed by a processor.

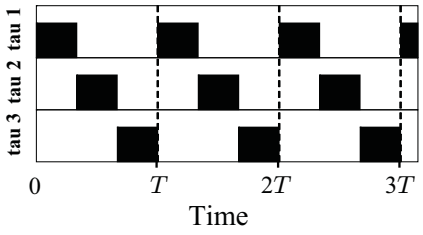

Fig. 3. The Gantt chart of 3 tasks. These tasks have the same relative deadlines which are equal to their periods T. A vertical dotted line represents a release time of a task.

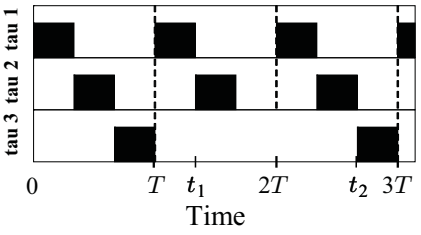

Fig. 4. The Gantt chart of 3 tasks with a job skipping policy. Agent 3 has a token at the initial time 0.

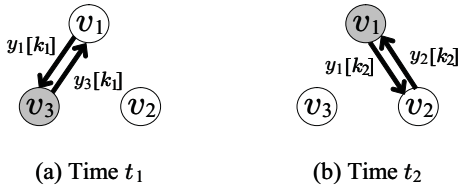

Fig. 5. Information exchanges at times t_1 (a) and t_2 (b). A shaded node represents that the corresponding agent has a token.

to achieve fair QoS levels because we can not satisfy the condition of Theorem 1 (b) which requires that a union of interaction graphs of agents has a spanning tree in some bounded time interval.

To avoid such a situation shown in Example 1, we assume that one of agents in an agent-based controller has a token which gives a priority of information exchange. We assume that agent i has a token and is exchangeable at time t_k. If there are other exchangeable agents at time t_k, agent i exchanges its QoS level with them. Then, agent i updates the resource of task τ_i to $x_i[k+1]$ based on the result of the information exchange, and cyclically passes the token to agent $i+1$. If none of agents except agent i are exchangeable at a release time of task τ_i, agent i skips a release of a job of task τ_i and suspends information exchange until other agents are exchangeable. Note that we can satisfy the conditions of Theorem 1 (b) by a job skipping policy because a token is cyclically passed among agents and there are always other exchangeable agents when some agent exchanges its QoS level. In this paper, we assume that a soft real-time system can accept such an occasional job skipping and does not cause any critical damage provided that most jobs are normally executed.

Example 2: We consider 3 tasks which have the same relative deadlines and periods T. We assume that agent 3 has a token at time 0 and the token is cyclically passed among 3 agents. We also assume that these tasks are scheduled by a job skipping policy to meet the condition of Theorem 1 (b). The Gantt chart of 3 tasks with the job skipping policy is shown in Fig. 4. In this example, none of agents except agent 3 are exchangeable at time T. Thus, agent 3 skips a release of a job of task τ_3 at time T. At time t_1, agents 3 and 1 are exchangeable. Hence, they exchange their QoS levels as shown in Fig. 5 (a), and the token is passed to agent 1 from agent 3. In Fig. 5, each node v_i of a digraph $G(V, E)$ represents agent i and the directed edge (v_i, v_j) indicates that agent j receives information of the QoS level $y_i[k]$ from agent i. At time t_2, agent 1 has the token, and agents 1 and 2 are exchangeable. Thus, they exchange their QoS levels as shown in Fig. 5 (b), and the token is passed to agent 2.

C. Fair QoS Control

The agent-based controller in Section IV-B can be modeled as a multi-agent system where a resource and a QoS level of each task are represented as a state and a performance value of each agent, respectively. We formulate resource allocation of each agent as a consensus protocol given by (2) and (3) with the following resource constraints:

$$\sum_{i=1}^{n} x_{i_\ell}[k] = R_\ell \quad \text{for all } k \text{ and } \ell, \tag{10}$$

where $R_\ell \in \mathbb{R}_+$ and $\mathbb{R}_+$ is the set of all positive real numbers. Fairness of QoS levels corresponds to performance consensus so that achieving fair QoS levels implies achieving consensus on QoS levels. The proposed resource allocation algorithm of agent i with a job skipping policy is summarized in Table I. If agent i has a token and is exchangeable, it exchanges its QoS level with other exchangeable agents. Then, these agents update resources of their tasks based on (2). If none of agents except agent i are exchangeable at a release time of task τ_i, agent i skips a release of a job of task τ_i and suspends information exchange until other agents are exchangeable. Note that the proposed fair QoS control enables asynchronous resource allocation in the sense that agents do not have to suspend information exchange until jobs of all tasks are completed and update their resources simultaneously.

By Theorem 1, we showed that (2) and (3) achieve fair QoS levels. In real-time systems, we also have to consider feasibility of resource allocation. Proposition 1 proves that soft real-time systems satisfy the resource constraints (10) whenever resource allocation is updated.

Proposition 1: Assume that $\sum_{i=1}^{n} x_{i_\ell}[0] = R_\ell$ for all ℓ. Then, all tasks satisfy the resource constraints (10) at all times if information exchange among agents can be modeled as an undirected graph and $b_{1_\ell}[k] = b_{2_\ell}[k] = \cdots = b_{n_\ell}[k] = b_\ell[k]$ for all ℓ.

Proof: Note that information exchange among agents can be modeled as an undirected graph, we have

$$p_{ij}[k]g_{ij}[k] = p_{ji}[k]g_{ji}[k] \quad \text{for all } i,\ j \text{ and } k.$$

TABLE I

RESOURCE ALLOCATION ALGORITHM OF AGENT i WITH A JOB SKIPPING POLICY FOR FAIR QOS CONTROL ($i \in \mathcal{I}$).

Input: Initial resource $x_i[0] > 0$.
Output: Sequence of QoS levels $\{y_i[k]\}$.
for $k = 1,\ 2,\ \ldots$ **do**
  if a job of task τ_i is completed **then**
    Monitor its QoS level $y_i[k]$;
  end if
  if agent i has a token and is exchangeable **then**
    if there are other exchangeable agents **then**
      Exchange its QoS level with them;
      Update a resource to $x_i[k+1]$ based on (2);
      if $i \neq n$ **then**
        Pass the token to agent $i+1$;
      else
        Pass the token to agent 1;
      end if
    else if none of agents except agent i are exchangeable at a release time of task τ_i **then**
      Skip a release of a job of task τ_i;
    end if
  else if agent i and the agent which has a token are exchangeable **then**
    Exchange its QoS level with the agent which has a token;
    Update a resource to $x_i[k+1]$ based on (2);
  end if
end for

We define $\tilde{e}_\ell$ as an $n \times 1$ vector whose ℓ-th component is 1 and other components are all 0. We also define e_ℓ as $e_\ell = [\tilde{e}_\ell\ \tilde{e}_\ell\ \cdots\ \tilde{e}_\ell]^T \in \mathbb{R}^{mn}$. Then, for all k and ℓ, we have

$$\begin{aligned} e_\ell^T B[k]P[k]y[k] &= \left(b_\ell[k]e_\ell^T I_{mn}\right) P[k]y[k] \\ &= b_\ell[k]e_\ell^T P[k]y[k] = 0. \end{aligned} \tag{11}$$

Suppose that $e_\ell^T x[k] = R_\ell$ for some k. From (6) and (11), we have

$$\begin{aligned} e_\ell^T x[k+1] &= e_\ell^T x[k] + e_\ell^T B[k]P[k]y[k] \\ &= e_\ell^T x[k] + 0 = R_\ell. \end{aligned} \tag{12}$$

Thus, all tasks satisfy the resource constraints (10) at all times. ■

The following proposition ensures that resources allocated to a set of tasks are positive at all times.

Proposition 2: Let $x_i[0] > 0$ for all i. The resource allocated to task τ_i is positive at all times if any edge weight $p_{ij}[k]$ with $d_i^{\text{in}}[k] > 0$ satisfies (9), and each QoS function satisfies $f_i(x_i) > 0$ for any $x_i > 0$.

Proof: Suppose that $x_i[k]$ is positive for some k. Since each QoS function satisfies $f_i(x_i) > 0$ for any $x_i > 0$, the QoS level $y_i[k]$ is positive. From Lemma 1, $W[k]$ is a stochastic matrix with positive diagonal entries if (9) holds, and hence $y_i[k+1]$ is also positive. Thus, from the assumption on the QoS function, $x_i[k+1]$ is positive.

Considering the preceding argument and the initial condition $x_i[k] > 0$, we conclude that the resource $x_i[k]$ is positive for all k. ■

V. SIMULATION

In this section, we provide a numerical example with 5 tasks τ_i ($i \in \mathcal{I} = \{1, 2, \ldots, 5\}$) and 3 resources X_ℓ ($\ell \in \mathcal{L} = \{1, 2, 3\}$), where X_1, X_2 and X_3 are a CPU utilization, memory size, and network bandwidth, respectively. Each task τ_i is periodic and its deadline D_i is equal to its period $T_i = 200$. Agent i in an agent-based controller manages resources allocated to task τ_i: the CPU utilization $x_{i_1}[k]$, the memory size $x_{i_2}[k]$ and the network bandwidth $x_{i_3}[k]$. QoS levels are evaluated by the following QoS functions:

$$f_1(x_{1_1}, x_{1_2}, x_{1_3}) = \frac{1}{2}x_{1_1}x_{1_2} + x_{1_3}^2, \tag{13}$$

$$f_2(x_{2_1}, x_{2_2}, x_{2_3}) = \frac{1}{5}x_{2_1}x_{2_3} + \frac{1}{5}\sin\left(\frac{\pi}{4}x_{2_2}\right), \tag{14}$$

$$f_3(x_{3_1}, x_{3_2}, x_{3_3}) = \frac{1}{4}x_{3_1} + \frac{1}{4}x_{3_2}x_{3_3}, \tag{15}$$

$$f_4(x_{4_1}, x_{4_2}, x_{4_3}) = \frac{1}{5}x_{4_1} + \frac{1}{5}x_{4_2}^2 + \frac{1}{5}x_{4_3}, \tag{16}$$

$$f_5(x_{5_1}, x_{5_2}, x_{5_3}) = \frac{1}{10}x_{5_1}x_{5_2} + \frac{1}{4}x_{5_2}x_{5_3}. \tag{17}$$

We set the weights $p_{ij}[k]$ and $b_{i_\ell}[k]$ as follows:

$$\begin{aligned} p_{ij}[k] = 0.8 \times \min\Bigg\{ &\left\{\left(\sum_{\ell=1}^{3} c_{i_\ell}^{\sup} b_{i_\ell}[k]\right) d_i^{\text{in}}[k]\right\}^{-1}, \\ &\left\{\left(\sum_{\ell=1}^{3} c_{j_\ell}^{\sup} b_{j_\ell}[k]\right) d_j^{\text{in}}[k]\right\}^{-1}\Bigg\}, \end{aligned} \tag{18}$$

$$b_{i_\ell}[k] = 1 \quad \text{for all } i,\ \ell \text{ and } k. \tag{19}$$

The EDF (Earliest Deadline First) algorithm is a well-known dynamic scheduling algorithm which gives the highest priority to the task with the earliest deadline [13], [14]. Suppose that all tasks are periodic and relative deadlines are equal to their periods. Then, it is known that the EDF scheduling algorithm can meet a deadline of each task if and only if

$$\sum_{i=1}^{n} x_{i1}[k] \leq 1.0 \quad \text{for all } k, \tag{20}$$

where $x_{i1}[k]$ is the CPU utilization of task τ_i and n is the number of tasks [25]. To satisfy the schedulability condition (20), we give the following resource constraints:

$$\sum_{i=1}^{5} x_{i_1}[k] = 1.0 \ \text{ for all } k, \tag{21}$$

$$\sum_{i=1}^{5} x_{i_2}[k] = 2.0 \ \text{ for all } k, \tag{22}$$

$$\sum_{i=1}^{5} x_{i_3}[k] = 1.5 \ \text{ for all } k. \tag{23}$$

Then, we have $c_{i_\ell}^{\sup}$ as shown in Table II where $0 \leq x_{i_1} \leq 1.0$, $0 \leq x_{i_2} \leq 2.0$ and $0 \leq x_{i_3} \leq 1.5$ ($i \in \mathcal{I},\ \ell \in \mathcal{L}$).

Fig. 6 illustrates QoS levels of 5 tasks with the EDF scheduling algorithm. The Gantt chart from time 0 to time 2000 is shown in Fig. 7. We assume that agent 5 has a token at time 0 and the token is cyclically passed among 5 agents. In this simulation, none of agents except agent 5 are exchangeable at time 200. Thus, agent 5 has to skip a release

TABLE II

VARIABLES $c_{i_\ell}^{\sup}$ $(i \in \mathcal{I},\ \ell \in \mathcal{L})$.

$c_{1_1}^{\sup}$	$c_{2_1}^{\sup}$	$c_{3_1}^{\sup}$	$c_{4_1}^{\sup}$	$c_{5_1}^{\sup}$
1.0	0.3	0.25	0.2	0.2

$c_{1_2}^{\sup}$	$c_{2_2}^{\sup}$	$c_{3_2}^{\sup}$	$c_{4_2}^{\sup}$	$c_{5_2}^{\sup}$
0.5	0.05π	0.375	0.8	0.475

$c_{1_3}^{\sup}$	$c_{2_3}^{\sup}$	$c_{3_3}^{\sup}$	$c_{4_3}^{\sup}$	$c_{5_3}^{\sup}$
3.0	0.2	0.5	0.2	0.5

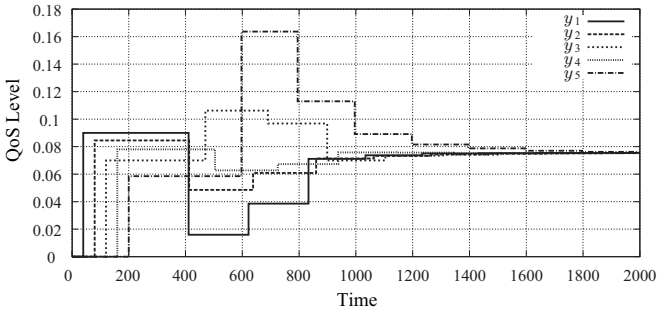

Fig. 6. QoS levels of 5 tasks from time 0 to time 2000 with the EDF scheduling.

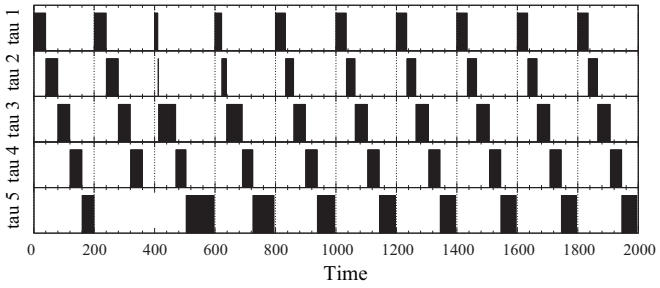

Fig. 7. The Gantt chart from time 0 to time 2000. Agent 5 has a token at time 0.

of a job of task τ_5 at time 200. At time 240, a job of task τ_1 is completed, and agents 5 and 1 are exchangeable. Hence, they exchange their QoS levels, and the token is passed to agent 1 from agent 5. In the same way, agent 1 has the token, and agents 1 and 2 are exchangeable at time 280. Thus, they exchange their QoS levels, and the token is passed to agent 2. Figs. 6 and 7 show that the set of tasks $\{\tau_1, \tau_2, \ldots, \tau_5\}$ achieves fair QoS levels without any deadline miss.

VI. CONCLUSIONS

This paper considered an application of a performance consensus problem to fair QoS control in soft real-time systems. We proposed an adaptive resource allocation algorithm with an agent-based controller where each agent has information of an allocated resource and a QoS level of each task. We provided sufficient conditions to achieve fair QoS levels by extending results of a consensus problem. In multi-resource fair QoS control, there are several feasible solutions for fair QoS levels. Future work includes optimization on the fair QoS levels.

REFERENCES

[1] W. Ren, R. W. Beard, and E. M. Atkins, "A Survey of Consensus Problems in Multi-agent Coordination," *Proceedings of the 2005 American Control Conference*, vol. 3, pp. 1859–1864, 2005.

[2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[3] W. Ren, R. W. Beard, and E. M. Atkins, "Information Consensus in Multivehicle Cooperative Control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[4] S. Martinez, J. Cortes, and F. Bullo, "Motion Coordination with Distributed Information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, 2007.

[5] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athens, "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

[7] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[8] R. Olfati-Saber and R. M. Murray, "Consensus Problems in Networks of Agents With Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[9] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[10] W. Ren and R. W. Beard, "Consensus Seeking in Multiagent Systems Under Dynamically Changing Interaction Topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[11] L. Xiao and S. Boyd, "Optimal Scaling of A Gradient Method for Distributed Resource Allocation," *Journal of Optimization Theory and Applications*, vol. 129, no. 3, pp. 469–488, 2006.

[12] F. Xiao and L. Wang, "Consensus problems for high-dimensional multi-agent systems," *Control Theory and Applications*, vol. 1, no. 3, pp. 830–837, 2007.

[13] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications Series*. Springer, 2005.

[14] J. W. S. Liu, *Real-Time Systems*. Prentice Hall, 2000.

[15] D. C. 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.

[16] C. Lu, J. A. Stankovic, G. Tao, and S. H. Son, "Feedback Control Real-Time Scheduling: Framework, Modeling, and Algorithms," *Real-Time Systems*, vol. 23, no. 1-2, pp. 85–126, 2002.

[17] P. Marti, C. Lin, S. A. Brandt, M. Velasco, and J. M. Fuertes, "Optimal State Feedback Based Resource Allocation for Resource-Constrained Control Tasks," *Proceedings of the 25th IEEE Real-Time Systems Symposium*, pp. 161–172, 2004.

[18] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, "A Resource Allocation Model for QoS Management," *Proceedings of the 18th IEEE Real-Time Systems Symposium*, pp. 298–307, 1997.

[19] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, "Practical Solutions for QoS-Based Resource Allocation Problems," *Proceedings of the 19th IEEE Real-Time Systems Symposium*, pp. 296–306, 1998.

[20] G. C. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni, "Elastic Scheduling for Flexible Workload Management," *IEEE Transactions on Computers*, vol. 51, no. 3, pp. 289–302, 2002.

[21] F. Harada, T. Ushio, and Y. Nakamoto, "Adaptive Resource Allocation Control for Fair QoS Management," *IEEE Transaction on Computers*, vol. 56, no. 3, pp. 344–357, 2007.

[22] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[23] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer Verlag, 2001.

[24] N. Hayashi and T. Ushio, "Performance Consensus Problem of Multi-agent Systems with Multiple State Variables," *IEICE Transactions on Fundamentals of Electronics*, vol. E91-A, no. 9, pp. 2403–2410, 2008.

[25] C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *Journal of the Association for Computing Machinery*, vol. 20, no. 1, pp. 46–61, 1973.