IEICE TRANS. INF. & SYST., VOL.E92–D, NO.2 FEBRUARY 2009

LETTER

# Schedulability Analysis on Generalized Quantum-Based Fixed Priority Scheduling*

**Moonju PARK**[†a)], ***Member***

**SUMMARY** This letter analyzes quantum-based scheduling of real-time tasks when each task is allowed to have a different quantum size. It is shown that generalized quantum-based scheduling dominates preemption threshold scheduling in the sense that if tasks are schedulable by preemption threshold scheduling then the tasks must be schedulable by generalized quantum-based scheduling, but the converse does not hold. To determine the schedulability of tasks in quantum-based scheduling, a method to calculate the worst case response time is also presented.
***key words:*** *real-time scheduling, quantum-based scheduling, fixed priority*

## 1. Introduction

Real-time system is defined as a system whose correctness of result is a function of time the result is delivered. Each task has "deadline" before which the task must complete execution. There are two kinds of scheme in scheduling real-time tasks. Fixed priority schedulers assign a priority to each task once and for all. In dynamic priority scheduling, the priorities of tasks are assigned at runtime. Although the Earliest Deadline First, which is a dynamic priority scheduling algorithm, is optimal for preemptive and non-preemptive real-time scheduling on a uniprocessor [1], fixed priority scheduling has been more widely used in real-time systems because it is very easy to implement [2].

In general, preemptive schedulers give better schedulability than non-preemptive schedulers. However, it has been shown that in the context of fixed priority scheduling, preemptive schedulers are not always superior to non-preemptive schedulers in [3], [4].

Preemption threshold scheduling [4] is a dual priority scheduling scheme that improves schedulability of fixed priority tasks. Introduction of a preemption threshold allows a task to disable preemption according to its preemption threshold, which may make the task schedulable, even though the task was not schedulable either by preemptive or non-preemptive schedulers. Further studies of preemption threshold scheduling can be found in [5], [6], and the application of the preemption threshold to dynamic priority scheduling is discussed in [7].

Another approach to enhance the schedulability in fixed priority scheduling is quantum-based scheduling. Gopalakrishnan and Parulkar studied both threshold preemption and quantum-based scheduling (called delayed preemption in their work), and reported empirical results that show quantum-based scheduling can increase processor utilization with RM priority assignment [8]. In [9], Anderson *et al.* defined quantum-based scheduling as follows: "Under quantum-based scheduling, processor time is allocated to tasks in discrete time units called quanta. When a processor is allocated to some task, that task is guaranteed to execute without preemption for $q$ time units, where $q$ is the length of the quantum, or until it terminates, whichever comes first." Quantum-based scheduling was applied to dynamic priority scheduling for efficient object sharing [9], flow protection in communication [10], and reduction of the scheduling overhead [11].

In this letter, quantum-based scheduling is studied in a more general form: each task is allowed to have its own quantum size that may be different from others. We define "generalized quantum-based scheduling" as quantum-based scheduling that allows different quantum sizes for different tasks. Adoption of generalized quantum-based scheduling to dynamic priority scheduling was analyzed in [11]. This letter analyzes generalized quantum-based scheduling for fixed priority scheduling. We show that generalized quantum-based scheduling dominates preemption threshold scheduling in the sense that a task set that is not schedulable by preemption threshold scheduling may be schedulable with quantum-based fixed priority scheduling, but all task sets that are schedulable by preemption threshold scheduling must be schedulable with generalized quantum-based scheduling.

The rest of this letter is organized as follows. Section 2 explains the task model in this letter. In Sect. 3, the worst case response time of a task in generalized quantum-based scheduling is calculated. Section 4 shows generalized quantum-based scheduling dominates preemption threshold scheduling. Finally, Sect. 5 concludes this work.

## 2. System Model

A task is denoted by $\tau_i$, and each $\tau_i$ is a three-tuple $(T_i, D_i, C_i)$ where $T_i$ is the period, $D_i$ is the relative deadline, and $C_i$ is the worst case execution time. It is easy to see that $D_i \geq C_i > 0$. Time is represented by an integer. Therefore time is discrete and clock ticks are indexed by integers, as in [1]. This requires that if the first instance of $\tau_i$ is invoked at time $t_x$, the following instances are in-

Manuscript received May 15, 2008.
Manuscript revised August 25, 2008.
†The author is with the Department of Computer Science and Engineering, University of Incheon, Korea.
*This work was supported by the University of Incheon Research Grant in 2008.
a) E-mail: mpark@incheon.ac.kr
DOI: 10.1587/transinf.E92.D.332

voked periodically at $t_x + kT_i$, where $k = 1, 2, 3, \ldots$, and $\tau_i$ must be allocated $C_i$ units of processor time in the interval $[t_x + (k-1)T_i, t_x + (k-1)T_i + D_i]$. Each $\tau_i$ is assigned a priority $\pi_i$. Without loss of generality, we can assume that $\pi_i$ is higher than $\pi_j$ if $i < j$. When $\pi_i$ is higher than $\pi_j$, it is denoted as $\pi_i \prec \pi_j$.

A concrete task has a specified release time, or the time of the first invocation. The difficulty of scheduling tasks can be affected by the release time [1]. When an instance of $\tau_i$ is invoked at time $t$ and finishes at time $t'$, $t' - t$ is the "response time" of the instance. The worst case response time (WCRT) of $\tau_i$ is defined as the maximum possible response time among all instances of $\tau_i$. We will denote the WCRT of $\tau_i$ as $WCRT_i$. A task $\tau_i$ is schedulable by a scheduling algorithm $S$ if and only if $WCRT_i \leq D_i$ when $\tau_i$ is scheduled by $S$. A concrete task set is schedulable by $S$ if and only if any concrete task is schedulable by $S$. A periodic task set is defined as schedulable by $S$ if and only if all concrete task sets that can be generated from the periodic task set are schedulable by $S$. We consider only periodic task sets in this letter.

For a given task $\tau_i$, we define $hp(\tau_i)$ as a subset of a task set, which consists of tasks with priority higher than or equal to $\tau_i$ (except $\tau_i$). On the other hand, $lp(\tau_i)$ is the set of tasks with lower priority than $\tau_i$. By definition, $lp(\tau_n) = \phi$ when the number of tasks is $n$. Tasks are all independent and cannot be suspended by themselves.

A quantum is a task variable, which is a unit of non-preemptive execution. The quantum of a task $\tau_i$ is denoted as $q_i$. When a task $\tau_i$ is scheduled, $\tau_i$ can execute during $q_i$ without being preempted by other tasks, unless it finishes its execution before the quantum expires. Thus, $\tau_i$ is non-preemptive for $q_i$ time units. If the task finishes before the quantum expires, another task is selected to execute. If $\tau_i$ does not finish after execution of $q_i$ units, it can be preempted by any ready task in $hp(\tau_i)$. If the quantum size $q_i = 1$, quantum-based scheduling of $\tau_i$ is equivalent to preemptive scheduling. On the other hand, if $q_i = C_i$, scheduling of $\tau_i$ is equivalent to non-preemptive scheduling. We define "feasible quantum" of $\tau_i$ as a quantum size which makes $WCRT_i \leq D_i$ in current scheduling settings.

In preemption threshold scheduling, each task is assigned another value called preemption threshold. The preemption threshold of a task $\tau_i$ is denoted as $\gamma_i$. $\gamma_i$ must be a priority value in the range of $[\pi_1, \pi_i]$. When a task $\tau_i$ is released, it competes for the processor at priority $\pi_i$. After $\tau_i$ starts its execution, another task $\tau_j$ can preempt $\tau_i$ if and only if $\pi_j \prec \gamma_i$. If $\gamma_i = \pi_i$, scheduling of $\tau_i$ is equivalent to preemptive scheduling. If $\gamma_i = \pi_1$, it is equivalent to non-preemptive scheduling. If $\tau_i$ is schedulable with $\gamma_i$ in current scheduling settings, $\gamma_i$ is called a feasible preemption threshold.

## 3. Worst Case Response Time of Quantum-Based Scheduling

In this section, we present a method to calculate the WCRT of a task in generalized quantum-based scheduling. By comparing the WCRT of a task with its deadline, we can determine whether a task is schedulable or not in current scheduling settings. It is shown in [3] that the concept of level-$i$ busy period is useful for calculation of the WCRT in non-preemptive scheduling. Level-$i$ busy period is defined as a processor busy period in which only instances of tasks in $hp(\tau_i)$ and $\tau_i$ execute [12].

**Lemma 1:** The WCRT of a task $\tau_i$ in non-preemptive fixed priority scheduling is found in a level-$i$ busy period obtained by simultaneously releasing $\tau_i$ with all tasks in $hp(\tau_i)$ at time 0 and the task which has the longest execution time in $lp(\tau_i)$ at time $-1$.

Lemma 1 tells us the longest level-$i$ busy period in non-preemptive scheduling is given by:

$$L_i = \max_{\tau_j \in lp(\tau_i)} \{C_j - 1\} + \sum_{\tau_j \in hp(\tau_i) \cup \{\tau_i\}} \left\lceil \frac{L_i}{T_j} \right\rceil C_j \qquad (1)$$

as shown in [3]. The calculation of the WCRT in generalized quantum-based scheduling is also based on the following analysis of non-preemptive scheduling in [3].

**Theorem 1:** For non-preemptive tasks with arbitrary fixed priorities, the worst case response time of any task $\tau_i$ is given by

$$WCRT_i = \max_{k=0,\ldots,K_i} \{w_{i,k} + C_i - kT_i\}$$

where

$$w_{i,k} = kC_i + \sum_{\tau_j \in hp(\tau_i)} \left(1 + \left\lfloor \frac{w_{i,k}}{T_j} \right\rfloor\right) C_j + B_i \qquad (2)$$

where $K_i = \lfloor L_i/T_i \rfloor$, $B_i = \max_{\tau_m \in lp(\tau_i)}\{C_m\} - 1$, and $L_i$ is given in Eq. (1).

In Theorem 1, $kC_i$ stands for the duration of the $k$ instances of $\tau_i$ released before $kT_i$. The second term stands for the maximum workload of tasks in $hp(\tau_i)$ in the interval $[0, w_{i,k}]$, and $B_i$ is the maximum delay caused by tasks in $lp(\tau_i)$. Once it has gained the processor at time $w_{i,k}$, the $(k+1)$-th instance of $\tau_i$ completes its execution by time $w_{i,k} + C_i$. Its response time is therefore $w_{i,k} + C_i - kT_i$.

Based on Theorem 1, the following theorem shows the WCRT of a task in generalized quantum-based scheduling can be calculated with the same time complexity. Note that $(C_i - 1)\ mod\ q_i + 1$ and $\left\lfloor \frac{C_i - 1}{q_i} \right\rfloor q_i$ are used to avoid the case $C_i\ mod\ q_i = 0$.

**Theorem 2:** The worst case response time of a task $\tau_i$ with quantum size $q_i$ is given by

$$WCRT_i = \max_{k=0,\ldots,K_i} \{w_{i,k} + ((C_i - 1)\ mod\ q_i) + 1 - kT_i\}$$

where

$$w_{i,k} = kC_i + \sum_{\tau_j \in hp(\tau_i)} \left(1 + \left\lfloor \frac{w_{i,k}}{T_j} \right\rfloor\right) C_j + \left\lfloor \frac{C_i - 1}{q_i} \right\rfloor q_i + B_i$$

where $K_i = \lfloor L_i/T_i \rfloor$ and $B_i = \max_{\tau_m \in lp(\tau_i)}\{q_m\} - 1$.

**Proof:** The maximum delay due to a lower priority task $\tau_j$ cannot exceed $q_j - 1$ in quantum-based scheduling. Thus, it is clear that $B_i = \max_{\tau_m \in lp(\tau_i)}\{q_m\} - 1$.

Let us consider a task set constructed by substituting $\tau_i$ with subtasks $\{\tau_{ij} = (T_i, D_i, C_{ij})\}$ where $C_{ij} = \min\{q_i, C_i - \sum_{k=1}^{j-1} C_{ij}\}$ and $j = 1, \ldots, \lceil C_i/q_i \rceil$. Let $\lceil C_i/q_i \rceil = Q_i$ and set all of $\tau_{ij}$ to have the same priority as $\tau_i$. Let $\tau_{ix}$ be the last task which finishes its job when all of $\tau_{ij}$ are simultaneously released. Then, $WCRT_i$ is the worst case response time of a task $\tau_{ix}$.

Without any loss of generality, we can assume that $\tau_{ix} = \tau_{iQ_i}$. By definition, $\tau_{iQ_i}$ has the worst case execution time of $((C_i - 1) \ mod \ q) + 1$, and we have $C_i - ((C_i - 1) \ mod \ q_i + 1) = \left\lfloor \frac{C_i - 1}{q_i} \right\rfloor q_i$. By Theorem 1,

$$\begin{aligned} w_{iQ_i,k} &= kC_i + \sum_{\tau_{jm} \in hp(\tau_{ik})} \left(1 + \left\lfloor \frac{w_i}{T_j} \right\rfloor\right) C_{jm} + B_i \\ &= kC_i + \sum_{\tau_j \in hp(\tau_i)} \left(1 + \left\lfloor \frac{w_{iQ_i,k}}{T_j} \right\rfloor\right) C_j \\ &\quad + \sum_{k=1}^{Q_i - 1} \left(1 + \left\lfloor \frac{w_{iQ_i,k}}{T_i} \right\rfloor\right) C_{ik} + B_i \\ &= kC_i + \sum_{\tau_j \in hp(\tau_i)} \left(1 + \left\lfloor \frac{w_{iQ_i,k}}{T_j} \right\rfloor\right) C_j + \left\lfloor \frac{C_i - 1}{q_i} \right\rfloor q_i + B_i. \end{aligned}$$

Thus the worst case response time is

$$WCRT_{iQ_i} = \max_{k=0,\ldots,K_i} \{w_{iQ_i,k} + C_{iQ_i} - kT_i\}. \quad (3)$$

Since $WCRT_i = WCRT_{iQ_i}$ and $w_{i,k} = w_{iQ_i,k}$,

$$\begin{aligned} WCRT_i &= WCRT_{iQ_i} \\ &= \max_{k=0,\ldots,K_i} \{w_{iQ_i,k} + C_{iQ_i} - kT_i\} \\ &= \max_{k=0,\ldots,K_i} \{w_{i,k} + ((C_i - 1) \ mod \ q) + 1 - kT_i\}. \end{aligned}$$

Therefore, the theorem is proved. ■

## 4. Comparison with Preemption Threshold Scheduling

In this section, schedulability of generalized quantum-based scheduling is compared with preemption threshold scheduling. We show that if a task is schedulable by preemption threshold scheduling, then it must be schedulable by generalized quantum-based scheduling. To this end, we first show that the blocking time (maximum delay caused by lower priority tasks) of each task in preemption threshold scheduling is larger than or equal to that in generalized quantum-based scheduling. Let us denote $B_i^q$ and $B_i^t$ as the blocking time experienced by $\tau_i$ in quantum-based scheduling and that in preemption threshold scheduling respectively. Also, we denote $WCRT_i^q$ and $WCRT_i^t$ as the WCRT of $\tau_i$ in quantum-based scheduling and that in preemption threshold scheduling respectively.

**Lemma 2:** For any set of feasible preemption threshold $\{\gamma_i\}$, we can find a set of feasible quantum $\{q_i\}$ resulting in smaller or equal blocking time for each task than preemption threshold scheduling.

**Proof:** We show the lemma by induction. Suppose that there are $n$ tasks in decreasing order of priority ($\pi_i \prec \pi_j$ if $i < j$). Note that $\tau_n$ has $B_n^q = B_n^t = 0$.

(1) Basis step: for a feasible preemption threshold $\gamma_n$, there is a feasible quantum $q_n$ such that $B_{n-1}^q \leq B_{n-1}^t$.

If $\gamma_n = \pi_n$, $WCRT_n^t$ is equal to $WCRT_n^q$ with $q_n = 1$. Thus if $\gamma_n = \pi_n$, we can see that $B_{n-1}^q = B_{n-1}^t = 0$. If $\gamma_n \prec \pi_n$, since $q_n \leq C_n$, then $B_{n-1}^q \leq B_{n-1}^t$. Therefore there must be a feasible quantum $q_n$ such that $B_{n-1}^q \leq B_{n-1}^t$.

(2) Inductive step: for feasible preemption thresholds $\gamma_i, \ldots, \gamma_n$ and feasible quantum $q_i, \ldots, q_n$, if $B_j^q \leq B_j^t$ for $i - 1 \leq j \leq n$, then there is a feasible quantum $q_{i-1}$ such that $B_{i-2}^q \leq B_{i-2}^t$ for a feasible preemption threshold $\gamma_{i-1}$.

If $\gamma_{i-1} = \pi_{i-1}$, then $B_{i-2}^t = B_{i-1}^t$ because $\tau_{i-1}$ cannot block $\tau_{i-2}$. On the other hand, if $q_{i-1} = 1$, we get $B_{i-2}^q = B_{i-1}^q$. Thus, when $\gamma_{i-1} = \pi_{i-1}$, $B_{i-2}^q \leq B_{i-2}^t$ since $B_{i-1}^q \leq B_{i-1}^t$. If $\gamma_{i-1} \prec \pi_{i-1}$, $B_{i-2}^t = \max\{C_{i-1}, B_{i-1}^t\}$. Because $B_{i-2}^q = \max\{q_{i-1}, B_{i-1}^q\}$ and $q_{i-1} \leq C_{i-1}$, we have $B_{i-2}^q \leq B_{i-2}^t$ in this case. Therefore there must be a feasible quantum $q_{i-1}$ such that $B_{i-2}^q \leq B_{i-2}^t$. ■

Based on Lemma 2 and the fact that $WCRT_i^t$ is smallest when $\gamma_i = \pi_1$, the following theorem shows generalized quantum-based scheduling can successfully schedule task sets that are schedulable by preemption threshold scheduling.

**Theorem 3:** If a task set is schedulable by preemption threshold scheduling, then it is schedulable by quantum-based scheduling.

**Proof:** The proof is by induction. Suppose that there are $n$ tasks in decreasing order of priority. We first show that if $\tau_n$ is schedulable by preemption threshold scheduling, then $\tau_n$ has a feasible quantum.

Let us assume that $\tau_n$ is schedulable with a preemption threshold assigned, but has no feasible quantum. Note that $\tau_n$ has $B_n = 0$. In this case, $WCRT_n^q > D_n$ even if $q_n = C_n$. It leads to a contradiction because when $q_n = C_n$, $WCRT_n^q$ is the same as $WCRT_n^t$ with $\gamma_n = \pi_1$. Therefore if $\tau_n$ is schedulable with a preemption threshold, it must have a feasible quantum.

(1) Basis step: if $\tau_n$ and $\tau_{n-1}$ are schedulable by preemption threshold scheduling, then $\tau_{n-1}$ must have a feasible quantum.

Let us assume that $\tau_{n-1}$ has no feasible quantum but it is schedulable with a preemption threshold. By Lemma 2, we can find $q_n$ which gives $B_{n-1}^q \leq B_{n-1}^t$. Furthermore, $q_n$ is a feasible quantum as shown above. By assumption, $WCRT_{n-1}^q > D_i$ even though $q_{n-1} = C_{n-1}$. But since $B_{n-1}^q \leq B_{n-1}^t$, $WCRT_{n-1}^q \leq WCRT_{n-1}^t$ even if $\gamma_{n-1} = \pi_1$. It leads to a contradiction. Thus, if $\tau_{n-1}$ is schedulable in preemption threshold scheduling then it must have a feasible quantum.

(2) Inductive step: if each task $\tau_i, \ldots, \tau_n$ is schedulable

**Table 1** Worst case response time of tasks.

| Task | $T_i$ | $D_i$ | $C_i$ | WCRT Preemptive | WCRT Non-preemptive | WCRT Threshold | WCRT with quantum=20 |
|---|---|---|---|---|---|---|---|
| $\tau_1$ | 70 | 50 | 25 | 25 | **59** | 44 | 44 |
| $\tau_2$ | 80 | 80 | 20 | 45 | 79 | 79 | 64 |
| $\tau_3$ | 200 | 100 | 35 | **125** | 80 | **105** | 80 |

by preemption threshold scheduling, then if $\tau_{i-1}$ has a feasible preemption threshold, $\tau_{i-1}$ must have a feasible quantum.

By Lemma 2, we can find feasible quantum $q_{i-1}, \ldots, q_n$ such that $B_j^q \le B_j^t$ for $j = i-1, \ldots, n$. Thus when $\gamma_{i-1} = \pi_1$ and $q_{i-1} = C_{i-1}$, $WCRT_{i-1}^q \le WCRT_{i-1}^t$. So if $\tau_{i-1}$ has no feasible quantum, it must be $WCRT_{i-1}^t > D_i$ even though $\gamma_{i-1} = \pi_1$, which contradicts the assumption. Thus if $\tau_{i-1}$ is schedulable with a preemption threshold assigned, $\tau_{i-1}$ must have a feasible quantum. ■

Now we show that there is a task set which is not schedulable by preemption threshold scheduling but schedulable by quantum-based scheduling. Suppose that there are three tasks $\tau_1 = (70, 50, 25)$, $\tau_2 = (80, 80, 20)$, and $\tau_3 = (200, 100, 35)$. This task set is not schedulable by preemption threshold scheduling as shown in Table 1. The WCRT of $\tau_3$ is 125 (> 100) when $\gamma_3 = \pi_3$ (WCRT Preemptive in the table) and 105 (> 100) when $\gamma_3 = \pi_2$ (WCRT Threshold in the table). Finally, if we let $\gamma_3 = \pi_1$ (WCRT Non-preemptive in the table), the WCRT of $\tau_1$ is 59 (> 50). Thus this task set is not schedulable by preemption threshold scheduling.

By choosing an appropriate quantum size, however, we can make the tasks schedulable. Among many possible values of the quantum size, as an example, we can choose 20 for all tasks as their quantum size, which makes the task set schedulable. As shown in Table 1, tasks become schedulable with $q_i = 20$.

## 5. Conclusion

In this letter, quantum-based fixed priority scheduling is analyzed when each task may have a different quantum size. An efficient method to calculate the worst case response time of tasks is presented. Also, it is shown that quantum-based fixed priority scheduling dominates preemption threshold scheduling in the sense that all tasks schedulable by preemption threshold scheduling must be schedulable by quantum-based scheduling, and we can find a task set that is not schedulable by preemption threshold scheduling, but schedulable by quantum-based scheduling.

### References

[1] K. Jeffay, D.F. Stanat, and C.U. Martel, "On non-preemptive scheduling of periodic and sporadic tasks," Proc. IEEE Real-Time Systems Symposium, pp.129–139, Dec. 1991.
[2] E. Bini, G.C. Buttazzo, and G.M. Buttazzo, "Rate monotonic analysis: The hyperbolic bound," IEEE Trans. Comput., vol.52, no.7, pp.933–942, 2003.
[3] L. George, N. Riviere, and M. Spuri, "Preemptive and non-preemptive real-time uniprocessor scheduling," Technical Report, INRIA, 1996.
[4] Y. Wang and M. Saksena, "Scheduling fixed-priority tasks with preemption threshold," Proc. 6th International Conference on Real-Time Computing Systems and Applications, pp.328–335, Dec. 1999.
[5] R. Ghattas and A.G. Dean, "Preemption threshold scheduling: Stack optimality, enhancements and analyis," Proc. 13th IEEE Real-Time and Embedded Technology and Applications Symposium, pp.147–157, April 2007.
[6] S. Kim, S. Hong, and T.H. Kim, "Integrating real-time synchronization schemes into preemption threshold scheduling," Proc. 5th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, p.145, April 2002.
[7] D.Z. He, F.Y. Wang, and W. Li, "Dynamic preemption threshold scheduling for specific real-time control systems," Proc. IEEE International Conference on Networking, Sensing and Control, pp.395–400, March 2005.
[8] R. Gopalakrishnan and G.M. Parulkar, "Bringing real-time scheduling theory and practice closer for multimedia computing," Proc. ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp.1–12, May 1996.
[9] J.H. Anderson, R. Jain, and K. Jeffay, "Efficient object sharing in quantum-based real-time systems," Proc. 19th IEEE Real-Time Systems Symposium, pp.346–355, Dec. 1998.
[10] T.M. Lim, B.S. Lee, and C.K. Yeo, "Quantum-based earliest deadline first scheduling for multiservices," IEEE Trans. Multimedia, vol.9, no.1, pp.157–168, 2007.
[11] S. Baruah, "The limited-preemption uniprocessor scheduling of sporadic task systems," Proc. 17th Euromicro Conference on Real-Time Systems, pp.137–144, July 2005.
[12] J.P. Lehoczky, "Fixed priority scheduling of periodic task sets with arbitrary deadlines," Proc. IEEE Real-Time Systems Symposium, pp.201–209, 1990.