Scheduling Control Tasks with Threshold-based Largest Dedication First

Hong Jin, Hongan Wang, Hui Wang, and Danli Wang

Institute of Software, Chinese Academy of Sciences,
Beijing 100080, P. R. China

{hjin, wha, hui.wang, dlwang}@iel.iscas.ac.cn

Abstract

Scheduling design of control tasks with fuzzy deadline is considered. By introducing the dedication concept to describe the scheduling criticality of a control task, the largest dedication first with preemption threshold is presented and the preemption threshold is used to control the context switching among control tasks. Every control task is dynamically assigned its priority and feasible preemption threshold that are dependent on its dedication degree and not restricted by the number of tasks. Simulation shows that the scheduling of controller tasks with fuzzy deadline can get implemented with less control performance cost and better scheduling performance. The proposed concept and scheduling algorithm can be widely applied to the design of computer-controlled systems, enrich researches in control and real-time scheduling theories further, and facilitate the application of real-time control technology.

Keyword: Scheduling Algorithm, Preemption Threshold, Fuzzy Deadline.

I. Introduction

The cross-fertilisation of control and scheduling gets more attention during 20th century [1]. The major contribution in this area was from the research group led by Arzen, K. E. [2][3]. Existing researches assume that all timing constraints of a control task are known and precise. However, in computer-controlled systems, the imprecise clock, overload or computer faults can lead to a relative variation or uncertainty for the sampling period or all. Some constraints thus can be considered as fuzzy or uncertainty, such as the deadline of a controller task can change randomly in an interval of its sampling period, as well as fuzzy slack, fuzzy execution time and fuzzy criticality. All these concerns were considered in [4][5]. We can use lingual terms, fuzzy model, statistical model or other uncertain models to describe uncertain timing constraints of control tasks, e.g., the sampling period of 5 seconds for a control task, which means that 5±0.1% seconds for the deadline admitted to have a small random or fuzzy variation around the sampling period can be considered to be acceptable although the sampling period is precise [6]. So, it is very important for us to study a scheduling method for control tasks with fuzzy-deadline in computer-controlled systems.

In this paper, the task scheduling with fuzzy deadline is considered. In existing researches, the deadline was usually defined as a fuzzy number or described by using a statistical model, the sigmoid function, and discrete random variable [6][7]. In following discussion, the trapezoid fuzzy number is used to describe the deadline of a controller task. From the control point of view, it is first to guarantee the stability of computer-controlled systems before implementing the scheduling of fuzzy control tasks. Under the assumption of the control performance, we study the scheduling design of controllers with fuzzy deadline again.

In the co-design of control and scheduling with fuzzy timing constraints, we need to consider the control performance cost as well as the scheduling performance, e.g., the missed deadline rate. In addition, the efficient utilization rate of central processing unit (CPU) is also important, and context switching may affect the performance stability of a control system. So controlling context switching is important and considered in achieving the scheduling design of controller tasks with fuzzy attributes, such that the waste of CPU resource is also avoided. Here the preemption threshold is adopted to control the number of context switching among control tasks in the proposed scheduling policy, which is especially important for preemptive multi-tasks in designing and implementing of embedded real-time software. In the proposed scheduling algorithm, dynamic assignment schemes of preemption threshold are presented and integrated into the co-design of control and scheduling because the proposed priority assigned for control task is dynamic [8].

The paper is arranged as follows: Section II discusses the fuzzy deadline model of controller task; the dedication concept and the threshold-based largest dedication first (TLDF) algorithm are presented in Section III; Section IV gives performance cost indexes and simulation comparisons; finally, the conclusion is given.

II. Fuzzy Deadline of Controller

A. *Controller Tasks*

In the computer-controlled system described in [2], the dispersion of the reference input $r(t)$ with the system output $y(t)$ is used as the input of the controller which recalculates the manipulated variable $u(t)$ used as the input of $G(s)$. The control aim is to make $y(t)$ follow $r(t)$ as closely and quickly as possible. The state update of the manipulated variable (Update State) and the calculation of system output (Calculate Output) will make up of the close loop of the whole system. The controller in computer-controlled system is considered as a control task.

In real-time scheduling community, the simplest model assumption about a controller task in computer-controlled systems is that the controller task is periodic or can be transformed to a periodic task, and has a fixed period, a known worst-case execute time and a hard deadline [2]. For the ith controller task T_i, in the following discussion, let P_i, d_i, C_i, p_i and h_i be the sampling period, the deadline, the execute time, the priority and the threshold respectively.

B. *Fuzzy Deadline*

In this paper, the trapezoid membership function is introduced to describe the fuzzy deadline as done in [4][9]. Consider the fuzzy expression of d_i.

Let $d_i \equiv \text{Trapezoid}(a_i, a_i', b_i', b_i)$ be the $[a_i, b_i]$-cut trapezoid fuzzy number of d_i, where $0 < a_i \le a_i' \le b_i' \le b_i \le P_i$, $\mu_i(t)$ be the trapezoid membership function:

$$\mu_i(t) = \begin{cases} (t-a_i)h_i/(a_i'-a_i) & a_i < t \le a_i' \\ h_i & a_i' < t \le b_i' \\ (b_i - t)h_i/(b_i - b_i') & b_i' < t \le b_i \\ 0 & others \end{cases}$$

And d_i is also called as $[a_i, b_i]$-cut trapezoid fuzzy deadline, where a_i and b_i are called as the earliest deadline and latest deadline of T_i respectively here. When $h_i = 2/[(b_i - a_i) + (b_i' - a_i')]$, $\int \mu_i(t)dt = 1$. Especially, d_i becomes to be of triangle fuzzy number as $a_i' = b_i'$ or uniform fuzzy number as $a_i = a_i'$ and $b_i' = b_i$.

III. Threshold-Based Largest Dedication First

A. Priority Assignment

Definition 1: $\forall x \in R^+$, the dedication degree of T_i as it will finish execution before the time of x can be defined as follows [9]:

$$Ded_i(x) \equiv \int_{-\infty}^{x} \mu_i(t)dt \Big/ \int_{-\infty}^{+\infty} \mu_i(t)dt$$

For the fuzzy deadline with $[a_i, b_i]$-cut membership function as described in above section, the above dedication function, for $x \in [a_i, b_i]$, will become,

$$Ded_i(x) \equiv \int_{a_i}^{x} \mu_i(t)dt \Big/ \int_{a_i}^{b_i} \mu_i(t)dt$$

Remark 1. The dedication value of T_i is between 0 and 1. If it can finish before its earliest deadline, its dedication is equal to 0, i.e., $Ded_i(x)=0$ for $x \le a_i$; if it can not finish before its latest deadline, its dedication is equal to 1, i.e., $Ded_i(x)=1$ for $x \ge b_i$.

Definition 2: Let t be the current time, $C_i(t)$ be the current remained execute time of task T_i, the current dedication of T_i can be defined as $Ded_i(t+C_i(t))$.

For the ideal case of no other tasks preempting resource, T_i will finish execution at $t+C_i(t)$. The current dedication can be used as the index scaling the criticality of a control task at the current time, denoted as Dedication INdex (*DIN*) of T_i, $DIN_i(t)=Ded_i(t+C_i(t))$.

Property 1: $C_i(t)$ is monotone decreasing.

Property 2: $Ded_i(x)$ is monotone increasing.

Property 3: The dedication index $DIN_i(t)$ is monotone increasing.
Proof: According to Property 1, for any t_1 and t_2 (and $t_2>t_1$), we have $C_i(t_1)-C_i(t_2) \le t_2-t_1$ or $t_2+C_i(t_2) \ge t_1+C_i(t_1)$, where the equation is met only when T_i continues to execute in the interval

of $[t_1, t_2]$ and is not preempted by other tasks. Then due to Property 2, $Ded_i(t_2+C_i(t_2)) \geq Ded_i(t_1+C_i(t_1))$ or $DIN_i(t_2) \geq DIN_i(t_1)$.

Definition 3: The priority of T_i, can be defined as its current dedication, $p_i(t)=DIN_i(t)$.

Remark 2. The larger the dedication of a task is, the higher its priority will be.

Remark 3. According to Definition 3, the higher the priority of a task is, the earlier it will be scheduled, i.e., tasks with high dedication will get scheduled first.

Remark 4. According to the proof of Property 3, the dedication of a task will maintain changeless as it is executing, and increase as it is preempted by other tasks.

B. Threshold Assignment

According to Remark 4, the executing task will maintain unchanged priority in scheduling, however, those tasks waiting for execution will have increasing priority. Once one of waiting tasks has a higher priority than the executing task, it will preempt resource to execute. According to Remark 1, this kind of preemption among tasks, known as the thrashing [8], will happen frequently and cause context switching.

Thrashing problem will increase overhead in system memory as well as causes the waste in CPU bandwidth, and will limit the application of the proposed algorithm, which is serious for control stability in computer-controlled systems. Therefore, we propose to enact preemptive constraints (i.e., preemption threshold). By using preemption threshold to control unnecessary preemption, the system overhead and context switching will get reduced [10].

Definition 4: According to Remark 3, the preemption threshold of a task is defined as its preempted priority upper bound. It can be preempted by another task with a priority larger than its preemption threshold.

Remark 5. The preemption threshold of a task is larger than or equal to its priority.

Remark 6. Let p_i and h_i be the priority and preemption threshold of T_i respectively, according to Remark 1, Remark 5 and Definition 3, it is meaningless for h_i larger than 1, so the interval of h_i is $[p_i, 1]$.

Remark 7. According to Remark 4, the preemption threshold changes along with the priority, and can't be determined in advance or off-line as done in [10].

Definition 5 [8]: The finished rate of a task is defined as the ratio of its finished time with its total execution time, where the finished time is the time it has executed.

Let p_i and h_i be the priority and preemption threshold of T_i respectively, two assignment schemes of h_i are given as follows.

Scheme 1: The preemption threshold (h_i) can be considered as a linear function of the priority (p_i). For example, for $\varepsilon \in [0, 1]$, $h_i=\varepsilon+(1-\varepsilon)p_i$.

Scheme 2: The preemption threshold (h_i) can be considered as a nonlinear function of the priority (p_i). For example, let f_i be the finished time of T_i. For a given referred finished rate, $\alpha \in [0, 1]$, $h_i = p_i$ if $f_i < \alpha$; otherwise $h_i = 1$.

IV. Simulation

A. Conditions

The computer-controlled system composed of two servo systems is considered. Each servo system can be described by using the same transfer function of $G(s)=1000/[s(s+1)]$, and controlled by a proportional-derivative controller (PD). The PD controllers for these two serve systems have the same discrete control algorithm form as described in [3]:

$$u(t)=Prop(t)+Der(t)$$
$$Prop(t)=K[r(t)-y(t)] \text{ and } Der(t)=\alpha_d Der(t-P)+\beta_d[y(t-P)-y(t)]$$

where, $\alpha_d=M/(NP+M)$, $\beta_d=NKM/(NP+M)$, K and M are control parameters, N is constant, P is the sampling period.

Controller 1 (T_1): P_1=14ms, K=1, M=0.04, N=100 and C_1=7ms;

Controller 2 (T_2): P_2=12ms, K=1.2, M=0.03, N=80 and C_2=6ms.

They execute as independent tasks. Their release times are all zero, and they have fuzzy deadlines of $d_1 \equiv Trapezoid(7,8,9,11)$(ms) and $d_2 \equiv Trapezoid(6,7,8,9)$(ms) respectively. TS=2s. The reference input of every subsystem is taken as a step-function, e.g., $r(t)=1$ when $t \leqq 0.5$s or $1\text{s} < t \leqq 1.5$s; or $r(t)=-1$ in other time interval.

For the case of the deadline not equal to the sampling period, the actual requested CPU utilization is $U=(C_1/d_1)+(C_2/d_2)$. Because d_i is of fuzzy uncertainty and can vary from its earliest deadline a_i to latest deadline b_i, the latest deadline b_i can be used instead of the actual deadline d_i to yield the lower bound of the utilization for the most optimistic case, $U \geq (C_1/b_1)+(C_2/b_2)=43/33>1$.

No matter how d_i varies in $[a_i, b_i]$, the utilization is always larger than 1. So the system is overloaded such that no scheduling policy can guarantee all instances of these two controller tasks to meet their latest deadlines. It goes without saying that their deadlines can be met.

B. Performance Indexes

Let $y_{ideal,i}$ and $y_{actual,i}$ be the required output and actual output of the ith control system respectively. An integrated performance cost of the whole multiple-task control system can be described as follows (CPC: Control Performance Cost):

$$CPC = \sum_{i=1}^{n} w_i CPC_i \text{ and } CPC_i = \int_0^{TS} \left(y_{ideal,i}(t) - y_{actual,i}(t)\right)^2 dt$$

where, CPC_i and w_i are the control performance cost of the ith control system and its weighed coefficient respectively, n is the number of controller tasks, TS is the simulation time.

Three scheduling indexes are introduced and used in following simulation evaluation:
1). Efficient Utilization Rate (EUR), defined as the ratio of the occupied CPU time by instances met their latest deadlines with the total simulation time *100%;

2). Context Switching Number (CSN), defined as the preempted number of deadline-missed tasks;

3). Missed Deadline Rate (MDR), defined as the ratio of the number of deadline-missed tasks with the total number of tasks;

4). Task Committed Rate (TCR), defined as the ratio of the number of committed tasks with the total number of tasks.

For the scheduling of control tasks with hard deadline, MDR=1-TCR. In addition, other performance index is value rate (i.e., all committed tasks contribute to the system) determined by the value or criticality of a control task.

For co-design of control and scheduling of multi-task in computer-controlled systems, it is always to hope that the total control performance cost, the missed deadline rate and the context switching number are smaller and smaller, but the efficient utilization rate is greater and greater. For different scheduling policy, the following integrated index can be considered.

$$COST=CPC/[(1-MDR)EUR] \text{ or } COST=CPC/[(1-MDR)(1-CSN)EUR]$$

If the system is not overload, the CPU utilization is used instead of EUR. For the proposed TLDF, it is used to choose the parameter ε or α in the scheme of threshold choice.

C. *Performance Comparisons*

Fig.1 and Fig.2 present the simulation results by using Scheme 1, and Fig.3 and Fig.4 by using Scheme 2. Fig.1 and Fig.3 give comparisons of CPC between System 1 and System 2, Fig.2 and Fig.4 give comparisons of MDR. Fig.5 gives comparison of EUR for two Schemes, and Fig.6 gives comparison of CSN.

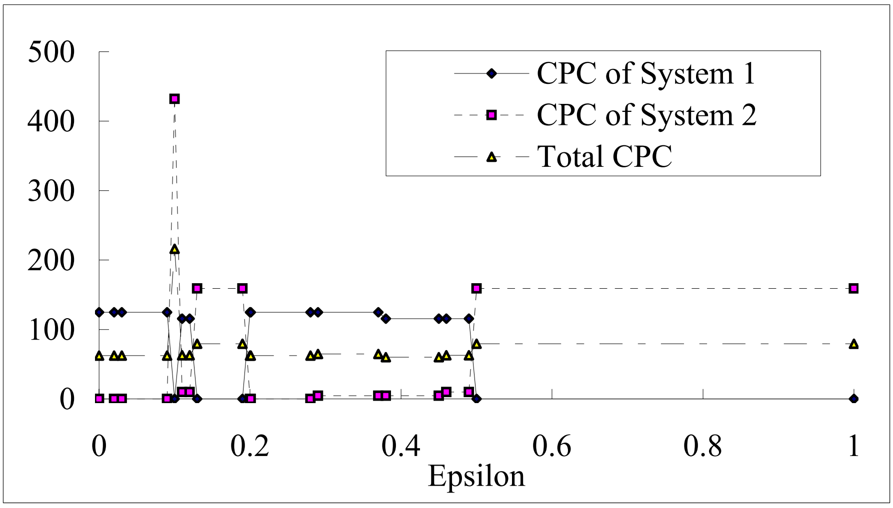

Figure 1. Comparison of CPC for Scheme 1

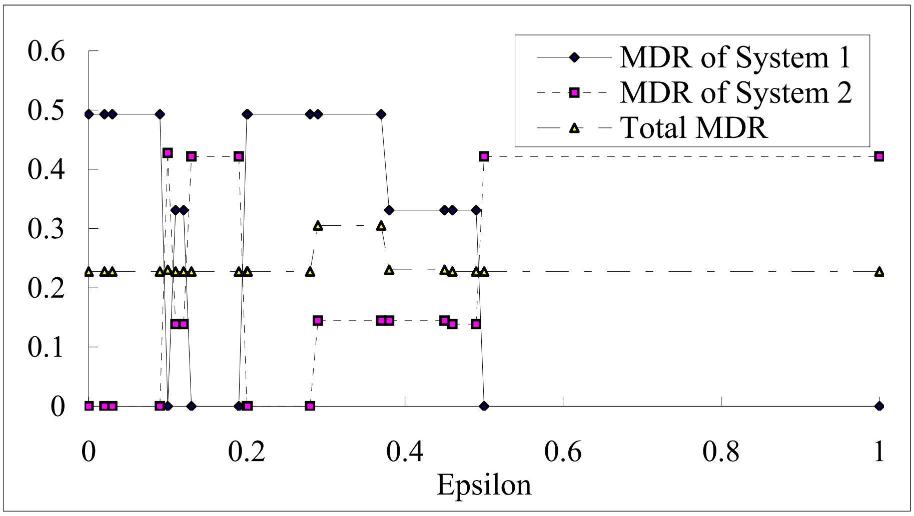

Figure 2. Comparison of MDR for Scheme 1

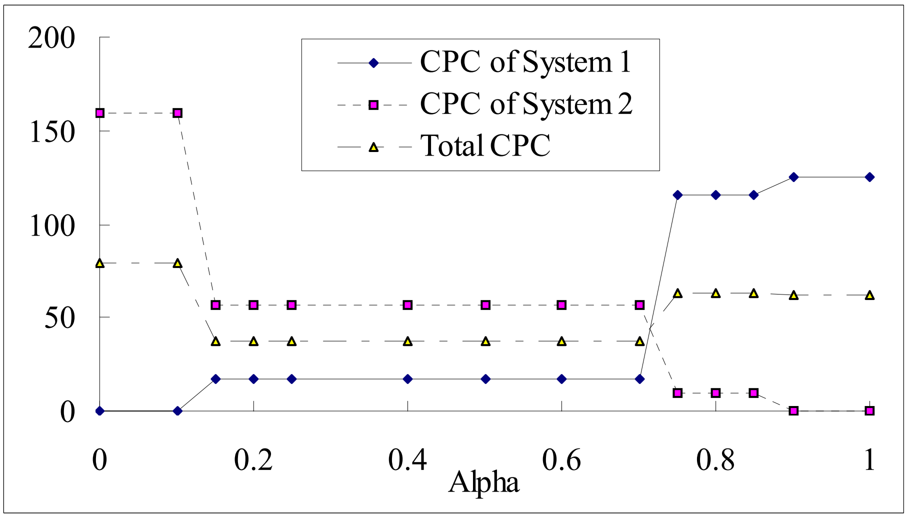

Figure 3. Comparison of CPC for Scheme 2

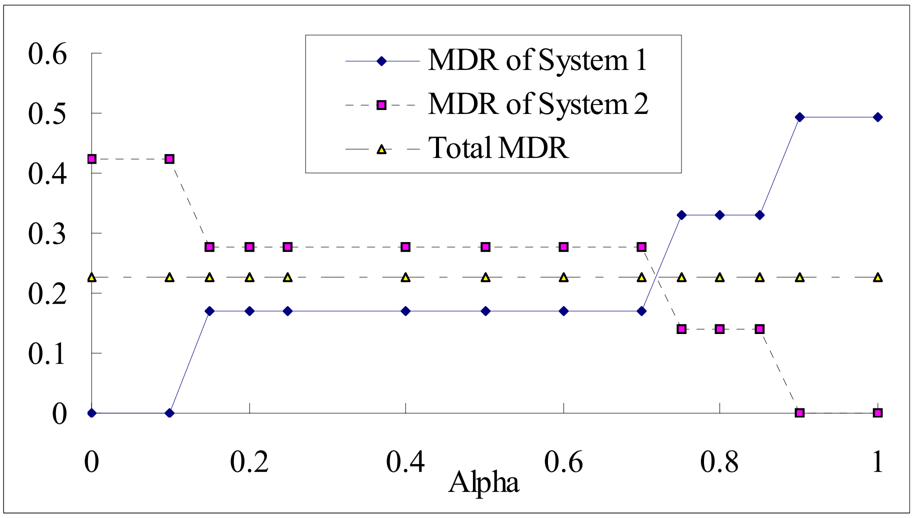

Figure 4. Comparison of MDR for Scheme 2

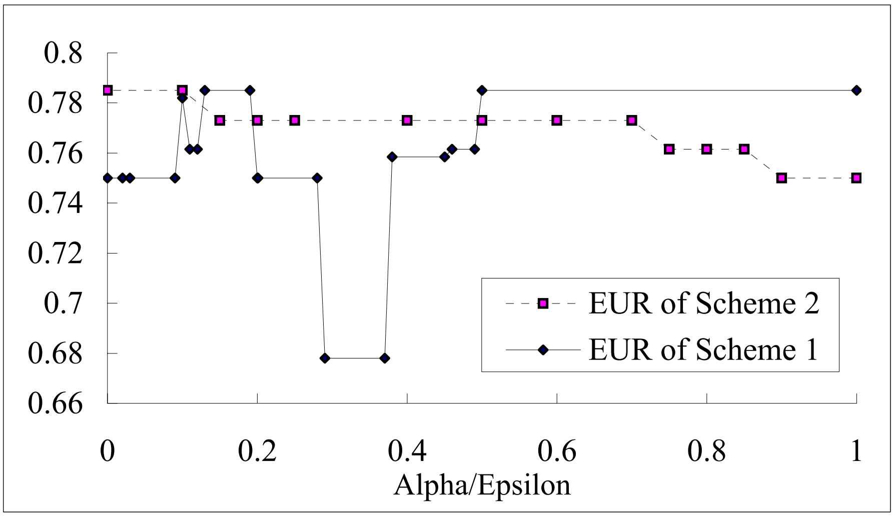

Figure 5. Comparison of EUR for two schemes

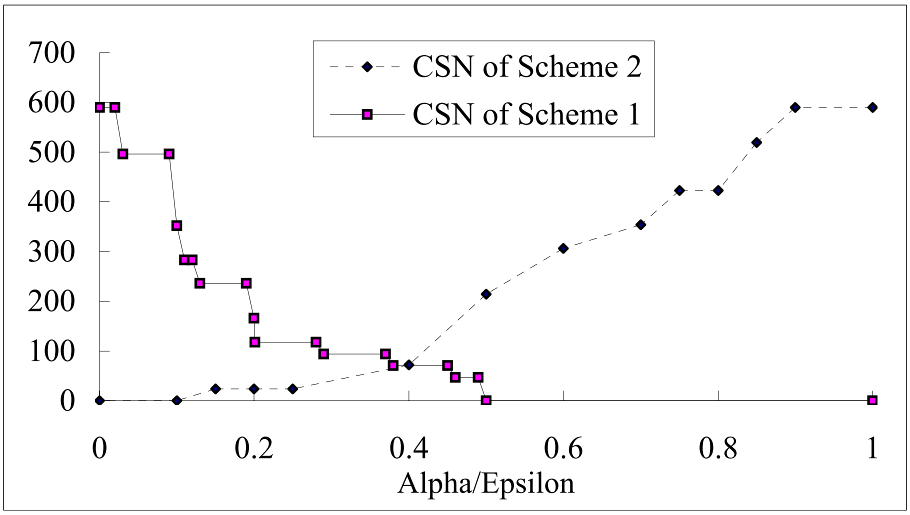

Figure 6. Comparison of CSN for two schemes

<u>For Scheme 1</u>

For $\varepsilon \geq 0.5$ and $\varepsilon \in [0.13, 0.19]$, all instances of Controller 1 meet their deadlines, and System 1 has no control cost, which is ideal case for System 1; the control cost of System 2 reaches to 159.0208 and the missed deadline rate is 0.4217; the efficient utilization rate reaches to 0.7850.

For $\varepsilon \leq 0.09$ and $\varepsilon \in [0.2, 0.28]$, all instances of Controller 2 meet their deadlines, and System 2 has no control cost, which is ideal case for System 2; the control cost of System 1 reaches to 124.8066 and the missed deadline rate is 0.4930; the efficient utilization rate reaches to 0.75.

For other value of ε, the control cost of System 1 is much larger than System 2; the missed deadline rate of Controller 1 is much higher than Controller 2.

The context switching number is decreasing along with ε increasing. The largest context switching number is 590 for small ε, and the context switching number for $\varepsilon \geq 0.5$ is zero.

<u>For Scheme 2</u>

Along with α increasing, the control cost and missed deadline rate of System 1 are monotone increasing but those of System 2 are monotone decreasing on the contrary; the efficient utilization rate is decreasing but the context switching number is increasing.

For full preemption, System 2 has no control cost and all instances of Controller 2 meet their deadlines; however, the control cost and missed deadline rate of System 1 reach to its largest.

For non-preemption, System 1 has no control cost and all instances of Controller 1 meet their deadlines; however, the control cost and missed deadline rate of System 2 reach to its largest.

The total missed deadline rate is always 0.2273.

As shown in above discussion, the context switching number of tasks can get greatly decreased by using the proposed method. Conditional limited preemption threshold controls the frequent preemption.

V. Conclusion

The scheduling problem of control tasks with uncertain timing constrains is discussed. The trapezoid fuzzy number is used to describe the uncertain deadline of a controller task. The concept of dedication is proposed. Along with waiting for execution, the dedication of a control task gets more and more large; and reflects its scheduling criticality very much. The larger the dedication is, the more exigent the task need be scheduled. So it is very appropriate to use the dedication as the metric of priority for a control task. And the task with higher dedication will be scheduled first, which is the basic principle of the proposed largest dedication first.

Acknowledgement

This work is supported by China NSF under Grant No. 60374058, 60373055 and 60373056.

References

[1] Q. Lin, P. C. Y. Chen, and P. A. Neow, “Dynamical scheduling of digital control systems,” in *Proceedings of IEEE Int’l Conference on Systems, Man and Cybernetics*. Washington, D. C., USA. Piscataway: IEEE Computer Society. 2003,Vol.5, pp.4098-4103.

[2] A. Cervin, “Improved scheduling of control tasks,” in *Proceedings of the 11th Euromicro Conference on Real-Time Systems*. York England, UK. Los Alamitos: IEEE Computer Society. 1999, pp.4-10.

[3] K. E. Arzen, A. Cervin, J. Eker, and L. Sha, “An introduction to control and scheduling co-design,” in *Proceedings of the 39th IEEE Conference on Decision and Control*. Sydney, NSW Australia. Piscataway: IEEE Computer Society. 2000,Vol.5, pp.4865-4870.

[4] M. Litoiu and R. Tadei, “Real-time task scheduling with fuzzy deadlines and processing times,” *Fuzzy Set and Systems*, 2001, Vol.117, Iss.1, pp.35-45.

[5] H. Jin, H. A. Wang, Q. Wang, Y. Fu, and H. Wang, “Fuzzy dynamic pre-emptive scheduling algorithms,” *Chinese Journal of Computers*, 2004, Vol.27, No.6, pp.812-818, (in Chinese with English Abstract)

[6] F. Terrier, L. Rioux, and Z. Chen, "Real time scheduling under uncertainty," In Nakanishi, S. (eds.): *Proceedings of the 4th IEEE Int'l Conference on Fuzzy Systems*. Yokohama Japan. Piscataway: IEEE Computer Society. 1995, Vol.3, pp.1177-1184.

[7] T. Zhou, X. Hu, and E. H. Sha, "A probabilistic performance metric for real-time system design," In: Jerraya, A., Lavagno, L., Vahid, F. (eds.): *Proceedings of the 7th Int'l Workshop on Hardware/Software Codesign*. Rome, Italy. New York: ACM Press. 1999, pp.90-94.

[8] H. Jin, H. A. Wang, Q. Wang, and G. Z. Dai, "An improved least-slack-first scheduling algorithm," *Chinese Journal of Software*, 2004, Vol.15, No.8, pp.1116-1123, (in Chinese with English Abstract)

[9] H. Jin, H. A. Wang, H. Wang, and D. L. Wang, "Scheduling design of controllers with fuzzy deadline," In: Wang, L., Jin, Y. (eds.): Fuzzy Systems and Knowledge Discovery. *Lecture Notes in Computer Science*. Springer-Verlag Berlin Heidelberg. 2005, Vol.3613, pp.861-864.

[10] Y. Wang and M. Saksena, "Scheduling fixed-priority tasks with preemption threshold," In: Gakkai, J. S. (eds.): *Proceedings of the 6th Int'l Conference on Real-Time Computing Systems and Applications*. Hong Kong, China. Los Alamitos: IEEE Computer Society. 1999, pp.328-335.

Hong Jin, Ph.D., is an associate professor in Institute of Software, Chinese Academy of Sciences. His current research interests include real-time scheduling, computer- controlled systems, real-time databases, wireless sensor networks, and fault-tolerant reliability.

Hong-an Wang, Ph.D., is a professor and doctoral supervisor in Institute of Software, Chinese Academy of Sciences. His current research interests include real-time intelligent system, real-time scheduling.

Hui Wang, Ph.D., is an associate professor in Institute of Software, Chinese Academy of Sciences. He performed post-doc study at Rutgers University from 1999 to 2002. His research interests include computational intelligence, cognitive engineering and human-computer interaction.

Dan-Li Wang, Ph.D., is an associate professor in Institute of Software, Chinese Academy of Sciences. Her main research interests include human-computer interaction, design method and usability engineering.