World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:4, No:8, 2010

Adaptive Fuzzy Control on EDF Scheduling

Xiangbin Zhu

International Science Index, Mathematical and Computational Sciences Vol:4, No:8, 2010 waset.org/Publication/2611

Abstract—EDF (Early Deadline First) algorithm is a very important scheduling algorithm for real- time systems . The EDF algorithm assigns priorities to each job according to their absolute deadlines and has good performance when the real-time system is not overloaded. When the real-time system is overloaded, many misdeadlines will be produced. But these misdeadlines are not uniformly distributed, which usually focus on some tasks. In this paper, we present an adaptive fuzzy control scheduling based on EDF algorithm. The improved algorithm can have a rectangular distribution of misdeadline ratios among all real-time tasks when the system is overloaded. To evaluate the effectiveness of the improved algorithm, we have done extensive simulation studies. The simulation results show that the new algorithm is superior to the old algorithm.

Keywords—Fuzzy control; real-time systems; EDF; misdeadline ratio.

I. Introduction

WITH the development of multimedia, embedded systems, mobile computing and ubiquitous computing, the real-time systems increasingly need to run on a dynamic and unpredictable environment. How to try to perform tasks' processing in prescriptive time and be able to control all real-time devices and tasks' coordinated running in dynamic and unpredictable environments bring new challenge to traditional real-time schedule theory.

In traditional real-time schedule theory, real-time scheduling policies are classified into two types, static scheduling and dynamic scheduling. The EDF algorithm is a venerable one of dynamic scheduling algorithms and an optimal algorithm by the criterion of schedulable utilization[1][2][3]. More and more applications, such as QoS, CAN and dynamic voltage scaling (DVS), employ the EDF algorithm to improve their performances. EDF scheduler has been exploited for scheduling not only in hard real-time systems, but also in soft real-time systems. For example, the open real-time system, which is an important one of the researches on real-time systems[4][5][6] ,employs EDF scheduler as the first scheduler.

But advances in these EDF applications in recent years have brought some problems. Traditionally, the EDF algorithm is employed in the hard real-time systems and there aren't any over-loading conditions in the real-time systems. But when the EDF algorithm is adopted in the soft real-time systems, the EDF scheduler has to work on over-loading conditions. For example, with the development of network and multimedia, many kinds of real-time appliances, including hard real-time systems, soft real-time systems and non-real-time systems, coexist simultaneously in a single system is more and more popular. Specially, the quickly development in hardware makes many embedded systems be able to play video and audio streams. So it is popular that a real-time system has hard applications and multimedia applications. For instance, a car control/monitoring system has a workstation, which collects data from many sensors with total sampling rate of 80Hz and show the dynamic graph. So the real-time system has a real-time task: communicating with sensors, and soft real-time processes: analyzer, GUI and play video or audio streams. In this environment, the real-time system may be to work on over-loading conditions.

Xiangbin Zhu is with College of Mathematics, Physics and Information Science, Zhejiang Normal University, China (E-mail: zhuxb@zjnu.cn).

Unfortunately, the EDF algorithm has some serious disadvantages when it is working on an over-loading system. The first problem is it is difficult to predict which tasks will miss their deadlines during overloads. Another problem is a late job which has already missed its deadline has a higher-priority than a job whose deadlines is still in the future. Consequently, if the execution of a late job is allowed to continue, it may cause some other jobs to be late[2].

So a good overrun management strategy is crucial to prevent this kind of instability. A common strategy is to schedule each late job at a lower priority than the jobs that are not late or to complete the late job, but schedule some functional noncritical jobs at the lowest priority until the system recovers from the overload condition. In a word, the scheduler either lowers the priorities of some late jobs, or discards some late jobs.

Actually, if a soft real-time system is overloaded, the distribution of misdeadline ratio is not rectangular in the EDF scheduling system. The usually scene is some tasks have many misdeadlines and some tasks have not any misdeadlines. Thus, many tasks may have noticeable degradation in quality of service when the real-time system is overloaded. So we should have a mechanism to distribute the misdeadlines uniformly among all tasks. In this paper, we use feedback control because feedback control is the very mechanism for a system to adapt to uncertainties and dynamics in its environment.

We employ the fuzzy control theory to dynamically adjust the priority of each task. The simulation results show the new scheduling method effectively reduces the variance of misdeadline ratio.

In this paper, a task is a sequence of invocations. So the entities being scheduler are invocations and tasks. Beginning from the start time of a task, invocations become ready at the start of fixed interval.

The rest of the paper is organized as follows: The scheduler and the task models are presented in Section 2. Section 3 presents the improved scheduling algorithm based on fuzzy control theory. Section 4 presents the simulation results and Section 5 summarizes our work.

II. Task Model

In this section, we define and explain important concepts and conventions used in this paper.

• Time slot: The unit interval between time t and time t+1 will be referred to as time slot t, $t \in N$.
• Interval of time: For integers a and b, [a, b]={a, ...,b-1}, and represents all time slots between and including a and b-1.
• Invocation: An invocation is an entity to be scheduled. The ready time r of an invocation is the earliest time the invocation can be scheduled; the deadline d of an invocation is the latest time by which the invocation must be completed.
• Task: a task is an infinite sequence of periodic invocations, defined by a tuple of 3 natural numbers:(c, d, p)

* c refers to the computation time.
* d refers to the deadline relative to the release of the task.
* p is the period and defines the length of the interval between the ready times of consecutive jobs.

• Task set: It is a set of p periodic tasks $S = \{\tau_0, \tau_1, ..., \tau_n\}$ Each task τ_j is characterized by 3 parameters which is defined above.
•CPU utilization U: it is the minimum required fraction of available service capacity of CPU. The utilization factor for n tasks is at least $U = \sum_{i=1}^{n} \frac{c_i}{p_i}$.
• Misdeadline: If a real-time task τ_i miss its deadlines, we say a misdeadline occurs.
• Misdeadline ratio: Misdeadline ratio is defined as the ratio of the number of misdeadlines in a task to the number of invocations in the task.

In this paper, the real-time system has the followed characters:
(1) Each task τ_i is periodic task.
(2) Tasks are preemptable and independable.
(3) The real-time system has only one processor.

III. THE FUZZY CONTROL SCHEDULING

A. Description of The Fuzzy Control Scheduling Algorithm

TABLE I
THE DISTRIBUTION OF MISDEADLINE RATIOS ISN'T UNIFORMLY

Tasks	invocations	misdeadlines	misdeadlines ratio
τ_1(4,5,5)	5	4	0.8
τ_2(2,3,5)	5	1	0.2

TABLE II
MISDEADLINE RATIOS ARE UNIFORMLY DISTRIBUTED

Tasks	invocations	misdeadlines	misdeadlines ratio
τ_1(4,5,5)	5	3	0.6
τ_2(2,3,5)	5	2	0.4

If a real-time system is overloaded, the distribution of misdeadlines ratio is usually not uniform with using EDF algorithm. The usually scene is several tasks have many misdeadlines and some tasks have not any misdeadlines. We expect the misdeadlines ratio of each task is almost uniform. For example, Table I shows a real-time task set. The task set has two tasks. One task's parameter is (4,5,5) and the other is (2,3,5). There are 5 misdeadlines till current time. We expect the distribution of misdeadlines likes Table II shows. So we should have a mechanism to uniform the distribution of misdeadline ratio. Obviously, we need cybernetic method because the new algorithm should dynamically adjust the priorities of tasks based on the distribution of misdeadline ratio.

In this paper, we use the fuzzy control theory to dynamically adjust the priority of each task because fuzzy theory, as a means of both capturing human expertise and dealing with uncertainty, have been applied to various control domains.

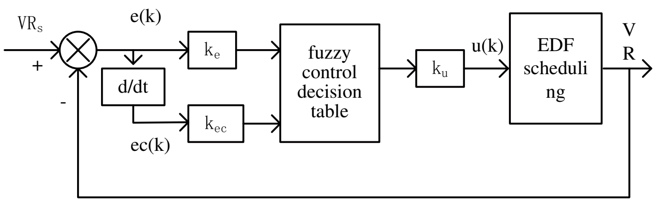

Fig. 1: Fuzzy control scheduling

We use the deadline to adjust the distribution of misdeadline because the priority of each task is based on deadline of each task in EDF scheduling. Obviously, misdeadline ratio is the controlled variable. We can use the number of task and CPU utilization to calculate the set point misdeadline ratio of each task. The fuzzy controller periodically monitors the controlled variable misdeadline ratio, calculate the error differenceΔVR from set point misdeadline ratio and the change of error, then through fuzzy inference based on fuzzy control rules and get the controlΔU. We use the controlΔU to modify the deadline.

Dynamic system loads result in the change of CPU utilization and the change of CPU utilization result in the change of misdeadline ratio. So in the actual application, real-time systems have dynamic system loads and dynamic CPU utilization. So we adopt the fuzzy controller with adaptive factor α. The main purpose of α is we needn't to modify the fuzzy control decision table when the CPU utilization has modified.

Figure 1 shows the fuzzy control scheduling of real-time systems[7]. Table III shows the fuzzy control decision table.

TABLE III
FUZZY CONTROL DECISION TABLE

ec \ e	-6	-5	-4	-3	-2	-1	0	2	2	3	4	5	6
-6	6	6	4	4	3	2	2	0	0	0	0	0	0
-5	5	4	3	2	2	2	1	1	0	0	0	0	0
-4	5	3	2	2	2	1	0	1	1	0	0	0	0
-3	4	2	2	1	1	1	0	1	1	1	0	0	0
-2	3	2	2	1	1	1	0	1	0	0	-1	0	0
-1	1	1	2	1	1	1	0	1	-1	-1	-1	-1	0
0	2	2	1	1	1	0	0	-1	-1	-1	-2	-2	-3
1	0	1	1	1	0	0	0	-1	-2	-2	-2	-2	-3
2	0	0	1	-1	0	0	0	-1	-2	-2	-2	-2	-3
3	0	0	0	0	-1	-1	0	-2	-2	-3	-3	-3	-3
4	0	0	0	0	1	-1	-1	-2	-2	-3	-3	-4	-4
5	0	0	0	0	0	-1	-1	-2	-3	-4	-4	-4	-5
6	0	0	0	0	0	0	-2	-3	-3	-4	-4	-5	-6

B. Sample Period

For every computer control system, it is an important problem to select a good sample period[8]. There are many factors to be thought. For example, the output signal must be existed for a while to the actuators. So the sample period should be long than the response time of the actuators. In general, the small sample period is more close to continues control and have more anti-jamming performance. But an over small sample period can produce great scheduling overhead, and may result in frequently adjusting the real-time scheduling system; on the contrary, at an over large sample period, the adjustment may be too slow to quickly adaptive adjust the real-time scheduling system. We get the following sample period strategy through many experiments.

For each taskτ_i, the sample period is, T=$\lambda y_i p_i$
, in which λ is a scalable sample rate factor. The scalable sample rate factor λ can be used to adjust the speed of adjustment based on the CPU utilization. With increases in the CPU utilization, λ is decreasing.

C. Fast Fuzzy Inference

In view of EDF scheduling system, we adopt general fuzzy sets, membership functions, and fuzzy rules, to get a widely applicable fuzzy decision. If the EDF scheduling system only servers for a specific application, then we can optimize the fuzzy components according to the actual characteristic of the application, or the real running data, and get an optimized fuzzy controller.

In the real-time systems, time is a key factor in control algorithm. So we utilizes the fast fuzzy inference architecture showed as figure 1. Using Min-Max compose inference algorithm, and maximum membership defuzzifacation method, we compute the basic fuzzy control decision offline and get a fuzzy decision table defined as table III[4]. In real scheduling, the process of scheduling control decision is as follows, monitor the errore$e(k)$, change of error *ec(k)*, and multiplied by k_e,k_{ec} to transform to the control table space, then query the fuzzy decision table and get the control, finally multiply k_u to transform to control space and get actual control $u(k)$. Where, quantity factor k_e, k_{ec}, proportion factor k_u is to implement the transformation of variable space. Suppose the space of variable x is X=[x_L, x_H], its fuzzy membership integer space is [m, n], then we can use formula (1) to calculate k_e,k_{ec}, and use formula (2) to calculate k_u. The adaptive α_1 and α_2 are modified with the increasing of CPU utilization.

$$k = \frac{n-m}{x_H - x_L}\alpha_1 \tag{1}$$

$$k_u = \frac{x_H - x_L}{n-m}\alpha_2 \tag{2}$$

D. Fuzzy Control Scheduling Algorithm

The fuzzy control-scheduling algorithm is working as follows:

```
Algorithm: Fuzzy Control Scheduling
{
Every sample period T, do Calculate e($k$) the error of misdeadline, and ec($k$) the change of e($k$), and multiply k$_e$, k$_{ec}$ to transform to fuzzy table space.
Query fuzzy decision table FDT, and multiply k$_u$ to transform to actual control space control $\Delta U(k)$.
Adjust the tasks window-constraint $W_i$ by:
deadline= deadline+ $\Delta U(k)$.
}
```

IV. Simulation Studies

In our simulation, we use two metrics to evaluate the performance of the two algorithms. One is the variance of misdeadline ratio and the other is the maximum number of misdeadlines in one task. We expect the misdeadline ratio of each task is almost equal. Thus, we use the variance of misdeadline ratio to study the result of the two algorithms. The maximum number of misdeadlines in one task also shows the uniform ability of the two algorithms to the distribution of misdeadlines in a certain extent.

We randomly produce task sets with the limit of CPU utilization U and the parameters: $p_i, c_i$and$d_i$are limited in some range. The parameters used in the simulation studies are given in Table IV. Each task set except for the last simulation contains approximately 12-20 tasks by fixing the schedule length to 1000. Each point in the performance curves (Figs.2-4) is the average of 5 simulation runs. In our simulations, the scheduling slot is 1.

TABLE IV
List of simulation parameter

Parameters	Explanation	Value in simulation
Max_C	Maximum execution time of invocations	6
Min_C	Minimum execution time of invocations	1
Slot	Scheduling granularity	1
Max_T	The max period of each task	8
Min_T	The min period of each task	3

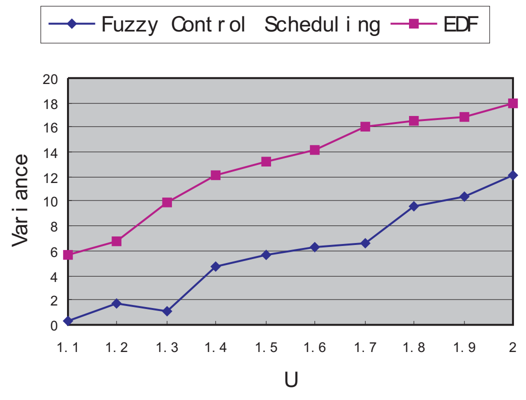

Fig. 2: Variance

In this simulation, the value of U is increased from 1.1 to 2.0. Fig.2 shows the effect on the variance of misdeadline ratio by the variation of CPU utilization. It is easy to see that the variances of the two algorithms are increased with the increasing of CPU utilization. So the variance is sensitive to CPU utilization. However, Fig.2 also shows that the fuzzy control scheduling EDF algorithm has better performance than the EDF algorithm. The variances of the new algorithm are lower than the variances of the old algorithm. Moreover, it can be seen that the variance of the EDF algorithm is more sensitive than the variance of the new algorithm to CPU utilization. This is because the fuzzy control scheduling can adapt itself to the dynamic CPU utilization.

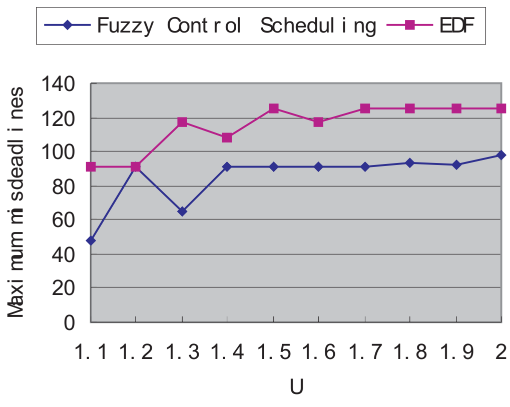

Fig. 3: Maximum misdeadlines

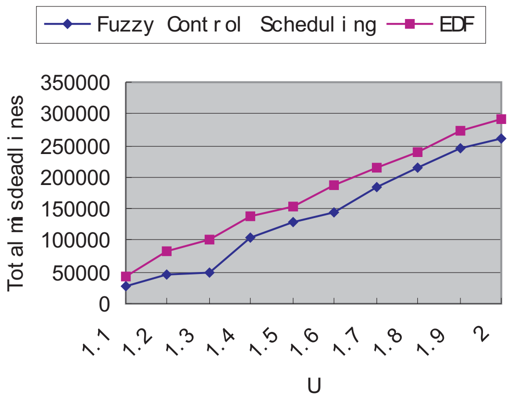

Fig. 4: Total misdeadlines

Fig.3 shows the effect on the maximum misdeadlines in one task by varying the CPU utilization. It should be clear in Fig.3 that the maximum misdeadlines of two algorithms are increased when CPU utilization is increased from 1.1 to 2.0. This is due to the misdeadlines of task sets are increased when CPU utilization is increased. After that, we observe that the maximum misdeadlines of the new algorithm are lower than the maximum misdeadlines of the old algorithm.

For investing whether the new scheduling algorithm affects schedulability, we measure the total misdeadlines of a task set in the two algorithms. Fig.4 shows the simulation results. We can see that the fuzzy control EDF scheduling algorithm has better performance than the EDF algorithm. The total misdeadlines of the new algorithm are lower than those of the old algorithm. So the schedulability of the EDF algorithm is better when using fuzzy control scheduling.

V. Conclusion

This paper proposes a new EDF scheduling algorithm, which uses fuzzy control theory to reduce the variance of misdeadline ratio. The simulation shows the new EDF algorithm has a good performance when CPU utilization is larger than 1. Moreover, when CPU utilization is changed, the performance of the new EDF algorithm is also good because it has some adaptive ability. The new EDF scheduling will improve the open real-time system scheduling performance

Acknowledgment

This research work is supported by Zhejiang Provincial Natural Science Foundation of China under Grant No.Y106285

References

[1] C L Liu and J W Layland, “Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment,” *Journal of the ACM* ,20(1), 1973: 40-61.
[2] Liu J W S, *Real-Time Systems*.Upper Saddle River: Prentice Hall, 2000
[3] Nimal Nissanke, *Real-time System*, Prentice Hall, 1997
[4] Deng Z, Liu JWS, Sun J. “A scheme for scheduling hard-real-time applications in open environment”. In: *Proceedings of the 9th Euromicro Workshop on Real-Time Systems*. Los Alamitos, CA: IEEE Computer Society Press, 1997: 155-185.
[5] Z. Deng and J. W. S. Liu, “Scheduling real-time applications in open envirovment,” In *IEEE Real- Time Systems Symposium*, San Francisco, December 1997.
[6] G. Lipari and G.C. Buttazzo,“Scheduling real-time multi-task applications in an open system”. In *Proceeding of the 11th Euromicro Workshop on Real-Time Systems*, York, UK, June 1999
[7] Huai Xiao-Yong, Zou Yong and Li Ming-Shu, “Adaptive Fuzzy Control Scheduling of Hybrid Real-time systems”, In *Proceedings of the First International Conference on Machine Learning and Cybernetics*, Beijing: IEEE, pp. 810-815, November 2002.
[8] Li Shiyong, *Fuzzy ControlNeuro Control and Intelligent Cybernetics*, Publication of Harbin Institute of Technology, 1998.9 Edition 2