FACTA UNIVERSITATIS
Series: **Mechanics, Automatic Control and Robotics** Vol. 6, Special Issue, 2007, pp. 9 - 24

THE INTERRUPT OF MASCOT ROBOT SYSTEM EMBEDDED IN RT MIDDLEWARE BASED ON FUZZY LOGIC

UDC 007.52+004.8(045)=111

Hai An Vu[1], Yoichi Yamazaki[1], Kazuya Fukuda[1], Yui Matsuura[1], Phuc Quang Le[1], Hitoho Oikawa[1], Kazuhiro Ohnishi[1], Yasufumi Takama[2], Fangyan Dong[1], Kaoru Hirota[1]

[1]Tokyo Institute of Technology, Yokohama, Japan
[2]Tokyo Metropolitan University, Tokyo, Japan
e-mail: Hirota@hrt.dis.titech.ac.jp; tou@hrt.dis.titech.ac.jp

Abstract. *A control method based on fuzzy priority measure and fuzzy selection criterion is proposed in order to solve the interrupt problem in Mascot Robot System. The fuzzy priority measure defines the priorities of interrupted instructions in order to optimize responsiveness of the system. The fuzzy selection criterion based on the fuzzy priority measure is used to determine the appropriate interrupted instruction for the next processing. This method is embedded in Robot Technology Middleware which is used to construct the communication environment among robots in Mascot Robot System. Five kinds of Eyeball robot's emotions, different in execution time, are used in the experiments on the real system. The fuzzy interrupt processing method improves the responsiveness of the system by more than 18.5% with an acceptable delay time of less than 0.32 seconds. Simulation experiments are carefully analyzed to classify systems and find out relationship between the two proposed fuzzy quantities and types of systems; this relationship is the key in optimizing different types of systems.*

Key words: *interrupt, robotics, fuzzy logic, middleware*

1. INTRODUCTION

Recently, robot systems for daily home environment are becoming common in the world, particularly in Japan. From 2005, the Japanese government has invested a lot of money in research on the New Energy and Industrial Technology Development Organization (NEDO). To implement open robot architecture, Robot Technology (RT) Middleware [5], a key software technology, has been developed and released by the Japan National Institute of Advanced Industrial Science and Technology (AIST) [4] from 2002. The RT-Middleware supports the construction of various networked robotic systems by the integration of diverse processing elements called RT-Components.

Received Ocober 30, 2008

In the NEDO project, many Japanese institutes and universities, such as NEC Inc., Tokyo Institute of Technology, Tokyo Metropolitan University and AIST, have cooperated to develop the Mascot Robot System (MRS) for 3 years [1], [2], [3]. In the home party's scenario, the system includes speech recognition modules, eyeball robots, a mobile robot, and a display module. These modules and robots are linked together under control of server by using RT-Middleware [4]. From the system's point of view, each of modules is presented as a RT-Component. The purpose of this scenario is to make the robot system responsible in real-time with friendly conversational situations.

One of the intrinsic properties of the Mascot Robot System is the existence of uninterruptible processes. Conventionally, these interrupted processes are not handled by system. For that reason, the system has been losing real-time responsiveness. In addition, it is required that every process's time consumption in the system must be known beforehand. To resolve this, fuzzy logic is used for selecting some interrupted processes for next actions of the system. To select the proper process, two fuzzy-quantities, *fuzzy-priority measure* and *fuzzy selection criterion* are used. The proposed measure is used to estimate priority of interrupted processes depending on their occurring-times. From that measure, the fuzzy selection criterion is calculated in order to determine the next appropriate process. The activators of the Mascot Robot System are human speeches. Therefore, they can cause interrupted processes while an uninterruptible process, such as eyeball robot's emotional expressions, is being executed. The proposed algorithm is implemented in the controller of eyeball robots to improve the performance. Experimental data includes simple Japanese words and their recognition results in eyeball robots.

The rest of the paper is organized as follows: Chapter 2 presents an overview of the whole Mascot Robot System and its components. Chapter 3 begins with a brief introduction of RT-Middleware and then explains the constructed network to connect the components in Mascot Robot System by using RT-Middleware. Chapter 4 presents the interrupt problem in the Mascot Robot System, related works and proposes the algorithm to handle interrupted processes. In chapter 5, implementation and experiments are presented. Finally, conclusion and future work are discussed in Chapter 6.

2. MASCOT ROBOT SYSTEM

The Mascot Robot System is an information retrieval and recommendation system for household environments which is able to conduct casual communication between humans and robots through human speech. This system is composed of four fixed robots (a TV, a dart board, a Display, and a mini_bar) and one mobile self-propelled robot, an information retrieval engine, and a server for overall management. The block diagram is showed as Fig.1.Every robot is composed of a controlled PC, a microphone, a Speech Recognition Module (SRM) and an Eye robot [1], [2]. In the Mascot Robots System, one directional and four non-directional microphones are setup, the first one for the mobile robot and one each for the robots. The SRM processes speech inputs and then indicates the recognition result on the PC display or write down the information in a log file through the RT Middleware network system. Depending on the recognition results, the eye robots express their corresponding emotions through designated actions.

The Speech Recognition Module in Fig.2 is compact and has high performance, e.g. it has a small size, low cost, low power consumption. The principle of speech recognition in the SRM is based on searching the word which matches the input in a pre-defined dictionary. This SRM is developed by NEC, Inc.

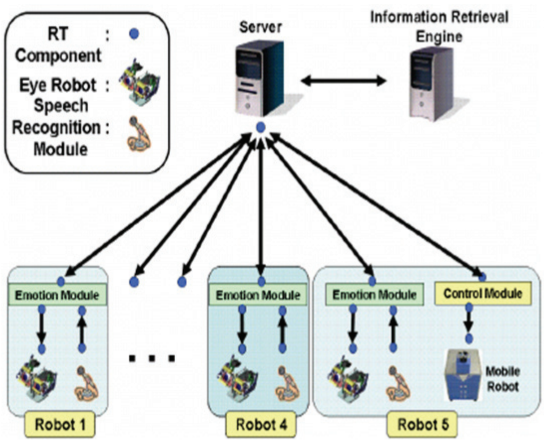

Fig. 1. Mascot Robot System

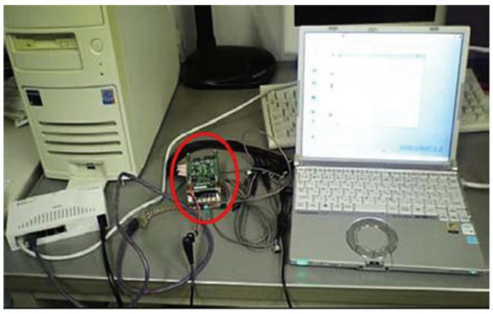

Fig. 2. Speech Recognition Module

As for the eye robot, there is a pair of eyeballs and eyelids. The eyeball movement is based on fuzzy control (up and down left and right) and serves to express different emotions such as happiness, sadness, surprise, anger. The movements are based on the mechanism of the human eye, and the eyeball has 3 degrees of freedom while the eyelid has 2 degrees, so the eyeball robot has a total 5 degrees of freedom. The Size of eyeball robot is 130 mm in width, 80 mm in height and 75 mm in depth. It uses 5 micro-servomotors as actuators and is controlled by a H8 micro-processor. Eyeball robot is shown in Fig.3

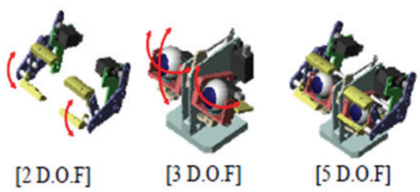

Fig. 3. Structure of Eye Robots

The operation of the eye robot is determined based on information from human speech which is acquired from the SRM.

The overall management server controls and supervises the whole system, making it possible to coordinate the action of the whole group of robots. The network system connecting the robots, the information retrieval engine, and the server is implemented using RT middleware.

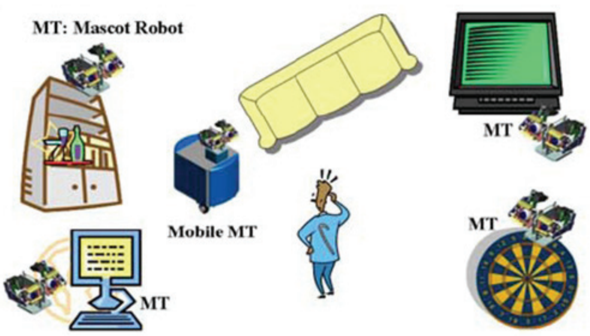

Fig. 4. The Mascot Robot System in a home environment

Making use of the above components, this system is set up in an interior space which simulates a home environment. The fixed robots are setup on furniture and appliances such as the TV, a PC or a closet. Their arrangement is shown in Fig.4.

The connection between the two servers and four fixed robots is done by wire LAN, but connected to the mobile robot with wireless LAN.

3. RT-MIDDLEWARE AND MRS

3.1 RT-Middleware

The RT-Middleware aims at establishing a common platform based on the distributed object technology which supports the construction of various networked robotic systems by the integration of various network enabled robotic elements. RT-Middleware was developed by National Institute of Advanced Industrial Science and Technology (AIST), Japan. In the RT-Middleware, functional element of the robot is segmented into the unit that is called RT-Component.

RT-middleware can work on Windows, Linux and other platforms. With one RT-component, i.e. it can be used in more than one system at the same time. With RT-middleware, the construction of multiple robot systems becomes very simple. RT-components are created first. From these components, it is possible to create many different systems depending on the communication objectives these between robots.

3.2 RT-Middleware for MRS

Fig. 5 shows the configuration of RT-Middleware for the Mascot robot system.

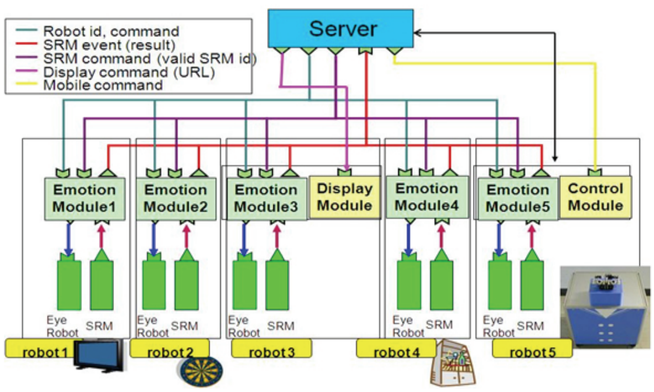

Fig. 5. RT-network of Mascot Robot System

There are six types of components created in the Mascot Robot System as follows:

a) SRM Component to recognize speech.
b) Emotion Component which receives the results from the SRM Components and then sends them to Server. This component also receives the response from Server and sends the emotion type to EyeRobot Component.
c) EyeRobot Component which controls the motion of Eyeball robot.
d) Controller Component which controls the motion of mobile robot.
e) Display Component which displays information upon retrieval by user.
f) Server Component which supervises the whole system.

The TimedString data which flows in the system is defined in RT-middleware. The sending and receiving of data in RT-network can be done either manually or automatically.

4. THE INTERRUPT PROBLEM IN THE MRS

4.1 Review of the Interrupt Problem

In computing, an "interrupt" is an asynchronous signal from hardware indicating the need for attention or a synchronous event in software indicating the need for a change in execution. A *hardware interrupt* causes the processor to save its state of execution via a context switch, and begin execution of an interrupt handler. *Software interrupts* are usually implemented as instructions in the instruction set, which cause a context switch to an interrupt handler similar to a hardware interrupt. Interrupts are a commonly used technique for computer multitasking, especially in real-time computing.

There are many ways to deal with the interrupt problem in conventional real-time systems such as computers. In such systems, instructions usually have two characteristics, first, they can be halted immediately when interrupts occur and second, their accomplished times are only able to calculate relatively. Priority assignments are key technique that is often used in both hardware and software to resolve the issue. In interrupt handlers, priority definitions depend on system producers or administrator.

Most of interrupt handlers operates are shown in Fig.6

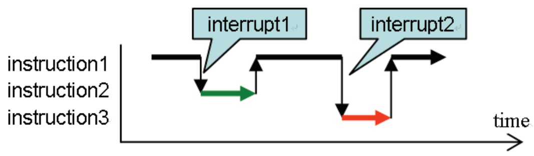

Fig. 6. Conventional Interrupt Process

When one instruction is being processed, an interrupt instruction occurs, system checks priority of interrupt instruction and compares with priority of processing instruction to select the next process

4.2 The Interrupt Problem on MRS

As mentioned above, Mascot Robot System is a real-time system, so interruption must to be respected and solved.

In Mascot Robot System, the input of system is the result of speech recognition from five sound recognition modules. Therefore, in one moment or short time, it might have many words put in from multi-users. Eyeball robot action is one of processes that can not be halted while being processing. In such case, it is necessary to wait for processing to get finished before adding the next instruction.

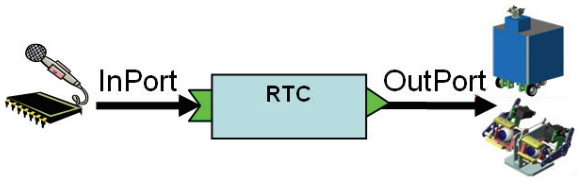

Fig. 7. Data flow in RT-Network

Suppose one instruction is being processes, the time needed to complete it is T_i, and during that time, there are n instructions coming in as shown in Fig.8.

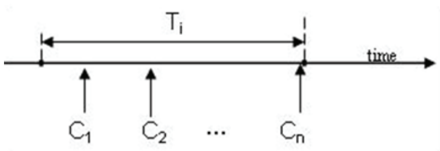

Fig. 8. Interrupt status

If the system cuts off all these interrupted instructions, it may be have no input when T_i elapses. This leads to low system performance. Furthermore, if all n interrupted instructions are executed in order, the system is delayed and the user can not understands when the responses shall come. This research gives the answer to the following questions:

Which one will be executed in the next process?

One or more instructions will be selected in the next process?

4.3 Definition of Interrupt Priority

In execution of eye robot, five kinds of emotion have the same role. When an input comes in, eye robot will execute suitable emotion immediately as soon as possible. When an interrupt occurs, the system only cares about the time of appearance of the interrupted instruction. Therefore, in Mascot Robot System, an attempt is made to try to define priority of instructions depending on the appearing time of them.

4.3.1 Temporal Priority Measure

Now, a definition of priority based on time parameters is proposed. Two things that are considered are the interrupt request time and the time when the interrupt process starts. If all of n interrupt requests appearing in time T_i are checked and if T_i is quite long, then the time between input and output is long. It can be said that the system's response is slow or system performance is not good. In this definition, only interrupt requests which are near the end of T_i are of interest as shown in Fig.9.

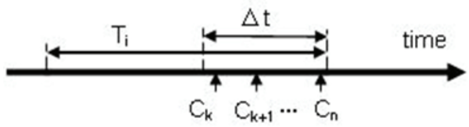

Fig. 9. Interrupt State

Because only the next interrupt requests which appear in a short time Δt are sought, the focus on the time axis is on the region [Ti – Δt, Ti] which can be presented as [0, Δt]. Normally, the priority is lowest at origin (0) and highest at the end of Ti (Δt). Temporal priority is proportional to time as illustrated in Fig.10:

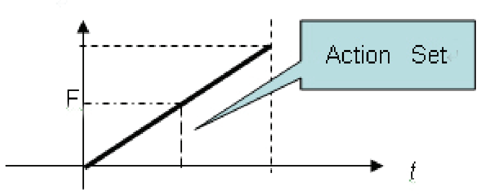

Fig. 10. Temporal Priority

F(C_t) is the priority function. H is the highest priority. If an interrupt request appears at time t then the priority of processing this interrupt is calculated as

$$F(C_t) = \frac{H \times t}{\Delta t} \ . \tag{1}$$

However, for interrupt requests which are very close to the end of T_i, their deviation can be neglected such that they end up having the same priority. Therefore, the fuzzy priority measure is considered.

4.3.2 Fuzzy Priority Measure

Basically, fuzzy priority measure is the same as temporal priority measure. At origin of time axis, the priority is equal to zero. And in the very short time dt closed to the end point of T_i, all the priorities of interrupts are the same and equal to 1. That means, preference is shown for instructions, which are near the finish point of execution. The most important problem in this method is determination of values of Δt and dt. There is no same answer for every system. But the value interval of them can be determined through experiments in chapter 5. The Fuzzy priority function is shown by Fig.11.

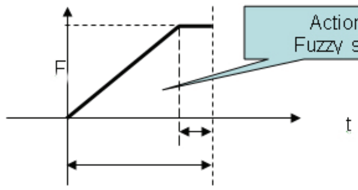

Fig. 11. Action Fuzzy Set

From the action fuzzy set, there are some obvious questions that need to be considered.

Which interrupt instructions will be chosen for the next process?

How many interrupt instruction will be chosen?

If in the time dt more than two interrupt instructions come in, does that mean that the different interrupt instruction have the same priority, then which one should be chosen?

These problems will be addressed in the next section based on two fuzzy quantities.

4.4 Fuzzy Interrupt System

Because the Mascot Robot System is an information retrieval and recommendation system, the response time of the system to the user's request is critical. As soon as the user's speech is recognized, the system must produce a corresponding response in as short a time as possible. Therefore, the proposed method is designed such that only one interrupt instruction is executed in the next process.

Thus, a method is proposed to determine which one of the incoming interrupts will be executed in the next instruction. Suppose that, there are two interrupt instructions with equal or only slightly different priorities. The higher priority interrupt instruction's process time is relatively longer than the lower priority one. In this case, if the higher priority interrupt instruction is chosen, the probability that other interrupt instruction will come in is high. This causes the system's responsiveness to deteriorate and lead to late reactions to the user's requests. So after the fuzzy priority measure is calculated, another measure is defined called the *fuzzy selection criterion* (FSC). T_t is the execution time of instruction C_t, and the FSC is calculated using

$$FSC(C_t) = \frac{F(C_t)}{T_t} \quad . \tag{2}$$

In the action fuzzy set, the FSC is calculated for all interrupt instructions, and the argument of maximum over them is chosen as the next process instruction.

The procedure for processing all incoming instructions in the fuzzy interrupt processing system includes the following six steps:

Step 1: Set the origin point of a processing instruction.

Step 2: Calculate the Fuzzy Priority Measure of interrupted instruction (FPM).

Step 3: Calculate the Fuzzy Selecting Criterion (FSC).

Step 4: Calculate the maximum of all the FSC, determine next instruction.

Step 5: if another interrupt instruction comes in, go back to Step2.

Step 6: When the current process is finished, put out the next instruction and set a new origin point.

The incoming instructions are processed in the proposed system as in Fig.12. The Fuzzy Interrupt Processing block role deals with interrupt instructions. In this block, the proposed FPM and FSC are calculated for interrupt instructions in order to find out the suitable instruction for the next process of our system. The Fuzzy Interrupt Processing module is the platform on which the proposed fuzzy quantities are implemented as shown in Fig.12. Details of the steps of Fuzzy Interrupt Processing module are shown in Fig.13.

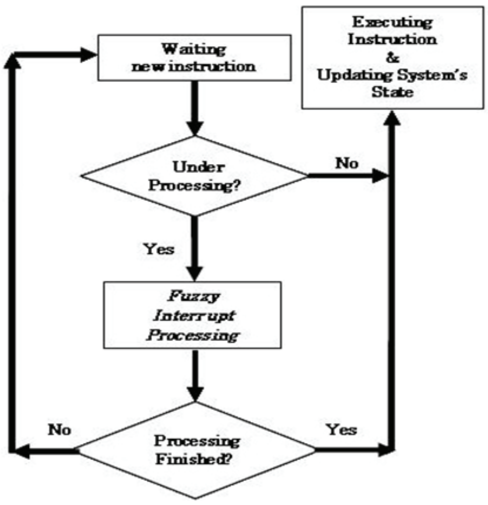

Fig. 12. Interrupt Processing Flow Chart

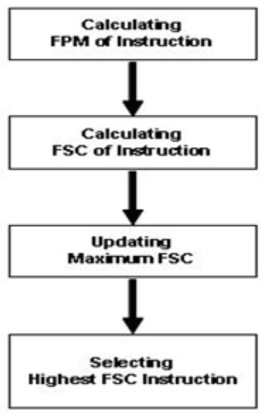

Fig. 13. Fuzzy Interrupt Processing details

5. SIMULATION PROGRAM AND EXPERIMENT ON MRS

In order to assess the advantages of the proposed method, the proposed interrupt handlers are embedded in RT components of the eye robots in the Mascot Robot System and tested.

In testing the system of interrupt processing, there are two input quantities, instructions' appearing points and instructions' time consumption. For each interrupt instructions, the system's interrupt handlers calculate the instruction's fuzzy priority and fuzzy selection criterion based on the input quantities. The fuzzy selection criterion of this instruction is used for comparison with other interrupt instructions. Finally, the next instruction of the system is determined based on the proposed algorithm in chapter 4.

The experiments on the Mascot Robot System are small-scale and require a lot of time. Therefore for future use of the proposed method on large-scale robot system, two kinds of evaluation experiments are necessary. The first one is on the real system and the second one is based on a simulation program. The main difference between the former and the later experiments is in the incoming instructions of the system. In the simulation program, the instructions' appearance points and time consumption are generated randomly. On the other hand, on the real system, instructions' parameters are determined by the robots.

5.1 Simulation Program for Fuzzy Interrupt

For a profound analysis of the proposed fuzzy interrupt processing algorithm's effect on the performance of the system, three types of system are examined depending on the interrupt instructions' execution time and the occurrence density of interrupt instructions.

Table 1. System Classification

System	Density	Execution time
Type 1	Low	
Type 2	medium	medium
Type 3	High	Short

In the first type, the occurrence of interrupt instructions density is low. The difference between the performance of the non-interrupt processing system and fuzzy interrupt processing system is not significant

In the second type, the occurrence of interrupt instructions density is medium, and interrupt instructions' execution time is long.

In the third type, the density of interrupt instructions is high, and interrupt instructions' execution time is short.

In the simulation experiments, 10.000 input instructions are generated randomly for testing the system. Two main input parameters are tuned; the density of instructions and the average of instructions' execution time to change between the three types of systems. For fuzzy interrupt processing, an attempt is made to establish the relationship between the effect on the performance and fuzzy parameters, Δ t and dt.

5.1.1 Experiments with Type-1-Systems

In Table 2, ADT is set as the average of the time difference between two successive instructions. ADT is inversely proportional to the instruction density. As shown in Table 2 and Fig. 14.

Table 2. Performance of Non-Interrupt and Fuzzy Interrupt method with Type-1-Sytems

No Interrupt (No. of inst.)	Fuzzy Interrupt (No. of inst.)	Delay Time (s)	ADT (s)	Execution Time (s)
8602	9432	0.04	5.09	3.61
8634	9446	0.04	5.11	3.60
9559	9932	0.01	6.12	3.60
9555	9909	0.01	6.10	3.60
9571	9899	0.01	6.10	3.61
9919	9995	0	7.11	3.60
9936	9996	0	7.11	3.59
9916	9993	0	7.09	3.60
9931	9990	0	7.11	3.60

With a system having a low incoming instruction's density, the number of executed instructions between non-interrupt processing and fuzzy interrupt processing is the slightly different. The longer the ADT is, the closer executed instructions between two systems are. In this case, the impact of Δt and dt is not significant, because case of interruption are few. In addition, when the interval between two successive instructions is increased, the proposed method still outperforms non-interrupt method with a very small delay time. This means that substituting the proposed method in the real application is acceptable.

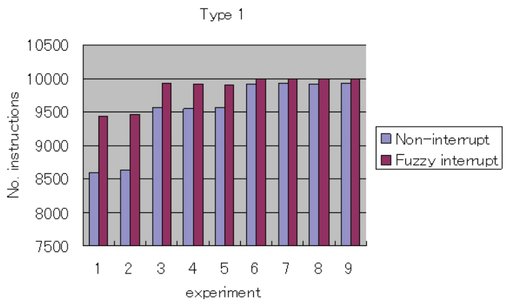

Fig. 14. Executed instructions in a Low Density System

5.1.2 Experiments with Type-2-Systems

In the experiment with type-2-systems, instructions' density is set to medium (around 1.4 to 1.7 seconds) and execution time is also set to medium (around 5 to 12 seconds). The main difference between type-1-systems and type-2-ones is the low responsiveness of the non-interrupt method. The underlying reason is the characteristic of type-2-system as mentioned in the system's classification.

A. Higher Performance of the Fuzzy Interrupt Process

Because the instructions' execution time is quite long, small values are chosen for Δt and dt in order to emphasis the responsiveness of the system. Table 3 and Fig. 15 show the experimental results with Δt = 0.1, dt = 0.02.

Table 3. Performance of Fuzzy method with Type-2-Systems

No Interrupt (No. of inst.)	Fuzzy Interrupt (No. of inst.)	Delay Time (s)	ADT (s)	Execution Time(s)
1253	1344	0.34	1.59	11.61
1270	1361	0.35	1.61	11.59
1665	1785	0.21	1.60	8.60
1662	1791	0.20	1.61	8.61
2106	2259	0.13	1.60	6.60
2103	2249	0.12	1.60	6.59
2399	2586	0.09	1.60	5.60
2399	2579	0.08	1.59	5.61

With a fixed ADT, when the execution time of instructions is reduced, the number of executed instructions is increased, and delay time is decreased. That means that in the Type 2 system, the proposed fuzzy method improves responsiveness.

B. Relationship between Δ t and System Performance

In order to examine the relation between fuzzy priority parameter Δt and system performance, the ADT and dt are set to 1.6 seconds and 0.02 seconds respectively. The system performance improvement is shown in 16, Fig.17 and Fig.18. In Table 4, the results show that system's delay time is proportional to Δt.

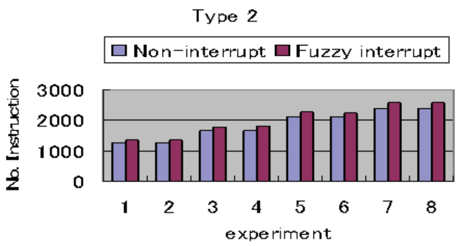

Fig. 15. Performance with changing execution time

Table 4. Relationship between Δt and system performance

Δt	Improvement (%)	Delay Time(s)	Execution Time(s)
0.3	**9.4**	1.03	11.54
0.2	**8.7**	0.84	11.54
0.1	**6.7**	0.35	11.54
0.3	**11.3**	0.99	9.6
0.2	**10.1**	0.71	9.6
0.1	**7.3**	0.25	9.6
0.3	**15**	0.75	6.6
0.2	**13.1**	0.42	6.6
0.1	**7.6**	0.12	6.6

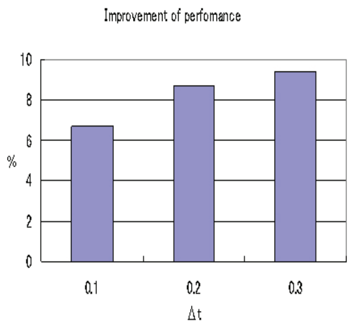

Fig.16. Sys. Performance (exe. time 11.54s)

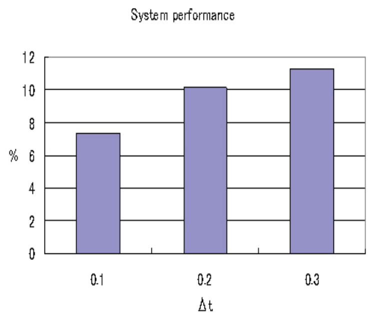

Fig.17. Sys. Performance (exe time 9.6s)

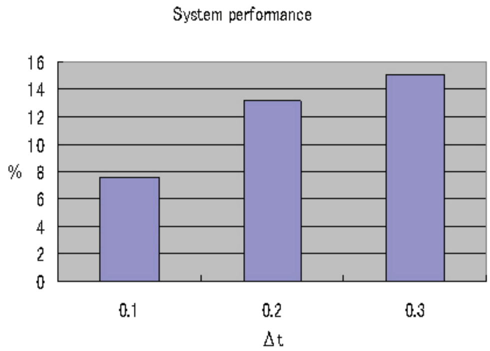

Fig. 18. Sys. Performance with execution time 6.6s

In order to increase the responsiveness of the system (low delay time), the improvement of the proposed method is reduced. The balance can be kept by adjusting between the two extreme cases. With the same value of Δt, it is observed that the improvement of the proposed method increases depending on execution time with inverse proportion. That happens when there is a change from type-1-systems to type-2-systems.

5.1.3 Experiments with Type-3-Systems

In experiments with type-3-systems, because of their features, they have a lot of similarities with the Mascot Robot System. In order to understand more carefully the effect of proposed method in dealing with interruptions in the Mascot Robot System, simulation of type-3-systems is analyzed in depth. In these experiments, Δt is set to 0.4 seconds, ADT to 0.4 seconds, and the controlled average execution time to 3.2 seconds along with instructions' execution time ranging from 2.0 seconds to 4.2 seconds. These parameters are characteristic of type-3-systems. In the Mascot Robot System, the emotions of eye robots usually consume around the same amount of time as well. An attempt is made to find out the best value of parameter dt to improve the performance of system. Parameter dt is tuned from 0.1 to 0.3, and for every value of dt, 10 experiments are performed and the average of instructions, system performance, and delay time are calculated. From the results are shown in Table 5, Fig.19, and Fig.20, it is easy to see that, when the value of dt is increased, system performance or number of executed instructions goes up as well. It also leads to longer delay time.

Table 5. Relation between dt and system performance

dt	No Interrupt	Fuzzy Interrupt	Improve (%)	Delay Time
0.1	1185	1408	18.8	0.31
0.2	1185	1458	23	0.36
0.3	1185	1495	26.1	0.43

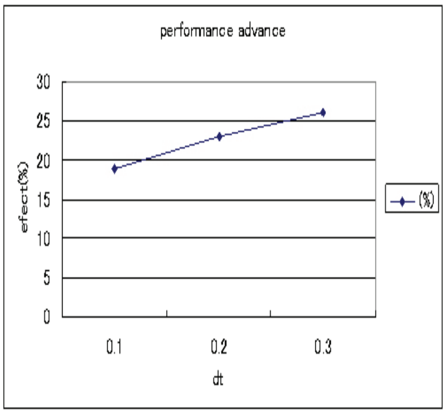

Fig. 19. System performance

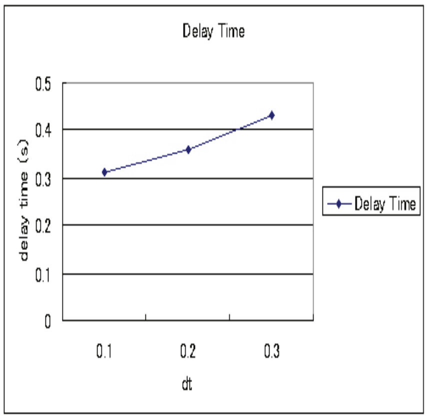

Fig. 20. Delay time

5.2 Experiments on the MRS

In this section, the results of experiments of the proposed method performed on eye robots in the Mascot Robot System are presented. To calculate exactly the time scale and to be able to control directly the emotion of the eye robot, the proposed method is embedded in the eye robot components of RT-middleware network.

The system is constructed by RT-middleware version 0.2.0, and running on Vine Linux 3.2 version. Five kinds of eye robot's emotions are used in experiments. It takes 2 second for the shortest emotion and 4.2 second for the longest emotion. To get high interrupted status come, the input of experiment is done by commands from the keyboard. five experiments are performed and the system's performance and delay time are calculated. In each experiment, the average of input instruction is 170. The result shows as Table 6.

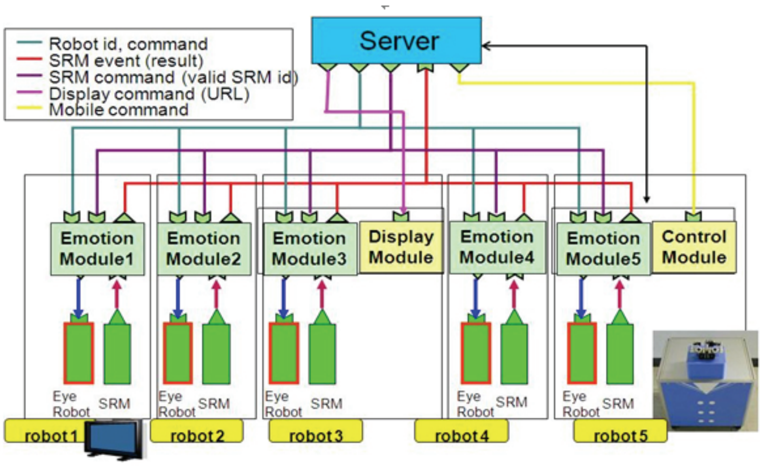

Fig. 21. Fuzzy processing embedded in RTM

Table 6. MRS Performance

ADT (s)	Improve (%)	Delay Time (s)	Execution Time (s)
0.55	35	0.28	3
1	18.5	0.32	3.1

In experiment of Mascot Robot System, since the execution time of the eye robot's emotions is short, the intervals are fixed such that the calculated interrupted instructions' priorities are 1 second and highest priority interval is 0.5 second. The result shows that, for Type 3 as well, the responsiveness of Mascot Robot System is improved, and the delay time of the system is about 10% of execution time on average.

6. CONCLUSIONS

The interruption control method for Mascot Robot System based on Fuzzy Priority Measure (FPM) and Fuzzy-Selecting Criterion (FSC) is proposed. Fuzzy Priority Measure is used to express preference of interrupted instructions which occur around the end point of current processing instruction. After calculating FPM, FSC is produced from FPM based on interrupted instruction's execution time. Finally, FSC is used as selecting criterion to determine the next instruction for system. In addition, the simulation experiments are carefully analyzed to classify systems and find out relationship between the two proposed fuzzy quantities and types of systems. This relationship is the key feature for applying the method in optimizing different types of systems.

In RT-Middleware network of Mascot Robot System, there are many places such as Emotion components, Server component, and Eyeball robot components where the proposed method can be implemented. In current testing system, the proposed method is implemented on the Eye robot components. There are some reasons for using two kinds of experiments, on real Mascot Robot System and on simulation. Experiments on the Mascot Robot System are difficult because of time and system environment. To find out the relationship between proposed quantities and several types of real-time system, the simulation experiments were examined. With our proposed method, the number of executed instructions is always higher than non-interrupt processing system. The tradeoff is the delay time.

Results of experiments on Mascot Robot System show that the responsiveness of system improved by more than 18.5%, with acceptable delay time 9% of 3.1 second in average. As analysis result, to classify systems in simulation testing, it is understood that Mascot Robot System is one of Type-3-systems, high density and short execution time of instructions. With simulation testing, the systems are classified into three types based on density execution time of instructions. The relationship between the responsiveness of the system and the proposed fuzzy quantities is different among types of system. For type-1-systems, the result of our proposed method is slightly better than that of non-interrupt processing method. For type-2-systems, the responsiveness of system mainly depends on the value of Δt, examining time interval. For type-3-systems, responsiveness of system involves both of Δt and dt, delay time of the system in average, however, largely depends on dt.

Future perspectives include automatic determination two fuzzy quantities based on acceptable delay time and instructions' execution times; specifying the suitable component in Mascot Robot System for implementing our proposed method; surveying the performance of type-2-systems when they can accept more than one interrupted instructions for next process.

REFERENCES

1. Yamazaki, Y., Hatakeyama, Y., Dong, F., Nomoto, K. and Kaoru, H. , "Fuzzy Inference based Mentality Expression for Eye Robot in Affinity Pleasure-Arousal Space", *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.12 No.3, pp.304-313,2008
2. Yamazaki, Y., Dong, F., Masuda, Y., Uehara, Y., Kormushev, P., Vu, H.A., Le, P.Q. and Hirota, K., "Intent Expression Using Eye Robot for Mascot Robot System", *International Symposium on Advanced Intelligent Systems*, pp. 576-580, 2007
3. Uehara, Y., Yamazaki, Y., Masuda, Y., Vu, H.A., Fukuda, K., Matsuura, Y., Le, P.Q., Hannachi, M.S., Dong, F., Takama, Y. and Hirota, K., "Speaker Emotion Inference Module and its Application to Mascot Robot System", *International Symposium on Management Engineering*, pp. 299-306, 2008.
4. Robot Technology Middleware (RTM) http://www.is.aist.go.jp/rt/OpenRTM-aist/html.

PREKID MASKOT ROBOTNIH SISTEMA UGRADJENIH U RT-MIDDLEWARE ZASNOVAN NA FAZI LOGICI

Hai An Vu, Yoichi Yamazaki, Kazuya Fukuda, Yui Matsuura, Phuc Quang Le, Hitoho Oikawa, Kazuhiro Ohnishi, Yasufumi Takama, Fangyan Dongm, Kaoru Hirota

Metod upravljanja zasnovan na merama fazi prioriteta i kriterijumima fazi odabira predlažu se u radu da bi se rešio problem prekida Maskot Robotnih Sistema. Mere fazi prioriteta odredjuju prioritete prekinutih uputstava da bi se optimalizovale reakcije sistema. Kriterijum fazi odabira zasnovan na meri fazi prioriteta se koristi da se odredi odgovarajući prekid uputstava za sledeće procesiranje. Ovaj metod se koristi u Robot Tehnologiji Middleware koja se koristi da bi se uspostavila atmosfera komunikacije medju robotima u Maskot Robot Sistemu. Pet vrsta Očnih robotskih emocija, različiti po vremenu izvršenja, korišćeni su u eksperimentima na realnim sistemima. Metod procesiranja fazi prekida poboljšava reakciju sistema za više od 18.5% sa prihvatljivim odstupanjem u vremenu sa manje od 0.32 sekundi. Eksperimenti simulacije su pažljivo analizirani da bi se klasifikovali sistemi i da bi se pronašao odnos izmedju predloženih fazi kvantiteta i tipova sistema; ovaj odnos je ključ za optimalizaciju različitih tipova sistema.

Ključne reči: *Procesiranje prekida, prekid, fazi logika, middleware, robotika*