*Advanced Robotics*, Vol. 11, No. 5, pp. 501–517 (1997)


*Paper*

# Distributed and cooperative object pushing by multiple mobile robots based on communication

KOICHI OZAKI, HAJIME ASAMA[1] and ISAO ENDO[1]

*Department of Mechanical Systems Engineering, Faculty of Engineering, Utsunomiya University, Ishii 2753, Utsunomiya, Tochigi 321, Japan*

[1]*The Institute of Physical and Chemical Research (RIKEN), Hirosawa 2-1, Wako, Saitama 351-01, Japan*

Received 21 December 1996; accepted 4 June 1997

**Abstract**—This paper describes the implementation of distributed and cooperative object pushing by multiple mobile robots based on communication. In order to realize a multi-functional, flexible and robust robotic system, we adopted a concept of functional distribution and cooperation. We have been developing a distributed autonomous robotic system, ACTRESS, which is composed of multiple autonomous mobile robots and several workstations. In this paper, cooperation is classified into individual action and collaborating action, and both actions are executed by the actual mobile robots in task assignment, team organization and synchronized motion based on communication. To verify the functional effect, we have integrated these strategies into a cooperative algorithm, and have experimented with it by which multiple mobile robots accomplished cooperatively the object pushing. Finally, we show the experimental results.

## 1. INTRODUCTION

Though flexible and robust robotic systems have been required in various fields, sufficient functionality has not yet been realized with currently available technologies in spite of all the efforts to develop intelligent robots. Instead of research oriented to a single sophisticated robot, research discussing multi-agent robotic systems such as DARS (distributed autonomous robotic systems) have recently attracted the attention of many researches as a new approach for flexible and robust systems [1].

Cooperative motion among multiple mobile robots can be classified into two types: individual action and collaborating action. As for individual action, studies on modest cooperation [2], traffic rule-based motion [3] and side-by-side motion [4] have been reported so far. On the other hand, as for collaborating action, studies have error recovery [5], cooperative handling motion [6] and cooperative object pushing [7] have been reported. Although the above studies have focused on some aspects of cooperation, the discussion has not sufficiently done yet. As for integrated actions, although Sugie *et al.* have demonstrated an integrated collaborating action [8], communication will be necessary to implement a distributed manner, because it is difficult for cooperation that each robot interacts only by using various sensors.

The aim of this paper is that every robot acquires autonomously information of its own situation by using communication, and decides how to cooperate by individual or collaborating action, even if knowledge on the other robots is not obtained previously.

We have developed a distributed autonomous robotic system, ACTRESS (*ACTor-based Robots and Equipment Synthetic System*), which is composed of multiple mobile robots and several workstations [9], and discussed the strategy how to achieve cooperative motion in a synthetic manner by using communication [10]. Taking an example of object pushing, this paper addresses the implementation of cooperative motion for the individual and the collaborating actions by multiple mobile robots based on the strategy by integrating various aspects of cooperation, i.e. task assignment [11], synchronized motion [12] and team organization [13], for collaboration. This paper describes the concept of ACTRESS and essential strategies for cooperation and discusses how to implement cooperative methods. Then, to verify the efficiency of implemented methods, the experimental results are shown.

## 2. DISTRIBUTED AUTONOMOUS ROBOTIC SYSTEM ACTRESS

### 2.1. Basic concept of ACTRESS

The concept of the ACTRESS is illustrated in Fig. 1. It is composed of multiple robotic agents called robotors (robotic actors). In this concept, communication between agents is assumed, taking account of the difficulty of message translation only by sensory information. Thus, each robot should be autonomous and have the ability to communicate with other robotors by message exchange.

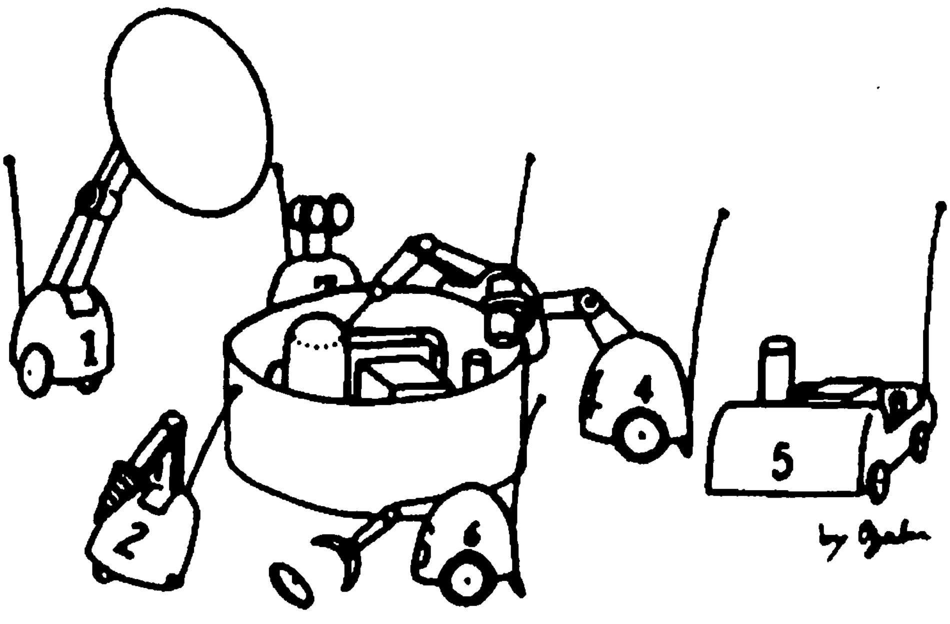


**Figure 1.** Concept of ACTRESS.

### 2.2. Prototype system of ACTRESS

Based on evaluation of functional distribution [14], we have developed a prototype system of ACTRESS. Figure 2 shows the configuration of the prototype system. This system is composed of three autonomous mobile robots and several stationary agents (workstations) with different functions such as Human Interface (HI/F) [15], Global Environment Manager (GEM) [16] and Gateway (GW) [17]. The HI/F sends messages or instructions from a human operator to other agents, and displays the messages from other agents to the operator. The GEM manages environmental information, updates it according to reports from robots and supplies it on demand. Based on the design of communication framework [18], we have also developed a communication system by contention access (CSMA/CD or CA access), which enables message passing from any agent to the other(s). The GW performs a role of a gateway forwarding messages between two communication media, i.e. wireless communication (radio LAN) for mobile agents and wired communication (Ethernet LAN) for stationary agents.

### 2.3. Basic strategy for cooperation

In order to develop a DARS including multiple autonomous mobile robots and to make the robotic system achieve various missions, it is necessary to discuss synthetically how each robot should deal cooperatively with various problems in any aspects. Cooperation by multiple mobile robots is classified into two types:

(1) *Individual action.* This is a type of cooperative action where each robot processes different tasks individually and in parallel. In this case, a conflict measure (i.e. functions to cope with conflicts in resources which should be shared by multiple robots) is required for deadlock avoidance or resolution. In other words, this is cooperation so as not to disturb other robots' actions.

(2) *Collaborating action.* This is another type of cooperative action where plural robots process a common task in a collaborative manner. In this case, a starvation

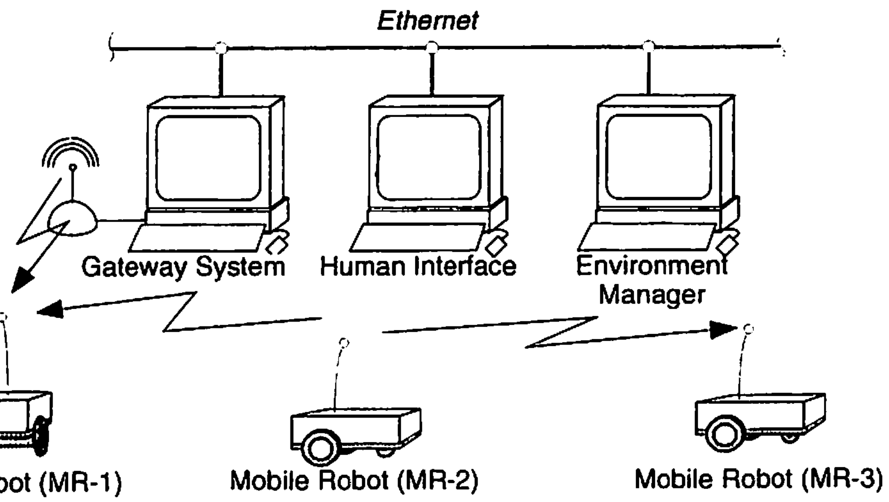


**Figure 2.** Configuration of ACTRESS.

measure (i.e. functions to cope with situation keeping on waiting for help forever) is required for organization of cooperative teams and collaboration. In other words, this is cooperation so as to support other robots' actions.

Each robot can perform both types of action. As for the collaborating action, it is necessary to achieve consistent behavior among several autonomous robots. Consequently, we considered one robot should become a coordinator which is responsible for task achievement. The coordinator can be determined depending on situations. Other robots than the coordinator are called cooperators. Thus, a team in collaborating action is composed of the following two types of robots:

(1) *Coordinator*. This is a robot which coordinates a collaborating team for the task execution. The coordinator organizes a team which consists of multiple robots, negotiating with the other robots with required functions. The coordinator communicates with cooperator(s) to accomplish a task which cannot be achieved by a single robot.

(2) *Cooperator*. This is a robot which supports the coordinator. In the collaborating action, one or several cooperator(s) follow the coordinator and carry out the motion commanded by the coordinator in order to accomplish the common target.

## 3. METHOD OF COOPERATIVE OBJECT PUSHING TASK

In this paper, we take an object pushing task [11] as a typical example of a cooperative task which requires both individual action and collaborating action among multiple mobile robots. The mission of the object push task is to clear a room by moving all the scattered objects to walls. The environmental information including the initial position of objects is given to each robot in advance. We assume that there are two types of objects (light objects and heavy ones). The light objects can be pushed by a single robot in the individual action, but the heavy ones should be pushed by multiple robots in the collaborating action. The weight of the objects is not known to robots

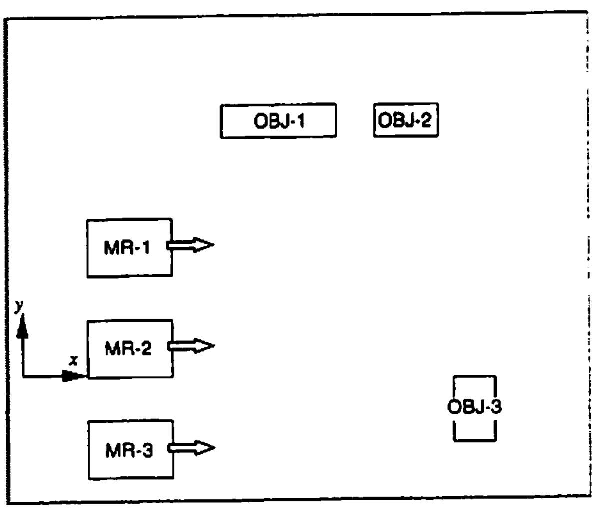


**Figure 3.** Environment of object pushing task.

and each robot recognizes the weight only when it tries to push it. Figure 3 represents an environment of the object pushing mission in a 2D space. OBJ-$i$ denotes an object and MR-$j$ denotes a robot, respectively. The white arrows in this figure indicate the initial headings of each robot.

In order to achieve the object pushing by actual mobile robots, it is necessary to deal with task assignment and team organization. In addition to motion control in individual action, synchronization for object pushing by multiple robots is required in collaborating action.

## 4. IMPLEMENTATION OF COOPERATIVE OBJECT PUSHING

### *4.1. Design of multiple processing system for individual and collaborating actions*

In order to realize the object pushing task by actual multiple mobile robots, it is necessary for each robot to execute parallel multiple processes such as control of motion, communication, etc. We designed a multiple processing system in a robot as shown in Fig. 4 and implemented it on a real-time OS (operating system). This processing system consists of a communication process, a self-motion process and a supporting-motion process. In the self-motion process, a procedure for individual action or a procedure of a coordinator for collaborating action is executed. On the other hand, in the supporting-motion process, a procedure of a cooperator for collaborating action is executed. The communication process monitors any event occurrence by message exchange and activates the self-motion process or the supporting-motion process alternatively. We describe these processes in detail in the following sections.

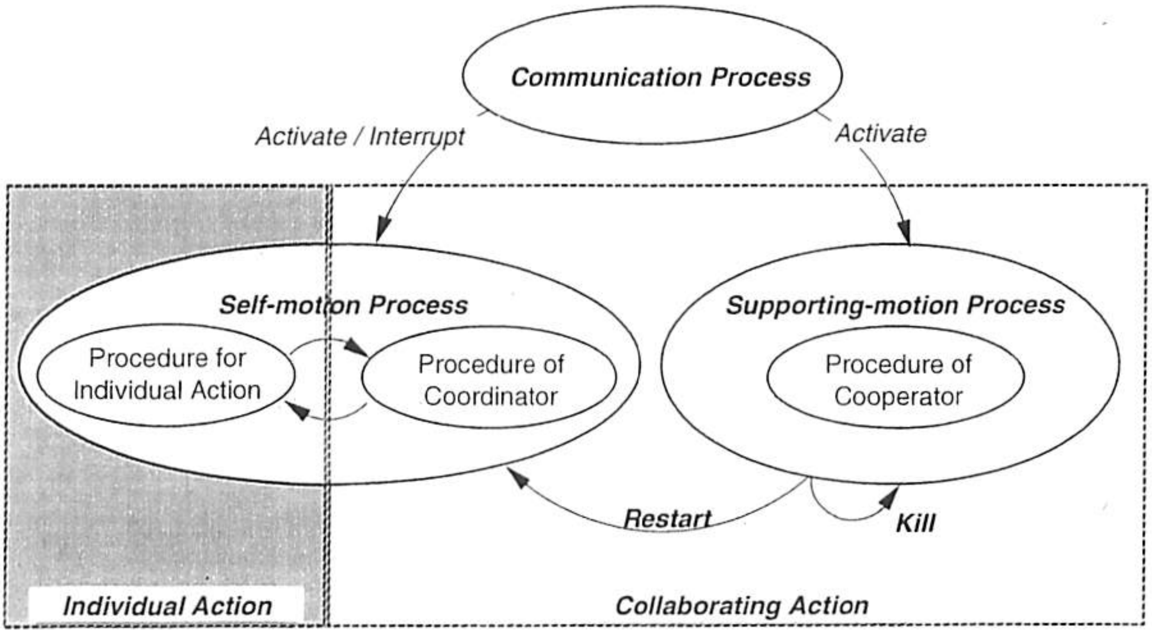


**Figure 4.** Outline of processing system for cooperative motion.

### *4.2. Communication process*

In order to implement the negotiation method (proposed so far [16]), we have designed the structure of the communication process as shown in Fig. 5. A status table, a knowledge on the other robots and protocol procedures in this process are provided for the communication process. Variables in the status table represent the current situations which include a priority of executing tasks, a current performance for executing tasks and an executing process. In this table, the task priority, the current performance and the executing process are coded by values of 0–255, 0–100, and characters *Individual action*, *Coordinator* and *Cooperator*, respectively. A low value of the task priority variable corresponds to high priority and a high value of the current performance variable means a high performance. The protocol procedure module is provided with procedures for five communication types: *negotiation*, *inquiry*, *offer*, *announcement*, *synchronization* and *reply*. These procedures are implemented based on *the message protocol core* [18]. The types of communication are specified in the protocol procedure according to a problem from the other processes.

During message passing between agents, the knowledge-base records acquire information of the message such as an ID, a value of performance, etc., of all the robots, and updates a knowledge-base related to the status of each agent from the recorded information. Consequently, the robot recognizes surrounding robots and learns the status of the other robots. Therefore, the robot does not have to communicate to acquire information about the status as it already has the knowledge about the status, which reduces the communication load and works the system efficiently.

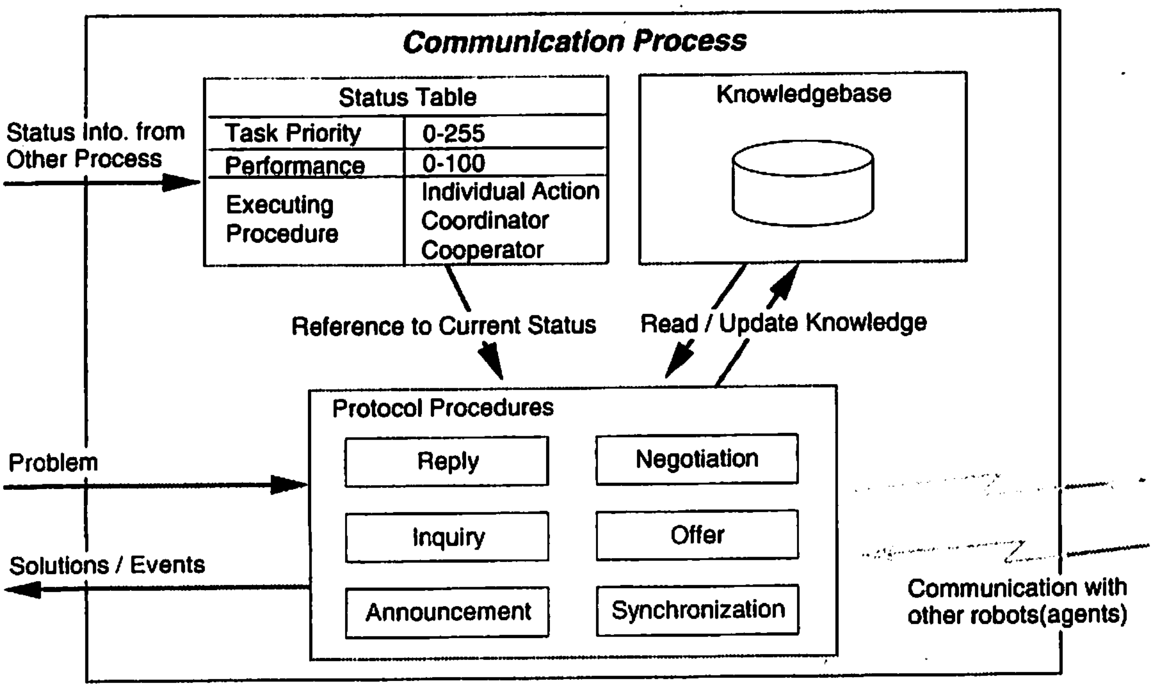


**Figure 5.** Structure of communication process.

### *4.3. Motion process*

Figure 6 shows details of the structure of the motion process. If a robot receives message, 'Start a mission' from the HI/F, each robot acquires environmental information negotiating with the GEM by communication and starts executing the procedure for the individual action. The environmental information includes the position and shape of each object in a room. Every robot can select one object as a target in the mission according to this information, based on criteria of minimum distance in this implementation. It adopts *the task assignment method* in order to achieve conflict resolution. The robot carries out motion planning. In this implementation, a planned path to the object is combined of segments of two straight lines and one circular arc. The robot moves to the object based on the path. When the robot arrives at the object, it detects the weight by using a force sensor. If the object is light, the robot can push it by the individual action. If the object is heavy, the robot executes a procedure of coordinator for the collaborating action. Namely, the robot becomes a coordinator. In this procedure, the coordinator organizes a collaborating team by using negotiation. During this negotiation, it transmits the position of the heavy object in a message to cooperator(s) and waits until cooperator(s) arrive at the position. Then the coordinator and cooperator push the object at the same time based on *the synchronized motion method*.

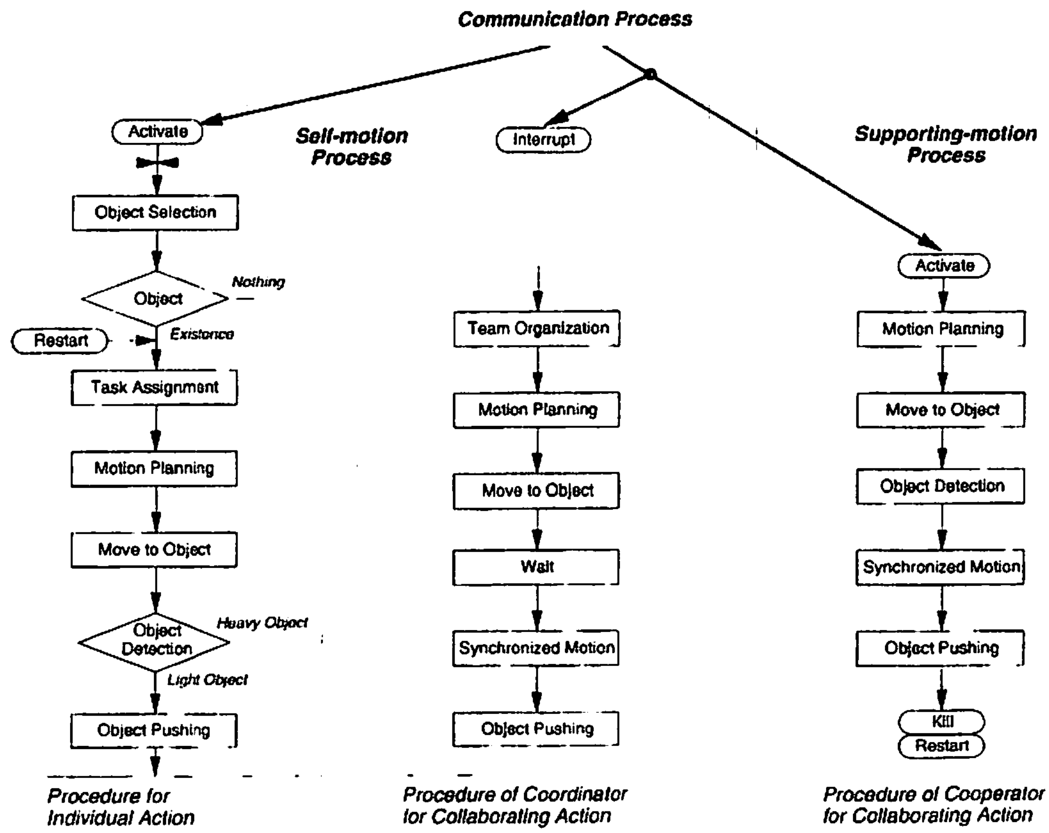


**Figure 6.** Structure of motion process.

When the communication process receives a request of collaboration from a coordinator, the robot compares the coordinator's status information with its own information on the status table and decides suitable behavior. In this implemented communication process, the robot accepts the request if the coordinator's task priority value is higher than its own one, if its own performance value is higher than the required performance and if it is executing the individual action or idling. In this case, the communication process in the robot interrupts the self-motion process and then activates the supporting-motion process to become a cooperator. The cooperator moves a position of a heavy object, which is given by the coordinator. If the cooperator(s) arrive at the position, it sends a message of arrival to the coordinator. Therefore, the cooperator can achieve the cooperative object pushing task based on the request from the coordinator.

### 4.4. Negotiation

To achieve object pushing based on the collaborating action, a coordinator needs to organize a collaborating team by using the negotiation procedure as shown in Fig. 7. The coordinator compares its own performance and required performance to push a heavy object. If the own performance does not satisfy the requirement, the coordinator broadcasts a message, 'request of object pushing' with 'the value of required performance' and 'the position of the heavy object' to all the robots in Fig. 7a. In Fig. 7b, if the current performance of each robot is higher than the required performance from the coordinator, the robot replies the performance and his position. Therefore, the coordinator can recognize the status of each robot with sufficient performance and selects near the existent robot(s) as cooperator(s). In Fig. 7c, the coordinator sends a message, 'accept,' to the selected robot(s) and also

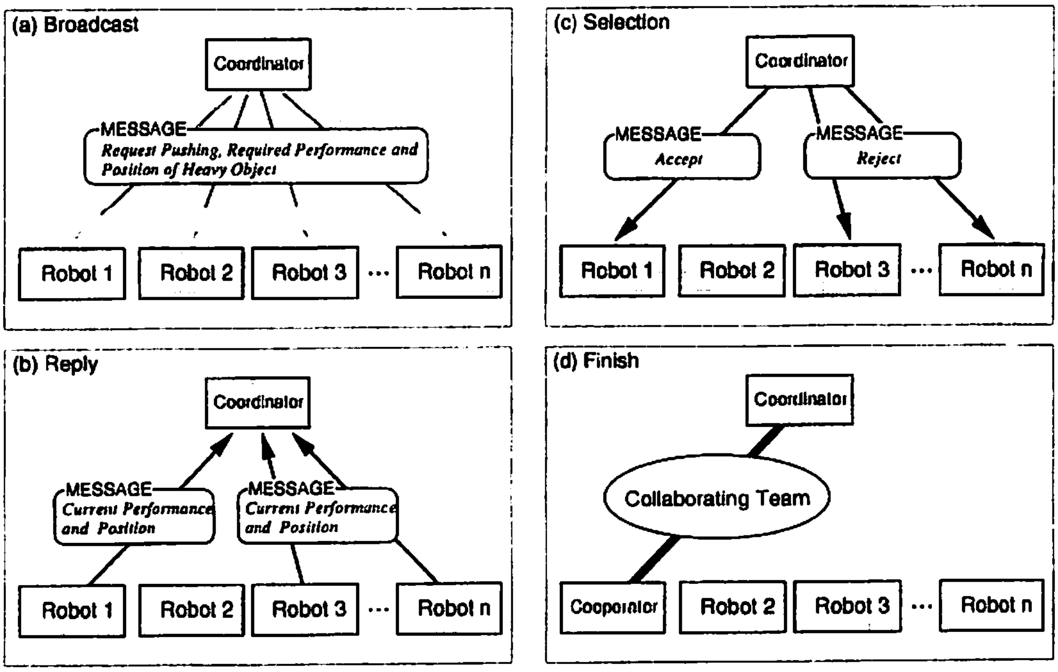


**Figure 7.** Procedure of team organization using negotiation.

sends a message, 'reject,' to the non-selected robot(s). Then coordinator and selected robot(s), the cooperator(s), are organized as a collaborating team in Fig. 7d.

## 5. EXPERIMENT OF OBJECT PUSHING BY THREE MOBILE ROBOTS

### *5.1. Hardware of the autonomous mobile robot*

Figure 8 illustrates the structure of an autonomous steering-type mobile robot used in the experiments. The locomotion system is composed of two front driving wheels driven by one DC servo motor (DC 24 V/80 W) through a differential gear, and one rear wheel to steer the robot through another DC servo motor (DC 24 V/3 W). The position and the orientation of the robot can be estimated from the rotating speeds of encoders on the left and right wheel axes.

The sensing system is composed of ultrasonic sensors in four directions (front, rear, left and right), one photoelectric sensor and three tactile sensors in front. The distances from the robot to the obstacles in each direction can be measured by the

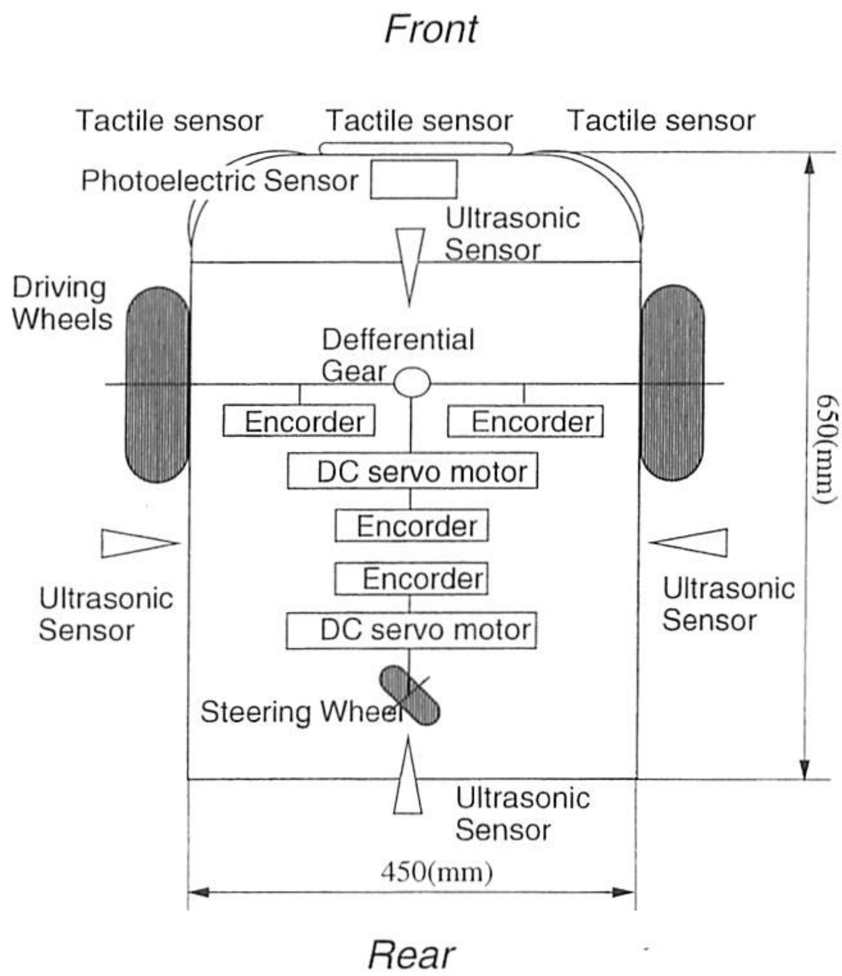


**Figure 8.** Hardware of mobile robot.

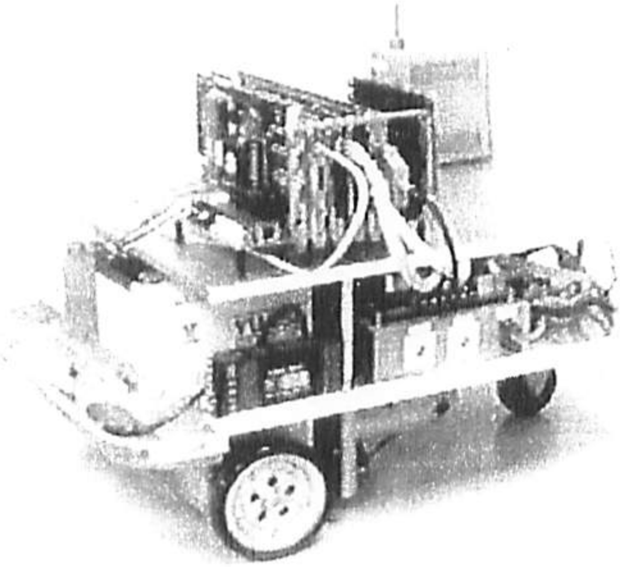

**Figure 9.** Autonomous mobile robot. The control system (the i486DX4 processor board) and the wireless LAN transmitter are mounted on the robot.

front ultrasonic sensor. The photoelectric sensor at the front is mounted for detecting collision against obstacles within a certain range. It can be judged whether the object which a robot is pushing is light or heavy by using the tactile sensors.

The control system is composed of a processor board, an ethernet board, two motor controllers and a digital I/O board on a common PC/AT ISA bus. The processing system is equipped with an i486DX4 processor (100 MHz) and 4 M byte memories. The ethernet board is connected with a wireless LAN transmitter for inter-robot communication. The motor controllers are used for the steering wheel control or for the driving wheels control, and the digital I/O board is used for the interface of the sensors, respectively. Therefore, the hardware structure mentioned above allows each robot to operate autonomously. Figure 9 shows the mobile robots.

### *5.2. Experimental environment and conditions*

In order to verify the functionality of the implemented system and processes for cooperation, we have experimented with the cooperative motion among the three mobile robots in object pushing. The experimental environment is shown in Fig. 3. Table 1 indicates names, IDs and positions of all the robots in this figure, and Table 2 indicates names, sizes and positions of all the objects, respectively. Here, we assume that a heavy object can be pushed by two mobile robots. Information concerning the weight of objects and the other robots is not given to all the robots. Therefore, every robot must negotiate with the other robots if it needs information to execute the task. In the experiment, the OBJ-1 is heavy.

**Table 1.**
Positions of mobile robots

| Robots | ID | $x$ (m) | $y$ (m) | $\theta$ (deg) |
|---|---|---|---|---|
| MR-1 | STMR0001 | 1.0 | 1.0 | 0 |
| MR-2 | STMR0002 | 1.0 | 0.2 | 0 |
| MR-3 | STMR0003 | 1.0 | −0.6 | 0 |

**Table 2.**
Positions of objects

| Objects | $W$ (m) | $B$ (m) | $x$ (m) | $y$ (m) | $\theta$ (deg) | Weight |
|---|---|---|---|---|---|---|
| OBJ-1 | 0.90 | 0.25 | 2.0 | 2.0 | 90 | heavy |
| OBJ-2 | 0.50 | 0.25 | 3.0 | 2.0 | 90 | light |
| OBJ-3 | 0.52 | 0.32 | 3.5 | −0.3 | 0 | light |

### *5.3. Experimental results*

A series of photographs in Fig. 10 shows experimental results of a demonstration motion of robots which are executing the object pushing task. Figure 11 shows the recorded communication logs of exchanged messages during this demonstration.

Firstly, when a mission, 'task start,' is given by the H/IF, all the robots start motion for object pushing. Since these robots do not know any positional information of the objects at the initial state, they acquire the information through the communication network by negotiating with the GEM. Each robot selects an object based on *the task assignment method*, and makes a plan to push the object (Step 1). As the result in this demonstration, MR-1, MR-2 and MR-3 select OBJ-1, OBJ-2 and OBJ-3, respectively. Based on the motion plan, they execute the motion to push the object (Step 2). When MR-1 reaches OBJ-1, MR-1 recognizes that the OBJ-1 is a heavy object and becomes a coordinator to organize a collaborating team to deal with the heavy object. In order to push the object based on the collaborating action, MR-1 shifts its position a little to the left side, i.e. it moves backward (Step 3), turns to the left and then to the right (Step 4), and approaches to the left side of the object (Step 5). Then, it starts negotiation to organize a collaborating team based on *the team organization method*. In this step, the coordinator broadcasts a message, which is the request for a cooperator to push the object. Log 1 in Fig. 11 denotes the requesting message. The format of this message is based on *the message protocol core* [18]. In this figure, *To*, *From*, *Control*, *Class*, *Type* and *Message* are the fields for a name of receiver, a name of sender, message control, message class, message type and message body, respectively. 'STMR' represents an attribute of a robot type, which indicates the steering-type mobile robot. The asterisk in the 'TO' field represents a wild-card. Namely, the message of Log 1 is sent only to all the steering-type mobile robots. The message, 'REQ_PUSH 80 200 200 90,' represents the request to push an object, the required performance and the position ($x$ [cm], $y$ [cm], $\theta$ [deg]), respectively. Namely, the coordinator requires a robot which has performance of 80 at the position of (200 [cm], 200 [cm], 90 [deg]). Both of Log 2 and 3 are reply

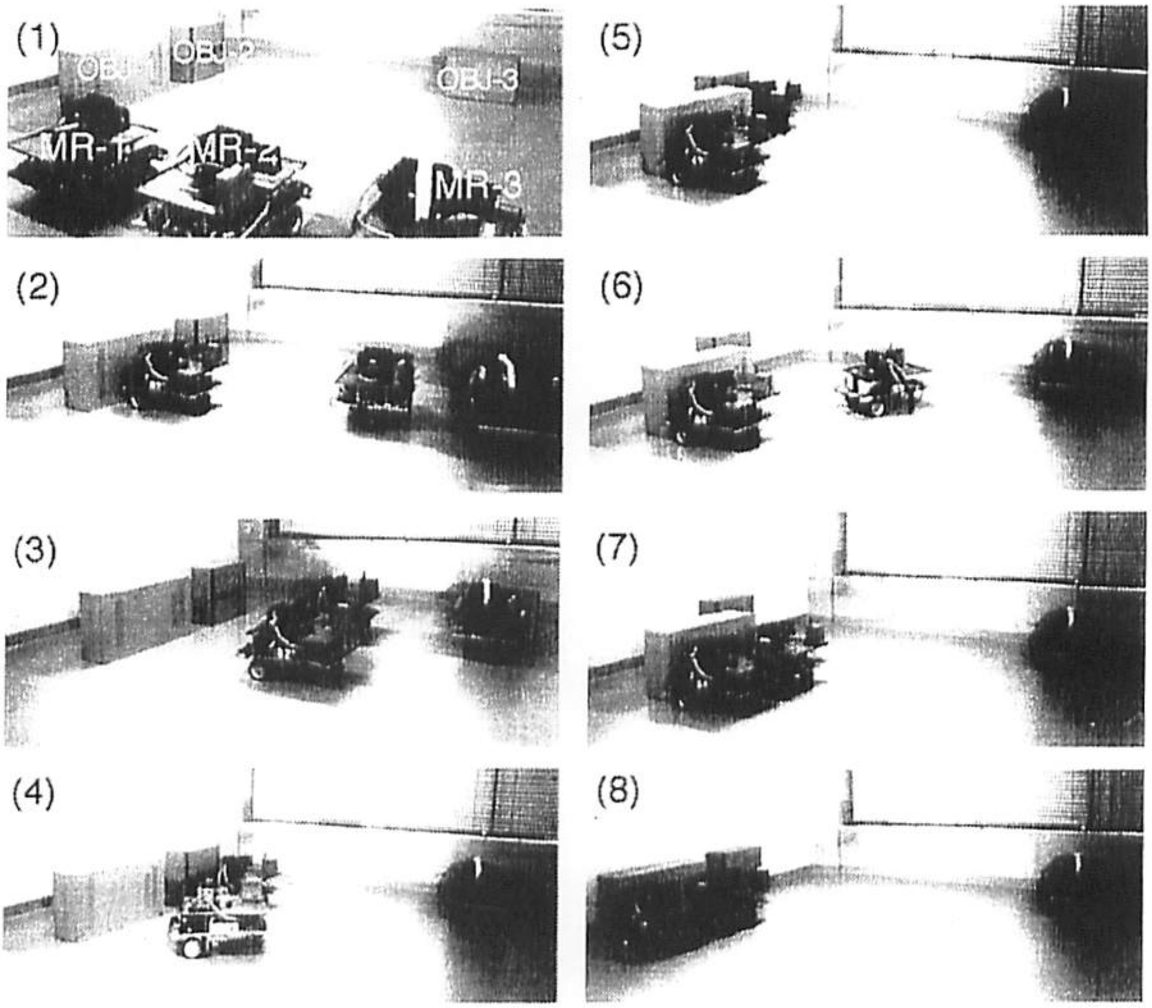


**Figure 10.** Experimental result. Each robot selects an object based on *the task assignment method* (Step 1) and moves to the position of the object (Step 2). MR-1 detects the heavy object, OBJ-1, and becomes the coordinator (Step 2). The coordinator backwards (Step 3), turns and moves the left side of the object (Step 5 and 6). It *organizes* a team consisting of the cooperator, MR-2 (Step 6). The cooperator moves to the right side of the object (Step 6 and 7). Finally, MR-1 and MR-2 achieve the cooperative object pushing by *the synchronized motion method.*

messages to this request. These messages include the performance and the current position of each robot. Accordingly, the coordinator can recognize the status of these robots and selects MR-2 as a cooperator since MR-2 is nearer to the coordinator in this demonstration. The coordinator sends the rejection message to MR-3 (Log 4) and also sends the acceptance message to MR-2 (Log 5). Then the collaborating team is organized to push the heavy object, which consists of MR-1 (the coordinator) and MR-2 (the cooperator). The cooperator moves to the right side of the heavy object (Step 6 and 7) and sends the message which represents the arrival at the position (Log 6). The coordinator makes a path plan for the synchronized motion to push the object in a coordinated manner, and informs the cooperator of this plan (Log 7). The message, 'SYNC_MOVE 20 50 0 0 0', represents the condition of the synchronized motion; the speed rate and the goal position ($x$ [cm], $y$ [cm] and $\theta$ [deg]) on the robot coordinates. Therefore, the cooperator can recognize the requirement; the speed 20

```
Log.1: Sat Aug 12 19:41:51 1995
TO      : STMR****
FROM    : STMR0001
CONTROL: -
CLASS   : CORE
TYPE    : COOP_NEGO_REQ
MESSAGE: REQ_PUSH 80 200 200 90

Log.2: Sat Aug 12 19:41:52 1995
TO      : STMR0001
FROM    : STMR0002
CONTROL: 1
CLASS   : CORE
TYPE    : COOP_NEGO_REP
MESSAGE: BID_PUSH 100 204 128 90

Log.3: Sat Aug 12 19:41:53 1995
TO      : STMR0001
FROM    : STMR0003
CONTROL: 1
CLASS   : CORE
TYPE    : COOP_NEGO_REP
MESSAGE: BID_PUSH 80 322 -200 179

Log.4: Sat Aug 12 19:41:56 1995
TO      : STMR0003
FROM    : STMR0001
CONTROL: 1
CLASS   : CORE
TYPE    : COOP_NEGO_RES
MESSAGE: REJECT

Log.5: Sat Aug 12 19:41:56 1995
TO      : STMR0002
FROM    : STMR0001
CONTROL: 1
CLASS   : CORE
TYPE    : COOP_NEGO_RES
MESSAGE: ACCEPT

Log.6: Sat Aug 12 19:42:55 1995
TO      : STMR0001
FROM    : STMR0002
CONTROL: 2
CLASS   : CORE
TYPE    : OFFER
MESSAGE: ARRIVAL

Log.7: Sat Aug 12 19:42:56 1995
TO      : STMR0002
FROM    : STMR0001
CONTROL: 2
CLASS   : CORE
TYPE    : INQ_SYNC
MESSAGE: SYNC_MOVE 20 50 0 0

Log.8:Sat Aug 12 19:42:57 1995
TO      : STMR0002
FROM    : STMR0001
CONTROL: 3
CLASS   : CORE
TYPE    : SYNC
MESSAGE: START
```

**Figure 11.** Communication logs during the object pushing task. Log 1 means that MR-1, coordinator, requests to push based on the collaborating action to all the robots. The messages of both Log 2 and Log 3 are replied to according to this request. As coordinator (MR-1) selects MR-2 as the cooperator, it sends the reject message to MR-3, Log 4, and sends the acceptance message to MR-2, Log 5. Log 6 means that the cooperator arrived at the right of object. Log 7 means the planned motion of the cooperator. Log 8 is the message for synchronization.

and the position (50 [m], 0 [m], 0 [deg]), which means the robot moves straight forward to 50 [cm] in front. After the coordinator transmits a synchronization signal (Log 8), the coordinator and the cooperator achieve collaborative object pushing by *the synchronized motion*. Figure 12 illustrates the trajectories of motions of each robot achieved in this experiment.

Figure 13 shows the trajectories of motions of each robot achieved in another experiment, in which all the objects are just light objects which can be pushed by a single robot in parallel.

Even if some robots are in the situation that they need cooperators to push heavy objects, the starvation deadlock, in which some robots are waiting for cooperators mutually in conflict with each other, can be avoided with the proposed method, because all the robots have an essential rule that they should support a coordinator than to become a coordinator. In addition, concentration of a number of messages does not occur because simultaneous communication can be excluded based on the CSMA method. Consequently, message exchange in the implemented system is carried out in sequence.

If an object is more heavy and needs to be pushed by plural robots with four or more cooperators, it is impossible to push this object in the implemented system. In this case a coordinator which selects the heavy object cannot organize a collaborating team, and then reports this situation to the HI/F. If the system is composed of a large

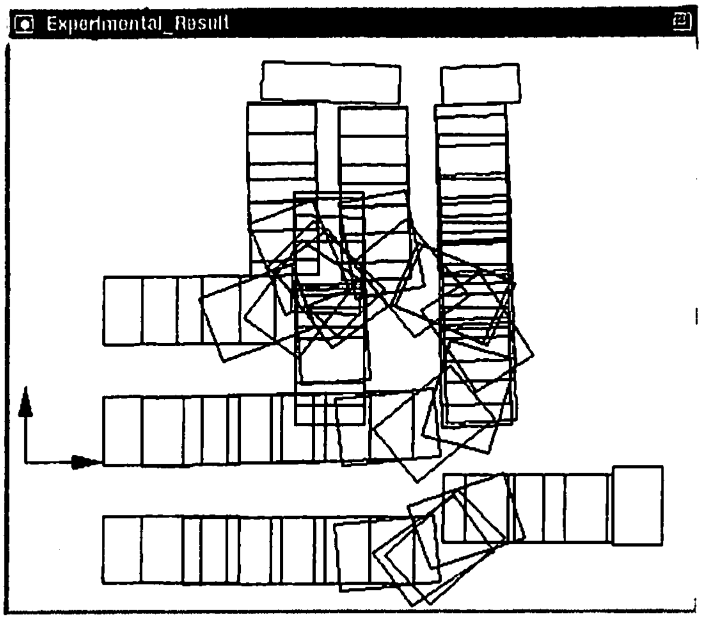

Figure 12. Trajectories of motions of each mobile robot by collaborating action.

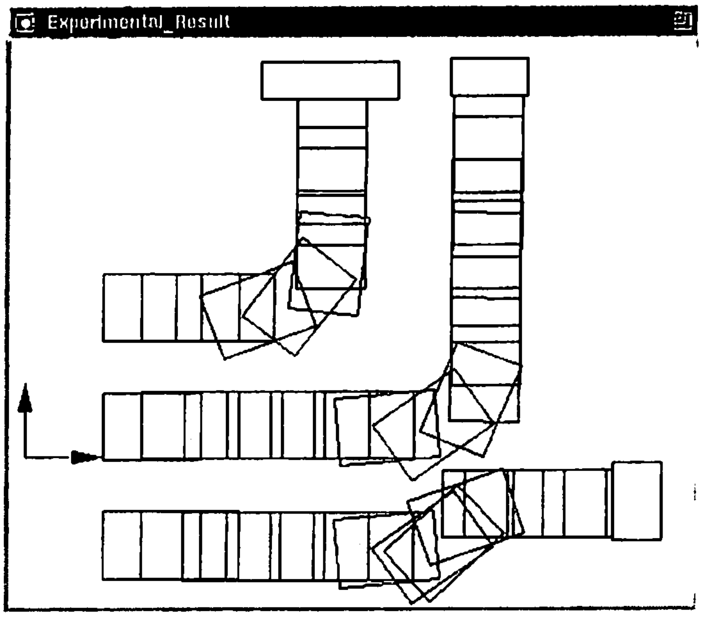

Figure 13. Trajectories of motions of each mobile robot by individual action.

number of robots and/or some high-performance robots, the coordinator can organize a collaborating team for the heavy object. The availability of organization such as above is shown in the experiment the simulation described previously [13].

## 6. CONCLUSION

In this paper, in order to achieve object pushing by multiple mobile robots, we have implemented strategies and methods for communication-based cooperation, by integrating *the task assignment method, the team organization method and the synchronized motion method.*

In this experiment, three actual mobile robots have achieved the object pushing task both by individual action and collaborative action. The features of the proposed method can be summarized as follows:

(1) By integration of both individual action and collaborating action, all the robots can achieve flexible and consistent cooperative motion.

(2) Each robot is able to behave autonomously, acquiring necessary information, which is unknown in advance, by exchanging messages through communication.

(3) Even if a robot finds an object which can not be processed by a single robot, it can organize a collaborating team autonomously and achieve the pushing task collaboratively.

Through the experimented results, the method for cooperative object pushing proposed in this paper was verified.


### *Acknowledgements*

The authors thank Professor A. Matsumoto (Toyo University), Professor K. Yokota (Utsunomiya University) and Mr. Y. Ishida (Kyushu University) for helpful suggestions and valuable comments on the methods presented in this paper.


## REFERENCES


1. H. Asama, T. Fukuda, T. Arai and I. Endo, eds, *Distributed Autonomous Robotic Systems.* Berlin: Springer-Verlag, 1994.
2. S. Yuta and S. Premvuti, "Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots," in *Proc. Int. Symp. on Distributed Autonomous Robotic Systems*, 1992, p. 173.
3. S. Kato and J. Takeno, "Fundamental studies on the application of traffic rules to the mobile robot world," in *Proc. 5th Int. Conf. on Advanced Robotics*, Pisa, Italy, 1991, pp. 1063–1068.
4. L. E. Parker, "Designing control laws for cooperative agent teams," in *Proc. IEEE Int. Conf. on Robotics Automat.*, Atlanta, GA, 1993, pp. 582–587.
5. F. R. Noreils, "Toward a robot architecture integration cooperation between mobile robots: application to indoor environment," *Int. J. Robotics Res.*, vol. 12, no. 1, pp. 79–98, 1993.
6. M. Kajitani, K. Fukuo and T. Masuda, "Basic study of cooperative motion of mobile robots," *J. Robotics Mechatron.*, vol. 2, no. 2, pp. 46–49, 1990.
7. L. E. Parker, "ALLIANCE: an architecure for fault tolerant, cooperative control of heterogeneous mobile robots," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Federal Armed Forces University, Munich, Germany, 1994, pp. 776–783.
8. H. Sugie, Y. Inagaki, S. Ono, H. Aisu and T. Unemi, "Cooperation among multiple mobile robots using intention inference," in *Proc. 4th IEEE Int. Conf. on Fuzzy Systems and 2nd Int. Fuzzy Engineering Symp.*, Yokohama, Japan, 1995, pp. 1707–1712.
9. H. Asama, A. Matsumoto and Y. Ishida, "Design of an autonomous and distributed robot system: ACTRESS," in *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, Tsukuba, Japan, 1989, pp. 283–290.
10. H. Asama, H. Kaetsu, I. Endo, K. Ozaki, Y. Ishida, K. Yokota and A. Matsumoto, "Strategy for cooperative motion of multiple robots in a distributed robotic system ACTRESS," *SME Trans. Robotics Res.*, MS94-236, pp. 1–11, 1994.

11. H. Asama, K. Ozaki, A. Matsumoto, Y. Ishida and I. Endo, "Development of task assignment system using communication for multiple autonomous robots," *J. Robotics Mechatron.*, vol. 4, no. 2, pp. 122–127, 1992.
12. K. Ozaki, H. Asama, Y. Ishida, A. Matsumoto, K. Yokota, H. Kaetsu and I. Endo, "Synchronized motion by multiple mobile robots using communication," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Yokohama, Japan, 1993, pp. 1164–1169.
13. K. Ozaki, H. Asama, Y. Ishida, K. Yokota, A. Matsumoto, H. Kaetsu and I. Endo, "Negotiation method for collaborating team organization among mutiple robots," in *Distributed Autonomous Robotic Systems*, H. Asama, T. Fukuda, T. Arai and I. Endo, eds, Berlin: Springer-Verlag, 1994, pp. 199–210.
14. H. Asama, M. K. Habib, I. Endo, K. Ozaki, A. Matsumoto and Y. Ishida, "Functional distribution among multiple mobile robots in an autonomous and decentralized robot system," in *Proc. IEEE Int. Conf. on Robotics Automat.*, Sacramento, CA, 1991, pp. 1921–1926.
15. K. Yokota, T. Suzuki, H. Asama, A. Matsumoto and I. Endo, "A human interface system for the multi-agent robotic system," in *Proc. IEEE Int. Conf. on Robotics Automat.*, San Diego, CA, 1994, pp. 1039–1044.
16. H. Asama, K. Ozaki, Y. Ishida, M. K. Habib, A. Matsumoto and I. Endo, "Negotiation between multiple mobile robots and an environment manager," in *Proc. 5th Int. Conf. on Advanced Robotics*, Pisa, Italy, 1991, pp. 533–538.
17. Y. Ishida, S. Tomita, H. Asama, K. Ozaki, A. Matsumoto and I. Endo, "Development of an integrated communication system for multiple robotic agents," in *Proc. Int. Symp. on Distributed Autonomous Robotic Systems*, Saitama, Japan, 1992, pp. 193–198.
18. Y. Ishida, H. Asama, I. Endo, K. Ozaki and A. Matsumoto, "Communication and cooperation in an autonomous and decentralized robot system," in *Proc. Int. Symp. on Distributed Intelligent Systems*, Arlington, 1991, pp. 299–304.


## ABOUT THE AUTHORS

**Koichi Ozaki** graduated from Toyo University in 1992 and received the Doctor degree in Engineering from Saitama University in 1996. He worked as a post-doctoral researcher in the Institute of Physical and Chemical Research (RIKEN) from 1996 to 1997 and joined Department of Mechanical Systems Engineering, Faculty of Engineering, Utsunomiya University in 1997. Currently, he is a lecturer. His main interests are distributed autonomous robotic systems, mobile robots and environmental systems. He is a member of RSJ, JSME, JSPE and Association of Environmental Instrumentation Control and Automation.

**Hajime Asama** graduated from the University of Tokyo in 1982 and received the Doctor degree of Engineering from the University of Tokyo in 1989. In 1986 he joined the Institute of Physical and Chemical Research (RIKEN) and he is currently a senior research scientist in RIKEN. He received JSME Robomec Award in 1995. He played a role in the editorship of '*Distributed Autonomous Robotics Systems*' and its second volume which were published by Springer-Verlag, Tokyo in 1994 and 1996, respectively. His main interests are distributed autonomous robotic systems and intelligent bioprocess systems. He is a member of RSJ, JSME, IEEE, New York Academy of Science, etc.

**Isao Endo** graduated from the University of Tokyo in 1965 and received the Doctor of Engineering degree from the University of Tokyo in 1970. He joined the Institute of Physical and Chemical Research (RIKEN) in 1970 and is currently a chief scientist of the Chemical Engineering Laboratory. In 1989 he was co-appointed as a Visiting Professor of Saitama University. In 1994 he received a Honorary Doctor of Engineering from Helsinki University of Technology. His main interests are bioprocess engineering and distributed autonomous robotic systems. He is a member of the New York Academy of Science, AIChE, ACS, Engineering Academy of Japan.