 8

Paper:

Development of Task Assignment System Using Communication for Multiple Autonomous Robots

H. Asama, K. Ozaki*, A. Matsumoto*, Y. Ishida and I. Endo**

Chemical Engineering Laboratory, RIKEN (The Institute of Physical and Chemical Research)
2-1 Hirosawa, Wako, Saitama 351-01, Japan
* Toyo University, Faculty of Engineering,
2100 Kujirai-Nakanodai, Kawagoe, Saitama 350, Japan
** University of Tokyo, Information Network System Operation Center,
2-11-16 Yayoi, Bunkyo-ku, Tokyo 113, Japan
[Received March 5 ,1992; accepted March 7, 1992]

1. Introduction

In order to realize a robot system having flexibility, robustness, and the capability of responding to various situation, the authors has been developing an autonomous and decentralized robot system called ACTRESS (ACTor-based Robot and Equipment Synthetic System).[1] The basic principle behind ACTRESS is that functions which should be possessed by the entire robot system are distributed to multiple robots. The robots are allowed to dynamically interact according to various situation. ACTRESS consists of multiple autonomous robotic agents, including robots, computing systems, and equipment with varying functions. It involves the premise that communication is possible between these agents. The current major problem in the actual development of this system is establishing how the multiple robots are allowed to cooperate using communication.

When multiple robots are at work in an ordinary environment, various conflicting problems arise. This paper focuses on the task assignment problem. The action modes for multiple robots are parallel independent action by each robot and cooperative action by multiple robots. This paper describes a new system which determines the task assignment while each robot autonomously switches both action modes by using communication according to the situation at hand.

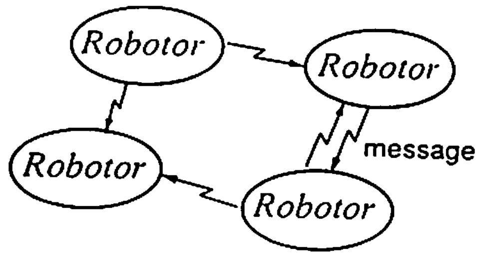

Fig. 1. Robotors and their message passing.

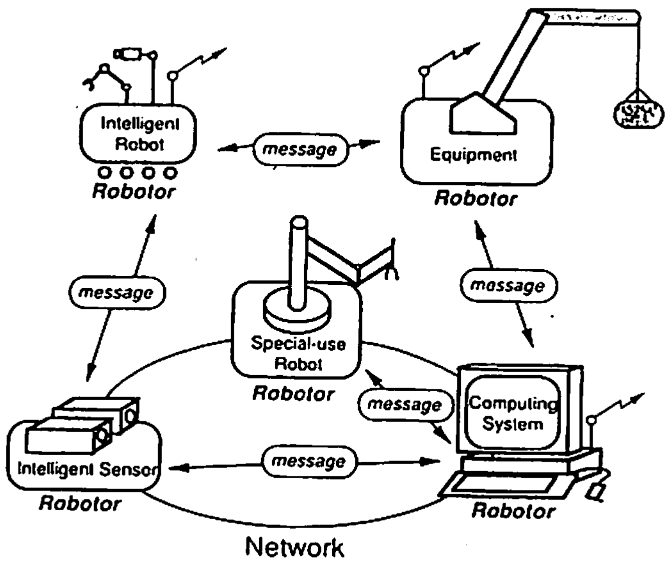

Fig. 2. Concept of ACTRESS.

2. Cooperative Motion of Multiple Robot System

2.1. Cooperation of Multiple Robots

Many methods have been proposed to realize cooperative action among multiple robots. Some examples are a method in which each robot is provided with a control structure for cooperation called multi-robot control level (MRC),[2] a scheduling method which combines centralized and distributed planning,[3] and a collision deadlock avoidance method by modest cooperation.[4] This study focuses on a task assignment problem. A framework has been proposed which allows communication for task assignment in the presence of multiple agents in a contract-net protocol.[5] It assumes an agent of centralized task management.

In contrast, ACTRESS is a system that incorporates multiple robots, or robotic agents, and provides intelligent group behavior. Each robotic agent is capable of functioning autonomously and, at the same time, communicating with the other agents. Based on the same concept, actor formalism has been proposed as a calculation model for information processing. This model executes information processing by objects, called actors, as well as message passing between them. The authors have expanded this concept into an autonomous and decentralized robot system. **Figure 1** contains a schematic diagram showing multiple agents (robotor: robotic actor) acting autonomously and message passing between them. The homogeneity of the robotic agents is not specified, instead, different functions are assumed for each robotic agent. This is because cooperation among agents with different functions can provide a wider variety of functions. **Figure 2** shows a conceptual diagram of ACTRESS. Here, the applicability of the system is considered, and elements such as robots, computers, and equipment with various functions are all regarded as robotic agent. ACTRESS is a decentralized system that does not require a supervisor. When a task is assigned, each robotic agent executes problem solving, performs information exchange by communication if necessary, and acts autonomously as well as cooperatively.

The authors previously reported research concerning cooperation among autonomous robots in ACTRESS, such as the development of the communication function between agents,[7,8] investigation of the functional distribution based on the evaluation of communications traffic,[9] the development of a communication simulator[10] and collision avoidance using communication.[11] The current study discusses how each autonomous robot determines task assignment in the decentralized system.

2.2. Action Modes of Multiple Robots

In the case that multiple robots act in a common environment and execute a common task, there are two types of action modes. When there are multiple tasks and each task can be executed by a single robot, the mode in which multiple robots act independently and parallelly (parallel action) is selected. In contrast, when each task cannot be executed by a single robot, mode in which multiple robots cooperatively perform a common task (cooperative action) is selected.

The advantages associated with the utilization of multiple robots are as follows.

(1) In the parallel mode, an increase in the number of robots results in improved efficiency by the effect of parallel processing.

(2) In the cooperative mode, an increase in the number of robots enables the performance of higher-level tasks.

(3) The presence of multiple robots allows the role of a malfunctioning robot to be assumed by another, this provides redundancy such as fault tolerance and robustness.

In the parallel mode, collisions and conflicts among robots should be avoided by assigning a different task to different robot, not by assigning the same task to multiple robots. In the cooperative mode, it is necessary, when determining task assignment to organize a task force by designating robots. This study addresses tasks in which both modes are present.

3. Communication System between Robots

The function of communication between robots is essential to the cooperative action of multiple robots, allowing a variety of information exchange between robotic agents. Assuming that there are autonomous mobile robots in the system, wireless communication is advantageous in terms of mobility. Furthermore, it is premised that each agent (robot, piece of equipment, or computer) behaves autonomously; therefore, a computer for intelligence should be incorporated in each agents. Consequently, a system for wireless communication among computers was developed in which a wireless modem is connected to each computer for communication.

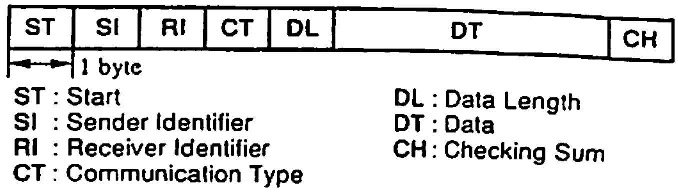

Fig. 3. Basic format of communication protocol.

Table 1. Communication type.

Symbol	Meaning
TXD	Sending Data
RXD	Acknowldge of Data Reception
ERR	Failure of Data Reception

A commercially available wireless modem was used; it sends and receives data with the frequencies of two channels (250–380MHz band). Data is received by interruption procedures. The wireless modem and the computer are connected by RS232C, and the transmission rate is 1200 bps.

Figure 3 illustrates the basic format of the communication protocol employed in the wireless communication system. In the figure, ST denotes a header, SI the ID of a sender, and RI the ID of a receiver. Furthermore, CT denotes a communication type shown in **Table 1**, DT the communication data, DL the data length of variable-length data DT, and CH is check sum for error checking.

The wireless modem has only two channels; therefore, reciprocal communication among three or more agents requires the management of communication. Consequently, communication by polling is adopted for communication among three or more agents. In this case, a specified agent passes messages among agents by sequentially executing one-to-one communication with each agent that is participating in the task. Therefore, multiple execution of one-to-one communication enables one-to-n communication, i.e. broadcasting. This method ensures the sending and receiving of messages; however, it has a disadvantage in that when the number of agents increases, communication takes time even when communication among agents is not busy.

3.2. Communication Procedures for Task Assignment

As communication functions required for task assignment, the following sending procedures are implemented:

(1) Report of ready state

This is used to confirm that all agents are ready for operation at the beginning of a task. When each agent is prepared and enters the ready state, a report is sent by broadcasting.

(2) Declaration of task selection

When each robot selects a certain task, a declaration is sent by broadcasting.

(3) Report of task completion

When each robot has finished the task, a report is sent by broadcasting.

(4) Request for cooperative processing

When a robot confronts a situation which requires cooperative processing, a request for cooperation is sent by broadcasting.

(5) Acceptance of cooperative processing

When a robot receives a request for cooperation, the reply of acceptance is sent by one-to-one communication.

(6) Report of completion of preparation for cooperation

In a cooperative task, when a robot completes preparation for the start of cooperative processing, a report is sent by one-to-one communication.

(7) Command for start of cooperative processing

In cooperative processing, after the robot which has requested cooperation receives the report of the completion of preparation for cooperation from another robot, the former sends the latter a command for the start of cooperative processing by broadcasting.

(8) Request for environmental information

When obtaining environmental information required for task selection and path planning, a request for environmental information is sent by broadcasting. If it is known in advance which agent has the environmental information, then the request is sent by one-to-one communication.

(9) Offer of task information

Responding to the request for the environmental information, the information is offered and sent by one-to-one communication.

4. Task Assignment System

4.1. Action Modes of Robots

A system has been implemented in which each robot performs a task which selecting it autonomously. Selection is done based on the information of processing situations managed by the robot and the acting situations of the other robots. In this system, each robot performs each task independently and parallelly in normal operation; however, when necessary, cooperative processing is allowed such as searching for a partner and requesting cooperation, as well as accepting and supporting the request for cooperation. This system does not control actual robots. Rather, it conducts decision making for the organization of robot groups which execute tasks in task assignment under situations that dynamically require independent or cooperative processing.

The following action modes were set up in order to express the state of each robot. These modes are based on the assumption that robots perform tasks by independent or cooperative processing while assigning tasks autonomically.

(1) Independent action mode:

State of independent and parallel operation to process a task that does not require cooperative action.

(2) Cooperative action 1 (request):

State of operating while requesting cooperation from another mobile robot in order to process a task that requires cooperative action.

(3) Cooperative action 2 (support):

State of operating while cooperating and supporting another mobile robot in order to process a task that requires cooperative action.

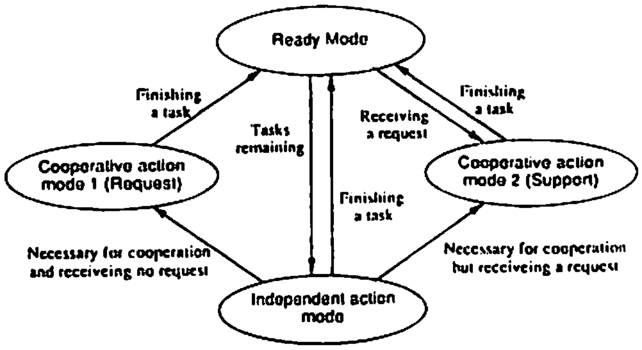

Fig. 4. Transition graph on action modes.

(4) Ready:

State of being prepared to process a task.

Figure 4 shows a transition graph of the action modes. From the ready mode, the independent action mode or cooperative action mode 2 (support) can be accessed depending on whether or not there is a request for cooperation. From the independent action mode, the cooperative action mode 1 (request) or cooperative action mode 2 (support) can be accessed depending on whether or not there is a request for cooperation. From any mode, control returns to the ready mode upon completion of each task.

When robots act cooperatively, each robot should be aware of the state of the other robots. Here, each robot is allowed to manage the action modes of the other robots in a decentralized manner. This is achieved based on the communication protocol which is described above. Furthermore, in order to determine task assignment through decision making by each robot, each robot should be aware of the states of tasks at an arbitrary point. Therefore, the states of tasks are managed in a decentralized manner based on the communication protocol described above.

4.2. Algorithm for Robot Actions

An algorithm, which dictates how a robot functions while determining task assignment, is described in the following. In the initial state, a robot assumes the ready mode and waits for all of the other robots to do the same. After the start of action, the robot checks for a request for cooperation. If there is a request, then the robot assumes cooperative action mode 2 (support); and if there is no request then it assumes an independent action mode.

In the independent action mode, tasks are executed according to the following procedures:

(1) Selection of a task
(2) Sending declaration of start of processing
(3) Path planning
(4) Moving to independent processing position
(5) Judgment as to whether or not cooperative processing is required
(6) Execution of processing by independent action
(7) Sending of report of task completion

Here, it was assumed that it cannot be judged whether the task is capable of being executed independently or requires cooperative processing until a robot moves to an independent processing position. This considers situations to dynamically generate new task requirements as processing progresses. If cooperative processing is not necessary, then execution proceeds according to this algorithm; however, if cooperation is required, then the presence of a request for cooperation is first checked. If there is a request, then a robot assumes cooperative action mode 2 (support); and if there is no request, then it assumes cooperative action model 1 (request).

In cooperative action mode 1 (request), a task is executed as follows:

(1) Sending request for cooperation
(2) Moving to cooperative processing position
(3) Waiting until completion of preparation for cooperation
(4) Sending command of start of cooperative processing
(5) Execution of task by cooperative processing
(6) Sending report of task completion

In cooperative action mode 2 (support), a task is executed as follows:

(1) Sending acceptance of cooperation
(2) Moving to cooperative processing position
(3) Sending completion of preparation for cooperation
(4) Wait until command of start of cooperative processing
(5) Execution of task by cooperative processing

Each time a task is completed, a robot assumes ready mode and checks the task situations. When all tasks have been completed, the robots judge that the mission has been completed and terminate operation. If unprocessed tasks remain, then the execution restarts. If the mission is not completed but there are no remaining unprocessed tasks, then there is a possibility of being requested by other robots. Therefore, the ready states is maintained until completion of the mission.

This algorithm features the following strategies:

(1) Each time a task is completed, a robot checks for the presence of a request for cooperation from other robots, if any, the request is accepted.

(2) Even if a robot judges that cooperation is necessary, it checks the presence of a request for cooperation from another robot before requesting for cooperation itself, if any, it accepts the request and abandons the task which it is facing.

In general, when cooperative action is required while each robot is parallelly performing independent action, the conflict of task execution takes place. This technique enables resolution of the conflict by incorporating a type of a "consideration mechanism" in which priority is given to the acceptance of a request for cooperation from another robot over acting independently or requesting cooperation from others.

5. Object Pushing Simulation

5.1. Object Pushing Task

An object pushing task was adopted as a typical task assignment problem which requires both parallel and cooperative action by multiple robots. A system has been developed to simulate execution of the task using the task assignment system. The object pushing task involves the transport of multiple objects, which are scattered in a room, to the walls. However, the target positions of objects to be pushed are not specified in advance. Two types of objects are used, a lightweight object that can be pushed by an independent robot, and a heavy object that requires cooperative pushing by multiple robots. Removal of the former can be achieved by parallel action, and remove of the latter requires cooperative action. Figure 5 shows an example of the environment of an object pushing task. Here, numbers are used to denote objects, and letters are used to denote robots. In order to manage task situations, the condition of each object was one of the following items.

(N) Not removed
(X) Robot X in removal action (X: robot ID)
(F) Removal completed.

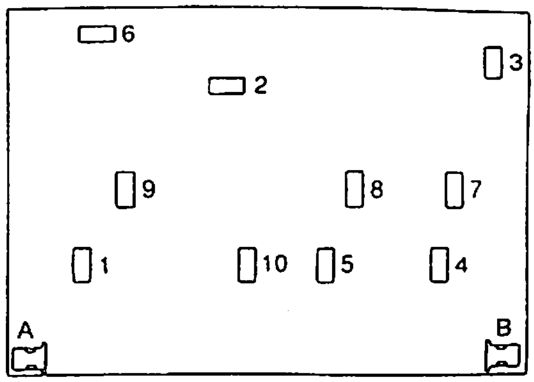

Fig. 5. Typical environment of object pushing task.

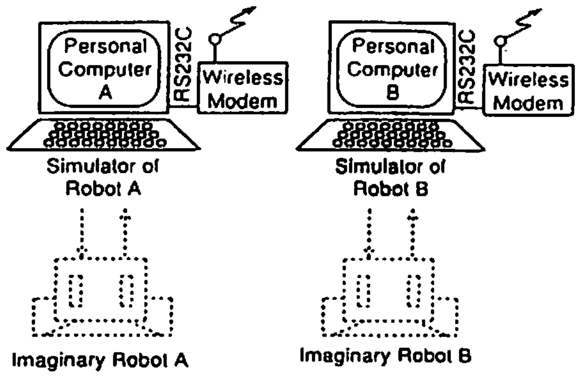

Fig. 6. Configuration of prototype system for simulation of object pushing task.

5.2. Configuration of Simulation System

In order to verify the developed task assignment technique, a prototype system to simulate the object pushing task was prepared. It consists of two personal computers to simulate two mobile robots and a wireless communication system. A task assignment system has been implemented for the computers. The configuration of the simulation system is illustrated in Fig.6. It is expected that in a practical application, a computer will be mounted on each autonomous mobile robot to execute the object pushing task. The computer simulates the decision making and action of each robot during task assignment. The trajectory of the robot in the simulation of the task is displayed on the computer. Heavy objects are removed by two cooperating robots. Furthermore, the task environment was treated as a two-dimensional space. In selecting an object, a robot is to choose the object whose center of gravity is closest to the current position of the robot. This is determined based on positional information of the object, referring to environmental information.

Path planning involves the following processes:

(1) Planning of a path of objects

The path of an object from its initial position to the wall

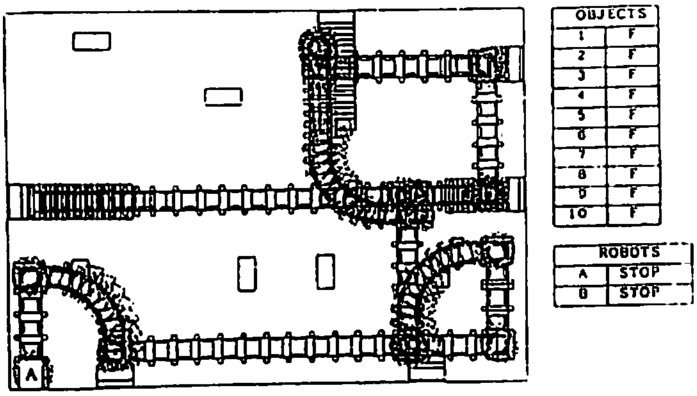

Fig. 7 (a). Simulated trajectory of robot A.

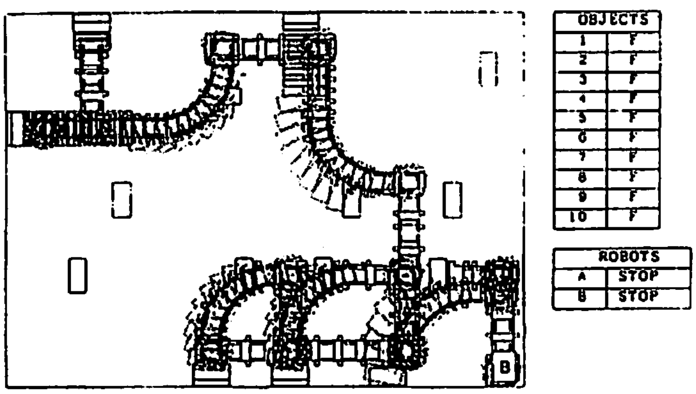

Fig. 7 (b). Simulated trajectory of robot B.

is planned as well as the target position of the object. The path must be planned without collision with respect to the positional information of unremoved objects.

(2) Planning of the initial position of a pushing robot

Based on the planned path for object transfer, the contact point with the object in the center of the face opposite to the transfer direction is determined as the initial position of the robot.

(3) Planning the path of the robot to the initial position

The path along which the robot moves from the current position to the initial position is planned. The path without collision must also be planned with respect to the positional information of unremoved objects.

5.3. Simulational Results

The simulation of an object pushing task was executed in the task environment as shown in Fig.5. Ten objects, 8 lightweight objects and 2 heavy objects, were used. Using two robots (robot A and robot B), the trajectory of each robot was displayed on each computer, as shown in Fig.7. Robot A removed object by independent processing, objects 4 and 8 by supporting robot B with cooperative processing, and then objects 3, 7, and 9 by independent processing. In addition, robot B removed object 4 by supporting robot A with cooperative processing, A, objects 5 and 10 by independent processing, object 8 by supporting robot A, and finally objects 2 and 6 by independent processing. The table in the upper right corner of Fig.7 lists the conditions of the objects manipulated by each robot. The table in the lower right corner lists the status of each robot. In the figure, the final conditions are shown.

These results indicate that the task assignment system enables each robot to autonomously determine task assignment and to remove all objects by parallel independent action or cooperative action. Furthermore, it was confirmed that the communication system accurately manages the situations of the objects along with the conditions of the robots.

6. Conclusion

A system has been developed in which task assignment is determined by communication among multiple autonomous robots. The system is capable not only of determining task assignment in the case of parallel and independent action by each robot, but also selecting a partner in the case that cooperative processing is required. In addition, a wireless communication system was developed which is necessary for the management of processing and robot conditions as well as for the organization of cooperative action. Finally, the system was applied to an object pushing task, and its functionality was confirmed by simulation.

Future research topics include the guarantee of a communication function when the number of agents increase, the realization of a function for dynamic collision avoidance, and the implementation of the system on actual robots.

References:

1) H. Asama, A. Matsumoto and Y. Ishida, "Design of an Autonomous and Distributed Robot System: ACTRESS," Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, pp.283-290, 1989.
2) F.R. Noreils, "Integrating Multi-Robot Coordination in a Mobile Robot Control System", Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, pp.43-49, 1989.
3) C. Le Pape, "A Combination of Decentralized and Distributed Methods for Multiagent Planning and Scheduling", Proc. IEEE Int. Conf. on Robotics and Automation, pp.488-493, 1990.
4) S. Premvuti and S. Yuta, "Consideration on the Cooperation of Mutiple Autonomous Mobile Robots," Proc. IEEE Int. Workshop on Intelligent Robots and Systems, pp.59-63, 1990.
5) R.G. Smith and R. Davis, "Frameworks for Cooperation in Distributed Problem Solving," IEEE Trans. on SMC, Vol.11, No.1, pp.61-70, 1981.
6) C. Hewitt, P. Bishop and R. Steiger, "A Universal Modular ACTOR Formalism for Artificial Intelligence," Proc. Int. Joint Conf. on Artificial Intelligence, pp.235-245, 1973.
7) A. Matsumoto, H. Asama, Y. Ishida, K. Ozaki and I. Endo, "Communication in the Autonomous and Decentralized Robot System ACTRESS," Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems '90, pp.835-840, 1990.
8) H. Asama, K. Ozaki, Y. Ishida, M.K. Habib, A. Matsumoto and I. Endo, "Negotiation between Multiple Mobile Robots and an Environment Manager," Proc. Int. Conf. on Advanced Robotics, pp.533-538, 1991.
9) H. Asama, M.K. Habib, K. Ozaki, Y. Ishida and I. Endo, "Functional Distribution among Multiple Mobile Robots in an Autonomous and Decentralized Robot System," Proc. IEEE Int. Conf. on Robotic and Automation, pp.1921-1926, 1991.

10) Y. Ishida, H. Asama, I. Endo, K. Ozaki and A. Matsumoto, "Communication and Cooperation in the Autonomous and Decentralized Robot System," Proc. IFAC Int. Symp. on Distributed Intelligence Systems, pp.299-304, 1991.

11) H. Asama, K. Ozaki, A. Matsumoto, Y. Ishida and I. Endo, "Collision Avoidance among Multiple Mobile Robots Based on Rules and Communication," Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems '91, pp.1215-1220, 1991.

Name:
Hajime ASAMA
Researcher, Ph D.

Affiliation:
The Institute of Physical and Chemical Research (RIKEN)
Chemical Engineering Lab.

Address:
2-1 Hirosawa, Wako, Saitama 351-01, Japan

Brief Biographical History:
1982 Entered The University of Tokyo (Graduate Course).
1986 Joined RIKEN (The Inst. of Physical and Chemical Research).

Main Works:
- "Development of an Expert system for Diagnosing Fermentation Processes", Kagaku Kogaku Ronbunshu, Vol. 17 NO. 3.
- "Dynamically Reconfigurable Robotic System (7th Report)", Trans. of the Japan Society of Mechanical Engineers(C), Vol. 57 No. 536.

Membership in Learned Societies:
- The Robotics Society of Japan (RSJ).
- The Japan Society of Precision Engineering.
- The Society of Chemical Engineers.

Name:
Akihiro MATSUMOTO
Assoc. Prof.

Affiliation:
Department of Mechanical Engineering, Toyo University

Address:
Kujirai-Nakanodai 2100, Kawagoe-shi, Saitama 350, Japan

Brief Biographical History:
1981-1983 Graduate School (Master Course), The University of Tokyo.
1983-1988 Research Associate, The University of Tokyo.
1988-1990 Assistant Professor, Toyo University.
1990- Associate Professor, Toyo University.

Main Works:
- "Communication in the Autonomous and Decentralized Robot System ACTRESS", Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '90), pp.835-840, Jul. 1990 .
- "SLIM and STROLIC: Standard Robot Languages in Japan", Proc. International Symposium on Industrial Robots (ISIR '91), pp.713-722, Oct. 1991.

Membership in Learned Societies:
- The Robotics Society of Japan (RSJ).
- Japan Society of Precision Engineering (JSPE).
- Japan Society of Mechanical Engineers (JSME).
- Institute of Electric and Electronic Engineers (IEEE).
- Japan Industrial Robot Association (JIRA).
- Japan Society of the Advancement of Automation (JSAA).

Name:
Koichi OZAKI

Affiliation:
Doctor Course Student, Saitama University

Address:
255, Shimo-ohkubo, Urawa, Saitama 338, Japan

Brief Biographical History:
1990 Graduated from Toyo University
1992 Doctor Course Student of Saitama University

Membership in Learned Societies:
- The Japan Society of Precision Engineering

Name:
Yoshiki ISHIDA

Affiliation:
Research Associate
Information Network System Operation Center, The University of Tokyo

Address:
2-11-6 Yayoi, Bunkyo-ku, Tokyo 113 Japan.

Brief Biographical History:
1986-1988 Graduate School (Master course), The University of Tokyo.
1988- Research Associate, The University of Tokyo.

Main Works:
- "Design of Communication System and Development of a Simulator for an Autonomous and Decentralized Robot System", Journal of the Robotics Society of Japan, Vol. 10, No. 4 (in printing).

Membership in Learned Societies:
- The Robotics Society of Japan(RSJ).
- Information Processing Society of Japan.
- The Institute of Electrical and Electronics Engineers.
- Internet Society, etc.

Name:.
Isao ENDO
Ph. D.

Affiliation:
Chief Engineer
The Institute of Physical and Chemical Research (RIKEN)

Address:
2-1 Hirosawa, Wako, Saitama 350-01, Japan

Brief Biographical History:
1965 B. Eng. Department of Chemical Engineering, The University of Tokyo.
1967 M. Eng. Department of Chemical Engineering, The University of Tokyo.
1970 Ph. D. Department of Chemical Engineering, The University of Tokyo.
1970 Joined Chemical Engineering Laboratory, RIKEN

Main Works:
- "Human Genome Analysis System", Nature, 352,89 (1991)
- "A Database System and an Expert System for Realizing Factory Automation", Bioproducts and Bioprocess, (1989)

Membership in Learned Societies:
- The Society of Chemical Engineering, Japan
- The Society of Fermentation Technology, Japan
- The Society of Enzyme Technology, Japan
- Japanese Society of Agricultural Chemistry
- American Institute of Chemical Engineering