

Design and Implementation of Multiple Autonomous Robots' Cooperative Behavior Based on Modest Cooperation

Shin'ichi YUTA and Suparerk PREMVUTI
Institute of Information Science and Electronics
University of Tsukuba
Tsukuba, Ibaraki, JAPAN 305

Abstract

In this paper, the authors present a concept on cooperation among autonomous mobile robots and describe how to adopt the concept to real robots.

At construction site, the use of autonomous mobile robots is very important. Robots are used for conveying materials or assembling. While constructing, several mobile robots have to share the same work space. For this reason, an architecture for cooperation of autonomous mobile robots must be considered. The concept of "modest cooperation" is proposed by the authors as a principle of cooperation behavior by multiple autonomous mobile robots.

The implementation and experiment of the cooperative behavior of real robots based on proposed concept is presented in this paper.

1 INTRODUCTION

Recently, many kinds of robots have been developed for construction site, which include not only the manipulator system but also the mobile robots for example a floor finishing robot. It is very important in automation in construction to give the mobile robots the ability of working autonomously. However, it's not easy because the environment of the robot in construction is usually very complicated and unstructured. So much more research on autonomous mobile robot are necessary to make the robots work in real construction environment.

One of the key subject for research of autonomous mobile robot is a cooperation of multiple robots. Because, there are many tasks of robot in the construction site, and large number of different types of robots or autonomous machine must work concurrently in the same area.

The problem of the multiple autonomous robot system is interested by several researchers, recently. Asama et.al. proposed a functional distributed multiple robot system and made an experimental system named ACTRESS [1]. To construct the system, in which multiple robots work cooperatively, the consideration on the architecture or basic structure is important. So, the authors proposed the concept of modest cooperation and the method to realize the cooperative behavior by combining the autonomous and centralized decision making [2]. The authors developed a experimental system with multi autonomous robots, to achieving the cooperation. In this paper, the design and implementation of multiple autonomous robots is described after the brief explanation on the basic concept for cooperative behavior.

2 BASIC CONCEPT ON THE AUTONOMOUS ROBOT'S COOPERATIVE BEHAVIOR

2.1 Classifying a Multiple robot system and its decision making mechanism

2.1.1 Classifying a Multiple Autonomous Robot System

Considering an autonomous multiple robot society. The state of the society will be divided into:

(1) There is common objective or mission to be performed, and the robots work toward their purpose together—A Common Objective System.

(2) Each robot has its own objective or mission, but there is some interference between robots during the execution of missions—An Individual Objective System.

The decision making mechanism is a key issue to solving the problem of organising a multiple robot system.

2.1.2 Multiple robots system provided with a common objective

The system is provided with an objective to be achieved. In this case, robots should share their duties and cooperate synchronously.

An example of this kind of cooperation is an assembly task done by two manipulators. In order to get an optimum solution of the task, the following important things need to be considered:

(1) Planning the jobs to be done and deciding the roles for each robot.

(2) Synchronization of the planned jobs.

In order to realize a specified task, information about the whole system should be collected at a central location, and the motion of each robot is decided based on this information. Keeping all the control information in one place is the best way to optimize the performance of the whole system. Even if each robot has independent controllers, the decision making of the whole system should be done centrally. The robots perform under instructions issued from the command center.

Suppose that each robot wants to decide its own movements independently. In order to make a proper decision, all information in the system must be supplied to all robots. And then, each robot's controller decides its motions while considering the optimization of the objectives of the whole system. This means, for example, all robots must solve the same path planning problem. However due to possible errors e.g. a message communication failure, one or more robots could make a plan not in accordance with the others. Thus, cooperation among the robots cannot be achieved.

This means, when the whole system has a common objective, the decision making mechanism should not be distributed to each robot, rather, it should be centralized.

In this kind of system, the integrated controller does not need to control the low level operations of a robot, such as servo motors.

2.1.3 A Multiple robot system in which each robot has a separate objective

Let's think of a world with many robots coexisting together, and each robot is provided with a separate task.

The cooperation of a robot in this case is quite passive. This type of cooperative behavior can be performed autonomously and thought as a primitive action of cooperation, according to the principle of "not to disturb each other". The authors named "modest cooperation" for this type of cooperative behavior. Only this case is considered as the scenario of a multi robot system with separate objectives. In this case, each robot knows about the existence of other robots and uses its "eyes" to see what the others are doing. Each robot can not know all the tasks which the other robots are going to do, but only the tasks to be performed by itself.

This type of system can be handled like this; all tasks are integrated together and then evaluation functions are defined. The solution given by a centralized method specifies each robot's task. However, since each robot has its own objective, integrating all the controls and parameters into one place is too complicated.

In our human society, which resembles the discussed model, each person does not require a global system efficiency. Instead, each person optimises his or her work. Similarly, multi-robot systems of this type are expected to introduce a cooperative mechanism to achieve tasks for all the autonomous robots in the system. The cooperative mechanism of distributed decision making could work efficiently in such a system.

If a robot decides its motion independently, and the motion is a service for another robot, autonomy of the robot is not ruined. In this situation, a robot can work autonomously, and at the same time cooperate with other robots.

In this case, each robot in the system has a different objective. Generally, each robot does not know the next action of the others. If a robot doesn't have any information about the next motion being done by the other robots, it is no use trying to cooperate with them. In order to make cooperation amongst robots possible. Each robot has to be able to examine what are they going to do. That is to say, the necessary ingredients in autonomous decision making is recognizing the type, position and actions of robots near by.

2.1.4 Coordinated Autonomous and Centerized Decision Making

In the previous sections, two types of systems were described. In one system all the robots had a common objective and in the other each robot had separate objectives. But in reality, multiple robot systems are not as simple as those already discussed. The whole system normally has a common objective but after task assignment each robot has its own objective. In this case, according to the objectives of the currently executing task, the type of cooperation can be classified dynamically into one of the two types described previously.

Considering the task objectives of the entire system hierarchically. For example, there are three levels(Fig. 1):

	ROBOT 1	ROBOT 2	ROBOT 3
LEVEL 1 Assigning Tasks	Cleaning Up the Room		
LEVEL 2 Deadlock Occurs	Floor Cleaning	Wall Cleaning	Ceiling Cleaning
LEVEL 3	Solving Deadlock State		

Fig. 1 Objective of operation of robots at each level

In this example, the first level is the task specifying level. This level assigns the task for each robot based on common objective. The second level is the robot objectives level. This level controls the robot according to autonomous objectives of each robot. The third level solves the deadlock state.

For example, consider a system with three robots, whose mission is to clean up a room. The delegation of work to each robot should be done by a centralized decision making process. As a result of this decision making each robot is given one of the following jobs; clean the floor, the wall, or the ceiling. When the roles of the robots are fixed, it means that each robot works to achieve independent

and separate objectives. At this level, the robots start doing their jobs of cleaning up the room. While moving around the room, the robots must avoid collisions with each other. Collision avoidance may be considered as a separate objective of cooperation among the robots and done autonomously by each robot. But, when the two cleaning robots go to positions that interfere with each others planned paths, a deadlock occurs. If the two robots don't shunt mutually, they cannot move any further. When the two robots are in this state, only these two robots are responsible for solving the deadlock problem. This problem can be considered as their common objective. The deadlock problem can be solved by one of the two robots.

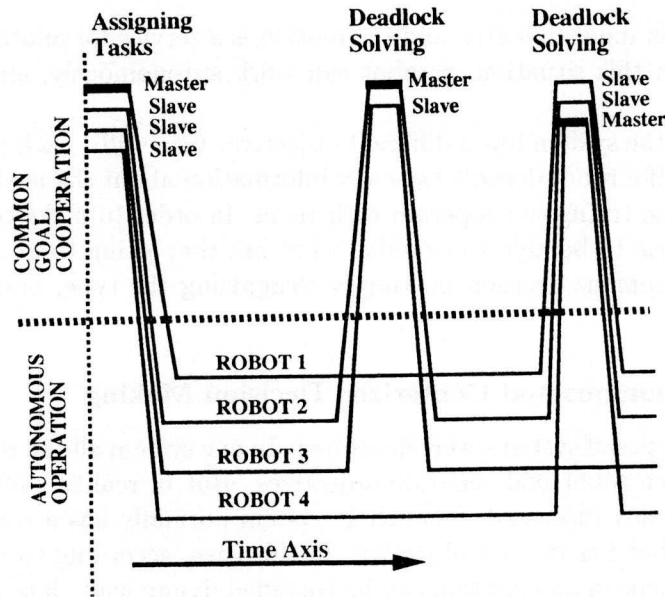


Fig.2 Timing chart shows switching between common goal cooperation and autonomous operation

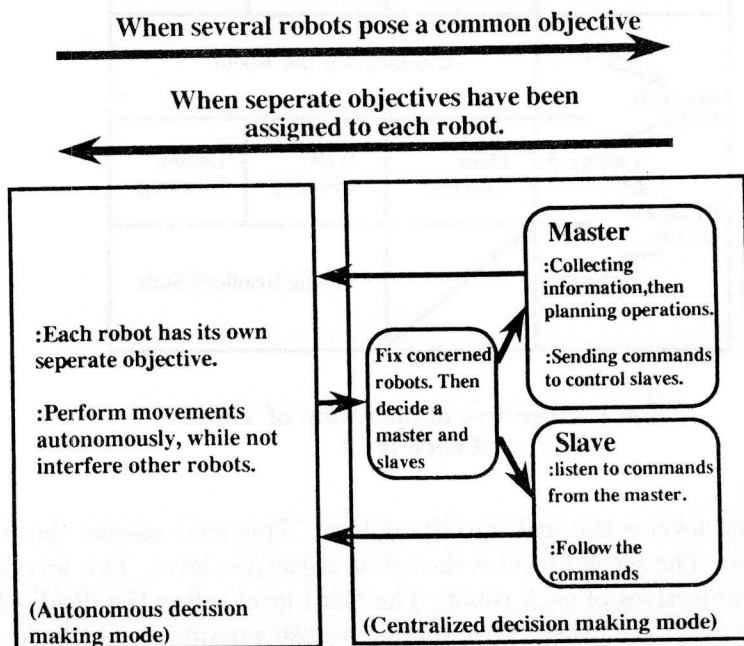


Fig.3 Control modes of a robot with cooperation capability

In the above example at level 1, decision making is done centrally to allocate the cleaning jobs to the individual robots. In level 2, each robot does its own decision making to clean part of the room, i.e. decision making is distributed. In level 3 to solve the deadlock situation, centralized decision making is necessary.

It can be seen from this example, that decision making continually switched between centralised and distributed modes.

Most of the time, robots operate autonomously. When they have common objective, e.g. a deadlock solving, the robots operate under centralized control. Once the dead lock is solved, every robot back to operating autonomously. An example timing chart showing the switching between common goal cooperation and autonomous operation, is shown in Fig. 2.

From the point of view of each robot, the control mode is switched between autonomous decision making and centralized decision making. In the centralized mode, a robot is a leader or a slave (Fig. 3).

The method to select the leader between concerned robots is an interesting problem. However, this issue is beyond the scope on this paper. Any simple function can select the leader.

3 AN EXAMPLE OF THE COOPERATIVE BEHAVIOR OF AUTONOMOUS ROBOTS — PROBLEM DESCRIPTION

The most important resource for mobile robots is the space occupied by the robot itself. As an typical example of the modest cooperation and distributed resource management, several mobile robots are considered to work on the roadway network. Here, each robot shares the roadway together as a common resources, and is expected to run on its independent course, to achieved his own objectives.

The environment is the roadway network, which is assumed as follows;

- (1) The every sections of roadway is straight and the robot cannot pass each other on it.
- (2) There is only three-fork type intersections i.e. the crossroad is not exist. This assumption makes it possible to treat the roadway sections as the units of resource.
- (3) The read can be recognized by the robot's sensor. So the robot must be added to run following to the roadway.

4 DESIGN OF ROBOT BEHAVIOR AND SIMULATION

To solve the problem given in the previous section. The each section of roadway between branches is managed as a resource, and the robot enter the section only when the roadway section is vacant. The vacancy of the section is realized by the robot using communication. So, advoiding the conflict caused by the communication delay, the each robot must declare not only the currently using resource but also a resource which is starting to be used. A deadlock can occur when two or more robots want to move into the section which the other robot exist. To solve the deadlock problem, shunting algorithm must be designed.

The authors simulated the motion of 7 robots operating in the same environment on a workstation(Fig.4). Robots move along their specified routes. Our algorithm states that robots must reserve a section (a roadway between two cross roads) before entering. This is done by issuing a request over the communication network.

For the case of deadlock occurred, a shunting algorithm was implemented. The simulator shows the shunting process for two and three robots.(Fig.5).

Shunting can be thought as a common goal for a group of robots for a small period of time. One robot involved in the deadlock is selected as the leader. This robot finds a solution by shunting and

sends instructions to the other robots. On the simulator's screen, the leader robot is displayed in a different way from a robot in the ordinary state.

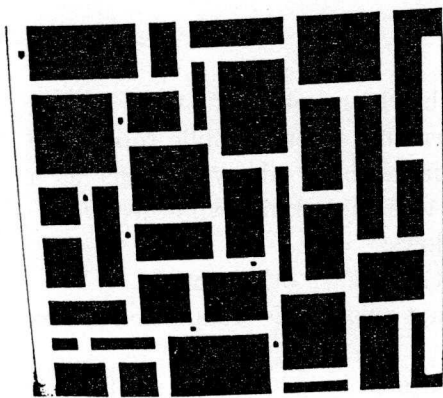
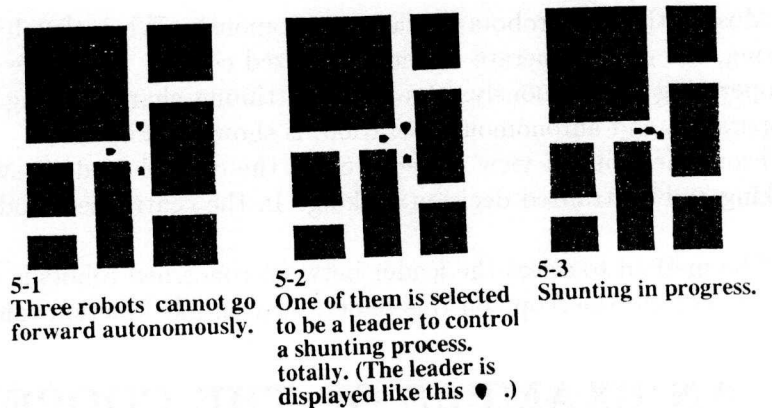


Fig.4 Simulation screen shows 7 robots moving in roadway network environment

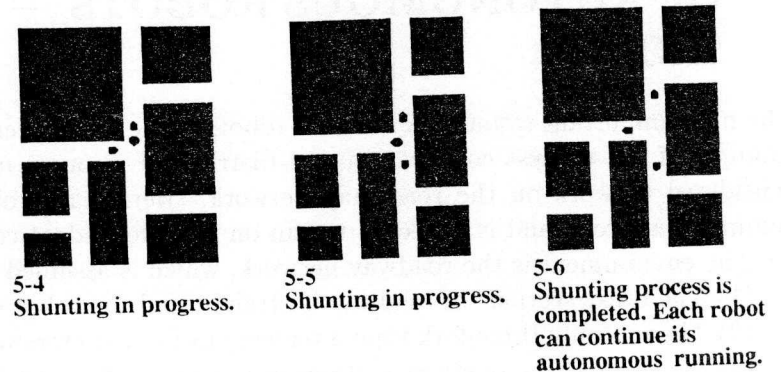


Fig.5 Shunting process of 3 robots (Deadlock solving)

5 IMPLEMENTATION OF THE ALGORITHM ON REAL ROBOTS AND THE RESULTING BEHAVIOR

The algorithm has been implemented on the Yamabico family of autonomous mobile robots [3].

5.1 About the robots and their environments

The control system of each "Yamabico" robot consists of several functional modules (Fig.6). Each module is responsible for a specific function of the robot. The main modules are as follows:

(1) Master module

This module executes a behavior program for a robot. While executing the program, this module passes execution commands to other specified modules in the Yamabico architecture.

(2) Locomotion module

This module executes locomotion functions sent from the master module. Basically, the module controls two powered wheels to move along a specified locus.

(3) Ultrasonic range finder module

This module measures the distance between the robot and obstacles in four directions; front, back, left and right.

(4) Communication network module[4]

This module supports broadcast communication and one to one communication. Broadcast communication is used for periodically sending information about the current state of the robot, for example, the current position of the robot. One to one communication is used between a leader and a slave while shunting in order to solve the deadlock problem.

The environment for robots is straight paths which form a network. The width of each path is 90 centimetres with both sides bounded by sound reflectors. Only three forked branches at intersections are allowed in the path network. A map of the environment is supplied to the robots prior to the commencement of operations. While the robots are moving along the paths, unpredicted obstacles such as pedestrians may pass in front of the robots at anytime.

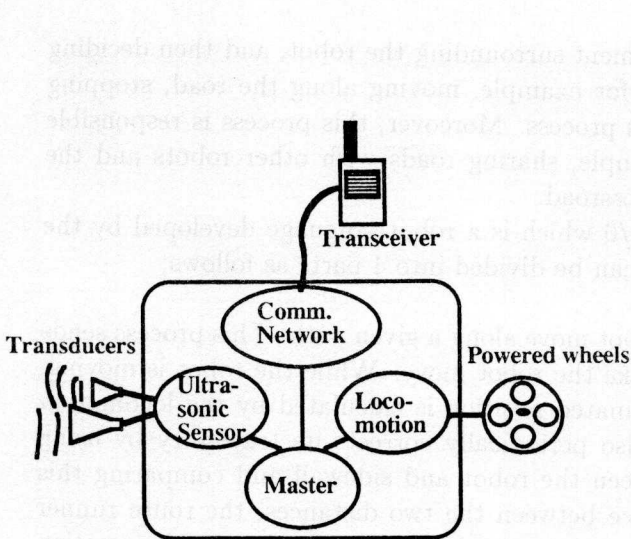


Fig.6 Functional modules structure of a "Yamabico" robot

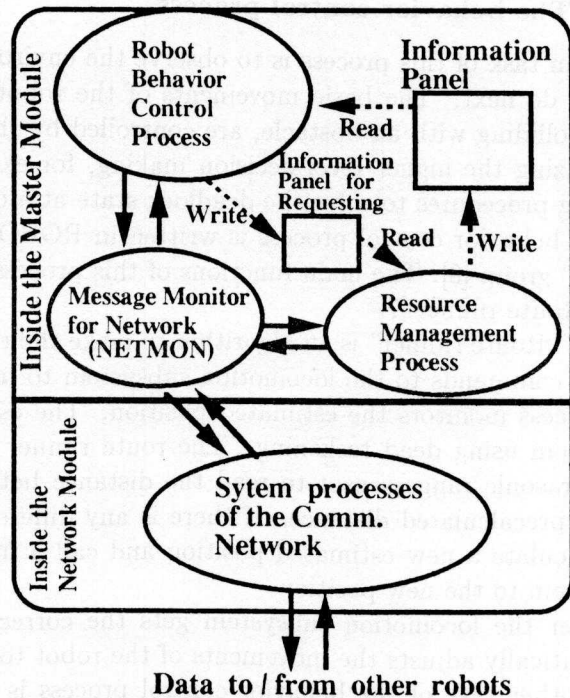


Fig.7 Show main processes concerning cooperative behavior and their data flows

5.2 Software for realizing the cooperation algorithm

On the master module of a Yamabico, besides the system processes, there are processes for controlling cooperative behaviors. The important processes are the "behavior control process" and the "resource management process" (Fig.7).

5.2.1 How to manage resources

A robot which performs its task in a "modest cooperative" way, must know about the state of shared resources. An state information monitoring panel [5] is used to provide this information. Each robot broadcasts information about its status to all other robots while it is moving. At the same time, each robot receives information broadcasts from the other robots. This information is put into the information panel.

Management of the information panel is done by a process called the "resource management process". The main tasks of this process are to update information about resources sent from other robots, and to try to secure resources requested by the "behavior control process".

The success or failure of responding to a request by the behavior control process is shown on the informational panel. In short, the resource management process is responsible for updating and managing the informational panel and the behavior control process uses the information stored in the informational panel.

5.2.2 The behavior control process

The main task of this process is to observe the environment surrounding the robot, and then deciding what to do next. The basic movements of the robot, for example, moving along the road, stopping before colliding with an obstacle, are controlled by this process. Moreover, this process is responsible for realizing the higher level decision making, for example, sharing roads with other robots and the shunting procedures to solve the deadlock state at a crossroad.

The behavior control process is written in ROBOL/0 which is a robot language developed by the authors' group [6]. The main functions of this process can be divided into 4 parts as follows;

(1) Route runner:[7]

The "Route runner" is an algorithm to make the robot move along a given path. This process sends motion commands to the locomotion subsystem to make the robot move. While the robot is moving, the process monitors the estimated position. The estimated position is calculated by the locomotion subsystem using dead reckoning. The route runner also periodically corrects its trajectory by using the ultrasonic range sensor to read the distance between the robot and sidewall and comparing this with a precalculated distance. If there is any difference between the two distances, the route runner will calculate a new estimated position and call a function to set the internal position of locomotion subsystem to the new position.

After the locomotion subsystem gets the correct position from the behavior control process, it automatically adjusts the movements of the robot to maintain the correct trajectory.

Another task of the behavior control process is to avoid collisions with obstacles in front of the robot. The robot avoids colliding with pedestrians by using its front ultrasonic sensor. When the distance between the robot and an obstacle is less than a specified value the process will call a function to stop the robot.

(2) Sharing resources by using the principle of modest cooperation:

The path for a robot consists of several unit straight roads which are called "sections" connected in series. A section is considered as a unit of resource for movements of the robot. Before a robot enters the next section, it must secure the section by requesting the section from the resource management process. The section can be secured for use if and only if the resource is not occupied by any other robot.

The behavior control process requests a section by writing the request on the information panel. The behavior control process checks the status of the required section on the information panel and decides whether the robot can enter the section or not. When the section is not or not going to be used, the process requests a section by writing the name of the required section on the information panel. The robot enters the next section only when the information panel shows that the section is not secured by others.

After the robot enters the next section, the behavior control process will try to secure the following section it expects to enter. If a section cannot be secured, the robot will stop.

(3) Deadlock detection:

While stopping at a crossroad the behavior control process will check the information panel to see if the following section is available or not and check if a deadlock state has occurred. The process will examine the number of robots which have stopped at the crossroad and the directions which each robot wants to go. From this information the process can know whether a deadlock has occurred or not.

(4) Solving the deadlock problem using shunting:

When the behavior control process detects a deadlock state at a crossroad. The process will try to gain leadership status. The leadership status is also considered like a resource. The result of the request, be it leader or be it slave, is shown on the information panel. The robot whose behavior control process gains leadership status must decide a shunting plan and control movements of the other slave robots. After shunting is finished, the leader robot will declare through the broadcast network that its leadership is over. Then, the robots at the crossroad return to their original states and can continue moving autonomously again.

5.3 Experiment

The environment for three autonomous mobile robot "Yamabico" is a network of roads shown in Fig.8. Each robot was assigned a course to move along repeatedly. Some parts of the network were shared between several robots.

Once the three robots started moving, they travelled along the assigned course successfully (Photo 1). The robots shared the roads by waiting at crossroads. Sometimes a deadlock state occurred, but the concerned robots performed shunting to solve the problem successfully.

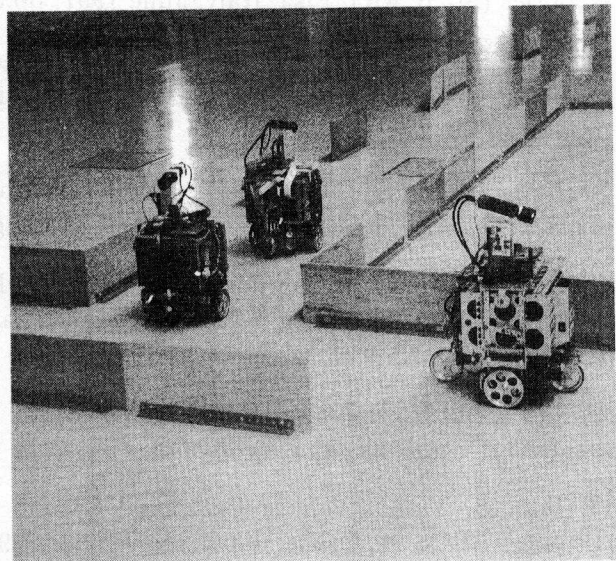
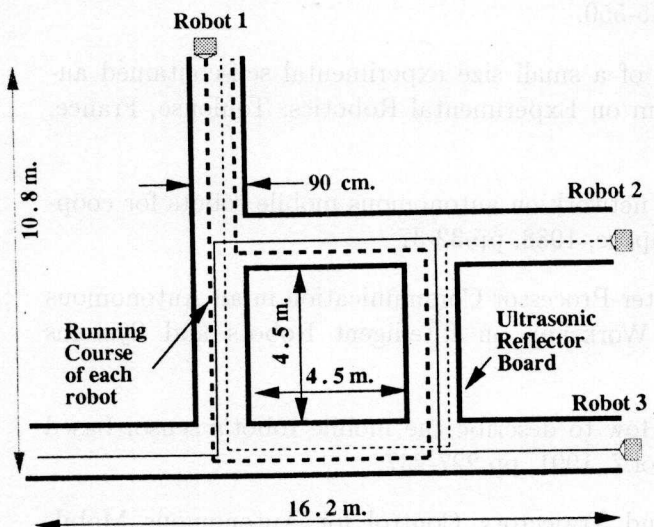


Fig. 8 Environment and running course of each robot

Photo 1 Show three robots running in the same environment

6 CONCLUSION

This paper considered a robot world with several self-contained autonomous robots gathered in a small space. Cooperation of the robots has been examined. When several robots work closely to each other, the most fundamental cooperation between them is a collision avoidance. To achieve this type of cooperation, the authors proposed the principle of modest cooperation.

In this paper, it was assumed that robots in a system had the same functions and abilities, and the robots moved with the same control algorithms. In the human society model, this assumption can be effective. In a more generalized model which consists of various type of construction robots, the problems of cooperation will become more complicated. Problems concerning the cooperation of robots with different control algorithms is an attractive research subject which needs attention in the future.

ACKNOWLEDGMENTS

The Yamabico Project is running with strong support from its members who are mainly located at the Intelligent Robot Laboratory at University of Tsukuba. Dr. Jun'ichi Iijima of Meisei University is a key person on the Yamabico project. Dr. Shigeki Iida and Mr. Shooji Suzuki have contributed greatly to the route runner algorithm and the programming environment. Mr. Yasunori Abe of Shinryo-Reinetsu Co.Ltd. helped to provide the experimental site. Dr. Alex Zelinsky of the University of Wollongong, Australia gave valuable suggestions in preparing and proof reading this paper. The authors thank the above mentioned people for their contributions.

References

- [1] H. Asama, M.K. Habib, I. Endo, K. Ozaki, A. Matsumoto, Y. Ishida, "Functional distribution among multiple mobile robots in an autonomous and decentralized robot system", Proceedings 1991 IEEE International Conference on Robotics and Automation, California, April 1991, pp.1921-1926.
- [2] S. Yuta and S. Premvuti, "Consideration on Cooperation of Multiple Autonomous Mobile Robots — Introduction to Modest Cooperation —", Proceedings Fifth International Conference on Advanced Robotics ('91 ICAR), Pisa, Italy, June 1991, pp.545-550.
- [3] S. Yuta, S. Suzuki and S. Iida, "Implementation of a small size experimental self-contained autonomous robot", Second International Symposium on Experimental Robotics, Toulouse, France, June 1991.
- [4] S. Premvuti and S. Yuta, "Radio communication network on autonomous mobile robots for cooperative motions", Proceedings of IECON'88, Singapore, 1988, pp.32-37.
- [5] S. Yuta, J. Iijima, "State Information Panel for Inter-Processor Communication in an Autonomous Mobile Robot Controller", IEEE International Workshop on Intelligent Robots and Systems (IROS)'90, Tsuchiura, Japan, June 1991, 1516.
- [6] S. Suzuki, M.K. Habib, J. Iijima and S. Yuta, "How to describe the mobile robot's sensor-based behavior?", *Robotics and Autonomous Systems*, Vol.7, 1991, pp.227-237.
- [7] S. Iida, S. Yuta, "Vehicle Command System and Trajectory Control for Autonomous Mobile Robots", IROS'91, Osaka, Japan, November 1991, pp.212-217.