A Design Method for Decentralized Control System applying System Connection

Toshiaki Tsuji*, Kouhei Ohnishi*
*Department of System Design Engineering, Keio University, Yokohama, JAPAN

***Abstract*—Decentralized control is a promising method for large scale systems. A critical design method for decentralized control systems is desirable for solving the complicated design problem. Simple and explicit controller design becomes available with the idea of "function." However, the hierarchy based on the network structure may impose restrictions on controller design. At the same time, various kinds of exceptions such as faults, performance limits and so on may also impose restrictions. This paper proposes a method of the controller design under these restrictions. The concept of system connection is applied to derive the hierarchy. An exception is treated the same as a communication blackout. The hierarchy structure is intentionally manipulated to decide the priority order of the functions. The unified design for the system with the hierarchical structure and exceptions can be realized through this method.**

I. INTRODUCTION

Decentralized control is a promising method for large scale systems. It is superior in many features such as flexibility, fault tolerance, expandability, rapid response and so on. Many studies have been performed on concepts and theories of decentralized control systems[1], [2]. Connection matrix is a useful tool to know the hierarchical structure of decentralized systems[3], [4]. Akuzawa and Ohnishi adopted the maximum eigenvalue of the connection matrix and its eigenvector as the design indices[5].

The idea of decentralized control has been also applied to many robot control systems[6], [7], [8], [9], [10]. Among them, Fukuda et. al. proposed the control method based on cell structure[6]. Decentralized control systems have been utilized for many fault tolerant systems[10].

We have shown a new design procedure of a decentralized control system based on the idea of "function"[13]. This idea is also applied and expanded for a bilateral control system with position/force scaling[14]. The design becomes simple and explicit since this idea clearly relates the system role to the controller structure. It was assumed in the method that communication between subsystems is complete. The decentralized system with incomplete communication has restrictions on its controller design because some of the commands may not reach to the other subsystems. By the same token, the decentralized system with a hierarchical structure has restrictions on its controller design. Furthermore, exceptions such as faults and performance limits also impose restrictions. This paper describes a way to design the decentralized control system under these restrictions. We apply connection matrix[3] to figure out the command flow of the decentralized system. A unified design method for a decentralized system under the hierarchical structure and exceptions is proposed.

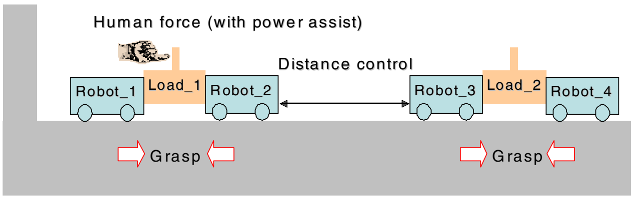


Fig. 1. Supposed system

II. SUPPOSED SYSTEM

This paper aims at applying the concept of system connection to the design of a decentralized control system. The following discussion is based on a supposed system shown in Fig. 1. The system consists of four robots active in 1 dimensional space. They are named robot_1, robot_2, robot_3 and robot_4 respectively. The method in this study is applicable for much complicated system in essence. We however verify it with a simple and fundamental system.

Two robots make a pair and each pair grasps a load. The pair of robot_1 and robot_2 grasps load_1 while the pair of robot_3 and robot_4 grasps load_2. The distance between the two pairs is held constant by control. An operator gives the manipulation force to the load or the robot. Operator force is detected by reaction force observer(RFOB)[12] and assisted so that the operator feels the load lighter.

III. CONCEPT AND EXPRESSION OF SYSTEM CONNECTION

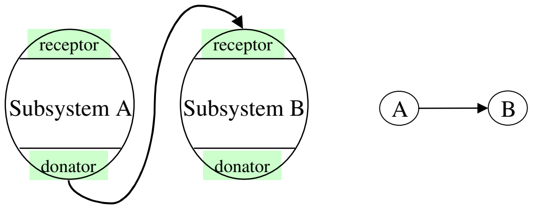
System connection is a concept for a network system to express the information flow. This section describes the expression of system connection as a preparation for this study[3], [4], [5].

A. Information Representation in Connected System

The internal connection among subsystems is considered so as to observe the entire system. Here, a subsystem stands for an individual robot in this study. We apply graph theory to express the informational flow. Fig. 2(a) shows the informational flow from subsystem A to subsystem B and Fig. 2(b) shows the equivalent representation.

A connection matrix $\boldsymbol{C}$ is defined against the decentralized system with several subsystems. c_{ij}, (i, j) element of $\boldsymbol{C}$, is defined as follows:

$$c_{ij} = \begin{cases} 1, & \text{if there is connection from} \\ & j\text{th subsystem to } i\text{th subsystem} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$



(a) (b)

Fig. 2. Equivalent expression of system connection

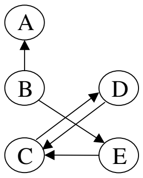


Fig. 3. Example of informationally connected system

The connection matrix for the example in Fig. 3 is represented in Boolean algebra as shown in (2).

$$\boldsymbol{C} = \begin{array}{c} \\ A_R \\ B_R \\ C_R \\ D_R \\ E_R \end{array} \begin{array}{c} \begin{array}{ccccc} A_D & B_D & C_D & D_D & E_D \end{array} \\ \left[\begin{array}{ccccc} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{array}\right] \end{array}. \qquad (2)$$

The subscripts D and R denote a donator and a receptor respectively. As shown in the example, matrix $\boldsymbol{C}$ describes the interaction among whole subsystems.

B. Reachability of Connected System

The connected system which forms a hierarchical structure is expressed using a reachability matrix.

The element c_{ij}^k in $\boldsymbol{C}^k$ represents the direct path from j to i with the length k. The sum of the power of the connection matrix $\boldsymbol{C}^1, \boldsymbol{C}^2, \boldsymbol{C}^3, \cdots, \boldsymbol{C}^k$ yields a reachability matrix $\boldsymbol{M}_R$.

$$\boldsymbol{M}_R = \sum_{k=1}^{n-1} \boldsymbol{C}^k + \boldsymbol{I} \qquad (3)$$

The reachability matrix represents the direct or indirect connections among all subsystems. The elements of it show whether information of subsystem j reaches to subsystem i. The example of a reachability matrix in Fig. 3 is calculated as follows:

$$\boldsymbol{M}_R = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \left[\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right] \end{array}. \qquad (4)$$

From this example, it is shown that information from subsystem E can reach subsystem C and D.

C. Hierarchical Structure of Connected System

Hierarchical information of the connected system is derived from the reachability matrix. The subsystem which sends information comes to the higher layer in a uniflow system. In the reachability matrix $\boldsymbol{M}_R$, the elements of the column represent reachability from the related subsystem to other subsystems. On the other hand, the elements of the row represent the reachability from other subsystems to the related subsystem. To derive hierarchical information, the elements of the reachability matrix is permutated in descending order based on the sum of column elements. The permutation matrix $\boldsymbol{P}$ in the example of Fig. 3 is shown in (5) and rows and columns are permutated by (6).

$$\begin{aligned} \boldsymbol{P} &= \left[\begin{array}{ccccc} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{array}\right] && (5) \\ \boldsymbol{M}'_R &= \boldsymbol{P}\boldsymbol{M}_R\boldsymbol{P}^T && (6) \end{aligned}$$

Then the replaced reachability matrix $\boldsymbol{M}'_R$ is obtained as follows:

$$\boldsymbol{M}'_R = \begin{array}{c} \\ B \\ E \\ C \\ D \\ A \end{array} \begin{array}{c} \begin{array}{ccccc} B & E & C & D & A \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}. \qquad (7)$$

Equation (7) shows that subsystem B sends its information to all of the other subsystems. In other words, B is on the highest layer. At the same time, all elements are 1 in the submatrix formed by the row C, D and the column C, D. It shows subsystem C and D send their information to each other, hence they exist in the same layer. The part where some subsystems exchange their information mutually is defined as one layer. The hierarchical structure is mathematically derived with the connection matrix in this manner.

IV. FUNCTION-BASED CONTROLLER DESIGN

Robots often need to execute multiple actions in parallel as the operation becomes complicated. For example, robots have to move after they grasp a load to achieve a conveying operation. In short, the robots have to “move” and “grasp” at the same time. Robots may be urged to execute and switch a wide variety of actions especially in a large scale system. However in previous design methods, these actions were not clearly related to the controllers. We have therefore proposed a method to express these actions by the idea of “function.” The controller design becomes explicit since this idea relates the individual actions to controllers. The following section briefly describes the method of function-based controller design[13].

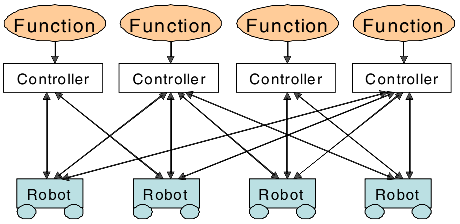


Fig. 4. Relation between controllers and robots

A. *Definition of Function*

At first, the word “system role” is defined as follows:

[*Definition 1*] “System role” is a description on the requirement from the user to the robot control system.

The designer has to decide the structure of the controller so as to satisfy the system role. It is more convenient to divide the system role into independent features since a system role is an abstract expression. These independent features are called “function.” In other words, function is a unit to express the desired actions of the robot. The idea of “function” is defined as follows[14]:

[*Definition 2*] “function” is the minimum component of a system role for a control system. Conversely, the system role is described as a combination of functions.

The examples of functions are shown in Table. I.

In this study, controllers of the decentralized system are designed based on functions. This approach is different from other conventional methods in the respect that the controllers are not directly associated with individual robots as shown in Fig. 4. Here, multiple controllers give inputs to multiple robots. A method to associate controllers with robots is indispensable to solve the complicated relationship. Hence coordinate transformation is applied. Robot coordinates are transformed into function coordinates, new coordinates based on functions. Function-based controllers are designed on this coordinate system.

B. *Coordinate Transformation*

TABLE I

FUNCTIONS FOR SUPPOSED SYSTEM

Type	Role of function	Based information
grasp	apply grasp force to a load	difference of two robots
coupling	control distance of two subsystems	difference of two robots
friction compensation	compensate friction of entire system	sum of robots
inertia manipulation	assist manipulation force from human	sum of robots
torque limit	limit excessive input	single data
velocity limit	slow down overspeed actuator	single data
position limit	avoid collisions	single data or difference of two robots

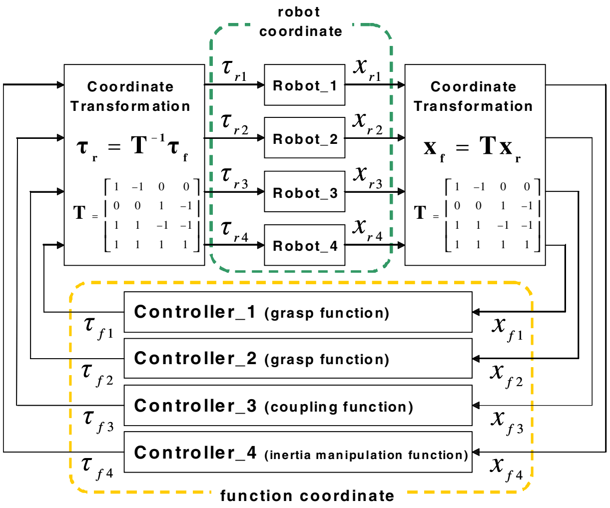


Fig. 5. Overview of control system

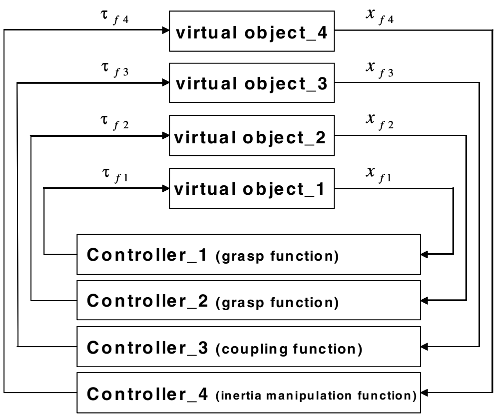


Fig. 6. Overview of equivalent system

The overview of the control system is shown in Fig. 5. The robot coordinates are transformed into function coordinates by a transformation matrix $\boldsymbol{T}$. If the respective rows in $\boldsymbol{T}$ are independent of each other, each motion realized by the function-based controller is also independent. Therefore it becomes possible to control as if respective controllers are connected to decoupled virtual objects as shown in Fig. 6. The behavior of robots realized by each function-based controller is named “function mode.” A function mode is the behavior of the virtual object in the function coordinate. On the other hand, the behavior of the entire system in robot coordinate space shows up as superposition of function modes.

The coordinate transformation from robot coordinates to function coordinates is shown in (8).

$$\begin{aligned} \boldsymbol{x}_f &= \boldsymbol{T}\boldsymbol{x}_r \\ \boldsymbol{x}_f &= \begin{bmatrix} x_{f1} & x_{f2} & \cdots & x_{fN} \end{bmatrix}^T \\ \boldsymbol{x}_r &= \begin{bmatrix} x_{r1} & x_{r2} & \cdots & x_{rM} \end{bmatrix}^T \end{aligned} \tag{8}$$

where x_f shows the position in function coordinate. Subscript denote the function number. N is the total number of mounted functions. x_r shows the position of the robot. Numbers in subscript denote the robot number. M is the total number of robots. $\boldsymbol{T}$ shows the transformation matrix.

Velocity, acceleration, input force $\boldsymbol{\tau}$ and external force $\boldsymbol{f}$ are all transformed in the same way.

$$\dot{\boldsymbol{x}}_f = \boldsymbol{T}\dot{\boldsymbol{x}}_r \tag{9}$$

$$\ddot{\boldsymbol{x}}_f = \boldsymbol{T}\ddot{\boldsymbol{x}}_r \tag{10}$$

$$\boldsymbol{\tau}_f = \boldsymbol{T}\boldsymbol{\tau}_r \tag{11}$$

$$\boldsymbol{f}_f = \boldsymbol{T}\boldsymbol{f}_r \tag{12}$$

C. Function Priority

The control system should automatically choose the functions to replace and the functions to execute when excessive functions exist. Hence priority order is introduced to functions.

The priority order is based on the following terms.

1) Mechanical limit
2) Safety
3) Importance of the task

Functions to reply to mechanical limits are the most important since mechanical limits are the absolute condition of a control system. Most of the fault redress functions such as torque limit function and velocity limit function belong to this category. Functions to assure the safety comes the next because safety is the priority for users. Grasp function also has a high priority since it may be dangerous to drop the grasped load in some situations. Except these, function priority is given manually based on the importance of the task.

V. DESIGN METHOD APPLYING SYSTEM CONNECTION

The hierarchical structure imposes restrictions on the system role. At the same time, the entire system is also restricted when communication is uncertain. Furthermore, various kinds of exceptions may restrict the operation task. We propose a design method to decide the functions under these restrictions. The concept of system connection is applied to derive the layer structure of the decentralized control system.

This section describes how to utilize the system connection for deciding the function order. Firstly, an example of manipulating the system connection intentionally is shown. Secondly, a unified method to decide the function order is shown. Finally, conditions for the function is introduced.

A. Manipulation of System Connection

A unilateral connection between subsystems frames the hierarchy. When the system connection alters, the hierarchical structure may also change. For example, Fig. 7 shows the alteration of the hierarchical structure when the communication from robot_3 to robot_2 is disconnected.

The flow of system connection could be decided arbitrarily if the system has reliable communication. Then the hierarchical structure is also arbitrarily decided. The restriction by this manipulated hierarchy could be positively applied to the design of the function order.

As shown in Fig. 8, the subsystem will close its receptor when any exceptions occur on it. The subsystem comes to the upper layer if the receptor is closed. Then the exception handling function on the subsystem acquires the higher priority. In sum, subsystem under exceptions can autonomously give up its cooperative activity and handle the exception in the first priority. Furthermore, the entire system can select the functions to carry on from the function order. Through this method, it may become possible for the decentralized system to autonomously redesign its controller and its hierarchical structure when any exceptions occur.

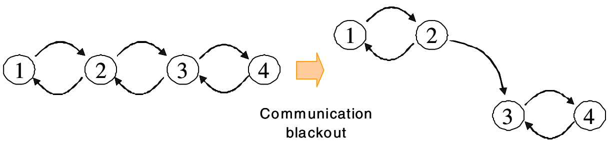


Fig. 7. Hierarchical structure due to communication

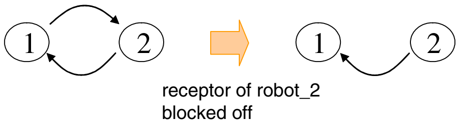


Fig. 8. Receptor block off in case of exceptions

B. Function Order based on Hierarchy

The hierarchy makes restrictions on function planning regardless if it is imposed arbitrarily. Therefore, function-based controller design should be expanded to consider these restrictions. The procedure to decide the function order under the hierarchical structure is given below.

1) Derive hierarchical structure

Hierarchical structure is derived by a reachability matrix $\boldsymbol{M}'_R$ as shown in III-C. Every subsystem, every robot in other words, belongs to individual layers. Robot_1 and robot_2 belong to the first layer and robot_3 and robot_4 belong to the second layer in the example shown in Fig. 7.

2) Distribute functions to each layer

This step should be executed one by one on each layer in the hierarchical order. The functions based on the subsystems in the corresponding or higher layer would belong to the corresponding layer. The function would not belong to the layer if it is related to any subsystems in the lower layer. The function based on robot_1 or robot_2 belong to the first layer in the example of Fig. 7. On the other hand, the function based on robot_1, robot_2 and robot_3 would belong to the second layer as shown in Fig. 9.

3) Choose active functions in each layer

This step should also be executed on each layer in the hierarchical order. The DOF of each layer is the sum DOF of robots in the layer. Functions are practicable as long as the DOF of functions do not exceed the DOF of the layer. Active functions are selected in the

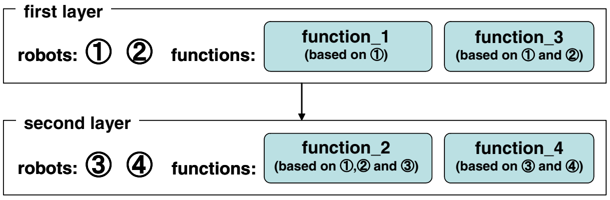


Fig. 9. Distribution of robots and functions to the layers

priority order in each layer. Priority order is given by the conventional way as shown in IV-C.

C. Conditions for Function

A function has conditions to be executed. The conditions are based on the state and the connection of the related robots. There are two types of conditions: the necessary condition and the arbitrary condition. The necessary condition is for diagnosis of function practicability. Each function has its inherent necessary conditions. For example, a grasp function has two necessary conditions. The output torque of related robots should be within the torque limits and information of one of the related robots should be reachable to all other related robots. The arbitrary condition is for a task shift. The designer sets conditions on the function so that the function works in a certain situation.

The function works when these conditions are satisfied. We define this state of functions as "active." On the other hand, function-based controllers do not give any inputs when the state or the connection is out of the condition. This state is named "inactive."

VI. SIMULATION

A simulation of the supposed system was executed to verify the validity of the proposed method. External force $F_{ext} = 0.35 \times \sin(t) + 0.06\ N$ was given to load_1. Fig. 5 shows the control system under no exceptions. The velocity limit of robot_2 was 2.0 m/s.

When the velocity of robot_2 went over the limit, the velocity limit function became active and robot_2 closed its receptor. The reachability matrix then became as follows:

$$\boldsymbol{M}'_R = \begin{array}{c} \\ 2 \\ 1 \\ 3 \\ 4 \end{array} \!\!\! \begin{array}{c} \begin{array}{cccc} 2 & 1 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}. \qquad (13)$$

Velocity limit function acquired the highest priority from the hierarchy order. Two grasp functions and coupling function took back seats. Inertia manipulation function gave way since velocity limit function became active.

Velocity responses on Fig. 10 show that velocity of robot_2 never overran the limit. Velocity of other robots were also repressed in tune with the robot_2 so as to achieve grasp and coupling functions. Force responses show that the grasp function was executed without any failure even when the velocity limit function became active. Small but rapid fluctuation on the force response occurred. It is due to the rapid input variation on the velocity limit function. It implies that rapid or extreme input variation may interfere with other functions in practice although functions are conceptually independent of each other. This influence is negligible if input variation is smooth. The function changeover therefore needs consideration for the boundary condition.

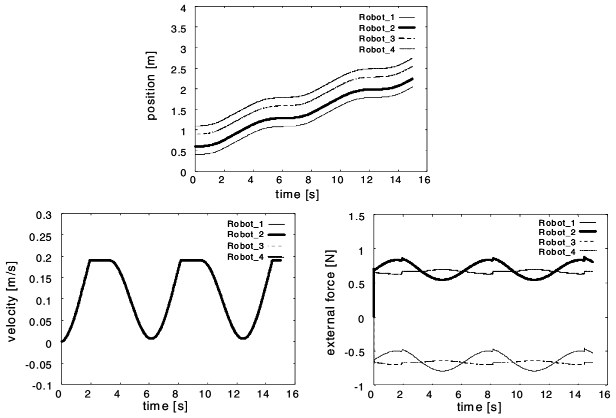


Fig. 10. When robot comes out of performance limit

VII. EXPERIMENT

We carried out an experiment on cooperative grasping with two robot manipulators shown in Fig. 11. Each manipulator has 1 DOF on a vertical rotation axis. Fig. 12 is an illustration of the experiment. Figs. 13 and 14 show the position response and the force response respectively. Sum and difference values of force response are shown since they correspond to functions directly. The sum value shows the human force and the difference value shows the grasping force. Vertical dashed lines in the figures denote the moment of function changeovers. Details are described as follows in the order of events:

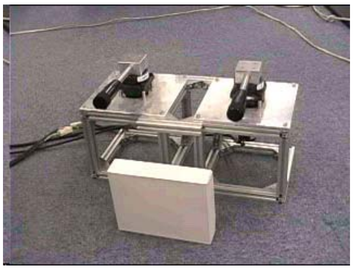


Fig. 11. Experimental system

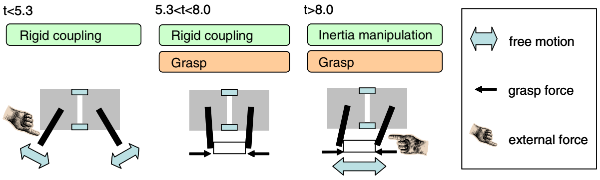


Fig. 12. Steps on experiment

1) $t < 5.3$

Only rigid coupling function is applied. The rigid coupling controller controls the robots so that the sum of position values becomes 0. Hence the motion of two robots becomes symmetric.

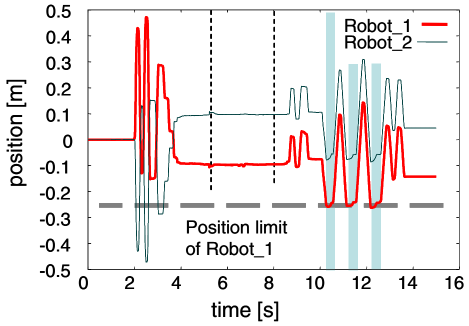


Fig. 13. Position response on experiment

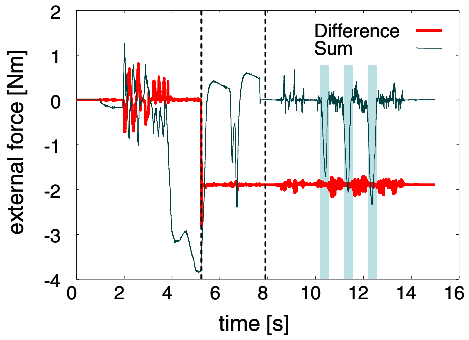


Fig. 14. Force response on experiment

2) $5.3 < t < 8.0$

A grasp function is added after the operator inserted a load between the robots. The sum value of the force response shows that the operator applied force while the rigid coupling function controlled the robots to stay. At the same time, the difference value of the force response was almost constant due to the stable grasping motion.

3) $t > 8.0$

The rigid coupling function is replaced by an inertia manipulation function. The grasped load therefore moved passively when operator applied force. A position limit function became active when robot_1 exceeded its position limit. The shaded areas show the term of the position limit function activated. Robot_1 stayed on the position limit while the operator applied approximately 2 Nm force. The grasping force hardly altered when the position limit function replaced the inertia manipulation function. The system connection was manipulated and the reachability matrix shifted from (14) to (15) when robot_1 exceeded its position limit. As a result, robot_1 came to the higher layer and the position limit function on robot_1 gained the first priority. In sum, exception handling is achieved through the manipulation of the system connection.

$$M'_R = \frac{1}{2}\begin{array}{c} \begin{array}{cc} 1 & \ \ 2 \end{array} \\ \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \end{array} \tag{14}$$

$$M'_R = \frac{1}{2}\begin{array}{c} \begin{array}{cc} 1 & \ \ 2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \end{array} \tag{15}$$

VIII. CONCLUSION

This paper described the design method of a decentralized control system. We applied the idea of function to achieve a simple and explicit controller design. A hierarchy due to the communication flow imposed restrictions to the role of function-based control system. The system connection was utilized to derive this hierarchical structure. A unified method to design the control system under these restrictions was proposed. Furthermore, these restrictions were utilized intentionally for the autonomous decision of functions. The validity of the proposed method was shown by the experiment and the simulation.

ACKNOWLEDGMENT

This work is supported in part by a Grant in Aid for the 21st century Center Of Excellence from the Ministry of Education, Culture, Sport, Science, and Technology in Japan.

REFERENCES

[1] N. Sebe: “A Unified Approach to Decentralized Controller Design,” Proc. IEEE Conf. Decision and Control, pp. 1086-1091 (2003)

[2] M. C. L. Sabatucci and A. Chella: “A Possible Approach to the Development of Robotic Multi-Agent Systems,” Proc. IEEE/WIC Intr. Conf. Intelligent Agent Technology, pp. 539-544

[3] T. Seki, Y. Fujimoto, T. Yakoh and K. Ohnishi: “An Expression of Dynamical System Connection in Decentralized Autonomous Control,” Tans. IEE Japan, Vol. 120-D, No. 1, pp. 80-87 (2000) (in Japanese)

[4] Y. Fujimoto, T. Seki, T. Yakoh and K. Ohnishi: “Dynamic Model of Informationally Connected Systems,” Trans. IEE Japan, Vol. 120-D, No. 1, pp. 19-28 (2000) (in Japanese)

[5] K. Akuzawa and K. Ohnishi: “Design Indices for Information Connection in Decentralized System,” Proc. IEEE Int. Conf. Industrial Electronics, Control and Instrumentation, pp. 2417-2422 (2003)

[6] T. Ueyama, T. Fukuda, F. Arai, Y. Katou, S. Matsumura and T. Uesugi: “A Study on Dynamically Reconfigurable Robotic Systems (10th Report, Distributed Control Structure for Organization using an Evaluation of Network Energy for Group Structure of Cebot),” J. of JSME, Part C, Vol. 58, No. 549, pp. 132-139 (1992) (in Japanese)

[7] Y. Arai, T. Fujii, H. Asama, S. Suzuki, H. Kaetsu and I. Endo: “Collision Avoidance in Multi-Robot Environment based on Local Communication,” J. of RSJ, Vol. 19, No. 1, pp. 45-57 (2001) (in Japanese)

[8] D. Gauthier, P. Freedman, G. Carayannis and A. S. Malowany: “Interprocess Communication for Distributed Robotics,” IEEE J. of Robotics and Automation, Vol. RA-3, No. 6, pp. 493-504 (1987)

[9] 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,” Proc. IEEE Intr. Conf. Robotics and Automation, pp. 1921-1926 (1991)

[10] Y. Fujimoto, T. Sekiguchi: “Fault-Tolerant Configuration of Distributed Discrete Controllers,” IEEE Trans. on Industrial Electronics, Vol. 50, No. 1, pp. 86-93 (2003)

[11] K. Ohnishi, M. Shibata, T. Murakami: “Motion Control for Advanced Mechatronics,” IEEE/ASME Trans. on Mechatronics, Vol. 1, No. 1, pp. 56-67 (1996)

[12] T. Murakami, F. Yu, K. Ohnishi: “Torque Sensorless Control in Multidegree-of-freedom Manipulator,” IEEE Trans. on Industrial Electronics, Vol. 40, No. 2, pp. 259-265, (1993)

[13] T. Tsuji and K. Ohnishi: “A Controller Design Method of Decentralized Control System,” Proc. 5th Intr. Power Electronics Conference (2005)

[14] T. Tsuji and K. Ohnishi: “Position/Force Scaling of Function-Based Bilateral Control System,” Proc. IEEE Intr. Conf. Industrial Technology (2004)