A Controller Design Method Based on Functionality

Toshiaki Tsuji, *Member, IEEE,* Kouhei Ohnishi, *Fellow, IEEE,* and Asif Šabanović, *Senior Member, IEEE*

***Abstract*—Robots are expected to expand their range of activities to human environment. Robots in human environment need redundancy for environmental adaptation. Furthermore, they have to automatically modify their controllers in response to varying conditions of the environment. Therefore, the authors have proposed a method to design a hyper-DOF control system efficiently. The method decouples a large control system into small independent components so-called function. Motion of the entire control system is expressed as superposition of multiple functions. Combination of some functions realizes many patterns of motion. Hence various motions are realized with much smaller efforts on controller design. Additionally, the controller design is explicit since a controller and a function corresponds directly.**

This study expands the method to multi-DOF robots in three-dimensional space since the conventional method was limited to a multi-robot system in one-dimensional space. A new problem of interference among function-based systems occurs along with the expansion. Disturbance observer is applied on each actuator to eliminate the interference. Procedures of controller design under varying conditions are also shown. The proposed method is applied to a grasping manipulator with 18DOF. Its experimental results show validity of the method.

***Index Terms*—fault tolerance, decentralized control, disturbance observer, acceleration control, motion control, mechatronics**

I. Introduction

Ability of motion control has recently improved due to development of mechatronics technology. From now on, motion control systems such as robots, electric vehicles and so on are expected to expand their applicable scope to human environment. Robots in human environment need redundancy for adaptation. Furthermore, they are often required to execute a complicated task concurrent with adaptation to environment. It is therefore necessary to solve a design problem of large-scale systems with a complicated task.

Decentralized control is a promising method for large-scale systems. It is preeminent in many features such as flexibility, fault tolerance, expandability, and rapid response. Many studies applied it to robot control systems. Among them, interesting concepts such as subsumption architecture[1], multi-agent system[2] and cell structure[3] have been proposed. Holonic architecture[4] is an interesting concept that allows reconfiguration of a large control system in manufacturing environments. Artificial intelligence is often introduced to solve the design problem of these methods. Decentralized control is also utilized for fault tolerant systems[5]. More explicit and simple framework in view of controller design is desired although the methods for decentralized control systems are interesting as concepts.

Decomposition block control[6] is one of the efficient solutions. It transforms a control system into BCD-form and simplifies the design problem. Arimoto and Nguyen showed that overall control input can be designed by linear superposition of all signals under the condition of unique stationary resolution of the controlled position variables[7]. Okada, Tatani and Nakamura proposed a method to symbolize the robot motion based on the singular value decomposition[8]. Lee and Li presented a decoupled design method that makes a bilateral control system behave as a common passive rigid mechanical body[9]. Control methods that apply the idea of modal decomposition have been recently developed[10], [11], [12]. Modal decomposition is a way to decompose a control system into multiple subsystems based on modal information. The word “mode” in these studies denotes essential information for the control system. For example, the study in [10] extracts two environmental modes: inclination and heaving modes. Note that the environment may have infinite modes due to its diversity. Biped locomotion on rough terrain was achieved by a hybrid control system decoupled to heaving and inclination modes controllers since the two modes are information essential for adaptation to environment. Tsuji, Nishi and Ohnishi extended the idea of environmental modes to function modes, which corresponds to other general tasks[13], [14]. Onal and Šabanović implemented a sensitive bilateral control using sliding mode control based on function modes[15]. Function modes provide a unified design method that deals with both task variation and exception handling. Although controller design becomes simple and explicit with the framework, the study was limited to one-dimensional space. This paper therefore extends the framework for robots in three-dimensional space. The largest problem here is dynamical interaction between decoupled modes. Disturbance observer(DOB)[16] is applied to cancel the dynamical interference and assure independence of each function mode. An extended form of function-based controller design is also described.

This paper is organized as follows. Section II describes the basic idea of functionality and extend it to three-dimensional systems. Section III shows a design flow of function-based controller design and describes the way of configurating the

T. Tsuji was with the Department of System Design Engineering, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Japan. He is now with the Department of Electrical and Electronic Systems, Saitama University, 255 Shimo-ohkubo, Sakura-ku, Saitama, Japan. (E-mail:tsuji@ees.saitama-u.ac.jp)

K. Ohnishi is with the Department of System Design Engineering, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Japan. (E-mail:ohnishi@sd.keio.ac.jp)

A. Šabanović is with the Mechatronics Program, Faculty of Engineering and Natural Sciences, Sabanci University, Orhanli, Tuzla, 34956 Istanbul, Turkey. (E-mail:asif@sabanciuniv.edu)

controller. Section IV shows an example of a control system for a parallel link manipulator. Section V shows its experimental result. Section VI is the conclusion of this paper.

II. Function Based Controller Design

A. Concept of functionality

In this study, a complicated control system is decoupled into small independent components based on modal information named function mode. Function mode is an idea proposed in [13]. Each function mode corresponds to a simple motion named function. Fig. 1a) shows one of the examples of mobile robots in one-dimensional space. In order to convey a load, robot A and B have to move the load after they grasp it. Entire motion of robot A and B can be decoupled to simplified motion of grasping and moving. These simple motions decoupled from a complicated motion are called function. Function mode is modal information that represents a function. Function mode is easily derived through a matrix $\boldsymbol{T}$ as shown in Fig. 1b). Here, x_A and x_B denote position of robot A and B respectively. x_{GR} and x_{MV} denote function mode of grasping and moving functions. Moving function is realized by a position controller on function mode x_{MV}. On the other hand, grasping function is realized by a force controller on mode x_{GR}. If the system has limited range of movement or velocity, exception handling such as position limit and velocity limit can be implemented as a function. Fig. 1c) shows an example when robot B comes to position limit. Here, x_{PL} denotes function mode of position limit. Position controller is applied on x_{PL}, equal to x_B in this situation, so that robot B does not exceed the position limit. Although moving function is halted then, grasping function is sustained under exception. The examples show that flexibility of controller design is enhanced by manipulating the combination of functions.

Assuming that functions are independent to each other, motion of the entire control system is represented as superposition of these functions. This property is named ”functionality” in this study. Combination of some functions realizes many patterns of motion. Hence various motions are realized with much smaller efforts on controller design. Furthermore, the controller design is explicit since a controller and a function corresponds directly.

The entire block diagram is shown in Fig. 2.

B. Advantage of function-based controller design

The originality of function-based controller design is to design each controller as a detachable component. It is similar to design of peripheral equipment for PC as shown in Fig. 3. Many kinds of function-based controllers are designed in advance like peripheral equipment. Among them, requisite functions are exerted depending on the varying system role. Great patterns of tasks are realized with such a framework. Furthermore, the design is still simple and explicit. In sum, this framework is useful for control of robots adaptive to complicated environments since it solves the issues of task variation and exception handling of complicated systems[14].

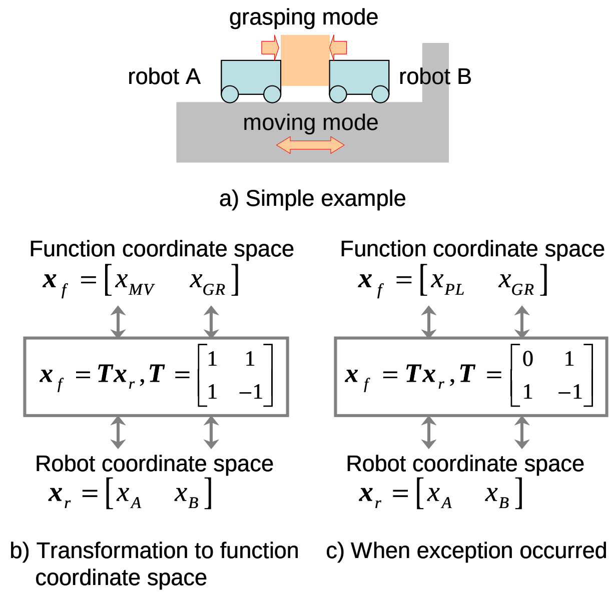

Fig. 1. Function mode in one dimensional space

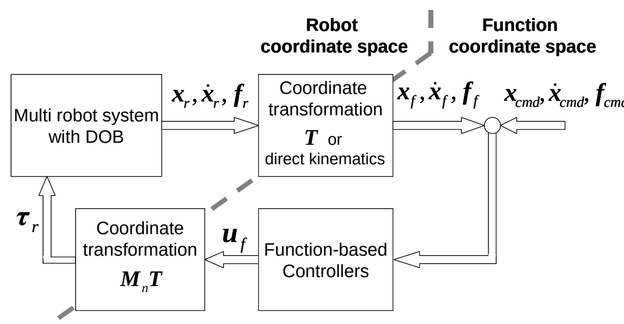

Fig. 2. Block diagram of function-based control system

C. Coordinate transformation based on function

The controller design based on functionality needs coordinate transformation. Motor information should be transformed into modal information, which corresponds to functions such as “moving function” and “grasping function”. This subsection describes an extended form of the coordinate transformation.

There exist many kinds of functions for tasks, exception handling, and so on. These functions require various kinds of information such as arm tip position, motor angles and other modal information. Multi-layered transformation is therefore introduced. An outline of the transformation is shown in Fig. 4.

The coordinate transformation introduced in [14] is to derive function coordinate space from workspace information of each robot. Note that workspace of a one-dimensional robot

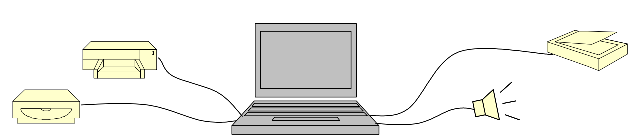

Fig. 3. Design as detachable component

corresponds to its joint space. A Jacobian matrix is known for transformation from joint space to workspace. Additionally, transformation from real motor coordinate space to virtual motor coordinate space of sum and differential motor is introduced for a twin drive system. The mechanism of the twin drive system is described in the Appendix.

Several coordinate spaces are transformed through transformation matrices. ${}^{f}\boldsymbol{T}_r$, a transformation matrix from real motor coordinate space to function coordinate space, is derived by multiplying the matrices between each space.

At first, function coordinate space is transformed from arm coordinate space (i.e. workspace of each robot) as follows:

$$\boldsymbol{x}_f = {}^{f}\boldsymbol{T}_a \boldsymbol{x}_a \quad (1)$$
$$\dot{\boldsymbol{x}}_f = {}^{f}\boldsymbol{T}_a \dot{\boldsymbol{x}}_a \quad (2)$$
$$\ddot{\boldsymbol{x}}_f = {}^{f}\boldsymbol{T}_a \ddot{\boldsymbol{x}}_a \quad (3)$$
$$\boldsymbol{f}_f = {}^{f}\boldsymbol{T}_a \boldsymbol{f}_a \quad (4)$$
$$\boldsymbol{x}_a = [\boldsymbol{x}_{a1}, \boldsymbol{x}_{a2}, \cdots, \boldsymbol{x}_{am}]^T$$
$$\boldsymbol{f}_a = [\boldsymbol{f}_{a1}, \boldsymbol{f}_{a2}, \cdots, \boldsymbol{f}_{am}]^T.$$

Here, $\boldsymbol{x}_{ai} \in \boldsymbol{R}^3$ and it denotes position of an end effector on the ith robot. $\boldsymbol{f}_{ai} \in \boldsymbol{R}^3$ and it denotes external force on the end effector. The subscript f denotes function coordinate space and the subscript a denotes arm coordinate space. ${}^{f}\boldsymbol{T}_a \in \boldsymbol{R}^{N\times M}$, m is total number of robots, M is total DOF of robots, and N is total DOF of functions.

${}^{f}\boldsymbol{T}_a$ corresponds to the transformation matrix in [14]. In most cases, it is composed of $1, 0$ and -1 to derive modal information of related arm tip variables.

As shown from (1) to (4), position, velocity, acceleration and external force are all transformed by ${}^{f}\boldsymbol{T}_a$. Position of arm tip is calculated by direct kinematics based on a real motor response. Force on arm tip is measured by reaction force observer (RFOB)[18] in this study while the proposed method is also applicable for robots with force sensors. Then, position and force information for function-based controller are derived from (1) and (4), respectively. Velocity and acceleration information on function coordinates are derived from a real motor response by (5) and (6).

$$\dot{\boldsymbol{x}}_f = {}^{f}\boldsymbol{T}_r \dot{\boldsymbol{x}}_r \quad (5)$$
$$\ddot{\boldsymbol{x}}_f = {}^{f}\boldsymbol{T}_r \ddot{\boldsymbol{x}}_r \quad (6)$$

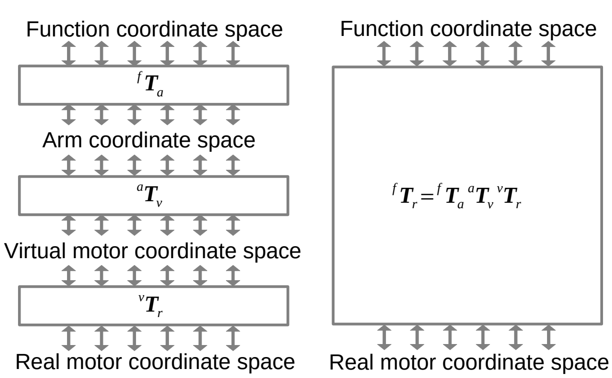

Fig. 4. Outline of coordinate transformation

$${}^{f}\boldsymbol{T}_r = {}^{f}\boldsymbol{T}_a {}^{a}\boldsymbol{T}_v {}^{v}\boldsymbol{T}_r \quad (7)$$

The subscript r denotes real motor coordinate space while the subscript v denotes virtual motor coordinate space for the twin drive system. ${}^{a}\boldsymbol{T}_v$ is a transformation matrix similar to a Jacobian matrix. It transforms virtual motor coordinate space to arm coordinate space. ${}^{v}\boldsymbol{T}_r$ is a transformation matrix from real motor coordinate space to virtual motor coordinate space. ${}^{a}\boldsymbol{T}_v \in \boldsymbol{R}^{M\times M}$ and ${}^{v}\boldsymbol{T}_r \in \boldsymbol{R}^{M\times M}$.

${}^{v}\boldsymbol{T}_r$ is a specific transformation matrix only for a twin drive system. It is a unit matrix $\boldsymbol{I}$ for other systems. In a one-dimensional system, the Jacobian matrix ${}^{a}\boldsymbol{T}_v$ is also a unit matrix $\boldsymbol{I}$.

${}^{f}\boldsymbol{T}_r$ can be explained as an extended Jacobian matrix. It is extended for a twin-drive system and cooperative work of a multi-robot system. It is therefore called "cooperative Jacobian matrix". ${}^{f}\boldsymbol{T}_a$, which is simply named "transformation matrix" in [14], is called "function matrix" for distinction.

Control input $\boldsymbol{u}_f$ is derived from controllers on function coordinate space. Here, $\boldsymbol{u}_f$ is in acceleration dimension. Torque input in real motor coordinate is derived from (8).

$$\boldsymbol{\tau}_r = \boldsymbol{M}_n {}^{f}\boldsymbol{T}_r^{+} \boldsymbol{u}_f \quad (8)$$
$${}^{f}\boldsymbol{T}_r^{+} = ({}^{f}\boldsymbol{T}_r^{T} {}^{f}\boldsymbol{T}_r)^{-1} {}^{f}\boldsymbol{T}_r^{T}$$

Here, $\boldsymbol{M}_n \in \boldsymbol{R}^{M\times M}$. $\boldsymbol{M}_n$ is a nominal inertia matrix of robots. The condition for deriving torque input is

$$rank(\boldsymbol{M}_n {}^{f}\boldsymbol{T}_r^{+}) = M. \quad (9)$$

Therefore, if any of functions are dependent on each other, a new function should be added. On the other hand, if $rank(\boldsymbol{M}_n {}^{f}\boldsymbol{T}_r^{+}) > M$, one of the functions with the lowest priority should be halted.

D. Dynamics in function coordinate space

It is to be anticipated from the name of cooperative Jacobian matrix that the coordinate transformation is for kinematics of a large-scale system. Virtual dynamics in a function coordinate interferes with each other, contrary to the method proposed in [14]. The interference occurs due to the generalization to three-dimensional systems.

DOB is applied to all of real motors in this method to cancel the interference. Fig. 5 shows a block diagram of DOB. DOB estimates and compensates disturbance on the control system. (10) shows the estimated disturbance value.

$$\hat{\tau}_{dis} = \frac{G_{dis}}{s + G_{dis}} \left(K_{tn} I_a^{ref} - \frac{G_v}{s + G_v} J_n \omega s \right) \quad (10)$$

Since the estimated disturbance value is proportional to acceleration value, DOB achieves acceleration control. It is well known that the plant works as a nominal system when acceleration control is acquired[16]. Hence inputs from position/force controller based on functions are superposed without any interference in the control frequency range lower than the cutoff frequency of DOB. Multirate control with a short sensor sampling rate[19] is a good candidate to heighten the cutoff frequency. Modal decomposition in acceleration dimension provides explicit controller design. In this point of view, this method has an advantage over other decomposition methods.

III. Configuration of Function-Based Control System

A. Procedures of controller design

A design flow of function-based control system is shown in Fig. 6. Firstly, the system role is determined by a designer of the control system. Secondly, the designer divides the system role into functions. Thirdly, a priority order of functions is determined. Important functions should be secured even if the number of active functions alters. Then, the transformation matrix ${}^{f}\boldsymbol{T}_r$ is derived. The number of functions is modified so that rank of $\boldsymbol{M}_n\,{}^{f}\boldsymbol{T}_r$ agrees with total DOF of robots M. Otherwise, (9) is unsatisfied. Finally, function-based controllers are designed individually.

B. Reconfiguration for alteration of system role

When the system role alters, combination of functions and its transformation matrix should be modified. At first, new combination of task functions should be given by the designer. Here, a task function is a function to acquire the system role while a performance-limit function is a function to deal with an exception. In the next place, the transformation matrix should be modified along with the functions. Majority of task functions control relative position or relative force between arm tips. In this study, ${}^{f}\boldsymbol{T}_a$ denotes the relation between arm tips. In sum, ${}^{f}\boldsymbol{T}_a$ should be modified in a similar way in [14] by modifying $\boldsymbol{T}$ when the system role alters.

C. Reconfiguration for exception handling

Reconfiguration for exception handling is more difficult compared to that for alteration of the system role. There are three reasons:

- exceptions occur all of a sudden;
- the control system should choose the combination of functions autonomously; and
- not only ${}^{f}\boldsymbol{T}_a$ but also ${}^{a}\boldsymbol{T}_v$ or ${}^{v}\boldsymbol{T}_r$ should be modified since performance-limit functions that deal with exceptions are often based on a real motor output or a virtual motor output.

A method to modify a transformation matrix is introduced below.

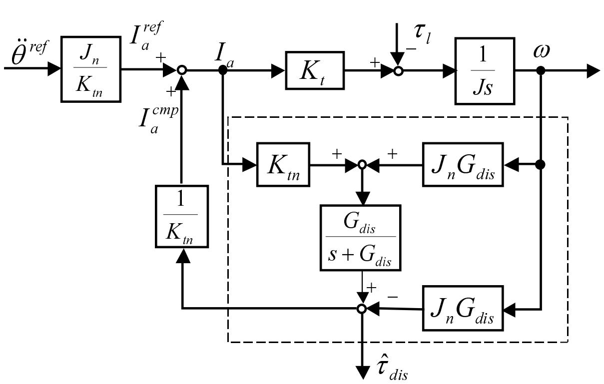

Fig. 5. Disturbance Observer

${}^{f}\boldsymbol{T}_r$ is described as follows:

$$ {}^{f}\boldsymbol{T}_r = \begin{bmatrix} {}^{f}\boldsymbol{t}_{r1}^T, & {}^{f}\boldsymbol{t}_{r2}^T, & \cdots, & {}^{f}\boldsymbol{t}_{rN}^T \end{bmatrix}^T. \tag{11} $$

${}^{f}\boldsymbol{t}_{ri} \in \boldsymbol{R}^M$, it extracts the coordinate of the ith function. It denotes a function mode and depends on the characteristics of the function. Function modes for task functions are derived all at once from (7).

On the other hand, performance-limit functions, which are activated in a special case also have their function modes. The function mode of the performance-limit function should be derived individually when the function is activated. The function mode of the performance-limit function is derived from various ways since performance-limits may exist in each layer of the multi-layered coordinate transformation. For example, a function mode of a velocity-limit function on the kth real motor is derived as follows:

$$ {}^{f}\boldsymbol{t}_{r,PL}^T = \begin{bmatrix} t_1, & t_2, & \cdots, & t_M \end{bmatrix} \\ \begin{cases} t_i = 1 & (i = k) \\ t_i = 0 & (otherwise). \end{cases} \tag{12} $$

Here, ${}^{f}\boldsymbol{t}_{r,PL}$ denotes a function mode of a performance-limit function.

A position-limit function for avoidance of a singular point is shown as another example of a performance-limit function. A joint angle of the twin drive system corresponds to a response value of a virtual differential motor. Hence a singular point is avoided by setting a position-limit on the virtual motor. A function mode of the position-limit function for the kth virtual motor is derived as follows:

$$ {}^{f}\boldsymbol{t}_{r,PL} = {}^{v}\boldsymbol{t}_{rk} \tag{13} $$

where, ${}^{v}\boldsymbol{T}_r = \begin{bmatrix} {}^{v}\boldsymbol{t}_{r1}^T, & {}^{v}\boldsymbol{t}_{r2}^T, & \cdots, & {}^{v}\boldsymbol{t}_{rN}^T \end{bmatrix}^T$. When the kth virtual motor response extracted by ${}^{v}\boldsymbol{T}_{r,PL}$ exceeds its limit, a position controller is implemented to the function mode to keep within the limit value.

A function mode of a position-limit function on an arm tip is derived as follows:

$$ {}^{f}\boldsymbol{t}_{r,PL} = {}^{a}\boldsymbol{t}_{rk} \tag{14} $$

where, ${}^{a}\boldsymbol{T}_r = \begin{bmatrix} {}^{a}\boldsymbol{t}_{r1}^T, & {}^{a}\boldsymbol{t}_{r2}^T, & \cdots, & {}^{a}\boldsymbol{t}_{rN}^T \end{bmatrix}^T$. In this case, it is assumed that the position limit is set for the kth element of $\boldsymbol{x}_a$.

A procedure for exception handling is as follows:

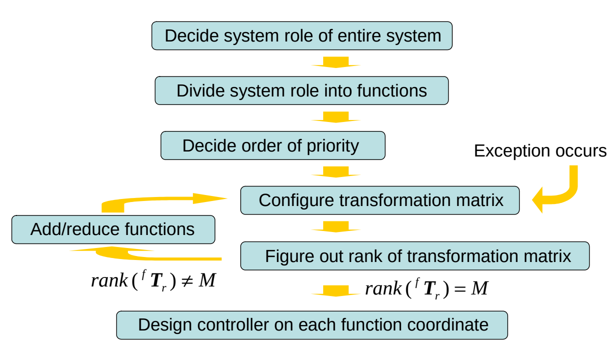

Fig. 6. Flow of controller design

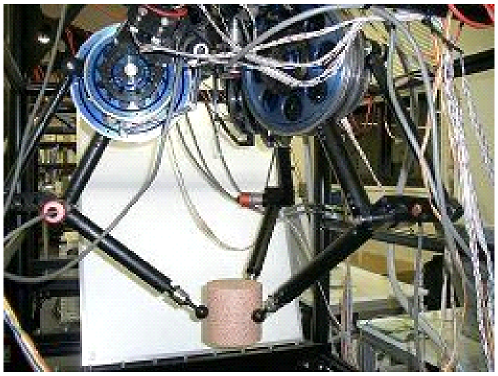

Fig. 7. Parallel link manipulators

1) keep observing variables for detecting exceptions;
2) select a relevant performance-limit function when one of the variables exceeds its limit;
3) derive ${}^{f}\boldsymbol{t}_{r,PL}$, a function mode of the performance-limit function;
4) derive ${}^{f}\boldsymbol{t}_{r,low}$, the function mode of the lowest-priority function;
5) derive ${}^{f}\boldsymbol{T}_{r,PL}$, the new transformation matrix for performance limit, by substituting ${}^{f}\boldsymbol{t}_{r,PL}$ to ${}^{f}\boldsymbol{t}_{r,low}$ in ${}^{f}\boldsymbol{T}_{r}$;
6) if $rank(\boldsymbol{M}_n {}^{f}\boldsymbol{T}_{r,PL}) \neq M$, select the function with the next-lowest priority, derive its function mode ${}^{f}\boldsymbol{t}_{r,low}$, and go to 5.
7) implement a function-based controller on each function coordinate

IV. Function-based Controller Design for Cooperative Grasping Motion

A control system for parallel link manipulators is shown in this section as a typical example of a function-based system. A picture of manipulators is shown in Fig. 7. The entire system consists of three parallel link manipulators with 3 DOF. There are 6 motors on each manipulator since the manipulator consists of twin drive systems. The details of the manipulators are shown in [20].

Three manipulators are fixed with orientation difference of 120 degrees respectively. Absolute position of the arm tip is presented by cylindrical coordinates as shown in (15).

$$\boldsymbol{x}_{ai} = [\ d_i, \quad \theta_i, \quad z_i\ ]^T \tag{15}$$

where d denotes distance from the z-axis based on the center of three manipulators, z denotes up-down position, and θ denotes rotation angle in a horizontal plane.

This study verifies the validity of the proposed method by an experiment of a human support operation with task variation. The operation is composed of 4 steps as illustrated in Fig. 8. Each step is described below.

Firstly in Step 1, the arm tips of the three manipulators move in compliance with external force only in the grasping mode, a mode that denotes sum of d_A,d_B and d_C. Step 2 starts after the operator inserts a cylindrical object between the three arm tips. In Step 2, the object is cooperatively grasped by the three arms while position and attitude of the object is kept constant under external force. In Step 3, the object moves in compliance with external force only in the pitching mode while it is grasped. The position of the object is kept constant at that time. In Step 4, it moves only in the up-down mode while its attitude is kept constant and it is grasped. Task functions for acquiring the system roles in Step 1 to Step 4 are shown in Table I. The overview of the coordinate transformation is shown in Fig. 9.

TABLE I
Functions for Parallel Link Manipulators

	Step 1	Step 2	Step 3	Step 4
Based on d				
Mode 1(Grasping)	**SC** (1)	**GR** (1)	**GR** (1)	**GR** (1)
Mode 2	RC (2)	RC (2)	RC (2)	RC (2)
Mode 3	RC (3)	RC (3)	RC (3)	RC (3)
Based on θ				
Mode 1(Rolling)	RC (9)	RC (9)	RC (9)	RC (9)
Mode 2	RC (8)	RC (8)	RC (8)	RC (8)
Mode 3	RC (7)	RC (7)	RC (7)	RC (7)
Based on z				
Mode 1(Up-down)	RC (6)	RC (6)	RC (6)	**SC** (6)
Mode 2(Pitching)	RC (5)	RC (5)	**SC** (5)	RC (5)
Mode 3(Yawing)	RC (4)	RC (4)	RC (4)	RC (4)
Based on virtual sum motors				
Mode 1	VC (10)	VC (10)	VC(10)	VC(10)
⋮	⋮	⋮	⋮	⋮
Mode 9	VC (18)	VC (18)	VC(18)	VC(18)

Here, RC, SC, VC, and GR denote functions of rigid coupling, spring coupling, velocity control, and grasping, respectively. Numbers in parentheses denote the priority order in task functions. The grasping function has higher priority to secure the object. Velocity control functions on sum motor coordinates keep velocity of virtual sum motors constant to cancel static friction. The velocity control functions therefore have lower priority since outputs of the functions have relatively small effects on the operation. The priority order of other task functions is given arbitrarily. Performance-limit functions

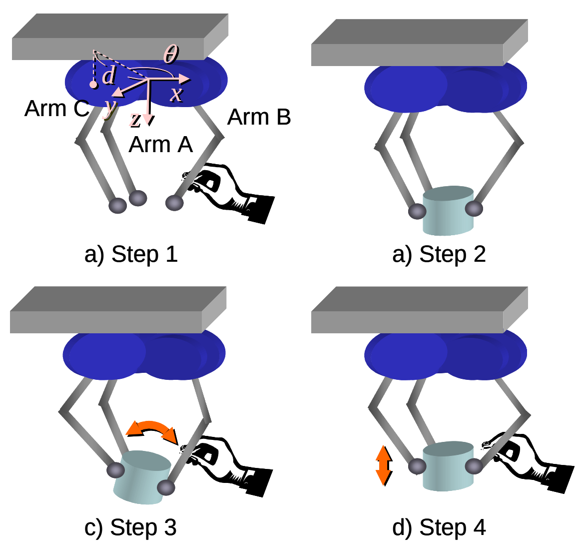

Fig. 8. Illustration of human support operation

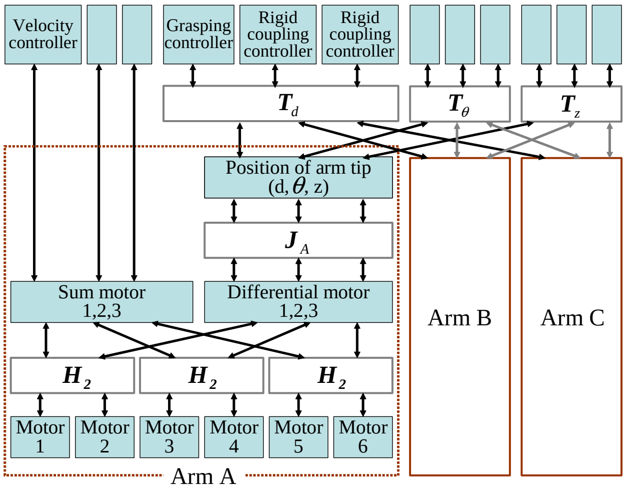

Fig. 9. Overview of entire coordinate transformation

exist in addition to the task functions. Priority of performance-limit functions is set higher than that of task functions so that they are compulsively activated when exceptions occur.

The function matrix ${}^f\boldsymbol{T}_a$ for such functions is given as follows:

$$
{}^f\boldsymbol{T}_a = \begin{bmatrix} \boldsymbol{I}_9 & & & \\ & \boldsymbol{T}_d & & \\ & & \boldsymbol{T}_\theta & \\ & & & \boldsymbol{T}_z \end{bmatrix} {}^f\boldsymbol{S}_a \tag{16}
$$

$$
\begin{aligned}
{}^f\boldsymbol{S}_a &= [\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_7, \boldsymbol{s}_8, \boldsymbol{s}_9, \boldsymbol{s}_{13}, \boldsymbol{s}_{14}, \boldsymbol{s}_{15}, \\
&\quad \boldsymbol{s}_4, \boldsymbol{s}_{10}, \boldsymbol{s}_{16}, \boldsymbol{s}_5, \boldsymbol{s}_{11}, \boldsymbol{s}_{17}, \boldsymbol{s}_6, \boldsymbol{s}_{12}, \boldsymbol{s}_{18}]^T \qquad (17) \\
\boldsymbol{s}_j &= [\begin{matrix} s_1, & s_2, & \cdots, & s_{18} \end{matrix}] \\
& \begin{cases} s_i = 1 & (i = j) \\ s_i = 0 & (otherwise). \end{cases}
\end{aligned}
$$

$$
\boldsymbol{T}_d = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \tag{18}
$$

$$
\boldsymbol{T}_\theta = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \tag{19}
$$

$$
\boldsymbol{T}_z = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & -1 & 2 \end{bmatrix} \tag{20}
$$

where, ${}^f\boldsymbol{S}_a$ is a permutation matrix to change an order of variables from an arm-based order to a function-based order. $\boldsymbol{I}_n$, an nth order unit matrix, corresponds to virtual sum motor coordinates. $\boldsymbol{T}_d$ denotes a function matrix in d coordinates while $\boldsymbol{T}_\theta$ and $\boldsymbol{T}_z$ denote that in θ and z coordinates. The first, second and third rows of $\boldsymbol{T}_d$ extract function modes named Mode 1, Mode 2 and Mode 3, respectively. Modes extracted by $\boldsymbol{T}_\theta$ and $\boldsymbol{T}_z$ are also named in the same way. Mode 1 is sum of three manipulators' responses. Mode 1 of d coordinate denotes grasping motion while that of θ coordinate denotes rolling motion and that of z coordinate denotes up-down motion. The second and the third rows of $\boldsymbol{T}_d$ and $\boldsymbol{T}_\theta$ are to derive the difference value of the Arm A and others. The second and the third rows of $\boldsymbol{T}_z$ extract pitching and yawing motion of the object, respectively.

${}^a\boldsymbol{T}_v$ in this study is as follows:

$$
{}^a\boldsymbol{T}_v = \begin{bmatrix} {}^a\boldsymbol{T}_{vA} & & \\ & {}^a\boldsymbol{T}_{vB} & \\ & & {}^a\boldsymbol{T}_{vC} \end{bmatrix} \tag{21}
$$

$$
{}^a\boldsymbol{T}_{vA} = \begin{bmatrix} \boldsymbol{I}_3 & \\ & \boldsymbol{J}_A \end{bmatrix} \tag{22}
$$

$$
{}^a\boldsymbol{T}_{vB} = \begin{bmatrix} \boldsymbol{I}_3 & \\ & \boldsymbol{J}_B \end{bmatrix} \tag{23}
$$

$$
{}^a\boldsymbol{T}_{vC} = \begin{bmatrix} \boldsymbol{I}_3 & \\ & \boldsymbol{J}_C \end{bmatrix}. \tag{24}
$$

Here, $\boldsymbol{J}_A$, $\boldsymbol{J}_B$ and $\boldsymbol{J}_C$ denote Jacobian matrices for arm A, B and C, respectively.

${}^v\boldsymbol{T}_r$ in this study is as follows:

$$
{}^v\boldsymbol{T}_r = \begin{bmatrix} {}^v\boldsymbol{T}_{rA} & & \\ & {}^v\boldsymbol{T}_{rB} & \\ & & {}^v\boldsymbol{T}_{rC} \end{bmatrix} \tag{25}
$$

$$
\begin{aligned}
{}^v\boldsymbol{T}_{rA} &= {}^v\boldsymbol{T}_{rB} = {}^v\,\boldsymbol{T}_{rC} \\
&= {}^v\boldsymbol{S}_r \begin{bmatrix} \boldsymbol{H}_2 & & \\ & \boldsymbol{H}_2 & \\ & & \boldsymbol{H}_2 \end{bmatrix} \qquad (26) \\
{}^v\boldsymbol{S}_r &= [\boldsymbol{s}_1, \boldsymbol{s}_3, \boldsymbol{s}_5, \boldsymbol{s}_2, \boldsymbol{s}_4, \boldsymbol{s}_6]^T \qquad (27) \\
\boldsymbol{s}_j &= [\begin{matrix} s_1, & s_2, & \cdots, & s_6 \end{matrix}] \\
& \begin{cases} s_i = 1 & (i = j) \\ s_i = 0 & (otherwise) \end{cases}
\end{aligned}
$$

$$
\boldsymbol{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{28}
$$

where, ${}^v\boldsymbol{S}_r$ is a permutation matrix to change an order of variables from real motors to virtual motors. $\boldsymbol{H}_2$ is a second-order Hadamard matrix.

Block diagrams of function-based controllers are shown in Fig. 10. Each function consists of a simple position/force controller.

V. Experiment

Experimental results are shown in this section. Table II shows control gains in the experiment. Figs. 11 and 12 show responses in d coordinate and in z coordinate, respectively.

When the operator maneuvered the Arm A in Step 1, all three manipulators moved only in grasping mode and accomplished open-close motion. Force responses of Arm A fluctuated due to the operator's force.

An object was grasped in Step 2 after the operator inserted the object. Then, force responses in grasping mode was about 13 N on average. The average is the grasping force. Grasping motion was retained while combination of functions was changed in later steps. Force responses in grasping mode fluctuated as the operator maneuvered the object. As indicated by the result, the condition to retain the grasping motion is to keep the external force smaller than the grasping force. Force responses in up-down mode show that about -3 N on average

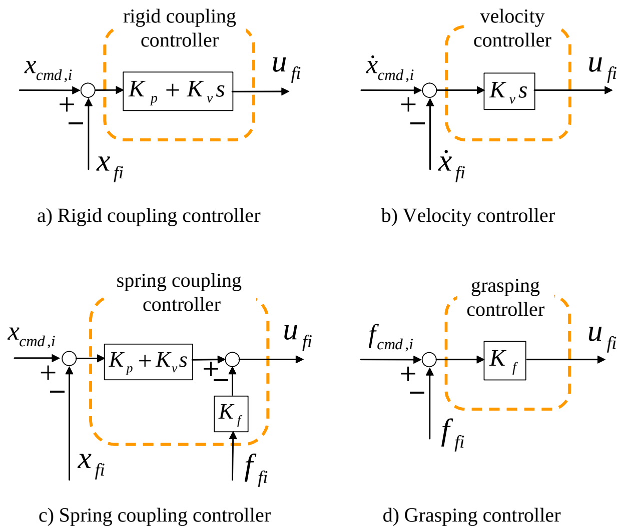

Fig. 10. Block diagram of functions

was acting on the manipulators. Since the object weighed 330 g, it seems that the average shows gravity force of the object.

The object was tilted in the pitching mode in Step 3 when the operator applied force in the z direction. On the other hand, the object went up and down in Step 4 when the operator applied force in the same direction. Position responses of Mode 1 in z coordinate was almost constant during Steps 1 to 3 while it varied relative to force response of Arm A during Step 4. At the same time, position responses of Mode 2 in z coordinate was almost constant during Steps 1,2 and 4 while it varied during Step 3. External force affected in all directions since the operator did not accurately maneuver. The object, however, moved only in the mode of spring coupling functions. The direction of free motion was changed by modifying the combination of functions while grasping motion was retained then. Interference between each mode rarely occurred due to acceleration control based on DOB.

TABLE II
CONTROL PARAMETERS

Position gain		K_p	600.0
Velocity gain		K_v	70.0
Force gain		K_f	8.0
Cutoff-frequency of DOB	[rad/sec]	G_{dis}	30.0
Cutoff-frequency of RFOB	[rad/sec]	G_f	15.0

VI. Conclusion

This study expanded the framework of function-based controller design to multi-DOF robots in three-dimensional space. The expanded form is also applicable to twin drive systems. A new problem of interference among function-based systems occurs after the expansion. DOB is applied on each actuator to eliminate the interference. The simplicity and explicitness of function-based controller design carry on despite the expansion since function-based systems are decoupled with DOB.

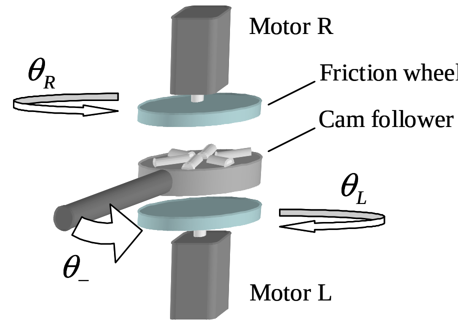

Fig. 13. Mechanism of twin drive system

Appendix

This section briefly describes a mechanism of a twin drive system[17]. Fig. 13 shows a schematic diagram of the twin drive system. The twin drive system is composed of a differential mechanism with two motors. Here, θ_R and θ_L denote the angle of motor R and motor L, respectively. Sum and difference of these two angles represent angles of a virtual sum motor and a virtual differential motor. These two virtual motors could be treated as two systems with independent coordinates. θ_-, the differential motor coordinate, appears as rotation of the joint. On the other hand, sum motor coordinate θ_+ do not affect the joint response. The velocity in sum motor coordinate $\dot{\theta}_+$ is controlled to hold a certain value to cancel the effect of static friction on real motors.

References

[1] R. A. Brooks: "A Robust Layered Control System For A Mobile Robot," *IEEE J. R & A*, Vol. RA-2, No. 1, pp. 14–23, 1986.

[2] M. C. L. Sabatucci and A. Chella: "A Possible Approach to the Development of Robotic Multi-Agent Systems," *Proc. IEEE/WIC Int. Conf. Intelligent Agent Technology*, pp. 539–544, 2003.

[3] 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. JSME, Part C*, Vol. 58, No. 549, pp. 132–139, 1992. (in Japanese)

[4] M. Sugi, Y. Maeda, Y. Aiyama, T. Harada and T. Arai, "A Holonic Architecture for Easy Reconfiguration of Robotic Assembly Systems," *IEEE Trans. on Robotics and Automation*, Vol. 19, No. 3, 2003.

[5] Y. Fujimoto, T. Sekiguchi, "Fault-Tolerant Configuration of Distributed Discrete Controllers," *IEEE Trans. on Industrial Electronics*, Vol. 50, No. 1, pp. 86–93, 2003.

[6] J. E. Hernandez, A. G. Loukianov, B. Castillo-Toledo and V. I. Utkin, "Observer Based Decomposition Control of Linear Delayed Systems," *Proc. IEEE Int. Conf. Decision and Control*, pp. 1867–1872, 2001.

[7] S. Arimoto, P. T. A. Nguyen, "Principle of Superposition for Realizing Dexterous Pinching Motions of a Pair of Robot Fingers with Soft-tips," *IEICE Trans. Fundamentals*, Vol. E84-A, No. 1, pp. 39–47, 2001.

[8] M. Okada, K. Tatani, Y. Nakamura, "Polynomial Design of the Nonlinear Dynamics for the Brain-Like Information Processing of Whole Body Motion," *Proc. IEEE Int. Conf. on R & A*, pp. 1410–1415, 2002.

[9] D. Lee and P. Y. Li, "Passive bilateral feedforward control of linear dynamically similar teleoperated manipulators," *IEEE Trans. Robotics & Automation*, Vol. 19, No. 3, pp. 443–456, 2003.

[10] M. Morisawa and K. Ohnishi, "Motion Control Taking Environmental Information into Account," *European Power Electronics and Drives Journal*, Vol. 12, No. 4, pp. 37–41, 2002.

[11] Y.Matsumoto, S.Katsura and K.Ohnishi, "An Analysis and Design of Bilateral Control Based on Disturbance Observer," *IEEE Int. Conf. Industrial Technology (ICIT'03)*, pp. 802–807, 2003.

[12] S. Katsura, K. Ohnishi and K. Ohishi, "Transmission of Force Sensation by Environment Quarrier Based on Multilateral Control," *IEEE Trans. Industrial Electronics*, Vol. 54, No. 2, 2007.

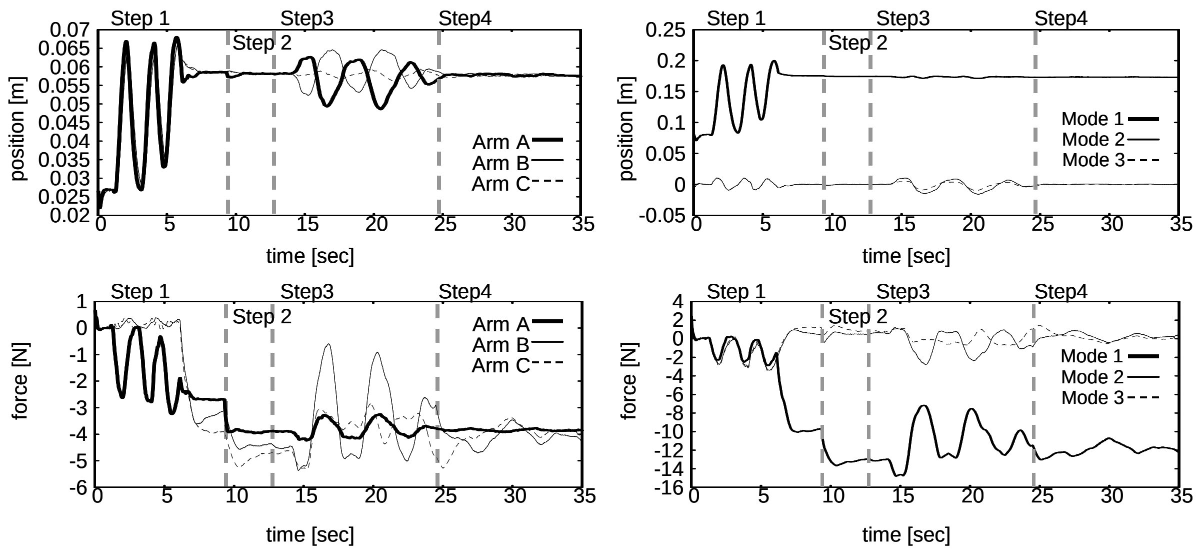

Fig. 11. Responses in d coordinate

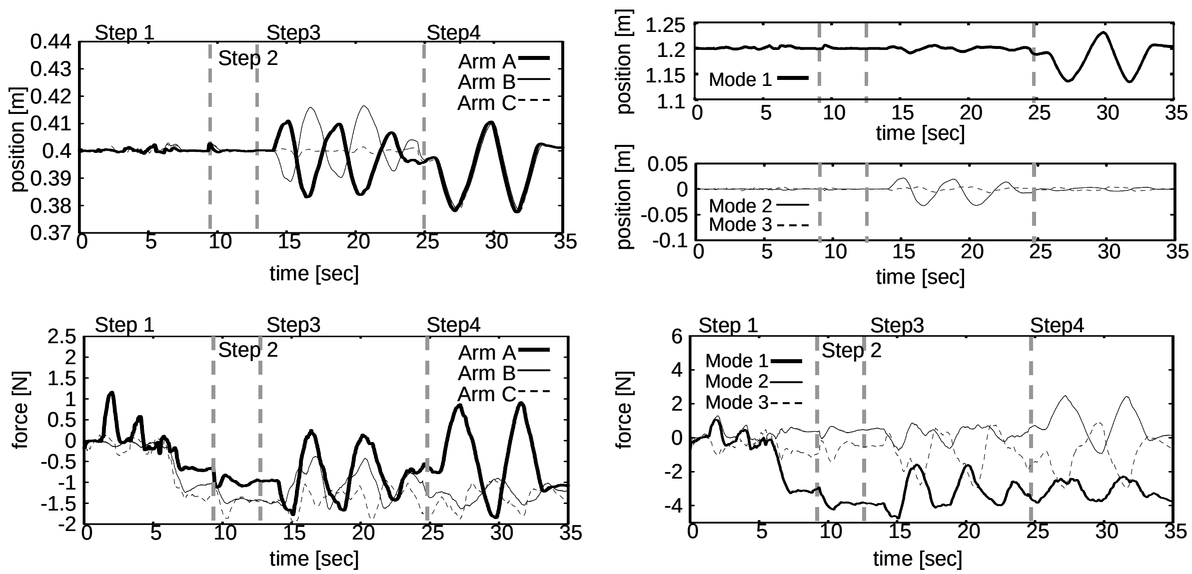

Fig. 12. Responses in z coordinate

[13] T. Tsuji, K. Natori, H. Nishi and K. Ohnishi, "Controller Design Method of Bilateral Control System," *European Power Electronics and Drives Journal*, Vol. 16, No. 2, pp. 22–28, 2006.

[14] T. Tsuji, H. Nishi and K. Ohnishi, "A Controller Design Method of Decentralized Control System," *IEEJ Trans. on Industry Applications*, Vol. 126-D, No. 5, pp. 630–638, 2006.

[15] C. D. Onal, A. Šabanović, "Bilateral Control with a Reflex Mechanism on the Slave Side," *Proc. of the 31st Annual Conf. of the IEEE Industrial Electronics Society (IECON2005)*, pp. 195–200, 2005.

[16] K. Ohnishi, M. Shibata, T. Murakami, "Motion Control for Advanced Mechatronics," *IEEE/ASME Trans. Mechatronics*, Vol. 1, No. 1, pp. 56–67, 1996.

[17] N. Hayashida, T. Yakoh, T. Murakami, K. Ohnishi, "A Friction Compensation in Twin Drive System," *Proc. 6th IEEE Int. Workshop on Advanced Motion Control*, pp.187–192, 2000.

[18] T. Murakami, R. Nakamura, F. Yu, K. Ohnishi, "Force Sensorless Compliant Control Based on Reaction Force Estimation Observer in Multi-Degrees-of-Freedom Robot," *Journal of RSJ*, Vol. 11, No. 5, pp. 765–768, 1993. (in Japanese)

[19] M. Mariko, T. Tsuji, K. Ohnishi, "Multirate Sampling Method for Acceleration Control System," *IEEE Trans. on Industrial Electronics*, Vol. 54, No. 3, pp. 1462–1471, 2007.

[20] T. Kageyama, K. Ohnishi, "An architecture of decentralized control for multi-degrees of freedom parallel manipulator," *Proc. 7th IEEE Int. Workshop on Advanced Motion Control*, pp. 74–79, 2002.

[21] 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 Int. Conf. R & A*, pp. 1921–1926, 1991.

[22] T. Sakuishi, Y. Izumikawa, K. Yubai and J. Hirai, "Fault-tolerant Control of Flexible Arm Based on Dual Youla Parameter Identification," *Proc. 9th IEEE Int. Workshop on Advanced Motion Control*, pp. 451–455, 2006.

[23] T. Tsuji, K. Ohnishi, A. Šabanović, "A controller design method based on functionality," *Proc. 9th IEEE Int. Workshop on Advanced Motion Control*, pp. 171–176, 2006.

PLACE
PHOTO
HERE

Toshiaki Tsuji (S'05, M'06) received a B.E.(2001), M.E. (2003), and Ph. D (2006) degrees from Keio university. He was a research associate in Department of Mechanical Engineering, Tokyo University of Science from 2006 to 2007. He is currently with Department of Electrical and Electronic Systems, Saitama University. His research interests include motion control, decentralized control and control of biped robots. He received the FANUC FA and Robot Foundation Original Paper Award in 2007.

PLACE
PHOTO
HERE

Kouhei Ohnishi (S'78, M'80, SM'00, F'01) received B.E. (1975), M.E. (1977), and Ph.D (1980) degrees, all in electrical engineering from University of Tokyo. Since 1980, he has been with Keio University. His research interests include robotics, motion control and haptics. He received the EPE-PEMC Council Award and the Dr.-Ing Eugene Mittelmann Achievement Award in 2004.

PLACE
PHOTO
HERE

Asif Šabanović (M'92, SM'04) received B. Sc. (1970), M. Sc. (1975), and Dr. Sc. (1979) degrees from the University of Sarajevo, respectively. From 1970 to 1991, he was with Energoinvest Institute for Control and Computer Sciences, Sarajevo. And then he was with the Department of Electrical Engineering, University of Sarajevo. Since 2000, he has been with the Faculty of Engineering and Natural Sciences, Sabanci University. He has been Visiting Researcher with the Institute of Control Science, Moscow, Russia (1975-1976) and Visiting Professor at the California Institute of Technology, Pasadena (1984-1985), Keio University, Yokohama, Japan (1991-1992), Yamaguchi University, Ube (1992-1993). He was head of CAD/CAM and the Robotics Department at TUBITAK-MRC, Istanbul, Turkey (1993-1995) and Head of the Engineering Department of B. H. Engineering and Consulting (1995-1999). His fields of interest include control systems, motion control systems, robotics, mechatronics, and power electronics.