

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-degrees-of-freedom (DOF) control system efficiently. The method decouples a large control system into small independent components 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 correspond directly. This paper expands the method to multi-DOF robots in 3-D space, since the conventional method was limited to a multirobot system in 1-D space. A new problem of interference among function-based systems occurs along with the expansion. A 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 18 DOF. Its experimental results show the validity of the method.

Index Terms—Acceleration control, decentralized control, disturbance observer (DOB), fault tolerance, mechatronics, motion control.

I. INTRODUCTION

THE ABILITY of motion control has recently improved due to the development of mechatronics technology. From now on, motion-control systems, such as robots, electric vehicles, and so on, are expected to expand their range of activity to human environments. Robots in human environments need redundancy for adaptation. Furthermore, they are often required to execute a complicated task concurrent with its adaptation to the 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], multiagent sys-

tem [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 [7] 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. Okada *et al.* [8] proposed a method to symbolize the robot motion based on the singular-value decomposition. Lee and Li [9] presented a decoupled design method that makes a bilateral-control system behave as a common passive rigid mechanical body. Control methods that apply the idea of modal decomposition have been recently developed [10]–[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 that are essential for the adaptation to the environment. Tsuji *et al.* [13], [14] extended the idea of environmental modes to function modes, which corresponds to other general tasks. Onal and Šabanović [15] implemented a sensitive bilateral control using a sliding-mode control based on function modes. 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, this paper was limited to 1-D space. This paper, therefore, extends the framework for robots in 3-D space. The largest problem here is dynamical interaction between decoupled modes. A 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 extends it to 3-D systems. Section III shows a design flow of function-based controller design and describes the way of configuring the controller. Section IV shows an example of a control system for a

Manuscript received September 11, 2006; revised July 24, 2007.

T. Tsuji was with the Department of System Design Engineering, Keio University, Yokohama 223-8521, Japan. He is now with the Department of Electrical and Electronic Systems, Saitama University, Saitama 338-8570, Japan (e-mail: tsuji@ees.saitama-u.ac.jp).

K. Ohnishi is with the Department of System Design Engineering, Keio University, Yokohama 223-8521, Japan (e-mail: ohnishi@sd.keio.ac.jp).

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

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIE.2007.906175

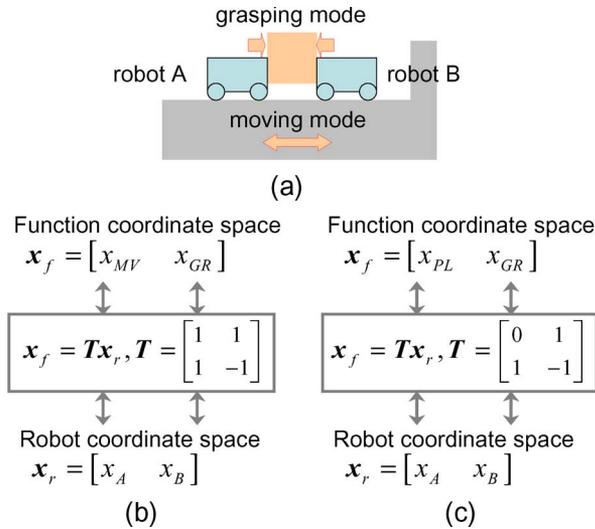


Fig. 1. Function mode in 1-D space. (a) Simple example. (b) Transformation to function coordinate space. (c) When an exception occurred.

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 paper, 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. 1(a) shows one of the examples of mobile robots in 1-D space. In order to convey a load, robots A and B have to move the load after they grasp it. The entire motion of robots A and B can be decoupled to the 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 T , as shown in Fig. 1(b). Here, x_A and x_B denote the position of robots A and B, respectively. x_{GR} and x_{MV} denote the function mode of grasping and moving functions. The 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 and velocity limits can be implemented as a function. Fig. 1(c) shows an example when robot B comes to a position limit. Here, x_{PL} denotes the function mode of position limit. A position controller is applied on x_{PL} , which is equal to x_B in this situation, so that robot B does not exceed the position limit. Although the moving function is halted then, the grasping function is sustained under exception. The examples show that flexibility of a controller design is enhanced by manipulating the combination of functions.

Assuming that functions are independent of each other, the motion of the entire control system is represented as superposition of these functions. This property is named “functionality”

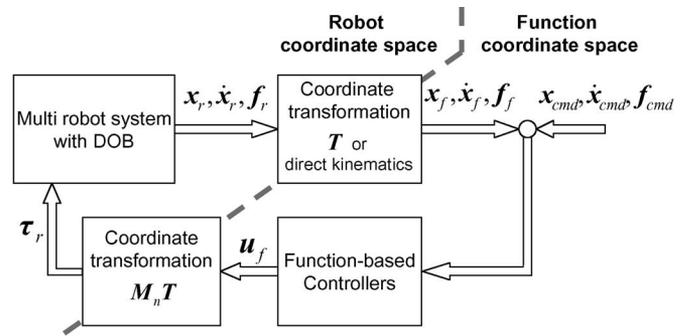


Fig. 2. Block diagram of function-based control system.

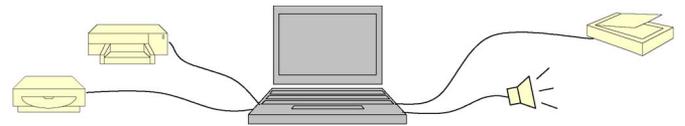


Fig. 3. Design as detachable component.

in this paper. 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 correspond 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 the 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 summary, 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].

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 section describes an extended form of the coordinate transformation.

There exists 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. Multilayered transformation is, therefore, introduced. An outline of the transformation is shown in Fig. 4.

The coordinate transformation introduced in [14] is to derive a function coordinate space from the workspace information of each robot. Note that the workspace of a 1-D robot corresponds

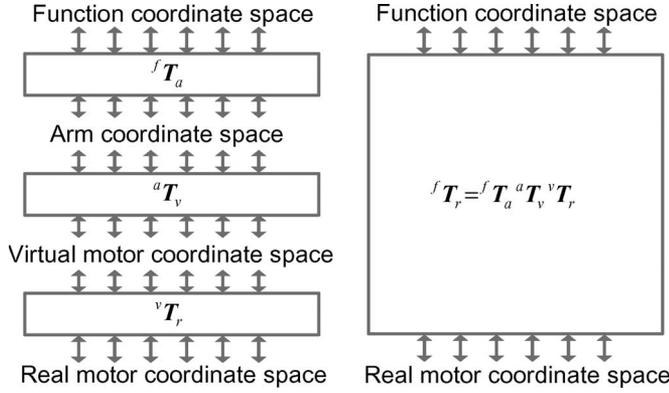


Fig. 4. Outline of coordinate transformation.

to its joint space. A Jacobian matrix is known for the transformation from joint space to workspace. Additionally, the 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. fT_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 an arm coordinate space (i.e., workspace of each robot) as follows:

$$\mathbf{x}_f = {}^fT_a \mathbf{x}_a \quad (1)$$

$$\dot{\mathbf{x}}_f = {}^fT_a \dot{\mathbf{x}}_a \quad (2)$$

$$\ddot{\mathbf{x}}_f = {}^fT_a \ddot{\mathbf{x}}_a \quad (3)$$

$$\mathbf{f}_f = {}^fT_a \mathbf{f}_a \quad (4)$$

$$\mathbf{x}_a = [\mathbf{x}_{a1}, \mathbf{x}_{a2}, \dots, \mathbf{x}_{am}]^T$$

$$\mathbf{f}_a = [\mathbf{f}_{a1}, \mathbf{f}_{a2}, \dots, \mathbf{f}_{am}]^T.$$

Here, $\mathbf{x}_{ai} \in \mathbf{R}^3$ and it denotes position of an end effector on the i th robot. $\mathbf{f}_{ai} \in \mathbf{R}^3$ and it denotes external force on the end effector. The subscript f denotes the function coordinate space, and the subscript a denotes the arm coordinate space. ${}^fT_a \in \mathbf{R}^{N \times M}$, where m is the total number of robots, M is the total degrees-of-freedom (DOF) of robots, and N is the total DOF of functions.

fT_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)–(4), position, velocity, acceleration, and external force are all transformed by fT_a . Position of the arm tip is calculated by direct kinematics based on a real motor response. Force on the arm tip is measured by a reaction force observer [18] in this paper, while the proposed method is also applicable for robots with force sensors. Then, the position and force information for function-based controllers are derived from (1) and (4), respectively. The velocity and acceleration information on function coordinates are derived from a real

motor response by (5) and (6), respectively.

$$\dot{\mathbf{x}}_f = {}^fT_r \dot{\mathbf{x}}_r \quad (5)$$

$$\ddot{\mathbf{x}}_f = {}^fT_r \ddot{\mathbf{x}}_r \quad (6)$$

$${}^fT_r = {}^fT_a {}^aT_v {}^vT_r. \quad (7)$$

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

vT_r is a specific transformation matrix only for a twin-drive system. It is a unit matrix \mathbf{I} for other systems. In a 1-D system, the Jacobian matrix aT_v is also a unit matrix \mathbf{I} .

fT_r can be explained as an extended Jacobian matrix. It is extended for a twin-drive system and cooperative work of a multirobot system. It is therefore called ‘‘cooperative Jacobian matrix.’’ fT_a , which is simply named ‘‘transformation matrix’’ in [14], is called ‘‘function matrix’’ for distinction.

Control input \mathbf{u}_f is derived from controllers on the function coordinate space. Here, \mathbf{u}_f is in acceleration dimension. Torque input in real motor coordinate is derived from

$$\boldsymbol{\tau}_r = \mathbf{M}_n {}^fT_r^+ \mathbf{u}_f$$

$${}^fT_r^+ = ({}^fT_r^T {}^fT_r)^{-1} {}^fT_r^T. \quad (8)$$

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

$$\text{rank}(\mathbf{M}_n {}^fT_r^+) = M. \quad (9)$$

Therefore, if any of the functions are dependent on each other, a new function should be added. On the other hand, if $\text{rank}(\mathbf{M}_n {}^fT_r^+) > M$, 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 interfere with each other, contrary to the method proposed in [14]. The interference occurs due to the generalization to 3-D systems.

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

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

Since the estimated disturbance value is proportional to the acceleration value, the DOB achieves acceleration control. It is well known that the plant works as a nominal system when the acceleration control is acquired [16]. Hence, inputs from the position/force controller based on functions are superposed

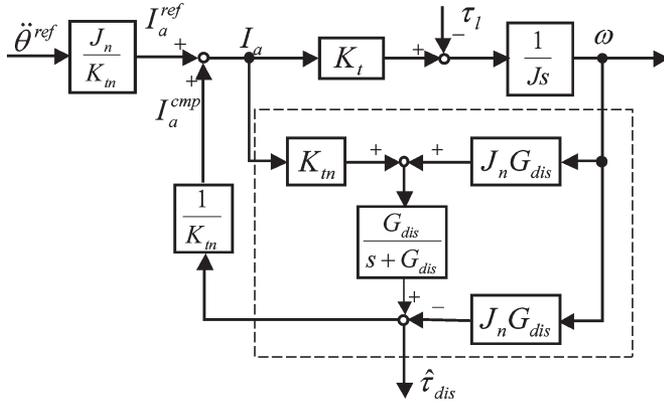


Fig. 5. Disturbance observer.

without any interference in the control-frequency range lower than the cutoff frequency of the DOB. Multirate control with a short sensor-sampling rate [19] is a good candidate to heighten the cutoff frequency. Modal decomposition in the 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 a function-based control system is shown in Fig. 6. First, the system role is determined by a designer of the control system. Second, the designer divides the system role into functions. Third, a priority order of functions is determined. Important functions should be secured even if the number of active functions alters. Then, the transformation matrix ${}^f T_r$ is derived. The number of functions is modified so that the rank of $M_n {}^f T_r$ agrees with the 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, a combination of functions and its transformation matrix should be modified. At first, a 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 the arm tips. In this paper, ${}^f T_a$ denotes the relation between arm tips. In sum, ${}^f T_a$ should be modified in a similar way in [14] by modifying T when the system role alters.

C. Reconfiguration for Exception Handling

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

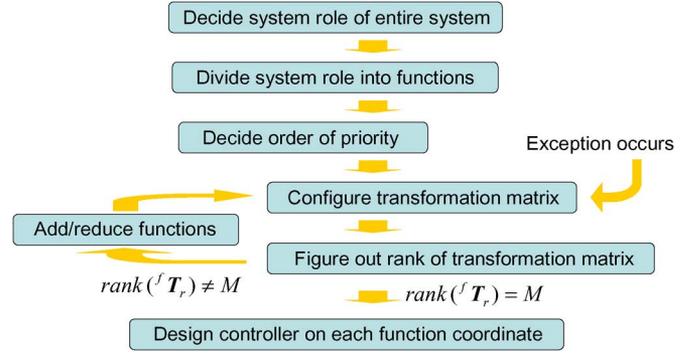


Fig. 6. Flow of controller design.

- 1) exceptions occur all of a sudden;
- 2) the control system should choose the combination of functions autonomously;
- 3) not only ${}^f T_a$ but also ${}^a T_v$ or ${}^v T_r$ 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 used to modify a transformation matrix is introduced below.

${}^f T_r$ is described as follows:

$${}^f T_r = [{}^f t_{r1}^T, {}^f t_{r2}^T, \dots, {}^f t_{rN}^T]^T. \quad (11)$$

${}^f t_{ri} \in \mathbf{R}^M$, it extracts the coordinate of the i th 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 multilayered coordinate transformation. For example, a function mode of a velocity-limit function on the k th real motor is derived as follows:

$${}^f t_{r,PL}^T = [t_1, t_2, \dots, t_M] \begin{cases} t_i = 1, & (i = k) \\ t_i = 0, & \text{otherwise.} \end{cases} \quad (12)$$

Here, ${}^f t_{r,PL}$ 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 k th virtual motor is derived as follows:

$${}^f t_{r,PL} = {}^v t_{rk} \quad (13)$$

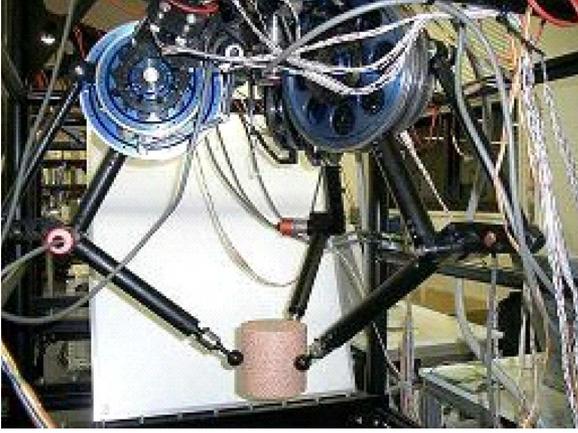


Fig. 7. Parallel-link manipulators.

where ${}^v\mathbf{T}_r = [{}^v\mathbf{t}_{r1}^T, {}^v\mathbf{t}_{r2}^T, \dots, {}^v\mathbf{t}_{rN}^T]^T$. When the k th virtual motor response extracted by ${}^v\mathbf{T}_{r,PL}$ exceeds its limit, a position controller is implemented to the function mode to keep it within the limit value.

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

$${}^f\mathbf{t}_{r,PL} = {}^a\mathbf{t}_{rk} \quad (14)$$

where ${}^a\mathbf{T}_r = [{}^a\mathbf{t}_{r1}^T, {}^a\mathbf{t}_{r2}^T, \dots, {}^a\mathbf{t}_{rN}^T]^T$. In this case, it is assumed that the position limit is set for the k th element of \mathbf{x}_a .

A procedure for exception handling is as follows:

- 1) keep observing variables for detecting exceptions;
- 2) select a relevant performance-limit function when one of the variables exceeds its limit;
- 3) derive ${}^f\mathbf{t}_{r,PL}$, a function mode of the performance-limit function;
- 4) derive ${}^f\mathbf{t}_{r,low}$, a function mode of the lowest priority function;
- 5) derive ${}^f\mathbf{T}_{r,PL}$, the new transformation matrix for performance limit, by substituting ${}^f\mathbf{t}_{r,PL}$ to ${}^f\mathbf{t}_{r,low}$ in ${}^f\mathbf{T}_r$;
- 6) if $\text{rank}(\mathbf{M}_n {}^f\mathbf{T}_{r,PL}) \neq M$, select the function with the next-lowest priority, derive its function mode ${}^f\mathbf{t}_{r,low}$, 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 six 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° , respectively. Absolute position of the arm tip is presented

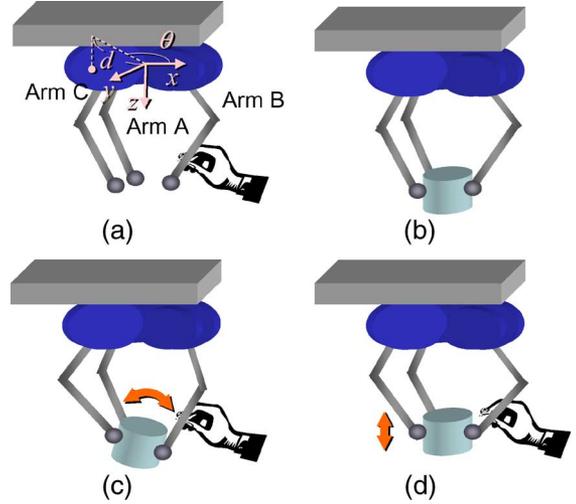


Fig. 8. Illustration of human-support operation. (a) Step 1. (b) Step 2. (c) Step 3. (d) Step 4.

by cylindrical coordinates as shown in

$$\mathbf{x}_{ai} = [d_i, \theta_i, z_i]^T \quad (15)$$

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 paper verifies the validity of the proposed method by an experiment of a human-support operation with task variation. The operation is composed of four steps, as shown in Fig. 8. Each step is described below.

First, 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 the 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 the 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 Steps 1–4 are shown in Table I. The overview of the coordinate transformation is shown in Fig. 9.

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

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)

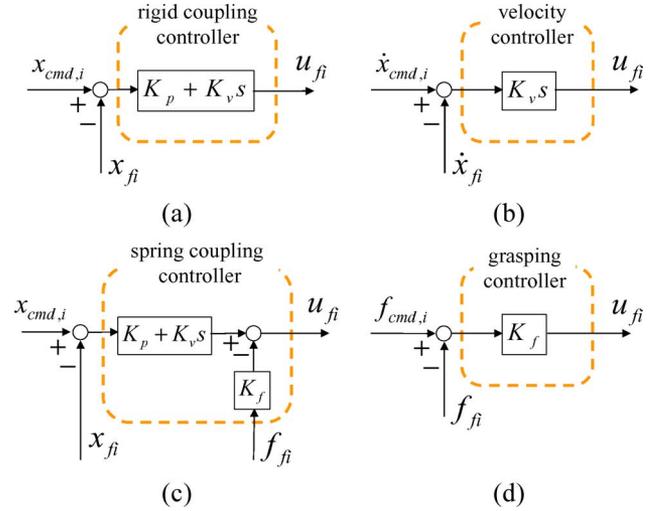


Fig. 10. Block diagram of functions. (a) Rigid-coupling controller. (b) Velocity controller. (c) Spring-coupling controller. (d) Grasping controller.

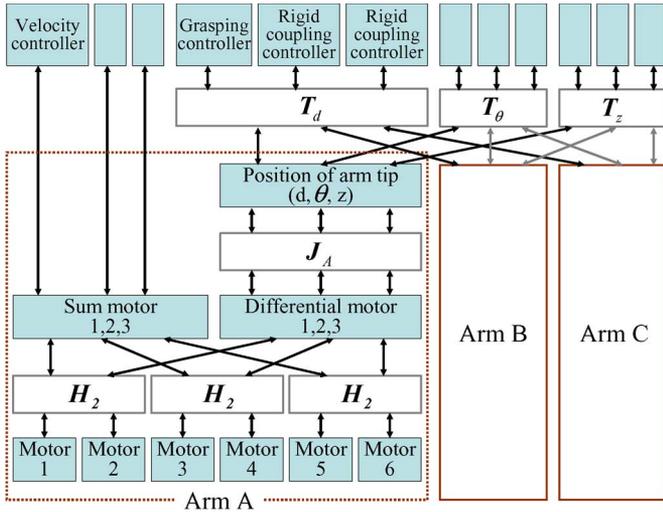


Fig. 9. Overview of the entire coordinate transformation.

The function matrix fT_a for such functions is given as follows:

$${}^fT_a = \begin{bmatrix} I_9 & & & \\ & T_d & & \\ & & T_\theta & \\ & & & T_z \end{bmatrix} {}^fS_a \quad (16)$$

$${}^fS_a = [s_1, s_2, s_3, s_7, s_8, s_9, s_{13}, s_{14}, s_{15}, s_4, s_{10}, s_{16}, s_5, s_{11}, s_{17}, s_6, s_{12}, s_{18}]^T \quad (17)$$

$$s_j = [s_1, s_2, \dots, s_{18}] \quad \begin{cases} s_i = 1, & (i=j) \\ s_i = 0, & \text{otherwise} \end{cases}$$

$$T_d = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \quad (18)$$

$$T_\theta = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \quad (19)$$

$$T_z = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & -1 & 2 \end{bmatrix} \quad (20)$$

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

where fS_a is a permutation matrix to change an order of variables from an arm-based order to a function-based order. I_n , an n th order unit matrix, corresponds to virtual sum motor coordinates. T_d denotes a function matrix in d coordinates while T_θ and T_z denote that in θ and z coordinates. The first, second, and third rows of T_d extract function modes named Mode 1, Mode 2, and Mode 3, respectively. Modes extracted by T_θ and T_z are also named in the same way. Mode 1 is the 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 T_d and T_θ are to derive the difference value of the arm A and others. The second and the third rows of T_z extract the pitching and yawing motions of the object, respectively.

aT_v in this paper is as follows:

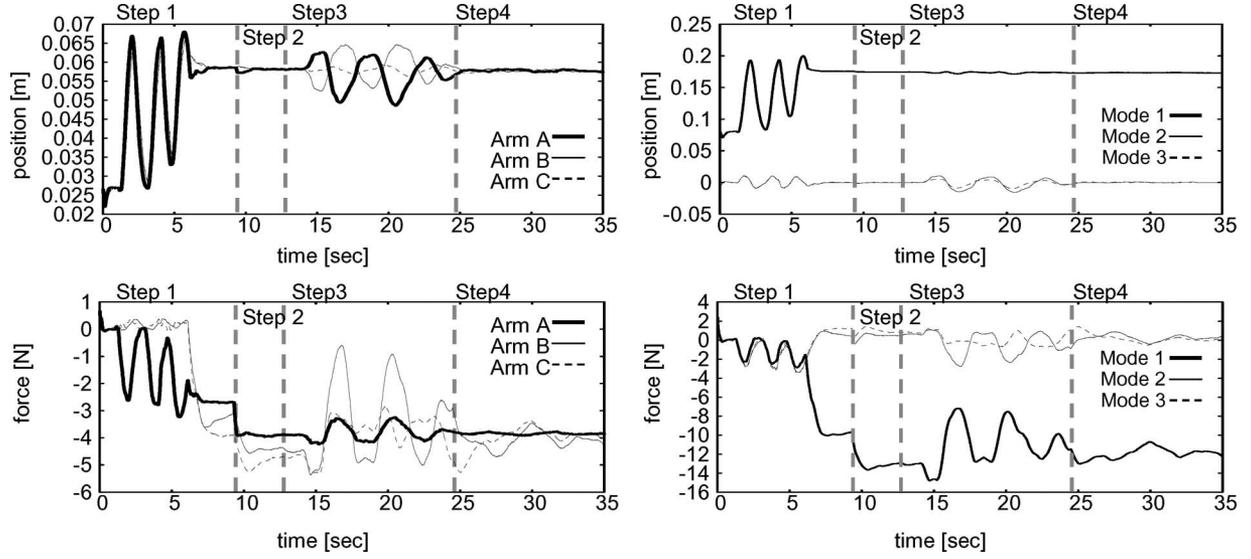
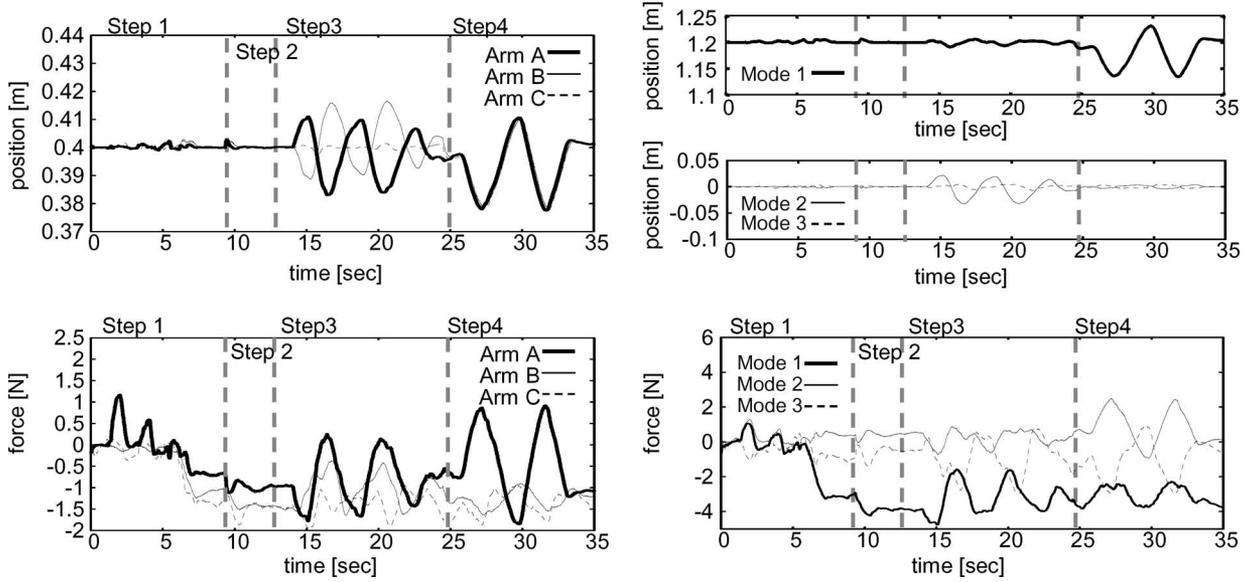
$${}^aT_v = \begin{bmatrix} {}^aT_{vA} & & \\ & {}^aT_{vB} & \\ & & {}^aT_{vC} \end{bmatrix} \quad (21)$$

$${}^aT_{vA} = \begin{bmatrix} I_3 & \\ & J_A \end{bmatrix} \quad (22)$$

$${}^aT_{vB} = \begin{bmatrix} I_3 & \\ & J_B \end{bmatrix} \quad (23)$$

$${}^aT_{vC} = \begin{bmatrix} I_3 & \\ & J_C \end{bmatrix}. \quad (24)$$

Here, J_A , J_B , and J_C denote Jacobian matrices for arms A, B, and C, respectively.


 Fig. 11. Responses in d coordinate.

 Fig. 12. Responses in z coordinate.

vT_r in this paper is as follows:

$${}^vT_r = \begin{bmatrix} {}^vT_{rA} & & \\ & {}^vT_{rB} & \\ & & {}^vT_{rC} \end{bmatrix} \quad (25)$$

$$\begin{aligned} {}^vT_{rA} &= {}^vT_{rB} = {}^vT_{rC} \\ &= {}^vS_r \begin{bmatrix} H_2 & & \\ & H_2 & \\ & & H_2 \end{bmatrix} \end{aligned} \quad (26)$$

$${}^vS_r = [s_1, s_3, s_5, s_2, s_4, s_6]^T \quad (27)$$

$$s_j = [s_1, s_2, \dots, s_6] \quad \begin{cases} s_i = 1, & (i = j) \\ s_i = 0, & \text{otherwise} \end{cases}$$

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (28)$$

where vS_r is a permutation matrix to change an order of variables from real to virtual motors. 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 and in z coordinates, respectively.

When the operator maneuvered 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 were about

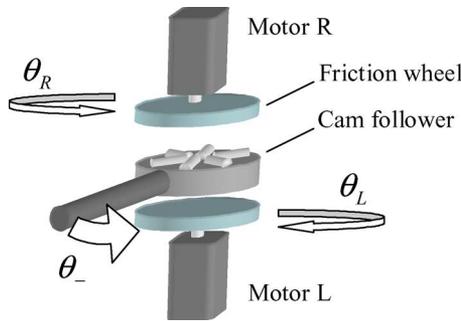


Fig. 13. Mechanism of a twin-drive system.

13 N on average. The average is the grasping force. Grasping motion was retained while a combination of functions were 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 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 1–3 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. Interference between each mode rarely occurred due to acceleration control based on DOB.

VI. CONCLUSION

This paper expanded the framework of function-based controller design to multi-DOF robots in 3-D 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.

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 motors R and L, respectively. The 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 a rotation of the joint. On the other hand, the sum motor coordinate θ_+ do not affect the joint response. The velocity in the 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. Robot. Autom.*, vol. RA-2, no. 1, pp. 14–23, Mar. 1986.
- [2] M. Cossentino, L. Sabatucci, and A. Chella, "A possible approach to the development of robotic multi-agent systems," in *Proc. IEEE/WIC Int. Conf. Intell. Agent Technol.*, 2003, pp. 539–544.
- [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. Jpn. Soc. Mech. Eng., 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 hollonic architecture for easy reconfiguration of robotic assembly systems," *IEEE Trans. Robot. Autom.*, vol. 19, no. 3, pp. 457–464, Jun. 2003.
- [5] Y. Fujimoto and T. Sekiguchi, "Fault-tolerant configuration of distributed discrete controllers," *IEEE Trans. Ind. Electron.*, vol. 50, no. 1, pp. 86–93, Feb. 2003.
- [6] J. E. Hernandez, A. G. Loukianov, B. Castillo-Toledo, and V. I. Utkin, "Observer based decomposition control of linear delayed systems," in *Proc. IEEE Int. Conf. Decision Control*, 2001, pp. 1867–1872.
- [7] S. Arimoto and P. T. A. Nguyen, "Principle of superposition for realizing dexterous pinching motions of a pair of robot fingers with soft-tips," *IEICE Trans. Fundam.*, vol. E84-A, no. 1, pp. 39–47, 2001.
- [8] M. Okada, K. Tatani, and Y. Nakamura, "Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 1410–1415.
- [9] D. Lee and P. Y. Li, "Passive bilateral feedforward control of linear dynamically similar teleoperated manipulators," *IEEE Trans. Robot. Autom.*, vol. 19, no. 3, pp. 443–456, Jun. 2003.
- [10] M. Morisawa and K. Ohnishi, "Motion control taking environmental information into account," *Eur. Power Electron. Drives J.*, 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," in *Proc. IEEE ICIT*, 2003, pp. 802–807.
- [12] S. Katsura, K. Ohnishi, and K. Ohishi, "Transmission of force sensation by environment quarrier based on multilateral control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 898–906, Apr. 2007.
- [13] T. Tsuji, K. Natori, H. Nishi, and K. Ohnishi, "Controller design method of bilateral control system," *Eur. Power Electron. Drives J.*, 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. Ind. Appl.*, vol. 126-D, no. 5, pp. 630–638, 2006.
- [15] C. D. Onal and A. Šabanović, "Bilateral control with a reflex mechanism on the slave side," in *Proc. 31st Annu. Conf. IEEE IECON*, 2005, pp. 195–200.
- [16] K. Ohnishi, M. Shibata, and T. Murakami, "Motion control for advanced mechatronics," *IEEE/ASME Trans. Mechatronics*, vol. 1, no. 1, pp. 56–67, Mar. 1996.
- [17] N. Hayashida, T. Yakoh, T. Murakami, and K. Ohnishi, "A friction compensation in twin drive system," in *Proc. 6th IEEE Int. Workshop Adv. Motion Control*, 2000, pp. 187–192.
- [18] T. Murakami, R. Nakamura, F. Yu, and K. Ohnishi, "Force sensorless compliant control based on reaction force estimation observer in multi-degrees-of-freedom robot," *J. Robot. Soc. Jpn.*, vol. 11, no. 5, pp. 765–768, 1993. (in Japanese).
- [19] M. Mizuochi, T. Tsuji, and K. Ohnishi, "Multirate sampling method for acceleration control system," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1462–1471, Jun. 2007.
- [20] T. Kageyama and K. Ohnishi, "An architecture of decentralized control for multi-degrees of freedom parallel manipulator," in *Proc. 7th IEEE Int. Workshop Adv. Motion Control*, 2002, pp. 74–79.

- [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," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1991, pp. 1921–1926.
- [22] T. Sakuishi, Y. Izumikawa, K. Yubai, and J. Hirai, "Fault-tolerant control of flexible arm based on dual Youla parameter identification," in *Proc. 9th IEEE Int. Workshop Adv. Motion Control*, 2006, pp. 451–455.
- [23] T. Tsuji, K. Ohnishi, and A. Šabanović, "A controller design method based on functionality," in *Proc. 9th IEEE Int. Workshop Adv. Motion Control*, 2006, pp. 171–176.



Toshiaki Tsuji (S'05–M'06) received the B.E., M.E., and Ph.D. degrees from Keio University, Yokohama, Japan, in 2001, 2003, and 2006, respectively.

From 2006 to 2007, he was a Research Associate with the Department of Mechanical Engineering, Tokyo University of Science, Tokyo, Japan. He is currently with the Department of Electrical and Electronic Systems, Saitama University, Saitama, Japan. His research interests include motion control, decentralized control, and control of biped robots.

Dr. Tsuji was the recipient of the FANUC FA and Robot Foundation Original Paper Award in 2007.



Kouhei Ohnishi (S'78–M'80–SM'00–F'01) received the B.E., M.E., and Ph.D. degrees from the University of Tokyo, Tokyo, Japan, in 1975, 1977, and 1980, respectively, all in electrical engineering.

Since 1980, he has been with the Department of System Design Engineering, Keio University, Yokohama, Japan. His research interests include robotics, motion control, and haptics.

Dr. Ohnishi was the recipient of the European Power Electronics and Drive Association–Power Electronics and Motion Control (EPE-PEMC) Council Award and the Dr.-Ing. Eugene Mittelmann Achievement Award in 2004.



Asif Šabanović (M'92–SM'04) received the B.Sc., M.Sc., and Dr.Sc. degrees from the University of Sarajevo, Sarajevo, Bosnia and Herzegovina, in 1970, 1975, and 1979, respectively.

From 1970 to 1991, he was with Energoinvest Institute for Control and Computer Sciences, Sarajevo. He was a Visiting Researcher with the Institute of Control Science, Moscow, Russia, from 1975 to 1976 and a Visiting Professor with the California Institute of Technology, Pasadena, from 1984 to 1985, Keio University, Yokohama, Japan, from 1991 to 1992,

and Yamaguchi University, Yamaguchi, Japan, from 1992 to 1993. He was the Head of CAD/CAM and the Robotics Department, TUBITAK-MRC, Istanbul, Turkey, from 1993 to 1995 and of the Engineering Department, B. H. Engineering and Consulting, Istanbul, Turkey, from 1995 to 1999. Then, he was with the Department of Electrical Engineering, University of Sarajevo. Since 2000, he has been with the Mechatronics Program, Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey. His fields of interest include control systems, motion-control systems, robotics, mechatronics, and power electronics.