Distributed Trajectory Generation for Cooperative Multi-Arm Robots via Virtual Force Interactions

Toshio Tsuji, Achmad Jazidie, and Makoto Kaneko

***Abstract*— A trajectory generation method for multi-arm robots through cooperative and competitive interactions among multiple end-effectors is proposed. The method can generate the trajectories of the multiple arms in a distributed manner based on a concept of a virtual interaction force which represents an interaction between an end-effector and an environment. It is shown that the method is effective not only for simple cooperative tasks such as positioning a common object, but also for more complicated tasks including relative motions among arms.**

I. Introduction

As regards generating a trajectory of a multi-arm robot, many interesting studies have been reported, up to the present, for a purpose of progressing abilities of robotic systems. Lee [1] proposed a method using a manipulability measure, which can deal with a dual-arm robot only. Al-Jarrah and Zheng [2] proposed a method for dual-arm robots handling flexible objects. Moon and Ahmad [3] applied a trajectory time scaling concept to the multi-arm robots, and later, in order to reduce computation time, Moon and Ahmad [4] developed a sub-time-optimal trajectory planning for cooperative multi-arm robots using a load distribution scheme. An algorithm for the time optimal trajectory generation was also proposed by Wang and Pu [5] based on a cell-to-cell mapping method. In the greater part of the previous researches, however, a desired spatial end-effector trajectory of a multi-arm robot for a task is assumed to be given beforehand, and attentions of the investigators have been paid for only simple cooperative tasks such as a pick-and-place motion of a common object.

On the other hand, let us consider a cooperative task between a left hand and a right hand to pare an apple. The left hand tightly holds the apple, and the right hand holds a knife and pares the apple. The left hand should control the position of the apple while the right hand should control a relative motion of the knife to the apple rather than its absolute position in the task space. If the cooperative task illustrated above becomes possible, a need of the multi-arm robots will increase, in particular, in unstructured and hazardous environments such as the space, undersea and nuclear power plants.

So far, only a few studies dedicated for such complicated tasks of the multi-arm robot have been reported. For example, Yamamoto and Mohri [6] proposed a trajectory generation method for a cooperative task of a multi-arm robot, where one arm is grasping and moving an object and others processing the surface of the object. Also, Tsuji [7] has proposed a method utilizing redundant degrees of freedom of a closed link system composed by multiple robotic arms. However, since all these methods generate trajectories based on geometrical constraints of a closed link structure composed by multiple robotic arms, planning a trajectory for each arm requires entire information on movements of all other arms. Thus, a centralized system for planning movements of all arms using a single computer will eventually face problems in terms of failure resistance, flexibility, expandability and so on, as the number of arms or the degrees of freedom of the joints increases.

Manuscript received April 26, 1995; revised August 29, 1996.

T. Tsuji and M. Kaneko are with the Department of Industrial and Systems Engineering, Hiroshima University, Higashi-Hiroshima, 739 Japan (e-mail: tsuji@huis.hiroshima-u.ac.jp).

A. Jazidie is with the Surabaya Institute of Technology, Sukolilo-Surabaya, 60111 Indonesia.

Publisher Item Identifier S 1083-4419(97)05206-0.

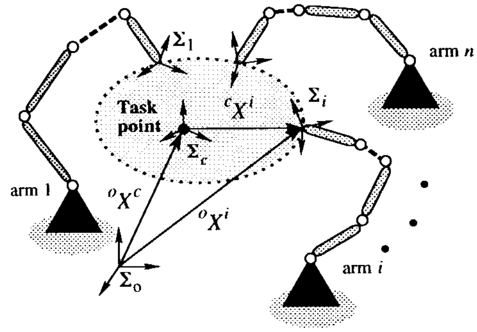


Fig. 1. Coordinate systems for an n-arm robot system performing a cooperative task. The first to n'th arms control the motion of the task point and the $(n'+1)$st to nth arms execute the relative motions between the task point and the end-point of each arm.

One possible approach that can be taken to overcome such problems is to construct an autonomous decentralized control system which is composed of a set of autonomous subsystems in a distributed manner [8], [9]. In this approach, various characteristics may be realized as follows: 1) partial failure in a subsystem can be handled locally, since no central controller exists; 2) by changing local interactions among subsystems, the overall system behavior can be regulated flexibly; and 3) the system structure composed by autonomous subsystems allows an easy expansion of the system without re-planning of the motion of the entire system.

Recently, a variety of control systems based on the concept of the autonomous decentralized system have been proposed (for example, [10] and [11]). However, these methods attempt to control a mobile robot system or a single-arm robot, not a multi-arm robot, by decentralizing it into a set of subsystems.

In the present paper, a new method for multi-arm robots is developed, which can generate trajectories of the multiple arms in a parallel and distributed manner through cooperative interactions among subsystems corresponding to each arm. In order to express information exchanging among subsystems, virtual dynamics are imagined for each manipulator, and virtual interaction forces arising from these virtual dynamics are used. Under the proposed method, it is possible to deal with not only simple cooperative tasks such as positioning a grasped object but also more complicated cooperative tasks including relative motions among the arms.

II. Basic Formulation of Multi-Arm Robot

A. Cooperative Task of Multi-Arm Robot

Let us consider a cooperative task of a multi-arm robot, which consists of two sub-tasks: a) holding and moving a common object and b) processing a surface of the object (see Fig. 1). A group of arms (the first to the n'th arms) control the motion of the object in the task space, while the other group of arms (the $(n'+1)$st to the nth arms) are processing the surface of the object. The second group of arms have to control relative positions of the end-effectors to the object rather than the absolute positions in the task space.

Now, we define a reference point of the object (e.g., the center of mass of the object) as the task point. Using the task point, the motion of multiple arms performing a cooperative task can be described as a) motion of the task point and b) relative motion between each arm and the task point. In motion planning of the task point, it is assumed that a virtual rigid link [12] is connected between the task point and the end-effector of each arm. Also, no slip motion between the end-effector of each arm and the object is assumed. On the other hand, the relative motion between each arm and the task point is used to represent more complicated tasks such as grinding and scraping the object. The relative motions among arms are expressed by combinations of relative motions between the end-effector of each arm and the task point.

B. Coordinate Systems

The joint degrees of freedom of each arm is denoted by m_i $(i = 1, 2, \cdots, n)$ where n is the number of arms, and the task space dimension is denoted by l. In the present paper, three different Cartesian coordinate systems are defined: 1) the world coordinate system, Σ_o; 2) the task coordinate system, Σ_c, which is a mobile coordinate system according to the motion of the task point; and 3) the end-point coordinate system, Σ_i $(i = 1, \cdots, n)$, the origin of which is located on the end-point of the arm i. Let the position and orientation vector of the task coordinate system, Σ_c, and of the end-point coordinate system, Σ_i, with respect to the world coordinate system, Σ_o, be denoted as

$$ {}^oX^c = [{}^op^{c^T}, {}^o\Phi^{c^T}]^T \in \Re^l $$

and

$$ {}^oX^i = [{}^op^{i^T}, {}^o\Phi^{i^T}]^T \in \Re^l \qquad i = 1, \cdots, n $$

respectively. Let also the position and orientation vector of the end-point coordinate system, Σ_i, represented in the task coordinate system, Σ_c, be denoted as

$$ {}^cX^i = [{}^cp^{i^T}, {}^c\Phi^{i^T}]^T \in \Re^l \qquad i = 1, \cdots, n $$

as shown in Fig. 1. Then, the position and orientation of the task point, ${}^oX^c$, is uniquely computed from ${}^oX^i$ and ${}^cX^i$.

In the three-dimensional (3-D) space $(l = 6)$, for example, the relationship among the position vectors ${}^op^c$, ${}^op^i$ and ${}^cp^i$ is given as follows:

$$ {}^op^c = {}^op^i - {}^oR_i({}^o\Phi^i)[{}^cR_i({}^c\Phi^i)]^T {}^cp^i \tag{1} $$

where the rotation matrices from Σ_i to Σ_o and from Σ_i to Σ_c are denoted as ${}^oR_i({}^o\Phi^i)$ and ${}^cR_i({}^c\Phi^i)$, respectively.

Also, the use of the Euler angle $\Phi = [\phi, \theta, \psi]^T$ for each orientation vector leads to the following expression for the rotational matrix ${}^oR_c({}^o\Phi^c)$ from Σ_c to Σ_o [13]

$$ \begin{aligned} {}^oR_c({}^o\Phi^c) &= {}^oR_i({}^o\Phi^i)[{}^cR_i({}^c\Phi^i)]^T \\ &= \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}. \end{aligned} \tag{2} $$

Then, based on the property of the Euler angle, the orientation vector ${}^o\Phi^c = [{}^o\phi^c, {}^o\theta^c, {}^o\psi^c]^T$ can be obtained in the following way.

a) When $\sin {}^o\theta^c \neq 0$,

$$ {}^o\phi^c = \operatorname{atan2}(\pm R_{23}, \pm R_{13}), \tag{3} $$

$$ {}^o\theta^c = \operatorname{atan2}(\pm\sqrt{R_{13}^2 + R_{23}^2}, R_{33}), \tag{4} $$

$$ {}^o\psi^c = \operatorname{atan2}(\pm R_{32}, \mp R_{31}). \tag{5} $$

b) When $\sin {}^o\theta^c = 0$,

$$ {}^o\phi^c = \text{arbitrary} \tag{6} $$

$$ {}^o\theta^c = \frac{\pi}{2}(1 - R_{33}) \tag{7} $$

$$ {}^o\psi^c = \operatorname{atan2}(R_{21}, R_{22}) - R_{33}\,{}^o\phi^c. \tag{8} $$

C. Kinematics of Multi-Arm Robot

As is well-known, the relationship between the end-point velocity vector, ${}^o\dot{X}^i$, of the arm i and the joint angular velocity vector, $\dot{q}^i \in \Re^{m^i}$, is given by

$$ {}^o\dot{X}^i = J^i\dot{q}^i \tag{9} $$

where $J^i \in \Re^{l \times m^i}$ is the Jacobian matrix of the arm i.

On the other hand, assuming that a virtual rigid link is connected between the task point and the end-point of the arm i, we can derive the relationships between the end-point of the arm i and the task point as follows:

$$ {}^oF^{ci} = G^iH^{io}F^i = G^{io}F^{\text{tri}} \tag{10} $$

$$ {}^o\dot{X}^{\text{tri}} = H^{io}\dot{X}^i = G^{i^T o}\dot{X}^c \tag{11} $$

where ${}^oF^i \in \Re^l$ expresses the force/moment vector of the end-point of the arm i represented in the world coordinate system, Σ_o; ${}^o\dot{X}^c$ and ${}^oF^{ci} \in \Re^l$ represent the velocity vector of the task point and the force/moment vector transmitted to the task point from the end-point of the arm i, represented in the world coordinate system, Σ_o, respectively. The contact type matrix $H^i \in \Re^{l_i \times l}$ and the matrix $G^i = S^iH^{i^T} \in \Re^{l \times l_i}$ express filtering effects that filter out some forces/moments of the end-point of the arm i and transmit other forces/moments to the task point depending on the contact mechanism. Also, the end-point velocity vector of the arm i transmitted from the task point is denoted as ${}^o\dot{X}^{\text{tri}} \in \Re^{l_i}$, and the vector of the forces/moments transmitted from the end-point of the arm i to the object as ${}^oF^{\text{tri}} \in \Re^{l_i}$ [14], [15], where l_i is the number of the degrees of freedom of the forces/moments that can be transmitted from the end-point of the arm i to the object. The matrix $S^i \in \Re^{l \times l}$ expresses the geometrical relationship between the task point and the end-point of the arm i, which is given by

$$ S^i = \begin{bmatrix} I & 0 \\ ({}^cp^i)_{\Sigma_o}\chi & I \end{bmatrix} \tag{12} $$

where I is the unit matrix; 0 is a zero matrix; and $({}^cp^i)_{\Sigma_o}$ is ${}^cp^i$ represented in Σ_o. In addition, χ is the cross operator that satisfies $(a\chi)u = a \times u$ for any vector a and u. When $a = [a, b, c]^T$, it is defined as [16]

$$ a\chi = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}. \tag{13} $$

Since the force/moment vector ${}^oF^c$ acting on the task point is the total sum of all force/moment vectors transmitted from the end-points of the arms to the task point, the net force/moment vector ${}^oF^c$ is obtained by

$$ {}^oF^c = \sum_{i=1}^{n} {}^oF^{ci}. \tag{14} $$

The kinematic relationships are summarized in Fig. 2. Based on the above formulations, a trajectory generation method of the end-effectors for multiple arms is explained in the next section.

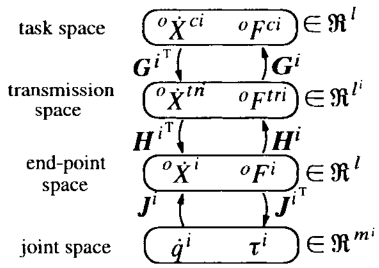


Fig. 2. Kinematic relationship of motion and force of arm i.

III. Distributed Trajectory Generation Based on Virtual Dynamics

The method proposed here involves a set of subsystems corresponding to each arm and can generate joint trajectories that satisfy kinematic constraints of the end-effector of each arm in a parallel and distributed manner through cooperative and competitive interactions among subsystems. In order to express the interactions among subsystems appropriately, virtual dynamics of each arm and the task point are introduced. Then virtual interaction forces/moments generated from the virtual dynamics and position constraints resulting from kinematic relationships between each arm and the task point are derived through analogy of mechanical systems to represent the interactions among subsystems.

A. Structure and Motion Equation of Subsystems

First of all, the virtual dynamics of the arm i is defined using the simplest second order differential equation as follows:

$$\ddot{q}^i = \tau^i + (H^i J^i)^T \lambda^i \tag{15}$$

where $\tau^i \in \Re^{m^i}$ is the virtual joint control torque vector of the arm i; and $\lambda^i \in \Re^{l_i}$ is the force/moment vector acting on the end-point of the arm i from the object represented in the world coordinate system, Σ_o.

Then, let us consider the motion of the task point. Since the virtual force/moment vector λ^i is exerted to the end-effector of the arm i from the object, a reacting virtual force/moment vector $-\lambda^i$ is exerted conversely to the object from each end-point. As a result, a net virtual force/moment vector acting on the object at the task point is given by

$${}^oF^c = \sum_{k=1}^{n} {}^oF^{ci} = \sum_{k=1}^{n} (-G^i \lambda^i). \tag{16}$$

On the other hand, virtual dynamics of the task point is assumed as

$$M_c {}^o\ddot{X}^c = {}^oF^c \tag{17}$$

where $M_c \in \Re^{l \times l}$ is interpreted as the virtual inertia matrix of the object; ${}^o\ddot{X}^c \in \Re^l$ is the acceleration vector of the task point represented in the world coordinate system, Σ_o.

Now, let us consider position constraints imposed on the end-point of each arm. The arms i $(i = 1, \cdots, n')$ must be constrained by the motion of the task point determined by the dynamics of equation (17). In other words, equation (11) leads to the relationship

$${}^o\ddot{X}^{\mathrm{tri}} = \dot{G}^{i^T} {}^o\dot{X}^c + G^{i^T} {}^o\ddot{X}^c. \tag{18}$$

On the other hand, in order to derive the kinematic constraints for the arm i $(i = n'+1, \cdots, n)$, we have to consider not only the motion of the task point but also the relative motions between the end-effector of each arm and the task point, ${}^cX^i \in \Re^l$, which are assumed to be given as the desired motion. Since we can denote as $l_i = l$ and $H^i = I_l$ (an $l \times l$ unit matrix) for arm i $(i = n'+1, \cdots, n)$, the end-point acceleration vector in the transmission space, ${}^o\ddot{X}^{\mathrm{tri}}$, can be written as

$$\begin{aligned} {}^o\ddot{X}^{\mathrm{tri}} = {}^o\ddot{X}^c &+ \begin{bmatrix} {}^oR_c & 0 \\ 0 & {}^oR_c \end{bmatrix} {}^c\ddot{X}^i + \begin{bmatrix} 2{}^o\dot{R}_c & 0 \\ 0 & {}^o\dot{R}_c \end{bmatrix} {}^c\dot{X}^i \\ &+ \begin{bmatrix} {}^o\ddot{R}_c & 0 \\ 0 & 0 \end{bmatrix} {}^cX^i. \end{aligned} \tag{19}$$

The end-effector acceleration vector ${}^o\ddot{X}^{\mathrm{tri}}$ computed by (18) and (19) must agree with the end-effector acceleration determined by the joint motion of the arms

$${}^o\ddot{X}^{\mathrm{tri}} = H^i \dot{J}^i \dot{q}^i + H^i J^i \ddot{q}^i. \tag{20}$$

Consequently, the joint acceleration of the arm i and the virtual interaction force/moment vector λ^i can be obtained using (15) and (20)

$$\begin{bmatrix} \ddot{q}^i \\ \lambda^i \end{bmatrix} = \begin{bmatrix} I_{m_i} & -(H^i J^i)^T \\ H^i J^i & 0 \end{bmatrix}^{-1} \begin{bmatrix} \tau^i \\ H^i \dot{J}^i \dot{q}^i - {}^o\ddot{X}^{\mathrm{tri}} \end{bmatrix} \tag{21}$$

where I_{m_i} is an $m_i \times m_i$ unit matrix. The resulted joint acceleration in turn generates the end-effector trajectory of each arm.

Now, the virtual joint control torque τ^i is computed using the target position of the task point, ${}^oX^{c^*}$, as follows:

$$\tau^i = (H^i J^i)^T (G^i)^+ \{K^i({}^oX^{c^*} - {}^oX^c)\} - B^i \dot{q}^i \tag{22}$$

where

$$(G^i)^+ = (G^{i^T} G^i)^{-1} G^{i^T} = H^i (S^i)^{-1} \in \Re^{l_i \times l}$$

is the pseudo-inverse matrix of G^i; $K^i \in \Re^{l \times l}$ is a positive definite position feedback gain matrix for the arm i; and $B^i \in \Re^{m_i \times m_i}$ is the positive definite velocity feedback gain matrix of the arm i. For the ith arm that is executing the relative motion $(i = n'+1, \cdots, n)$, we can easily show $G^i = 0$ and $(G^i)^+ = 0$. Then, the virtual joint control torque for the ith arm $(i = n'+1, \cdots, n)$ is reduced to

$$\tau^i = -B^i \dot{q}^i. \tag{23}$$

The trajectory generation method proposed here is illustrated in Fig. 3. Each subsystem generates a trajectory cooperatively using the virtual end-point force/moment vector λ^i as the information of the interactions via the virtual dynamics of the task point. Since each subsystem can operate independently, it is not necessary to modify the motion equations of all subsystem if the purpose of the arm changes from the task of holding and moving the object to the task including a relative motion, or if a new arm is added to the system. In the following section, stability of the system and the kinematic property of the equilibrium point is analyzed.

B. Stability Analysis

Let us consider an energy function H that is composed of two types of energy functions H_1 and H_2 as follows:

$$H = H_1 + H_2, \tag{24}$$

$$H_1 = \sum_{i=1}^{n'} E^i + Q_c + \tfrac{1}{2} \sum_{i=1}^{n'} \dot{q}^{i^T} \dot{q}^i \tag{25}$$

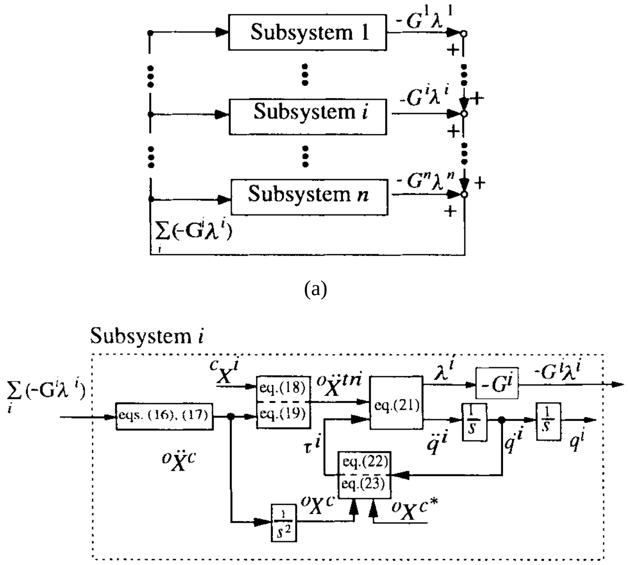


Fig. 3. Distributed trajectory generation for a multi-arm robot.

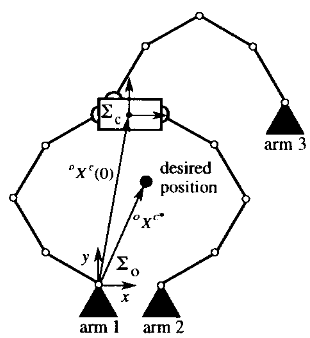


Fig. 4. Model of a three-arm robot and its initial posture used in simulations. The task coordinate system and the desired position of the task point are shown.

$$H_2 = \tfrac{1}{2} \sum_{i=n'+1}^{n} \dot{q}^{i^T} \dot{q} \tag{26}$$

$$E^i = \tfrac{1}{2} ({}^oX^{c^*} - {}^oX^c)^T K^i ({}^oX^{c^*} - {}^oX^c) \tag{27}$$

$$Q_c = \tfrac{1}{2} {}^o\dot{X}^{c^T} M_c {}^o\dot{X}^c. \tag{28}$$

H_1 and H_2 are energy functions for the motion of the task point and the relative motion between each arm i $(i = n' + 1, \cdots, n)$ and the task point, respectively; E^i represents the squared position error between the target and current positions of the task point calculated at each subsystem i $(i = 1, 2, \cdots, n')$; and Q_c expresses the virtual kinetic energy of the task point.

Now, let us consider the energy function for the motion of the task point, H_1. The time derivative of the energy function, $\dot{H}_1$, can be derived as

$$\dot{H}_1 = \sum_{i=1}^{n'} \dot{E}^i + \dot{Q}_c + \sum_{i=1}^{n'} \left[\dot{q}^{i^T} \ddot{q}^i \right] \tag{29}$$

$$\dot{E}^i = -({}^o\dot{X}^c)^T K^i ({}^oX^{c^*} - {}^oX^c) \tag{30}$$

$$\dot{Q}_c = {}^o\dot{X}^{c^T} M_c {}^o\ddot{X}^c. \tag{31}$$

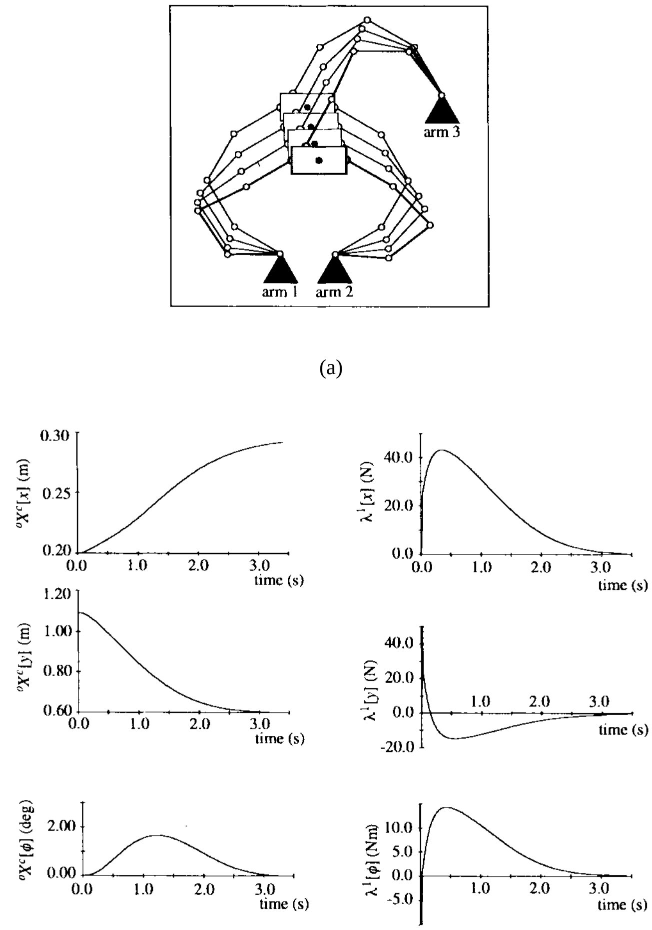


Fig. 5. An example of the generated trajectory for positioning the task point. (a) Stick pictures, (b) time history of the position of the task point, and (c) time history of the virtual interaction forces/moments of the arm 1.

Substituting (15)–(17) and (22) into (29)–(31), the following equation can be obtained:

$$\dot{H}_1 = -{}^o\dot{X}^{c^T} \sum_{i=n'+1}^{n} G^i \lambda^i - \sum_{i=1}^{n'} \dot{q}^{i^T} B^i \dot{q}^i. \tag{32}$$

On the other hand, the time derivative of the energy function, $\dot{H}_2$, can be given by

$$\dot{H}_2 = \sum_{i=n'+1}^{n} [-\dot{q}^{i^T} B^i \dot{q}^i + {}^o\dot{X}^{\mathrm{tri}} \lambda^i]. \tag{33}$$

As a result, from (32) and (33) the time derivative of the energy function of the whole system can be obtained as

$$\dot{H} = -\sum_{i=1}^{n} \dot{q}^{i^T} B^i \dot{q}^i. \tag{34}$$

Since B^i is of positive definite, we have $\dot{H} \leq 0$ and the energy function H decreases monotonically until $\dot{H} = 0$, i.e., $\dot{q}^i = 0$ $(i = 1, 2, \cdots, n)$ and ${}^o\dot{X}^c = 0$. This means that the motion of the whole system is asymptotically stable.

C. Kinematic Property of the Equilibrium Point

Kinematic meaning of the equilibrium point of the energy function, H, is analyzed under an assumption that the relative motions between the end-points and the task point, ${}^cX^i$ $(i = n'+1, \cdots, n)$, are given as the desired motion satisfying ${}^c\dot{X}^i = {}^c\ddot{X}^i = 0$ at a certain time, t_r, where t_r is the time duration required for relative motions. It means that, for $t > t_r$, there is no relative motion between end-points and the task point, and therefore, $\lambda^i = 0$ $(i = n'+1, \cdots, n)$.

Since the task point is at a standstill in the equilibrium point, using (16) and (17) and $\lambda^i = 0$ $(i = n'+1, \cdots, n)$, we have

$$-\sum_{i=1}^{n'} G^i \lambda^i = 0. \tag{35}$$

Substituting (15) and (22) into (35) and remembering that at the equilibrium position, $\dot{q}^i = \ddot{q}^i = 0$ and ${}^o\dot{X}^c = 0$, the following equation can be obtained:

$$\left[\sum_{i=1}^{n'} G^i (G^i)^+ K^i\right] ({}^oX^{c^*} - {}^oX^c) = 0 \tag{36}$$

where $G^i(G^i)^+$ is a positive semi-definite matrix and K^i is a positive definite matrix from its definition. Provided that all forces/moments acting on the task point can be controlled using the joint torque of the arm i $(i = 1, 2, \cdots, n')$ and n' is sufficiently large, the matrix of $\Sigma_{i=1}^{n'} G^i(G^i)^+ K^i$ in (36) may be expected to be a nonsingular. For example, if we define that $K^i = K$ $(i = 1, 2, \cdots, n')$, then (36) reduces to

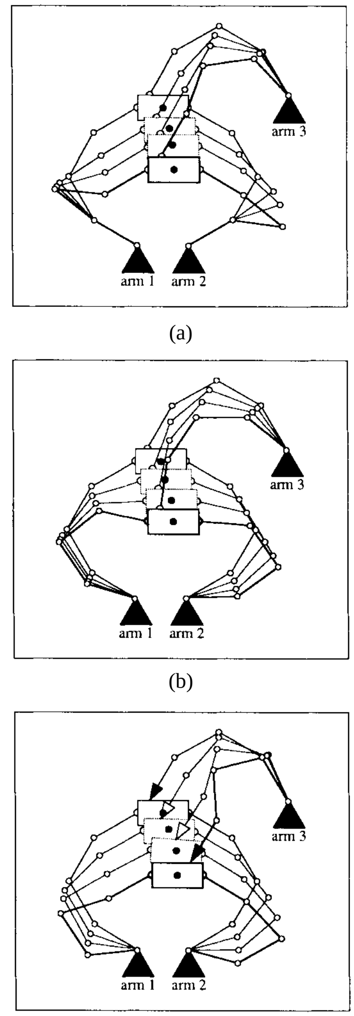
$$\left[\sum_{i=1}^{n'} G^i (G^i)^+\right] K({}^oX^{c^*} - {}^oX^c) = 0. \tag{37}$$

Assuming that at least one manipulator, for example arm k, is connected to the object through the rigid grasping contact type, i.e., $l_k = l$, we have $G^k(G^k)^+ = I_l$ (an $l \times l$ unit matrix). Consequently the matrix of $\Sigma_{i=1}^{n'} G^i(G^i)^+$ in (37) is assured to be nonsingular. As a result, the solution of (37) becomes ${}^oX^c = {}^oX^{c^*}$, and the equilibrium point of the task point agrees with the corresponding target position. If there is an uncontrollable force/moment of the task point, (36) becomes indefinite. Therefore the equilibrium point does not always agree with the target position. Even in this case, stability of the multi-arm robot system can be guaranteed as described in the previous section.

Summing up, the distributed trajectory generation method for multi-arm robots has been explained in this section. Also, the stability of the motion of the whole system and the characteristics of the equilibrium point of the task point have been analyzed. In the next section, effectiveness of the proposed method will be verified by simulation experiments.

IV. Simulation Experiments

Computer simulations were carried out using a three-arm planar robot (Fig. 4). Each arm is a four-joint type, and lengths of all links are set to 0.4 (m). The task point is defined as the center of gravity of the object, which is the origin of the task coordinate system (see Fig. 4). The parameters used for the simulations are follows: the position feedback gains $K^i = \text{diag.}\,[100(\text{N/m}), 100(\text{N/m}), 100(\text{N/rad})]$; the velocity feedback gains $B^i = \text{diag.}\,[10, 10, 10, 10](\text{Nm/(rad/s)})$ for $i = 1, 2, 3$; and $M_c = \text{diag.}\,[50(\text{kg}), 50(\text{kg}), 50(\text{kgm}^2)]$. Note that diag. $[\cdot]$ denotes a diagonal matrix.



(c)

Fig. 6. Results of trajectory generations. (a) The first joints of arms 1 and 2 are fixed, (b) end-effector of each arm is connected to the object through a point contact, and (c) end-effector of arm 3 performs a relative motion along the surface of the object.

Fig. 5 shows examples of the results where the position of the task point is moved to the target position from the initial position indicated in Fig. 4. Fig. 5(a) shows stick pictures, while Fig. 5(b) expresses the time history of the position of the task point, and Fig. 5(c) gives the time history of the virtual interaction forces/moment generated between the arm 1 and the object. In this case, all of the arms control the motion of the task point and all arms grasp the object rigidly, $l_i = 3$ for $i = 1, 2, 3$.

On the other hand, Fig. 6 shows other results using the same initial arm posture and the target position of the task point as Fig. 5. Fig. 6(a) indicates the simulation result for the case in which the first joints of the arm 1 and arm 2 are fixed. In Fig. 6(b), the end-point of each arm is connected to the object through the point-contact, i.e., the end-point forces can be transmitted in any direction to the object, but the end-point moments cannot be transmitted $(l_i = 2$ for $i = 1, 2, 3)$. In Fig. 6(c), the end-point of the arm 3 performs a relative motion respect to the task point along the surface of the object. From Fig. 6, it can be seen that in every case, the position of the task point reaches the target position, whereas the intermediate trajectories and final postures are different considerably.

Since two joints were fixed in Fig. 6(a), both of the arms 1 and 2 moved using only three joints each. The proposed method can easily deal with such constraints, because fixing of a joint is handled within the subsystem locally and does not directly affect other subsystems.

In Fig. 5(a), the angle between the end-point of each arm and the object is kept constant. On the other hand, they changes during motion in Fig. 6(b), since the end-point moment cannot be transmitted to the object, i.e., the end-point can rotate freely. The method can generate the trajectories under various contact mechanism between the end-points of the arms and the object.

In Fig. 6(c), the arms 1 and 2 control the task point, and the end-point of the arm 3 carried out a relative motion with respect to the task point. The relative motion of the end-effector of the arm 3 is given as a function of the time as follows:

$$ {}^{c}X^{3}(t) = \begin{cases} [0.1t^2 - 0.1(\mathrm{m}), 0.1(\mathrm{m}), \frac{4}{3}\,\pi(\mathrm{rad})]^T \\ \qquad \text{if } 0 \le t \le 1 \\ [-0.1t^2 + 0.4t - 0.3(\mathrm{m}), 0.1(\mathrm{m}), \frac{4}{3}\,\pi(\mathrm{rad})]^T \\ \qquad \text{if } 1 \le t \le 2 \\ [0.1(\mathrm{m}), 0.1(\mathrm{m}), \frac{4}{3}\,\pi(\mathrm{rad})]^T \\ \qquad \text{if } t \ge 2 \end{cases}. \tag{38} $$

It can be seen from Fig. 6(c) that the cooperative motion can be realized maintaining the closed link structure.

V. Conclusion

This paper has proposed the new trajectory generation method for multi-arm robots using the concept of the virtual dynamics. This method is based on an effort to express interactions among the arms by using the virtual forces/moments transmitted from each arm to the task point, so that the method can generate the trajectories of the multiple arms in a parallel and distributed manner through cooperation and competition among subsystems corresponding to each arm. In fact, it should be noted that for multiple robots involved in coordination, the proposed method is not the only one which can treat individual robots in a distributed manner. The conventional dynamic analysis will also lead to the generation of joint trajectories of all the manipulators. In comparison with other approaches, the advantages of the proposed method are summarized as follows.

1) Using the virtual dynamics, the mobility of the object depending on the direction of the motion can be regulated. Also, if a virtual inertia of each arm is included on the basis of the concept of the virtual dynamics, the mobility of each arm relative to other arms and the object can be set according to the given task.
2) The subsystem corresponding to each arm can work independently, so that the advantages of the autonomous decentralized system in terms of failure resistance, expandability and parallel computation are also held with the proposed method.
3) The relative motion between the end-point of each arm and the task point can be given as the desired trajectory which are treated as the constraint on manipulator motion.

References

[1] S. Lee, "Dual redundant arm configuration optimization with task oriented dual arm manipulability," *IEEE Trans. Robot. Automat.*, vol. 5, no. 1, pp. 78–97, 1989.

[2] O. Al-Jarrah and Y. F. Zheng, "Efficient trajectory planning for two manipulators to deform flexible material," in *Proc. 1994 Int. Conf. Intelligent Robots Systems*, 1994, vol. 1, pp. 1056–1063.

[3] S. B. Moon and S. Ahmad, "Time-scaling of cooperative multirobot trajectories," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 900–908, Apr. 1991.

[4] ——, "Sub-time-optimal trajectory planning for cooperative multi-manipulator systems using the load distribution scheme," in *Proc. 1993 IEEE Int. Conf. Robotics Automation*, 1993, vol. 1, pp. 1037–1042.

[5] F. Y. Wang and B. Pu, "Planning time-optimal trajectory for coordinated robot arms," in *Proc. 1993 IEEE Int. Conf. Robotics Automation*, vol. 1, pp. 245–250.

[6] M. Yamamoto and A. Mohri, "A study for coordinated tasks of multi-robot system—problem statement and task planning for two-robot system," in *Proc. 9th Annu. Conf. Robotics Soc. Japan*, 1991, pp. 489–492, (in Japanese).

[7] T. Tsuji, "Trajectory generation of a multi-arm robot utilizing kinematic redundancy," *J. Robot. Mechatronics*, vol. 5, no. 6, pp. 601–605, 1993.

[8] T. Fukuda, S. Kawauchi, and H. Asama, "Analysis and evaluation of cellular robotics (cebot) as a distributed intelligent system by communication information amount," in *Proc. 1990 IEEE/RSJ Int. Workshop Intelligent Robots Systems*, 1990, pp. 827–834.

[9] R. C. Arkin, T. Balch, and E. Nitz, "Communication of behavioral state in multi-agent retrieval tasks," in *Proc. 1993 IEEE Int. Conf. Robotics Automation*, 1993, vol. 3, pp. 588–594.

[10] H. Asama, M. K. Habib, and I. Endo, "Functional distribution among multiple mobile robots in an autonomous and decentralized robot systems," in *Proc. 1991 IEEE Int. Conf. Robotics Automation*, pp. 1921–1926.

[11] T. Tsuji, S. Nakayama, and K. Ito, "Distributed feedback control for redundant manipulators based on virtual arms," in *Proc. Int. Symp. Autonomous Decentralized Systems*, 1993, pp. 143–149.

[12] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for coordination of two robots," in *Proc. 1988 IEEE Int. Conf. Robotics Automation*, pp. 350–356.

[13] K. S. Fu, R. C. Gonzales, and S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill, 1987.

[14] M. R. Cutkosky and I. Kao, "Computing and controlling the compliance of a robotics hand," *IEEE Trans. Robot. Automat.*, vol. 5, no. 2, pp. 151–165, 1989.

[15] M. T. Mason and J. K. Salisbury, *Robot Hands and the Mechanics of Manipulation*. Cambridge, MA: MIT Press, 1985.

[16] R. Featherstone, *Robot Dynamics Algorithm*. Norwell, MA: Kluwer, 1987.