MECATRONICS

2eme Congrès FRANCO-JAPONAIS de MECATRONIQUE

第２回メカトロニクスに関する国際会議

Proceedings of the

2nd JAPAN-FRANCE CONGRESS on MECHATRONICS

(INTERNATIONAL)

Volume 2

November 1-3, 1994

TAKAMATSU, KAGAWA
Japan

Co-organized by
KAGAWA UNIVERSITY
JAPAN SOCIETY for PRECISION ENGINEERING
L'INSTITUT des MICROTECHNIQUES de FRANCHE-COMTE

in cooperation with
JSME,IEEJ,RSJ,SICE,IEICE,IPSJ,IFToMM-J
AFRI,IEEE(RAS)

Supported by
MINISTRY of INTERNATIONAL TRADE AND INDUSTRY of JAPAN
CNRS(FRANCE)
KAGAWA-PREFECTURAL GOVERNMENT, CITY of TAKAMATSU
DEPARTMENT du DOUBS et REGION de FRANCHE-COMTE(FRANCE)

Distributed Trajectory Generation of A Multi-Arm Robot Performing Cooperative Tasks

Achmad JAZIDIE, Toshio TSUJI, Makoto KANEKO and Mitsuo NAGAMACHI

Faculty of Engineering, Hiroshima University
1-4-1, Kagamiyama, Higashi-Hiroshima, 724 Japan

Abstract In this paper, a method for generating the end-points' trajectories in a parallel and distributed manner through the cooperation of subsystems corresponding to each arm is proposed. The method involves the dynamic modeling of multi-arm robots and an effort to carry out exchanges of information among subsystems by means of interactive forces/moments generated by the dynamics of each arm and the object. The method can deal not only with simple tasks such as positioning a grasped object but also with more complicated cooperative tasks including relative motions among the arms. Simulation experiments are carried out to show the effectiveness of the proposed method.

Key Words Multi-arm systems, distributed control, interactive forces, trajectory generation, cooperative tasks.

1 Introduction

Up to the present, there are many researches have been done in developing trajectory generation methods for multi-arm robots. Since all these methods generate trajectories based on conditions of geometrical constraints of a closed link structure composed by multiple arms, the planning of a trajectory for each arm can only be carried out if the information on the actions of all the other arms is provided. Thus, the centralized system of the motion planning of all the arms by means of a single computer will eventually face problems in terms of failure resistance, flexibility, expandability, etc., as the number of arms increases.

One of the possible approach that can be taken to overcome such problems is to construct an autonomous decentralized control system which is composed by a set of autonomous subsystem in a distributed manner [1]. Recently, a variety of control systems based on the concept of this autonomous decentralized system have been proposed [2], [3]. These methods attempt to control a mobile robot system or a single-arm robot by decentralizing it into subsystems, but do not deal with multi-arm robots.

In the present paper, a new trajectory generation method for multi-arm robots is proposed in a parallel and distributed way through cooperative interactions among subsystems corresponding to each arm. According to the proposed method, the interaction forces between arms arising from dynamics of each arm through the object are used to exchange information among subsystems. 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 containing relative motions among the arms.

2 Formulation and Kinematics of Multi-Arm Robots

2.1 Formulation of Multi-Arm Robots

Let's now consider an n-arm robot performing a task as illustrated in **Fig.1**. The joint degrees of freedom of each arm is denoted by m^i (i=1,...,n) and the task space dimension is expressed by l. Then a single task point according to the objective of the task is defined.

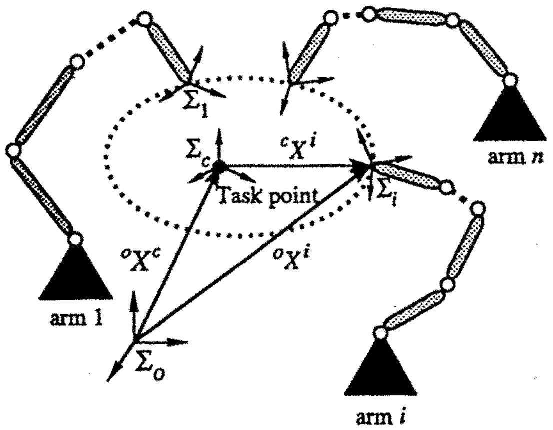

Fig.1 An n-arm robot performing a cooperative task

Here, three different coordinate systems are defined: 1) the base coordinate system Σ_o, 2) the task coordinate system Σ_c having its origin at the position of the task point, and 3) the end-point coordinate system Σ_i, i=1,...,n. The position and orientation of the task point ${}^oX^c = [{}^op^{cT}\ {}^o\Phi^{cT}]^T \in \Re^l$ can be determined uniquely from ${}^oX^i = [{}^op^{iT}\ {}^o\Phi^{iT}]^T \in \Re^l$ and ${}^cX^i = [{}^cp^{iT}\ {}^c\Phi^{iT}]^T \in \Re^l$ (i=1,..,n) (see Fig.1).

In the case of a task in the three-dimensional space (l=6), for example, the following results can be obtained. If the rotational matrix from Σ_i to Σ_o is denoted as ${}^oR_i({}^o\Phi^i)$ and the rotational matrix from Σ_i to Σ_c as ${}^cR_i({}^c\Phi^i)$, then the relationship among the position vectors ${}^op^c$, ${}^op^i$ and ${}^cp^i$ is given as follows:

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

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

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

Then, based on the nature of the Euler angle, the orientation vector ${}^{o}\Phi^c = [{}^{o}\phi^c, {}^{o}\theta^c, {}^{o}\psi^c]^{\mathrm{T}}$ can be obtained [4].

2.2 Kinematics of Multi-Arm Robots

The kinematic relationships of a multi-arm robot are summarized in **Fig.2.** $J^i \in \Re^{l \times m^i}$ is the Jacobian matrix of the arm i.; ${}^{o}\dot{X}^i$ and $\dot{q}^i \in \Re^{m^i}$ are the end-point velocity vector and the joint angular velocity vector, respectively; ${}^{o}F^i \in \Re^l$ expresses the force/moment vector of the end-point of the arm i represented in Σ_o; ${}^{o}\dot{X}^c$ and ${}^{o}F^{ci} \in \Re^l$ represent the velocity vector of the task point and the force/moment vector transmitted to the task point by the arm i, represented in Σ_o, respectively. It is assumed that the virtual rigid link is connected between the task point and the end-point of the arm i.

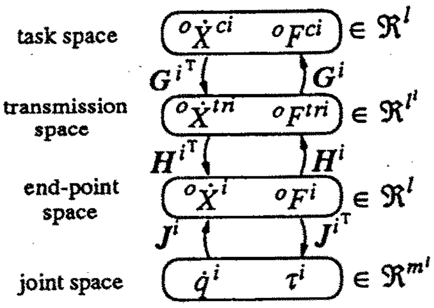

Fig.2 Kinematic relationship of motion and force variables

Using the contact type matrix $H^i \in \Re^{l_i \times l}$ expressing the filter characteristics which filter out some forces/moments of the end-point of the arm i and transmit other forces/moments to the object depending on the contact mechanism and the matrix $G^i = S^i H^{i\mathrm{T}} \in \Re^{l \times l_i}$, we denote the end-point velocity vector of the arm i transmitted from the task point as ${}^{o}\dot{X}^{tri} \in \Re^{l_i}$, and the vector of the forces/moments transmitting from the end-point of the arm i to the object as ${}^{o}F^{tri} \in \Re^{l_i}$ [5], [6]. Here, l_i is the degree of freedom of the forces/moment 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.

On the basis of the above formulations, a method of distributed planning of end-effector trajectories for multiple arms is explained in the next section.

3 Distributed Trajectory Generation for Multi-Arm Robots

Firstly, the dynamics of each arm and the task point are introduced. Then the interaction force/moment vector generated from these dynamics and the position constraints resulting from kinematic relationships between each arm and the task point are used to express interactions among the subsystems.

3.1 Composition of Subsystems

Let's now consider that the first to (n')-th arms control the motion of the task point and the $(n'+1)$st to n-th arms execute the relative motions between the task point and the end-point of each arm. Firstly, the motion equation of the arm i is given by

$$W^i \ddot{q}^i = \tau^i + Y^i - (V^i)^{\mathrm{T}} + (H^i J^i)^{\mathrm{T}} \lambda^i , \quad (3)$$

where $W^i \in \Re^{m^i \times m^i}$ is the non-singular inertia matrix of the arm i; $(V^i)^{\mathrm{T}} \in \Re^{m^i}$ is the nonlinear term representing the joint torque vector due to the centrifugal, Coriolis and friction forces of the arm i; $Y^i \in \Re^{m^i}$ is the gravitational torque vector of the arm i; $\tau^i \in \Re^{m^i}$ is the 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 Σ_o.

Next, let's consider the motion of the task point. Since a force/moment vector λ^i is exerted to the end-point of the arm i from the object, the reacting force/moment vector $-\lambda^i$ is exerted conversely to the object from each end-point. For this reason, the net force/moment vector acting on the object at the task point is given by

$${}^{o}F^c = \sum_{i=1}^{n} {}^{o}F^{ci} = \sum_{i=1}^{n} (-G^i \lambda^i) . \quad (4)$$

On the other hand, the motion equation of the task point, i.e. the object, is given by

$$M_c {}^{o}\ddot{X}^c = \begin{bmatrix} m_c g \\ -{}^{o}\omega_c \times [{}^{o}I_c {}^{o}\omega_c] \end{bmatrix} + {}^{o}F^c , \quad (5)$$

$$M_c = \begin{bmatrix} m_c I & 0 \\ 0 & {}^{o}I_c \end{bmatrix} , \quad (6)$$

where m_c is the mass of the object; ${}^{o}\ddot{X}^c \in \Re^l$ is the acceleration vector of the task point; $g \in \Re^3$ is the gravitational vector; ${}^{o}I_c \in \Re^{3\times3}$ and ${}^{o}\omega_c \in \Re^{3\times3}$ are the moment inertia tensor and the angular velocity vector of the object, respectively. "×" denotes the vector cross product.

Let's now consider the position constraints imposed on the end-point of each arm. First, the arms $i = 1,....,n'$ must be constrained by the motion of the task point determined by the dynamics of Eq. (5). From the kinematic relationship, we have

$$ {}^{o}\ddot{X}^{tri} = \dot{G}^{i\mathrm{T}}\,{}^{o}\dot{X}^{c} + G^{i\mathrm{T}}\,{}^{o}\ddot{X}^{c} \ . \tag{7} $$

On the other hand, in order to derive the position constraints for the arms i ($i = n'+1,\ldots,n$), we have to consider not only the motion of the task point but also the relative motions between the end-point of each arm and the task point, ${}^{c}X^{i} \in \Re^{l}$, which are given as the desired motions. The position constraint of these arms can be written as

$$ {}^{o}\ddot{X}^{tri} = {}^{o}\ddot{X}^{c} + \begin{bmatrix} {}^{o}R_c & 0 \\ 0 & {}^{o}R_c \end{bmatrix} {}^{c}\ddot{X}^{i} + \begin{bmatrix} 2\,{}^{o}\dot{R}_c & 0 \\ 0 & {}^{o}\dot{R}_c \end{bmatrix} {}^{c}\dot{X}^{i} + \begin{bmatrix} {}^{o}\ddot{R}_c & 0 \\ 0 & 0 \end{bmatrix} {}^{c}X^{i} . \tag{8} $$

Equations (7) and (8) are the end-point motion constraints of each arm.

Then, the joint trajectory of the arm i and the interaction force/moment vector λ^{i} can be obtained by using Eq. (3) and kinematic relationship of each arm as follows:

$$ \begin{bmatrix} \ddot{q}^{i} \\ \lambda^{i} \end{bmatrix} = \begin{bmatrix} W^{i} & -(H^{i}J^{i})^{\mathrm{T}} \\ -H^{i}J^{i} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \tau^{i} + Y^{i} - (V^{i})^{\mathrm{T}} \\ H^{i}\dot{J}^{i}\dot{q}^{i} - {}^{o}\ddot{X}^{tri} \end{bmatrix} . \tag{9} $$

Now, we will explain the computation method of the joint torque to generate the trajectory in parallel and distributed way. For the arm i ($i = 1, \ldots, n'$), the joint control torque τ^{i} is computed by using the target position of the task point, ${}^{o}X^{c^*}$, as follows:

$$ \tau^{i} = (H^{i}J^{i})^{\mathrm{T}} H^{i} (S^{i})^{-1} \{ K^{i}({}^{o}X^{c^*} - {}^{o}X^{c}) - B^{i}_{obj}\,{}^{o}\dot{X}^{c} - g^{i}_{obj}({}^{o}\omega_c) \} - Y^{i} - B^{i}_{j}\dot{q}^{i}, \tag{10} $$

where $K^{i}, B^{i}_{obj} \in \Re^{l\times l}$ are the positive definite matrices for the position and velocity feedback gain of the arm i, respectively; $B^{i}_{j} \in \Re^{m^i \times m^i}$ is the positive definite viscous friction matrix of the arm i; $g^{i}_{obj}({}^{o}\omega_c) \in \Re^{l}$ is the force vector contributed by the arm i to compensate the centrifugal and gravitational forces of the object.

On the other hand, for the other arms which are executing the relative motion, the joint torque is given by

$$ \tau^{i} = -Y^{i} - B^{i}_{j}\dot{q}^{i} \quad (i = n'+1,\ldots,n) \ . \tag{11} $$

The trajectory generation method proposed here is now illustrated in Fig.3. Each subsystem generates a trajectory cooperatively each other using the end-point force/moment vector λ^{i} as an information via the dynamics of the task point. In the following section, the stability of this system will be analyzed.

3.2 Stability of the System

Let's now consider an energy function H composed by two types of energy function H_1 and H_2 as given below:

$$ H = H_1 + H_2 \ , \tag{12} $$

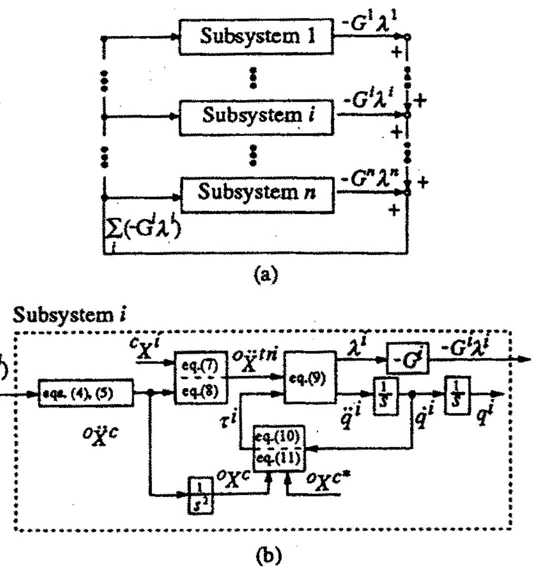

Fig.3 Composition of a subsystem

$$ H_1 = \sum_{i=1}^{n'} E^{i} + Q_c + \frac{1}{2}\sum_{i=1}^{n'} \dot{q}^{i\mathrm{T}} W^{i} \dot{q}^{i} \ , \tag{13} $$

$$ H_2 = \frac{1}{2}\sum_{i=n'+1}^{n} \dot{q}^{i\mathrm{T}} W^{i} \dot{q}^{i} \ , \tag{14} $$

$$ E^{i} = \frac{1}{2}({}^{o}X^{c^*} - {}^{o}X^{c})^{\mathrm{T}} K^{i} ({}^{o}X^{c^*} - {}^{o}X^{c}) \ , \tag{15} $$

$$ Q_c = \frac{1}{2}\,{}^{o}\dot{X}^{c\mathrm{T}} M_c\,{}^{o}\dot{X}^{c} \ . \tag{16} $$

H_1 and H_2 are energy functions for the motion of the task point and the relative motion between each arm i ($i = n'+1,\ldots,n$) 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 ($i = 1,\ldots,n'$), and Q_c expresses the kinetic energy of the task point.

It can be shown that the $\dot{H}$ is given by

$$ \dot{H} = -\sum_{i=1}^{n} (\dot{q}^{i})^{\mathrm{T}} B^{i}_{j} \dot{q}^{i} - \sum_{i=1}^{n'} ({}^{o}\dot{X}^{c})^{\mathrm{T}} B^{i}_{obj} ({}^{o}\dot{X}^{c}) \ , \tag{17} $$

using the property: $(\dot{q}^{i})^{\mathrm{T}}\dot{W}^{i}\dot{q}^{i} = 2(\dot{q}^{i})^{\mathrm{T}}(V^{i})^{\mathrm{T}}$. Since B^{i}_{j} and B^{i}_{obj} are positive definite matrices, we have $\dot{H} \le 0$ and that energy function H decreases monotonically until $\dot{H} = 0$, i.e. $\dot{q}^{i} = 0$ ($i = 1,\ldots,n$) and ${}^{o}\dot{X}^{c} = 0$. This means that the whole system is asymptotically stable.

The distributed trajectory generation method for multi-arms robots has been explained in this chapter and the stability of the whole system has also been analyzed. In the next chapter, the effectiveness of the method will be verified by simulation experiments.

4 Simulation Experiments

By applying the proposed method to a cooperative task by three planar arms, computer simulations were carried out (see **Fig.4**). The link parameters of each arm and the object parameters are shown in **Table 1**. In this case, the position of the task point is set at the center of gravity of the object (the origin of the task coordinate system), and the motion of the task point starts from the initial position ${}^{o}X^{c}(0) = [0.2(\mathrm{m}), 1.1(\mathrm{m}), 0(\mathrm{rad})]^{T}$ to the target position ${}^{o}X^{c*} = [0.3(\mathrm{m}), 0.5(\mathrm{m}), 0(\mathrm{rad})]^{T}$ (see Fig.4). In addition, the dynamic parameters of each arm used in the simulations are set as $K^{i} = \mathrm{diag}.[100(\mathrm{N/m}), 100(\mathrm{N/m}), 100(\mathrm{Nm/rad})]$; $B^{i}_{obj} = \mathrm{diag}.[11.6(\mathrm{N/(m/s)}), 11.6(\mathrm{N/(m/s)}), 5.9(\mathrm{Nm/(rad/s)})]$; and $B^{i}_{j} = \mathrm{diag}\,[1, 1, 1, 1]\,(\mathrm{Nm/(rad/s)})$ where "diag .[·]" denotes a diagonal matrix.

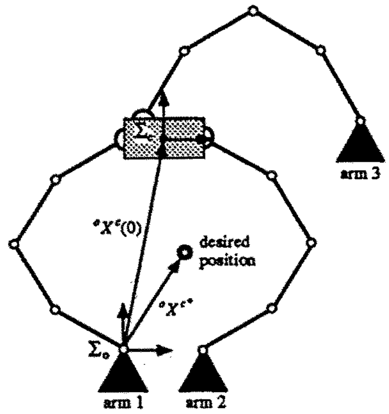

Fig.4 Initial posture and the desired position of the task point

Table 1 Link parameters of each arm and the object parameters

	link i ($i = 1, 2, 3, 4$) and the object
mass (kg)	1.0
length (m)	0.4
center of mass (m)	0.2
moment of inertia (kgm^2)	0.267

Figure 5 shows the simulation result for the case in which the end-point of the arm 3 undergoes a relative motion respect to the task point along the surface while the arm 1 and the arm 2 control the motion of the task point. In this case, arm 1 and arm 2 grasp the object rigidly. The relative motion of the end-point of the arm 3 is given as a function of the time as follows:

$$
{}^{c}X^{3}(t) = \begin{cases} [2.5t^{2} - 0.1(\mathrm{m}), 0.1(\mathrm{m}), \frac{4}{3}\pi(\mathrm{rad})]^{T} & \text{if } 0 \le t \le 0.2 \\ [-2.5t^{2} + 2.0t - 0.3(\mathrm{m}), 0.1(\mathrm{m}), \frac{4}{3}\pi(\mathrm{rad})]^{T} & \text{if } 0.2 \le t \le 0.4 \\ [0.1(\mathrm{m}), 0.1(\mathrm{m}), \frac{4}{3}\pi(\mathrm{rad})]^{T} & \text{if } t \ge 0.4 \end{cases} \qquad (18)
$$

It is seen from the Fig.5 that even in the case of the end-points carrying out relative motions, a cooperative task can be realized maintaining the closed link structure.

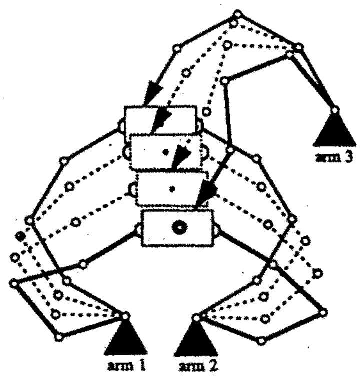

Fig.5 Result of the trajectory generation

5 Conclusions

A method of generating the end-points' trajectories of multi-arm robots in a parallel and distributed way has been proposed. In the method, each arm was expressed as a subsystem containing its dynamics and the interactions among the arms were expressed by using the interaction forces/moments between each arm and the task point. On the other hand, the position constraints were derived from the task point in order to maintain the closed chains formed by the arms and the object.

Since the joint torques for the controlling the motion of the task point are calculated using only the position error of the task point, it may be possible for the system become deadlock because of constraints imposed on the motions of the each arm such as restriction of the joint motions. Therefore, future work will be directed to avoid this problem, for example, by making positive use of the redundant joint degrees of freedom to regulate joint motion as the sub task.

References

[1] T. Fukuda, S. Kawauchi and H. Asama: Analysis and evaluation of cellular robotics (CEBOT) as a distributed intelligent system by communication information amount, Proc. of the 1990 IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS '90), pp. 827-834 (1990).

[2] H. Asama, M. K. Habib and I. Endo: Functional distribution among multiple mobile robots in an autonomous and decentralized robot systems, Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation, pp. 1921-1926 (1991).

[3] T. Tsuji, S. Nakayama and K. Ito: Distributed feedback control for redundant manipulators based on virtual arms, Proc. of the Int. Symp. on Autonomous Decentralized Systems, pp. 143-149 (1993).

[4] K. S. Fu, R. C. Gonzales and S. G. Lee: Robotics: Control, Sensing, Vision, and Intelligence. McGraw-Hill Book Company (1987)

[5] M. R. Cutkosky and I. Kao: Computing and controlling the compliance of a robotic hand. IEEE Trans. on Robotics and Automation, 5(2): 151-165 (1989)