Journal of Intelligent and Robotic Systems **22:** 99–116, 1998.

Path Planning and Control of a Cooperative Three-Robot System Manipulating Large Objects

COSTAS S. TZAFESTAS
Laboratoire de Robotique de Paris, CNRS-UPMC-UVSQ, 10-12 Avenue de l'Europe, 78140 Vélizy, France; e-mail: tzaf@robot.uvsq.fr

PLATON A. PROKOPIOU and SPYROS G. TZAFESTAS
Intelligent Robotics and Automation Laboratory, Department of Electrical and Computer Engineering, National Technical University of Athens, Greece; e-mail: tzafesta@softlab.ece.ntua.gr

(Received: 5 February 1997; accepted in final form: 10 November 1997)

Abstract. After a brief review of the current research on multi-robot systems, the paper presents a path planning and control scheme for a cooperative three-robot system transferring/manipulating a large object from an initial to a desired final position/orientation. The robots are assumed to be capable of holding the object at three points that define an isosceles triangle. The mode of operation adopted is that of a "master-and-two-slave robots". The control scheme employs the differential displacement of the object which is transformed into that of the end-effector of each robotic arm, and then used to compute the differential displacements of the joints of the robots. The scheme was applied to several 3-robot systems by simulation and proved to be adequately effective, subject to certain conditions regarding the magnitude of the differential displacements. Here, an example is included which concerns the case of three Stäubli RX-90L robots.

Key words: multi-robot systems, cooperative three-robot system, path planning-control, master-and-two-slaves mode, large object manipulation.

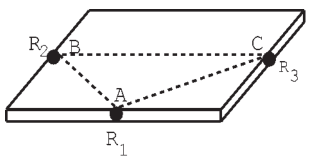
1. Introduction

The study of multiple cooperating robots was initiated about two decades ago with many important results already available [1–54]. The field of multiple cooperating robots is very important since, although many industrial operations and tasks can be performed efficiently by a single robot, there are other tasks that need the coordinated and cooperative work of more than one robots for satisfactory and economic performance. The majority of results so far derived concern the case of two cooperating robots, although in some cases the theory developed is applicable to more than two cooperating robots. Actually, most of the publications in the open literature present theoretical investigations, and only a few of them provide practical experimental results. Tasks that need the cooperation of more than one robots include the manipulation and transportation of large objects or long and heavy bars or large flexible objects or objects without special features (e.g., handles).

Although the capabilities of 2-robot systems are substantially increased over single-robot systems, they are still unable to handle (grasp, hold, manipulate, transport, etc.) very large, very heavy or peculiar objects (such as large cardboard boxes). Therefore, attention must be turned to the case of using three (or more) cooperating robots.

The purpose of this paper is to present a path planning/control procedure for a 3-robot arm system that moves large objects from an initial to a desired position/orientation, by holding it at three different points defining a triangle. For completeness, a short review of the state-of-art of the multiple cooperating robots field is first provided, which shows the problems so far investigated and the techniques used.

Single robot tasks can be performed by controlling the robot's hand such as to follow a desired path, without controlling the exact time at which the hand passes through the particular points of the trajectory. The orientation of the robot's hand during the motion may be irrelevant. This is not true in multi-robot systems where, once the two or more hands grasp the object, their relative positions and orientations with respect to each other must remain invariant during the entire operation. Actually, in cooperating multi-robot systems each hand must pass through a particular point on its trajectory at exactly the right time, and the orientations of the hands must also be the proper ones.
In the 3-robot case (R_1, R_2, R_3):



two possible strategies can be followed:

(I) *Master-Slave-Driver:* Here the robot R_2 is the master, R_3 the slave, and R_1 the driver (that orients the plane ABC in space).

(II) *Master-and-two-Slaves* (Leader-and-two-Follower Robots): The motion of R_1 (master) follows directly from the motion planning, whereas the motion of R_2 and R_3 (the slaves) must also satisfy the constraints posed by the rigid object. Actually there are two ways of specifying the initial and final positions/orientations of the object; one using the grasping points A, B, C of the master and the two slaves, and the other using the center of gravity of the object (see Figure 2).

The control of the robots is performed incrementally with the aid of the differential relations between the object and the three robot arms. The differential change of the object is transformed into that of each robot arm, and then the differential change of each joint of the three robots is derived. Numerical simulation results are provided which demonstrate the applicability and effectiveness of the proposed planning-control algorithm.

2. Multi-Robot Systems: A Review

In general, multi-robot systems are distinguished in two broad classes:

Class-1: Each robot performs its own task independently in a shared common workspace (e.g., [1, 7–9, 48]).

Class-2: All robots cooperate to perform a given task (e.g., [2, 4, 6, 13, 20, 21, 22, 25, 48]).

The main problem that has to be faced in *class-1* multi-robot systems is that of path planning so as to avoid collisions. Here the robots have no physical interactions during the execution of the desired task. The complexity of the problem, especially when there are objects moving in the common workspace, has constrained the studies to point-to-point tasks which are performed in a pseudo-concurrent manner. Most of the algorithms developed are suitable for off-line application. Frequently, the path planning needs time delay actions and/or changes of the robot paths in space.

Class-2 multirobots have been designed by several control schemes, in which the robots are assumed to have direct interaction while performing the desired tasks. The robots in class-2 systems are strongly coupled, and the desired path of the object fully determines the task space of each robotic arm. Here, three principal kinds of tasks have been considered throughout the years. The first category of tasks, where there is no relative motion between the object and the end-effectors, is that of grasping a unique rigid object and transferring it from an initial position/orientation to a new one.

Problems that have been studied for this category of tasks are:

- Decoupling control [39, 40],
- Force control [10, 11, 17, 24, 43],
- Hybrid position/force control [12, 15, 26, 27, 42, 44],
- Force/load distribution [23, 28–36, 38, 49].

In the second category of tasks, the robot arms are called to manipulate objects having movable parts like a pair of pliers [20, 43]. Finally, the third category of tasks concerns the treatment of large objects which cannot be grasped by end-effectors but only pushed by them (e.g., large cardboard boxes) [18, 46, 47, 51]. In this case, the object can be pushed by *enveloping grasp* using the robot links [18–51] or by *open-palm end-effectors* [19, 46, 47]. The system constraints are unilateral, and the object may slide or roll along the contact surfaces (i.e., the constraints are nonholonomic).

Most of the systems studied in the literature involve only two robots. The case of more than two robots has been considered principally in connection with the force and load distribution problem. The class of multi-robot systems also involves the so-called multifingered robots [41–50], which can be treated by similar techniques. The majority of publications in the open literature present theoretical investigations with simulation results. Examples of real experimental works are provided in [16–19, 37]. The case of dual-arm robotic system with joint flexibility was studied in [45].

We now proceed to a little more detailed description of a few representative works.

In [2] a hierarchical control structure is proposed for a 2-robot system operating in the master-slave mode. Each robot arm is controlled such that the error between the actual and desired paths is minimized. The desired path for the master robot is specified, while the desired path of the slave manipulator is expressed with respect to the path of the master robot. This relative position error between the two end-effectors is controlled by the two-robot coordination control computer. The controller involves a joint position predictor for the master robot, a coordinate transformation unit, and a slave command modifier.

In [3], feedback linearization is introduced, and the pole placement technique is used to the resulting state-space model which describes the dynamics of the multi-robot system. The controller needs the exact knowledge of the model parameters, and the transients are controlled by the suitable choice of the system eigenvalues. For a general 6-joint robot (like the PUMA 600 considered in the simulations) the technique needs a large number of computations and no attention was paid to control the interacting forces.

In [4], each joint is controlled by a proportional type controller with the error being expressed in Cartesian space. When two or more robots grasp the same object, the interacting forces and positions are constrained. Therefore, for achieving coordinated operation of the multi-robot system, these force constraints must be determined including the natural and the artificial ones. Such formulations are presented from several points of view in [10–12, 15, 17, 26, 27, 42–44].

In [10], the master-slave mode is considered, where the master arm is controlled by a position PID controller with a feedforward term, and the slave moves in cooperation with the master while its force is controlled so as to balance the interactive force exerted by the master via the object.

In [12], the controllers of the two robot arms are designed using the MIMO discrete-time autoregressive stochastic (ARX) model with external inputs, where the parameters are estimated on-line recursively. The controllers are adaptive with time-varying gains, they depend on predicted errors of the two robots, and they need complete state information (position, velocity, force) of the two cooperating robots, as well as the current parameter estimates of the ARX models. The design specifications (position/orientation paths of the master and slave robots) are expressed in terms of a performance index that has to be minimized subject to the ARX robot models. The discrete-time controllers $u_1(k)$ and $u_2(k)$, so obtained, are of the self-tuning type [52, 53].

The optimal force/load distribution problem, which is a kind of inverse dynamics problem, has been solved using various optimization techniques based on different types of objective functions and optimization constraints [28–36, 38, 49]. The simplest methods are based on the *linear programming* (LP) technique. A noticeable contribution is presented in [29], where the so-called *compact dual LP* technique was developed, which uses inequality constraints and can be implemented in real-time. The duality principle of LP was used in order to reduce the size of the problem by exchanging the variables with constraints (the actual number of which is much smaller than the number of variables). The compact dual LP technique is very efficient from a computational point of view, but possesses the limitations of LP techniques, i.e., any quadratic quantities (e.g., force norm, effort of torque, power, energy) cannot be dealt with, and also undesirable discontinuities may be introduced into the solution. These disadvantages are not possessed by nonlinear programming (NLP)-based techniques such as those presented in [25, 30, 32–38].

In [25], the objective function represents a minimum-norm force, and the quadratic inequality constraints represent the friction cone. The problem size is not considerably reduced, and the computational requirements of the solution remain high. In [30] the problem is formulated so as to minimize the energy consumption of the two cooperating robots subject to joint torque constraints, and the internal force without any joint torque constraint.

In [32], the problem is to minimize the joint torque effort with quadratic inequality constraints on the maximum joint torque of dual robots. The method adopted is based on Lagrange multipliers combined with the force ellipsoid.

In [36], a quadratic criterion is minimized subject to linear inequality constraints for the multi-finger force-distribution problem. The resulting quadratic programming (QP) algorithm is very fast, but it is expected to be slower when applied to a cooperating multi-robot system, since here the number of constraints is much larger than that of a multi-finger robotic system.

This drawback was eliminated in [38] by combining the duality principle with the QP technique, and introducing quadratic constraints to deal with force-norm constraints without approximating them by linear ones. Therefore, the algorithm of [38] reduces the size of the problem via the duality principle (exchange of variables with constraints) and does not possess the drawbacks of LP-based techniques, such as that proposed in [29]. Note that the number of variables in a conventional algorithm is $6n$, and the number of constraints is $2mn + 6$ ($m =$ common number of degrees of freedom of robots, $n =$ number of robots). In the dual QP algorithm they decrease to $2n$ and $2n$, respectively. The application of the dual-QP algorithm to the case of a 2-PUMA-robot system showed that indeed it can be implemented in real time (the computation time was less than (1/8)-times as that of a standard QP algorithm) [38].

In [19], a 2-robot control system is presented which uses the full dynamics of the robots and manipulates large objects through the approach of enveloping grasp [18, 51]. The proposed architecture, called TRACS (two robotic arm coordination system) involves two PUMA 250 robots, an AT PC-based coordination controller (80286 compatible), and a number of sensors. The PC-AT communicates with the Unimation controllers of the two robots via a parallel interface (16 bits to each controller), where the parallel lines are directly connected to the joint microprocessors through the arm interface board (i.e., bypassing the LS-11 processor). The two processors are scheduled concurrently (so as to realize a 400 Hz sampling rate). This ensures real-time control under the MS-DOS operating system. The coordination controller of the 2-robot system includes a dynamic force controller, a rolling control algorithm, and a planning algorithm all of which are fully described in [19]. This controller which was extensively experimentally tested is capable of handling large industrial objects (e.g., cardboard boxes), waste disposal objects (e.g., barrels), military objects (e.g., crates), and space objects (e.g., satellites).

In [54], a simple robust scheme for the on-line concurrent motion planning of multi-robot systems is provided. This scheme uses a linear system of equations for each robot taking into consideration a vector for motion planning, and an original procedure for the proper perturbation of the pseudoinverse matrix. The proposed scheme can deal simultaneously with both real-time motion coordination and singularities prevention in a sensor-based environment. The approach adopted is based on, conceptually considering redundant robotic manipulators, formulating for each one of them an inverse kinematics problem under an inexact context. Although the "manipulability index" is usually employed for avoiding singularities, in [54] it is used for motion planning. The algorithm was tested by simulation on a dual robot system involving two planar redundant robots. The results showed the efficiency of the proposed scheme which, due to its properties, is suitable for autonomous or telerobotic systems operations.

3. Cooperative 3-Robot Arm Kinematics

In the following, the kinematics equations of the 3-robot-arms system will be derived for the master-and-two-slaves configuration (using homogeneous transformations).

3.1. THE 3-ROBOT SYSTEM: MASTER-AND-TWO-SLAVES CONFIGURATION

We consider the symmetric configuration shown in Figure 1.
The world coordinate (w-c) system is defined to be the coordinate system $x_0 y_0 z_0$ of the master's base. Therefore the position and orientation of an object with respect to w-c is described by an homogeneous matrix A^m:

$$A^m = \left[\begin{array}{ccc|c} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ \hline & \mathbf{0} & & 1 \end{array}\right], \qquad \mathbf{p} = \left[p_x, p_z, p_z\right]^{\mathrm{T}}, \tag{1}$$

where the vectors $\mathbf{n}$, $\mathbf{o}$ and $\mathbf{a}$ define the orientation of the object, and $\mathbf{p}$ its position (the position of the origin of the coordinate system $[\mathbf{n}, \mathbf{o}, \mathbf{a}]$). The position and

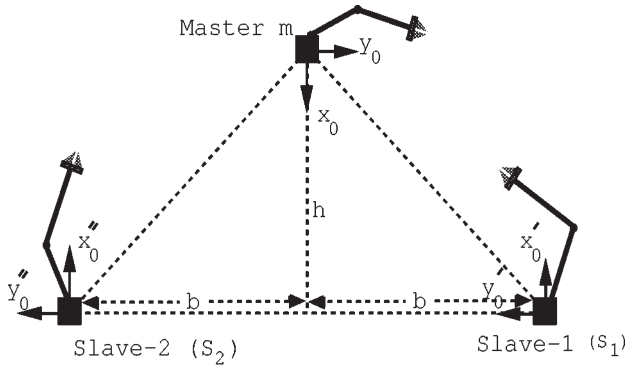


Figure 1. Symmetric master-and-two-slaves configuration (all axes z_0, z_0', z_0'' are normal to the plane $m - s_1 - s_2$).

orientation of the same object with respect to the coordinate systems of the bases of S_1 and S_2 is given by

$$A^{S_1} = S_1^{-1} A^m \quad \text{and} \quad A^{S_2} = S_2^{-1} A^m, \tag{2}$$

where the matrices S_1 and S_2 define the coordinate systems of the slaves S_1 and S_2, respectively, and are given by (see Figure 1):

$$S_1 = \begin{bmatrix} -1 & 0 & 0 & h \\ 0 & -1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad S_2 = \begin{bmatrix} -1 & 0 & 0 & h \\ 0 & -1 & 0 & -b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

One can see that in this symmetric configuration: $S_i^{-1} = S_i$ $(i = 1, 2)$, whereas the transformation from $x_0' y_0' z_0'$ to $x_0'' y_0'' z_0''$ is equal to:

$$S_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad S_{21} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = S_{12}^{-1}.$$

It must be remarked that in practice the distances b and h must be carefully selected and depend on the shape of the workspaces of the three robots as well as on the overall motion of the three-robot system. Usually, one can find suitable values of b and h that depend on the application at hand.

3.2. PLANAR OBJECT MANIPULATION

Here we consider a particular application, where a "plane" (planar object) has to be transferred from an initial to a final position. The three robots grasp the object

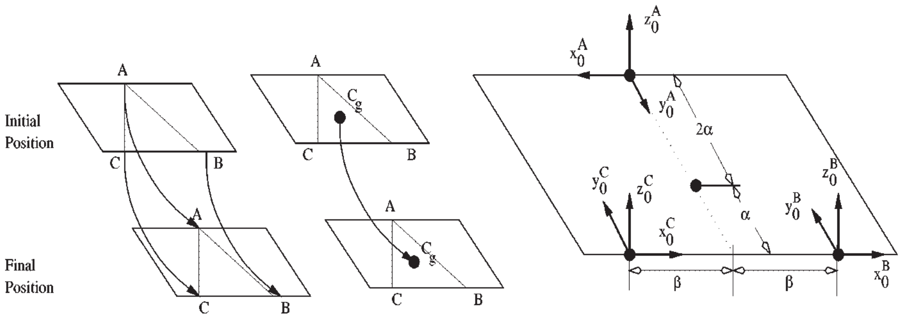


Figure 2. Two ways of specifying the initial and final positions (C_g: center of gravity).

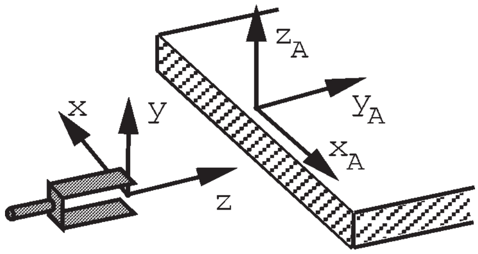


Figure 3. Coordinate systems attached to the end-effector and grasping point.

at three points A, B and C that define a triangle. Therefore one can either define the initial and final positions of the vertices of this triangle, or the initial and final position of the center of gravity of the triangle and the initial and final orientation of the plane (Figure 2).

The initial and final positions or the path of the object (defined in one of the two ways shown in Figure 2) are used to determine the path (position and orientation) that must be followed by the end-effector of each arm. The position and orientation of an end-effector, with respect to the robot-base reference frame, is described by an homogeneous transformation (4×4 matrix) H of the type described by Equation (1).

The coordinate systems of the end-effectors and the grasping points are assumed as shown in Figure 3, and therefore:

$$G^{\mathrm{A}} = H^{m}\Theta, \qquad G^{\mathrm{B}} = H^{S_1}\Theta, \qquad G^{\mathrm{C}} = H^{S_2}\Theta, \tag{4}$$

where G^{A}, G^{B}, G^{C} are the coordinate systems attached to the grasping points A, B and C of the master, slave-1 and slave-2, respectively, expressed in the corresponding robot reference frame, and:

$$\Theta = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

The matrices that define the coordinate systems attached to B and C with respect to the coordinate system attached to A are:

$$K_{\mathrm{B}}^{*}=\begin{bmatrix}-1&0&0&-\beta\\0&-1&0&3\alpha\\0&0&1&0\\0&0&0&1\end{bmatrix},\qquad K_{\mathrm{C}}^{*}=\begin{bmatrix}-1&0&0&\beta\\0&-1&0&3\alpha\\0&0&1&0\\0&0&0&1\end{bmatrix}. \tag{6}$$

4. The Path Planning and Control Algorithm

4.1. ABSOLUTE MOTION EQUATIONS

Our purpose here is to define the motion of the planar object in space. Consider first the motion of the point A (grasped by the master arm). This motion is defined by a time-varying homogeneous transformation matrix $M(t)$ which determines the linear and angular displacements needed for the point A to go from the initial to the desired final position and orientation. The matrix $M(t)$ is given by

$$M(t)=\begin{bmatrix}r_xr_xv(\tau\phi)+c(\tau\phi) & r_yr_xv(\tau\phi)-r_zs(\tau\phi) & r_zr_xv(\tau\phi)+r_ys(\tau\phi) & \tau x\\ r_xr_yv(\tau\phi)+r_zs(\tau\phi) & r_yr_yv(\tau\phi)+c(\tau\phi) & r_zr_yv(\tau\phi)-r_xs(\tau\phi) & \tau y\\ r_xr_zv(\tau\phi)-r_ys(\tau\phi) & r_yr_zv(\tau\phi)+r_xs(\tau\phi) & r_zr_zv(\tau\phi)+c(\tau\phi) & \tau z\\ 0&0&0&1\end{bmatrix}, \tag{7}$$

where $\tau = t/t_f$ is normalized time (t_f is the time in which the motion has to be completed), $s(\cdot) = \sin(\cdot)$, $c(\cdot) = \cos(\cdot)$, $v(\cdot) = 1 - \cos(\cdot)$, $\mathbf{p} = [x, y, z]^{\mathrm{T}}$ is the position displacement vector from the initial to the final position, and the vector $\mathbf{r} = [r_x, r_y, r_z]^{\mathrm{T}}$ defines the axis about which the initial coordinate system must rotate by an angle ϕ_* to obtain the final orientation.

Now, if $G^{\mathrm{A}}(0)$ is the matrix defining the initial position/orientation of the point A, then the time-varying position/orientation of A with respect to the w-c system is given by

$$G^{\mathrm{A}}(t) = G^{\mathrm{A}}(0)M(t) \tag{8}$$

and the final one is given by

$$G_f^{\mathrm{A}} = G^{\mathrm{A}}(t_f) = G^{\mathrm{A}}(0)M(t_f), \tag{9}$$

where

$$x = \mathbf{n}^{\mathrm{T}}(0)\left[\mathbf{p}(t_f) - \mathbf{p}(0)\right], \qquad y = \mathbf{o}^{\mathrm{T}}(0)\left[\mathbf{p}(t_f) - \mathbf{p}(0)\right],$$
$$z = \mathbf{a}^{\mathrm{T}}(0)\left[\mathbf{p}(t_f) - \mathbf{p}(0)\right], \tag{10a}$$

$$\phi_* = \cos^{-1}\left[\frac{1}{2}(\mathbf{n}^{\mathrm{T}}(0)\mathbf{n}(t_f) + \mathbf{o}^{\mathrm{T}}(0)\mathbf{o}(t_f) + \mathbf{a}^{\mathrm{T}}(0)\mathbf{a}(t_f) - 1)\right], \tag{10b}$$

$$\mathbf{r} = \begin{bmatrix} \mathbf{a}^{\mathrm{T}}(0)\mathbf{o}(t_f) - \mathbf{o}^{\mathrm{T}}(0)\mathbf{a}(t_f) \\ \mathbf{n}^{\mathrm{T}}(0)\mathbf{a}(t_f) - \mathbf{a}^{\mathrm{T}}(0)\mathbf{n}(t_f) \\ \mathbf{o}^{\mathrm{T}}(0)\mathbf{n}(t_f) - \mathbf{n}^{\mathrm{T}}(0)\mathbf{o}(t_f) \end{bmatrix}. \tag{10c}$$

The motion of the points B and C of the object is defined by

$$G^{\mathrm{B}}(t) = S_1 G^{\mathrm{A}}(0)M(t)K_{\mathrm{B}}^* \quad \text{and} \quad G^{\mathrm{C}}(t) = S_2 G^{\mathrm{A}}(0)M(t)K_{\mathrm{C}}^*. \tag{11}$$

The motion of the end-effectors of the three arms grasping the points A, B and C is defined by the transformations $H^m(t)$, $H^{S_1}(t)$ and $H^{S_2}(t)$, which can be determined by equating the right-hand sides of (4) and (8), (11), respectively, and solving the resulting equations:

$$\begin{aligned} &H^m(t) = G^{\mathrm{A}}(0)M(t)\Theta, \qquad H^{S_1}(t) = S_1 G^{\mathrm{A}}(0)M(t)K_{\mathrm{B}}^*\Theta, \\ &H^{S_2}(t) = S_2 G^{\mathrm{A}}(0)M(t)K_{\mathrm{C}}^*\Theta, \end{aligned} \tag{12}$$

where the relations $S_i^{-1} = S_i$ $(i = 1, 2)$ and $\Theta^{-1} = \Theta$ were used.

4.2. INCREMENTAL MOTION EQUATIONS

We now determine the incremental (differential) motion equations of the three-robot arm system. Let

$$\mathbf{D} = \left[\mathrm{d}_x, \mathrm{d}_y, \mathrm{d}_z; \mathrm{d}\phi_x, \mathrm{d}\phi_y, \mathrm{d}\phi_z\right]^{\mathrm{T}}$$

be the differential motion vector, where d_x, d_y, d_z are differential linear displacements and $\mathrm{d}\phi_x, \mathrm{d}\phi_y, \mathrm{d}\phi_z$ are differential angular displacements with respect to the axes x, y, z, respectively.

Consider the grasping point A. The coordinate system of A at time $(t + \mathrm{d}t)$ is given by

$$G^{\mathrm{A}}(t + \mathrm{d}t) = G^{\mathrm{A}}(t) + \mathrm{d}G^{\mathrm{A}}(t) = G^{\mathrm{A}}(t)[I + \Delta], \tag{13a}$$

where I is the 4×4 unit matrix, and

$$\Delta = \begin{bmatrix} 0 & -\mathrm{d}\phi_z & \mathrm{d}\phi_y & \mathrm{d}x \\ \mathrm{d}\phi_z & 0 & -\mathrm{d}\phi_x & \mathrm{d}y \\ -\mathrm{d}\phi_y & \mathrm{d}\phi_x & 0 & \mathrm{d}z \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{13b}$$

Similarly, the differential transformations for the three arms are defined by

$$H^k(t + \mathrm{d}t) = H^k(t)\left[I + \Delta^k\right], \tag{14a}$$

$$\Delta^k = \begin{bmatrix} 0 & -\mathrm{d}\phi_z^k & \mathrm{d}\phi_y^k & \mathrm{d}x \\ \mathrm{d}\phi_z^k & 0 & -\mathrm{d}\phi_x^k & \mathrm{d}y \\ -\mathrm{d}\phi_y^k & \mathrm{d}\phi_x^k & 0 & \mathrm{d}z \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{14b}$$

for $k = m$, S_1 and S_2, respectively.

From the analysis of Section 4.1, it follows that:

$$\begin{aligned} &H^m(t) = G^{\mathrm{A}}(t)\Theta, \qquad S_1 H^{S_1}(t) = G^{\mathrm{A}}(t)K_{\mathrm{B}}^*\Theta, \\ &S_2 H^{S_2}(t) = G^{\mathrm{A}}(t)K_{\mathrm{C}}^*\Theta. \end{aligned} \tag{15}$$

Similar equations hold for the time instant $(t + \mathrm{d}t)$.

Now using (13a), (13b), (14a), (14b) and (15), and solving for Δ^m, Δ^{S_1} and Δ^{S_2} we obtain:

$$\begin{aligned} \Delta^m &= \left[H^m(t)\right]^{-1} G^{\mathrm{A}}(t)\Delta\Theta, \\ \Delta^{S_1} &= \left[H^{S_1}(t)\right]^{-1} S_1 G^{\mathrm{A}}(t)\Delta K_{\mathrm{B}}^*\Theta, \\ \Delta^{S_2} &= \left[H^{S_2}(t)\right]^{-1} S_2 G^{\mathrm{A}}(t)\Delta K_{\mathrm{C}}^*\Theta, \end{aligned}$$

which by (15) reduce to:

$$\Delta^m = \Theta\Delta\Theta, \qquad \Delta^{S_1} = \Theta K_{\mathrm{B}}^*\Delta K_{\mathrm{B}}^*\Theta, \qquad \Delta^{S_2} = \Theta K_{\mathrm{C}}^*\Delta K_{\mathrm{C}}^*\Theta. \tag{16}$$

Equation (16) gives the differential displacements of the three robots end-effectors in terms of the differential displacement matrix Δ of the point A. Using (13b) and the definition of Θ in (5) we get the following:

Master Arm

$$\begin{aligned} &\mathrm{d}\phi_x^m = -\mathrm{d}\phi_x, \qquad \mathrm{d}\phi_y^m = \mathrm{d}\phi_z, \qquad \mathrm{d}\phi_z^m = \mathrm{d}\phi_y, \\ &\mathrm{d}x^m = -\mathrm{d}x, \qquad \mathrm{d}y^m = \mathrm{d}z, \qquad \mathrm{d}z^m = \mathrm{d}y. \end{aligned}$$

Slave-1

$$\begin{aligned} &\mathrm{d}\phi_x^{s_1} = \mathrm{d}\phi_x, \qquad \mathrm{d}\phi_y^{s_1} = \mathrm{d}\phi_z, \qquad \mathrm{d}\phi_z^{s_1} = -\mathrm{d}\phi_y, \\ &\mathrm{d}x^{s_1} = \mathrm{d}x - 3\alpha\mathrm{d}\phi_z, \quad \mathrm{d}y^{s_1} = \mathrm{d}z + \beta\mathrm{d}\phi_y + 3\alpha\mathrm{d}\phi_x, \quad \mathrm{d}z^{s_1} = -\mathrm{d}y + \beta\mathrm{d}\phi_z. \end{aligned}$$

Slave-2

$$\begin{aligned} &\mathrm{d}\phi_x^{s_2} = \mathrm{d}\phi_x, \qquad \mathrm{d}\phi_y^{s_2} = \mathrm{d}\phi_z, \qquad \mathrm{d}\phi_z^{s_2} = -\mathrm{d}\phi_y, \\ &\mathrm{d}x^{s_2} = \mathrm{d}x - 3\alpha\mathrm{d}\phi_z, \quad \mathrm{d}y^{s_2} = \mathrm{d}z - \beta\mathrm{d}\phi_y + 3\alpha\mathrm{d}\phi_x, \quad \mathrm{d}z^{s_2} = -\mathrm{d}y - \beta\mathrm{d}\phi_z. \end{aligned}$$

4.3. THE COOPERATIVE PATH PLANNING/CONTROL ALGORITHM

To develop the proposed incremental motion control algorithm (for each robotic arm) the total linear and angular displacement of the point A ($\mathbf{p} = [x, y, z]^{\mathrm{T}}$ and ϕ_*) given by (10a), (10b) is divided in a large number of small (nearly infinitesimal) displacements $\delta\mathbf{p}$ and $\delta\boldsymbol{\phi}$. From these displacements and the above relations one can compute the corresponding displacements $\delta\mathbf{p}^m$, $\delta\boldsymbol{\phi}^m$; $\delta\mathbf{p}^{s_1}$, $\delta\boldsymbol{\phi}^{s_1}$; $\delta\mathbf{p}^{s_2}$, $\delta\boldsymbol{\phi}^{s_2}$ of the three arms.

Let q_i $(i = 1, 2, \ldots, 6)$ be the displacement of each joint, and $\mathrm{d}q_i$ the corresponding differential displacement. Then we can write:

$$\begin{aligned} &H^m[\mathrm{d}x^m, \mathrm{d}y^m, \mathrm{d}z^m; \mathrm{d}\phi_x^m, \mathrm{d}\phi_y^m, \mathrm{d}\phi_z^m]^\mathrm{T} \\ &\quad = J^m(q_1, \ldots, q_6)[\mathrm{d}q_1^m, \mathrm{d}q_2^m, \ldots, \mathrm{d}q_6^m]^\mathrm{T}, \end{aligned} \tag{17}$$

where J^m is the Jacobian matrix of the master arm. Similar equations hold also for the slave arms.

Given the small displacement $\delta x^m, \ldots, \delta\phi_z^m$ (determined as discussed previously) can find the corresponding δq_i^m $(i = 1, \ldots, 6)$, by solving Equation (17), under the assumption that the robot does not pass very near to or via singular configurations. On the basis of the above analysis the incremental motion control algorithm is as follows.

- *Step 0* (initialization): Determine the initial position (the q_i's) of each robotic arm, and the final position/orientation of the master arm. Also specify the desired time t_f for the task completion.
- *Step 1*: Compute the linear displacement vector $\mathbf{p} = [x, y, z]^\mathrm{T}$, the axis of rotation $\mathbf{r} = [r_x, r_y, r_z]$ and the total rotation angle ϕ_* from Equations (10a)–(10c). Determine the number of elementary segments to which the motion from the initial to the final position/orientation will be splitted, and compute the corresponding $\mathrm{d}\mathbf{p}$ and $\mathrm{d}\boldsymbol{\phi}$ of each one of them.
- *Step 2*: Set $\delta q_i = 0$ $(i = 1, 2, \ldots, 6)$.
- *Step 3*: At each time t compute $\delta\mathbf{p}^m$, $\delta\boldsymbol{\phi}^m$; $\delta\mathbf{p}^{s_1}$, $\delta\boldsymbol{\phi}^{s_1}$; $\delta\mathbf{p}^{s_2}$, $\delta\boldsymbol{\phi}^{s_2}$; using (16).
- *Step 4*: Using the $\delta\mathbf{p}^j$ and $\delta\boldsymbol{\phi}^j$ $(j = m, s_1, s_2)$, found in step 3, compute the δq_i^j $(j = m, s_1, s_2; i = 1, \ldots, 6)$ by solving the Jacobian Equation (17).
- *Step 5*: Update the q_i^j's as: $q_{i,\mathrm{new}}^j = q_{\mathrm{old}}^j + \delta q_i^j$ and repeat from step 3, until the final time t_f is reached. Here of course $q_{i,\mathrm{new}}^j$ is used as initial value of q_i^j at the next time instant $t + \delta t$.

5. Representative Simulation Example

The above scheme has been tested in simulation using different triads of robots (KUKA, ABB, STÄUBLI). Here we give a representative example where three Stäubli RX-90L robots were used. RX-90 has a kinematic structure similar to that of a PUMA 700 robot with 6 rotational degrees of freedom and a spherical workspace of a radius around 120 cm.

A series of numerical simulations has been performed. The modelled task consists of picking up a horizontal plate and performing a vertical translation of 30 cm

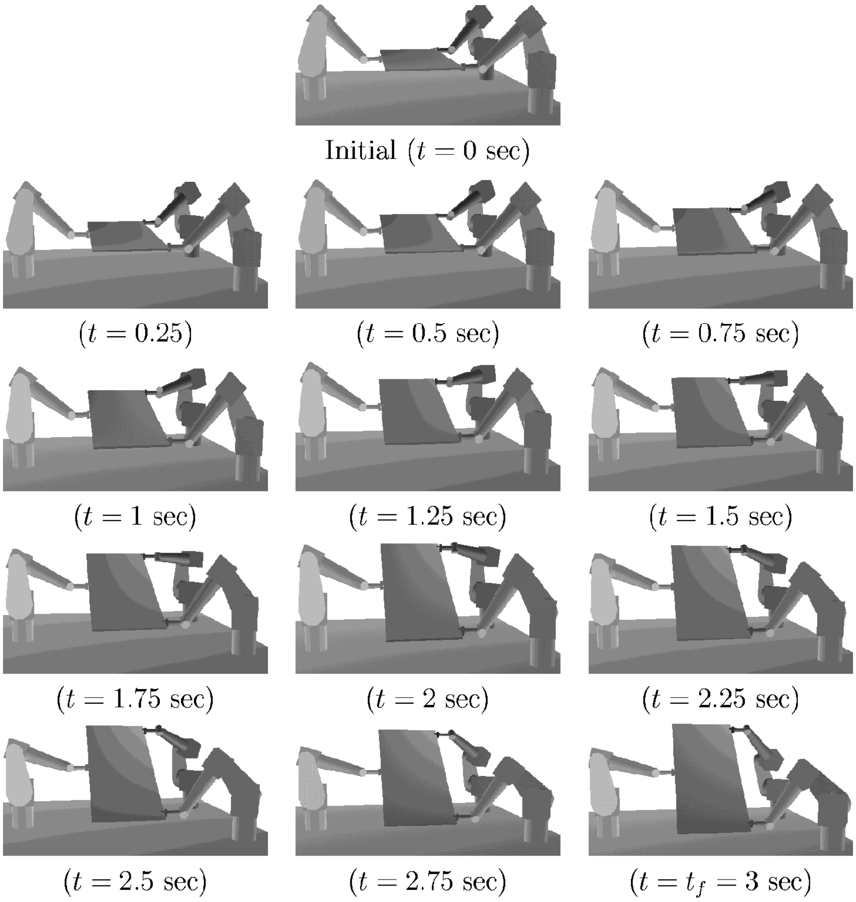


Figure 4. Graphical animation of the simulated 3-robot coordination task. A sequence of configurations: initial ($t = 0$), intermediate ($t = 0.25-2.75$ sec) and final configuration ($t = 3$ sec).

as well as a rotation of 40 degrees about an axis parallel to the x-axis of the master-robot coordinate frame. The dimensions of the plate are taken to be ($180 \times 80 \times 4$) cm.

Initial and final configurations (as well as eleven intermediate ones) are shown in Figure 4. The motion of each robot is planned by making small incremental, linear and angular displacements, as discussed in Section 4. In order to test the efficiency of the method, we varied the number N of increments. To evaluate quantitatively the performance of the algorithm we used a "relative-positioning error" measure ε_p, defined as*

$$\varepsilon_p = \sqrt{e^2_{p_{s1,m}} + e^2_{p_{s2,m}} + e^2_{p_{s2,s1}}},$$

* A similar expression was computed and used for the "relative-orientation error". The results are analogous to those obtained for the "relative-positioning error".

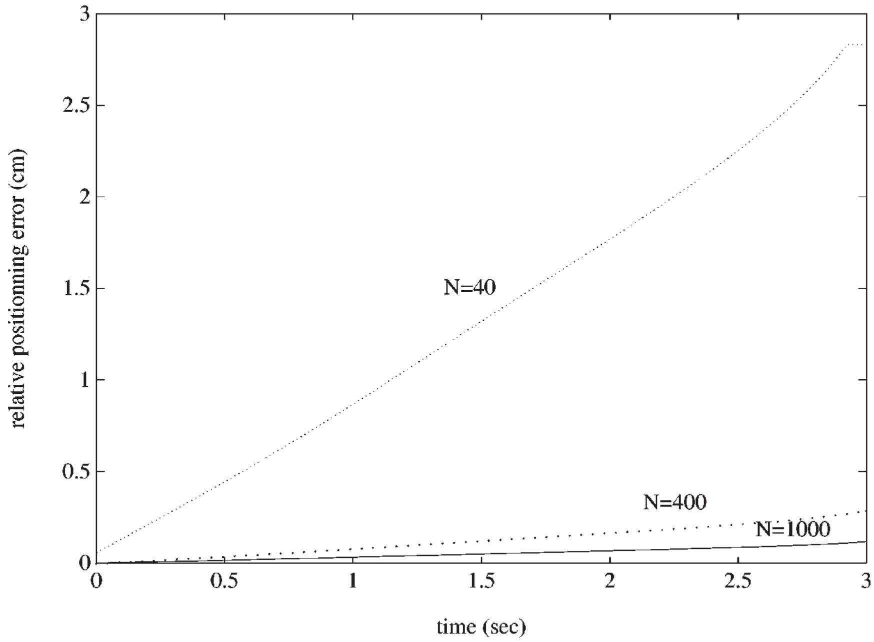


Figure 5. Relative-positioning error for the robots' end-effectors.

where

$$e^2_{p_{j,i}} = \left|\mathbf{p}^{(i)}_{j,i} - \mathbf{d}^{(i)}_{j,i}\right|^2 \quad (i = m, s_1,\ j = s_1, s_2 \text{ with } i \neq j),$$

$\mathbf{p}^{(i)}_{j,i}$: the relative position of the j-robot end-effector, with respect to the i-robot endpoint, expressed in the ith robot local tool frame.

$\mathbf{d}^{(i)}_{j,i}$: the desired (reference), relative-position vectors from the i- to the j-robot end-effector, expressed in the local ith tool frame. These reference position vectors are imposed by the geometry of the manipulated object and the choice of the grasping points. In our case:

$$\mathbf{d}^{(m)}_{s_1,m} = \left[\beta, 0, 3\alpha\right]^{\mathrm{T}}, \qquad \mathbf{d}^{(m)}_{s_2,m} = \left[-\beta, 0, 3\alpha\right]^{\mathrm{T}}, \qquad \mathbf{d}^{(s_1)}_{s_2,s_1} = \left[2\beta, 0, 0\right]^{\mathrm{T}}.$$

This error gives a measure of the magnitude of the "internal forces" that may appear during execution of the task. Figure 5 shows the results obtained for three different numbers N of differential increments ($N = 40, 400, 1000$) and $t_f = 3$ sec. The presence of cumulative errors is practically eliminated (inferior to 1 mm) if sufficient number of steps ($N = 400, 1000$) is used, which corresponds to a differential linear displacement of less than 1 mm and a differential angular displacement of 0.1 degrees or less. Satisfying these conditions, the obtained results show that the proposed method can be easily implemented and efficient for the case of three-robots coordinated task.

6. Conclusions

A path planning method for the trajectory control of three cooperating robots is presented in this paper. The proposed algorithm consists of performing incremental, linear and angular displacements which are computed, using homogeneous transformations, from the desired motion of the manipulated object.

Numerical simulations show the applicability and the effectiveness of the proposed method, under certain conditions regarding the magnitude of the differential displacements, which is related to the number of increments used. Nevertheless, complete elimination of cumulative errors may require the use of the inverse geometric model of the robots in a periodic way, in order to reinitialize the undesirable resulting relative positionning errors. This is under current implementation.

Further work is in progress by the present authors in the area of 3-robot systems. This includes the problems of decoupling/decomposition [55–57], hybrid position/force control [58] and robust trajectory control using the sliding mode approach [59–61]. Another problem which will be considered is the manipulation of large objects with three cooperating robots equipped with open-palm end effectors. This will be usefull for the cases where two robots are not sufficient to do the job (e.g., when long cylindrical, or large spherical objects without handles have to be transported).

References

1. Freund, E. and Hoyer, H.: Collision avoidance in multi-robot systems, in: H. Hanafusa and H. Inoue (eds), *Robotics Research: The Second Int. Symp.*, MIT Press, Cambridge, MA, 1985, pp. 135–146.
2. Alford, C. O. and Belyeu, S. M.: Coordinated control of two robot arms, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA, March 1984, pp. 468–473.
3. Freund, E.: On the design of multirobot systems, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA, March 1984, pp. 477–490.
4. Fujii, S. and Kurono, S.: Coordinated computer control of a pair of manipulators, in: *Proc. 4th IFTOMM World Congress*, Univ. of Newcastle, U.K., September 1975, pp. 411–417.
5. Zapata, R., Fournier, A., and Dauchez, P.: True cooperation of robots in multi-arms tasks, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, 1987, pp. 1255–1260.
6. Fortune, S., Wilfong, G., and Yap, C.: Coordinated motion of two robot arms, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, San Fransisco, CA, 1986, pp. 1216–1223.
7. Hoyer, H.: On-line collision avoidance for industrial robots, in: *Proc. IFAC Symp. on Robot Control (SYROCO'85)*, Barcelona, Spain, 1985, pp. 477–485.
8. Henrich, D. and Cheng, X.: Fast distance computation for on-line collision detection with multi-arm robots, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Nice, France, 1992, pp. 2514–2519.
9. Tournassoud, P.: A strategy for obstacle avoidance and its application to multi-robot systems, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, San Fransisco, CA, 1986, pp. 1224–1229.
10. Ishida,T.: Force control in coordination of two arms, in: *Proc. 5th Int. Conf. on Artificial Intelligence*, August 1977, pp. 411–415.
11. Kim, K. I. and Zheng, Y. F.: Two strategies of position and force control for two industrial robots handling a single object, *Robot. Autonom. Systems* **5** (1989), 395–403.

12. Koivo, A. J.: Adaptive position-velocity-force control of two manipulators, in: *Proc. 24th IEEE Conf. on Decision and Control*, Ft. Lauderdale, FL, December 1985, pp. 334–337.
13. Koivo, A. J. and Bekey, G. A.: Report of workshop on coordinated multiple robot manipulators: Planning, control and applications, *IEEE J. Robotics Automat.* **4**(1) (1988), 91–93.
14. Lim, J. and Ghyung, D. H.: On a control scheme for two cooperating robot arms, in: *Proc. 24th IEEE Conf. on Decision and Control*, Ft. Lauderdale, FL, December 1985, pp. 334–337.
15. Yoshikawa, T. and Zheng, X.: Coordinated dynamic hybrid position/force control for multiple robot manipulators handling constrained objects, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Cincinnati, Ohio, 1990, pp. 1178–1183.
16. Su, S.: Coordinated control of multiple manipulator systems: Experimental results, in: *Proc. 1992 Int. Conf. on Robotics and Automation*, Nice, France, 1992, pp. 2199–2204.
17. Schneider, S. A. and Cannon, R. H., Jr: Object impedance control for cooperative manipulation: theory and experimental results, *IEEE J. Robotics Automat.* **4**(1) (1988), 91–93.
18. Paljug, E., Yun, X., and Kumar, V.: Control of rolling contacts in multiple robotic manipulation, in: *Proc. Int. Conf. on Advanced Robotics*, Pisa, Italy, 1991, pp. 591–596.
19. Paljug, E. and Yun, X.: Experimental study of two robot arms manipulating large objects, *IEEE Trans. Control Systems Tech.* **3**(2) (1995), 177–188.
20. Zheng, Y. F. and Luh, J. Y. S.: Control of two coordinated robots in motion, in: *Proc. IEEE Conf. on Decision and Control*, Ft. Lauderdale, FL, December 1985, pp. 1761–1764.
21. Walker, M. W., Kim, D., and Dionise, J.: Adaptive coordinated motion control of two manipulator arms, in: *Proc. 1989 Int. Conf. Robotics and Automation*, Scottsdale, AZ, 1989, pp. 1084–1090.
22. Li, Z., Hsu, P., and Sastry, S.: Dynamic coordination of a multiple robotic system with point contact, in: *Proc. 1988 American Control Conf.*, Atlanta, GA, 1988, pp. 505–510.
23. Kreutz, K. and Lokshin, A.: Load balance and closed chain multiple arm control, in: *Proc. 1988 American Control Conf.*, Atlanta, GA, 1988, pp. 2148–2155.
24. McClamroch, H. N. and Wang, D.: Feedback stabilization and tracking of constrained robots, *IEEE Trans. Automat. Control* **33**(5) (1988), 419–426.
25. Nakamura, Y., Nagai, K., and Yoshikawa, T.: Mechanics of coordinative manipulation by multiple robotic mechanisms, in: *Proc. 1987 IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, 1987, pp. 991–998.
26. Hayati, S.: Hybrid position/force control of multi-arm cooperating robots, in: *Proc. 1986 IEEE Int. Conf. Robotics and Automation*, San Fransisco, CA, 1986, pp. 82–89.
27. Uchiyama, M. and Dauchez, P.: A symmetric hybrid position/force control scheme for the coordination of two robots, in: *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA, pp. 350–356.
28. Pittelkau, M. E.: Adaptive load-sharing force control for two-arm-manipulators, in: *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA, 1988, pp. 498–503.
29. Orin, D. E. and Oh, S. Y.: Control of force distribution in robotic mechanisms containing closed kinematic chains, *ASME J. Dyn. Syst. Meas. Control* **103**(2) (1981), 134–141.
30. Zheng, Y. F. and Luh, J. Y. S.: Optimal load distribution for two industrial robots handling a single object, in: *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, 1988, pp. 344–349.
31. Cheng, F. T. and Orin, D. E.: Efficient algorithm for optimal force distribution – The compact-dual LP method, *IEEE Trans. Robotics Automat.* **6**(2) (1990), 178–187.
32. Choi, M. H., Ko, M. S., and Lee, B. H.: Optimal load distribution for two cooperating robots using force ellipsoid, *Robotica* **11** (1993), 61–72.
33. Choi, M. H. and Lee, B. H.: A real-time optimal load distribution for multiple cooperating robots, in: *Proc. 1995 IEEE Conf. on Robotics and Automation*, 1995, pp. 1211–1216.
34. Lu, W. S. and Meng, Q. H.: An improved load distribution scheme for coordinating manipulators, in: *Proc. 1993 IEEE Int. Conf. on Robotics and Automation*, 1993, pp. 523–528.

35. Klein, C. A. and Kittivatcharapong, S: Optimal force distribution for the legs of a walking machine with friction cone constraints, *IEEE Trans. Robotics Automat.* **6**(1) (1990), 73–85.
36. Nahon, M. A. and Angeles, J.: Real-time force optimization in parallel kinematic chains under inequality constraints, *IEEE Trans. Robotics Automat.* **8**(4) (1992), 439–450.
37. Luecke, G. R. and Gardner, J. F.: Experimental results for force distribution in cooperating manipulator systems using local joint control, *Int. J. Robotics Res.* **13**(6) (1994), 471–480.
38. Kwon, W. and Lee, B. H.: Optimal force distribution of multiple cooperating robots using nonlinear programming dual method, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, 1996, pp. 2408–2413.
39. Unseren, M. A. and Koivo, A. J.: Reduced-order model and decoupled control architecture for two manipulators holding an object, in: *Proc. 1989 IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, 1989, pp. 1240–1245.
40. Unseren, M. A.: Rigid body dynamics and decoupled control architecture for two strongly interacting manipulators, *Robotica* **9**(4) (1991), pp. 421–430.
41. Hsu, P, Li, Z., and Sastry, S.: On grasping and coordinated manipulation by a multifingered robot hand, in: *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, 1988, pp. 384–389.
42. Wen, J. and Kreutz, K.: Motion and force control for multiple cooperative manipulators, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, 1989, pp. 1246–1251.
43. Dellinger, W. F. and Anderson, J. N.: Interactive force dynamics of two robotic manipulators grasping a nonrigid object, in: *Proc. 1992 IEEE Int. Conf. on Robotics and Automation*, 1992, pp. 2205–2210.
44. Hayati, S. A.: Position and force control of coordinated multiple arms, *IEEE Trans. Aerospace Electronic Systems* **24**(5) (1988), 584–590.
45. Ahmad, S. and Guo, H.: Dynamic Coordination of dual-arm robotic systems with joint flexibility, in: *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, 1988, pp. 332–337.
46. Yun, X.: Object handling using two arms without grasping, *Int. J. Robotics Res.* **12**(1) (1993), 99–106.
47. Yun, X.: Coordination of two-arm pushing, in: *Proc. 1991 IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, 1991.
48. Roach, J. W. and Boaz, M. N.: Coordinating the motions of robot arms in a common space, *IEEE J. Robotics Automat.* **3**(5) (1987), 437–444.
49. Chung, W. and Waldron, K. J.: Force distribution by optimizing friction angles for multifinger system, in: *Proc. 1993 IEEE Int. Conf. on Robotics and Automation*, 1993, 717–722.
50. Kerr, J. and Roth, B.: Analysis of multifingered hands, *Int. J. Robotics Res.* **4**(4) (1986), 3–17.
51. Yun, X., Kumar, V., Sarkar, N., and Paljug, E.: Control of multiple arms with rolling constraints, in: *Proc. 1992 IEEE Int. Conf. on Robotics and Automation*, Nice, France, 1992.
52. Tzafestas, S. G.: Digital PID and self tuning control, in: S. G. Tzafestas (ed.), *Applied Digital Control*, North-Holland, Amsterdam, 1985, pp. 1–49.
53. Tzafestas, S. G. and Pal, J. K. (eds): *Real-Time Microcomputer Control of Industrial Processes*, Kluwer, Dordrecht/Boston, 1991.
54. Mayorga, R. V. and Wong, A. K. C.: A robust method for the concurrent motion planning of multi-manipulators systems, *J. Intell. Robotics Systems* **19**(1) (1997), 73–88.
55. Tzafestas, S. G. (ed.): *Intelligent Robotic Systems*, Marcel Dekker, New York, 1991.
56. Tzafestas, S. G., Stavrakakis, G., and Zagorianos, A.: Robot model reference adaptive control through lower/upper part dynamic decomposition, *J. Intell. Robotic Systems* **1** (1988), 163–184.
57. Stavrakakis, G., Zagorianos, A., and Tzafestas, S. G.: Combined Euler–Lagrange/Newton–Euler robot modelling and identification, in: S. G. Tzafestas, A. Eisenberg, and L. Carotenuto (eds), *System Modelling and Simulation*, North-Holland, 1989, pp. 209–220.
58. Vukobratovic, M.: How to control robots interacting with dynamic environment, *J. Intell. Robotic Systems* **19** (1997), 119–152.

59. Tzafestas, S. G., Raibert, M., and Tzafestas C. S.: Robust sliding-mode control applied to a 5-link biped robot, *J. Intell. Robotic Systems* **15**(1) (1996), 67–133.
60. Tzafestas, S. G. and Prokopiou, P. A.: Compensation of teleoperator modeling uncertainties with a sliding-mode controller, *Robotics and Computer Integrated Manufacturing* **13**(1) (1997), 9–20.
61. Slotine, J.-J. and Li, W.: *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.