Control and Cybernetics

vol. **29** (2000) No. 2

A differential motion planning algorithm for controlling multi-robot systems handling a common object

by

C.S. Tzafestas, P.A. Prokopiou and S.G. Tzafestas

Intelligent Robotics and Automation Laboratory
Department of Electrical and Computer Engineering
National Technical University of Athens
Zografou 15773, Athens, Greece
E-mail: tzafesta@softlab.ece.ntua.gr

Abstract: Multi-robot systems have substantially increased capabilities over single-robot systems and can handle very large or peculiar objects. This paper presents a differential (incremental) motion planning algorithm for an m-robot system ($m \geq 2$) to cooperatively transfer an object from an initial to a desired final position / orientation by rigidly holding it at given respective points $Q_1, Q_2, \ldots, Q_m$. One of the robots plays the role of a "master" while other robots operate in the "slave" mode maintaining invariant their relative positions and orientations during the system motion. The method employs the differential displacements of the end-effector of each robot arm. Then, the differential displacements of the joints of the m robots are computed for the application of incremental motion control. The algorithm was tested on many examples. A representative of them is shown here, concerning the case of three STAUBLI RX-90L robots similar to 6-dof PUMA robots. The results obtained show the practicality and effectiveness of the method, which, however, needs particular care for completely eliminating the cumulative errors that may occur.

Keywords: multi-robot systems, cooperative robots, incremental robot motion planning, master and two-slaves system, multi-robot kinematics, rigidity condition.

1. Introduction

Multi-robot systems have attracted increasing attention over the years for both their theoretical and practical interest, with many important results already available (Alford and Belyeu, 1984, Freund, 1984, Fujii and Kurono, 1975, Ishida, 1977, Kim and Zheng, 1989, Koivo, 1985, Koivo and Bekey, 1988, Lim

Freund and Hoyer, 1985, Hoyer, 1985, Tournassoud, 1986, Zapata et al., 1987, Schneider and Cannon, Jr., 1988, Yoshikawa and Zheng, 1990, Paljug et al., 1991, Henrich and Cheng, 1992, Su, 1992, Mayorga and Wong, 1997, Wang et al., 1997, Unseren and Koivo, 1989, Unseren, 1991, Choi and Lee, 1995, Kwon and Lee, 1996). Many industrial operations and tasks can be performed efficiently by a single robot with or without redundant degrees of freedom (Tzafestas, 1991a, Zagorianos, Tzafestas and Stavrakakis, 1995, Zagorianos, Kontoyiannis and Tzafestas, 1994, Tzafestas et al., 1988, Tzafestas, 1989, Stavrakakis et al., 1990, Tzafestas, 1991b, Tzafestas et al., 1996, Tzafestas and Prokopiou, 1997). However, there are tasks which need two or more cooperating robots for satisfactory and economic performance. The case of two cooperating robots handling large objects or long flexible bars has been investigated by several researchers (Alford and Belyeu, 1984, Freund, 1984, Fujii and Kurono, 1975, Ishida, 1977, Kim and Zheng, 1989, Koivo, 1985, Koivo and Bekey, 1988, Lim and Ghyung, 1985, Paljug and Yun, 1995, Zheng and Luh, 1985). Most of these publications present theoretical investigations and only a few provide practical experimental studies (e.g. Paljug and Yun, 1995). For example, in Freund (1984) feedback linearization is introduced, and the pole placement technique is applied to the desired linear state-space model. In Fujii and Kurono (1975), each joint is controlled by a proportional type controller with the error being expressed in Cartesian space. In Ishida (1977), 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 Koivo (1985), the controllers of the two arms are designed using the MIMO discrete ARX model with external inputs, where the parameters are estimated on-line recursively. Experimental real-time results are presented in Koivo (1985) for two PUMA 250 robot arms that manipulate large objects. In Wu (1997), the problem of generating collision-free, near time-optimal trajectories for two cooperative redundant robots between two sets of end-points is treated. First, the time-optimal trajectory of one robot is found, and then the collision-free trajectory for the other robot is determined, by regarding the first robot as a moving obstacle. Then, the travelling time is minimized by an iterative scheme which scales down the time profiles of the robot trajectories.

Although the capabilities of 2-robot-systems are considerably increased over single-robot-systems, they are still unable to handle (grasp, manipulate, transfer etc.) very large, very heavy or flexible objects. Therefore, attention must be turned to the case of using three (or more) cooperating robots. Some studies of multi-robot systems can be found in (Freund and Hoyer, 1985, Hoyer, 1985, Tournassoud, 1986, Zapata et al., 1987, Schneider and Cannon, Jr., 1988, Yoshikawa and Zheng, 1990, Paljug et al., 1991, Henrich and Cheng, 1992, Su, 1992, Mayorga and Wong, 1997, Wang et al., 1997, Unseren and Koivo, 1989, Unseren, 1991, Choi and Lee, 1995, Kwon and Lee, 1996). In general, the prin-

1988), combined position and force control (e.g. Koivo, 1985, Yoshikawa and Zheng, 1990. Wang et al., 1997), decoupling control (e.g. Unseren and Koivo, 1989, Unseren, 1991) and force / load distribution (e.g. Choi and Lee, 1995, Kwon and Lee, 1996). For example, in Mayorga and Wong (1997) a robust scheme for on-line concurrent motion planning of multi-robot systems is developed, which uses a linear set of equations for each robot and takes into account a vector for motion planning. This scheme can coordinate in real-time the motion of the robots, and prevent singularities employing sensor-based information. In Wang et al. (1997), the control problem of multi-robot systems is decomposed into motion-control and internal-force control. It is shown that, under the rigidity assumption (no slippage of the end-effectors on the object), the motion control subsystem does not depend on the internal force control, and so any advanced motion-control law developed for a single robot can be applied directly to the motion control of the multi-robot system. The above paper contains experimental results for a system consisting of two RTX robotic manipulators.

The purpose of the present paper is to treat the problem of motion planning of m-robot systems aiming at moving large objects from an initial to a desired position / orientation under the rigidity assumption. The algorithm is based on the technique of Paljug and Yun (1995) which is properly extended to the m-robot case ($m \geq 2$). Single robot tasks can be performed by controlling the robot's end-effector such as to follow a desired path, without controlling the exact time at which the end-effector passes through the particular points on the trajectory. The orientation of the robot's end-effector during the motion may also be irrelevant. This is not true in multi-robot systems, where, once the two or more end-effectors 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 end-effector must pass through a particular point on its trajectory at exactly the right time, and the orientations of the end-effectors must also be the proper ones.

Section 2 presents the multi-robot kinematic equations and the general kinematic constraints which the robots have to satisfy due to the rigidity condition. Section 3 derives the absolute and incremental motion equations of the m-robot system, and develops the proposed differential motion planning algorithm. Section 4 provides the full study of a 3-robot (master-and-two-slaves) example, where it is assumed (without loss of generality) that the three grasping points define an isosceles triangle, and that the three robots are placed in a symmetric lay-out on the shop floor. The performance of the system is expressed using suitable "relative positioning and orientation" error measures. Finally, Section 5 gives the conclusions and indicates some directions for further investigation.

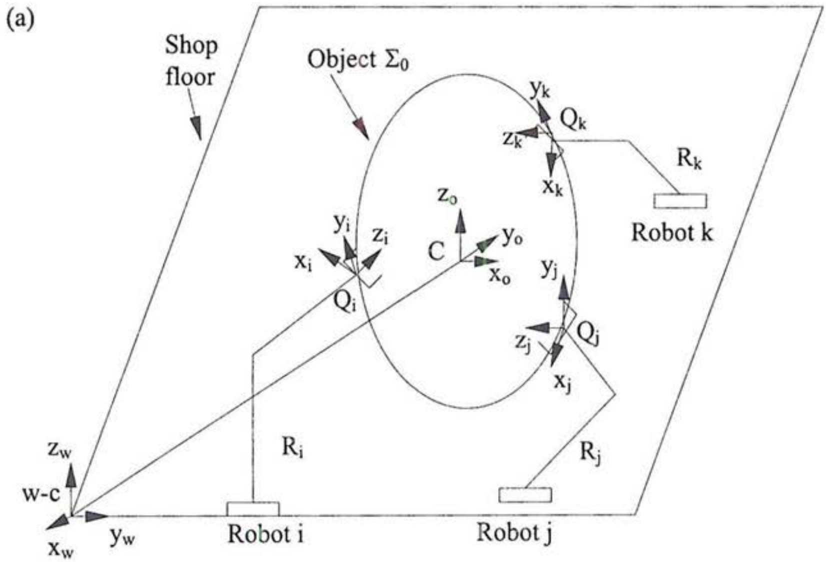
2. Multi-robot kinematics

Consider m robots R_i ($i = 1, 2, \ldots, m$) in a common work space, which aim to

(x_w, y_w, z_w), then the position and orientation of the object with respect to w-c is given by an homogenous matrix $\underline{A}^{\mathrm{o}}$:

$$\underline{A}^{\mathrm{o}} = \left[\begin{array}{ccc:c} \underline{n} & \underline{o} & \underline{a} & \underline{p} \\ \hdashline & 0 & & 1 \end{array} \right], \ \underline{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{1}$$

where the 3-dimensional vectors $\underline{n}, \underline{o}, \underline{a}$ define the orientation of the object, and $\underline{p}$ is its position vector (position of the origin of the coordinate system $[\underline{n}, \underline{o}, \underline{a}]$, which is usually placed at the center of gravity of the object).



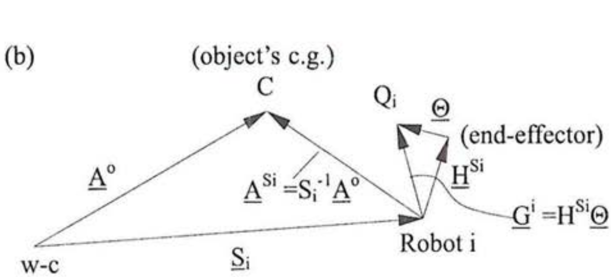


Figure 1. (a) An object handled by a system of m cooperating robots (C = object's center of gravity, $(x_{\mathrm{o}}, y_{\mathrm{o}}, z_{\mathrm{o}})$ = object coordinate frame, (x_i, y_i, z_i) = coordinate frame fixed to the i-th end effector). (b) Arrow diagram of the

The position and orientation of this object with respect to the coordinate systems $\underline{S}_i$ $(i = 1, 2, \ldots, m)$ of the bases of the robots is given by (see Fig. 1b):

$$\underline{A}^{Si} = \underline{S}_i^{-1}\underline{A}^{\mathrm{o}},\ i = 1, 2, \ldots, m. \tag{2}$$

In practice, the relative positions of the bases of the m robots on the shopfloor influence the shape of the robot workspaces as well as the overall motion of the system.

Figure 1a shows an object grasped by the m cooperating robots in a workspace with a total of n degrees of freedom.

The robots grasp the objects rigidly at the points Q_i $(i = 1, 2, \ldots, m)$, i.e. no slippage at the grasping points occurs. Therefore, one can either define the initial and final positions of these points, or the initial and final position of the center of gravity C of the object, plus the initial / final object's orientation. The initial and final positions, or the path of the object, defined in one of the above two ways, are used to determine the motion 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 corresponding robot-base reference frame is described by an homogenous 4×4 transformation matrix $\underline{H}^{Si}$ of the type described by (1). The coordinate systems of the end-effectors and of the grasping points Q_i $(i = 1, 2, \ldots, m)$ are as shown in Fig. 2.

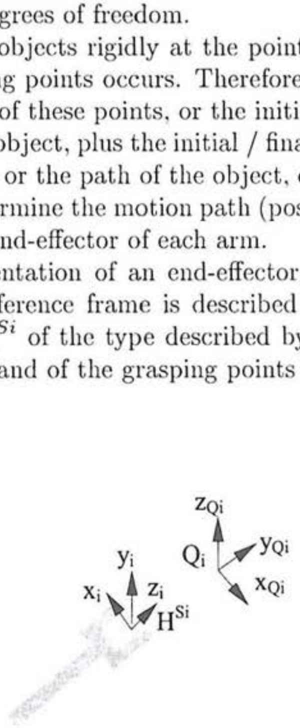


Figure 2. Coordinate systems attached to the end-effector i and grasping point Q_i

Therefore, if $\underline{G}^i$ is the coordinate system attached to the grasping point $\underline{Q}_i$ $(i = 1, 2, \ldots, m)$ expressed in the corresponding robot reference frame, then

$$\underline{G}^i = \underline{H}^{Si}\underline{\Theta} \tag{3a}$$

where

$$\underline{\Theta} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3b}$$

is the fixed transformation from $\underline{H}^{Si}$ to $\underline{G}^{i}$ (see Fig. 2). Usually, we consider one of the robots (say the robot j) to be the master robot and define the coordinate systems attached to the points Q_i ($i = 1, 2, \ldots, m$, $i \neq j$), with respect to the coordinate system attached to the master grasping point Q_j. This calculation depends on the geometry defined by the Q_i's.

The generalized Cartesian position / orientation vector $\underline{P}$ of the object frame (x_o, y_o, z_o) attached to its center of gravity can be expressed in terms of the joint position vector $\underline{q}_i$ of the robot as:

$$\underline{P} = \underline{L}_i(\underline{q}_i), \; i = 1, 2, \ldots, m. \tag{4a}$$

Clearly, $\underline{P}$ involves the Cartesian coordinates of the object's center of gravity C and the angles of rotation of the object's frame. Thus, inverting (4a) gives:

$$\underline{q}_i = \underline{L}_i^{-1}(\underline{P}), \; i = 1, 2, \ldots, m. \tag{4b}$$

From (4a) it follows that the joint positions of the m robots must satisfy the following kinematic constraints:

$$\underline{L}_1(\underline{q}_1) = \underline{L}_2(\underline{q}_2) = \ldots = \underline{L}_m(\underline{q}_m) \tag{5}$$

If $\underline{J}_{oi}$ is the transformation matrix from the object frame to the ith robot end-effector frame, and $\underline{J}_i$ is the Jacobian matrix of the ith robot, then

$$\underline{\dot{P}} = \underline{J}_{0Li}\underline{\dot{q}}_i, \; i = 1, 2, \ldots, m \tag{6a}$$

where $\underline{J}_{0Li} = \partial L_i(q_i)/\partial q_i$ is practically computed by:

$$\underline{J}_{oLi} = \underline{J}_{oi}^{-1} J_1. \tag{6b}$$

Therefore, the following velocity constraints hold:

$$\underline{J}_{0L1}\underline{\dot{q}}_1 = \underline{J}_{0L2}\underline{\dot{q}}_2 = \ldots = \underline{J}_{0Lm}\underline{\dot{q}}_m. \tag{7}$$

Differentiating (6a) and (7) yields the respective relations and constraints for the object's generalized Cartesian-space acceleration vector and the joint accelerations of the m robots, i.e:

$$\underline{\ddot{P}} = \underline{\dot{J}}_{0Li}\underline{\dot{q}}_i + \underline{J}_{0Li}\underline{\ddot{q}}_i \tag{8a}$$

$$\underline{\dot{J}}_{0L1}\underline{\dot{q}}_1 + \underline{J}_{0L1}\underline{\ddot{q}}_1 = \underline{\dot{J}}_{0L2}\underline{\dot{q}}_2 + \underline{J}_{0L2}\underline{\ddot{q}}_2 = \ldots = \underline{\dot{J}}_{0Lm}\underline{\dot{q}}_m + \underline{J}_{0Lm}\underline{\ddot{q}}_m \tag{8b}$$

Equations (4a), (6a) and (8a) can be used for checking if any constraints regarding the positions, velocities and accelerations of the joints are violated by the desired position, velocity and acceleration of the object grasped and transferred by the robots. Equations (5), (7) and (8b) represent the kinematic conditions for

3. Multi-robot motion planning

3.1. Absolute motion equations

Here, the motion equations of the object in space will be provided. Consider first the motion of the point Q_j (grasped by the master arm). This motion is defined by a time-varying homogeneous transformation matrix $\underline{M}(t)$ which determines the linear and angular displacements needed for the point Q_j to go from the initial to the desired final position and orientation. The matrix $\underline{M}(t)$ is given by

$$\underline{M}(t) = \tag{9}$$
$$\begin{bmatrix} r_x r_x \upsilon(\tau\varphi) + \mathrm{c}(\tau\varphi) & r_x r_y \upsilon(\tau\varphi) - r_z \mathrm{s}(\tau\varphi) & r_z r_x \upsilon(\tau\varphi) + r_y \mathrm{s}(\tau\varphi) & \tau \cdot x \\ r_x r_y \upsilon(\tau\varphi) + r_z \mathrm{s}(\tau\varphi) & r_y r_y \upsilon(\tau\varphi) + \mathrm{c}(\tau\varphi) & r_y r_z \upsilon(\tau\varphi) - r_x \mathrm{s}(\tau\varphi) & \tau \cdot y \\ r_x r_z \upsilon(\tau\varphi) - r_y \mathrm{s}(\tau\varphi) & r_y r_z \upsilon(\tau\varphi) + r_x \mathrm{s}(\tau\varphi) & r_z r_z \upsilon(\tau\varphi) + \mathrm{c}(\tau\varphi) & \tau \cdot z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

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

$$\underline{G}^A(t) = \underline{G}^A(0) \cdot \underline{M}(t) \tag{10}$$

and the final one is given by

$$\underline{G}^A_f = \underline{G}^A(t_f) = \underline{G}^A(0) \cdot \underline{M}(t_f) \tag{11}$$

where

$$x = \underline{n}^{\mathrm{T}}(0)[\underline{p}(t_f) - \underline{p}(0)],\ y = \underline{o}^{\mathrm{T}}(0)[\underline{p}(t_f) - \underline{p}(0)],$$
$$z = \underline{a}^{\mathrm{T}}(0)[\underline{p}(t_f) - \underline{p}(0)] \tag{12a}$$

$$\varphi_* = \cos^{-1}\left[\left(\frac{1}{2}\right)\left(\underline{n}^{\mathrm{T}}(0)\underline{n}(t_f) + \underline{o}^{\mathrm{T}}(0)\underline{o}(t_f) + \underline{a}_{\mathrm{T}}(0)\underline{a}(t_f) - 1\right)\right] \tag{12b}$$

$$\underline{r} = \begin{bmatrix} \underline{a}^{\mathrm{T}}(0)\underline{o}(t_f) - \underline{o}^{\mathrm{T}}(0)\underline{a}(t_f) \\ \underline{n}^{\mathrm{T}}(0)\underline{a}(t_f) - \underline{a}^{\mathrm{T}}(0)\underline{n}(t_f) \\ \underline{o}^{\mathrm{T}}(0)\underline{n}(t_f) - \underline{n}^{\mathrm{T}}(0)\underline{o}(t_f) \end{bmatrix}. \tag{12c}$$

The motion of the other points Q_i $(i \neq j)$ of the object is defined by

$$\underline{G}^i(t) = \underline{S}_i^{-1}\underline{G}^A(0)\underline{M}(t)\underline{K}_i^*,\ i = 1, 2, 3, \ldots, m,\ i \neq j \tag{13}$$

where the matrices $\underline{S}_i$ $(i \neq j, i = 1, 2, \ldots, m)$ define the coordinate frames of the

Q_i $(i \neq j)$ with respect to the coordinate frame attached to Q_j (grasped by the master). The arrow diagram of the transformations involved in (13) is shown in Fig. 3.

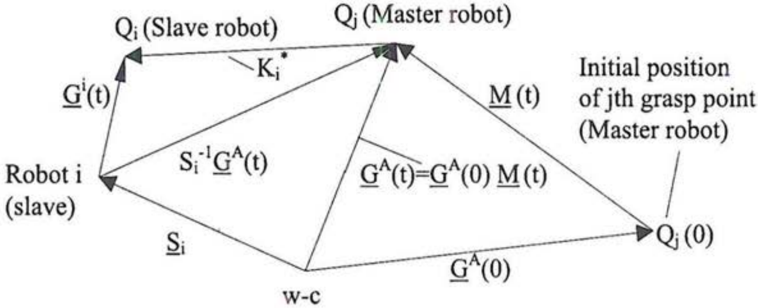


Figure 3. Arrow diagram of the transformations $\underline{G}^A(0)$ (master), $\underline{M}(t)$, $\underline{G}^A(t)$, $\underline{S}_i$, $\underline{K}_i^*$, and $\underline{G}^i(t) = \underline{S}_i^{-1}\underline{G}^A(0)\underline{M}(t)\underline{K}_i^*$ {see Eq. (13)}

The motion of the end-effectors of the robots at the point Q_j (for the master) and Q_i ($i \neq j$ for the slaves) is defined by the transformations $\underline{H}^M(t)$ and $\underline{H}^{Si}(t)$, $i \neq j$, respectively, which can be determined by equating the right-hand sides of (see (3a,b)):

$$\underline{G}^A(t) = \underline{H}^M(t) \cdot \underline{\Theta},\ \underline{G}^i(t) = \underline{H}^{Si}(t) \cdot \underline{\Theta},\ i \neq j$$

with the right-hand sides of (10) and (13) respectively, and solving the resulting equations, namely:

$$\underline{H}^M(t) = \underline{G}^A(0)\underline{M}(t)\underline{\Theta},\ \underline{H}^{Si}(t) = \underline{S}_i^{-1}\underline{G}^A(0)\underline{M}(t)\underline{K}_i^*\underline{\Theta},\ i \neq j \tag{14}$$

where the relation $\underline{\Theta}^{-1} = \underline{\Theta}$ was used.

3.2. Incremental motion equations

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

$$\underline{D} = [d_x, d_y, d_z; d\varphi_x, d\varphi_y, d\varphi_z]^{\mathrm{T}}$$

be the differential motion vector, where d_x, d_y, d_z are differential linear displacements, and $d\varphi_x, d\varphi_y, d\varphi_z$ are differential angular displacements with respect to

Consider the master's grasping point Q_j. The coordinate system of Q_j at the time $(t + dt)$ is given by

$$\underline{G}^A(t+dt) = \underline{G}^A(t) + d\underline{G}^A(t) = \underline{G}^A(t)[\underline{I} + \underline{\Delta}] \tag{15a}$$

where $\underline{I}$ is the 4×4 unit matrix, and

$$\underline{\Delta} = \begin{bmatrix} 0 & -\mathrm{d}\varphi_z & \mathrm{d}\varphi_y & \mathrm{d}x \\ \mathrm{d}\varphi_z & 0 & -\mathrm{d}\varphi_x & \mathrm{d}y \\ -\mathrm{d}\varphi_y & \mathrm{d}\varphi_x & 0 & \mathrm{d}z \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{15b}$$

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

$$\underline{H}^k(t+\mathrm{d}t) = \underline{H}^k(t)[\underline{I} + \underline{\Delta}^k] \tag{16a}$$

$$\underline{\Delta}^k = \begin{bmatrix} 0 & -\mathrm{d}\varphi_z^k & \mathrm{d}\varphi_y^k & \mathrm{d}x \\ \mathrm{d}\varphi_z^k & 0 & -\mathrm{d}\varphi_x^k & \mathrm{d}y \\ -\mathrm{d}\varphi_y^k & \mathrm{d}\varphi_x^k & 0 & \mathrm{d}z \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{16b}$$

with $k = M$ for the master, and $k = S_i$ $(i \neq j,\ i = 1, 2, \ldots, m)$ for the slaves.

From the analysis of Section 3.1, it follows that:

$$\underline{H}^M(t) = \underline{G}^A(t)\underline{\Theta},\ \underline{S}_i\underline{H}^{Si}(t) = \underline{G}^A(t)\underline{K}_i^*\underline{\Theta},\ i = 1, 2, \ldots, m\ i \neq j. \tag{17}$$

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

Now, using (15a,b), (16a,b) and (17), and solving for Δ^M and Δ^{Si} $(i \neq j)$ one obtains:

$$\begin{aligned} \underline{\Delta}^M &= [\underline{H}^M(t)]^{-1}\underline{G}^A(t)\underline{\Delta}\underline{\Theta} \\ \underline{\Delta}^{Si}(t) &= [\underline{H}^{Si}(t)]^{-1}\underline{S}_i^{-1}\underline{G}^A(t)\underline{\Delta}\,\underline{K}_i^*\underline{\Theta},\ i = 1, 2, \ldots, m,\ i \neq j \end{aligned}$$

which by (17) reduces to:

$$\underline{\Delta}^M = \underline{\Theta}\,\underline{\Delta}\,\underline{\Theta} \tag{18a}$$

$$\underline{\Delta}^{Si}(t) = \underline{\Theta}\,\underline{K}_i^{*-1}\underline{\Delta}\,\underline{K}_i^*\underline{\Theta},\ i = 1, 2, \ldots, m,\ i \neq j. \tag{18b}$$

Equations (18a,b) give the displacements of the robot end-effectors in terms of the differential displacement matrix $\underline{\Delta}$ of the master's grasping point Q_j.

Using (15b) and the definition (3b) of $\underline{\Theta}, \underline{\Delta}^M$ in (18a) gives for the master robot:

$$\begin{aligned} \mathrm{d}\phi_x^M &= -\mathrm{d}\phi_x,\ \mathrm{d}\phi_y^M = \mathrm{d}\phi_x,\ \mathrm{d}\phi_z^M = \mathrm{d}\phi_y \\ \mathrm{d}x^M &= -\mathrm{d}x,\ \mathrm{d}y^M = \mathrm{d}z,\ \mathrm{d}z^M = \mathrm{d}y. \end{aligned} \tag{19}$$

Similar equations can be derived for the slave robots if one computes (18b) using

3.3. The differential motion planning algorithm

To develop the proposed incremental motion control algorithm (for each robotic arm) the total linear and angular displacement of the master grasping point Q_j ($\underline{p} = [x, y, z]^{\mathrm{T}}$ and φ_*) given by (12a,b) is divided in a large number of small (nearly infinitesimal) displacements $\delta\underline{p}$ and $\delta\underline{\varphi}$. (The planning algorithm uses practical small displacements which are denoted by $\delta\underline{p}$ and $\delta\underline{\varphi}$ to distinguish them from the theoretical differential displacements $\mathrm{d}\underline{p}$ and $\mathrm{d}\underline{\varphi}$.) From these displacements and the above relations one can compute the corresponding displacements $\delta\underline{p}^M, \delta\underline{\varphi}^M$, $\delta\underline{p}^{Si}, \delta\underline{\varphi}^{Si}$ ($i = 1, 2, \ldots, m,\ i \neq j$) of all robot arms.

Let q_k ($k = 1, 2, \ldots, 6$) be the displacement of each joint, and δq_k the corresponding small displacement. Then, one can write for the master robot arm:

$$\begin{aligned}&[\delta x^M, \delta y^M, \delta z^M; \delta\phi_x^M, \delta\phi_y^M, \delta\phi_z^M]^{\mathrm{T}}\\ &= \underline{J}^M(q_1, \ldots, q_6) \cdot [\delta q_1^M, \delta q_2^M, \ldots, \delta q_6^M]^{\mathrm{T}}\end{aligned} \tag{20}$$

where $\underline{J}^M$ is the Jacobian matrix of the master arm. Similar equations hold for the slave robot arms, too.

Given the small displacements $\delta x^M, \ldots, \delta\varphi^M$ (determined as previously discussed) one can find the corresponding δq_k^M ($k = 1, \ldots, 6$) by solving the Jacobian equation (20), assuming that the robot does not pass via, or very near to, the singular configurations. This must be tested by simulation prior to the application of the algorithm in a practical case.

On the basis of the above analysis the incremental control algorithm is as follows:

- Step 0 (Initialization): Determine the initial position (the q_js; $j = M, S_i$, $i = 1, 2, \ldots, m,\ i \neq M$) of each robotic arm, and the final position / orientation of the master arm. Also specify the desired time t_f for task completion.
- Step 1: Compute the linear displacement vector $\underline{p} = [x, y, z]^{\mathrm{T}}$, the axis of rotation $\underline{r} = [r_x, r_y, r_z]^{\mathrm{T}}$, and the total rotation angle φ_*, from the equation (12a,b,c). Determine the number of elementary segments into which the motion from the initial to the final position / orientation will be split, and compute the corresponding $\mathrm{d}\underline{p}$ and $\mathrm{d}\underline{\varphi}$ of each of them.
- Step 2: Set $\delta q_k^j = 0$ ($k = 1, 2, \ldots, 6$).
- Step 3: At each time t compute $\delta\underline{p}^M, \delta\underline{\varphi}^M, \delta\underline{p}^{Si}, \delta\underline{\varphi}^{Si}$ ($i = 1, 2, \ldots, m$, $i \neq M$) using (18a,b)
- Step 4: Using the $\delta\underline{p}^j, \delta\underline{\varphi}^j$ ($j = M, S_i = 1, 2, \ldots, m, i \neq M$) found in Step 3, compute the δq_k^j ($j = M,\ S_i, i = 1, 2, \ldots, m,\ i \neq M,\ k = 1, 2, \ldots, 6$) by solving the Jacobian equation (20).
- Step 5: Update the q_k^j's as

$$q_{k,\mathrm{new}}^j = q_{k,\mathrm{old}}^j = \delta q_k^j$$

and repeat from Step 3 until time t_f is reached. Here, of course, $q_{k,\mathrm{new}}^j$ is

The computational requirements of this algorithm are comparable to those of other m-robot motion planning / control algorithms (see e.g. Zapata et al., 1987, Wang et al., 1997). The application of suitable parallel task scheduling / grouping techniques can substantially reduce the actual computation / implementation time (Tzafestas, 1991c, Tzafestas and Triantafyllakis, 1993, 1994, Tzafestas et al., 1995), thus giving the algorithm more practical value.

4. A master-and-two-slaves example

Here a 3-robot system will be fully treated, where the robot R_1 is considered to be the master, cooperating with the two slave robots R_2 and R_3. It is assumed that the robots grasp a planar object at three points $A = Q_0$, $B = Q_1$ and $C = Q_2$, which define an isosceles triangle as shown Fig. 4. The coordinate frames attached

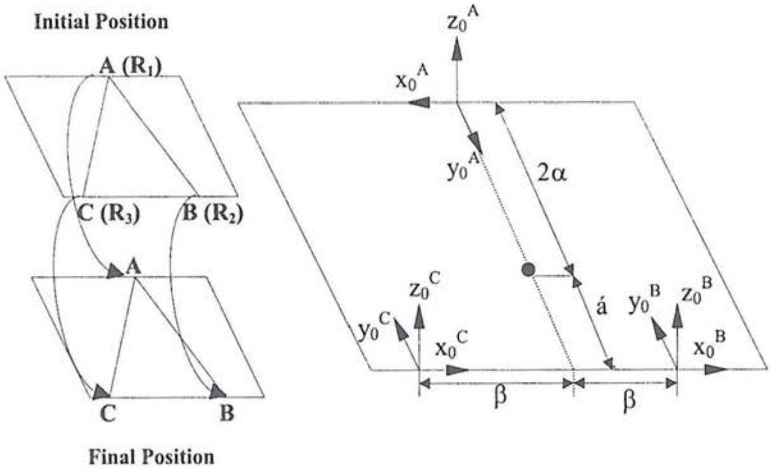


Figure 4. The master R_1 grasps the object at point A, and the slaves R_2 and R_3 at points B and C, respectively. The triangle ABC is isosceles: $(AB) = (AC)$

to the grasping points A, B and C, and expressed in the corresponding robot reference frame, are (see (3a,b) or (13)):

$$\underline{G}^A = \underline{H}^M \cdot \underline{\Theta},\ \underline{G}^B = \underline{H}^{S_1} \cdot \underline{\Theta},\ \underline{G}^C = \underline{H}^{S_2} \cdot \underline{\Theta}.$$

Here, the matrices $\underline{K}_B^*$ and $\underline{K}_C^*$ that define the coordinate frames attached to grasping point A, are (see Fig. 4):

$$\underline{K}_B^* = \begin{bmatrix} -1 & 0 & 0 & -\beta \\ 0 & -1 & 0 & 3a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; \underline{K}_C^* = \begin{bmatrix} -1 & 0 & 0 & \beta \\ 0 & -1 & 0 & 3a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{21}$$

The w-c reference frame is defined to be the coordinate frame $(x_\mathrm{o}, y_\mathrm{o}, z_\mathrm{o})$ of the master's base, and the 3-robot system is assumed to possess the symmetric configuration shown in Fig. 5.

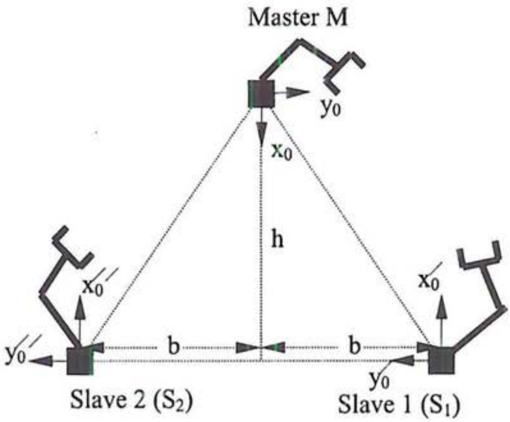


Figure 5. Symmetric master-and-two-slaves configuration (all axes $z_\mathrm{o}, z'_\mathrm{o}, z''_\mathrm{o}$ are normal to the plane M-S_1-S_2)

From Fig. 5 it follows that the matrices $\underline{S}_1$ and $\underline{S}_2$ defining the coordinate frames of the slaves 1 and 2, respectively, are given by:

$$\underline{S}_1 = \begin{bmatrix} -1 & 0 & 0 & h \\ 0 & -1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; \underline{S}_2 = \begin{bmatrix} -1 & 0 & 0 & h \\ 0 & -1 & 0 & -b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

One can see that in this symmetric configuration: $\underline{S}_i^{-1} = \underline{S}_i$ $(i = 1, 2)$, whereas the transformation from $x'_\mathrm{o}, y'_\mathrm{o}, z'_\mathrm{o}$ to $x''_\mathrm{o}, y''_\mathrm{o}, z''_\mathrm{o}$ is equal to:

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

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.

The displacements of the end-effectors of the two slaves in terms of the differential displacement matrix $\underline{\Delta}$ of the master at the grasping point A, are given by (18b). Therefore, using (3b), (15b) and (21) in (18b) one finds:

<u>Slave 1</u>

$$d\varphi_x^{S_1} = d\varphi_x,\ d\varphi_y^{S_1} = d\varphi_z,\ d\varphi_z^{S_1} = -d\varphi_y$$
$$dx^{S_1} = dx - 3\alpha d\varphi_z,\ dy^{S_1} = dz + \beta d\varphi_y + 3\alpha d\varphi_z,\ dz^{S_1} = -dy + \beta d\varphi_z$$

<u>Slave 2</u>

$$d\varphi_x^{S_2} = d\varphi_x,\ d\varphi_y^{S_2} = d\varphi_z,\ d\varphi_z^{S_2} = -d\varphi_y$$
$$dx^{S_2} = dx - 3\alpha d\varphi_z,\ dy^{S_2} = dz - \beta d\varphi_y + 3\alpha d\varphi_z,\ dz^{S_2} = -dy - \beta d\varphi_z.$$

The proposed differential motion algorithm was applied to several trials of simulated robots. Here we present the results obtained for three Stäubli RX-90L robots that are similar to the PUMA 700 robot, and possess six revolute joints and a workspace of a radius of about 120 cm.

The simulated task consists of picking up an horizontal plate and performing a vertical translation of 30 cm as well as 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 cm × 80 cm × 4 cm).

Initial and final configurations (as well as eleven intermediate ones) are shown in Fig. 6. The motion of each robot is planned by making small incremental, linear and angular displacements, as discussed in Section 3. 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

$$\varepsilon_p = \sqrt{e^2_{p,s1,M} + e^2_{p,s2,M} + e^2_{p,s2,s1}}$$
$$e^2_{p_{j,i}} = |\underline{p}^{(i)}_{j,i} - \underline{d}^{(i)}_{j,i}|^2 \ (i = M, S_1,\ j = S_1, S_2 \text{ with } i \neq j)$$

where $\underline{p}^{(i)}_{j,i}$ is the relative position of the j-th robot end-effector, with respect to the i-th robot endpoint, expressed in the i-th robot local tool frame and $\underline{d}^{(i)}_{j,i}$ is the desired (reference), relative-position vector from the i-th to the j-th robot end effector, expressed in the local i-th 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:

$$d^{(m)} = [\beta\ 0\ 3\alpha]^T,\ d^{(m)} = [-\beta\ 0\ 3\alpha]^T,\ d^{(S_1)} = [\ldots]^T$$

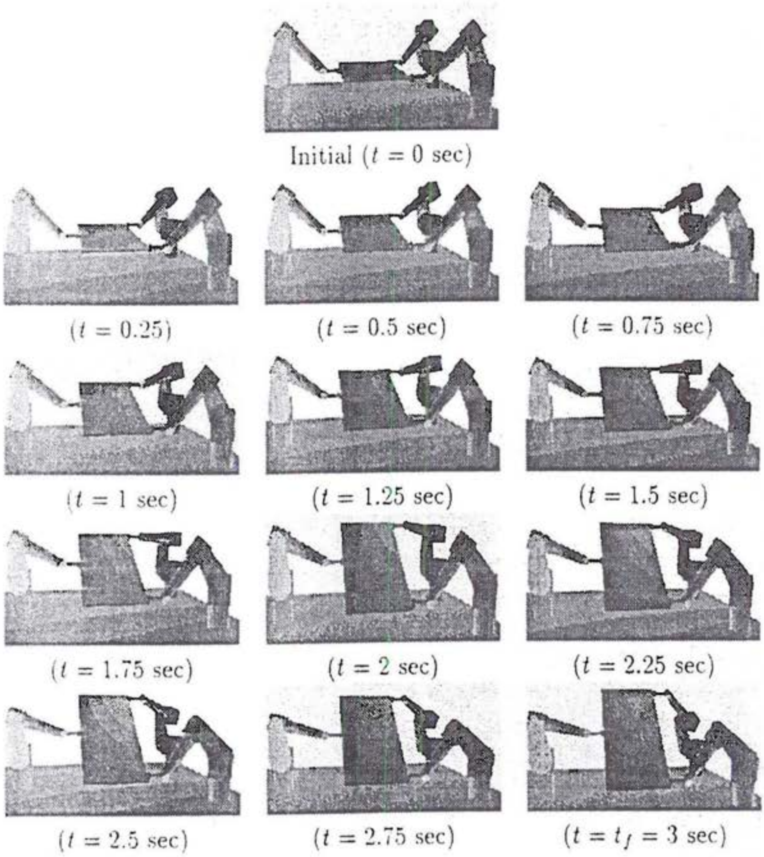


Figure 6. Graphical animation of the simulated 3-robot coordination task. A sequence of configurations: initial ($t = 0$), intermediate ($t = 0.25$–2.75 sec) and final configuration ($t = 3$ sec)

This error gives a measure of the magnitude of the "internal forces" that may appear during execution of the task. Fig. 7 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 (less than 1 mm) if sufficient number of steps ($N = 400, 1000$) is used, which corresponds to a differential angular displacement of 0.1 degrees or less. Satisfying these conditions, the results obtained show that the proposed method can be easily and efficiently implemented for the case of three-robot coordinate tasks.

A similar "relative — orientation error" expression was also used for the orientation trajectories of the end-effectors, which gave analogous results to

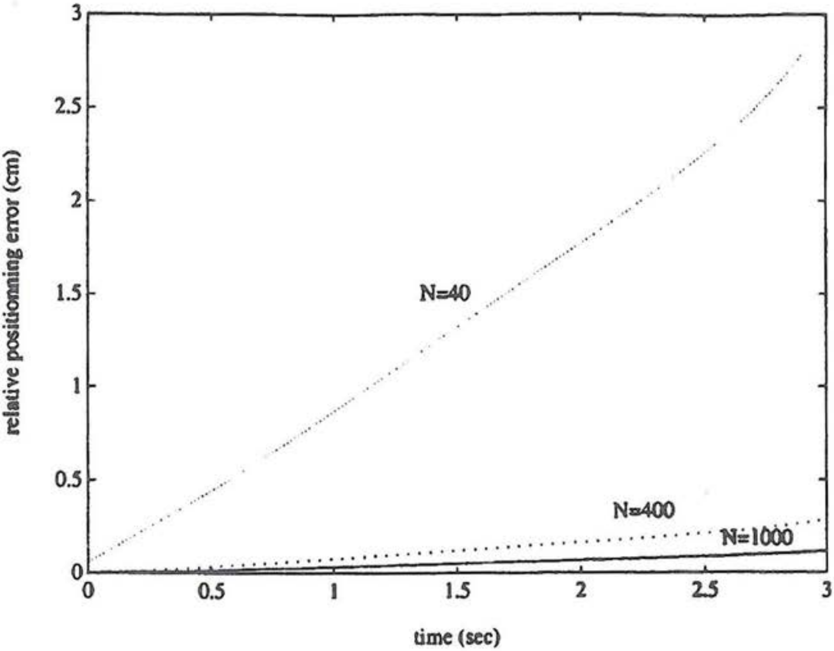


Figure 7. Relative — positioning error for the robots' end-effectors

5. Conclusions

In this paper we have considered the problem of transferring an object grasped by m cooperating robots from an initial to a desired position / orientation under the assumption that the end-effectors grasp rigidly the object (no slippage of the end-effectors at the grasping points is allowed). A path / motion planning algorithm was presented which consists of performing incremental linear and angular displacements computed from the desired motion of the manipulated object using homogeneous transformations.

Numerical simulations showed applicability 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 reference to the inverse geometric model of the robots in a periodic way, in order to reinitialize the resulting undesirable relative positioning errors. This is currently under investigation by the authors, along with the problem of adaptive and sliding-mode robust trajectory control (Tzafestas et al., 1988, 1992, 1996; Tzafestas, 1989, 1991b; Stavrakakis et al., 1990; Tzafestas and Prokopiou, 1997), using the decomposition

Acknowledgment. The authors would like to thank the reviewers for their constructive comments and suggestions which were followed in revising the paper.

References

ALFORD, C.O. and BELYEU, S.M. (1984) Coordinated control of two robot arms. *Proc. IEEE Intern. Conference on Robotics and Automation*, Atlanta, GA, March 1984, 468-473.

CHOI, M.H. and LEE, B.H. (1995) A real-time optimal load distribution for multiple cooperating robots. *Proc. 1995 IEEE Conf. on Robotics and Automation*, 1211-1216.

FREUND E. (1984) On the design of multirobot systems. *Proc. IEEE Intern. Conference on Robotics and Automation*, Atlanta, GA, March 1984, 477-490.

FREUND, E. and HOYER, H. (1985) Collision avoidance in multi-robot systems. in: H. Hanafusa and H. Inoue, eds., *Robotics Research: The Second Intl. Symposium*, MIT Press, Cambridge, MA, 135-146.

FUJII, S. and KURONO, S. (1975) Coordinated computer control of a pair of manipulators. *Proc. 4th IFTOMM World Congress*, Univ. Newcastle, U.K., Sept. 1975, 411-417.

HENRICH, D. and CHENG, X. (1992) Fast distance computation for on-line collision detection with multi-arm robots. *Proc. 1992 IEEE Intl. Conf. on Robotics and Automation*, Nice, France, 2514-2519.

HOYER, H. (1985) On line collision avoidance for industrial robots. *Proc. IFAC Symp. on Robot Control (SYROCO'85)*, Barcelona, Spain, 477-485.

ISHIDA, T. (1977) Force control in coordination of two arms. *Proc. 5th Int. Conf. on Artificial Intelligence*, August 1977, 411-415.

KIM, K.I. and ZHENG, Y.F. (1989) Two strategies of position and force control for two industrial robots handling a single object. *Robotics and Autonomous Systems*, **5**, 395-403.

KOIVO, A.J. (1985) Adaptive position-velocity-force control of two manipulators. *Proc. 24th IEEE Conf. on Decision and Control*, Ft. Lauderdale, Fl., Dec. 1985, 334-337.

KOIVO, A.J. and BEKEY, G.A. (1988) Report of workshop on coodinated multiple robot manipulators: Planning, control and applications. *IEEE Journal Robotics and Automation*, **4**, 1, 91-93.

KWON, W., AND LEE, B.H. (1996) Optimal force distribution of multiple cooperating robots using nonlinear programming dual method. *Proc. 1996 IEEE Intl. Conf. on Robotics and Automation*, 2408-2413.

LIM, J. and GHYUNG, D.H. (1985) On a control scheme for two cooperating robot arms. *Proc. 24th IEEE Conf. on Decision and Control*, Ft. Lauderdale, Fl., Dec. 1985, 334-337.

MAYORGA, R.V. and WONG, A.K.C. (1997) A robust method for the concurrent motion planning of multi-manipulator systems. *J. Intell. and Robotic*

PALJUG, E. and YUN, X. (1995) Experimental study of two robot arms manipulating large objects. *IEEE Trans. Control Syst. Tech.*, **3**, 2, 177-188.

PALJUG, E., YUN, X. AND KUMAR, V. (1991) Control of rolling contacts in multiple robotic manipulation. *Proc. Intl. Conf. on Advanced Robotics*, Pisa, Italy, 591-596.

SCHNEIDER, S.A. and CANNON, JR., R.H. (1988) Object impedance control for cooperative manipulation: Theory and experimental results. *IEEE J. Robotics and Automation*, **4**, 1, 91-93.

STAVRAKAKIS, G., ZAGORIANOS, A. and TZAFESTAS, S.G. (1990) Combined computed torque and model reference adaptive control of robot systems. *J. Franklin Inst.*, **327**, 273-294.

SU, S. (1992) Coordinated control of multiple manipulator systems: experimental results. *Proc. 1992 IEEE Intl. Conf. on Robotics and Automation*, Nice, France, 2199-2204.

TOURNASSOUD, P. (1986) A strategy for obstacle avoidance and its application to multi-robot systems. *Proc. IEEE Intl. Conf. on Robotics and Automation*, San Francisco, CA, 1224-1229.

TZAFESTAS, S.G. (1989) Dynamic modelling and adaptive contol of industrial robots. *Systems Anal. Model. Simul.*, **6**, 4, 243-266.

TZAFESTAS, S.G. (1991A) *Intelligent Robotic Systems*. Marcel Dekker, N.Y.

TZAFESTAS, S.G. (1991B) Adaptive, robust and rule-based control of robotic manipulators. In: Tzafestas (1991a), 313-419.

TZAFESTAS, S.G. (1991C) Task grouping and scheduling for parallel processing. in: T. Ono and F. Kozin, eds., *Systems and Control — Topics in Theory and Applications*, MITA Press, Tokyo, 401-419.

TZAFESTAS, S.G., KALOBATSOS B., Stavrakakis G. and Zagorianos A. (1992) Some results concerning the autonomous trajectory generation and adaptive control of industrial robots. *J. Franklin Inst.*, **329**, 1, 1-14.

TZAFESTAS, S.G. and PROKOPIOU, P.A. (1997) Compensation of teleoperator uncertainties with a sliding-mode controller. *J. Robotics and Computer Integrated Manufacturing*, **13**, 1, 9-20.

TZAFESTAS, S.G., RAIBERT, M. and TZAFESTAS, C.S. (1996) Robust sliding-mode control applied to a 5-link biped robot. *J. Intelligent and Robotic Systems*, **15**, 1, 67-133.

TZAFESTAS, S.G., STAVRAKAKIS, G. and ZAGORIANOS, A. (1988) Robot model reference adaptive control through lower / upper part dynamic decomposition. *J. Intelligent and Robotic Systems*, **1**, 163-184.

TZAFESTAS, S.G. and TRIANTAFYLLAKIS, A. (1993) Deterministic scheduling in computing and manufacturing systems: A survey of models and algorithms. *Mathematics and Computers in Simulation*, **35**, 5, 397-434.

TZAFESTAS, S.G. and TRIANTAFYLLAKIS, A. (1994) A new adaptively weighted combinatorial dispatching rule for complex scheduling problems. *Com-*

TZAFESTAS, S.G., TRIANTAFYLLAKIS, A. and RIZOS, G. (1995) Scheduling dependent tasks on identical machines using a novel heuristic criterion: A Robotic Computation Example. *J. Intelligent and Robotic Systems*, **12**, 229-237.

UNSEREN, M.A. (1991) Rigid body dynamics and decoupled control architecture for two strongly interacting manipulators. *Robotica*, **9**, 4, 421-430.

UNSEREN, M.A. and KOIVO, A.J. (1989) Reduced-order model and decoupled control architecture for two manipulators holding an object. *Proc. 1989 IEEE Intl. Conf. on Robotics and Automation*, Scottsdale, Arizona, 1240-1245.

WANG, J., DODDS, S.J. and BAILEY, W.N. (1997) Co-ordinated control of multiple robotic manipulators handling a common object — Theory and experiments. *IEE Proc. — Control Theory Appl.*, **144**, 1, 73-84.

WU, C.-J. (1997) Collision-free trajectory planning for two cooperative redundant manipulators using the minimum-time criterion. *J. Intell. and Robotics Systems*, **18**, 1, 17-45.

YOSHIKAWA, T. and ZHENG, X. (1990) Coordinated dynamic hybrid position / force control for multiple robot manipulators handling constrained objects. *Proc. IEEE Intl. Conf. on Robotics and Automation*, Cincinnati, Ohio, 1178-1183.

ZAGORIANOS, A., KONTOYIANNIS, E. and TZAFESTAS, S.G. (1994) Resolved motion model-based predictive control of redundant robots. *Math. & Computers in Simulation (IMACS Trans.)*, **37**, 195-205.

ZAGORIANOS, A., TZAFESTAS, S.G. and STAVRAKAKIS, G.S. (1995) On-Line discrete-time control of industrial robots. *Robotics & Autonomous Systems*, **14**, 289-299.

ZAPATA, R., FOURNIER, A. and DAUCHEZ, P. (1987) True cooperation of robots in multi-arm tasks. *Proc. IEEE Intl. Conf. on Robotics and Automation*, Raleigh, NC, 1255-1260.

ZHENG, Y.F. and LUH, J.Y.S. (1985) Control of two coordinated robots in motion. *Proc. 24th IEEE Conf. on Decision and Control*, Ft. Lauderdale, Fl., Dec. 1985, 1761-1764.