Direct and Inverse Kinematics for Coordinated Motion Tasks of a Two-Manipulator System

P. Chiacchio

S. Chiaverini

B. Siciliano
Assoc. Mem. ASME

Dipartimento di Informatica e Sistemistica,
Università degli Studi di Napoli Federico II,
Via Claudio 21, 80125 Napoli, Italy

A new formulation for direct kinematics of a system of two manipulators is presented. This allows a straightforward description of general coordinated motion tasks in terms of meaningful absolute and relative variables. An effective inverse kinematics algorithm is devised which exploits the above formulation where the task Jacobians are expressed in terms of the Jacobians of the single manipulators. The scheme is extended to handle the presence of redundant degrees of freedom in the system. Different types of grasp between the end effectors and a commonly held object are treated with minimum reformulation effort. Case studies are developed throughout the paper for a system of two PUMA 560 manipulators which illustrate the capabilities of the scheme.

1 Introduction

Various robotic applications require the adoption of two-arm systems in lieu of single robot manipulators. These include, for instance, manipulation of heavy or non-rigid objects and mating of mechanical pieces. In all such cases the two robots should operate in a coordinated way so as to synchronize the relative motions, avoid undesired collisions, maintain the grasp between the arms and the object, etc.

The problem of motion coordination of a two-arm robot system was addressed in a number of previous works including the leader-follower strategy [1], [2], the symmetric formulation [3]–[5], the constrained motion and grasp analysis [6]–[9].

One important point in coordinated motion of two manipulators is the definition of meaningful task space variables that allow a straightforward specification of the assigned task. This becomes critical in the case of a task formulation involving both position and orientation for which the specification of variables describing orientation is not always obvious even for a skilled user.

The contribution of this work is to obtain a direct kinematics model of a system of two manipulators which allows a clear description of coordination tasks in terms of absolute and relative components. The case of planar arms was presented in [10, 11]. Here an effective definition of absolute and relative task variables for direct kinematics of a two-manipulator system is presented with particular concern to specification of orientation in space.

The kinematic model used for coordination is expressed in terms of the direct kinematics of the two manipulators; in particular the task Jacobians are computed on the basis of the Jacobians of the single manipulators. An inverse kinematics scheme is devised to find the joint motions for the single manipulators corresponding to a given coordinated motion task. The scheme is easily extended to handle the presence of redundant degrees of freedom in the system.

The proposed task formulation allows specification of coordinated motions of a two-manipulator system, and thus it does not necessarily assume that an object is actually held by the two end effectors.

Nevertheless, when an object is present, different types of grasp may occur and the kinematic model of contact has to be properly accounted for [12]. The case of tight grasp was treated in [13], and preliminary work on different types of non-tight grasp is reported in [14–16]. Here it is formally shown how the proposed task formulation can be retained to encompass the cases of rolling contact and sliding contact. This is achieved by adding to the direct kinematics equations for the two manipulators as many virtual joint variables as the number of degrees of freedom of the grasp.

It should be remarked that the resulting joint motions can be used as reference inputs to some control scheme for the two-manipulator system. However, the focus of the present work is purely on the kinematics aspects—not on the control aspects—of the coordinated motion problem for a two-manipulator system.

A system composed by two PUMA 560 robot manipulators is considered to work out numerical case studies throughout the paper. These are aimed at showing the effectiveness of the inverse kinematics scheme when the proposed task-oriented formulation is applied to nonredundant and redundant systems.

The paper is organized as follows. Section 2 presents the task formulation for coordinated motion of a system of two manipulators. The inverse kinematics scheme is developed in Section 3. The utilization of redundant degrees of freedom is discussed in Section 4. The cases of rolling contact and sliding contact are treated in Sections 5 and 6, respectively. Conclusions are drawn in a final section.

2 Task Formulation for Coordinated Motion

Consider a system of two manipulators. For each manipulator ($i = 1, 2$), let $\mathbf{p}_i^b$ be the (3×1) vector denoting the end-effector position. Let also $\mathbf{R}_i^b$ be the (3×3) rotation matrix expressing the end-effector orientation, i.e., its columns represent the unit vectors of a frame attached to the end effector. Both quantities are referred to a common base frame (superscript "b").

The end-effector linear velocity is directly given as the time derivative of the position vector, that is $\dot{\mathbf{p}}_i^b$. The end-effector angular velocity is given by the (3×1) vector $\boldsymbol{\omega}_i^b$, which is related to the time derivative of the rotation matrix $\mathbf{R}_i^b$ through the relationship [17]

$$\dot{\mathbf{R}}_i^b = \mathbf{S}(\boldsymbol{\omega}_i^b)\mathbf{R}_i^b, \tag{1}$$

where $\mathbf{S}(\cdot)$ is the (3×3) skew-symmetric operator performing the cross product.

Contributed by the Dynamic Systems and Control Division for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received by the DSCD February 13, 1995. Associate Technical Editor: J. E. Colgate.

It is desired to find a *task formulation* that allows specification of a *coordinated motion* for the two-manipulator system. To this purpose, it should be obvious that taking the end-effector position and orientation of each manipulator as task variables is inadequate, since the system would be regarded as composed by two independent manipulators and coordination management would be left to the user.

On the other hand, most available task formulations are based on a global description of the system through the use of the so-called grasp matrix, assuming that a common object is held by the two manipulators [3, 4, 6, 8]. Linear velocities $\dot{\mathbf{p}}_i^b$ and angular velocities $\boldsymbol{\omega}_i^b$ of the two manipulators are mapped by the transpose of the grasp matrix, giving the *absolute* linear and angular velocities of the object; then, by exploiting the null space of the grasp matrix, linear and angular velocities of one manipulator *relative* to the other are computed. The result is a complete description of the task at differential kinematics level. Position task variables can be easily found by integration of linear velocities while a problem arises for orientation task variables, in view of the non-integrability of angular velocities. It might be argued that a description of orientation could be found by means of a minimal representation, e.g., Euler angles, but the resulting variables would not allow a clear specification of a coordination task for the user.

In the following, an effective formulation is proposed which fully characterizes a coordinated motion task and, at the same time, it allows the user to naturally specify the task in terms of a set of meaningful variables.

In order to establish this task formulation, a suitable frame is to be introduced to specify coordinated motions of the two-manipulator system. Let such frame be termed as absolute frame.

The *absolute position* of the system can be defined as the origin of the absolute frame (denoted by subscript "a") which can be expressed as a function of the positions of the two end effectors. One simple choice is

$$\mathbf{p}_a^b = \tfrac{1}{2}(\mathbf{p}_1^b + \mathbf{p}_2^b). \tag{2}$$

In order to define the absolute orientation of the system, consider the matrix operator $\mathbf{R}_k(\vartheta)$ expressing the rotation by the angle ϑ about the axis aligned with the unit vector $\mathbf{k} = [k_x\ k_y\ k_z]^T$ [17]; note that the following property holds:

$$\dot{\mathbf{R}}_k(\lambda\vartheta) = \lambda\dot{\mathbf{R}}_k(\vartheta), \tag{3}$$

where λ is a constant. Then, the rotation matrix giving the *absolute orientation* can be defined as

$$\mathbf{R}_a^b = \mathbf{R}_1^b\mathbf{R}_{k_{12}^1}^1(\vartheta_{12}/2), \tag{4}$$

where $\mathbf{k}_{12}^1$ and ϑ_{12} are, respectively, the unit vector and the angle that realize the rotation described by $\mathbf{R}_2^1$, i.e., the orientation of frame 2 with respect to frame 1. Therefore, the above choice corresponds to make a rotation about axis $\mathbf{k}_{12}^1$ by an angle which is half the angle needed to align $\mathbf{R}_2^b$ with $\mathbf{R}_1^b$.

The absolute position and orientation describe the task in terms of the composition of the position and orientation of the single manipulators. It is clear that there exist infinite end-effector configurations giving the same absolute position and orientation. Therefore, in order to fully describe a coordinated motion, the position and orientation of one manipulator relative to the other is also of concern.

The *relative position* between the two end effectors can be defined as

$$\mathbf{p}_r^b = \mathbf{p}_2^b - \mathbf{p}_1^b. \tag{5}$$

The *relative orientation* between the two end effectors can be defined with reference to the end-effector frame of either manipulator—say the first one—in terms of the rotation matrix

$$\mathbf{R}_r^1 = \mathbf{R}_2^1. \tag{6}$$

Remark 1. To simplify specification of a coordination task, it may be appropriate to choose $\mathbf{p}_i^b$ and $\mathbf{R}_i^b$ other than the actual end-effector position and orientation of the two manipulators. In that case proper constant transformation matrices have to be introduced in the direct kinematics of the two manipulators. ■

Remark 2. To be independent of the absolute motion of the system, it is more convenient to specify the relative position with reference to the absolute frame, i.e., $\mathbf{p}_r^a$. The relationship between $\mathbf{p}_r^a$ and $\mathbf{p}_r^b$ is given by

$$\mathbf{p}_r^b = \mathbf{R}_a^b\mathbf{p}_r^a \tag{7}$$

with $\mathbf{R}_a^b$ as in (4). ■

Remark 3. A distinguished feature of the proposed formulation is that coordinated motion of the system is achieved without necessarily assuming that the two manipulators are kinematically constrained through the presence of an object between the two end effectors. Nevertheless, if the two end effectors hold a common object, general manipulation tasks can be described by the above formulation. For instance, if the task is to move a tightly grasped object without deforming it, a trajectory has to be assigned to $\mathbf{p}_a^b$ and $\mathbf{R}_a^b$ while $\mathbf{p}_r^a$ and $\mathbf{R}_r^1$ have to be kept constant. Yet, if the task is to stretch, bend or shear the object, suitable trajectories have to be specified for the relative variables too. Cases of non-tight grasp will be discussed later. ■

Having established a task formulation for the direct kinematics of the two-manipulator system, it is useful to derive also the differential kinematics relating the coordinated (absolute and relative) velocities to the corresponding velocities of the two manipulators. This is of interest not only for characterizing the velocity mappings, similar to task formulations using the grasp matrix, but also for solving the inverse kinematics of the two-manipulator system as well as for handling the presence of redundant degrees of freedom in the system. These issues will be treated in the next two sections.

The absolute linear velocity of the system is obtained as the time derivative of (2), i.e.,

$$\dot{\mathbf{p}}_a^b = \tfrac{1}{2}(\dot{\mathbf{p}}_1^b + \dot{\mathbf{p}}_2^b). \tag{8}$$

Differentiating (4) with respect to time and using (1), (3) yields

$$\mathbf{S}(\boldsymbol{\omega}_a^b)\mathbf{R}_a^b = \mathbf{S}(\boldsymbol{\omega}_1^b)\mathbf{R}_a^b + \tfrac{1}{2}\mathbf{S}(\boldsymbol{\omega}_{12}^b)\mathbf{R}_a^b \tag{9}$$

where $\boldsymbol{\omega}_{12}^b$ denotes the angular velocity of frame 2 with respect to frame 1. Since $\boldsymbol{\omega}_{12}^b = \boldsymbol{\omega}_2^b - \boldsymbol{\omega}_1^b$, the absolute angular velocity is given by

$$\boldsymbol{\omega}_a^b = \tfrac{1}{2}(\boldsymbol{\omega}_1^b + \boldsymbol{\omega}_2^b). \tag{10}$$

The relative linear velocity of the system is obtained as the time derivative of (5), i.e.,

$$\dot{\mathbf{p}}_r^b = \dot{\mathbf{p}}_2^b - \dot{\mathbf{p}}_1^b. \tag{11}$$

If the relative position is expressed as $\mathbf{p}_r^a$, from (7) the relative linear velocity in the base frame is related to $\dot{\mathbf{p}}_r^a$ by

$$\dot{\mathbf{p}}_r^b = \mathbf{R}_a^b\dot{\mathbf{p}}_r^a + \mathbf{S}(\boldsymbol{\omega}_a^b)\mathbf{p}_r^b \tag{12}$$

with $\boldsymbol{\omega}_a^b$ as in (9).

Finally, differentiating (6) with respect to time and using (1) yields

$$\mathbf{S}(\boldsymbol{\omega}_r^1)\mathbf{R}_r^1 = \mathbf{S}(\boldsymbol{\omega}_{12}^1)\mathbf{R}_2^1, \tag{13}$$

and thus the relative angular velocity is

$$\boldsymbol{\omega}_r^b = \boldsymbol{\omega}_2^b - \boldsymbol{\omega}_1^b \tag{14}$$

which has been expressed in the base frame.

3 Inverse Kinematics Algorithm

The inverse kinematics problem for a two-manipulator system can be stated as that to compute the joint variable trajectories corresponding to given coordinated motion trajectories for the absolute and relative task variables.

Finding closed-form solutions is possible only for special manipulator geometries and simple coordination tasks, and thus an algorithmic approach should be pursued. Algorithmic solutions to the inverse kinematics problem are based on the computation of the Jacobians associated with the task variables of interest. Since these variables have been expressed in the previous section as a function of the position and orientation of the two end effectors, the sought Jacobians are remarkably related to the Jacobians of the single manipulators.

Without loss of generality, consider a system of two cooperative 6-degree-of-freedom manipulators. For each manipulator, let $\mathbf{q}_i$ indicate the (6×1) vectors of joint variables. The geometric Jacobian $\mathbf{J}_i^b(\mathbf{q}_i)$ is the (6×6) matrix relating the joint velocity vectors $\dot{\mathbf{q}}_i$ to the linear and angular end-effector velocities in the base frame as

$$\begin{bmatrix} \dot{\mathbf{p}}_i^b \\ \boldsymbol{\omega}_i^b \end{bmatrix} = \mathbf{J}_i^b(\mathbf{q}_i)\dot{\mathbf{q}}_i \quad i = 1, 2. \tag{15}$$

At this point, combining (8), (10) and taking into account (15) yields

$$\begin{bmatrix} \dot{\mathbf{p}}_a^b \\ \boldsymbol{\omega}_a^b \end{bmatrix} = \mathbf{J}_a^b(\mathbf{q}_1, \mathbf{q}_2)\begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{bmatrix}, \tag{16}$$

where the (6×12) *absolute Jacobian* matrix is defined as

$$\mathbf{J}_a^b = [\tfrac{1}{2}\mathbf{J}_1^b \quad \tfrac{1}{2}\mathbf{J}_2^b]. \tag{17}$$

Further, combining (11), (13) and taking into account (15) yields

$$\begin{bmatrix} \dot{\mathbf{p}}_r^b \\ \boldsymbol{\omega}_r^b \end{bmatrix} = \mathbf{J}_r^b(\mathbf{q}_1, \mathbf{q}_2)\begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{bmatrix}. \tag{18}$$

where the (6×12) *relative Jacobian* matrix is defined as

$$\mathbf{J}_r^b = [-\mathbf{J}_1^b \quad \mathbf{J}_2^b]. \tag{19}$$

An effective *inverse kinematics algorithm* is given by the closed-loop scheme based on the computation of the inverse of the manipulator Jacobian [18]. The joint velocity solution can be written in the form

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})(\mathbf{v}_d + \mathbf{K}\mathbf{e}) \tag{20}$$

where $\mathbf{q}$ is the vector of joint variables, $\mathbf{J}$ is the Jacobian—assumed to be square and singularity-free—associated with the velocity mapping, $\mathbf{v}_d$ is the desired task velocity, $\mathbf{K}$ is a suitable diagonal positive gain matrix, and $\mathbf{e}$ is the algorithmic error between the desired and current task variables. Notice that the physical robot system is not involved since the algorithm only serves the purpose to invert the kinematics of the system along a given task trajectory, i.e., $\dot{\mathbf{q}}$ is given by (20), and $\mathbf{q}$ is computed by integrating $\dot{\mathbf{q}}$.

The closed-loop inverse kinematics algorithm based on (20) avoids the typical numerical drift of open-loop resolved-rate schemes. The solution can be made robust with respect to singularities of $\mathbf{J}$ by resorting to a damped least-squares inverse of the matrix [19], [20].

The above algorithm can be keenly applied to solve the inverse kinematics for the two-manipulator system at issue. In detail, define

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix}. \tag{21}$$

The Jacobian is

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_a^b \\ \mathbf{J}_r^b \end{bmatrix}, \tag{22}$$

where $\mathbf{J}_a^b$, $\mathbf{J}_r^b$ are given as in (17), (19). The error is

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_a \\ \mathbf{e}_r \end{bmatrix}. \tag{23}$$

The *absolute error* has a position and an orientation component [21] and is given by

$$\mathbf{e}_a = \begin{bmatrix} \mathbf{p}_{ad}^b - \mathbf{p}_a^b \\ \tfrac{1}{2}(\mathbf{S}(\mathbf{n}_a^b)\mathbf{n}_{ad}^b + \mathbf{S}(\mathbf{s}_a^b)\mathbf{s}_{ad}^b + \mathbf{S}(\mathbf{a}_a^b)\mathbf{a}_{ad}^b) \end{bmatrix}, \tag{24}$$

where $\mathbf{p}_{ad}^d$ is the desired absolute position specified by the user in the base frame, $\mathbf{p}_a^b$ is the actual absolute position that can be computed as in (2), $\mathbf{n}_{ad}^b$, $\mathbf{s}_{ad}^b$, $\mathbf{a}_{ad}^b$ are the column vectors of the rotation matrix $\mathbf{R}_{ad}^b$ giving the desired absolute orientation specified by the user in the base frame, and $\mathbf{n}_a^b$, $\mathbf{s}_a^b$, $\mathbf{a}_a^b$ are the column vectors of the rotation matrix $\mathbf{R}_a^b$ in (4). The relative error is given by

$$\mathbf{e}_r = \begin{bmatrix} \mathbf{R}_a^b\mathbf{p}_{rd}^a - \mathbf{p}_r^b \\ \tfrac{1}{2}\mathbf{R}_1^b(\mathbf{S}(\mathbf{n}_r^1)\mathbf{n}_{rd}^1 + \mathbf{S}(\mathbf{s}_r^1)\mathbf{s}_{rd}^1 + \mathbf{S}(\mathbf{a}_r^1)\mathbf{a}_{rd}^1) \end{bmatrix}. \tag{25}$$

The rotation $\mathbf{R}_a^b$ is aimed at expressing the desired relative position $\mathbf{p}_{rd}^a$, assigned by the user in the absolute frame (see Remark 2), in the base frame; in this way, the specification of the desired relative position between the two end effectors is not affected by the absolute frame orientation. Further in (25) notice that: $\mathbf{p}_r^b$ can be computed as in (5); $\mathbf{n}_{rd}^1$, $\mathbf{s}_{rd}^1$, $\mathbf{a}_{rd}^1$ are the column vectors of the rotation matrix $\mathbf{R}_{rd}^1$ giving the desired relative orientation specified by the user in the end-effector frame of the first manipulator; $\mathbf{n}_r^1$, $\mathbf{s}_r^1$, $\mathbf{a}_r^1$ are the column vectors of the rotation matrix $\mathbf{R}_r^1$ in (6); and the rotation $\mathbf{R}_1^b$ is aimed at expressing the orientation error in the base frame.

Finally, the desired velocity is

$$\mathbf{v}_d = \begin{bmatrix} \mathbf{v}_{ad} \\ \mathbf{v}_{rd} \end{bmatrix}. \tag{26}$$

The *absolute velocity* term is given by

$$\mathbf{v}_{ad} = \begin{bmatrix} \dot{\mathbf{p}}_{ad}^b \\ \boldsymbol{\omega}_{ad}^b \end{bmatrix}, \tag{27}$$

where $\dot{\mathbf{p}}_{ad}^b$ and $\boldsymbol{\omega}_{ad}^b$ are respectively the desired absolute linear and angular velocities specified by the user in the base frame. The *relative velocity* term is given by

$$\mathbf{v}_{rd} = \begin{bmatrix} \mathbf{R}_a^b\dot{\mathbf{p}}_{rd}^a + \mathbf{S}(\boldsymbol{\omega}_a^b)\mathbf{R}_a^b\mathbf{p}_{rd}^a \\ \boldsymbol{\omega}_{rd}^1 \end{bmatrix}, \tag{28}$$

where $\dot{\mathbf{p}}_{rd}^a$ is the desired relative linear velocity specified by the user in the object frame and $\boldsymbol{\omega}_{rd}^1$ is the desired relative angular velocity specified by the user in the end-effector frame of the first manipulator. Notice that the expression of the translational part of the relative velocity presents an additional term which is a consequence of having assigned the relative position with reference to the absolute frame.

Case Study 1. A system composed by two PUMA 560 manipulators is considered to develop two case studies for a nonredundant and a redundant system, respectively; SI units are used throughout the various case studies. The base of manipulator 1 is located at $(0, -0.1501, 0)$ and the initial configuration places the end effector at $\mathbf{p}_1^b = [0.4 \quad 0 \quad 0.5]^T$; in view of Remark 1, a constant rotation matrix has been used so that $\mathbf{R}_1^b = \mathbf{I}_3$ where $\mathbf{I}_3$ denotes the (3×3) identity matrix. The base of manipulator 2 is located at $(1, 0.1501, 0)$ and the initial configuration places the end effector at $\mathbf{p}_2^b = [0.6 \quad 0 \quad 0.5]^T$ with $\mathbf{R}_2^b = \mathbf{I}_3$. By using (2), (4), (5), (6), the initial values for the task variables are:

$$\mathbf{p}_a^b = [0.5 \quad 0 \quad 0.5]^T \quad \mathbf{R}_a^b = \mathbf{I}_3$$

$$\mathbf{p}_r^b = [0.2 \quad 0 \quad 0]^T \quad \mathbf{R}_r^1 = \mathbf{I}_3.$$

A sketch of the initial configuration is depicted in Fig. 1 where, for the purpose of illustration, an object is represented. Notice that the actual end-effector orientation of the two manipulators, as represented in Fig. 1, differs from the adopted rotation matrices $\mathbf{R}_1^b$ and $\mathbf{R}_2^b$.

The task is to move the origin of the absolute frame to the position

$$\mathbf{p}_a^b = [0.5 \quad 0 \quad 0.7]^T$$

realizing a displacement of 0.2 along the z axis, and to rotate the absolute frame at

$$\mathbf{R}_a^b = \begin{bmatrix} \cos \pi/4 & -\sin \pi/4 & 0 \\ \sin \pi/4 & \cos \pi/4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

corresponding to a rotation of $\pi/4$ about the z axis.

A rectilinear path has been specified from the initial to the final position, whereas an angular motion about the z axis has been specified to rotate from the initial to the final orientation. Smooth trajectories have been assigned using 5th-order interpolating polynomials for the path coordinate as well as for the angle of rotation about the fixed axis [17]; null initial and final velocities and accelerations have been imposed with a time duration of 1 s.

As for the relative variables, it is convenient to define the task with reference to the absolute frame. The task is to keep $\mathbf{p}_r^a$ and $\mathbf{R}_r^1$ constant; this corresponds to maintain a tight grasp with the object. Notice the simplicity of task specification when $\mathbf{p}_r$ is referred to the absolute frame, otherwise a suitable trajectory should be specified for $\mathbf{p}_r^b$ depending on the absolute orientation $\mathbf{R}_a^b$.

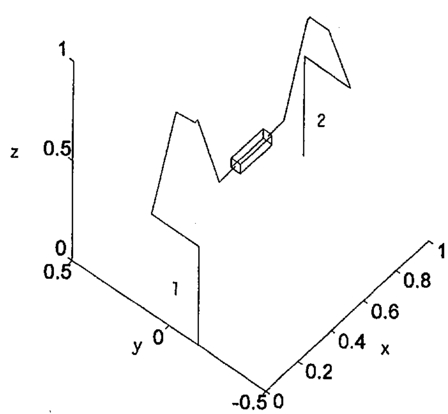

Fig. 1 Initial configuration of the system for Case Study 1

The inverse kinematics algorithm based on (20) has been implemented in MATLAB at 1 ms sampling time; the gain matrix has been chosen as $\mathbf{K} = \text{block diag}\,\{500\mathbf{I}_6, 1000\mathbf{I}_6\}$.

The resulting final system configuration is shown in Fig. 2, whereas the time history of the norm of position and orientation components of both absolute and relative errors (Fig. 3) confirm the good performance of the inverse kinematics algorithm. This means that the task trajectories corresponding to the joint trajectories generated by the algorithm accurately reproduce the specified task trajectories. ■

4 Redundant System

If the system possesses redundant degrees of freedom, the Jacobian becomes a matrix with more columns than rows. In this case, the solution (20) is modified into [22]

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{v}_d + \mathbf{K}\mathbf{e}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\dot{\mathbf{q}}_0 \tag{29}$$

with obvious adjustment of the dimensions of vector and matrix quantities. In (29) $\mathbf{J}^\dagger$ denotes the Moore-Penrose pseudoinverse of $\mathbf{J}$, and the operator $(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})$ projects the vector of arbitrary joint velocities $\dot{\mathbf{q}}_0$ (aimed at utilizing the redundant degrees of freedom) into the null space of $\mathbf{J}$ so as not to interfere with the primary end-effector task [23]. In the case of manipulators with mixed revolute and prismatic joints, the use of a weighted generalized inverse is advised [24].

A possible choice of $\dot{\mathbf{q}}_0$ in (29) is

$$\dot{\mathbf{q}}_0 = k_c \left(\frac{\partial c(\mathbf{q})}{\partial \mathbf{q}} \right)^T \tag{30}$$

where $c(\mathbf{q})$ is a constraint function of the joint variables to be optimized locally and k_c is a signed constant.

It should be pointed out that kinematic redundancy of the system of two manipulators may be due either to the effective presence of additional joint variables, i.e., more than 6 degrees of freedom for either manipulator, or to relaxation of some coordination task variables.

Case Study 2. The system of the two PUMA 560 manipulators is made kinematically redundant by placing the base of manipulator 2 on a track aligned with y axis. Hence $\mathbf{q}_2$ becomes a (7×1) vector, where the seventh component denotes the track displacement. The initial configuration is the same as in Case Study 1 (Fig. 1) and the same task is assigned for the absolute and relative variables. Further it is required to minimize the constraint $c = 0.5(q_{2,1}(t) - q_{2,1}(0))^2$, i.e., to keep the base revolute joint of manipulator 2 as closely as possible to the initial value.

The inverse kinematics algorithm based on (29), (30) has been implemented in MATLAB at 1 ms sampling time with the same $\mathbf{K}$ as above and $k_c = 3000$.

Figure 4 displays the resulting final system configuration. The errors are of the same order of magnitude as in Fig. 3, and they have not been reported for brevity. It is easy to recognize that the task has been successfully executed. A comparison with the nonredundant case reveals that the absolute frame is taken to the final position and orientation and the relative position and orientation are kept constant; further, the base revolute joint of manipulator 2 is not changed (compare with Fig. 1) thanks to the presence of the additional degree of freedom in the track. ■

5 Rolling Contact

The above direct and inverse kinematics allow performing coordinated motion of a system of two manipulators. As anticipated in Remark 3, the task formulation can be directly applied to specify tasks for a tightly grasped object by assigning suitable values to the absolute and relative quantities.

If the grasp is not tight, a new set of task variables should be derived but this may be difficult in general cases. For instance,

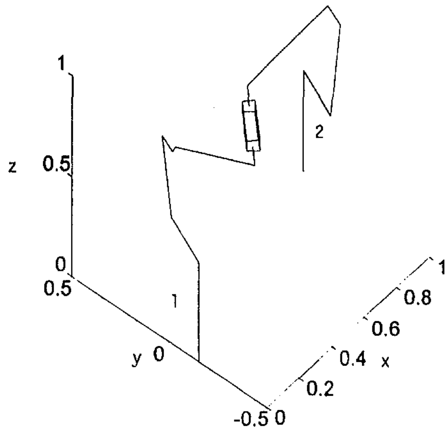

Fig. 2 Final configuration of the system for Case Study 1

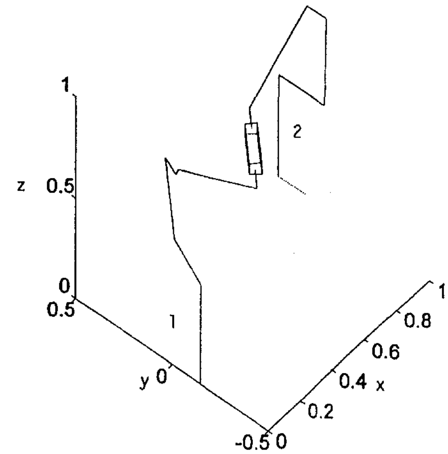

Fig. 4 Final configuration of the system for Case Study 2

suppose that either manipulator is allowed a one-degree-of-freedom rolling contact; in terms of the coordination task description, this implies that the dimension of the task space should be decreased. However, it is not easy to identify which task variable to relax except for special cases. For instance, for a system of planar arms the rolling degree-of-freedom can be chosen as the arms' relative orientation angle [10].

A keen solution to the above problem is to retain the task formulation expressed by (2), (4), (5), (6) as a whole, but to add virtual degrees of freedom to the kinematics of the manipulator making the rolling contact.

To embed the possibility of handling a general three-degree-of-freedom rolling point contact in either manipulator, three virtual revolute joints are introduced at the end effector with their axes realizing three independent revolute degrees of freedom that describe the kinematics of the contact. It is clear, for instance, that the axis of the first virtual joint must not be aligned with the axis of the last link of the manipulator. The result is an augmented kinematics expressing the position and orientation of the virtual end effector. In particular, the position of the virtual end effector coincides with that of the actual end effector, while their orientations differ to properly account for the rolling.

It is worth emphasizing that with the adoption of the virtual end-effector concept, the user continues to specify a coordination task in the same formal way as above, no matter whether an object with tight or non-tight grasp is held by the two end effectors.

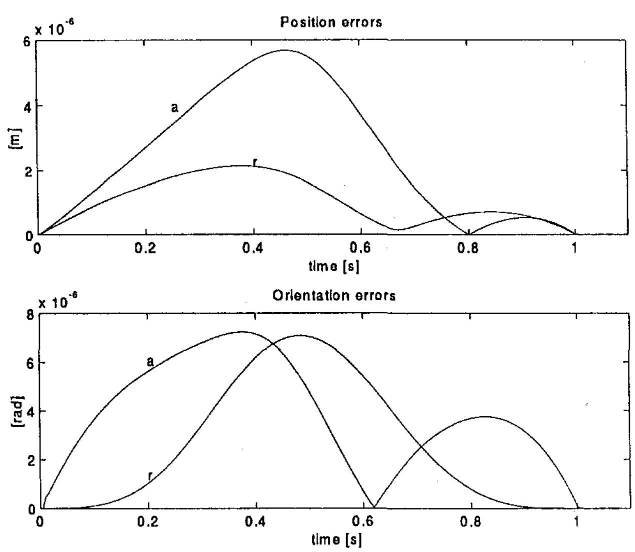

Fig. 3 Norm of position and orientation components of absolute and relative errors for Case Study 1

Let $\mathbf{q}_R$ denote the (3×1) vector of the additional joint variables. Without loss of generality, assume that manipulator 2 is making the rolling contact; this implies that $\mathbf{R}_2^b$ becomes a function of both $\mathbf{q}_2$ and $\mathbf{q}_R$. As a consequence, the virtual end effector still makes a tight grasp with the object and the foregoing task formulation can be retained, leaving to the inverse kinematics algorithm the job of handling the degrees of freedom describing the rolling contact.

The task space differential kinematics corresponding to (8), (10), (11), (13) becomes:

$$\begin{bmatrix} \dot{\mathbf{p}}_a^b \\ \boldsymbol{\omega}_a^b \end{bmatrix} = \mathbf{J}_a^b \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_R \end{bmatrix} \tag{31}$$

for the absolute part, where

$$\mathbf{J}_a^b(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_R) = [\tfrac{1}{2}\mathbf{J}_1^b(\mathbf{q}_1) \quad \tfrac{1}{2}\mathbf{J}_2^b(\mathbf{q}_2, \mathbf{q}_R)], \tag{32}$$

and

$$\begin{bmatrix} \dot{\mathbf{p}}_r^b \\ \boldsymbol{\omega}_r^b \end{bmatrix} = \mathbf{J}_r^b \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_R \end{bmatrix} \tag{33}$$

for the relative part, where

$$\mathbf{J}_r^b(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_R) = [-\mathbf{J}_1^b(\mathbf{q}_1) \quad \mathbf{J}_2^b(\mathbf{q}_2, \mathbf{q}_R)]. \tag{34}$$

At this point, by defining $\mathbf{q} = [\mathbf{q}_1^T \;\; \mathbf{q}_2^T \;\; \mathbf{q}_R^T]^T$ and $\mathbf{J} = [\mathbf{J}_a^{bT} \;\; \mathbf{J}_r^{bT}]^T$, the algorithm based on (29) and (30) can be adopted to compute the inverse kinematics solution.

Remark 4. If the grasp makes a one- or a two-degree-of-freedom rolling contact, then it is sufficient to add only one or two virtual joints at the end effector and to define a properly augmented kinematics. The solution will be formally analogous to the previous case. ■

Remark 5. The solution obtained with the above inverse kinematics algorithm provides the trajectories both for the actual joints of the two manipulators and for the virtual joints. Only the former will be used as reference inputs to some control schemes, whereas the motion described by the latter will be realized through the closed kinematic chain formed by the two end effectors and the object. ■

Remark 6. Notice that a rolling contact introduces kinematic redundancy into the system, thanks to the extra joint variables $\mathbf{q}_R$ available. These can be exploited to satisfy additional task constraints. ■

Numerical case studies for a system of two manipulators with a rolling contact, with and without using redundancy, can be found in [16].

6 Sliding Contact

A conceptually similar reformulation of the task can be worked out to describe an object grasp with a sliding contact. In terms of the previously introduced concept of virtual end effector, a sliding contact can be realized so that the rotation of the actual and virtual end effectors coincide while their positions differ according to the geometry of the sliding surface.

It can be recognized that a sliding contact requires at most two degrees of freedom. In the case of a planar surface, two virtual prismatic joints are added at the end effector with their axes realizing two degrees of freedom along the surface; of course, the two axes must not be aligned. The result is an augmented kinematics expressing the location of the virtual end effector.

Let $\mathbf{q}_S$ denote the (2×1) vector of the additional joint variables. Again, manipulator 2 is assumed to make the sliding contact; this implies that $\mathbf{p}_2$ becomes a function of both $\mathbf{q}_2$ and $\mathbf{q}_S$. Hence the virtual end effector still makes a tight grasp with the object and the procedure follows along the same lines as above.

The task space differential kinematics corresponding to (8), (10), (11), (13) in this case become:

$$\begin{bmatrix} \dot{\mathbf{p}}_a^b \\ \boldsymbol{\omega}_a^b \end{bmatrix} = \mathbf{J}_a^b \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_S \end{bmatrix} \tag{35}$$

for the absolute part, where

$$\mathbf{J}_a^b(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_S) = [\tfrac{1}{2}\mathbf{J}_1^b(\mathbf{q}_1) \quad \tfrac{1}{2}\mathbf{J}_2^b(\mathbf{q}_2, \mathbf{q}_S)], \tag{36}$$

and

$$\begin{bmatrix} \dot{\mathbf{p}}_r^b \\ \boldsymbol{\omega}_r^b \end{bmatrix} = \mathbf{J}_r^b \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_S \end{bmatrix} \tag{37}$$

for the relative part, where

$$\mathbf{J}_r^b(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_S) = [-\mathbf{J}_1^b(\mathbf{q}_1) \quad \mathbf{J}_2^b(\mathbf{q}_2, \mathbf{q}_S)]. \tag{38}$$

Then, by defining $\mathbf{q} = [\mathbf{q}_1^T \quad \mathbf{q}_2^T \quad \mathbf{q}_S^T]^T$ and $\mathbf{J} = [\mathbf{J}_a^{bT} \quad \mathbf{J}_r^{bT}]^T$, the algorithm based on (29) and (30) can be adopted to compute the inverse kinematics solution.

Remark 7. In the case of a curved surface, sliding contact still requires at most two degrees of freedom. However, three virtual prismatic joints have to be introduced at the end effector with a geometric constraint so as to realize two independent degrees of freedom along the surface. ■

Remark 8. The considerations in Remarks 5 and 6 apply also to this case. ■

Remark 9. More general types of contact can be modeled under the proposed framework by a proper combination of rolling and sliding contact. ■

Numerical case studies for a system of two manipulators with a sliding contact, with and without using redundancy, can be found in [15].

7 Conclusions

The direct and inverse kinematics problem for a nonredundant or redundant system of two manipulators achieving coordinated motion has been discussed in this work. A general task formulation has been presented which allows a straightforward description by means of suitable absolute and relative variables. The formulation is effectively used to derive an inverse kinematics algorithm that makes use of the kinematics of the two manipulators.

If an object is held by the two manipulators, it has been shown how a minimum reformulation effort allows description of different types of grasp via a suitable introduction of virtual joints providing virtual degrees of freedom that describe the grasp geometry.

Numerical case studies have been worked out for a system of two PUMA 560 manipulators holding a common object, which have demonstrated the effectiveness of the inverse kinematics algorithm to utilize redundant degrees of freedom.

The joint motions resulting from the inverse kinematics scheme can be fed as the reference inputs to the joint control systems of the individual robot manipulators (two-stage kinematic control strategy). This can be performed in either an off-line or an on-line fashion. With reference to commercially available hardware, the computational burden of the proposed inverse kinematics algorithm for motion coordination of two six-degree-of-freedom manipulators permits real-time implementation with sampling times as low as few milliseconds. Experiments are currently undergoing for a set-up of two industrial robots with six and seven joints, respectively, by running the control algorithms on a PC 486/66 at 2 ms; the results will be described in a future paper.

On the other hand, if the end effectors of the two manipulators hold a common object, relying on the joint control systems of the individual manipulators is practically inadequate, and a different control strategy has to be pursued. In [25] a new scheme was proposed that still uses the above joint motions as reference inputs but achieves control on object space variables so as to gain robustness to uncertainty or model inaccuracy in the system. However, only the planar case was considered and the extension to motion in space on the basis of the task formulation proposed in this paper is currently being investigated [26].

Acknowledgments

This work was supported partly by Ministero dell'Università e della Ricerca Scientifica e Tecnologica under 60% funds.

References

1 Luh, J. Y. S., and Zheng, Y. F., "Constrained Relations Between Two Coordinated Industrial Robots for Motion Control," *Int. J. of Robotics Research*, Vol. 6, No. 3, pp. 60–70, 1987.

2 Zheng, Y. F., "Kinematics and Dynamics of Two Industrial Robots in Assembly," *Proc. 1989 IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, pp. 1360–1365, 1989.

3 Uchiyama, M., and Dauchez, P., "A Symmetric Hybrid Position/Force Control Scheme for the Coordination of Two Robots," *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA, pp. 350–356, 1988.

4 Dauchez, P., Descriptions de Tâches en Vue de la Commande Hybride Symétrique d'un Robot Manipulateur a Deux Bras, Thèse d'Etat, Université des Sciences et Techniques du Languedoc, Montpellier, F, 1990.

5 Uchiyama, M., "A Unified Approach to Load Sharing, Motion Decomposing, and Force Sensing of Dual Arm Robots," *Robotics Research—5th Int. Symp.*, MIT Press, Cambridge, MA, pp. 225–232, 1990.

6 Nakamura, Y., Nagai, K., and Yoshikawa, T., "Mechanics of Coordinative Manipulation by Multiple Robotic Mechanisms," *Proc. 1987 IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, pp. 991–998, 1987.

7 Khatib, O., "Augmented Object Concept in Multi-Robot Manipulator Control," *Proc. 1988 American Control Conf.*, Atlanta, GA, pp. 2140–2147, 1988.

8 Li, Z., Hsu, P., and Sastry, S., "Grasping and Coordinated Manipulation by a Multifingered Robot Hand," *Int. J. of Robotics Research*, Vol. 8, No. 4, pp. 33–50, 1989.

9 Walker, I. D., Freeman, R. A., and Marcus, S. I., "Analysis of Motion and Internal Loading of Objects Grasped by Multiple Cooperating Manipulators," *Int. J. of Robotics Research*, Vol. 10, pp. 396–409, 1991.

10 Chiacchio, P., Chiaverini, S., and Siciliano, B., "Dexterous Reconfiguration of a Two-Arm Robot System," *Proc. IEEE Int. Conf. on Control '91*, Edinburgh, GB, pp. 347–351, Mar. 1991.

11 Chiacchio, P., Chiaverini, S., and Siciliano, B., "Kineto-Static Analysis of Cooperative Robot Manipulators Achieving Dexterous Configurations," *9th CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators*, Udine, I, Sept. 1992, in RoManSy 9, Lecture Notes in Control and Information Sciences 187, A. Morecki, G. Bianchi, K. Jaworek, eds., Springer-Verlag, Berlin, D, pp. 93–100, 1993.

12 Bruyninckx, H., Dutré, S., and De Schutter, J., "Application of Computer Aided Kinematics to Modelling of Contacts in Robotic Manipulation," NATO ASI on Computer Aided Kinematics and Dynamics of Rigid and Flexible Mechanical Systems, Troia, P, July 1993.

13 Chiacchio, P., Chiaverini, S., and Siciliano, B., "Task-Oriented Kinematic Control of Two Cooperative 6-dof Manipulators," *Proc. 1993 American Control Conf.*, San Francisco, CA, pp. 336–340, 1993.

14 Chiacchio, P., Chiaverini, S., and Siciliano, B., "User-Oriented Task Description for Cooperative Spatial Manipulators: One-Degree-of-Freedom Rolling Grasp," *Proc. 32nd IEEE Conf. on Decision and Control*, San Antonio, TX, pp. 1126–1127, 1993.

15 Chiacchio, P., Chiaverini, S., and Siciliano, B., "Redundancy Resolution for Two Cooperative Spatial Manipulators with a Sliding Contact," *10th CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators*, Gdańsk, PL, Sept. 1994, in Theory and Practice of Robots and Manipulators, CISM Courses and Lectures 361, A. Morecki, G. Bianchi, K. Jaworek, eds., Springer-Verlag, Wien, A, pp. 119–124, 1995.

16 Chiacchio, P., and Chiaverini, S., "User-Oriented Task Description for Cooperative Spatial Manipulators: Rolling Grasp," *5th Int. Symp. on Robotics and Manufacturing*, Wailea, HI, Aug. 1994, in *Robotics and Manufacturing: Recent Trends in Research, Education, and Applications*, Vol. 5, M. Jamshidi, C. Nguyen, R. Lumia and J. Yuh, eds., ASME Press, pp. 431–436, 1994.

17 Sciavicco, L., and Siciliano, B., *Modeling and Control of Robot Manipulators*, McGraw-Hill, New York, 1996.

18 Siciliano, B., "A Closed-Loop Inverse Kinematic Scheme for On-Line Joint-Based Robot Control," *Robotica*, Vol. 8, pp. 231–243, 1990.

19 Nakamura, Y., and Hanafusa, H., "Inverse Kinematic Solution with Singularity Robustness for Robot Manipulator Control," ASME JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL, Vol. 108, pp. 163–171, 1986.

20 Wampler, C. W., "Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 16, pp. 93–101, 1986.

21 Luh, J. Y. S., Walker, M. W., and Paul, R. P. C., "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE Trans. on Automatic Control*, Vol. 25, pp. 468–474, 1980.

22 Chiacchio, P., Chiaverini, S., Sciavicco, L., and Siciliano, B., "Closed-Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy," *Int. J. of Robotics Research*, Vol. 10, pp. 410–425, 1991.

23 Nakamura, Y., Hanafusa, H., and Yoshikawa, T., "Task-Priority Based Redundancy Control of Robot Manipulators," *Int. J. of Robotics Research*, Vol. 6, No. 2, pp. 3–15, 1987.

24 Doty, K., "Inverting Kinematic Equations: The Weighted Generalized Inverse Versus the Moore-Penrose Inverse," *1993 IEEE International Conference on Robotics and Automation*, in Tutorial on 'Force and Contact Control in Robotic Systems: A Historical Perspective and Current Technologies', Atlanta, GA, 1993.

25 Chiacchio, P., Chiaverini, S., and Siciliano, B., "Cooperative Control Schemes for Multiple Robot Manipulator Systems," *Proc. 1992 IEEE Int. Conf. on Robotics and Automation*, Nice, F, pp. 2218–2223, 1992.

26 Chiacchio, P., and Chiaverini, S., "PD-Type Control Schemes for Cooperative Manipulator Systems," *Int. J. of Intelligent Automation and Soft Computing*, Vol. 2, No. 2, 1996.