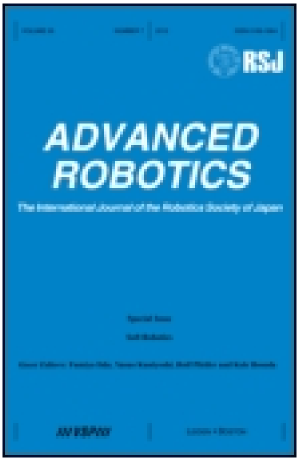


Advanced Robotics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tadr20

An algorithmic approach to coordinate transformation for robotic manipulators

A. Balestrino [a], G. De Maria [b], L. Sciavicco [c] & B. Siciliano [d]

[a] Dipartimento di Informatica e Sistemistica, Università di Napoli, Via Claudio 21, 80125 Napoli, Italy

[b] Dipartimento di Informatica e Sistemistica, Università di Napoli, Via Claudio 21, 80125 Napoli, Italy

[c] Dipartimento di Informatica e Sistemistica, Università di Napoli, Via Claudio 21, 80125 Napoli, Italy

[d] Dipartimento di Informatica e Sistemistica, Università di Napoli, Via Claudio 21, 80125 Napoli, Italy

Version of record first published: 02 Apr 2012.

To cite this article: A. Balestrino , G. De Maria , L. Sciavicco & B. Siciliano (1987): An algorithmic approach to coordinate transformation for robotic manipulators, Advanced Robotics, 2:4, 327-344

To link to this article: http://dx.doi.org/10.1163/156855388X00029

Advanced Robotics, Vol. 2, No. 4, pp. 327–344 (1988)

Paper

An algorithmic approach to coordinate transformation for robotic manipulators

A. BALESTRINO,* G. DE MARIA, L. SCIAVICCO† and B. SICILIANO
Dipartimento di Informatica e Sistemistica, Università di Napoli, Via Claudio 21, 80125 Napoli, Italy

Received for *AR* 12 May 1987

Abstract—Coordinate transformation is one of the most important issues in robotic manipulator control. Robot tasks are naturally specified in work space coordinates, usually a Cartesian frame, while control actions are developed on joint coordinates. Effective inverse kinematic solutions are analytical in nature; they exist only for special manipulator geometries and geometric intuition is usually required. Computational inverse kinematic algorithms have recently been proposed; they are based on general closed-loop schemes which perform the mapping of the desired Cartesian trajectory into the corresponding joint trajectory.

The aim of this paper is to propose an effective computational scheme to the inverse kinematic problem for manipulators with spherical wrists. First an insight into the formulation of kinematics is given in order to detail the general scheme for this specific class of manipulators. Algorithm convergence is then ensured by means of the Lyapunov direct method. The resulting algorithm is based on the hand position and orientation vectors usually adopted to describe motion in the task space. The analysis of the computational burden is performed by taking the Stanford arm as a reference. Finally a case study is developed via numerical simulations.

1. INTRODUCTION

Robotic manipulators consist of a series of $n+1$ rigid bodies linked together by n actuated joints. Depending on the structure of the kinematical chain, the basis for all advanced manipulator control techniques is the relationship between the Cartesian coordinates of the end effector and the joint coordinates of the manipulator. It is usual to distinguish between direct kinematics, or simply kinematics, and inverse kinematics of the manipulator. The former is concerned with expressing the external Cartesian coordinates of the end effector in terms of the internal joint coordinates; that is, passing from the so-called drive oriented space to the so-called task oriented space. The latter involves the inverse transformation of the external coordinates into the internal coordinates.

The method of characterizing direct kinematics is straightforward since the position and orientation of the end effector are completely specified once all the joint coordinates are given. The point at issue in this paper is the inverse kinematic problem; that is, determining the joint coordinates corresponding to a given Cartesian position and orientation of the manipulator's end effector. While there is only one end effector state for a given set of joint coordinates, there are a number of different configurations which all place the end effector in the same position and orientation

*Present address: Istituto di Elettrotecnica, Università di Pisa, Via Diotisalvi 2, 56100 Pisa, Italy.
†To whom correspondence should be addressed.

[1]. If one is concerned with computing the joint variable trajectory corresponding to a given Cartesian trajectory, only one of the possible solutions for each kinematic configuration is desired. In particular, the joint trajectory must be congruent with the initial arm configuration.

The paper is organized as follows. First the coordinate transformation problem is stated and a concise review of analytical [2] and computational methods [3, 4] is given. Then the kinematics of a class of manipulators whose wrist position is independent of the hand orientation is formulated; the orientation is specified in terms of unit vectors, and the structural kinematical properties of the hand orientation are fully investigated. The general idea first presented in ref. 3 and later in ref. 4 of adopting a dynamical scheme to solve the inverse kinematic problem is followed here. By taking into account the structural kinematic properties, an effective conversion algorithm is presented. No geometric intuition is required since the resultant algorithm is based on direct kinematics. Simulation results are shown for the six-degree-of-freedom Stanford manipulator (Fig. 1).

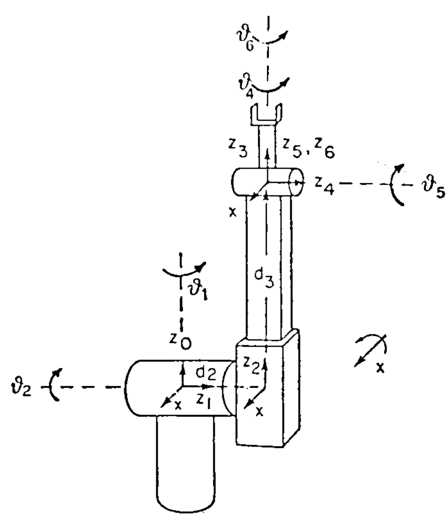


Figure 1. The Stanford arm [2].

1.1. Statement of the problem and previous work

For any given robot with known geometrical dimensions the kinematics can be written as [5]

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \tag{1}$$

where $\mathbf{q}$ is the vector of joint coordinates, $\mathbf{x}$ is the vector of Cartesian coordinates, and $\mathbf{f}$ is a non-linear vector valued function. Differentiating with respect to time gives

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{2}$$

which relates the joint velocity vector $\dot{\mathbf{q}}$ to the Cartesian velocity vector $\dot{\mathbf{x}}$ through the Jacobian matrix $\mathbf{J}(\mathbf{q}) = \partial\mathbf{f}/\partial\mathbf{q}$.

Obtaining a solution for the joint coordinates, that is, inverting (1), is of the utmost importance in order to provide the manipulator, controlled in the joint space, with a target trajectory $\hat{\mathbf{q}}(t)$, or else $(\hat{\mathbf{q}}(t), \dot{\hat{\mathbf{q}}}(t))$. (See refs. 3, 6–8 for some joint based control schemes). An alternative approach leads to Cartesian-based control schemes [9–11]. Here, the joint variables are sensed and then transformed into the Cartesian variables. The trajectory conversion problem is replaced by some kind of coordinate conversion inside the servo loop. The disadvantage of these techniques is that the resulting system may run at a lower sampling frequency compared with joint-based control systems. This would, in general, degrade the stability and disturbance rejection capability of the control system. Consequently, in this paper attention will be focused on the inverse kinematic problem with the purpose of generating a joint trajectory for joint-based control schemes.

The most popular method of providing a controlled manipulator with reference joint coordinates is the one proposed by Paul [2] based on homogeneous transforms [12]. In short, the Cartesian state vector $\mathbf{x}$ in (1) is rearranged in a compact matrix notation [5] which identifies a coordinate frame of the end effector with reference to the base coordinate frame. The solution method is trigonometric in nature as there is a set of non-trivial equations given by the above transforms. For a six-degree-of-freedom manipulator, for instance, 12 equations will yield the required six joint coordinates. The solution is then obtained in a sequential manner, isolating each variable by pre-multiplication—in some cases post-multiplication—by a certain number of transforms in each equation, thus requiring geometric intuition [2]. It should be mentioned, however, that the above equations must be combined with the use of the arctangent function of two arguments in order to avoid inherent problems of angle quadrant ambiguity. This solution technique is apparently valid for kinematically simple manipulators whose geometry is well understood in advance [13].

Efficient procedures for performing transformations from the position and velocity of the end effector to the corresponding joint angles and velocities have been established by Featherstone [14]. An iterative type solution for a robot with any type of configuration has been proposed by Takano [15].

In view of the above, computational solution techniques for the inverse kinematic problem have recently been proposed for general robot structures [3, 4]. A simple closed-loop dynamical system is used which, when driven by the desired Cartesian trajectory $\hat{\mathbf{x}}(t)$, yields not only a joint trajectory $\mathbf{q}(t)$, but also $\dot{\mathbf{q}}(t)$, such that $\mathbf{q}(t) \rightarrow \mathbf{f}^{-1}(\hat{\mathbf{x}}(t)) = \hat{\mathbf{q}}(t)$. The dynamical system is shown in Fig. 2. By adopting a very simple structure of the gain block (a constant positive definite matrix) in ref. 4 it is shown that the error $\hat{\mathbf{q}}(t) - \mathbf{q}(t)$ can be made arbitrarily small. A slight modification is then suggested to generate joint accelerations $\ddot{\mathbf{q}}(t)$ if required. In ref. 3, convergence of the error to zero is assured by using a more complicated non-linear $\mathbf{q}$-dependent

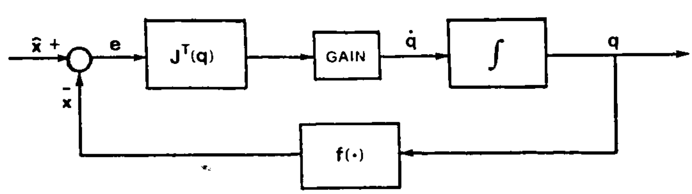


Figure 2. The general inverse kinematic scheme.

feedback gain. Both schemes are only general since they require that the position and orientation of the end effector be uniquely determined by means of six paramters. As a matter of fact, an explicit dependence of any three scalar orientation parameters (e.g. Euler angles, RPY angles) on the joint variables cannot be achieved, so the direct kinematic function (1) cannot be directly computed [16].

In this paper, manipulators with spherical wrists are considered and the hand orientation is defined by means of the usual unit vector hand frame [2]. An accurate analysis of the kinematical properties is performed with the purpose of reducing the overall computational burden.

2. KINEMATICS

A robot task is naturally specified in terms of end effector Cartesian coordinates. As shown in Fig. 3, the position of the end effector is described by a position vector $\mathbf{p}(t)$, whereas the orientation is defined through a unit approach vector $\mathbf{a}(t)$, a unit sliding

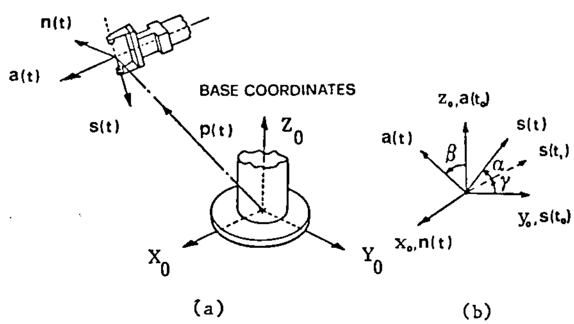


Figure 3. (a) Wrist position and hand orientation vectors; (b) Euler angles of the hand orientation [10].

vector $\mathbf{s}(t)$, and a normal unit vector $\mathbf{n}(t)=\mathbf{s}(t)\times\mathbf{a}(t)$. All these vectors are defined with reference to the base frame of the manipulator [2]. If non-redundant manipulators with spherical wrists are involved, the vector $\mathbf{q}$ in (1) can be partitioned since the wrist position is independent of the hand orientation. Let

$$\mathbf{q}=\begin{bmatrix}\mathbf{q}_p\\ \mathbf{q}_h\end{bmatrix} \tag{3}$$

be the partition, where $\mathbf{q}_p \in R^3$ are the joint coordinates which determine the position of the wrist and $\mathbf{q}_h$ are the other joint coordinates which, together with the previous ones, specify the orientation of the hand. Under this assumption, the kinematics of the wrist are obtained simply via the following set of equations:

$$\mathbf{p}=\mathbf{f}_p(\mathbf{q}_p), \qquad \mathbf{p}\in R^3 \tag{4}$$

$$\dot{\mathbf{p}}=\mathbf{J}_p(\mathbf{q}_p)\dot{\mathbf{q}}_p, \qquad \dot{\mathbf{p}}\in R^3 \tag{5}$$

where $\mathbf{f}_p$ denotes the non-linear vector valued function which defines the direct kinematics of the wrist and $\mathbf{J}_p$ is the associated Jacobian matrix $\partial\mathbf{f}_p/\partial\mathbf{q}_p$ which relates joint velocities $\dot{\mathbf{q}}_p$ to Cartesian velocities $\dot{\mathbf{p}}$ of the wrist.

Progressing similarly for the hand, but observing that the kinematics of the hand also depend on the wrist, we obtain the following set of equations for the hand unit vectors:

$$\left.\begin{aligned} \mathbf{s} &= \mathbf{f}_s(\mathbf{q}) \\ \mathbf{a} &= \mathbf{f}_a(\mathbf{q}) \\ \mathbf{n} &= \mathbf{f}_n(\mathbf{q}) \end{aligned}\right\} \qquad \mathbf{s}, \mathbf{a}, \mathbf{n} \in R^3 \tag{6}$$

$$\left.\begin{aligned} \dot{\mathbf{s}} &= \mathbf{J}_{sp}(\mathbf{q})\dot{\mathbf{q}}_p + \mathbf{J}_s(\mathbf{q})\dot{\mathbf{q}}_h \\ \dot{\mathbf{a}} &= \mathbf{J}_{ap}(\mathbf{q})\dot{\mathbf{q}}_p + \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}_h \\ \dot{\mathbf{n}} &= \mathbf{J}_{np}(\mathbf{q})\dot{\mathbf{q}}_p + \mathbf{J}_n(\mathbf{q})\dot{\mathbf{q}}_h. \end{aligned}\right\} \qquad \dot{\mathbf{s}}, \dot{\mathbf{a}}, \dot{\mathbf{n}} \in R^3 \tag{7}$$

In (6), $\mathbf{f}_s$, $\mathbf{f}_a$, $\mathbf{f}_n$ indicate the non-linear vector valued functions which uniquely define the direct kinematics of the hand in terms of the unit vectors **s**, **a**, **n**, respectively. In (7), $\mathbf{J}_{sp}$, $\mathbf{J}_{ap}$, $\mathbf{J}_{np}$ are the associated Jacobian matrices $\partial\mathbf{f}_s/\partial\mathbf{q}_p$, $\partial\mathbf{f}_a/\partial\mathbf{q}_p$, and $\partial\mathbf{f}_n/\partial\mathbf{q}_p$ which relate the contributions of joint velocities $\dot{\mathbf{q}}_p$ to the Cartesian velocities of the hand unit vectors, and finally $\mathbf{J}_s$, $\mathbf{J}_a$, $\mathbf{J}_n$ are the associated Jacobian matrices $\partial\mathbf{f}_s/\partial\mathbf{q}_h$, $\partial\mathbf{f}_a/\partial\mathbf{q}_h$, $\partial\mathbf{f}_n/\partial\mathbf{q}_h$ which relate the proper contributions of joint velocities $\dot{\mathbf{q}}_h$ to the Cartesian velocities of the hand unit vectors.

The kinematics of the hand can be further clarified on the basis of the structural relationship existing for the hand unit vectors **s**, **a**, **n** and their derivatives. Indeed, while there is no restriction on the value of **p** in (4), provided that the manipulator can reach the desired position, the vectors **s**, **a**, **n** must always be so that

$$\mathbf{s}^T\mathbf{s} = \mathbf{a}^T\mathbf{a} = \mathbf{n}^T\mathbf{n} = 1 \tag{8}$$

$$\mathbf{s}^T\mathbf{a} = \mathbf{a}^T\mathbf{n} = \mathbf{n}^T\mathbf{s} = 0 \tag{9}$$

where $(\ )^T$ denotes the transpose. Differentiating (8) and (9) with respect to time yields

$$\mathbf{s}^T\dot{\mathbf{s}} = \mathbf{a}^T\dot{\mathbf{a}} = \mathbf{n}^T\dot{\mathbf{n}} = 0 \tag{10}$$

$$\mathbf{s}^T\dot{\mathbf{a}} + \mathbf{a}^T\dot{\mathbf{s}} = \mathbf{a}^T\dot{\mathbf{n}} + \mathbf{n}^T\dot{\mathbf{a}} = \mathbf{n}^T\dot{\mathbf{s}} + \mathbf{s}^T\dot{\mathbf{n}} = 0. \tag{11}$$

On reduction of (7) and (10) the set of equations involving the hand associated Jacobians can be written in compact form as

$$\mathbf{u}^T\mathbf{J}_{up}\dot{\mathbf{q}}_p = -\mathbf{u}^T\mathbf{J}_u\dot{\mathbf{q}}_h, \qquad \mathbf{u} = \mathbf{s}, \mathbf{a} \ \text{ or } \ \mathbf{n}. \tag{12}$$

Since each of (12) must hold for any $\dot{\mathbf{q}}_p$, $\dot{\mathbf{q}}_h$, being $\mathbf{J}_u^T\mathbf{u}$ independent of the joint velocities, both members of (12) must be null. To this end, only two cases are possible:

$$\left.\begin{aligned} &\text{(i)}\ \ \dot{\mathbf{q}}_p \perp \mathbf{J}_{up}^T\mathbf{u}, && \dot{\mathbf{q}}_h \perp \mathbf{J}_u^T\mathbf{u} \\ &\text{(ii)}\ \ \mathbf{u} \in \mathcal{N}(\mathbf{J}_{up}^T), && \mathbf{u} \in \mathcal{N}(\mathbf{J}_u^T) \end{aligned}\right\} \quad \mathbf{u} = \mathbf{s}, \mathbf{a} \ \text{or} \ \mathbf{n} \tag{13}$$

where $\mathcal{N}(\mathbf{A})$ denotes the null space of matrix **A**. For the same reason as above, (i) cannot hold for any $\dot{\mathbf{q}}_p$, $\dot{\mathbf{q}}_h$, thus the following set of characteristic relations is obtained:

$$\begin{aligned}\mathbf{J}_{sp}^T\mathbf{s}&=\mathbf{J}_s^T\mathbf{s}=\mathbf{0}\\ \mathbf{J}_{ap}^T\mathbf{a}&=\mathbf{J}_a^T\mathbf{a}=\mathbf{0}\\ \mathbf{J}_{np}^T\mathbf{n}&=\mathbf{J}_n^T\mathbf{n}=\mathbf{0}.\end{aligned}\tag{14}$$

Furthermore, on reduction of (7) and (11), the orthogonality conditions yield

$$\begin{aligned}\mathbf{s}^T\mathbf{J}_{ap}\dot{\mathbf{q}}_p+\mathbf{s}^T\mathbf{J}_a\dot{\mathbf{q}}_h&=-\mathbf{a}^T\mathbf{J}_{sp}\dot{\mathbf{q}}_p-\mathbf{a}^T\mathbf{J}_s\dot{\mathbf{q}}_h\\ \mathbf{a}^T\mathbf{J}_{np}\dot{\mathbf{q}}_p+\mathbf{a}^T\mathbf{J}_n\dot{\mathbf{q}}_h&=-\mathbf{n}^T\mathbf{J}_{ap}\dot{\mathbf{q}}_p-\mathbf{n}^T\mathbf{J}_a\dot{\mathbf{q}}_h\\ \mathbf{n}^T\mathbf{J}_{sp}\dot{\mathbf{q}}_p+\mathbf{n}^T\mathbf{J}_s\dot{\mathbf{q}}_h&=-\mathbf{s}^T\mathbf{J}_{np}\dot{\mathbf{q}}_p-\mathbf{s}^T\mathbf{J}_n\dot{\mathbf{q}}_h\end{aligned}\tag{15}$$

and proceeding as above gives

$$\begin{aligned}\mathbf{J}_{sp}^T\mathbf{a}&=-\mathbf{J}_{ap}^T\mathbf{s}, &\quad \mathbf{J}_s^T\mathbf{a}&=-\mathbf{J}_a^T\mathbf{s}\\ \mathbf{J}_{ap}^T\mathbf{n}&=-\mathbf{J}_{np}^T\mathbf{a}, &\quad \mathbf{J}_a^T\mathbf{n}&=-\mathbf{J}_n^T\mathbf{a}\\ \mathbf{J}_{np}^T\mathbf{s}&=-\mathbf{J}_{sp}^T\mathbf{n}, &\quad \mathbf{J}_n^T\mathbf{s}&=-\mathbf{J}_s^T\mathbf{n}.\end{aligned}\tag{16}$$

On the basis of the above equations relating the Jacobians associated with the hand, additional analysis on the geometrical meaning of the generic Jacobian can be carried out. Referring to the generic hand unit vector $\mathbf{u}$ and assuming the wrist to be at rest for simplicity ($\dot{\mathbf{q}}_p=\mathbf{0}$), the vector $\dot{\mathbf{u}}=\mathbf{J}_u\dot{\mathbf{q}}_h$ must be able to span the whole plane orthogonal to $\mathbf{u}$. As $\mathbf{J}_u$ establishes the physical relationship between the joint velocities $\dot{\mathbf{q}}_h$ and the Cartesian velocities $\dot{\mathbf{u}}$, $\operatorname{rank}(\mathbf{J}_u)=2$ and $\mathscr{R}(\mathbf{J}_u)$, where $\mathscr{R}(\mathbf{A})$ denotes the range of matrix $\mathbf{A}$, is the plane orthogonal to $\mathbf{u}$. Since, moreover, $\mathscr{R}(\mathbf{J}_u)^\perp:=\mathscr{N}(\mathbf{J}_u^T)$, where $\perp$ denotes the orthogonal complement, the null space of $\mathbf{J}_u^T$ is spanned only by the direction of $\mathbf{u}$. Thus, in short

$$\operatorname{rank}(\mathbf{J}_s)=\operatorname{rank}(\mathbf{J}_a)=\operatorname{rank}(\mathbf{J}_n)=2, \qquad \text{for any} \quad \mathbf{q}\tag{17}$$

$$\left.\begin{aligned}\mathscr{N}(\mathbf{J}_s^T)&=\operatorname{span}(\mathbf{s})\\ \mathscr{N}(\mathbf{J}_a^T)&=\operatorname{span}(\mathbf{a})\\ \mathscr{N}(\mathbf{J}_n^T)&=\operatorname{span}(\mathbf{n}).\end{aligned}\right\}\ \text{for any } \mathbf{q}.\tag{18}$$

The same lines can be followed to derive similar properties for $\mathbf{J}_{sp}$, $\mathbf{J}_{ap}$ and $\mathbf{J}_{np}$.

Last but not least, the vectors in (16) ($\mathbf{J}_s^T\mathbf{a}$, $\mathbf{J}_a^T\mathbf{n}$, $\mathbf{J}_n^T\mathbf{s}$) can be shown to be linearly independent, thus building up a base in R^3. Denoting the above vectors as $\mathbf{k}_1$, $\mathbf{k}_2$, $\mathbf{k}_3$ and accounting for (16), they must be such that

$$\begin{aligned}\mathbf{k}_1&\in\mathscr{R}(\mathbf{J}_s^T)\cap\mathscr{R}(\mathbf{J}_a^T)\\ \mathbf{k}_2&\in\mathscr{R}(\mathbf{J}_a^T)\cap\mathscr{R}(\mathbf{J}_n^T)\\ \mathbf{k}_3&\in\mathscr{R}(\mathbf{J}_n^T)\cap\mathscr{R}(\mathbf{J}_s^T).\end{aligned}\tag{19}$$

If it were, for instance, $\mathscr{R}(\mathbf{J}_s^T)\equiv\mathscr{R}(\mathbf{J}_a^T)$, $\mathbf{k}_2$ would be parallel to $\mathbf{k}_3$, i.e. $\mathbf{k}_2=\alpha\mathbf{k}_3$, resulting in the absurdum $\mathbf{J}_n^T(\alpha\mathbf{s}-\mathbf{a})=\mathbf{0}$; in fact, $\alpha\mathbf{s}-\mathbf{a}$ is a vector in the plane orthogonal to $\mathbf{n}$, whereas $\mathbf{J}_n^T$ has its null space only in the direction of $\mathbf{n}$. Now it is understood that the three ranges of $\mathbf{J}_s^T$, $\mathbf{J}_a^T$, $\mathbf{J}_n^T$, respectively, do not coincide and, moreover, they cannot have a line in common, as in such a case the above three vectors $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ would all be along that line; this never happens since the vectors do not result parallel two by two. Consequently, the three vectors $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ are linearly independent and actually form a base of R^3.

3. INVERSE KINEMATIC SOLUTION TECHNIQUE

In the following, the inverse kinematic problem is solved by constructing a dynamical system whose input is the target trajectory in the task space and whose outputs are the corresponding joint position and velocity trajectories. This approach only requires the computation of direct kinematics. For manipulators with spherical wrists, a two-stage cascade solution algorithm is considered (Fig. 4). The inverse kinematic problem is divided into two subproblems: first the determination of $\mathbf{q}_p$, then the determination of $\mathbf{q}_h$.

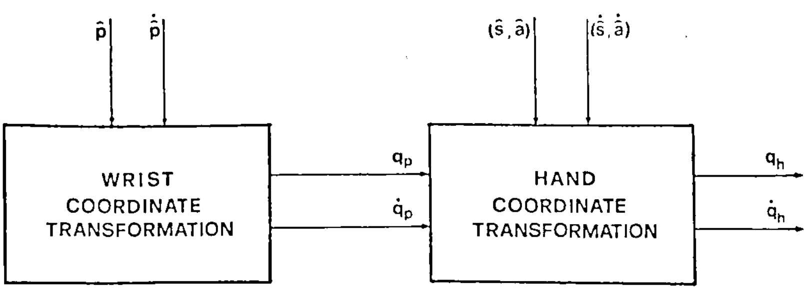


Figure 4. Two-stage coordinate transformation scheme.

3.1. *Wrist position*

Formulating the coordinate transformation problem for the wrist position as a result of the convergence of the state variables $\mathbf{q}_p$ to the desired variables $\hat{\mathbf{q}}_p$ of a dynamic system allows the following definition of a position error in the task space:

$$\mathbf{e}_p = \hat{\mathbf{p}} - \mathbf{p} \tag{20}$$

where $\hat{\mathbf{p}}$ is the desired position vector corresponding to $\hat{\mathbf{q}}_p$, and similarly $\mathbf{p}$ to $\mathbf{q}_p$, both via (4). In order to ensure the above convergence, error dynamics must be taken into account, i.e. via (5),

$$\dot{\mathbf{e}}_p = \dot{\hat{\mathbf{p}}} - \mathbf{J}_p(\mathbf{q}_p)\dot{\mathbf{q}}_p. \tag{21}$$

The point then is to relate $\dot{\mathbf{q}}_p$ to $\mathbf{e}_p$ so as to guarantee that the position error $\mathbf{e}_p$ goes asymptotically to zero and consequently $\mathbf{q}_p \to \hat{\mathbf{q}}_p$. Let

$$v_p = 0.5\mathbf{e}_p^T\mathbf{e}_p \tag{22}$$

be a positive definite Lyapunov function associated with $\mathbf{e}_p$. Differentiating with respect to time and accounting for (21) yields

$$\dot{v}_p = \mathbf{e}_p^T\dot{\hat{\mathbf{p}}} - \mathbf{e}_p^T\mathbf{J}_p\dot{\mathbf{q}}_p. \tag{23}$$

Up to now, $\dot{\mathbf{q}}_p$ can be chosen as

$$\dot{\mathbf{q}}_p = \gamma_p \mathbf{J}_p^T \mathbf{e}_p \tag{24}$$

with γ_p taken such that $\dot{v}_p$ is negative definite. To this end, the following choice for γ_p is easily seen to prove suitable:

$$\gamma_p = \alpha_p + \frac{\mathbf{e}_p^T \dot{\hat{\mathbf{p}}}}{\mathbf{e}_p^T \mathbf{J}_p \mathbf{J}_p^T \mathbf{e}_p}, \qquad \alpha_p > 0. \tag{25}$$

The corresponding scheme is shown in Fig. 5. The convergence rate and precision attainable for a given settling time are directly proportional to the gain α_p when $\mathbf{e}_p(0) \neq \mathbf{0}$. It could be argued that $\mathbf{J}_p^T \mathbf{e}_p$ vanishes when $\mathbf{e}_p \in \mathcal{N}(\mathbf{J}_p^T)$; however, observing that $\mathcal{N}(\mathbf{J}_p^T)$ is only in that direction along which a Cartesian force applied to the wrist is completely neutralized by the mechanical constraints of the manipulator, it is reasonable to assume that the target trajectory is planned consistently with the mechanical constraints. Then, from the implementation viewpoint, it is always $\mathbf{J}_p^T \mathbf{e}_p \neq \mathbf{0}$.

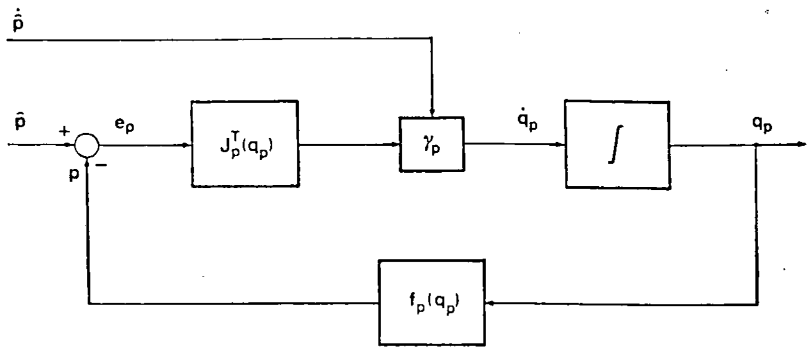


Figure 5. Coordinate transformation scheme for the wrist.

A further remark can be made about (25). Provided that $\mathbf{e}_p(0) = \mathbf{0}$, since (25) always guarantees a null tracking position error, it naturally introduces, in the neighborhood of $\mathbf{e}_p = \mathbf{0}$, an equivalent gain, dependent on state variables, which tends to ∞. This point leads to the generation of a $\dot{\mathbf{q}}_p$ rich in harmonics, whose effect on $\mathbf{q}_p$, however, is cut off by the filtering nature of the integrators. Therefore, whenever joint velocities are also needed, either additional filtering on $\dot{\mathbf{q}}_p$ must be introduced or finite tracking position errors must be admitted via the purely proportional control law

$$\dot{\mathbf{q}}_p = \alpha_p \mathbf{J}_p^T \mathbf{e}_p. \tag{26}$$

With this choice $\dot{v}_p$ is negative definite only outside a region in the error space which contains $\mathbf{e}_p = \mathbf{0}$ that is attractive for all trajectories. Obviously the maximum tracking position error will depend directly on $\|\dot{\hat{\mathbf{p}}}\|$ and inversely on α_p; it must be emphasized, however, that the steady-state ($\dot{\hat{\mathbf{p}}} = 0$) position error is identically zero.

An alternative coordinate transformation algorithm can be conceived by considering

$$\dot{\mathbf{q}}_p = \mathbf{J}_p^{-1}(\mathbf{M}_p \mathbf{e}_p + \dot{\hat{\mathbf{p}}}) \tag{27}$$

where $\mathbf{M}_p$ is a positive definite matrix whose eigenvalues affect the position error convergence rate to zero. If the Jacobian degenerates, (27) can still be used by considering an appropriate reduced order Jacobian. The control law (27) recalls a solution similar to the inverse Jacobian control [16]; it should be emphasized, however, that the main goal outlined in ref. 16 is on-line Cartesian-based control and not explicitly solving the inverse kinematic problem.

3.2. Hand orientation

As far as the hand orientation of the manipulator is concerned, a convergent algorithm similar to the one proposed for the wrist position can be synthesized, but with some substantial modifications. In actual fact, one must take into account the structural constraints due to the choice of the orthogonal unit vectors as the orientation variables. The point here is to define unambiguously an orientation error between the desired unit vectors ($\hat{\mathbf{s}}$, $\hat{\mathbf{a}}$, $\hat{\mathbf{n}}$) and the actual unit vectors ($\mathbf{s}$, $\mathbf{a}$, $\mathbf{n}$) which describes the orientation of the actual frame with respect to the desired one. To this end, the orientation error can be defined as

$$\mathbf{e}_h = \begin{bmatrix} \hat{\mathbf{s}} - \mathbf{s} \\ \hat{\mathbf{a}} - \mathbf{a} \end{bmatrix}; \tag{28}$$

then error dynamics, via (7), are described by

$$\dot{\mathbf{e}}_h = \begin{bmatrix} \dot{\hat{\mathbf{s}}} - \mathbf{J}_{sp}\dot{\mathbf{q}}_p - \mathbf{J}_s\dot{\mathbf{q}}_h \\ \dot{\hat{\mathbf{a}}} - \mathbf{J}_{ap}\dot{\mathbf{q}}_p - \mathbf{J}_a\dot{\mathbf{q}}_h \end{bmatrix}. \tag{29}$$

Similarly to the wrist position algorithm, $\dot{\mathbf{q}}_h$ must be related to $\mathbf{e}_h$ so as to ensure that the orientation error $\mathbf{e}_h$ goes asymptotically to zero ($\mathbf{s} = \hat{\mathbf{s}}$, $\mathbf{a} = \hat{\mathbf{a}}$, and obviously $\mathbf{n} = \hat{\mathbf{n}}$) and consequently $\mathbf{q}_h$ to $\hat{\mathbf{q}}_h$, where $\hat{\mathbf{q}}_h$ are the desired joint variables to be tracked. For this purpose let

$$v_h = 0.5\mathbf{e}_h^T\mathbf{e}_h \tag{30}$$

be a positive-definite Lyapunov function associated with $\mathbf{e}_h$. Differentiating with respect to time and accounting for (14) gives

$$\dot{v}_h = -\mathbf{s}^T\dot{\hat{\mathbf{s}}} - \mathbf{a}^T\dot{\hat{\mathbf{a}}} - \hat{\mathbf{s}}^T\dot{\mathbf{s}} - \hat{\mathbf{a}}^T\dot{\mathbf{a}} \tag{31}$$

which, by means of (7), yields

$$\dot{v}_h = -\mathbf{s}^T\dot{\hat{\mathbf{s}}} - \mathbf{a}^T\dot{\hat{\mathbf{a}}} - (\hat{\mathbf{s}}^T\mathbf{J}_{sp} + \hat{\mathbf{a}}^T\mathbf{J}_{ap})\dot{\mathbf{q}}_p - (\hat{\mathbf{s}}^T\mathbf{J}_s + \hat{\mathbf{a}}^T\mathbf{J}_a)\dot{\mathbf{q}}_h. \tag{32}$$

It can first be observed that, in virtue of (10) and (14), $v_h = \dot{v}_h = 0$ as $\mathbf{s} = \hat{\mathbf{s}}$ and $\mathbf{a} = \hat{\mathbf{a}}$, as may be expected. In order to investigate better the last term in (32) related to $\dot{\mathbf{q}}_h$, i.e. the control to be synthesized, projecting $\hat{\mathbf{s}}$ and $\hat{\mathbf{a}}$ on the frame ($\mathbf{s}$, $\mathbf{a}$, $\mathbf{n}$) and taking into account (14) yields

$$\hat{\mathbf{s}}^T\mathbf{J}_s + \hat{\mathbf{a}}^T\mathbf{J}_a = \hat{s}_a\mathbf{a}^T\mathbf{J}_s + \hat{s}_n\mathbf{n}^T\mathbf{J}_s + \hat{a}_n\mathbf{n}^T\mathbf{J}_a + \hat{a}_s\mathbf{s}^T\mathbf{J}_a \tag{33}$$

where the scalar quantities with the subscripts denote the components of $\hat{\mathbf{s}}$ and $\hat{\mathbf{a}}$ along the axes of the frame ($\mathbf{s}$, $\mathbf{a}$, $\mathbf{n}$). On the basis of (16), (33) can be rewritten as

$$\hat{\mathbf{s}}^T\mathbf{J}_s + \hat{\mathbf{a}}^T\mathbf{J}_a = (\hat{a}_s - \hat{s}_a)\mathbf{s}^T\mathbf{J}_a - \hat{a}_n\mathbf{a}^T\mathbf{J}_n + \hat{s}_n\mathbf{n}^T\mathbf{J}_s. \tag{34}$$

By observing that the vectors ($\mathbf{s}^T\mathbf{J}_a$, $\mathbf{a}^T\mathbf{J}_n$, $\mathbf{n}^T\mathbf{J}_s$) give a base in R^3 [see vectors ($\mathbf{k}_1$, $\mathbf{k}_2$,

$\mathbf{k}_3$) in Section 2], (34) vanishes if and only if

$$\begin{aligned}\hat{a}_s &= \hat{s}_a \\ \hat{a}_n &= \hat{s}_n = 0.\end{aligned} \tag{35}$$

Related to the conditions (35), two possible indeterminacies of the mutual orientation of the two frames arise:

$$\text{(i)}\quad \hat{\mathbf{n}} = -\mathbf{n}, \quad \text{for any} \quad \mathbf{s}, \hat{\mathbf{s}}, \mathbf{a}, \hat{\mathbf{a}} \tag{36}$$

$$\text{(ii)}\quad \hat{\mathbf{n}} = \mathbf{n}, \quad \hat{\mathbf{s}} = -\mathbf{s}, \quad \hat{\mathbf{a}} = -\mathbf{a}. \tag{37}$$

In the first case (36), it results that $\|\mathbf{e}_h\| = 2$, where $\|\cdot\|$ denotes the Euclidean norm of the vector, whereas in the second case (37), $\|\mathbf{e}_h\| = \|\mathbf{e}_{h\max}\| = 2\sqrt{2}$; the occurrence of these situations is apparently of no interest for a convergent algorithm. In other words, provided that the algorithm starts with the same initial conditions ($\mathbf{s} = \hat{\mathbf{s}}$, $\mathbf{a} = \hat{\mathbf{a}}$), or eventually with $\|\mathbf{e}_h\| < 2$, realizing a locally negative definite $\dot{v}_h$ always guarantees that the algorithm works, ensuring a good tracking accuracy and avoiding the above indeterminacies.

To this extent, since the first terms in (32), as $\mathbf{s} \to \hat{\mathbf{s}}$ and $\mathbf{a} \to \hat{\mathbf{a}}$, go to zero as (34), the following choice for $\dot{\mathbf{q}}_h$ proves adequate:

$$\dot{\mathbf{q}}_h = \gamma_h \mathbf{sgn}(\mathbf{J}_s^T \hat{\mathbf{s}} + \mathbf{J}_a^T \hat{\mathbf{a}}) \tag{38}$$

where $\mathbf{sgn}\ \mathbf{w} := (\operatorname{sgn} w_1 \ \ldots \ \operatorname{sgn} w_r)^T$, $\mathbf{w} \in R^r$, with

$$\begin{aligned}\gamma_h \geq {} & [\|\dot{\hat{\mathbf{s}}}\|_{\max} + \|\dot{\hat{\mathbf{a}}}\|_{\max} + \|\dot{\mathbf{q}}_p\|_{\max}(|\Lambda(\mathbf{J}_{sp})| + |\Lambda(\mathbf{J}_{ap})|)] \\ & \times (|\lambda(\mathbf{J}_s)| + |\lambda(\mathbf{J}_a)|)^{-1}\end{aligned} \tag{39}$$

where $\Lambda(\mathbf{A})$ and $\lambda(\mathbf{A})$ denote the maximum and the minimum eigenvalue of matrix $\mathbf{A}$, respectively.

For this algorithm similar remarks as for the wrist are in order. Indeed, if one admits finite tracking orientation errors, once $\mathbf{e}_h(0) = \mathbf{0}$, the purely proportional control law (Fig. 6).

$$\dot{\mathbf{q}}_h = \gamma_h (\mathbf{J}_s^T \hat{\mathbf{s}} + \mathbf{J}_a^T \hat{\mathbf{a}}) \tag{40}$$

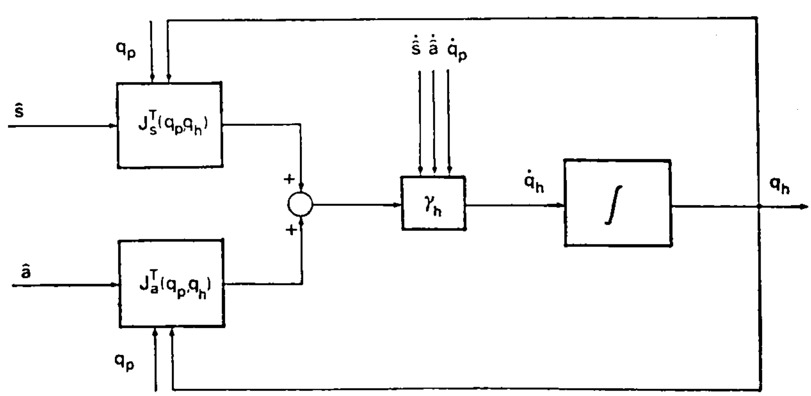


Figure 6. Coordinate transformation scheme for the hand.

ensures the convergence of the algorithm with zero errors at the steady state ($\dot{\hat{\mathbf{s}}}=\dot{\hat{\mathbf{a}}}=\mathbf{0}$ and $\dot{\mathbf{q}}_p=\mathbf{0}$), while the tracking errors will depend directly on the velocities $\|\dot{\hat{\mathbf{s}}}\|$, $\|\dot{\hat{\mathbf{a}}}\|$ and $\|\dot{\mathbf{q}}_p\|$, and inversely on the gain γ_h.

Incidentally, it can be observed that augmenting the orientation error space dimension by introducing an additional component $\hat{\mathbf{n}}-\mathbf{n}$ would lead to a positive-definite Lyapunov function

$$v_h=0.5(\mathbf{e}_s^T\mathbf{e}_a^T\mathbf{e}_n^T)\begin{bmatrix}\mathbf{e}_s\\ \mathbf{e}_a\\ \mathbf{e}_n\end{bmatrix} \tag{41}$$

whose derivative, accounting for (14) and (16) and projecting as above, would give

$$\begin{aligned}\dot{v}_h= & -\mathbf{s}^T\dot{\hat{\mathbf{s}}}-\mathbf{a}^T\dot{\hat{\mathbf{a}}}-\mathbf{n}^T\dot{\hat{\mathbf{n}}}-(\hat{\mathbf{s}}^T\mathbf{J}_{sp}+\hat{\mathbf{a}}^T\mathbf{J}_{ap}+\hat{\mathbf{n}}^T\mathbf{J}_{np})\dot{\mathbf{q}}_p \\ & -[(\hat{a}_s-\hat{s}_a)\mathbf{s}^T\mathbf{J}_a+(\hat{n}_a-\hat{a}_n)\mathbf{a}^T\mathbf{J}_n+(\hat{s}_n-\hat{n}_s)\mathbf{n}^T\mathbf{J}_s]\dot{\mathbf{q}}_h.\end{aligned} \tag{42}$$

Even in this case, of course, $\dot{v}_h$ would suffer at least from the same indeterminacies as above; thus, from the implementation viewpoint, selecting only two components of the orientation error, as in (28), will suffice. This point stems from the fact that $\mathbf{n}$ is uniquely determined as the vector product $\mathbf{n}=\mathbf{s}\times\mathbf{a}$. In summary, the orientation error (28) serves the purpose of determining a simple algorithm which assures the asymptotic stability of the constructed dynamical system (Fig. 6) for the solution of the hand orientation problem. Other techniques of defining the orientation error [10] are more complicated from a computational viewpoint since they do not allow a coordinate conversion algorithm as simple as the one proposed to be found.

4. ANALYSIS OF DIGITAL IMPLEMENTATION AND A NUMERICAL EXAMPLE

The Stanford arm (Fig. 1) was selected as a reference to evaluate the digital implementation performance of the coordinate transformation algorithm set forth in this paper. The kinematics of the Stanford arm are fully reformulated in the Appendix.

For this manipulator the trigonometric closed-form solution for the inverse transformation of equations (A1) and (A2) does exist; adopting the solution proposed by Paul [2] requires 14 transcendental function calls, 31 floating point multiplications, 15 additions, together with the use of the arctangent function of two arguments. An increased number of mathematical computations are required if joint velocities are needed as well [2].

The method described in this paper requires a greater number of mathematical computations (10 transcendental function calls, 86 floating point multiplications and 46 additions). No additional computations are required, however, either to generate the joint velocities, if needed, or to select a joint solution among the feasible ones. To be more specific regarding the latter point, the analytical method [2] requires that the robot programming language be provided with a set of instructions allowing the user to choose the joint solution congruent with the desired kinematical configuration of the arm (e.g. elbow up or elbow down, etc.). The computational method proposed above overcomes this drawback since, once the initial configuration of the arm has been specified, the solution algorithm progresses with continuity along the given trajectory, the tracking errors always being very small. The algorithm has also been digitally implemented on a single dedicated microprocessor system [17].

In order to show the effectiveness of the algorithm, a numerical example was simulated. The target trajectory in the Cartesian space consisted of:

(a) a straight line from point $\hat{\mathbf{p}}_0 \equiv (0.3, -0.3, 0.2)$ m to point $\hat{\mathbf{p}}_f \equiv (0.4, 0.05, 0.04)$ m; and
(b) Euler angle excursions from $(\hat{\alpha}_0, \hat{\beta}_0, \hat{\gamma}_0) \equiv (45, 75, 105)^\circ$ to $(\hat{\alpha}_f, \hat{\beta}_f, \hat{\gamma}_f) \equiv (135, 165, 195)^\circ$.

Two different velocity profiles were imposed for both the wrist position and the hand orientation:

(a) the triangular one (acceleration, deceleration) of Fig. 7a; and
(b) the trapezoidal one (acceleration, cruise, deceleration) of Fig. 7b, with very small acceleration times.

Four pairs of values for the maximum velocities v_{max} in Fig. 7 were chosen in order to investigate the influence of velocity on the tracking accuracy (see Table 1); the slower trajectories are typical of those tasks where the robot is required to perform some work, whereas the faster ones refer to material handling tasks.

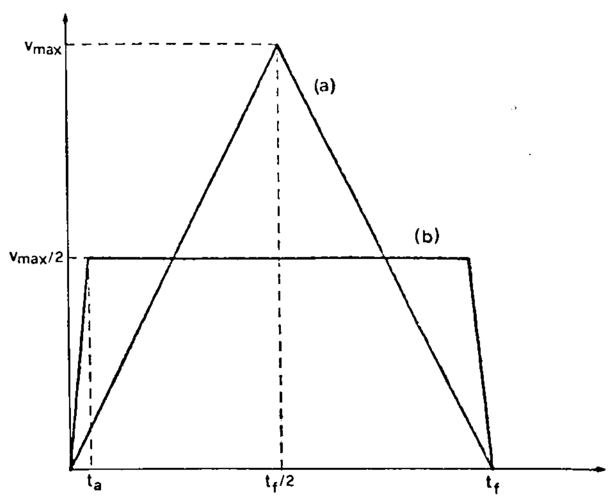


Figure 7. Velocity profiles in the work space: (a) triangular; (b) trapezoidal.

Table 1.
Selected parameters for the velocity profiles of Fig. 7

t_a(s)	t_f(s)	$\|\dot{p}_{max}\|$ (m/s)	$\dot{\alpha}_{max} = \dot{\beta}_{max} = \dot{\gamma}_{max}$ (°/s)
0.05	1	0.8	90
0.2	4	0.2	22.5
0.5	10	0.08	9
2	40	0.02	2.25

Since the simulated system is a sample data system with a solution sample period of 2 ms, if the discontinuous solutions for the control laws $\dot{\mathbf{q}}_p$ and $\dot{\mathbf{q}}_h$ [(24) and (38) respectively] are adopted, finite tracking errors will be expected. As a consequence, the proportional control laws (26) and (40) were chosen to consider the worst case since finite tracking errors are already expected for the continuous time system. The inherent advantage with this choice, however, is that smooth joint velocity trajectories are also generated directly at no additional cost. The values α_p in (26) and γ_h in (40) at the designer's disposal were set at 950 and 450 respectively.

Figures 8 and 9 show the tracking position errors $\|\mathbf{e}_p\|$ for the slowest and the fastest trajectories, respectively, while in Figs 10 and 11 are reported the maximum tracking orientation errors, in this case $\mathbf{e}_a$, which are evaluated as the arc-cosine of the

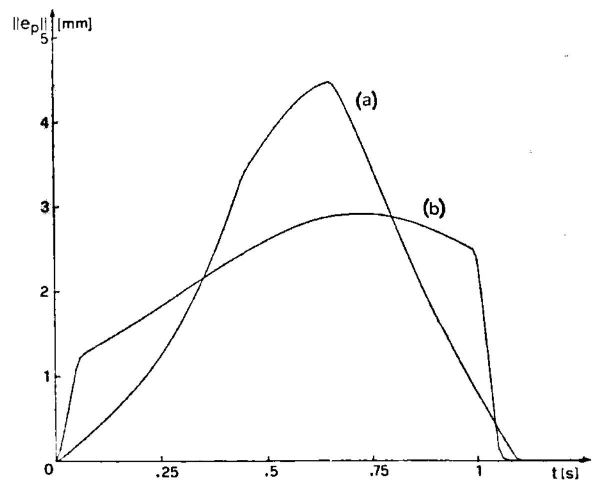


Figure 8. Tracking position errors for the fastest trajectories: (a) triangular profile; (b) trapezoidal profile.

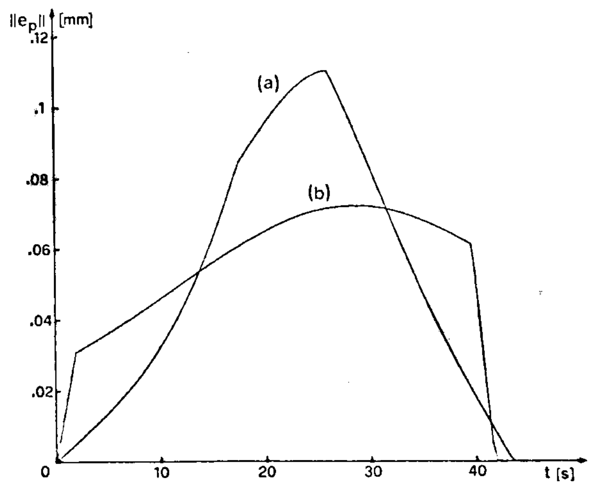


Figure 9. Tracking position errors for the slowest trajectoires: (a) triangular profile; (b) trapezoidal profile.

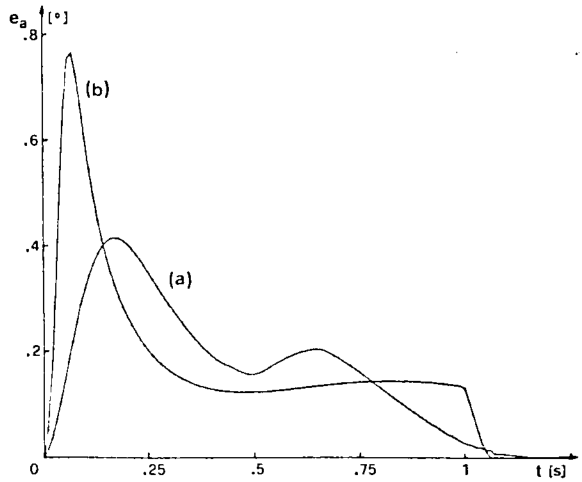


Figure 10. Tracking orientation errors for the fastest trajectories: (a) triangular profile; (b) trapezoidal profile.

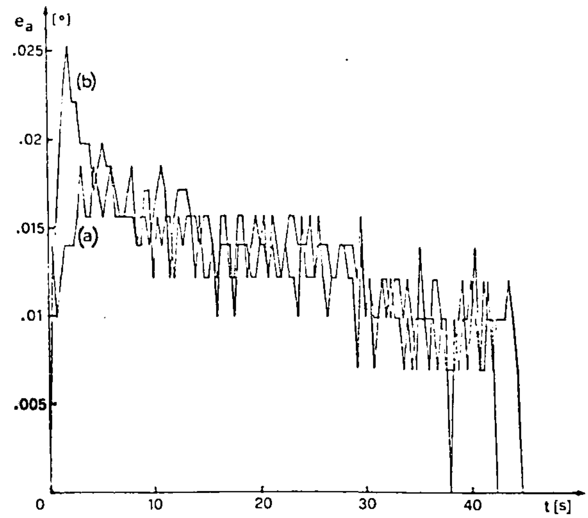


Figure 11. Tracking orientation errors, for the slowest trajectories: (a) triangular profile; (b) trapezoidal profile.

scalar products $\mathbf{a}^T\hat{\mathbf{a}}$. Since the same initial conditions occur $[\mathbf{e}_p(0)=\mathbf{e}_h(0)=0]$, the tracking errors are very small along all the trajectories, whereas at the steady state they vanish in virtue of the closed-loop structure of the coordinate transformation schemes of Figs 5 and 6. Finally, in Fig. 12, for the two different velocity profiles, the maximum position errors are reported vs. the four values of the maximum wrist velocities of Table 1. It is apparent that there is a strict direct proportionality between these errors and the velocities as anticipated in Section 3. This result is acceptable in practice, since a higher tracking performance is expected for slow assembly and processing tasks than in fast material handling tasks.

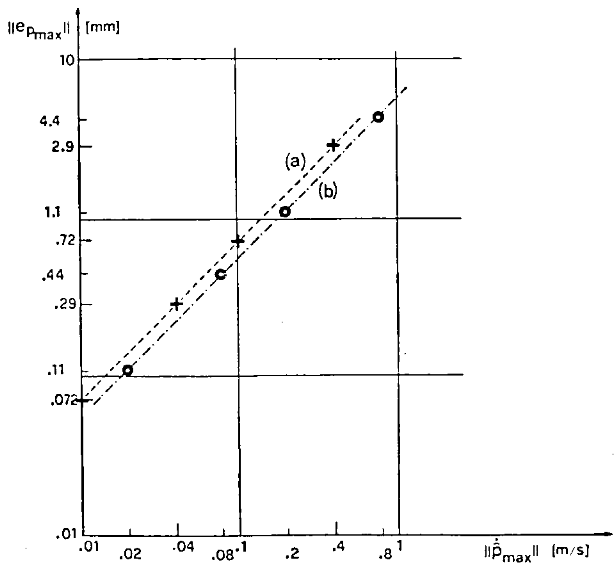


Figure 12. Maximum tracking position errors vs. maximum wrist velocities: (a) triangular profile; (b) trapezoidal profile.

5. CONCLUSIONS

This paper has presented a computational method for solving the inverse kinematic problem for robotic manipulators with spherical wrists. The approach is based on a general closed-loop dynamic scheme which, once the target trajectory in the task space is given as input, gives the joint trajectories (position + velocity) as its output. The main advantages of this technique are:

(a) it is based only on direct kinematics and does not require any geometric intuition;
(b) the use of the transpose of the Jacobian may avoid problems in correspondence of kinematic singularities; and
(c) given the initial arm configuration, uniqueness of the solution is ensured, since the algorithm generates, at each step, solutions which are adjacent to the preceding ones.

The structural properties of the kinematics of the unit vectors defining the orientation of the end effector were investigated with the purpose of achieving a two-stage algorithm which decreases the overall computational burden. As a matter of fact, the hand orientation algorithm is even simpler than the wrist position algorithm. It was also shown how the joint velocities can be generated directly on condition that very small tracking errors are permitted. Finally, a case study was developed and the simulation results for the Stanford arm show the effectiveness of the scheme proposed with two different velocity profiles.

Current research developments are dedicated to the application of this computational technique to more complicated manipulator geometries. The case of two-by-two intersecting axes at the end effector has already been developed [18, 19]. Non-converging axes are treated in refs 20 and 21. The extension to the case of redundant manipulators is also being currently investigated [21, 22].

Acknowledgement

This work was partially supported by MPI.

REFERENCES

1. T. Lozano-Pérez, "Robot programming," *Proc. IEEE*, vol. 71, no. 7, pp. 821–841, 1983.
2. R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control.* Cambridge, MA: MIT Press, 1982.
3. A. Balestrino, G. De Maria and L. Sciavicco, "Robust control of robotic manipulators," in *Proc. 9th IFAC World Congress*, Budapest, Hungary, July 1984.
4. W. A. Wolovich and H. Elliott, "A computational technique for inverse kinematics," in *Proc. 23rd IEEE CDC*, Las Vegas, NV, Dec. 1984.
5. J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *ASME J. Appl. Mech.*, pp. 215–221, 1955.
6. A. Balestrino, G. De Maria and L. Sciavicco, "An adaptive model following control for robotic manipulators', *ASME J. Dynamic Syst., Meas., Contr.*, vol. 105, no. 3, pp. 143–151, 1983.
7. E. Freund, "Fast nonlinear control with arbitrary pole placement for industrial robots and manipulators," *Int. J. Robotics Res.*, vol. 1, no. 1, pp. 65–78, 1982.
8. J.-J. E. Slotine, "The robust control of robot manipulators," *Int. J. Robotics Res.*, vol. 4, no. 2, pp. 49–64, 1985.
9. D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man–Machine Syst.*, vol. MMS-10, no. 2, pp. 47–53, 1969.
10. J. Y. S. Luh, M. W. Walker and R. P. C. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Trans. Automat. Contr.*, vol. AC-25, no. 3, pp. 468–474, 1980.
11. A. Balestrino, G. De Maria and L. Sciavicco, "Adaptive control of manipulators in the task oriented space," in *Proc. 13th ISIR/Robots 7*, Chicago, IL, April 1983.
12. R. P. Paul, B. Shimano and G. E. Mayer, "Kinematic control equations for simple manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, no. 6, pp. 449–455, 1981.
13. D. L. Pieper, *The Kinematics of Manipulators under Computer Control*, Memo. AIM-72, Stanford Artificial Intelligence Project, Stanford, CA, 1968.
14. R. Featherstone, "Position and velocity transformations between robot end-effector coordinates and joint angles," *Int. J. Robotics Res.*, vol. 2, no. 2, pp. 35–45, 1983.
15. M. Takano, "A new effective solution for inverse kinematics problem (synthesis) of a robot with any type of configuration," *J. Fac. Eng., Univ. Tokyo*, vol. 38, no. 2, pp. 107–135, 1985.
16. J. J. Craig, *Introduction to Robotics: Mechanics and Control.* Reading, MA: Addison Wesley, 1986.
17. G. De Maria and R. Marino, "A discrete algorithm for solving the inverse kinematic problem for robotic manipulators," in *Proc. '85 ICAR*, Tokyo, Japan, Sept. 1985.
18. L. Sciavicco and B. Siciliano, "Coordinate transformation: a solution algorithm for one class of robots," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, no. 4, pp. 550–559, 1986.
19. L. Sciavicco and B. Siciliano, "An inverse kinematic solution algorithm for robots with two-by-two intersecting axes at the end effector," in *Proc. IEEE Int. Conf. Robotics, Automation*, San Francisco, CA, April 1986.
20. G. De Maria, L. Sciavicco and B. Siciliano, "A general solution algorithm to coordinate transformation for robotic manipulators," in *Proc. '85 ICAR,* Tokyo, Japan, Sept. 1985.
21. L. Sciavicco and B. Siciliano, "Solving the inverse kinematic problem for robotic manipulators," in *Proc. 6th CISM–IFToMM Ro. Man. Sy.*, Kraców, Poland, Sept. 1986.
22. L. Sciavicco and B. Siciliano, "A dynamic solution to the inverse kinematic problem for redundant manipulators," in *Proc. IEEE Int. Conf. Robotics, Automation*, Raleigh, NC, March–April 1987.

APPENDIX

Figure 1 shows the six-degree-of-freedom Stanford arm [2]; starting from the base there are two revolute joints (θ_1, θ_2) and one prismatic joint (d_3) which identify $\mathbf{q}_p$, then another three revolute joints $(\theta_4, \theta_5, \theta_6)$, i.e. $\mathbf{q}_h$. Using the short notations $\sin\theta_i = S_i$ and $\cos\theta_i = C_i$, equations (4) and (6) can be written respectively as

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} C_1S_2d_3 - S_1d_2 \\ S_1S_2d_3 + C_1d_2 \\ C_2d_3 \end{bmatrix} \tag{A1}$$

$$\mathbf{s} = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \begin{bmatrix} C_1 - C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6 - S_1(-S_4C_5S_6 + C_4C_6) \\ S_1 - C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6 + C_1(-S_4C_5S_6 + C_4C_6) \\ S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6 \end{bmatrix}$$

$$\mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5 \\ S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5 \\ -S_2C_4S_5 + C_2C_5 \end{bmatrix} \tag{A2}$$

$$\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} C_1C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6 - S_1(S_4C_5C_6 + C_4S_6) \\ S_1C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6 + C_1(S_4C_5C_6 + C_4S_6) \\ -S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6 \end{bmatrix}.$$

Differentiating **p**, **s**, **a**, and **n** with respect to **q** yields the Jacobian matrices defined in (5) and (7) which can be written in compact form as

$$\mathbf{J}_p(\mathbf{q}_p) = [\mathbf{z}\times\mathbf{p} \,|\, \mathbf{t}\times\mathbf{p} \,|\, \mathbf{r}] \tag{A3}$$

$$\mathbf{J}_{sp}(\mathbf{q}) = [\mathbf{z}\times\mathbf{s} \,|\, \mathbf{t}\times\mathbf{s} \,|\, \mathbf{0}], \quad \mathbf{J}_s(\mathbf{q}) = [\mathbf{r}\times\mathbf{s} \,|\, S_6\mathbf{a} \,|\, -\mathbf{n}]$$

$$\mathbf{J}_{ap}(\mathbf{q}) = [\mathbf{z}\times\mathbf{a} \,|\, \mathbf{t}\times\mathbf{a} \,|\, \mathbf{0}], \quad \mathbf{J}_a(\mathbf{q}) = [\mathbf{r}\times\mathbf{a} \,|\, C_6\mathbf{n} - S_6\mathbf{s} \,|\, \mathbf{0}] \tag{A4}$$

$$\mathbf{J}_{np}(\mathbf{q}) = [\mathbf{z}\times\mathbf{n} \,|\, \mathbf{t}\times\mathbf{n} \,|\, \mathbf{0}], \quad \mathbf{J}_n(\mathbf{q}) = [\mathbf{r}\times\mathbf{n} \,|\, -C_6\mathbf{a} \,|\, \mathbf{s}]$$

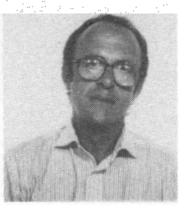
where $\mathbf{z} = (0 \quad 0 \quad 1)^T$, $\mathbf{t} = (-S_1 \quad C_1 \quad 0)^T$, $\mathbf{r} = (C_1S_2 \quad S_1S_2 \quad C_2)^T$ and $\mathbf{0} = (0 \quad 0 \quad 0)^T$. It must be emphasized that some handy reductions have been carried out in deriving the above Jacobians since, for a given kinematic configuration, **p**, **s**, **a**, and **n** are already known from (A1) and (A2); in this way, the computation time can be conveniently reduced.

For the sake of peculiarity, equations (12), (16), (17), and (18) together with the property that $(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3)$ represent a base of R^3 can all be easily verified by the expressions given in (A2) and (A4).

Bruno Siciliano was born in Napoli, Italy, on October 27, 1959. He received the Laurea degree and the Ph.D. degree in electronic engineering from the Università di Napoli, in 1982 and 1987 respectively.

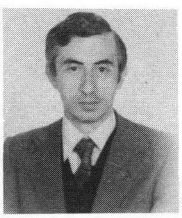
Since 1982 he has been a Research Assistant at the Dipartimento di Informatica e Sistemistica, Università di Napoli. His research interests include robust control of non-linear systems, kinematics dynamics and control of robotic manipulators, and control of lightweight flexible arms. From Fall 1985 to Spring 1986 he has been a Visiting Scholar at the G. W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia, under a NATO Research Fellowhsip.

In December 1986, Dr. Siciliano received the Jean Vertut National Award for the most original paper published in robotics.

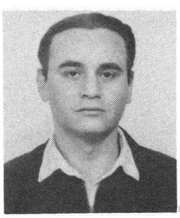


Lorenzo Sciavicco was born in Roma, Italy, on December 8, 1938. He received the Doctoral degree in electronic engineering from the Università di Roma in 1963.

Since 1970 he has been with the Università di Napoli, Italy where he is presently Professor in the area of Automatic Control at the Dipartimento di Informatica e Sistemistica. His fields of interest are in automatic control theory and applications; in particular, stability of nonlinear control systems and robotics.



Aldo Balestrino was born near Salerno, Italy, in 1941. He graduated in electronic engineering at the University of Naples in 1965. From 1967 to 1979 he was Research Fellow and Assistant Professor at the Institute of Electrical Engineering of the University of Naples in the field of Automatic Control. Full professor of Systems Theory at the same Institute since 1980 and temporary teacher at Naval Universitary Institute of Naples from 1980 to 1982, in 1983 he joined the University of Pisa as full professor of Automatic Control. Currently he is the Head of the Institute of Electrical Engineering in Pisa, Italy. His research interests include the fields of systems theory, variable structure control systems, multivariable control and robotics.



Giuseppe De Maria was born in Naples, Italy, on 11 December 1948. He received the doctoral degree in Electronic Engineering from Naples University in 1973. Since 1975 he has been at the Istituto Elettrotecnico; recently he has joined the Dipartimento di Informatica e Sistemistica, University of Naples. Since 1978 he has been associate professor of automatic control. His research interests are in adaptive control, non-linear systems and robotics.