

## A GENERAL SOLUTION ALGORITHM TO COORDINATE TRANSFORMATION FOR ROBOTIC MANIPULATORS

G. De Maria   L. Sciavicco   B. Siciliano

Dipartimento di Informatica e Sistemistica, University of Naples  
via Claudio, 21 - 80125 Naples, Italy.

### ABSTRACT

Coordinate transformation plays one of the most important roles in robotic manipulator control. Typical robot tasks are specified in work space coordinates, usually a Cartesian frame, whereas control actions are developed on joint space coordinates. The goal of the paper is to establish a solution algorithm which works for any nonredundant kinematic structure with the last three revolute axes: concurrent, two-by-two intersecting, nonintersecting at all. The coordinate transformation is turned into a dynamic problem, and the solution convergence is assured by means of the Lyapunov direct method. The resultant algorithms prove effective since they only make use of direct kinematics. Examples are finally developed.

### INTRODUCTION

As it is well known, a serial-link manipulator consists of a sequence of mechanical links connected by actuated joints. The relationship between two connective joint coordinates is well defined through four link parameters; one is a joint variable and the others are geometric parameters. The basis for all manipulator control techniques is the relationship between the Cartesian coordinates of the end effector and the joint coordinates. As a rule, the direct (joint-to-Cartesian space) relationship is unique, whereas the inverse (Cartesian-to-joint) transformation is not. Actually, while there is only one end effector state for a given set of joint coordinates, there are a number of different joint configurations which all place the end effector in the same position and orientation, and in each case geometric intuition is required to find one solution. Besides, if one is tempted to apply without exception the well known trigonometric method [1], not all kinematic structures allow for a closed form solution for the inverse transformation [2].

The goal of this paper is to present a general solution algorithm for the inverse kinematic problem which only makes use of direct kinematics of the manipulator. The kinematics of a general manipulator is formulated and structural kinematic properties are evidenced. The conversion problem is formulated as a dynamical one and convergent solution algorithms are derived accounting for the kinematic properties of the three basic mechanical structures considered, as regards the last three joints geometric parameters. Two robotic arms, the PUMA 560 [7] and the prototype presented in [2], are taken as examples in order to apply the proposed coordinate transformation algorithms along a trajectory specified in the Cartesian space.

### KINEMATICS

For an  $N$ -degree of freedom manipulator there will be conventionally  $N$  links and  $N$  joints. The relationship between the intermediate joint coordinate frames  $n-1$  and  $n$  can be expressed in terms of four kinematic parameters [3], see Fig. 1. A link  $L_n$  is characterized by its length  $a_n$ , a common normal distance between the axes of the two joints related to the link, and by the twist angle  $\alpha_n$  between the two joint axes. Each joint axis has two normals attributed to it, one for each link. The position of two links,  $L_{n-1}$  and  $L_n$ , relative to each other is defined by the distance

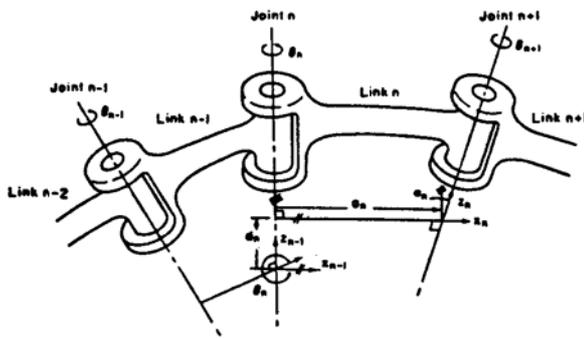


Fig. 1. Kinematic parameters.

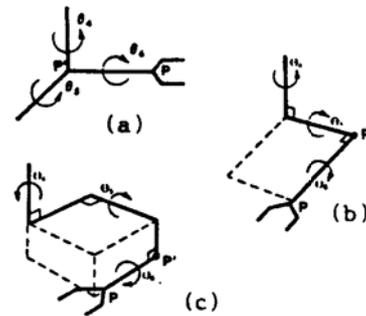


Fig. 2. a) Concurrent axes, b) 2-by-2 intersecting axes, c) nonintersecting axes.

$d_n$  between the links, and by the angle  $\theta_n$  between the links;  $d_n$  is the distance between the normals along the joint  $J_n$  axis, and  $\theta_n$  is the angle between the normals measured in the plane normal to the joint axis. The geometric parameters, that is  $a_n$  and  $\alpha_n$ , plus  $d_n$  or  $\theta_n$  for a revolute joint or a prismatic joint respectively, are those which accomplish the motion of joint  $J_n$ .

The structures here considered are characterized through the following constraints on the geometric parameters of the last three joints,

- a) concurrent axes:  $a_4 = a_5 = d_5 = 0$  (Fig. 2a),
- b) two-by-two intersecting axes:  $a_4 = a_5 = 0$ ,  $d_5 \neq 0$  (Fig. 2b),
- c) nonintersecting axes:  $a_4 \neq 0$ ,  $a_5 \neq 0$  (Fig. 2c).

A robot task is naturally specified in terms of end effector Cartesian coordinates  $(p_x, p_y, p_z, \alpha, \beta, \gamma)$  with respect to the base frame;  $p_i$ 's are the components of the end effector position vector  $\underline{p}$ , and  $\alpha, \beta, \gamma$  are the Euler angles which define its orientation (roll, pitch and yaw angles can be adopted as well). The orientation, however, can be conveniently described through a unit approach vector  $\underline{a}$ , a unit sliding vector  $\underline{s}$ , and a unit normal vector  $\underline{n}$ , [4], see Fig. 3.

The orientation frame  $(\underline{s}, \underline{a}, \underline{n})$  defined with reference to the base coordinates of the manipulator, can be easily determined starting from the Euler angles. Such frame will be referred to in the following since it allows for a unique definition of the orientation in terms of direct relationship with the joint variables. Under these assumptions, for any robot kinematic structure with known geometric parameters, the direct kinematics can be written as

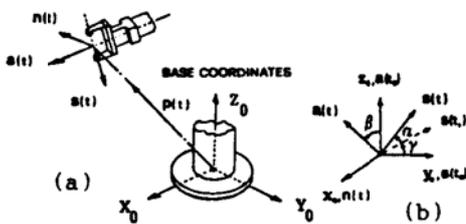


Fig. 3. a) Orientation unit vectors, b) Euler angles.

$$\underline{p} = \underline{f}_p(\underline{q}), \quad \underline{s} = \underline{f}_s(\underline{q}), \quad \underline{a} = \underline{f}_a(\underline{q}), \quad (1)$$

where  $\underline{q}$  is the  $(6 \times 1)$  vector of joint coordinates, and  $\underline{f}_p, \underline{f}_s, \underline{f}_a$  are nonlinear vector functions which are always unique;  $\underline{n} = \underline{f}_n(\underline{q})$  is redundant since it can be determined as the vector product  $\underline{s} \times \underline{a}$ .

At this extent, since the aim of the work is to seek a convergent algorithm which could solve for  $\underline{q}$  by inverting (1), the dynamics of such an algorithm involves the differentiation of (1) with respect to time, i.e.

$$\dot{\underline{p}} = J_p \dot{\underline{q}}, \quad \dot{\underline{s}} = J_s \dot{\underline{q}}, \quad \dot{\underline{a}} = J_a \dot{\underline{q}}, \quad (2)$$

where  $J_p, J_s, J_a$  are the Jacobian matrices of the kind  $J(\underline{q}) = \partial f / \partial \underline{q}$ . The issue of defining the end effector orientation in terms of unit vectors  $\underline{s}, \underline{a}$  implicitly requires that the three independent constraints, expressed by

$$\underline{s}^T \underline{s} = \underline{a}^T \underline{a} = 1, \quad \underline{s}^T \underline{a} = 0 \quad (3)$$

be satisfied (T denotes the transpose). Differentiating (3) with respect to time also yields

$$\underline{s}^T \dot{\underline{s}} = \underline{a}^T \dot{\underline{a}} = 0, \quad \underline{s}^T \dot{\underline{a}} + \underline{a}^T \dot{\underline{s}} = 0 \quad (4)$$

which involve three independent constraints on velocities. Substitution of (2) in (4) leads to the following useful kinematic properties for the Jacobians  $J_s, J_a$ , [5]

$$i) \text{rank}(J_s) = \text{rank}(J_a) = 2, \quad \forall \underline{q}, \quad (5)$$

$$ii) N(J_s^T) = \text{span}(\underline{s}), \quad N(J_a^T) = \text{span}(\underline{a}), \quad \forall \underline{q}, \quad (6)$$

$$iii) \text{given } \underline{x}, \underline{y} \in \mathbb{R}^3, \quad J_s^T \underline{x} + J_a^T \underline{y} = \underline{0} \text{ if } \underline{x} \in \text{span}(\underline{s}), \underline{y} \in \text{span}(\underline{a}); \quad (7)$$

the other orientation singularities of iii) are of no interest [5]. N is null space.

#### THE SOLUTION ALGORITHMS

The inverse kinematic problem is reconceived as a dynamic one in order to achieve a general approach which only involves the computation of direct kinematics (1), [6]. Denoting by  $\hat{\underline{q}}$  a solution of (1) relative to the assigned Cartesian vectors  $\hat{\underline{p}}, \hat{\underline{s}}, \hat{\underline{a}}$ , the following errors can be defined between the above vectors and the corresponding ones obtained from the algorithm state variables  $\underline{q}$ ,

$$\underline{e}_p = \hat{\underline{p}} - \underline{f}_p(\underline{q}), \quad \underline{e}_s = \hat{\underline{s}} - \underline{f}_s(\underline{q}), \quad \underline{e}_a = \hat{\underline{a}} - \underline{f}_a(\underline{q}) \quad (8)$$

In order to assure the convergence of  $\underline{q}$  to  $\hat{\underline{q}}$ , errors dynamics are involved, i.e., via (2),

$$\dot{\underline{e}}_p = \dot{\hat{\underline{p}}} - J_p \dot{\underline{q}}, \quad \dot{\underline{e}}_s = \dot{\hat{\underline{s}}} - J_s \dot{\underline{q}}, \quad \dot{\underline{e}}_a = \dot{\hat{\underline{a}}} - J_a \dot{\underline{q}}. \quad (9)$$

The point then is to relate  $\dot{\underline{q}}$  to  $\underline{e}_p, \underline{e}_s, \underline{e}_a$  so as to guarantee that such errors go asymptotically to zero, and consequently  $\underline{q}$  to  $\hat{\underline{q}}$ . Such a general scheme is showed in Fig. 4, [5].

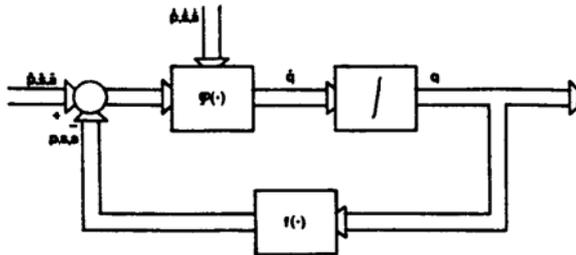


Fig. 4. General coordinate transformation scheme.  $\underline{p}' = \underline{p} - d_6 \underline{a}$ , (10)

in the following  $\underline{p}'$  will be assumed as position vector and indicated by  $\underline{p}$ , without loss of generality. In addition, depending upon the particular structure, it follows that the position vector  $\underline{p}$  is only dependent on:

However, if one is tempted to find a general convergent algorithm corresponding to the above scheme, eventual simplification deriving from each particular kinematic structure cannot be accomplished. To be more specific, with reference to Fig. 2, the end effector position vector  $\underline{p}$  and the approach unit vector  $\underline{a}$  are always independent of the last rotation  $\theta_6$ . Hence, as

- a) the first three joint variables  $(q_1, q_2, q_3)$ ,
- b) the first four joint variables  $(q_1, q_2, q_3, \theta_4)$ ,
- c) the first five joint variables  $(q_1, q_2, q_3, \theta_4, \theta_5)$ .

respectively. This issue suggests the realization of two stage algorithms which can be obtained by opportunely partitioning the vector  $\underline{q}$ , while taking into account the kinematic constraints of the different structures. So far in the following the conversion algorithms, relative to the three cases of Fig. 2, will be separately treated.

a) Concurrent axes.

In this case the position of the end effector does not depend on its orientation (Fig. 2a), which clearly simplifies the inverse kinematic problem. Indeed for such structure closed form solutions do exist [1], [7], [8], but they suffer from singularities and are based on geometric intuition. For this structure, the vector  $\underline{q}$  in (1) can be partitioned as

$$\underline{q}^T = (\underline{q}_p \quad \underline{q}_h), \quad \underline{q}_p^T = (q_1 \quad q_2 \quad q_3), \quad \underline{q}_h^T = (\theta_4 \quad \theta_5 \quad \theta_6), \quad (11)$$

where  $\underline{q}_p$  are the joint coordinates which determine the position of the "wrist", and  $\underline{q}_h$  are the other joint coordinates which, together with the previous ones, specify the orientation of the "hand". Under this assumption (1), (2) become in this case

$$\underline{p} = \underline{f}_p(\underline{q}_p) \quad (12)$$

$$\dot{\underline{p}} = J_p(\underline{q}_p)\dot{\underline{q}}_p \quad (13)$$

and

$$\underline{s} = \underline{f}_s(\underline{q}), \quad \underline{a} = \underline{f}_a(\underline{q}) \quad (14)$$

$$\dot{\underline{s}} = J_{sp}(\underline{q})\dot{\underline{q}}_p + J_s(\underline{q})\dot{\underline{q}}_h, \quad \dot{\underline{a}} = J_{ap}(\underline{q})\dot{\underline{q}}_p + J_a(\underline{q})\dot{\underline{q}}_h, \quad (15)$$

for the wrist and the hand respectively; in (13)  $J_p$  is the wrist Jacobian  $\partial \underline{f}_p / \partial \underline{q}_p$ , while in (15)  $J_{sp}$ ,  $J_{ap}$  and  $J_s$ ,  $J_a$  are the hand Jacobians  $\partial \underline{f}_s / \partial \underline{q}_p$ ,  $\partial \underline{f}_a / \partial \underline{q}_p$  and  $\partial \underline{f}_s / \partial \underline{q}_h$ ,  $\partial \underline{f}_a / \partial \underline{q}_h$  respectively. Such a partitioning suggests a two stage algorithm; specifying (9) gives

$$\dot{\underline{e}}_p = \dot{\underline{p}} - J_p \dot{\underline{q}}_p \quad (16)$$

$$\begin{bmatrix} \dot{\underline{e}}_s \\ \dot{\underline{e}}_a \end{bmatrix} = \begin{bmatrix} \dot{\underline{s}} \\ \dot{\underline{a}} \end{bmatrix} - \begin{bmatrix} J_{sp} \\ J_{ap} \end{bmatrix} \dot{\underline{q}}_p - \begin{bmatrix} J_s \\ J_a \end{bmatrix} \dot{\underline{q}}_h. \quad (17)$$

Assuming first

$$\dot{\underline{q}}_p = \gamma_p J_p^T \underline{e}_p, \quad \gamma_p = \alpha_p + \frac{\underline{e}_p^T \dot{\underline{p}}}{\underline{e}_p^T J_p J_p^T \underline{e}_p}^{-1}, \quad \alpha_p > 0 \quad (18)$$

guarantees that the wrist position error goes to zero. This issue can be recognized by considering the error Lyapunov function  $V = \frac{1}{2} \underline{e}_p^T \underline{e}_p$  and verifying that its derivative is negative definite in virtue of (18), [5].

Progressing then in a similar fashion for the hand, assuming as error Lyapunov function  $V_h = \frac{1}{2} (\underline{e}_s^T \underline{e}_s + \underline{e}_a^T \underline{e}_a)$ , and accounting for the kinematic properties (5)-(7) lead to the following choice

$$\dot{\underline{q}}_h = \gamma_h \text{sgn}(J_s^T \underline{s} + J_a^T \underline{a}), \quad \gamma_h = \frac{|\dot{\underline{s}}|_{\max} + |\dot{\underline{a}}|_{\max} + k_p}{\Lambda(J_{sp}) + \Lambda(J_{ap})}, \quad (19)$$

where  $\Lambda(A)$  denotes the maximum eigenvalue of matrix  $A$ , and  $(\text{sgn } \underline{w})^T = (\text{sgn } w_1 \dots \text{sgn } w_r)$ ,  $\underline{w} \in R^r$ ; such a choice guarantees that  $\underline{e}_s \rightarrow 0$  and  $\underline{e}_a \rightarrow 0$ . The reader is referred to [5] for further details.

b) Two-by-two intersecting axes.

In this case the position vector  $\underline{p}$  is not uniquely determined through the first three degrees of freedom of the structure, since also  $\theta_4$  concurs to determine it (Fig. 2b). By four d.o.f. it is allowed to position the vector  $\underline{p}$  by  $=1$  values of  $q_1, q_2, q_3, \theta_4$ . In order to obtain a unique solution the following mechanical constraint must be considered

$$\underline{\hat{a}}^T \underline{z}_4 = \cos \alpha_5, \quad (20)$$

where  $\underline{\hat{a}}$  is given in the Cartesian space, and  $\underline{z}_4$  depends on  $q_1, q_2, q_3, \theta_4$ . So far, partitioning with respect to the first four joint variables allows the definition of the following errors:

$$\begin{aligned} \underline{e}_p &= \underline{\hat{p}} - \underline{f}_p(q_p) \\ \underline{e}_{z_4} &= \cos \alpha_5 - \underline{\hat{a}}^T \underline{z}_4 \end{aligned} \quad \underline{q}_p^T = (q_1 \ q_2 \ q_3 \ \theta_4) \quad (21)$$

Differentiating with respect to time yields

$$\begin{pmatrix} \dot{\underline{e}}_p \\ \dot{\underline{e}}_{z_4} \end{pmatrix} = \begin{pmatrix} \dot{\underline{\hat{p}}} \\ -\dot{\underline{\hat{a}}}^T \underline{z}_4 \end{pmatrix} - \begin{pmatrix} J_p \\ \underline{\hat{a}}^T J_{z_4} \end{pmatrix} \dot{\underline{q}}_p, \quad (22)$$

where  $J_p$  is the (3x4) matrix  $\partial \underline{f}_p / \partial \underline{q}_p$  and  $J_{z_4}$  is the (3x4) matrix  $\partial \underline{z}_4 / \partial \underline{q}_p$ . The matrix premultiplied to  $\dot{\underline{q}}_p$  has rank 4 almost everywhere. The following choice assures the convergence of the algorithm

$$\begin{aligned} \dot{\underline{q}}_p &= \gamma_p J_p^T \underline{e}_p + \gamma_{z_4} \underline{\text{sgn}}(J_{z_4}^T \underline{\hat{a}} \underline{e}_{z_4}), \quad \gamma_p = \alpha_p + \frac{\underline{e}_p^T \underline{\hat{p}} (\underline{e}_p^T J_p J_p^T \underline{e}_p)^{-1}}{\|\underline{\hat{p}}\|}, \quad \alpha_p > 0 \\ \gamma_{z_4} &= \|\underline{\hat{a}}\|_{\max} \end{aligned} \quad (23)$$

In this way the first stage of the algorithm guarantees that  $\underline{\hat{a}}$  and  $\underline{a}$  lie on the same cone of angle  $\alpha_5$  and vertex  $P'$ , but they do not necessarily have the same direction, as  $\theta_5$  has not been determined yet. Then, in order to align  $\underline{\hat{a}}$  with  $\underline{a}$  and  $\underline{\hat{s}}$  with  $\underline{s}$ , the second stage of the algorithm must be able to determine  $\theta_5$  and  $\theta_6$ . Progressing as for the previous case gives

$$\begin{aligned} \underline{e}_s &= \underline{\hat{s}} - \underline{f}_s(q) \\ \underline{e}_a &= \underline{\hat{a}} - \underline{f}_a(q_p, \theta_5) \end{aligned} \quad (24)$$

and

$$\begin{pmatrix} \dot{\underline{e}}_s \\ \dot{\underline{e}}_a \end{pmatrix} = \begin{pmatrix} \dot{\underline{\hat{s}}} \\ \dot{\underline{\hat{a}}} \end{pmatrix} - \begin{pmatrix} J_{sp} \\ J_{ap} \end{pmatrix} \dot{\underline{q}}_p - \begin{pmatrix} J_s \\ J_a \end{pmatrix} \dot{\theta}_h, \quad \underline{q}_h^T = (\theta_5 \ \theta_6), \quad (25)$$

where  $J_{sp}$ ,  $J_{ap}$ ,  $J_s$ ,  $J_a$  are substantially identical to the Jacobians in (17) except for their dimensions. The convergence is then assured by

$$\dot{\theta}_h = \gamma_h \underline{\text{sgn}}(J_s^T \underline{\hat{s}} + J_a^T \underline{\hat{a}}), \quad \gamma_h = \|\underline{\hat{s}}\|_{\max} + \|\underline{\hat{a}}\|_{\max} + \|\dot{\underline{q}}_p\|_{\max} (|A(J_{sp})| + |A(J_{ap})|). \quad (26)$$

Moreover, since the algorithm is guaranteed to converge, the realistic case  $\underline{q}(0) = \underline{\hat{q}}(0)$  avoids any problem of indeterminacy concerned with  $\cos \alpha_5 = \cos(-\alpha_5)$ .

c) Nonintersecting axes.

In this last case  $\theta_4$  and  $\theta_5$ , together with  $(q_1, q_2, q_3)$ , concur to determine the position of the vector  $\underline{p}$  (Fig. 2c). By 5 d.o.f. it is allowed to position the vector  $\underline{p}$  by  $=2$  values of  $q_1, q_2, q_3, \theta_4, \theta_5$ . As a unique solution is desired, one must account for the two constraints expressed by

$$\frac{\hat{a}^T z_4}{\hat{a}^T a} = \cos \alpha_5 \quad (27)$$

$$\frac{\hat{a}^T a}{\hat{a}^T a} = 1. \quad (28)$$

The former concerns with the particular mechanical structure, whereas the latter is introduced since, by 5 d.o.f., position and direction are uniquely determined. As a consequence the errors are defined through

$$\begin{aligned} e_p &= \hat{p} - f_p(q_p) \\ e_{z_4} &= \cos \alpha_5 - \frac{\hat{a}^T f_{z_4}(q_1, q_2, q_3, \theta_4)}{\hat{a}^T z_4} \\ e_a &= 1 - \frac{\hat{a}^T f_a(q_p)}{\hat{a}^T a} \end{aligned} \quad (29)$$

and their derivatives are

$$\begin{bmatrix} \dot{e}_p \\ \dot{e}_{z_4} \\ \dot{e}_a \end{bmatrix} = \begin{bmatrix} \dot{\hat{p}} \\ -\dot{\hat{a}}^T z_4 \\ -\dot{\hat{a}}^T a \end{bmatrix} - \begin{bmatrix} J_{p_p} \\ \hat{a}^T J_{z_4} \\ \hat{a}^T J_a \end{bmatrix} \dot{q}_p, \quad q_p^T = (q_1 \ q_2 \ q_3 \ \theta_4 \ \theta_5) \quad (30)$$

with analogous meanings of the Jacobians in the matrix premultiplied to  $\dot{q}_p$  which has rank 5 almost everywhere. The choice for  $\dot{q}_p$  is then

$$\begin{aligned} \dot{q}_p &= \gamma_p J_{p-p}^T e_p + \gamma_{z_4} \frac{\text{sgn}(J_{z_4}^T \hat{a} e_{z_4})}{|J_{z_4}^T \hat{a} e_{z_4}|} + \gamma_a \frac{\text{sgn}(J_a^T \hat{a} e_a)}{|J_a^T \hat{a} e_a|}, \quad \gamma_p = \alpha_p + \frac{e_p^T \dot{\hat{p}} (e_p^T J_{p-p}^T J_{p-p}^T e_p)^{-1}}{|e_p|}, \quad \alpha_p > 0 \\ \gamma_{z_4} &= \gamma_a = \frac{|\dot{\hat{a}}|}{|\hat{a}|} \max. \end{aligned} \quad (31)$$

The second stage of the algorithm, now, is only required to align  $s$  with  $\hat{s}$ . As by means of (31)  $a = \hat{a}$ , only  $\theta_6$  performs this task. The error and its derivative are respectively

$$e_s = \hat{s} - f_s(q_p, \theta_6) \quad (32)$$

$$\dot{e}_s = \dot{\hat{s}} - J_{s p} \dot{q}_p - J_s \dot{\theta}_6. \quad (33)$$

The choice

$$\dot{\theta}_6 = \gamma_s \frac{\text{sgn}(J_s^T \hat{s} e_s)}{|J_s^T \hat{s} e_s|}, \quad \gamma_s = \frac{|\dot{\hat{s}}|}{|\hat{s}|} \max + \frac{|\dot{q}_p|}{|q_p|} \max |J_{s p}| \quad (34)$$

assures the convergence.

In sum, once the kinematic structure of the manipulator is known, one can select the right algorithm to perform the coordinate transformation of (1). Moreover if one is interested in obtaining a solution for (2) as well, that is  $\dot{q}$  corresponding to  $\dot{\hat{p}}, \dot{\hat{s}}, \dot{\hat{a}}$ , as required by typical advanced control techniques [6], [9], [10], the sgn laws in (19), (23), (26), (31), (34) can be replaced by proportional laws, which avoid the generation of undesirable joint velocities rich of harmonics; reasonably small tracking errors occur, but steady-state errors are identically zero [5].

Finally, an appealing feature of the proposed algorithms is that the number of additions, floating point multiplies and transcendental calls required is contained, allowing for a solution sample rate that can be conveniently increased up to the same values of joint servos sample rate; this issue suggests digital implementation by means of a single dedicated microprocessor system [11].

#### EXAMPLES

In order to show the effectiveness of the proposed coordinate transformation algorithms two arms have been chosen: the PUMA 560 (Fig. 5) which has three intersecting revolute joint axes at the end effector, and whose direct kinematics can be

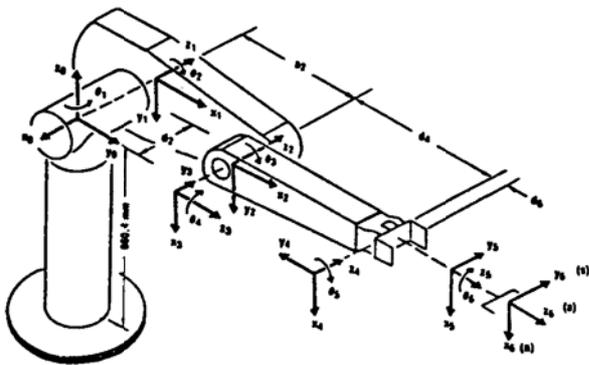


Fig. 5. The PUMA 560 arm.

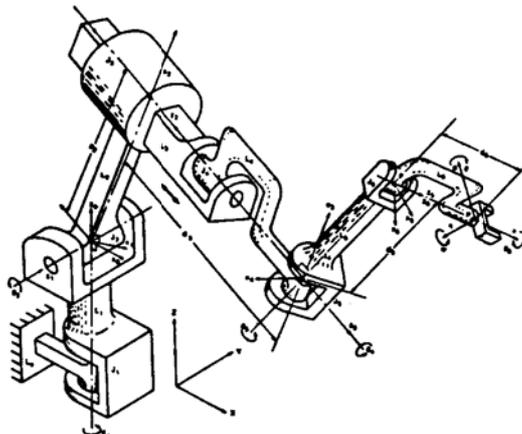


Fig. 6. The prototype arm of [2].

found in [7], and the prototype (Fig. 6) which has two-by-two intersecting revolute joint axes at the end effector, and whose direct kinematics is reported in [2]. The assigned trajectory to track in the Cartesian space consists of a 40 cm. straight line, along with null Euler angles excursions; it should be noted that this does not imply that some joints must not move. Trapezoidal velocity profiles have been imposed with two couples of values of maximum velocities and time intervals  $t_f$ : ( $\|\dot{p}\|_{\max} = 0.6\text{m/s}$ ,  $t_f = 1\text{s}$ ), ( $\|\dot{p}\|_{\max} = 0.06\text{m/s}$ ,  $t_f = 10\text{s}$ ); the slower trajectory is typical of those tasks where the robot is required to work, whereas the faster one refers to material handling tasks.

Proportional type laws have been chosen, gaining the inherent advantage to directly generate joint velocities; finite tracking errors are then expected. However, since the simulated system is a sample data system with a solution sample period of 2ms, finite tracking errors would have occurred even with sgn type laws. The feedback gains  $\gamma$  have always been set up at the inverse of the sample period, that is 500. Fig. 7 shows tracking position errors for the two trajectories, respectively for the two arms, while in Fig. 8 reported are the maximum tracking orientation errors,  $e_a$  (or  $e_s$ ), which are evaluated as  $\cos^{-1} \frac{a}{\hat{a}}$  (or  $\cos^{-1} \frac{s}{\hat{s}}$ ). Since the same initial conditions occur ( $\underline{q}(0) = \hat{\underline{q}}(0)$ ), tracking errors maintain in any case very small along the whole trajectory, whereas at steady-state they vanish in virtue of the closed loop structure of the developed conversion algorithms.

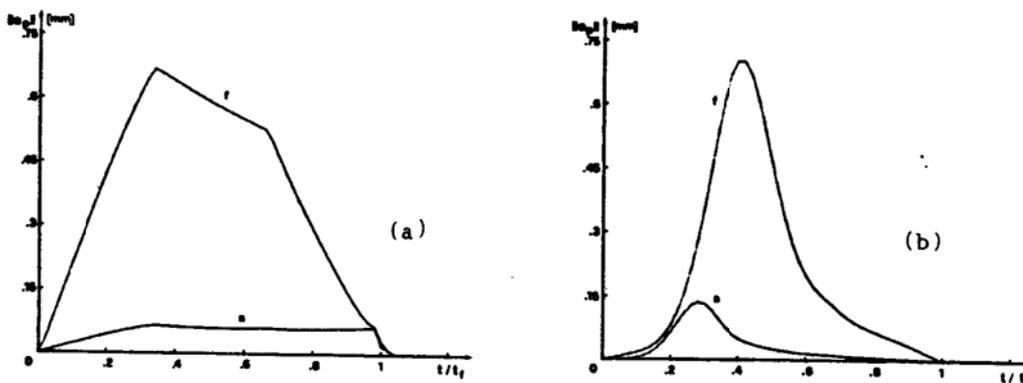


Fig. 7. Tracking position errors: a) PUMA 560, b) arm [2] (f=fast, s=slow).

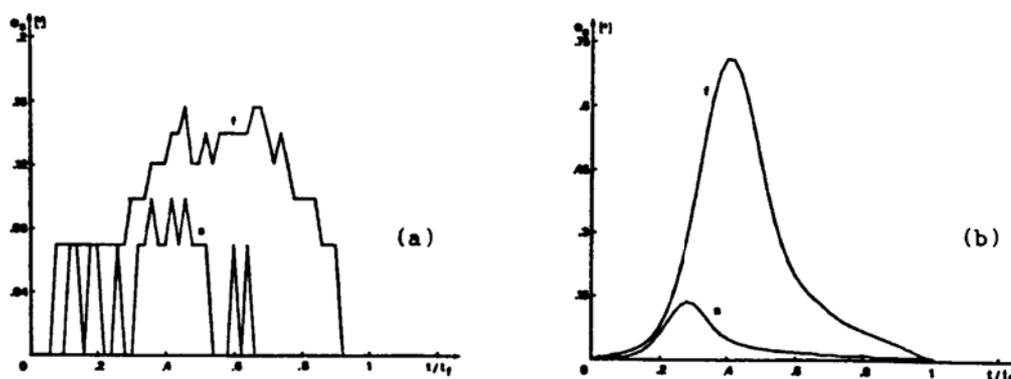


Fig. 8. Tracking orientation errors: a) PUMA 560, b) arm |2| (f=fast, s=slow).

#### CONCLUSIONS

This paper has presented a general solution algorithm for the inverse kinematic problem. Three basic mechanical structures have been considered differing from one another as regards the end effector axes configuration. Going back from the end effector terminal point through the structure, the algorithm has been partitioned at an opportune point whose position, dependent on a reduced number of joint variables, can be expressed in terms of the Cartesian coordinates of the given task. In this way two-stage algorithms have been obtained. As the numerical algorithms implemented always provide, at each step, solutions adjacent to the preceding ones, uniqueness of solutions is assured. Finally, since the computational burden is contained, solution sample rates equal to those of joint servos are allowed, thus avoiding further interpolations. Future developments are devoted to extend such algorithms to kinematically redundant arms with potential benefits over current designs.

#### REFERENCES

- [1] R.P. Paul, Robot Manipulators: Mathematics, Programming, and Control. Cambridge, MA: MIT Press, 1982.
- [2] V.J. Lumelsky, "Iterative coordinate transformation procedure for one class of robots," IEEE Trans. Syst., Man, Cyber., vol. 14, pp. 500-505, May/June 1984.
- [3] J. Denavit, R.S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," ASME J. Appl. Mech., June 1955.
- [4] J.Y.S. Luh, "An anatomy of industrial robots and their controls," IEEE Trans. Automat. Control, vol. 28, pp. 133-153, Feb. 1983.
- [5] A. Balestrino, G. De Maria, L. Sciavicco, B. Siciliano, "A novel approach to coordinate transformation for robotic manipulators," subm. IEEE Tr. SMC, 1985.
- [6] A. Balestrino, G. De Maria, L. Sciavicco, "Robust control of robotic manipulators," Proc. 9th IFAC World Congress, Budapest, Hungary, July 1984.
- [7] C.S.G. Lee, "Robot arm kinematics, dynamics, and control," Comput., vol. 15, no. 12, pp. 62-80, Dec. 1982.
- [8] R.P. Paul, B. Shimano, G.E. Mayer, "Kinematic control equations for simple manipulators," IEEE Trans. Syst., Man, Cybern., vol. 11, pp. 449-455, June 1981.
- [9] J.Y.S. Luh, M.W. Walker, R.P.C. Paul, "Resolved acceleration control of mechanical manipulators," IEEE Trans. Aut. Cont., vol. 25, pp. 468-474, June 1980.
- [10] A. Balestrino, G. De Maria, L. Sciavicco, "An adaptive model following control for robotic manipulators," ASME J.DSMC, vol. 105, pp. 143-151, Sept. 1983.
- [11] G. De Maria, R. Marino, "A discrete algorithm for solving the inverse kinematic problem of robotic manipulators," 85' ICAR, Tokyo, Japan, Sept. 1985.