

Solving the Inverse Kinematic Problem for Robotic Manipulators

L. Sciavicco and B. Siciliano

Universita' di Napoli, Dipartimento di Informatica e Sistemistica, Napoli, Italy

ABSTRACT

Based on a dynamic approach, a general solution algorithm for the inverse kinematic problem for robotic manipulators is presented. It requires only the computation of direct kinematics. Two-stage algorithms are then derived for three basic kinematical structures in order to comply with the mechanical constraints of each structure. Applicability of the algorithm to redundant manipulators with obstacle collision avoidance and limited joint range availability is finally shown.

INTRODUCTION

The crucial point in advanced control of robotic manipulators is the capability of transforming the task space coordinates into the configuration space coordinates, that is solving the Inverse Kinematic Problem. As a matter of fact control is implemented in the joint space, identified by the n -dimensional joint vector q , whereas motion is specified in the Cartesian space, by means of an m -dimensional Cartesian vector x . The direct kinematic problem allows to specify in a straightforward manner [1] the relationship between the joint variables and the Cartesian variables as

$$\underline{x} = \underline{f}(q), \quad (1)$$

where \underline{f} is a continuous nonlinear function, whose structure and parameters are known. It is unique, while the inverse transformation

$$q = \underline{f}^{-1}(x) \quad (2)$$

is not unique and, because of complexity of (1), is hard to be expressed analytically.

The most common approach for solving the inverse kinematic problem is certainly to obtain a closed-form solution to (2), [2]. Only manipulators with a spherical wrist, however, allow closed-form solutions. The problem becomes more critical for kinematically redundant manipulators, for which the number of degrees of freedom exceeds the six coordinates which are usually required to specify the position and the orientation of the end effector in the Cartesian space.

An iterative technique based on a nonlinear optimization algorithm for solving (2) has been proposed in [3]. Though it seems to be quite general, being applicable to any kinematical structure, it involves a great amount of computation to converge to the desired solution, which makes it impractical for tracking control purposes.

The approach illustrated in this paper is based on a dynamic formulation of the problem first proposed in [4], and later also in [5]. The main advantage of the method is that it only makes use of direct kinematics (1). Besides uniqueness of the solution is assured, computation time is drastically reduced, joint velocities are automatically generated and occurrence of kinematic singularities may not represent a drawback. Even if

the resultant algorithm is general and manipulator-independent, an even smaller number of computations are required and better results are achieved if the algorithm is customized to the particular kinematical structure. Furthermore, the same dynamic concept which is at the basis of the method allows the solution of the inverse kinematic problem for redundant manipulators in presence of obstacles in the workspace and/or with mechanical constraints imposed of the joint variables.

THE GENERAL SOLUTION ALGORITHM

The inverse kinematic problem is conceived as a dynamical one in order to get a general solution algorithm which requires only the computation of direct kinematics (1). Let $\hat{q}(t)$ be a solution of (1) relative to a given Cartesian trajectory $\hat{x}(t)$. The following error vector $e(t)$ can be defined between the Cartesian trajectory and the corresponding one obtained from the algorithm state variables $q(t)$,

$$e(t) = \hat{x}(t) - x(t). \quad (3)$$

Recall that Cartesian velocities are related to joint velocities through the Jacobian matrix $J(q)$ associated with the relationship

$$\dot{x}(t) = J(q)\dot{q}(t). \quad (4)$$

In order to assure the convergence of $q(t)$ to $\hat{q}(t)$, error dynamics is involved, i.e. via (4) (dropping the time dependence),

$$\dot{e} = \dot{\hat{x}} - J(q)\dot{q}. \quad (5)$$

With the choice

$$\dot{q} = \gamma J^T(q)e, \quad \gamma = \alpha + (e^T \dot{\hat{x}})(e^T J J^T e)^{-1}, \quad \alpha > 0 \quad (6)$$

the dynamic system of fig. 1 assures that $e \rightarrow 0$, and then $q(t) \rightarrow \hat{q}(t)$. This issue can be recognized by considering the error Lyapunov function $V = .5e^T e$ and verifying that its derivative is negative definite in virtue of (6), [4], [5]. The choice of α in (6) determines the convergence rate of the closed loop system of fig. 1. Starting with the same initial conditions $q(0) = \hat{q}(0)$ will always guarantee good tracking accuracy, avoiding thus the solution nonuniqueness drawback concerned with those techniques which provide joint trajectories as result of an interpolation between a finite set of via points [6].

As it had been anticipated, only direct kinematics is to be computed, which drastically reduces the computational burden. In fact only one iteration per cycle is to be done as far as the digital implementation of the algorithm for trajectory tracking is concerned. The implementation of the algorithm on a single dedicated microprocessor system has been realized in lab and is described in [7].

It should be emphasized also that the dynamical system of fig. 1 will produce joint velocities $\dot{q}(t)$ at no additional cost, which is very useful for advanced control, see model reference adaptive control [8] for instance.

Finally it might be noted that, since no inversion is required (J^{-1}, J^{-1}), the algorithm may provide solutions even in case of kinematic singularities, on condition that the trajectory in the task space is properly planned. To be more specific, no problem will arise if the singularity belongs to the trajectory, whereas in case of a trajectory passing in the proximity of a singularity, indeed, high velocities are mechanically involved; in this case

the convergence of the algorithm is likely to be derated.

NONREDUNDANT MANIPULATORS

A robot task is naturally specified in terms of the end effector Cartesian vector $\underline{x}^1 = (p_x, p_y, p_z, \alpha, \beta, \gamma)$ with respect to the base frame; p_i 's are the components of the end effector position vector \underline{p} , and α, β, γ are the Euler angles (or roll, pitch and yaw angles) which define its orientation. In order to obtain a unique specification of the orientation, however, a unit approach vector \underline{a} and a unit sliding vector \underline{s} can be adopted; the unit normal vector \underline{n} which usually completes the frame at the end effector is redundant since $\underline{n} = \underline{s} \times \underline{a}$. The above vectors can be easily determined from the Euler angles, [9]. In view of the preceding, for any robot kinematical structure, (1) becomes

$$\underline{p} = \underline{f}_p(\underline{q}) \quad \underline{s} = \underline{f}_s(\underline{q}) \quad \underline{a} = \underline{f}_a(\underline{q}) \quad (7)$$

subjected to the constraints

$$\underline{s}^T \underline{s} = \underline{a}^T \underline{a} = 1 \quad \underline{s}^T \underline{a} = 0. \quad (8)$$

Similarly (4) gives

$$\dot{\underline{p}} = J_p(\underline{q}) \dot{\underline{q}} \quad \dot{\underline{s}} = J_s(\underline{q}) \dot{\underline{q}} \quad \dot{\underline{a}} = J_a(\underline{q}) \dot{\underline{q}} \quad (9)$$

subjected to

$$\underline{s}^T \dot{\underline{s}} = \underline{a}^T \dot{\underline{a}} = 0 \quad \underline{s}^T \dot{\underline{a}} + \underline{a}^T \dot{\underline{s}} = 0. \quad (10)$$

Nonredundant manipulators require six degrees of freedom to identify uniquely the position and the orientation of the end effector. Typical kinematical structures have three revolute joints ($\theta_4, \theta_5, \theta_6$) at the end effector, whereas the first three joints (q_1, q_2, q_3) are either all revolute, such as the PUMA arm [10], or two revolute and one prismatic, such as the JPL arm [11]. As far as the last three joints, three basic configurations are illustrated in fig. 2. The case a) is of particular interest since it is possible to decouple the position of the end effector from its orientation (spherical wrist). The cases b) and c) may also occur in practical robot design. The three structures can be conveniently characterized through the following constraints on the geometric parameters of the last three joints. More specifically, the lengths a_n and the distances d_n , [1], are respectively in the three cases:

- a) parallel 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) nonconverging axes: $a_4 \neq 0, a_5 \neq 0$ (fig. 2c).

With reference to fig. 2, the position vector \underline{p} and the approach unit vector \underline{a} are always independent of the last rotation. Hence

$$\underline{p}' = \underline{p} - d_6 \underline{a} \quad (11)$$

can be assumed as position vector, and still indicated by \underline{p} without loss of generality. Furthermore the position vector \underline{p} depends on

- a) the first three joint variables (q_1, q_2, q_3),
- b) the first four joint variables (q_1, q_2, q_3, θ_4),
- c) the first five joint variables ($q_1, q_2, q_3, \theta_4, \theta_5$).

respectively in the three cases of fig. 2. As a consequence the vector of joint variables q in (1) can be partitioned as

$$q^T = (q_p^T \mid q_h^T), \quad (12)$$

where q_p are the joint variables which determine the position p , and q_h are the remaining joint variables which, together with q_p , determine the orientation s, a . Accordingly the general algorithm can be partitioned into two stages (first determine q_p , then q_h) as follows in the three cases of fig. 2, [12].

Spherical wrist

For this structure (fig. 2a) the vector q in (12) is partitioned into

$$q_p^T = (q_1 \ q_2 \ q_3) \quad q_h^T = (\theta_4 \ \theta_5 \ \theta_6). \quad (13)$$

The resulting two-stage inverse kinematic algorithm is

$$\dot{q}_p = \gamma_p J_p^T \dot{e}_p, \quad \gamma_p = \alpha_p + (\dot{e}_p^T \hat{e}) (\dot{e}_p^T J_p J_p^T \dot{e}_p)^{-1}, \quad \alpha_p > 0 \quad (14a)$$

$$\dot{q}_h = \gamma_h (J_s^T \hat{s} + J_a^T \hat{a}), \quad (14b)$$

$$\gamma_h > (\|\hat{s}\|_{\max} \|\hat{a}\|_{\max} \|\dot{q}_p\|_{\max}) (|\lambda(J_{sp})| + |\lambda(J_{ap})|) (|\lambda(J_s)| + |\lambda(J_a)|)^{-1}$$

where $J_p = \partial f_p / \partial q_p$, $J_{sp} = \partial f_s / \partial q_p$, $J_{ap} = \partial f_a / \partial q_p$, $J_s = \partial f_s / \partial q_h$, $J_a = \partial f_a / \partial q_h$; $\lambda(A)$ and $\lambda(A)$ denote the maximum and the minimum eigenvalue of matrix A respectively. Further details on the derivation of (14) can be found in [13].

Two-by-two intersecting axes

In this case also θ_4 concurs to determine the position vector p (fig. 2b). The partition of q in (12) gives

$$q_p^T = (q_1 \ q_2 \ q_3 \ \theta_4) \quad q_h = (\theta_5 \ \theta_6). \quad (15)$$

In order to obtain a unique solution for q_p , the following mechanical constraint is to be incorporated in the first stage:

$$\hat{a}^T z_4 = \cos \alpha_5, \quad (16)$$

where α_5 is the constant twist angle between the fifth and the sixth link, \hat{a} is given in the Cartesian space and z_4 is the unit vector along the fifth link axis, which is determined through q_p . The inverse kinematic algorithm results then, [14], [15],

$$\dot{q}_p = \gamma_p (J_p^T \dot{e}_p + J_{z_4}^T \hat{a} e_{z_4}), \quad \gamma_p > (\|\hat{a}\|_{\max} \|\dot{q}_p\|_{\max}) \left| \begin{matrix} J_p \\ \hat{a}^T J_{z_4} \end{matrix} \right|^{-1} \quad (17a)$$

$$\dot{q}_h = \gamma_h (J_s^T \hat{s} + J_a^T \hat{a}), \quad (17b)$$

$$\gamma_h > (\|\hat{s}\|_{\max} \|\hat{a}\|_{\max} \|\dot{q}_p\|_{\max}) (|\lambda(J_{sp}^T J_{sp} + J_{ap}^T J_{ap})|)^{-5} (|\lambda(J_s^T J_s + J_a^T J_a)|)^{-5}$$

where $e_{z_4} = \cos \alpha_5 - \hat{a}^T f_{z_4}(q_p)$, and $J_{z_4} = \partial f_{z_4} / \partial q_p$.

Nonconverging axes

In this last case five degrees of freedom determine the position of p (fig. 2c). The partition obviously results

$$\underline{q}_p^T = (q_1 \ q_2 \ q_3 \ \theta_4 \ \theta_5) \quad q_h = \theta_6 \quad (18)$$

For the same reason as above, the two following constraints must be added in order to get a unique solution for \underline{q}_p :

$$\hat{\underline{a}}^T \underline{z}_4 = \cos \alpha_5 \quad (19a)$$

$$\hat{\underline{a}}^T \underline{a} = 1. \quad (19b)$$

The resulting algorithm is, [12],

$$\dot{\underline{q}}_p = \gamma_p (J_p^T \underline{e}_p + J_{z4}^T \hat{\underline{a}} e_{z4} + J_a^T \hat{\underline{a}}), \quad \gamma_p > (\|\hat{\underline{e}}\|_{\max} + \|\hat{\underline{a}}\|_{\max}) \lambda \left| \hat{\underline{a}}^T J_{z4} \right|^{-1} \quad (20a)$$

$$\dot{q}_h = \gamma_s (J_s^T \hat{\underline{s}}), \quad \gamma_s > (\|\hat{\underline{s}}\|_{\max} + \|\dot{\underline{q}}_p\|_{\max}) \lambda (J_{sp}^T J_{sp})^{-.5} |\lambda (J_s^T J_s)|^{-.5} \quad (20b)$$

where it might be noted that, since \underline{a} is determined in the first stage, the second stage is only required to align \underline{s} with $\hat{\underline{s}}$ by means of θ_6 .

REDUNDANT MANIPULATORS

A manipulator is termed kinematically redundant if the number of degrees of freedom exceeds the number of task space coordinates. Redundancy can be conveniently exploited to solve the inverse kinematic problem with obstacle collision avoidance and/or limited joint range availability. In these cases solutions have been proposed, based on the use of the generalized inverse, [15]-[19]. It seems, however, that the amount of computation involved is still too large for real time control.

With reference to the general scheme of fig. 1, the two kinds of constraints can be successfully incorporated in the dynamic approach if the task space state vector in (1) is enlarged

Suppose first that, while the manipulator is tracking a desired trajectory in the task space, one or more links along its kinematical structure happen to be much too close to an obstacle in the workspace. Since the inverse kinematic algorithm (6) provides joint configurations which are adjacent to each other as the manipulator proceeds, one or more constraints can be introduced in order to avoid the collision with the obstacle.

More precisely, let \underline{c}_i ($i=1, \dots, n$) indicate the position vectors of those points of the obstacle which are closest to each link l_i of the manipulator; a point at minimum distance from the obstacle on each link is automatically individuated and let \underline{p}_i indicate the corresponding position vector. Both vectors are defined with respect to the same base frame, see fig. 3 for a planar example. If the distance between the two points $\|\underline{d}_i\|$, where $\underline{d}_i = \underline{p}_i - \underline{c}_i$, is less than a threshold distance \hat{d} , there is a danger of a collision, and the joint velocities which represent the control inputs to the system of fig. 1 should be modified accordingly to the new situation. This can be accomplished as follows. Define the errors

$$e_{dj} = .5(\hat{d}^2 - \underline{d}_i^T \underline{d}_i), \quad j = 1, \dots, k, \quad (21)$$

where k is the number of active constraints. Differentiating (21) with respect to time gives

$$\dot{e}_{dj} = - J_{d1j} \dot{\underline{q}} \quad (22)$$

where

$$J_{d_i} = \underline{d}_i^T J_{p_i}, \quad (23)$$

being J_{p_i} the Jacobian matrix associated with vector p_i . Then the control (6) can be modified into

$$\dot{\underline{q}} = G_d(J^T \underline{e} + J_d^T \underline{e}_d) \quad (24)$$

which still guarantees that $\underline{x} \rightarrow \hat{\underline{x}}$, but also assures that $\underline{d}_i^T \underline{d}_i \rightarrow \hat{d}^2$; G_d is a positive definite diagonal matrix.

In this way the motion of those degrees of freedom which influence the motion of the point p_i is braked preventing the link l_i from approaching the obstacle. As a matter of fact, a link which is candidate to a collision, in virtue of (24), is forced to move tangentially around the imaginary sphere of center in \underline{c}_i and radius \hat{d} .

Although (24) is the basis of the obstacle avoidance scheme proposed here, proper decision making in charge of a higher control level is equally important to successful operation of the algorithm. The threshold distance \hat{d} , which should include the thickness of manipulator links, must be programmed accordingly to the sample rate at which p_i and \underline{c}_i are updated so as to get a security gap. In addition, in order not to introduce a discontinuity, the feedback gains of G_d should be tapered as a function of distance.

It must be remarked, however, that the computational burden of the pure inverse kinematic algorithm remains contained as it may be checked in (24).

Conceptually similar is the inclusion of mechanical constraints on joint variables into the inverse kinematic scheme (6). If the joint variables q_i are constrained between two extremal values $q_{i\min}$ and $q_{i\max}$, i.e.

$$q_{i\min} \leq q_i \leq q_{i\max}, \quad i = 1, \dots, n, \quad (25)$$

it is possible to define again a threshold \hat{d}' and the errors

$$e_{q_j} = \hat{d}' - d_{q_j}, \quad j = 1, \dots, r, \quad (26)$$

where either $d_{q_j} = q_j - q_{i\min}$ or $d_{q_j} = q_{i\max} - q_j$, depending on which limit is involved. Progressing as above yields the modified control of type (6)

$$\dot{\underline{q}} = G_q(J^T \underline{e} \pm \underline{e}_q) \quad (27)$$

which assures that $\underline{x} \rightarrow \hat{\underline{x}}$ and $d_{q_j} \rightarrow \hat{d}'$, the sign - applying for $q_{i\min}$ and the sign + for $q_{i\max}$; G_q is a positive definite diagonal matrix. In this way the joint variables are prevented from approaching either of the two limits. Remarks on the adequate choice of \hat{d}' and G_q as for the case of obstacle avoidance are in order also in this case.

Last but not least, it must be emphasized that in order to comply with all the constraints given by (21) and (26) it must be checked that

$$k + r \leq n - m, \quad (28)$$

In other words the enlargement of the error space can be made up to cover the degrees of redundancy available which are in number of $n - m$.

CONCLUDING REMARKS

This paper presented a general solution algorithm for the inverse kinematic problem for robotic manipulators. For each kinematically

nonredundant structure the algorithm is conveniently partitioned into two stages so as to better cope with the kinematical structure.

Simulation studies dedicated to several robotic manipulators can be found in [7], [12]-[15] and have not been fully reported here due to lack of space. Tracking errors have always resulted to be contained, on the average of 1 mm (position) and .1° (orientation) for typical velocities of 1m/sec and 90°/sec. Steady-state errors are practically null due to the closed loop structure of the dynamic system of fig. 1; this issue proves that a precise solution can be provided by the algorithm in all those cases when computation time is not the main concern and the purpose is just to get the set of joint variables corresponding to a given configuration of the end effector. It must be emphasized also that uniqueness of the solution is always assured, and a kinematic singularity along the given trajectory does not involve any large errors, as proved for instance by simulation results derived in [14].

Finally it has been shown how the same dynamic approach can be successfully adopted for solving the inverse kinematic problem for redundant manipulators in presence of obstacles in the workspace and/or with mechanical constraints imposed on the joint variables

REFERENCES

- [1] J. Denavit and R.S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," J. Appl. Mech., June 1955.
- [2] R.P. Paul, B. Shimano and G.E. Meyer, "Kinematic control equations for simple manipulators," IEEE Trans. Syst., Man, Cybern., vol. 11, no. 6, June 1981.
- [3] A.A. Goldenberg, B. Benhabib and R.G. Fenton, "A complete generalized solution to the inverse kinematics of robots," IEEE J. Rob. Autom., vol. 1, no. 1, March 1985.
- [4] A. Balestrino, G. De Maria and L. Sciavicco, "Robust control of robotic manipulators," Proc. 9th IFAC World Congress, Budapest, Hungary, July 1984.
- [5] W.A. Wolovich and H. Elliott, "A computational technique for inverse kinematics," Proc. 23rd CDC, Las Vegas, NV, Dec. 1984.
- [6] R.P. Paul, Robot Manipulators: Mathematics, Programming, and Control. Cambridge, MA, MIT Press, 1981.
- [7] G. De Maria and R. Marino, "A discrete algorithm for solving the inverse kinematic problem for robotic manipulators," Proc. '85 ICAR, Tokyo, Japan, Sept. 1985.
- [8] A. Balestrino, G. De Maria and L. Sciavicco, "An adaptive model following control for robotic manipulators," ASME J. Dynamic Syst., Meas., Contr., vol. 105, Sept. 1983.
- [9] J.Y.S. Luh, "An anatomy of industrial robots and their controls," IEEE Trans. Autom. Contr., vol. 28, no. 2, Feb. 1983.
- [10] C.S.G. Lee, "Robot arm kinematics, dynamics, and control," IEEE Computer, vol. 15, no.12, Oct. 1982.
- [11] V.D. Scheinman, "Design of a computer manipulator," Stanford A.I. Lab, Memo. AIM-92, 1969.
- [12] G. De Maria, L. Sciavicco and B. Siciliano, "A general solution algorithm to coordinate transformation for robotic manipulators," Proc. '85 ICAR, Tokyo, Japan, Sept. 1985.
- [13] A. Balestrino, G. De Maria, L. Sciavicco and B. Siciliano, "A novel approach to coordinate transformation for robotic manipulators," IEEE Trans. Syst., Man, Cybern., submitted for publication.
- [14] L. Sciavicco and B. Siciliano, "Coordinate transformation: a solution algorithm for one class of robots," IEEE Trans. Syst., Man, Cybern., submitted for publication.

- [15] L. Sciavicco and B. Siciliano, "An inverse kinematic solution algorithm for robots with two-by-two intersecting axes at the end effector," *Proc. IEEE Int. Conf. Rob. Autom.*, San Francisco, CA, April 1986.
- [16] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. 7, no. 12, Dec. 1977.
- [17] C.A. Klein and C.H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. 13, no. 2, Feb. 1983.
- [18] M. Vukobratovic and M. Kircanski, "A dynamic approach to nominal trajectory synthesis for redundant manipulators," *IEEE Trans. Syst., Man, Cybern.*, Jan./Feb. 1984.
- [19] M. Kircanski and M. Vukobratovic, "Trajectory planning for redundant manipulators in the presence of obstacles," *Proc. 5th CISM-IFTOMM Ro.Man.Sy.*, Udine, Italy, June 1984.

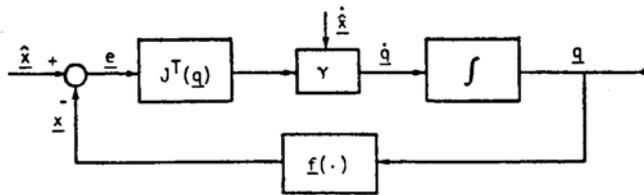


Fig. 1. The general dynamic inverse kinematic algorithm.

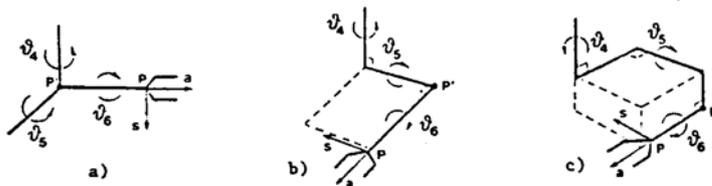


Fig. 2. Three basic kinematical configurations at the end effector: a) spherical wrist, b) 2-by-2 intersecting axes, c) nonconverging axes.

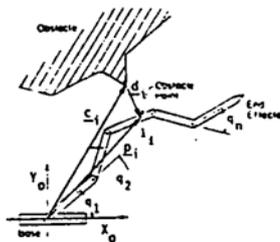


Fig. 3. Geometry of a planar manipulator showing the point nearest to the obstacle.