
Chapter 16

Achieving singularity robustness: an inverse kinematic solution algorithm for robot control

P. Chiacchio and B. Siciliano

1. PROBLEM STATEMENT AND PREVIOUS WORK

1.1 Robot Kinematics

For any robot with known geometrical dimensions, the direct kinematic equation describes the mapping of the $(n \times 1)$ vector of joint coordinates \underline{q} into the $(m \times 1)$ vector of robot's end-effector task (Cartesian) coordinates \underline{x} as (Denavit and Hartenberg (1))

$$\underline{x} = \underline{f}(\underline{q}) \quad (1.1)$$

where \underline{f} is a continuous nonlinear function, whose structure and parameters are known. Differentiating Eq.1.1 with respect to time yields the mapping between the joint velocity vector $\dot{\underline{q}}$ and the end-effector task velocity vector $\dot{\underline{x}}$, through the $(m \times n)$ Jacobian matrix $J(\underline{q}) = \partial \underline{f} / \partial \underline{q}$, i.e.

$$\dot{\underline{x}} = J(\underline{q}) \dot{\underline{q}}. \quad (1.2)$$

1.2 The Inverse Kinematic Problem

A robot is usually commanded by assigning a desired motion $\dot{\underline{x}}(t)$ to its end-effector. Hence, it is necessary to solve Eq.1.1 for $\dot{\underline{q}}(t)$ such that a control system can be designed which guarantees tracking of the desired joint motion $\dot{\underline{q}}(t)$. Solving the inverse kinematic problem becomes more dramatic for on-line applications, when the end-effector's motion is re-programmed on the basis of sensor information.

1.2.1 Previous work. The most natural way of solving the inverse kinematic problem relies upon the possibility of finding a closed-form analytical solution to Eq.1.1. Pieper (2) shows that this is true only for nonredundant structures ($m = n$) having simple geometries. In particular, a sufficient condition is given which establishes that the kinematic structure is solvable if it contains three consecutive joint rotation axes intersecting at a common point. For instance, all the robots having spherical wrists are solvable; there do exist, however, mechanical designs that do not satisfy the above condition. In addition, if the

joint velocities are needed by the control servos, Eq.1.2 must be solved for \dot{q} , thus requiring also the inversion of the Jacobian matrix $J(q)$. Therefore, the two shortcomings of the above technique, namely the solvability of the structure and the computational burden, have inspired the research to finding alternative solution techniques to the inverse kinematic problem which be applicable to any kinematic structure as well as be efficient from the computational viewpoint.

The other approach to the problem, commonly followed in the robotics literature, is based on the use of the inverse of the Jacobian matrix in Eq.1.2. It does not require any special assumption on the kinematic structure. In particular, it can be shown that the general solution to Eq.1.2, for a kinematically redundant structure ($m < n$), is given by (Whitney (3))

$$\dot{q} = J^{\dagger} \dot{x} + [I - J^{\dagger}(q)J(q)] \dot{q}_0 \quad (1.3)$$

where J^{\dagger} is the $(n \times m)$ Moore-Penrose pseudo-inverse matrix defined as $J^{\dagger} = J^T(JJ^T)^{-1}$, I is the $(n \times n)$ identity matrix and \dot{q}_0 is an $(n \times 1)$ arbitrary joint velocity vector. It can be noticed that the solution given in Eq.1.3 composes of the least-square solution term of minimum norm plus a homogeneous solution term created by the projection operator $(I - J^{\dagger}J)$ which selects the components of \dot{q}_0 in the null space (space of redundancy) of the mapping J . The vector \dot{q}_0 is usually adopted to optimize some additional criterion, such as obstacle avoidance, limited joint range etc. (see Klein (4) for a short survey on various choices of the vector \dot{q}_0).

1.2.2 Kinematic Singularities. The other drawback encountered in the solution to the inverse kinematic problem is certainly the occurrence of a kinematic singularity. A joint configuration is detected as a singular configuration if the determinant of the Jacobian matrix in Eq.1.2 vanishes. In case of redundant structures, a joint configuration is singular if the Jacobian matrix is not a full rank matrix. It can be seen that at a singularity either one or more columns of the Jacobian are null vectors or there is colinearity between two or more columns. The former case corresponds to having one or more joints whose motions do not produce any change of the end-effector location, like the "shoulder singularity" in a PUMA-like geometry (Hollerbach (5)). This type of singularity is the most critical one since it may fall into the robot's reachable workspace, thus constituting a problem for end-effector correct motion planning. On the other hand, in the latter case two or more joint infinitesimal motions produce the same infinitesimal change of the end-effector location, like the "elbow singularity" in a PUMA-like geometry (5). This kind of singularity is not as bad as the former since in those particular configurations the end-effector is at the boundaries of the robot's reachable workspace. Another singularity which usually occurs for manipulators having a spherical wrist is known as "wrist singularity" (5, Aboaf and Paul (6)), at which the wrist

cannot accommodate rotations about one of its orientation axes, which then becomes a degenerate axis.

The main problem concerned with a solution of the kind of Eq.1.3 is that kinematic singularities are not avoided in any practical sense, since the joint velocities are minimized only instantaneously (Baillieul et al (7)). Nevertheless, a solution based on Eq.1.3 with singularity robustness has been recently proposed by Nakamura and Hanafusa (8).

2. AN INVERSE KINEMATIC SOLUTION ALGORITHM WITH SINGULARITY ROBUSTNESS

The inverse kinematic solution algorithm with singularity robustness, which is the main contribution of this work, is naturally derived from a general computational method recently established in the literature (Balestrino et al (9), Siciliano (10), Balestrino et al (11)) which is applicable to any redundant and nonredundant robot geometry. Hence, the next subsection is devoted to briefly recall that general algorithm.

2.1 The General Inverse Kinematic Solution Algorithm

The inverse kinematic problem is solved by constructing the dynamic system of Fig.2.1, whose input is the end-effector target trajectory $\underline{\hat{x}}(t)$ and whose outputs are the corresponding joint position and velocity trajectories, $\underline{q}(t)$ and $\underline{\dot{q}}(t)$ respectively; K is a positive definite diagonal matrix.

Lemma. The dynamic system of Fig.2.1 assures that the tracking error $\underline{e}(t) = \underline{\hat{x}}(t) - \underline{x}(t)$ can be made arbitrarily small by increasing the minimum element of K .

Proof. According to the Lyapunov direct method for the analysis of the stability of nonlinear systems, define the positive definite Lyapunov function

$$v(\underline{e}, t) = \frac{1}{2} \underline{e}^T(t) \underline{e}(t). \quad (2.1)$$

Its time derivative results, via Eq.1.2 (dropping the time dependence)

$$\dot{v}(\underline{e}) = \underline{e}^T \dot{\underline{x}} - \underline{e}^T J(\underline{q}) K J^T(\underline{q}) \underline{e}. \quad (2.2)$$

It can be recognized that $\dot{v}(\underline{e})$ is negative definite only outside a region in the error space containing $\underline{e} = \underline{0}$, which is attractive for all trajectories. The maximum tracking errors will depend directly on $\|\underline{\hat{x}}\|$ and inversely on the minimum element of K . It must be emphasized that the steady-state error ($\underline{\dot{x}} = \underline{0}$) is identically zero.

From the above lemma it follows that the application of the dynamic system of Fig. 1 to solve the inverse kinematic problem for a general structure is twofold. It can be used off-line to make $\underline{q}(t)$ approach a desired constant solution $\underline{\hat{q}}$ to Eq.1.1, with $\underline{q}(0) \neq \underline{\hat{q}}$, arbitrarily fast. It can be adopted on-line to guarantee that $\underline{x}(t)$ will track the desired end-effector trajectory $\underline{\hat{x}}(t)$ with an arbitrarily fast

decaying error.

The advantages of this technique can be summarized as:

- a) it is applicable to any robot since it does not require any special assumption regarding the kinematic structure,
- b) it is computationally efficient since it is based only on direct kinematic functions (\underline{f} and J), generating joint velocities at no additional cost,
- c) the use of the transpose of the Jacobian may avoid problems when kinematic singularities occur (this point will be faced up in the following subsection),
- d) given the initial configuration of the structure, uniqueness of the solution is assured as the algorithm, generates adjacent solutions step-by-step.

The same algorithm can be partitioned into two stages to better account for the particular geometry of the structure, with the inherent advantage of further decreasing the computational burden ((9), (10), (11), De Maria et al (12), Sciavicco and Siciliano (13,14,15)).

2.2 Making the Algorithm Robust to Singularities

Based on the remarks of subsection 1.2.2, only kinematic singularities that cause one column of the Jacobian matrix of Eq.1.2 to vanish are considered in what follows. Thus, assume that \underline{j}_i be the null column vector of J . This implies that the motion of the corresponding joint q_i does not produce any change of the end-effector location. Similarly, if the computational scheme of Fig.2.1 is adopted to solve the inverse kinematic problem, it is easy to recognize that there will be no motion at the joint q_i . This result is consistent with the mechanical interpretation that it is not worth moving the joint q_i .

The real drawback to the solution of the inverse kinematic problem, however, occurs when the trajectory assigned to the end-effector passes in the proximity of a singularity. In that case, indeed, the norm of the vector \underline{j}_i approaches zero and higher values for the corresponding joint velocity \dot{q}_i are expected to allow the end-effector to track the desired trajectory. On the other hand, from the point of view of the scheme of Fig.2.1 with constant gains in the matrix K , it happens that the weight of the tracking error $\underline{e}(t)$ on the control \dot{q}_i is "masked" by the small value of the norm of \underline{j}_i , compared to the other joints. This implies that the joint q_i cannot move as fast as required by the end-effector trajectory and the tracking error tends to increase.

In order to overcome the above problem, here it is proposed to adjust the elements k_i of the matrix K of the scheme of Fig.2.1 according to the current joint configuration, such that the algorithm be robust to the occurrence of kinematic singularities. The suggestion is to modify the elements k_i into

$$k_i / \|\underline{j}_i(q)\| \quad (2.3)$$

which assures that, in the proximity of a singularity when $\|j_i\|$ takes a small value, only the weight for the control \dot{q}_i increases, guaranteeing a contained tracking error. Obviously, if the trajectory crosses the singular point, the above weight is not allowed to take an infinitely large value. In other words, there must be a numerical threshold for $\|j_i\|$ so as to avoid division by zero in Eq.2.3.

It has to be emphasized that, in light of the choice in Eq.2.3, the control at each joint becomes

$$\dot{q}_i = (k_i / \|j_i(q)\|) j_{ie}^T \quad (2.4)$$

which corresponds to making the actual weight quasi-independent (see the angle of the inner product j_{ie}) of the particular configuration q attained by the structure along the trajectory.

This point turns out to be advantageous for the discrete-time implementation of the algorithm. It can be recognized, indeed, that there does exist a maximum value for the equivalent gains of K which depends inversely on the sampling time. To this purpose, the above "normalization" serves as a design tool to set the k_i 's regardless of the desired end-effector trajectory to track.

3. A PRACTICAL EXAMPLE

In order to show the effectiveness of the proposed inverse kinematic solution algorithm with singularity robustness, a case study has been worked out. The robot prototype DEXTER available at FIAR S.p.A., Italy (Fig.3.1) has been selected. It has seven degrees of freedom (redundant) and a PUMA-like geometry as regards the joints q_2 , q_3 and q_4 . Only the first four joints are considered in the two sets of simulations that follow. It is quite straightforward to recognize that a shoulder singularity occurs at any point along the axis of the shoulder joint q_2 .

In the first set, the sliding joint q_1 is assumed to be blocked such that the structure be nonredundant for an end-effector positioning task. The desired trajectory is a straight line parallel to the floor, 15 cm. away from the above singularity axes. Fig.3.2a shows that the end-effector position tracking error becomes considerably smaller if the modification of the gains k_i 's (Eq.2.3) occurs. The shoulder joint q_2 anticipates its motion, according to the increased sensitivity to the tracking error (Fig.3.2b).

In the second set, q_1 is released such that the structure becomes redundant for the same kind of task as above. The desired trajectory is a straight line parallel to the floor, crossing the plane formed by the axes of the sliding joint q_1 and the shoulder joint q_2 , with the peculiarity that the trajectory does have a component on the axis of q_1 , which is then required to slide. An improvement on the tracking error can be seen (Fig.3.3a) as in the previous case. In addition, the joint velocity \dot{q}_1 decreases as all the joints concur to better accomplish the required motion of the robot's end-effector (Fig.3.3b).

REFERENCES

1. Denavit, J., and Hartenberg, R.S., 1955, 'A kinematic notation for lower-pair mechanisms based on matrices', ASME J. Appl. Mech., 22, 215-221.
2. Pieper, D.L., 1969, 'The Kinematics of Manipulators under Computer Control', Ph.D. dissertation, Stanford University, U.S.A..
3. Whitney, D.E., 1972, 'The mathematics of coordinated control of prosthetic arms and manipulators', ASME J. Dyn. Syst., Meas., Contr., 94, 303-309.
4. Klein, C.A., 1985, 'Use of redundancy in the design of robotic systems', Robotics Research: The Second International Symposium, MIT Press, 207-214.
5. Hollerbach, J.M., 1985, 'Evaluation of redundant manipulators derived from the PUMA geometry', Proc. of the ASME Winter Annual Meeting, 187-192.
6. Aboaf, E.W., and Paul, R.P., 1987, 'Living with the singularity of robot wrists', Proc. of the 1987 IEEE International Conference on Robotics and Automation, 1713-1717.
7. Baillieul, J., Hollerbach, J., and Brockett, R.W., 1984, 'Programming and control of kinematically redundant manipulators', Proc. of the 23rd IEEE CDC, 768-774.
8. Nakamura, Y., and Hanafusa, H., 1986, 'Inverse kinematic solutions with singularity robustness for robot manipulator control', ASME J. Dyn. Syst., Meas., Contr., 108, 163-171.
9. Balestrino, A., De Maria G., and Sciavicco L., 1984, 'Robust control of robotic manipulators', Prep. of the 9th IFAC World Congress, 6, 80-85.
10. Siciliano, B., 1986, 'Solution Algorithms to the Inverse Kinematic Problem for Manipulation Robots (in Italian)', Ph.D. dissertation, Università di Napoli, Italy.
11. Balestrino, A., De Maria, G., Sciavicco, L. and Siciliano, B., 1987, 'An algorithmic approach to coordinate transformation for robotic manipulators', Advanced Robotics, 2, to be published.
12. De Maria, G., Sciavicco, L. and Siciliano, B., 1985, 'A general solution algorithm to coordinate transformation for robotic manipulators', Proc. of the Second International Conference on Advanced Robotics, 251-258.

13. Sciavicco, L., and Siciliano, B., 1986, 'An inverse kinematic solution algorithm for robots with two-by-two intersecting axes at the end effector', Proc. of the 1986 IEEE International Conference on Robotics and Automation, 673-678.
14. Sciavicco, L., and Siciliano, B., 1986, 'Coordinate transformation: A solution algorithm for one class of robots', IEEE Trans. Syst., Man, Cybern., 16, 550-559.
15. Sciavicco, L., and Siciliano, B., 1986, 'Solving the inverse kinematic problem for robotic manipulators,' Prep. of the Sixth CISM-IFTOMM Ro.Man.Sy, 82-89.

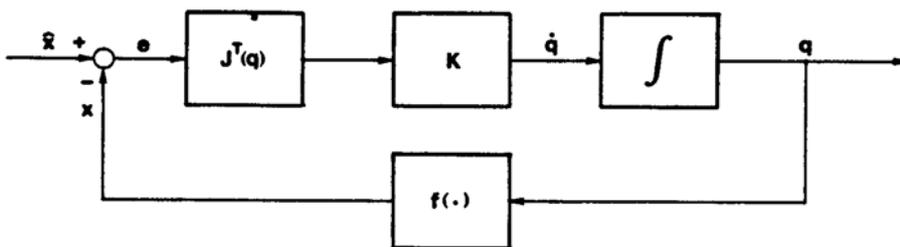


Fig.2.1 The closed-loop scheme of the inverse kinematic solution algorithm.

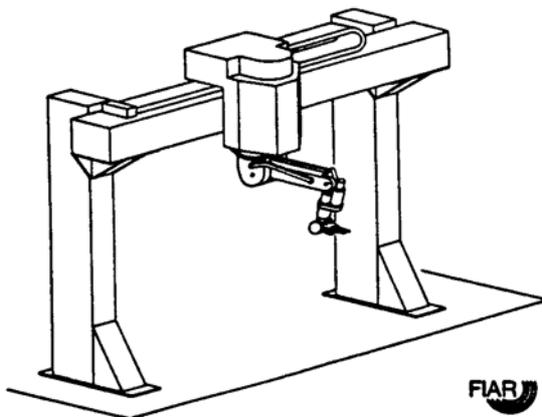


Fig.3.1 The prototype robot DEXTER (courtesy of FIAR S.p.A.)

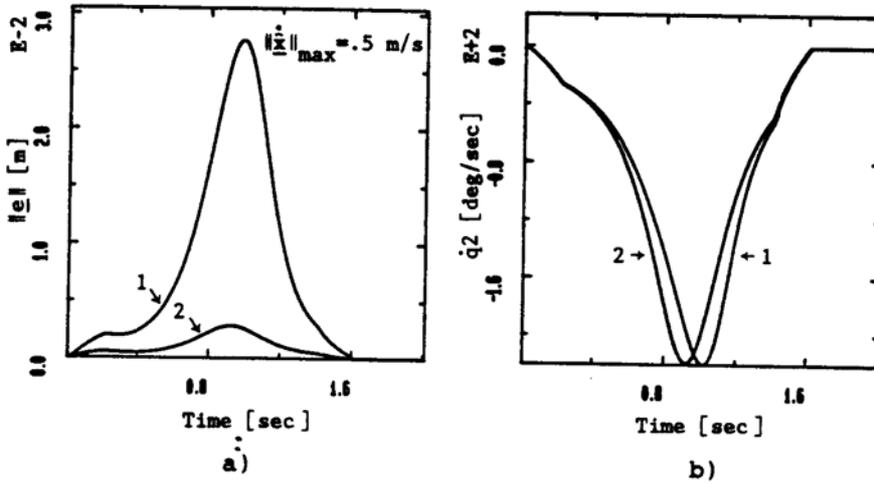


Fig.3.2 a) End-effector tracking error.
 b) Shoulder joint velocity \dot{q}_2 .
 1 - without gain adjusting,
 2 - with gain adjusting.

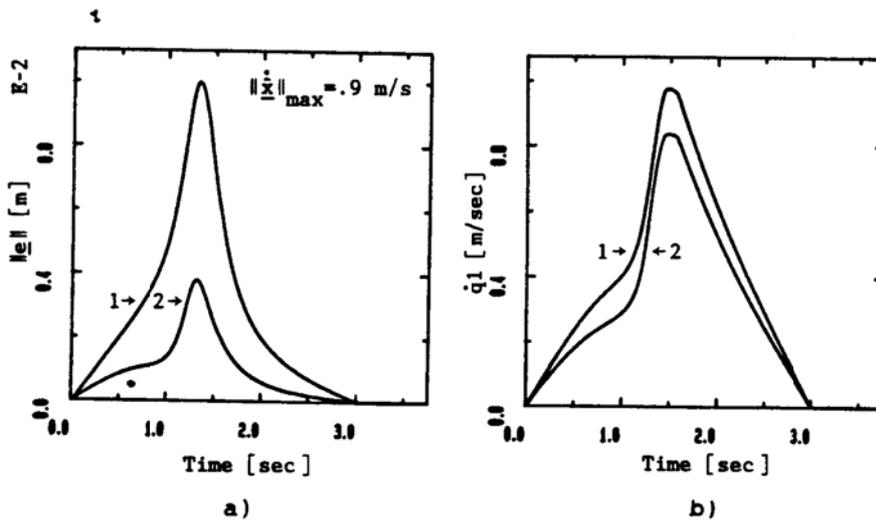


Fig.3.3 a) End-effector tracking error.
 b) Sliding joint velocity \dot{q}_1 .
 1 - without gain adjusting,
 2 - with gain adjusting.