

A COMPUTATIONAL TECHNIQUE FOR SOLVING ROBOT END-EFFECTOR TRAJECTORIES INTO JOINT TRAJECTORIES

Lorenzo Sciavicco and Bruno Siciliano

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

The Problem

For any robot with known geometrical parameters, the direct kinematic equation can be written as [1]

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

where t is the time variable, $\underline{x}$ is the $(m \times 1)$ vector of task coordinates, $\underline{q}$ is the $(n \times 1)$ vector of joint coordinates, and $\underline{f}$ is a nonlinear vectorial function, whose structure and parameters are known.

It is well known that the solution of the inverse kinematic problem, i.e. solving eq. (1), is of fundamental importance for robot control. A typical robot task is specified as a trajectory assigned to the end-effector, say $\hat{\underline{x}}(t)$, $\dot{\hat{\underline{x}}}(t)$, $\ddot{\hat{\underline{x}}}(t)$. This must be solved into a joint trajectory, say $\hat{\underline{q}}(t)$, $\dot{\hat{\underline{q}}}(t)$, $\ddot{\hat{\underline{q}}}(t)$, which constitutes the reference input to the joint control servos.

The most popular approach to the problem relies on the possibility of finding a closed-form analytical solution to the robot kinematic equation. It is recognized that this is true only for robots having simple geometries [2], such as the spherical wrist, the elbow manipulator etc. Further, the analytical solution is usually non-unique and sequential, and requires the computation of Atan2 functions.

Besides eq. (1), the other direct kinematic relation is

$$\dot{\underline{x}}(t) = \underline{J}(\underline{q}(t))\dot{\underline{q}}(t). \quad (2)$$

where $\underline{J} = \partial \underline{f} / \partial \underline{q}$ is the Jacobian matrix.

Previous Work and the Extended Algorithm

This paper originates from a rather different approach to the problem which is applicable to any robot structure, whose kinematic equation and Jacobian are known. The idea, independently proposed in [3] and [4], is to reformulate the inverse kinematic problem as a tracking problem for a simple dynamic system. Let

$$\underline{e}(t) = \hat{\underline{x}}(t) - \underline{f}(\underline{q}(t)) \quad (3)$$

denote the end-effector error vector between the desired end-effector configuration vector $\hat{\underline{x}}(t)$ and the dynamic system output vector $\underline{x}(t) = \underline{f}(\underline{q}(t))$. Since the initial robot configuration can be assumed to be known, it is $\underline{e}(0) = \underline{0}$. The error dynamics results, via (2),

$$\dot{\underline{e}} = \dot{\hat{\underline{x}}} - \underline{J}(\underline{q})\dot{\underline{q}} \quad (4)$$

This work was supported by MPI Research Funds.

where the explicit dependence on time has been dropped. In [3,4] it was shown that a control law of type

$$\dot{\underline{q}} = \underline{KJ}^T(\underline{q})\underline{e} \quad (5)$$

with $\underline{K}$ a positive definite matrix, guarantees that the tracking error is bounded and the positional error ($\hat{\underline{x}} = \underline{0}$) is null. Further, the choice [3]

$$\dot{\underline{q}} = \underline{J}^{-1}(\underline{\hat{x}} + \underline{K}\underline{e}) \quad (6)$$

with $m = n$ for simplicity (otherwise a generalized inverse of $\underline{J}$ must be used), guarantees that the tracking error is identically zero. The joint velocity vector $\dot{\underline{q}}$ is also directly generated by the algorithm. In particular, the control law (5) has the advantage that the algorithm does not require any matrix inversion; thus it may overcome the problem of inverting eq. (1) in the neighborhood of a kinematic singularity, and it can be applied to redundant manipulators ($m < n$). Further, the general inverse kinematic algorithm based on eq. (5) has been customized to the special cases of manipulators having a spherical wrist [5], two-by-two intersecting axes at the end-effector [6], and to the case of redundant manipulators with a set of constraints on obstacle avoidance and limited joint range [7], dexterity and task compatibility [8].

One of the limits of the preceding approach is that the joint acceleration vector $\ddot{\underline{q}}$ is not directly generated. Having also that feature may be nice for designing a joint space dynamic control, say computed torque for instance [9]. To this purpose, eq. (2) can be further derived with respect to time, yielding

$$\ddot{\underline{x}} = \dot{\underline{J}}(\underline{q})\dot{\underline{q}} + \underline{J}(\underline{q})\ddot{\underline{q}}. \quad (7)$$

As a consequence, the 2nd order error dynamics results

$$\ddot{\underline{e}} = \ddot{\hat{\underline{x}}} - \dot{\underline{J}}(\underline{q})\dot{\underline{q}} - \underline{J}(\underline{q})\ddot{\underline{q}}. \quad (8)$$

and a suitable control law is to be sought in order to guarantee error convergence.

In [4] the extension of the 1st order scheme based on the control law (5) to a 2nd order scheme was proposed, but a formal proof of algorithm convergence was not given. Later, in [10] the control law

$$\ddot{\underline{q}} = \underline{J}^{-1}(\underline{q})[-\dot{\underline{J}}(\underline{q})\dot{\underline{q}} + \underline{K}_p\underline{e} - \underline{K}_d\dot{\underline{x}}] \quad (9)$$

has been proposed, with $\underline{K}_p$ and $\underline{K}_d$ positive definite matrices, which ensures a limited tracking error (depending on the shape of $\hat{\underline{x}}(t)$) and a null positional error. Also in [11] a 2nd order posi-

tional scheme based on the control law

$$\ddot{\mathbf{q}} = \mathbf{J}^T(\mathbf{K}_p \mathbf{e} + \mathbf{K}_d \dot{\mathbf{e}}) \quad (10)$$

has been proposed as an alternative scheme for redundant manipulators which avoids the inversion of the Jacobian and the computation of the term $\mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$.

A new 2nd order inverse kinematic scheme can be derived by adopting the same "inverse dynamics" technique which is typical of a resolved acceleration control [9]. Indeed, the simple control law

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})[\ddot{\mathbf{x}} - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{K}_p \mathbf{e} + \mathbf{K}_d \dot{\mathbf{e}}] \quad (11)$$

leads to the error dynamics equation

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \mathbf{0}. \quad (12)$$

As emphasized above, the initial robot configuration is known, that is $\mathbf{e}(0) = \dot{\mathbf{e}}(0) = \mathbf{0}$. Therefore, the control law (12) ensures that $\mathbf{x}(t)$ exactly tracks the reference trajectory $\hat{\mathbf{x}}(\bar{t})$, i.e. $\mathbf{q}(t) = \mathbf{f}^{-1}(\hat{\mathbf{x}}(t)) = \hat{\mathbf{q}}(t)$. The solution (11) is easily recognized as improving the solution (9), since they both require the on-line computation of $\mathbf{J}^{-1}(\mathbf{q})$ and $\dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}$ and ensure a null positional error, but only (11) guarantees a null tracking error. The closed-loop scheme based on the control law (11) is illustrated in Fig. 1.

A final remark concerns with the computational burden and the drawback of matrix inversion ($\mathbf{J}^{-1}(\mathbf{q})$ and $\dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}$). Adopting a solution similar to the one proposed in [12] for the task space control would lead to an inverse kinematic algorithm which requires the computation of the transpose of the Jacobian only, similarly to what happens for the 1st order solution in (5). In this case, the control law would be formally analogous to (10), but the matrices $\mathbf{K}_p$ and $\mathbf{K}_d$ should be properly designed in order to guarantee that the tracking error be confined to an attractive region of the error space containing the origin.

An Example

The first three degree-of-freedom's of a PUMA robot [9] are considered to apply the proposed inverse kinematic algorithm. The reference task trajectory consists of a straight line of 0.5 m to be executed in 0.5 s, with a sinusoidal velocity profile; the initial robot configuration ($q_1 = q_2 = q_3 = 0$ deg) places the end-effector in the reference initial position. $\mathbf{K}_p = \text{diag}(100 \ 100 \ 100)$ and $\mathbf{K}_d = \text{diag}(20 \ 20 \ 20)$. The sampling time is 2 ms. The norms of the error vector and its derivative are reported in Fig. 2; it can be recognized that the scheme of Fig. 1 performs the kinematic inversion of the given end-effector trajectory, guaranteeing a small tracking error, essentially due to the discrete-time implementation, and a null positional error.

References

- [1] J. Denavit and R.S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *J. Appl. Mech.*, June 1955.
- [2] D.L. Pieper, *The Kinematics of Manipulators Under Computer Control*. Ph.D. Thesis, Stanford University, CA, 1968.
- [3] A. Balestrino, G. De Maria, and L. Sciavicco, "Robust control of robotic manipulators," *9th IFAC World Congress*, Budapest, Hungary, July 1984.
- [4] W.A. Wolovich and H. Elliott, "A computational technique for inverse kinematics," *23rd IEEE CDC*, Las Vegas, NV, Dec. 1984.
- [5] A. Balestrino, G. De Maria, L. Sciavicco, and B. Siciliano, "An algorithmic approach to coordinate transformation for robotic manipulators," *Advanced Robotics*, to appear, 1988.
- [6] L. Sciavicco and B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," *IEEE Trans. Syst., Man, Cybern.*, 16, 4, 1986.
- [7] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE J. Rob., Autom.*, to appear, 1988.
- [8] L. Sciavicco and B. Siciliano, "An inverse kinematic solution algorithm for dexterous redundant manipulators," *3rd ICAR*, Versailles, France, Oct. 1987.
- [9] J.J. Craig, *Introduction to Robotics: Mechanics and Control*. Reading, Mass.: Addison-Wesley, 1986.
- [10] W.A. Wolovich and K.F. Flueckiger, "Inverse kinematic-based robot control," *Workshop Space Telerob.*, JPL, Pasadena, CA, Jan. 1987.
- [11] J.-J.E. Slotine and D.R. Yoerger, "A rule-based inverse kinematics algorithm for redundant manipulators," *Int. J. Rob., Autom.*, 2, 2, 1987.
- [12] A. Balestrino, G. De Maria, and L. Sciavicco, "Adaptive control of manipulators in the task oriented space," *13th ISIR/Robots 7*, Chicago, IL, Apr. 1983.

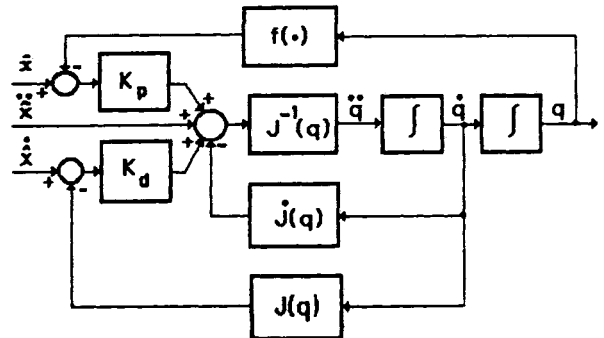


Fig.1 - The 2nd order inverse kinematic scheme

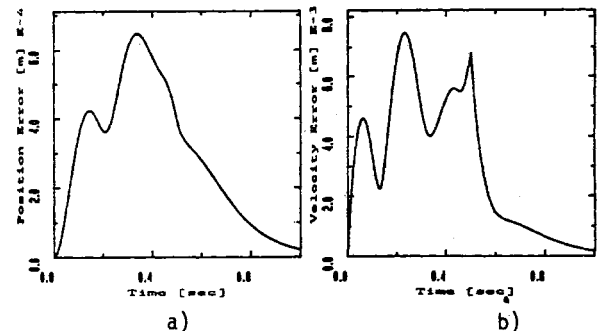


Fig.2 - Tracking errors: a) position, b) velocity