

Bruno Siciliano

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

CLOSED-LOOP COMPUTATIONAL SCHEMES OF ROBOT INVERSE KINEMATICS

Abstract

This paper is aimed at surveying a class of computational inverse kinematic schemes for robots of arbitrary architecture. The supporting idea is to reformulate the inverse kinematic problem as a control problem for a simple closed-loop dynamic system. First-order and second-order schemes are formally derived. They can be distinguished into two categories, namely those which utilize the inverse of the robot's Jacobian and those which are based on the transpose of the Jacobian matrix. The former are analogous to widely used resolved-rate and resolved-acceleration controls. The latter are more efficient from the computational viewpoint. Algorithm convergence for the latter is guaranteed from Lyapunov stability analysis.

Introduction

The solution of the inverse kinematic problem has been a research topic investigated by many roboticists in the last decade. It has been well-known, since one of the earlier works in inverse kinematics [1], that analytical inverse kinematic solutions exist only for special manipulator geometries. For all those structures which are not solvable in closed-form, a number of computational schemes have been proposed in the literature most of which are based on the computation of the robot's Jacobian. The pioneering resolved motion rate control [2] solves the linearized kinematic equation for joint velocities which are integrated to obtain joint displacements. The iterative method based on a nonlinear optimization algorithm [3] uses a modified Newton-Raphson method. The numerical method derived from the formulation of invariants in the rotational part of the closure equations [4] is yet another solution to the problem. Another numerical method for the solution of systems of nonlinear equations is described in [5]. The approach based on the use of continuation methods [6] differs, indeed, from the previous methods.

In the context of solving for inverse kinematics, it is usually desired to generate not only joint displacements but also joint velocities, and accelerations eventually. This would provide the setpoints for the control system in the joint space. Here, computation time becomes an important concern for those on-line sensor-driven tasks when it is required to compute the inverse kinematics at the same rate as the joint servo rate [7]. Special computer architectures have been introduced for inverse kinematic position computation [8]. Also, for simple manipulator geometries such as six d.o.f. wrist-partitioned architectures, closed-form solutions have been found for joint velocities [9] as well as for joint accelerations [10] which avoid the explicit inversion of the Jacobian matrix. On the other hand, if it is desired to control the robot directly in the task space, the solution of the inverse kinematics is apparently avoided, but the control becomes more sophisticated [11].

A rather different approach to the solution of the inverse kinematic problem is obtained by constructing a simple closed-loop stable dynamic system, whose input is the desired end-effector trajectory and whose outputs are the joint displacement, velocity (and acceleration) trajectories. The original idea has been independently proposed in [12] and [13] for solving the position component of the end-effector

trajectory. There, a first-order scheme based on the computation of the transpose of the Jacobian is devised which generates the joint displacements and velocities while guaranteeing a null positional error and a norm-bounded tracking error. The extension of the scheme to account for the orientation component of the end-effector has been described in [14,15] and the investigation of a special non-solvable structure has been presented in [14,16]. The application to the case of robots with redundancy has been discussed in [14,17,18,19,20], and recently also in [21,22]. The issue of kinematic singularity robustness of the scheme has been addressed in [23]. The convergence of all above schemes is ensured by Lyapunov stability analysis which leads to establishing estimates of the region of attractiveness of the solutions. Also, the algorithms are remarkably based on the sole computation of direct kinematic functions, and therefore avoid the numerical instabilities associated with any matrix inversion or pseudo-inversion.

Alternatively, first-order schemes based on the computation of the inverse of the Jacobian have been suggested in [24,25], but they are useful only as positional schemes (e.g. constant end-effector location). A tracking scheme based on the same concept has been designed in [12,14], which is formally analogous to resolved-rate control [2]. Moreover, a second-order scheme corresponding to resolved-acceleration control [27] has been given in [26]. If the inversion of the Jacobian is to be avoided, the solution proposed in [22] can be adopted which serves as a second-order positional scheme. A new tracking scheme, instead, is derived here from sliding mode control [28]. In the following, this class of first-order and second-order schemes based on the closed-loop formulation of the inverse kinematic problem is surveyed. Advantages and limitations of each solution are discussed.

Closed-loop formulation of the inverse kinematic problem

It is well-known that the vector of end-effector coordinates x (usually position + orientation) is described as a function of the vector of joint displacements q by the direct kinematic transformation

$$x = f(q) \quad (1)$$

where f is a continuous nonlinear function whose structure and parameters are uniquely determined for each given robot [7]. A sufficient condition for an analytical closed-form solution to exist is that three consecutive joint revolute axes intersect at a common point [1]; the so-called elbow manipulator, however, is an example of solvable structure which does not satisfy that condition [7]. The majority of computational approaches in robot inverse kinematics are based on the differential linear mapping between x and q

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

where $J(q) \triangleq \partial f / \partial q$ is the joint configuration-dependent Jacobian matrix; the upper dot indicates time-derivative, although time-dependence is not explicitly evidenced in (1) and (2). Thus, the so-called resolved rate or Jacobian control [2] is given by

$$\dot{q} = J^{-1}(q)\dot{x} \quad (3)$$

which is integrated over time to provide q . In (3) it is assumed that an inverse to J does exist; a pseudo-inverse must be used if the Jacobian degenerates or if the manipulator is redundant [29].

A different approach to the inverse kinematic problem which is independent of the particular robot geometry is illustrated in the following. The idea is to reformulate the problem as a tracking problem for a simple dynamic system [12,13]. Let

$$e \triangleq \dot{x}_d - \dot{x} = \dot{x}_d - f(q) \quad (4)$$

denote the error vector between the desired end-effector location vector x_d and the vector x which is thought as computed from the current joint configuration vector q via (1). The definition of the error for end-effector position is immediate, whereas for end-effector orientation the reader is referred to [14,15,16,27]. The closed-loop scheme of Fig. 1 can be constructed. If the control \dot{q} is chosen so that the system is guaranteed to be stable, i.e. $e \rightarrow 0$, it can be concluded that the system performs a first-order kinematic inversion; namely, given x_d (and \dot{x}_d if needed), it generates q and \dot{q} . In addition, differentiating (4) with respect to time gives

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

where the vector \dot{x} is computed via (2). Similarly to Fig. 1, the closed-loop scheme of Fig. 2 can be constructed. If the control \ddot{q} is chosen so that the system is guaranteed to be stable, i.e. $e \rightarrow 0$, it can be concluded that the system performs a second-order kinematic inversion; namely, given x_d and \dot{x}_d (and \ddot{x}_d if needed), it generates q , \dot{q} and \ddot{q} .

First-order schemes

Theorem 1. If \dot{x}_d belongs to the class of C^1 functions, and the matrix $J(q)$ has full rank for all joint configurations q 's, the control law

$$\dot{q} = J^{-1}(q)[\dot{x}_d + Ke] \quad (6)$$

with K a positive definite matrix (all its eigenvalues are strictly in the right half complex plane), ensures that $e \rightarrow 0$ [12,14].

Proof. Direct substitution of (6) in (5) gives

$$\dot{e} + Ke = 0 \quad (7)$$

which, in force of the positive definiteness of K , guarantees that $e \rightarrow 0$. Notice that a pseudo-inverse of J must be used if $\dim(q) > \dim(x)$ in (1).
End of proof.

Theorem 2. If x_d belongs to the class of C^1 functions, the control law

$$\dot{q} = (1 + e^T K^T \dot{x}_d / e^T K^T J J^T K e) J^T K e \quad (8)$$

with K a (not necessarily symmetric) positive definite matrix, ensures that $e \rightarrow 0$ [12,14]. Note that the dependence on q in J has been dropped for the sake of notation compactness.

Proof. Define the Lyapunov function candidate of the error vector e in (4) as

$$V = \frac{1}{2} e^T K e. \quad (9)$$

Its time derivative along the trajectories of the system (5) results in

$$\dot{V} = e^T K^T \dot{x}_d - e^T K^T J \dot{q}. \quad (10)$$

Direct substitution of (8) in (10) gives

$$\dot{V} = -e^T K^T J J^T K e \leq 0 \quad (11)$$

which in turn implies that $e \rightarrow 0$. According to Lyapunov stability, however, the case $\dot{V} = 0$ must be analyzed. From (11) it is seen that the condition $\dot{V} = 0$ implies $e = 0$, except when Ke belongs to the null space of J^T , where the algorithm may in principle

get "stuck". This is possible, for example, when the desired trajectory x_d extends outside the workspace, so that the manipulator will converge to the closest point on the boundary of the workspace, with the remaining error perpendicular to the boundary [30]. On the other hand, if the manipulator is at an internal singularity, one can easily show that such equilibrium point is unstable, and the time evolution of x_d will contribute to decrease \dot{v} again.

End of proof.

Corollary 1. If the desired end-effector location is constant, i.e. $\dot{x}_d = 0$, the control law (8) ensuring that $e \rightarrow 0$, reduces to [14,21]

$$\dot{q} = J^T K e. \quad (12)$$

Proof. Direct substitution of (12) in (10) with $\dot{x}_d = 0$ trivially leads to (11). In this case, however, if Ke belongs to the null space of J^T , x_d must be slightly perturbed in order to avoid that the solution gets stuck at $e \neq 0$.

End of proof.

Theorem 3. If x_d belongs to the class of C^1 functions and its time derivative is norm-bounded, i.e. $\|\dot{x}_d\| \leq v$, the control law (12) ensures that e can be made arbitrarily small by increasing the minimum eigenvalue of K [12,13,14].

Proof. Define the Lyapunov function candidate as in (9). Its time derivative along the trajectories of the system (5) under the control (12) results in

$$\dot{v} = e^T K^T \dot{x}_d - e^T K^T J J^T K e. \quad (13)$$

Let λ_J denote the minimum eigenvalue of $J J^T$ ($\lambda_J > 0$). Also, let λ_K denote the minimum eigenvalue of K ($\lambda_K > 0$). By the assumption on $\|\dot{x}_d\|$, it follows that

$$\dot{v} \leq \|e\| \lambda_K v - \|e\|^2 \lambda_K^2 \lambda_J \quad (14)$$

which implies that $\dot{v} \leq 0$ as long as

$$\|e\| \geq v / \lambda_K \lambda_J \quad (15)$$

i.e. the error enters an attractive region of the error space containing the origin $e = 0$ which can be made arbitrarily small by increasing λ_K . Notice that at steady-state ($\dot{x}_d = 0$), it is trivially $e = 0$.

End of proof.

A few comments are in order concerning the three first-order schemes suggested above. The first scheme (6) resembles resolved-rate control with additional feedback Ke which avoids the typical drawback of cumulative errors of (3). Notice, however, that the error e for all the schemes presented here is not to be intended as actually measured at the end-effector (or computed from joint measurements via (1)); it is just an "algorithmic" error computed from the outputs of the scheme of Fig. 1 via (1). The second and the third schemes have in common the nice feature over the first scheme that they are based on the transpose of J . They avoid the numerical instabilities associated with matrix inversion as well as their computational burden is reduced. One drawback of the second scheme (8), however, is that it introduces, in the neighborhood of $e = 0$, an equivalent gain which tends to ∞ . The analysis of the second term generated on the right hand side of (8), indeed, reveals that this is given by the ratio of two quantities that go to zero, as $e \rightarrow 0$, with the same order two. Therefore, in order to avoid so-called "chattering" time evolution in the joint velocities \dot{q} , it seems more convenient to adopt the third scheme (12) which is also less computational-demanding. Furthermore, an appealing feature of the solution (12) lies in the following physical interpretation [12,21]. It is well known that the relationship between the joint torque vector τ and the end-effector force vector γ is given by [7]

$$\tau = J^T(q)\gamma \quad (16)$$

which is dual of (2). As a consequence, the control law (12) is analogous to applying an elastic force Ke at the end-effector of an ideal manipulator with the same structure as the manipulator of interest, but having an inertia matrix equal to the identity matrix, and operating in the absence of gravity or friction. In force of this analogy, it can be recognized, for instance, that in the case when Ke is in the null space of J^T discussed above, this corresponds to applying end-effector forces in a direction along which the manipulator cannot move.

The solution (12) suggests that the tracking error can be made arbitrarily small by choosing λ_k large enough. It should be emphasized, however, that the implementation of the discrete-time solution algorithm, and then the sampling rate, limits the maximum value of λ_k . In order to establish an optimum for that value, a discrete-time stability proof should be undertaken, as done for instance in [22], but this goes beyond the scopes of the present work.

Second-order schemes

Theorem 4. If x_d belongs to the class of C^2 functions, and the matrix $J(q)$ has full rank for all joint configurations q 's, the control law

$$\ddot{q} = J^{-1}(q)[\ddot{x}_d - \dot{J}(q)\dot{q} + K_p e + K_d \dot{e}] \quad (17)$$

with $\dot{J}(q) \triangleq d/dt[J(q)]$ and K_p, K_d positive definite matrices such that the matrix

$$M \triangleq \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} \quad (18)$$

is a Hurwitz matrix, ensures that $e \rightarrow 0$ [26].

Proof. Differentiating (5) with respect to time yields

$$\ddot{e} = \ddot{x}_d - \ddot{x} = \ddot{x}_d - \dot{J}(q)\dot{q} - J(q)\ddot{q}. \quad (19)$$

Direct substitution of (17) in (19) gives

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \quad (20)$$

which, in force of (18), guarantees that $e \rightarrow 0$.

End of proof.

Theorem 5. If x_d belongs to the class of C^2 functions, the control law

$$\ddot{q} = (1 + s^T K^T z / s^T K^T J J^T K s) J^T K s, \quad (21)$$

where K is a (not necessarily symmetric) positive definite matrix,

$$s = \dot{e} + \Lambda e \quad (22)$$

with Λ being a (not necessarily symmetric) positive definite matrix, and

$$z = \ddot{x}_d - \dot{J}\dot{q} + \Lambda \dot{e}, \quad (23)$$

ensures that $s \rightarrow 0$, which in turn implies that $e \rightarrow 0$ [28].

Proof. Define the Lyapunov function candidate of the error "sliding" vector in (22) as

$$V = \frac{1}{2} s^T K s. \quad (24)$$

Its time derivative along the trajectories of the system (19) results in

$$\dot{V} = s^T K^T \dot{z} - s^T K^T J \ddot{q} \quad (25)$$

with z defined in (23). Direct substitution of (21) in (25) gives

$$\dot{V} = -s^T K^T J J^T K s \leq 0 \quad (26)$$

which implies that $s \rightarrow 0$. Since V is lower bounded by zero, s is bounded and then e and \dot{e} in (22) are bounded. Thus, in force of the positive definiteness of K , (26) also implies that $e \rightarrow 0$ [28]. The occurrence of rank deficiencies in J leads to similar considerations as those for the first-order schemes above. End of proof.

Corollary 2. If the desired end-effector location is constant, i.e. $\dot{x}_d = \ddot{x}_d = 0$, the control law (21) can be modified into

$$\ddot{q} = J^T K s \quad (27)$$

which ensures that $\dot{e} \rightarrow 0$ and $\dot{q} \rightarrow 0$ [21].

Proof. The proof is somewhat not as straightforward as for corollary 1. Define the Lyapunov function candidate

$$V = \frac{1}{2} (\dot{q}^T \dot{q} + e^T K \Delta e). \quad (28)$$

Its time derivative along the trajectories of the system (5), with $\dot{x}_d = 0$, under the control (27) results in

$$\dot{V} = -\dot{e}^T K s + \dot{e}^T K \Delta e \quad (29)$$

which, by virtue of (22) becomes

$$\dot{V} = -\dot{e}^T K \dot{e} \leq 0 \quad (30)$$

implying that $\dot{e} \rightarrow 0$, and then $\dot{q} \rightarrow 0$ and $e \rightarrow 0$ [21].

End of proof.

Theorem 6. If x_d belongs to the class of C^2 functions and the quantity z in (23) is norm-bounded, i.e. $\|z\| \leq a$, the control law (27) ensures that s , and then e , can be made arbitrarily small by increasing the minimum eigenvalue of K .

Proof. Define the Lyapunov function candidate as in (24). Its time derivative along the trajectories of the system (19) under the control (27) results in

$$\dot{V} = s^T K^T \dot{z} - s^T K^T J J^T K s. \quad (31)$$

Let λ_J and λ_K denote the minimum eigenvalues of $J J^T$ and K , respectively. By the assumption on $\|z\|$, it follows that

$$\dot{V} \leq \|s\| \lambda_K a - \|s\|^2 \lambda_K^2 \lambda_J \quad (32)$$

which implies that $\dot{V} \leq 0$ as long as

$$\|s\| \geq a / \lambda_K \lambda_J \quad (33)$$

i.e. the sliding vector enters an attractive region containing the sliding surface $s = 0$ which can be made arbitrarily small by increasing λ_K . At steady-state ($\dot{x}_d = \ddot{x}_d = 0$), it is $e = 0$ from corollary 2. Also, notice that the assumption on the

norm-boundedness of x can in turn be relaxed to the assumption that \ddot{x}_d is norm-bounded. This can be explained as follows. Since v is lower bounded by zero and s is bounded in force of (33), then e and \dot{e} in (22) are bounded. This implies that q and \dot{q} are bounded too. Since $\dot{J}(q)$ can be factored as $H(q)\dot{q}$ with $H(q)$ bounded [11], in sum the second and third terms on the right hand side of (23) are bounded, and thus the only requirement is that \ddot{x}_d be bounded.
End of proof.

Similar remarks as for the first-order schemes are in order also for the second-order schemes just presented. In short, the first scheme (17) resembles resolved-acceleration control as in [27]. The second and third schemes both avoid the inversion of J , but the third scheme (27) is to be preferred over the second (21) from the implementation viewpoint since it avoids chattering accelerations \ddot{q} and is less time-consuming. In the light of the velocity/force duality described by (2)/(16), the control law (27) is analogous to applying an elastic/damping force Ks at the end-effector of a simple manipulator having a unitary inertia matrix and no gravity or friction. The minimum eigenvalue λ_{\min} in (27) is upper bounded by the finite sampling rate of the discrete-time algorithm.

Conclusions

A class of computational schemes of robot inverse kinematics have been illustrated which are originated from a closed-loop dynamic reformulation of the problem. First-order schemes which solve for joint displacements and velocities and second-order schemes which solve for joint accelerations too have been formally derived. It is easy to recognize that all the schemes presented do not require any special assumption about the kinematic structure. For the formal derivation of inverse kinematic schemes for redundant manipulators, however, the reader is referred to [14,17,18,19,20] where a so-called task space augmentation strategy is proposed to solve for the redundant d.o.f.'s. Also, due to lack of space, no case studies for practical robot structures have been developed in this work. Extensive numerical results, however, can be found in [14,15,16,17,18,20,23,26,30] which demonstrate the effectiveness of applying the closed-loop computational schemes to solve the inverse kinematic problem for a variety of manipulator architectures.

Acknowledgements

The research described in this work has been supported by the Ministero della Pubblica Istruzione under MPI 40% and 60% funds.

References

- [1] D.L. Pieper, *The Kinematics of Manipulators Under Computer Control*, Ph.D. dissertation, Stanford University, 1968.
- [2] D.E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on Man-Machine Systems*, vol. MMS-10, no. 2, pp. 47-53, 1969.
- [3] A.A. Goldenberg, B. Benhabib, and R.G. Fenton, "A complete generalized solution to the inverse kinematics of robots," *IEEE Journal of Robotics and Automation*, vol. RA-1, no. 1, pp. 14-20, 1985.
- [4] J. Angeles, "On the numerical solution of the inverse kinematic problem," *International Journal of Robotics Research*, vol. 4, no. 2, pp. 21-37, 1985.
- [5] J. Lenarcic, "An efficient numerical approach for calculating the inverse kinematics for robot manipulators," *Robotica*, vol. 3, no. 1, pp. 21-26, 1985.
- [6] L.-W. Tsai and A.P. Morgan, "Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods," *ASME Journal of Mechanism, Transmission, and Automation in Design*, vol. 107, no. 2, pp. 189-200,

- 1985.
- [7] R.P. Paul, Robot Manipulators: Mathematics, Programming and Control. Cambridge, MA: The MIT Press, 1981.
 - [8] C.S.G. Lee and P.R. Chang, "A maximum pipelined CORDIC architecture for inverse kinematic position computation," IEEE Journal of Robotics and Automation, vol. RA-3, no. 5, pp. 445-458, 1987.
 - [9] R. Featherstone, "Calculation of robot joint rates and actuator torques from end effector velocities and applied forces," Mechanism and Machine Theory, vol. 18, no. 3, pp. 193-198, 1983.
 - [10] J.M. Hollerbach and G. Sahar, "Wrist-partitioned, inverse kinematic acceleration and manipulator dynamics," International Journal of Robotics Research, vol. 2, no. 4, pp. 61-76, 1983.
 - [11] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," IEEE Journal of Robotics and Automation, vol. 3, no. 1, pp. 43-53, 1987.
 - [12] A. Balestrino, G. De Maria, and L. Sciavicco, "Robust control of robotic manipulators," 9th IFAC World Congress, Budapest, Hungary, July 1984.
 - [13] W.A. Wolovich and H. Elliott, "A computational technique for inverse kinematics," 23rd IEEE Conference on Decision and Control, Las Vegas, NV, Dec. 1984.
 - [14] B. Siciliano, Solution Algorithms to the Inverse Kinematic Problem for Manipulation Robots (in Italian), Ph.D. dissertation, Università di Napoli, Dec. 1986.
 - [15] A. Balestrino, G. De Maria, L. Sciavicco, and B. Siciliano, "An algorithmic approach to coordinate transformation for robotic manipulators," Advanced Robotics, vol. 2, no. 4, pp. 327-344, 1987.
 - [16] L. Sciavicco and B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-16, no. 4, pp. 550-559, 1986.
 - [17] L. Sciavicco and B. Siciliano, "An inverse kinematic solution algorithm for dexterous redundant manipulators," 3rd International Conference on Advanced Robotics, Versailles, France, Oct. 1987.
 - [18] L. Sciavicco, B. Siciliano, and P. Chiacchio, "On the use of redundancy in robot kinematic control," American Control Conference, Atlanta, GA, June 1988.
 - [19] L. Sciavicco and B. Siciliano, "On the solution of inverse kinematics of redundant manipulators," NATO Advanced Research Workshop: Robots With Redundancy, Salò, Italy, June/July 1988.
 - [20] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," IEEE Journal of Robotics and Automation, vol. 4, no. 4, pp. 303-310, 1988.
 - [21] J.-J.E. Slotine and D.R. Yoerger, "A rule-based inverse kinematics algorithm for redundant manipulators," International Journal of Robotics and Automation, vol. 2, no. 2, pp. 86-89, 1987.
 - [22] H. Das, J.-J.E. Slotine, and T.B. Sheridan, "Inverse kinematic algorithms for redundant systems," IEEE International Conference on Robotics and Automation, Philadelphia, PA, Apr. 1988.
 - [23] P. Chiacchio and B. Siciliano, "Achieving singularity robustness: An inverse kinematic solution algorithm for robot control," IEE Control Engineering Series 36 Robot Control: Theory and Applications, K. Warwick and A. Pugh (Editors), London, UK: Peter Peregrinus Ltd., ch. 16, pp. 149-156, 1988.
 - [24] W.A. Wolovich and K.F. Flueckiger, "Inverse kinematic-based robot control," Workshop on Space Telerobotics, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, Jan. 1987.
 - [25] R.J. Vaccaro and S.D. Hill, "A joint-space command generator for Cartesian control of robotic manipulators," IEEE Journal of Robotics and Automation, vol. RA-4, no. 1, pp. 70-76, 1988.
 - [26] L. Sciavicco and B. Siciliano, "A computational technique for solving robot end-effector trajectories into joint trajectories," American Control Conference, Atlanta, GA, June 1988.
 - [27] J.Y.S. Luh, M.W. Walker, and R.P.C. Paul, "Resolved-acceleration control of

- mechanical manipulators," IEEE Transactions on Automatic Control, vol. AC-25, no. 3, pp. 468-474, 1980.
- [28] J.-J.E. Slotine and S.S. Sastry, "Tracking control of nonlinear systems using sliding surfaces with application to robot manipulators," International Journal of Control, vol. 38, no. 2, pp. 465-492, 1983.
- [29] C.A. Klein and C.-H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, no. 3, pp. 245-250, 1983.
- [30] L. Sciavicco and B. Siciliano, "An algorithm to compute the reachable workspace for 2R and 3R planar pair mechanical arms," IEEE International Conference on Robotics and Automation, Philadelphia, PA, Apr. 1988.

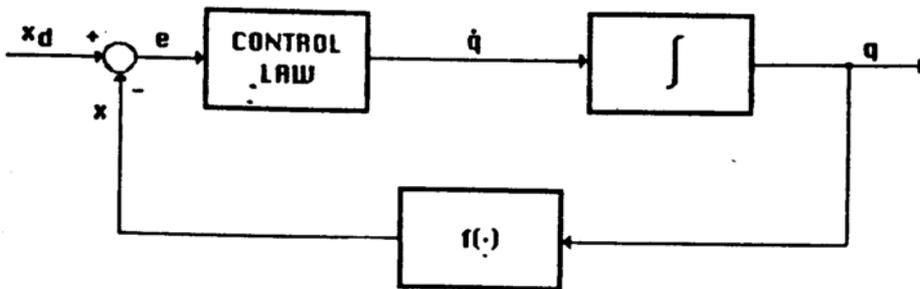


Fig. 1 - Block diagram of a 1st-order closed-loop inverse kinematics scheme.

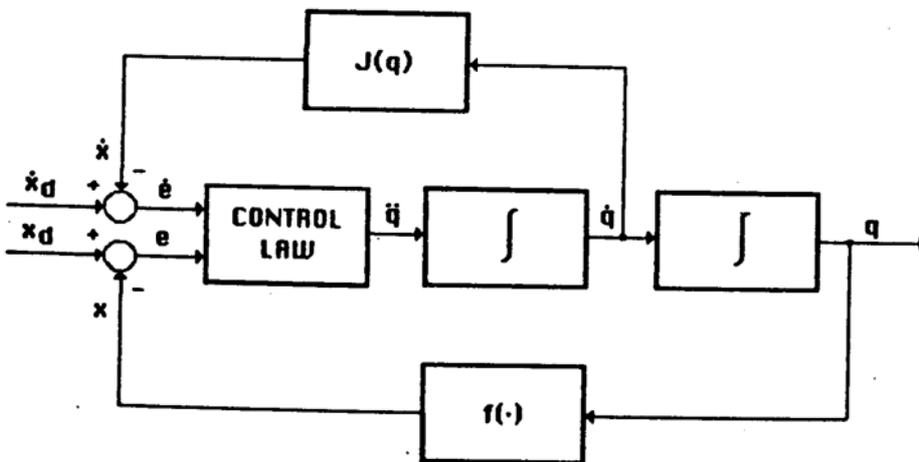


Fig. 2 - Block diagram of a 2nd-order closed-loop inverse kinematics scheme.