A Complete Generalized Solution to the Inverse Kinematics of Robots

ANDREW A. GOLDENBERG, MEMBER, IEEE, B. BENHABIB, AND ROBERT G. FENTON

Abstract—**The kinematic transformation between task space and joint configuration coordinates is nonlinear and configuration dependent. A solution to the inverse kinematics is a vector of joint configuration coordinates that corresponds to a set of task space coordinates. For a class of robots closed form solutions always exist, but constraints on joint displacements cannot be systematically incorporated in the process of obtaining a solution. An iterative solution is presented that is suitable for any class of robots having rotary or prismatic joints, with any arbitrary number of degrees of freedom, including both standard and kinematically redundant robots. The solution can be obtained subject to specified constraints and based on certain performance criteria. The solution is based on a new rapidly convergent constrained nonlinear optimization algorithm which uses a modified Newton–Raphson technique for solving a system nonlinear equations. The algorithm is illustrated using as an example a kinematically redundant robot.**

I. INTRODUCTION

FOR ADVANCED CONTROL of robot manipulators, necessary capabilities are offline programming of the end effector path and control in terms of Cartesian coordinates. Although control of Cartesian trajectory of the end effector is a basic requirement for many industrial applications, most robot manipulators lack this ability.

The "inverse kinematics" control of a robot manipulator requires the transformation of end effector Cartesian task space coordinates into corresponding joint configuration space coordinates. The common approach for solving this problem is to obtain a closed-form solution to the inverse transformation [1]. However, only certain classes of robots (e.g., spherical wrist manipulators, such as the PUMA 560 robot) allow closed-form inverse kinematics solutions. The problem becomes more critical for kinematically redundant robots, for which the number of degrees of freedom (DOF) exceeds the required six coordinates necessary to attain arbitrary locations in the three-dimensional work space [2], [4].

A second approach for solving the inverse transformation problem of n DOF robots is based on the use of iterative procedures for solving a system of nonlinear equations [3], [4], [5], [6]. Alternatively, the kinematics model of the manipulator can be divided into subsystems such that an iterative procedure can be applied to determine some of the joint variables, while the other variables are obtained by a closed-form solution [7], [8]. The method presented herein considers the kinematic model as a whole and determines all joint variables using a rapidly converging iterative procedure.

In a three-dimensional space a manipulator must have at least six DOF in order to be able to attain any arbitrary end effector position and orientation. If the manipulator has six DOF, a system of six nonlinear equations has to be solved for the six joint variables. However, if the manipulator has more than six DOF, the system of six nonlinear equations is underdetermined. For such cases an optimization procedure may be used to determine the best set of solutions subject to a given objective function [4], [9].

A number of iterative methods for the analysis of spatial mechanisms presented in earlier works use matrix algebra to formulate the kinematic relations [11], [24].

In literature the most common algorithms proposed to solve the nonlinear kinematic equations use Newton's methods based on simultaneous successive linear interpolation of nonlinear equations [12], [13]. Since Newton-like algorithms are considered "local methods," which require an initial close estimate to the exact solution, a proper step size control is necessary to avoid divergence due to a "bad" initial estimate [14].

This work consists of five sections. Section II presents a general formulation of the inverse kinematics problem of n DOF robot arms. A review of solutions of nonlinear equations is presented in Section III. An efficient and general iterative technique is discussed in Section IV for both kinematically determinate (standard) and redundant robots. The technique is based on a modified Newton–Raphson method for solving nonlinear equations. A numerical example of a seven degree of freedom robot illustrates the method in Section V.

II. PROBLEM FORMULATION

The motion of an articulated arm (robot) can be analysed by assigning certain coordinate frames to each link [15], in order to obtain a transformation that relates joint to task space coordinates. The task space coordinates are the position and orientation of the end effector with respect to a reference (base) frame. The relative position and orientation of two link coordinate frames are expressed in terms of transformation matrices $A_i \in R^{4\times4}$ [1] defined as follows

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad i=1, n \tag{1}$$

where a_i, α_i, d_i, θ_i are the link length, twist, distance, and joint

Manuscript received September 26, 1984; revised February 1985. This work was supported in part by the National Science and Engineering Research Council under Grant A-4664.

The authors are with the Robotics and Automation Laboratory, Department of Mechanical Engineering, University of Toronto, Toronto, ON, Canada.

variable, respectively. For simplicity we assume that only rotary joints are considered. Using (1), the transformation T_n representing the position and orientation of the end effector with respect to the base frame is

$$T_n(q)=\prod_{i=1}^{n} A_i(q)=\begin{pmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{2}$$

where $\boldsymbol{n}$, $\boldsymbol{o}$, $\boldsymbol{a}$ are the orientation vectors, $\boldsymbol{p}$ is the position vector, and $\boldsymbol{q}$ is the configuration space vector consisting of joint variables θ_i, $\boldsymbol{q}$ *member* R^n.

The inverse kinematics problem is the determination of a vector $\boldsymbol{q}^*$ which corresponds to a given end effector target transformation $T_n{}^t$. The approach, first introduced in [5], is based on a residual vector definition $\boldsymbol{r}$ *member* R^6 which represents the residual position and orientation between $T_n{}^t$ and the actual (present) end effector transformation $T_n{}^a$. Clearly $\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{q})$ is defined as follows

$$\boldsymbol{r}=(r_x r_y r_z r_\phi r_\theta r_\psi) \tag{3}$$

where $(r_x r_y r_z)$ and $(r_\phi r_\theta r_\phi)$ represent the residual position and the residual orientation, respectively. The elements of residual position are defined as [16]

$$r_x=\boldsymbol{n}^a \cdot (\boldsymbol{p}^t-\boldsymbol{p}^a)$$

$$r_y=\boldsymbol{o}^a \cdot (\boldsymbol{p}^t-\boldsymbol{p}^a)$$

$$r_z=\boldsymbol{a}^a \cdot (\boldsymbol{p}^t-\boldsymbol{p}^a) \tag{4}$$

where $\boldsymbol{p}^t$ and $\boldsymbol{p}^a$ are the position vector of the target and the actual frames with respect to the base frame, respectively, and $(\boldsymbol{n}^a\boldsymbol{o}^a\boldsymbol{a}^a)$ are the orientation unit vectors of the actual end effector frame with respect to the base frame.

For the elements of the residual orientation vector one can use any suitable set of rotation angles with a predefined sequence of rotations. For example, for Euler angles the elements are defined as

$$r_\phi=a\ \tan\ 2[(\boldsymbol{o}^a \cdot \boldsymbol{a}^t),\ (\boldsymbol{a}^a \cdot \boldsymbol{a}^t)] \qquad \mathrm{r}_\phi=r_\phi+180^\circ$$

$$r_\theta=a\ \tan\ 2[((\boldsymbol{n}^a \cdot \boldsymbol{a}^t)\ \cos\ r_\phi+(\boldsymbol{o}^a \cdot \boldsymbol{a}^t)\ \sin\ r_\phi),\ (\boldsymbol{a}^a \cdot \boldsymbol{a}^t)]$$

$$r_\psi=a\ \tan\ 2[(-(\boldsymbol{n}^a \cdot \boldsymbol{n}^t)\ \sin\ r_\phi+(\boldsymbol{o}^a \cdot \boldsymbol{n}^t)\ \cos\ r_\phi),$$

$$(-(\boldsymbol{n}^a \cdot \boldsymbol{o}^t)\ \sin\ r_\phi+(\boldsymbol{o}^a \cdot \boldsymbol{o}^t)\ \cos\ r_\phi)] \tag{5}$$

For yaw-pitch-roll angles the elements are defined as

$$r_\phi=a\ \tan\ 2[(\boldsymbol{o}^a \cdot \boldsymbol{n}^t),\ (\boldsymbol{n}^a \cdot \boldsymbol{n}^t)] \qquad r_\phi=r_\phi+180^\circ$$

$$r_\theta=a\ \tan\ 2[-(\boldsymbol{a}^a \cdot \boldsymbol{n}^t),\ ((\boldsymbol{n}^a \cdot \boldsymbol{n}^t)\ \cos\ r_\phi+(\boldsymbol{o}^a \cdot \boldsymbol{n}^t)\ \sin\ r_\phi)]$$

$$r_\psi=a\ \tan\ 2[((\boldsymbol{n}^a \cdot \boldsymbol{a}^t)\ \sin\ r_\phi-(\boldsymbol{o}^a \cdot \boldsymbol{a}^t)\ \cos\ r_\phi),$$

$$(-(\boldsymbol{n}^a \cdot \boldsymbol{o}^t)\ \sin\ r_\phi+(\boldsymbol{o}^a \cdot \boldsymbol{o}^t)\ \cos\ r_\phi)] \tag{6}$$

For a set of $x - y - z$ rotation axes the elements are

$$r_\phi=1/2(\boldsymbol{a}^a \cdot \boldsymbol{o}^t-\boldsymbol{a}^t \cdot \boldsymbol{o}^a)$$

$$r_\theta=1/2(\boldsymbol{n}^a \cdot \boldsymbol{a}^t-\boldsymbol{n}^t \cdot \boldsymbol{a}^a)$$

$$r_\psi=1/2(\boldsymbol{o}^a \cdot \boldsymbol{n}^t-\boldsymbol{o}^t \cdot \boldsymbol{n}^a) \tag{7}$$

The solution to the target transformation $\boldsymbol{q}^*$ is obtained when the actual end effector transformation $T_n{}^a$ is coincident with the target transformation $T_n{}^t$. Such a solution exists if and only if $\boldsymbol{q} = \boldsymbol{q}^*$, i.e.

$$\boldsymbol{r}(\boldsymbol{q}^*)=0 \tag{8}$$

The algorithm presented herein generates $\boldsymbol{q}^*$ such that (8) holds with $\boldsymbol{q}^a$ (actual configuration) being the initial estimate of $\boldsymbol{q}^*$.

III. NUMERICAL METHODS FOR NONLINEAR KINEMATIC EQUATIONS

A. Determination of the Jacobian

The most common methods for solving nonlinear equations approximate the nonlinear system by a linear one such as (8), then solve the problem iteratively [17], [18], [19]. For example, the Newton–Raphson method [18] takes into account only the first order terms of the Taylor series expansion of $\boldsymbol{r}(\boldsymbol{q})$ in (3). A solution for this system of equations is determined using, in an iterative manner, the following approximation [12]

$$\boldsymbol{q}^{(k+1)}=\boldsymbol{q}^{(k)}+\boldsymbol{\delta}^{(k)} \tag{9}$$

where $\boldsymbol{\delta}^{(k)}$ solves the linear system and

$$r_j(\boldsymbol{q}^{(k)})+\sum_{i=1}^{n} J_{ji}{}^{(k)}\boldsymbol{\delta}_i{}^{(k)}=0, \qquad j=1,\ 6 \tag{10}$$

where J_{ji} is defined as

$$J_{ji}{}^{(k)}=[\partial r_j/\partial q_i]_{q=q^{(k)}}, \qquad i=1,\ n;\ j=1,\ 6. \tag{11}$$

The function r_j is continuously differentiable and J_{ji} is nonsingular in the neighborhood of the solution $\boldsymbol{q} = \boldsymbol{q}^*$.

The manipulator Jacobian matrix, which is required to solve (10), has to be determined analytically or numerically. Analytical expressions for the Jacobian matrix corresponding to the residual vector $\boldsymbol{r}(\boldsymbol{q})$ given by (4) and (7) are shown in Appendix I [10]. Since the computations are lengthy, the user may decide either to compute the Jacobian only after every m iterations, or approximate it using the following function values [19]

$$[J]^{(k+1)}=[J]^{(k)}+[D]^{(k)} \tag{12}$$

where

$$[D]^{(k)}=\frac{\boldsymbol{r}^{(k)}(\boldsymbol{z}^{(k)})^T}{(\boldsymbol{z}^{(k)})^T\boldsymbol{\delta}^{(k)}} \tag{13}$$

and vector $\boldsymbol{z}^{(k)}$ which is chosen to be orthogonal to $\boldsymbol{\delta}^{(1)} \cdots \boldsymbol{\delta}^{(k-1)}$ is determined using the Gram–Schmidt orthogonalization process, so that

$$[D]^{(k)}(\boldsymbol{\delta}^{(l)})=0, \qquad 1\leq k-l\leq n. \tag{14}$$

Following the determination of the Jacobian matrix, the next task is to invert it for solving (10).

B. Inverse of the Jacobian

Case 1: For manipulator arms with $n < 6$ DOF, the $(6 \times n)$ Jacobian matrix is nonsquare, so that an inverse can be only obtained using generalized inverses [17] as follows

$$G=[J]^{+}=([J]^{T}[J])^{-1}[J]^{T} \tag{15}$$

The inverse J^{+} is unique, minimizes the Euclidean norm of $\boldsymbol{q}$ and satisfies the following relations [17]

$$JJ^{+}J=J \tag{16a}$$

$$J^{+}JJ^{+}=J^{+} \tag{16b}$$

$$(J^{+}J)^{T}=J^{+}J \tag{17a}$$

$$(JJ^{+})^{T}=JJ^{+} \tag{17b}$$

Case 2: For manipulator arms with $n = 6$ DOF, the Jacobian matrix is square and it can be inverted if it is nonsingular. Then $J^{+} = J^{-1}$ (the ordinary inverse) in (15).

Case 3: For kinematically redundant robots with $n > 6$ DOF, the $(6 \times n)$ Jacobian matrix is nonsquare and its inverse can be obtained using pseudoinverses [4]. Although, the pseudoinverse of the $(6 \times n)$ Jacobian matrix, $n > 6$, can easily be obtained, it can only be used if the objective function is the minimum Euclidean norm. For more general objective functions a new type of Jacobian inversion technique is presented in the next section.

IV. SOLUTION TO THE INVERSE KINEMATICS

A. The Modified Newton–Raphson Method for $n > 6$ DOF

The purpose of this method is to solve the nonlinear system of equations (8) subject to arbitrary objective functions (optimization criteria). Since $n > 6$ in (10), the $(6 \times n)$ Jacobian matrix can be partitioned into a (6×6) and a $(6 \times (n - 6))$ matrix as in [4],

$$(\boldsymbol{r}(\boldsymbol{q}^{(k)}))+[J^{R}]^{(k)}(\boldsymbol{\delta}^{R})^{(k)}+[J^{A}]^{(k)}(\boldsymbol{\delta}^{A})^{(k)}=0 \tag{18}$$

where $[J^{R}]$ is the (6×6) reduced Jacobian, evaluated at $\boldsymbol{q} = \boldsymbol{q}^{(k)}$, $[J^{A}]$ is the $(6 \times (n - 6))$ matrix corresponding to $(\boldsymbol{\delta}^{A})$ obtained by excluding $(n - 6)$ columns from $[J]$, $(\boldsymbol{\delta}^{R})^{(k)} = (\boldsymbol{q}^{R})^{(k+1)} - (\boldsymbol{q}^{R})^{(k)}$ are the six joint correction variables, and $(\boldsymbol{\delta}^{A})^{(k)} = (\boldsymbol{q}^{A})^{(k+1)} - (\boldsymbol{q}^{A})^{(k)}$ are the $(n - 6)$ free joint correction variables.

The only condition in choosing the free variables $\boldsymbol{\delta}^{A}$ is to guarantee the invertibility of the reduced Jacobian matrix $[J^{R}]$ which is necessary to obtain the following equations,

$$(\boldsymbol{\delta}^{R})=(\boldsymbol{a})-[b](\boldsymbol{\delta}^{A}) \tag{19}$$

where

$$(\boldsymbol{a})=[J^{R}]^{-1}(\boldsymbol{r})$$

$$[b]=[J^{R}]^{-1}[J^{A}]. \tag{20}$$

Values of $\boldsymbol{\delta}^{R}$ obtained from (19), corresponding to certain $\boldsymbol{\delta}^{A}$ values, can now be substituted into any objective function of the form

$$\min\ Z=f(\boldsymbol{q}) \tag{21}$$

and the optimum $\boldsymbol{\delta}^{A}$ can be determined using an $(n - 6)$ dimensional optimization routine.

B. Constraints

The system of nonlinear equations (8), $\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{q})$, is always subject to constraints imposed on the solution such as

$$\boldsymbol{q}^{l}\leq\boldsymbol{q}\leq\boldsymbol{q}^{u} \tag{22}$$

where $\boldsymbol{q}^{l}$ and $\boldsymbol{q}^{u}$ are the lower and upper limits on the joint variable displacements q_i, $i = 1, n$.

In general, for Newton–Raphson methods, when the generalized inverse is used to obtain the inverse Jacobian matrix, no control is exercised on the computed correction vector $\boldsymbol{\delta}^{(k)}$, at iteration step k. It is possible, that $\boldsymbol{q}^{(k+1)}$ determined from $\boldsymbol{\delta}^{(k)}$, (9) does not satisfy (22) and the algorithm converges to a (nonfeasible) solution outside the permissible joint ranges of the robot (outside the work space). Under such circumstances there may be two strategies to prevent convergence to a nonfeasible solution: 1) Check the solution $\boldsymbol{q}^{(k+1)}$ against the limits and correct those variable values to their nearest limit which are out-of-range, or 2) Reduce the step size of Newton–Raphson iteration, $\|\boldsymbol{\delta}_{NR}{}^{(k)}\|$, if a joint exceeds its limits. This strategy will be discussed in detail in Section IV-C.

For the modified Newton's method presented in Section IV-A, for $n > 6$, joint limits on the free variables, $\boldsymbol{\delta}^{A}$, are determined from constraints equation (22) specified for all joints prior to every algorithm step. Let $q_i{}^{(k)}$ be the current value of the ith joint variable. Then, using (19) and (22), the constraints become

$$q_i{}^{l}-q_i{}^{(k)}=\delta_i{}^{l}\leq\delta_i{}^{(k)}\leq\delta_i{}^{u}=q_i{}^{u}-q_i{}^{(k)} \tag{23}$$

and

$$(\boldsymbol{\delta}^{R})^{l}\leq[-(\boldsymbol{a})^{(k)}-[b]^{(k)}(\boldsymbol{\delta}^{A})^{(k)}]\leq(\boldsymbol{\delta}^{R})^{u}. \tag{24}$$

If the total number of variables is $n = 7$, the constraint equations will simplify as follows,

$$(\boldsymbol{\delta}^{A})^{L}\leq(\boldsymbol{\delta}^{A})^{(k)}\leq(\boldsymbol{\delta}^{A})^{U} \tag{25}$$

where

$$(\boldsymbol{\delta}^{A})^{L}=\max\ [(\boldsymbol{\delta}^{A})^{l},\ (a_j-(\boldsymbol{\delta}_j{}^{R})^{l})/b_j;\ j=1,\ 6]$$

$$(\boldsymbol{\delta}^{A})^{U}=\min\ [(\delta^{A}),\ (a_j-(\boldsymbol{\delta}_j{}^{R})^{u})/b_j;\ j=1,\ 6] \tag{26}$$

where a_j and b_j are the elements of $(\boldsymbol{a})$ and $(\boldsymbol{b})$, respectively.

C. Step Size Control

The classical Newton–Raphson iteration often fails to converge when the initial estimate $\boldsymbol{q}^{(0)}$ is not sufficiently close, in the Euclidean sense, to the solution of the system (8). A number of useful modifications have been developed to Newton's method, which provide reliable convergence, even when a close initial estimate is not available [17], [18], [19]. In [18], Eq. (9) is replaced by the expression

$$q^{(k+1)} = q^{(k)} + \lambda^{(k)}\delta^{(k)} \tag{27}$$

where $\lambda^{(k)}$ is a positive number chosen to satisfy the following inequality,

$$G(q^{(k+1)}) < G(q^{(k)}) \tag{28}$$

where

$$G(q) = \sum_{j=1}^{6} [r_j(q)]^2 \tag{29}$$

Equation (28) is expected to hold [18], since if $[J]^{(k)}$ is nonsingular, then

$$\left[\frac{\partial}{\partial\lambda} G(q^{(k)} + \lambda\delta^{(k)})\right]_{\lambda=0} = -2G(q^{(k)}) < 0 \tag{30}$$

unless $q^{(k)}$ is already a solution of (8).

A different modification to the classical Newton–Raphson iteration, which is known as the Levenberg–Marquardt iteration [20], [21], replaces (9) by

$$q^{(k+1)} = q^{(k)} + \eta^{(k)} \tag{31}$$

where η solves the linear system

$$([J]^T[J] + \nu^{(k)}[I])\eta^{(k)} + [J]^T(r(q^{(k)})) = 0 \tag{32}$$

where $\nu^{(k)}$ is a positive number.

A common strategy to both methods is to retain direction, but to restrict step length if necessary. If (28) is satisfied, the full Newton–Raphson correction $\delta_{NR}{}^{(k)}$ is maintained by setting $\lambda^{(k)} = 1$ in (27) and $\nu^{(k)} = 0$ in (32).

As mentioned in Section IV-B, step size control can be effectively used to prevent nonfeasible solutions. If such a situation is encountered during the iterative procedure, the step size is restricted using a new algorithm presented in Section IV-D.

It is noticed that the iterative procedure of the Levenberg–Marquardt method (see Appendix II) [20] is a least squares solution, and can not be easily applied for $n > 6$ DOF robots if the modified Newton's method of Section IV-A is used. On the other hand, the step size restriction procedure using (27), as it will be explained in Section IV-D, is more suitable for the new algorithm of Section IV-A, since the correction $\delta^{(k)}$ is restricted only after the full Newton–Raphson correction $\delta_{NR}{}^{(k)}$ is computed.

D. Algorithm of the Iterative Procedure

The algorithm proposed in this paper seeks a solution q^* to the problem defined by (8) subject to the constraints of (22). This algorithm generalizes the algorithm presented earlier for $n > 6$ DOF (redundant) robots [6]. In Steps 1 and 2 the manipulator Jacobian matrix is computed and inverted to obtain the full Newton–Raphson correction vector $\delta_{NR}{}^{(k)}$. The user has the option to use either generalized inverses or the inversion procedure described in Section IV-A. In Steps 3–6 the step length, defined as $\|\delta^{(k)}\|$, may be restricted to prevent nonfeasible solutions. The constant $\lambda^{(k)}$ in (27) is computed as a multiplier of the Newton–Raphson correction vector $\delta_{NR}{}^{(k)}$ that was determined in the earlier steps. In Step 7 the value of $G(q^{(k+1)})$ from (29) is compared with a specified threshold value ϵ for convergence check. The algorithm consists of the following step-by-step procedure.

Step 1: Determine the Jacobian matrix $[J]$ (see (11)) corresponding to the residual vector r (see (3)) at $q^{(k)}$.

Step 2: Evaluate the full Newton–Raphson correction $\delta_{NR}{}^{(k)}$ as follows.

Case 1 (for arbitrary n DOF):

a) using generalized inverses (see (15)–(17)) obtain $[J]^+$

b) determine the correction vector $\delta_{NR}{}^{(k)} = \delta^{(k)}$ (see (10)).

Case 2 (for $n > 6$ DOF):

a) partition the Jacobian matrix into $[J^R]$ and $[J^A]$ (see (18))

b) determine the constants (a) and $[b]$ (see (20))

c) determine the feasible domain of the free variables (δ^A) (see (23)–(26))

d) determine an optimal solution $(\delta^A)^*$ for a given objective function (see (21))

e) determine the optimal correction vector $\delta_{NR}{}^{(k)}$ (see (19)).

Step 3: Evaluate the gradient $(g)^{(k)}$ as follows [22],

$$(g_j)^{(k)} = \left[\frac{\partial}{\partial q_i} G(q)\right]_{q=q^{(k)}} = 2\sum_{i=1}^{n} [J_{ij} r_j(q)]_{q=q^{(k)}} \tag{33}$$

Step 4: Check if the solution is converging to a local minimum using

$$G(q^{(k)}) \geq \xi \|g^{(k)}\| \tag{34}$$

where ξ is usually set to an overestimate of the distance $q^{(0)}$ to the solution q^* [22]. If (34) holds, i.e., there is convergence to a local minimum, then terminate the process, else proceed to the next step.

Step 5: Restrict the step size by the following procedure, if necessary.

1) If

$$\|\delta_{NR}{}^{(k)}\| \leq \Delta^{(k)} \tag{35}$$

and

$$\mu\|g^{(k)}\| \geq \Delta^{(k)} \tag{36}$$

then set

$$\delta^{(k)} = \delta_{NR}{}^{(k)}$$

where $\Delta^{(k)}$ is the maximum allowable step size initially set to $\Delta^{(0)} = \max$ (DSTEP, min [DMAX, $\mu\|g\|$]); and $\mu = \|g^{(k)}\|/\|Jg^{(k)}\|^2$; DSTEP is chosen to avoid computer roundoff errors.

2) If equality (36) holds and (35) does not, then

$$\delta^{(k)} = \Delta^{(k)} g^{(k)}/\|g^{(k)}\| \tag{37}$$

3) If both inequalities (35) and (36) do not hold,

$$\boldsymbol{\delta}^{(k)}=(1-\gamma)\mu \boldsymbol{g}^{(k)}+\gamma \boldsymbol{\delta}_{NR}{}^{(k)} \tag{38}$$

such that the positive number γ satisfies

$$\|(1-\gamma)\mu \boldsymbol{g}^{(k)}+\gamma \boldsymbol{\delta}_{NR}{}^{(k)}\|=\Delta^{(k)}. \tag{39}$$

Step 6: Verify whether both inequalities (23) and (28) hold (check (28) only if Case 2 of algorithm Step 2 is used).

1) If they do not hold, reduce $\Delta^{(k)}$ as follows and return to Step 5,

$$\Delta^{(k)}=\max\ (\beta\Delta^{(k)},\ \text{DSTEP}) \tag{40}$$

where $0.0 < \beta < 1.0$.

2) If they hold, then set

$$\boldsymbol{q}^{(k+1)}=\boldsymbol{q}^{(k)}+\boldsymbol{\delta}^{(k)} \tag{9}$$

Step 7: Verify the convergence to final solution as follows,

$$G(\boldsymbol{q}^{(k+1)})<\epsilon, \tag{41}$$

where $\epsilon > 0.0$ is a threshold value supplied by the user.

1) If (41) holds, then $\boldsymbol{q}^* = \boldsymbol{q}^{(k+1)}$ is the optimal solution of (8).

2) If (41) does not hold, let $k = k + 1$ and return to algorithm Step 1.

The proofs of convergence of the algorithm described above can be found in [18].

V. A NUMERICAL EXAMPLE

A seven DOF, all rotary joints, two elbow robot with a maximum reach of 3.0 is given. The link lengths (defined according to [10]) are $a_1 = 0$, $a_2 = 1$, $a_3 = 1$, $a_4 = 1$, $a_5 = 0$, $a_6 = 0$, $a_7 = 0$. The residual vector $\boldsymbol{r}(\boldsymbol{q})$, expressed by (4) and (7), and the corresponding Jacobian matrix (using the formulation given in Appendix I) are utilized in the algorithm described in Section IV-D.

The robot is positioned initially at $\theta_i{}^{(0)}$ corresponding to task space $T_7{}^{(0)}$, and it is required to relocate its end effector to a task space $T_7{}^{(1)}$, where

$$\theta_i{}^{(0)}=(0.2172,\ 0.3914,\ 0.1651,\ 0.4032,\ 1.0723,\ -0.4137,\ 0.0601) \tag{42}$$

$$T_7{}^{(0)}=\begin{bmatrix} 0.522 & -0.467 & 0.714 & 1.688 \\ -0.821 & -0.047 & 0.569 & 0.373 \\ -0.232 & -0.883 & -0.408 & 2.347 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \tag{43}$$

$$T_7{}^{(1)}=\begin{bmatrix} 0.943 & 0.207 & 0.261 & 2.231 \\ -0.292 & 0.894 & 0.352 & 0.749 \\ -0.160 & -0.395 & 0.899 & 1.923 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \tag{44}$$

The optimization criterion used in this example is a weighted average of three criteria and it is defined as

$$\min\ Z=0.5M_1+0.3M_2+0.2M_3 \tag{45}$$

where

$$M_1=\max\ [\tau_i(\Delta\theta_i);\ i=1,\ 7] \tag{46}$$

TABLE I
KINEMATIC VARIABLES

Joint	1	2	3	4	5	6	7
Maximum Velocity (1/s)	2	2	2	2	10	10	10
Maximum Acceleration ($1/s^2$)	10	10	10	10	25	25	25
Lower Limit (rad)	$-\pi$	$-\pi$	$-\pi$	$-\pi$	$-\pi$	$-\pi$	$-\pi$
Upper Limit (rad)	π	π	π	π	π	π	π
Mid-Range (rad)	0	0	0	0	0	0	0

$$M_2=\sum_{i=1}^{7}(\Delta\theta_i)^2 \tag{47}$$

$$M_3=\sum_{i=1}^{7}(\theta_i-\theta_i{}^M)^2 \tag{48}$$

where τ_i is the motion time of ith joint, and $\theta_i{}^M$ is the operational midrange value of the joint angle of the ith joint.

The kinematic variables corresponding to each of the joints are shown in Table I. The optimal solution obtained for the objective function (45) is

$$\theta_i{}^{(1)}=(0.3533,\ 0.2841,\ 0.9001,\ 0.0966,\ -0.8976,\ -0.2368,\ 0.9067) \tag{49}$$

where

$$\epsilon=1\times 10^{-10} \qquad \mu^{(0)}=0.5$$

$$\text{DSTEP}=0.00001 \qquad \text{DMAX}=20.0$$

$$\Delta^{(0)}=0.9967$$

and the number of iterations required for this calculation was 13.

VI. CONCLUSIONS

This paper presents a complete generalized solution to the inverse kinematics of robots with arbitrary number of degrees of freedom using the concept of residuals. The solution is robot independent and is obtained using an iterative procedure.

The procedure is a modified Newton–Raphson algorithm for which the step length is only restricted to avoid nonfeasible solutions. It is however important to define a sufficiently large initial maximum allowable step length $\Delta^{(0)}$. The procedure also considers constraints on the variables as well as an objective function to be minimized. The procedure requires that the Jacobian is either analytically or numerically computed. In the case of analytically determined Jacobian the rate of convergence is considerably improved.

APPENDIX I
THE MANIPULATOR JACOBIAN

Each column of the manipulator Jacobian matrix $[J]$ can be obtained from $\partial T_n/\partial\theta_i$. In order to determine the columns of the $(6 \times n)$ Jacobian matrix with respect to the end effector frame, the ${}^{i-1}T_n$, $i = 1, n$ transformations have to be developed as follows

$$ {}^{i-1}T_n = (A_i A_{i+1} \cdots A_n), \qquad i=1, \cdots, n $$

where A_i are the (4×4) transformation matrices between frames $(i - 1)$ and (i).

The Jacobian matrix is denoted as

$$ [J] = \begin{bmatrix} d_{1_x} & d_{2_x} & & d_{n_x} \\ d_{1_y} & d_{2_y} & & d_{n_y} \\ d_{1_z} & d_{2_z} & \cdots & d_{n_z} \\ \delta_{1_x} & \delta_{2_x} & & \delta_{n_x} \\ \delta_{1_y} & \delta_{2_y} & & \delta_{n_y} \\ \delta_{1_z} & \delta_{2_z} & & \delta_{n_z} \end{bmatrix} \tag{A1} $$

where, if the joint is revolute,

$$ \begin{aligned} \boldsymbol{d}_i = (-n_x p_y + n_y p_x)\boldsymbol{i} + (-o_x p_y + o_y p_x)\boldsymbol{j} \\ + (-a_x p_y + a_y p_x)\boldsymbol{k} \end{aligned} \tag{A2} $$

$$ \boldsymbol{\delta}_i = n_z\boldsymbol{i} + o_z\boldsymbol{j} + a_z\boldsymbol{k} \tag{A3} $$

and if the joint is prismaic

$$ \boldsymbol{d}_i = n_z\boldsymbol{i} + o_z\boldsymbol{j} + a_z\boldsymbol{k} \tag{A4} $$

$$ \boldsymbol{\delta}_i = 0\boldsymbol{i} + 0\boldsymbol{j} + 0\boldsymbol{k} \tag{A5} $$

where $\boldsymbol{n}$, $\boldsymbol{o}$, $\boldsymbol{a}$, and $\boldsymbol{p}$ are the column vectors of ${}^{i-1}T_n$.

APPENDIX II
THE LEVENBERG–MARQUARDT ITERATIVE PROCEDURE

The algorithm of the Levenberg–Marquardt procedure given by (31)–(32) solves the nonlinear set of equations (3)

$$ \boldsymbol{r} = (r_x, r_y, r_z, r_\phi, r_\theta, r_\psi) = \boldsymbol{r}(\boldsymbol{q}) \tag{3$'$} $$

$$ \boldsymbol{q}^{(k+1)} = \boldsymbol{q}^{(k)} + \boldsymbol{\eta}^{(k)} \tag{31$'$} $$

where $\boldsymbol{\eta}$ solves the linear system

$$ ([J]^T[J] + \nu^{(k)}[I])\boldsymbol{\eta}^{(k)} + [J]^T(\boldsymbol{r}(\boldsymbol{q}^{(k)})) = 0.0. \tag{32$'$} $$

The iterative procedure is as follows.

1) Normalize the Jacobian matrix $[J]$ and the gradient $(\boldsymbol{g})$, computed at $\boldsymbol{q} = \boldsymbol{q}^{(k)}$, as

$$ \hat{J}_{ij} = = \left(\frac{J_{ij}}{\sqrt{J_{ij}}\ \ \sqrt{J_{ij}}} \right) \tag{A6} $$

$$ \hat{g}_j = (g_j/\sqrt{J_{jj}}). \tag{A7} $$

2) Determine $\nu^{(k)}$ as follows.

Let $\xi > 1.0$, and $\nu^{(k-1)}$ denote the value of ν from the previous iteration; initially let $\nu^{(0)} = 10^{-2}$.

Compute $G^{(k+1)}(\nu^{(k-1)})$ and $G^{(k+1)}(\nu^{(k-1)}/\xi)$, from (29) as follows:

a) If $G^{(k+1)}(\nu^{(k-1)}/\xi) \le G^{(k)}$, let $\nu^{(k)} = \nu^{(k-1)}/\xi$.

b) If $G^{(k+1)}/(\nu^{(k-1)}/\xi) > G^{(k)}$, and $G^{(k+1)}(\nu^{(k-1)}) \le G^{(k)}$, let $\nu^{(k)} = \nu^{(k-1)}$.

3) Solve (A8) for the normalized correction vector $\hat{\boldsymbol{\delta}}$,

$$ ([\hat{J}]^{(k)} + \lambda^{(k)}[I])(\hat{\boldsymbol{\delta}})^{(k)} = (\hat{\boldsymbol{g}})^{(k)}. \tag{A8} $$

4) Solve (A9) for correction vector $\boldsymbol{\delta}$,

$$ \delta_j = \hat{\delta}_j/\sqrt{J_{jj}}\ . \tag{A9} $$

The constant $\nu^{(k)}$ is selected in order to satisfy inequality (28). The normalization procedure is sometimes referred as "scaling" and widely used in linear least squares problems as a way of improving the performance of an algorithm [20], [23].

REFERENCES

[1] R. P. Paul, B. Shimano, and G. E. Mayer, "Kinematic control equations for simple manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, no. 6, pp. 456–460, 1981.

[2] M. Benati, P. Morasso, and V. Togliasco, "The inverse kinematic problem of anthropomorphic manipulator arms," *ASME J. Dynamic Syst., Meas., Contr.*, vol. 104, pp. 110–113, 1982.

[3] A. A. Goldenberg and B. Benhabib, "End effector optimal path generation," *IEEE Trans. Syst., Man, Cybern.*, submitted for publication.

[4] R. G. Fenton, B. Benhabib, and A. A. Goldenberg, "Optimal point to point control of robots with redundant degrees of freedom," in *Proc. ASME Winter Ann. Meeting,* New Orleans, PED-Vol. 13, pp. 93–109; *ASME J. Eng. Ind.* to appear.

[5] A. A. Goldenberg and D. L. Lawrence, "A generalized solution to the inverse kinematics of robotic manipulators," *ASME J. Dynamic Syst., Meas., Contr.*, vol. 107, pp. 103–106, Mar. 1985.

[6] B. Benhabib, A. A. Goldenberg, and R. G. Fenton, "A solution to the inverse kinematics of redundant manipulators," *ASME J. Dynamic Syst., Meas., Contr.*, submitted for publication; *Proc. ACC,* 1985.

[7] V. J. Lumelsky, "Iterative coordinate transformation procedure for one class of robots," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC—14, no. 3, pp. 500–505, 1984.

[8] R. Featherstone, "Position and velocity transformations between robot end effector coordinates and joint angles," *Int. J. Robot. Res.*, vol. 3, no. 2, pp. 35–45, 1983.

[9] A. Liegeois, "Automatic supervisory control of the configuration and behaviour of multibody mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, no. 12, pp. 868–871, 1977.

[10] R. P. Paul, *Robot Manipulators, Mathematics, Programming and Control.* Cambridge, MA: *Mass. Inst. Tech.*, 1981.

[11] J. J. Uicker, J. Denavit, and R. S. Hartenberg, "An iterative method for the displacement analysis of spatial mechanisms," *ASME J. App. Mech.*, pp. 309–314, 1964.

[12] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Non-linear Equations in Several Variables.* New York: Academic, 1970.

[13] J. F. Traub, *Iterative Methods for the Solution of Equations.* Englewood Cliffs, NJ: Prentice-Hall, 1964.

[14] D. D. Whitney, "Optimum step size control for Newton-Raphson solution of non-linear vector equations," *IEEE Trans. Automat. Contr.*, pp. 572–574, 1969.

[15] J. Denavit, and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *ASME J. Appl. Mech.*, pp. 215–221, 1955.

[16] P. R. Paul and C. H. Wu, "Resolved motion force control of robot manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-12, no. 3, pp. 226–275, 1982.

[17] R. Fletcher, "Generalized inverse methods for the best least squares solution of systems of non-linear equations," *Comput. J.*, no. 10, pp. 392–399, 1968.

[18] P. Rabinowitz, *Numerical Methods for Non-linear Algebraic Equations.* New York: Gordon and Breach, 1970, pp. 87–114.

[19] J. G. P. Barnes, "An algorithm for solving non-linear equations based on the secont method," *Comput. J.*, no. 8, pp. 66–72, 1965.

[20] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Indust. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.

[21] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quart. Appl. Math.*, no. 2, pp. 164–168, 1944.

[22] P. Rabinowitz, *Numerical Methods for Non-linear Algebraic Equations.* New York: Gordon and Breach, 1970, pp. 115–161.

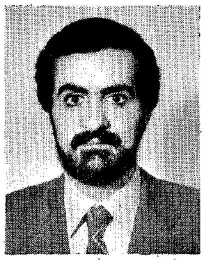
[23] J. More, *et al., User guide for Minpack-1.* Argonne, IL: Argonne National Laboratory, 1980.

[24] S. Y. Oh, D. Orin, and M. Bech, "An inverse kinematic solution for kinematically redundant robot manipulators," *J. Robot. Syst.*, vol. 1, no. 3, pp. 235–249, 1984.

Andrew A. Goldenberg (S'73–M'76) was born in Bucharest, Romania. He received the B.Sc. and M.Sc. degrees at the Technion, Haifa, Israel, in 1969 and 1972, respectively, and the Ph.D. degree at the University of Toronto, Toronto, ON, Canada, in 1976, all in electrical engineering.

From 1975–1981 he was employed by Spar Aerospace Ltd., Toronto, where he worked mainly on control, analysis, and design of the shuttle space remote manipulator system. During 1981–1982 he was an Assistant Professor of Electrical Engineering at the University of Toronto, and since 1982 he has been an Associate Professor of Mechanical Engineering, cross-appointed in electrical engineering, at the University of Toronto. He is presently Head of the Robotics and Automation Laboratory at the University of Toronto. His current research interests are in the field of robotics and automation in manufacturing, kinematics, control and dynamics of robots, and simulation. He is a Consultant to IBM of Canada Ltd., Northern Telecom Ltd., Ontario Hydro, the Ministry of Transportation and Communication of Ontario, and the National Research Council.

Dr. Goldenberg is a member of the American Society of Mechanical Engineers, the Society of Manufacturing Engineers, and the Association of Professional Engineers of Ontario.



B. Benhabib was born in Istanbul, Turkey in 1957. He received the B.Sc. degree in mechanical engineering from Bogazici University, Istanbul, Turkey, in 1980, and the M.Sc. degree in mechanical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1982.

He is currently working towards the Ph.D. degree in mechanical engineering at the University of Toronto, Toronto, ON, Canada. His research interests are in kinematics and control of robotic systems.

Robert G. Fenton was born in Budapest, Hungary, on April 3, 1931. He received the Dipl.Ing. degree with honors from the Technical University of Budapest, Budapest, Hungary, in 1953, and the Ph.D. degree from the University of New South Wales, Australia, in 1968, both in mechanical engineering.

After several years of work in industry he joined the faculty of the University of New South Wales in 1963, and subsequently moved to the University of Toronto, Toronto, ON, Canada, in 1968, where he is currently a Professor of Mechanical Engineering.

Dr. Fenton is a member of the Association of Professional Engineers of Ontario and the Society of Manufacturing Engineers. He received the Winter Arbitration Prize in 1969 and the P. Henderson Memorial Prize in 1970, from the Institution of Mechanical Engineers, Great Britain, for his work on metal removal processes.