Rachid Manseur
Keith L. Doty
Machine Intelligence Laboratory
Department of Electrical Engineering
University of Florida
Gainesville, Florida

A Fast Algorithm for Inverse Kinematic Analysis of Robot Manipulators

Abstract

To solve the inverse kinematics problem, we obtain with little effort a reduced and complete set of equations by a convenient choice of end-effector frame and application of rotation orthogonality. This approach does not require computation of the forward kinematics and can be used with manipulators of any geometry, although it is most efficient when applied to orthogonal manipulators, a class of robot arms defined in this paper. For manipulators requiring numerical techniques, but for which knowledge of one joint variable allows closed-form solutions of the remaining joint variables, an iterative inverse kinematic method, simple and fast enough to be suitable for real-time manipulator control, has been developed. The concepts and techniques presented in this paper are illustrated with two examples. The iterative method developed here performs a kinematic inversion of a 6-degree-of-freedom manipulator with no closed-form solutions in less than 30 ms using a desktop computer, an order of magnitude faster than times found in the literature.

1. Introduction

The inverse kinematics problem is to find a set of joint-variable values that will place the end-effector of a robot manipulator into a given pose (i.e., position and orientation). This problem is an important part of computer control algorithms for open serial kinematic chains (manipulators). Some 6- and 5-degree-of-freedom (DOF) arms with simple geometries allow closed-form inverse solutions. Pieper (1968) has shown that a sufficient condition for a manipulator to have a closed-form solution is that three adjacent joint axes intersect. If the intersecting axes are the last three, the so-called wrist-partitioned type of manipulator is obtained. Computationally efficient position, velocity, and acceleration inverse kinematics for this type of arm have been presented (Featherstone 1983; Hollerbach and Sahar 1983; Low and Dubey 1986; Paul and Zhang 1986).

Numerical techniques for determining a manipulator configuration that will position and orient the end-effector in a desired fashion can be found in the literature for general geometry arms as well (Goldenberg, Benhabib, and Fenton 1985; Goldenberg and Lawrence 1985; Angeles 1985, 1986). These numerical methods use multidimensional Newton-Raphson or similar techniques to provide a solution. Their computational efficiency is hindered by the need to compute the inverse Jacobian of the manipulator at several points.

Tsai and Morgan (1984) described a remarkable homotopy map method, guaranteed to find all solutions of a system of polynomial equations in several variables and applied it to the inverse kinematic problem of 5- or 6-revolute-DOF arms of arbitrary architecture. The computational complexity of the method makes it impractical for on-line use. However, in the process, the inverse kinematic problem is reduced to four equations in only four of the joint variables. In this paper, we show that this simplification and considerable algebraic reduction can be obtained with much less effort by a convenient choice of joint frames and proper application of rotation orthogonality. The power of this simplification procedure is enhanced when applied to orthogonal manipulators, which are defined in this paper.

The technique for finding a reduced set of equations is shown to be helpful in solving the inverse kine-

The International Journal of Robotics Research,
Vol. 7, No. 3, June 1988,
© 1988 Massachusetts Institute of Technology.

matics for arms that allow closed-form solutions as well. The PUMA 560 inverse kinematics problem is solved to illustrate the power of this approach.

Finally, we present an original and fast iterative technique, based on the reduced set of equations, that is suitable for real-time control of manipulators for which knowledge of one joint variable allows a closed-form computation of the remaining variables. In our second example, this iterative technique is applied to an existing 6-DOF arm and programmed on a desktop computer. The average inversion time is found to be less than 30 ms, an inversion time at least one order of magnitude better than those found in the literature.

2. Notation and Manipulator Frame Assignment

A manipulator is an open chain of rigid bodies (links) connected together by joints. Each link is free to rotate about or slide along a joint axis with respect to the preceding link. Using the Denavit-Hartenberg parameters (1955), each link i is assigned a frame of reference F_i with a location and orientation entirely described by the four parameters d_i, Θ_i, a_i, α_i with respect to the preceding frame F_{i-1} along the chain. For an n-link, n-DOF manipulator, the frames are numbered from 0 to n, with frame 0 being the base frame and frame n the end-effector frame. Link i can either rotate about or slide along axis z_{i-1}. Since there is no link $n+1$, frame F_n can be chosen so that $\alpha_n = a_n = d_n = 0$, if joint n is revolute, and $\alpha_n = a_n = \Theta_n = 0$, if it is prismatic. Frame F_0 can be positioned such that $d_1 = 0$, if joint 1 is revolute. These assignments simplify the computation without loss of generality.

A vector expression in frame F_i and its expression in frame F_{i-1} are related by the homogeneous matrix transforms $\mathbf{A}_i$ and $(\mathbf{A}_i)^{-1}$ given by

cos θ
sin θ
sin α
cos α

$$\mathbf{A}_i = \begin{bmatrix} C_i & -S_i\tau_i & S_i\sigma_i & a_iC_i \\ S_i & C_i\tau_i & -C_i\sigma_i & a_iS_i \\ 0 & \sigma_i & \tau_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i & & & \mathbf{l}_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (1)$$

Я

$$(\mathbf{A}_i)^{-1} = \begin{bmatrix} C_i & S_i & 0 & -a_i \\ -S_i\tau_i & C_i\tau_i & \sigma_i & -\sigma_i d_i \\ S_i\sigma_i & -C_i\sigma_i & \tau_i & -\tau_i d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i^T & & & (-\mathbf{R}_i^T\mathbf{l}_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (2)$$

where $C_i = \cos\Theta_i$, $S_i = \sin\Theta_i$, $\tau_i = \cos\alpha_i$, and $\sigma_i = \sin\alpha_i$. The upper left 3×3 matrix in $\mathbf{A}_i$ is the rotation matrix $\mathbf{R}_i$ necessary to align the unit vectors of F_i with those of F_{i-1}. Rotation matrices are orthogonal, so $\mathbf{R}_i^{-1} = \mathbf{R}_i^T$. Vector $\mathbf{l}_i = [a_iC_i, a_iS_i, d_i]^T$ positions the origin of F_i with respect to F_{i-1}.

Given an end-effector pose $\mathbf{P}$ expressed with respect to the base frame F_0 by the matrix

$$\mathbf{P} = \begin{bmatrix} n_x & b_x & t_x & p_x \\ n_y & b_y & t_y & p_y \\ n_z & b_z & t_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{b} & \mathbf{t} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & \mathbf{R} & & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (3)$$

where

$$\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix},$$

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} n_x & b_x & t_x \\ n_y & b_y & t_y \\ n_z & b_z & t_z \end{bmatrix},$$

the basic inverse kinematic problem for an n-DOF arm is to find the values of all joint variables for which the following matrix equation holds

$$\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5 \cdots \mathbf{A}_n = \mathbf{P}. \qquad (4)$$

At least 6 DOF are required to arbitrarily position and orient a rigid body in space. If n is larger than six, the manipulator is redundant, and the system of equations implied by (4) is underconstrained. If $n < 6$, the system becomes overconstrained. In this paper we will restrict our discussion to the exactly specified system obtained with $n = 6$, although the simplification techniques presented below can be of assistance for other values of n as well.

In the following, a leading superscript will be used to designate the frame of expression of a given vector (for example ${}^3\mathbf{p}$ represents vector $\mathbf{p}$ expressed in frame 3).

3. Inverse Kinematics Equations

For a 6-DOF arm, Eq. (4) yields 12 nontrivial scalar equations in the six unknown variables. It is desirable to reduce this system to a minimal number of equations involving as few of the joint variables as possible. For all-revolute, 6-DOF manipulators, Tsai and Morgan (1984) have identified that with respect to frame F_3, the z-component of the position vector ${}^3\mathbf{p}$ and that of vector ${}^3\mathbf{t}$ along with the inner products $({}^3\mathbf{t} \cdot {}^3\mathbf{p})$ and $({}^3\mathbf{p} \cdot {}^3\mathbf{p})$ provide four equations in only four of the unknowns, thereby reducing the complexity of the problem. The process of obtaining these four equations involved multiplying the A matrices and simplifying the expressions obtained for the elements of ${}^3\mathbf{t}$ and ${}^3\mathbf{p}$. Besides being lengthy, this method does not allow insight into the mechanisms that make the simplifications possible. The approach presented here provides the same results with much less effort and greater insight by taking advantage of the properties of rotation transformations.

By writing the product of two A matrices in the form

$$\mathbf{A}_i\mathbf{A}_j = \begin{bmatrix} \mathbf{R}_i\mathbf{R}_j & (\mathbf{R}_i\mathbf{l}_j + \mathbf{l}_i) \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix},$$

we can divide Eq. (4) into a position equation and an orientation equation, which can be expressed, respectively, as

$$\mathbf{p} = \mathbf{R}_1(\mathbf{R}_2(\mathbf{R}_3(\mathbf{R}_4(\mathbf{R}_5\mathbf{l}_6 + \mathbf{l}_5) + \mathbf{l}_4) + \mathbf{l}_3) + \mathbf{l}_2) + \mathbf{l}_1 \quad (5)$$

and

$$\mathbf{R} = \mathbf{R}_1\mathbf{R}_2\mathbf{R}_3\mathbf{R}_4\mathbf{R}_5\mathbf{R}_6. \quad (6)$$

With the frame assignment conventions discussed, $\mathbf{l}_6 = \mathbf{0}$ when joint 6 is revolute. Equation (5) then simplifies to

$$\mathbf{p} = \mathbf{R}_1(\mathbf{R}_2(\mathbf{R}_3(\mathbf{R}_4\mathbf{l}_5 + \mathbf{l}_4) + \mathbf{l}_3) + \mathbf{l}_2) + \mathbf{l}_1. \quad (5')$$

Three independent scalar equations for p_x, p_y, and p_z can be obtained from (5) and three more equations can be selected out of the nine scalar equations implied by (6).

Since rotations are orthogonal transformations, they leave inner products invariant.

$$\mathbf{Qu} \cdot \mathbf{Qv} = \mathbf{u} \cdot \mathbf{v} \quad (7)$$

for any rotation matrix $\mathbf{Q}$ and any vectors $\mathbf{u}$ and $\mathbf{v}$. A special case of (7) that is sometimes useful is

$$\mathbf{Qu} \cdot \mathbf{v} = \mathbf{u} \cdot \mathbf{Q}^{-1}\mathbf{v}. \quad (8)$$

These properties are extremely efficient in eliminating algebraic terms and unnecessary joint variables when applied to Eqs. (5) and (6), provided it is further recognized that

$$\mathbf{R}_i^{-1}\mathbf{l}_i = [a_i,\ d_i\sigma_i,\ d_i\tau_i]^{\mathrm{T}} \quad (9)$$

and

$$\mathbf{R}_i^{-1}\mathbf{z} = [0,\ \sigma_i,\ \tau_i] \qquad \text{where } \mathbf{z} = [0, 0, 1]^{\mathrm{T}} \quad (10)$$

are independent of θ_i, when joint i is revolute. Also, due to the frame assignments discussed earlier, $\mathbf{R}_6\mathbf{z} = \mathbf{z}$ in all cases, since frame F_6 can be chosen to force $\alpha_6 = 0$.

By using Eqs. (7) and (8) repeatedly, we obtain four reduced equations:

t_z equation

$$\begin{aligned} t_z &= \mathbf{t} \cdot \mathbf{z} = (\mathbf{Rz}) \cdot \mathbf{z}, \\ t_z &= (\mathbf{R}_1\mathbf{R}_2\mathbf{R}_3\mathbf{R}_4\mathbf{R}_5\mathbf{R}_6\mathbf{z}) \cdot \mathbf{z}, \\ t_z &= (\mathbf{R}_1\mathbf{R}_2\mathbf{R}_3\mathbf{R}_4\mathbf{R}_5\mathbf{z}) \cdot \mathbf{z}, \\ t_z &= \mathbf{z} \cdot (\mathbf{R}_5^{-1}\mathbf{R}_4^{-1}\mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{R}_1^{-1}\mathbf{z}). \end{aligned} \quad (11)$$

p_z equation

$$\mathbf{p} = \mathbf{R}_1\mathbf{R}_2\mathbf{R}_3\mathbf{R}_4\mathbf{q}$$

with

$$\begin{aligned} \mathbf{q} &= \mathbf{l}_5 + \mathbf{R}_4^{-1}(\mathbf{l}_4 + \mathbf{R}_3^{-1}(\mathbf{l}_3 + \mathbf{R}_2^{-1}(\mathbf{l}_2 + \mathbf{R}_1^{-1}\mathbf{l}_1))), \\ p_z &= \mathbf{p} \cdot \mathbf{z} = \mathbf{q} \cdot (\mathbf{R}_4^{-1}\mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{R}_1^{-1})\mathbf{z}. \end{aligned} \tag{12}$$

p.t equation

$$\mathbf{p} \cdot \mathbf{t} = \mathbf{R}_5^{-1}\mathbf{q} \cdot \mathbf{z}. \tag{13}$$

p.p equation

$$\mathbf{p} \cdot \mathbf{p} = \mathbf{p}^2 = \mathbf{q} \cdot \mathbf{q} = \mathbf{q}^2. \tag{14}$$

Since $\mathbf{R}_1^{-1}\mathbf{l}_1$ and $\mathbf{R}_1^{-1}\mathbf{z}$ are independent of Θ_1 (Eqs. (9) and (10)), vector $\mathbf{q}$ and Eqs. (11)–(14) are easily seen to be independent of the first and last joint variables and therefore form a system of four equations in four unknowns.

These four equations determine candidate solutions for joint variables 2, 3, 4, and 5. Once this system of equations is solved, the remaining two variables can be found using more equations from (4) and then tested for consistency. The power of this approach will become apparent for specific manipulators, since further simplification using Eqs. (7)–(10) becomes obvious. Furthermore, simplification by use of rotation inner product invariance is not only computationally economical, but it also provides greater insight into the structure and properties of the inverse kinematic equations.

Equations (7)–(10) are necessary, but not sufficient. Although they are satisfied by all solution sets of Eq. (4), they are also, in general, satisfied by extraneous solutions. This problem was reported by Tsai and Morgan (1984) as well.

Another problem with considering Eqs. (11)–(14) alone is the presence of sign ambiguities. In many practical situations, one of the equations will allow a closed-form solution for either the sine or the cosine function of a revolute variable Θ. The other function needs to be computed using the Pythagorean identity, which offers two values opposite in sign. Although both signs can be tried in the search for a solution, in some cases the number of sign ambiguities can be reduced by considering more constraints from Eqs. (5) and (6). Additional equations will also help filter out extraneous solutions and in some cases will ease the solution finding process rather than complicate it. The x- and y-components of vectors $\mathbf{t}$ and $\mathbf{p}$ provide convenient additional constraints at the cost of introducing the variable Θ_1. Equations

$$t_x = \mathbf{R}_1\mathbf{R}_2\mathbf{R}_3\mathbf{R}_4\mathbf{R}_5\mathbf{z} \cdot \mathbf{x}, \tag{15}$$

$$t_y = \mathbf{R}_1\mathbf{R}_2\mathbf{R}_3\mathbf{R}_4\mathbf{R}_5\mathbf{z} \cdot \mathbf{y}, \tag{16}$$

$$p_x = (\mathbf{R}_1(\mathbf{R}_2(\mathbf{R}_3(\mathbf{R}_4\mathbf{l}_5 + \mathbf{l}_4) + \mathbf{l}_3) + \mathbf{l}_2) + \mathbf{l}_1) \cdot \mathbf{x}, \tag{17}$$

$$p_y = (\mathbf{R}_1(\mathbf{R}_2(\mathbf{R}_3(\mathbf{R}_4\mathbf{l}_5 + \mathbf{l}_4) + \mathbf{l}_3) + \mathbf{l}_2) + \mathbf{l}_1) \cdot \mathbf{y}, \tag{18}$$

where $\mathbf{x} = [1, 0, 0]^T$ and $\mathbf{y} = [0, 1, 0]^T$ are the usual canonical unit vectors, are still independent of Θ_6.

4. Orthogonal Manipulators

Definition

An n-axis, serial kinematic chain of revolute or prismatic joints is orthogonal if all twist angles α_i, $i = 1, \ldots, n$, along the chain are 0 or $\pi/2$. An open orthogonal kinematic chain will be called an orthogonal manipulator (Doty 1986).

Six-DOF orthogonal manipulators can be classified in terms of the values of their twist angles α_i, $i = 1, \ldots, 6$. Since α_6 can always be chosen 0, there are only $2^5 = 32$ distinct classes of orthogonal manipulators, 8 of which have four or more adjacent parallel joint axes, which reduces their capability to less than 6 DOF. As a result, there are only 24 types of six joints orthogonal manipulators with full spatial position and orientation capability.

A convenient notation for this classification of orthogonal manipulators is obtained by assigning a 6-bit binary number to each of these 24 types in which bit i is 0 if $\alpha_i = 0$ and bit i is 1 if $\alpha_i = \pi/2$. For example, a manipulator with twist angles $\alpha_6 = 0$, $\alpha_5 = \pi/2$, $\alpha_4 = \pi/2$, $\alpha_3 = 0$, $\alpha_2 = 0$, and $\alpha_1 = \pi/2$ belongs to the class 011-001 of orthogonal manipulators. Twist angle α_6 is always 0. Thus, the leading bit can be omitted, and a 5-bit notation for all 24 classes can be used.

Since most industrial robot arms are orthogonal, it is worthwhile to consider the inverse kinematic problem with respect to these manipulators. The A-matrices associated with orthogonal arms have one of the two following forms:

$$\mathbf{A}_i(\alpha = 0) = \begin{bmatrix} C_i & -S_i & 0 & a_iC_i \\ S_i & C_i & 0 & a_iS_i \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (19)$$

or

$$\mathbf{A}_i(\alpha = \pi/2) = \begin{bmatrix} C_i & 0 & S_i & a_iC_i \\ S_i & 0 & -C_i & a_iS_i \\ 0 & 1 & 0 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (20)$$

Further computational simplification is obtained with orthogonal manipulators since

$$\mathbf{R}_i\mathbf{z} = \mathbf{R}_i^{-1}\mathbf{z} = \mathbf{z} \qquad \text{if } \alpha_i = 0$$

and

$$\mathbf{R}_i^{-1}\mathbf{z} = \mathbf{y} \qquad \text{if } \alpha_i = \pi/2.$$

Using this approach, Doty (1986) has shown that, of the 24 classes of nontrivial orthogonal manipulators, those with two nonzero twist angles (classes 01-001, 01-010, 01-100, 10-100, and 10-010) have closed-form solutions. The inverse kinematic analysis of the remaining classes is still under investigation.

5. Closed-Form Example: PUMA 560 Inverse Kinematics

A popular orthogonal manipulator geometry, the PUMA 560, is described by the kinematic parameters given in Table 1. This manipulator has a spherical wrist and therefore allows closed-form solutions (Pieper 1968). Inverse kinematic solutions have been proposed by numerous authors for this type of arm (Lee and Ziegler 1984; Craig 1986; Paul and Zhang 1986). This example is included here to demonstrate

Table 1. PUMA 560 Kinematic Parameters

Joint	d	Θ	a	α
1	0	Θ_1	0	$\pi/2$
2	0	Θ_2	a_2	0
3	d_3	Θ_3	a_3	$\pi/2$
4	d_4	Θ_4	0	$\pi/2$
5	0	Θ_5	0	$\pi/2$
6	0	Θ_6	0	0

the utility of the approach already outlined and to contrast it with the geometric and algebraic approaches taken by the previous authors.

In the following equations C_{ij} and S_{ij} stand for the cosine and sine of $\Theta_i + \Theta_j$, respectively. Without computing the forward kinematics, we will illustrate how Eqs. (11)-(14) may be easily obtained. For quick reference in the following discussion, we write the equations immediately.

$$t_z = S_{23}C_4S_5 + C_{23}C_5, \qquad (11')$$

$$p_z = a_2S_2 + a_3S_{23} - d_4C_{23}, \qquad (12')$$

$$\mathbf{p}\cdot\mathbf{t} = (a_3 + a_2C_3)C_4S_5 + d_3S_4S_5 - C_5(d_4 + a_2S_3) \qquad (13')$$

$$(\mathbf{p}^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2)/2a_2 = d_4S_3 + a_3C_3. \qquad (14')$$

To illustrate the simplification obtained by the frame assignment described earlier and inner-product invariance under rotations, we give in detail the development of Eq. (14′). With $\mathbf{l}_1 = \mathbf{l}_5 = \mathbf{l}_6 = \mathbf{0}$ and $\mathbf{l}_4 = d_4\mathbf{z}$, Eq. (5) yields

$$\mathbf{p} = \mathbf{R}_1(\mathbf{R}_2(\mathbf{R}_3\mathbf{l}_4 + \mathbf{l}_3) + \mathbf{l}_2),$$
$$\mathbf{p} = \mathbf{R}_1\mathbf{R}_2\mathbf{R}_3[\mathbf{l}_4 + \mathbf{R}_3^{-1}\mathbf{l}_3 + \mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2].$$

By orthogonality, the inner product $\mathbf{p}\cdot\mathbf{p}$ has the same value as the inner product of the term in brackets; hence

$$\mathbf{p}\cdot\mathbf{p} = [\mathbf{l}_4 + \mathbf{R}_3^{-1}\mathbf{l}_3 + \mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2]\cdot[\mathbf{l}_4 + \mathbf{R}_3^{-1}\mathbf{l}_3 + \mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2].$$

The inner product of each term in brackets with itself is the square of the length of that vector. For example,

$$\mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2 \cdot \mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2 = \mathbf{l}_2 \cdot \mathbf{l}_2 = a_2^2 + d_2^2 = \mathbf{l}_2^2.$$

These inner-product manipulations represent a considerable algebraic simplification that requires little or no mental effort. Further, they provide a methodology and considerable insight into how to find other algebraic reductions.

Some of the cross terms also reduce; for instance,

$$\mathbf{R}_3^{-1}\mathbf{l}_3 \cdot \mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2 = \mathbf{l}_3 \cdot \mathbf{R}_2^{-1}\mathbf{l}_2.$$

Complete expansion of Eq. (14) and application of the reduction techniques just discussed lead to

$$(\mathbf{p}^2 - \mathbf{l}_4^2 - \mathbf{l}_3^2 - \mathbf{l}_2^2)/2 = \mathbf{l}_4 \cdot [\mathbf{R}_3^{-1}\mathbf{l}_3 + \mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2] + \mathbf{l}_3 \cdot \mathbf{R}_2^{-1}\mathbf{l}_2.$$

For this manipulator, vectors $\mathbf{l}_4 = [0, 0, d_4]^T = d_4 z$, $\mathbf{R}_3^{-1}\mathbf{l}_3 = [a_3, d_3, 0]^T$, and $\mathbf{R}_2^{-1}\mathbf{l}_2 = [a_2, 0, 0]^T = a_2\mathbf{x}$ allow us to simplify the last equation:

$$(\mathbf{p}^2 - \mathbf{l}_4^2 - \mathbf{l}_3^2 - \mathbf{l}_2^2)/2 = d_4 z \cdot [\mathbf{R}_3^{-1}\mathbf{l}_3 + a_2\mathbf{R}_3^{-1}\mathbf{x}] + a_2\mathbf{l}_3 \cdot \mathbf{x},$$

(e.g., $\mathbf{l}_4 \cdot \mathbf{R}_3^{-1}\mathbf{l}_3$ is obviously 0, which eliminates Θ_4 from this equation).

Without any matrix multiplication required, we obtain the fully simplified relation involving Θ_3 only:

$$(\mathbf{p}^3 - a_2^2 - a_3^2 - d_3^2 - d_4^2)/2 = a_2(d_4 S_3 + a_3 C_3).$$

The last equation is of the form $aS + bC = d$, where S and C are the sine and cosine of some angle Θ. Such an equation has two solutions, when $a^2 + b^2 \geq d^2$,

$$\Theta = \text{atan2}[d, \pm\sqrt{a^2 + b^2 - d^2}] - \text{atan2}(b, a)$$

where atan2(v, w) returns the angle Arctan(v/w) adjusted to the proper quadrant according to the sign of the real numbers v and w.

At this point Eq. (14′) can be solved for Θ_3, yielding two solutions. After applying trigonometric identities for angle sums to (12′), we get

$$p_z = a_2 S_2 + a_3(S_2 C_3 + S_3 C_2) - d_4(C_2 C_3 - S_2 S_3),$$

and grouping terms, we obtain

$$(a_2 + a_3 C_3 + d_4 S_3)S_2 + (a_3 S_3 - d_4 C_3)C_2 = p_z.$$

With Θ_3 known, two values can be obtained for Θ_2. Doty (1986) has shown that all 4-DOF manipulators have closed-form solutions with at most two distinct solution sets. This means that if two angles of a 6-DOF manipulator can be found in closed form, the entire angle set is solvable.

With Θ_2 and Θ_3 known, (11′) and (13′) become functions of Θ_4 and Θ_5 only. Although this system of two equations in two unknowns can theoretically be solved, its solution is not obvious. A simpler solution exists if Eqs. (15)-(18) are considered.

$$\mathbf{p} \cdot \mathbf{x} = \mathbf{R}_1(\mathbf{R}_2(\mathbf{R}_3\mathbf{l}_4 + \mathbf{l}_3) + \mathbf{l}_2) \cdot \mathbf{x} = p_x, \quad (17')$$

$$\mathbf{p} \cdot \mathbf{y} = \mathbf{R}_1(\mathbf{R}_2(\mathbf{R}_3\mathbf{l}_4 + \mathbf{l}_3) + \mathbf{l}_2) \cdot \mathbf{y} = p_y, \quad (18')$$

or

$$(d_4 S_{23} + a_3 C_{23} + a_2 C_2)C_1 + d_3 S_1 = p_x,$$
$$-d_3 C_1 + (d_4 S_{23} + a_3 C_{23} + a_2 C_2)S_1 = p_y.$$

The last two equations form a linear system in S_1 and C_1 and provide a unique value for Θ_1.

Equations (15) and (16) along with (11′) provide a way to solve for Θ_4 and Θ_5:

$$t_x = C_1 C_{23} C_4 S_5 + S_1 S_4 S_5 - C_1 S_{23} C_5, \quad (15')$$

$$t_y = S_1 C_{23} C_4 S_5 - C_1 S_4 S_5 - S_1 S_{23} C_5. \quad (16')$$

Solving for C_5 in (11′) and substituting in the last two equations give (after grouping terms)

$$C_1 C_4 S_5 + S_1 C_{23} S_4 S_5 = t_x C_{23} + t_z C_1 S_{23},$$
$$S_1 C_4 S_5 - C_1 C_{23} S_4 S_5 = t_y C_{23} + t_z S_1 S_{23}.$$

This linear system can be solved uniquely for the products $C_4 S_5$ and $S_4 S_5$. When $S_5 \neq 0$, two solutions for Θ_4 are then obtained:

$$\Theta_4 = \text{atan2}(S_4 S_5, C_4 S_5) \quad \text{or}$$
$$\Theta_4 = \text{atan2}(-S_4 S_5, -C_4 S_5).$$

When $S_5 = 0$, joint axes $\mathbf{z}_3$ and $\mathbf{z}_5$ are aligned and the manipulator loses 1 DOF. Only the sum $\Theta_4 + \Theta_6$ can be found by use of Eqs. (5) and (6).

With Θ_4 known, the t_x and t_y equations constitute a linear system of equations that yields a unique solution for Θ_5. The last joint variable Θ_6 can then be obtained from two more equations from (6) such as the n_z and b_z equations. This procedure will yield eight solutions, which must then be checked for joint-variable range limitations. We end the discussion of the PUMA example with the observation that the forward kinematics were never determined in order to obtain the inverse kinematic solution!

6. Iterative Procedure

For many manipulator geometries, closed-form solutions cannot be found; therefore numerical techniques must be used to solve the inverse kinematic system of equations. The numerical techniques found in the literature are based on multidimensional Newton-Raphson, or similar, methods that require use of the manipulator inverse Jacobian (Goldenberg, Benhabib, and Fenton 1985; Goldenberg and Lawrence 1985; Angeles 1985, 1986). The method proposed here takes advantage of the reduced set of inverse kinematic equations discussed earlier to provide an algebraically simpler and computationally faster iterative technique. This new technique can be applied to any 6-DOF manipulator for which all joint variables can be found in closed form when one joint variable is known. For manipulators satisfying this last condition, the inverse kinematic problem can be reduced to a one-dimensional root-finding process for which simple and fast numerical techniques, such as the one-dimensional Newton-Raphson or the secant method, are well suited.

Equation (4) can also be expressed as

$$\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6 = \mathbf{A}_1^{-1}\mathbf{P} = {}^1\mathbf{P} = \begin{bmatrix} {}^1\mathbf{n} & {}^1\mathbf{b} & {}^1\mathbf{t} & {}^1\mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (21)$$

where the right side is a function only of Θ_1. Assuming Θ_1 known, the problem reduces to five unknowns and, for many manipulators, can be solved in closed form by the techniques presented here.

From Eq. (21),

$${}^1\mathbf{p} = \mathbf{R}_2(\mathbf{R}_3(\mathbf{R}_4\mathbf{l}_5 + \mathbf{l}_4) + \mathbf{l}_3) + \mathbf{l}_2 = \mathbf{R}_2\mathbf{R}_3\mathbf{R}_4{}^1\mathbf{q} \qquad (22)$$

with

$${}^1\mathbf{q} = \mathbf{l}_5 + \mathbf{R}_4^{-1}(\mathbf{l}_4 + \mathbf{R}_3^{-1}(\mathbf{l}_3 + \mathbf{R}_2^{-1}\mathbf{l}_2)). \qquad (23)$$

Again with repetitive use of rotation orthogonality, we have

$$\begin{aligned} {}^1t_z &= {}^1\mathbf{t} \cdot \mathbf{z} = \mathbf{R}_2\mathbf{R}_3\mathbf{R}_4\mathbf{R}_5\mathbf{z} \cdot \mathbf{z} \\ &= \mathbf{z} \cdot (\mathbf{R}_5^{-1}\mathbf{R}_4^{-1}\mathbf{R}_3^{-1}\mathbf{R}_2^{-1})\mathbf{z}, \end{aligned} \qquad (24)$$

$$\begin{aligned} {}^1p_z &= {}^1\mathbf{p} \cdot \mathbf{z} = {}^1\mathbf{q} \cdot \mathbf{R}_4^{-1}\mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{z} \\ &= \mathbf{R}_1^{-1}(\mathbf{p} - \mathbf{l}_1) \cdot \mathbf{z}, \end{aligned} \qquad (25)$$

$${}^1\mathbf{p} \cdot {}^1\mathbf{t} = \mathbf{R}_5^{-1}{}^1\mathbf{q} \cdot \mathbf{z} = \mathbf{p} \cdot \mathbf{t}, \qquad (26)$$

$${}^1\mathbf{p} \cdot {}^1\mathbf{p} = {}^1\mathbf{q} \cdot {}^1\mathbf{q}. \qquad (27)$$

Equations (24)–(27) provide four equations independent of Θ_2 and of Θ_6. These equations are the basis of the iterative technique proposed here; however, to alleviate the problem of extraneous solutions and sign ambiguities, the following additional equations, which introduce the variable Θ_2, can be of assistance.

$${}^1t_x = \mathbf{z} \cdot (\mathbf{R}_5^{-1}\mathbf{R}_4^{-1}\mathbf{R}_3^{-1}\mathbf{R}_2^{-1})\mathbf{x} = \mathbf{R}_1^{-1}\mathbf{t} \cdot \mathbf{x}, \qquad (28)$$

$${}^1t_y = \mathbf{z} \cdot (\mathbf{R}_5^{-1}\mathbf{R}_4^{-1}\mathbf{R}_3^{-1}\mathbf{R}_2^{-1})\mathbf{y} = \mathbf{R}_1^{-1}\mathbf{t} \cdot \mathbf{y}, \qquad (29)$$

$$\begin{aligned} {}^1p_x &= {}^1\mathbf{p} \cdot \mathbf{z} = {}^1\mathbf{q} \cdot \mathbf{R}_4^{-1}\mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{x} \\ &= \mathbf{R}_1^{-1}(\mathbf{p} - \mathbf{l}_1) \cdot \mathbf{x}, \end{aligned} \qquad (30)$$

$$\begin{aligned} {}^1p_y &= 1\mathbf{p} \cdot \mathbf{z} = {}^1\mathbf{q} \cdot \mathbf{R}_4^{-1}\mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{y} \\ &= \mathbf{R}_1^{-1}(\mathbf{p} - \mathbf{l}_1) \cdot \mathbf{y}. \end{aligned} \qquad (31)$$

The application examples below will illustrate this discussion and elaborate the general steps involved in implementing the method for an arbitrary manipulator.

6.1. Outline of the Iterative Method

1. Derive explicitly Eqs. (24)–(31).
2. Assume θ_1 known and verify that the other joint variables can be obtained in closed form from a proper selection of Eqs. (24)–(31). This is a condition for the applicability of this method.
3. Find a real-valued function of Θ_1 by using one of the equations that can be computed for any value of Θ_1. For example, if given Θ_1 and using the remaining equations, Eq. (26) can be evaluated, then a good candidate function is

$$f(\Theta_1) = \mathbf{R}_5^{-1} \times (\mathbf{l}_5 + \mathbf{R}_4^{-1}(\mathbf{l}_4 + \mathbf{R}_3^{-1}(\mathbf{l}_3 + \mathbf{R}_2^{-1}\mathbf{l}_2))) \times \mathbf{z} - \mathbf{p} \cdot \mathbf{t}.$$

4. Implement a Newton-Raphson or other numerical method for finding a root of the function f from an initial guess of Θ_1.

Once a root of f is found, the values of the remaining joint variables can then be computed in closed form.

6.2. Computing the Derivative of f

If Newton-Raphson is to be used, then the derivative of the function f must be computed as well as the function itself. From Eq. (11), it can be seen that the derivative of the right side can be explicitly computed as a function of Θ_1.

Again using the example function of step 3 and assuming an all-revolute 6-DOF arm, $df/d\Theta_1$ will depend on the values of C_i, S_i, $dC_i/d\Theta_1$, and $dS_i/d\Theta_1$ for $i = 3, 4, 5$. By differentiating Eqs. (24)–(31) as necessary, we obtain $dC_i/d\Theta_1$ and $dS_i/d\Theta_1$ for $i = 3, 4, 5$. A useful additional relation is provided by differentiating the Pythagorean constraint on C_i and S_i with respect to Θ_1:

$$C_i^2 + S_i^2 = 1, \qquad (32)$$

so that

$$C_i \frac{dC_i}{d\Theta_1} + S_i \frac{dS_i}{d\Theta_1} = 0. \qquad (33)$$

Table 2. GP66 Manipulator Kinematic Parameters (joint 3 is prismatic)

Joint	d	Θ	a	α
1	0	Θ_1	0	$\pi/2$
2	0	Θ_2	a_2	$\pi/2$
3	d_3	0	0	0
4	0	Θ_4	0	$\pi/2$
5	d_5	Θ_5	0	$\pi/2$
6	0	Θ_6	0	0

In practical situations, the derivative of f can be approximated numerically by

$$\frac{df}{d\Theta_1} = \frac{f(\Theta_1 + \delta) - f(\Theta_1)}{\delta}$$

with a small value of δ.

A root of f will correspond to a true solution of the inverse kinematic problem or to an extraneous solution. To avoid extraneous solutions, select f so that its computation requires the use of several of the constraint equations, Eqs. (24)–(31). In several applications, satisfactory results were obtained by selecting either (26) or (27) to define the function f.

With this method, sometimes a division by S_i or C_i needs to be performed. If either of these variables becomes zero, a pertinent value of Θ_i from the set $\{0, \pi/2, \pi, 3\pi/2\}$ along with the current value of Θ_1 should allow solving for the remaining variables in closed form for that particular iteration.

7. Iterative Method Example: GP66 Manipulator

Consider the manipulator geometry with kinematic parameters given in Table 2. This robot arm is an existing industrial manipulator that belongs to the 11–011 class of orthogonal arms and does not allow closed-form solutions. An iterative method that exactly computes the position, but approximates the orientation, was proposed for this type of geometry by Lumelsky (1984). The technique presented here differs in that it solves for both the orientation and the position with the same precision, and it is applicable to a larger variety of manipulators.

Assuming a guess of Θ_1, we can compute the corresponding x- and z-components of ${}^1\mathbf{p}$ and ${}^1\mathbf{t}$:

$$ {}^1p_x = p_x C_1 + p_y S_1, \tag{34a}$$

$$ {}^1p_z = p_x S_1 - p_y C_1, \tag{34b}$$

$$ {}^1t_x = t_x C_1 + t_y S_1, \tag{34c}$$

$$ {}^1t_z = t_x S_1 - t_y C_1. \tag{34d}$$

Next, we derive Eqs. (24)–(27) as applied to this manipulator. For this robot $\mathbf{l}_1 = \mathbf{l}_4 = \mathbf{l}_6 = \mathbf{0}$, $\mathbf{R}_2^{-1}\mathbf{l}_2 = a_2\mathbf{x}$, $\mathbf{l}_3 = d_3\mathbf{z}$, $\mathbf{l}_5 = d_5\mathbf{z}$, and $\mathbf{R}_3 = I$. With these values, (5) yields

$$\mathbf{p} = \mathbf{R}_1\mathbf{R}_2(d_5\mathbf{R}_4\mathbf{z} + d_3\mathbf{z} + a_2\mathbf{x}).$$

After multiplication by $\mathbf{R}_1^{-1}$,

$$ {}^1\mathbf{p} = \mathbf{R}_2(d_5\mathbf{R}_4\mathbf{z} + d_3\mathbf{z} + a_2\mathbf{x}). \tag{35}$$

Vector ${}^1\mathbf{t}$ simplifies to

$$ {}^1\mathbf{t} = \mathbf{R}_1^{-1}\mathbf{t} = \mathbf{R}_2\mathbf{R}_4\mathbf{R}_5\mathbf{z}. \tag{36}$$

Computing ${}^1t_z$

Using Eq. (36), and Eqs. (7) and (8) as necessary, we obtain

$$ {}^1t_z = {}^1\mathbf{t}\cdot\mathbf{z} = \mathbf{R}_5\mathbf{z}\cdot\mathbf{R}_4^{-1}\mathbf{y}.$$

Since $\mathbf{R}_5\mathbf{z} = [S_5, -C_5, 0]^T$ and $\mathbf{R}_4^{-1}\mathbf{y} = [S_4, 0, -C_4]^T$, the preceding equation becomes

$$ {}^1t_z = S_4 S_5. \tag{37}$$

Computing ${}^1p_z$

Since ${}^1p_z = {}^1\mathbf{p}\cdot\mathbf{z}$, from Eq. (35), Eq. (8), and $\mathbf{R}_2^{-1}\mathbf{z} = \mathbf{y}$,

$$ {}^1p_z = (d_5\mathbf{R}_4\mathbf{z} + d_3\mathbf{z} + a_2\mathbf{x})\cdot\mathbf{y},$$

which is easily seen to produce

$$ {}^1p_z = -d_5 C_4. \tag{38}$$

Computing ${}^1\mathbf{t}\cdot{}^1\mathbf{p}$

Equations (35) and (36) and use of Eqs. (7) and (8) yield

$$ {}^1\mathbf{t}\cdot{}^1\mathbf{p} = \mathbf{t}\cdot\mathbf{p} = \mathbf{R}_5\mathbf{z}\cdot(d_5\mathbf{z} + d_3\mathbf{R}_4^{-1}\mathbf{z} + a_2\mathbf{R}_4^{-1}\mathbf{x}).$$

With $\mathbf{R}_4^{-1}\mathbf{z} = \mathbf{y}$ and $\mathbf{R}_5\mathbf{z}\cdot d_5\mathbf{z} = 0$, this equation reduces to

$$\mathbf{t}\cdot\mathbf{p} = -d_3C_5 + a_2C_4S_5. \tag{39}$$

Computing $\mathbf{p}\cdot\mathbf{p}$

The inner product directly produces

$$ {}^1\mathbf{p}\cdot{}^1\mathbf{p} = \mathbf{p}\cdot\mathbf{p} = d_5^2 + d_3^2 + a_2^2 + 2a_2d_5S_4 \tag{40}$$

without any matrix operations.

Equation (39) can be used to define a real function of Θ_1:

$$f(\Theta_1) = -d_3C_5 + a_2C_4S_5 - \mathbf{t}\cdot\mathbf{p}. \tag{39$'$}$$

Values of Θ_1 that yield a solution to the inverse kinematics of this manipulator must be zeros of the function f. Equations (40), (38), and (37) provide a way to compute f, given Θ_1. With Θ_1 known, Eq. (34) gives ${}^1p_x$, ${}^1p_z$, ${}^1t_x$, and ${}^1t_z$. Equation (38) gives

$$C_4 = -{}^1p_z/d_5 \tag{41}$$

and

$$S_4 = u_4 \operatorname{Trig}(C_4), \tag{42}$$

where $u_4 = 1$ or -1 expresses a sign ambiguity and the function Trig is defined by $\operatorname{Trig}(x) = (1 - x^2)^{1/2}$.

The prismatic variable d_3 can then be found from (40):

$$d_3 = (\mathbf{p}^2 - a_2^2 - d_5^2 - 2a_2d_5S_4)^{1/2}. \tag{43}$$

From (37), the value of S_5 can be computed, if S_4 is not zero:

$$S_5 = {}^1t_z/S_4 \tag{44}$$

and

$$C_5 = u_5 \text{ Trig } (S_5), \tag{45}$$

where $u_5 = 1$ or -1 is another sign ambiguity. This additional sign ambiguity can be avoided if more equations involving Θ_2 are considered. Indeed, Eqs. (30) and (31), when applied to this manipulator, yield a system of two equations that can be readily solved for S_2 and C_2:

$$S_2 = (d_3{}^1p_x + k_0p_z)/(d_3^2 + k_0^2), \tag{46a}$$

$$C_2 = (k_0{}^1p_x - d_3p_z)/(d_3^2 + k_0^2), \tag{46b}$$

where $k_0 = (a_2 + d_5S_4)$. The value of C_5 can then be obtained from either (29),

$$C_5 = (tz - S_2C_4S_5)/C_2, \tag{47}$$

or (28),

$$C_5 = (C_2C_4S_5 - {}^1t_x)/S_2. \tag{47'}$$

With the computed values of d_3, C_4, C_5, and S_5, the value of $f(\Theta_1)$ is fully determined.

The derivative of f can also be evaluated. By differentiating (38) with respect to Θ_1, we obtain $dC_4/d\Theta_1$:

$$dC_4/d\Theta_1 = -{}^1p_x/d_5, \tag{48}$$

where we substituted $d({}^1p_z)/d\Theta_1 = {}^1p_x$. Using (33), we find the value of $dS_4/d\Theta_1$:

$$\frac{dS_4}{d\Theta_1} = -C_4\left(\frac{dC_4}{d\Theta_1}\right)\bigg/S_4 = \frac{C_4{}^1p_x}{S_4d_5}. \tag{49}$$

Differentiating (40) yields

$$\begin{aligned}\frac{dd_3}{d\Theta_1} &= -a_2d_5\left(\frac{dS_4}{d\Theta_1}\right)\bigg/d_3,\\ \frac{dd_3}{d\Theta_1} &= \frac{-a_2d_5C_4{}^1p_x}{S_4d_5d_3},\end{aligned} \tag{50}$$

and from (37), we get $dS_5/d\Theta_1$:

$$\frac{dS_5}{d\Theta_1} = \left[{}^1t_x - S_5\left(\frac{dS_4}{d\Theta_1}\right)\right]\bigg/S_4. \tag{51}$$

Once again, using (33), we have

$$\frac{dC_5}{d\Theta_1} = -S_5\left(\frac{dS_5}{d\Theta_1}\right)\bigg/C_5, \tag{52}$$

and we finally obtain $df/d\Theta_1$ by differentiating (39′):

$$\begin{aligned}\frac{df}{d\Theta_1} = a_2&\left(C_4\frac{dS_5}{d\Theta_1} + S_5\frac{dC_4}{d\Theta_1}\right)\\ &-\left(d_3\frac{dC_5}{d\Theta_1} + C_5\frac{dd_3}{d\Theta_1}\right).\end{aligned} \tag{53}$$

From the one-dimensional Newton-Raphson iterative method, we get a new estimate for Θ_1:

$$\Theta_{1,\text{new}} = \Theta_1 - \frac{f(\Theta_1)}{df/d\Theta_1}.$$

Once Θ_1 is obtained to the desired accuracy, the remaining joint variables Θ_2, Θ_4, Θ_5 are then computed from the values of their sines and cosines as obtained, along with d_3, from the last iteration. A vector equation in Θ_6 can be obtained from (6) by right multiplying both sides by $\mathbf{R}_6^{-1}\mathbf{R}_5^{-1}\mathbf{z}$:

$$\mathbf{R}\mathbf{R}_6^{-1}\mathbf{y} = \mathbf{R}_1\mathbf{R}_2\mathbf{R}_4\mathbf{z}, \tag{54}$$

where we used $\mathbf{R}_5^{-1}\mathbf{z} = \mathbf{y}$. This equation can be solved uniquely for Θ_6.

When the Θ_1 estimate is close enough to a solution, the complexity can be reduced by computing $df/d\Theta_1$ numerically at any iteration using the values of Θ_1 and $f(\Theta_1)$ in the preceding iteration:

$$\left(\frac{df}{d\Theta_1}\right)^i = \frac{f^{i-1} - f^i}{\Theta_1^{i-1} - \Theta_1^i},$$

where the superscript represents the iteration number at which the variable is computed. This saves the computational cost of Eqs. (48)–(53) and avoids the prob-

Table 3. GP66 Trajectory Tracking Points (all angles are in degrees, $a_2 = 0.36$, $d_5 = 0.19$)

θ_1	θ_2	d_3	θ_4	θ_5	θ_6
−19.072	54.427	1.192	−140.114	−137.013	−121.439
−15.319	54.980	1.090	−135.196	−135.357	−125.247
−11.061	55.823	0.992	−129.853	−133.343	−129.428
−6.234	57.063	0.901	−124.100	−130.873	−134.024
−0.773	58.831	0.820	−118.000	−127.817	−139.068
5.374	61.276	0.751	−111.700	−124.006	−144.568
12.239	64.532	0.697	−105.467	−119.245	−150.474
19.805	68.657	0.662	−99.716	−113.360	−156.644
27.968	73.551	0.649	−94.958	−106.315	−162.840
36.488	78.908	0.660	−91.649	−98.352	−168.788
45.000	84.279	0.694	−90.000	−90.000	−174.278

lem of special cases that occur when division by a number close to zero is needed in any of those equations.

The procedure just described was programmed to compute the joint variables for 10 equidistant points on a linear trajectory with constant orientation that will move the end-effector from the initial pose

$$\mathbf{P} = \begin{bmatrix} \sqrt{2}/2 & 0 & \sqrt{2}/2 & 1 \\ \sqrt{2}/2 & 0 & -\sqrt{2}/2 & -\frac{1}{2} \\ 0 & 1 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

to the position $[\frac{1}{2}, \frac{1}{2}, \frac{1}{10}]^T$.

Table 3 shows the output of the program. The maximum number of iterations needed per point was six. The guess for each point was the value of Θ_1 at the preceding point. The experiment was started with a guess of $-20°$. Convergence at every point was obtained when six or more points were taken along the trajectory. Although Table 3 gives the joint variables to only three decimal places, they were computed with a precision of 10^{-5}. The program was written in C and run on an AT&T 3B2/310 desktop computer. The kinematic inversion for the 11 points took 0.32 s, which gives an average time per kinematic inversion of 29.1 ms, clearly suitable for real-time high-precision inverse kinematics.

8. Conclusion

This paper has addressed the inverse kinematics problem of 6-DOF manipulators. The problem is simplified by a convenient choice of base and end-effector frames for manipulators with revolute first and last joints. The invariance of inner product under rotation is then shown to allow full simplification of the inverse kinematic equations without multiplying out the homogeneous matrices and performing the usual lengthy simplifications of the scalar equations. This simplification process also provides better insight into the structure of the inverse kinematic problem. We have also shown how the same techniques allow reduction of the complexity of the problem to four equations in only four of the unknowns. Although in this paper we only show this when the equations are expressed in base or first frames, it still holds in any of the frames along the manipulator structure (Doty 1986).

The paper also defines the important set of orthogonal manipulators and shows that there are only 24 distinct classes of orthogonal manipulators with 6-DOF capability. A simple notation for the 24 classes is proposed. It can be shown that the five classes of 6-DOF orthogonal manipulators with only two of the six twist angles equal to $\pi/2$ will always yield closed-form solutions.

Finally we provide a fast iterative inverse kinematic method based on a one-dimensional Newton-Raphson technique. This method neither requires computation of the Jacobian nor the inverse Jacobian of the manipulator. Its computational simplicity allows its use in real-time manipulator control. It can be applied to any manipulator that does not allow closed-form solutions, but for which knowledge of one of the joint variables allows closed-form solutions for the remaining joint variables. The convergence properties and possible improvements and generalization of this method are the subject of ongoing research. The method typically converges in five iterations for a guess within 10° from a solution although it has been observed that the convergence depends highly on the end-effector pose to be solved as well as the manipulator geometry. In some rare instances, a much closer guess is required before convergence can occur.

As examples, this paper presented the new inverse