

SOLVING THE 6R INVERSE POSITION PROBLEM USING A GENERIC-CASE SOLUTION METHODOLOGY

CHARLES WAMPLER and ALEXANDER MORGAN

Mathematics Department, General Motors Research Laboratories, Warren, MI 48090, U.S.A.

(Received 19 November 1989)

Abstract—This paper considers the computation of all solutions to the inverse position problem for general six-revolute-joint manipulators. Instead of reducing the problem to one highly complicated input-output equation, we work with a system of 11 very simple polynomial equations. Although the total degree of the system is large (1024), using the "method of the generic case" we show numerically that the generic number of solutions is 16, in agreement with previous works. Moreover, we present an efficient numerical method for finding all 16 solutions, based on *coefficient-parameter polynomial continuation*. We present a set of 41 test problems, on which the algorithm used an average of less than 10 s of CPU time on an IBM 370-3090 in double precision FORTRAN. The methodology applies equally well to other problems in kinematics that can be formulated as polynomial systems.

1. INTRODUCTION

The inverse position problem for six-link manipulators, in which the hand position and orientation are given and the joint displacements are to be found, is of fundamental importance to computer controlled robots. It is also equivalent to the displacement analysis of seven-link single-loop spatial mechanisms. While the problem can be expressed in closed-form for a variety of special cases, such as when three consecutive axes intersect in a common point, this appears to be impossible for more general cases. Of particular note is the most general 6R manipulator, which has six revolute joints (hence "6R") and general link parameters. In the absence of a closed-form solution, a problem can be considered "solved" when two objectives have been met: (1) a tight upper bound on the number of solutions has been established and (2) an efficient, numerically sound method for computing all solutions has been demonstrated. In this paper, we present a numerical approach, based on polynomial continuation, that addresses both objectives for the 6R problem. Moreover, the scope of the methodology extends beyond this problem to a host of other kinematic synthesis and analysis problems. This methodology is based on the "method of the generic case", a rigorous computational technique for proving results about polynomial systems. This includes a general method for finding (and proving) tight upper bounds on the number of solutions.

The inverse position problem for manipulators has been studied for at least two decades, while the analysis of spatial mechanisms extends back a century. A landmark treatise on the subject of 6R manipulators came in 1968 when Pieper [1] developed closed-form solutions for the case where any three consecutive axes intersect in a common point. His work also included a polynomial formulation for the general 6R problem, but it had a total degree of 64,000. The first major accomplishment in the attack on the general 6R problem came in 1973 when Roth *et al.* [2] showed that there are at most 32 solutions.† Albala and Angeles addressed the problem in 1979 [3]. Then in 1980, Duffy and Crane [4] derived a 32nd degree polynomial for the problem. Duffy [5] also presented lower degree polynomials for a wide variety of special cases involving prismatic or cylindrical joints as well as for the 6R case with the consecutive axes parallel. In 1984, Tsai and Morgan [6] cast the problem as eight second-degree equations (total degree 256), which they solved numerically using polynomial continuation. Reformulated as a two-homogeneous system, the Tsai-Morgan system was reduced to two-homogeneous degree 96 by Morgan and Sommese [7] and

†Unless there are an infinite number. This is a qualification we will generally omit, since it is always a possibility: for example, in singular configurations in which one or more axes may turn without affecting the hand position and orientation.

effectively to degree 64 in Ref. [8]. (See Appendix A.) Computations with both formulations supported the conjecture that, for generic parameters, there are always 16 (sometimes complex) finite solutions. Motivated by this result and numerical solutions by Crane, Primrose [9] was the first (1986) to prove conclusively that there are at most 16 solutions by showing that the remaining 16 solutions to the Duffy–Crane polynomial had pure imaginary parts. Finally, in 1988 Lee and Liang [10] were able to reduce the problem to a single 16th degree polynomial. This work was simplified and extended in Ref. [11]. Complementing these results, Manseur and Doty [12] have found an example where all 16 solutions are real.

Although a 16th degree polynomial reduction of the 6R problem has been established, it does not necessarily follow that the fastest, most numerically stable numerical method for this problem must be based on such a reduction. Issues of algebraic stability, for example, have lead some researchers [11] to evoke hundreds of digits of precision for portions of the computation. Further (numerical) analysis will be required to clarify which combinations of methods are most effective.

Almost two decades have elapsed from the time that Pieper first considered the general 6R manipulator problem as a polynomial system until a tight bound on the number of solutions was firmly established. This indicates the difficulty in reducing even a moderately complicated polynomial system to a single polynomial of minimum degree. In this paper we use a methodology for general polynomial systems that can be used both to determine the generic number of solutions numerically and also to calculate all of the geometrically isolated† solutions. This methodology does not involve reducing the system to a single polynomial. It is instead based on new developments in polynomial continuation. We apply this general methodology to a new model for the 6R inverse position problem, presented in Section 2. In Section 3, we give an overview of polynomial continuation and the related polynomial theory, focusing on the recent developments that have improved the method dramatically. Section 4 describes our code for solving the problem and documents its performance on a number of examples. Section 5 gives the conclusions.

2. MODELLING

We model the 6R manipulator problem using the Denavit–Hartenburg formalism [13] (see Fig. 1). That is, each link is represented by the line along its joint axis and the common normal to the next joint axis. (In the case of parallel joints, any of the common normals can be chosen.) We place a unit vector \mathbf{z}_i along joint axis i and a unit vector \mathbf{x}_{i+1} along its common normal to joint axis $i + 1$. (Boldface type denotes a vector in three-space.) A fixed vector \mathbf{x}_1 perpendicular to \mathbf{z}_1 defines a fixed reference frame, and an end-effector axis \mathbf{z}_7 is attached to link 6. The origin point O and the endpoint P are defined as in Fig. 1. Following the usual convention, each link has an offset d_i along \mathbf{z}_i and a length a_i along \mathbf{x}_{i+1} . Also there is a twist angle α_i between \mathbf{z}_i and \mathbf{z}_{i+1} . The joint rotation θ_i is the angle between \mathbf{x}_i and \mathbf{x}_{i+1} . The parameters d_i, a_i, α_i for $i = 1, \dots, 6$ are constants for a given 6R manipulator, whereas $\theta_1, \dots, \theta_6$ are the variable joint angles. Now, given an end-effector position $\mathbf{p} = \overrightarrow{OP}$ and end-effector orientation $\mathbf{x}_7, \mathbf{z}_7$, the problem is to find all the corresponding sets of joint angles $(\theta_1, \dots, \theta_6)$. (In the equivalent 7R spatial mechanism problem, \mathbf{z}_7 is the final joint axis and we wish to find the joint angles as \mathbf{z}_7 is rotated about \mathbf{z}_7 .)

We may formulate the problem as follows. (The symbols “ \cdot ” and “ \times ” represent the vector scalar product and the vector cross product operations, respectively.) The unit lengths imply for $i = 1, \dots, 7$

$$\mathbf{z}_i \cdot \mathbf{z}_i = 1, \quad (1)$$

$$\mathbf{x}_i \cdot \mathbf{x}_i = 1. \quad (2)$$

†A solution is *geometrically isolated* if it is not part of a positive-dimensional solution set (e.g. a curve or surface of solutions). We can be assured of finding only the geometrically isolated solutions using polynomial continuation (or any other general method). If a polynomial system does not have an infinite number of solutions, then every solution is geometrically isolated. A solution is said to be *nonsingular* if the Jacobian matrix of the system at the solution is nonsingular. Nonsingular solutions are geometrically isolated, but the converse is not true.

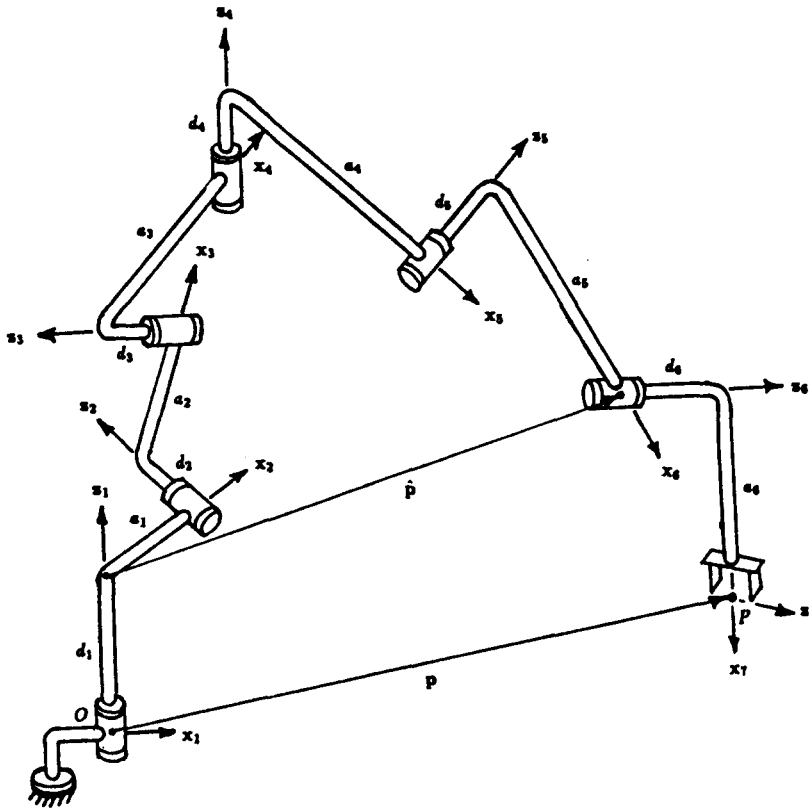


Fig. 1. Denavit-Hartenburg notation for a 6R manipulator.

The definition of x_{i+1} as the common normal to z_i and z_{i+1} gives for $i = 1, \dots, 6$

$$x_{i+1} \cdot z_i = 0, \quad (3)$$

$$x_{i+1} \cdot z_{i+1} = 0. \quad (4)$$

The definitions of the twist angles imply for $i = 1, \dots, 6$,

$$z_i \cdot z_{i+1} = \cos \alpha_i \quad (5)$$

$$\sin \alpha_i x_{i+1} = z_i \times z_{i+1}. \quad (6)$$

Similarly, the joint angle definitions imply for $i = 1, \dots, 6$

$$x_i \cdot x_{i+1} = \cos \theta_i \quad (7)$$

$$\sin \theta_i z_i = x_i \times x_{i+1}. \quad (8)$$

Finally, the endpoint position can be computed as

$$p = \sum_{i=1}^6 (d_i z_i + a_i x_{i+1}). \quad (9)$$

Given x_1, z_1, x_7, z_7, p , this may be viewed as a system of equations for the unknowns x_i, z_i , $i = 2, \dots, 6$ and θ_i , $i = 1, \dots, 6$. We have more equations than unknowns, but they can be shown to be consistent. Our first action is to reduce the number of variables and equations.

Equations (1)–(9) are basically just the definitions of the usual Denavit–Hartenburg parameters. From these relations, one can easily write x_i, z_i in terms of $x_{i-1}, z_{i-1}, \theta_i$ and thereby rederive the conventional transformation matrix formulation of the problem. We take an alternative approach, which is to first calculate x_i, z_i in equations (1)–(6) and (9) and then evaluate θ_i from equations

(7) and (8). That is, given all the joint axis directions and their common normals, we compute the joint angles as

$$\theta_i = \tan^{-1} \left(\frac{\mathbf{z}_i \cdot \mathbf{x}_i \times \mathbf{x}_{i+1}}{\mathbf{x}_i \cdot \mathbf{x}_{i+1}} \right). \quad (10)$$

If the joint twists for links 1–5 are general, that is, if $\sin \alpha_i \neq 0$, $i = 1, \dots, 5$, then we can solve for \mathbf{x}_{i+1} from equation (6):

$$\mathbf{x}_{i+1} = (\mathbf{z}_i \times \mathbf{z}_{i+1}) / \sin \alpha_i. \quad (11)$$

From this and equations (1) and (5), equations (2)–(4) follow easily. Also, we can calculate \mathbf{z}_6 as

$$\mathbf{z}_6 = \sin \alpha_6 \mathbf{z}_7 \times \mathbf{x}_7 + \cos \alpha_6 \mathbf{z}_7. \quad (12)$$

Defining the known quantity $\hat{\mathbf{p}}$ as

$$\hat{\mathbf{p}} = \mathbf{p} - d_6 \mathbf{z}_6 - a_6 \mathbf{x}_7 - d_1 \mathbf{z}_1, \quad (13)$$

a complete polynomial system of 12 equations in 12 unknowns \mathbf{z}_i , $i = 2, 3, 4, 5$ is as follows:

$$\mathbf{z}_i \cdot \mathbf{z}_i = 1, \quad i = 2, 3, 4, 5, \quad (14)$$

$$\mathbf{z}_i \cdot \mathbf{z}_{i+1} = c_i, \quad i = 1, 2, 3, 4, 5, \quad (15)$$

$$e_1 \mathbf{z}_1 \times \mathbf{z}_2 + \sum_{i=2}^5 (d_i \mathbf{z}_i + e_i \mathbf{z}_i \times \mathbf{z}_{i+1}) = \hat{\mathbf{p}}, \quad (16)$$

where $c_i = \cos \alpha_i$, and $e_i = a_i / \sin \alpha_i$, ($i = 1, \dots, 5$). Remember that \mathbf{z}_1 is given and \mathbf{z}_6 is computed in equation (12). (Note that \mathbf{x}_7 and \mathbf{z}_7 are also given.) This is the system of equations which we used for our computations. After solving for the joint axes \mathbf{z}_i , we find the common normals from equation (11) and then the joint angles from equation (10).

System (14)–(16) consists of two linear [equation (15) $i = 1$ and $i = 5$] and 10 quadratic equations, for a total degree of $2^{10} = 1024$. Because we take $\mathbf{z}_1 = (0, 0, 1)$, equation (15) simply sets the third component of \mathbf{z}_2 to be c_1 , so this variable can be easily eliminated. Thus, the system we actually submit to the numerical solver consists of 11 equations in 11 unknowns. Note that the sparsity of the system is a plus in minimizing the cost of function and Jacobian matrix evaluations. This sparsity could also be exploited to decrease the cost of solving linear systems involving the Jacobian matrix. Since we are mainly interested in general geometries, the restriction to nonzero twist angles is not viewed as a difficulty. For completeness, Appendix B shows how to write a similar system for a manipulator with parallel joints.

3. POLYNOMIAL THEORY AND POLYNOMIAL CONTINUATION

From now on we refer to the 6R inverse position problem as the “IPP”. In Section 3.1 we outline “the method of the generic case”, which is the basis from polynomial theory for our numerical technique. Then in Section 3.2 we present the numerical technique itself: “coefficient-parameter polynomial continuation”.

3.1. The method of the generic case

“The method of the generic case” refers to a mathematical theory analyzing the structure of the solution sets of polynomial systems. Here we outline the basic ideas. The results are precisely stated and rigorously proven in Ref. [14]. We denote by C^i the complex Euclidean space of dimension i , and by R^i the real Euclidean space of dimension i .

We begin with the assumption that we have a system of polynomial equations F that varies with a set of parameters Q ; thus, we have

$$F(c[q], w) = 0, \quad (17)$$

where w is a set of complex variables, $w \in C^n$, q is a set of parameters, say $q \in Q$, and c is a set of analytic functions giving the coefficients of the polynomials in terms of the parameters. Thus $c[q]$ denotes an analytic map from the parameter space Q to the coefficient space, which we may

take to be some C^1 . We need to have $w \in C^n$ rather than $w \in R^n$ for theoretical reasons, even if only real solutions have physical meaning. Similarly, we will assume Q is an open subset of some C^1 , although the theory allows much more general spaces. (See Section 4, discussion of “Part 2” of the problem set, for a case in which complex solutions have physical meaning for the IPP.)

We wish to talk about “possibly physical solutions”. We make this precise as follows: we assume we have a particular collection of polynomial “side conditions”, $s(c[q], w) = 0$; that is, additional polynomial equations that also vary with q and have w as the independent variable. We further assume that any solution obeying the side conditions cannot be a physical solution. The *possibly physical solutions* are those solutions to $F(c[q], w) = 0$ that do not obey the side conditions $s(c[q], w) = 0$. In any particular case, these possibly physical solutions may not be physical; for example, some of them may include imaginary numbers. We want the generic set of possibly physical solutions to be as small as possible. This will yield the tightest bound on the number of physical solutions. The best case occurs when there is a choice of parameters for which all the possibly physical solutions are physical, demonstrating that the bound cannot be any tighter. For the IPP we have such a tight bound, 16.

In physical problems, it is natural to conceptualize q as being the set of physical parameters. However, often it is desirable to let q be derived from these parameters rather than take q be the physical parameters themselves.

Regardless of the way in which q is defined, we have the following result: for all generic choices of q_0 in Q , the structure of the solution set of $F(c[q_0], w) = 0$ is the same. In practice, we identify a *random* choice of q_0 from Q with a *generic* choice. Although this is not strictly correct mathematically, it is plausible and defensible. The structure of the solutions of $F(c[q], w) = 0$ includes such qualities as the number of geometrically isolated finite solutions, the number of geometrically isolated solutions at infinity, the type and dimensionality of positive-dimensional solution components, the number of geometrically isolated strictly imaginary solutions and strictly real solutions. Also, if $s(c[q], w) = 0$ is a polynomial system that also varies with q and if a certain number of solutions of $F(c[q_0], w) = 0$ also satisfy $s(q_0, w) = 0$, then that same number of solutions to $F(c[q], w) = 0$ will also satisfy $s(c[q], w) = 0$ for any generic choice of q .

Suppose that we numerically solve $F(c[q_0], w) = 0$ for a randomly chosen q_0 . Then, with the qualifications that

- (1) q_0 is random instead of generic,

and

- (2) the computational solution of $F(c[q_0], w) = 0$ is subject to error,

the revealed structure of the solution sets of generic $F(c[q], w) = 0$ is rigorously proven. In particular, the number of nonsingular possibly physical solutions is determined.

Since generic q are dense in Q , we obtain results for nongeneric systems from arguments about limits. (The arguments include some subtleties. See Ref. [14].) Thus, for example:

- the number of geometrically isolated (nonsingular, respectively) finite solutions of a nongeneric $F(c[q], w) = 0$ is bounded above by the generic number,
- the number of geometrically isolated (nonsingular, respectively) possibly physical solutions is bounded above by the generic number.

In Section 4, we summarize the result of applying the method of the generic case to the IPP system developed in Section 2. In Appendix A we do this for the formulation of the IPP from Ref. [6]. In both cases a description of the solution set results, including a computational proof that there are exactly 16 (generic) possibly physical solutions.

3.2. Coefficient-parameter polynomial continuation

Continuation is the process of tracking solutions of a previously solved problem as its parameters are gradually changed to those of an unsolved target problem of similar form. Such an approach is particularly effective for solving systems of polynomial equations because of the completeness of the theory in this case.

“Polynomial continuation” refers to methods for computing the full set of geometrically isolated solutions to systems of n polynomial equations in n unknowns using numerical continuation (path tracking) techniques. Polynomial continuation consists of two parts:

(1) defining the continuation system,

and

(2) numerically computing the resulting continuation paths.

We are focusing here on Part 1. Part 2 is important, but reliable techniques of polynomial path tracking are available and they need not be customized to the specific problem being solved. Several approaches are surveyed in Refs [16, 17], although we favor the predictor-corrector technique described in Ref. [17, Chap. 4] because it is customized to polynomial paths. See Ref. [18].

We use a theory of polynomial continuation called coefficient-parameter continuation, which we will outline briefly here. An elementary overview of polynomial continuation is given in Ref. [17], while Refs [18–21] summarize a number of recent developments. The theory of coefficient-parameter polynomial continuation is precisely developed in Ref. [14].

Consider a family of a polynomial systems of the form of equation (17). Suppose we let the parameters q be continuous functions of a dummy parameter t . Then the solutions of equation (17) will vary continuously with t . The continuation method consists of tracking solutions from an initial set of parameters q_0 , for which the solutions are known, to a target set q_1 . (These “tracked solutions” are called *continuation paths*.) Typically, we choose parameter formulas to be simple linear interpolation, such as

$$H(w, t) \equiv F(c[(1-t)q_0 + tq_1], w), \quad (18)$$

where t increases from 0 to 1. We call this a *coefficient-parameter continuation* to distinguish it from *coefficient continuation*, in which the coefficients are interpolated directly. $H(w, 0)$ is the *start system* and $H(w, 1)$ is the *target system*. The continuation paths are collectively the solutions to $H(w, t) = 0$ for $0 \leq t \leq 1$.

Now we proceed as follows: we wish to construct a computer code to solve equation (17) for any choice of $q = q_1$. We are interested in only the possibly physical solutions. Associated with equation (17) is a system of side conditions $s(c[q], w) = 0$ that picks out solutions that are not possibly physical, as discussed in Section 2.1. The first step of our technique is to choose q_0 from Q at random, and solve† $F(c[q_0], w) = 0$.

We discard the solutions that satisfy $s(q_0, w) = 0$ and save the rest. (For the IPP formulation given in Section 2, this amounts to discarding the solutions at infinity.) The solutions we have saved are the *start points*. Then we define the continuation system $H(w, t)$ by condition (18). (There are a number of alternative ways to set up H , described in Refs [14, 21]. Note that here q_0 consists of random complex numbers.) Each start point is a solution to the start system $H(w, 0) = 0$.

In the second step of the methodology, we proceed, for any choice of q_1 , to track the continuation paths defined by equation (18) as t goes from 0 to 1, beginning at each start point. The endpoints of these paths will include all the (geometrically isolated) possibly physical solutions of the target system.

Since the first step (solving a random problem) is done only once, a moderately large degree system can be tolerated. For many problems in kinematics, we can expect the number of possibly physical solutions to be only a small fraction of the total, so that the computation cost in the second step will be significantly reduced over the first. See Section 4.

4. NUMERICAL RESULTS

The continuation system we use is equation (18) from Section 3, where F is defined by equations (14)–(16) and the parameters q are the 20 constants c , for $i = 1-5$, e , for $i = 1-5$, d , for $i = 2-5$, z , and \hat{p} (these last two accounting for three parameters each), respectively. Note that these are

†A traditional continuation method can be used, for which published computer codes are available; for example, CONSOL8T from Ref. [17] or POLSYS/HOMPACK from Ref. [16].

Table 1. Random coefficients for the IPP2 start system

Parameter	Real Part	Imaginary Part
c_1	0.642935654806024D + 0	0.819555356316414D + 0
c_2	0.266880023988051D + 0	0.452565255665832D + 0
c_3	0.264255517449783D + 0	0.342483846503455D + 0
c_4	0.126010863921692D + 0	0.864590917688474D + 0
c_5	0.179560356711590D + 0	0.870916656600457D + 0
e_1	0.496254299763677D + 0	0.546020011741348D + 0
e_2	0.958341609741183D + 0	0.847442419999425D + 0
e_3	0.964759562277322D + 0	0.713970745189727D + 0
e_4	0.706319991204509D + 0	0.120097702052590D + 0
e_5	0.482079337580217D + 0	0.307430483491771D + 0
d_2	0.984138451804029D + 0	0.414967172346223D + 0
d_3	0.353268870498141D + 0	0.389909226887539D + 0
d_4	0.204379350350791D + 0	0.374294529683539D - 2
d_5	0.907681632783351D + 0	0.405209293447115D + 0
$z_{0,1}$	0.352598136811139D + 0	0.116888144319091D + 0
$z_{0,2}$	0.539042485524988D + 0	0.687058436891675D + 0
$z_{0,3}$	0.391154215376446D + 0	0.128900893181708D + 0
$\tilde{p}_{0,1}$	0.437312713587744D + 0	0.914780691357009D + 0
$\tilde{p}_{0,2}$	0.719086796332704D + 0	0.691791591267247D + 0
$\tilde{p}_{0,3}$	0.941279842398217D + 0	0.903185535898956D - 01

The "D" indicates powers of 10. Thus

$$0.374294529683539D - 2$$

denotes

$$0.374294529683539 \times 10^{-2}.$$

not the "natural" parameters of the problem, which are a_i , d_i , α_i , \mathbf{p} , \mathbf{x}_7 , and \mathbf{z}_7 . Instead, the parameters we choose are derived from the natural parameters by the simple formulas given in Section 2.

The start system parameters q_0 were chosen as random complex numbers (given in Table 1). This implements the "method of the generic case", discussed in Section 3. We solved this randomly chosen start system using a two-homogeneous version of the CONSOL8T code from Ref. [17]. (See Refs [7, 8, 19, 21] for a discussion of the two-homogeneous approach.) Since the two-homogeneous Bezout number of the system is 320, we therefore tracked 320 paths for this computation. [We could have used a more traditional one-homogeneous code (e.g. the original CONSOL8T), in which case we would have had to have tracked 1024 paths, the total degree number.] The run took about 30 min of CPU time on an IBM 370-3090 in double precision FORTRAN. In this case, being at infinity rules out a solution as being possibly physical.† Of the 320 continuation path endpoints yielding solutions to this system, only 16 were finite (i.e. not at infinity). Therefore, the IPP system has no more than 16 finite (geometrically isolated) solutions for *any* choice of parameters. This computation amounts to a new proof of this important result. Aside from this theoretical result, we have used this computation to construct a continuation computer code to solve the IPP.

Thus, defining the continuation system by condition (18) and using as start points the 16 finite solutions discovered in the computation discussed above, we can solve any IPP problem (defined by a choice of q_1) by path tracking the associated 16 paths. [As noted in Section 2, if any twist angle equals zero, we must use a modification of equations (14)–(16). See Appendix B.] We used the predictor-corrector path tracker from CONSOL8T [17] to compute the continuation paths. We call the resulting method IPP2.

To test IPP2 we solved a collection of test problems (in five parts, as described below) and compared the results to the IPP1-64 method from Ref. [8]. IPP1-64 is discussed in Appendix B. We give both CPU and NFE values in Table 2. CPU denotes seconds of CPU time on an IBM 370-3090 using double precision FORTRAN with the OPT(3) option. The CPU values, although of interest, should be viewed with some suspicion. They vary because of the time-sharing environment and other factors. (We experienced timer variations of several tenths of a second in rerunning sample problems.) The NFE value denotes "number of function evaluations". For the path tracker we are using, this is the number of function and Jacobian matrix evaluations and also the number of linear systems solved. Thus, it is a fairly good representation of the total

†To implement the two-homogeneous presentation of our system, we must divide the variables into two groups; in this case, $\{z_2, z_4\}$ and $\{z_1, z_3\}$. The side condition is $w_{0,1}w_{0,2} = 0$, where $w_{0,i}$ represents the homogeneous coordinate for the i th group.

Table 2. Run statistics for problem sets solved with IPP1-64 and IPP2 (see Section 4)

Prob.	IPP1-64		IPP2	
	NFE	CPU (s)	NFE	CPU (s)
1	16359	20.3	4995	8.0
2	18766	22.8	8262	13.0
3	20142	24.4	13078	20.5
4	24666	30.4	8402	13.3
5	24665	30.5	8368	13.4
6	18000	22.2	8087	13.0
7	15067	19.0	5032	8.1
8	14996	18.6	5033	8.1
9	14719	18.2	4977	8.0
10	14250	17.7	5251	8.5
11	13686	17.2	5512	8.9
12	14934	18.6	6149	9.8
13	13766	16.7	3418	5.6
14	13847	17.3	4061	6.5
15	13268	16.4	4057	6.6
16	15930	19.6	4756	7.6
17	18787	23.1	5444	8.6
18	20369	24.6	5639	8.9
19	22395	27.1	7857	12.3
20	21307	25.5	5949	9.3
21	14294	17.5	5250	8.4
22	13842	17.0	4946	8.0
23	14250	17.6	5077	8.2
24	14142	17.1	5026	8.2
25	13280	16.1	5172	8.4
26	13321	16.2	4504	7.3
27	14099	17.1	4698	7.7
28	14818	18.0	4912	8.0
29	14500	17.6	4189	6.9
30	14486	17.8	4130	6.8
31	15600	18.9	4145	6.8
32	15461	18.7	4612	7.5
33	15216	18.4	4424	7.2
34	15565	18.8	3883	6.5
35	14911	18.0	3650	6.0
36	15377	18.6	3589	5.9
37	15450	18.8	3624	5.9
38	16308	19.8	3651	6.0
39	16607	20.2	3739	6.1
40	16139	19.6	4077	6.6
41	16384	19.9	4172	6.8

computational work; it subsumes the amount of algebra needed to evaluate the system (and its partial derivatives) and to solve the linear systems (via Gaussian elimination). It is generally proportional to the CPU timings for comparisons of the same method on different problems.

However, in comparing IPP1-64 and IPP2, the NFE values are not comparable without making adjustments based on the following facts: The IPP1-64 system consists of eight equations, while the IPP2 systems consists of 11 (thus, the one does 8×8 linear algebra, the other 11×11). On the other hand, IPP1-64 requires more algebra for each function and Jacobian matrix evaluation than IPP2 (thus, the "sparsity" of the IPP2 system referenced in Section 2). In fact, on the problems we tested, IPP1-64 averaged about 815 function evaluations per second, and IPP2 about 630. Thus, IPP2 takes about 1.3 times longer per function evaluation than IPP1-64. In addition, the different structures of the models yield computational differences, the principle being:

- (1) The IPP2 model becomes more and more ill-conditioned as twist angles are chosen closer and closer to zero. This is not the case for IPP1-64, which, in fact, allows zero twist angles,

and

- (2) IPP2 tracks only 16 paths, while IPP1-64 tracks 64.

Typically, the ratio of 4 to 1 paths gives IPP2 a considerable edge over IPP1-64. It is generally 2 to 3 times faster. On the other hand, on problems with nearly zero twist angles, the two codes can run at almost the same speed. In practice, one would run the alternative version of IPP2 noted in Appendix B when twist angles are zero, but these tests measure the sensitivity of the code to this factor. The reliability and accuracy of both programs is excellent. Each program found all

geometrically isolated solutions to all problems. However, this reliability is strongly a function of the choice of path tracker, and we have chosen one which is less sensitive to ill-conditioned problems. (Note Ref. [18] on this important issue.)

Now we will describe the five problem sets in detail. All twist angle values given below are in degrees. The solutions to the first four problem sets are listed in Table 3.

Part 1 consists of Problems 1–3. The first problem (given in Table 4) is a moderately scaled random manipulator with a moderate random hand position and orientation. It provides a nominal start problem for this series of three. The second problem is identical to the first, except that the second and fourth twist angles are changed to 1° and 359° , respectively. The third problem is identical to the first, except that these angles are changed to 0.1° and 359.9° , respectively. This series is designed to quantify the nominal performance of IPP2 and then the degradation in performance caused by nearly zero twist angles. The IPP1-64 performance provides a control, since this code is not sensitive to nearly zero twist angles. All runs produced the correct two real solutions, but the computational work increases dramatically for IPP2 from Problem 1–3. The ratio of NFE's is 2.6 from Problem 1–3 for IPP2, while, for IPP1-64, this ratio is only 1.2. In comparing the CPU values for IPP1-64 and IPP2, we see that IPP2 is 2.5 times faster for the first problem, but only 1.14 times faster for the third.

Part 2 consists of Problems 4 and 5. These problems show the increased challenge of solving the IPP when the manipulator is near the boundary of its reachable area. Table 5 gives the Problem 4 data. This problem has two real solutions. Problem 5 is the same as Problem 4, except that the first three link lengths have been rounded to four digits from six digits. (Precisely, the rounded link lengths are $a_1 = 0.123D + 3$, $a_2 = 0.1426D + 3$ and $a_3 = 0.1625D + 3$.) The run statistics are virtually identical, but this problem has four real solutions. We observe that Problem 4 has a complex pair of solutions with small imaginary parts (on the order of 10^{-2}). This complex pair resolves into a pair of (ill-conditioned) real solutions under the perturbation of the polynomial system that results from rounding the first three link lengths to four digits. This pair of problems points out several subtleties:

1. The "nearly real" pair of solutions to Problem 4 indicates that the manipulator could almost (but not quite) attain the specified hand position and orientation with an additional pair of configurations. A user might want to be informed of this fact, which would be signaled by a display of the magnitudes of the imaginary parts of the complex solutions. Thus, knowing the complex solutions yields physical information.
2. Ill-conditioned endpoints can occur for physically meaningful solutions, and the path tracker must be able to deal with them.

Part 3 consists of Problem 6–12. These are variations of the example of Manseur and Doty [12] that gives 16 real solutions to the IPP. In fact, Problems 6–9 have this property, and then Problems 10–12 have 8, 4 and 0 real solutions, respectively. Problem 6 is defined in Table 6. It is identical with the Manseur and Doty example, except that their three zero twist angles have been replaced by 1° angles. In Problem 7, these 1° angles are replaced by 10° angles. Keeping the twist angles fixed as in Problem 7, the rest of the problems in this set replace each of the manipulator parameters that have the value of 0.0 with larger and larger values; namely, 0.01, 0.1, 0.3, 0.5 and 1.0 in Problems 8–12, respectively. Of interest is the persistence (and finally, degradation) of the "16 real solutions" property under greater and greater perturbations, and the stabilizing of the IPP2 run statistics as soon as the twist angles become larger than 1° .

Part 4 consists of Problems 13–20. It explores how increasingly severe singularities due to special hand positions and orientations affect the performance of the solvers. Problem 13 is a moderate problem, the manipulator from the second example of Ref. [6], with a randomly chosen hand position and orientation (Table 7). For Problems 14–16 this same manipulator is solved using a hand position and orientation that forces $(\theta_2, \theta_3, \theta_4, \theta_5)$ to include at least one solution with values $(0, 0, 0, 0)$, $(180, 180, 0, 0)$ and $(180, 180, 180, 180)$, respectively (Table 8). Thus at least one singular solution occurs. In fact, Problems 14 and 15 include two singular endpoints and Problem 16 has three. Compare, in particular, the NFE values for Problem 16 with those of Problems 14 and 15 (in Table 2). To "make matters worse" for Problem 17, we set all the offsets (d_i) equal to zero,

Table 3. Real solutions to the test problems (θ_j for $j = 1-6$, in degrees)

Prob.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	-98.3580	-162.6711	76.6759	-5.5721	73.4399	39.0772
	-118.1634	134.1567	156.0316	-12.2721	84.8259	43.4999
2	-94.0000	-174.0000	79.0000	11.0000	67.0000	33.0000
	-113.8130	75.1231	-153.5567	2.6442	77.2033	26.6143
3	-93.4624	-174.4208	78.3077	13.0225	66.0180	32.4839
	-113.3878	68.5254	-147.5634	2.6050	77.0359	25.0332
4	143.0000	0.9883	-0.0059	0.1527	167.0000	53.0000
	144.3383	-0.9370	0.1002	0.5868	166.5701	53.0388
5	142.4743	1.7446	-0.0475	-0.0239	167.1868	53.0011
	144.8641	-1.6935	0.1420	0.7517	166.4180	53.0695
	142.8577	1.0848	0.1358	-8.8051	-167.0113	76.7711
	144.4604	-1.2252	0.2690	-8.6323	-166.5008	77.7557
6	2.5172	108.0759	112.0431	-10.5230	0.0051	-0.1095
	2.5172	108.0759	-67.9569	-169.4770	179.9949	179.8905
	88.6785	-176.7247	3.2709	-116.7581	22.8041	-39.5633
	88.6785	-176.7247	-176.7291	-63.2419	157.1959	140.4367
	168.3219	-103.8922	146.6038	-17.2409	-171.8792	98.1651
	168.3219	-103.8922	-33.3962	-162.7591	-8.1208	-81.8349
	113.8436	5.3064	2.2557	-124.0758	-117.0152	136.6227
	113.8436	5.3064	-177.7443	-55.9242	-62.9848	-43.3773
	-12.9429	-105.0963	65.0246	176.9766	172.5830	100.5782
	-12.9429	-105.0963	-114.9754	3.0234	7.4170	-79.4218
	-96.2845	-6.2736	179.9689	38.4860	52.5499	-39.4047
	-96.2845	-6.2736	-0.0311	141.5140	127.4501	140.5953
	-120.7884	172.3344	0.9272	148.6680	-33.2848	-37.1791
	-120.7884	172.3344	-179.0728	31.3320	-146.7152	142.8209
	-178.1262	108.1916	32.2662	-174.3067	-15.3254	-0.4195
	-178.1262	108.1916	-147.7338	-5.6933	-164.6746	179.5805
7	23.3264	113.4456	112.1746	-34.9459	0.4566	-1.0267
	23.3264	113.4456	67.8254	-145.0541	179.5434	178.9733
	88.9601	176.2877	17.8067	118.4329	28.1414	35.1583
	88.9601	-176.2877	-162.1933	-61.5671	151.8586	144.8417
	113.7476	9.7132	13.8371	-125.3720	123.4263	129.5092
	113.7476	9.7132	-166.1629	-54.6280	56.5737	50.4908
	165.3442	-101.2557	147.9020	-24.4507	179.2226	97.8675
	165.3442	-101.2557	-32.0980	-155.5493	0.7774	-82.1325
	-21.9930	-101.7730	59.6376	162.4936	163.5698	100.5588
	-21.9930	-101.7730	-120.3624	17.5064	16.4302	-79.4412
	-96.2859	-11.1618	172.0617	37.9273	58.0958	-37.0103
	-96.2859	-11.1618	-7.9383	142.0727	121.9042	142.9897
	-121.2563	165.2800	175.7050	31.0083	-152.2889	145.3650
	-121.2563	165.2800	-4.2950	148.9917	-27.7111	-34.6350
	-161.3488	115.0559	25.4373	169.1701	-15.4092	-4.4428
	-161.3488	115.0559	-154.5627	10.8299	-164.5908	175.5572
8	22.4916	112.0565	114.1508	-33.9243	0.2336	0.3106
	22.4916	114.8123	-69.3955	-144.5878	179.5534	178.7418
	88.8581	-177.3825	-161.0037	-62.6399	153.0120	144.9976
	89.7344	-174.8276	16.2300	-118.9962	28.2052	-35.3762
	164.9044	-101.7499	-32.3602	-154.6302	0.2461	-81.1792
	165.8229	-101.8299	149.0939	-24.9005	-179.8072	97.3188
	113.8852	8.5343	-165.0086	-54.3529	-56.7753	-50.4638
	115.2645	10.4586	13.4153	-125.1292	-122.1107	129.0219
	-21.2557	-102.3271	59.2809	163.0431	163.6912	99.5142
	-22.6369	-99.5565	-121.3900	17.6573	16.7902	-78.1195
	-94.1474	-13.0521	174.2336	36.9809	57.3119	-37.5845
	-95.8677	-10.3109	-8.7075	142.4922	122.0354	142.8451
	-121.8303	166.2178	-5.1467	150.7160	-28.7019	-34.2967
	-122.0697	163.8500	177.1279	30.1500	-151.9854	145.6911
	-160.8461	115.4933	-154.2453	10.4749	-163.7822	174.8222
	-162.0120	115.0121	24.7543	170.7711	-16.3489	-3.3858
9	16.8717	98.2950	131.5822	-26.4830	-1.3744	4.2524
	30.4278	127.2051	-85.2921	-138.4371	179.3597	174.7179
	84.8749	171.6077	-148.7853	-75.1136	163.3560	147.3729
	98.3927	-161.2939	1.7567	-123.1325	26.2868	-37.8298
	114.3602	-2.0962	-154.7715	-52.0886	-57.6938	-49.8448
	128.4112	17.9808	9.5063	-123.4288	-110.3114	124.1456
	159.9762	-106.7078	-34.5365	-146.0969	3.8597	-72.1292
	169.6530	-107.1256	159.9721	-29.4477	-170.9979	92.4108
	-15.1681	-106.8375	54.7481	167.5683	164.5775	89.4197
	-33.1818	-73.2506	-134.9465	19.6907	23.2223	63.2920
	-67.4887	-34.6633	-163.0851	27.8066	44.1617	45.1417
	-92.0356	-2.6492	-15.7344	145.9543	123.5967	141.4979
	-125.3293	175.5695	-13.6346	164.8421	-37.7510	31.9035
	-132.1602	148.6549	-168.6365	20.8289	-150.0883	150.2847
	-154.3574	121.9345	-152.8378	9.6291	-156.1923	166.6609
	-167.7509	112.5338	19.1076	175.5597	-24.6721	7.5558

—continued opposite

Table 3—continued

10	14.6997	51.2019	173.1827	-15.6332	-4.5602	-0.4541
	102.1531	-25.5049	-134.1903	-44.2042	-50.3553	-43.8796
	156.2957	43.8474	1.3873	-123.9633	-82.1190	104.0899
	176.2835	-118.5406	-174.7523	-42.5474	-150.3013	80.2031
	179.5357	91.5570	6.7962	-146.1939	-46.9120	46.5374
	-4.1009	-110.8949	33.8088	175.3863	165.7201	62.8124
	-82.4338	14.8241	-32.5608	151.9686	129.8160	137.8728
-129.6185	-159.9117	-36.2711	-167.1353	-57.7226	-27.8057	
11	3.0460	-99.7140	-1.1005	179.5759	167.8233	38.3513
	-69.1949	34.1054	-53.1396	156.7353	140.8344	132.0347
	-146.0495	-134.7539	-65.9888	-131.8841	-74.1285	-15.8973
	-177.3113	-126.9879	-144.1177	-61.9760	-125.1324	59.5309
12	No real solutions.					
13	1.3571	21.8656	49.5020	-135.3516	177.2369	118.2644
	22.0000	11.0000	73.0000	-86.0000	-163.0000	67.0000
	22.2259	-83.0189	-155.8732	-64.4355	79.3048	138.8233
	33.3157	9.1797	89.1082	-68.5913	-154.5087	52.2032
	55.8623	-13.4494	-143.1730	-142.0105	63.7681	157.7112
	122.3995	125.6131	57.1077	-64.9907	7.7464	103.8062
14	13.2851	12.5493	-1.3806	28.9590	8.8568	80.5781
	22.0000	0.0000	0.0000	0.0000	0.0000	67.0000†
	90.3382	-8.8447	111.2940	23.9813	28.5286	46.5138
15	10.0457	-4.7161	26.5377	17.7366	-146.0340	-97.1835
	22.0000	180.0000	-180.0000	0.0000	0.0000	67.0000†
	36.6812	92.1914	-149.8650	125.9562	6.5082	100.1607
	42.7762	-3.1257	89.7629	32.1047	-129.8878	-77.3090
	82.7315	162.1436	98.7035	-37.1989	-28.1860	56.7000
16	22.0000	179.9997	179.9999	-180.0000	-179.9999	67.0003‡
	57.5074	-32.4328	-168.7493	90.0807	138.4922	-15.6728
	141.6415	143.6943	0.2970	17.2663	126.8306	158.5497
	-53.9826	-40.6097	-42.2725	-79.6420	78.0916	-9.2603
17	22.0000	0.0000	0.0000	0.0000	0.0000	67.0000§
18	22.0000	0.0000	0.0000	0.0000	0.0000	67.0000§
19	22.0000	0.0000	180.0000	0.0000	0.0000	67.0000§
	22.0000	180.0000	0.0000	0.0000	-180.0000	-113.0000§
20	No geometrically isolated real solutions.					

†Multiplicity two. ‡Multiplicity three. §Multiplicity four.

and generated a hand position and orientation yielding a solution with theta values (0, 0, 0, 0). (The hand positions and orientations for Problems 17–20 are given in Table 9.) Then in Problem 18 we set all the link lengths equal to 0.45 and otherwise the parameters are as in Problem 17. In Problem 19 the hand position and orientation for this last manipulator was chosen with one theta set equal to (0, 180, 0, 0), and finally for Problem 20 the same manipulator with theta set equal to (180, 180, 180, 180). For Problems 17 and 18 the number of continuation paths with singular endpoints is 8, for Problems 19 and 20 it is 12. Problem 20 is particularly degenerate. It possesses no real geometrically isolated solutions at all! [It has exactly two complex pairs of geometrically

Table 4. Problem I

Manipulator parameters a , d , and α (i.e., the first row gives a_j for $j = 1$ to 6, the second row d_j for $j = 1$ to 6, and the last row α_j for $j = 1$ to 6):						
0.7530D + 0	0.2190D + 0	0.4320D + 0	0.3430D + 0	0.4310D + 0	0.6420D + 0	
0.7550D + 0	0.9320D + 0	0.6530D + 0	0.3380D + 0	0.1280D + 0	0.5260D + 0	
0.2200D + 2	0.1000D + 2	0.7000D + 2	0.3490D + 3	0.8800D + 2	0.2600D + 2	
Hand position:						
-0.233420031755010D + 1	0.216941600256900D + 0	0.277358043344060D + 1				
Hand orientation matrix:						
-0.454444346169680D + 0	-0.458410443991960D - 1	-0.889594815004450D + 0				
0.260250280436430D + 0	0.948269069850210D + 0	-0.181811888220130D + 0				
0.851909694608040D + 0	-0.314140684747780D + 0	-0.419005372780620D + 0				

Table 5. Problem 4

Manipulator parameters a , d , and α (i.e., the first row gives a_j for $j = 1$ to 6, the second row d_j for $j = 1$ to 6, and the last row α_j for $j = 1$ to 6):

0.123366D + 3	0.142564D + 3	0.162544D + 3	0.343000D + 0	0.789300D + 0	0.123000D + 0
0.968500D + 0	0.563000D + 0	0.190000D + 0	0.163000D + 0	0.100000D + 0	0.876000D + 0
0.100000D + 1	0.350000D + 2	0.200000D + 2	0.660000D + 2	0.180000D + 3	0.370000D + 2

Hand position:

-0.34499607188930D + 3	0.25338291709575D + 3	0.27112740408867D + 1
------------------------	-----------------------	-----------------------

Hand orientation matrix:

0.74470324934565D + 0	-0.66674687490185D + 0	-0.29422359229639D - 1
0.31353420472692D + 0	0.38842979699426D + 0	-0.86649789109568D + 0
0.58916328201975D + 0	0.63605887904785D + 0	0.49831288314081D + 0

Table 6. Problem 6

Manipulator parameters a , d , and α (i.e., the first row gives a_j for $j = 1$ to 6, the second row d_j for $j = 1$ to 6, and the last row α_j for $j = 1$ to 6):

0.30D + 00	0.10D + 01	0.00D + 00	0.15D + 01	0.00D + 00	0.00D + 00
0.00D + 00	0.00D + 00	0.20D + 00	0.00D + 00	0.00D + 00	0.00D + 00
0.90D + 02	0.10D + 01	0.90D + 02	0.10D + 01	0.90D + 02	0.10D + 01

Hand position:

-0.1140175D + 01	0.0000D + 00	0.0000D + 00
------------------	--------------	--------------

Hand orientation matrix:

-0.760117D + 00	-0.641689D + 00	0.102262D + 00
0.133333D + 00	0.000000D + 00	0.991071D + 00
-0.635959D + 00	0.766965D + 00	0.855580D - 01

Table 7. Problem 13

Manipulator parameters a , d , and α (i.e., the first row gives a_j for $j = 1$ to 6, the second row d_j for $j = 1$ to 6, and the last row α_j for $j = 1$ to 6):

0.45D + 0	0.55D + 0	0.75D + 0	0.75D + 0	0.55D + 0	0.45D + 0
0.50D + 0	0.60D + 0	0.40D + 0	0.10D + 1	0.40D + 0	0.60D + 0
0.80D + 2	0.93D + 2	0.12D + 3	0.12D + 3	0.93D + 2	0.80D + 2

Hand position:

0.16829945912573D + 1	0.48415813071319D + 0	0.82083875430520D + 0
-----------------------	-----------------------	-----------------------

Hand orientation matrix:

-0.58314721777709D + 0	0.18785983973287D + 0	0.79034676124748D + 0
0.39371509906310D + 0	0.91635422339387D + 0	0.72686711563778D - 1
-0.71058267862971D + 0	0.35355850701653D + 0	-0.60833267128083D + 0

isolated solutions. The rest of the solutions are included in positive-dimensional solution sets. However, some of these include real (and thus physical) solutions.] The path tracker abandoned some paths near their endpoints, due to the high degree of singularity, which is why Problem 20 is solved in less time than Problem 19. The program reports "no solutions" for this last problem, which we believe is sensible for such a degenerate case.

Part 5 is taken from Ref. [10], where the problems and their solutions are given in Table 1 (10 problems) and Table 2 (11 problems) of that reference. (The Table 1 problems are taken from Ref. [4].) We have numbered these as Problems 21–30 and 31–41, respectively. In Table 2, we include our run statistics. No run statistics are given in Refs [4, 10].†

†For methods involving reduction computations, the total CPU cost would include the cost of generating the polynomial to be solved, the cost of solving the polynomial, and the cost of back substituting to generate the physical solutions. The need for very high precision to assure the numerical stability of the elimination process can significantly increase the CPU cost.

Table 8. Hand positions and orientations for Problems 14–16

 Problem 14 hand position:

$$0.32597195693809D + 1 \quad 0.54315555639162D + 0 \quad 0.57283794190814D + 0$$

Problem 14 hand orientation matrix:

$$\begin{array}{lll} 0.64815428818396D + 0 & 0.79161767330574D - 01 & 0.75738328031575D + 0 \\ -0.56119424077973D + 0 & -0.62263053294554D + 0 & 0.54533681661851D + 0 \\ 0.51473978166099D + 0 & -0.77850153117194D + 0 & -0.35913552196705D + 0 \end{array}$$

Problem 15 hand position:

$$0.21439821463962D + 1 \quad 0.36828870507612D + 0 \quad 0.62213244813290D + 0$$

Problem 15 hand orientation matrix:

$$\begin{array}{lll} 0.98126643087233D - 1 & -0.40480958842247D + 0 & 0.90912065152963D + 0 \\ 0.80017195261471D + 0 & 0.57524060710887D + 0 & 0.16977364395507D + 0 \\ -0.59168911445555D + 0 & 0.71079352913082D + 0 & 0.38036370854822D + 0 \end{array}$$

Problem 16 hand position:

$$0.12868352974634D + 1 \quad -0.15143991415139D + 1 \quad 0.16034363026702D + 1$$

Problem 16 hand orientation matrix:

$$\begin{array}{lll} 0.30240098111430D + 0 & 0.21069244605705D + 0 & 0.92960332389445D + 0 \\ 0.29457522407360D + 0 & -0.94818038668236D + 0 & 0.11907725086208D + 0 \\ 0.90652031636533D + 0 & 0.23782902994669D + 0 & -0.34879545371395D + 0 \end{array}$$

Table 9. Hand positions and orientations for Problems 17–20

 Problem 17 hand position:

$$0.31195801861115D + 1 \quad 0.89001270156765D + 0 \quad 0.23163290174744D + 0$$

Problem 17 hand orientation matrix:

$$\begin{array}{lll} 0.64815428818396D + 0 & 0.79161767330574D - 1 & 0.75738328031575D + 0 \\ -0.56119424077973D + 0 & -0.62263053294554D + 0 & 0.54533681661851D + 0 \\ 0.51473978166099D + 0 & -0.77850153117194D + 0 & -0.35913552196705D + 0 \end{array}$$

Problem 18 hand position:

$$0.23778331024581D + 1 \quad 0.59032742683492D + 0 \quad 0.23163290174744D + 0$$

Problem 18 hand orientation matrix:

$$\begin{array}{lll} 0.64815428818396D + 0 & 0.79161767330574D - 1 & 0.75738328031575D + 0 \\ -0.56119424077973D + 0 & -0.62263053294554D + 0 & 0.54533681661851D + 0 \\ 0.51473978166099D + 0 & -0.77850153117194D + 0 & -0.35913552196705D + 0 \end{array}$$

Problem 19 hand position:

$$-0.72607275531045D + 0 \quad 0.12646314554293D + 0 \quad 0.14167404085940D + 0$$

Problem 19 hand orientation matrix:

$$\begin{array}{lll} -0.68631115723422D + 0 & -0.18558900960193D - 2 & -0.72730567929020D + 0 \\ 0.65563580573354D + 0 & 0.43129177251048D + 0 & -0.61978149149914D + 0 \\ 0.31483120190978D + 0 & -0.90221059771914D + 0 & -0.29478356749881D + 0 \end{array}$$

Problem 20 hand position:

$$0.55331317605649D + 0 \quad 0.30113181787028D + 0 \quad 0.40793414236440D + 0$$

Problem 20 hand orientation matrix:

$$\begin{array}{lll} 0.30240098111430D + 0 & 0.21069244605705D + 0 & 0.92960332389445D + 0 \\ 0.29457522407360D + 0 & -0.94818038668236D + 0 & 0.11907725086208D + 0 \\ 0.90652031636533D + 0 & 0.23782902994669D + 0 & -0.34879545371395D + 0 \end{array}$$

5. SUMMARY AND CONCLUSIONS

In this paper we present a new model for the 6R inverse position problem (in Section 2) and define a solution approach based on polynomial continuation (in Section 4). The new model assumes that twist angles are not zero, but a modification is given in Appendix B which covers the zero twist angle cases. We report on testing the resulting 16 path continuation method on 41 test problems and include for comparison the outcome of testing a 64 path continuation method from Ref. [8] on the same problem set (Section 4). In Section 3 we outline the theoretical basis for our numerical technique: the method of the generic case and coefficient-parameter polynomial continuation.

We conclude:

1. The new approach is reliable on problems with twist angles of at least $1/10^\circ$. CPU times vary in our tests from 5.9 to 20.5 s. The average time is less than 10 s. This compares with cpu times varying from 16.1 to 30.5 with an average less than 20 s for the method from Ref. [8].
2. The methodology of the approach is applicable to any polynomial system and is potentially a very powerful tool for the analysis and solution of systems of polynomials that arise in kinematics.

Acknowledgement—This work represents a cooperative interdisciplinary effort. The order of the names given on this paper has no significance. There is no primary author.

REFERENCES

1. D. Pieper, Ph.D. Thesis, Stanford University (1968).
2. B. Roth, J. Rastegar and V. Scheinman, *1st CISM-IFTOMM Symp.*, Vol 1, pp. 93-113 (1973).
3. H. Albala and J. Angeles, *Proc. 5th World Congr. Theory Mach. Mech.*, pp 1008-1011 (1979).
4. J. Duffy and C. Crane, *Mech. Mach. Theory* 15, 153-169 (1980).
5. J. Duffy, *Analysis of Mechanisms and Robot Manipulators*. Wiley, New York (1980).
6. L.-W. Tsai and A. P. Morgan, *ASME JI Mech. Transm. Automn Des.* 107, 48-57 (1985).
7. A. P. Morgan and A. J. Sommese, *Appl. Math. Comput.* 24, 101-113 (1987).
8. A. P. Morgan and A. J. Sommese, *Appl. Math. Comput.* 24, 115-138 (1987).
9. E. J. F. Primrose, *Mech. Mach. Theory* 21, 509-510 (1986).
10. H.-Y. Lee and C.-G. Liang, *Mech. Mach. Theory* 23, 219-226 (1988).
11. R. Madhusudan and B. Roth, Preprint (1989).
12. R. Manseur and K. L. Doty, *Int. J. Robot. Res.* 8, 75-79 (1989).
13. R. S. Hartenberg and J. Denavit, *Kinematic Synthesis of Linkages*. McGraw-Hill, New York (1964).
14. A. P. Morgan and A. J. Sommese, *Appl. Math. Comput.* 29, 123-160 (1989).
15. E. L. Allgower and K. Georg, *SIAM Rev.* 22, 28-85 (1980).
16. L. T. Watson, S. C. Billups and A. P. Morgan, *ACM Trans. Math. Softw.* 13, 281-310 (1987).
17. A. P. Morgan, *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*. Prentice-Hall, Englewood Cliffs, N.J. (1987).
18. A. P. Morgan, A. J. Sommese and L. T. Watson, *Trans. Math. Softw.* (1989).
19. A. P. Morgan, *Impact of Recent Computer Advances on Operations Research (Proc. ORSA Conf.)*, Williamsburg, Va., (Ed. R. Sharda) (1989).
20. A. P. Morgan and A. J. Sommese, *Lectures in Applied Mathematics: Computational Solution of Nonlinear Systems of Equations (Proc. AMS-SIAM Summer Sem., Colorado State University)* (Eds E. Allgower and K. Georg) 1989).
21. C. W. Wampler, A. P. Morgan and A. J. Sommese, *ASME JI Mech. Des.* 112, 59-68 (1990).

APPENDIX A

The IPP1 Model

In this Appendix we outline the IPP 1-64 method for solving the 6R inverse position problem. This method was introduced in Refs [7, 8] as an example of the m -homogeneous approach to polynomial continuation. IPP1-64 is an improvement of the method proposed in Ref. [6].

In Ref. [6], the 6R problem is reduced to solving a system of eight second-degree equations in eight unknowns of the following form (see also Ref. [17, Chap. 10]):

$$\begin{aligned}
 F_l = & a_{l,1} w_1 w_3 + a_{l,2} w_1 w_4 + a_{l,3} w_2 w_5 + a_{l,4} w_2 w_6 \\
 & + a_{l,5} w_5 w_7 + a_{l,6} w_5 w_8 + a_{l,7} w_6 w_7 + a_{l,8} w_6 w_8 \\
 & + a_{l,9} w_1 + a_{l,10} w_2 + a_{l,11} w_3 + a_{l,12} w_4 \\
 & + a_{l,13} w_5 + a_{l,14} w_6 + a_{l,15} w_7 + a_{l,16} w_8 \\
 & + a_{l,17} \quad (l = 1, \dots, 4), \\
 F_l = & w_{2l-9}^2 + w_{2l-8}^2 - 1 \quad (l = 5, \dots, 8),
 \end{aligned} \tag{A.1}$$

where the coefficients a_{ij} are given by the manipulator parameters. The total degree of this system is 256, and therefore a traditional (one-homogeneous) polynomial continuation method must track 256 paths to solve the system. This was implemented in Ref. [6]. However, by grouping the variables into the two groups:

$$\{w_1, w_2, w_5, w_6\} \quad \{w_3, w_4, w_7, w_8\},$$

the resulting two-homogeneous structure yields a Bezout number of 96. We further observe that there are eight multiplicity-four fixed solutions to this two-homogeneous system (that do not depend on the choice of parameters). This is proven as Theorem 4 in Ref. [8]. Thus, as discussed in Ref. [8], we may set up a continuation with only 64 paths. In fact, we use a continuation system of the form

$$H(w, t) = (1-t)\gamma G(w) + tF(w),$$

in which the start system G is the second example in [6] and γ is a randomly chosen complex number.

The 64 paths are tracked using the method from Ref. [17, Chap. 4] (the CONSOL8 path tracker). The 64 endpoints are 64 solutions to (A.1), which for generic choices of parameters partition into four groups of 16 each, as follows:

- (1) 16 finite solutions to the 6R problem,
- (2) 16 finite extraneous solutions,
- (3) 16 solutions at infinity to the 6R problem

and

- (4) 16 extraneous solutions at infinity.

This is established by solving equation (A.1) for a random choice of parameters and evoking the method of the generic case. This generic system is solved using a modification of the code used to accomplish the analogous purpose for IPP2. (See Section 4.)

Thus, for this model, our use of side conditions in evoking the method of the generic case is a little more complicated than in Section 4. First, we have side conditions that *pick out* possibly physical solutions; specifically, these side conditions are the equations of the original unreduced IPP system. Solutions to equation (A.1) that do not satisfy these equations are called "extraneous". Then, we have side conditions that *rule out* solutions being possibly physical; specifically, the equations that fix a solution at infinity. The IPP2 analysis required only this second type of side condition. (See Refs [14, 21] for more on the way side conditions can be used.) The result is that the 64 path endpoints can be partitioned generically into four sets of 16, as above.

IPP1-64 solves the problems from Section 4 in CPU times ranging from 16.1 to 30.5 s. The CPU times reported in Ref. [6] (for three other problems) range from 212.363 to 266.513 s. We would generally expect IPP1-64 to outperform the Ref. [6] method, since it tracks a quarter of the paths and there is no reason to think it would be less reliable.

APPENDIX B

Parallel Joints

If one or more of the joint twists is 0 or 180°, then the corresponding $e_i = a_i/\sin \alpha_i$ is undefined and the reduction to equations (14–16) is illegal. Moreover, for very small $\sin \alpha_i$, the system will be ill-conditioned. (For quantitative information on this, see the experimental results in Section 4.) To avoid the difficulty, we retain x_{k+1} as a variable and eliminate z_{k+1} instead using the relation

$$z_{k+1} = -\sin \alpha_k z_k \times x_{k+1} + \cos \alpha_k z_k.$$

To avoid increasing the degree, we introduce a new variable

$$v^k = z_k \times x_{k+1},$$

so that

$$z_{k+1} = -\sin \alpha_k v_k + \cos \alpha_k z_k, \quad x_{k+1} = v_k \times z_k. \tag{B.1}$$

Now, we write the system as

$$\begin{aligned}
 z_i \cdot z_j &= 1 \quad i \neq k+1, \quad v_k \cdot v_k = 1, \\
 z_i \cdot z_{k+1} &= c, \quad i \neq k, \quad v_k \cdot z_k = 0, \\
 \sum_{\substack{1 \leq i \leq 5 \\ i \neq k}} (d_i z_i + e_i z_i \times z_{k+1}) + d_k z_k + a_k v_k \times z_k &= \hat{p}.
 \end{aligned}$$

where equation (B.1) is used everywhere z_{k+1} appears.

SOLUTION DU PROBLEME DE LA POSITION INVERSE POUR MANIPULATEURS A SIX JOINTS ROTATOIRES UTILISANT LA "METHODE DU CAS GÉNÉRIQUE"

Résumé—Dans cet article on considère le calcul de toutes les solutions du problème de la position inverse pour manipulateurs généraux à six joints rotatoires. Au lieu de réduire le problème à une équation très complexe d'entrée-sortie, on travaille avec un système de 11 équations polynômiales simples. En dépit du fait que le degré total du système soit très élevé (1024), en utilisant la "méthode du cas générique" on montre numériquement que le nombre générique des solutions est 16, en accord avec les preuves préalables. En outre, on présente une méthode numérique, basée sur la continuation des paramètres des polynômes, qui permet de trouver efficacement toutes les 16 solutions. On présente un ensemble de 41 problèmes dans lesquels l'algorithme numérique nécessite en moyenne moins de 10 sec CPU dans un IBM 370-3090 en FORTRAN double précision. Cette méthodologie est également applicable à d'autres problèmes de cinématique qui sont aussi formulés en systèmes d'équations polynômiales.