MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 854 **March, 1986**

A Closed Form Solution for Inverse Kinematics of Robot Manipulator with Redundancy

Pyung H. Chang

Abstract. A closed form equation for inverse kinematics of manipulator with redundancy is derived, using the Lagrangian multiplier method. The proposed equation is proved to provide the exact equilibrium state for the resolved motion method, and is shown to be a general expression that yields the extended Jacobian method. The repeatability problem in the resolved motion method does not exist in the proposed equation. The equation is demonstrated to give more accurate trajectories than the resolved motion method.

Acknowledgment. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124 and in part by the Systems Development Foundation.

1 Introduction

A kinematically redundant robot manipulator is a manipulator that has more degrees of freedom than are necessary to locate the end effector at a desired position and orientation. For example, if we want to locate some point on a two-dimensional end effector at a specified position and orientation, we need three degrees of freedom. Thus, a robot manipulator with more than three degrees of freedom is kinematically redundant in the two-dimensional space.

The major advantages to adding redundant degrees of freedom to a robot manipulator are as follows:

1. One achieves greater dexterity in maneuvering in a workspace with obstacles.

2. One can avoid singular configurations of the manipulators.

1.1 The Resolved Motion Method vs. the Inverse Kinematic Method

Because of these significant advantages, an increasing amount of research has focused on the kinematically redundant manipulator, and the progress in this field has been rapid [Whitney,1972; Liégeois,1977; Klein and Huang,1983; Yoshikawa,1984; Hollerbach,1984]. Much of this research has involved the resolution of motion using the pseudoinverse of the Jacobian matrix — also known as the Moore-Penrose generalized inverse matrix — in order to resolve the redundancy. This resolved motion technique first determines the joint velocity using the pseudoinverse matrix, and then incrementally determines the joint displacement; it thus transforms from workspace to joint space via joint velocity. In contrast to this direction of research, relatively little research [Benati et al,1982; Hollerbach,1984] has involved the inverse kinematics — the direct mapping from the workspace to the joint space — for kinematically redundant manipulators.

The advantages and disadvantages in the inverse kinematic method over the resolved motion method are well known in nonredundant robot manipulators. The resolved motion method — now using the inverse of the Jacobian matrix instead of the pseudoinverse matrix — is well-defined for general manipulator kinematics, except for numerical problems near kinematic singularities. Furthermore, the joint velocities can be efficiently computed from the workspace velocities using the Jacobian matrix without requiring an explicit matrix inverse. The resolved motion method, however, has some weak points:

- The method has intrinsic inaccuracy because of linear approximation characteristics of the Jacobian matrix; thus it accumulates errors, which become even larger as the velocity increases.
- The method does not give directly the joint values for a given position and orientation of the end effector.

On the other hand, the inverse kinematic method has symbolic solutions only in some types of manipulator kinematics [Pieper,1969]. For general manipulator kinematics, it has only iterative solutions based on numerical methods, which can be computationally expensive unless we have initial conditions sufficiently near to the solution. However, this method — be it symbolic or numerical — is attractive because of the direct mapping from the workspace to joint space, fixing most of the aforementioned problems of the resolved motion method.

1.2 Inverse Kinematic Method in the Redundant Manipulator

The comparison between the two methods remains essentially unchanged in redundant robot manipulators; thus we have good reasons to choose the inverse kinematic method.

As in the nonredundant case, no symbolic solution has been developed yet for the general redundant manipulator, for we cannot obtain symbolic solutions unless certain conditions are met by the manipulator structure. For example, in [Benati et al,1982; Hollerbach,1984] only some of the joint variables were obtained symbolically. To obtain these solutions the manipulator structure and the number of degrees of freedom were fixed — explicitly in [Hollerbach,1984] and implicitly in [Benati et al,1982].

An additional difficult task for a redundant manipulator — regardless of whether or not it has symbolic solutions — is to rationally (or optimally) use the extra degrees of freedom to achieve the objectives such as singularity avoidance or obstacle avoidance. In other words, the task is to resolve the redundancy while achieving some objectives.

To our knowledge, the general, closed-form equation to resolve the redundancy at the inverse kinematic level has not appeared yet. In this paper, we propose a method, or a general equation — derived from the Lagrangian multiplier method — to resolve the redundancy; thus fully specifying the kinematic equations. The resulting system of equations will be qualitatively compared with the two existing methods: the extended Jacobian method[Baillieul,1985] and the resolved motion method [Liégeois ,1977; Klein and Huang,1983], which uses the pseudoinverse matrix. From this comparison, the relationships between the proposed method and each of the two methods will be examined. To numerically evaluate the proposed method, the system of equations — which requires numerical iterative solutions — is solved for a simulated task, and its solutions are compared with those of the resolved motion method.

In Section 2, the proposed method will be derived. In Section 3, the comparison and the relationships will be covered. The proposed method will be evaluated by simulations in Section 4, and the results will be discussed in Section 5, and conclusions in Section 6.

2 Derivation of the Proposed Equation

In this section, we will derive extra equations which, together with the kinematic equations of the manipulator, can fully specify the under-determined problem.

The kinematic equation for the redundant manipulator is given as the following vector equation:

$$\mathbf{x} = \mathbf{f}(\vec{\theta}) \tag{1}$$

where $\mathbf{x}$ is an m-dimensional vector representing the position and orientation of the end effector in the workspace, $\vec{\theta}$ is n-dimensional vector representing joint variables, and $\mathbf{f}$ is a vector function consisting of m scalar functions, with $m < n$. Eq.(1) may be rewritten as

$$\begin{aligned} \mathbf{F}(\vec{\theta}) &= \mathbf{f}(\vec{\theta}) - \mathbf{x} \\ &= \mathbf{0} \end{aligned} \tag{1a}$$

Let $H(\vec{\theta})$ be some criteria function to be minimized, which quantitatively represents the desired performance — for instance, singularity avoidance or obstacle avoidance. Here, <u>any</u> criteria function can be used, as long as the function is expressed in terms of joint variables — which is easy in most cases by using Eq.(1) — and the function has first partial derivatives.

Let us define the Lagrangian function $L(\vec{\theta})$ as the following:

$$L(\vec{\theta}) = \vec{\lambda}^T \mathbf{F}(\vec{\theta}) + H(\vec{\theta}) \tag{2}$$

where $\vec{\lambda}$ is an m-dimensional Lagrangian multiplier vector. At the minimum of L,

$$\begin{aligned} \frac{\partial L}{\partial \vec{\theta}} &= \vec{\lambda}^T \frac{\partial \mathbf{F}}{\partial \vec{\theta}} + \frac{\partial H}{\partial \vec{\theta}} \\ &= \mathbf{0} \end{aligned} \tag{3}$$

where the $m \times n$ matrix, $\frac{\partial \mathbf{F}}{\partial \vec{\theta}}$, is the Jacobian matrix $\mathbf{J}$. The second term in the r.h.s. of Eq.(3) may be expressed as

$$\mathbf{h}^T = \frac{\partial H}{\partial \vec{\theta}}$$

where $\mathbf{h} = (h_1, h_2, ..., h_n)^T$ with

$$h_i = \frac{\partial H}{\partial \theta_i}, \quad (i = 1, 2, .., n)$$

Thus Eq.(3) becomes the following:

$$\vec{\lambda}^T \mathbf{J} = -\mathbf{h}^T \tag{4}$$

If we transpose Eq.(4), we get

$$\mathbf{J}^T \vec{\lambda} = -\mathbf{h}$$

or

$$\begin{bmatrix} (J^1)^T \\ (J^2)^T \\ \cdot \\ \cdot \\ (J^n)^T \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \cdot \\ \cdot \\ \lambda_m \end{bmatrix} = - \begin{bmatrix} h_1 \\ h_2 \\ \cdot \\ \cdot \\ h_n \end{bmatrix} \tag{5}$$

where $(J^i)^T$ denotes the transpose of i-th column vector of the Jacobian matrix. If we select m linearly independent rows from $\mathbf{J}^T$, which are, without loss of the generality, the first m rows, and constitute a nonsingular matrix, $\mathbf{J_m}$, then $\vec{\lambda}$ can be obtained from Eq.(5), as

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \cdot \\ \cdot \\ \lambda_m \end{bmatrix} = -\mathbf{J_m}^{-1} \begin{bmatrix} h_1 \\ h_2 \\ \cdot \\ \cdot \\ h_m \end{bmatrix}$$

Substituting this into Eq.(5), we have

$$- \begin{bmatrix} (J^{m+1})^T \\ (J^{m+2})^T \\ \cdot \\ \cdot \\ (J^n)^T \end{bmatrix} \mathbf{J_m}^{-1} \begin{bmatrix} h_1 \\ h_2 \\ \cdot \\ \cdot \\ h_m \end{bmatrix} = - \begin{bmatrix} h_{m+1} \\ h_{m+2} \\ \cdot \\ \cdot \\ h_n \end{bmatrix} \tag{6}$$

For simplicity, let us denote

$$\mathbf{J_{n-m}} = \begin{bmatrix} (J^{m+1})^T \\ (J^{m+2})^T \\ \cdot \\ \cdot \\ (J^n)^T \end{bmatrix}$$

where $\mathbf{J_{n-m}}$ is an $(n-m) \times m$ matrix. If we define

$$\mathbf{Z} = \left[\mathbf{J_{n-m}} \mathbf{J_m}^{-1} : -\mathbf{I_{n-m}} \right] \tag{6a}$$

where $\mathbf{I_{n-m}}$ is an identity matrix of rank $(n-m)$, then Eq.(6) becomes

$$\mathbf{Zh} = \mathbf{0} \tag{7}$$

where $\mathbf{Z}$ and $\mathbf{h}$ are defined as above. If we combine the kinematic equation, Eq.(1), with Eq.(7), as a system of equations, we get

$$\begin{cases} \mathbf{x} & = \mathbf{f}(\vec{\theta}) \\ \mathbf{Zh} & = \mathbf{0} \end{cases} \tag{7a}$$

Since $\mathbf{Z}$ is an $(n-m) \times n$ matrix, and $\mathbf{h}$ is an n-dimensional vector, Eq.(7) consists of $(n-m)$ scalar equations with n unknowns, $\vec{\theta}$. On the other hand, the kinematic equation, Eq.(1), has m scalar equations. Therefore, Eq.(7a) has n independent nonlinear equations which now fully specify the n unknowns. Note that Eq.(7a) has to be solved numerically.

The additional set of equations, Eq.(7), resolve the redundancy — at the inverse kinematic level — in such a way that the criteria function, $H(\vec{\theta})$, may be minimized. This resolution of redundancy may be viewed as the direct counterpart of the resolution technique in the resolved motion method, which uses the null space to resolve the redundancy. We will discuss the relationship in more detail in the subsequent section. Note the generality of Eq.(7); no assumption was made that could limit it.

3 Comparison with Other Methods

In this section, the present result in Eq.(7) will be compared with two existing methods: the extended Jacobian method and the resolved motion method which uses the pseudoinverse. The relationships will be investigated on the basis of comparisons.

3.1 Relationship with Extended Jacobian Method

An alternative method to resolve the redundancy — also at the inverse kinematic level — is presented in [Baillieul,1985]. This method derives the additional equations by using the orthogonality between the gradient vector of the criteria function and the null space vector, $\mathbf{n}_J$, such as

$$\begin{aligned} G(\vec{\theta}) &= \mathbf{n}_J \mathbf{h} \\ &= 0 \end{aligned} \tag{8}$$

where

$$\mathbf{n}_J = (\triangle_1, \triangle_2, .., \triangle_n)^T \tag{9}$$

with

$$\triangle_i = (-1)^{i+1} det(J^1, J^2, .., J^{i-1}, J^{i+1}, .., J^n) \tag{10}$$

and where J^k is the k-th column vector of the Jacobian matrix. From the resulting fully specified system of equations, the new Jacobian matrix, called the extended Jacobian, is derived, just in the same way as in the nonredundant case.

This method, however, is limited to redundant manipulators that have only one redundant degree of freedom, that is, only if $n = m+1$. Obviously, this limitation is not desirable.

As proved in Appendix A, the additional equation, Eq.(8) can be also obtained from Eq.(7) in the case that $n = m+1$. Therefore, we may regard Eq.(7) as a general equation from which Eq.(8) could be derived.

Moreover, as Eqs.(9) and (10) show, elements of $\mathbf{n}_J$ are the determinants of n $m \times m$ submatrices made from the original Jacobian matrix — making n different combinations from n column vectors, and obtaining n determinants. Compared to this, however, Eq.(7) has only one inversion of an $m \times m$ matrix; accordingly, it is somewhat easier to manipulate either symbolically or numerically.

Therefore, we may say that Eq.(7) is the more general equation with expressions somewhat simpler to treat and more efficient to compute than Eq.(8).

3.2 Relationship with Pseudoinverse Control

A general solution for the equation

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\vec{\theta}},$$

when $n > m$, was given as [Ben-Israel and Greville,1980]

$$\dot{\vec{\theta}} = \mathbf{J}^{+}\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\mathbf{h} \tag{11}$$

where $\mathbf{J}^{+}$ is the pseudoinverse matrix, defined as

$$\mathbf{J}^{+} = \mathbf{J}^{T}(\mathbf{J}\mathbf{J}^{T})^{-1} \tag{12}$$

and $(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})$ is the null space of $\mathbf{J}$, with $\mathbf{h}$ an arbitrary vector. Eq.(11) gives a way to resolve the redundancy at the velocity level.

Liégeois[1977] developed a formulation of resolution of redundancy, such that a scalar criteria function may be minimized, by setting the vector $\mathbf{h}$ as

$$\mathbf{h} = \nabla H \tag{12a}$$

where H is the criteria function to be minimized. He expresses Eq.(11) and Eq.(12a) in terms of the infinitesimal displacement, as

$$d\vec{\theta} = \mathbf{J}^{+}d\mathbf{x} + (\mathbf{I} - \mathbf{J}^{+}\mathbf{J}) \nabla H \tag{13}$$

In Eq.(13)(or Eq.(11)), the first term in the r.h.s. is responsible for the displacement of the end effector. The second term, on the other hand, forces joints to have self-motion to achieve an equilibrium (or optimum), $\vec{\theta}^{*}$, where H has a local minimum, for an instantaneous location of the end effector, $\mathbf{x}(t)$. In practice, however, while the joint variables are searching for a $\vec{\theta}^{*}$ for a given $\mathbf{x}$, $\mathbf{x}(t)$ continuously varies — thus requiring different $\vec{\theta}^{*}$'s. Therefore, the joint variables never reach the optimal configuration, but are slightly different from that configuration in the direction determined by the first term in Eq.(13), $\mathbf{J}^{+}d\mathbf{x}$.

Because of these characteristics, the resolved motion method in Eq.(13)(or Eq.(11)) has an undesirable property: it does not preserve repeatablity of joint values for repeated end effector motions. More specifically, two factors that cause this property are as follows:

1. Because of the directionality in the first term of Eq.(13), the joint variables have different values depending on the direction of the repeated path — a cyclic path, for instance — in the workspace. Note,

however, that the repeatablility can be preserved when tracing only one direction of the cyclic path even in the presence of this factor [Baillieul,1985].

2. Because of the irreversibility of the second term, they never return to the original configuration, once the joint variables achieve a $\vec{\theta}^*$. This situation can happen at the initial transient period when initial joint values are far from the optimum; for the optimal joint values cannot be known in advance.

In Appendix B, we prove that Eq.(7) is the necessary and sufficient condition to be satisfied when Eq.(13) reaches its equilibrium states with $d\mathbf{x} = \mathbf{0}$. In other words, Eq.(7a) gives the exact equilibrium state at which Eq.(13) will eventually arrive — the optimal joint configuration — for a given end effector location. Thus, we may regard Eq.(13) in the resolved motion method as an approximated equation linearized at states that are exactly determined by Eq.(7a).

4 Simulation

In this section, we select a kinematically redundant manipulator and apply Eq.(7). The resulting system of equations are solved numerically for $\mathbf{x}$, the end effector location, which makes a cyclic path. In parallel to using Eq.(7a), the resolved motion method is applied to the same manipulator with the same tip motion. The points we try to examine or verify through the simulation are as follows:

1. Whether the resulting system in Eq.(7a) gives kinematically correct joint variables for a given $\mathbf{x}$, achieving the performance represented by the criteria function we select.

2. How the present method compares to the resolved motion method in terms of accuracy and repeatability.

3. Whether the present method gives, in fact, the same equilibrium states as the resolved motion method will eventually reach.

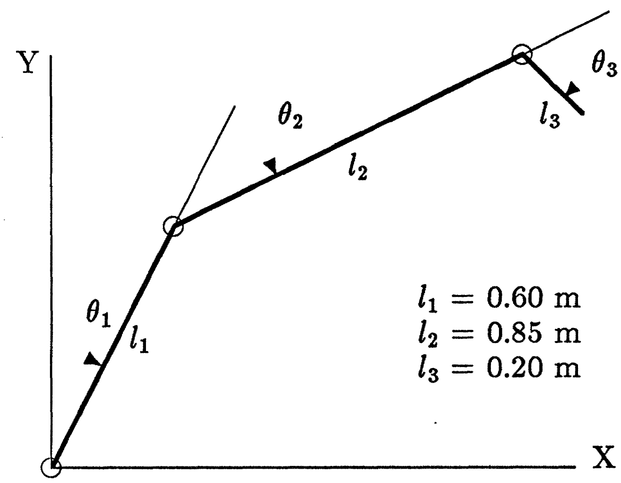

Figure 1: The Schematic Diagram Of The Redundant Manipulator

We use the same manipulator presented in the paper[Yoshikawa,1984] for the sake of comparison of data obtained in that paper: a 3 degrees of freedom manipulator with the end effector moving in the (x,y) plane; thus kinematically redundant. The schematic diagram with necessary parameters is in Figure 1.

The task is to make a square path — thus a cyclic path — while avoiding singularities. A good criteria function for this objective may be the manipulability[Yoshikawa,1984], which is given as

$$H = det(\mathbf{J}\mathbf{J}^{\mathrm{T}}) \tag{14}$$

The kinematic equation is given as

$$x = l_1 s_1 + l_2 s_{12} + l_3 s_{123} \tag{15}$$

$$y = l_1 c_1 + l_2 c_{12} + l_3 c_{123} \tag{16}$$

where l_1,l_2, and l_3 represent the length of each link, while the variables with subscripts are defined as

$$s_i = sin(\theta_i), \quad s_{ij...k} = sin(\theta_i + \theta_j + ... + \theta_k),$$
$$c_i = cos(\theta_i), \quad c_{ij...k} = cos(\theta_i + \theta_j + ... + \theta_k), \quad i, j, .., k = 1, 2, 3$$

Then, the Jacobian matrix is obtained as

$$\mathbf{J} = \begin{pmatrix} vc_{123} + uc_{12} + c_1 & vc_{123} + uc_{12} & vc_{123} \\ -vs_{123} - us_{12} + s_1 & -vs_{123} - us_{12} & -vs_{123} \end{pmatrix} \tag{17}$$

where

$$u = \frac{l_2}{l_1}, \quad v = \frac{l_3}{l_1}$$

By applying Eq.(7), we get

$$\begin{aligned} &2u^2v^3(s_{23}s_{33} - c_{23}s_3^2) + uv^3(2s_{23}s_{233} - 3s_{2233}s_3) + 2u^3v^2s_2s_{33} + \\ &u^2v^2(c_{33} - c_{22}) + 2uv^2(s_2s_{2233} - c_2s_{23}^2) - u^3vs_{22}s_3 - 2u^2vs_2s_3 \\ &= 0 \end{aligned} \tag{18}$$

where the same definition is used for the subscripts as in Eqs.(15) and (16).

The system of equations, Eq.(15), Eq.(16), and Eq.(18) now fully specify the originally under-determined system of equations, Eq.(15) and Eq.(16), while minimizing H in Eq.(14). The system of equations may be solved either purely numerically, or by symbolically reducing variables — in this example, θ_1 — and then by using numerical methods. Incidentally, this example suggests that it is possible to reduce variables, thus reducing the order of the system of nonlinear equations, after resolving the redundancy first with all the joint variables. The system of equations were numerically solved for joint values, tracing each (x, y) on a square command path in the workspace by using a subroutine called ZSOLVE, translated from the **IMSL** library into MACSYMA. As shown in Figure 3, the x-y coordinates of the four vertices of the command path are given as follows:

(446.00, 91.514),(546.00, 91.514)

(446.00,-84.865),(546.00,-84.865),

where the units are mm.

Meanwhile, the resolved motion method in Eq.(11) is also applied to this manipulator for the same path. Since Eq.(11) is a system of differential equations, a FORTRAN program, called DYSYS — which uses the Runge-Kutta integration method — was used together with the LINPACK subroutines. Note that this numerical scheme allows much more accurate solutions than the normal scheme for the resolved motion method. That is, the Runge-Kutta method evaluates the derivative four times at each time interval and calculates the weighted average, while the normal scheme evaluates it only once at each time interval.

The simulation for the resolved motion method was made with initial joint angle values of (-40.5006, 141.6408, 78.4169) in degrees, which are far from equilibrium states, that were deliberately selected to examine repeatability. The same initial value was also used as the initial guess for the above nonlinear equations for fair comparison of the two methods.

5 Results and Discussions

5.1 Results

The numerical results of simulations are listed in Figures 2-3 and Tables 1-3:

- Figure 2 is the plot of joint variables solved with the two methods. Note that the 3-D trajectory of joint variables is represented with two 2-D plots: θ_1 vs. θ_2 and θ_1 vs. θ_3.

  On the other hand, Figure 3 shows actual trajectories, with the two methods, of the end effector in the workspace, as compared to the command path. The actual trajectory was determined by forward kinematics with joint values obtained by each method.

- In Table 1, some representative solutions are listed to numerically compare the accuracy of the end effector position obtained with each method, while, in Table 2, the repeatability was examined for a simple reciprocal path.

  Table 3, on the other hand, verifies the relationship proved in Appendix B : the solution with the resolved motion method when $d\mathbf{x} = \mathbf{0}$ is the same as that with the proposed method.

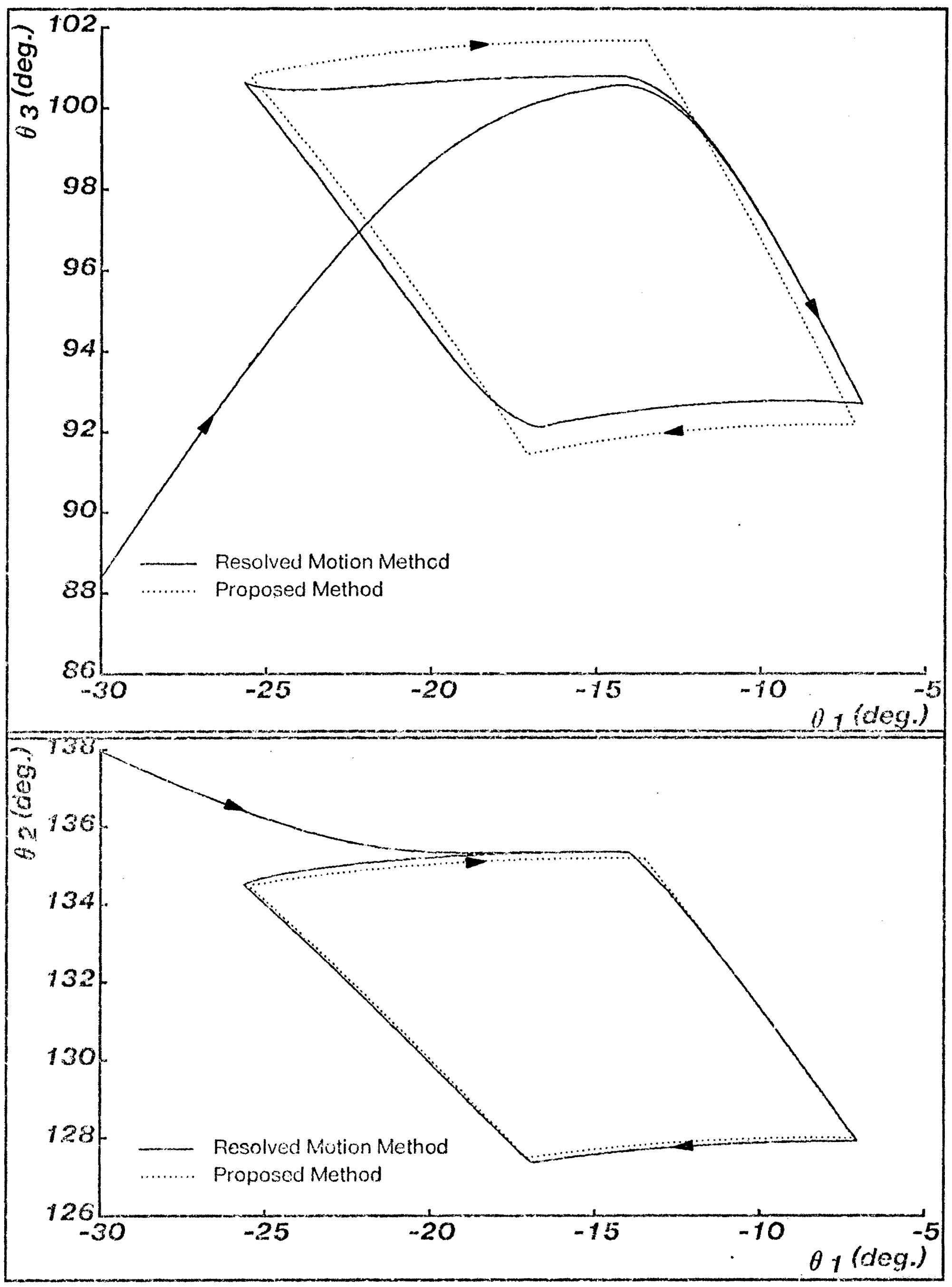

Figure 1: Joint Trajectories Obtained With The Two Methods

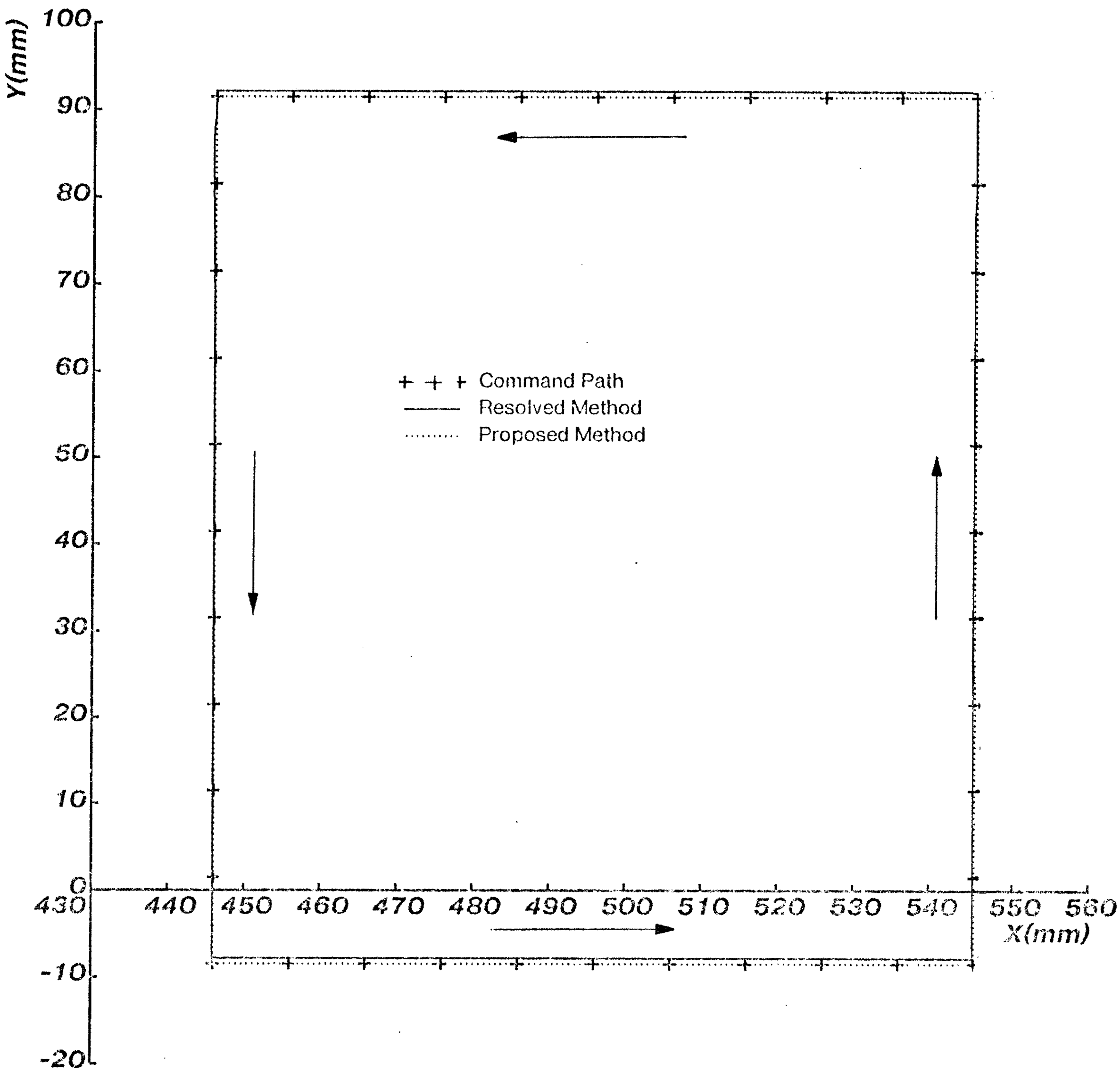

Figure 2: The Command Path And Actual Workspace Trajectories Obtained With The Two Methods

Table 1: The Comparison Of Accuracy Obtained With The Two Methods

	X(mm)	Y(mm)	θ_1(deg)	θ_2(deg)	θ_3(deg)
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
Resolved Method	: 446.00	91.514	-40.5006	141.6408	78.4169
Command Path	: 446.00	-84.866			
Proposed Method	: 446.00	-84.866	-13.4927	135.1801	101.6627
Resolved Method	: 445.74	-65.824	-14.0445	135.3160	101.2448
Command Path	: 546.00	-84.866			
Proposed Method	: 546.00	-84.863	-7.1232	128.0020	92.1837
Resolved Method	: 545.52	-68.216	-7.1924	127.9635	92.4919
Command Path	: 546.00	91.514			
Proposed Method	: 546.00	91.514	-17.0753	127.4846	91.4484
Resolved Method	: 545.73	92.947	-17.0519	127.3890	91.7938
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
Resolved Method	: 445.97	93.167	-25.7427	134.4867	100.7127

Table 2: The Repeatability Test With The Two Methods

	X(mm)	Y(mm)	θ_1(deg)	θ_2(deg)	θ_3(deg)
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
Resolved Method	: 445.89	93.140	-25.7472	134.4929	100.7206
Command Path	: 446.00	-84.866			
Proposed Method	: 446.00	-84.866	-13.4927	135.1801	101.6627
Resolved Method	: 445.63	-66.244	-14.0476	135.3249	101.2557
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
Resolved Method	: 445.60	92.912	-25.4178	134.3821	101.2799
Command Path	: 446.00	-84.866			
Proposed Method	: 446.00	-84.866	-13.4927	135.1801	101.6627
Resolved Method	: 445.57	-66.184	-14.0526	135.3293	101.2610

Table 3: The Comparison Of Solutions: Proposed Method Vs. Resolved Method With dx=0

	X(mm)	Y(mm)	θ_1(deg)	θ_2(deg)	θ_3(deg)
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
Resolved Method	: 446.00	91.514	-25.5115	134.4894	100.8164
Command Path	: 446.00	-84.866			
Proposed Method	: 446.00	-84.866	-13.4927	135.1801	101.6627
Resolved Method	: 446.00	-84.868	-13.4927	135.1801	101.6626
Command Path	: 546.00	-84.866			
Proposed Method	: 546.00	-84.863	-7.1232	128.0020	92.1837
Resolved Method	: 546.00	-84.863	-7.1232	128.0020	92.1837
Command Path	: 546.00	91.514			
Proposed Method	: 546.00	91.514	-17.0753	127.4846	91.4484
Resolved Method	: 546.00	91.514	-17.0752	127.4846	91.4484

5.2 Discussions

From the results of simulations, we may evaluate the proposed method in terms of accuracy and repeatability by comparing it with the resolved motion method. We can also derive some useful ideas from the relationship between the two methods.

5.2.1 Accuracy

As shown in Table 1 and Figure 3, the proposed method gives joint variables which exactly correspond to the commanded x and y, minimizing the criteria function to avoid singularities. Clearly, we see that the accuracy in the workspace achieved with the proposed method is better than that with the resolved motion method.

Therefore, the proposed method provides a useful mean for an accurate position control of the end-effector, when the manipulator is kinematically redundant.

5.2.2 Repeatability

Table 2 and Figure 2 show that the repeatability is not preserved with the resolved motion method because of the two factors mentioned in Section 3: the initial joint variables which are far from optimal joint values (Figure 2) and the direction — clockwise or counterclockwise — of the path to be traced (Table 2).

The lack of repeatablility can be a considerable drawback in robot manipulators which perform cyclic tasks, because, as the end effector traces the cyclic path, joint variables evolve into states which cannot be predicted in advance.

On the other hand, it is obvious that the proposed method preserves the repeatability regardless of direction. In other words, the method allows a fixed transformation from workspace to joint space. The property of fixed transformation is useful not only for the prediction of joint variables, but also for the precomputation of position dependent terms such as the Jacobian matrix and the inertia matrix.[Raibert and Horn,1978]

5.2.3 Relationship between the Two Methods

The result in Table 3 shows a nearly perfect agreement of both solutions, verifying that Eq.(7) in the proposed method is the equilibrium equation at which Eq.(11) will finally arrive.

Because of the relationship between the two methods, we may use them interchangeably as follows:

- The exact equilibrium state can be determined either directly with the proposed method, or indirectly with the resolved motion method by setting $d\mathbf{x} = \mathbf{0}$. The latter, however, would require more computations than the former.

- The incremental displacement $d\vec{\theta}$, which Eq.(11) in the resolved motion method easily provides, can be also obtained by first differentiating Eq.(7a) to get the Jacobian matrix and then by inverting it.

We can also make use of the relationship complementarily: The proposed method could be more effective if the initial guess for the method is provided by the resolved motion method.

6 Conclusions

In the paper, we have derived a general equation which resolves the redundancy in the kinematically redundant manipulator. This equation, together with the kinematic equations, constitutes a system of equations, the solution of which generates accurate joint values which make the end effector trace the commanded path in the workspace, while minimizing the criteria function.

We have compared the equation with two other methods. We have proved that the equation derived is a general equation that generates the extended Jacobian method. We have also shown that the proposed system of equations is an equilibrium equation toward which the resolved motion method converges. This point was proved and verified in the simulation results. The results of simulation also show that the proposed method gives better joint values than the resolved motion method in terms of accuracy and repeatability.

Acknowledgment.

In this research, the idea of the fixed transformation from the workspace to the joint space was first suggested by Berthold Horn, which the author gratefully acknowledges. He is also thankful to Thomás Lozano-Pérez and Ken Salisbury for their detailed comments on drafts.

References

1. Baillieul,J., "Kinematic Programming Alternatives for Redundant Manipulators," *IEEE Conference for Robotics and Automation*, St. Louis, March 25-28, 1985

2. Benati,M., Morasso,P., and Tagliasco,V., "The Inverse Kinematic Problem for Anthropomorphic Manipulator Arms," *ASME J. of Dynamic Systems, Measurements, and Control*, vol 104, 1982, pp 110-113

3. Ben-Israel,A. and Greville,T., *Generalized Inverses: Theory and Applications*, New York, Robert E. Krieger Publishing Co., 1980.

4. Hollerbach,J., "Optimum Kinematic Design for a Seven Degree of Freedom Manipulator," *2nd Int'l Symp. on Robotics Research*, Kyoto, Japan, Aug.20-23, 1984

5. Klein,C., and Huang,C., "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," *IEEE Trans on System, Man and Cybernetics*,vol. SMC-13, pp245-250, 1983

6. Liégeois,A., "Automatic Supervisory Control of the Configuration and Behaviour of Multibody Mechanisms," *IEEE Trans on System, Man and Cybernetics*,vol. SMC-7, no.12, 1977

7. Pieper,D., "The Kinematics of Manipulators under Computer Control," Ph.D. Thesis, Dep't of Computer Science, Stanford University, 1968

8. Raibert,M., and Horn,B., "Manipulator Control Using the Configuration Space Method," *Industrial Robot* 5, 2, June, 1978

9. Whitney,D., "The Mathematical Coordinated Control of Manipulators and Human Prostheses," *ASME J. Dynamic System, Measure., Control*, pp303-309, 1972

10. Yoshikawa,T.,"Analysis and Control of Robot Manipulators with Redundancy," *The 1st Int'l Symp.*,ed. Brady,M. and R.Paul, Cambridge, Mass., MIT Press, pp.735-748, 1984

7 Appendix A. The Derivation of the Extended Jacobian Method

In this Appendix, we will prove that the extended Jacobian method is, in fact, a special case of Eq.(7).

The additional equation to resolve the redundancy in the extended Jacobian method is given[Baillieul,1985] in Eqs.(8),(9), and (10) as

$$\begin{aligned} G(\vec{\theta}) &= \mathbf{n}_J \mathbf{h} \\ &= 0 \end{aligned} \tag{8}$$

where

$$\mathbf{n}_J = (\triangle_1, \triangle_2, ..., \triangle_n) \tag{9}$$

with

$$\triangle_i = (-1)^{i+1} det(J^1, J^2, ..J^{i-1}, J^{i+1}, .., J^n) \tag{10}$$

and where J^k is the k-th column vector of the Jacobian matrix, $\mathbf{J}$, derived from Eq.(1). When $n = m + 1$, our result, Eq.(7), is specified as

$$\begin{aligned} G_2(\vec{\theta}) &= [(J^n)^T \mathbf{J_m}^{-1}, -1]\mathbf{h} \\ &= 0 \end{aligned} \tag{A1}$$

where $(J^n)^T$ is the transpose of the n-th column vector of the Jacobian matrix, $\mathbf{J}$. $\mathbf{J_m}^{-1}$ can be derived as

$$\mathbf{J_m}^{-1} = \frac{1}{D_m} \mathbf{A_m} \tag{A2}$$

where $\mathbf{A_m}$ is the adjoint matrix of $\mathbf{J_m}$ and D_m is the determinant of $\mathbf{J_m}$. Thus $\mathbf{A_m}$ is expressed as

$$\mathbf{A_m} = \begin{pmatrix} Cof_{11} & Cof_{21} & \dots & Cof_{m1} \\ Cof_{12} & Cof_{22} & \dots & Cof_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ Cof_{1m} & Cof_{2m} & \dots & Cof_{mm} \end{pmatrix} \tag{A3}$$

where Cof_{ij} is the cofactor of j_{ij} of the Jacobian matrix, $\mathbf{J}$. From Eqs.(A1),(A2), and (A3), we get

$$(J^n)^T \mathbf{J_m}^{-1} = \frac{1}{D_m}[(J^n)^T A_m^1, (J^n)^T A_m^2, ...(J^n)^T A_m^m] \tag{A4}$$

where A_m^i is the i-th column vector of $\mathbf{A_m}$. Since

$$\begin{aligned}(J^n)^T A_m^1 &= j_{1n} Cof_{11} + j_{2n} Cof_{12} + .. + j_{mn} Cof_{1m} \\ &= -\triangle_1\end{aligned}$$

we get likewise

$$(J^n)^T A_m^i = -\triangle_i \tag{A5}$$

From the definition of $\triangle_n$, we get

$$\begin{aligned}\triangle_n &= det(\mathbf{J_m}^T) \\ &= det(\mathbf{J_m}) \\ &= D_m\end{aligned} \tag{A6}$$

Therefore,

$$((J^n)^T \mathbf{J_m}^{-1}, -1) = (-\frac{\triangle_1}{\triangle_n}, -\frac{\triangle_2}{\triangle_n}, ..., -1) \tag{A7}$$

Since $\mathbf{J_m}$ is nonsingular — and thus $\triangle_n$ is nonzero, we can multiply by it on both sides of Eq.(A1), resulting in Eq.(8). Thus we have proved that Eq.(7) is a general expression which yields the additional equation in the extended Jacobian method.

8 Appendix B. The Relationship Between The Proposed Method And The Resolved Motion Method

The matrix in Eq.(6a) is again,

$$\mathbf{Z} = \left[\mathbf{J_{n-m}} \mathbf{J_m}^{-1} : -\mathbf{I_{n-m}}\right] \tag{6a}$$

Since the rank of any matrix is the dimension of its largest nonsingular submatrix, which is $\mathbf{I_{n-m}}$ for $\mathbf{Z}$, the rank of $\mathbf{Z}$ (and $\mathbf{Z}^T$, too) is $n-m$. In addition, since

$$\begin{aligned}\mathbf{J}\mathbf{Z}^T &= [\mathbf{J_m}^T : \mathbf{J_{n-m}}^T] \begin{bmatrix} (\mathbf{J_m}^{-1})^T (\mathbf{J_{n-m}}^{-1})^T \\ -\mathbf{I_{n-m}} \end{bmatrix} \\ &= \mathbf{0},\end{aligned} \tag{B1}$$

column vectors of $\mathbf{Z}^T$ are a set of basis vectors which are orthogonal to $\mathbf{J}$. Thus, row vectors of $\mathbf{J}$, together with column vectors of $\mathbf{Z}^T$ constitute the basis of n-dimensional vector space.

Accordingly, any n-dimensional vector $\mathbf{h}$ can be represented as

$$\mathbf{h} = \mathbf{J}^T\mathbf{h}_1 + \mathbf{Z}^T\mathbf{h}_2 \tag{B2}$$

where $\mathbf{h}_1$ and $\mathbf{h}_2$ are arbitrary vectors of m and $n-m$ dimensions, respectively. Premultiplying Eq.(B2) by $\mathbf{J}$, we have

$$\mathbf{J}\mathbf{h} = \mathbf{J}\mathbf{J}^T\mathbf{h}_1$$

Thus,

$$\mathbf{h}_1 = (\mathbf{J}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{h} \tag{B3}$$

Similarly, if we multiply Eq.(B2) with $\mathbf{Z}$ and solve for $\mathbf{h}_2$, we get

$$\mathbf{h}_2 = (\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{h} \tag{B4}$$

From Eq.(B2), Eq.(B3), and Eq.(B4), we obtain the following relationship:

$$(\mathbf{I} - \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}\mathbf{J})\mathbf{h} = \mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{h}$$

or, from Eq.(12),

$$(\mathbf{I} - \mathbf{J}^+\mathbf{J})\mathbf{h} = \mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{h} \tag{B5}$$

For a constant location of the end effector (no tip motion), when $\mathbf{dx} = \mathbf{0}$, Eq.(13), with Eq.(B5), becomes

$$d\vec{\theta} = \mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{h} \tag{B6}$$

If $\mathbf{Z}\mathbf{h} = \mathbf{0}$ as in Eq.(7), then from Eq.(B6), $d\vec{\theta} = \mathbf{0}$, which means that joint variables reach an equilibrium state — or a stationary state — which is mostly the optimal configuraion for a given tip location. Conversely, if $d\vec{\theta} = \mathbf{0}$, we have $\mathbf{Z}\mathbf{h} = \mathbf{0}$; since the rank of $\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}$ in Eq.(B6) is $n-m$ and $\mathbf{Z}\mathbf{h}$ is an $(n-m)$-dimensional vector.

Therefore, $\mathbf{Z}\mathbf{h} = \mathbf{0}$, Eq.(7), of the proposed method is the necessary and sufficient condition to be satisfied when Eq.(13) of the resolved motion method reaches its equilibrium state.