C. R. Mirman
Engineering Department
Wilkes University
Wilkes-Barre, Pennsylvania 18766

K. C. Gupta
Department of Mechanical Engineering
University of Illinois at Chicago
Chicago, Illinois 60680

Identification of Position-Independent Robot Parameter Errors Using Special Jacobian Matrices

Abstract

Common production manipulators are designed to be controlled through the use of a closed-form kinematic solution algorithm. This closed-form solution uses manufacturer-specified nominal parameter values to model the given manipulator. These position-independent parameters, link lengths, twists, and offsets are geometrical and do not change as the manipulator is driven. However, manufacturing tolerances or usage may cause the actual parameters to deviate from the nominal robot parameters. In some cases, this deviation may be large and thus may cause the closed-form inversion to yield inaccurate joint value results.

To assure proper end-effector positioning, it is important to identify these position-independent error values. The identification algorithm to be presented uses the Denavit-Hartenberg (D-H) parameter notation to model any given M-jointed manipulator. For the given model, a set of special Jacobian matrices are formulated with respect to the four D-H parameters ($\mathbf{J}_\theta$, $\mathbf{J}_s$, $\mathbf{J}_\alpha$, and $\mathbf{J}_a$). This set of four special Jacobian matrices is used to determine a set of D-H error parameter values, which will allow one to estimate the actual D-H parameter values.

The iterative regression algorithm for parameter identification uses the end-effector information at several arbitrary manipulator positions, as well as nominal parameter and joint variable information to compute the parameter error values. The use of the special Jacobian matrices allows for easy determination of mathematical singularity conditions. In addition, statistical methods are employed to provide numerical uniformity between examples and aid in the evaluation process. Through several examples, it will be shown that the use of this parameter identification algorithm produces accurate results. This article will discuss the possibilities of deterministic as well as probabilistic parameter errors. Through the use of this parameter identification algorithm, one is able to identify the D-H parameter error values that will otherwise cause a loss in end-effector accuracy.

The International Journal of Robotics Research,
Vol. 12, No. 3, June 1993,

Introduction

Manufacturing tolerances, setup, and prolonged usage may alter the actual link parameters (lengths, twists, and offsets) from the nominal or specified link parameters, as well as affect the joint encoder accuracy. In order to obtain end-effector accuracy by compensating for built-in errors, one must formulate a method for identifying these parameter error values. A method will be presented that will allow identification of the amount of error in the position-independent parameters.

Numerous approaches have been used to determine error values in industrial manipulators. Ibarra and Perreira (1986) develop a model in which all link and joint parameter errors are taken as position independent. Hsu and Everett (1985), Tang and Ching-Cheng (1987), and Veitschegger and Wu (1987) include a small rotational term to overcome linear dependence; therefore, five variables per joint must now be found. Whitney et al. (1986) include nongeometric error sources (i.e., gear transmission error) in the model. Huang (1989) sets up the manipulator error model but does not discuss the solution algorithm, and Jarvis (1987) formulates a method by which errors are determined through a triangulation process using an optical measuring system. In this article the error parameters are divided into two groups: position-independent (geometrical parameters) and position-dependent (variable joint parameters). Several statistical approaches will be employed to ensure consistency between examples, as well as to aid in evaluation. Several examples of current production robots will be used to show the results of the parameter identification algorithm under varying built-in error levels. The use of the proposed parameter identification method is a very simple and straightforward process. The proposed identification algorithm will allow determination of position-independent parameter error values, which may then be employed to increase end-effector accuracy.

Jacobian Matrix Formulation

It is possible to form a model for any given M-jointed manipulator, which will allow for the formation of a set of four Jacobian matrices (Gupta 1985; Mirman and Gupta 1991). One may write a relation between the end-effector reference system and the base reference system, $\mathbf{A}_h$, in terms of adjacent link transformation D-H matrices, $\mathbf{A}_i$ (Denavit and Hartenberg 1955; Pieper and Roth 1969).

$$\mathbf{A}_h = \mathbf{A}_1 * \mathbf{A}_2 * \mathbf{A}_3 * \cdots * \mathbf{A}_M = \left[\begin{array}{c|c} \mathbf{R}_h & \mathbf{p}_h \\ \hline \mathbf{0} & 1 \end{array}\right], \quad (1)$$

where the rotation matrix $\mathbf{R}_h$ is given by,

$$\mathbf{R}_h = \mathbf{R}_1 * \mathbf{R}_2 * \mathbf{R}_3 * \cdots * \mathbf{R}_M. \quad (2)$$

The 4×4 link transformation matrix may be partitioned into 3×3 rotation matrix, $\mathbf{R}_m$, and a 3×1 translation vector, $\mathbf{p}_m$, and may be written in the form

$$\mathbf{A}_m^L = \mathbf{A}_1 * \mathbf{A}_2 * \mathbf{A}_3 * \cdots * \mathbf{A}_{m-1} = \left[\begin{array}{c|c} \mathbf{R}_m^L & \mathbf{p}_m \\ \hline \mathbf{0} & 1 \end{array}\right], \quad (3)$$

and the rotation matrix $\mathbf{R}_m^L$ is given by,

$$\mathbf{R}_m^L = \mathbf{R}_1 * \mathbf{R}_2 * \mathbf{R}_3 * \cdots * \mathbf{R}_{m-1}, \quad (4)$$

where $\mathbf{0}$ is a 3×1 vector. It should be noted that the terms $\mathbf{R}_m^L$ and $\mathbf{A}_m^L$ represent rotation and transformation relationships, respectively, between the mth link system and the base reference system. For each link, a special rectangular coordinate system, $(x_m y_m z_m)$, must be established. The D-H notation is comprised of three-dimensional parameters and one joint variable, each describing an angle or length, which are needed to transform coordinates from the $(m+1)$th link system to the mth link system (Fig. 1). Most industrial manipulators are comprised of two joint types: a revolute joint in which θ is denoted as the joint variable, while s is fixed; and a prismatic joint in which s denotes the joint variable, while θ is fixed. The 4×4 D-H matrix for transformation from the $(m+1)$th link system to the mth link system is given by (Denavit and Hartenberg 1955; Pieper and Roth 1969),

$$\mathbf{A}_m = \left[\begin{array}{ccc|c} \cos\theta_m & -\sin\theta_m \cos\alpha_m & \sin\theta_m \sin\alpha_m & a_m \cos\theta_m \\ \sin\theta_m & \cos\theta_m \cos\alpha_m & -\cos\theta_m \sin\alpha_m & a_m \sin\theta_m \\ 0 & \sin\alpha_m & \cos\alpha_m & s_m \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \quad (5)$$

The transformation matrix $\mathbf{A}_m$ is shown in its partitioned form, as in equation (3). The transformation of a point on the end effector into the base coordinate system is given by

$$\begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix} = \mathbf{A}_1 * \mathbf{A}_2 * \cdots * \mathbf{A}_M \begin{bmatrix} \mathbf{P}_o \\ 1 \end{bmatrix} = \mathbf{A}_h \begin{bmatrix} \mathbf{P}_o \\ 1 \end{bmatrix}, \quad (6)$$

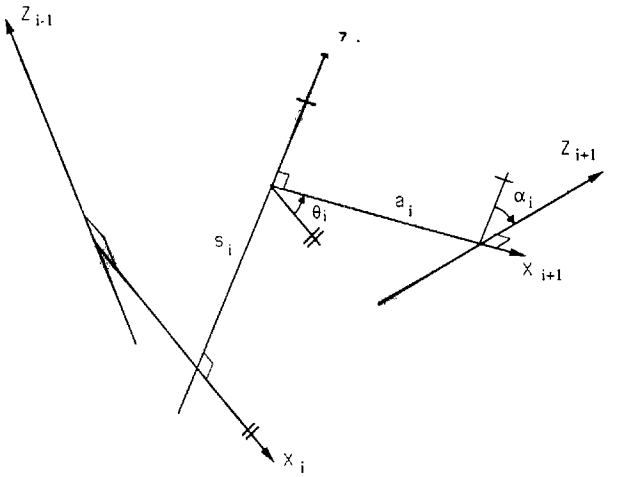

Fig. 1. Definition of Denavit-Hartenberg (D-H) parameters.

where $\mathbf{P}_o$ is the location of a point on the end effector in the end-effector system, and $\mathbf{P}$ is its location in the base system. Differentiating both sides of equation (6) and eliminating $\mathbf{P}_o$, one finds

$$\begin{bmatrix} d\mathbf{P} \\ 0 \end{bmatrix} = \begin{bmatrix} d\boldsymbol{\eta}_p \\ 0 \end{bmatrix} = (d\mathbf{A}_h)\mathbf{A}_h^{-1} \begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix}, \quad (7)$$

where $d\mathbf{P}$ is the infinitesimal change, or error in translation, of point $\mathbf{P}$ measured in the base reference system; this error may also be denoted by $d\boldsymbol{\eta}_p$.

Using the first-order kinematics, the relation between the translational and rotational error differentials may be written as

$$d\boldsymbol{\eta}_p = d\boldsymbol{\eta}_{o'} + \boldsymbol{\nu} d\phi \times \underline{\mathbf{o}'\mathbf{P}}, \quad (8)$$

where $\boldsymbol{\nu} d\phi$ is a 3×1 end-effector rotational error vector and $d\boldsymbol{\eta}_{o'}$ is a 3×1 end-effector translational error vector with respect to the base coincident point $\mathbf{o}'$; for simplicity of notation, $d\boldsymbol{\eta}_{o'} \equiv d\boldsymbol{\eta}$. One should also note that in equation (8), $\underline{\mathbf{o}'\mathbf{P}}$ denotes the distance between the end-effector point $\mathbf{P}$ and the end-effector point coincident with the origin $\mathbf{o}'$. The form of the end-effector rotational error vector and its corresponding 3×3 skew-symmetric matrix is

$$\boldsymbol{\nu} d\phi = \begin{bmatrix} \nu_x \\ \nu_y \\ \nu_z \end{bmatrix} d\phi \qquad \mathbf{N} d\phi = \begin{bmatrix} 0 & -\nu_z & \nu_y \\ \nu_z & 0 & -\nu_x \\ -\nu_y & \nu_x & 0 \end{bmatrix} d\phi \quad (9)$$

Equation (8) may now be written in the skew-symmetric form,

$$d\boldsymbol{\eta}_p = d\boldsymbol{\eta} + \mathbf{N} d\phi\, (\underline{\mathbf{o}'\mathbf{P}}) \quad (10)$$

or

$$\begin{bmatrix} d\boldsymbol{\eta}_p \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{N} d\phi & d\boldsymbol{\eta} \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix} \quad (11)$$

Comparison of equations (7) and (11) yields the following important relation among the various errors.

$$(d\mathbf{A}_h)\mathbf{A}_h^{-1} = \left[\begin{array}{c|c} \mathbf{N}d\phi & d\boldsymbol{\eta} \\ \hline \mathbf{0} & 0 \end{array}\right] \tag{12}$$

One must find a relation for the error quantities given on the right side of equation (12). The matrix $d\mathbf{A}_h$ may be written as,

$$d\mathbf{A}_h \cong \mathbf{A}_h^{\text{actual}} - \mathbf{A}_h^{(\text{nominal/current})}. \tag{13}$$

This will allow the use of nominal or updated D-H parameters in the error calculations; the latter is for the use of an iterative parameter estimation algorithm. Using the partitioned form of matrix $\mathbf{A}_h$ and its inverse, $(d\mathbf{A}_h)\mathbf{A}_h^{-1}$ may be written as,

$$\begin{aligned}(d\mathbf{A}_h)\mathbf{A}_h^{-1} &= \left[\begin{array}{c|c} d\mathbf{R}_h & d\mathbf{p}_h \\ \hline \mathbf{0} & 0 \end{array}\right]\left[\begin{array}{c|c} \mathbf{R}_h^T & -\mathbf{R}_h^T\mathbf{p}_h \\ \hline \mathbf{0} & 1 \end{array}\right] \\ &= \left[\begin{array}{c|c} (d\mathbf{R}_h)\mathbf{R}_h^T & d\mathbf{p}_h - (d\mathbf{R}_h)\mathbf{R}_h^T\mathbf{p}_h \\ \hline \mathbf{0} & 0 \end{array}\right] \end{aligned} \tag{14}$$

Comparing equations (12) and (14) allows one to write,

$$\mathbf{N}d\phi = (d\mathbf{R}_h)\mathbf{R}_h^T \qquad d\boldsymbol{\eta} = d\mathbf{p}_h - (d\mathbf{R}_h)\mathbf{R}_h^T\mathbf{p}_h. \tag{15}$$

The two relations shown in equation (15) will be used to obtain the rotational and translational end-effector error values, respectively.

Examining the left side of equation (12), $\mathbf{A}_h$ is a product of M transformation matrices (1), and each link transformation matrix, $\mathbf{A}_m$, is comprised of the four D-H parameters (5). Using the chain rule and postmultiplying by $\mathbf{A}_h^{-1}$, $(d\mathbf{A}_h)\mathbf{A}_h^{-1}$ may be written as,

$$\begin{aligned}(d\mathbf{A}_h)\mathbf{A}_h^{-1} = &\sum_{m=1}^{M} \frac{\partial \mathbf{A}_h}{\partial \theta_m}\mathbf{A}_h^{-1}d\theta_m + \sum_{m=1}^{M} \frac{\partial \mathbf{A}_h}{\partial s_m}\mathbf{A}_h^{-1}ds_m \\ &+ \sum_{m=1}^{M} \frac{\partial \mathbf{A}_h}{\partial a_m}\mathbf{A}_h^{-1}da_m + \sum_{m=1}^{M} \frac{\partial \mathbf{A}_h}{\partial \alpha_m}\mathbf{A}_h^{-1}d\alpha_m. \end{aligned} \tag{16}$$

We may now look at each summation of partial derivatives in equation (16) individually. Expanding the first summation group with respect to θ_m,

$$\frac{\partial \mathbf{A}_h}{\partial \theta_m} = (\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_{m-1})\frac{\partial \mathbf{A}_m}{\partial \theta_m}(\mathbf{A}_{m+1}\cdots\mathbf{A}_M). \tag{17}$$

Postmultiplying by the inverse of $\mathbf{A}_h$ given in equation (1),

$$\frac{\partial \mathbf{A}_h}{\partial \theta_m}\mathbf{A}_h^{-1} = (\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_{m-1})\frac{\partial \mathbf{A}_m}{\partial \theta_m}\mathbf{A}_m^{-1}(\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_{m-1})^{-1}. \tag{18}$$

Taking the partial derivative of $\mathbf{A}_m$ with respect to θ_m, one may obtain the relation

$$\frac{\partial \mathbf{A}_m}{\partial \theta_m}\mathbf{A}_m^{-1} = \left[\begin{array}{ccc|c} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right]. \tag{19}$$

Three unit vectors ($\mathbf{i}$ $\mathbf{j}$ $\mathbf{k}$), one in each axis direction of the mth link system, may be described in skew-symmetric form as

$$\mathbf{I} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{J} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad \mathbf{K} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{20}$$

Rewriting equation (19),

$$\frac{\partial \mathbf{A}_m}{\partial \theta_m}\mathbf{A}_m^{-1} = \left[\begin{array}{c|c} \mathbf{K} & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right]. \tag{21}$$

Substituting equation (21) into equation (18),

$$\frac{\partial \mathbf{A}_h}{\partial \theta_m}\mathbf{A}_h^{-1} = (\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_{m-1})\left[\begin{array}{c|c} \mathbf{K} & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right](\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_{m-1})^{-1}. \tag{22}$$

Equation (22) may be reduced further, using equation (3):

$$\frac{\partial \mathbf{A}_h}{\partial \theta_m}\mathbf{A}_h^{-1} = \left[\begin{array}{c|c} \mathbf{R}_m^L & \mathbf{p}_m \\ \hline \mathbf{0} & 1 \end{array}\right]\left[\begin{array}{c|c} \mathbf{K} & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right]\left[\begin{array}{c|c} \mathbf{R}_m^L & \mathbf{p}_m \\ \hline \mathbf{0} & 1 \end{array}\right]^{-1}. \tag{23}$$

Using the explicit matrix inversion, equation (23) may be written in the following form,

$$\frac{\partial \mathbf{A}_h}{\partial \theta_m}\mathbf{A}_h^{-1} = \left[\begin{array}{c|c} \mathbf{R}_m^L\mathbf{K}(\mathbf{R}_m^L)^T & -\mathbf{R}_m^L\mathbf{K}(\mathbf{R}_m^L)^T\mathbf{p}_m \\ \hline \mathbf{0} & 0 \end{array}\right]. \tag{24}$$

Equation (24) may be simplified further, if additional terms are introduced. The mth link system unit vectors ($\mathbf{i}$ $\mathbf{j}$ $\mathbf{k}$) must also be found with respect to the base coordinate system; these are denoted by ($\mathbf{u}_m$ $\mathbf{v}_m$ $\mathbf{w}_m$). The link system unit vectors may be mapped into the base system by using the relations,

$$\mathbf{u}_m = \mathbf{R}_m^L\mathbf{i} \qquad \mathbf{v}_m = \mathbf{R}_m^L\mathbf{j} \qquad \mathbf{w}_m = \mathbf{R}_m^L\mathbf{k}. \tag{25}$$

The skew-symmetric matrix representations of the base unit vectors ($\mathbf{u}_m$ $\mathbf{v}_m$ $\mathbf{w}_m$) are denoted by $\mathbf{U}_m$, $\mathbf{V}_m$, and $\mathbf{W}_m$, respectively. From the vector cross-product, $\mathbf{j} = \mathbf{k} \times \mathbf{i}$, the skew-symmetric form may be written; $\mathbf{j} = \mathbf{K}\mathbf{i}$. Combining this expression with equation (25), one finds

$$\mathbf{v}_m = \mathbf{R}_m^L\mathbf{K}(\mathbf{R}_m^L)^T\mathbf{u}_m. \tag{26}$$

Computation of the vector cross-product in the base system gives $\mathbf{v}_m = \mathbf{w}_m \times \mathbf{u}_m$, or in the skew-symmetric form,

$$\mathbf{v}_m = \mathbf{W}_m\mathbf{u}_m. \tag{27}$$

Comparing equations (26) and (27) reveals that

$$\mathbf{W}_m = \mathbf{R}_m^L \mathbf{K} (\mathbf{R}_m^L)^T. \tag{28}$$

Equation (24) may now be written in the following form:

$$\frac{\partial \mathbf{A}_h}{\partial \theta_m} \mathbf{A}_h^{-1} = \left[\begin{array}{c|c} \mathbf{W}_m & -\mathbf{W}_m \mathbf{p}_m \\ \hline \mathbf{0} & 0 \end{array}\right]. \tag{29}$$

The skew-symmetric relation $(-\mathbf{W}_m \mathbf{p}_m)$ may also be written in the vector form $\mathbf{p}_m \times \mathbf{w}_m$, where $\mathbf{p}_m$ is the position vector of the origin of the mth link system in the base system. Since its cross-product with the mth joint axis $\mathbf{w}_m$ is being found, one is not limited to the origin of the mth coordinate system, and any point $\boldsymbol{\rho}_m$ on the joint axis $\mathbf{w}_m$ may be used, or

$$\mathbf{p}_m \times \mathbf{w}_m \equiv \boldsymbol{\rho}_m \times \mathbf{w}_m. \tag{30}$$

Therefore, equation (29) may now be written in its most compact form:

$$\frac{\partial \mathbf{A}_h}{\partial \theta_m} \mathbf{A}_h^{-1} = \left[\begin{array}{c|c} \mathbf{W}_m & \boldsymbol{\rho}_m \times \mathbf{w}_m \\ \hline \mathbf{0} & 0 \end{array}\right]. \tag{31}$$

The expression for $(\partial \mathbf{A}_h / \partial s_m)\mathbf{A}_h^{-1}$ shall now be found. To establish this relationship, one may employ equations (17) and (18), substituting the D-H parameter s_m for θ_m. Taking the partial derivative of $\mathbf{A}_m$ with respect to the parameter s_m, and multiplying by $\mathbf{A}_m^{-1}$, one finds

$$\frac{\partial \mathbf{A}_m}{\partial s_m} \mathbf{A}_m^{-1} = \left[\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right]. \tag{32}$$

After substitution, one may write the expression corresponding to (22) as

$$\frac{\partial \mathbf{A}_h}{\partial s_m} \mathbf{A}_h^{-1} = (\mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_{m-1}) \left[\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{k} \\ \hline \mathbf{0} & 0 \end{array}\right] \times (\mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_{m-1})^{-1} \tag{33}$$

or, expanding further,

$$\begin{aligned} \frac{\partial \mathbf{A}_h}{\partial s_m} \mathbf{A}_h^{-1} &= \left[\begin{array}{c|c} \mathbf{R}_m^L & \mathbf{p}_m \\ \hline \mathbf{0} & 1 \end{array}\right] \left[\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{k} \\ \hline \mathbf{0} & 0 \end{array}\right] \\ &\quad \times \left[\begin{array}{c|c} (\mathbf{R}_m^L)^T & -(\mathbf{R}_m^L)^T \mathbf{p}_m \\ \hline \mathbf{0} & 1 \end{array}\right] \\ &= \left[\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{w}_m \\ \hline \mathbf{0} & 0 \end{array}\right]. \end{aligned} \tag{34}$$

The summation group with respect to α_m, shown in equation (16), will now be expanded. Again, equations (17) and (18) will be the basis for the relationship, substituting the D-H parameter α_m for θ_m. Following the procedure employed earlier,

$$\frac{\partial \mathbf{A}_m}{\partial \alpha_m} \mathbf{A}_m^{-1} = \left[\begin{array}{ccc|c} 0 & 0 & \sin\theta_m & -s_m \sin\theta_m \\ 0 & 0 & -\cos\theta_m & s_m \cos\theta_m \\ -\sin\theta_m & \cos\theta_m & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right]. \tag{35}$$

Equation (35) has a form uncommon to the preceding D-H derivations, but using the following substitutions, a simple compact form may be found. Let $\mathbf{u}_{m+1} = \mathbf{R}_{m+1}^L \mathbf{i}_{m+1}$, where $\mathbf{u}_{m+1}$ is the unit vector along axis x_{m+1} transformed into the base system. If the unit vector $\mathbf{i}_{m+1}$ in the $(m+1)$th link system is transformed into the mth system by rotation matrix $\mathbf{R}_m$, one finds,

$$\begin{bmatrix} \cos\theta_m & -\sin\theta_m & 0 \\ \sin\theta_m & \cos\theta_m & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta_m \\ \sin\theta_m \\ 0 \end{bmatrix} = \mathbf{i}_{m+1}^m. \tag{36}$$

The unit vector $\mathbf{i}_{m+1}^m$ may also be written in skew-symmetric matrix form

$$\mathbf{I}_{m+1}^m = \begin{bmatrix} 0 & 0 & \sin\theta_m \\ 0 & 0 & -\cos\theta_m \\ -\sin\theta_m & \cos\theta_m & 0 \end{bmatrix}. \tag{37}$$

Therefore, one may now write,

$$\frac{\partial \mathbf{A}_m}{\partial \alpha_m} \mathbf{A}_m^{-1} = \left[\begin{array}{c|c} \mathbf{I}_{m+1}^m & -s_m \mathbf{I}_{m+1}^m \mathbf{k} \\ \hline \mathbf{0} & 0 \end{array}\right]. \tag{38}$$

Using equations (17) and (18),

$$\begin{aligned} \frac{\partial \mathbf{A}_h}{\partial \alpha_m} \mathbf{A}_h^{-1} &= \left[\begin{array}{c|c} \mathbf{R}_m^L & \mathbf{p}_m \\ \hline \mathbf{0} & 1 \end{array}\right] \left[\begin{array}{c|c} \mathbf{I}_{m+1}^m & -s_m \mathbf{I}_{m+1}^m \mathbf{k} \\ \hline \mathbf{0} & 0 \end{array}\right] \\ &\quad \times \left[\begin{array}{c|c} (\mathbf{R}_m^L)^T & -(\mathbf{R}_m^L)^T \mathbf{p}_m \\ \hline \mathbf{0} & 1 \end{array}\right]. \end{aligned} \tag{39}$$

Using substitutions similar to those described earlier,

$$\mathbf{U}_{m+1} = \mathbf{R}_m^L \mathbf{I}_{m+1}^m (\mathbf{R}_m^L)^T, \tag{40}$$

where $\mathbf{U}_{m+1}$ is the skew-symmetric representation of $\mathbf{u}_{m+1}$. With the use of these simplifications, equation (39) may be written as follows:

$$\begin{aligned} \frac{\partial \mathbf{A}_h}{\partial \alpha_m} \mathbf{A}_h^{-1} &= \left[\begin{array}{c|c} \mathbf{U}_{m+1} & -\mathbf{U}_{m+1}\mathbf{p}_m - s_m \mathbf{U}_{m+1}\mathbf{w}_m \\ \hline \mathbf{0} & 0 \end{array}\right] \\ &= \left[\begin{array}{c|c} \mathbf{U}_{m+1} & (\mathbf{p}_m + s_m \mathbf{w}_m) \times \mathbf{u}_{m+1} \\ \hline \mathbf{0} & 0 \end{array}\right]. \end{aligned} \tag{41}$$

For simplification, the point $\mathbf{q}_{m+1} = (\mathbf{p}_m + s_m \mathbf{w}_m)$ is the intersection of $\mathbf{x}_{m+1}$ and $\mathbf{z}_m$, but the cross-product allows one to choose any point $\boldsymbol{\sigma}_{m+1}$ on the $\mathbf{x}_{m+1}$ axis, yielding

$$(\mathbf{p}_m + s_m \mathbf{w}_m) \times \mathbf{u}_{m+1} \equiv \mathbf{q}_{m+1} \times \mathbf{u}_{m+1} \equiv \boldsymbol{\sigma}_{m+1} \times \mathbf{u}_{m+1}. \tag{42}$$

It is now possible to write the most compact form of equation (41)

$$\frac{\partial \mathbf{A}_h}{\partial \alpha_m}\mathbf{A}_h^{-1} = \left[\begin{array}{c|c} \mathbf{U}_{m+1} & \boldsymbol{\sigma}_{m+1} \times \mathbf{u}_{m+1} \\ \hline \mathbf{0} & 0 \end{array}\right]. \tag{43}$$

The same analysis used for the preceding three D-H parameters can now be used for the final parameter, link length a_m:

$$\frac{\partial \mathbf{A}_m}{\partial a_m}\mathbf{A}_m^{-1} = \left[\begin{array}{ccc|c} 0 & 0 & 0 & \cos\theta_m \\ 0 & 0 & 0 & \sin\theta_m \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right]. \tag{44}$$

Using predefined terms,

$$\frac{\partial \mathbf{A}_m}{\partial a_m}\mathbf{A}_m^{-1} = \left[\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{i}_{m+1}^m \\ \hline \mathbf{0} & 0 \end{array}\right], \tag{45}$$

$$\frac{\partial \mathbf{A}_h}{\partial a_m}\mathbf{A}_h^{-1} = \left[\begin{array}{c|c} \mathbf{R}_m^L & \mathbf{p}_m \\ \hline \mathbf{0} & 1 \end{array}\right]\left[\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{i}_{m+1}^m \\ \hline \mathbf{0} & 0 \end{array}\right] \times \left[\begin{array}{c|c} (\mathbf{R}_m^L)^T & -(\mathbf{R}_m^L)^T\mathbf{p}_m \\ \hline \mathbf{0} & 1 \end{array}\right], \tag{46}$$

or,

$$\frac{\partial \mathbf{A}_h}{\partial a_m}\mathbf{A}_h^{-1} = \left[\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{u}_{m+1} \\ \hline \mathbf{0} & 0 \end{array}\right]. \tag{47}$$

Now that the partial derivative expressions with respect to the four D-H parameters at each of the M joints have been determined, they may be substituted into equation (16).

$$(d\mathbf{A}_h)\mathbf{A}_h^{-1} = \sum_{m=1}^{M}\left[\begin{array}{c|c} \mathbf{W}_m & \boldsymbol{\rho}_m \times \mathbf{w}_m \\ \hline \mathbf{0} & 0 \end{array}\right] d\theta_m + \sum_{m=1}^{M}\left[\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{w}_m \\ \hline \mathbf{0} & 0 \end{array}\right] ds_m + \sum_{m=1}^{M}\left[\begin{array}{c|c} \mathbf{U}_{m+1} & \boldsymbol{\sigma}_{m+1} \times \mathbf{u}_{m+1} \\ \hline \mathbf{0} & 0 \end{array}\right] d\alpha_m + \sum_{m=1}^{M}\left[\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{u}_{m+1} \\ \hline \mathbf{0} & 0 \end{array}\right] da_m \tag{48}$$

In the expression for $(d\mathbf{A}_h)\mathbf{A}_h^{-1}$ given in equation (12), the 4×4 end-effector error matrix has been partitioned into a 3×3 skew-symmetric matrix and a 3×1 vector. Each of the four matrices shown in equation (48) is also shown in its partitioned form. If one equates the vector representation of the skew-symmetric matrices on the left and right sides of equation (48), along with the 3×1 vector partition on both sides of the equality, one obtains the following vector relations:

$$\begin{bmatrix} \nu d\phi \\ d\boldsymbol{\eta} \end{bmatrix} = \sum_{m=1}^{M}\begin{bmatrix} \mathbf{w}_m \\ \boldsymbol{\rho}_m \times \mathbf{w}_m \end{bmatrix} d\theta_m + \sum_{m=1}^{M}\begin{bmatrix} \mathbf{0} \\ \mathbf{w}_m \end{bmatrix} ds_m + \sum_{m=1}^{M}\begin{bmatrix} \mathbf{u}_{m+1} \\ \boldsymbol{\sigma}_{m+1} \times \mathbf{u}_{m+1} \end{bmatrix} d\alpha_m + \sum_{m=1}^{M}\begin{bmatrix} \mathbf{0} \\ \mathbf{u}_{m+1} \end{bmatrix} da_m \tag{49}$$

or equation (49) may be written in terms of the following matrices:

$$\begin{bmatrix} \nu d\phi \\ d\boldsymbol{\eta} \end{bmatrix} = \mathbf{J}_\theta d\boldsymbol{\theta} + \mathbf{J}_s d\mathbf{s} + \mathbf{J}_\alpha d\boldsymbol{\alpha} + \mathbf{J}_a d\mathbf{a}, \tag{50}$$

where $\mathbf{J}_\theta$, $\mathbf{J}_s$, $\mathbf{J}_\alpha$, and $\mathbf{J}_a$ are special Jacobian matrices (dimensions $6 \times M$) identified with the associated parameters, and $d\boldsymbol{\theta}$, $d\mathbf{s}$, $d\boldsymbol{\alpha}$, $d\mathbf{a}$ are the vectors (dimensions $M \times 1$) containing the D-H error parameters. Bennett and Hollerbach (1989) comment on the formulation of the special Jacobian matrices, but a formal derivation is not presented. The structure of these special Jacobian matrices is more clearly seen in the following explicit forms:

$$\mathbf{J}_\theta = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_M \\ \boldsymbol{\rho}_1 \times \mathbf{w}_1 & \boldsymbol{\rho}_2 \times \mathbf{w}_2 & \cdots & \boldsymbol{\rho}_M \times \mathbf{w}_M \end{bmatrix}, \tag{51}$$

$$\mathbf{J}_s = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_M \end{bmatrix}, \tag{52}$$

$$\mathbf{J}_\alpha = \begin{bmatrix} \mathbf{u}_2 & \mathbf{u}_3 & \cdots & \mathbf{u}_{M+1} \\ \boldsymbol{\sigma}_2 \times \mathbf{u}_2 & \boldsymbol{\sigma}_3 \times \mathbf{u}_3 & \cdots & \boldsymbol{\sigma}_{M+1} \times \mathbf{u}_{M+1} \end{bmatrix}, \tag{53}$$

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{u}_2 & \mathbf{u}_3 & \cdots & \mathbf{u}_{M+1} \end{bmatrix}. \tag{54}$$

The columns of $\mathbf{J}_\theta$ and $\mathbf{J}_\alpha$ are Plucker's line coordinates for the link's z- and x-axes, respectively. The columns of $\mathbf{J}_s$ and $\mathbf{J}_a$ contain the null vectors together with the unit vectors along the link's z- and x-axes, respectively. Intuitively, the columns of $\mathbf{J}_\theta$ can be related to real or fictitious revolute joints along z-axes; similarly, the columns of $\mathbf{J}_\alpha$ can be related to fictitious revolute joints along the x-axes of the D-H link systems. On the other hand, the columns of $\mathbf{J}_s$ and $\mathbf{J}_a$ can be related to real or fictitious prismatic joints along the z- and x-axes, respectively, of the D-H link systems. In kinematic control of the robot, the only Jacobian encountered is a hybrid of $\mathbf{J}_\theta$ and $\mathbf{J}_s$ (Gupta 1985), depending on the type of joint used, which is simply denoted as Jacobian matrix $\mathbf{J}$. In the parameter identification and compensation problem, these four special Jacobian matrices given in equations (51) through (54) may be used to solve for the Denavit and

Hartenberg error parameters, which are given by

$$\mathbf{d\theta} = \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ \vdots \\ d\theta_M \end{bmatrix} \qquad \mathbf{ds} = \begin{bmatrix} ds_1 \\ ds_2 \\ \vdots \\ ds_M \end{bmatrix}$$

$$\mathbf{d\alpha} = \begin{bmatrix} d\alpha_1 \\ d\alpha_2 \\ \vdots \\ d\alpha_M \end{bmatrix} \qquad \mathbf{da} = \begin{bmatrix} da_1 \\ da_2 \\ \vdots \\ da_M \end{bmatrix} \tag{55}$$

Iterative Regression Algorithm for Parameter Identification

Using the relation between the four special Jacobian matrices given in equation (50), one may obtain values for the D-H error parameters $d\boldsymbol{\theta}$, $d\mathbf{s}$, $d\boldsymbol{\alpha}$, and $d\mathbf{a}$. From equation (50), one is able to determine a total of $4M$ D-H error parameter values; for a six-jointed robot, there are 24 error parameters. Most industrial robots use two joint types: the revolute joint and the prismatic joint. In each of these joints, the amount that the joint rotates or translates is denoted by a joint variable: s for the prismatic joint and θ for the revolute joint. For the revolute and prismatic joints, $d\theta$ and ds, respectively, are the corresponding D-H errors, and these values will be dependent on the corresponding joint variable values. Because the joint value error parameters will vary according to the joint angle values, these errors are said to be position-dependent. The remaining three D-H parameter errors associated with each link system are fixed in value, or position independent, as are the associated D-H parameters. Therefore, equation (50) may be rewritten in terms of position-dependent (pd) and position-independent (pi) D-H error values,

$$\begin{bmatrix} \nu d\phi \\ d\eta \end{bmatrix} - (\mathbf{J}_\theta d\boldsymbol{\theta} + \mathbf{J}_s d\mathbf{s})^{pd} = (\mathbf{J}_\theta d\boldsymbol{\theta} + \mathbf{J}_s d\mathbf{s})^{pi} + \mathbf{J}_\alpha d\boldsymbol{\alpha} + \mathbf{J}_a d\mathbf{a}. \tag{56}$$

Using equation (56), one may compute the $3M$ position-independent D-H error parameters. By measurement, the end-effector position and orientation matrix, $\mathbf{A}_h^{\text{actual}}$, and the M values of the position-dependent error values may be found at each manipulator position, although the latter may be set to zero if they are not significant. Using the preceding general formulation, a Fortran program was developed to implement parameter identification. The program allows one to employ either the $4M$-parameter or the $3M$-parameter model; however, only the latter will be considered in detail since the former lacks a sound theoretical basis. The program will allow for the use of any manipulator that is comprised of revolute or prismatic joints. All data presented was compiled using a PUMA-type industrial manipulator.

Each of the $3M$ position-independent error parameters included in the initial model may not be of equal significance. Therefore, it would be desirable to arrive at a method by which one has some control over the position-independent error parameters that are to be included in the model. In the PUMA example, one may want to include some combination of $d\alpha_m$, da_m, and ds_m; the largest model case would involve computing all $3M$ D-H error parameters. Employing the prescribed parameter identification methodology, each robot end-effector position will yield a set of six independent equations, as shown in equation (56). The computation of the largest model case would require a minimum of N=$3M/6$ positions (for PUMA, 3 positions). When a larger number of positions is used ($N > M/2$), then the techniques of regression analysis must be employed (Draper and Smith, 1966). The choice of the positions is arbitrary. At first, it was decided to allow only significant error parameters in the model. To determine the best subset of position-independent error parameters, the R-Best algorithm from the IMSL (1988) numerical library was chosen. This algorithm assesses the best linear subset regressions with subsets containing $3M$, $3M-1$, ..., and one position-independent D-H error parameters within the given manipulator model. This solution method is time consuming, but because this is an off-line procedure, computational time is not a constraint. The R-Best algorithm chooses among all of the possible combinations in each of the $3M$ subsets by examining the error of linear regression, using the R^2 criterion (Pieper and Roth 1969).

$$R^2 = \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2}, \tag{57}$$

where y_i is the dependent variable or, loosely, the end-effector error value on the left side of equation (56), $\hat{y}_i$ is the value found through the regression, and $\bar{y}_i$ is the mean of the dependent value; as $R^2 \rightarrow 1$, the linear regression fit improves.

In all test cases, built-in position-independent D-H error parameter levels of $\pm ds$, $\pm d\alpha$, and $\pm da$ were used; for simplicity, alternating ($\pm$) built-in error levels were used. In some examples, a built-in position-dependent parameter error level of $\pm d\theta$ was included; then the left side of equation (56) may be reformulated by allowing $\theta_m^{\text{actual}} = \theta_m^{\text{nominal}} + d\theta_m$, at each position. Table 1 gives the joint values for a PUMA-type manipulator at each of the six positions that were used in the example. Although configuration singularities may be avoided through position choices, mathematical singularities can occur. For example, if $\alpha_m = 0$, then $z_m \parallel z_{m+1}$, and ds_m and ds_{m+1} become dependent error parameters. Other mathematical singularities may also exist, but they do not occur in the PUMA examples. The effects of parameter identification may be seen in the following

example of a PUMA with built-in, or induced, parameter error levels of ±0.01 radians and ±0.001 inches in the rotational and translational parameters, respectively. Examining Table 2, column 2 shows the built-in parameter error levels, and column 3 shows the identified parameter error levels after standard linear regression using the R-Best algorithm on the full $3M$-parameter model. As one can see, the value of ds_3 has been set to zero; parameter errors ds_2 and ds_3 are dependent, since the nominal value of α_2 is zero. From Table 2, one is able to see that the translational parameter error recovery is poor. Table 3 shows the worst-case percentage rotational parameter (top entry) and translational parameter (lower entry) identification using the $3M$-parameter model and standard regression at various built-in rotational and translational parameter error levels. The percent identification of rotational error parameters is quite good at all built-in error levels. However, as built-in rotational error levels increase and built-in translational error levels decrease, the percentage identification of translational error parameters gets quite poor. Table 4 shows the worst-case percentage rotational parameter (top entry) and translational parameter (lower entry) identification results that are obtained through the use of the subset regression, R-Best, at certain error parameter levels. Because of the dependence of ds_2 and ds_3, the 18-parameter model is not possible, and because of the nature of the induced error used in the given examples, the subsets containing 16 and 17 parameters yield similar results. As one can see, the 15-parameter model produces poor results. Owing to the inconsistent qual-

Table 1. Nominal PUMA D-H Values (angles given in radians and lengths given in inches)

Link	θ	s	α	a
1	θ_1	26.0	1.570796	0.0
2	θ_2	−8.0	0.0	17.0
3	θ_3	0.0	1.570796	0.0
4	θ_4	17.0	1.570796	0.0
5	θ_5	0.0	1.570796	0.0
6	θ_6	−2.25	3.141593	0.0

Position	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	0.7854	−0.5236	−1.2618	0.2618	−1.6981	−0.3490
2	0.5708	−0.3491	−1.4363	0.1745	−0.7854	−1.5360
3	0.1745	−1.0944	−0.9199	0.1745	−0.7854	−0.2618
4	1.0454	−0.1230	−0.9090	1.2323	−1.2420	−0.7897
5	1.0000	−0.3000	−0.5656	0.7890	−1.0000	−0.5400
6	0.5943	−0.2340	−0.3405	1.0900	−0.7897	−0.1000

Table 2. Induced Position-Independent Parameter Error Values and Values Obtained After Using the Iterative Regression Algorithm

D-H Parameter	Induced Parameter Error	Iteration 1	Iteration 2	Iteration 3
α_1	−0.010	−0.010005	−0.010000	−0.010
α_2	0.010	0.009949	0.010000	0.010
α_3	−0.010	−0.010007	−0.010000	−0.010
α_4	0.010	0.010010	0.010000	0.010
α_5	−0.010	−0.009999	−0.010000	−0.010
α_6	0.010	0.009994	0.010000	0.010
a_1	0.001	−0.000493	0.000999	0.001
a_2	−0.001	−0.001106	−0.001000	−0.001
a_3	0.001	0.000495	0.001000	0.001
a_4	−0.001	0.000306	−0.001000	−0.001
a_5	0.001	0.001713	0.001000	0.001
a_6	−0.001	−0.002790	−0.001000	−0.001
s_1	−0.001	−0.001242	−0.000999	−0.001
s_2	0.001	−0.001031	0.000994	0.001
s_3	−0.001	0.000000	−0.000994	−0.001
s_4	0.001	0.002058	0.001000	0.001
s_5	−0.001	−0.001020	−0.000999	−0.001
s_6	0.001	0.001061	0.001000	0.001

Table 3. Worst-Case Percentage Recovery of Position-Independent Parameter Error Values After Standard Regression Using the Parameter Identification Algorithm*

Induced Translational Parameter Error Levels (Inches)	Induced Rotational Parameter Error Levels (Radians) ±0.01	±0.005	±0.001	±0.0001
±0.2	99.5063	99.7565	99.8686	99.8736
	98.2847	99.3197	99.8545	99.9899
±0.1	99.5002	99.7504	99.9313	99.9365
	97.5430	99.1319	99.8573	99.9999
±0.05	99.4971	99.7473	99.9515	99.9679
	96.0592	95.7573	99.8628	99.9887
±0.01	99.4947	99.7448	99.9490	99.9931
	81.7728	95.7610	99.7499	99.9885
±0.005	99.4944	99.7445	99.9487	99.9951
	69.3534	90.7657	99.5982	99.9863
±0.001	99.4941	99.7443	99.9484	99.9949
	—	54.5375	98.1339	99.9749
±0.0001	99.4941	99.7442	99.9484	99.9948
	—	—	81.6550	99.8388

*Top and bottom values denote rotation and translation, respectively.

Table 4. Optimal Subset Regression Using the R-Best Algorithm*

Number of Variables in Model	Induced Parameter Error Levels (Rotation in Radians and Translation in Inches) ±0.001 ±0.1	±0.001 ±0.01	±0.001 ±0.001	±0.001 ±0.0001
18	—	—	—	—
	—	—	—	—
17	98.7500	99.9455	99.8750	99.8630
	99.7511	99.7335	98.0017	79.8130
16	98.0721	99.9093	99.8748	99.8635
	99.7667	99.7410	98.0017	79.8180
15	22.8900	23.4700	92.4430	99.3400
	89.5790	41.3224	42.3000	41.5670

*Top and bottom values denote rotation and translation, respectively.

ity of parameter identification by standard regression in the examples shown, it was decided to use an iterative regression algorithm.

The Iterative Regression Algorithm (IRA) consists of updating the $3M$ D-H parameter error values by adding the associated parameter error values obtained through the linear regression, reformulating the linear system given in equation (56), and repeating the regression procedure until convergence occurs. The criteria used to determine convergence is a check on each position-independent parameter error value after each iteration. If the absolute change in position-independent parameter error values between successive iterations is less than $\epsilon = 10^{-7}$, convergence is achieved, and the iteration process is terminated. To judge the effectiveness of the identification, one can examine the end-effector error. This error has two parts, rotational and translational, each comprised of a sum of squares of the components (given in equation (15)) (i.e., $(\nu d\phi_x^2 + \nu d\phi_y^2 + \nu d\phi_z^2)$ and $(d\eta_x^2 + d\eta_y^2 + d\eta_z^2)$), respectively. During the iteration process, if the end-effector error quantities showed an increase during successive iterations, a step-cut was used, and the ith iteration parameter error values were halved, up to eight times, until both sum of squares decreased.

The linear regression used to solve the $6N \times 3M$ overdetermined linear system, **Bx=w**, uses the least squares algorithm (Pieper and Roth 1969), and the solution may be expressed as,

$$\mathbf{x} = (\mathbf{B}^T\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{w}) \tag{58}$$

where $\mathbf{x}$ is the vector of unknown error parameters or regressors, $\mathbf{B}$ is the matrix of regressor coefficients, and $\mathbf{w}$ is the vector of known end-effector error values. To determine dependence between two regressors, the algorithm examines a decomposition of the sum of squares and cross-products matrix $\mathbf{B}^T\mathbf{B}$. The use of this dependence declaration process did not ensure uniformity in several examples that were considered. A statistical improvement was implemented through regressor scaling, with respect to the mean and standard deviation, of the corresponding coefficients during the regression. Thus, the correlation matrix $\mathbf{r} = [r_{ij}]$ was used instead of the sum of squares and cross-products matrix $\mathbf{B}^T\mathbf{B}$ in equation (58). This substitution provided a uniform method for declaring linear dependence of regressors; dependence is declared for the ith and jth regressors if $r_{ij} \sim 1$. The system of $6N$ equations given in equation (56) may be transformed into the correlation form using the relation

$$\sum_{i=1}^{3M} \frac{(b_i - \bar{b}_i)}{\sigma_i} x_i' = \frac{(w - \bar{w})}{\sigma_w}, \tag{59}$$

where b_i is the ith regressor coefficient value and x_i' is the ith unknown regressor value; the actual regressor value may be found by $x_i = x_i' \sigma_w / \sigma_i$. Table 2 shows the improvements obtained after successive iterations using built-in rotational error parameter levels of ± 0.01 radians and translational error parameter levels of ± 0.001 inches. After the first iteration (i.e., the standard regression), the translational error parameter identification is very poor, but after an additional iteration, there is an almost perfect parameter identification. The identification algorithm converges after three iterations. Because this is an off-line procedure, a large number of iterations is not a constraint; at most error level combinations, the number of iterations needed for convergence was approximately three. The use of this iterative parameter identification algorithm allows for the determination of the position-independent D-H parameter error values, which, if not included in the actual manipulator model, would result in large end-effector errors.

Parameter Identification of Random Parameter Errors

In the previous examples, the built-in parameter errors and the end-effector information were exact. The use of exact values helps illustrate the effects that may be obtained through the use of the parameter identification algorithm, but in actuality, all measured values will have some amount of uncertainty built in. For example, the instrumentation used to find the end-effector information at each position will be accurate to some tolerance. In this section, two situations will be examined in which random errors are prevalent in either the built-in parameter error values or in the end-effector information.

Table 5. Average and Standard Deviation of the Random Built-in Parameter Error Values and the Associated Error Parameters Identified Using the Iterative Regression Algorithm

	Average and Standard Deviation of Built-in D-H Parameter Error (Standard deviation given in parenthesis)					
Parameter	1	2	3	4	5	6
α	+2.00E−7 (0.57E−5)	+5.00E−7 (0.56E−5)	−4.00E−7 (0.61E−5)	−3.00E−7 (0.56E−5)	0.0 (0.68E−5)	+6.00E−7 (0.55E−5)
a	+0.43E−5 (0.58E−4)	+0.44E−5 (0.51E−4)	−0.58E−5 (0.57E−4)	+0.12E−5 (0.58E−4)	+0.44E−5 (0.53E−4)	−0.23E−5 (0.57E−4)
s	+0.42E−5 (0.65E−4)	+0.28E−5 (0.59E−4)	+0.08E−5 (0.54E−4)	+0.60E−5 (0.53E−4)	+0.47E−5 (0.67E−4)	+0.48E−5 (0.55E−4)

	Identified D−H Parameter Error Values					
Parameter	1	2	3	4	5	6
α	+0.15E−5	+0.11E−5	+0.12E−5	−0.02E−5	−0.13E−5	+0.07E−5
a	−0.03E−4	+0.63E−4	+0.21E−4	+0.69E−4	−0.56E−4	+0.25E−4
s	−0.26E−4	+0.32E−4	+0.0	−0.18E−4	−0.12E−4	+0.55E−4

The first situation is one in which a large group of robots is fabricated and assembled. Due to manufacturing tolerances and the assembly process, the position-independent parameter errors may exhibit some distribution within the group. In other words, the built-in error levels in the position-independent parameters will be distributed around some parameter error value. The assumption of finding a common parameter value for a group of randomly distributed error values is valid only if the deviations about the base error are small; because of precision manufacturing and assembly of robots, this assumption is valid. In the examples that will be presented, a built-in base error of zero was chosen, and a random position-independent error value was assigned to each corresponding parameter. The random error was within a $\pm$ range and has a Gaussian distribution. In this case, the use of many positions will yield a more accurate solution. The expected outcome is for each of the $3M$ identified parameter error values to be in the neighborhood of the mean parameter error value of the given parameter (i.e., $\bar{x}_i \pm \sigma_i$). Table 5 shows the identified parameter error values using the Iterative Regression Algorithm when random position-independent parameters are included. In this example, 50 independent positions were used, along with random rotational parameter errors within the range ± 0.00001 radians and random translational parameter error values within the range ± 0.0001 inches. From Table 5, the identification of all position-independent error parameters is within the expected range of $\bar{x}_i \pm \sigma_i$, with the exception of a_2 and a_4, although they are in the range $\bar{x}_i \pm 2\sigma_i$. Under all test cases that include position-independent random parameter error ranges (only one case is shown in Table 5), the identified rotational error parameters fell within one standard deviation of the mean built-in error value. The translational error parameter identification, on the other hand, was unpredictable and is dependent on the ranges of the built-in error parameters. This parameter identification algorithm allows one to identify the position-independent parameter error values for a group of manufactured robots in which there is some unknown random distribution of position-independent parameter errors.

An orthogonality correction was also used. This correction uses the orthogonal properties of the end-effector orientation, in which the cross-product of any two columns will produce the third. It was determined that the results were unreliable because of the fact that large errors in the orientations propagated through the cross-product, and therefore, this correction was not utilized.

The second situation in which random errors may arise is in the measurement of the actual position and orientation of the robot end effector. The precision of the measuring instrumentation has a large impact on the accuracy of the identified parameters. To approximate this effect, each value of the end-effector position and orientation matrix $\mathbf{A}_h^{\text{actual}}$ was altered by some random percentage within a given range; the ranges were $\pm 0.1\%$, $\pm 0.01\%$, and $\pm 0.001\%$. Table 6 shows the worst-case percentage identification of rotational and translational parameter error values, after the first and final iterations, for various built-in parameter error levels. In Table 6, there is a large improvement in the recovery percentage between the first

Table 6. Worst-Case Percentage Identification After the Initial and Final Iteration With Random Percentage Deviations Introduced into the End-effector Information, Under Various Built-in Parameter Error Levels*

Built-in Parameter	exact		0.001%		0.01%		0.1%	
Error Values	Initial	Final	Initial	Final	Initial	Final	Initial	Final
0.01	98.598	100.00	98.594	99.992	98.557	99.905	97.069	97.931
0.1	92.407	100.00	92.656	98.501	93.712	99.288	86.016	86.449
0.01	98.696	100.00	98.698	99.992	98.724	99.944	97.435	98.768
0.001	—	100.00	—	67.372	—	—	—	—
0.001	99.853	100.00	99.776	99.849	99.092	99.091	91.178	91.110
0.01	99.291	99.911	95.053	94.910	81.474	81.245	56.644	56.861
0.001	99.862	100.00	99.813	99.892	99.421	99.370	93.439	93.311
0.001	99.291	99.911	95.191	92.161	—	—	—	—
0.0001	99.976	100.00	98.924	98.944	92.573	92.590	—	—
0.01	99.997	99.991	97.933	97.916	75.887	75.879	—	—

*Top and bottom values denote rotation and translation, respectively, and [—] indicates very poor identification percentage.

Table 7. End-effector Rotational and Translational Errors After the Initial and Final Iteration With Random Percentage Deviations Introduced into the End-effector Information, Under Various Built-in Parameter Error Levels*

Built-in Parameter	exact		0.001%		0.01%		0.1%	
Error Values	Initial	Final	Initial	Final	Initial	Final	Initial	Final
0.01	0.1148	0.0	0.1148	0.0	0.1149	0.0001	0.1146	0.0013
0.1	2.6483	0.0	2.6483	0.0011	2.6498	0.0104	2.6679	0.1357
0.01	0.1148	0.0	0.1148	0.0	0.1148	0.0001	0.1152	0.0009
0.001	2.0125	0.0	2.0124	0.0008	2.0109	0.0103	2.0130	0.0734
0.001	0.0115	0.0	0.0115	0.0	0.0115	0.0001	0.0115	0.0011
0.01	0.2666	0.0	0.2670	0.0012	0.2685	0.0096	0.3276	0.1225
0.001	0.0115	0.0	0.0115	0.0	0.0115	0.0001	0.0117	0.0011
0.001	0.2052	0.0	0.2052	0.0011	0.2072	0.0098	0.2219	0.0952
0.0001	0.0001	0.0	0.0011	0.0	0.0012	0.0001	0.0016	0.0011
0.01	0.1526	0.0	0.1526	0.0012	0.1532	0.0131	0.1790	0.0871

*Top and bottom values denote rotation and translation, respectively.

and final iterations when the exact end-effector information is used. As the percentage of random error in $\mathbf{A}_h^{\text{actual}}$ increases, the percentage of position-independent parameter identification between the first and final iterations shows less of an increase, and in some cases, it decreases. At certain built-in position-independent parameter error levels, the inclusion of random $\mathbf{A}_h^{\text{actual}}$ percentage error produces very poor percent recoveries at all iterations, signified by the symbol (–). Table 7 shows the changes in the end-effector error values under various levels of percentage error in $\mathbf{A}_h^{\text{actual}}$, after the first and final iterations. As the percentage of error in $\mathbf{A}_h^{\text{actual}}$ increases, the ratio between the initial and final error at the end-effector decreases, thus revealing a decrease in the effectiveness of the formulated parameter identification algorithm when the accuracy of the end-effector measurement system is poor. If one can estimate the accuracy of end-effector measurements, then by knowing the parameter error levels that are identified, one may determine the confidence levels in the identified values.

Conclusion

An Iterative Regression Algorithm has been presented that allows one to identify the D-H parameter error values for any given manipulator. With the given formulation, one may model any M-jointed robot comprised of some

combination of revolute and prismatic joints. The formulation of the error model with special Jacobian matrices allows for a simple determination of dependency between two error parameters, thus avoiding mathematical singularities. Through coefficient scaling, the correlation matrix was used for linear regression, thus ensuring consistency in the dependence declaration among error parameters. The formulation of the parameter identification algorithm can also allow for the calculation of all $4M$ D-H parameter error values, although because of the reasons presented, a $3M$-parameter error model based only on position-independent errors was used in this article. In addition, one is able to choose the number of positions that are used, ($N > M/2$); because this is an off-line procedure, computational time is not a constraint. Various examples using a PUMA-type robot have been presented in which the Iterative Regression Algorithm is used to identify position-independent parameter error values. The use of statistical methods has been shown, both for standard regression and in the formation of the best error parameter subsets, although both were rejected because of poor identification at various built-in parameter error levels. It is also demonstrated how poor measurement accuracy can affect the results of the Iterative Regression Algorithm and to what extent. The results of the use of parameter identification in the preceding cases are further explored by Mirman (1990). Through the use of this iterative parameter identification algorithm, the unknown parameter error values can be determined accurately and used to produce a high level of end-effector accuracy.

References

Bennett, D., and Hollerbach, J. M. 1989. Identifying the kinematics of non-redundant serial chain manipulators by a closed loop approach. *Proceedings of the 4th ICAR*, Columbus, Ohio.

Denavit, J., and Hartenberg, R. S. 1955. A kinematic notation of lower pair mechanisms based on matrices. *J. Appl. Mechanics* 22:215–221.

Draper, N. R., and Smith, H. 1966. *Applied Regression Analysis*. New York: J. Wiley and Sons.

Gupta, K. C. 1985. Discussion on the formulation of the Jacobian matrix. *J. Mech. Transmissions Automation Design* 107:237–238.

Hsu, T. W., and Everett, L. J. 1985 (Boston). Identification of the kinematic parameters of a robot manipulator for positional accuracy improvement. *Computers in Engineering Conference*, Vol. 1, pp. 263–267.

Huang, Z. 1989. Error analysis of position and orientation in robot manipulators. *J. Mech. Machine Theory* 22(6):577–581.

Ibarra, R., and Perreira, N. 1986. Determination of linkage parameter and pair variable errors in open chain kinematic linkages using a minimal set of pose measurement data. *J. Mech. Transmissions Automation Design* 108:159–166.

International Mathematical and Statistical Libraries (IMSL). 10th ed. Houston, Texas.

Jarvis, J. 1987. Microsurveying: Towards robot accuracy. *IEEE Conference on Robotics and Automation*, pp. 1660–1665.

Mirman, C. R. 1990. Iterative parameter identification and compensation for robots using special Jacobian matrices. Ph.D. dissertation, University of Illinois at Chicago, Chicago, IL.

Mirman, C. R., and Gupta, K. C. 1991. Robot parameter identification and compensation using sets of Jacobian matrices. *IFToMM World Congress*.

Pieper, D. L., and Roth, B. 1969. The kinematics of manipulators under computer control. *Proceedings of the Second IFToMM World Congress*, Kupari, Yugoslavia, pp. 159–168.

Tang, S., and Ching-Cheng, W. 1987. Computation of the effects of link deflections and joint compliance on robot positioning. *IEEE Conference on Robotics and Automation*, pp. 910–915.

Veitschegger, W., and Wu, C. H. 1987. A method for calibrating and compensating robot kinematic errors. *IEEE Conference on Robotics and Automation*, Vol. 1, pp. 39–44.

Whitney, D. E., Lozinski, C. A., and Rourke, J. M. 1986. Industrial robot forward calibration method and results. *J. Dyn. Sys. Measurement Control* 108:1–8.