A VIRTUAL SENSOR APPROACH TO ROBOT KINEMATIC IDENTIFICATION: THEORY AND EXPERIMENTAL IMPLEMENTATION*

Patrick F. Muir

Intelligent Machines Principles Division
Sandia National Laboratories
Albuquerque, NM 87185

ABSTRACT

Kinematic parameter estimates are required for the implementation of robot servo-control algorithms. The goal of robot kinematic identification is to accurately estimate robot kinematic parameters from experimental data. A system for identifying robot kinematics is developed which applies a novel *virtual sensor approach* to parameter estimation. The system incorporates the uncertainty of experimental measurements through the introduction of computationally simple variance matrix approximations. A novel line-fitting procedure which requires no additional sensors or accurately machined components is devised for experimentally comparing the parameter estimates with the robot design parameters. Experimental data is acquired from and the kinematic parameters are estimated for an Adept II robot manipulator using a commercially available visual three-dimensional position sensor.

1. INTRODUCTION

1.1 Parameter Estimation

Robot kinematic identification, like many engineering identification applications, requires the accurate estimation of parameters from sets of experimental data. In general, the parameters can be computed by a *simple, determined* mapping from a *small* set of data (e.g., the estimation of a plane from the measured values of three points on the plane). However, the accuracy of such parameter estimates is limited by the accuracy of the individual measurements. To increase the accuracy of parameter estimates, engineers typically acquire a *large* set of data and compute the parameter estimates by a *complex, overdetermined* algorithm. The parameter estimates thus obtained can be more accurate than the individual measurements, but this approach must be tailored to a specific application and often involves linearization.

A *generally applicable* parameter estimation framework is introduced in this paper based upon the *virtual sensor approach* which combines the conceptual simplicity of the small data set approach with the numerical accuracy of the large data set approach. The estimation algorithm computes the *unbiased minimum variance estimate* of any desired parameter (whether linearly or nonlinearly related to the measurements) without linearization. Intuitive, computationally-simple approximations to variance matrices are introduced and implemented to tractably incorporate measurement uncertainties into the estimation process and thereby speed parameter convergence with reduced data sets. The estimation framework is applied to kinematic parameter identification of the Adept II robot manipulator, but has a multitude of robotics applications.

The literature on parameter estimation is found in such diverse areas as probability, statistics, identification, stochastic systems, and signal processing. This paper treats parameter estimation, first, from a general viewpoint and second, as it is applied to robot kinematic identification. Because of the proliferation of parameter estimation literature and the application described in this paper, only the robot kinematics literature is considered.

1.2 Robot Kinematics Identification

The specific goal of robot kinematic parameter identification is to estimate the parameters which interrelate the positions, velocities and accelerations of the robot joints and end-effector. I refer the reader to two recent survey papers [4,5] in the area of robot kinematic identification (also referred to as robot calibration) for the details of previous works. According to the classifications given in [5], the robot identification presented here is a level 2 robot calibration; i.e., all of the kinematic parameters but none of the dynamic parameters are identified.

The virtual sensor approach is distinct from previous approaches in that linearization is not required and the uncertainties of the data are advantageously utilized in the estimation algorithm. This study also differs from most previous approaches in that a commercial visual position sensor is employed. The only other known robot kinematic identification work utilizing this sensor is documented in [3]. The advantages of the present application of the sensor are expounded upon in Sections 6 and 7. The approach of this paper bears similarities to that described in [7] in that it is founded upon the identification of lines and circles traced by the robot end-effector in space. The theoretical and implementational details of the virtual sensor approach are, however, more advanced.

1.3 Overview of the Paper

The general parameter estimation problem is the topic of Section 2. The problem is defined formally and the virtual sensor solution is detailed. In Section 3, the prototypical problems of estimating three-dimensional lines and circles from sets of point data are described in terms of the virtual sensor approach. In Section 4, the robot kinematic parameter identification problem is reduced to estimating three-dimensional lines and circles from point data. A novel method of comparing the relative accuracies of estimated parameters with the robot design parameters is detailed in Section 5. Then in Section 6, the robot kinematic identification procedure is applied to the Adept II manipulator. The summarized results and conclusions are presented in Section 7.

2. VIRTUAL SENSOR MINIMUM VARIANCE ESTIMATION

2.1 The Virtual Sensor Approach

The general parameter estimation problem is one of estimating a $(t \times 1)$ parameter vector[1] $\mathbf{f}$ from a set $Q=\{\mathbf{m}_i: i=1...n\}$ of $(m \times 1)$

* This research has been supported by the U.S. Department of Energy at Sandia National Laboratories under Contract DE-AC04-76DP00789. The author would like to thank Larry Ray of Sandia National Laboratories for many helpful discussions during the course of this work.

[1] In this paper, scalars are denoted by lower-case plainface type (e.g., a), vectors by lower-case boldface type (e.g., **a**), matrices by upper-case boldface type (e.g., **A**), and sets are denoted by upper-case italix type (e.g., *A*).

measurement vectors $\mathbf{m}_i$. The measurement vectors are the actual outputs of physical sensors, and n is the number of measurements. To implement the virtual sensor approach, it is assumed that a data-to-parameter mapping $\mathbf{f}=\mathbf{g}(Q_s)$ from any s-measurement subset Q_s of the set of measurements Q to the parameter vector $\mathbf{f}$ is known. The number of measurements s involved in this mapping is the minimum number of measurements required to determine the parameter vector. For example, the minimum number of point measurements required to determine a plane is three (s=3), and the mapping from the coordinates of three points to the three characteristic parameters of a plane is the solution of three simultaneous linear equations.

The virtual sensor approach breaks the estimation problem into two independent steps as shown in Figure 1. In the first step, the parameter vector (this is termed a virtual sensor measurement) and its variance matrix are computed according to the data-to-parameter mapping. In the second step, a minimum variance parameter vector estimate is computed from all of the virtual sensor measurements. *Virtual sensor* measurements are so named because the combination of *physical sensors* and the *data-to-parameter mapping* computations produce a measurement of the desired parameter vector, which to the user appears as if there were a *virtual sensor* measuring the parameter vector directly. A standard minimum variance estimation algorithm is applied in the second step resulting in provably unbiased estimates.

The two independent steps of the virtual sensor approach are formally stated as follows:

Step 1: Compute a set of virtual sensor pairs $\{(\mathbf{f}_{vj}, \Sigma_{vj})$: j=1...r$\}$, where $\mathbf{f}_{vj}$ is a (t × 1) virtual sensor measurement and $\Sigma_{vj}=\mathbf{Var}(\mathbf{f}_{vj})$ is the corresponding (t × t) virtual sensor variance, for all subsets Q_s of the original set of measurements; and

Step 2: Compute the (t × 1) minimum variance parameter estimate $\mathbf{f}_e$ from the r virtual sensor pairs.

Step 1 is detailed in Section 2.2 and Step 2 in Section 2.3.

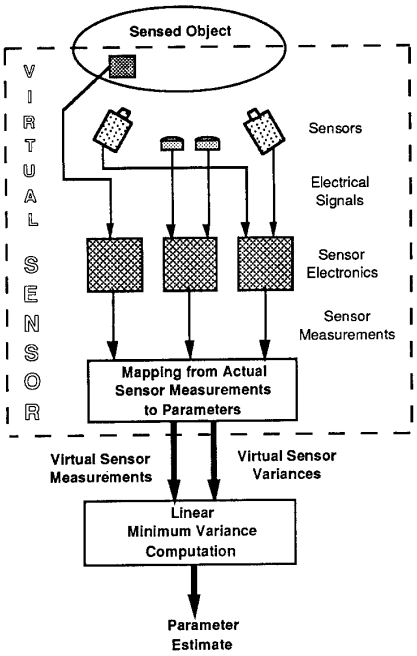


Figure 1: The Virtual Sensor Concept

2.2 Virtual Sensor Computations

The first step in the virtual sensor approach is discussed in this section. To compute the virtual sensor measurements one must first group the actual set of measurements into subsets. The number r_{max} of unique subsets Q_{sj} possible is equal to the number of permutations of the n physical sensor measurements taken s at a time:

$$r_{max} = \binom{n}{s} = \frac{n!}{s!(n-s)!} . \qquad (1)$$

In practice, the actual number r of virtual sensor measurements computed is chosen to be significantly less than r_{max} to lower computation time. Random and windowing methods for forming measurement subsets will be described in Section 6.

Once the subsets of measurements Q_{sj} for j=1...r have been formed, the virtual sensor measurements $\mathbf{f}_{vj}$ are computed directly according to the mapping:

$$\mathbf{f}_{vj} = \mathbf{g}(Q_{sj}) \qquad \text{for } j=1...r. \qquad (2)$$

To compute the variance of the virtual sensor measurements in (2), one may write the mapping in linear form[1] about the point Q_{sj}:

$$\mathbf{f}_{vj} = \sum_{k=1}^{s} \mathbf{A}_{kj}\mathbf{m}_{kj} + \mathbf{b}_j \qquad \text{for } j=1...r \qquad (3)$$

where the subset Q_{sj} consists of the measurement vectors $\mathbf{m}_{kj}$ for k=1...s, and the constant (t × m) matrix $\mathbf{A}_{kj}$ and (t × 1) vector $\mathbf{b}_i$ are:

$$\mathbf{A}_{kj} = \left(\frac{\delta \mathbf{g}(Q_{sj})}{\delta \mathbf{m}_{kj}} \Big| Q_{sj} \right); \quad \text{and} \qquad (4)$$

$$\mathbf{b}_j = -\sum_{k=1}^{s} \left(\frac{\delta \mathbf{g}(Q_{sj})}{\delta \mathbf{m}_{kj}} \Big| Q_{sj} \right) \mathbf{m}_{kj} + (\mathbf{f}_{vj} | Q_{sj}) . \qquad (5)$$

The variance of the virtual sensor measurement follows from (3) [1]:

$$\Sigma_{vj} = \sum_{k=1}^{s} \left(\mathbf{A}_{kj} \Sigma_{kj} \mathbf{A}_{kj}^T \right) \qquad (6)$$

where $\Sigma_{kj}=\mathbf{Var}(\mathbf{m}_{kj})$ is the variance matrix of measurement $\mathbf{m}_{kj}$. In Sections 3 and 4, it will be shown that for the robot kinematic parameters of interest, computationally simple approximations to the virtual sensor variances have been identified which reduce the amount of required computations significantly. And so, (3-6) need not be computed for the robot kinematic identification application. These variance computations are included here for completeness.

2.3 Minimum Variance Computations

To realize Step 2 of the virtual sensor approach, a linear estimate of $\mathbf{f}$ computed from the virtual sensor measurements is desired because of its conceptual simplicity [6]; i.e.:

$$\mathbf{f}_e = \sum_{j=1}^{r} \left(\mathbf{W}_j \mathbf{f}_{vj} \right) \qquad (7)$$

where the (t × t) weights $\mathbf{W}_j$ for j=1...r must be chosen to minimize the variance of the estimate $\Sigma_e=\mathbf{Var}(\mathbf{f}_e)$:

$$\Sigma_e = \sum_{j=1}^{r} \left(\mathbf{W}_j \Sigma_{vj} \mathbf{W}_j^T \right) . \qquad (8)$$

Further, a matrix unity sum constraint on the weights must be satisfied to ensure that we obtain an *unbiased* estimate [2]:

$$\sum_{j=1}^{r} \mathbf{W}_j = \mathbf{I}_t \qquad (9)$$

[1] by computing the first two terms of the Taylor series expansion

where $\mathbf{I}_t$ is the (t x t) identity matrix. We apply the method of Lagrange multipliers to solve the problem by minimizing the scalar objective function J:

$$J = tr\left(\sum_{j=1}^{r}\left(\mathbf{W}_j \Sigma_{vj} \mathbf{W}_j^T\right) + \lambda\left(\mathbf{I}_t - \sum_{j=1}^{r}\mathbf{W}_j\right)\right) \qquad (10)$$

where *tr* is the matrix trace operator. Upon differentiating (10) with respect to the weight $\mathbf{W}_j$ and equating to zero, we obtain

$$\frac{\delta J}{\delta \mathbf{W}_j} = \mathbf{W}_j \Sigma_{vj} + \mathbf{W}_j^T \Sigma_{vj} - \lambda \mathbf{I}_t = \mathbf{0}_t \qquad (11)$$

The matrix $\mathbf{0}_t$ is the (t x t) null matrix. We show below that $\mathbf{W}_j$ is symmetric. Thus, solving (11) for $\mathbf{W}_j$ gives:

$$\mathbf{W}_j = \frac{\lambda}{2}\Sigma_{vj}^{-1} . \qquad (12)$$

Because a variance matrix is positive definite by construction, the matrix inverse Σ_{vj}^{-1} in (12) exists. Equation (12) must hold for all j, and the matrix unity sum constraint in (9) can be written as:

$$\sum_{j=1}^{r}\left(\frac{\lambda}{2}\Sigma_{vj}^{-1}\right) = \mathbf{I}_t . \qquad (13)$$

Solving (13) for $\lambda\mathbf{I}_t$ yields:

$$\lambda\mathbf{I}_t = 2\left(\sum_{j=1}^{r}\Sigma_{vj}^{-1}\right)^{-1} . \qquad (14)$$

We now substitute (14) into (12) to compute the minimum variance weights:

$$\mathbf{W}_j = \left(\sum_{j=1}^{r}\Sigma_{vj}^{-1}\right)^{-1}\Sigma_{vj}^{-1} \quad \text{for } j=1\ldots r \qquad (15)$$

The variance matrices Σ_{vj} are symmetric by definition and by inspection of (15) the weights $\mathbf{W}_j$ are also symmetric. Substituting (15) into (7) yields the desired unbiased linear minimum variance estimator of **f**:

$$\mathbf{f}_e = \left(\sum_{j=1}^{r}\Sigma_{vj}^{-1}\right)^{-1}\left(\sum_{j=1}^{r}\left(\Sigma_{vj}^{-1}\mathbf{f}_{vj}\right)\right) \qquad (16)$$

This estimator can also be shown to be identical to the maximum likelihood estimator of the parameter vector assuming Gaussian virtual sensor distributions. The estimator in (16) can be easily written in recursive form so that the parameter estimate can be updated by with each new virtual sensor measurement. The recursive form is applied in Section 6 to view the convergence of the parameter estimate as the number of virtual sensor readings is increased. The virtual sensor methodology is completely defined within this section except for the data-to-parameter mapping requirements in Step 1. In the next section, mappings are developed for two prototypical estimation problems, that of identifying a three-dimensional line and circle from point measurements. The line and circle mappings are the only mappings required for the robot kinematic identification described in Section 4.

3. LINE AND CIRCLE MAPPINGS

3.1 Three-Dimensional Line Mapping

In this section, the mappings required in Step 1 of the virtual sensor approach for three-dimensional line estimation are developed. A three-dimensional line can be represented by a (3 x 1) point **e** and a (3 x 1) direction **d**. The line being defined by the equation:

$$\mathbf{y} = \mathbf{d}\,\xi + \mathbf{e} . \qquad (17)$$

In (17), **y** takes on all points on the line as ξ is varied from $-\infty$ to ∞. Separate mappings are required for the point parameter and the direction parameter.

The mapping from point measurements $\mathbf{m}_i$ to the point parameter **e** is one-to-one:

$$\mathbf{e} = \mathbf{m}_i . \qquad (18)$$

Thus only one measurement is required for the mapping (i.e., s=1). According to (18), the variance of the point parameter is simply the variance of the measurement.

$$\Sigma_e = \Sigma_i \qquad (19)$$

And so, the virtual sensor pair for the point parameter $(\mathbf{e},\Sigma_e)$ is defined by (18) and (19) as required for Step 1.

Two *unique* points $\mathbf{m}_i$ and $\mathbf{m}_j$ define a direction vector (i.e., s=2). The mapping for computing the direction vector is the difference of the points:

$$\mathbf{d} = \mathbf{m}_i - \mathbf{m}_j . \qquad (20)$$

The direction parameter variance can be computed as in (6). However, an intuitive approximation to the direction variance which is inversely related to the distance between the two points is introduced:

$$\Sigma_d = \frac{\mathbf{I}_2}{\|\mathbf{m}_i - \mathbf{m}_j\|} \qquad (21)$$

where $\|\cdot\|$ denotes the norm of the vector. The appropriateness of this approximation can be visualized. As the distance between two points increases, the line defined by them varies less and less for a constant variation in the position of either of the points. Thus, the variance of the direction decreases as the distance between the points increases. The virtual sensor pair for the direction parameter $(\mathbf{d},\Sigma_d)$ is defined by (20) and (21) as required for Step 1.

Equations (18-19) and (20-21) are the required mappings from one and two point measurements, respectively, to the desired point and direction parameters and may thus be applied to compute the virtual sensor measurements of a three-dimensional line. These mapping and variance computations are applied in Section 4 to identify the kinematics of a *translational* robot joint.

3.2 Three-Dimensional Circle Mapping

In this section, the mappings required in Step 1 of the virtual sensor approach for three-dimensional circle estimation are developed. A three-dimensional circle can be represented by a (3 x 1) axis **h**, a (3 x 1) center **c**, and a scalar radius u. The circle being represented by the two simultaneous equations:

$$\mathbf{y}^T\mathbf{h} = 1 \quad \text{and} \qquad (22)$$

$$\|\mathbf{y} - \mathbf{c}\| = u \qquad (23)$$

where **y** is any point on the circle. Equation (22) specifies that **y** must lie in the plane of the circle. Equation (23) specifies that the distance between any point on the circle and the circle center is the circle radius. Mappings and variance approximations for the axis and center parameters are developed in this section. The radius parameter is not required for the robot kinematic identification application described in Section 4.

The axis parameter is a vector normal to the plane of the circle. Therefore, the mapping for the axis parameter is the same as the mapping for the direction parameter of the plane of the circle. Three *non-colinear* points uniquely define a plane (i.e., s=3). as specified in (22). To compute the mapping from three point measurements to the axis vector **h** consider the three points: $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. Applying (22) to each of these points and combining equations yields:

$$\begin{pmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T\end{pmatrix}\mathbf{h} = \begin{pmatrix}1 & 1 & 1\end{pmatrix}^T \qquad (24)$$

For the non-colinear case, (24) can be solved for the axis vector:

$$\mathbf{h} = \begin{pmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T\end{pmatrix}^{-T}\begin{pmatrix}1 & 1 & 1\end{pmatrix}^T \qquad (25)$$

The axis variance can be computed by the variance expression in (6). However, an intuitive, computationally-simple approximation to the axis variance may be introduced as the inverse of the area of the triangle defined by the three points:

$$\Sigma_h = \frac{I_3}{\sqrt{\alpha\,(\alpha-\|\mathbf{p}_1-\mathbf{p}_2\|)\,(\alpha-\|\mathbf{p}_2-\mathbf{p}_3\|)\,(\alpha-\|\mathbf{p}_3-\mathbf{p}_1\|)}} \qquad (26)$$

where $\alpha = (1/2)(\|\mathbf{p}_1-\mathbf{p}_2\|+\|\mathbf{p}_2-\mathbf{p}_3\|+\|\mathbf{p}_3-\mathbf{p}_1\|)$.

By considering many different sets of three points, the reader will be able to convince himself that the variance of the axis defined by the points is inversely correlated with the area of the triangle defined by the three points. As the area of the triangle increases, the axis variations due to point variations decreases. And so, the virtual sensor pair for the circle axis parameter $(\mathbf{h},\Sigma_h)$ is defined by (25) and (26) as required for Step 1.

The circle center mapping can be obtained by applying an observation about a three-dimensional circle:

The plane normal to and bisecting the line segment between any two points on a circle includes the center of the circle.

From three unique points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ on the circle, the circle center can then be expressed as the intersection of three planes: the plane of the points, the bisecting plane for points $\mathbf{p}_1$ and $\mathbf{p}_2$, and the bisecting plane for points $\mathbf{p}_2$ and $\mathbf{p}_3$ as shown in Figure 2. (The bisecting plane for $\mathbf{p}_3$ and $\mathbf{p}_1$ could also have been applied.)

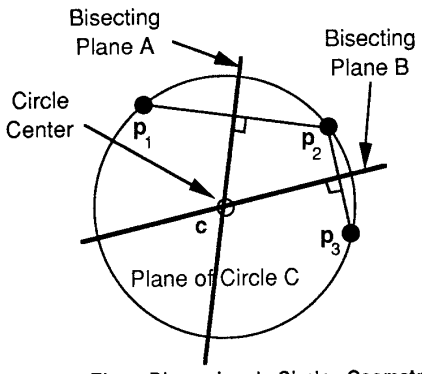


Figure 2: Three-Dimensional Circle Geometry

The plane bisecting points 1 and 2 (Plane A in Figure 2) can be represented as

$$\mathbf{y}^T\mathbf{v}_a = k_a \qquad (27)$$

where $\mathbf{v}_a$ is a vector normal to the plane and k_a is a nonzero scalar constant. Since, the line segment between points 1 and 2 is normal to Plane A, the direction between points 1 and 2 is the vector $\mathbf{v}_a$. The point mid-way between points 1 and 2 is a point on Plane A. Thus, to compute k_a, the vector $\mathbf{v}_a=(\mathbf{p}_1-\mathbf{p}_2)$ and point $\mathbf{y}=(\mathbf{p}_1+\mathbf{p}_2)/2$ are substituted into (27)

$$k_a = (\mathbf{p}_1+\mathbf{p}_2)^T(\mathbf{p}_1-\mathbf{p}_2)/2 \quad . \qquad (28)$$

The plane bisecting points 2 and 3 (Plane B in Figure 2) is computed analogously:

$$\mathbf{y}^T\mathbf{v}_b = k_b \qquad (29)$$

where $\mathbf{v}_b=(\mathbf{p}_2-\mathbf{p}_3)$ and $k_b=(\mathbf{p}_2+\mathbf{p}_3)^T(\mathbf{p}_2-\mathbf{p}_3)/2$. The plane of the three points (Plane C in Figure 2) is computed according to the mapping developed in (22) and (25). The circle center $\mathbf{c}$ is thus the point which satisfies equations (22), (27) and (29) simultaneously:

$$\mathbf{c}^T\begin{pmatrix}\mathbf{h} & \mathbf{v}_a & \mathbf{v}_b\end{pmatrix} = \begin{pmatrix}1 & k_a & k_b\end{pmatrix} . \qquad (30)$$

The mapping from any three unique points on a three-dimensional circle to the center vector is then

$$\mathbf{c} = \begin{pmatrix}\mathbf{h} & \mathbf{v}_a & \mathbf{v}_b\end{pmatrix}^{-T}\begin{pmatrix}1 & k_a & k_b\end{pmatrix}^T \qquad (31)$$

The same approximation is applied for the center variance as is applied for the axis variance:

$$\Sigma_c = \Sigma_h \qquad (32)$$

As the area of the triangle defined by the points increases, the axis variance and center variance decreases. And so, the virtual sensor pair for the circle center $(\mathbf{c},\Sigma_c)$ is defined by (31) and (32) as required for Step 1. These mappings and variance computations are applied in Section 4 to identify the kinematics of a *rotational* robot joint.

4. APPLICATION TO ROBOT KINEMATIC IDENTIFICATION

The development of the previous sections is now applied to identify robot kinematics. The position of a target point on the robot end-effector is measured as each robot joint is moved independently. If the joint is translational, the target point will traverse a line in space. If the joint is rotational, the target point will traverse a circle in space. The line and circle mappings in Sections 3.1 and 3.2, respectively are applied directly to compute the joint axis and a point on the axis. The coordinate system for the joint axis is then located such that the z-axis is aligned with the joint axis, and the point on the axis coincides with the origin of the joint coordinate system. These correspondences can be shown in the homogeneous transformation matrix from the sensor coordinate system S to joint 1 coordinate system.

$${}^S\mathbf{T}_1 = (\mathbf{n}\ \mathbf{o}\ \mathbf{a}\ \mathbf{p}) = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (33)$$

The homogeneous transformation matrix ${}^S\mathbf{T}_1$ denotes the position orientation of the joint 1 coordinate system relative to the sensor coordinate system. The following correspondences can be made:

$$\mathbf{a} = \begin{cases} \dfrac{\mathbf{d}}{\|\mathbf{d}\|} \text{ for a translational joint} \\ \dfrac{\mathbf{h}}{\|\mathbf{h}\|} \text{ for a rotational joint} \end{cases} \qquad (34)$$

$$\mathbf{p} = \begin{cases} \mathbf{e} \text{ for a translational joint} \\ \mathbf{c} \text{ for a rotational joint} \end{cases} \qquad (35)$$

The position of the joint coordinate system is thus completely determined and its orientation is constrained to a rotation about the z-axis. To compute the remaining elements of the homogeneous transformation matrix, the Euler angles form of a transformation matrix is employed [5]:

$$\mathbf{T} = \begin{pmatrix} c\varphi c\theta c\psi - s\varphi s\psi & -c\varphi c\theta s\psi - s\varphi c\psi & c\varphi s\theta & dx \\ s\varphi c\theta c\psi + c\varphi s\psi & -s\varphi c\theta s\psi + c\varphi c\psi & s\varphi s\theta & dy \\ -s\theta c\psi & s\theta s\psi & c\theta & dz \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (36)$$

where c denotes the trigonometric cosine function, s denotes the trigonometric sin function and φ, θ, ψ are rotations about the z, y and z-axes, respectively. By equating (33) and (36), the remaining elements of the homogeneous transformation matrix may be solved in terms of a rotation ψ about the z-axis:

$$\mathbf{n} = \frac{1}{a_{xy}}\begin{pmatrix} a_x a_z c\psi - a_y s\psi \\ a_y a_z c\psi + a_x s\psi \\ -c\psi\, a_{xy}^2 \end{pmatrix} \qquad (37)$$

$$o = \frac{1}{a_{xy}} \begin{pmatrix} -a_x a_z s\psi - a_y c\psi \\ a_y a_z s\psi + a_x c\psi \\ s\psi a_{xy}^2 \end{pmatrix}$$

where $a_{xy}=\sqrt{a_x^2+a_y^2}$.

The free variable ψ representing a rotation about the z-axis can be chosen to align the shaft encoder zero position of the robot with the positive x-axis of the joint coordinate system. The shaft encoder measurements will thereby correspond directly to angles with respect to the joint coordinate system.

The result of the previous computations is the homogeneous transformation matrix from the sensor frame to the joint coordinate system. The procedure is repeated for all joints. However, the kinematics of a robot consist of the homogeneous transformation matrices *from each joint coordinate system to the next*. These robot homogeneous transformation matrices can be computed from a pair of sensor-to-joint transformation matrices as:

$$^1T_2 = {}^ST_1^{-1}\ {}^ST_2 \qquad (38)$$

Once (38) has been computed for all robot joints, the position and orientation of the base coordinate frame of the robot may be computed. The simplest method for doing this is to define the origin of the reference coordinate system for the robot workspace to be where the end effector coordinate system is located when all joints are set to their zero position. The position and orientation of the base of the four-axis Adept II robot $\mathbf{T}_1$ is then computed as:

$$\mathbf{T}_1 = {}^4T_E^{-1}\ {}^3T_4^{-1}\ {}^2T_3^{-1}\ {}^1T_2^{-1} \qquad (39)$$

The position and orientation of the four-axis Adept II robot end-effector in space $\mathbf{T}_E$ at any time is then computed as:

$$\mathbf{T}_E = \mathbf{T}_1\ \Phi_1\ {}^1T_2\ \Phi_2\ {}^2T_3\ \Phi_3\ {}^3T_4\ \Phi_4\ {}^4T_E \qquad (40)$$

where the Φ matrices represent joint translation x or rotation θ and have the form:

$$\Phi = \begin{pmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{for a rotational joint}$$

$$\Phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{for a translational joint} \qquad (41)$$

5. EVALUATION OF THE ESTIMATES

Once kinematic identification is completed, a method is desired for determining the accuracy of the estimates. If the exact parameter variances are applied throughout the development, we may apply the Chebyshev inequality [6] to arrive at a probability that the parameters are in error by any specified amount. However, when applying approximations to the parameter variances, the Chebyshev inequality does not necessarily apply. A novel experimental method for comparing the relative accuracies of two sets of robot parameters has been conceived to compare the relative accuracy of the estimated robot parameters to the robot design parameters obtained from the manufacturer.

The experimental method requires that a thin wire be pulled taught and fixed within the robot workspace. A taught wire defines a very accurate line in space. The robot end-effector is then manually positioned at several points along the wire and the robot joint angles are recorded at each one. It is noted that these measurements will each be in error by some unknown amount because of the manual positioning. These manual positioning errors will not be biased towards one set of parameters over another. That is, for a large number of points the manual positioning errors will adversely effect the results of any set of parameters equally. Each parameter set is then applied to compute the end-effector position in (40) for each recorded point along the wire. The resulting set of end-effector points generated with a particular parameter set should follow a line in space except for manual positioning errors and parameter errors. Since the manual positioning errors are uncorrelated with either set of parameters, their effects average out. The accuracy with which each parameter set defines a line is then a measure of the parameter accuracies. And the parameter set which most closely defines a line in space (e.g., in the least-squares sense) is the more accurate parameter set. We can thereby determine whether the identified robot parameters are more or less accurate than the robot design parameters.

6. ADEPT II KINEMATIC IDENTIFICATION

6.1 Data Collection and Estimation

The kinematics of the Adept II robot is experimentally identified by applying the virtual sensor framework described in Sections 2 to 4. Point data are collected using a Watsmart visual position sensor [8]. The sensor consists of two cameras which view the robot, an LED mounted on the end effector, and processing electronics which compute the three-dimensional coordinates of the LED by triangulation. Each joint of the robot is moved through a range of motion while three-dimensional position measurements of the LED are collected. The four-axis Adept II was programmed to sequentially move each joint through a range of motion beginning with the last joint and working towards the robot base. Joints 1, 2, and 4 were repeatedly moved through ranges of 53°, 50° and 170°, respectively. The translational joint 3 and rotational joint 4 physically share the same axis; thus, the joint 4 data is applied to identify both axes. One thousand point measurements are taken at a rate of 20 measurements per second while each joint is in motion. The entire data collection process takes only 5 minutes. An attempt was made to move each joint manually with robot power off and all but one joint brake energized to avoid any robot motor electrical noise from effecting the position sensor readings and to avoid any servo errors on non-moving joints from effecting the data. But this method of data collection produced no better results and required more manual operations.

Some implementational details are noteworthy. The sensor calibration grid (a black cubic structure with 40 LEDs at calibrated positions) was applied before collecting data and a 0.88mm calibration error was achieved. The visual position sensor responds to sunlight, so the windows to the laboratory are covered with black cloth. Overhead florescent lights cause no problems. The robot workspace is kept free of extraneous objects which can reflect the light of the LEDs causing irregular measurements. The surface of the visible robot area is also covered by black cloth to avoid reflections. The sensor has the capability of viewing several LEDs at once. However, it was found that if two or more LEDs were used with the robot motors enabled, the LEDs interfered with each other producing inaccurate position readings. The motions of the joints are keep to within the volume of the sensor calibration grid to ensure accuracy of the measurements.

The virtual sensor parameter estimation computations are programed within Mathematica (a symbolic mathematics package) running on a Macintosh II personal computer. Windowing and random selection of data subsets are used to compute the virtual sensor measurements. For the windowing case, points are chosen from the data set at equal intervals. Many window sizes were tested, but convergence of the parameters was slow in all cases. A random selection of points was also tried. The random sampling in general provided good convergence of the parameters. But, infrequent sets of bad data points (e.g., a set of three points which happened to be measured with the robot at the same position) would throw off the convergence. To remove these bad sets of data points, a criteria

was applied to the data before computation of the virtual sensor measurements. If the area of the triangle defined by three points for a rotational joint (or the distance between two points for a translational joint) is less than a manually selected threshold, the data set was not accepted for computation of virtual sensor measurements. This method is successful for eliminating bad data sets and enables the position parameters typically to converge to within 0.5mm with 300 virtual sensor measurements. For the results presented below, 2000 virtual sensor measurements are computed to ensure convergence.

6.2 Evaluation of Results

According to the method of Section 4, the homogeneous transformation matrices for the Adept II robot were identified. The accuracy of the estimated parameters (i.e., the elements of the transformation matrices) relative to the manufacturer's design parameters is determined by the line-fitting procedure introduced in Section 5. The design parameters for the Adept II robot are obtained from the robot manufacturer. The results using the design parameters and the estimated parameters are plotted in Figures 3, and 4, respectively. For the design parameter results in Figure 3, the largest deviation from a line is approximately 2mm; whereas, the estimated parameter results in Figure 4 show a maximum 9mm deviation. It is found that the parameter estimates produce good results but are not as accurate as the design parameters. One can visually ascertain that the points generated by the design

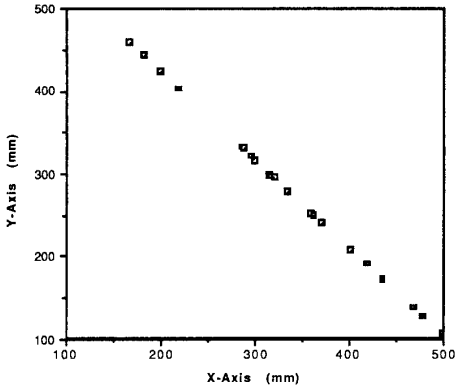


Figure 3: Line-Fitting Results Using the Design Parameters

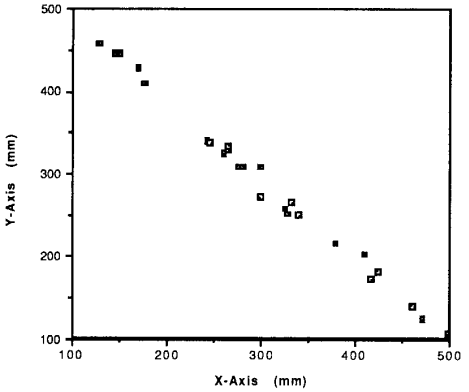


Figure 4: Line-Fitting Results using the Estimated Parameters

parameters more closely define a line than does the points generated by the estimated parameters. These results verify the results of [3] in which the same sensor was applied to identify the kinematics of the MIT Serial Link Direct Drive Arm using a least squares parameter identification technique. Because the virtual sensor estimation approach is provably unbiased and the estimates converge, it is concluded that the position sensor is the source of the inaccuracies.

7. CONCLUSIONS

This work contributes to the robot parameter identification literature [4,5] by virtue of the following features. The virtual sensor approach is a theoretically founded, general-purpose method of *including the uncertainty of sensor measurements* into the parameter identification process. This is an advance upon previous studies which have not formally included uncertainty in the parameter identification process. The approach is *simple* requiring a single LED and *no a-priori kinematic information* (in contrast to the 6 LEDs at known relative positions and the nominal kinematic parameters required in [3]). Novel, *computationally inexpensive approximations to parameter variances* are conceived and applied to eliminate the computationally expensive, theoretical variance computations. Implementational procedures beyond those presented in [3] for experimental collection of useful visual position data have been reported. The virtual sensor framework is equally *applicable to linear and nonlinear estimation problems*, in contrast to the majority of existing estimation methods which linearize nonlinear problems for implementation. Additionally, a novel line-fitting procedure which requires no additional sensors or accurately machined components is presented for experimentally comparing the parameter estimation results with the robot design parameters.

The results suggest that the virtual sensor parameter estimation approach combined with the visual position sensor can be advantageously applied to perform kinematic identification for robots which are constructed less accurately than the Adept II. To obtain kinematic parameter estimates which will improve the controlled performance of the Adept II requires a more accurate sensor. An area of future work towards enhancing the virtual sensor methodology is the development of windowing techniques (methods for choosing measurement subsets) which optimize parameter convergence.

8. BIBLIOGRAPHY

[1] W.H. Beyer, (editor), *CRC Standard Mathematical Tables*, CRC Press, Inc., Boca Raton, FL, 1981.

[2] W.B. Davenport, *Probability and Random Processes*, McGraw-Hill Book Company, New York, NY, 1970.

[3] J. M. Hollerbach and D. J. Bennett, "Automatic Kinematic Calibration Using a Motion Tracking System," *Robotics Research: The Fourth International Symposium*, edited by R. Bolles and B. Roth, MIT Press, Cambridge, MA, 1988, pp. 191-198.

[4] J. M. Hollerbach, "A Survey of Kinematic Calibration," in *The Robotics Review 1988*, edited by O. Khatib, J.J. Craig, and T. Lozano-Perez, MIT Press, Cambridge, MA, 1988, pp.207-242.

[5] Z. S. Roth, B. W. Mooring, and B. Ravani, "An Overview of Robot Calibration," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 5, October 1987, pp. 377-385.

[6] M.R. Spiegel, *Probability and Statistics*, McGraw-Hill Book Company, New York, NY, 1975.

[7] H.W. Stone, A.C. Sanderson and C.P.Neuman, "Arm Signature Identification," Proceedings of the 1986 IEEE Conference on Robotics and Automation, San Francisco, April 1986, pp. 41-48.

[8] *Watsmart Manual, Version 2.7*, Northern Digital Inc., 403 Albert Street, Waterloo, Ontario, Canada N2L3V2, 1989.