ROBOT CALIBRATION WITHOUT SCALING

A Thesis

by

THOMAS W. IVES

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 1995

Major Subject: Mechanical Engineering

ROBOT CALIBRATION WITHOUT SCALING

A Thesis

by

THOMAS W. IVES

Submitted to Texas A&M University
in Partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Louis J. Everett (Chair of Committee)	Jeff Trinkle (Member)
Harry Hogan (Member)	George P. Peterson (Head of Department)

May 1995

Major Subject: Mechanical Engineering

ABSTRACT

Robot Calibration without Scaling. (May 1995)
Thomas W. Ives, B.S., The University of Texas at Austin
Chair of Advisory Committee: Dr. Louis J. Everett

Many engineering problems are studied through numerical schemes that use large matrices to mathematically describe the systems being analyzed. Some of the problems require iterative algorithms. In many cases the matrices used in these numerical schemes are found to be "ill-conditioned", or close to singularity. The amount of ill-conditioning is measured with a condition number. Ill-conditioned matrices significantly slow, and usually prevent, convergence toward a solution when using iterative methods. Scaling is a common way of improving the condition number for a matrix. Researchers in other fields have developed specific methods of scaling matrices to improve the condition number. However, robotics researchers have not specifically addressed, in robotics literature, their approaches to improving the condition of matrices that are used in kinematic parameter identification (calibration).

A new calibration method was developed that uses matrices with significantly better condition numbers than matrices used in standard calibration methods, and scaling of the kinematic parameters describing the robot is not required. This method was tested on rotational joint robots only. The key feature of this new method is that it takes advantage of the rotational kinematic parameters' independence from the translational kinematic parameters. This feature is utilized as a constraint that simplifies the identification technique. The method is similar to standard methods in that it is iterative. There are two steps to each iteration. In the first step, the translational parameters are solved separately in a closed form solution. In the second step, the rotational parameters are estimated using a new type of Jacobian matrix. This new Jacobian has a condition number that is significantly better than those derived from standard methods. Furthermore, the new method is easy to understand and implement.

ACKNOWLEDGEMENTS

I would first like to thank my wife for her patience with, and contribution to, my higher education. She has allowed and enabled me to pursue one of the greater joys in my life, and I often feel that I go off to play while she is left home to do the more important work for our family. It is to her that I dedicate this work. I love you Sue.

Second, I feel fortunate to have worked under Dr. Louis Everett. I hope to emulate many of his qualities and strengths in the years to come. I am especially grateful to him for his patience with the many delays that this work encountered.

Third, I would like to thank Dr. Hogan and Dr. Trinkle for being on my committee.

Fourth, I would like to thank Michael J. Wienen for the great care he took in reviewing the progress of this thesis.

And last and greatest, I thank you God that you have given us such a world as this to explore. Thank You for putting it here for us to enjoy. It is with great anticipation that I look forward to the next one You are preparing for us.

TABLE OF CONTENTS

Page

ABSTRACT....................

ACKNOWLEDGEMENTS....................

TABLE OF CONTENTS....................

LIST OF FIGURES....................

1. INTRODUCTION....................1

2. THE CALIBRATION PROCESS....................4

 2.1 Modeling....................4
 2.2 Measurement....................5
 2.3 Identification....................6
 2.4 Implementation....................9

3. PROBLEM DEFINITION....................10

 3.1 Unconstrained and Constrained Minimization Defined....................10
 3.2 Unconstrained Minimization in Kinematic Identification....................10
 3.3 Overview of Problem....................12

4. THE NEW CALIBRATION PROCEDURE....................13

 4.1 Calculation of Translational Parameters....................14
 4.2 Identification of Rotational Parameters....................15
 4.3 Overview of the New Procedure....................16

5. RESULTS....................17

 5.1 Calibration of a Theoretical Robot....................17
 5.2 Calibration of a Robot Using Real Measurements....................18

6. CONCLUSIONS....................20

REFERENCES....................21

APPENDIX A T MATRICE DESCRIPTIONS....................22

APPENDIX B INPUT FILES....................23

 B.1 Description of Input Files....................23
 B.2 GMF Robot Input File....................25
 B.3 PUMA Robot Input File....................26

APPENDIX C MEASUREMENTS USED FOR PUMA ROBOT....................27

VITA....................31

LIST OF FIGURES

Page

Figure 1. Pictorial Representations of a Nominal Kinematic Model and an Actual Physical Model of a Robot 3

Figure 2. Comparison of Standard and New Methods Using Real Measurement Data on a Puma Robot 18

Figure 3. GMF SR-110 Robot 25

Figure 4. PUMA 560 Robot 26

1. INTRODUCTION

When robots first appeared, they seemed to be very promising devices in the realm of automated manufacturing. They could be programmed to perform a variety of tasks autonomously. They could replace expensive hired labor to perform repetitive or potentially dangerous work. They could work for longer time periods with increased production speed and production quality. Their main disadvantage was the considerable amount of time it took to program them. The robot would have to be manually taught several points for a new task. Manual teaching is when a human controls the robot's tool position. When positioned, the point is stored with other points in a program written for the task. This would not seem to be a disadvantage compared to the long hours of fast reliable production the robot gives in return.

However, after many hours have been invested in teaching data points for the long, detailed programs, a robot's accuracy eventually declines. There are two general causes for this loss in accuracy. The first general cause is small changes in the robot due to wear. These small changes result in deviations from the taught positions. Replacing the worn parts with new parts does not correct this problem, because there are also slight differences between the same parts when they are new. The second general cause is movement of pieces within the robot's workspace. Examples of this include movement of tool racks, or movement of fixtures that hold the parts being worked on by the robot. Once these things have been moved with respect to the robot, the robot's programmed points have been invalidated by the amount of movement. This is most simply illustrated by a blind person who memorizes the placement of furniture in a room. If someone comes in and rearranges the furniture, he begins tripping over it and must re-learn the location of each item. One would think that this problem could be corrected by shifting the origins of the workspace items in the robot's programs, but experience has taught that this rarely corrects the problem completely. Too many positions still have serious errors, because the equations in the robot controller describing the robot's kinematics do not predict the new positions correctly.

The classical controls engineering fix to such problems is to add feedback. In the realm of automated manufacturing, robot's operate without feedback control; they are open loop devices. In order for open loop control to have exact accuracy, the model of the system must be known exactly. Thus, the robot must be manufactured exactly as specified, or the exact model must be determined after manufacturing is complete. However, even when using the tightest manufacturing tolerances, the robot will still have positioning errors. These errors are due to small manufacturing errors that are inevitable, and it is impossible to formulate an exact kinematic model for the robot with current techniques.

The style and format of this thesis document follow that modeled by *IEEE Transactions on Robotics and Automation.*

There appear to be two possible approaches to the solution of this problem. The first possible approach is to implement feedback control. In a manufacturing environment, it is very difficult to implement feedback control, because the position sensing circuitry on the end manipulator of the robot would interfere with the robot's work. The second possible approach is to attempt open loop control by formulating an adequate kinematic model. Once this model is in hand, determine the parameters of that model. This second approach is now commonly known as robot calibration. While this study works specifically with robots, the calibration methods presented herein can be used to identify the kinematic parameters of any machine.

A manufacturer typically designs its robot's linkages to be orthogonal to the robot's base frame when all the joint angles are at their zero reference positions. This is illustrated by the robot in figure 1 that has coordinate frames with unit vectors x_i, y_i, and z_i. The set of kinematic equations used to describe this robot will be referred to as the nominal model (NM). The actual robot after manufacturing (which is exaggerated for clarity) might have coordinate frames such as those shown with unit vectors x'_i, y'_i, and z'_i. The kinematic equations used to describe the actual robot represented by x'_i, y'_i, and z'_i will be referred to as the actual physical model (APM) of the robot. (This model is never known exactly.) Unfortunately, the model usually programmed into the controller to calculate robot positions is the one represented by x_i, y_i, and z_i. (The reason for this is simply ignorance. The manufacturer knows of no better model to use, and they do not know how to find an adequate kinematic model after manufacturing is complete). Consider one difference between the NM and APM to see how positioning errors occur. Since the nominal model does not account for the true orientation of joint 1, the controller is ignorant of the plane in which the tool point moves as joint 1 is operated. Therefore, it will never be able to position the tool point correctly from calculated data; it must be shown (taught) the desired position. Some of the other possible differences between the two models compound the errors further: actual base position, length and direction of link 1, orientation of joint 2, the length and direction of the tool, and the orientation of the tool point.

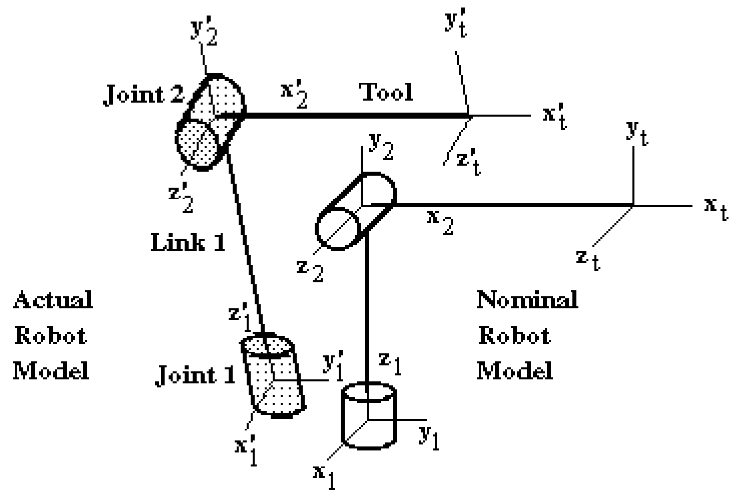

Figure 1. Pictorial Representations of a Nominal Kinematic Model
and an Actual Physical Model of a Robot

As stated above, this thesis looks at kinematic calibration of robots. More specifically, it deals with a specific problem encountered by engineers performing kinematic calibration. When performing the iterative numerical analysis used to determine the values of the kinematic parameters, the solution does not typically converge without scaling a certain key matrix that will be named later. This work describes a new calibration procedure that eliminates the need to scale any of the matrices used in the analysis, and this new method is simple to implement.

2. THE CALIBRATION PROCESS

Mooring, Roth and Driels [1], explain the calibration strategy as follows. Consider the development of a relationship between joint space and task space that has the following functional form.

$$\mathbf{P} = f(\eta, \theta) \tag{2.1}$$

P is the predicted pose[1] of the robot's tool in the task space, θ is a vector of the joint values, and η is a set of constant elements. The elements of η are placed in the formulation to describe the robot's deviation from the originally intended geometry, (described by the NM), to the actual physical geometry, (described by the APM).

The goal of calibration is to make the predicted positions of the robot match the actual positions of the robot for a given set of θ. The success of this goal depends upon the accomplishment of two tasks. The first task is to find the best formulation of equation 2.1. The second task is to accurately find the values for each element of η. The entire calibration process can be further divided into four distinct phases: modeling; measurement; parameter identification; and implementation.

2.1 Modeling

Modeling is the determination of an adequate functional form for equation 2.1. Everett and Hsu [2] have shown for geometric models that there are a maximum of N elements, or parameters, in η.

$$N = 4R + 2P + 6 \tag{2.2}$$

where R is the number of rotational joints in the robot, and P is the number of prismatic joints in the robot. For each rotational joint, there are four constant parameters belonging to η that must be determined: two translational parameters and two rotational parameters. For each prismatic[2] joint, there are two rotational parameters to identify. The "6" represents the three rotational parameters and three translational parameters that describe the position of the tool point with respect to the last joint frame. The remainder of this thesis will be restricted to the formulation of rotational joint robots only.

[1] The pose of a robot is comprised of the orientation information and the 3-D coordinate position information of the end manipulator.

[2] A prismatic joint moves along a straight line in similar fashion to a telescoping antenna.

Furthermore, the N parameters of η will commonly be broken into two groups in section 4. N_R will represent the number of rotational parameters in η, and N_T will represent the number of translational parameters in η. The reason for this division will become clear in section 4.

The modeling technique used in this work divides the formulation of the model into individual T matrices. Appendix A shows the six basic T matrices used in the kinematic model formulation. Each T matrix can contain a single parameter of η, a single constant rotation in some multiple of 90°, a single constant translation, or a single variable joint rotation. The only T matrices considered to have unknown values in them are those that contain elements of η.

The T matrices containing constant rotations in multiples of 90° are not necessary for the development of the model. They are used only for clarity and will be referred to as orthogonal rotations. By using them to always orient the z-axis of the joint frame along the axis of the joint, the parameters of η can be arranged in like manner for each joint. This simplifies the design (and troubleshooting) of the formulation such that the modeler need only specify the correct orthogonal rotations between each joint. For illustration, this technique will be applied to the robot shown in figure 1 while using the restriction of equation 2.2.

A mathematical model (henceforth refered to as a model) for a robot is developed by ordering individual T matrices describing the robot's geometry from the base frame to the tool point frame. When these T matrices are multiplied together, the result will be a predicted pose of the robot's tool point with respect to the base frame of the robot. This is commonly called a forward solution in robotics. Any result of such a calculation in this thesis will be labeled T_t. A model of the robot shown in figure 1 is described by equation 2.3. This equation is shown in groups for clarity. The parameters of η are arranged in the manner described above. The first group in equation 2.3 contains the T matrices that contain the parameters of η for joint one. In the second group, the first T matrix contains joint one's rotation, and the remaining T matrices perform orthogonal rotations. The third group contains T matrices with parameters of η for joint two. The fourth group has one T matrix describing joint two's rotation. The last group has T matrices containing parameters of η for the tool, or end effector. Sometimes the tool rotation terms are not used in the development of the model. The reason for this will be discussed later. Note that in between each subset of η parameters, there is a set of T matrices containing joint displacement and possible orthogonal rotations. The manufacturer believes it has delivered a robot that is adequately modeled with the NM. Values for the parameters of η are sought by the modeler that will compensate for the manufacturing errors.

Once the values of η are accurately known, the pose of this robot can be predicted more accurately using equation 2.3. If this method were applied to a robot with more than two joints, the pattern would simply be extended to include extra terms for each additional joint.

$$T_t = (T_{R_{X1}} T_{R_{Y1}} T_{T_{X1}} T_{T_{Y1}}) \, (T_{R_{Z1}\theta_1} T_{R_{X1}90^\circ} T_{R_{Y1}90^\circ}) \, (T_{R_{X2}} T_{R_{Y2}} T_{T_{X2}} T_{T_{Y2}}) \, (T_{R_{Z2}\theta_2}) \, (T_{T_{Xt}} T_{T_{Yt}} T_{T_{Zt}} T_{R_{Xt}} T_{R_{Yt}} T_{R_{Zt}})_t \qquad \textbf{(2.3)}$$

2.2 Measurement

This phase of the calibration process has two sub-phases: joint displacement recording; and pose measurement recording. Joint displacements can be taken directly from the encoder readings of the robot. Pose measurements are obtained using an external measurement system such as a coordinate measurement machine (CMM). The robot is moved into a pose, and the joint displacements and pose measurements are taken. The number of pose measurements needed is determined by the number of unknown parameters, N. Usually, an experimenter will measure more than the required number of poses in order to average out the effects of measurement error. Measurement error is a limiting factor when trying to find the best values of η. The modified calibration technique used in this work will be analyzed for its effectiveness in averaging out measurement error.

2.3 Identification

2.3.1 With Tool Orientation Measurements

In order to make an estimation of the correct parameters of η, it is necessary to know how each element of the pose is affected by each element of η. The matrix classically used to describe these relational changes is known as the Jacobian (J). For a generic robot, J would have the form of equation 2.4. The first row contains changes in the x coordinate of the tool point with respect to the changes in each of the parameters of η. (This is the x coordinate with respect to the robot's base frame). In similar fashion, the next two rows contain changes in the y and z coordinates of the tool point. The last three rows contain rotational changes about the same x, y and z axes of a frame at the robot's tool point that has the same orientation as the robot's base frame.

$$J = \begin{bmatrix} \frac{\partial x}{\partial \eta_1} & \frac{\partial x}{\partial \eta_2} & \cdots & \frac{\partial x}{\partial \eta_i} & \cdots & \frac{\partial x}{\partial \eta_N} \\ \frac{\partial y}{\partial \eta_1} & \frac{\partial y}{\partial \eta_2} & \cdots & \frac{\partial y}{\partial \eta_i} & \cdots & \frac{\partial y}{\partial \eta_N} \\ \frac{\partial z}{\partial \eta_1} & \frac{\partial z}{\partial \eta_2} & \cdots & \frac{\partial z}{\partial \eta_i} & \cdots & \frac{\partial z}{\partial \eta_N} \\ \frac{\partial \phi_x}{\partial \eta_1} & \frac{\partial \phi_x}{\partial \eta_2} & \cdots & \frac{\partial \phi_x}{\partial \eta_i} & \cdots & \frac{\partial \phi_x}{\partial \eta_N} \\ \frac{\partial \phi_y}{\partial \eta_1} & \frac{\partial \phi_y}{\partial \eta_2} & \cdots & \frac{\partial \phi_y}{\partial \eta_i} & \cdots & \frac{\partial \phi_y}{\partial \eta_N} \\ \frac{\partial \phi_z}{\partial \eta_1} & \frac{\partial \phi_z}{\partial \eta_2} & \cdots & \frac{\partial \phi_z}{\partial \eta_i} & \cdots & \frac{\partial \phi_z}{\partial \eta_N} \end{bmatrix} \qquad \textbf{(2.4)}$$

Initial estimates for the values of η are set to make the model match the NM, because, before the identification phase of the calibration is complete, there is no idea of what these values should be. Consider the robot of figure 1 to be in a pose with a given set of joint angles. If an error (E) between the measured pose and the calculated pose (the calculated pose is found using equation 2.3) can be determined, then J can be used to estimate the parameters of η with equation 2.5. The reason for saying that η is estimated rather than found will be explained below. The negative sign in equation 2.5 serves to direct the adjustments of $\Delta\eta$ in the correct direction.

$$\begin{bmatrix} dx \\ dy \\ dz \\ \delta\phi x \\ \delta\phi y \\ \delta\phi z \end{bmatrix} = - \begin{bmatrix} \frac{\partial x}{\partial \eta_1} & \frac{\partial x}{\partial \eta_2} & \cdots & \frac{\partial x}{\partial \eta_i} & \cdots & \frac{\partial x}{\partial \eta_N} \\ \frac{\partial y}{\partial \eta_1} & \frac{\partial y}{\partial \eta_2} & \cdots & \frac{\partial y}{\partial \eta_i} & \cdots & \frac{\partial y}{\partial \eta_N} \\ \frac{\partial z}{\partial \eta_1} & \frac{\partial z}{\partial \eta_2} & \cdots & \frac{\partial z}{\partial \eta_i} & \cdots & \frac{\partial z}{\partial \eta_N} \\ \frac{\partial \phi_x}{\partial \eta_1} & \frac{\partial \phi_x}{\partial \eta_2} & \cdots & \frac{\partial \phi_x}{\partial \eta_i} & \cdots & \frac{\partial \phi_x}{\partial \eta_N} \\ \frac{\partial \phi_y}{\partial \eta_1} & \frac{\partial \phi_y}{\partial \eta_2} & \cdots & \frac{\partial \phi_y}{\partial \eta_i} & \cdots & \frac{\partial \phi_y}{\partial \eta_N} \\ \frac{\partial \phi_z}{\partial \eta_1} & \frac{\partial \phi_z}{\partial \eta_2} & \cdots & \frac{\partial \phi_z}{\partial \eta_i} & \cdots & \frac{\partial \phi_z}{\partial \eta_N} \end{bmatrix} \begin{bmatrix} \Delta\eta_1 \\ \Delta\eta_2 \\ \vdots \\ \Delta\eta_i \\ \vdots \\ \Delta\eta_N \end{bmatrix} \text{, or} \quad E = -J\Delta\eta \qquad \textbf{(2.5)}$$

The challenge at this point is finding E, and J. The detailed treatment of this development is explained by Paul [3]. The determination of E is considered first. As the measured pose (Tm) approaches the calculated pose (Tc), the operation performed in equation 2.6 approaches the form of the T matrix shown in equation 2.6. This partially explains why equation 2.5 provides an estimate of η. The rotational terms obtained from the result of equation 2.6 only provide estimated values of the error. As Tc approaches Tm, this estimate becomes more accurate. The rotational errors for the separate x, y and z contributions are respectively represented by each of the terms of $\delta\phi$. The elements of the E vector are extracted from the results of equation 2.6 and arranged as shown.

$$Tc^{-1}(Tm - Tc) = Tc^{-1}Tm - I \cong \begin{bmatrix} 0 & -\delta\phi z & \delta\phi y & dx \\ \delta\phi z & 0 & -\delta\phi x & dy \\ -\delta\phi y & \delta\phi x & 0 & dz \\ 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and } \quad E = \begin{bmatrix} dx \\ dy \\ dz \\ \delta\phi x \\ \delta\phi y \\ \delta\phi z \end{bmatrix} \tag{2.6}$$

Now consider the determination of J. Let each of the T matrices in equation 2.3 be referred to as T_i. For clarity in the following discussion, a transformation matrix U_{i+1} will be created. This U_{i+1} transformation describes the tool location with respect to frame i for i **= 1** to **n**. It is calculated as shown in equation 2.7 where n is the number of T matrices in the model.

$$U_{i+1} = T_{i+1}T_{i+2}\cdots T_n \tag{2.7}$$

In order to construct J, ${\delta T_t}/{\delta\eta_i}$ must be found for each parameter of η. Recall that T_t is the result of the forward solution of the model. This calculation can be performed according to equation 2.8. The columns of each J can then be built by extracting elements from the result of equation 2.9.

$$\delta T_t / \delta\eta_i = \sum_{i=1}^{n} U_{i+1}^{-1}\, \delta T_i / \delta\eta_i \, U_{i+1} \tag{2.8}$$

$$\left(\delta T_t/\delta\eta_i\right)^{-1} Tm - I = \begin{bmatrix} 0 & -d\phi z/d\eta_i & d\phi y/d\eta_i & dx/d\eta_i \\ d\phi z/d\eta_i & 0 & -d\phi x/d\eta_i & dy/d\eta_i \\ -d\phi y/d\eta_i & d\phi x/d\eta_i & 0 & dz/d\eta_i \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{2.9}$$

This is a linear approximation for J at the current values of η. This completes the explanation of why equation 2.5 provides an estimate of η. The solution process for η using equation 2.5 will become clearer from the following discussions.

Currently, the Jacobian has dimension $6 \times N_{R,T}$, where $N_{R,T}$ represents the number of unknown rotational and translational terms in η, and 6 represents the number of elements in the robot's pose. To have the right number of equations for the given number of unknowns in equation 2.5, the row dimension of the Jacobian must increase. This can only be done by including more poses. Therefore, the row dimension of the Jacobian will increase to $6 \times M$ where **M** represents the number of measured poses. Each $6 \times N_{R,T}$ Jacobian will represent a single robot pose. The combination of Jacobians would be as shown in equation 2.10, and each J_i and E_i are of the forms in equation 2.5 where **i = 1** to **M**. If the resulting matrices in equation 2.10 are not square, the least squares method is used [4].

$$\begin{bmatrix} E_1 \\ \vdots \\ E_i \\ \vdots \\ E_M \end{bmatrix} = - \begin{bmatrix} J_1 \\ \vdots \\ J_i \\ \vdots \\ J_M \end{bmatrix} \Delta\eta \tag{2.10}$$

Now estimates can be made for the elements of η through the solution of equation 2.10 using the estimates for each J_i and E_i. Once the changes in η ($\Delta\eta$) are determined from solving equation 2.10, the new values of η are updated to η + $\Delta\eta$. The process is repeated until $\Delta\eta$ and the magnitude of E approach zero (within the desired tolerance), or until the solution has converged to the lowest possible error. However, sometimes convergence is not achieved due to the problems that will be discussed in section 3.

2.3.2 Without Tool Orientation Measurements

The process of solving for η changes only slightly when it is undesirable to collect tool orientation measurements. If tool orientation data is not collected, the parameters of η associated with the tool orientation (or end manipulator orientation) cannot be determined. The reason is that these three parameters are independent from all other frames in the robot, and they can only be determined from orientation measurements of the tool point. Therefore, the number of identifiable terms of η reduces to

$$N = 4R + 2P + 3 \qquad \textbf{(2.11)}$$

Also, equation 2.4 is reduced to equation 2.12, and equations 2.5 and 2.10 change accordingly.

$$J = \begin{bmatrix} \frac{\partial x}{\partial \eta_1} & \frac{\partial x}{\partial \eta_2} & \cdots & \frac{\partial x}{\partial \eta_i} & \cdots & \frac{\partial x}{\partial \eta_N} \\ \frac{\partial y}{\partial \eta_1} & \frac{\partial y}{\partial \eta_2} & \cdots & \frac{\partial y}{\partial \eta_i} & \cdots & \frac{\partial y}{\partial \eta_N} \\ \frac{\partial z}{\partial \eta_1} & \frac{\partial z}{\partial \eta_2} & \cdots & \frac{\partial z}{\partial \eta_i} & \cdots & \frac{\partial z}{\partial \eta_N} \end{bmatrix}. \qquad \textbf{(2.12)}$$

Orientation parameters of the tool point are omitted when they are deemed non critical. Times when tool orientation data is most critical are when orientation of the tool point is critical. Drilling deep holes would be an example of when the tool orientation is critical. The degree to which tool orientation is important must ultimately be determined by the robot user.

2.4 Implementation

One must understand the limits of the improved model obtained from calibration. The new model may only make good predictions over a small subset of the robot's entire workspace. Even with the limitations that exist, there are creative methods of implementation for the improved model found by calibration. A comprehensive review of these methods is beyond the focus of this work. However, the reader is encouraged to look at one valuable implementation technique used to safeguard large amounts of manually taught data. This technique is described by Everett and Ives [5].

3. PROBLEM DEFINITION

3.1 Unconstrained and Constrained Minimization Defined

The identification procedure described in section 2.3 is a form of *unconstrained minimization*. For ease of reference it will be called the standard method for the remainder of this thesis. During each iteration on the solution of equation 2.10, all the elements of η are *unconstrained* (they are all allowed to change simultaneously) while iterating to *minimize* the errors between the calculated and measured poses. If a constraint is applied to the analysis, the problem can be divided into two steps. During the first step, the constraint is applied, and it reduces the number of unknowns. The constraint used in this thesis reduces the number of η's values being sought to η's rotational parameters. Then, once the rotational parameters of η are found, the constraint is used to determine the remainder of η's parameters. This is an example of *constrained minimization*.

3.2 Unconstrained Minimization in Kinematic Identification

3.2.1 Previous Problems with Unconstrained Minimization

3.2.1.1 Problems with Convergence

Since there is no general all purpose optimization or minimization algorithm that can work for any numerical analysis problem, some ingenuity is required on the part of the researcher [6] when he is performing his particular analysis. The algorithms developed for kinematic identification have been proven successful, but experimenters have consistently encountered one problem in their identification analysis schemes. This problem has been discussed at conferences amongst robotics researchers, but it has not (up to this point) been addressed in the robotics literature [7]-[9]. It is a problem often encountered by those performing numerical solutions of matrices in general [10],[11]. The problem is that convergence toward a solution requires scaling the Jacobian matrix when solving equation 2.10. One simple example is the calibration of a PUMA 560 robot. Researchers at Texas A&M University found that convergence can be achieved simply by converting the original translational parameters from millimeters to inches. While this is nice and convenient for this particular case, it does not prove successful in general. There is no “foolproof” guideline that defines the correct amount of scaling for every numerical problem, and Forsythe, Malcolm and Moler [10] advise that scaling not be used unless necessary, because scaling can affect the accuracy of the computed solution through the resulting changes in the pivots and interchanges used during Gaussian elimination. Furthermore, once scaled, the Gaussian elimination technique *will* produce small residuals, but it will not necessarily produce small errors. The condition number (discussed in section 3.2.2) describes the relation between the error and the residual [10],[11].

3.2.1.2 Parameters of Different Units

Equation 2.4a is repeated here as equation 3.1 for convenience. Consider the terms in equation 3.1. There will be a collection of four groups of units: 1) length per length; 2) length per angle; 3) angle per length; and 4) angle per angle. The magnitude difference between these terms can be significant, especially as robots with large degrees of freedom are analyzed and more terms are multiplied together. When the Jacobians are combined as in equation 2.10, and the resulting Jacobian is rectangular, applying the least squares method will cause the magnitudes to deviate even further. The final result is a Jacobian that is closer to being singular. There is a method of determining how near the Jacobian is to being singular that will be discussed in the next section.

$$J = \begin{bmatrix} \frac{\partial x}{\partial \eta_1} & \frac{\partial x}{\partial \eta_2} & \cdots & \frac{\partial x}{\partial \eta_i} & \cdots & \frac{\partial x}{\partial \eta_N} \\ \frac{\partial y}{\partial \eta_1} & \frac{\partial y}{\partial \eta_2} & \cdots & \frac{\partial y}{\partial \eta_i} & \cdots & \frac{\partial y}{\partial \eta_N} \\ \frac{\partial z}{\partial \eta_1} & \frac{\partial z}{\partial \eta_2} & \cdots & \frac{\partial z}{\partial \eta_i} & \cdots & \frac{\partial z}{\partial \eta_N} \\ \frac{\partial \phi_x}{\partial \eta_1} & \frac{\partial \phi_x}{\partial \eta_2} & \cdots & \frac{\partial \phi_x}{\partial \eta_i} & \cdots & \frac{\partial \phi_x}{\partial \eta_N} \\ \frac{\partial \phi_y}{\partial \eta_1} & \frac{\partial \phi_y}{\partial \eta_2} & \cdots & \frac{\partial \phi_y}{\partial \eta_i} & \cdots & \frac{\partial \phi_y}{\partial \eta_N} \\ \frac{\partial \phi_z}{\partial \eta_1} & \frac{\partial \phi_z}{\partial \eta_2} & \cdots & \frac{\partial \phi_z}{\partial \eta_i} & \cdots & \frac{\partial \phi_z}{\partial \eta_N} \end{bmatrix} \quad \textbf{(3.1)}$$

Researchers at Texas A&M University have typically scaled the translational parameters by the sum of all translational parameters in the robot geometry to ensure that the solution of equation 2.10 converges. This has the effect of reducing the large differences in magnitude between the terms in equation 3.1. Though the author has never used a model that would not converge when this type of scaling was used, this topic remains relevant as explained below.

3.2.2 Condition Number

The condition number is explained in detail by Forsythe, Malcolm and Moler [10], and by Cook [11]. In short, the reciprocal of the condition number provides a measure of “nearness to singularity.” The condition number can be defined in different ways [10], and should therefore be thought of as a relative measure. For this thesis, the condition number of a matrix $\mathbf{A}$ is defined as

$$\mathrm{cond}([\mathrm{A}]) = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{3.2}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of the matrix. The condition numbers of Jacobians used in this study were noted to significantly improve when the translational parameters were scaled as described above. These results are compared to the condition numbers obtained from the new calibration method in section 5.

3.3 Overview of Problem

The fortunate point for the standard method is that the solutions in robot calibrations are driven by the current model’s error from actual measurements. There is never any guess about whether one has reached the best possible solution for the model being used. Furthermore, researchers have found that there is one unique solution to a model [12]. Therefore, the question arises, why focus on this problem if convergence is achieved through current scaling techniques? This work does not present theory that provides determination of the correct amount of scaling. It presents a new method by which scaling becomes unnecessary. Furthermore, this new method uses a different approach that may be simpler to implement.

4. THE NEW CALIBRATION PROCEDURE

Consider a robot that has been modeled as described in section 2. Furthermore, consider that the T matrices are left in symbolic form. The symbolic solution of T_t for this robot model would contain elements having the functional forms shown in equation 4.1. The limits of the sumations and products are not shown, because they will depend upon the particular robot. For the following discussion, it is only important to note the general functional form. Each r_i is defined to be a rotational element of the robot being acted upon by a trigonometric function of sine or cosine. These rotational elements include rotational parameters of η, joint angles and orthogonal rotations. Each t_i is a translational parameter of η which could be translations of linkages between joints, base translations, and tool translations.

$$T_t = \begin{bmatrix} \sum_j \left(\pi_i r_i\right)_j & \sum_j \left(\pi_i r_i\right)_j & \sum_j \left(\pi_i r_i\right)_j & \sum_k \left(\sum_j \left(\pi_i r_i\right)_j\right) t_k \\ \sum_j \left(\pi_i r_i\right)_j & \sum_j \left(\pi_i r_i\right)_j & \sum_j \left(\pi_i r_i\right)_j & \sum_k \left(\sum_j \left(\pi_i r_i\right)_j\right) t_k \\ \sum_j \left(\pi_i r_i\right)_j & \sum_j \left(\pi_i r_i\right)_j & \sum_j \left(\pi_i r_i\right)_j & \sum_k \left(\sum_j \left(\pi_i r_i\right)_j\right) t_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(4.1)}$$

The upper left 3x3 partition of T_t contains equations that describe the orientation of the robot's tool, or end manipulator, with respect to its base. These equations contain rotational elements only. The upper right 3x1 partition of T_t contains equations describing the x, y, and z position of the tool with respect to the base. Each of these equations has products containing both rotational and translational terms, and each of these products includes one translational parameter of η. Therefore, the translational terms of η can be factored out of these equations, and this will result in an equation of the following form.

$$[R][t]=[P], \text{ or } \begin{bmatrix} R_{11} & \cdots & R_{1i} & \cdots & R_{1n} \\ R_{21} & \cdots & R_{2i} & \cdots & R_{2n} \\ R_{31} & \cdots & R_{3i} & \cdots & R_{3n} \end{bmatrix} \begin{bmatrix} t_1 \\ \vdots \\ t_i \\ \vdots \\ t_n \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.2}$$

The translational terms of η are represented by lower case t's to keep the reader from confusing them with T matrices. The equations with the rotational parameters in them are represented by upper case R's to help the reader remember that the elements of this matrix are equations comprised of trigonometric functions, sine and cosine, acting on the rotational parameters (the rotational parameters themselves are still represented by lower case r's). The results of these three equations are the x, y and z positions of the robot's tool point (end manipulator) and will be denoted as P.

From looking at equation 4.2, one can see that the rotational parameters of the robot are independent from the translational parameters of the robot[1]. Therefore, in the new procedure, the parameters of η are estimated in two steps. In the first step, the translational parameters of η are estimated using a closed form solution of an augmented form of equation 4.2. During this closed form solution, the rotational parameters are considered to be correct. In the second step, the parameters of η describing rotations are estimated using a procedure similar to the standard calibration method, but an augmented version of equation 4.2 is used as a constraint. During this second step, the affects of changes in η's rotational parameters on the x, y, and z position of the robot's tool point are analyzed. This analysis includes the effects of the changes in η's rotational parameters on η's translational parameters and will be discussed in section 4.2. This two step procedure is repeated until the desired or best possible convergence is achieved.

4.1 Calculation of Translational Parameters

The R matrix from equation 4.2 has dimensions $3 \times N_T$. However, all of the measurements can be used in the solution for the translational parameters by applying the

[1]Regardless of the solution method used, it is true in kinematics that the translational parameters are dependent upon the rotational parameters. The method shown in section 2.3 does not take advantage of this. The current procedure does.

least squares method [4]. Each measurement will have a corresponding $[R]_i$ and $[P]_i$ as shown in equation 4.2, and there will be **M** measurements. Several of these will be augmented together into the form of equation 4.3. The large matrix comprised of the $[R]_i$'s will be referred to as R_B for the remainder of this thesis. The large vector comprised of $[P]_i$'s will be referred to as P_B. It is important to note that the solution for t using equation 4.3 is only as good as the current estimates for the rotational parameters of η. This is understandable when one considers that this method is developed using the premise that the translational parameters are dependent upon the rotational parameters.

$$\begin{bmatrix} [R]_1 \\ \vdots \\ [R]_i \\ \vdots \\ [R]_M \end{bmatrix} \begin{bmatrix} t_1 \\ \vdots \\ t_i \\ \vdots \\ t_{N_T} \end{bmatrix} = \begin{bmatrix} [P]_1 \\ \vdots \\ [P]_i \\ \vdots \\ [P]_M \end{bmatrix} \text{, or } R_B t = P_B \tag{4.3}$$

4.2 Identification of Rotational Parameters

The new method for determining the rotational parameters of η is similar to the standard method of determining all of η 's parameters. The error vector for this method will be referred to as the new error vector (E_N). The Jacobian will be called the pseudo Jacobian (J_P). The formulation of E_N and J_P is as follows. Consider a Taylor series about the calculated x, y and z positions of the end manipulator. The result is shown in equation 4.4 with the higher ordered terms dropped. In this equation, $P_{B,Meas}$ is the measured position and $P_{B,Calc}$ is the position calculated from equation 4.3 using the current estimates of the rotational parameters of η.

$$[P_{B,Calc}] = [P_{B,Meas}] + \begin{bmatrix} \frac{\partial P_B}{\partial r_1} & \cdots & \frac{\partial P_B}{\partial r_j} & \cdots & \frac{\partial P_B}{\partial r_{N_R}} \end{bmatrix} \Delta r + \cdots \tag{4.4}$$

It is desired to make changes to the rotational parameters of η (Δr) such that they minimize error between the calculated and measured poses. By rearrangement of equation 4.4, E_N becomes

$$E_N = [P_{B,Meas}] - [P_{B,Calc}] = J_P \Delta r \quad \textbf{(4.5)}$$

where

$$J_P = \left[\frac{\partial P_B}{\partial r_1} \quad \cdots \quad \frac{\partial P_B}{\partial r_j} \quad \cdots \quad \frac{\partial P_B}{\partial r_{N_R}} \right] \quad \textbf{(4.6)}$$

J_P shown in equation 4.6 is found as follows. The derivative of equation 4.3 with respect to each of η's rotational parameters is shown by equation 4.7. The solution of equation 4.7 for each r_j will result in one vector.

$$\frac{\partial P_B}{\partial r_j} = \left[\frac{\partial R_B}{\partial r_j} t + R_B \frac{\partial t}{\partial r_j} \right]_j \quad \textbf{(4.7)}$$

There will be N_R of these vectors. Once these vectors are augmented together in order from **j** = **1** to N_R, J_P is formed as shown in equation 4.8.

$$J_P = \left[\left[\frac{\partial R_B}{\partial r_1} t + R_B \frac{\partial t}{\partial r_1} \right]_1 \quad \cdots \quad \left[\frac{\partial R_B}{\partial r_j} t + R_B \frac{\partial t}{\partial r_j} \right]_j \quad \cdots \quad \left[\frac{\partial R_B}{\partial r_{N_R}} t + R_B \frac{\partial t}{\partial r_{N_R}} \right]_{N_R} \right] \quad \textbf{(4.8)}$$

J_P gives the estimated path that Δr must follow in order to minimize E_N. The solution of equation 4.5 is an estimate of Δr for each iteration. The rotational elements of η in the model are then adjusted by Δr.

The vector $\frac{\partial t}{\partial r_j}$ in equations 4.7 and 4.8 is a new term that must be determined. There will be N_R of these vectors, and each will have dimensions of $N_T \times 1$. In order to solve for these vectors, an approximation must be made. As Δr approaches zero, equation 4.7 approaches zero as well. In order to solve for $\frac{\partial t}{\partial r_j}$, the left side of equation 4.5 will be set to zero, and from this approximation equation 4.9 is formed. Once the $\frac{\partial t}{\partial r_j}$ are found, they are placed back into their respective places in J_P.

$$[R_B] \left[\frac{\partial t}{\partial r_j} \right] = - \left[\frac{\partial R_B}{\partial r_j} t \right] \quad \textbf{(4.9)}$$

One can see that J_P will have elements that are close in magnitude by looking at equation 4.8, because, rather than it being an unbalanced mixture of dimensional ratios as described in section 3 when using the standard method, each of its elements are comprised of the same dimensional ratios. This will be illustrated in section 5 by comparing the condition number of J_P to the condition number of J.

4.3 Overview of the New Procedure

There are two major steps in the new procedure for each iteration: 1) solve for the translational parameters of η; 2) solve for the rotational parameters of η. A substep of step 2, estimates each $\frac{\partial t}{\partial r_j}$. After step one, the model is updated with new estimates of η's translational parameters. After step 2, the model is updated with the new estimates of η's rotational parameters. The steps are repeated until the the error is minimized.

Note that the new method cannot be used to identify the three rotational parameters of the tool point. Consider equation 4.3. Tool point orientations are not collected for this equation. As stated previously, calibrations are often performed without identifying these parameters, because they are rarely required, and because the data is difficult to acquire. If desired, the standard method can be used to solve for them separately. Since only rotational parameters are being identified, the problems mentioned in section 3 will not occur. Furthermore, since only three parameters are being identified, the required number of measurements and calculations will be minimal.

5. RESULTS

Calibrations were performed using theoretical and real robot measurement data. The method for creating robot model input files for the code used in this work is described in Appendix B.1. Using this description, the reader will be able to understand the models used for the two robots in this study. These two models are in Appendices B.2 and B.3.

5.1 Calibration of a Theoretical Robot

Theoretical measurement data was generated by making the model match the NM from the manufacturer (i.e. η 's values were set to make the model match the manufacturer's NM). In a real test, this would be the NM, but it was treated as the APM for this test. Forward solutions were calculated using this model with several sets of joint angles to generate theoretical measurement data. Then, the APM was altered by changing η 's values by small random amounts. This model was then treated as the NM. A calibration was performed on this NM using the theoretical measurements and random joint position data. By the end of the calibration, the values of η for the model being treated as the NM should be adjusted back to what they were for the model being treated as the APM.

5.1.1 Without Measurement Error

The first tests were done without measurement error by using theoretical measurement data. This is extremely convenient, because the performance of the calibration methods can be tested without introducing this uncertainty. The robot model used for the theoretical studies was that of a GMF SR-110A. The standard and new calibration methods were applied to this model.

First, the standard method was used without scaling. The translational parameters were entered into the model in millimeters. Convergence could not be achieved. In this method, the Jacobian matrix was first made square by use of the least squares method. The resulting condition number of the modified Jacobian was 64869.6. The condition number reduced to 235.09 after scaling using the method described in section 3.2.1.

When the new calibration method was applied to this same model using the same theoretical measurements and the same joint angles, convergence was reached without any scaling, and the resulting condition number of J_P after the least squares method was applied was 55.12. The condition number of R_B after having least squares applied was 14.69. The models from each method were identical.

5.1.2 With Measurement Error

Artificial measurement error was introduced into the theoretical x, y, and z measurement data with variations in magnitude of approximately 2 tenths of a millimeter or less. The same modified data file was used for each calibration method. Models were determined from each method using this data. Then the performance of each model was tested by calculating forward solutions and comparing them to the forward solutions calculated using the exact model. The differences in the results between the two methods were negligible (e.g. 1E-9 or less). The condition numbers were the same as in section 5.1.1.

5.2 Calibration of a Robot Using Real Measurements

The experimental data used in this section was obtained by Dr. Chiang-Kuo Lei [13] in an independent study using a PUMA 560. The model for this PUMA is in Appendix B.3. The measurement device was a Mitutoya CX-D2 coordinate measuring machine with a claimed repeatability of 0.01mm and a claimed accuracy of 0.1mm. The data used from his study is contained in Appendix C. This data includes joint angles, and position data used for calibration and performance testing.

Both methods were applied to this robot using the data from Appendix C. Since the exact model for a real robot will never be known, it was not possible to compare the results in the same way as in section 5.1.2. In this case, the measured data was divided in half. The robot model was calibrated with both methods using the first half of the data (the joint angles and their corresponding poses). Then the performance of these two models was checked with the second half of the data.

The performance was measured as follows. Calculated poses were obtained from forward solutions using the models obtained from each method and the second set of joint angles. The calculated poses were then compared to the second half of measured poses. Comparisons of the error between these calculated and measured poses for each method are shown in figure 2. These errors are the magnitude of the upper right 3x1 partition (i.e.

the translation vector) in the T_t matrix found using equation 2.6.

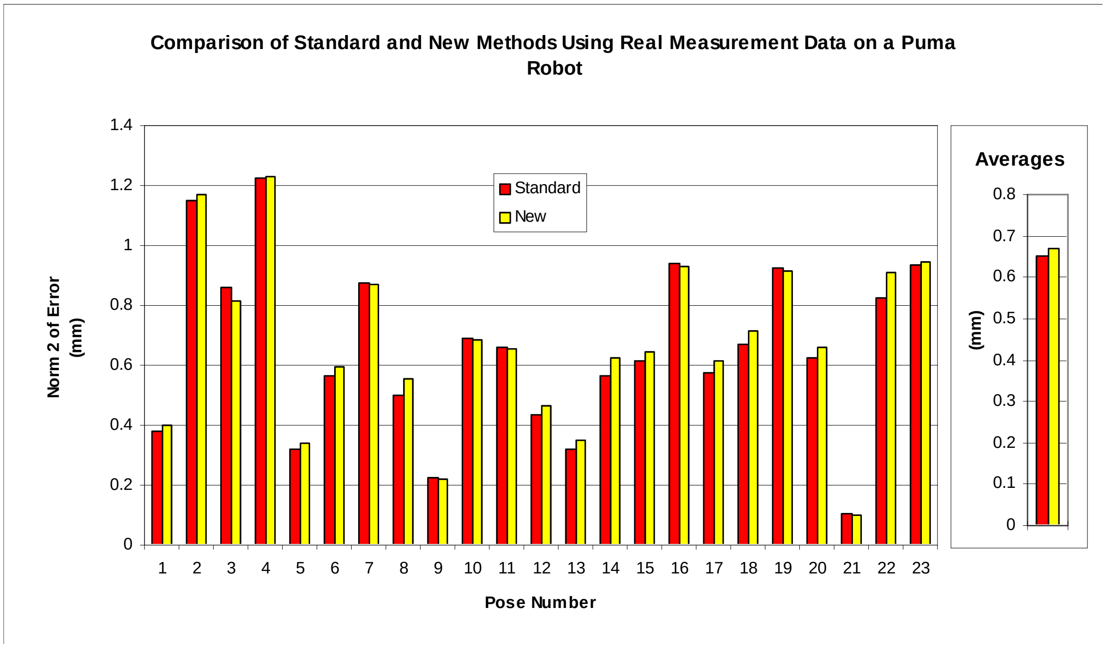

Figure 2. Comparison of Standard and New Methods Using Real Measurement Data on a Puma Robot

The condition number of **J** was 2.788e+9, the condition number of $\mathbf{J_P}$ was 1606, and the condition number of $\mathbf{R_B}$ was 8.

It is obvious from the differences in the predictions of the two models that measurement errors have slightly different effects on the two methods. These effects can be explained by considering the way the two methods determine the values of η's parameters. In the standard method, all of the parameters, rotational and translational, are estimated together. In the new method, the determination of the translational parameters is completely dependent on the accuracy of the rotational parameters. Furthermore, the error calculated for the standard method is influenced by rotational and translational effects; the calculated error in the new method is simply the difference between the calculated and measured [x y z] vectors. Thus, when measurement error is introduced, the kinematics identified by each method will be slightly different, because the kinematic adjustments made by each method are driven by different mechanisms. However, from looking at the results, these differences are slight.

The average error for the standard method is less than 0.02mm better than the average error for the new method. This difference is five times less than the CMM's accuracy, and it could not even be seen on a measurement. The difference between each calculated and measured pose was also well below the CMM accuracy. Thus, the question of which method handles measurement error best remains unanswered, but, for the CMM used, the differences would have been hidden.

Consider what would have happened if the models had been compared by an actual experiment. The robot would have been driven to several different poses. Each pose would have been measured, and the joint angles corresponding to each pose would have been recorded. During the measurement, some accuracy would be lost due to the accuracy of the CMM. Predicted poses for each set of joint angles would have been calculated using the model from each method, and the performance of each model would have been compared. However, the minimum resolution of the robot which is 0.1mm for the PUMA 560 would have caused an additional uncertainty between the measured and predicted poses. Thus, the differences between the two methods would have been more hidden.

6. CONCLUSIONS

A new identification method for robot calibration that does not require scaling of any kinematic parameters was applied using simulated measurements on a GMF SR-110A robot, and actual measurements on a PUMA 560 robot. The new method is straight-forward and easy to implement. Its main advantage is that it eliminates the need for scaling to improve condition numbers when performing the numerical identification analysis phase of robot calibration. Condition numbers of matrices developed in the new method are much better than matrices used by the standard method, because each element in the new method's Jacobian has the same dimensional ratio. The standard method could have from two to four different dimensional ratios. Comparisons between the two methods indicate that there is no significant difference in the accuracy of the models obtained from each method. However, the new procedure bypasses the scaling problems encountered when calibrating robots.

This work did not specifically investigate how variations in scaling affect the condition number of a matrix. However, finding a general relation for the optimal scale factor for kinematic identification problems would be a valuable topic for further research.

REFERENCES

[1] B. W. Mooring, Z. S. Roth, M. R. Driels, *Fundamentals of Manipulator Calibration.* New York, NY: John Wiley & Sons Inc., 1991.

[2] L. J. Everett, T. Hsu, "The Theory of Kinematic Parameter Identification for Industrial Robots," *Trans. ASME*, vol. 110, pp. 96-100, 1988.

[3] R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control.* Cambridge, MA: MIT Press, 1981.

[4] W. Kaplan, *Advanced Calculus.* Reading, MA: Addison-Wesley, 1984.

[5] L. J. Everett, T. W. Ives, "Automatic Maintenance of Robot Programs," *IEEE Trans. Robotics Automat.*, Accepted for Publication December 1994.

[6] D. G. Luenberger, *Optimization by Vector Space Methods*, New York, NY: John Wiley & Sons Inc., 1968.

[7] H. Zhuang, Z. S. Roth, "A Linear Solution to the Kinematic Parameter Identification of Robot Manipulators," *IEEE Trans. on Robotics Automat.*, Vol. 9, No. 2, pp. 721-731, 1993.

[8] K. Tchon', "Calibration of Manipulator Kinematics: A Singularity Theory Approach," *IEEE Trans. on Robotics Automat.*, Vol. 8, No. 5, pp. 671-678, 1992.

[9] C. A. Edwards II, R. L. Galloway Jr., "A Single-Point Calibration Technique for a Six Degree-of-Freedom Articulated Arm," *Int. J. of Robotics Res.*, Vol. 13, No. 3, pp. 189-198, 1994.

[10] G. E. Forsythe, M. A. Malcolm, C. B. Moler, *Computer Methods for Mathematical Computations.* Englewood Cliffs, NJ: Prentice-Hall, 1977.

[11] R. D. Cook, *Concepts and Applications of Finite Element Analysis.* New York, NY: John Wiley & Sons Inc., 1974.

[12] J. B. Bennett, *The Control of Human Arm Movement: Models and Mechanical Constraints.* Cambridge, MA: PhD. Dissertation, Dept. Brain and Cog. Sci., MIT, June 1990.

[13] C. K. Lei, *An Experimental Comparison of Robot Kinematic Calibration Methods.* PhD. Dissertation, Dept. Mech. Eng., Texas A&M Univ., May 1992.

APPENDIX A
T MATRICE DESCRIPTIONS

A T matrix is represented by a single T. The first level of subscripting indicates the type of T matrix. A rotational one would be T_R, and a translational one would be T_T. Each rotational T matrix represents a rotation about a single axis of a coordinate frame. Each translational T matrix represents a translation along a single axis of a coordinate frame. The second level of subscripting represents the single axis in question. Therefore, T_{R_x} represents a T matrix for a rotation about the x axis.

The forms of the individual T matrices used in this study are shown below.

$$T_{R_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_{T_x} = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{R_Y} = \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_{T_Y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{R_z} = \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_{T_z} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, consider equation 2.3 from the body of the thesis that is repeated below. When a $\theta_{\#}$ follows the subscripting, it means that this T matrix represents a joint movement. When a number follows the subscripting, then this means that the T matrix represents a constant motion of the amount specified.

$$T_t = (T_{R_{X1}} T_{R_{Y1}} T_{T_{X1}} T_{T_{Y1}})_1 \, (T_{R_{Z1}\theta_1} T_{R_{X1}90^\circ} T_{R_{Y1}90^\circ}) \, (T_{R_{X2}} T_{R_{Y2}} T_{T_{X2}} T_{T_{Y2}})_2 \, (T_{R_{Z2}\theta_2})$$

$$(T_{T_{Xt}} T_{T_{Yt}} T_{T_{Zt}} T_{R_{Xt}} T_{R_{Yt}} T_{R_{Zt}})_t$$

APPENDIX B

B.1 Description of Input Files

Consider the following model.

$$T_{R_X} T_{R_Y} T_{T_X} T_{T_Y} T_{R_Z\theta_1} T_{R_{X1}90^\circ}$$

The first line of the input file must contain a description of the type of measurements that will be supplied. For example, if the experimenter is supplying orientational and translational measurements, the first line contains all **a**'s as such

a,a,a,a,a,a

The first three **a**'s represent the "availability" of orientation measurements about the x, y and z axes. The second three represent the translational measurements along the x, y and z axes. Therefore, if any measurement is not available, that **a** should be replaced with an **n**. If the orientational measurements were not available, the first line would be

n,n,n,a,a,a.

Each T matrix in the robot model is represented by two lines in the input file for the code that was used. The first line describes the nature of the T matrix. The line that follows it contains the value for the single variable contained in that T matrix.

The next two lines of the input file represent the first T matrix, and they are

R, C, I, X
0.0

This means that the first T matrix in the model is a rotation by a constant amount that needs to be identified, and the rotation is about the X axis. Thus, the **R** is for rotation, the **C** means it is a constant rotation, the **I** means that this parameter needs to be identified, and the **X** means that the rotation is about the X axis. Finally, the initial guess for the value of the rotation is 0°. The next two lines in the model would therefore be

R, C, I, Y
0.0

For the constant translations along the X and Y axes, the inputs would then be

T,C,I,X
0.0
T,C,I,Y
0.0

Currently, the above four values are initially set to zero. They can be set to whatever a good initial guess would be for the model under consideration.

The next T matrix encountered in the model represents the joint rotation. The line inputs for this T matrix would be

R,A,K,Z
0.0

The **A** stands for active meaning that the value is dynamic. The **K** stands for known meaning the joint angle is a known value. The next two lines represent two orthogonal transformations that are made for ease of modeling as described in the body of the thesis.

R,C,K,X
90.0
R,C,K,Y
90.0

The rest of the models representation through input lines can be understood from the explanations above. If comments are desired, they can be inserted into the input file by starting them with an exclamation (!) mark. This way the program will disregard the entire contents of the line following the ! mark.

B.2 GMF Robot Input File

```
! Standard Param Model
for GMF
n,n,n,a,a,a
! joint one
r,c,i,x
1.0
r,c,i,y
-1.0
t,c,i,x
2.0
t,c,i,y
-2.0
r,a,k,z
0
! joint two
r,c,k,x
90.0
r,c,k,y
90.0
r,c,i,x
-1.0
r,c,i,y
1.0
t,c,i,x
-2.0
t,c,i,y
2.0
r,a,k,z
0
! joint three
r,c,i,x
2.0
r,c,i,y
-2.0
t,c,i,x
-1.0
t,c,i,y
751.0
r,a,k,z
0
! joint four
r,c,i,x
-2.0
r,c,i,y
2.0
t,c,i,x
897.0
t,c,i,y
3.0
r,a,k,z
! joint five
r,c,k,x
90.0
r,c,k,y
90.0
r,c,i,x
1.0
r,c,i,y
-1.0
t,c,i,x
2.0
t,c,i,y
-107.0
r,a,k,z
0
! joint six
r,c,k,z
90.0
r,c,k,x
-90.0
r,c,i,x
-1.0
r,c,i,y
1.0
t,c,i,x
-2.0
t,c,i,y
-128.0
r,a,k,z
0
! tool
r,c,k,x
0.0
r,c,k,y
0.0
r,c,k,z
0.0
t,c,i,x
3.0
t,c,i,y
-210.0
t,c,i,z
-210.0
! the end
```

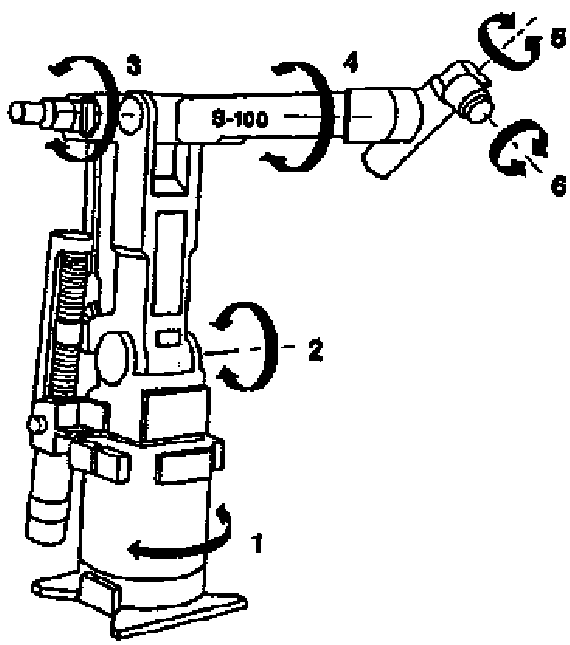

Figure 3. GMF SR-110 Robot

B.3 PUMA Robot Input File

```
! Puma Model
n,n,n,a,a,a
! joint one
r,c,k,x
-90
r,c,i,x
0
r,c,i,y
0
t,c,i,x
-380
t,c,i,y
-850
r,a,k,z
0
! joint two
r,c,k,y
-90
r,c,k,x
-90
r,c,i,x
0
r,c,i,y
0
t,c,i,x
358
t,c,i,y
0
r,a,k,z
0
! joint three
r,c,i,x
0
r,c,i,y
0
t,c,i,x
0
t,c,i,y
432
r,a,k,z
0.0
! joint four
r,c,k,x
90
r,c,k,y
90
r,c,i,x
0
r,c,i,y
0
t,c,i,x
-20
t,c,i,y
150
r,a,k,z
0
! joint five
r,c,k,x
-90
r,c,i,x
0
r,c,i,y
0
t,c,i,x
0
t,c,i,y
-433
r,a,k,z
0
! joint six
r,c,k,x
90
r,c,i,x
0
r,c,i,y
0
t,c,i,x
0
t,c,i,y
0
r,a,k,z
0
! the tool
t,c,i,x
0
t,c,i,y
0
t,c,i,z
56
```

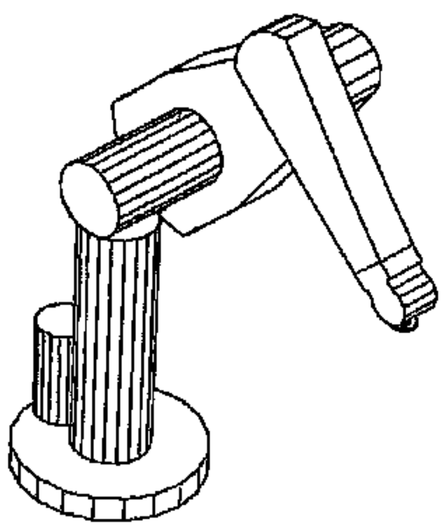

Figure 4. PUMA 560 Robot

APPENDIX C

MEASUREMENTS USED FOR PUMA ROBOT

Measurements Used for Calibration

Joint Angles

-107.287000	-156.176000	62.221000	-77.624000	18.737000	52.420000
-107.183000	166.998000	78.003000	143.036000	-29.905000	-183.350000
-115.527000	-164.553000	0.599000	149.821000	-76.685000	-222.390000
-112.604000	144.701000	-172.513000	26.922000	-66.539000	-13.360000
-81.343000	157.994000	155.803000	167.668000	44.484000	199.760000
-80.118000	138.472000	131.012000	-89.517000	-27.378000	15.420000
-73.790000	136.978000	138.054000	93.928000	99.970000	-264.500000
-73.427000	139.482000	-152.781000	-6.031000	-93.845000	-15.520000
-53.795000	169.514000	130.869000	-55.327000	-45.972000	61.360000
-54.124000	165.179000	81.843000	61.545000	42.380000	-107.710000
-33.981000	-163.889000	2.648000	57.211000	80.156000	-62.610000
-33.695000	142.828000	-167.294000	120.800000	79.574000	-123.080000
53.723000	-28.136000	128.765000	101.909000	34.475000	90.850000
52.366000	25.071000	83.331000	110.067000	41.155000	28.140000
44.714000	42.984000	53.718000	94.543000	78.519000	-0.270000
35.365000	-61.672000	-170.760000	111.275000	64.391000	42.620000
75.855000	21.912000	29.888000	23.406000	39.825000	178.710000
76.608000	42.984000	51.449000	95.663000	58.920000	-1.050000
67.148000	42.127000	55.701000	89.127000	99.970000	20.910000
67.582000	41.721000	-27.751000	9.921000	99.976000	7.390000
105.255000	10.223000	55.305000	146.036000	-29.158000	29.470000
104.865000	17.644000	98.130000	32.855000	-30.745000	40.440000
113.258000	-16.743000	-177.501000	24.730000	-76.432000	-211.170000
114.142000	37.760000	-8.866000	152.628000	-64.056000	-2.530000

Pose Measurements

-0.820850	-0.568000	-0.036220	-21.382482	-0.679030	-0.734600	-0.059820	-28.131008
-0.566900	0.821763	0.002064	483.236524	-0.735410	0.677900	0.020531	61.052227
0.028297	0.020697	-0.997620	39.633195	0.025516	0.058382	-1.000030	59.185454
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.489250	-0.864650	-0.112100	-44.056300	-0.974520	-0.211600	-0.062800	-47.779432
-0.873230	0.488528	0.046090	51.367258	-0.210450	0.976103	-0.052980	479.031046
0.015489	0.121844	-0.992020	379.161040	0.073233	-0.038320	-0.994990	318.045338
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.999030	-0.003050	-0.034860	-370.705380	-0.091390	-0.954030	0.288402	-373.824500
-0.005360	0.998554	0.036502	483.194360	-0.996450	0.095934	0.000757	47.731528
0.034437	0.036851	-0.997630	58.499375	-0.028030	-0.286080	-0.957770	29.637431
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
0.171924	-0.065850	0.982267	-416.961320	-0.842730	-0.506190	-0.187370	-394.698220
-0.007000	-0.997860	-0.067430	77.982648	-0.440380	0.844122	-0.309670	475.597728
0.983691	0.004409	-0.173190	86.807396	0.314689	-0.176250	-0.932140	334.645508
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.993530	0.106765	-0.014290	-763.686560	-0.439610	-0.896350	-0.054850	-762.695960
0.106132	0.991968	-0.004440	480.403154	-0.897440	0.441142	0.001414	59.614664
0.011656	-0.005360	-0.998160	70.381114	0.023798	0.049371	-0.998060	65.908453
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.552710	-0.831520	-0.051310	-763.993900	-0.900530	0.435158	-0.008080	-761.735840
-0.832720	0.555093	-0.017750	57.724777	0.436342	0.899708	-0.001640	482.006656
0.043924	0.033485	-0.998000	386.505450	0.005004	-0.003890	-1.000340	388.806436
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.993940	0.109251	0.032322	-49.054004	-0.703340	-0.708720	-0.025110	-35.037268
0.105661	0.991772	-0.062510	491.240826	-0.707670	0.704193	-0.051890	47.269756
-0.039130	-0.057590	-0.996330	44.240196	0.054530	-0.019170	-0.997230	53.825775
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
0.013635	-0.836260	-0.544200	-7.876540	-0.963270	-0.263690	-0.067450	-33.634172
-0.999040	-0.038020	0.036396	28.014600	-0.261900	0.965202	-0.044280	481.859844
-0.053200	0.545055	-0.835880	159.371944	0.077350	-0.023790	-0.997000	341.952072
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.993580	0.122547	-0.032610	-374.924320	0.030457	-0.700630	-0.713280	-419.955980
0.123923	0.993356	0.022579	476.348806	-0.997730	-0.068340	0.025259	43.769686
0.034337	0.019776	-1.001450	50.213184	-0.066170	0.711033	-0.700710	50.260936
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.068420	-0.202840	-0.974930	-308.325520	0.982608	-0.082220	0.162085	-365.361220
-0.942140	0.334837	-0.002840	21.512378	-0.011320	-0.920920	-0.390710	471.403426
0.325887	0.919296	-0.214020	85.895612	0.182103	0.381771	-0.905510	364.804960
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.977630	-0.205750	-0.008700	-767.133340	0.516341	-0.855490	-0.001120	-759.848620
-0.205690	0.977055	-0.040490	469.875362	-0.856560	-0.513940	0.020805	50.076075
0.016629	-0.038350	-0.998300	64.410285	-0.020680	-0.007780	-0.998130	67.356406
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.804100	-0.590090	-0.036890	-745.048040	-0.899980	-0.432700	-0.017700	-752.403880
-0.591780	0.806354	0.006766	64.296417	-0.432860	0.900148	-0.044780	476.480632
0.026922	0.026348	-0.997500	377.172220	0.035596	-0.032920	-0.997450	378.232416
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000

Measurements Used for Performance Comparison

Joint Angles

-106.057000	152.584000	106.941000	119.290000	-20.468000	-149.030000
-97.300000	137.170000	133.297000	-89.297000	-81.282000	-126.310000
-118.383000	-116.708000	-7.509000	136.664000	-41.105000	-145.170000
-81.035000	-138.741000	29.713000	25.703000	19.160000	-121.280000
-81.145000	-177.369000	42.380000	12.343000	45.549000	-40.580000
-73.504000	-162.087000	-19.501000	16.584000	91.697000	-96.100000
-73.372000	-98.630000	-32.635000	24.192000	43.731000	-15.840000
-53.943000	-152.298000	54.937000	-99.915000	-35.728000	107.140000
-53.756000	154.962000	101.942000	-105.886000	-40.314000	-4.160000
-53.481000	137.043000	131.847000	86.726000	96.295000	49.770000
-34.069000	-116.774000	-8.855000	69.857000	63.320000	-19.110000
53.822000	7.015000	61.595000	65.418000	43.539000	115.600000
51.888000	11.547000	110.468000	122.871000	48.807000	-14.050000
35.041000	-14.821000	178.660000	121.734000	80.442000	-34.080000
34.618000	36.107000	-7.350000	60.793000	76.981000	170.710000
75.965000	-41.292000	155.500000	-32.003000	-28.932000	205.040000
75.833000	-2.258000	141.636000	162.015000	51.850000	-53.790000
65.275000	-17.891000	-158.346000	151.617000	93.082000	-83.690000
65.308000	-77.800000	-145.849000	147.832000	51.729000	7.160000
105.298000	-27.697000	130.732000	50.323000	-20.643000	166.600000
105.172000	24.977000	83.578000	39.716000	-24.505000	45.210000
104.678000	42.330000	52.531000	94.197000	-88.770000	-72.520000
114.423000	-64.160000	-165.718000	34.579000	-46.599000	-195.710000

Pose Measurements

-0.771300	-0.627600	-0.092050	-44.378626	0.017887	0.029184	0.999015	-106.151172
-0.632720	0.774581	0.014954	62.294770	0.707339	-0.708650	0.007688	48.028555
0.062166	0.069915	-0.994150	52.793646	0.707515	0.705979	-0.034490	120.041899
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.983800	-0.175330	-0.039030	-9.911080	0.292633	-0.952980	-0.062450	-379.056900
-0.174040	0.984564	0.000978	485.348534	-0.955450	-0.291220	-0.047900	480.678236
0.037733	0.008150	-0.998230	375.087134	0.028047	0.074518	-0.995610	52.648180
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.743650	-0.665340	-0.048330	-374.619520	0.273318	-0.962930	-0.035340	-366.666780
-0.666590	0.745488	-0.018680	55.501108	-0.964150	-0.271220	-0.019060	53.286000
0.048393	0.018724	-0.997530	55.835398	0.009442	0.041635	-0.999410	387.339078
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.990020	-0.130550	-0.060050	-369.384580	-0.994450	-0.082500	-0.033650	-765.677920
-0.130970	0.990930	-0.018940	484.707692	-0.084880	0.995629	0.012181	483.826820
0.061635	-0.009720	-0.997200	389.395462	0.033268	0.014080	-0.997340	66.268625
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
0.564250	-0.825340	0.025911	-762.187960	0.303640	0.468136	0.827601	-682.419260
-0.825400	-0.562910	0.018950	57.632244	0.770670	-0.633930	0.073011	39.625549
-0.001050	-0.031810	-0.998560	68.703647	0.558806	0.615585	-0.553330	143.884955
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.932290	0.359032	-0.048830	-763.094740	-0.957520	-0.283690	-0.057160	-44.734226
0.359692	0.930975	-0.024470	474.781118	-0.284390	0.958752	-0.003310	475.388432
0.035612	-0.039420	-0.997040	386.299964	0.055519	0.013695	-0.998250	29.430624
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.343620	-0.938440	-0.042060	-35.233864	-0.527180	-0.846870	-0.081250	-34.817812
-0.938500	0.345763	-0.040860	50.241479	-0.846560	0.532316	-0.058070	51.645769
0.053191	0.026356	-0.998260	63.884251	0.092919	0.039659	-0.995510	346.641928
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.991830	0.108200	-0.068990	-36.169346	-0.966480	-0.247200	-0.069180	-380.766320
0.107360	0.993576	-0.009080	489.328460	-0.250040	0.967514	0.019233	478.304606
0.067994	-0.015990	-0.996240	354.656136	0.061502	0.036693	-0.996660	56.839536
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.235770	-0.970620	-0.037270	-377.870720	0.045587	-0.993160	-0.114750	-370.873020
-0.971850	0.238339	-0.025810	52.466164	-0.999900	-0.039300	-0.047610	52.797278
0.034210	0.031319	-0.997970	57.238062	0.043093	0.118760	-0.992570	374.492520
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.901970	-0.428270	-0.032800	-365.622840	-0.911460	0.410677	-0.008440	-766.188460
-0.429010	0.902875	-0.034450	473.298520	0.408468	0.912763	0.018339	473.775532
0.044704	-0.016630	-0.997390	370.913152	0.017785	0.011596	-0.999130	70.542760
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
0.311874	-0.950000	-0.016710	-764.214880	0.268175	-0.069990	0.961032	-686.394360
-0.950360	-0.311560	0.017366	49.871706	-0.028510	-0.997690	-0.065680	38.435102
-0.022060	0.011275	-0.999470	70.558609	0.962513	-0.009400	-0.269970	160.142733
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
-0.998520	-0.021300	-0.032900	-751.362480				
-0.022530	0.998617	0.032739	478.867216				
0.031938	0.032914	-0.998370	389.217154				
0.000000	0.000000	0.000000	1.000000				

VITA

Thomas W. (Sibenthal) Ives was born September 24, 1961 became Thomas Wayne Ives in 1970 through his mother's second marriage. He received his Bachelor of Science Degree in Mechanical Engineering from The University of Texas at Austin in December 1984. He held his first engineering position in the Naval Nuclear Power Program with Westinghouse Electric Corporation. This position gave him direct contact with the Naval Nuclear program as a civilian. He completed Officer's Naval Nuclear Power School in Orlando, Florida in January 1986. He completed Naval Nuclear Prototype Training at the Idaho National Engineering Laboratory (INEL) in November 1986 receiving the Naval qualification of Engineering Officer of the Watch for operation of the A1W Plant which is a prototype of one of the four U.S.S. Enterprise dual reactor plants. In March of 1987, he received Naval qualification as an Engineering Duty Officer. In July of 1987, he became a qualified Nuclear Plant Engineer. In November of 1987, he passed the Naval Reactors Engineers Exam, and subsequently received the qualification of Shift Test Engineer. As a Shift Test Engineer, he directed a crew of nuclear navy personnel through the daily requirements of a comprehensive test program following the overhaul of the A1W prototype which also involved overseeing reactor plant conditions, operations, and maintenance. In June of 1989, he left Westinghouse to accept a position at the Texas A&M University (TAMU) Nuclear Science Center (NSC). While there, he managed the operations of a 1MW research reactor and commenced his graduate work in Mechanical Engineering. He left his position at the TAMU NSC in July of 1991 to concentrate more fully on his graduate studies. He has worked as a Research Assistant and as a Teaching Assistant for the Mechanical Engineering Department at TAMU, and he has also worked as a Research Assistant for the Electrical Engineering Department. He received his Master of Science degree in May of 1995. His Master of Science research was in the field of robot calibration.

He married Nola Sue Nichols in March of 1988. In July of 1993, God blessed them with a son, Gabriel Thomas Ives. In September of 1996, God blessed them with a daughter, Anna Michelle Ives. The Ives family is currently living in Boise, Idaho where Thom is employed by SCP Global Technologies.