AUTOMATIC CALIBRATION OF SPACE BASED MANIPULATORS AND MECHANISMS

A FINAL REPORT
for
GRANT NA~~S~~6 9-192

Submitted to

Charles R. Price
Teleoperator Systems Branch
Nasa Johnson Space Center
Houston, Texas 77058

by

Louis J. Everett
Mechanical Engineering Department
Texas A&M University
College Station, Texas 77843

January 6, 1987 to June 5, 1988

AUTOMATIC CALIBRATION OF SPACE BASED MANIPULATORS AND MECHANISMS

A FINAL REPORT
for
GRANT NAS 9-192

Submitted to

Charles R. Price
Teleoperator Systems Branch
Nasa Johnson Space Center
Houston, Texas 77058

by

Louis J. Everett
Mechanical Engineering Department
Texas A&M University
College Station, Texas 77843

January 6, 1987 to June 5, 1988

CONTENTS

LIST OF FIGURES

Part 1
EXECUTIVE SUMMARY

1.1 Task Overview

Grant NAG 9-192 supported twelve months of research comprising four related tasks in manipulator kinematic calibration. This section of the report summaries the objectives and status of each task.

1.1.1 Redundant Manipulator Calibration

There have been several manipulator configurations proposed for use on a space station. One manipulator has seven degrees of freedom, another fourteen. NASA personnel expressed concern about the ability to calibrate redundant degree of freedom manipulators. Because of this concern we wanted to demonstrate that redundant manipulators present no unique problems.

Calibration of a seven degree of freedom manipulator has been simulated. Calibration of redundant manipulators presents no unique difficulties.

Calibration is a regression problem in which several unknown parameters are chosen to minimize error between a calculated and measured tool location [1]. As an optimization problem, it remains well defined regardless of the number of joints and measurements.

1.1.2 Closed Loop Manipulator Calibration

There is increased interest in the study of direct drive manipulators because they reduce (or eliminate) several non-geometry sources of error such as backlash and gear harmonics. In addition, they produce much higher stiffnesses. Because direct drive electric manipulators require very large motors, they are sometimes impractical. To overcome this problem, designers have utilized mechanism transmissions to produce optimal power transmission from motor to linkage. For example, Bajpai and Roth [2] analyzed the basic kinematic geometry and workspace properties of a simple five-bar-closed-loop robot. There are many examples of closed-loop joint actuation manipulators in commercially available systems. Some manipulators designed by GMF and Cincinnati Milacron have linkage transmissions.

This work presents a calibration model that can be applied on a closed-loop robot. It is an expansion of open-loop kinematic calibration algorithms subject to constraints. A closed-loop robot with a five-bar linkage transmission has been tested. Results show that the algorithm converges within a few iterations.

1.1.3 Study of Calibration Models

This study formalizes the concept of model differences. Differences are categorized as structural and numerical; structural differences are emphasized here. The work demonstrates that "geometric" manipulators can be visualized as points in a vector space with the dimension of the space depending solely on the number and type of manipulator joints. Visualizing parameters in a kinematic model as the coordinates locating the manipulator in vector space enables a standard evaluation of the usefulness and accuracy of various manipulator models. Key results include a derivation of the maximum number of parameters necessary for models, a formal discussion on the inclusion of *extra* parameters, and a method to predetermine a minimum model structure for a kinematic manipulator.

1.1.4 Using Single Point Sensors

Single point sensors can measure the position of only one point fixed to the manipulator's end effector. When single point sensors have been used for calibration, it has not been possible to calibrate the orientation of the tool. Furthermore, it has often been difficult to calibrate the sensor system. Results in the literature seldom provide a complete calibration of the manipulator. Presented here is a technique that enables single point sensors to gather sufficient information to complete the calibration. In addition, the method can also reduce the burden of calibrating the sensor system itself.

1.2 Travel Supported by the Grant

In December 1987, the principal investigator attended the ASME Winter Annual Meeting in Boston. One objective of the visit was to evaluate interest in the subject of robot acceptance tests. A paper session on the general topic of acceptance testing for manipulators has preliminary approval from the chairman of the Robotics panel of the Dynamic Systems, Measurement, and Control division of ASME. The principal investigator intends to organize an acceptance testing session in an appropriate upcoming conference.

In March 1988, the principal investigator attended the Joint Applications in Instrumentation, Process, and Computer Control mini symposium held at the University of Houston, Clear Lake. The objective was to remain current with robotics and automation pertaining to space applications.

In April 1988, the principal investigator and graduate student Cheng-Yang Lin attended the IEEE Robotics and Automation Conference in Philadelphia. The objectives of the trip were to present two papers generated from grant money. Both papers were well received, each inspired several questions and discussion. In addition to presenting the papers, the principal investigator contributed to a half day short course on manipulator calibration which also was well received.

In May 1988, the principal investigator visited NASA Marshall Space Flight Center. The trip's objectives were to present research results, become familiar with Marshall's activities, and discuss possibilities for continued NASA support. The trip resulted in a better understanding of NASA's ground based automation requirements.

1.3 Bibliography Generated from the Grant

To date, six papers generated from grant support, have been submitted for publication and two have appeared in print. The papers, *Kinematic Calibration of Manipulators with Closed Loop Actuated Joints*, and *A Study of Kinematic Models for Forward Calibration of Manipulators* have appeared in Volume 2 of the 1988 IEEE International Conference on Robotics and Automation Proceedings.

Two journal versions of the former papers have been submitted for publication. The papers *Forward Calibration of Closed Loop Jointed Manipulators* and *Similarity in Structurally Different Kinematic Models in Forward Calibration* were submitted to the **International Journal of Robotics Research**. Reviewers comments have not been received to date.

Another paper titled: *Completing the Forward Kinematic Calibration of Open Loop Manipulators When Single Point Position Sensors are Used* has been submitted to the **Journal of Robotic Systems**. Reviewers of the paper asked for specific changes and resubmission. The changes are nearly complete at this time.

Lastly, a conference paper titled: *A New Method for Calibrating the Final Ori-*

entation Parameters of a Manipulator has been submitted for publication at the 1988 Winter Annual Meeting of ASME.

1.4 Future Plans

This section briefly discusses five projects which are planned for the near future.

1.4.1 Nonkinematic Models

Kinematic calibration models for open and closed loop manipulators are well understood. Reported results from this grant have shown how structurally different kinematic models of manipulators can be compared. Future work is to develop similar theory for so called nonkinematic models.

Nonkinematic models will be studied by allowing the constant parameters in kinematic models to vary. Each constant will be expressed as a series function of quantities including joint positions, load, and velocity. Careful choice of the series functions used should enable the development of a generic calibration model representing a superset of currently existing models.

The project currently has one student assigned to it but is not funded.

1.4.2 Numerically Different Calibration Models

The reported results on structural differences in kinematic models will be extended to numerical differences. The work may provide new understanding about how nearly singular points affect calibration, and manipulation tasks. Additionally, the work may discover the causes of ill-conditioned calibration Jacobians. This knowledge relates to locating calibration measurement devices in the workspace and inverse kinematic solutions.

Numerical differences are currently being studied by the principal investigator.

1.4.3 Existence and Uniqueness of Solution

Although calibration algorithms generally converge there is no formal proof that calibration solutions are unique. Solution existence can be assured through proper problem definition but uniqueness remains a problem. This ongoing work will attempt to determine the conditions which guarantee unique solutions. The problem is being addressed using convex sets, and stability theories.

1.4.4 Using Calibration for Fault Isolation

Calibration results can be used for detecting faults in a manipulator. If, for example, a joint position sensor is providing incorrect output. It may be possible to perform a quick calibration to detect the faulty sensor. This process was performed to a small extent by Mooring and Pack [3]. Currently this work is not funded nor being investigated although it may have considerable impact on NASA operations.

1.4.5 Closed Loop Calibration - Continuation

The closed loop mechanisms calibration project is not complete. It has been shown that it is possible to calibrate manipulators having closed loops but there remains several areas to be investigated. For example, the singularities arising when the loops are nearly planar need to be better understood. A formulation which models internal forces may also be important. Fundamental to all future closed loop calibration work will be a sensitivity study to determine when calibration is required and what improvement can be achieved using calibration.

Part 2
TECHNICAL REPORT

2.1 Introduction

The configuration of a manipulator is specified one of two ways: (1) by a set of joint variables or (2) by the tool position and orientation (the pose [4]). There is typically one measurable joint variable for each degree of freedom of motion. The joint measurement usually consists of the relative displacement of each joint. Pose can be specified by a 4 by 4 homogeneous transform (${}^{w}T^{t}$) [5]. Because tool position feedback is uncommon, it may be impossible to measure tool pose accurately during motion. Manipulators that do not have accurate measurements of tool pose are often operated in a teach playback mode, where an operator places the end effector in a desired position and the controller "remembers" the joint configuration. During manipulation, the controller replays the memorized sequence of joint configurations. It may be possible (even in the absence of pose measurement) to specify pose rather than teaching if one mathematically relates pose to joint position. This can be done with a kinematic model [5]. When utilizing a kinematic model, the controller relies on accurate joint positioning for accurate tool positioning.

A recognized problem with kinematic models is their unsatisfactory accuracy. For various reasons [6,7], including manufacturing tolerances, accurate mathematical relations between tool pose and joint configurations are difficult to obtain. Manipulator calibration has been proposed as a means for reducing the error in the mathematical relation between joint positions and tool pose. A complete discussion of calibration can be found in Whitney, Lozinski, and Rourke [6], Chen and Chao [8], Mooring and Tang [7], Hyatti [9] Stone, Sanderson, and Neuman [10], Everett, Driels and Mooring [1], and Everett and Hsu [11].

2.1.1 Types of Calibration

As pointed out by Roth, Mooring, and Ravani [12], the term calibration represents three significantly different processes. Level I, joint level calibration, consists of calibrating the joint feedback sensors. The homing process on some manipulators with incremental joint encoders is Level I calibration. A variation of Level I, workspace calibration, is the process of determining the position and orientation of the mechanism base relative to a fixture holding the manipulated objects. A discussion of workspace calibration can be found in [13]. Level I calibration is relatively simple and does not require sophisticated measurement devices. Level II calibration (the subject of this research) is the calculation of an accurate mapping between joint position and tool pose by measuring this relationship at a certain number of locations and performing a regression analysis to fit model parameters to the measurements. Level III calibration, dynamic calibration, is the calculation of inertia and similar terms that affect the motion of the tool.

A further subdivision of Level II calibration is discussed by Whitney, Lozinski, and Rourke [6] and Shamma and Whitney [14]. Like the inverse kinematic problem, inverse calibration seeks a direct relation with tool pose as input and joint position as output. There are two advantages of inverse calibration: (1) the error sources need not be known, and (2) results do not require an inverse kinematic solution. A major disadvantage is that inverse calibration requires extensive position measurements. Considering the current expense of measurement, the method may not be suitable for some applications.

Forward calibration [6,7,8,9,10,11,12,15] assumes a model based on assumptions of error sources and attempts to determine the "best fit" set of model constants that causes the computed tool pose to approximate measured pose. There are two advantages of forward over inverse calibration: (1) it may require fewer measurements, and (2) it gives insight to the sources of error, which can be valuable when fed back to the manipulator designer. This research addresses forward calibration only.

2.1.2 Forward Calibration

The first step in performing forward calibration is to choose a model that is assumed capable of relating joint to tool configurations. Authors usually present their version of a kinematic model and proceed. Because numerical results differ with each model, results are difficult to compare.

It is assumed that a manipulator is characterized by a set of constants (e.g. Hartenberg-Denavit constants [5]) called parameters. These parameters typically consist of rigid body transformations, including translations such as link lengths, and rotations such as twist angles. Parameters can be specified by screw operators [7]. The jth parameter of a manipulator whether it is a rotation or translation is refered to as C_j. The complete set of parameters is referred to collectively as the vector $\overline{C}$. A specific manipulator is characterized by listing a set of parameters as in $\overline{C} = (C_1, \ldots, C_m)$ where m is the number of parameters. Since the manipulator's joints are lower pair, joint i, lying between bodies i and $i + 1$, moves along or rotates about a single line specified with unit vector $\vec{n}_i$. The set of joint variables is referenced collectively as $\overline{\Theta}$.

Most algorithms discussed in the literature are intimately tied to a particular modeling strategy. The software used in this study was written with the ability to model and calibrate manipulators in a variety of ways. The software has been used with several of the popular modeling techniques such as those of Whitney [6], Chen [8], Hayati [15], and Everett and Hsu [16].

2.2 Redundant Manipulator Calibration

The objective of the redundant manipulator calibration task was to demonstrate that redundant manipulators can be calibrated. The calibration problem was simulated since a redundant manipulator was unavailable. A seven degree of freedom manipulator similar to a PUMA arm was calibrated.

The calibration problem differs significantly from inverse kinematic problems. In inverse problems, one must calculate joint positions corresponding to a desired pose of the tool. This pose specification consists of six given quantities (constraints). If the number of joints (free variables) is less than six, it may not be possible to satisfy all the constraints. If the number of joints is greater than six, as for redundant manipulators, there will be an infinite number of solutions. In calibration, however, there can be 30 free variables (for a 6 degree of freedom PUMA) these are constrained to produce minimum error between measured and calculated manipulator pose. Hence, calibration is a regression problem and therefore is well defined regardless of the number of free variables.

2.2.1 The Calibration Algorithm

The pose of a manipulator's tool can be expressed as a four by four transformation matrix T_o, see Paul [5]. It is common that one computes the pose of a tool as a product of matrices, A_i, each fixed to a link of the robot.

$$T_o = A_1 A_2 \cdots A_i \cdots A_n \tag{1}$$

The four by four matrix A_i is the relationship between successive link fixed coordinate systems. One well known convention for such matrices is the Hartenberg-Denavit transformation [17]. It is easily understood that different manipulator configurations and sizes have different A_i matrices.

Let dT be the differential change between the measured T matrix, T_m, and the calculated T matrix, T_o:

$$dT = T_m - T_o \tag{2}$$

and define δT_o as:

$$dT = T_o \delta T_o \tag{3}$$

Here δT_o means a differential change of the calculated T matrix relative to the coordinates of the tool. Kinematic calibration is the process of choosing parameters in A_i so that dT approaches zero. The solution is found iteratively.

Using first order terms of a Taylor series to relate dT to small changes in parameter values, results in:

$$dT = \sum_{k=1}^{p} \frac{\partial T_o}{\partial P_k} (dP_k) \tag{4}$$

Here P_k represents the k'th unknown parameter and p is the number of parameters (unknowns) in the model.

It can be shown [5] that combining equations 2, 3, and 4 produces the equation:

$$T_o^{-1}(T_m - T_o) = \delta T = \begin{bmatrix} 0 & -\delta z & \delta y & dx \\ \delta z & 0 & -\delta x & dy \\ -\delta y & \delta x & 0 & dz \\ 0 & 0 & 0 & 0 \end{bmatrix} = [J][dP] \tag{5}$$

Here δT is thought of as an error between actual and calculated tool location, $[dP]$ is a vector of kinematic parameter errors, and $[J]$ is a Jacobian matrix (containing partial derivatives) relating $[dP]$ and δT. Quantities dx, dy, and dz are the translation errors between computed and measured positions, δx, δy, and δz are the orientation errors.

For each measurement, at most six independent equations can be extracted from equation 5. Therefore, if there are n parameters to identify, there must be

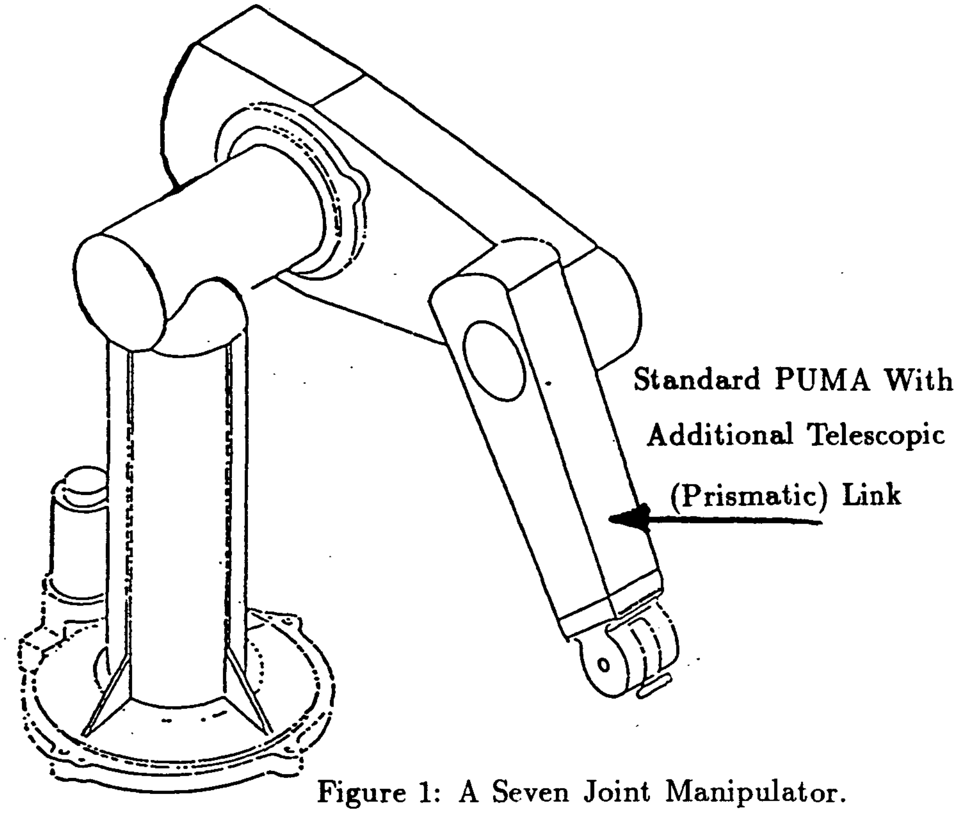

Figure 1: A Seven Joint Manipulator.

a minimum of $n/6 = m$ measurements made. According to theory developed by Everett et.al. [1] and [11], the number of independent kinematic parameters is $6 + 4R + 2P$, where R is the number of revolute joints and P is the number of prismatic joints. For the case of a seven revolute joint manipulator, 34 parameters need to be identified. This requires a minimum of six measurements.

2.2.2 Example

A simulated calibration was performed for a seven joint manipulator shown in figure 1. The open-loop kinematic equation is constructed of transformations from the base to the tool which can be expressed as:

$$T_o = A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 \tag{6}$$

The unknowns for this problem are the constant parameters in the A matrices.

The solution process begins with selection of values for the unknown A matrices. This is equivalent to choosing the kinematics of the manipulator. The values chosen (hence the correct calibration solutions) are:

$$A_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(2) & -\mathrm{S}(2) & 0 \\ 0 & \mathrm{S}(2) & \mathrm{C}(2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(1) & 0 & \mathrm{S}(1) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(1) & 0 & \mathrm{C}(1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & .67 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$A_1 = \begin{bmatrix} \mathrm{C}(\theta_1) & -\mathrm{S}(\theta_1) & 0 & 0 \\ \mathrm{S}(\theta_1) & \mathrm{C}(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & .05 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-90) & -\mathrm{S}(-90) & 0 \\ 0 & \mathrm{S}(-90) & \mathrm{C}(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$A_2 = \begin{bmatrix} \mathrm{C}(\theta_2) & -\mathrm{S}(\theta_2) & 0 & 0 \\ \mathrm{S}(\theta_2) & \mathrm{C}(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & .42 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(2) & -\mathrm{S}(2) & 0 \\ 0 & \mathrm{S}(2) & \mathrm{C}(2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(-2) & 0 & \mathrm{S}(-2) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(-2) & 0 & \mathrm{C}(-2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$A_3 = \begin{bmatrix} \mathrm{C}(\theta_3) & -\mathrm{S}(\theta_3) & 0 & 0 \\ \mathrm{S}(\theta_3) & \mathrm{C}(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & .17 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(90) & -\mathrm{S}(90) & 0 \\ 0 & \mathrm{S}(90) & \mathrm{C}(90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(1) & -\mathrm{S}(1) & 0 \\ 0 & \mathrm{S}(1) & \mathrm{C}(1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(-2) & 0 & \mathrm{S}(-2) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(-2) & 0 & \mathrm{C}(-2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$\begin{gathered} A_5 = \begin{bmatrix} \mathrm{C}(\theta_5) & -\mathrm{S}(\theta_5) & 0 & 0 \\ \mathrm{S}(\theta_5) & \mathrm{C}(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & .1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-90) & -\mathrm{S}(-90) & 0 \\ 0 & \mathrm{S}(-90) & \mathrm{C}(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{gathered} \tag{12}$$

$$\begin{gathered} A_6 = \begin{bmatrix} \mathrm{C}(\theta_6) & -\mathrm{S}(\theta_6) & 0 & 0 \\ \mathrm{S}(\theta_6) & \mathrm{C}(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & .05 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(90) & -\mathrm{S}(90) & 0 \\ 0 & \mathrm{S}(90) & \mathrm{C}(90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{gathered} \tag{13}$$

$$\begin{gathered} A_7 = \begin{bmatrix} \mathrm{C}(\theta_7) & -\mathrm{S}(\theta_7) & 0 & 0 \\ \mathrm{S}(\theta_7) & \mathrm{C}(\theta_7) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & .25 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & .05 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(1) & -\mathrm{S}(1) & 0 \\ 0 & \mathrm{S}(1) & \mathrm{C}(1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(-2) & 0 & \mathrm{S}(-2) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(-2) & 0 & \mathrm{C}(-2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{gathered} \tag{14}$$

A forward kinematic solution is performed using the correct A matrices to calculate the relation between input joint values and the pose of the tool. The input angles were chosen as:

Measurement No.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
1	10	20	30	.4	50	60	70
2	-10	-30	-40	.43	-40	-30	90
3	20	50	70	.16	-10	20	-50
4	-60	15	90	.20	30	45	10
5	-30	45	-30	.30	20	-10	-30
6	45	60	10	.1	70	-46	30
7	-45	-20	-10	.25	-50	-60	20
8	0	-60	-70	.3	-20	-20	45
9	60	90	-90	.35	-30	30	-45
10	30	-50	40	.50	-70	90	-90
11	10	0	45	-.1	75	35	0
12	-90	-90	-45	-.2	25	-45	-20

In an actual calibration, one would position the manipulator and measure the joint angles and tool pose.

With the simulated calibration data, the calibration is performed. This requires computing the partial derivatives of the kinematic equations. In addition, the algorithm requires an initial estimate of the unknowns. The initial estimates are:

$$A_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(0) & -\mathrm{S}(0) & 0 \\ 0 & \mathrm{S}(0) & \mathrm{C}(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(0) & 0 & \mathrm{S}(0) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(0) & 0 & \mathrm{C}(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & .67 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$A_1 = \begin{bmatrix} \mathrm{C}(\theta_1) & -\mathrm{S}(\theta_1) & 0 & 0 \\ \mathrm{S}(\theta_1) & \mathrm{C}(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-90) & -\mathrm{S}(-90) & 0 \\ 0 & \mathrm{S}(-90) & \mathrm{C}(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$A_2 = \begin{bmatrix} \mathrm{C}(\theta_2) & -\mathrm{S}(\theta_2) & 0 & 0 \\ \mathrm{S}(\theta_2) & \mathrm{C}(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & .43 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(0) & -\mathrm{S}(0) & 0 \\ 0 & \mathrm{S}(0) & \mathrm{C}(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(0) & 0 & \mathrm{S}(0) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(0) & 0 & \mathrm{C}(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$A_3 = \begin{bmatrix} \mathrm{C}(\theta_3) & -\mathrm{S}(\theta_3) & 0 & 0 \\ \mathrm{S}(\theta_3) & \mathrm{C}(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.15 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(90) & -\mathrm{S}(90) & 0 \\ 0 & \mathrm{S}(90) & \mathrm{C}(90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(0) & -\mathrm{S}(0) & 0 \\ 0 & \mathrm{S}(0) & \mathrm{C}(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(0) & 0 & \mathrm{S}(0) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(0) & 0 & \mathrm{C}(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

$$A_5 = \begin{bmatrix} \mathrm{C}(\theta_5) & -\mathrm{S}(\theta_5) & 0 & 0 \\ \mathrm{S}(\theta_5) & \mathrm{C}(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-90) & -\mathrm{S}(-90) & 0 \\ 0 & \mathrm{S}(-90) & \mathrm{C}(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

$$A_6 = \begin{bmatrix} \mathrm{C}(\theta_6) & -\mathrm{S}(\theta_6) & 0 & 0 \\ \mathrm{S}(\theta_6) & \mathrm{C}(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(90) & -\mathrm{S}(90) & 0 \\ 0 & \mathrm{S}(90) & \mathrm{C}(90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

$$
\begin{aligned}
A_7 = & \begin{bmatrix} C(\theta_7) & -S(\theta_7) & 0 & 0 \\ S(\theta_7) & C(\theta_7) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & .2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
& \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(0) & -S(0) & 0 \\ 0 & S(0) & C(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(0) & 0 & S(0) & 0 \\ 0 & 1 & 0 & 0 \\ -S(0) & 0 & C(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{22}
$$

With this data, the algorithm was used to estimate the correct parameters. After convergence, the two norm of the difference between estimated and correct parameters was $2.7069719 \times 10^{-29}$.

2.2.3 Conclusions

The problems of calibrating redundant manipulators are fundamentally the same as nonredundant manipulators. As a result there is no need to study redundant manipulator calibration as a separate issue.

2.3 Calibration of Closed Loop Manipulators

A method for performing forward kinematic calibration of manipulators with one or more closed-loop actuated joints is presented. Closed loop manipulators are unique from conventional manipulators. Closed loop manipulators may contain ball and socket joints and develop significant internal forces.

The technique used for calibration is an extension of the algorithm designed for open-loop jointed manipulators and is equivalent to minimizing a constrained objective function. The constraints arise from the closed-loop mechanisms in the manipulator. The objective function is taken as the integral of end effector position and orientation error and the closed loop constraints are dealt with using the method of Lagrange multipliers.

2.3.1 Modeling

Since forward calibration is investigated, the model is expressed such that joint position is input and tool pose is output. In addition to expressing tool pose, it is necessary to express each closed-loop because the tool relationship must be consistent with the physical constraints of the mechanism.

The component of the model relating joint configuration to tool pose is the open-loop transformation, T_o. The component expressing the closed-loop is the closed-loop transformation, T_c. There are several methods for expressing these transformations; we use four-by-four homogeneous transformation matrices. In this case, the open-loop equation is written:

$$T_o = A_{o_1} A_{o_2} \cdots A_{o_i} \cdots A_{o_n} \tag{23}$$

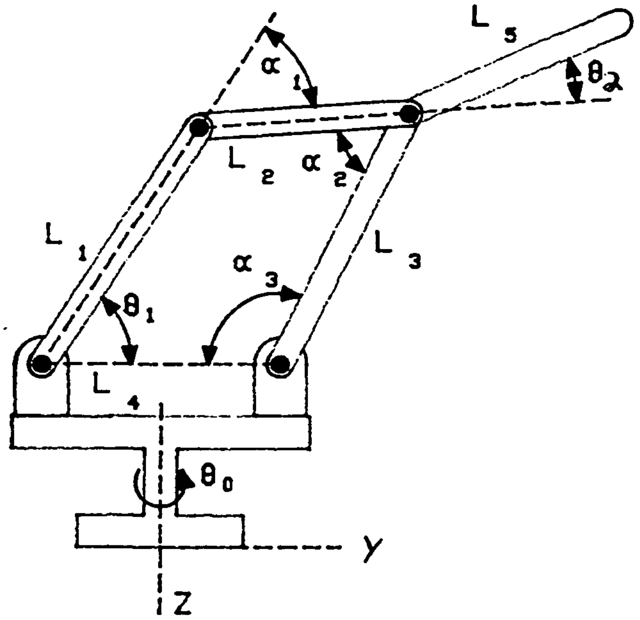

Figure 2: A Simple Manipulator With a Single Four Bar Closed-Loop.

The T_c transform has the form:

$$T_c = A_{c_1} A_{c_2} \cdots A_{c_i} \cdots A_{o_n} = [I] \tag{24}$$

Again A_{o_i} and A_{c_i} represent homogeneous matrices. There is one equation like equation 24 for every closed-loop in the mechanism. Although equations 23 and 24 appear simple, they are nonlinear functions of a large number of unknowns.

Three types of quantities may appear in equations 23 and 24. The first type is the set of measurable joint positions ($\vec{\Theta}$); the second is a set of imprecisely known constants ($\vec{C}$); the third is a set of unmeasurable variables ($\vec{\alpha}$).

Note the two-dimensional manipulator shown in figure 2. Although the mathematics applies to a much more general class of three-dimensional high degree of freedom manipulators, this simple example clarifies the concepts. In the two dimensional manipulator, an open-loop transformation could have the form:

$$T_o = \begin{bmatrix} C(\theta_0) & -S(\theta_0) & 0 & 0 \\ S(\theta_0) & C(\theta_0) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\theta_1) & -S(\theta_1) & 0 \\ 0 & S(\theta_1) & C(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\alpha_1) & -S(\alpha_1) & 0 \\ 0 & S(\alpha_1) & C(\alpha_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (25)$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\theta_2) & -S(\theta_2) & 0 \\ 0 & S(\theta_2) & C(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Here $C(\theta)$ and $S(\theta)$ denote $\cos(\theta)$ and $\sin(\theta)$, respectively.

The closed-loop transformation could have the form:

$$T_c = I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\theta_1) & -S(\theta_1) & 0 \\ 0 & S(\theta_1) & C(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\alpha_1) & -S(\alpha_1) & 0 \\ 0 & S(\alpha_1) & C(\alpha_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\alpha_2) & -S(\alpha_2) & 0 \\ 0 & S(\alpha_2) & C(\alpha_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\alpha_3) & -S(\alpha_3) & 0 \\ 0 & S(\alpha_3) & C(\alpha_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\Delta) & -S(\Delta) & 0 \\ 0 & S(\Delta) & C(\Delta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (26)$$

In these equations, $\vec{\Theta} = (\theta_0, \theta_1, \theta_2)$, $\vec{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$, and $\vec{C} = (X_0, Y_0, Z_0, L_1, L_2, L_3, L_4, L_5, \Delta)$.

2.3.2 The Objective Function

The calibration problem is to determine the unknowns in equations 23 and 24. The unknowns are selected by minimizing the square error between model predicted and measured tool pose. Although pose is computed by a four-by-four homogeneous matrix, it is convenient to represent error as a six by one vector of the individual errors.

If tool pose is computed by equation 23, and measured pose is expressed as T_m, pose error can be expressed as:

$$\vec{F}_{(\vec{\Theta},\vec{\alpha},\vec{C})} = \vec{S}\left[T_o^{-1}\left(T_m - T_o\right)\right] \tag{27}$$

The last matrix in equation 27 has the same form as in equation 5. Therefore the operator $\vec{S}\,[]$ from equation 27 can be defined as:

$$\vec{S}\left[T_o^{-1}\left(T_m - T_o\right)\right] = (dx/l, dy/l, dz/l, \delta x, \delta y, \delta z)^T \tag{28}$$

where l is a normalizing length. Since $\vec{F}$ is a function of joint position, it varies over the workspace; hence a scalar objective function is defined as an integration over the entire workspace:

$$f = \int_V \frac{1}{2}\vec{F}^T\vec{F}d\vec{\Theta} \tag{29}$$

Since it is impractical to compute the integration, it is approximated as a summation over joint positions. After simplification, the objective function becomes:

$$f = \sum_{i=1}^{m}\frac{1}{2}\vec{F}^T_{(\vec{\Theta}_i,\vec{\alpha}_i,\vec{C})}\vec{F}_{(\vec{\Theta}_i,\vec{\alpha}_i,\vec{C})} \tag{30}$$

Here m represents the number of joint sets or measurements used in the calibration. Since not all variables in equation 30 are independent, the objective function must be minimized subject to the constraints, T_c. This can be done several ways.

One method for dealing with constrained optimization problems is to solve the constraints explicitly for the dependent variables and substitute these into the objective function. This is difficult but has been successfully performed in some mechanism design cases [18]. The advantage of this technique is that it reduces the number of variables in the optimization problem. The disadvantage is that the constraints are not easy to solve explicitly.

Another method for solving constrained optimization problems is penalty weighting, which has been used in mechanism design problems [19]. With the penalty weighting method, the constraint equations are expressed as:

$$0 = \lambda \int_V \vec{S}^t \left[T_c - [I]\right] \vec{S} \left[T_c - [I]\right] d\vec{\Theta} \tag{31}$$

The objective function is modified:

$$f' = \int_V \left[\frac{1}{2} \vec{F}^T \vec{F} + \lambda \vec{S}^t \left[T_c - [I]\right] \vec{S} \left[T_c - [I]\right]\right] d\vec{\Theta} \tag{32}$$

where λ is a large arbitrarily chosen constant. As λ increases, any solution not satisfying the constraints highly penalizes the objective function. The technique differs from the Lagrange multiplier method [20] in that λ is chosen arbitrarily. This difference results from the restriction that the Lagrange multiplier method is valid only when the optimum of the modified objective function is not an extremum of the constraint equations. Because the constraint equations are a squared form, the optimal solution of the objective function is also a minimum of the constraint equations; hence the restriction is violated. Two advantages of the penalty weighting method are (1) it introduces no unknown variables, and (2) it does not require explicit solutions to the constraint equations. A disadvantage is that as λ increases, the profile of the objective function tends toward a deep, narrow, long valley which can cause the iterative solution process to oscillate and converge slowly [21].

The method chosen for this work is the Lagrange multiplier technique. To apply this method, the constraints are modified to:

$$0 = \vec{\lambda}^T_{(\vec{\Theta})} \vec{S} \left[T_c - [I]\right] \tag{33}$$

Note that in this case $\vec{\lambda}$ is an unknown vector function of $\vec{\Theta}$. These unknown functions number six times the number of closed-loops. The objective function is also modified as:

$$f' = \int_V \left[\frac{1}{2}\vec{F}^T \vec{F} + \vec{\lambda}^T_{(\vec{\Theta})}\vec{S}\left[T_c - [I]\right]\right] d\vec{\Theta} = \vec{\lambda}^T_{(\vec{\Theta})}\vec{G} \tag{34}$$

The unknowns in equation 34, $\vec{\alpha}$, $\vec{C}$, and $\vec{\lambda}_{(\vec{\Theta})}$ are treated as independent. The modified objective function is discretized to become:

$$\begin{gathered} f' = \sum_{i=1}^m \left\{\left[\tfrac{1}{2}\vec{F}^T_{(\vec{\Theta}_i,\vec{\alpha}_i,\vec{C})}\vec{F}_{(\vec{\Theta}_i,\vec{\alpha}_i,\vec{C})} + \right.\right. \\ \left.\left. \vec{\lambda}^T_{(\vec{\Theta}_i)}\vec{S}\left[T_{c(\vec{\Theta}_i,\vec{\alpha}_i,\vec{C})} - [I]\right]\right]\right\} \end{gathered} \tag{35}$$

2.3.3 Critical Points of the Objective Function

The relative maximums or minimums of the objective function are located at the critical points of the function. The critical points of the objective function are defined by:

$$\begin{gathered} \sum_{i=1}^m \left\{\vec{F}^T_{(\vec{\Theta}_i,\vec{\alpha}_i,\vec{C})}\frac{\partial \vec{F}}{\partial \vec{C}^T} + \vec{\lambda}^T_{(\vec{\Theta}_i)}\frac{\partial \vec{G}}{\partial \vec{C}^T}\right\} = \vec{r}^{\,T} = 0 \\ \vec{F}^T_{(\vec{\Theta}_j,\vec{\alpha}_j,\vec{C})}\frac{\partial \vec{F}}{\partial \vec{\alpha}^T} + \vec{\lambda}^T_{(\vec{\Theta}_j)}\frac{\partial \vec{G}}{\partial \vec{\alpha}^T} = \vec{\gamma}_j^{\,T} = 0 \\ \vec{G}_{(\vec{\Theta}_j,\vec{\alpha}_j,\vec{C})} = \vec{\Delta}_j = 0 \end{gathered} \tag{36}$$

The terms $\frac{\partial \vec{v}}{\partial \vec{w}^T}$ represent the partial derivatives of a column vector $\vec{v}$ with respect to a column vector $\vec{w}$, which is a matrix with row i column j given as $\frac{\partial v_i}{\partial w_j}$. The free subscript j in equation 36 assumes numbers from one to m.

The Newton iteration process is used to solve equations 36 for the critical point.

The iterative equations can be expressed as:

$$
\begin{aligned}
-\vec{r}_k = \textstyle\sum_{i=1}^{m} \Big[& \left(\vec{F}^T_{,\vec{p}\vec{a}^T}\vec{F} + \vec{F}^T_{,\vec{p}}\vec{F}_{,\vec{a}^T} + \vec{G}^T_{,\vec{p}\vec{a}^T}\vec{\lambda}\right)_{ki}\left(\vec{\alpha}_{(k+1)i} - \vec{\alpha}_{ki}\right) \\
& + \left(\vec{F}^T_{,\vec{p}\vec{p}^T}\vec{F} + \vec{F}^T_{,\vec{p}}\vec{F}_{,\vec{p}^T} + \vec{G}^T_{,\vec{p}\vec{p}^T}\vec{\lambda}\right)_{ki}\left(\vec{C}_{(k+1)} - \vec{C}_k\right) \\
& + \vec{G}^T_{,\vec{p}}\left(\vec{\lambda}_{(k+1)i} - \vec{\lambda}_{ki}\right) \\
-\vec{\gamma}_{kj} = & \left(\vec{F}^T_{,\vec{a}\vec{a}^T}\vec{F} + \vec{F}^T_{,\vec{a}}\vec{F}_{,\vec{a}^T} + \vec{G}^T_{,\vec{a}\vec{a}^T}\vec{\lambda}\right)_{kj}\left(\vec{\alpha}_{(k+1)j} - \vec{\alpha}_{kj}\right) \\
& + \left(\vec{F}^T_{,\vec{a}\vec{p}^T}\vec{F} + \vec{F}^T_{,\vec{a}}\vec{F}_{,\vec{p}^T} + \vec{G}^T_{,\vec{a}\vec{p}^T}\vec{\lambda}\right)_{kj}\left(\vec{C}_{(k+1)} - \vec{C}_k\right) \\
& + \vec{G}^T_{,\vec{a}}\left(\vec{\lambda}_{(k+1)j} - \vec{\lambda}_{kj}\right) \\
-\vec{\Delta}_{kj} = & \vec{G}_{,\vec{a}^T}\left(\vec{\alpha}_{(k+1)j} - \vec{\alpha}_{kj}\right) + \vec{G}_{,\vec{p}^T}\left(\vec{C}_{(k+1)} - \vec{C}_k\right)
\end{aligned}
\tag{37}
$$

The subscript k represents estimates at the k iteration. In equation 37, a comma represents partial differentiation; the free variable j represents the j joint configuration. Consider the bilinear terms such as $(\vec{F}^T_{,\vec{a}\vec{p}^T}\vec{F})_{kj}$, to be interpreted as a matrix in which the r c element equals $\sum_l \left(\frac{\partial^2 F_l}{\partial a_r \partial p_c} F_l\right)$. The term F_l is the l row of $\vec{F}$, α_r is the r row of $\vec{\alpha}$, and p_c is the c row of $\vec{C}$.

It is possible to arrange equations 37 into the following matrix form:

$$-\begin{bmatrix} \vec{\gamma} \\ \begin{bmatrix} \vdots \\ \vec{r}_{j-1} \\ \vec{r}_{j} \\ \vec{r}_{j+1} \\ \vdots \end{bmatrix} \\ \begin{bmatrix} \vdots \\ \vec{\Delta}_{j-1} \\ \vec{\Delta}_{j} \\ \vec{\Delta}_{j+1} \\ \vdots \end{bmatrix} \end{bmatrix}_k = \begin{bmatrix} \sum_{i=1}^{m} \left(\vec{F'}_{,\vec{p}\vec{p}}\right)_i & \cdots & X & \vec{F'}_{,\vec{p}\vec{a}j} & X & \cdots & X & \vec{G'}_{,\vec{p}j} & X & \cdots \\ \vdots & . & \vdots & \vdots & \vdots & . & \vdots & \vdots & \vdots & . \\ X & \cdots & X & 0 & 0 & \cdots & X & 0 & 0 & \cdots \\ \vec{F'}_{,\vec{a}\vec{p}j} & \cdots & 0 & \vec{F'}_{,\vec{a}\vec{a}j} & 0 & \cdots & 0 & \vec{G'}_{,\vec{a}j} & 0 & \cdots \\ X & \cdots & 0 & 0 & X & \cdots & 0 & 0 & X & \cdots \\ \vdots & . & \vdots & \vdots & \vdots & . & \vdots & \vdots & \vdots & . \\ X & \cdots & X & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \\ \vec{G'}_{,\vec{p}j} & \cdots & 0 & \vec{G'}_{,\vec{a}j} & 0 & \cdots & 0 & 0 & 0 & \cdots \\ X & \cdots & 0 & 0 & X & \cdots & 0 & 0 & 0 & \cdots \\ \vdots & . & \vdots & \vdots & \vdots & . & \vdots & \vdots & \vdots & \cdots \end{bmatrix}_k$$

$$\begin{bmatrix} \left(\vec{C}_{k+1} - \vec{C}_k\right) \\ \begin{bmatrix} \vdots \\ \left(\vec{\alpha}_{(k+1)(j-1)} - \vec{\alpha}_{(k)(j-1)}\right) \\ \left(\vec{\alpha}_{(k+1)(j)} - \vec{\alpha}_{(k)(j)}\right) \\ \left(\vec{\alpha}_{(k+1)(j+1)} - \vec{\alpha}_{(k)(j+1)}\right) \\ \vdots \end{bmatrix} \\ \begin{bmatrix} \vdots \\ \left(\vec{\lambda}_{(k+1)(j-1)} - \vec{\lambda}_{(k)(j-1)}\right) \\ \left(\vec{\lambda}_{(k+1)(j)} - \vec{\lambda}_{(k)(j)}\right) \\ \left(\vec{\lambda}_{(k+1)(j+1)} - \vec{\lambda}_{(k)(j+1)}\right) \\ \vdots \end{bmatrix} \end{bmatrix} \tag{38}$$

Here X implies a nonzero matrix, and $\left(\vec{F'}_{,\vec{p}\vec{p}}\right)_i = \left(\vec{F}^T_{,\vec{p}\vec{p}^T}\vec{F} + \vec{F}^T_{,\vec{p}}\vec{F}_{,\vec{p}^T} + \vec{G}^T_{,\vec{p}\vec{p}^T}\vec{\lambda}\right)_i$. Other terms are defined similarly. The square block symmetric matrix in equation 38 differs from the Jacobian matrices used in the calibration of open loop manipulators [7,10,11,22]. Even if one throws out all terms involving the constraint equations, the bilinear terms remain. These bilinear terms do not appear in standard open-loop calibration Jacobians. Since the purpose of the Jacobian is simply to predict a new estimate for the unknowns, it need not be exactly correct. Note that the square matrix in equation 38 is the Hessian matrix for the original objective function. As such, it plays an important role in the study of the uniqueness of the critical point. When interpreted as the Hessian matrix, it is imperative to include the bilinear terms. Since this study simply identifies unknowns producing a critical point, the bilinear terms are discarded to reduce unnecessary numerical difficulty.

2.3.4 Model Completeness

It can be shown [21] that the correct values of $\vec{\lambda}$ are computed from:

$$\vec{\lambda}^T = -\frac{\partial f'}{\partial \vec{x}^T}\left(\frac{\partial \vec{G}}{\partial \vec{x}^T}\right)^{-1} \tag{39}$$

where $\vec{x}$ represents the vector of constrained unknowns of the problem. Note that $\vec{x}$ contains $\vec{\alpha}$ and some $\vec{C}$. The inverted matrix in equation 39 will be nonsingular if the constraint equations ($\vec{G}$) can be used to uniquely determine the constrained quantities. Unfortunately, this is not always the case for calibration. For example, consider the manipulator with a single closed-loop planar four bar mechanism shown in figure 2. If the manipulator is modelled with complete three-dimensional transformations throughout, there must be six constraint equations (three position and three orientation) to guarantee that the single loop remains closed. With the formulation used here, $6m$ Lagrange multipliers would be introduced. Since there are only three unknown variables in the loop, there will be only $3m+C$ independent constraints, where C is some constant. We know C must be constant because there is a finite number of design variables in any mechanism [18]. This implies that potentially not all the $6m$ Lagrange multipliers can be uniquely determined. This manifests itself in the fact that $\left(\frac{\partial \vec{G}}{\partial \vec{x}^T}\right)$ is singular. In any case, an optimal solution can be obtained, but the solution algorithm must deal with singular matrices.

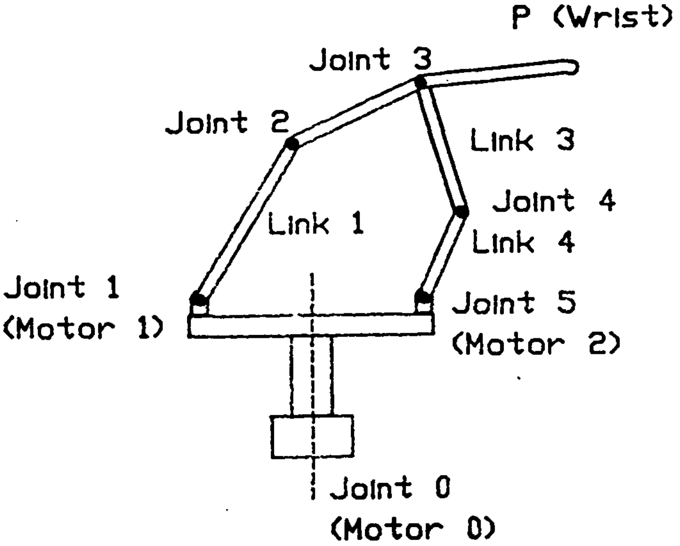

Figure 3: Robot Used in the Example.

2.3.5 Example

The algorithm described was used to identify a manipulator with a five-bar actuated joint (Figure 3). Links one and four are inputs driven by two motors fixed to the base. For simplicity, the closed-loop mechanism is assumed to cause only a two dimensional motion at point P on the tip of the arm. The manipulator can move in three dimensions with a rotation axis on joint 0.

The open-loop kinematic equation is constructed of transformations from the base to joint one, then to joints two and three, and finally to point P. The closed-loop equation can be expressed as a product of transforms from joint three to joints four, five, one, two, and finally back to joint three. These are expressed as:

$$\begin{aligned} T_o &= A_0 A_1 A_2 A'_3 \\ T_c &= A_3 A_4 A_5 A_1 A_2 \end{aligned} \tag{40}$$

The unknown variables in this problem are the angles between links one and two, two and three, and three and four. The measurable angles are assumed to be about joints zero, one and five. Since the closed-loop produces only planar motion, it is possible to reduce the number of constraints from six to three, thereby requiring only three Lagrange multipliers per measurement. This simplifies the problem tremendously and avoids the problems of a singular Jacobian matrix. The unknowns for this problem are the constant parameters in the A matrices, plus three Lagrange multipliers and three unknown angles per measurement.

The solution process begins with selection of values for the unknown A matrices. This is equivalent to choosing the kinematics of the manipulator. The values chosen (hence the correct calibration solutions) are:

$$A_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\theta_0+180) & -\mathrm{S}(\theta_0+180) & 0 & 0 \\ \mathrm{S}(\theta_0+180) & \mathrm{C}(\theta_0+180) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 4.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-90) & -\mathrm{S}(-90) & 0 \\ 0 & \mathrm{S}(-90) & \mathrm{C}(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(180) & -\mathrm{S}(180) & 0 & 0 \\ \mathrm{S}(180) & \mathrm{C}(180) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{41}$$

$$A_1 = \begin{bmatrix} \mathrm{C}(\theta_1) & -\mathrm{S}(\theta_1) & 0 & 0 \\ \mathrm{S}(\theta_1) & \mathrm{C}(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{42}$$

$$A_2 = \begin{bmatrix} \mathrm{C}(\alpha_1) & -\mathrm{S}(\alpha_1) & 0 & 0 \\ \mathrm{S}(\alpha_1) & \mathrm{C}(\alpha_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{43}$$

$$A_3 = \begin{bmatrix} \mathrm{C}(\alpha_2) & -\mathrm{S}(\alpha_2) & 0 & 0 \\ \mathrm{S}(\alpha_2) & \mathrm{C}(\alpha_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{44}$$

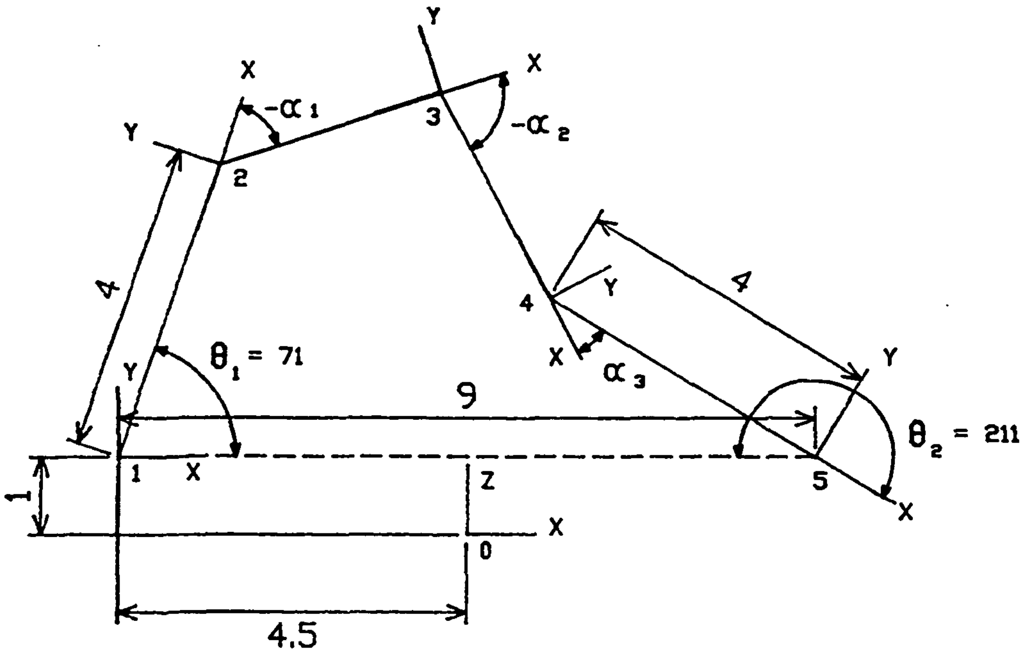

Figure 4: Coordinate System Placement for One Set of Input Data.

$$A_4 = \begin{bmatrix} C(\alpha_3) & -S(\alpha_3) & 0 & 0 \\ S(\alpha_3) & C(\alpha_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{45}$$

$$A_5 = \begin{bmatrix} C(\theta_2) & -S(\theta_2) & 0 & 0 \\ S(\theta_2) & C(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 9 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(180) & -S(180) & 0 & 0 \\ S(180) & C(180) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{46}$$

$$A'_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{47}$$

Figure 4 shows the appearance of the coordinate systems for one set of input data. A forward kinematic solution is performed using the correct matrices to calculate the relation between input joint values and the position of point P. The input angles chosen and the corresponding positions of P are:

Measurement No.	θ_0	θ_1	θ_2	P_x	P_y	P_z
$i = 1$	0	71	211	5.365	0	7.552
2	10	74	214	5.146	0.907	7.4215
3	20	77	217	4.809	1.750	7.135
4	30	68	220	4.415	2.549	8.633
5	40	65	205	4.390	3.684	7.463
6	50	60	200	3.923	4.676	7.106
7	60	57	197	3.171	5.492	6.793
8	70	50	190	2.357	6.476	5.853
9	80	45	185	1.257	7.132	5.076
10	90	40	180	0	7.536	4.283
11	100	89	210	-0.524	2.972	-0.053
12	-10	10	130	8.001	-1.410	-0.664
13	-20	83	224	3.682	-1.340	0.716
14	-30	5	126	7.192	-4.152	-0.443
15	-40	0	120	6.428	-5.393	-0.393

This data is a simulation of the actual measurement process. In an actual calibration, one would position the joints and measure the input angles and the position of the end effector.

With the simulated calibration data, the calibration process is applied. This requires computing the partial derivatives of the open and closed-loop kinematic equations. In addition, the algorithm requires an initial estimate of the unknowns.

The initial estimates are:

$$
\begin{aligned}
A_0 = & \begin{bmatrix} 1 & 0 & 0 & .1 \\ 0 & 1 & 0 & .2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(1) & -S(1) & 0 \\ 0 & S(1) & C(1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(1) & 0 & S(1) & 0 \\ 0 & 1 & 0 & 0 \\ -S(1) & 0 & C(1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
& \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(\theta_0 + 180) & -S(\theta_0 + 180) & 0 & 0 \\ S(\theta_0 + 180) & C(\theta_0 + 180) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 4.8 \\ 0 & 1 & 0 & .2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
& \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(-88) & -S(-88) & 0 \\ 0 & S(-88) & C(-88) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(1) & 0 & S(1) & 0 \\ 0 & 1 & 0 & 0 \\ -S(1) & 0 & C(1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(180) & -S(180) & 0 & 0 \\ S(180) & C(180) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{48}
$$

$$
A_1 = \begin{bmatrix} C(\theta_1 + .2) & -S(\theta_1 + .2) & 0 & 0 \\ S(\theta_1 + .2) & C(\theta_1 + .2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 4.2 \\ 0 & 1 & 0 & .1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{49}
$$

$$
A_2 = \begin{bmatrix} C(\alpha_1) & -S(\alpha_1) & 0 & 0 \\ S(\alpha_1) & C(\alpha_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 3.2 \\ 0 & 1 & 0 & .1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{50}
$$

$$
A_3 = \begin{bmatrix} C(\alpha_2) & -S(\alpha_2) & 0 & 0 \\ S(\alpha_2) & C(\alpha_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 3.2 \\ 0 & 1 & 0 & .2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{51}
$$

$$
A_4 = \begin{bmatrix} C(\alpha_3) & -S(\alpha_3) & 0 & 0 \\ S(\alpha_3) & C(\alpha_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 4.2 \\ 0 & 1 & 0 & .2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(1) & -S(1) & 0 & 0 \\ S(1) & C(1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{52}
$$

$$
A_5 = \begin{bmatrix} C(\theta_2 + .1) & -S(\theta_2 + .1) & 0 & 0 \\ S(\theta_2 + .1) & C(\theta_2 + .1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 & 8.7 \\ 0 & 1 & 0 & .2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(180) & -S(180) & 0 & 0 \\ S(180) & C(180) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{53}
$$

$$A'_{3} = \begin{bmatrix} C(2) & -S(2) & 0 & 0 \\ S(2) & C(2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & .1 \\ 0 & 1 & 0 & .2 \\ 0 & 0 & 1 & .2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(2) & -S(2) & 0 \\ 0 & S(2) & C(2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(3) & 0 & S(3) & 0 \\ 0 & 1 & 0 & 0 \\ -S(3) & 0 & C(3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{54}$$

With this data, the algorithm was used to estimate the unknowns. After nine iterations, the algorithm produced an objective function of the order 10^{-14} and all Lagrange multipliers were computed as zero.

2.3.6 Conclusions

An algorithm has been presented that can kinematically calibrate closed-loop manipulators. The algorithm is equivalent to minimizing a constrained objective function. The chosen objective function was the error between the measured end effector pose and the kinematically calculated pose. Error was expressed as the Euclidean norm of the difference in pose integrated over the entire workspace. For practical reasons, the integration was reduced to a discrete sum.

Constraint equations arise because the closed-loop must remain closed for all joint configurations. The constraints appear as nonlinear algebraic restrictions on the kinematic equations. In general, there are six constraints per closed-loop. For practical reasons, the constraints are approximated as individual constraint equations applied to discrete joint configurations. This causes the number of constraint equations to equal six times the number of measurements used in the calibration.

Since the constraint equations cannot be easily inverted, the method of Lagrange multipliers was used. By modifying the objective function, the constrained optimization is treated as a unconstrained problem. The disadvantage of this is that the number of unknowns increases.

The method was successfully applied to two and three dimensional manipulators, a simple five-bar actuated joint mechanism was included as an example. Although the example demonstrated that the technique has application, more research is needed. Problems to be investigated include those associated with very large optimization problems, determining the required number of parameters, and dealing with internal forces. In addition, more research is needed to better understand the singular conditions arising from planar and nearly planar mechanisms.

2.4 A Study of Calibration Models

Many of the publications dealing with calibration introduce a *new* kinematic model. Because results differ, conclusions concerning the utility of the models are difficult to formulate. This part of the report formalizes model differences thereby enabling model comparison.

A manipulator is considered to be a system consisting $n+1$ rigid bodies. One end of the manipulator is the world, and is called body 1; the tool is numbered $n+1$ and intermediate bodies are numbered sequentially with body 2 connected to the world via joint 1 see figure 5. Manipulators are characterized by a set of constants called parameters, which typically consist of rigid body transformations. Parameters may also be specified by screw operators [7]. The jth parameter of a manipulator whether it is a rotation or translation is denoted C_j. The complete set of parameters is referred to as vector $\overline{C}$. A specific manipulator is characterized by listing a set of parameters as in $\overline{C} = (C_1, \ldots, C_p)$ where p is the number of parameters. Since the manipulator's joints are lower pair, joint i, lying between bodies i and $i+1$, moves along or rotates about a single line specified with unit vector $\vec{n}_i$. The set of joint variables is referenced as $\overline{\Theta}$. A forward kinematic model, referred to as the function $F[\overline{C}, \overline{\Theta}] = F[(C_1, \ldots, C_p), \overline{\Theta}]$, calculates the pose of a tool held by a manipulator.

Models of manipulators can differ. These differences are either structural or numerical. Structural differences are the result of a different number, type, or order of parameters. For example, a model with a rotation parameter followed by a translation is structurally different from a model with a translation followed by a rotation. Numerical differences arise from using different numerical values for the parameters. We emphasize that the models $F_s[\overline{C}_s, \overline{\Theta}]$ and $F_p[\overline{C}_p, \overline{\Theta}]$ are structurally

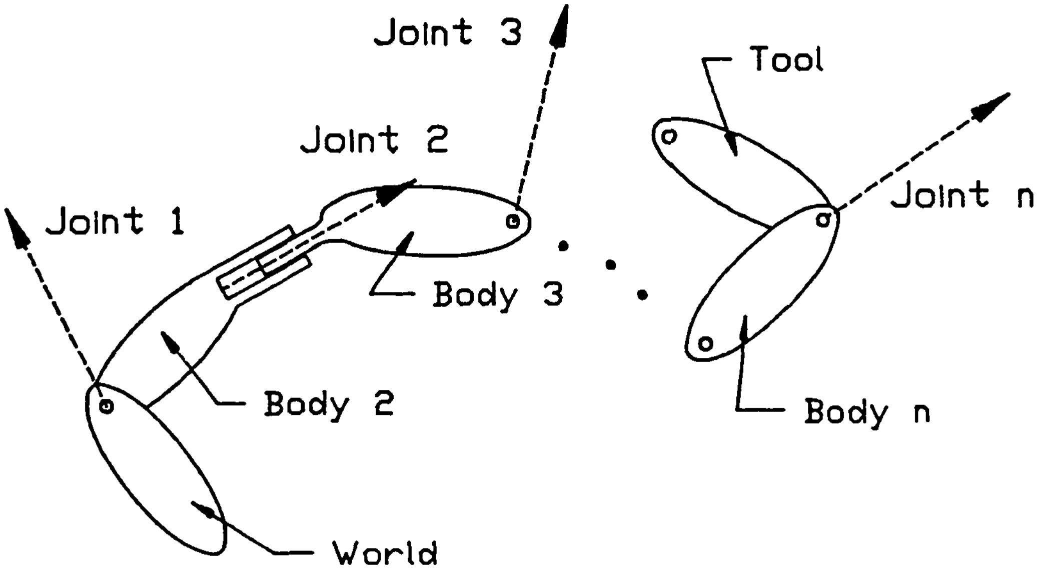

Figure 5: Body and Joint Numbering Scheme.

different by the subscripts s and p. Only structural differences are studied.

Two manipulators are identical if they pose a tool exactly alike for identical joint variables. A model exactly matches a manipulator if it exactly predicts the pose of a tool held by the manipulator for all values of the joint variables. Two manipulators $F_1[\overline{C}_1,\overline{\Theta}]$ and $F_2[\overline{C}_2,\overline{\Theta}]$ are called nearby if $\|F_1[\overline{C}_1,\overline{\Theta}]-F_2[\overline{C}_2,\overline{\Theta}]\|$ is small for all $\overline{\Theta}$.

2.4.1 Redundant Parameters

All parameters for a particular model structure can be classified as either redundant or essential. A parameter C_j is redundant relative to $\overline{C}=(C_1,\ldots,C_j,\ldots,C_p)$ if there are numbers $(C_1^*,\ldots,C_p^*)$ and a positive number δ such that:

$$F_s[(C_1,\ldots,C_j,\ldots,C_p),\overline{\Theta}] = F_s[(C_1^*,\ldots,C_j+\epsilon_j,\ldots,C_p^*),\overline{\Theta}] \qquad (55)$$

for any $\epsilon_j < \delta$ for all $\overline{\Theta}$. Essential parameters include all parameters that are not redundant. It is easy to show that if parameter C_j is redundant relative to $\overline{C}$, it is also redundant relative to $\overline{C}^*$, where $\overline{C}$ and $\overline{C}^*$ are defined in 55. Redundancy must be expressed as relative to a set of parameters because it is possible for a parameter to be redundant relative to one set of parameters, and essential relative to another.

The significance of essential parameters is that all such parameters in a model must be known precisely. To demonstrate this consider the following. Suppose a robot is modelled exactly as

$$F_s[(C_1,\ldots,C_j,\ldots,C_p),\overline{\Theta}] = F_s[\overline{C},\overline{\Theta}]$$

and we wish to calibrate a model. Beginning with a nearby model with the same structure

$$F_s[(C_1',\ldots,C_j+\epsilon_j,\ldots,C_p'),\overline{\Theta}]$$

we modify all parameters but C_j in an attempt to make our model match the manipulator. After this process our model appears as:

$$F_s[(C_1^*,\ldots,C_j+\epsilon_j,\ldots,C_p^*),\overline{\Theta}]$$

By definition, if it is possible that:

$$F_s[(C_1,\ldots,C_j,\ldots,C_p),\overline{\Theta}] = F_s[(C_1^*,\ldots,C_j+\epsilon_j,\ldots,C_p^*),\overline{\Theta}]$$

parameter C_j must be redundant relative to the manipulator $F_s[\overline{C},\overline{\Theta}]$. If not, the equality cannot exist and our result cannot model the manipulator exactly.

2.4.2 Testing for Redundant Parameters

To develop a necessary and sufficient condition for redundant parameters, consider two nearby models given by $F_s[\overline{C}+\overline{\epsilon},\overline{\Theta}] \approx F_s[\overline{C},\overline{\Theta}]$. Since forward kinematic models contain parameters appearing alone or as arguments in forward trigonometric functions, they are everywhere continuous with respect to the parameters. Because they are continuous, it is possible to expand one model in a Taylor series. After expansion, the two nearby models are related by:

$$\begin{aligned} F_s[\overline{C}+\overline{\epsilon},\overline{\Theta}] &\equiv F_s[C_1+\epsilon_1,\ldots,C_j+\epsilon_j,\ldots,C_p+\epsilon_p,\overline{\Theta}] \\ &\approx F_s[C_1,\ldots,C_j,\ldots,C_p,\overline{\Theta}]+\textstyle\sum_{k=1}^{p}\left\{\left[\frac{\partial F_s}{\partial C_k}\Big|_{\overline{C},\overline{\Theta}}\right]\epsilon_k\right\}+\cdots \end{aligned} \tag{56}$$

Rewriting and keeping only the first order terms, we obtain:

$$F_s[\overline{C}+\overline{\epsilon},\overline{\Theta}]-F_s[\overline{C},\overline{\Theta}]\approx\sum_{k=1}^{p}\left\{\left[\frac{\partial F_s}{\partial C_k}\bigg|_{\overline{C},\overline{\Theta}}\right]\epsilon_k\right\} \tag{57}$$

If there are redundant parameters relative to $\overline{C}$, it is possible that the two nearby models are exactly the same. If so

$$[0]=\sum_{k=1}^{p}\left\{\left[\frac{\partial F_s}{\partial C_k}\bigg|_{\overline{C},\overline{\Theta}}\right]\epsilon_k\right\} \tag{58}$$

is satisfied for all choice of $\overline{\Theta}$. Although $\frac{\partial F_s}{\partial \overline{C}k}$ represents a 4 by 4 matrix, it is possible to rewrite equation 58 as a 16 row by p matrix times the p by 1 vector $\overline{\epsilon}$ by forming a row in the equation for each term in the original 4 by 4 matrix. Although it is impossible to express equation 58 for all possible values of $\overline{\Theta}$, it is possible to construct several equations using different values of $\overline{\Theta}$. If this is done for m values of the joint variables, the equation becomes:

$$[0]=\begin{bmatrix} \frac{\partial F_s}{\partial C_k}\Big|_{\overline{C},\overline{\Theta}_1} \\ \vdots \\ \frac{\partial F_s}{\partial C_k}\Big|_{\overline{C},\overline{\Theta}_m} \end{bmatrix}\overline{\epsilon}=[J]\overline{\epsilon} \tag{59}$$

In this equation, $[J]$ is $16m$ by p. If there are redundant parameters then there will be a nontrivial solution for $\overline{\epsilon}$ and $[J]$ must have rank less than p. Since all nontrivial solutions lie in the null space of $[J]$, the number of redundant parameters equals the rank of the null space of $[J]$. Conversely, the number of essential parameters is equal to the rank of $[J]$. This test is only sufficient because one can never express equation 59 for all possible values of $\overline{\Theta}$.

2.4.3 Basis Independent Calibration Model

To demonstrate that equation 59 has a maximum rank regardless of the model structure used, review equation 57. Divide equation 57 by a small pseudo time increment Δt and take the limit as Δt approaches zero. In the limit, equation 57 represents the derivative of $F[\overline{C},\overline{\Theta}]$ with respect to t. If the parameters $\overline{C}$ are considered functions of t, equations 57 and 59 can be written as:

$$[J]\frac{\overline{\epsilon}}{\Delta t} = [J]\dot{\overline{C}} = \frac{dF[\overline{C},\overline{\Theta}]}{dt} \tag{60}$$

Physically, equation 60 is a velocity equation for a large degree of freedom system in which the model parameters $\overline{C}$ are functions of time and contribute to the motion of the tool. Note that equation 60 differs from the conventional velocity of the manipulator since conventional velocity is the time derivative with respect to the joint variables. To avoid confusion, equation 60 is called the "extended velocity" equation. Also note that $\frac{\overline{\epsilon}}{\Delta t} = \dot{\overline{C}}$ is constant relative to joint variables $\overline{\Theta}$.

Equation 60 is rewritten using vector mechanics since velocities can be expressed as vectors. For example, the time derivative of the ith rotational parameter which is applied in the jth body of the model structure (after the application of joint variable $j-1$ but before joint variable j) is expressed as $\vec{\omega}_j^i$. Derivatives of the ith distance parameter applied in the jth body is expressed as $\vec{v}_j^i$. The sum of $\vec{v}_j^i$ over i is represented as $\vec{V}_j$. The quantity O_j represents the number of $\vec{\omega}_j^i$ terms located in body j.

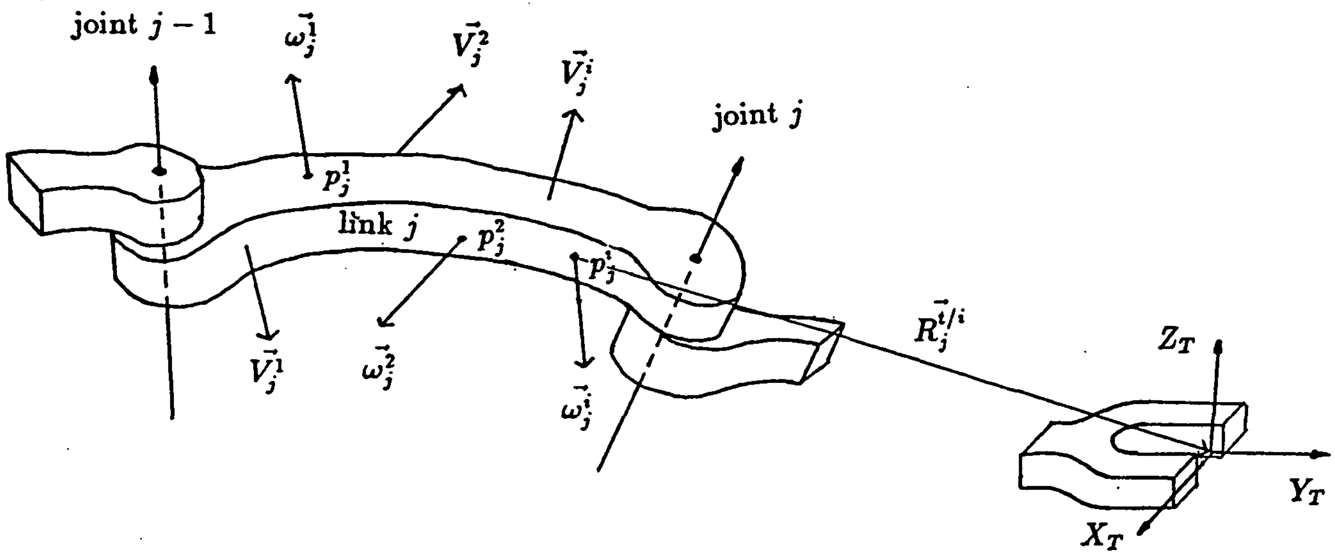

Figure 6: Vector Notation For Manipulator Models.

With standard kinematic formula, the extended angular and linear velocities of the tool for a single joint configuration can be written as:

$$\begin{aligned} {}^w\vec{\Omega}^E &= \textstyle\sum_{j=1}^{n+1} \left(\sum_{i=1}^{O_j} \vec{\omega}_j^i\right) = \sum_{j=1}^{n+1} \vec{\Omega}_j \\ {}^w\vec{V}^E &= \textstyle\sum_{j=1}^{n+1} \left\{\vec{V}_j + \sum_{i=1}^{O_j} \left[\vec{\omega}_j^i \times \left(\vec{r}^{\,t/Q_{j+1}} + \vec{r}^{\,Q_j/P_j^i}\right)\right]\right\} \end{aligned} \tag{61}$$

E denotes a coordinate system fixed in the tool (end effector) that represents the Cartesian position of the manipulator. P_j^i represents a point on the line of action of $\vec{\omega}_j^i$, and hence is fixed in body j, see figure 6. Q_j is any point on the (j-1)st joint after the joint variable; hence Q_j is fixed in body j. The vector $\vec{r}^{\,A/B}$ is the position of point A with respect to point B. Point Q_j^* is coincident with point Q_j when the manipulator is in a zero position and is fixed to body j-1. Points Q_j and Q_j^* differ only when joint j is prismatic and the joint is displaced. Point t is located at the origin of E.

Equation 61 is representative of equation 59 for a single joint configuration and any conceivable kinematic model, including those based on screw theory [7].

Since the changes in parameters represent infinitesimal angular motions, they can be treated as vectors and the order of their application is irrelevant. Consider the special case of a Hartenberg-Denavit [17] model. The quantity $\vec{\Omega}_j$ equals $\Delta\theta\vec{z}+\Delta\alpha\vec{x}$ and $\vec{V}_j$ equals $\Delta l\vec{z}+\Delta r\vec{x}$, where $\vec{z}$ is a unit vector along the motion axis between bodies $j-1$ and j, and the unit vector $\vec{x}$ points along the x axis.

2.4.4 Essential Model Parameters

Equation 61 is expressed for general joint configurations and analyzed to determine the maximum possible rank of the Jacobian matrix. The analysis consists of performing row and column operations on the vectors in the equation and therefore it is necessary to evaluate the vectors in a common coordinate system. The end effector coordinate system is arbitrarily chosen for the evaluation of these vector quantities.

Consider only the jth terms in equation 61:

$$\begin{aligned} {}^{w}\vec{\Omega}^{E} &= \cdots + \textstyle\sum_{i=1}^{O_j} \vec{\omega}_j^i + \cdots \equiv \cdots + \vec{\Omega}_j + \cdots \\ {}^{w}\vec{V}^{E} &= \cdots + \vec{V}_j + \textstyle\sum_{i=1}^{O_j} \left[\vec{\omega}_j^i \times \left(\vec{r}^{\,t/Q_{j+1}} + \vec{r}^{\,Q_j/P_j^i}\right)\right] + \cdots \end{aligned} \tag{62}$$

Now place the manipulator in a new joint configuration by moving or rotating every joint relative to its current position. For example, if joint l is revolute, then it is rotated θ_l^k degrees from its position corresponding to equation 62. When joint l rotates, all vectors appearing in the model structure after joint ls motion appear unchanged in the end effector frame, but vectors applied before joint l appear to rotate an amount $-\theta_l^k$ about joint ls rotation axis ($\vec{n}_l$) (see figure 7). This vector rotation is denoted by a rotation operator $R_{(\vec{n}_l,-\theta_l^k)}$. If joint l is prismatic, motion of d_l^k units changes only some position vectors. Prismatic joint motion is depicted in figure 8 and denoted by the operator $D_{(\vec{n}_l,-d_l^k)}$.

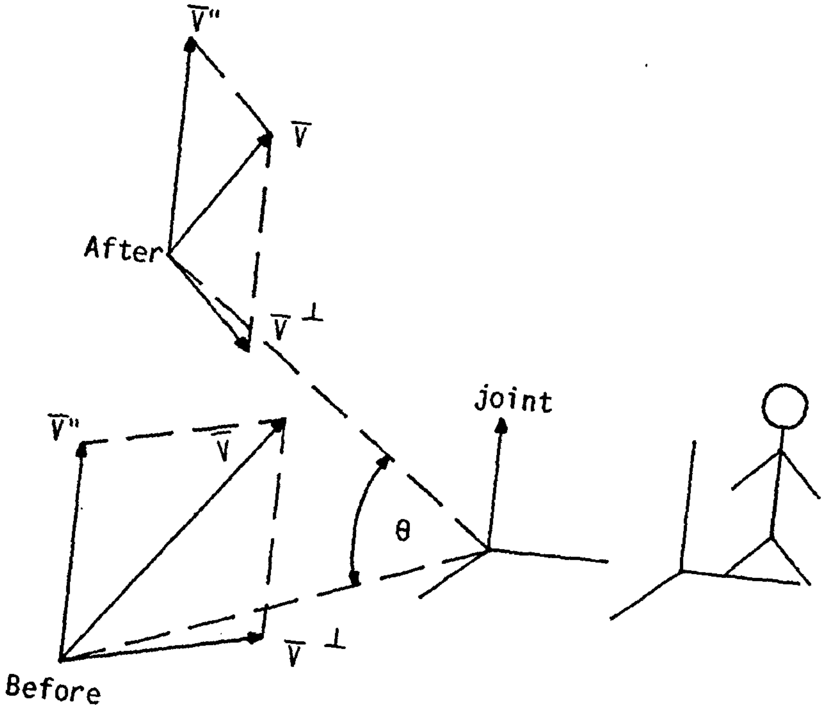

Figure 7: Appearance of Vectors Before and After Rotation.

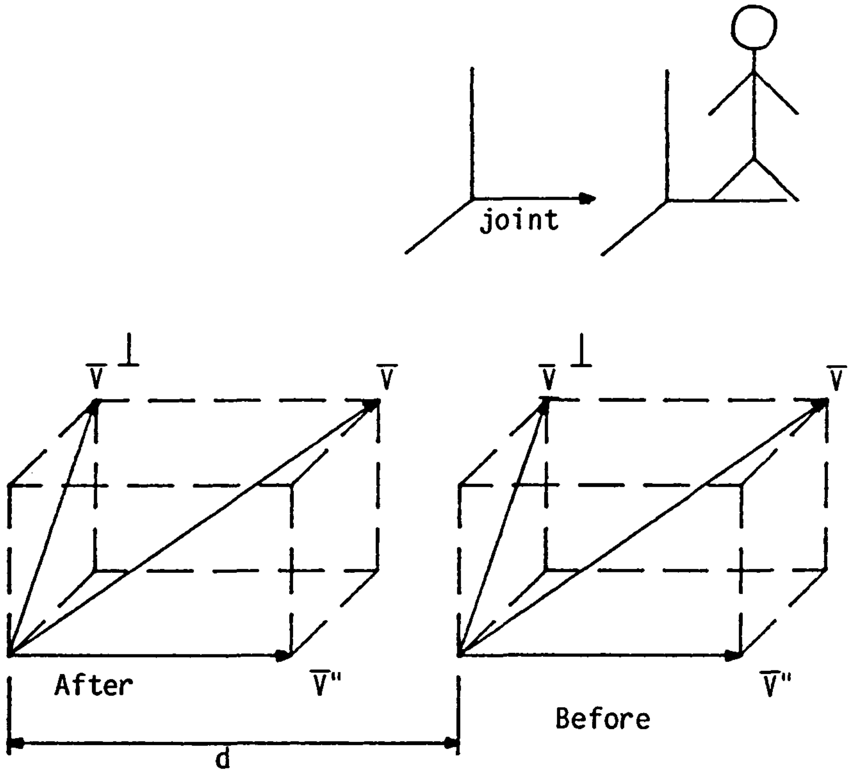

Figure 8: Appearance of Vectors Before and After Translation.

2.4.5 The Special Case When Joint j is Revolute

When all n joints of the manipulator change into the kth joint configuration, the jth terms from equation 61 appear as:

$$\begin{gathered} {}^{w}\vec{\Omega}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_j,-\theta_j^k)} \vec{\Omega}_j + \cdots \\ {}^{w}\vec{V}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_j,-\theta_j^k)} \left\{ \vec{V}_j + \sum_{i=1}^{O_j} \left[\vec{\omega}_j^i \times \left(\vec{r}^{\,t/Q_{j+1}} + \vec{r}^{\,Q_j/P_j^i} \right) \right] \right\} + \cdots \end{gathered} \tag{63}$$

This assumes that the jth joint is revolute; another analysis follows demonstrating what happens when joint j is prismatic. By recognizing that the position vector $\vec{r}^{\,t/Q_{j+1}}$ is unchanged by joint j motion, equation 63 becomes:

$$\begin{gathered} {}^{w}\vec{\Omega}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_j,-\theta_j^k)} \vec{\Omega}_j + \cdots \\ {}^{w}\vec{V}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_j,-\theta_j^k)} \left[\vec{V}_j + \sum_{i=1}^{O_j} \left(\vec{\omega}_j^i \times \vec{r}^{\,Q_j/P_j^i} \right) \right] + \\ \left[R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \left(R_{(\vec{n}_j,-\theta_j^k)} \vec{\Omega}_j \right) \times \vec{r}^{\,t/Q_{j+1}} \right] + \cdots \end{gathered} \tag{64}$$

The term $\left[\vec{V}_j + \sum_{i=1}^{O_j} \left(\vec{\omega}_j^i \times \vec{r}^{\,Q_j/P_j^i} \right) \right]$ can be lumped into a single constant vector called $\vec{V}_j'$.

Denoting by $\overline{A}^{\perp}$ and $\overline{A}^{||}$, components of vector $\overline{A}$ that are respectively perpendicular and parallel to unit vector $\vec{n}_j$, expression 64 can be written as:

$$\begin{gathered} {}^{w}\vec{\Omega}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_j,-\theta_j^k)} \vec{\Omega}_j^{\perp} + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \vec{\Omega}_j^{||} + \cdots \\ {}^{w}\vec{V}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_j,-\theta_j^k)} \left[\left(\vec{V}_j' \right)^{\perp} \right] + \\ \left[R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \left(R_{(\vec{n}_j,-\theta_j^k)} \vec{\Omega}_j^{\perp} \right) \times \vec{r}^{\,t/Q_{j+1}} \right] + \\ R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \left[\left(\vec{V}_j' \right)^{||} + \vec{\Omega}_j^{||} \times \vec{r}^{\,Q_{j+2}/Q_{j+1}} \right] + \\ R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+2},-\theta_{j+2}^k)} \left[\left(R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \vec{\Omega}_j^{||} \right) \times \vec{r}^{\,t/Q_{j+2}} \right] + \cdots \end{gathered} \tag{65}$$

It is important to observe the $j+1$st terms from equation 61, which are similar to the jth terms of expression 64. When added to equation 65 the following results:

$$
\begin{gathered}
{}^{w}\vec{\Omega}^{E} = \cdots + R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_j,-\theta^k_j)} \vec{\Omega}^{\perp}_{j} + R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \vec{\Omega}^{\|}_{j} + \\
R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \vec{\Omega}_{j+1} + \cdots
\end{gathered}
$$

$$
\begin{gathered}
{}^{w}\vec{V}^{E} = \cdots + R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_j,-\theta^k_j)} \left[\left(\vec{V}'_j \right)^{\perp} \right] + \\
\left[R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left(R_{(\vec{n}_j,-\theta^k_j)} \vec{\Omega}^{\perp}_{j} \right) \times \vec{r}^{\,t/Q_{j+1}} \right] + \\
R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left[\left(\vec{V}'_j \right)^{\|} + \vec{\Omega}^{\|}_{j} \times \vec{r}^{\,Q_{j+2}/Q_{j+1}} \right] + \\
R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+2},-\theta^k_{j+2})} \left[R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left(\vec{\Omega}^{\|}_{j} \right) \times \vec{r}^{\,t/Q_{j+2}} \right] + \\
R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left[\vec{V}_{j+1} + \textstyle\sum_{i=1}^{O_{j+1}} \left(\vec{\omega}^{i}_{j+1} \times \vec{r}^{\,Q_{j+1}/P^i_{j+1}} \right) \right] + \\
\left[R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+2},-\theta^k_{j+2})} \left(R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \vec{\Omega}_{j+1} \right) \times \vec{r}^{\,t/Q_{j+2}} \right] + \cdots
\end{gathered}
\tag{66}
$$

By combining terms with identical rotation operators, equation 66 can be expressed as:

$$
{}^{w}\vec{\Omega}^{E} = \cdots + R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_j,-\theta^k_j)} \vec{\Omega}^{\perp}_{j} + R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left(\vec{\Omega}^{\|}_{j} + \vec{\Omega}_{j+1} \right) + \cdots
$$

$$
\begin{gathered}
{}^{w}\vec{V}^{E} = \cdots + R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_j,-\theta^k_j)} \left[\left(\vec{V}'_j \right)^{\perp} \right] + \\
\left[R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left(R_{(\vec{n}_j,-\theta^k_j)} \vec{\Omega}^{\perp}_{j} \right) \times \vec{r}^{\,t/Q_{j+1}} \right] + \\
R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left[\left(\vec{V}'_j \right)^{\|} + \vec{V}_{j+1} + \right. \\
\left. \textstyle\sum_{i=1}^{O_{j+1}} \left(\vec{\omega}^{i}_{j+1} \times \vec{r}^{\,Q_{j+1}/P^i_{j+1}} \right) + \vec{\Omega}^{\|}_{j} \times \vec{r}^{\,Q_{j+2}/Q_{j+1}} \right] + \\
\left\{ R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+2},-\theta^k_{j+2})} \left[R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left(\vec{\Omega}^{\|}_{j} + \vec{\Omega}_{j+1} \right) \right] \times \vec{r}^{\,t/Q_{j+2}} \right\} + \cdots
\end{gathered}
\tag{67}
$$

Since the vectors $\vec{\Omega}^{\|}_{j}$ and $\vec{\Omega}_{j+1}$ are constant, their sum in the first line of expression 67 can be replaced with another constant vector $\vec{\Omega}'_{j+1}$. Similarly, through proper definition of other constant vectors, expression 67 can be reduced to:

$$
{}^{w}\vec{\Omega}^{E} = \cdots + R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_j,-\theta^k_j)} \vec{\Omega}^{\perp}_{j} + R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \vec{\Omega}'_{j+1} + \cdots
$$

$$
\begin{gathered}
{}^{w}\vec{V}^{E} = \cdots + R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_j,-\theta^k_j)} \left[\left(\vec{V}'_j \right)^{\perp} \right] + \\
\left[R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left(R_{(\vec{n}_j,-\theta^k_j)} \vec{\Omega}^{\perp}_{j} \right) \times \vec{r}^{\,t/Q_{j+1}} \right] + \\
R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \left(\vec{V}'_{j+1} \right) + \\
\left[R_{(\vec{n}_n,-\theta^k_n)} \cdots R_{(\vec{n}_{j+2},-\theta^k_{j+2})} \left(R_{(\vec{n}_{j+1},-\theta^k_{j+1})} \vec{\Omega}'_{j+1} \right) \times \vec{r}^{\,t/Q_{j+2}} \right] + \cdots
\end{gathered}
\tag{68}
$$

Equation 68 is the exact form as the jth and $j+1$st terms would appear in equation 64. This series of manipulations shows that the jth kinematic parameters can be decomposed into components parallel and perpendicular to the jth rotation axis and that the parallel components can be shifted past the rotation axis and combined with the original $j+1$st terms.

2.4.6 The Special Case When Joint j is Prismatic.

When joint j is prismatic, the jth terms appear as:

$$\begin{gathered} {}^{w}\vec{\Omega}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots D_{(\vec{n}_j,-d_j^k)} \vec{\Omega}_j + \cdots \\ {}^{w}\vec{V}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots D_{(\vec{n}_j,-d_j^k)} \left\{ \vec{V}_j + \textstyle\sum_{i=1}^{O_j} \left[\vec{\omega}_j^i \times \left(\vec{r}^{\,t/Q_{j+1}} + \vec{r}^{\,Q_j/P_j^i} \right) \right] \right\} + \cdots \end{gathered} \tag{69}$$

Since the representation of constant vectors $\vec{\omega}_j^i$ and $\vec{V}_j$ is unaffected by prismatic joint motion, these terms can be rewritten as:

$$\begin{gathered} {}^{w}\vec{\Omega}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \vec{\Omega}_j + \cdots \\ {}^{w}\vec{V}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \left(\vec{\Omega}_j^{\perp} \times d_j^k \vec{n}_j \right) + \\ R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \left\{ \vec{V}_j + \textstyle\sum_{i=1}^{O_j} \left[\vec{\omega}_j^i \times \vec{r}^{\,Q_{j+1}^*/P_j^i} \right] + \vec{\Omega}_j \times \vec{r}^{\,t/Q_{j+1}} \right\} + \cdots \end{gathered} \tag{70}$$

Including the $j+1$st terms as in the last section, expression 70 becomes:

$$\begin{gathered} {}^{w}\vec{\Omega}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \left(\vec{\Omega}_j + \vec{\Omega}_{j+1} \right) + \cdots \\ {}^{w}\vec{V}^{E} = \cdots + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \left(\vec{\Omega}_j^{\perp} \times d_j^k \vec{n}_j \right) + \\ R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{j+1},-\theta_{j+1}^k)} \left\{ \vec{V}_j + \vec{V}_{j+1} + \right. \\ \textstyle\sum_{i=1}^{O_j} \left[\vec{\omega}_j^i \times \vec{r}^{\,Q_{j+1}^*/P_j^i} \right] + \sum_{i=1}^{O_{j+1}} \left[\vec{\omega}_{j+1}^i \times \vec{r}^{\,t/P_{j+1}^i} \right] + \\ \left. \vec{\Omega}_j \times \vec{r}^{\,t/Q_{j+1}} \right\} + \cdots \end{gathered} \tag{71}$$

Note that all of the jth terms can be absorbed into the $j+1$st terms with the exception of the single quantity $\left(\vec{\Omega}_j^{\perp} \times d_j^k \vec{n}_j\right)$, which represents the effect of prismatic joint j.

2.4.7 Summary

The operations described above can be performed on all the terms of equation 61 from joint 1 through joint n. When this is done, the equation expressed for the kth joint configuration takes the form:

$$
\begin{gathered}
{}^{w}\vec{\Omega}^{t} = R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_1,-\theta_1^k)} \vec{\Omega}_1^{\perp} + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_2,-\theta_2^k)} \left(\vec{\Omega}_2'\right)^{\perp} + \cdots + \\
R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{l-1},-\theta_{l-1}^k)} \left(\vec{\Omega}_{l-1}'\right)^{\perp} + R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{l+1},-\theta_{l+1}^k)} \left(\vec{\Omega}_{l+1}'\right)^{\perp} + \cdots + \\
R_{(\vec{n}_n,-\theta_n^k)} \left(\vec{\Omega}_n'\right)^{\perp} + \vec{\Omega}_{n+1}'
\end{gathered}
$$

$$
\begin{gathered}
{}^{w}\vec{V}^{t} = R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_1,-\theta_1^k)} \left(\vec{V}_1'\right)^{\perp} + \left\{ R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_2,-\theta_2^k)} \left[R_{(\vec{n}_1,-\theta_1^k)} \left(\vec{\Omega}_1'\right)^{\perp} \right] \times \vec{r}^{\,t/Q_2} \right\} + \\
R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_2,-\theta_2^k)} \left(\vec{V}_2'\right)^{\perp} + \left\{ R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_3,-\theta_3^k)} \left[R_{(\vec{n}_2,-\theta_2^k)} \left(\vec{\Omega}_2'\right)^{\perp} \right] \times \vec{r}^{\,t/Q_3} \right\} + \cdots + \\
R_{(\vec{n}_n,-\theta_n^k)} \cdots R_{(\vec{n}_{l+1},-\theta_{l+1}^k)} \left(\vec{\Omega}_l^{\perp} \times d_l^k \vec{n}_l\right) + \cdots + \\
R_{(\vec{n}_n,-\theta_n^k)} \left(\vec{V}_n'\right)^{\perp} + \left[R_{(\vec{n}_n,-\theta_n^k)} \left(\vec{\Omega}_n'\right)^{\perp} \right] \times \vec{r}^{\,t/Q_{n+1}} + \vec{V}_{n+1}'
\end{gathered}
\tag{72}
$$

Here joint l has been assumed to be prismatic; joints 1, 2, $l-1$, $l+1$, and n are revolute.

2.4.8 Discussion

Equation 72 is a representation of several rows of the Jacobian matrix defined in equation 59 for a single joint configuration. Using equation 72 we wish to determine the maximum possible rank of the Jacobian matrix. Notice that in the first of the two equations in 72 most of the unknowns present are perpendicular components of a vector. Since these perpendicular components are two-dimensional vectors, they total at most twice the number of revolute joints. Also note that vector

$\vec{\Omega}'_{n+1}$ is three-dimensional; hence it can contribute three independent (essential) unknowns. The first equation cannot determine any parameters that appear just before a prismatic joint, since none of the prismatic parameters appear in the first of the two equations. It is also impossible to use any of these equations for determining prismatic parameters. This last fact is consistent with published results [16].

Consider the second of the two equations. Again, most of the parameters appear as perpendicular components of vectors, hence they are two-dimensional. In the latter equation, there are two dimensional vectors $\left(\vec{V}'_j\right)^{\perp}$ and $\left(\vec{\Omega}'_j\right)^{\perp}$ for every revolute joint j. Because position vector $\vec{r}^{\,t/Q_{j+1}}$ changes with joint variables $j+1$ to n, these unknown vectors cannot be combined. As a result, they can produce four independent (essential) parameters per revolute joint. Because $\left(\vec{\Omega}'_j\right)^{\perp}$ appears in the first equation, when the two equations are taken together, the number of independent unknowns remains four per revolute joint. It is especially important that the two-dimensional vector $\left(\vec{\Omega}'_l\right)^{\perp}$ corresponding to prismatic joint l appears only in the second equation; hence it produces two extra independent parameters for every prismatic joint. Also since $\vec{V}'_{n+1}$ is three-dimensional, it may contribute three extra essential parameters. Since $\vec{\Omega}'_{n+1}$ appears only in the first equation it cannot be computed when only the second equation is used.

The outcome of this analysis was predicted by Everett and Hsu [11] with a different approach. They gave no formal proof but suggested an equation that relates the number and type of joints to the number of independent parameters. Since the two findings are mutually supportive, their equation can formalize one result of this study. The maximum number of kinematic parameters that must be identified for any manipulator, regardless of the modelling scheme, is

$$N = 4R + 2P + 6 \tag{73}$$

Equation 73 shows the number of independent parameters N for a robot with R revolute and P prismatic joints.

There are cases where a robot model does not have as many parameters as indicated. One explanation is that the rotation and/or the translation of the joints create dependencies between the otherwise independent parameters. Detailed analysis of these occurrences requires a study of number differences of kinematic models and will be presented in a future work.

One simple example of joint position prohibiting a full ranked Jacobian can be observed from these results. Consider what happens during row operations on the Jacobian matrix, for example the jth terms from equation 72 for joint configurations k and $k+1$. To prepare for the analysis, suppose the kth equation is operated on from the left with $R_{(\vec{n}_{j+1},-\theta_{j+1}^k)}\cdots R_{(\vec{n}_n,-\theta_n^k)}$. The jth and $j+1$st columns appear as:

$$
\begin{gathered}
R_{(\vec{n}_j,-\theta_j^k)}\left(\vec{\Omega}_j'\right)^{\perp} + \left(\vec{\Omega}_{j+1}'\right)^{\perp} \\
R_{(\vec{n}_j,-\theta_j^k)}\left(\vec{V}_j'\right)^{\perp} + \left[R_{(\vec{n}_j,-\theta_j^k)}\left(\vec{\Omega}_j'\right)^{\perp}\right] \times \vec{r}^{\,t/Q_{j+1}} + \left(\vec{V}_{j+1}'\right)^{\perp} + \left(\vec{\Omega}_{j+1}'\right)^{\perp} \times \vec{r}^{\,t/Q_{j+2}}
\end{gathered}
\tag{74}
$$

During solution, to eliminate the $j+1$st column, subtract rows k and $k+1$. The result will be:

$$
\begin{gathered}
R_{(\vec{n}_j,-\theta_j^k)}\left(\vec{\Omega}_j'\right)^{\perp} - R_{(\vec{n}_j,-\theta_j^{k+1})}\left(\vec{\Omega}_j'\right)^{\perp} + 0 \\
R_{(\vec{n}_j,-\theta_j^k)}\left(\vec{V}_j'\right)^{\perp} + \left[R_{(\vec{n}_j,-\theta_j^k)}\left(\vec{\Omega}_j'\right)^{\perp}\right] \times \vec{r}^{\,t/Q_{j+1}} - \\
R_{(\vec{n}_j,-\theta_j^{k+1})}\left(\vec{V}_j'\right)^{\perp} - \left[R_{(\vec{n}_j,-\theta_j^{k+1})}\left(\vec{\Omega}_j'\right)^{\perp}\right] \times \vec{r}^{\,t/Q_{j+1}} + 0
\end{gathered}
\tag{75}
$$

Some of these vectors are depicted in figure 9. If θ_j^k and θ_j^{k+1} are the same, not only the $j+1$st term disappears but also the jth, thereby reducing the rank of the Jacobian matrix. Hence to prevent the jth unknowns from completely disappearing, one must guarantee that the jth joint is rotated. Likewise for prismatic joints; if the

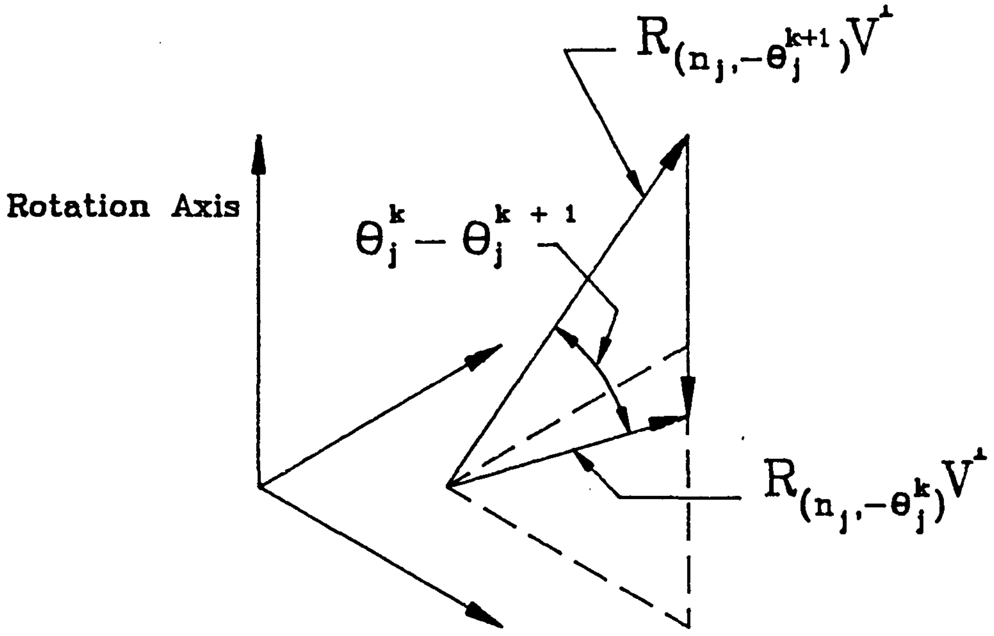
$R_{(n_j,-\theta_j^{k+1})}V^{\perp}$
Rotation Axis
$\theta_j^k - \theta_j^{k+1}$
$R_{(n_j,-\theta_j^k)}V^{\perp}$

Figure 9: Remainder of J'th Column Ater Row Operation.

joint is not moved ($d_l^k = d_l^{k+1}$), the unknowns just before the joint axis will disappear during row operations on the Jacobian matrix. This last discussion supports the idea that in calibration, it is important to exercise the manipulator in joint space to insure that the joint rotations and translation have appreciable magnitude.

Not only has this analysis demonstrated that there is a well defined number of essential parameters for any manipulator, but it has also demonstrated which parameters they are. For example, the analysis indicates that the important parameters should provide a two-dimensional rotation vector perpendicular to the immediately following revolute or prismatic axis and a two dimensional translation displacement vector also perpendicular to an immediately following revolute joint. In addition, the model must contain a three-dimensional displacement and three-dimensional rotation after the last motion axis. These parameters are essential, and any model that lacks them is not complete.

2.4.9 Example

Consider a three revolute joint manipulator shown in figure 10. The method of this paper will be used to establish calibration models which have maximum rank Jacobian matrices.

One full rank model is shown in figure 10 and can be expressed as:

$$\begin{bmatrix} 1 & 0 & 0 & \epsilon_{x1} \\ 0 & 1 & 0 & \epsilon_{y1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(\alpha_{x1}) & -\mathrm{S}(\alpha_{x1}) & 0 \\ 0 & \mathrm{S}(\alpha_{x1}) & \mathrm{C}(\alpha_{x1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{y1}) & 0 & \mathrm{S}(\alpha_{y1}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{y1}) & 0 & \mathrm{C}(\alpha_{y1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{C}(\theta_1) & -\mathrm{S}(\theta_1) & 0 & 0 \\ \mathrm{S}(\theta_1) & \mathrm{C}(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

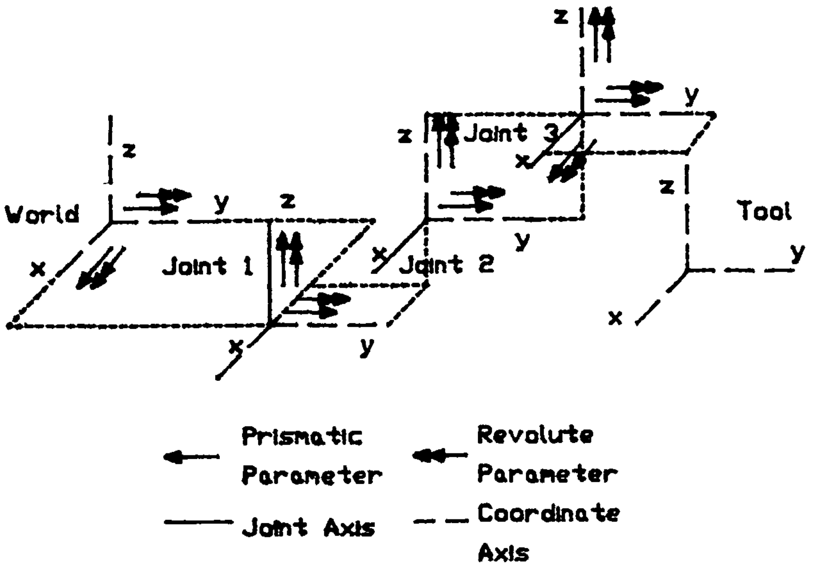

Figure 10: A Complete Model for a 3 Joint Manipulator.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \epsilon_{y2} \\ 0 & 0 & 1 & L+\epsilon_{z2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{z2}) & -\mathrm{S}(\alpha_{z2}) & 0 & 0 \\ \mathrm{S}(\alpha_{z2}) & \mathrm{C}(\alpha_{z2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{y2}) & 0 & \mathrm{S}(\alpha_{y2}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{y2}) & 0 & \mathrm{C}(\alpha_{y2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-\theta_2) & -\mathrm{S}(-\theta_2) & 0 \\ 0 & \mathrm{S}(-\theta_2) & \mathrm{C}(-\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & D+\epsilon_{y3} \\ 0 & 0 & 1 & \epsilon_{z3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{z3}) & -\mathrm{S}(\alpha_{z3}) & 0 & 0 \\ \mathrm{S}(\alpha_{z3}) & \mathrm{C}(\alpha_{z3}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{y3}) & 0 & \mathrm{S}(\alpha_{y3}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{y3}) & 0 & \mathrm{C}(\alpha_{y3}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-\theta_3) & -\mathrm{S}(-\theta_3) & 0 \\ 0 & \mathrm{S}(-\theta_3) & \mathrm{C}(-\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

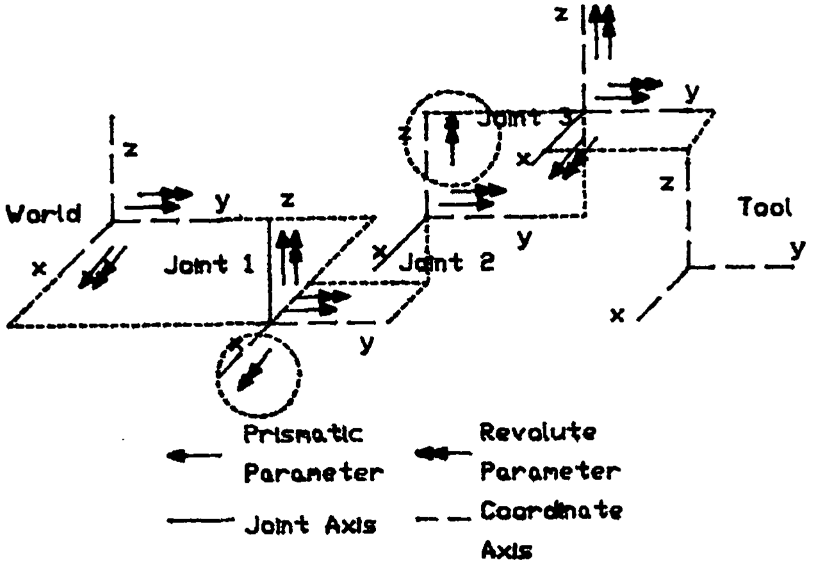

Figure 11: A Complete Model for a 3 Joint Manipulator Demonstrating Parameter Shifting.

$$\begin{bmatrix} 1 & 0 & 0 & \epsilon_{xt} \\ 0 & 1 & 0 & \epsilon_{yt} \\ 0 & 0 & 1 & \epsilon_{zt} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(\alpha_{xt}) & -\mathrm{S}(\alpha_{xt}) & 0 \\ 0 & \mathrm{S}(\alpha_{xt}) & \mathrm{C}(\alpha_{xt}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{yt}) & 0 & \mathrm{S}(\alpha_{yt}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{yt}) & 0 & \mathrm{C}(\alpha_{yt}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} \mathrm{C}(\alpha_{zt}) & -\mathrm{S}(\alpha_{zt}) & 0 & 0 \\ \mathrm{S}(\alpha_{zt}) & \mathrm{C}(\alpha_{zt}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In this particular model, we have used α to represent an angle to be calibrated, ϵ is a length to be calibrated, and θ is a measured relative encoder angle. In all the models given here, the absolute zero of each encoder reading is arbitrary and can be conveniently selected. The model includes rotations and translations perpendicular to the upcoming rotation axis and is therefore complete. After joint 3 comes three rotations and translations. This model will have a full rank Jacobian except for special cases occurring, for example, when some of the α are near 90 degrees.

Another full rank model is shown in figure 11 and can be expressed as:

$$\begin{bmatrix} 1 & 0 & 0 & \epsilon_{x1} \\ 0 & 1 & 0 & \epsilon_{y1} \\ 0 & 0 & 1 & L+\epsilon_{z2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(\alpha_{x1}) & -\mathrm{S}(\alpha_{x1}) & 0 \\ 0 & \mathrm{S}(\alpha_{x1}) & \mathrm{C}(\alpha_{x1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{y1}) & 0 & \mathrm{S}(\alpha_{y1}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{y1}) & 0 & \mathrm{C}(\alpha_{y1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{C}(\theta_1+\alpha_{z2}) & -\mathrm{S}(\theta_1+\alpha_{z2}) & 0 & 0 \\ \mathrm{S}(\theta_1+\alpha_{z2}) & \mathrm{C}(\theta_1+\alpha_{z2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & \epsilon_{xt} \\ 0 & 1 & 0 & \epsilon_{y2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{y2}) & 0 & \mathrm{S}(\alpha_{y2}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{y2}) & 0 & \mathrm{C}(\alpha_{y2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-\theta_2) & -\mathrm{S}(-\theta_2) & 0 \\ 0 & \mathrm{S}(-\theta_2) & \mathrm{C}(-\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & D+\epsilon_{y3} \\ 0 & 0 & 1 & \epsilon_{z3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{z3}) & -\mathrm{S}(\alpha_{z3}) & 0 & 0 \\ \mathrm{S}(\alpha_{z3}) & \mathrm{C}(\alpha_{z3}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{y3}) & 0 & \mathrm{S}(\alpha_{y3}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{y3}) & 0 & \mathrm{C}(\alpha_{y3}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-\theta_3) & -\mathrm{S}(-\theta_3) & 0 \\ 0 & \mathrm{S}(-\theta_3) & \mathrm{C}(-\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \epsilon_{yt} \\ 0 & 0 & 1 & \epsilon_{zt} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(\alpha_{xt}) & -\mathrm{S}(\alpha_{xt}) & 0 \\ 0 & \mathrm{S}(\alpha_{xt}) & \mathrm{C}(\alpha_{xt}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{yt}) & 0 & \mathrm{S}(\alpha_{yt}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{yt}) & 0 & \mathrm{C}(\alpha_{yt}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{C}(\alpha_{zt}) & -\mathrm{S}(\alpha_{zt}) & 0 & 0 \\ \mathrm{S}(\alpha_{zt}) & \mathrm{C}(\alpha_{zt}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The unknown quantities α and ϵ will differ from those above. This model demonstrates that parameters parallel with the upcoming joint axis may be shifted passed that axis.

Another good model is shown in figure 12 and is given by:

$$\begin{bmatrix} 1 & 0 & 0 & \epsilon_{x1} \\ 0 & 1 & 0 & \epsilon_{y1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(\alpha_{x1}) & -\mathrm{S}(\alpha_{x1}) & 0 \\ 0 & \mathrm{S}(\alpha_{x1}) & \mathrm{C}(\alpha_{x1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{y1}) & 0 & \mathrm{S}(\alpha_{y1}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{y1}) & 0 & \mathrm{C}(\alpha_{y1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{C}(\theta_1) & -\mathrm{S}(\theta_1) & 0 & 0 \\ \mathrm{S}(\theta_1) & \mathrm{C}(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \epsilon_{y2} \\ 0 & 0 & 1 & L+\epsilon_{z2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{z2}) & -\mathrm{S}(\alpha_{z2}) & 0 & 0 \\ \mathrm{S}(\alpha_{z2}) & \mathrm{C}(\alpha_{z2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{y2}) & 0 & \mathrm{S}(\alpha_{y2}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{y2}) & 0 & \mathrm{C}(\alpha_{y2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-\theta_2+\alpha_{x2}) & -\mathrm{S}(-\theta_2+\alpha_{x2}) & 0 \\ 0 & \mathrm{S}(-\theta_2+\alpha_{x2}) & \mathrm{C}(-\theta_2+\alpha_{x2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & D+\epsilon_{y3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{z3}) & -\mathrm{S}(\alpha_{z3}) & 0 & 0 \\ \mathrm{S}(\alpha_{z3}) & \mathrm{C}(\alpha_{z3}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{y3}) & 0 & \mathrm{S}(\alpha_{y3}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{y3}) & 0 & \mathrm{C}(\alpha_{y3}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(-\theta_3) & -\mathrm{S}(-\theta_3) & 0 \\ 0 & \mathrm{S}(-\theta_3) & \mathrm{C}(-\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & \epsilon_{xt} \\ 0 & 1 & 0 & \epsilon_{yt} \\ 0 & 0 & 1 & \epsilon_{zt} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathrm{C}(\alpha_{xt}) & -\mathrm{S}(\alpha_{xt}) & 0 \\ 0 & \mathrm{S}(\alpha_{xt}) & \mathrm{C}(\alpha_{xt}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{C}(\alpha_{yt}) & 0 & \mathrm{S}(\alpha_{yt}) & 0 \\ 0 & 1 & 0 & 0 \\ -\mathrm{S}(\alpha_{yt}) & 0 & \mathrm{C}(\alpha_{yt}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{C}(\alpha_{zt}) & -\mathrm{S}(\alpha_{zt}) & 0 & 0 \\ \mathrm{S}(\alpha_{zt}) & \mathrm{C}(\alpha_{zt}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This model demonstrates that a rotational parameter (α_{x2}) parallel to the upcoming joint axis with a displacement parameter (ϵ_{y3}) perpendicular to the joint axis can be used as polar coordinates to replace a displacement parameter.

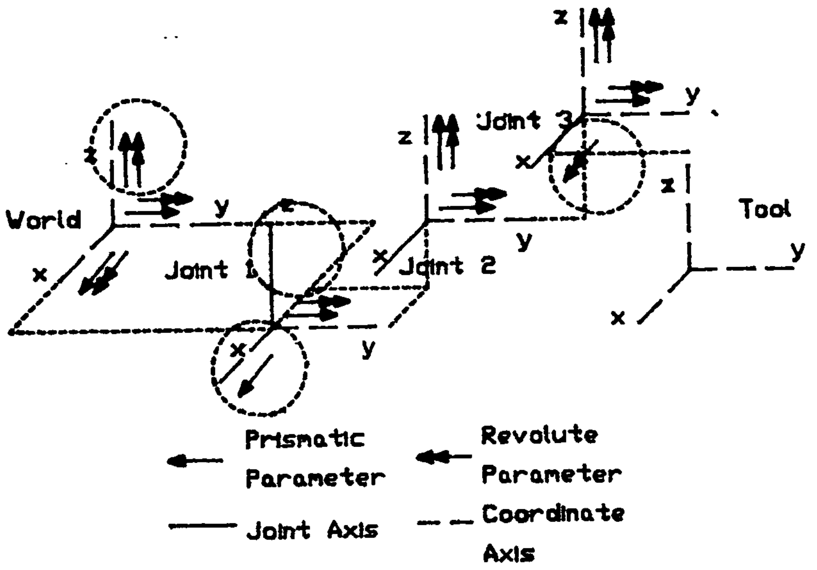

Figure 12: A Complete Model for a 3 Joint Manipulator Demonstrating Polar Coordinates.

These examples attempt to show that defining a revolute joint axis requires locating a point on the axis and the orientation of the axis. This can be done in a number of ways as shown above.

2.4.10 Conclusions

This formal proof that there are a maximum number of independent parameters in kinematic calibration models shows that the maximum number depends on the number and type of joints. It is possible to predetermine which parameters in a model are independent and which are not.

For each revolute joint, four parameters must be determined. At least two of the four parameters must be rotational. The remainder must provide two translations. A satisfactory set of parameters can be determined if the approximate orientation of each joint axis is known. Two orientation parameters applied about noncolinear axes before and perpendicular to joint axis j should be selected for identification, and two prismatic parameters applied along noncolinear axes before and perpendicular to joint axis j should be selected.

For a prismatic joint, only two parameters need to be determined. The two parameters must be rotation parameters and should be applied about noncolinear axes before and perpendicular to the prismatic joint axis.

For specifying the end effector, six parameters must be determined. These can be chosen as three rotational and three translation parameters applied after the last joint axis along three mutually perpendicular axes.

2.5 Using Single Point Sensors

When a manipulator is calibrated, a relation between tool pose relative to the world and the joint configuration is obtained. To express the tool pose relative to the world requires specification of at least two coordinate systems. One system, fixed in the world, is called the world reference frame. The other system, fixed in the hand or tool, is called the tool reference frame. Refer to figure 13.

The tool and world frames must be physical, easily located systems so the user can specify their. Some calibration algorithms place the tool reference on the manipulator's last rotation axis. Although this is well defined mathematically, it is difficult (if not impossible) for a human operator to specify positions relative to it therefore should be avoided.

A third coordinate system is the sensor reference frame. When a sensor is used to collect calibration data, the measured points lie in a sensor coordinate system. Note that in general the sensor reference can differ from the tool reference, and usually is fixed in the tool. Because measurements are taken using the sensor, calibration algorithms relate sensor pose relative to the world. In contrast, the ultimate objective is to compute the tool pose relative to the world. Hence it is sometimes necessary to relate the sensor reference to the tool, and until this is done, the calibration is not complete. This part of the report discusses one way this relation between sensor and tool frames can be computed.

2.5.1 Single Point Sensors

A variety of sensors have been used for forward calibration of manipulators, many of which measure positions of individual points. The theodolite system of Whitney, Lozinski and Rourke [6] is capable of determining the position of a very

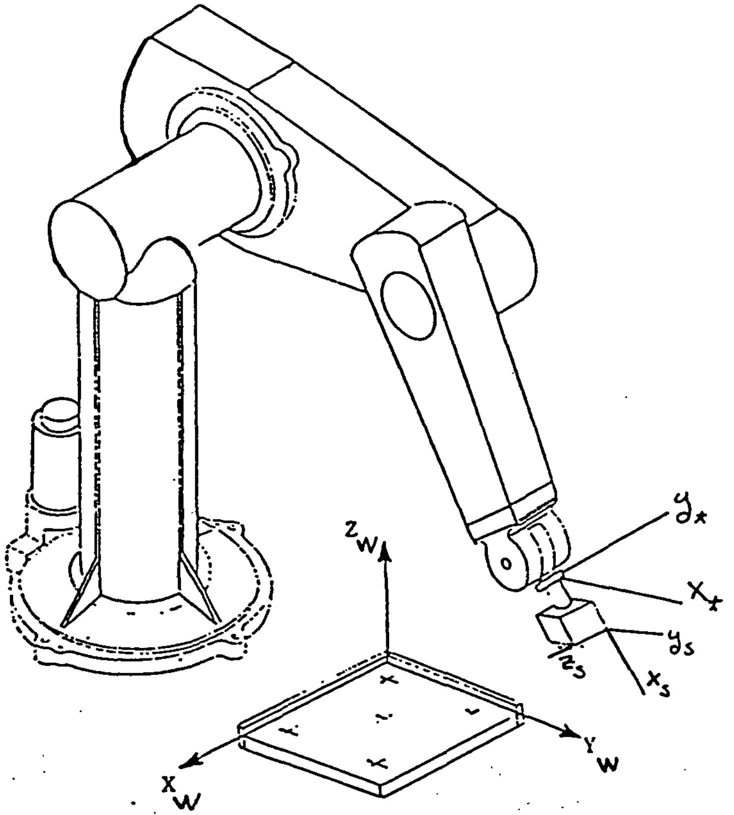

Figure 13: The World, Tool, and Sensor Coordinate Systems.

small target sphere (a point) relative to a global reference system. Chen and Chao [8] use a similar device. The output of the theodolite system is a set of coordinates defining the position of an arbitrary point. The system of Stone, Sanderson, and Neuman [10] acoustically measures the position of a spark source (a point) which, as in the theodolite system, is arbitrarily located. Veitschegger and Wu [23] used a pointer device where a point (the tip of a pointer) is at a known location in the workspace, see figure 14. Veitschegger's sensor system, can accurately locate the pointer tip only when it contacts a target location. These systems might be called point sensors since they locate points. Not all sensor systems are point sensors. The five degree-of-freedom, laser interferometer system developed by Lau, Hocken, and Haynes [24] for example, measures part of a body's orientation as well.

It is possible to use point sensors to sense the orientation of a body by measuring the positions of multiple points on a stationary rigid body. This is simple math-

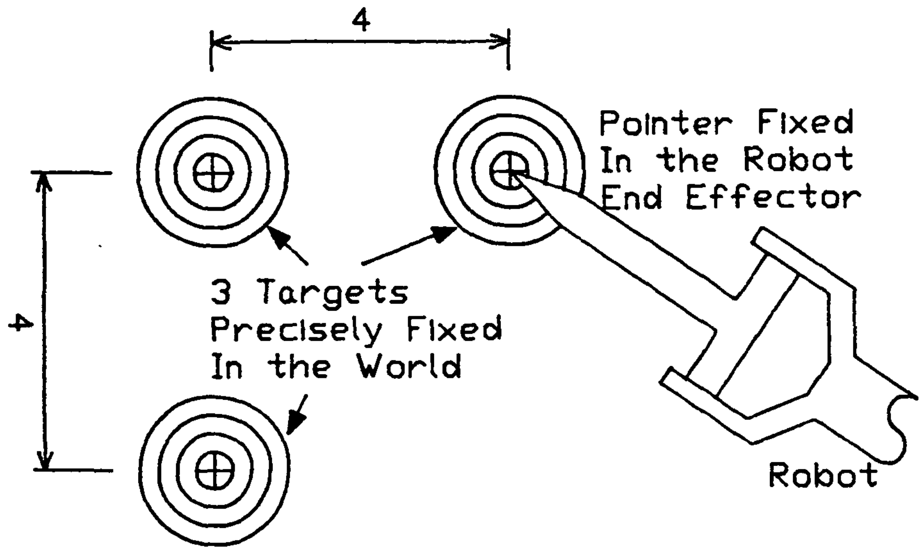

Figure 14: A Pointer and Target Measurement System.

ematically, but it requires a measurement system capable of measuring multiple points without motion of the body. A sensor capable of measuring multiple points on a body will be called a multipoint sensor. The theodolite system for example is a multipoint sensor.

Of main concern here are single point sensors, instruments that can determine the position of only one point at a time. For example the sensor systems of Hyatti and Veitschegger are single point sensors. By defining multiple target points, it is possible to use Veitschegger's sensor to determine information about multiple points of the manipulator, but the system operates by locating a single point at a time. The reason for isolating single point sensor systems is that they can be very inexpensive, highly repeatable, and reliable.

It is convenient to classify points as either physical or nonphysical. A physical point can be touched, and hence measured, with conventional measurement techniques. For example, a small sphere mounted on a manipulator creates a physical point. A nonphysical point cannot be touched or measured easily. For example, a proximity probe usually trips when it comes within a certain distance of an object. The trip point is a constant distance from the object, but it is invisible and cannot be preset or precomputed. It will be shown that there is a simple method by which manipulators can be completely calibrated using nonphysical single point sensors.

2.5.2 Completing the Calibration Using a Single Point Sensor

The objective of forward calibration is to determine a relation between joint configuration and the pose of the tool or hand with respect to the world or global coordinate system. A sensor produces a relation of the sensor reference system with respect to the world because the sensor measures only the position of the sensor frame not the position of the tool frame. If the sensor system is physical, it may be possible to measure, with conventional devices, the constant relation between sensor and tool reference thereby allowing the computation of tool pose given sensor pose. If the sensor system is nonphysical, the sensor/tool pose may not be easy to determine. In addition, since single point sensors do not provide orientation information, it may not be possible to calibrate the correct orientation of the sensor frame. This subsection presents a method by which nonphysical, single point sensors can gather enough information to complete the calibration.

To determine the sensor/tool pose, it is necessary to locate multiple points in the tool. A manipulator held fixture (called the orientation fixture) shown in figure 15 has been used on a manipulator. It is possible to use three of the four physical points on the orientation fixture to define a coordinate system. This orientation frame differs from the sensor frame used for calibration. One point of the orientation fixture becomes the origin of the orientation frame. Another arbitrarily selected point defines the X axis, and a third point defines the X, Y plane. The coordinates

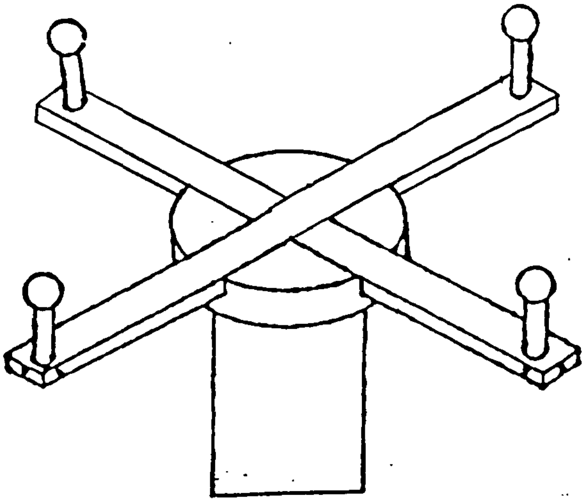

Figure 15: The Orientation Fixture.

of the four physical points of the orientation fixture are measurable and can be expressed as:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} S_0 \\ 1 \end{bmatrix} ; \begin{bmatrix} X_1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} S_1 \\ 1 \end{bmatrix} ; \begin{bmatrix} X_2 \\ Y_2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} S_2 \\ 1 \end{bmatrix} ; \begin{bmatrix} X_3 \\ Y_3 \\ Z_3 \\ 1 \end{bmatrix} = \begin{bmatrix} S_3 \\ 1 \end{bmatrix} \qquad (76)$$

Note that since the orientation frame is physical, it is easy to measure its pose with respect to the tool.

Suppose a nonphysical single point sensor was used to calibrate a manipulator, therefore an accurate relation between joint configuration and sensor pose is available. This relation can be expressed as the four-by-four transformation ${}^{w}T^{s}$. Note that the relation is well defined although the sensor frame is nonphysical and cannot be touched. What we desire is to determine the transform ${}^{w}T^{t}$, which is the tool pose relative to the world.

The tool pose relative to the world can be found from:

$$ {}^wT^t = {}^wT^{s\,s}T^{o\,o}T^t \tag{77}$$

Here ${}^wT^t$ is unknown and desired, ${}^wT^s$ can be accurately computed since we have calibrated the manipulator, and ${}^sT^o$, the pose of the orientation frame relative to the sensor frame is unknown but constant. The transform ${}^oT^t$, the pose of the tool relative to the orientation fixture, is measurable since both frames are physical. The objective is to compute the unknown transform ${}^sT^o$ thereby making the computation of ${}^wT^t$ possible.

To compute the unknown transform, one can use the same single point sensor used for calibrating the manipulator. For example, the pointer of Veitschegger's sensor could be fixed in the workspace and four targets placed on the manipulator. The manipulator would be driven, four separate times, so that each target on the manipulator is pointed to. Based on the four joint configurations corresponding to pointing at each target, the unknown transform can be computed as will be demonstrated next.

Consider the transform equation:

$$ {}^wT^o = {}^wT^{s\,s}T^o = {}^wT^s \begin{bmatrix} {}^s\vec{n}^o & {}^s\vec{o}^o & {}^s\vec{a}^o & {}^s\vec{p}^o \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{78}$$

When target i on the manipulator is brought to the fixed pointer, regardless of the approach direction, the target's coordinates (also the pointer's coordinates) can be expressed as:

$$ \begin{bmatrix} \vec{X} \\ 1 \end{bmatrix} = {}^wT^s_i \begin{bmatrix} {}^s\vec{n}^o & {}^s\vec{o}^o & {}^s\vec{a}^o & {}^s\vec{p}^o \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_i \\ 1 \end{bmatrix} \tag{79}$$

The vector $\vec{X}$ is a constant but unknown position of the target relative to the world. The transforms ${}^wT^s_i$ are computed from the joint configuration measured when target i of the manipulator is pointed at. Rewriting equation 79 yields:

$$ {}^sT^w_i \begin{bmatrix} \vec{X} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^s\vec{n}^{\,o} S_{xi} + {}^s\vec{o}^{\,o} S_{yi} + ({}^s\vec{n}^{\,o} \times {}^s\vec{o}^{\,o}) S_{zi} \\ 1 \end{bmatrix} + \begin{bmatrix} {}^s\vec{p}^{\,o} \\ 0 \end{bmatrix} \qquad (80) $$

The terms S_{xi}, S_{yi}, and S_{zi} are the x, y and z components of S_i. Equation 80 represents four matrix equations, each corresponding to a different target on the manipulator. Combining equations 80 with 76 yields:

$$ \begin{gathered} \left[({}^sT^w_1 - {}^sT^w_0) \begin{bmatrix} \vec{X} \\ 1 \end{bmatrix} \right] / X_1 = \begin{bmatrix} {}^s\vec{n}^{\,o} \\ 0 \end{bmatrix} \\ \left[({}^sT^w_2 - {}^sT^w_0) \begin{bmatrix} \vec{X} \\ 1 \end{bmatrix} - X_2({}^s\vec{n}^{\,o}) \right] / Y_2 = \begin{bmatrix} {}^s\vec{o}^{\,o} \\ 0 \end{bmatrix} \qquad (81) \\ ({}^sT^w_3 - {}^sT^w_0) \begin{bmatrix} \vec{X} \\ 1 \end{bmatrix} - X_3({}^s\vec{n}^{\,o}) - Y_3({}^s\vec{o}^{\,o}) - Z_3\,({}^s\vec{n}^{\,o} \times {}^s\vec{o}^{\,o})^t = 0 \end{gathered} $$

Equations 81 can be iterated to compute the quantity $\vec{X}$. When $\vec{X}$ is known, equations 80 can be used for the unknown transformation ${}^sT^o$.

2.5.3 Example

Equations 81 are solved in this subsection. To demonstrate the effects of measurement error, random normal deviates are injected into simulated data. The simulation process is depicted in the flow chart in figure 16.

Recall that the single point sensor is placed in an unknown but fixed position $\vec{X}$ in the world reference. For the simulation, $\vec{X}$ was arbitrarily chosen as:

$$ \vec{X} = (11.0, -2.0, 3.0)^T \text{ inches} \qquad (82) $$

The unknown transformation ${}^sT^o$ (the desired result of the simulation) is arbi-

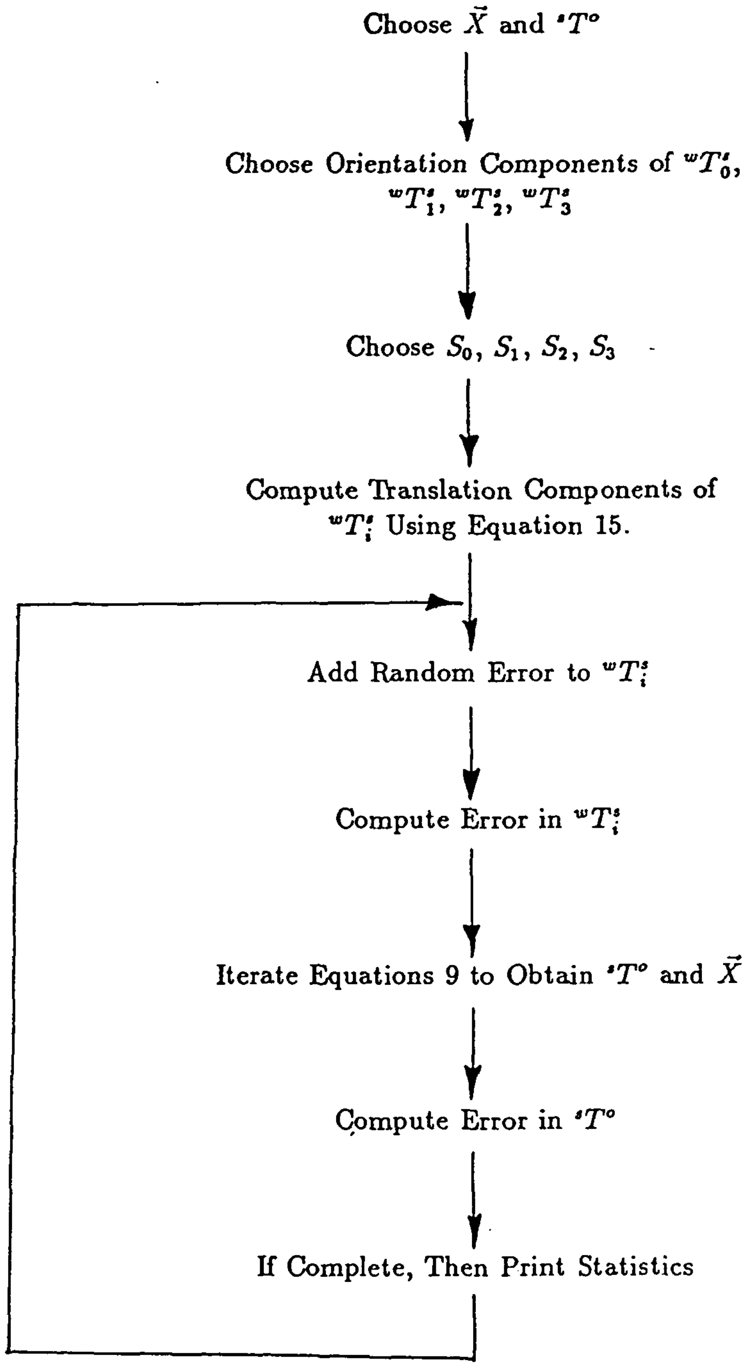
Choose $\vec{X}$ and ${}^{s}T^{o}$
Choose Orientation Components of ${}^{w}T^{s}_{0}$, ${}^{w}T^{s}_{1}$, ${}^{w}T^{s}_{2}$, ${}^{w}T^{s}_{3}$
Choose S_0, S_1, S_2, S_3
Compute Translation Components of ${}^{w}T^{s}_{i}$ Using Equation 15.
Add Random Error to ${}^{w}T^{s}_{i}$
Compute Error in ${}^{w}T^{s}_{i}$
Iterate Equations 9 to Obtain ${}^{s}T^{o}$ and $\vec{X}$
Compute Error in ${}^{s}T^{o}$
If Complete, Then Print Statistics

Figure 16: Flow Chart of the Simulation Procedure.

trarily chosen. By choosing ${}^sT^o$, it is possible to compute the remainder of the input data. When the input is known, equations 81 are solved. The result of equations 81 is compared to the chosen ${}^sT^o$ and $\vec{X}$.

When choosing transformations, it is important that they be consistent with the properties of transform matrices; therefore transforms are chosen by specifying screw matrix parameters. The screw axis, angle and displacement for matrix ${}^sT^o$ are chosen as:

$$(-.5, .5, .707),\ 45^o,\ (-2., 11., 3.) \text{ inches} \tag{83}$$

These parameters produce the following transformation matrix:

$$\begin{bmatrix} 0.7803301 & -0.5732233 & 0.2500000 & -2.000000 \\ 0.4267767 & 0.7803301 & 0.4571068 & 11.00000 \\ -0.4571068 & -0.2500000 & 0.8535534 & 3.000000 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^sT^o \tag{84}$$

The next step is to compute the four matrices ${}^wT_i^s$ that represent the manipulator's sensor reference pose relative to the world when point i of the orientation fixture is placed at position $\vec{X}$. The orientation component of these matrices can be arbitrarily selected without violating any kinematic equations. Therefore their screw axes and angles were selected as:

$$\begin{gathered} (-.25, .5, .829),\ 30^o \\ (.5, .25, .829),\ 45^o \\ (.1, .75, .654),\ -15^o \\ (.75, .1, .654),\ 50^o \end{gathered} \tag{85}$$

Next the positions of the targets on the orientation fixture are selected. Their positions relative to the orientation fixture are arbitrary so they were initially selected (in inches) as:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} S_0 \\ 1 \end{bmatrix}; \begin{bmatrix} 10 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} S_1 \\ 1 \end{bmatrix}; \begin{bmatrix} 10 \\ 10 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} S_2 \\ 1 \end{bmatrix}; \begin{bmatrix} 0 \\ 10 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} S_3 \\ 1 \end{bmatrix} \tag{86}$$

The only unknown input data at this point are the translation components of the matrices ${}^wT_i^s$. These are calculated from the following equation:

$$\begin{bmatrix} \vec{X} \\ 1 \end{bmatrix} = {}^wR_i^s\,{}^sT^o \begin{bmatrix} S_i \\ 1 \end{bmatrix} + \begin{bmatrix} {}^w\vec{p}_i{}^s \\ 1 \end{bmatrix} \tag{87}$$

Here ${}^wR_i^s$ is a pure rotation transformation matrix equivalent to the orientation part of ${}^wT_i^s$, and ${}^w\vec{p}_i{}^s$ is the translation component of ${}^wT_i^s$. The solution of equation 87 makes it possible to express the four ${}^wT^s$ matrices as:

$$\begin{gathered}
\begin{bmatrix} 0.8743988 & 0.3978313 & -0.2777715 & 9.205970 \\ -0.4313249 & 0.8995190 & -0.06945706 & -12.54899 \\ 0.2222285 & 0.1805429 & 0.9581329 & -1.415914 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^wT_0^s \\
\begin{bmatrix} 0.7803301 & 0.6229137 & -0.0553495 & -3.125949 \\ -0.5496903 & 0.7254126 & 0.4142669 & -9.234571 \\ 0.2982038 & -0.2928398 & 0.9084709 & 7.167713 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^wT_1^s \\
\begin{bmatrix} 0.9662666 & -0.1666693 & 0.1963422 & 15.57589 \\ 0.1717805 & 0.9850926 & -0.009172767 & -24.77669 \\ -0.1918864 & 0.0425910 & 0.9804925 & 6.022668 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^wT_2^s \\
\begin{bmatrix} 0.8437196 & 0.5276574 & 0.0985639 & 7.552853 \\ -0.4740756 & 0.6463597 & 0.5978891 & -18.11831 \\ 0.2517729 & -0.5511776 & 0.7954959 & 14.91298 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^wT_3^s
\end{gathered} \tag{88}$$

The data in equations 86 and 88 can be used to iterate equations 81 to obtain the unknowns $\vec{X}$ and ${}^{s}T^{o}$. Instead, measurement error is simulated to determine its effect on the solution.

To simulate measurement error, the four matrices given in equations 88 are modified. A normally distributed random number is added to the components of the screw parameters given in equation 85. Eight normal deviates with zero mean and a chosen standard deviation ($\sigma_a = 0.001$) are generated and added to the x and y components of the four screw axis parameters. The four z components are determined so that the axis vector is a unit length. Four new normal deviates with zero mean and standard deviation $\sigma_\theta = 0.1^o$ are generated and added to the four angles in equations 85. From the modified screw parameters, modified orientation parts for ${}^{w}T_{i}^{s}$ are determined. Finally, twelve more zero mean normal deviates with standard deviation $\sigma_p = 0.1$ inches are generated and added to the translation components of the matrices given in equation 88. One particular set of modified ${}^{w}T^{s}$ matrices follow:

$$\begin{gathered}
\begin{bmatrix} 0.8744398 & 0.3971986 & -0.2785468 & 9.366833 \\ -0.4311531 & 0.8994911 & -0.07087093 & -12.54547 \\ 0.2224006 & 0.1820687 & 0.9578043 & -1.347170 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{w}T^{s}_{0m} \\
\begin{bmatrix} 0.7807935 & 0.6223629 & -0.0550087 & -2.964712 \\ -0.5499712 & 0.7264028 & 0.4121537 & -9.225114 \\ 0.2964677 & -0.2915537 & 0.9094523 & 7.144633 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{w}T^{s}_{1m} \\
\begin{bmatrix} 0.9665945 & -0.1658109 & 0.1954533 & 15.52145 \\ 0.1708459 & 0.9852560 & -0.0090689 & -24.85198 \\ -0.1910678 & 0.0421583 & 0.9806711 & 5.999934 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{w}T^{s}_{2m} \\
\begin{bmatrix} 0.8436117 & 0.5278046 & 0.0986999 & 7.633105 \\ -0.4738018 & 0.6452208 & 0.5993346 & -18.10223 \\ 0.2526483 & -0.5523698 & 0.7943906 & 14.97786 \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{w}T^{s}_{3m}
\end{gathered} \tag{89}$$

From the four modified matrices of equation 89, equation 81 is solved for matrix ${}^{s}T^{o}$. In all cases, the initial estimate of $\vec{X}$ is zero. When the matrices in equation 89 are used, the solution is:

$$\begin{bmatrix} 0.7847148 & -0.5669553 & 0.2497151 & -1.892561 \\ 0.4280652 & 0.7875659 & 0.4436474 & 11.01522 \\ -0.4483111 & -0.2414573 & 0.8607085 & 2.918883 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{90}$$

The simulation results express the error between matrices. Matrix error is calculated as the one norm (sum of absolute values) of matrix difference, e.g. Error $= \|T_{\text{Correct}} - T_{\text{Incorrect}}\|$. To obtain statistics, the simulation of measurement error was repeated multiple times. At each iteration, the process started with the correct screw parameters in equations 85 and correct translation components of ${}^{w}T^{s}$ in the fourth column of the matrices in equations 88. From this point, new random numbers were generated to determine modified ${}^{w}T^{s}$ matrices. At each iteration, errors in the modified and original ${}^{w}T^{s}$ matrices were determined and recorded. After solution, the error between the computed and correct ${}^{s}T^{o}$ matrix was computed and recorded. After $n = 100$ iterations, the average input data error was computed as the sum of the ${}^{w}T^{s}$ errors divided by $4n = 400$. The mean solution error was computed as the sum of ${}^{s}T^{o}$ error divided by $n = 100$. The sample standard deviations were similarly computed. Typical results are in tables 1, 2 and 3.

Based on the results shown, the method can be used to complete the forward calibration problem when single point sensors are used. The solution error seems to be slightly sensitive to the orientation fixture target spacing and may require consideration in designing the fixture. The method of solution was a standard zero crossing algorithm. It may be possible to obtain more accurate results with a minimization algorithm. Since the introduction of measurement error may cause the equations to have no solution, the minimization problem may be better defined.

In this work, the zero crossing algorithm was iterated 1000 times; if it did not reach convergence within that time, the last best estimate of $\vec{X}$ was used. The results given include the errors when the algorithm failed to converge. The algorithm used required the input of only four ${}^{w}T^{s}$ matrices; by collecting more data than necessary and averaging them, it may be possible to reduce the error in the result. To do this, one should define the problem as the minimization of an objective function, collect more data than necessary, and determine the best fit ${}^{s}T^{o}$ matrix. This is essentially what is done with many of the forward calibration algorithms.

Table 1: Summary of Input Data and Results of the Simulation

Correct x, y, z coordinates of $\vec{X}$	11.	-2.	3.
Screw Parameters for ${}^{w}T^{s}$			
x, y, components of Screw axis, θ	-.5	.5	45.
x, y, z components of Displacement	-2.	11.	3.
Coordinates of point 0 of Orientation fixture	0	0	0
Coordinates of point 1 of Orientation fixture	10	0	0
Coordinates of point 2 of Orientation fixture	10	10	0
Coordinates of point 3 of Orientation fixture	0	10	0
Screw Parameters for ${}^{w}T^{s}_{0}$			
x, y, components of Screw axis, θ	-.25	.5	30.
Screw Parameters for ${}^{w}T^{s}_{1}$			
x, y, components of Screw axis, θ	.5	.25	45.
Screw Parameters for ${}^{w}T^{s}_{2}$			
x, y, components of Screw axis, θ	.1	.75	-15.
Screw Parameters for ${}^{w}T^{s}_{3}$			
x, y, components of Screw axis, θ	.75	.1	50.
Standard deviation for Screw axis	0.001		
Standard deviation for Screw angle		0.1	
Standard deviation for Displacement			0.1
Number of iterations	100		
Input Data Mean and Standard Deviation	0.242	0.102	
Resulting Mean and Standard Deviation	0.713	0.378	

Table 2: Summary of Input Data and Results of the Simulation

Correct x, y, z coordinates of $\vec{X}$	11.	-2.	3.
Screw Parameters for ${}^wT^s$			
x, y, components of Screw axis, θ	-.5	.5	45.
x, y, z components of Displacement	-2.	11.	3.
Coordinates of point 0 of Orientation fixture	0	0	0
Coordinates of point 1 of Orientation fixture	10	0	0
Coordinates of point 2 of Orientation fixture	10	10	0
Coordinates of point 3 of Orientation fixture	0	10	0
Screw Parameters for ${}^wT^s_0$			
x, y, components of Screw axis, θ	-.25	.5	30.
Screw Parameters for ${}^wT^s_1$			
x, y, components of Screw axis, θ	.5	.25	45.
Screw Parameters for ${}^wT^s_2$			
x, y, components of Screw axis, θ	.1	.75	-15.
Screw Parameters for ${}^wT^s_3$			
x, y, components of Screw axis, θ	.75	.1	50.
Standard deviation for Screw axis	0.001		
Standard deviation for Screw angle		0.1	
Standard deviation for Displacement			0.1
Number of iterations	1000		
Input Data Mean and Standard Deviation	0.245	0.105	
Resulting Mean and Standard Deviation	0.752	0.391	

Table 3: Summary of Input Data and Results of the Simulation

Correct x, y, z coordinates of $\vec{X}$	11.	-2.	3.
Screw Parameters for $^{w}T^{s}$			
x, y, components of Screw axis, θ	-.5	.5	45.
x, y, z components of Displacement	-2.	11.	3.
Coordinates of point 0 of Orientation fixture	0	0	0
Coordinates of point 1 of Orientation fixture	5	0	0
Coordinates of point 2 of Orientation fixture	5	5	0
Coordinates of point 3 of Orientation fixture	0	5	0
Screw Parameters for $^{w}T^{s}_{0}$			
x, y, components of Screw axis, θ	-.25	.5	30.
Screw Parameters for $^{w}T^{s}_{1}$			
x, y, components of Screw axis, θ	.5	.25	45.
Screw Parameters for $^{w}T^{s}_{2}$			
x, y, components of Screw axis, θ	.1	.75	-15.
Screw Parameters for $^{w}T^{s}_{3}$			
x, y, components of Screw axis, θ	.75	.1	50.
Standard deviation for Screw axis	0.001		
Standard deviation for Screw angle		0.1	
Standard deviation for Displacement			0.1
Number of iterations	100		
Input Data Mean and Standard Deviation	0.242	0.102	
Resulting Mean and Standard Deviation	0.787	0.397	

2.5.4 Conclusions

This work demonstrated how nonphysical single point sensors can gather sufficient position information to complete a forward kinematic calibration. Single point sensors, which can measure the position of only one point fixed to the tool of the manipulator, have several advantages over other types. They are simpler, less expensive, highly accurate, and can provide feedback to the manipulator controller. This feedback often consists of "go", "no-go" type information that can be easily interfaced to digital input lines of the control system. The feedback can "drive" the manipulator automatically, enabling the automatic collection of calibration data.

Single point sensors have been used successfully for forward kinematic calibration but have been unable to compute the correct orientation of the tool. In addition, some single point sensors are nonphysical, which means they cannot be accurately located relative to the tool. Using nonphysical sensors presents difficulty because the forward calibration can compute the pose of the sensor only and not of the tool.

The method presented utilizes a fixture held in the tool and the single point sensing system fixed in the workspace to complete the forward calibration. The method can be performed with any manipulator and produces accurate results even in the presence of measurement errors.

New sensor designs may relax the need for accurate position sensing. With this method, it is possible to perform accurate manipulator calibration with a sensor system that produces repeatable results, not necessarily accurate measurements.

REFERENCES

[1] Louis Everett, Morris Driels, and Benjamin Mooring. Kinematic modelling for robot calibration. In *Robotics and Automation*, pages 183–190, IEEE, IEEE Computer Society Press, 1730 Massachusetts Avenue, NW, Washington, D.C., April 1987.

[2] Atul Bajpai and Bernard Roth. Workspace and mobility of a close-loop manipulator. *The International Journal of Robotics Research*, 5(2):131–142, 1986.

[3] B. W. Mooring and T. J. Pack. Determination and specification of robot repeatability. *IEEE*, 1017–1023, February 1986.

[4] J. C. Colson. *Performance Measures for Robotic Systems.* Master's thesis, University of Texas at Austin, December 1984.

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

[6] D. E. Whitney, C. A. Lozinski, and J. M. Rourke. Industrial robot forward calibration method and results. *Journal of Dynamic Systems, Measurement and Control*, 108(1), March 1986.

[7] B. W. Mooring and G. R. Tang. An improved method for identifying the kinematic parameters in a six-axis robot. In *International Computers in Engineering Conference and Exhibit*, pages 79–84, 1984.

[8] J. Chen and L. M. Chao. Positioning error analysis for robot manipulator with all rotary joints. In *IEEE International Conference on Robotics and Automation*, San Francisco, California, April 7-10 1986.

[9] S. A. Hayati. Robot arm geometric link parameter estimation. In *22nd IEEE Conference on Decision and Control*, pages 1477–1483, December 1983.

[10] Henry W. Stone, Arthur C. Sanderson, and Charles P. Neuman. Arm signature identification. In *Robotics and Automation*, pages 41–48, IEEE, IEEE Computer Society Press, 1730 Massachusetts Avenue, NW, Washington, D.C., April 1986.

[11] L. J. Everett and T. W. Hsu. The theory of kinematic parameter identification for industrial robots. *To appear in the Journal of Dynamic Systems, Measurement, and Control*, March 1988.

[12] Z. Roth, B. W. Mooring, and B. Ravani. Robot precision and calibration issues in electronic assembly. In *IEEE Southcon Conference*, Orlando, Florida, 1986. Program Session Record 21.

[13] L. J. Everett. Summer faculty research program results. August 1985. IBM Inc.

[14] J. S. Shamma and D. E. Whitney. A method for inverse robot calibration. *Journal of Dynamic Systems, Measurement and Control*, 109(1):36–43, March 1987.

[15] S. A. Hayati and M. Mirmirani. Puma 600 robot arm geometric calibration. In *IEEE International Conference on Robotics*, Atlanta, Georgia, 1984.

[16] T. W. Hsu and L. J. Everett. Identification of the kinematic parameters of a robot manipulator for postional accuracy improvement. In *Computers In Engineering Conference and Exhibition*, pages 263–267, 1985.

[17] J. Denavit and R. S. Hartenberg. A kinematic notation for lower pair mechanism based on matrices. *Journal of Applied Mechanics*, 215–221, June 1955.

[18] George N. Sandor and Arthur G. Erdman. *Advanced Mechanism Design: Analysis and Synthesis, Vol. II.* Prentice-Hall, 1984.

[19] R. L. Fox and K. D. Willmert. Optimum design of curve-generating linkages with inequality constraints. *Journal of Engineering for Industry*, 144–152, February 1967.

[20] Cornelius Lanczos. *The Variational Principles of Mechanics.* University of Toronto Press, Toronto, Canada, 1957.

[21] Arthur E. Bryson Jr. and Yu-Chi Ho. *Applied Optimal Control.* John Wiley and Sons, 1975.

[22] Tsing-Wong Hsu. *Robot Accuracy Improvement Through Kinematic Parameter Identification.* PhD thesis, Texas A&M University, Mechanical Engineering, May 1987.

[23] W. K. Veitschegger and C. H. Wu. A method for calibrating and compensating robot kinematic errors. In *Robotics and Automation*, pages 39–44, IEEE, IEEE Computer Society Press, 1730 Massachusetts Avenue, NW, Washington, D.C., Apr 1987.

[24] K. Lau, R. Hocken, and L. Haynes. *Robot Performance Measurements Using Automatic Laser Tracking Techniques.* Technical Report, Center for Manufacturing Engineering, National Bureau of Standards.