ERROR MODELING OF THE CARPAL WRIST

by

Gregory Nicholas Saccoccio

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Mechanical Engineering

APPROVED:

C. F. Reinholtz, Chairman

L. D. Mitchell

P. H. Tidwell

August, 1996
Blacksburg, Virginia

Keywords: Robotic Wrist, Kinematics, Error Modeling, Sensitivity Analysis

C.2

LD
5655
V855
1996
S237
c.2

ERROR MODELING OF THE CARPAL WRIST

by

Gregory Nicholas Saccoccio

Charles F. Reinholtz, Chairman

Mechanical Engineering

(ABSTRACT)

In recent years, increased emphasis has been placed on the development of parallel-architecture mechanisms for use as robotic manipulators. Parallel robots offer the benefits of higher load-carrying capacity, greater positioning accuracy and lower weight when compared to serial devices. However, robotic wrist development has traditionally focused on serial mechanisms having a large, spherical workspace and simpler kinematic solutions. The Carpal wrist is a unique parallel mechanism consisting of a fixed base and a movable output plane connected via three serial kinematic chains. The forward and inverse kinematic problems of the Carpal wrist are solved closed-form, making the device suitable for use as a new type of robotic wrist. The closed-form solutions are dependent upon the assumptions that the fixed and moving planes are symmetric about a mid-plane and that the three kinematic chains connecting the planes are identical. This thesis investigates the errors that result from those assumptions being violated due to manufacturing and assembly errors. In the non-ideal model, pose error is found by iteratively solving a system of equations describing the output plane position and orientation and comparing them with the ideal solution. The error model is a tool for predicting the effects of kinematic parameter errors on the positioning accuracy and reachable workspace of the Carpal wrist. In this work, a general error model is developed and validated for a range of parameter error values. Special-case results are presented for errors in the individual parameters.

In loving memory of my mother,

Adrienne Saccoccio

Acknowledgments

I would first like to thank my graduate advisor, Dr. Charles F. Reinholtz, for all of his help while I have been at Virginia Tech. He has shown himself not only to be a top-notch professor and research leader, but also a career advisor and all-around "nice guy". Thanks also to Dr. Paul Tidwell and Dr. Larry Mitchell for sharing their input and experience as members of my graduate committee.

Steve Canfield and Tony Ganino deserve special thanks for the many hours spent familiarizing me with their previous work on the Carpal wrist and helping to establish the scope of this thesis. In addition to Steve and Tony, the other members of the Robotics and Mechanisms Group -- Randy Soper, Susan Larkin, Dana Coombs and Matt Doyle -- along with my good friend Manik Tewari, have always been there with wit, wisdom, badgering and hysterical laughter.

I am especially grateful for the support and encouragement I have received from my family over the past several years. They have always lent a helping hand or a sympathetic ear, often reminding me that there is light at the end of the tunnel.

Table of Contents

1. Introduction **1**

1.1 Literature Review 7

1.1.1 Overview of Robotic Wrists 8

1.1.2 Sensitivity Analysis and Error Modeling 12

1.1.3 Numerical Methods for Solving Nonlinear Equations 14

1.2 Motivation and Objective 23

2. Kinematics of the Carpal Wrist **24**

2.1 Forward Kinematics 25

2.2 Inverse Kinematics 29

3. Error Model of the Carpal Wrist **35**

3.1 Forward Kinematics of the Error Model 36

3.1.1 Locating the Mid-plane Spheric Joints 36

3.1.2 Locating the Distal Revolute Joints 39

3.1.3 Solving the System of Equations 41

3.2 Definition of Error Quantity 45

3.3 Overview of Computer Solution Techniques 46

4. Results of Carpal Wrist Error Modeling **48**

4.1 Kinematic Ratios 49

4.2 Parameter Error Values 52

4.3 Implementation of the Carpal Wrist Error Model 54

4.3.1 Residual Error in the Solution Technique 56

4.4 Individual Kinematic Parameter Errors 57

4.4.1 Basal Leg Length 59

4.4.2 Basal Plate Dimensions 67

4.4.3 Basal Revolute Locator Angles 71

4.4.4 Basal Revolute Joint Out-of-Plane Offset 77

4.4.5 Basal Revolute Joint Out-of-Plane Tilt Angle 77
4.4.6 Distal Leg Length 81
4.4.7 Distal Revolute Connectors 81
4.5 Combined Dimensional Errors 85
4.5.1 Two Basal Legs 87
4.5.2 Upper and Lower Legs - Connected 89
4.5.3 Upper and Lower Legs - Not Connected 89
4.5.4 Basal Revolute Locators Increased 92
5. Conclusions and Recommendations **94**
References **97**
Appendix 1: Listing of Matlab Codes **101**
A1.1 Sensitivity Analysis Driver 101
A1.2 Inverse Kinematics Function 106
A1.3 Newton-Raphson Function 109
A1.4 Rotation Matrix Function 111
Vita **112**

List of Figures

1.1 Kinematic Diagram of a Serial Configuration 4
1.2 Kinematic Diagram of a Parallel Configuration 5
1.3 Load Transmission in Serial vs. Parallel Manipulators 6
1.4 Roll-Pitch-Roll and Pitch-Yaw-Roll Kinematic Configurations 9
1.5 Carpal Wrist Prototype 11
1.6 Graphical Representation of Newton-Raphson Method 19
1.7 Function with Multiple Roots 20
1.8 Divergence Due to Root at an Inflection Point 20
1.9 Divergence Due to Local Maxima or Minima 21
2.1 Kinematic Diagram of the Carpal Wrist 25
2.2 Top View of Basal Plane 26
2.3 Inverse Kinematic Model 30
3.1 Top View of Non-Ideal Basal Plane 37
3.2 Basal Plane with $\boldsymbol{u}_i$ Out of Plane 38
3.3 Distal Plate Showing Segments g_i and Location of Distal Axis Intersections 39
3.4 Interconnection of Mid-joints and Distal Plane 40
3.5 Ideal Carpal Wrist Distal Axis Intersection Points 44
3.6 Computer Solution Technique Flow Diagram 47
4.1 Spherical Workspace Shells for the Carpal Wrist 50
4.2 Examples of Solid Angles 51
4.3 Effects of Kinematic Ratios on Maximum Solid Angle Workspace 52
4.4 Ideal Workspace of Carpal Wrist, $R_d = 0.875$ and $R_b = 0.375$ 54
4.5 Definition of Bend Axis and Bend Angle 56
4.6 Error Residual for Carpal Wrist with $R_d = 0.875$ and $R_b = 0.375$ 57
4.7 Plot of Averaged Error vs. Bend Axis Angle, α 58
4.8 Plot of Averaged Error vs. Bend Angle, ϕ 59

4.9 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta l_1 = 0.5\%$ 62
4.10 Lower Half of Workspace Sphere Showing Unreachable Points 64
4.11 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta l_2 = 0.5\%$ 65
4.12 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta l_3 = 0.5\%$ 66
4.13 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta b_1 = 0.5\%$ 68
4.14 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta b_2 = 0.5\%$ 69
4.15 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta b_3 = 0.5\%$ 70
4.16 Relationship of Basal Revolute Locator Angles, β_i 71
4.17 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\beta_1 = +0.6°$ 73
4.18 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\beta_1 = -0.6°$ 74
4.19 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\beta_3 = +0.6°$ 75
4.20 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\beta_3 = -0.6°$ 76
4.21 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\eta_1 = 0.5\%$ 79
4.22 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\mu_1 = 0.6°$ 80
4.23 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta l_4 = 0.5\%$ 83
4.24 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta g_{d1} = 0.5\%$ 84
4.25 Plot of Averaged Error vs. Bend Angle, ϕ, for Combined Parameter Errors 86
4.26 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for Combined Variations δl_1 and $\delta l_2 = 0.5\%$ 88
4.27 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for Combined Variations δl_1 and $\delta l_4 = 0.5\%$ 90
4.28 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for Combined Variations δl_1 and $\delta l_5 = 0.5\%$ 91
4.29 Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for Increasing $\delta\beta_1$ and $\delta\beta_3$ by $+0.6°$ 93

List of Tables

1.1 Overview of Joint Types 3

2.1 Closure Criteria for Choice of Mid-joint Location 34

4.1 Values of Kinematic Parameter Variations 53

4.2 Results of Varying Basal Leg Length, l_i 61

4.3 Results of Varying Basal Plate Dimension, b_i 67

4.4 Results of Varying Basal Revolute Locator Angles, β_i 72

4.5 Results of Varying Basal Revolute Offset, η_i, and Tilt Angle, μ_i 78

4.6 Results of Varying Distal Leg Lengths, l_{i+3}, and Revolute Connectors, g_{di} 82

4.7 Results of Combined Errors 92

5.1 Summary of Pose Error for Individual Parameter Errors 95

Chapter 1

Introduction

The Robotic Industries Association defines a robot as "a programmable, multi-functional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks." (Holzbock, 1986). Although a wide variety of kinematic structures are possible, most industrial manipulators are of the wrist-partitioned class (Craig, 1989). Industrial manipulators of this class can be divided into two sub-structures, an arm and a wrist. The robot arm is often referred to as the positioning structure. If the manipulator is to have N degrees of freedom, the robot wrist frequently contains N-3 degrees-of-freedom and serves as the orienting structure for the robot end-effector. An end-effector is a tool, probe, gripper, sprayer, etc., that is rigidly attached to the wrist. An object in space, in this case the end-effector, can be fully described relative to a fixed frame through six quantities, three for position and three for orientation. In practice, a robot has a total of five or six degrees of

freedom to allow for full spatial positioning and orientation, often referred to as the *pose*, of the end-effector (Craig, 1989).

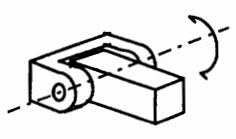
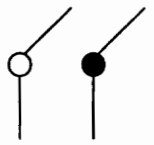
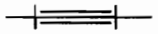
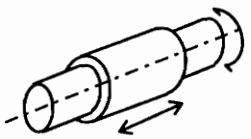
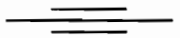
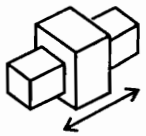
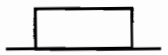
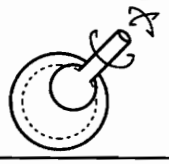
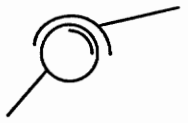
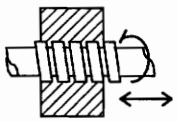
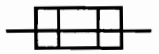
This thesis presents the development of a tool to predict the effects of kinematic parameter errors on the workspace of the Carpal wrist, a new type of parallel-architecture robotic wrist. In this chapter, an overview of kinematics and robotics concepts will first be presented to familiarize the reader with the subject matter. This is followed by a three-part review of the literature, covering previous work in robotic wrist design, including the Carpal wrist, error modeling and sensitivity analysis, and methods for solving systems of nonlinear equations. The final section of the chapter presents the motivation and objective of this work.

To proceed with the discussion of the Carpal wrist and robotics in general, some kinematic concepts must be described. The term *mechanism* generally refers to a combination of rigid bodies connected to realize a given motion. The rigid bodies are often referred to as links, and the connection allowing relative motion between neighboring links makes a kinematic pair, or joint. Kinematic pairs can be divided into two groups. A *lower-pair* refers to a connection with surface contact, such as a revolute joint. A *higher-pair* refers to connection at a point or along a line, such as between gear teeth or in a ball bearing.

In robotics, kinematic links are typically connected by one-degree-of-freedom joints of the lower-pair class. If two links are connected by a joint, the term "degree of freedom" refers to the number of independent motions one link can achieve while the other link is held fixed. For example, a revolute joint, or pin joint, has only one degree of freedom because if one link is held fixed, the other link can only be rotated about one axis. On the other hand, a cylindric joint has two degrees of freedom, since the moving link can be rotated and translated. Types of mechanism joints are summarized in Table 1.1 (Mabie and Reinholtz, 1987).

A mechanism whose motion can be represented fully in two-dimensional space is called a *planar mechanism.* If the motion of a mechanism takes place in three-dimensional space, it is referred to as a *spatial mechanism.* Although planar mechanisms are useful in many applications, most industrial robots are spatial mechanisms.

Table 1.1: Overview of Joint Types (adapted from Mabie And Reinholtz, 1987)

Joint Type	Physical Form	Schematic Representation		Degrees of Freedom
		Planar	Spatial	
Revolute (*R*)				1
Cylindric (*C*)				2
Prismatic (*P*)				1
Spheric (*S*)				3
Screw or Helical (*H*)				1

A *serial* device, or *open-loop* mechanism, consists of several links connected in sequence to form a single, open-ended chain. As shown in Fig. 1.1, the motion of the output link is a function of the motion of each joint in the chain.

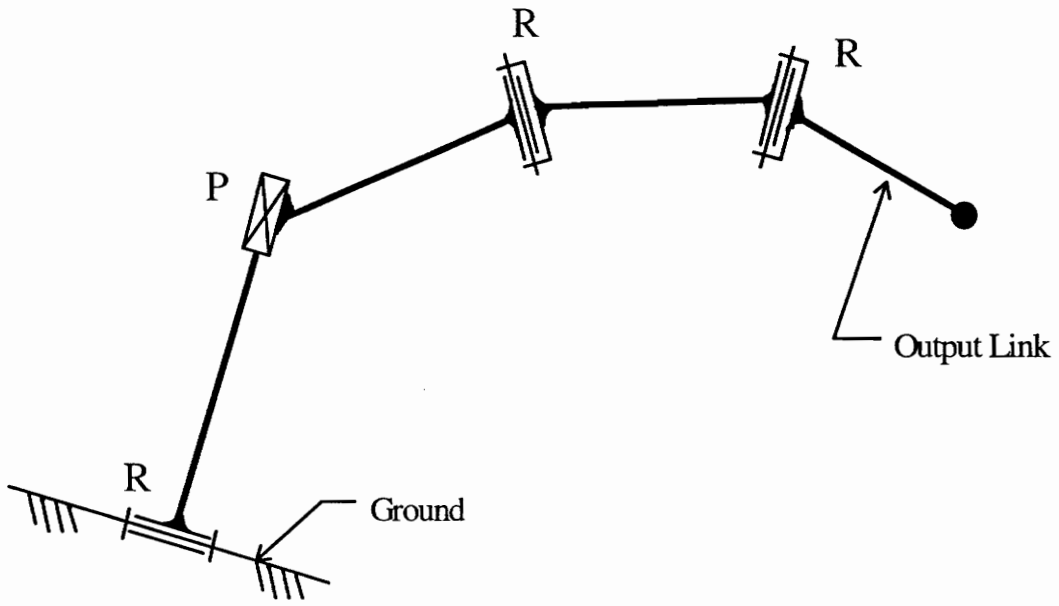


Fig. 1.1 - Kinematic Diagram of a Serial Configuration

A *closed-loop*, or *parallel-chain*, mechanism is one in which the output motion is controlled through the interaction of two or more kinematic chains connected in parallel. The connecting chains themselves can either be serial or parallel. Motion of the output can be achieved by moving the kinematic chains, either individually or in combinations. An example of a parallel device is shown in Fig. 1.2.

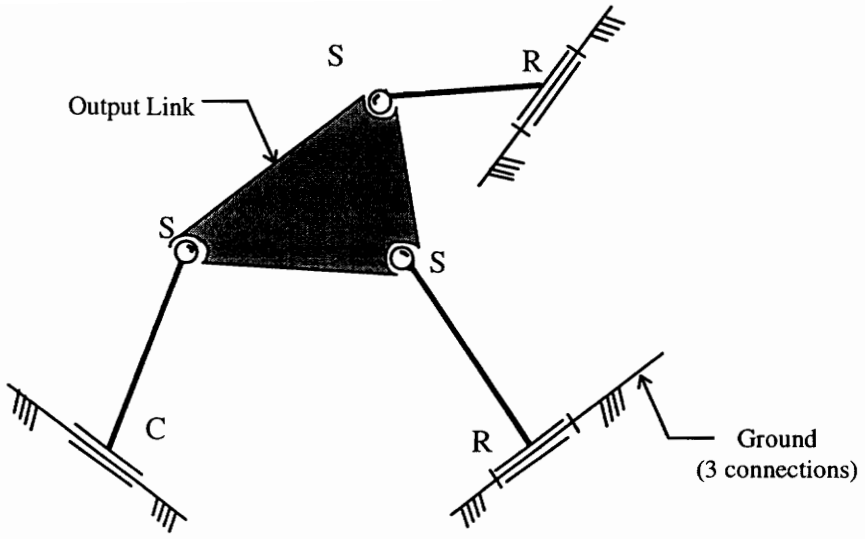


Fig. 1.2 - Kinematic Diagram of a Parallel Configuration

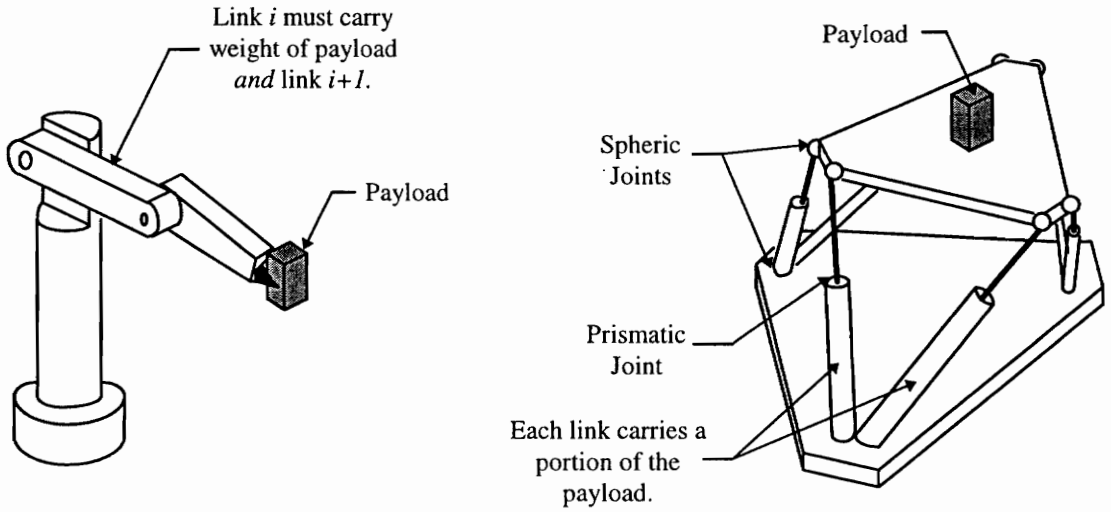
The number of degrees-of-freedom, or mobility, of a mechanism can be computed using the Kutzbach equation for spatial mechanisms

$$M = 6(n-1) - \sum_{j=1}^{5} (6-j) f_j \tag{1.1}$$

where n = the number of links in the mechanism (including the ground link) and f is the number of joints with j degrees-of-freedom (Mabie and Reinholtz, 1987). It should also be noted that a mechanism can have *idle* degrees-of-freedom, which have no effect on the output motion. Such is the case with two spheric joints connected by one link, in which the link can be rotated about the axis between spheric joint centers without moving the position of either spheric joint.

Until recently, most investigations into robot manipulator design were limited to serial devices. Similar in nature to human limbs, the motion of a serial device is intuitive. Since most industrial robot applications involve tasks previously done by humans, such

as spray painting, welding, component insertion and assembly, a device which replicated human motion was thought to be easier to implement. Unfortunately, serial devices suffer from poor strength-to-weight characteristics. When carrying a large payload at the output (end-effector), each link and joint in the kinematic chain must support the forces and torques due to the load, as illustrated in Fig. 1.3.



(a) Serial Manipulator

(b) Parallel Manipulator (Stewart's Platform)

Fig. 1.3 - Load Transmission in Serial vs. Parallel Manipulators

To counter the resulting static and dynamic effects, the links and joints must be very carefully designed, with significant stiffness and torque transmission capabilities. In an effort to reduce the weight of the manipulator, joint actuators are often located on the robot base, or ground link. This requires complex gear trains and linkage transmission systems, which can be difficult to design and can add mechanical backlash to the system, resulting in positioning errors at the end-effector. Designing serial manipulators to avoid

these problems, i.e., by using precision gearing, powerful actuators and high-performance materials, can add significantly to the cost of the robot.

Parallel devices have often been overlooked for use as industrial manipulators. Many designs of parallel manipulators lack the large workspace required to perform typical industrial tasks. Merlet (1994) and Uchiyama (1994) cite several advantages of parallel manipulators. An excellent payload-to-weight ratio is the most prominent. High stiffness permits greater positioning accuracy. End-effector position is less sensitive to error because the devices do not suffer from accumulation of actuator errors. In addition, parallel manipulators can be designed so that the actuators are attached to the ground link, reducing overall inertia of the moving components. However, the kinematic solutions for parallel manipulators are often not in closed-form, i.e. a solution must be found numerically. This can make parallel manipulators difficult to control and slower in operation.

1.1 Literature Review

The review of literature that follows is divided into three parts. In the first, a general review of previous work in robotic wrist design is presented. This material serves to establish the nature of the devices being discussed within this thesis, as well as the significance of the Carpal wrist to the state of the art. Although the work presented in this thesis is unique to the Carpal wrist, the second section aims to familiarize the reader with previous work in modeling kinematic error in mechanisms, with focus on robotic applications. The third and final section details work in solving nonlinear systems of equations, such as those that arise in the error model presented in Chapter 3.

1.1.1 Overview of Robotic Wrists

Traditionally, robotic wrists have been designed as serial, or open-chain, devices with intersecting joint axes. Craig (1989) and Holzbock (1986) define robotic wrists as usually having two or three degrees of freedom (d.o.f.), generally described in some combination of roll, pitch or yaw. With three orthogonal axes, any orientation of an object can theoretically be achieved. The intersecting, orthogonal axes also ensure a closed-form kinematic solution. This is beneficial, since it implies that the wrist can be attached to the end of a robot arm without significant computational penalty.

In a comprehensive overview of robotic wrist technology, Rosheim (1989) divides serial wrists into two categories, roll-pitch-roll and pitch-yaw-roll. These two configurations involve connecting revolute joints in series at different orientations, permitting three rotational degrees of freedom, as illustrated in Fig. 1.4. Roll-pitch-roll wrists are described as being more common in industry, but suffer from the drawback of a cone-shaped region of singularities in the workspace when the axes are not orthogonal. That is, when the wrist is moving away from a position where the two roll axes are aligned, a conical region is encountered in which velocity and torque become excessively high, resulting in a loss of dexterity, reduced dynamic performance or an unachievable position.

In presenting the mechanical design of the Carpal wrist, Ganino (1996) details several previous efforts using parallel devices for robotic wrists. Agrawal, et al (1994) describe a three-degree-of-freedom parallel actuated wrist device which is capable of pointing an axis anywhere within a cone-shaped work envelope, then rotating about that axis. Originally a component of a mechanical spine-type manipulator, the wrist was designed to be compact, but not to have a very large workspace. Agrawal's mechanism features two plates connected by three kinematic chains, one provided with rotary actuation and two with linear actuation. The device is suggested for use as a general robotic wrist, but little detail is given regarding its dexterity characteristics.

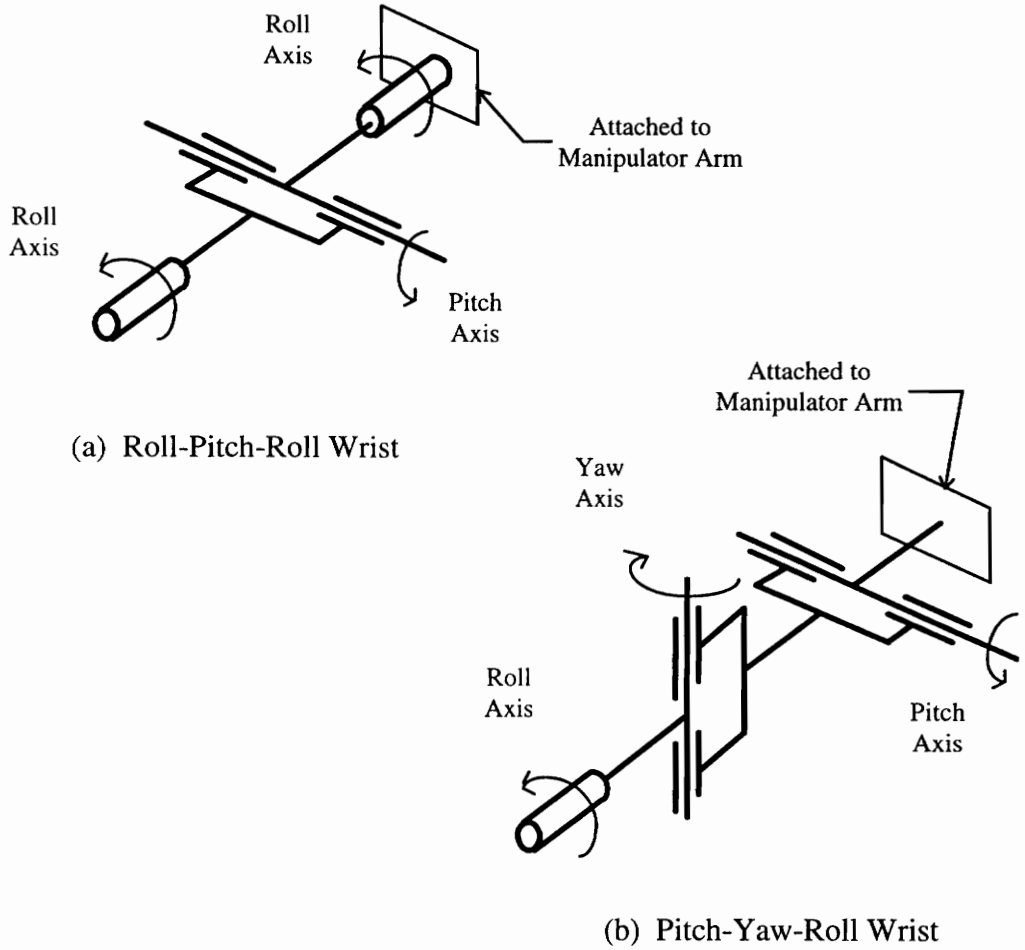


Fig. 1.4 - Roll-Pitch-Roll and Pitch-Yaw-Roll Kinematic Configurations

Milenkovic (1987) describes a singularity free wrist using an actuated double Carden joint design with pitch and yaw capability. Hudgens and Tesar (1988) propose a parallel device with six degrees-of-freedom to be used as micro-manipulator, placed in between the terminal link of the robot arm and the end-effector in order to produce high-resolution positioning. Rosheim (1989) presents a few parallel actuated wrist designs, notably one from the Association des Ouvriers en Instruments de Precision in France and one from the Cranfield Institute of Technology in England.

Fujimoto, et al. (1990) present the design of an octahedral wrist joint, which features six spheric-prismatic-spheric (SPS) chains connecting the fixed and moving plates. Actuation is provided via direct drive pneumatic cylinders mounted along the legs. In designing force-reflecting manual controllers for use on tele-operator robotic systems, Kim and Tesar (1990) study three spherical wrist types: a revolute-revolute-revolute (3R) serial device with intersecting revolute axes, a parallel device with two plates connected by three 3R chains, again with intersecting revolute axes, and a hybrid device combining two parallel mechanisms connected in series. The kinematics for each device are detailed, along with their advantages and disadvantages.

Nguyen, et al. (1991) describe another six degree-of-freedom wrist device, based on a Stewart platform shown in Fig. 1.3b (Stewart, 1965), for studying precision robotic assembly in a space environment. Lee and Chang (1992) utilize a parallel architecture wrist for a package handling system for the United Parcel Service, citing the benefits of large payload, high speed, high repeatability and large workspace. An echelon-form solution technique is presented by Innocenti and Parenti-Castelli (1993) for the forward kinematic problem of general parallel wrists, with examples for several different kinematic chain types connecting the upper and lower plates.

Hashimoto and Imamura (1994) introduce the design of a compliant parallel wrist having two rotational and one translational degree-of-freedom. The authors introduce passive mechanical compliance as a necessary feature when the robot is to perform interactive tasks, such as grinding and assembly. Hertz and Hughes (1994) present a survey of parallel devices, investigating Variable Geometry Trusses (VGTs) and double-tripod devices for use as robotic manipulators. The double-tripod mechanism, derived from a constant velocity coupling patented by Clemens (1872), is suggested as a three degree-of-freedom spatial pointing device. The authors present a solution technique for the forward and inverse kinematics for general configurations of both devices.

The Carpal wrist, shown in Fig. 1.5, was first presented by Salerno, et al. (1995) as an evolution of research into constant velocity couplings. The proposed wrist is a symmetric, parallel actuated three degree-of-freedom wrist resembling the Clemens coupling.

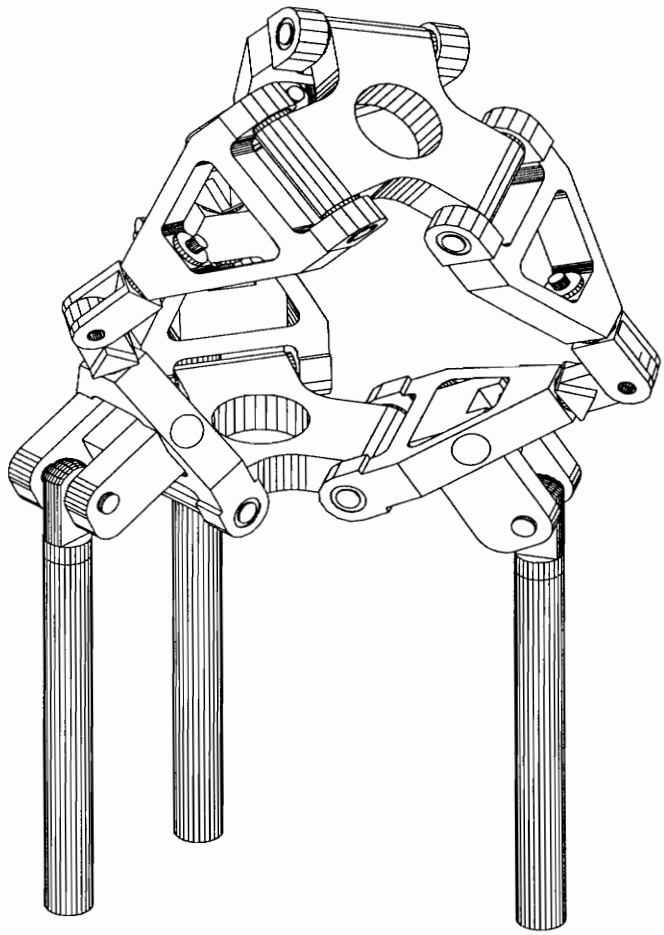


Fig. 1.5 - Carpal Wrist Prototype

The wrist is constructed from a basal plate and a distal plate, connected by three revolute-spheric-revolute (RSR) kinematic chains. The authors also suggest the addition of a

fourth degree of freedom which would involve a rotation of the entire assembly about the roll axis of the wrist. The Carpal wrist has the advantages of a high strength-to-weight ratio and a large dexterous workspace free of singularities, as well as a protected tunnel for routing cables and hoses (Canfield, et al., 1996). The detailed mechanical design of a proof-of-concept prototype was undertaken by Ganino (1996).

1.1.2 Sensitivity Analysis and Error Modeling

Sensitivity analysis and error modeling are expansive topics in the literature. In this section, the discussion is limited to publications dealing with mechanisms, robots and robotic wrists. Even with this limitation, the subject can be further reduced by the nature of the errors being examined. The work in this thesis deals only with kinematic errors, i.e., errors in the link lengths, offsets and other geometric attributes, in an effort to predict performance of a wrist with inclusion of manufacturing tolerances. With that in mind, the literature presented in this section deals with creation of the error models and presentation of the results.

Much has been written in the area of sensitivity analysis of planar mechanisms. Ting and Long (1994) employ the Taguchi method to achieve a robust design that is insensitive to dimensional variations rather than controlling the sources of the variations. They introduce a sensitivity Jacobian to relate the performance and dimensional tolerances, then introduce a sensitivity index to measure the sensitivity of performance quality to the dimensional tolerances of the entire system. The ideas are presented as generally applicable to any mechanism, although the example given involves synthesis of a planar four-bar linkage for rigid-body guidance.

In another application to planar four-bar mechanisms, Cleghorn, et al. (1993) present the error sensitivity analysis in two sections. In the first half of the analysis, the most error-sensitive link dimension is found using sensitivity indices. This establishes which link

will most effect the mechanism output for a given dimensional error. The concept is then extended for combinations of errors in the link lengths. With this information, the second portion of the analysis involves dimensional tolerance synthesis, which can be used to obtain the most accurate period within the input motion cycle, the best combination of input error parameters and the proper limits of tolerance on the inputs.

In the defining the Transmission Merit Parameter, Wu and Lankarani (1992) establish a quantity to relate the transmission quality and output sensitivity of a mechanism to a dimensional disturbance. In an example using a combined cam-linkage mechanism, the Transmission Merit Parameter is related to the mechanical advantage. The parameter is then generalized as the absolute value of the determinant of the Jacobian matrix for any planar mechanism. No application to spatial mechanisms or robotic manipulators is presented.

A stochastic error propagation technique is presented by Necsulescu, et al. (1994) for serial-type robot arms. A general model is developed for propagating errors with different distributions. The model assumes that the dependence of end-effector errors on geometric errors in the links is linear for a given configuration. Examples are given for both two- and three-degree-of-freedom manipulator arms.

Within the area of error modeling for parallel robot manipulators, Zhuang, et al. (1995) present the kinematic calibration of a Stewart platform. The Stewart platform, shown in Fig. 1.3b (Stewart, 1965), is a parallel mechanism consisting of six extensible legs connected to a fixed base via universal joints and to a movable top plate by spheric joints. In creating the error model, an assumption is made that the universal joints, ball joints and prismatic joints in the legs are all perfect. The kinematic errors considered are limited to those within the leg lengths, base plate and top plate dimensions. Due to the complexity of the forward kinematics of the Stewart platform, the authors utilize an error model based on the inverse kinematics and pose measurements for calibration.

Zobel and Di Stefano (1994) present a method to synthesize tolerances in spatial robots, focusing on the use of a parameter which balances the relation between positioning error and cost. The method is then applied to the Delta robot, a parallel, four degree-of-freedom robot used for light weight, repetitive motion applications (Clavel, 1988). The error model is constructed considering the geometric errors evident in link lengths and joint offsets.

Notash and Podhorodeski (1995) describe in great detail the construction of the error model for a three-branch parallel device. The manipulator is a six-degree-of-freedom hand controller consisting of a base plate connected to the top plate via three revolute-revolute-revolute-spheric (RRRS) kinematic chains. The identification model is constructed in two stages. The first stage uses only geometric parameter values provided by the robot manufacturer. The second, more complete, stage includes all possible joint offsets, out-of-plane conditions for the joints attached to the base and top plates, as well as link twisting between joints. Also addressed are methods in which the end-effector pose error can be expressed. If the end-effector position function is taken as the centroid of the three branch-end positions, the resulting error values will have some of the inaccuracies filtered out. Instead, the authors suggest using three separate error functions for each of the branch-end distances on the top plate.

1.1.3 Numerical Methods for Solving Nonlinear Equations

The kinematics of the Carpal wrist, fully described in Chapter 2, provide for a closed-form solution. Key to the existence of this closed-form solution is the geometric symmetry of the device. In formulating the model for inclusion of errors in the kinematic parameters, the assumptions of symmetry cannot be used. For this more general case, the kinematic model of the Carpal wrist yields a system of nine nonlinear equations in nine unknowns, which is derived using the three-dimensional distance equation of the form

$$(x-x_0)^2+(y-y_0)^2+(z-z_0)^2=d^2 \tag{1.2}$$

A detailed approach to the kinematic analysis will be presented in Chapter 3. The material below describes general approaches for solving systems of nonlinear equations.

Arun (1990) presents the solution to the kinematics of Variable-Geometry Truss (VGT) manipulators using homotopy continuation methods. Homotopy methods are based on mapping the known solution from one system of equations to that of the system in question. With $F(x) = 0$ being the system to which the solution is desired, and $G(x) = 0$ a simpler system of known solution, a homotopy path which transforms the trivial problem into the actual system can be found such that the following conditions are satisfied

$$H(\lambda,x)=\begin{cases}G(x) & \text{when } \lambda=0\\ F(x) & \text{when } \lambda=1\end{cases} \tag{1.3}$$

The objective of homotopy methods is to solve $H(\lambda,x) = 0$ numerically for x for increasing values of λ, through the range from the known solution at $\lambda = 0$ to the desired solution at $\lambda = 1$ (Coetzee and Stonick, 1995). When properly applied, homotopy will yield all the roots of a system of polynomial equations.

Burns (1994) presents the monomial method for solving systems of nonlinear equations by constructing a sequence of approximating monomial systems. Specifically, the system of nonlinear equations, $F_k(x_1,x_2,...,x_n) = 0$ with $k = 1, 2, \ldots, n$, is replaced with a system of positive single-termed polynomials of the form

$$C_k\prod_{j=1}^{n}x_j^{A_{jk}}=1 \tag{1.4}$$

where C_k is a vector of coefficients formed from weighted sums of the exponents of the equations in the original system. When transformed into log space using $z_j = \ln(x_j)$, the monomial system becomes linear and can be expressed as

$$\sum_{j=1}^{n} A_{jk} z_j = -\ln C_k \tag{1.5}$$

The resulting linear system can be easily solved, and the solution serves as the operating point for the next iteration. Burns (1994) claims such a method outperforms Newton's method. However, he cites problems which may arise in mechanism design, specifically when the nonlinear equations in the system have an identical exponent structure, such as the system of distance equations of the Error Model presented in Chapter 4. In such a case, the monomial method may fail because the weighted coefficients, C_k, may cause the corresponding rows of the linearized system to become linearly dependent.

One of the most common methods for solving systems of nonlinear equations is the Newton-Raphson method, often referred to as simply Newton's method or the method of tangents. Much literature exists on its derivation, performance characteristics and implementation. Acton (1970), Hostetter, et al. (1991), Strang (1986), and Volkov (1990) are some of the more comprehensive mathematics references. Haug (1989) and Mabie and Reinholtz (1987) present Newton-Raphson as an important tool in kinematics.

In general, a system of n equations in n unknowns can be expressed in the form

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \tag{1.6}$$

or in vector notation as $f(x) = 0$. Writing the Taylor-series expansion about the vector x, retaining only linear terms and setting each expression equal to zero yields

$$\begin{aligned} 0 &= f_1(x) + \frac{\partial f_1}{\partial x_1}\delta_1 + \frac{\partial f_1}{\partial x_2}\delta_2 + \cdots + \frac{\partial f_1}{\partial x_n}\delta_n \\ 0 &= f_2(x) + \frac{\partial f_2}{\partial x_1}\delta_1 + \frac{\partial f_2}{\partial x_2}\delta_2 + \cdots + \frac{\partial f_2}{\partial x_n}\delta_n \\ &\vdots \\ 0 &= f_1(x) + \frac{\partial f_1}{\partial x_1}\delta_1 + \frac{\partial f_1}{\partial x_2}\delta_2 + \cdots + \frac{\partial f_1}{\partial x_n}\delta_n \end{aligned} \tag{1.7}$$

where δ_i represents a small deviation from x_i (Acton, 1970). Letting the current approximation to the solution be x^k, Eq. (1.7) can be rearranged as

$$\Phi^k \delta^k = -f(x^k) \tag{1.8}$$

which has the form of a linear system, $Ax = b$. In Eq. (1.8), Φ^k is a Jacobian matrix of partial derivatives

$$\Phi^k = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{1.9}$$

and δ is the difference between the current iteration and the next iteration, $\delta = (x^{k+1} - x^k)$.

Algorithms which implement the Newton-Raphson method are numerous in the literature. In general, an initial "guess" of the solution is made. Commonly denoted x^0, it serves as a starting point for the iterative process. The system functions are evaluated at x^0. If the condition $f(x)=0$ is met, the process is stopped; the solution has been found. In practice, this step is performed with a small equation error tolerance, ε_e, such that $f(x)\leq\varepsilon_e$ is evaluated. If the initial guess is not the solution to the system, the Jacobian, Φ^k, is then evaluated and the linear system of Eq. (1.8) is solved. The next iteration begins with the system solution x^{k+1} computed from $\delta=(x^{k+1}-x^k)$. Convergence to the actual solution occurs when $\delta=0$; convergence has been shown to occur quadratically (Strang, 1986). As with the function evaluation, a solution error tolerance, i.e., $\delta=\varepsilon_s$, is often used as the stopping criteria instead of $\delta=0$. It should also be noted that if the Jacobian is singular when evaluated at the initial guess, x^0 , and $f(x)>\varepsilon_e$, the process must be started with a new guess (Haug, 1989).

The way in which the Newton-Raphson method works can easily be visualized for a single nonlinear equation of one variable (Strang, 1986; Haug, 1989). The function $g(x)$ has a solution x^* at the point where the curve crosses the horizontal, i.e., where $g(x) = 0$. Figure 1.6 below shows how a line is drawn tangent to $g(x)$ at the guess x^0. Where this line intersects the x-axis is the next value of solution, x^1. The process continues until the solution is reached.

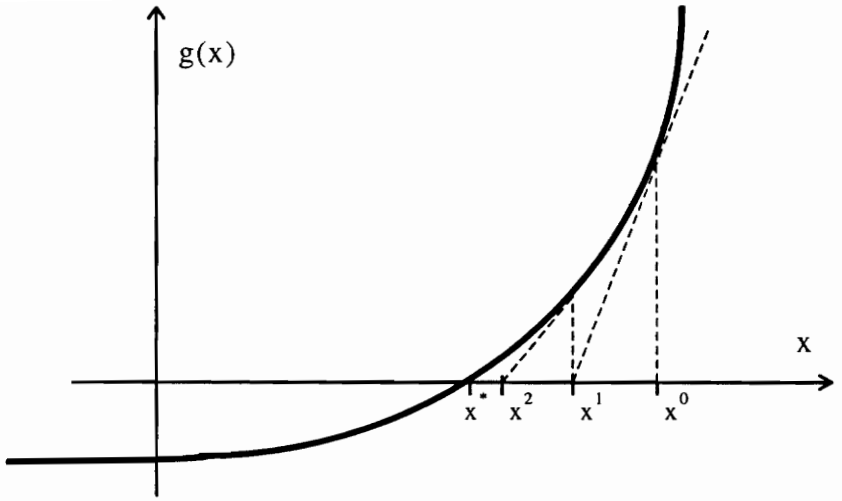


Fig. 1.6 - Graphical Representation of Newton-Raphson Method

Certain problems that arise with the Newton-Raphson method can also be illustrated using the one-dimensional case. The initial guess is critical, as it must be in the neighborhood of the actual solution for the method to work. Use of the Newton-Raphson method also depends on the function being easily differentiable, or it must be done numerically. Figures 1.7 through 1.9 show problems which may occur with multiple roots, divergence due to inflection points and due to local maxima or minima.

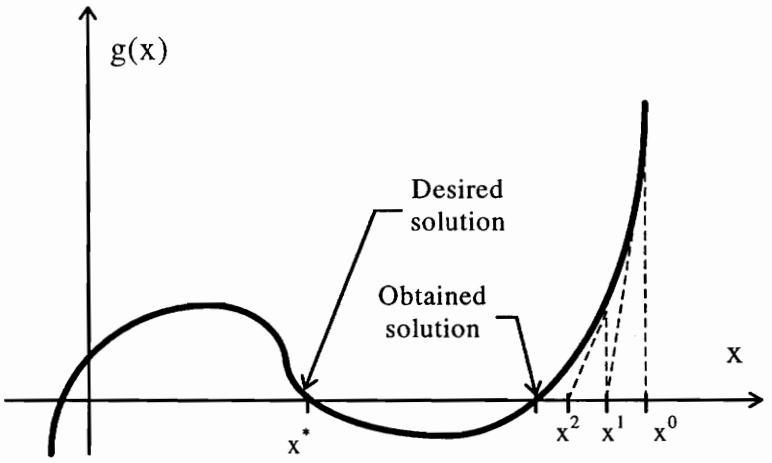


Fig. 1.7 - Function with multiple roots

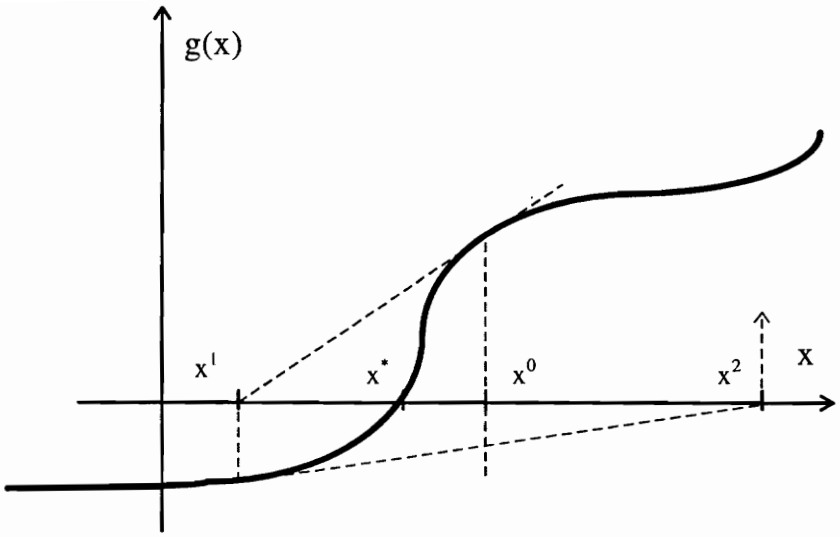


Fig. 1.8 - Divergence due to root at an inflection point

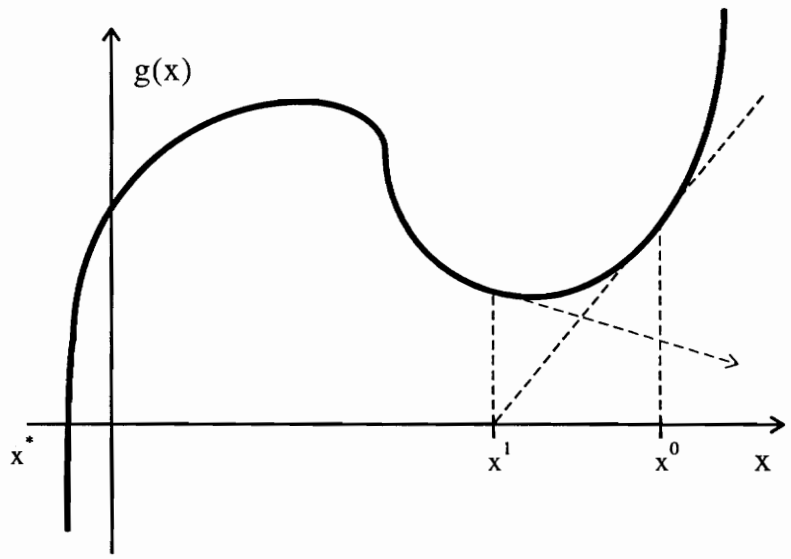


Fig. 1.9 - Divergence due to local maxima or minima

Many authors have proposed improvements to the Newton-Raphson method. Strang (1986) presents several slightly different approaches. The Modified Newton method, in

which only one Jacobian is used throughout all iterations, is useful if the system functions are not easily differentiable. In the Quasi-Newton method, also discussed extensively by Arun (1990), the Jacobian is updated at each iteration using step information instead of recomputing the derivatives, such as

$$J^{k+1} = J^k + \frac{(\Delta g - J^k \delta)\delta^T}{\delta^T \delta} \qquad \text{with} \quad \Delta g = g(x^{k+1}) - g(x^k) \tag{1.10}$$

Strang (1986) also suggests the method of steepest descent if the derivatives are impossible and no Jacobian can be found. In this case, the identity matrix, or a multiple thereof, is used. However, this method is often more useful for finding minima of a function, rather than roots.

Gerlach (1994) proposes achieving higher-order convergence through modifying the given functions. The premise is that the more the function looks linear, the faster the iterations will converge. Hence, the author suggests making the function look nearly linear in the neighborhood of the roots by applying higher-order derivative terms and a weighting factor. It also states that the weighting factor can be difficult to determine, and there are several criteria regarding existence of the higher-order derivatives.

Alefeld, et al. (1994) present a new iteration stopping criteria, which when satisfied uses the last three iterates of the Newton-Raphson method to compute an interval vector that should contain the solution. The technique combines the properties of the Krawczyk operator and the Newton-Kantorovich theorem. The resulting error of the approximation to the solution is shown to be on the order of machine precision. Nguyen (1993) suggests using a neural network architecture to improve performance. At each iteration, the linearized equation system is transformed to a quadratic function which is minimized to find the value of δ_k. The neural network approach achieves ultra high-speed

computations due to the parallel processing capability, and the computing time is independent of the system dimensions or structure.

Saraç, et al (1994) present an algorithm for solving systems of nonlinear equations with application to air-compressor system design. The procedure closely follows the conventional Newton-Raphson method in formation of the linearized system of equations $Ax = b$. An auxiliary matrix D is introduced based on the physical structure of the system. The introduction of D introduces a Gauss elimination into the iterative algorithm, and the linearized system of equations becomes

$$ADy = b, \text{ with } Dx = y \tag{1.11}$$

The authors also provide selection criteria for the matrix D, but they are not well described for general systems of equations.

There are several other methods for solving nonlinear systems which have not been covered. Those described here have been used extensively in mechanism problems. In Chapter 3, the Error Model for the Carpal wrist will be presented and it will be shown that the Newton-Raphson method can be easily applied to the problem. The system equations are easily differentiable, so an analytical expression for the Jacobian can be used, and the system is relatively small. The issue of an initial guess is resolved by using the solution from the ideal Carpal wrist kinematics, which should lie within the vicinity of the actual solution.

1.2 Motivation and Objective

The ultimate goal of the development of a device such as the Carpal wrist is its applicability and acceptance in industry. The contribution of the design to the field of kinematics has been proven in previous works (Salerno, et al., 1995; Canfield, et al., 1996). Error modeling of the Carpal wrist involves the creation of a tool to predict the effect of kinematic parameter errors on the workspace of the wrist. As an assessment of "how good" the device is, this work is intended to enable the designer to answer the following questions.

1. How does error in a kinematic parameter, or combinations of parameters, affect the workspace pose of the Carpal wrist?

2. Conversely, in the manufacture of the Carpal wrist, which components must be manufactured to the most critical tolerances?

By completing this investigation, it is hoped that this work will aid in further proving the uniqueness and importance of the device. The work will not only aid in more advanced "on paper" development, but will also influence the manufacture of future Carpal wrists.

Chapter 2

Kinematics of the Carpal Wrist

The kinematic analysis of the Carpal wrist evolved from research into the development of an all-revolute constant-velocity coupling (Canfield, et al. 1994; Canfield and Reinholtz, 1995). Salerno, et al (1995) proposed the geometry for use as an innovative robotic wrist with four degrees of freedom (pitch, yaw, plunge and tool rotation), and presented closed-form solutions to the forward and inverse kinematics. Ganino (1996) details the mechanical design of the Carpal wrist, as well as the fabrication of a "proof of concept" prototype. The prototype utilizes only three degrees of freedom, pitch, yaw and plunge. The work of this thesis also focuses on such a wrist. The forward and inverse kinematic analyses are presented to provide the basis for construction of the Error Model in Chapter 3 .

2.1 Forward Kinematics

The forward kinematics problem can be stated as follows: given a set of input angles, θ_1, θ_2, θ_3, and the fixed geometric dimensions of the device, find the position and orientation of the tool coordinate frame. Looking at the kinematic diagram of Fig. 2.1, as well as the top view of the basal plane in Fig. 2.2, the forward kinematic solution begins at the origin of the basal plate coordinate frame, denoted c_B. Note that in Fig. 2.1, the vectors b_i, m_i and d_i all originate at the basal center c_B, but are not drawn in to avoid confusion.

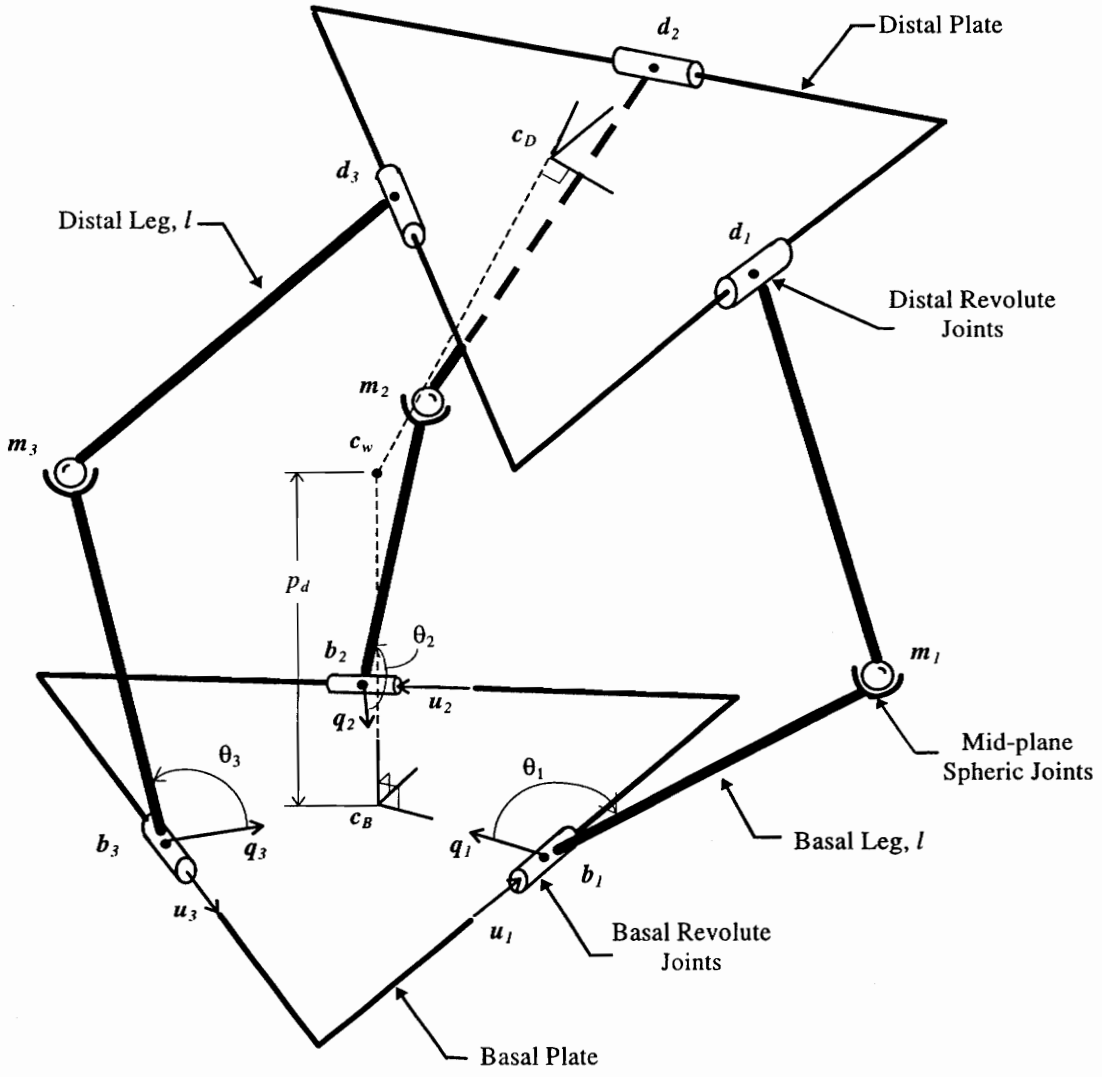


Fig. 2.1 - Kinematic Diagram of the Carpal Wrist

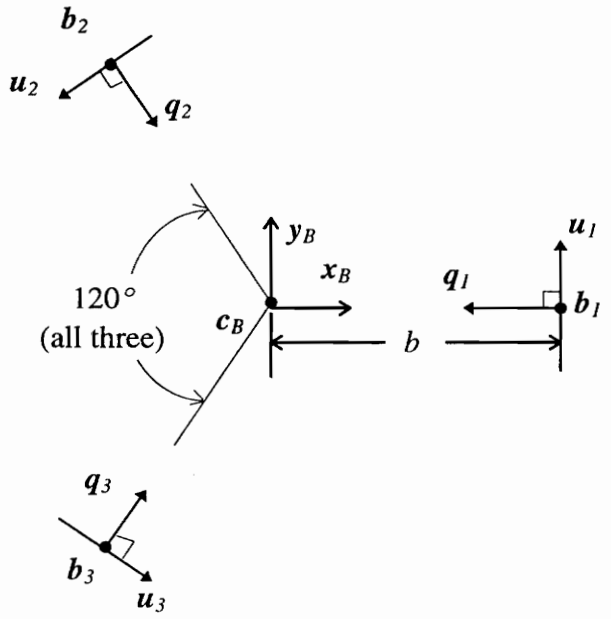


Fig. 2.2 - Top View of Basal Plane

The basal revolute joint centers are then located a distance b from the basal center, spaced 120° apart. The basal revolute joint axes are denoted by unit vectors, $\boldsymbol{u}_i$ (i = 1, 2, 3), the intersection of which forms an equilateral triangle. Unit vectors normal to the revolute axes point towards $\boldsymbol{c}_\mathbf{B}$ and are denoted $\boldsymbol{q}_i$.

In this work, the convention followed aligns the first revolute axis, $\boldsymbol{u}_1$, with the $\boldsymbol{y}_B$-axis of the basal coordinate frame, and $\boldsymbol{q}_1$ with the *negative* $\boldsymbol{x}_B$ to preserve the Cartesian system. All vectors are numbered from one to three in a clockwise sense viewed from the top. Vectors locating the revolute joint centers in the $\{B\}$-frame can then be written as

$$\boldsymbol{b}_i = b(-\boldsymbol{q}_i) \qquad i = 1,\ 2,\ 3 \tag{2.1}$$

A general rotation matrix which rotates a vector about vector $\boldsymbol{k}$ by an angle ϕ in the right-hand sense can be written as (Craig, 1989)

$$\mathbf{R}_{[k,\phi]} = \begin{bmatrix} k_x^2 v\phi + c\phi & k_x k_y v\phi - k_z s\phi & k_x k_z v\phi + k_y s\phi \\ k_x k_y v\phi + k_z s\phi & k_y^2 v\phi + c\phi & k_y k_z v\phi - k_x s\phi \\ k_x k_z v\phi - k_y s\phi & k_y k_z v\phi + k_x s\phi & k_z^2 v\phi + c\phi \end{bmatrix} \tag{2.2}$$

where $c\phi = \cos\phi$, $s\phi = \sin\phi$ and $v\phi = (1 - \cos\phi)$. The unit vectors, $\boldsymbol{q}_i$, can be rotated about the revolute axes, $\boldsymbol{u}_i$, by an amount of the input angle, θ_i, using this general rotation matrix. From the basal revolute joint centers, the location of the spheric joints, referred to as the mid-plane nodes, $\boldsymbol{m}_i$, can then be located using the leg length, l, such that, for $i = 1, 2, 3$

$$\boldsymbol{m}_i = \boldsymbol{b}_i + l\,\mathbf{R}_{[u_i,\theta_i]}\,\boldsymbol{q}_i \tag{2.3}$$

A set of mid-plane coefficients, which also defines a vector normal to the mid-plane, can be found from the vector cross product

$$\begin{bmatrix} A_m \\ B_m \\ C_m \end{bmatrix} = (\boldsymbol{m}_2 - \boldsymbol{m}_1) \times (\boldsymbol{m}_3 - \boldsymbol{m}_2) \tag{2.4}$$

and a unit vector normal to the mid-plane can be defined from the coefficients as

$$\boldsymbol{n}_m = \frac{\begin{bmatrix} A_m & B_m & C_m \end{bmatrix}^T}{\sqrt{A_m^2 + B_m^2 + C_m^2}} \tag{2.5}$$

With this information, the perpendicular distance from the basal revolute points to the mid-plane can be found as

$$\delta_i = \boldsymbol{n}_m \cdot (\boldsymbol{m}_i - \boldsymbol{b}_i) \tag{2.6}$$

An important assumption in the usual kinematic analysis of the Carpal wrist is that the device is symmetric about the mid-plane formed by the spheric joints, $\boldsymbol{m}_i$. This symmetry results in a closed-form forward kinematic solution. The distal plate revolute points, $\boldsymbol{d}_i$, are located from $\boldsymbol{c}_{\mathbf{B}}$ by

$$\boldsymbol{d}_i = \boldsymbol{b}_i + 2\delta_i \, \boldsymbol{n}_m \tag{2.7}$$

The distal plane is defined by the points $\boldsymbol{d}_i$, with the distal coordinate center, $\boldsymbol{c}_D$, located at the centroid of the equilateral triangle

$$\boldsymbol{c}_D = \tfrac{1}{3}(\boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3) \tag{2.8}$$

The distal frame, $\{D\}$, can then be described with coordinate axes defined from the distal revolute information, beginning with the outward pointing z_D-axis

$$z_D = \frac{(\boldsymbol{d}_2 - \boldsymbol{d}_1) \times (\boldsymbol{d}_3 - \boldsymbol{d}_2)}{\left|(\boldsymbol{d}_2 - \boldsymbol{d}_1) \times (\boldsymbol{d}_3 - \boldsymbol{d}_2)\right|} \tag{2.9}$$

and the corresponding x- and y-axes

$$\boldsymbol{x}_D = \frac{\boldsymbol{d}_1 - \boldsymbol{c}_D}{\left|\boldsymbol{d}_1 - \boldsymbol{c}_D\right|} \tag{2.10}$$

$$\boldsymbol{y}_D = \boldsymbol{z}_D \times \boldsymbol{x}_D \tag{2.11}$$

With the distal coordinate frame defined, a rotation matrix describing the relative rotation between $\{B\}$ and $\{D\}$ is the direction cosines of the joint axes

$$ {}^{B}_{D}\boldsymbol{R} = [\boldsymbol{x}_D \quad \boldsymbol{y}_D \quad \boldsymbol{z}_D] \tag{2.12} $$

and the overall transformation matrix can be expressed as

$$ {}^{B}_{D}\boldsymbol{T} = \left[\begin{array}{c|c} {}^{B}_{D}\boldsymbol{R} & \boldsymbol{c}_D \\ \hline 0 & 1 \end{array}\right] \tag{2.13} $$

From these expressions, the location of the end-effector, or tool, can be found given any values of the input angles, θ_i, and all the wrist dimensions.

2.2. Inverse Kinematics

The parallel architecture of the Carpal wrist is unique in that it provides two rotational and one translational degree-of-freedom (pitch, yaw and plunge), instead of the conventional three rotations. This presents some complication in the solution to the inverse kinematics problem, which can be stated as: *given an end-effector pose and a wrist plunge distance, find the input angles, θ_i , required to achieve the result.* The first step in the inverse solution procedure is to determine if the end-effector pose, or goal, is attainable within the wrist workspace. The goal can be restricted to three variables, corresponding to the two angular and one translational descriptors. According to Canfield and Reinholtz (1995), no relative rotation exists between the basal and distal planes. Therefore, no such rotation should be included in the goal specifications.

At any plunge distance, p_d , the wrist can be modeled as two links of length p_d connected by a spheric joint, as shown in Fig. 2.3. The goal point, which is the location of the center of the distal coordinate frame $\{D\}$, can then be described as a rotation, ϕ, from the axis $\boldsymbol{z}_B$ about an axis $\boldsymbol{u}_{bend}$. The axis $\boldsymbol{u}_{bend}$ is parallel to the $\boldsymbol{x}_B$-$\boldsymbol{y}_B$ plane and is located an

angular distance α from x_B. The angle ϕ is often referred to as the bend angle, while α is the bend-axis angle.

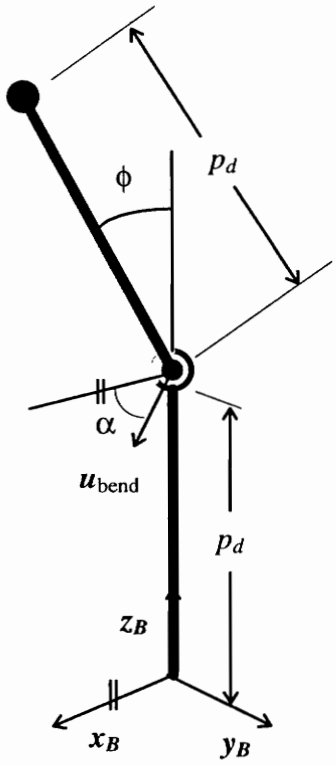


Fig. 2.3 - Inverse Kinematic Model

From the specified tool rotation matrix, the z_D-axis components can be obtained as the last column. Once the goal is specified, the bend axis and bend angle can be found from

$$u_{bend} = \frac{z_B \times z_D}{|z_B \times z_D|} \tag{2.14}$$

$$\phi = \cos^{-1}(z_B \cdot z_D) \tag{2.15}$$

With the wrist center, $\boldsymbol{c_w}$, as the location of the spheric joint in the two-link model, the location of the distal revolute joints, $\boldsymbol{d_i}$, can be found by rotating their undeflected positions, $\boldsymbol{d_i'}$, about the bend axis such that

$$\boldsymbol{d}_i = \boldsymbol{R}_{[u_{bend},\phi]}\boldsymbol{d}_i' + \boldsymbol{c}_w \tag{2.16}$$

where

$$\boldsymbol{d}_i' = \begin{bmatrix} b_{ix} \\ b_{iy} \\ p_d \end{bmatrix} \tag{2.17}$$

$$c_w = \begin{bmatrix} 0 \\ 0 \\ p_d \end{bmatrix} \tag{2.18}$$

With the distal revolute locations known, three points which must lie on the plane of symmetry, or mid-plane, can be found by bisecting the vector between the basal and distal revolute joints as follows

$$\boldsymbol{p}_i = \frac{\boldsymbol{d}_i + \boldsymbol{b}_i}{2} \tag{2.19}$$

Although these points do not represent the mid-joint locations in most cases, they do lie on the same plane. This enables the mid-plane coefficients to be found from the vector cross product.

$$\begin{bmatrix} A_m \\ B_m \\ C_m \end{bmatrix} = (\boldsymbol{p}_2 - \boldsymbol{p}_1) \times (\boldsymbol{p}_3 - \boldsymbol{p}_1) \tag{2.20}$$

$$D_m = -\begin{bmatrix} A_m \\ B_m \\ C_m \end{bmatrix} \cdot \boldsymbol{p}_1 \tag{2.21}$$

Since the basal and distal plates are joined by revolute-spheric-revolute (RSR) kinematic chains, it can be envisioned that the locus of possible spheric joint locations relative to both the basal and distal legs form circles which intersect at the mid-joints. With this in mind, the following constraint equations can be written:

$$\left|\boldsymbol{m}_i - \boldsymbol{b}_i\right| = l \tag{2.22}$$

$$A_{c_i} m_{ix} + B_{c_i} m_{iy} + C_{c_i} m_{iz} + D_{c_i} = 0 \tag{2.23}$$

$$A_m m_{ix} + B_m m_{iy} + C_m m_{iz} + D_m = 0 \tag{2.24}$$

where $\boldsymbol{m}_i$ is the vector locating mid-plane node i, having components m_{ix}, m_{iy} and m_{iz}, and A_c, B_c, C_c and D_c are plane coefficients of the circle formed by basal leg i. Equations (2.23) and (2.24) can be solved simultaneously for the y- and z-components of $\boldsymbol{m}_i$ in terms of m_{ix}, such that

$$m_{iy} = Rm_{ix} + S \tag{2.25}$$

$$m_{iz} = Tm_{ix} + U \tag{2.26}$$

where the coefficients are

$$R = \frac{A_{c_i} C_m - C_{c_i} A_m}{C_{c_i} B_m - B_{c_i} C_m} \tag{2.27}$$

$$S = \frac{D_{c_i} C_m - C_{c_i} D_m}{C_{c_i} B_m - B_{c_i} C_m} \tag{2.28}$$

$$T = \frac{B_{c_i} A_m - A_{c_i} B_m}{C_{c_i} B_m - B_{c_i} C_m} \tag{2.29}$$

$$U = \frac{B_{c_i} D_m - D_{c_i} B_m}{C_{c_i} B_m - B_{c_i} C_m} \tag{2.30}$$

The above relations can be substituted into Eq. 2.22, and expanded to solve for m_{ix} in the quadratic equation

$$Am_{ix}^2 + Bm_{ix} + C = 0 \tag{2.31}$$

where the coefficients are

$$\begin{aligned} A &= R^2 + T^2 + 1 \\ B &= 2\left(RS + TU - b_{ix} - Rb_{iy} - Tb_{iz}\right) \\ C &= S^2 + U^2 - 2Sb_{iy} - 2Ub_{iz} + b^2 + l^2 \end{aligned} \tag{2.32}$$

Equation (2.31) can then be solved using the quadratic formula, and the value of m_{ix} substituted into Eqs. (2.25) and (2.26) to obtain m_{iy} and m_{iz}. With the midplane nodes, $\boldsymbol{m}_i$, computed, the required basal leg input angles can be determined using the associated unit vector, $\boldsymbol{q}_i$, from the dot product

$$\theta_i = \cos^{-1}\left(\frac{(\boldsymbol{m}_i - \boldsymbol{b}_i)\cdot \boldsymbol{q}_i}{|\boldsymbol{m}_i - \boldsymbol{b}_i||\boldsymbol{q}_i|}\right) \tag{2.33}$$

Ganino (1996) notes that Eq. (2.31) results in two solutions for each of the three legs, for a total of eight possible mechanism closures. Also, Eq. (2.33) results in two solutions for each input angle, implying that a closure is reflected about the base. With this in mind, the solution with the mid-plane normal containing a positive z-component is chosen. For

the mechanism, two configurations are possible: an inward closure with the leg angles pointing toward the wrist center, and an outward closure in which the legs point away from the center. Table 2.1 below summarizes the choice of roots for the mid-nodes.

Table 2.1: Closure criteria for choice of mid-joint location

Leg Pair	Outward Closure	Inward Closure
1	$\max(m_{1x})$	$\min(m_{1x})$
2	$\min(m_{2x})$	$\max(m_{2x})$
3	$\min(m_{3x})$	$\max(m_{3x})$

Thus, the forward and inverse kinematics of the Carpal wrist are solved in closed-form. In Chapter 3, the error model will be developed in order to account for kinematic errors, which may occur due to manufacturing tolerances. It will be shown that the assumption of a plane of symmetry about which the distal plate is reflected is no longer valid. This will result in a set of equations for the forward kinematics which must be solved iteratively to find the tool pose.

Chapter 3

Error Model of the Carpal Wrist

In this chapter, the Error Model for the Carpal wrist is constructed based on a forward kinematic solution. As previously discussed, the term Error Model refers to a solution to the Carpal wrist which accounts for errors in the kinematic parameters, particularly geometric errors in the link lengths, basal and distal plates. Such errors would occur in an actual wrist as a result of manufacturing tolerances and assembly problems. Throughout the work, such a wrist will also be referred to as "non-ideal". This term is appropriate because the uniqueness of the Carpal wrist design is heavily dependent on three "ideal" qualities: basal and distal revolutes arranged about identical equilateral triangles, six legs of equal length, and a reflected tripod arrangement, symmetric about the mid-plane.

3.1 Forward Kinematics of the Error Model

Construction of the forward kinematic Error Model begins, as with the ideal Carpal wrist, at the center of the basal coordinate system, denoted by $\boldsymbol{c_B}$. For a given Carpal wrist, this center would be established based on the bolt pattern used to attach it to the manipulator arm. It is assumed that this frame would be rigidly attached to the end of the robot arm and that any misalignment in the attachment would be accounted for in an error model of the robot arm, not in that of the wrist. The forward kinematic analysis is divided into three sub-sections. The first section presents a method for determining the location of the mid-plane spheric joints, which is a relatively simple procedure. In the second section, a technique for locating the distal revolute joint centers is developed. In the third section a method for solving the resulting system of nonlinear equations is detailed.

3.1.1 Locating the Mid-plane Spheric Joints

The basal revolutes are again described with unit vectors $\boldsymbol{u_i}$ along their respective axes ($i = 1, 2, 3$), with unit vectors $\boldsymbol{q_i}$ normal to the axis vectors and pointing inward. As with the ideal model, the sign convention is such that the first revolute axis, $\boldsymbol{u_1}$, is aligned with the $\boldsymbol{y_B}$-axis of the basal coordinate frame, and $\boldsymbol{q_1}$ with the *negative* $\boldsymbol{x_B}$ to preserve the Cartesian system. In the ideal Carpal wrist, the revolutes were equidistant from the basal center and spaced at 120° about the center. For the non-ideal model, each revolute is located a distances b_i from $\boldsymbol{c_B}$, where $b_i = b_{ideal} + \delta b_i$. Additionally, the vectors $\boldsymbol{b_i}$ locating each revolute from the center are located at angles β_i, as shown in Fig. 3.1, where the sum of the angles β_i must be equal to 360°. Using the aforementioned sign convention, the vectors $\boldsymbol{u_i}$ and $\boldsymbol{q_i}$ can be expressed in terms of β_i as

$$\boldsymbol{u}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \boldsymbol{q}_1 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \tag{3.1}$$

$$\boldsymbol{u}_2 = \begin{bmatrix} -\sin\beta_1 \\ \cos\beta_1 \\ 0 \end{bmatrix} \qquad \boldsymbol{q}_2 = \begin{bmatrix} -\cos\beta_1 \\ -\sin\beta_1 \\ 0 \end{bmatrix} \tag{3.2}$$

$$\boldsymbol{u}_3 = \begin{bmatrix} \sin\beta_3 \\ \cos\beta_3 \\ 0 \end{bmatrix} \qquad \boldsymbol{q}_3 = \begin{bmatrix} -\cos\beta_3 \\ \sin\beta_3 \\ 0 \end{bmatrix} \tag{3.3}$$

and the $\boldsymbol{b}_i$ vectors can then be found from

$$\boldsymbol{b}_i = b_i(-\boldsymbol{q}_i) \tag{3.4}$$

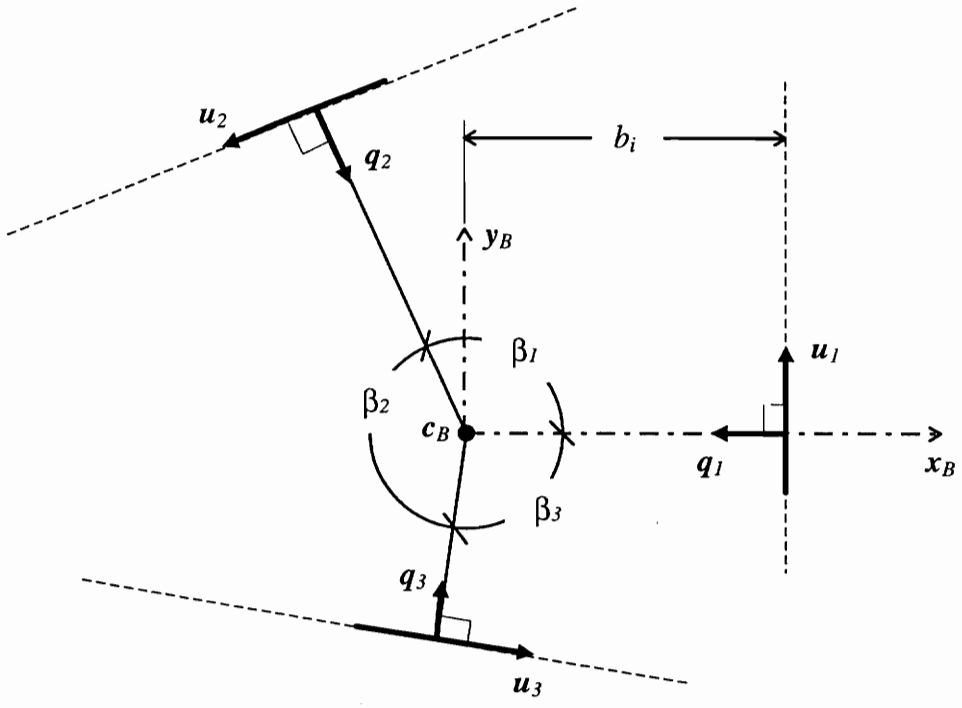


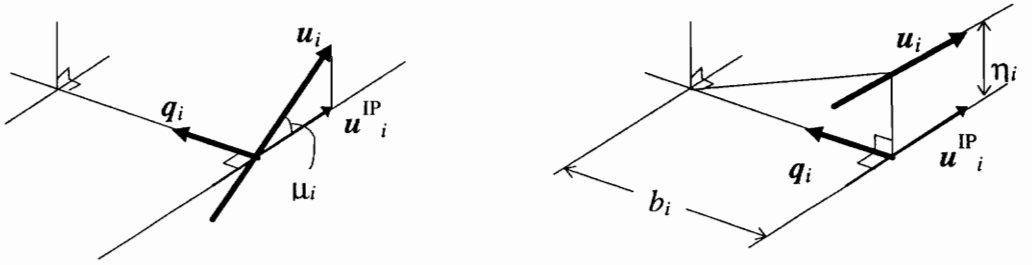
Fig. 3.1 - Top View of Non-Ideal Basal Plane

In the ideal Carpal wrist, the revolute joints and basal center are assumed to be co-planar. The Error Model can be extended to account for variations in this criteria. As shown in Fig. 3.2, if the vector $\boldsymbol{u}_i$ is rotated about $\boldsymbol{q}_i$ by an angular amount μ_i, an expression for $\boldsymbol{u}_i$ can be written as

$$\boldsymbol{u}_i = \boldsymbol{u}_i^{IP}\left(1 + tan\,\mu_i\right) \tag{3.5}$$

where $\boldsymbol{u}_i^{\text{IP}}$ is the in-plane revolute axis vector. If the vector $\boldsymbol{u}_i$ is translated above the basal plane by an distance η_i, as shown in Fig. 3.2, Eq. (3.4) becomes

$$\boldsymbol{b}_i = b_i(-\boldsymbol{q}_i) + \eta_i \boldsymbol{z}_B \tag{3.6}$$



(a) Out of Plane Tilt, μ_i (b) Out of Plane Offset, η_i

Fig. 3.2 - Basal plane with $\boldsymbol{u}_i$ out of plane

Another characteristic of the ideal Carpal wrist is that all leg lengths are equal. In the non-ideal model, the six leg dimensions will be denoted as the scalar l_i, $i = 1 \ldots 6$, to account for variation in their dimension, such that $l_i = l_{ideal} + \delta l_i$. Legs l_1, l_2 and l_3 connect to the respective ith basal revolutes. The upper legs, which will be used in the next section of the analysis, will be denoted l_{i+3}, that is, l_4 connects $\boldsymbol{m}_1$ to $\boldsymbol{d}_1$ and so on.

As with the ideal model, the mid-plane spheric joints, $\boldsymbol{m}_i$, are found by summing the vector locating the revolute joint and the $\boldsymbol{q}_i$ vector rotated by the input angle θ_i. The rotation matrix $\boldsymbol{R}_{[u_i,\theta_i]}$ is found using Eq. (2.2) For the non-ideal model, the mid-plane spheric joints are located by

$$\boldsymbol{m}_i = \boldsymbol{b}_i + l_i \boldsymbol{R}_{[u_i,\theta_i]}\boldsymbol{q}_i \tag{3.7}$$

3.1.2 Locating the Distal Revolute Joints

Having located the mid-plane spheric joints in the previous section, a method can now be presented locate the distal revolute joint centers, $\boldsymbol{d}_i$. It is important to recall that symmetry about the midplane is not assumed for the Error Model. The approach taken to find the distal revolute center points is somewhat similar to that used in solving Variable Geometry Truss (VGT) manipulators in that it uses geometric constraint equations for the upper-octahedral portion of the structure (Tidwell, 1989; Padmanabhan, et al., 1992). On the distal plate, the revolute points are connected by segments g_i, as shown in Fig 3.3.

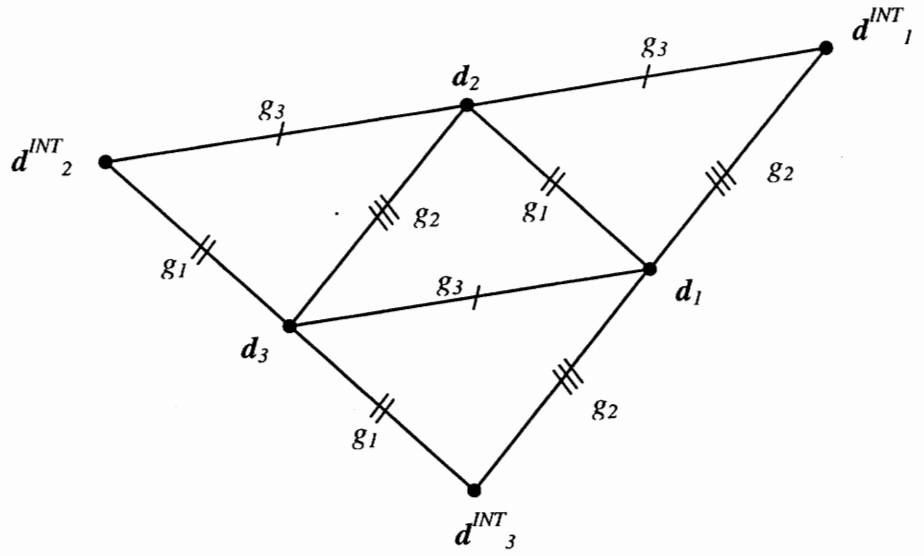


Fig 3.3 - Distal Plate Showing Segments g_i and Location of Distal Axis Intersections

Lines parallel to the revolute connectors can be drawn extending from the opposing revolute joint, as shown in Fig 3.3. By similar triangles, these parallel segments will also be of length g_i. The intersections of these "pseudo-axes" form a triangle, with vertices labeled $\boldsymbol{d}_i^{\text{int}}$ which will be located from the mid-joints in the following solution.

Looking at Fig 3.4, it can be seen that triangles can be constructed with vertices at mid-joints, $\boldsymbol{m}_i$, distal revolutes, $\boldsymbol{d}_i$, and distal axis intersections, $\boldsymbol{d}_i^{\text{int}}$. The resulting geometry is similar to an octahedral cell of a VGT, with the distal plane being analogous to the batten plane and the segments connecting the axis intersection points to the mid-joints analogous to longerons (Tidwell, 1989). The mid-plane is similar to the actuator plane of a double-octahedron, in that the points $\boldsymbol{m}_i$ are movable, but no battens connect the mid-joints in the Carpal wrist. The legs of each triangle are known dimensions, one being the previously defined g_i and the other the upper leg, l_{i+3}. If the upper legs are assumed to be perpendicular to the distal revolute axes, the hypotenuse can be computed by the Pythagorean Theorem to yield the edge dimensions of the constructed octahedron.

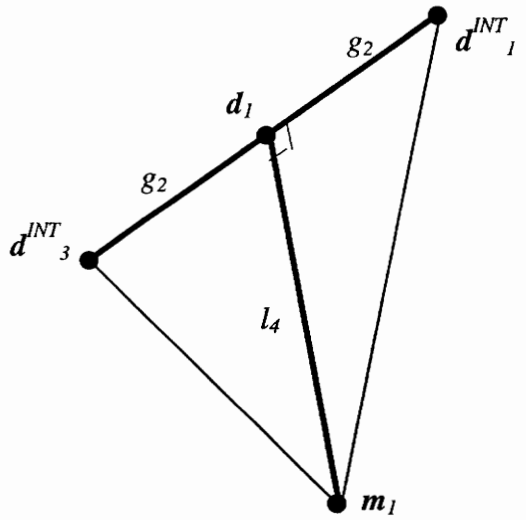


Fig 3.4 - Interconnection of Mid-joints and Distal Plane

From this geometry, a set of six distance equations can be written to describe the edges of the octahedron.

$$\left(d_{1x}^{\text{int}} - m_{1x}\right)^2 + \left(d_{1y}^{\text{int}} - m_{1y}\right)^2 + \left(d_{1z}^{\text{int}} - m_{1z}\right)^2 = l_4^2 + g_2^2 \tag{3.8}$$

$$\left(d_{3x}^{\text{int}} - m_{1x}\right)^2 + \left(d_{3y}^{\text{int}} - m_{1y}\right)^2 + \left(d_{3z}^{\text{int}} - m_{1z}\right)^2 = l_4^2 + g_2^2 \tag{3.9}$$

$$\left(d_{1x}^{\text{int}} - m_{2x}\right)^2 + \left(d_{1y}^{\text{int}} - m_{2y}\right)^2 + \left(d_{1z}^{\text{int}} - m_{2z}\right)^2 = l_5^2 + g_3^2 \tag{3.10}$$

$$\left(d_{2x}^{\text{int}} - m_{2x}\right)^2 + \left(d_{2y}^{\text{int}} - m_{2y}\right)^2 + \left(d_{2z}^{\text{int}} - m_{2z}\right)^2 = l_5^2 + g_3^2 \tag{3.11}$$

$$\left(d_{2x}^{\text{int}} - m_{3x}\right)^2 + \left(d_{2y}^{\text{int}} - m_{3y}\right)^2 + \left(d_{2z}^{\text{int}} - m_{3z}\right)^2 = l_6^2 + g_1^2 \tag{3.12}$$

$$\left(d_{3x}^{\text{int}} - m_{3x}\right)^2 + \left(d_{3y}^{\text{int}} - m_{3y}\right)^2 + \left(d_{3z}^{\text{int}} - m_{3z}\right)^2 = l_6^2 + g_1^2 \tag{3.13}$$

Three more distance equations can be written to describe the connections between the axis intersection points themselves, such that

$$\left(d_{2x}^{\text{int}} - d_{1x}^{\text{int}}\right)^2 + \left(d_{2y}^{\text{int}} - d_{1y}^{\text{int}}\right)^2 + \left(d_{2z}^{\text{int}} - d_{1z}^{\text{int}}\right)^2 = \left(2g_3\right)^2 \tag{3.14}$$

$$\left(d_{3x}^{\text{int}} - d_{2x}^{\text{int}}\right)^2 + \left(d_{3y}^{\text{int}} - d_{2y}^{\text{int}}\right)^2 + \left(d_{3z}^{\text{int}} - d_{2z}^{\text{int}}\right)^2 = \left(2g_1\right)^2 \tag{3.15}$$

$$\left(d_{1x}^{\text{int}} - d_{3x}^{\text{int}}\right)^2 + \left(d_{1y}^{\text{int}} - d_{3y}^{\text{int}}\right)^2 + \left(d_{1z}^{\text{int}} - d_{3z}^{\text{int}}\right)^2 = \left(2g_2\right)^2 \tag{3.16}$$

3.1.3 Solving the System of Equations

The resulting system, as described in the previous section with Eqs. (3.8) through (3.16), contains nine nonlinear equations in nine unknowns. As discussed in Chapter 1, many techniques exist for solving such systems. Of these, the Newton-Raphson method was selected based on three criteria:

1. Analytical expressions of the partial derivatives of each equation can be easily obtained to form a Jacobian matrix.
2. An excellent initial approximation, x^0, to the solution of the system exists, i.e. the solution from the ideal Carpal wrist.
3. The system is relatively small in size.

Referring to the procedure of the Newton-Raphson method presented in Section 1.1.3, the solution vector, $\boldsymbol{x}$, is defined as a column vector of the desired quantities, the distal axis intersection points, and can be written for this system as follows

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_9 \end{bmatrix}^T = \begin{bmatrix} d_{1x}^{\text{int}} & d_{1y}^{\text{int}} & d_{1z}^{\text{int}} & d_{2x}^{\text{int}} & d_{2y}^{\text{int}} & d_{2z}^{\text{int}} & d_{3x}^{\text{int}} & d_{3y}^{\text{int}} & d_{3z}^{\text{int}} \end{bmatrix}^T \quad (3.21)$$

In terms of the solution vector, Eqs. (3.8) through (3.16) can be rewritten in the form of Eq. (1.5). The functions can also be re-ordered such that the Jacobian will be in a better form for solution.

$$f_1 = (x_1 - m_{1x})^2 + (x_2 - m_{1y})^2 + (x_3 - m_{1z})^2 - (l_4^2 + g_2^2) = 0 \quad (3.22)$$

$$f_2 = (x_1 - m_{2x})^2 + (x_2 - m_{2y})^2 + (x_3 - m_{2z})^2 - (l_5^2 + g_3^2) = 0 \quad (3.23)$$

$$f_3 = (x_4 - x_1)^2 + (x_5 - x_2)^2 + (x_6 - x_3)^2 - (2g_3)^2 = 0 \quad (3.24)$$

$$f_4 = (x_4 - m_{3x})^2 + (x_5 - m_{3y})^2 + (x_6 - m_{3z})^2 - (l_6^2 + g_1^2) = 0 \quad (3.25)$$

$$f_5 = (x_4 - m_{2x})^2 + (x_5 - m_{2y})^2 + (x_6 - m_{2z})^2 - (l_5^2 + g_3^2) = 0 \quad (3.26)$$

$$f_6 = (x_7 - x_4)^2 + (x_8 - x_5)^2 + (x_9 - x_6)^2 - (2g_1)^2 = 0 \quad (3.27)$$

$$f_7 = (x_7 - m_{1x})^2 + (x_8 - m_{1y})^2 + (x_9 - m_{1z})^2 - (l_4^2 + g_2^2) = 0 \quad (3.28)$$

$$f_8 = (x_7 - m_{3x})^2 + (x_8 - m_{3y})^2 + (x_9 - m_{3z})^2 - (l_6^2 + g_1^2) = 0 \quad (3.29)$$

$$f_9 = (x_1 - x_7)^2 + (x_2 - x_8)^2 + (x_3 - x_9)^2 - (2g_2)^2 = 0 \tag{3.30}$$

An analytical expression for the nine-by-nine Jacobian matrix (Eq. 1.8) can then be written as

$$\Phi = \begin{bmatrix}
2(x_1 - m_{1x}) & 2(x_2 - m_{1y}) & 2(x_3 - m_{1z}) & & & & & & \\
2(x_1 - m_{2x}) & 2(x_2 - m_{2y}) & 2(x_3 - m_{2z}) & & & & & & \\
-2(x_4 - x_1) & -2(x_5 - x_2) & -2(x_6 - x_3) & 2(x_4 - x_1) & 2(x_5 - x_2) & 2(x_6 - x_3) & & & \\
 & & & 2(x_4 - m_{3x}) & 2(x_5 - m_{3y}) & 2(x_6 - m_{3z}) & & & \\
 & & & 2(x_4 - m_{2x}) & 2(x_5 - m_{2y}) & 2(x_6 - m_{2z}) & & & \\
 & & & -2(x_7 - x_4) & -2(x_8 - x_5) & -2(x_9 - x_6) & 2(x_7 - x_4) & 2(x_8 - x_5) & 2(x_9 - x_6) \\
 & & & & & & 2(x_7 - m_{1x}) & 2(x_8 - m_{1y}) & 2(x_9 - m_{1z}) \\
 & & & & & & 2(x_7 - m_{3x}) & 2(x_8 - m_{3y}) & 2(x_9 - m_{3z}) \\
2(x_1 - x_7) & 2(x_2 - x_8) & 2(x_3 - x_9) & & & & -2(x_1 - x_7) & -2(x_2 - x_8) & -2(x_3 - x_9)
\end{bmatrix}$$

where the blank matrix elements are equal to zero.

As mentioned earlier, an initial guess with which to begin the Newton-Raphson solution technique can be made using the information from the ideal Carpal wrist. Although the intersection points of the distal revolute axes, $\boldsymbol{d}_i^{int}$, are not solved for in the ideal wrist kinematics, they can easily be found by simple geometry. Recall that for the ideal wrist, the distal plate is an equilateral triangle with dimension b from center to each revolute joint. As shown in Fig. 3.5, the distance interconnecting the revolute joints is $\sqrt{3}b$. Since the triangle $\Delta \boldsymbol{d}_i \boldsymbol{d}_i^{int} \boldsymbol{d}_{i+1}$ is also equilateral, the distance from the revolute points to the axis intersection points is also $\sqrt{3}b$. The ideal distal revolute unit vectors can be found from the following expressions

$$\boldsymbol{u}_{d1} = \frac{\boldsymbol{d}_2 - \boldsymbol{d}_3}{|\boldsymbol{d}_2 - \boldsymbol{d}_3|} \qquad \boldsymbol{u}_{d2} = \frac{\boldsymbol{d}_3 - \boldsymbol{d}_1}{|\boldsymbol{d}_3 - \boldsymbol{d}_1|} \qquad \boldsymbol{u}_{d3} = \frac{\boldsymbol{d}_1 - \boldsymbol{d}_2}{|\boldsymbol{d}_1 - \boldsymbol{d}_2|} \tag{3.32}$$

The intersection of the distal revolute axes, which coincide with the vertices of the larger equilateral triangle, can be found from vector addition as

$$\boldsymbol{d}_i^{int,ideal} = \boldsymbol{d}_i + \left(\sqrt{3}b\right)\boldsymbol{u}_{d_i} \tag{3.33}$$

These ideal intersection points can then be used as the initial approximation to the solution of the system of equations which describe the distal point locations for the non-ideal wrist.

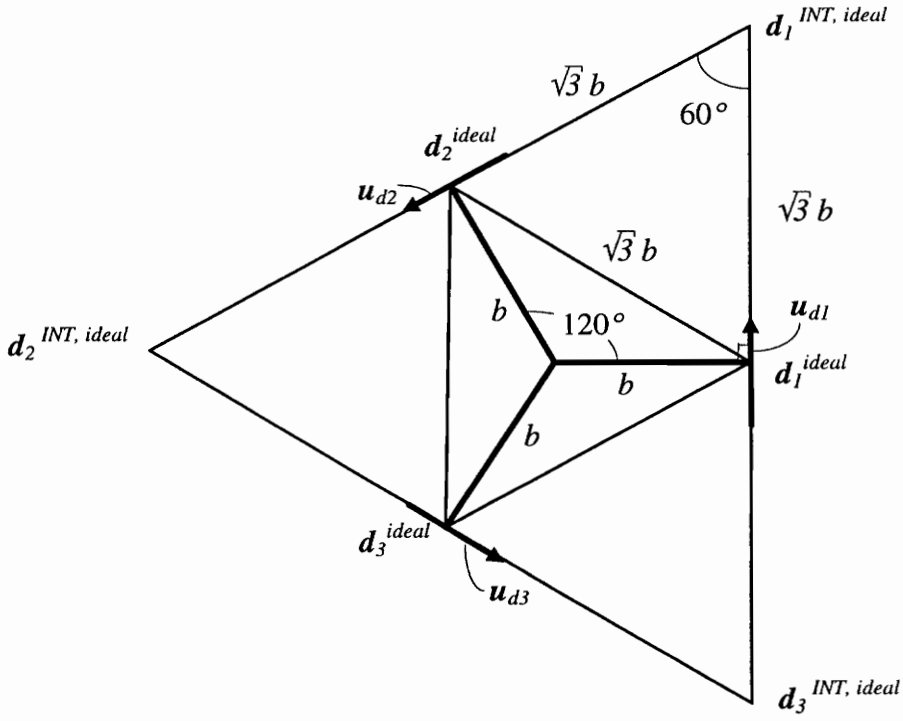


Fig. 3.5 - Ideal Carpal Wrist Distal Axis Intersection Points

Once the Newton-Raphson technique is used to find the solution for the distal axis intersection point locations of the non-ideal wrist, the distal revolute center locations, $\boldsymbol{d}_i^{actual}$, can be determined. Referring back to Fig 3.3, the procedure involves first finding

unit vectors connecting the intersection points, then performing a vector addition using the g_i dimensions defined earlier. The points are located using the following equations

$$\boldsymbol{d}_1^{actual} = \boldsymbol{d}_3^{\text{int}} + g_2 \left(\frac{\boldsymbol{d}_1^{\text{int}} - \boldsymbol{d}_3^{\text{int}}}{\left| \boldsymbol{d}_1^{\text{int}} - \boldsymbol{d}_3^{\text{int}} \right|} \right) \tag{3.34}$$

$$\boldsymbol{d}_2^{actual} = \boldsymbol{d}_1^{\text{int}} + g_3 \left(\frac{\boldsymbol{d}_2^{\text{int}} - \boldsymbol{d}_1^{\text{int}}}{\left| \boldsymbol{d}_2^{\text{int}} - \boldsymbol{d}_1^{\text{int}} \right|} \right) \tag{3.35}$$

$$\boldsymbol{d}_3^{actual} = \boldsymbol{d}_2^{\text{int}} + g_1 \left(\frac{\boldsymbol{d}_3^{\text{int}} - \boldsymbol{d}_2^{\text{int}}}{\left| \boldsymbol{d}_3^{\text{int}} - \boldsymbol{d}_2^{\text{int}} \right|} \right) \tag{3.36}$$

3.2 Definition of Error Quantity

In constructing the Error Model for the Carpal wrist, much consideration was given to what quantities would be compared to the ideal wrist in order to evaluate "error". Initially, it was felt that a comparison would be made based upon the location of the distal center, $\boldsymbol{c_D}$. The forward kinematic solution developed in the previous section does not locate the distal center for the non-ideal wrist. The reasoning behind this is as follows. It is assumed that, for a given Carpal wrist, the actual values of the kinematic parameters, i.e., the values of $l_i^{\ actual}$, $b_i^{\ actual}$, etc. , might be found by mechanical measurements. If one were to be presented with a distal plate from the wrist, finding the center of the plate would be more difficult than finding the revolute joint locations.

The output pose can be expressed in six quantities: three for position and three for orientation. The pose error would then be the difference between the actual and ideal values of these quantities. However, a convenient method does not exist for plotting all of these quantities over the three-dimensional workspace of the wrist, i.e., versus the bend angle and bend axis angle. Therefore, the results are compared by considering a vector from each actual distal revolute joint center to its corresponding ideal location. Summing the magnitudes of these three vectors gives

$$E = \sum_{i=1}^{3} \left\| \boldsymbol{d}_i^{actual} - \boldsymbol{d}_i^{ideal} \right\| \tag{3.37}$$

which incorporates the position and orientation, or pose, error into one scalar quantity. This error can then be computed over the entire workspace, for a variety of non-ideal wrist configurations. Results of such analyses will be presented in Chapter 4.

3.3 Overview of Computer Solution Techniques

The iterative solution to the forward kinematics of the Error Model is easily implemented using computer programs. Figure 3.6 shows a flow diagram of the computer solution technique. With the kinematic parameters and variations established, the procedure begins with first choosing a goal point (the bend angle, ϕ, and bend axis angle, α, introduced in Chapter 2). At this value, the inverse kinematics of the ideal Carpal wrist are solved to obtain the required input angles, θ_1, θ_2, and θ_3, to attain the goal point. These input angles are then used for the forward kinematic solution of the non-ideal model, which gives the location of the distal plane. The ideal and actual values are then compared, and the process is repeated for the next goal point in the workspace. For this work, the Matlab software package from The MathWorks was used for computer implementation. All of the Matlab code is included in Appendix A.

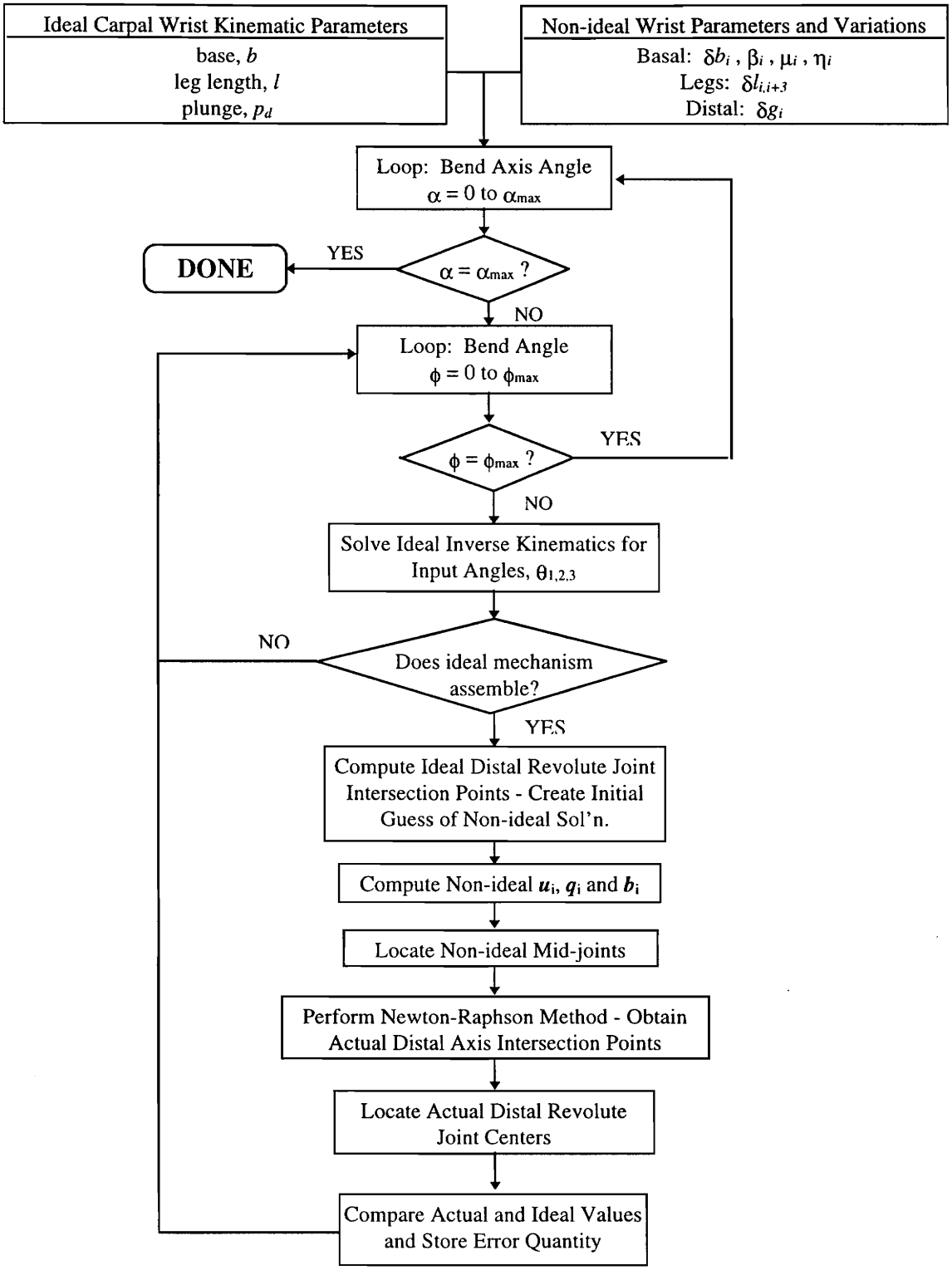


Fig. 3.6 - Computer Solution Technique Flow Diagram

Chapter 4

Results of Carpal Wrist Error Modeling

In Chapter 3, an error model was developed for the Carpal wrist based on a forward kinematic solution that accounts for errors in the kinematic parameters. This chapter will present examples demonstrating the use of the error model as a design and evaluation tool. In the first section, kinematic ratios are introduced to describe different wrist configurations. Values of the parameter errors are categorized in the second section, based on manufacturing tolerances for the Carpal wrist prototype. The third section will present the results of using the error model with errors in the individual parameters. In the last section, superposition will be used to extend the error model to combinations of parameter errors.

4.1 Kinematic Ratios

Two kinematic ratios for the double-octahedral parallel mechanism were defined by Canfield and Reinholtz (1995) in an investigation into constant velocity couplings. Similar ratios are utilized in the design of the Carpal wrist, as detailed by Salerno, et al. (1995) and Ganino (1996). Recall, from Figs. 2.1 and 2.3, that the distance from the basal center, $\boldsymbol{c_B}$, to the wrist center, $\boldsymbol{c_w}$, is defined as the plunge distance, p_d, while the leg length, l, is the distance from a mid-plane spheric joint to either a basal or distal revolute joint. The plunge ratio, R_d, is used as a method of quantifying the spherical shells within the volumetric workspace. The plunge ratio relates plunge distance to leg length as

$$R_d = \frac{p_d}{l} \tag{4.1}$$

The base dimension, b, is defined as the distance from the center point, $\boldsymbol{c_B}$, to the revolute joints, $\boldsymbol{b_i}$. The base-to-leg ratio, R_b, can be considered a quantity describing the geometric structure of the wrist, such that

$$R_b = \frac{b}{l} \tag{4.2}$$

Figure 4.1 illustrates the workspace at three different plunge ratios for a fixed base-to-leg ratio. Each value of the plunge ratio produces a partial spherical surface with radius equal to the plunge distance. The size of the spherical shell represents the workspace achieved by the pitch and yaw motions of the mechanism.

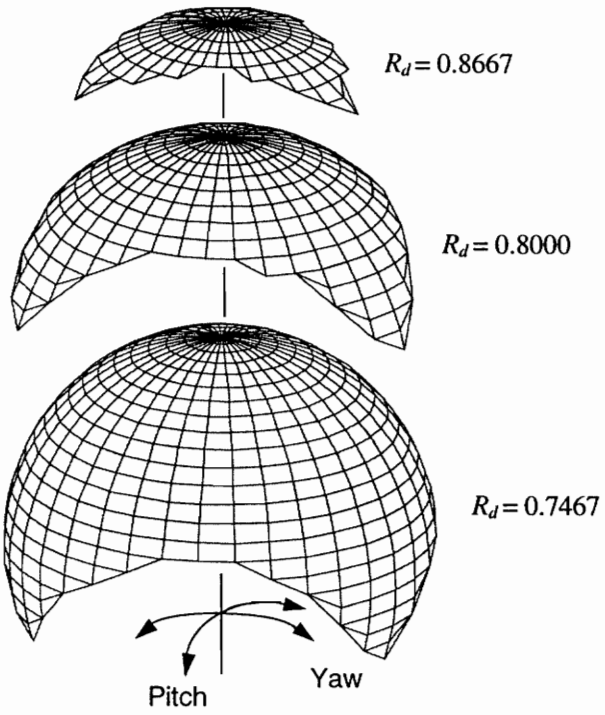


Fig. 4.1 - Spherical Workspace Shells for the Carpal Wrist with $R_b = 0.667$ and Various Values of R_d

The kinematic ratios of the Carpal wrist directly effect the achievable solid angle workspace for a given geometric configuration. Solid angle workspace, as illustrated in Fig. 4.2, refers to the volume created by rotating an angle about an axis, in this case about the basal z-axis, to form a conical section of a sphere of radius p_d, with vertex at the center and base formed from the portion of the spherical surface. A large solid angle workspace is important in a robotic wrist because it allows for many orientations to be achieved without moving the positioning structure, or robot arm.

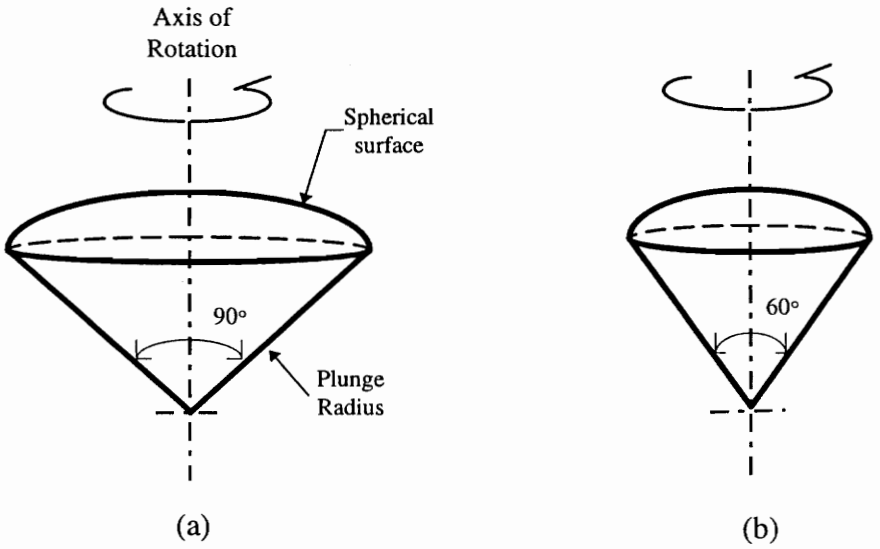


Fig. 4.2 - Examples of Solid Angles, (a) 90° and (b) 60° Workspaces

Canfield and Reinholtz (1995) describe how the kinematic ratios can be related to the solid angle workspace, as shown by Fig. 4.3. This relationship was also utilized in design of a Carpal wrist proof-of-concept prototype, aiding in the selection of an appropriate starting configuration for the iterative design process (Ganino, 1996). In general, the ratios are a convenient way to describe the wrist geometry and workspace capability, and are used as such within this chapter.

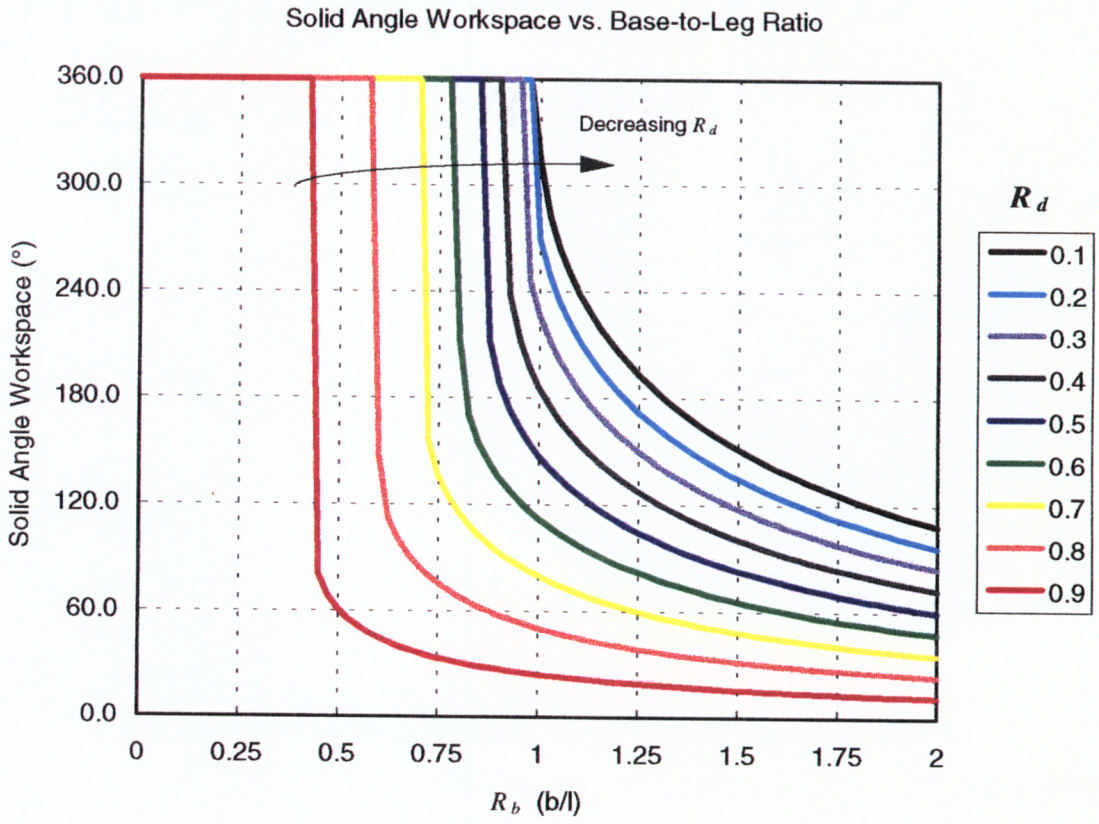


Fig. 4.3 - Effects of Kinematic Ratios on Maximum Solid Angle Workspace of the Carpal Wrist (Adapted from Canfield and Reinholtz, 1995)

4.2 Parameter Error Values

The error model developed in Chapter 3 is a forward kinematic solution for the non-ideal Carpal wrist with provision for errors in the kinematic parameters. Such variations would arise due to manufacturing tolerances and assembly errors. Analyses using the error model were performed for errors described as percentages of the nominal parameters. Using a percentage-based set allows for a more general application of this work as one of

the Carpal wrist design and evaluation tools. Regardless of the specific dimensions of the device, if the kinematic ratios and parameter error percentage values are known, an estimate of pose error can be found by looking at this work without conducting a specific analysis. As a starting point, the tolerances specified in design of the Carpal wrist prototype described by Ganino (1996) were used to establish the percentages. The design tolerances values are shown below in Table 4.1.

Table 4.1: Values of Kinematic Parameter Variations (from Ganino, 1996)

Parameter	Variable	Nominal Value	Maximum Error	Percentage
Basal Plate Dimension	b	3 cm (1.18 in.)	± 0.013 cm (± 0.005 in.)	0.43
Basal Revolute Locator Angle	β	120°	$\pm 0.5°$	0.42
Basal Revolute Vertical Offset	η	0	± 0.013 cm (± 0.005 in.)	0.43
Basal Revolute Tilt Angle	μ	0	$\pm 0.5°$	0.42
Leg Length	l	8 cm (3.15 in.)	± 0.033 cm (± 0.013 in.)	0.41
Distal Revolute Connector	g	5.19 cm (2.04 in.)	± 0.025 cm (± 0.010 in.)	0.48

It should be noted that the kinematic leg length error represents the "worst case" due to stack-up of tolerances from the assembly of the leg, a flanged bushing and the mid-joint. This is also reflected in the percentage of the parameter. It should also be noted that percentages are assigned to the Basal Revolute Vertical Offset and Tilt Angle based on the nominal basal dimension b and β. For other Carpal wrist configurations, the parameter values would be dependent on the specified tolerances of the manufactured

parts and purchased components. Percentages could then be established from the tolerances.

4.3 Implementation of the Carpal Wrist Error Model

Ganino (1996) introduced an iterative design methodology for the Carpal wrist based on the kinematic ratios R_b and R_d described in Section 4.1. A prototype is constructed with ratios R_b = 0.375 and R_d = 0.875 in order to achieve a 180° solid angle workspace. Theoretically, these ratios yield a 360° solid angle workspace, as shown in Fig. 4.4. However, in the prototype design, consideration was given to the physical limitations of the components, i.e., joint interference would prevent the distal plate from folding fully underneath the basal plate to reach the lower regions of the sphere. The wrist with ratios R_b = 0.375 and R_d = 0.875 will be used to validate the error model, with consideration given to the entire 360° solid angle workspace.

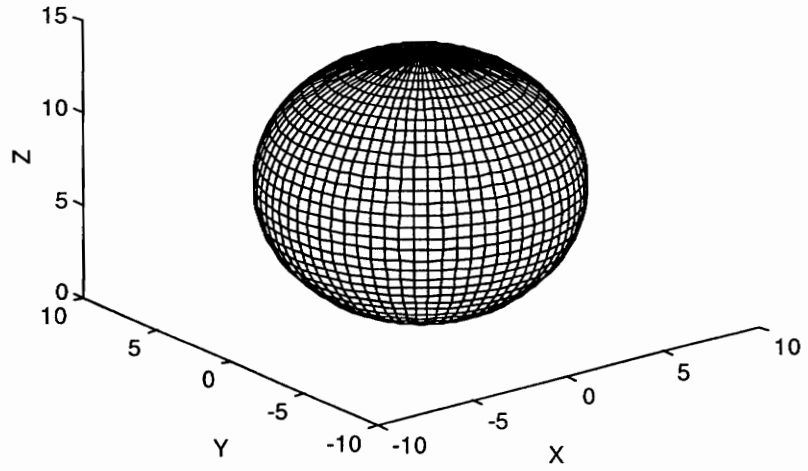


Fig. 4.4 - Ideal Workspace of Carpal Wrist, R_d = 0.875 and R_b = 0.375 (axes units = cm.)

Before continuing with the validation and examples of the error model, the concepts of bend axis angle and bend angle will be reviewed. The inverse kinematic solution of the Carpal wrist, presented in Chapter 2, defines the bend axis angle, α, as a right-hand rotation from the x-axis which locates the bend axis, $\boldsymbol{u}_{\text{bend}}$. The bending motion is then defined as a rotation by an amount ϕ, the bend angle, about the bend axis. As an example, bending motion at $\alpha = 0°$ indicates that the wrist is bending about the positive x-axis. At $\alpha = 90°$, the wrist is bending about the positive y-axis in the direction of the first basal revolute joint, $\boldsymbol{b}_1$. These concepts are illustrated by Fig. 4.5. The importance in restating this concept is that the error plots are presented as error versus α and ϕ, where the magnitude of the error is displayed using a color scale.

As the examples will illustrate, the affects of introducing parameter errors vary considerably over the workspace sphere. The output pose error, as described by Eq. (3.37), will be presented in a workspace plot as error versus bend angle, ϕ, and bend axis angle, α. Additionally, the error will be summarized for each case in the following ways:

1. Maximum error value in the entire workspace.
2. Average error value within the design specified 180° solid angle workspace ($\phi = 0$ to 90°, $\alpha = 0$ to 360°).
3. Amount of workspace surface area lost due to parameter error.

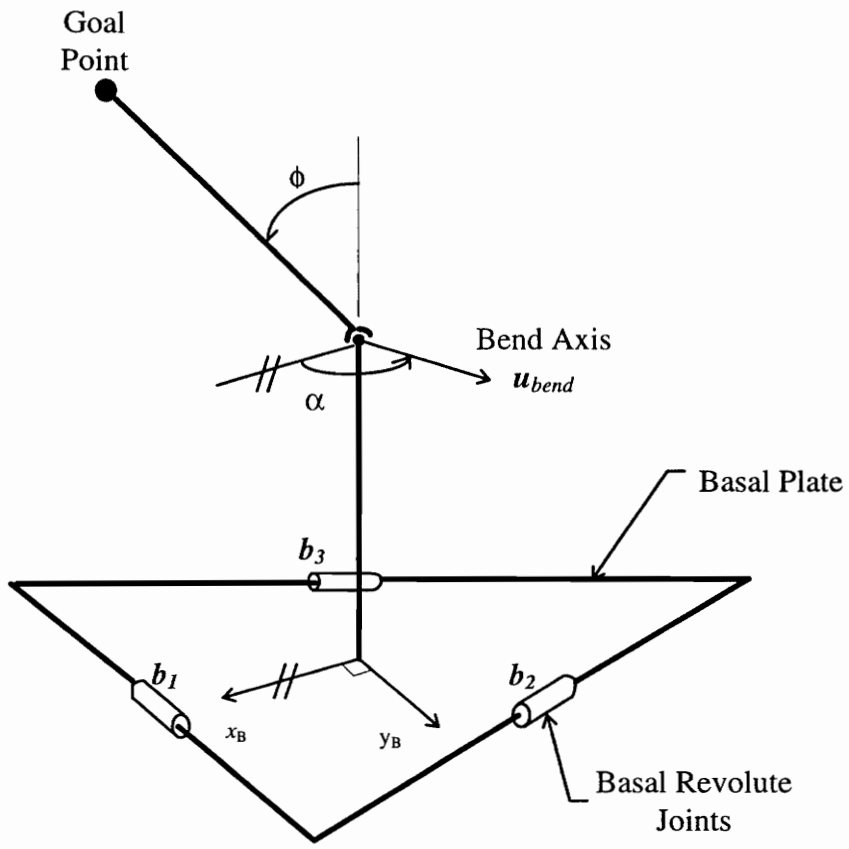


Fig. 4.5 - Definition of Bend Axis and Bend Angle

4.3.1 Residual Error in the Solution Technique

As discussed in Chapter 3, the error model differs from the ideal Carpal wrist model in that the solution cannot be carried out in closed-form. Here, the pose error is computed for the case in which there are no kinematic parameters errors in the error model in order to compare it with the ideal solution. Figure 4.6 shows that the magnitude of the error residual plotted against the bend angle, α, and bend axis angle, ϕ, over the workspace is not greater than $9.5(10^{-15})$, thus verifying the forward kinematics solution of the error model.

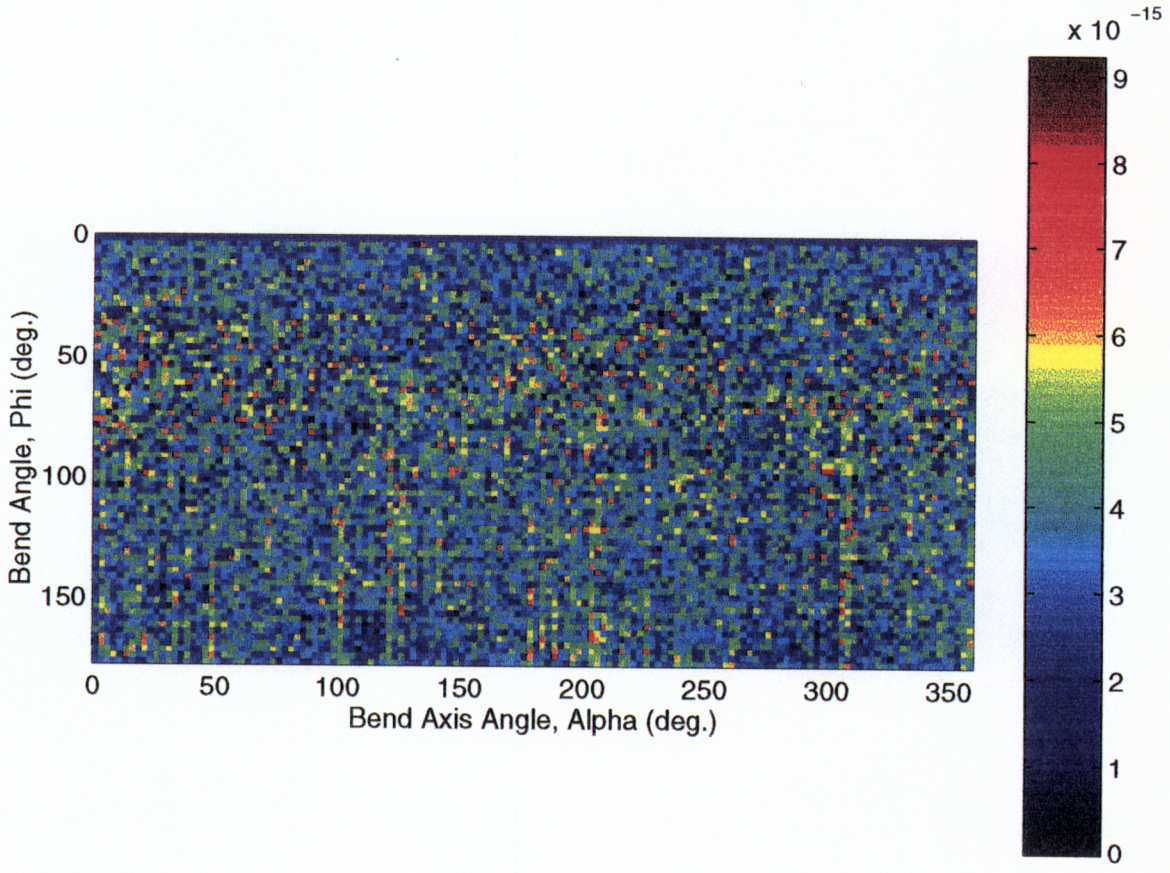


Fig 4.6 - Error Residual for Carpal Wrist with $R_d = 0.875$ and $R_b = 0.375$

4.4 Individual Kinematic Parameter Errors

Over a range of parameter errors taken as percentages of the nominal parameter values, pose error was found in order to assess the approximate linear behavior of the error model. The results were plotted in two ways. Figure 4.7 shows the plot of the pose error, averaged over the range of ϕ, versus α. It can be seen that the error model has nearly linear behavior for small variations in the error. The error plots at $\delta l_1 = 0.10$, 0.25 and

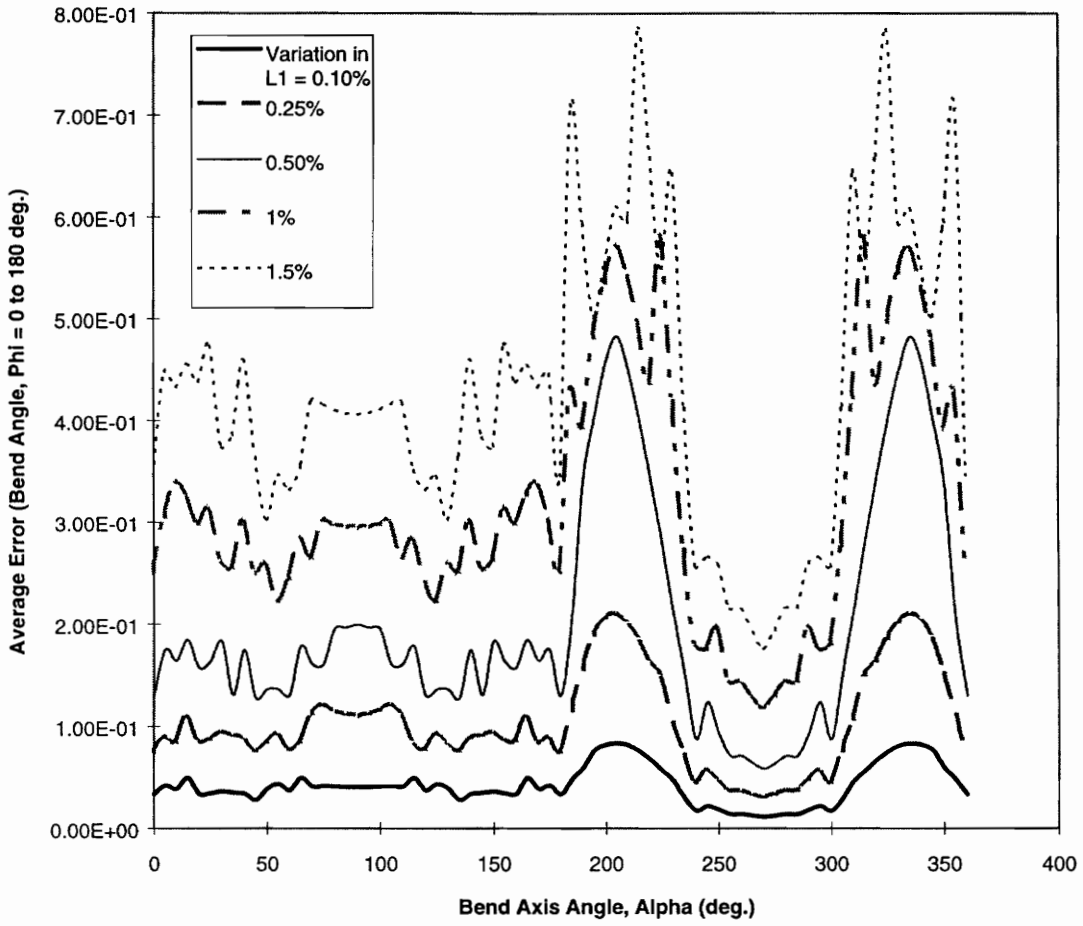


Fig. 4.7 - Plot of Averaged Error vs. Bend Axis Angle, α

Figure 4.8 shows the same parameter error percentages plotted in terms of average error, this time over the range of α, versus ϕ. Again, the nearly linear behavior is illustrated by the similar curves for the errors of δl_1 = 0.10, 0.25 and 0.50 %, while the behavior is lost at δl_1 = 1 and 1.5%. Note that ϕ = 0 to 90° corresponds to the 180° solid angle workspace.

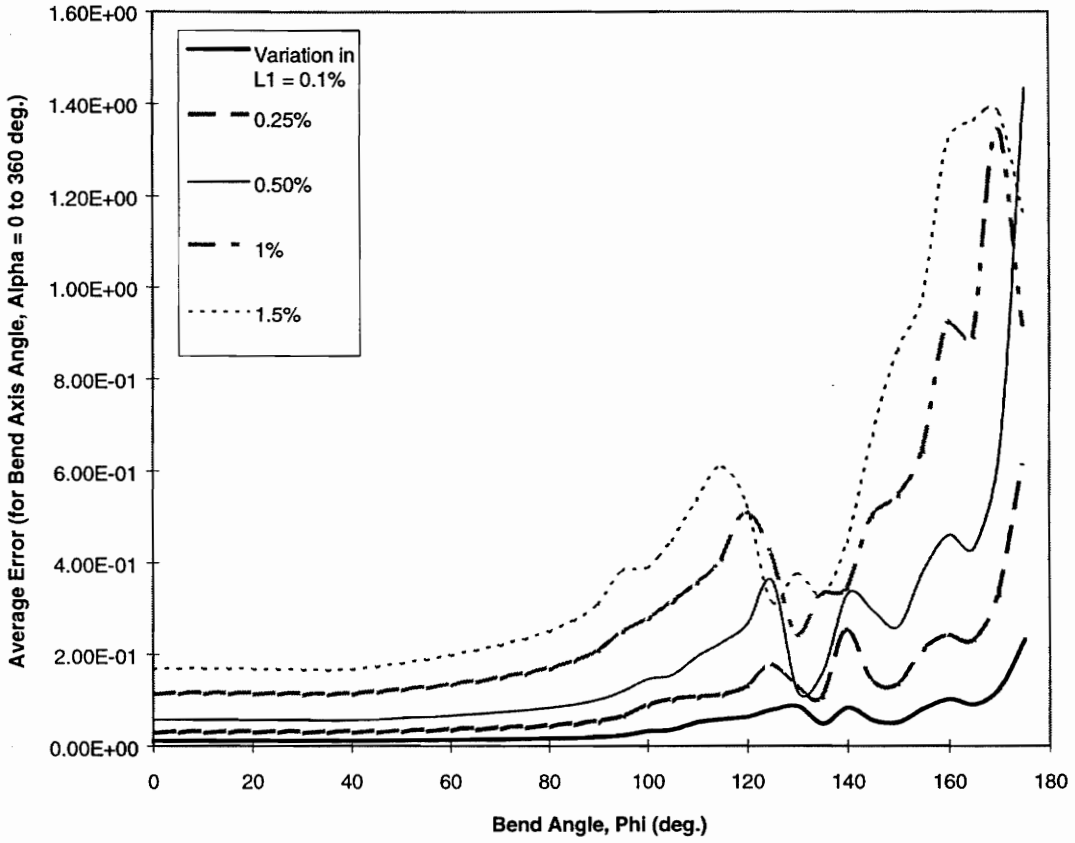


Fig. 4.8 - Plot of Averaged Error vs. Bend Angle, ϕ

In the following sections, results will be presented for variations equal to 0.5%, which allows the evaluation of a Carpal wrist which is slightly out-of-tolerance, yet well within the nearly linear range of the model.

4.4.1 Basal Leg Length

From Table 4.1 and the results of the prototype design by Ganino (1996), the error in the leg length parameter is expected to be about $\pm$ 0.033 cm ($\pm$ 0.013 IN.), or 0.43% of the

nominal value of 8 cm (3.15 in.). An error value of $\delta l_1 = 0.5\%$ was used in the error model, and the first basal leg dimension can be described as

$$l_1 = l + 0.005(l) = (1 + 0.005)(8\text{cm}) = 8.04\text{cm} \tag{4.3}$$

Plots of the resultant pose error over the wrist workspace are shown in Fig. 4.9. From the plot of Fig. 4.9a, it is worth noting that the intended-use 180° solid angle workspace, as defined by the region $0° \leq \phi \leq 90°$ and $0° \leq \alpha \leq 360°$, the pose error quantity has an average value of only 0.0648 cm (0.0255 in.). In the full workspace plot of Fig. 4.9b, the white regions represent workspace where the solution to the non-ideal forward kinematics does not exist, i.e., the mechanism cannot physically assemble. In theory, these points have the highest error since they cannot be attained. The highest values of error for which the Carpal wrist still assembles occur in the extreme lower regions of the workspace, specifically at $\alpha = 225°$ and 315° for $\phi = 177.5°$. If the wrist were bending over at an orientation of $\alpha = 270°$, i.e., directly between legs 2 and 3, this maximum error occurs 45° on either side of this bend. At such a bend angle, the wrist is virtually folded over onto itself. Fig. 4.10 is a top view of the *lower* half of the spherical workspace, which illustrates the unreachable workspace relative to the revolute joint locations.

Figures 4.11 and 4.12 show the workspace pose error due to a 0.5% error in the second and third basal legs, l_2 and l_3, respectively. The results are virtually identical to those due to error in the first leg, but they are shifted in the workspace by an amount $\alpha = 120°$. This corresponds to the leg placement interval about the basal center point. The values of maximum error, average error in the 180° solid angle workspace and percentage of workspace lost are summarized for all three cases in Table 4.2. The small differences in the numerical values are the result of numerical errors in the computation.

Table 4.2: Results of Varying Basal Leg Length, l_i

Leg w/ $\delta l = 0.005l$	Max. Error Value	Avg. Error in 180° Solid Angle Workspace	Approximate % Workspace Lost
l_1	9.02 cm (3.55 in.) at $\alpha = 225°, 315°$ and $\phi = 177.5°$	0.0648 cm (0.0255 in.)	6.55
l_2	9.02 cm (3.55 in.) at $\alpha = 75°, 345°$ and $\phi = 177.5°$	0.0650 cm (0.0256 in.)	6.52
l_3	9.02 cm (3.55 in.) at $\alpha = 105°, 195°$ and $\phi = 177.5°$	0.0647 cm (0.0255 in.)	6.55

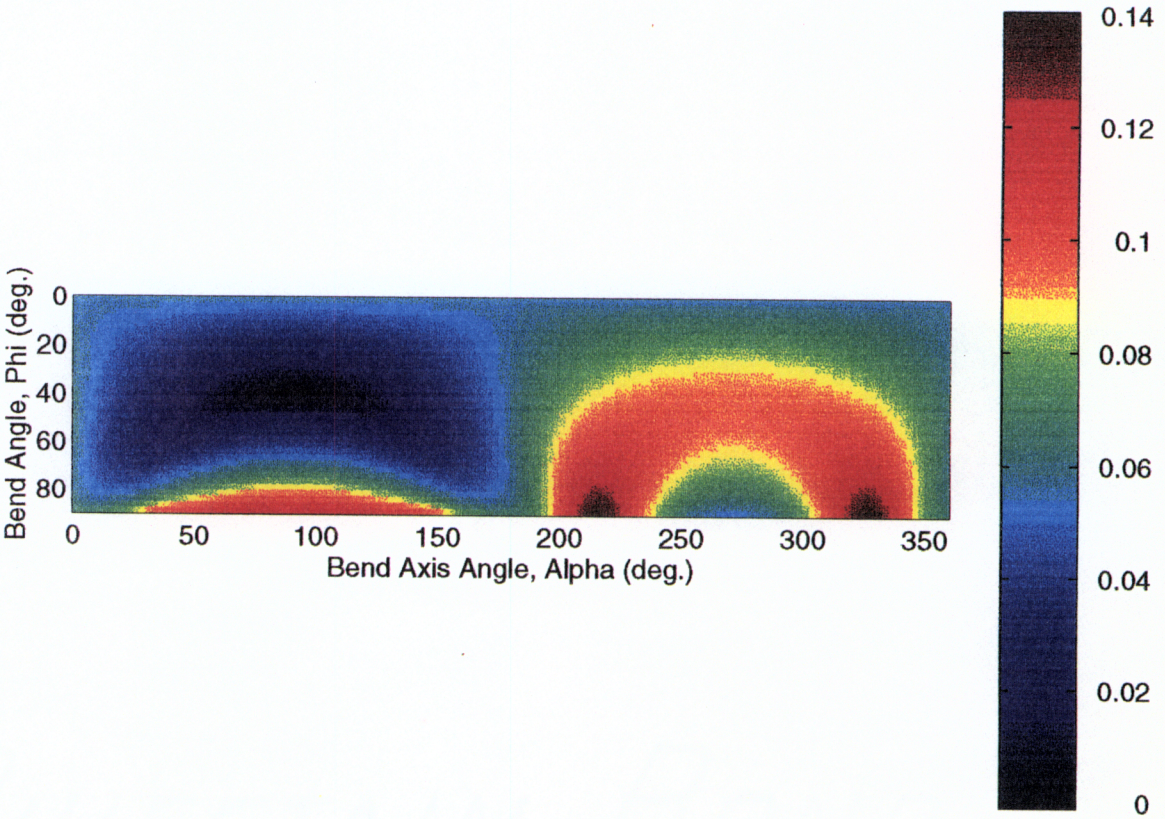


Fig. 4.9a - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta l_1 = 0.5\%$ for Upper 180° Solid Angle Workspace

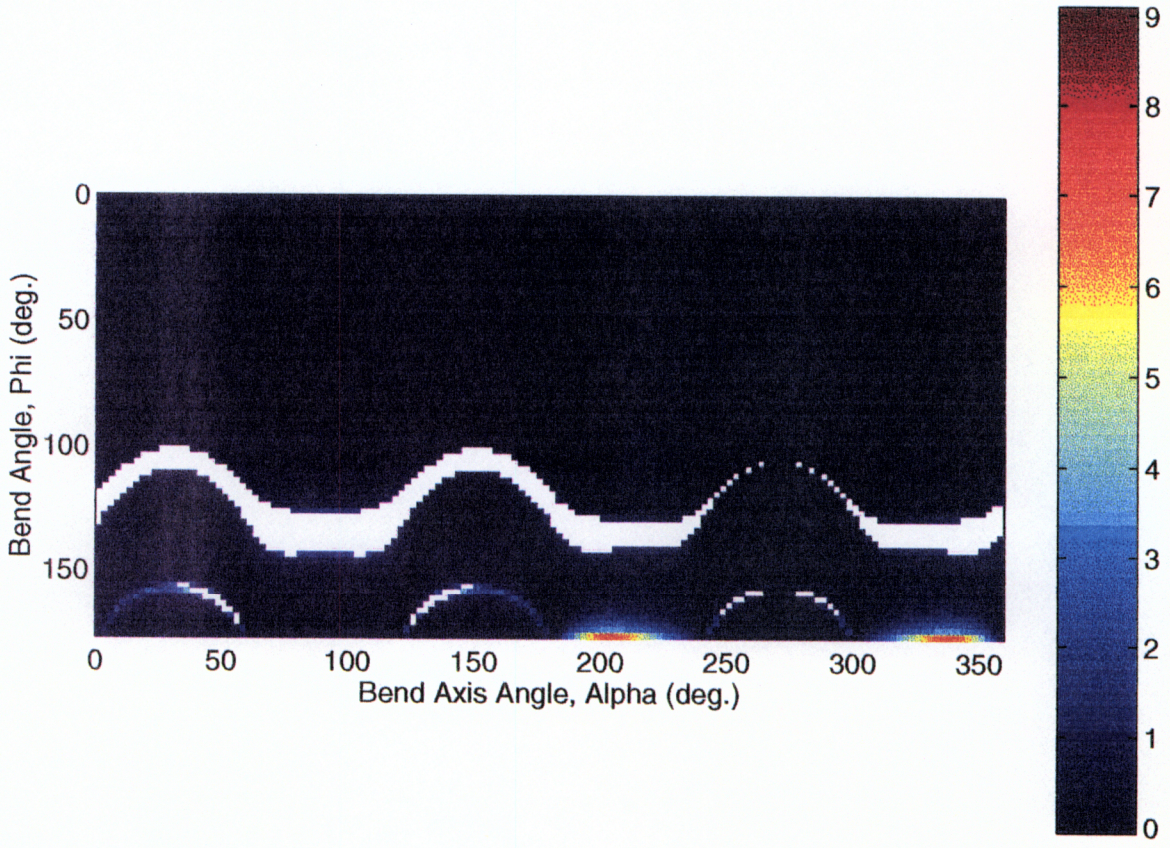


Fig. 4.9b - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta l_1 = 0.5\%$ for Entire 360° Solid Angle Workspace

(White regions represent unreachable workspace)

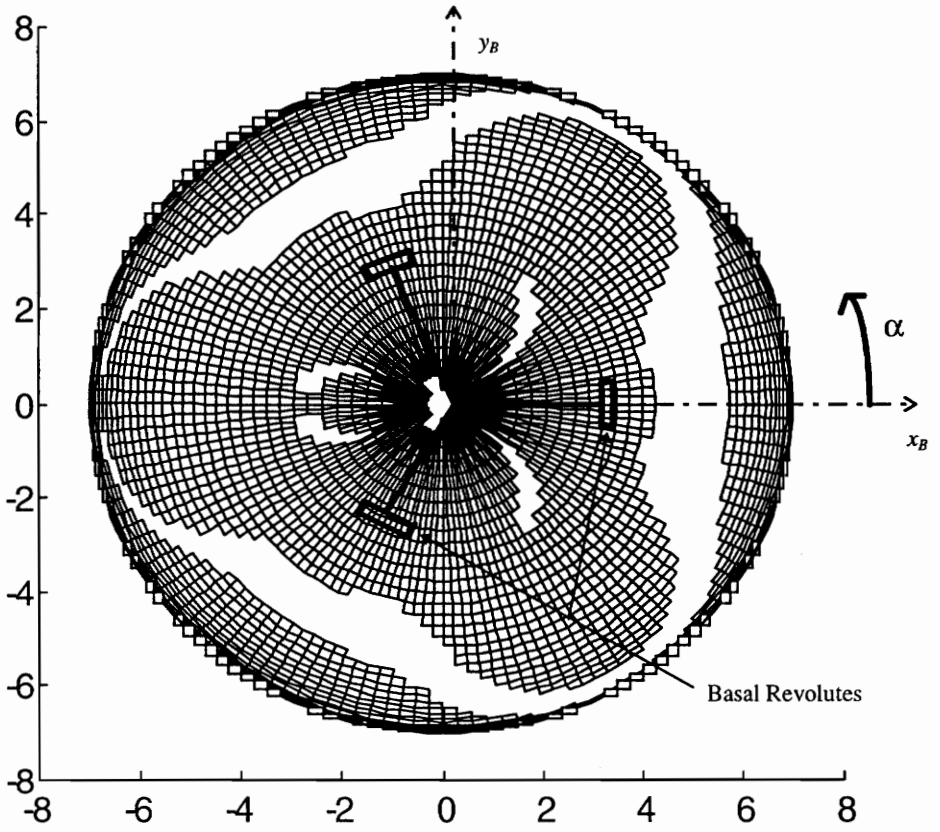


Fig. 4.10 - Lower Half of Workspace Sphere Showing Unreachable Points Relative to Basal Plate Geometry

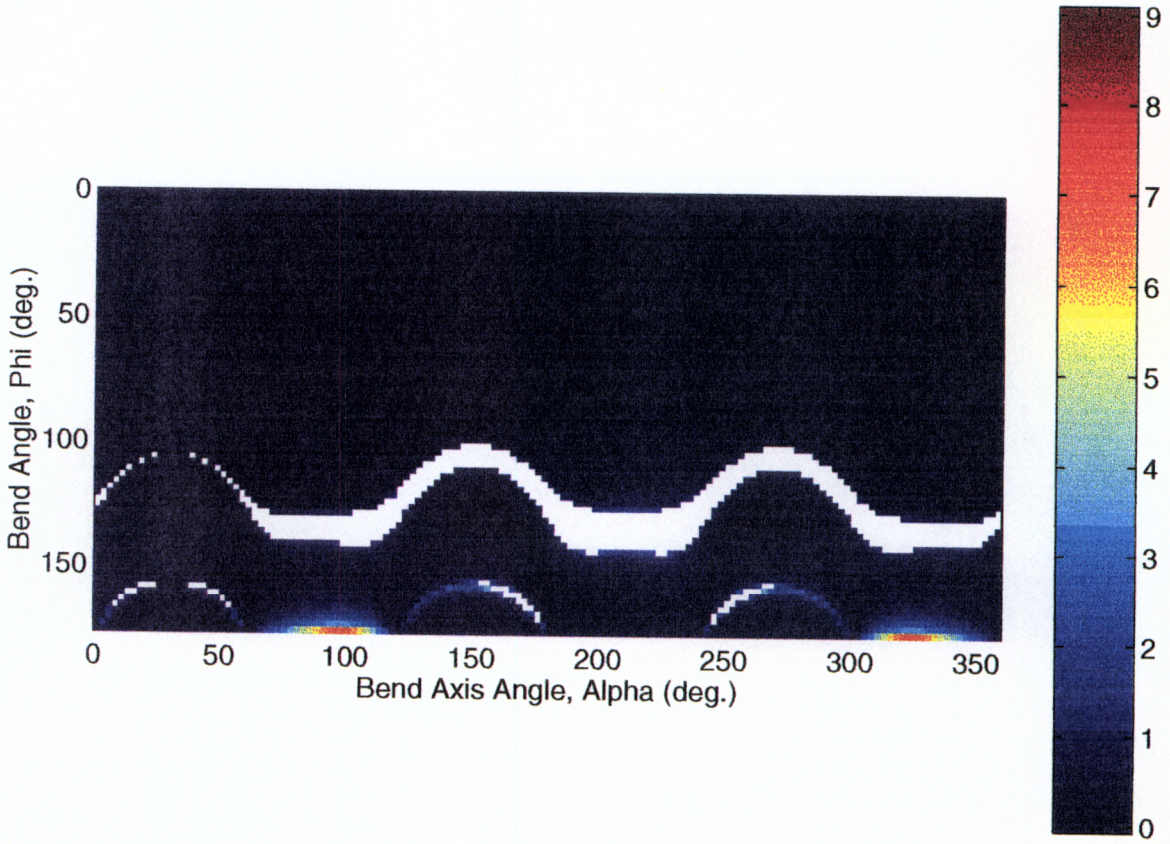


Fig. 4.11 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta l_2 = 0.5\%$

(White regions represent unreachable workspace)

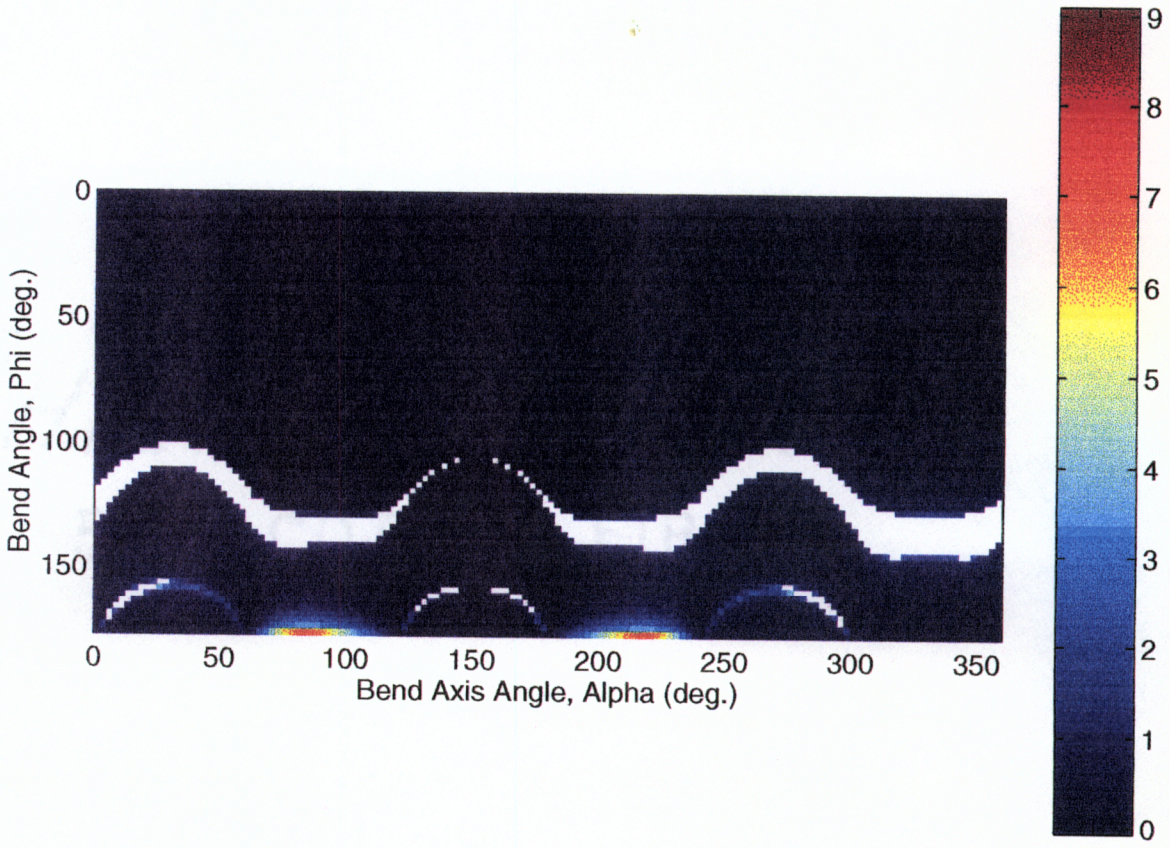


Fig. 4.12 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta l_3 = 0.5\%$

(White regions represent unreachable workspace)

4.4.2 Basal Plate Dimensions

Introduction of an 0.5% error in the basal plate dimensions, b_i, produces pose error similar to the errors resulting from variations in the leg lengths, but to a lesser magnitude. For a variation in b_1, such that

$$b_1 = b + 0.005b = (1 + 0.005) \cdot 3\text{cm} = 3.015\text{cm} \tag{4.4}$$

the resulting error plot over the workspace is shown in Fig. 4.13. The maximum pose error again occur as the wrist bends over between the second and third legs. Both the average error in the 180° solid angle space and the percentage of workspace lost are smaller than for errors in the basal leg lengths. As with variation in the leg length parameters, introducing error into the second and third basal plate dimensions produces identical errors which are shifted α = 120° in the workspace, as shown in Figs. 4.14 and 4.15. Table 4.3 summarizes the error results for the basal dimension variations.

Table 4.3: Results of Varying Basal Plate Dimension, b_i

Base w/ $\delta b = 0.005b$	Max. Error Value	Avg. Error in 180° Solid Angle Workspace	Approximate % Workspace Lost
b_1	2.27 cm (0.894 in.) α = 222.5°, 317.5° and ϕ = 177.5°	0.0236 cm (0.0093 in.)	1.86
b_2	2.27 cm (0.894 in.) α = 77.5°, 342.5° and ϕ = 177.5°	0.0237 cm (0.0093 in.)	1.84
b_3	2.27 cm (0.894 in.) α = 102.5°, 197.5° and ϕ=177.5°	0.0236 cm (0.0093 in.)	1.87

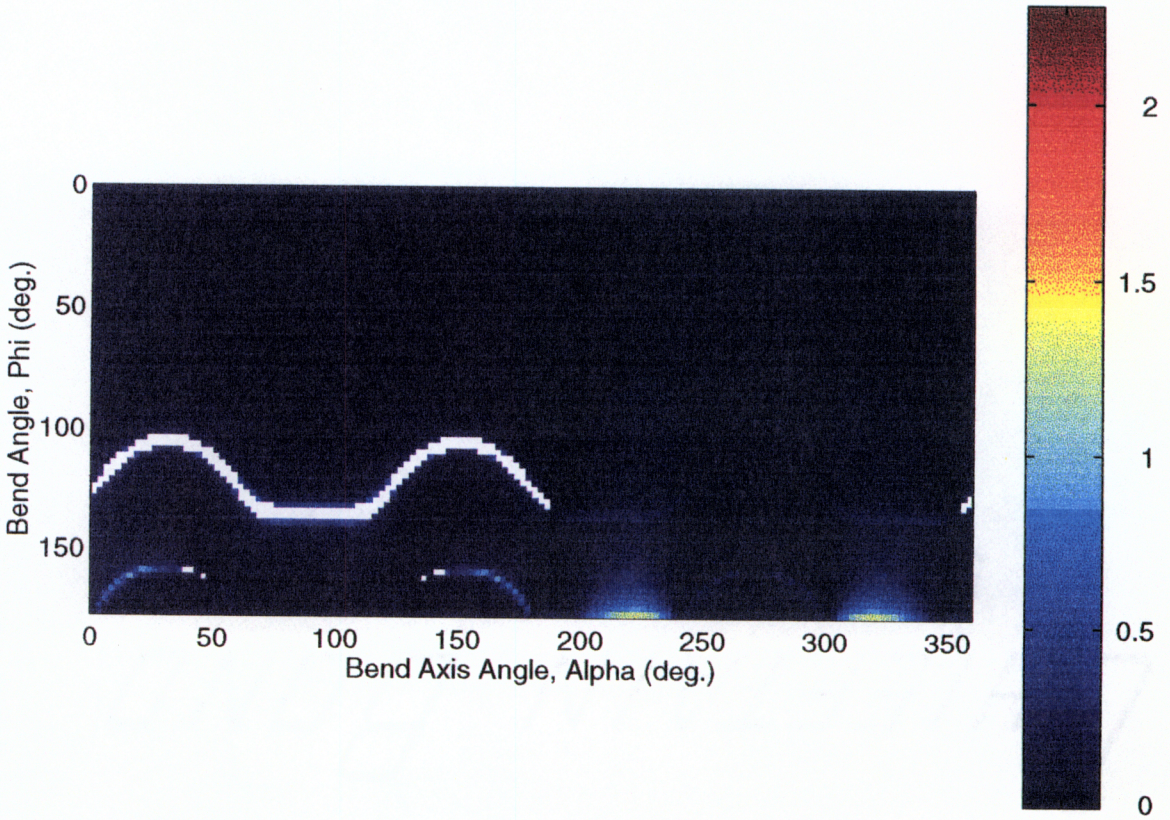


Fig. 4.13 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta b_1 = 0.5\%$

(White regions represent unreachable workspace)

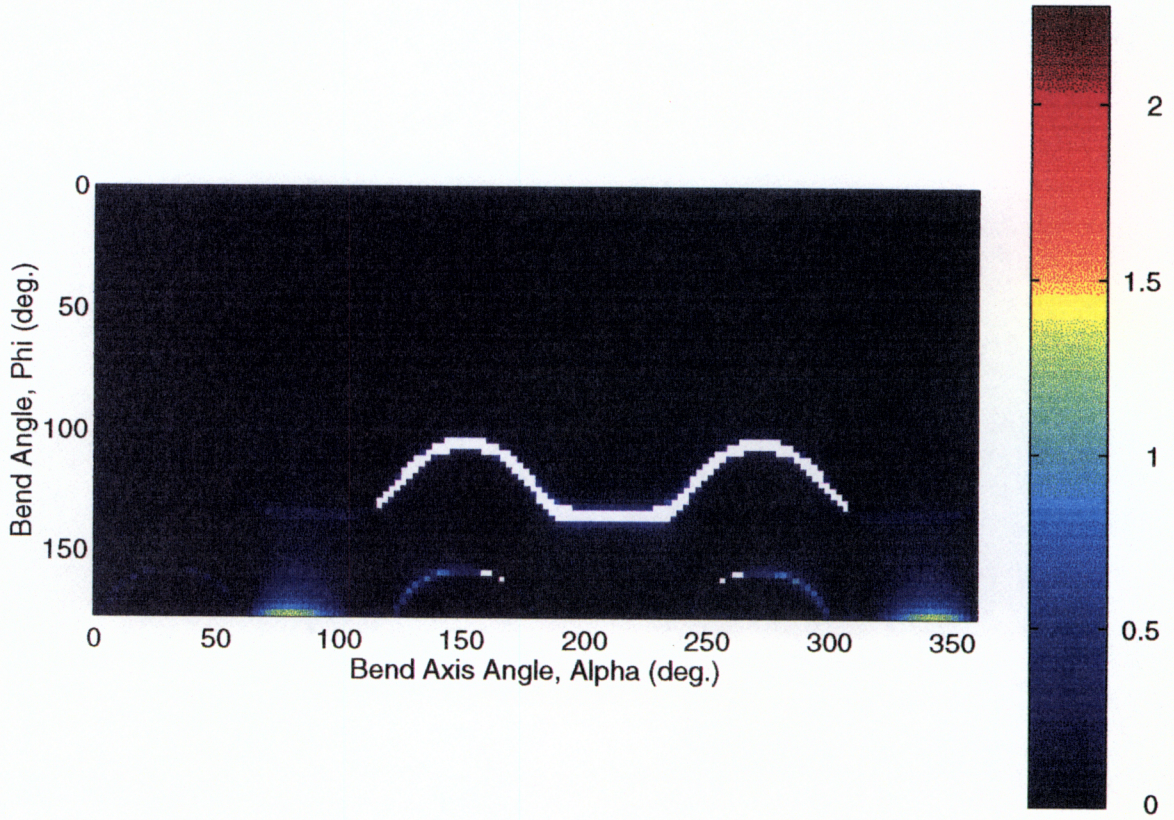


Fig. 4.14 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta b_2 = 0.5\%$

(White regions represent unreachable workspace)

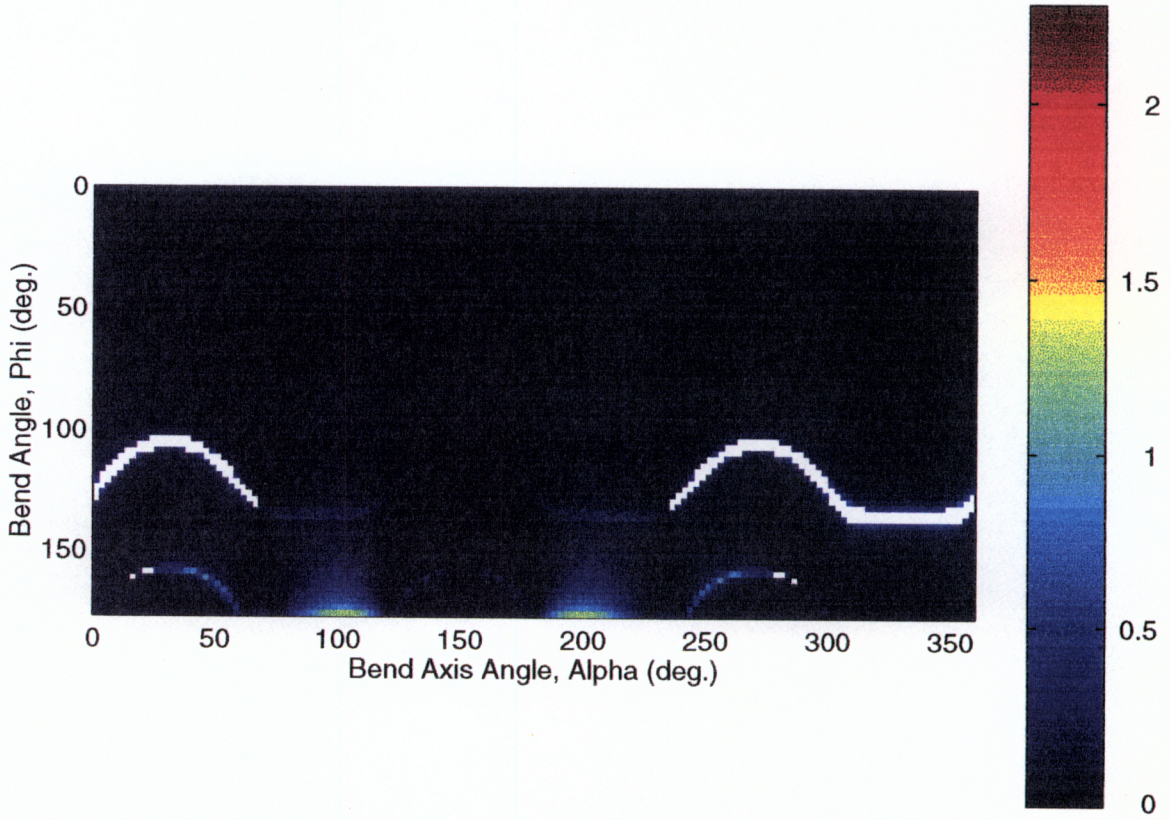


Fig. 4.15 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta b_3 = 0.5\%$

(White regions represent unreachable workspace)

4.4.3 Basal Revolute Locator Angles

In Chapter 3, the x_B-axis was defined to be perpendicular to the first revolute joint axis and in the plane containing the three revolute joints. The angle β_1 is defined from the x_B -axis and locating a line perpendicular to the second basal revolute axis. Likewise, the third basal angle, β_3, is also measured from the fixed x_B -axis, as shown in Fig. 4.16. Note that, in order to use the error model, the values of β_i must sum to 360°.

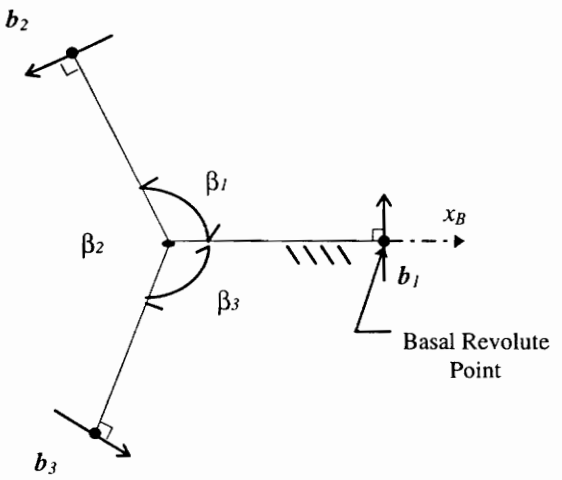


Fig. 4.16 - Relationship of Basal Revolute Locator Angles, β_i

Since this section deals only with variation of individual parameters, it can be seen from Fig. 4.16 that if angle β_3 remains equal to 120°, an increase in β_1 must also have a decrease in β_2, to maintain the 360° criteria. This pattern holds true for a change in any of the angles.

The results of introducing an error into the basal revolute locator angles, which effectively moves the basal revolute joint around the basal center point, are summarized in Table 4.4. The magnitude of the parameter error is again 0.5% and is based on the information of Table 4.1. Since the nominal value of β is always 120°, the error is

$$\delta\beta = 0.005(120°) = 0.6° \quad (4.5)$$

In Fig. 4.17 it can be seen that when β_1 is increased, a region of unreachable workspace develops exists when the wrist is bending in between basal revolutes $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. As the wrist bends between the second and third basal revolutes, $\boldsymbol{b}_2$ and $\boldsymbol{b}_3$, the goal points can be reached, but the error values are highest. In either case, the high error values are not within the upper 180° solid angle space, and less than 2% of the workspace is unattainable. Figure 4.18 illustrates a decrease in β_1. As expected, the numerical results are identical, except that they are reflected about a line at $\alpha = 210°$.

Figures 4.19 and 4.20 represent the error plots for an increase and decrease of β_3, respectively. The results are similar to those obtained by changing β_1, except that they are shifted in the workspace by an amount of $\alpha = 120°$.

Table 4.4: Results of Varying Basal Revolute Locator Angles, β_i

Angular Variation of $\delta\beta = 0.6°$	Max. Error Value	Avg. Error in 180° Solid Angle Workspace	Approximate % Workspace Lost
Increase β_1	1.09 cm (0.429 in.) $\alpha = 292.5°$ and $\phi = 115°$	0.0909 cm (0.0358 in.)	1.94
Decrease β_1	1.09 cm (0.429 in.) $\alpha = 127.5°$ and $\phi = 115°$	0.0909 cm (0.0358 in.)	1.94
Increase β_3	1.09 cm (0.429 in.) $\alpha = 247.5°$ and $\phi = 115°$	0.0909 cm (0.0358 in.)	1.97
Decrease β_3	1.09 cm (0.429 in.) $\alpha = 52.5°$ and $\phi = 115°$	0.0909 cm (0.0358 in.)	1.94

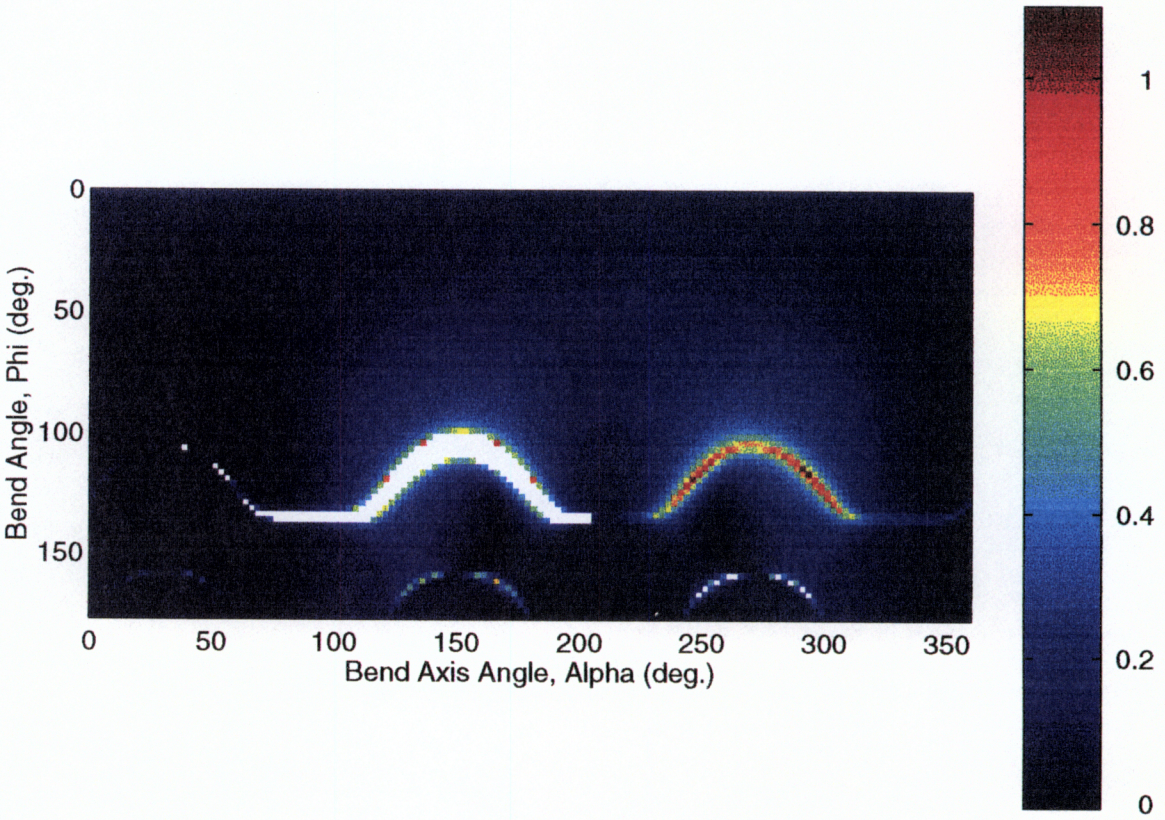


Fig. 4.17 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\beta_I = +0.6°$

(White regions represent unreachable workspace)

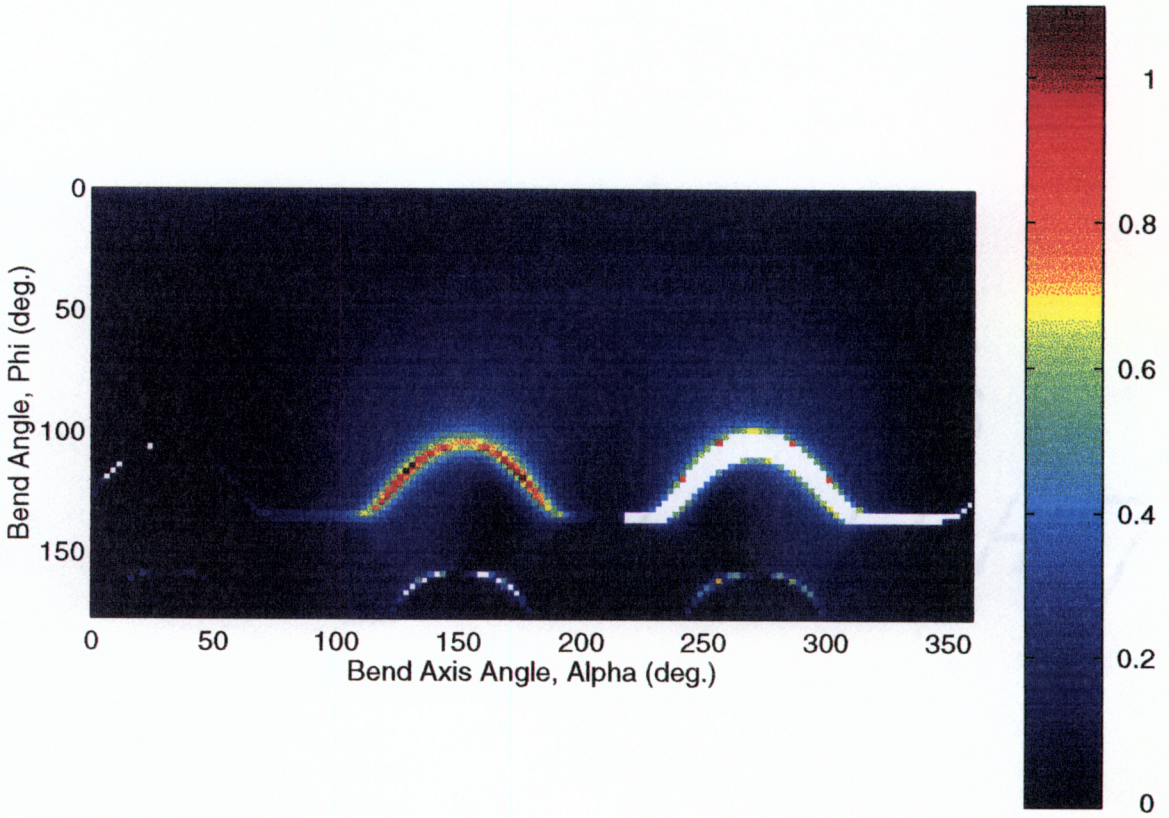


Fig. 4.18 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\beta_l$ = - 0.6°

(White regions represent unreachable workspace)

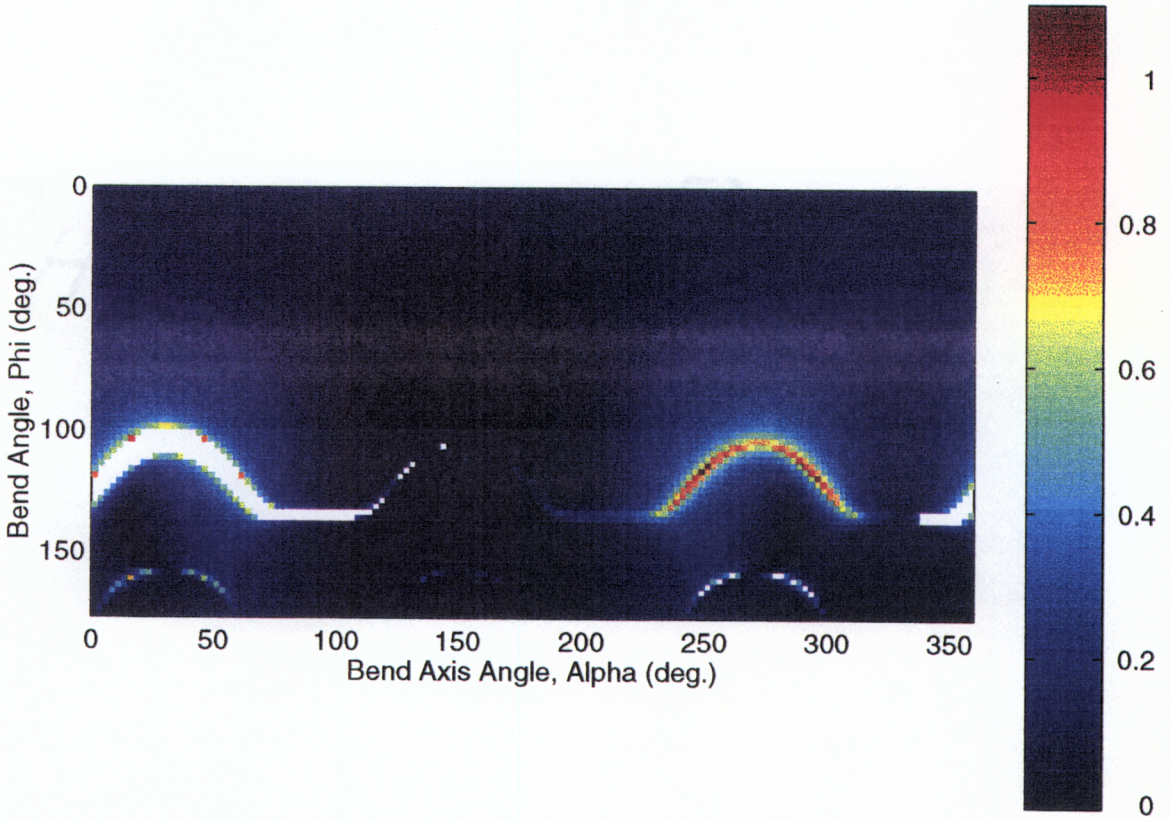


Fig. 4.19 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\beta_3 = +0.6°$

(White regions represent unreachable workspace)

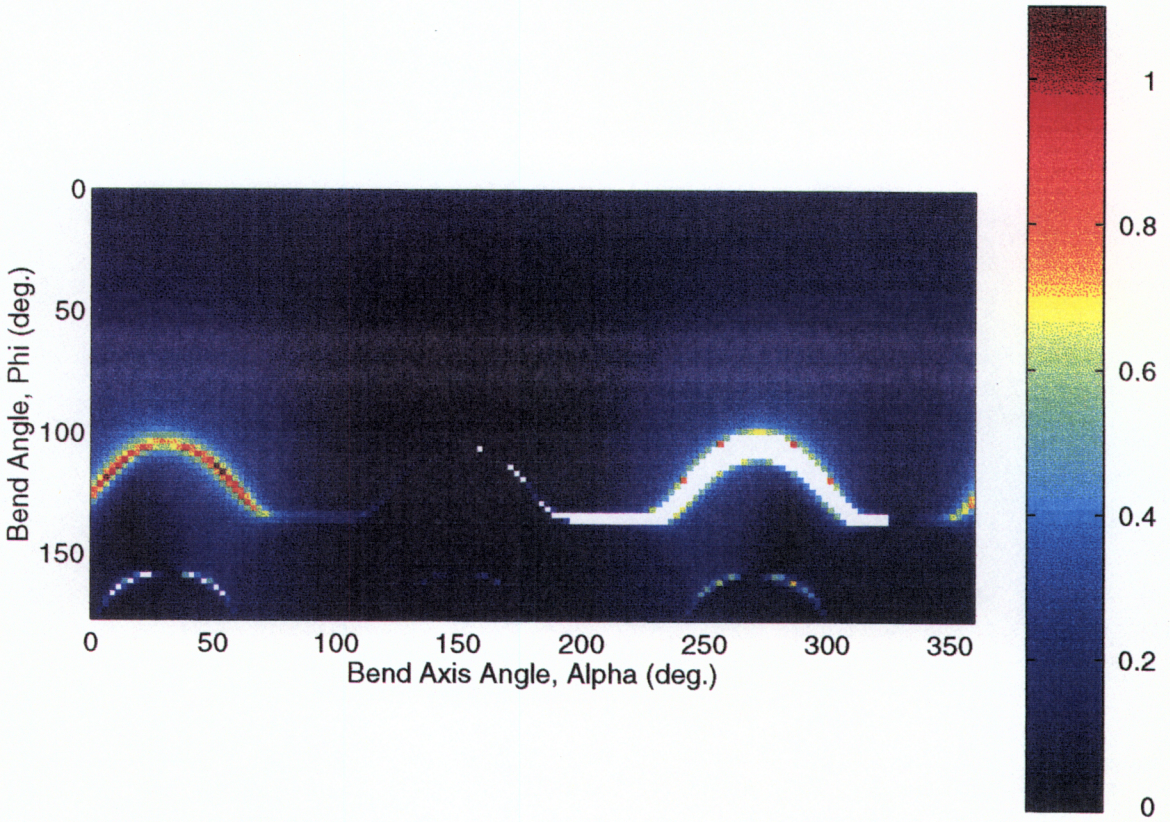


Fig. 4.20 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\beta_3 = -0.6°$

(White regions represent unreachable workspace.)

4.4.4 Basal Revolute Joint Out-of-Plane Offset

In Sections 4.4.1 and 4.4.2, it was shown that the results for an error in the second or third values for a parameter, i.e., l_2 and l_3 for parameters l_i, can be found by shifting the results for the first value, i.e., l_1, by an amount $\alpha = 120°$. In the interest of brevity, only the results of error in the first value will be presented for the remaining individual parameters. As illustrated in Fig. 3.2, the basal out-of-plane offset is a translation of the basal revolute point along the z_B-axis by an amount η_i. A variation of 0.5% of basal dimension, b, was used for the distance, η_i, that the basal revolute would be offset. Figure 4.21 shows the results of introducing such an error into the Carpal wrist. As expected, pose error is exhibited as the wrist bends along the first leg, with loss of workspace when bending in the direction of the second and third legs. The highest errors in reachable workspace are again when the wrist is folded over onto itself in the directions of the second and third legs. Table 4.5 summarizes the maximum and average over the 180° solid angle error values and the percentage of lost workspace.

4.4.5 Basal Revolute Out-of-Plane Tilt Angle

For the instance where a basal revolute is tilted out of plane, a variation of 0.5% produces the results in Table 4.5 and Fig. 4.22. Tilting the revolute about the (-x)-axis by a small amount produces interesting results in the workspace. The maximum error again occurs while bending into the second leg, with the wrist folded over onto itself. The regions of unreachable workspace for this case is unique. The asymmetry about $\alpha = 90°$ is especially interesting.

Table 4.5: Results of Varying Basal Revolute Offset, η_i, and Tilt Angle, μ_i

Variation	Max. Error Value	Avg. Error in 180° Solid Angle Workspace	Approximate % Workspace Lost
$\delta\eta_i = 0.5\%b$	5.16 cm (2.03. in.) $\alpha = 207.5°$ and 330°, $\phi = 177.5°$	0.0353 cm (0.0139 in.)	1.85
$\delta\mu_i = 0.6°$	12.98 cm (5.11 in.) $\alpha = 207.5°$ and $\phi = 177.5°$	0.0833 cm (0.0328 in.)	2.56

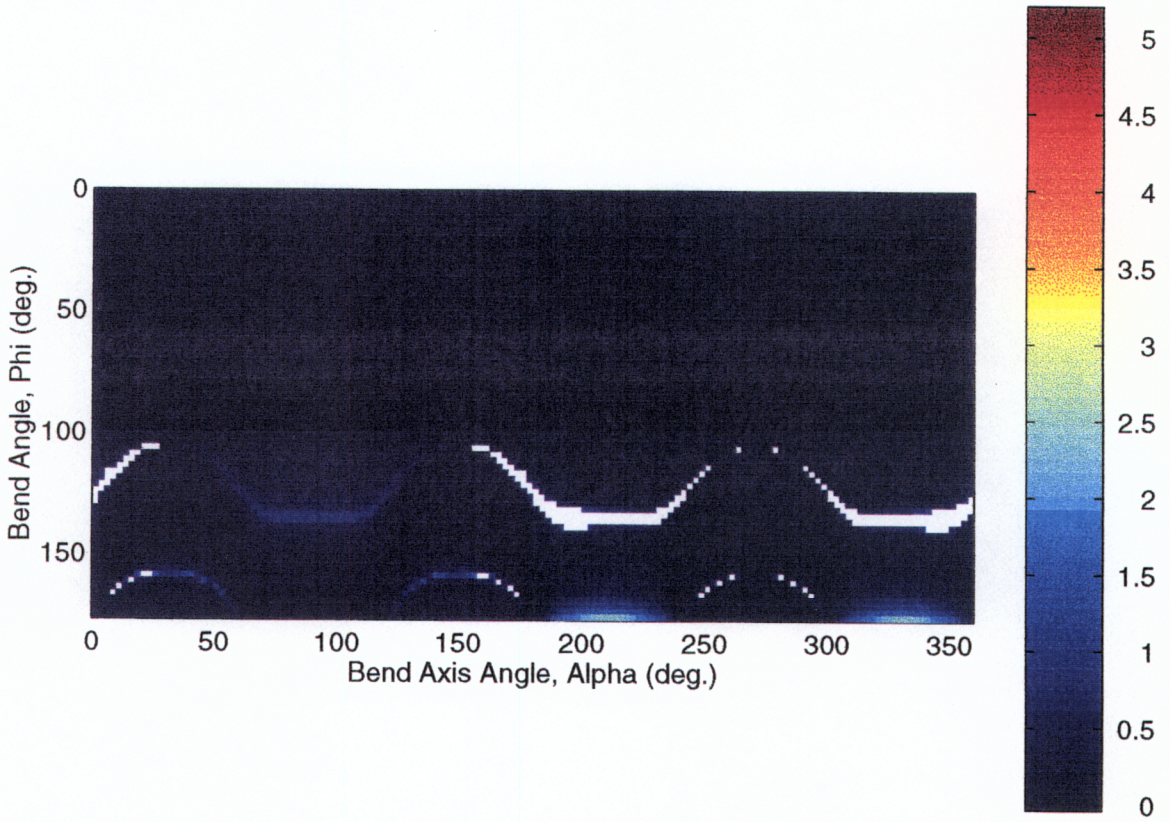


Fig. 4.21 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\eta_1 = 0.5\%$

(White regions represent unreachable workspace.)

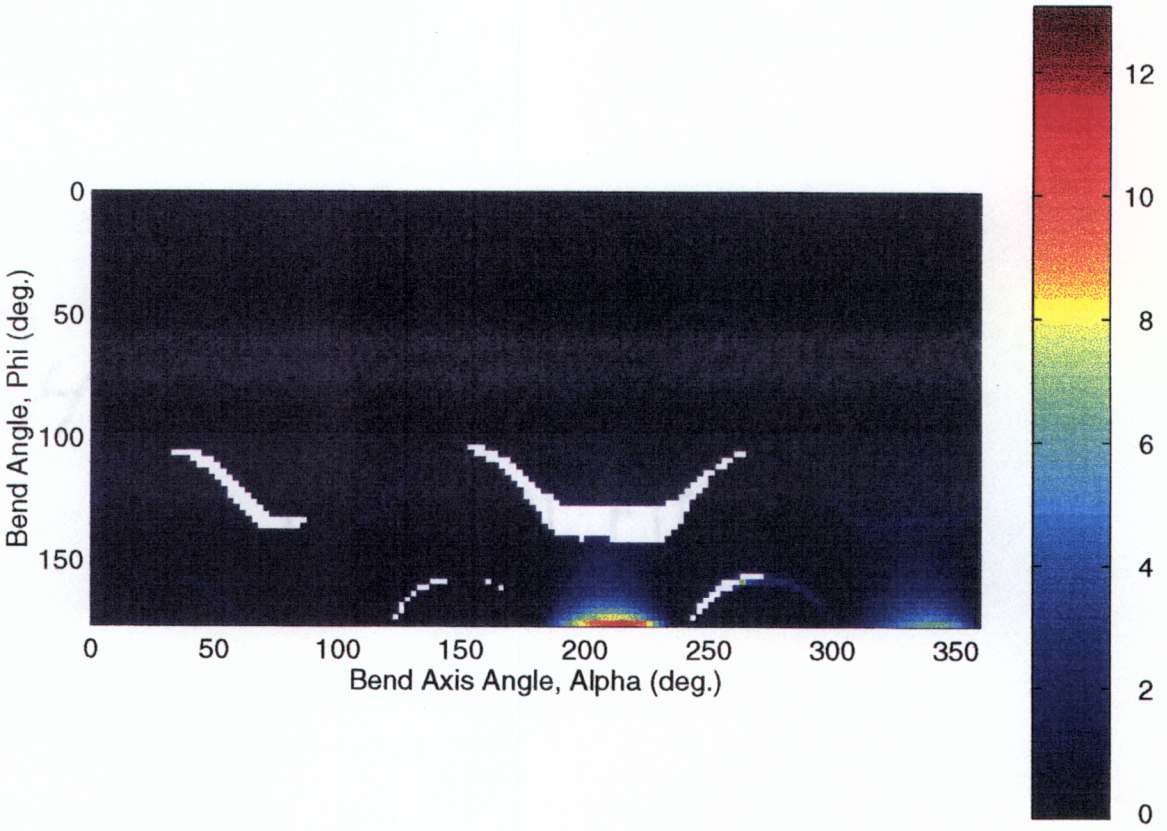


Fig. 4.22 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta\mu_1 = 0.6°$

(White regions represent unreachable workspace.)

4.4.6 Distal Legs

Introducing a 0.5% error into the first distal leg parameter, l_4, yields the workspace error plot shown in Fig. 4.23. Again, the 120° symmetry of the wrist will be used to predict the effect of errors in l_5 and l_6. An interesting result of error in the distal leg length is that much of the workspace remains intact, with only 0.54% being unreachable. This compares with about a 6.5% workspace loss for a similar percentage variation in the basal leg length. However, the average error over the upper 180° solid angle is nearly double that of the basal leg case, with a magnitude of 0.1162 cm (0.0457 in.).

4.4.7 Distal Revolute Connectors

The final kinematic error to consider is in the distal revolute point connectors, g_{di}. Ideally, the length of g_{di} is equal to $\sqrt{3}b$. From this dimension, and again considering a 0.5% variation in the parameter, the variation is found as

$$\delta g_{d_i} = 0.005(\sqrt{3})(3\text{cm}) = 0.026\text{cm} \tag{4.6}$$

The workspace plot shown in Fig. 4.24 illustrates the results for this case. Recall that the dimension g_{d1} connects the distal revolutes $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$. A gap in the workspace occurs when the wrist is bending in the direction of the *third* basal revolute, which geometrically opposes the increased dimension. Only 1.13% of the work region cannot be reached by the wrist. The highest errors occur when the wrist is folded over onto itself, bending in the direction of the first and second basal revolutes. Error results for the variation of distal legs and distal revolute connectors are summarized in Table 4.6.

Table 4.6: Results of Varying Distal Leg Lengths, l_{i+3}, and Revolute Connectors, g_{di}

Variation	Max. Error Value	Avg. Error in 180° Solid Angle Workspace	Approximate % Workspace Lost
$\delta l_{i+3} = 0.5\% l$	11.86 cm (4.67 in.) α = 195° and 345°, ϕ = 177.5°	0.1162 cm (0.0457 in.)	0.54
$\delta g_{di} = 0.5\%(\sqrt{3}b)$	6.69 cm (2.63 in.) α = 92.5° and 207.5°, ϕ = 177.5°	0.0423 cm (0.0167 in.)	1.13

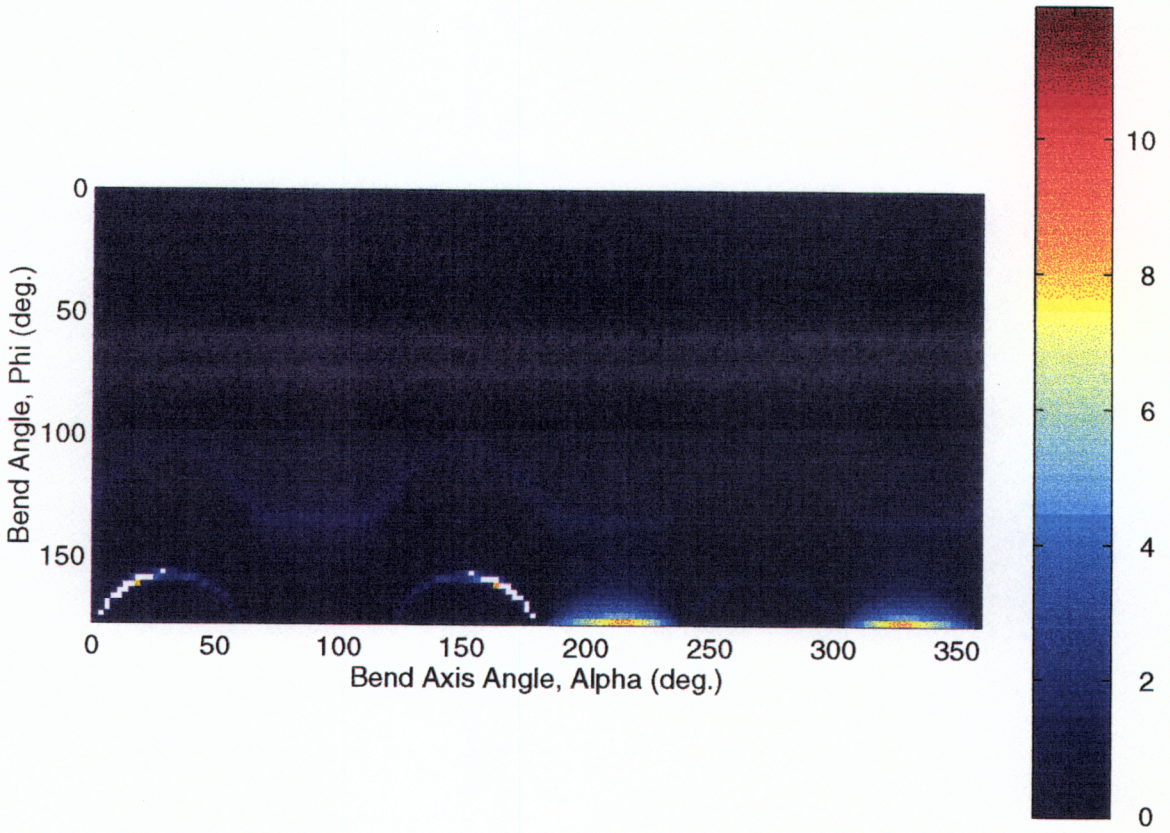


Fig. 4.23 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta l_4 = 0.5\%$

(White regions represent unreachable workspace)

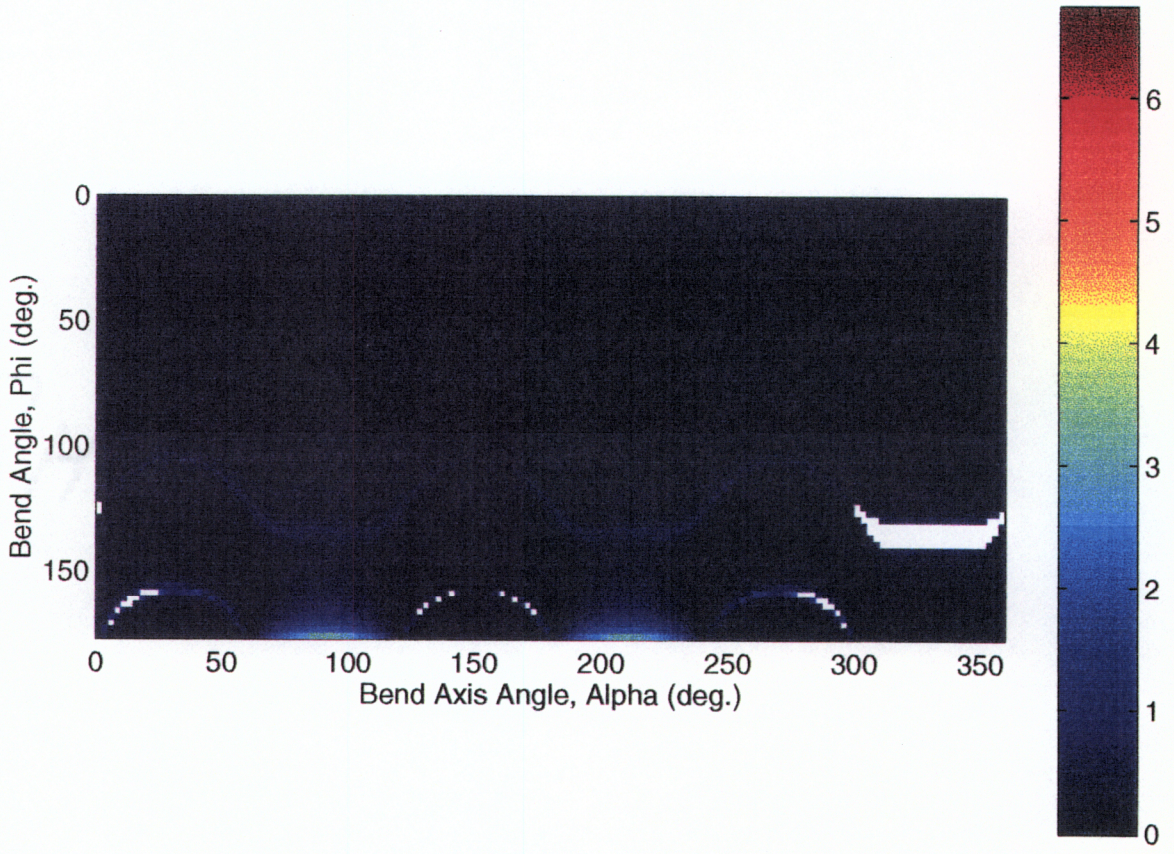


Fig. 4.24 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for $\delta g_{dl} = 0.5\%$

(White regions represent unreachable workspace.)

4.5 Combined Dimensional Errors

For a given Carpal wrist configuration, errors will exist simultaneously in all of the kinematic parameters. Early in Section 4.4, it was shown that the error model exhibits nearly linear behavior in the range of 0 to 1% parameter error. If the system is truly linear, the principle of superposition can be used to predict the effects of combinations of errors in the wrist. The expression of pose error, restated below in Eq. (4.7), utilizes the sum of the vector norms of the error at each distal revolute point.

$$E = \sum_{i=1}^{3} \left\| \boldsymbol{d}_i^{actual} - \boldsymbol{d}_i^{ideal} \right\| \tag{4.7}$$

For multiple error sources, the scalar quantity of Eq. (4.7) cannot simply be summed to find the resultant error. Instead, consider the error vectors for the individual distal revolute joint locations, expressed as

$$\boldsymbol{e}_{d_i} = \boldsymbol{d}_i^{actual} - \boldsymbol{d}_i^{ideal} \tag{4.9}$$

Superposition of these error vectors for variations in n parameters can be expressed as

$$\left(\boldsymbol{e}_{d_i}\right)_{overall} = \left(\boldsymbol{e}_{d_i}\right)_1 + \left(\boldsymbol{e}_{d_i}\right)_2 + \ldots + \left(\boldsymbol{e}_{d_i}\right)_n \tag{4.10}$$

An example of superposition is shown in Fig. 4.25. In the graph, the dashed line corresponds to the vector sum of the errors due to individual variations in basal leg lengths l_1 and l_2; the program was run for error in each length, then the vector errors were summed. The solid line represents the error for the combined variations, that is, the program was run with errors in both parameters. Ideally, the curves in Fig. 4.25 would be identical. For the error model, the curves are nearly identical in the range of ϕ = 0 to

120°, which allows for the use of superposition over the majority of the workspace. As the wrist bends beyond $\phi = 120°$, the curves become dissimilar.

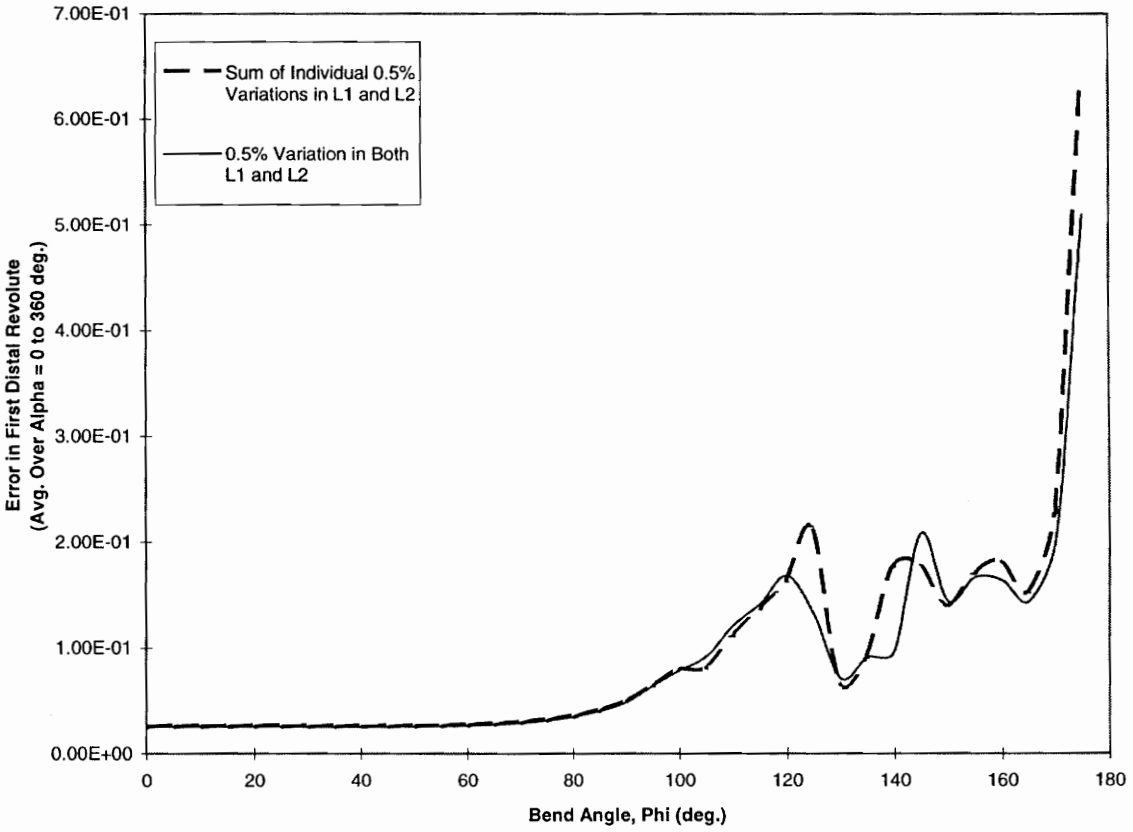


Fig. 4.25 - Plot of Averaged Error vs. Bend Angle, ϕ, for Combined Parameter Errors

Use of superposition in the Carpal wrist error model allows for broad application of this work to future wrist design. It is impossible to consider every combination of errors for all possible configurations of the Carpal wrist. In the sections that follow, sample error combinations are presented. It is understood that, for a wrist with multiple error sources, a statistical error analysis would yield more definitive results. Here, the combinations of

parameter errors were used to further validate the error model as a tool. In future wrist designs, the error model program could be run for the specific Carpal wrist configuration at hand.

4.5.1 Two Basal Legs

The data for combined variations in two basal legs, in this case l_1 and l_2, is shown in the error plot of Fig. 4.26. The resulting workspace suffers from a complete loss of the lower region of the sphere. Although the lower points are reachable, the wrist could never move to those points from the upper region without being physically disassembled. The amount of workspace lost based only on the white regions shown in the plot is 9.28%, but in reality the loss is 36.2% due to the aforementioned physical restriction. This additional loss of workspace is certainly a concern. The maximum and averaged errors and workspace lost are summarized in Table 4.7.

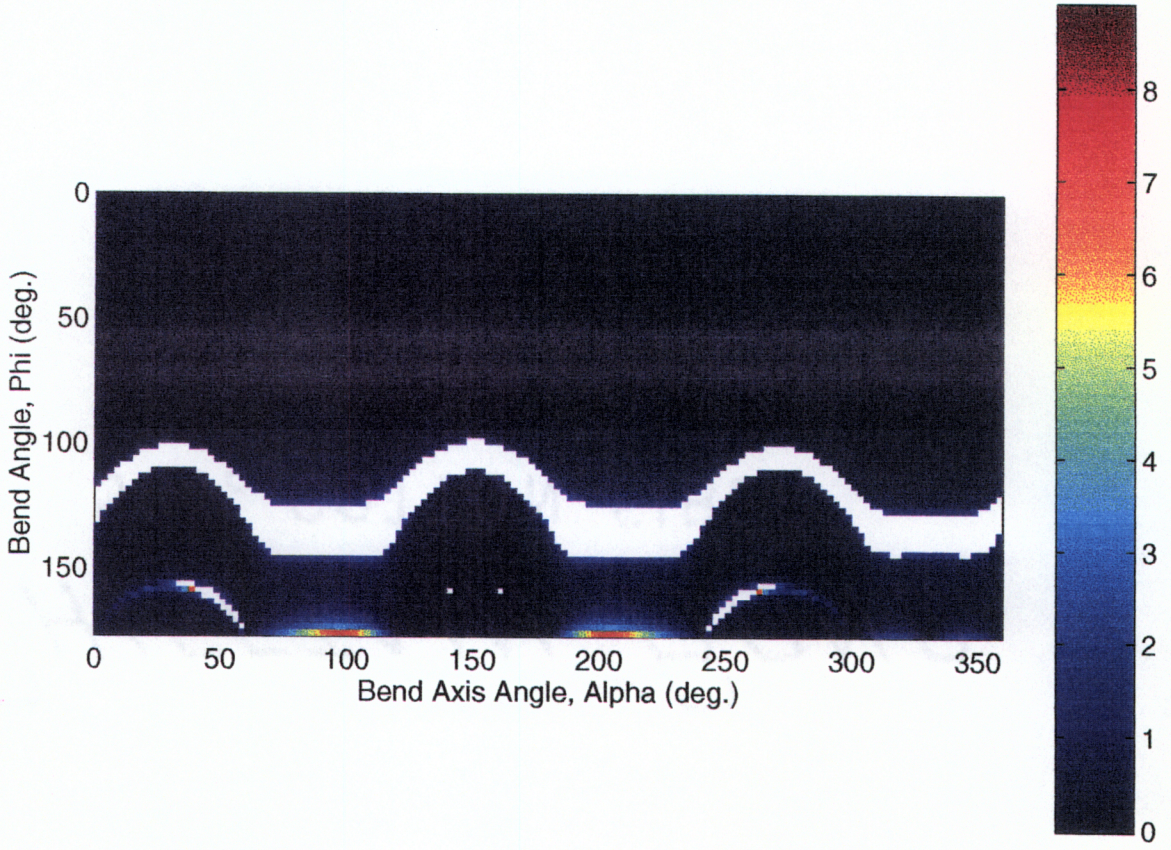


Fig. 4.26 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for Combined Variations δl_1 and $\delta l_2 = 0.5\%$

(White regions represent unreachable workspace.)

4.5.2 Upper and Lower Legs - Connected

In this case, errors of 0.5% are introduced into both a basal leg, l_1, and a distal leg, l_4, which are joined by a common spheric joint. An interesting result is that there is not a loss of reachable workspace. The combination of extended legs cancels out the losses due to the individual variations, as shown by Fig. 4.27 The values are summarized in Table 4.7.

4.5.3 Upper and Lower Legs - Not Connected

As a companion to the previous section, the instance in which the two extended legs are *not* connected by a common spheric joint yields the plot in Fig. 4.28. With the combined errors, less workspace is lost compared to variation in l_1 alone (see Fig. 4.9). However, as the wrist bends over onto itself, three regions of high error are experienced, as opposed to two for the single parameter variation. The error values are summarized in Table 4.7.

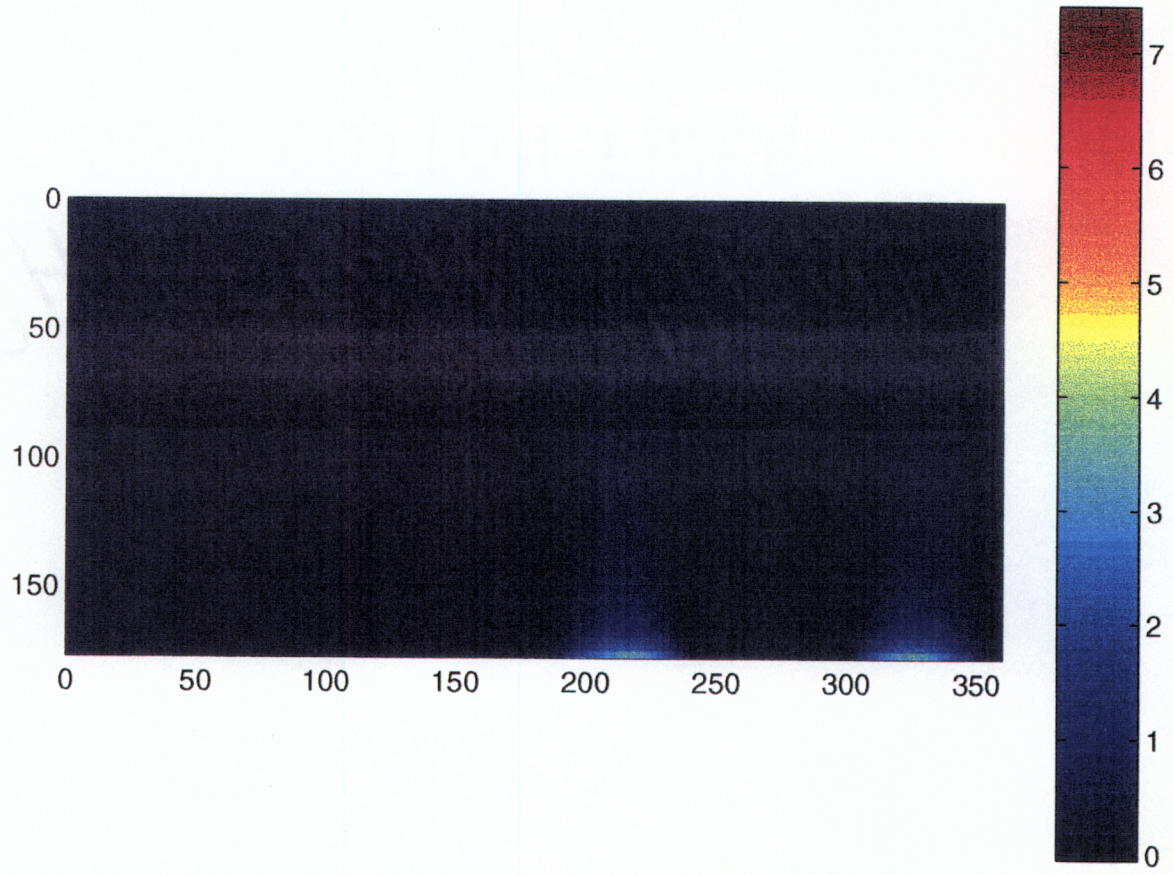


Fig. 4.27 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for Combined Variations δl_1 and $\delta l_4 = 0.5\%$

(White regions represent unreachable workspace.)

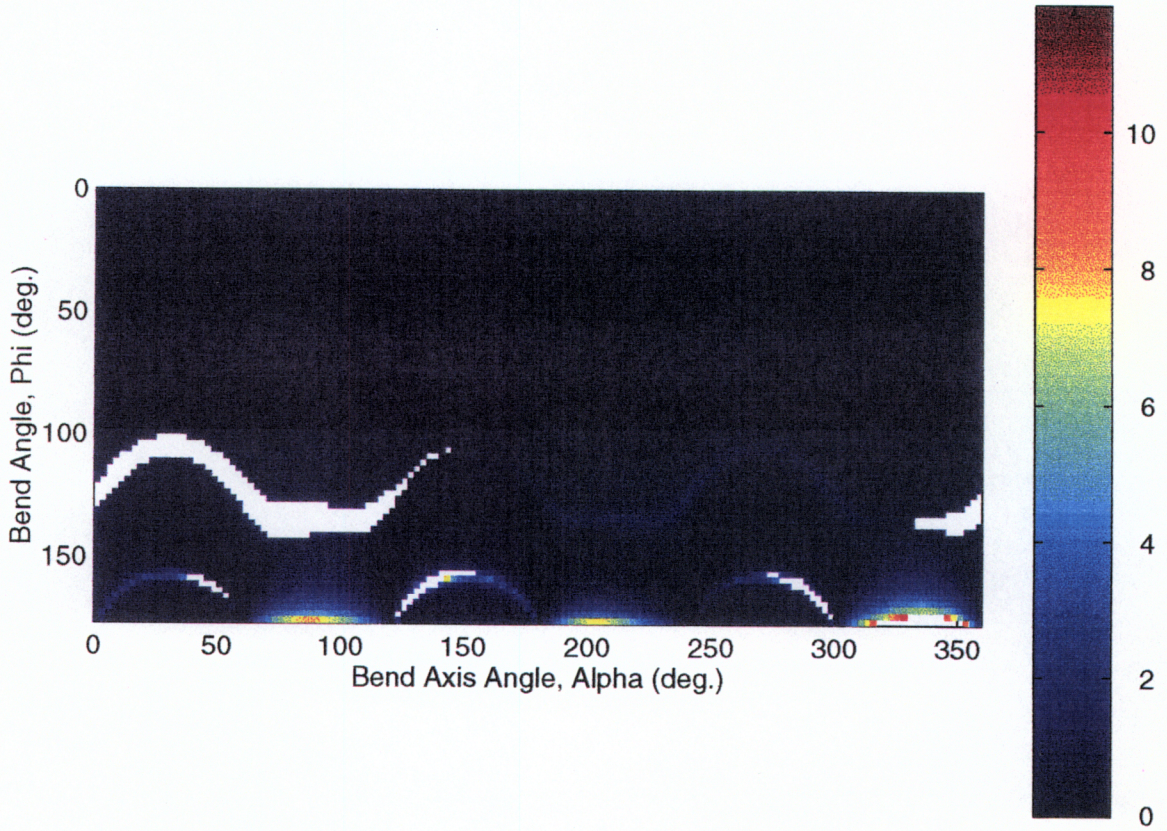


Fig. 4.28 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for Combined Variations δl_1 and $\delta l_5 = 0.5\%$

(White regions represent unreachable workspace.)

4.5.4 Basal Revolute Locators Increased

In Section 4.4.3, individual variations in β_i were examined, and it was noted that regions of highest error, with the mechanism still assembled, did not occur at the extreme values of bend angle, ϕ. Here, an increase in both β_1 and β_3 produces similar effects, as shown in Fig. 4.29. The region of highest error in the reachable workspace is within $225° \leq \alpha \leq 325°$ and $100° \leq \phi \leq 140°$, as the wrist is bending between the second and third leg pairs. Another interesting result is that some moderately high error values are experienced within the upper 180° solid angle workspace.

Table 4.7: Results of Combined Errors

Variation	Max. Error Value	Avg. Error in 180° Solid Angle Workspace	Approximate % Workspace Lost
$\delta l_1 = 0.5\% l$ and $\delta l_2 = 0.5\% l$	8.84 cm (3.48 in.) $\alpha = 75°$ and 225°, $\phi = 177.5°$	0.0802 cm (0.0316 in.)	36.24
$\delta l_1 = 0.5\% l$ and $\delta l_4 = 0.5\% l$	7.35 cm (2.89 in.) $\alpha = 215°$ and 325°, $\phi = 177.5°$	0.1572 cm (0.0619 in.)	0
$\delta l_1 = 0.5\% l$ and $\delta l_5 = 0.5\% l$	11.80 cm (4.65 in.) $\alpha = 105°$ $\phi = 177.5°$	0.1154 cm (0.0454 in.)	3.75
$\delta\beta_1$ and $\delta\beta_3 =$ +0.6°	1.50 cm (0.59 in.) $\alpha = 295°$ $\phi = 117.5°$	0.1425 cm (0.0561 in.)	3.35

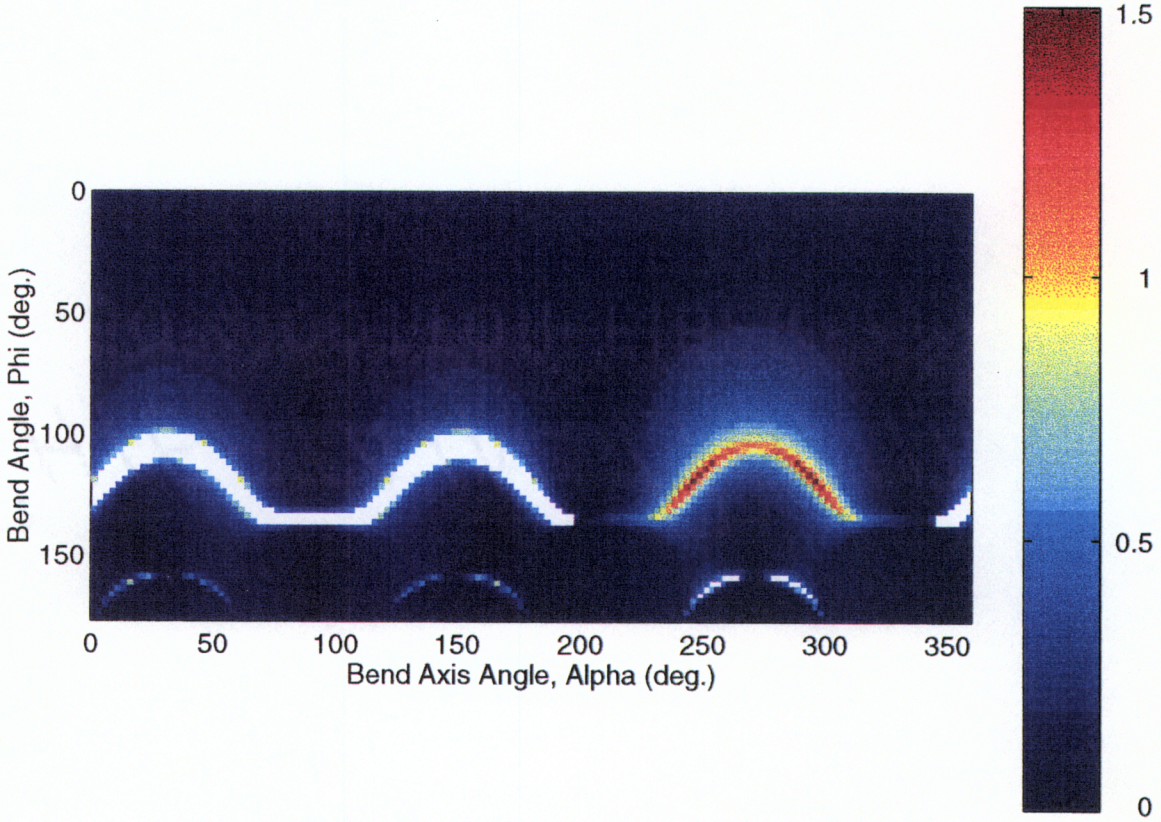


Fig. 4.29 - Error vs. Bend Angle, ϕ, and Bend Axis Angle, α, for Increasing $\delta\beta_1$ and $\delta\beta_3$ by +0.6°

(White regions represent unreachable workspace.)

Chapter 5

Conclusions and Recommendations

A tool has been presented to predict the effects of kinematic parameter errors on the workspace pose of the Carpal wrist. Such errors would occur as a result of manufacturing tolerances and assembly problems. In the ideal Carpal wrist, the forward and inverse kinematic problems can be solved in closed-form, based on assumptions of symmetry. These assumptions are not valid for the error model forward kinematic solution. Instead, a system of equations describing the position of the output plane relative to the mid-plane of the wrist must be solved iteratively.

The error model was shown to exhibit nearly linear behavior for errors in the range of 0 to 1% of the nominal parameter value. It was tested for errors of 0.5% on a Carpal wrist having a base-to-leg ratio of $R_b = 0.375$ and a plunge ratio of $R_d = 0.875$. Such a wrist configuration has a theoretical workspace of a full sphere, allowing for pose error evaluation over a maximal region. These ratios were also chosen for the proof-of-concept

prototype designed by Ganino (1996), which utilizes only the upper 180° solid angle workspace.

For each parameter error, the output pose error was summarized in three ways: maximum error over the entire workspace, average error over the upper 180° solid angle workspace, and as a percentage of workspace lost. The highest and lowest values of each category for errors in the individual parameters are shown in Table 5.1. As shown in the plots in Chapter 4, the maximum error over the entire workspace was generally encountered only when the wrist was folded over onto itself, which would not be possible with a physical device. This was true for all of the parameters except the basal revolute locator angles, β_i, where the maximum error values occur just beyond the workspace equator. For combinations of parameter errors, it was shown that the principle of superposition can be applied over the range of errors for which the model is nearly linear (0 to 1% of the nominal parameter value).

Table 5.1: Summary of Pose Error for Individual Parameter Errors

Quantity		Parameter	Value
% Workspace Lost	high	Basal Leg, l_i	6.5%
	low	Distal Leg, l_{i+3}	0.5%
Average Over 180° Solid Angle	high	Distal Leg, l_{i+3}	0.116 cm
	low	Basal Plate Dim., b_i	0.024 cm
Maximum Over Workspace	high	Basal Revolute Tilt, μ_i	12.98 cm
	low	Basal Revolute Locator, β_i	1.09 cm

The focus of this work has been to develop an error model for the Carpal wrist. Further development of the model is possible with the inclusion of additional parameters which exist in the physical structure of the wrist. For example, the ideal Carpal wrist has a fixed base and movable distal plate, connected by three revolute-spheric-revolute (RSR) serial

kinematic chains. The Carpal wrist prototype designed by Ganino (1996) utilizes equivalent revolute-revolute-revolute (3R) joints in place of the spheric joints. The 3R joint has intersecting joint axes in order to achieve the spherical motion. Errors in this component could be incorporated into the error model as offsets which affect the axes intersection point.

Other possible error sources in the physical wrist arise from clearance within the assembled components and compliance in the links and joints. Clearance, or backlash, could exist in the drive components or in the RSR chains. Since the wrist is designed to carry significant loads, link and joint compliance for both static and dynamic situations could be considered. For the combinations of parameter errors, further work could involve an optimization problem to determine which combinations of errors, and at which percentage of the nominal parameter value, result in the worst pose error. A statistical error analysis would provide confidence intervals for dimensional tolerances. One reason for the creation of an error model for a manipulator is calibration. As described by Voruganti (1995), Notash and Podhorodeski (1995) and others, the error model is used to incorporate not only geometric errors, but dynamic and non-geometric errors, such as the previously mentioned compliance and backlash. The system is then updated based on the error-corrected kinematic parameters to account for the position and orientation errors at the output. Calibration could be performed on the wrist itself, but more likely it would be used in conjunction with a manipulator calibration scheme.

The story of the Carpal wrist is just beginning. This work will contribute a tool to the continually expanding designer's toolbox. Following the recent fabrication of the proof-of-concept prototype and a U.S. Patent application, these design tools will be important as the device is developed for industrial applications.

References

Acton, F.S., 1970, *Numerical Methods That Work*, Harper & Row, New York, NY.

Alefeld, G., Gienger, A., and Potra, F., 1994, "Efficient Numerical Validation of Solutions of Nonlinear Systems," *SIAM Journal of Numerical Analysis*, Vol. 31, No. 1, pp. 252-260.

Arun, V., 1990, "The Solution of Variable-Geometry Truss Problems Using New Homotopy Continuation Methods," *Ph.D. Dissertation*, Department of Mechanical Engineering, Virginia Polytechnic Institute & State University, Blacksburg, VA.

Burns, S.A., 1994, "The Monomial Method: Extensions, Variations and Performance Issues," *International Journal for Numerical Methods in Engineering*, Vol. 37, pp. 2093-2107.

Canfield, S. L. and Reinholtz, C. F., 1995, "Development of an All-Revolute-Jointed Constant-Velocity Coupling," *Journal of Applied Mechanisms and Robotics,* Vol. 2, No. 3, pp. 13-19.

Canfield, S.L., Ganino, A.J., Salerno, R.J., and Reinholtz, C.F., 1996, "Singularity and Dexterity Analysis of the Carpal Wrist," Accepted for Publication in *Mechanisms Synthesis and Analysis, Proceedings of the ASME 24th Biennial Mechanisms Conference*, Vol. and pp. not yet disclosed.

Canfield, S. L, Salerno, R. J. and Reinholtz, C. F., 1994, "Kinematic Design of a Large-Angular-Motion Constant-Velocity Coupling," *Mechanism Synthesis and Analysis, Proceedings of the ASME 23rd Biennial Mechanisms Conference*, DE-Vol. 70, pp. 167-174.

Clavel, R., 1988, "Delta, A Fast Robot with Parallel Geometry," *Proceedings of the 18th International Symposium on Industrial Robots*, pp. 91-100.

Cleghorn, W.L., Fenton, R.G. and Fu, J.F., 1993, "Optimum Tolerancing of Planar Mechanisms Based on an Error Sensitivity Analysis," *Journal of Mechanical Design*, Vol. 115, No. 2, pp. 306-313.

Clemens, M., 1872, "Universal-Joint Coupling for Connecting Shafts," *U. S. Patents, 125,880 & 125,881.*

Coetzee, F.M. and Stonick, V.L., 1995, "Sequential Homotopy-Based Computation of Multiple Solutions to Nonlinear Equations," *Proceedings - ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 1356-1359.

Craig, J.J., 1989, *Introduction to Robotics: Mechanics and Control*, Addison Wesley Publishing Co., Reading, MA.

Fujimoto, K., Tadokoro, S. and Takamori, T., 1990, "Development of a 6-D.O.F. Direct-Drive Wrist Joint by Pneumatic Linear Actuator Drive," *Proceedings of the 16th Annual Conference of IEEE Industrial Electronics Society - IECON '90: Signal Processing and System Control of Factory Automation*, Vol. 1, pp. 427-432.

Ganino, A.J., 1996, "Mechanical Design of the Carpal Wrist: A Parallel-Actuated, Singularity-Free Robotic Wrist," *M.S. Thesis*, Department of Mechanical Engineering, Virginia Polytechnic Institute & State University, Blacksburg, VA.

Gerlach, J., 1994, "Accelerated Convergence of Newton's Method," *SIAM Review*, Vol. 36, No. 2, pp. 272-276.

Haug, E.J., 1989, *Computer Aided Kinematics and Dynamics of Mechanical Systems, Volume 1: Basic Methods*, Allyn and Bacon, Needham Heights,MA.

Holzbock, W.G., 1986, *Robotic Technology: Principles and Practice*, Van Nostrand Reinhold Co., Inc., New York, NY.

Hostetter, G.H., Santina, M.S., D'Caprio-Montalvo, P., 1991, *Analytical, Numerical, and Computational Methods for Science and Engineering*, Prentice-Hall, Inc., Englewood Cliffs, NJ.

Hudgens, J.C. and Tesar, D., 1988, "A Fully-Parallel Six Degree-of-Freedom Micromanipulator: Kinematic Analysis and Dynamic Model," *Trends and Developments in Mechanisms, Machines and Robotics, Proceedings of the ASME 20th Biennial Mechanisms Conference*, DE-Vol.15-3, pp. 29-37.

Innocenti, C. and Parenti-Castelli, V., 1993, "Echelon Form Solution of Direct Kinematics for the General Fully-Parallel Spherical Wrist," *Mechanism and Machine Theory*, Vol. 28, No. 4, pp. 553-561.

Kim, W.K. and Tesar, D., 1990, "Study on the Structural Design of Force-Reflecting Manual Controllers," *Flexible Mechanism, Dynamics and Robot Trajectories, Proceedings of the ASME 21st Biennial Mechanisms Conference*, DE-Vol. 24, pp. 481-488.

Lee, J.J. and Chang, S.-L., 1992, "On the Kinematics of the UPS Wrist for Real Time Control," *Robotics, Spatial Mechanisms and Mechanical Systems, Proceedings of the ASME 22nd Biennial Mechanisms Conference*, DE-Vol. 45, pp. 305-312.

Mabie, H.H. and Reinholtz, C.F., 1987, *Mechanisms and Dynamics of Machinery, 4th ed.*, John Wiley & Sons, New York, NY.

Merlet, J.P., 1994, "Parallel Manipulators: State of the Art and Perspectives," *Advanced Robotics*, Vol. 8, No. 6, pp. 589-596.

Nguyen, C.C., Antrazi, S.C., Zhou, Z.-L. and Campbell, C.E., 1991, "Analysis and Implementation of a 6 DOF Stewart Platform-based Robotic Wrist," *Computers and Electrical Engineering*, Vol. 17, No. 3, pp. 191-203.

Nguyen, T.T., 1993, "Neural Network Architecture for Solving Nonlinear Equation Systems," *Electronics Letters*, Vol. 29, No. 16, pp. 1403-1406.

Notash, L. and Podhorodeski, R.P., 1995, "Kinematic Calibration of Parallel Manipulators," *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4, pp. 3310-3315.

Padmanabhan, B., Tidwell, P.H., Salerno, R.J. and Reinholtz, C.F., 1992, "VGT-Based Gimbals: Practical Construction and General Theory," *Flexible Mechanism, Dynamics and Analysis, Proceedings of the ASME 22nd Biennial Mechanisms Conference*, DE-Vol. 47, pp. 437-443.

Salerno, R. J., Canfield, S. L., Ganino, A. J. and Reinholtz, C. F., 1995, "A Parallel, Four Degree-of-Freedom Robotic Wrist," *Advances in Design Automation, Proceedings of the 1995 ASME Design Technical Conferences*, DE-Vol. 82, pp. 765-771.

Saraç, H.I., Güven, H.R., Sözbir, N., and Uysal, Ü., 1994, "A New Algorithm for Solution of the Non-linear Equation Systems Occurring in the Design of Air-Compression Systems," *Energy*, Vol. 19, No. 10, pp. 1067-1075.

Stewart, D., 1965, "A Platform with Six Degrees of Freedom," *Proceedings of the Institute of Mechanical Engineering*, Vol. 180, No. 1, pp. 371-386.

Strang, G., 1986, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA.

Tidwell, P.H., 1989, "Design and Construction of an Octahedral-Octahedral Variable Geometry Truss Manipulator", *M.S. Thesis*, Department of Mechanical Engineering, Virginia Polytechnic Institute & State University, Blacksburg, VA.

Ting, K-L. and Long, Y., 1994, "Performance Quality, Sensitivity, and Tolerance Specification of Mechanisms," *Mechanism Synthesis and Analysis, Proceedings of the ASME 23rd Biennial Mechanisms Conference*, DE-Vol. 70, pp. 387-396.

Uchiyama, M., 1994, "Structures and Characteristics of Parallel Manipulators," *Advanced Robotics*, Vol. 8, No. 6, pp. 545-557.

Volkov, E. A., 1990, *Numerical Methods*, Hemisphere Publishing Corp., New York, NY.

Voruganti, R. S., 1995, "Robot System Characterization: Error Modeling, Identification, Analysis, and Minimization," *Ph.D. Dissertation*, Department of Mechanical Engineering, Virginia Polytechnic Institute & State University, Blacksburg, VA.

Wu, F. and Lankarani, H.M., 1992, "A New Parameter for Transmission Quality and Output Sensitivity Analysis of Mechanisms," *Mechanism Design and Synthesis, Proceedings of the ASME 22nd Biennial Mechanisms Conference*, DE-Vol. 46, pp. 103-109.

Zhuang, H., Masory, O. and Yan, J., 1995, "Kinematic Calibration of a Stewart Platform Using Pose Measurements Obtained by a Single Theodolite," *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems: Human Robot Interaction and Cooperative Robots*, Vol. 2, pp. 329-334.

Appendix 1: Listing of Matlab Codes

The Matlab software package by The MathWorks was used to implement the Carpal wrist error model. The *m-file* codes used are listed below, beginning with the driver `sensitiv.m`, followed by the inverse kinematics solver, `inv_sol.m`, the Newton-Raphson Method function, `newt_rap.m`, and the general rotation matrix function, `rotation.m`. The codes have been commented to facilitate better understanding during future use, but the user may also want to review Chapters 2 and 3 for a more complete definition of terms.

A1.1: Sensitivity Analysis Driver

```
% sensitiv.m - Carpal Wrist Error Model and Sensitivity Analysis Driver
%
% By:     Greg Saccoccio
% Date:   8/30/96
%
% Purpose:    This m-file, to be run with Matlab version 4.2 or higher,
%             implements the Carpal wrist error model solution technique.
%             It requires the user to input the nominal kinematic parameter
%             values as well as any errors in the system, as noted in the
%             comments below.

% Some general constants to be used throughout

   angle_step = 2.5;       % Increment for bend angle and bend axis angle
   dtr = pi/180;           % Convert degrees to radians for trig functions
   rtd = 180/pi;           % Convert radians to degrees
   half_pi = pi/2;         % Define pi/2 for simplicity


% Define values of kinematic parameters and corresponding errors

   plunge = 7;                          % Set plunge distance

   leg = 8;                             % Leg length nominal dimension
   delta_l = [0 0 0 0 0 0];             % Leg length error vector
   l = leg*ones(1,6)+delta_l;           % Actual leg length vector

   base = 3;                            % Basal and distal nominal dimension
   delta_b = [0 0 0];                   % Basal error
   b_dim = base*ones(1,3)+delta_b;      % Actual basal values

   beta_nom = [2*pi/3 2*pi/3 2*pi/3]; % Basal plate nominal angles (radians)
```

```
    delta_beta = [0 0 0];              % Basal angle variation (radians)
    beta = beta_nom + delta_beta;      % Actual basal plate angles

    eta = [0 0 0];                     % Basal Revolute Out-of-Plane Offset
    mu = [0 0 0];                      % Basal Revolute Tilt (radians)

    g_id = sqrt(3)*base;               % Ideal basal or distal point connectors
    delta_g_d = [0 0 0];               % Variation for distal point connector
    g_d = g_id*ones(1,3) + delta_g_d; % Actual distal point connector values


% Compute Basal plate information

    % First revolute axis u_1 aligned with Y_B and q_1 along -X_B
    u_ip(:,1) = [ 0; 1; 0];
    q(:,1) = [-1; 0; 0];

    % u_2 and q_2 in terms of beta_1
    u_ip(:,2) = [ -sin(beta(1));  cos(beta(1)); 0 ];
    q(:,2) = [ -cos(beta(1)); -sin(beta(1)); 0 ];

    % u_3 and q_3 in terms of beta_3
    u_ip(:,3) = [  sin(beta(3)); cos(beta(3)); 0 ];
    q(:,3) = [ -cos(beta(3)); sin(beta(3)); 0 ];

    for j = 1:3
       % Account for u-vectors tilted out-of-plane
       u(:,j) = u_ip(:,j) + tan(mu(j))*[0;0;1];

       % Compute b vectors (and account for out-of-plane offset)
       b(:,j)=-b_dim(j)*q(:,j) + eta(j)*[0;0;1];
    end;

% Begin Workspace Loop of alpha (bend axis angle) and phi (bend angle)

    % Initialize index counters
    v = 1;
    w = 1;

    for alpha = 0:angle_step:360

       closure_flag = 1;    % Re-initialize closure flag for ideal mechanism
       v = 1;               % Re-initialize matrix column counter
       w = w + 1;           % Increment matrix row counter

       % Store value of alpha in first row of output matrices
       x_id(1,w) = alpha;
       y_id(1,w) = alpha;
       z_wk(1,w) = alpha;
       e_d_1(1,w) = alpha;
       e_d_2(1,w) = alpha;
       e_d_3(1,w) = alpha;
       E(1,w) = alpha;

       for phi = 0:angle_step:179
```

```
v = v + 1;       % Increment matrix column counter

% Store value of phi in first column of output matrices
x_id(v,1) = phi;        % Ideal workspace matrices
y_id(v,1) = phi;
z_wk(v,1) = phi;
x_sp(v,1) = phi;        % Actual workspace matrices
y_sp(v,1) = phi;
z_sp(v,1) = phi;
e_d_1(v,1) = phi;       % Error in first distal revolute
e_d_2(v,1) = phi;       % Error in second distal revolute
e_d_3(v,1) = phi;       % Error in third distal revolute
E(v,1) = phi;           % Pose error matrix

% Solve Inverse Kinematics of Ideal Case to get input angles
% by calling function inv_sol.m
[theta1,theta2,theta3,cdx_id,cdy_id,cdz_id,d_id,closure_flag] =
      inv_sol(alpha,phi,leg,base,plunge,closure_flag);

% Check if mechanism assembles using criteria of Inverse solution
if closure_flag == 0

   % Mechanism does not assemble, store point as empty (NaN)
   x_id(v,w) = NaN;
   y_id(v,w) = NaN;
   z_wk(v,w) = NaN;
   x_sp(v,w) = NaN;
   y_sp(v,w) = NaN;
   z_sp(v,w) = NaN;
   e_d_1(v,w) = NaN;
   e_d_2(v,w) = NaN;
   e_d_3(v,w) = NaN;
   E(v,w) = NaN;

else

   % Store IDEAL data in matrices
   x_id(v,w) = cdx_id;
   y_id(v,w) = cdy_id;
   z_wk(v,w) = cdz_id;

   % Compute IDEAL distal revolute axis unit vectors
   u_d_id(:,1) = (d_id(:,2)-d_id(:,3))./norm(d_id(:,2)-d_id(:,3));
   u_d_id(:,2) = (d_id(:,3)-d_id(:,1))./norm(d_id(:,3)-d_id(:,1));
   u_d_id(:,3) = (d_id(:,1)-d_id(:,2))./norm(d_id(:,1)-d_id(:,2));

   % Convert IDEAL distal revolute points to IDEAL intersection
   % points of revolute axes
   d_prime_id(:,1) = d_id(:,1) + g_id*u_d_id(:,1);
   d_prime_id(:,2) = d_id(:,2) + g_id*u_d_id(:,2);
   d_prime_id(:,3) = d_id(:,3) + g_id*u_d_id(:,3);

   % Begin Forward Kinematics of NON-IDEAL model
```

```
% Find rotation matrices and store in 9x3 matrix
R = [rotation(u(:,1),theta1)
     rotation(u(:,2),theta2)
     rotation(u(:,3),theta3)];

% Locate Mid-plane Spheric Joints by computing m-vectors
i = 0;
for j = 1:3
   m(:,j) = b(:,j) + l(j)*R(j+i:j+i+2,:)*q(:,j);
   i = i + 2;
end;

% Form 9x1 initial solution guess vector as [d'1x d'1y...d'3z]'
X_guess = [ d_prime_id(:,1); d_prime_id(:,2); d_prime_id(:,3)];

% Send initial guess into Newton-Raphson function
[X,it_num,it_stuck,partials] = newt_rap(X_guess,m,l,g_d);

if it_stuck == 1    % Check convergence counter

   % Store as non-assembled point
   x_sp(v,w) = NaN;
   y_sp(v,w) = NaN;
   z_sp(v,w) = NaN;
   E(v,w) = NaN;

else

   % Rename solution as d_prime_actual as columns of matrix
   d_prime_ac = [   X(1) X(4) X(7)
                    X(2) X(5) X(8)
                    X(3) X(6) X(9) ];

   % ACTUAL Distal revolute axis unit vectors
   u_d_ac(:,1) = (d_prime_ac(:,1)-
      d_prime_ac(:,3))./norm(d_prime_ac(:,1)-d_prime_ac(:,3));
   u_d_ac(:,2) = (d_prime_ac(:,2)-
      d_prime_ac(:,1))./norm(d_prime_ac(:,2)-d_prime_ac(:,1));
   u_d_ac(:,3) = (d_prime_ac(:,3)-
      d_prime_ac(:,2))./norm(d_prime_ac(:,3)-d_prime_ac(:,2));

   % ACTUAL Distal revolute points
   d_ac(:,1) = d_prime_ac(:,3) + g_d(2)*u_d_ac(:,1);
   d_ac(:,2) = d_prime_ac(:,1) + g_d(3)*u_d_ac(:,2);
   d_ac(:,3) = d_prime_ac(:,2) + g_d(1)*u_d_ac(:,3);

   % Distal revolute error quantities
   e_d_1 = norm(d_ac(:,1)-d_id(:,1));
   e_d_2 = norm(d_ac(:,2)-d_id(:,2));
   e_d_3 = norm(d_ac(:,3)-d_id(:,3));

   % Pose error quantity
   E(v,w) = e_d_1 + e_d_2 + e_d_3;

end;
```

```
            end;
        end;    % End phi loop
    end;        % End alpha loop

% Workspace edge points (for plotting)
i = 0;
j = 0;
for j = 2:w
    for i = 2:v
        % Ideal workspace edge
        if z_wk(i,j) <= 9999
            edge_x(j) = x_id(i,j);
            edge_y(j) = y_id(i,j);
            edge_z(j) = z_wk(i,j);
        end;
        % Non-ideal workspace edge
        if z_sp(i,j) <= 9999
            edge_Ex(j) = x_sp(i,j);
            edge_Ey(j) = y_sp(i,j);
            edge_Ez(j) = z_sp(i,j);
        end;
    end;
end;

% Count empty workspace elements and compute
% percentage lost due to error
empties = 0;
count = 0;
i = 0;
j = 0;
for i = 2:v
    for j = 2:w

        if E(i,j) <= 9999
            count = count + 1;
        end;
    end;
end;
count;
empties = (w-1)*(v-1) - count;
percent_lost = 100*(empties/((w-1)*(v-1)))

% Statements to generate error plot
pcolor(E(1,2:w),E(2:v,1),E(2:v,2:w));
shading('flat');
axis('ij');
axis('image');
colormap(jet);
colorbar;
xlabel('Bend Axis Angle, Alpha (deg.)');
ylabel('Bend Angle, Phi (deg.)');
```

A1.2: Inverse Kinematics Function

```
function [theta1,theta2,theta3,x_id,y_id,z_wk,d_id,closure_flag] =
   inv_sol(alpha,phi,leg,base,plunge,closure_flag)

% inv_sol.m - INVERSE KINEMATIC Solution Function
%
% By:     Greg Saccoccio
% Date:   8/30/96
%
% INPUTS: alpha  - bend axis angle (in degrees)
%         phi    - bend angle (in degrees)
%
%         leg    - nominal leg dimension
%         base   - nominal base dimension
%         plunge - plunge distance
%
%         closure_flag - mechanism assembly criteria
%
% OUTPUTS:   theta1,2,3 - value of input angles (in radians)
%
%         x_id
%         y_id      - IDEAL distal center where mechanism assembles
%         z_wk
%
%         d_id      - IDEAL distal revolute intersection points
%
%         closure_flag - updated mechanism assembly status (1-assembled,
%                        0-not assembled)
%

% Define Ideal Basal angle
beta = [0 2*pi/3 4*pi/3];

for j = 1:3

   % Computing u and q vectors
   u(:,j)=[-sin(beta(j));  cos(beta(j)); 0];
   q(:,j)=[-cos(beta(j)); -sin(beta(j)); 0];
   % Compute b vectors
   b(:,j)= -base*q(:,j);

end;

% Wrist modelled as two rods joined by spheric joint

   dtr = pi/180;       % Convert degrees to radians
   alpha = alpha*dtr;
   phi   = phi*dtr;

   % Define Z-axes in Basal frame
   z_b = [0; 0; 1];
```

```
z_d = [sin(alpha)*sin(phi); -cos(alpha)*sin(phi); cos(phi)];

% Store goal (IDEAL distal center) point
goal = plunge*(z_b + z_d);
x_id = goal(1);
y_id = goal(2);
z_id = goal(3);
z_wk = z_id;

if closure_flag == 1    % If mechanism assembles, solve for input angles

    u_bend = [cos(alpha); sin(alpha); 0];
    [R_bend] = rotation(u_bend,phi);

    for j = 1:3
        % Distal revolute location
        d(:,j) = [b(1,j); b(2,j); plunge];
        d(:,j) = R_bend*d(:,j) + [0; 0; plunge];
    end;

    % Three points on plane of symmetry
    p = (d + b)./2;

    n = cross((p(:,2)-p(:,1)),(p(:,3)-p(:,1)));% Mid plane normal vector
    Am = n(1);
    Bm = n(2);
    Cm = n(3);
    Dm = dot(-n,p(:,1));

    Dc = 0;

    for j = 1:3

        temp1 = [0 0];
        Ac = u(1,j);
        Bc = u(2,j);
        Cc = u(3,j);

        denom = Cc*Bm-Bc*Cm;    % Solving for the mid-plane nodes
        R = (Ac*Cm-Cc*Am)/denom;
        S = (Dc*Cm-Cc*Dm)/denom;
        T = (Bc*Am-Ac*Bm)/denom;
        U = (Bc*Dm-Dc*Bm)/denom;

        A = R*R + T*T + 1;
        B = 2*R*S + 2*T*U - 2*b(1,j) - 2*R*b(2,j) - 2*T*b(3,j);
        C = S*S + U*U - 2*S*b(2,j) - 2*U*b(3,j) + base*base - leg*leg;

        radical = B*B - 4*A*C;

        % Check for assembly of mechanism
        if radical >= 0
            % Solve for m_ix roots and Store in t vector
            temp1(1) = (-B+sqrt(radical))/(2*A);
            temp1(2) = (-B-sqrt(radical))/(2*A);
```

```
            else
               closure_flag = 0;
               break;
            end;

            % Closure criteria for selecting root of m_ix
            if j == 1
               m(1,j) = max(temp1);
            else
               m(1,j) = min(temp1);
            end;

            % Find m_iy and m_iz
            m(2,j) = R*m(1,j)+S;
            m(3,j) = T*m(1,j)+U;

            % Find input angle and store in theta matrices
            temp2 = m(:,j)-b(:,j);
            temp5 = dot(temp2,q(:,j));
            temp6 = norm(temp2)*norm(q(:,j));
            temp3 = acos(temp5/temp6);
            temp4 = dot(cross(q(:,j),temp2),u(:,j));

            if temp4 < 0
               if j == 1
                  theta1 = 2*pi-temp3;
               elseif j == 2
                  theta2 = 2*pi-temp3;
               else
                  theta3 = 2*pi-temp3;
               end;
            else
               if j == 1
                  theta1 = temp3;
               elseif j == 2
                  theta2 = temp3;
               else
                  theta3 = temp3;
               end;
            end;
         end;
      end;

      if closure_flag == 0          % If mechanism does not assemble

         theta1 = NaN;
         theta2 = NaN;
         theta3 = NaN;
         z_wk = NaN;

      end;

   d_id = d; % Set IDEAL distal revolute points
```

A1.3: Newton-Raphson Function

```
function [x_done,it_num,it_stuck,partials] = newt_rap(X_guess,m,l,g_d)

% newt_rap.m - NEWTON-RAPHSON METHOD SOLVING FUNCTION
%
% By:     Greg Saccoccio
% Date:   8/30/96
%
%  An m-file to solve the system of 9 equations/9 unknowns in order
%  to find the points (d_i prime) which locate the intersection of
%  the distal revolute axes for the non-ideal model.
%
% Inputs: X_guess - the initial guess of the solution, location of
%                   intersection points from ideal solution
%
%         m - matrix of mid-plane node vectors, stored as columns of 3 x 3
%
%         l - leg lengths, 1 x 6 vector
%
%         g_d - 3 x 1 vector of VGT triangle half-side lengths
%
% Outputs:    x_done - 9 X 1 solution vector to the system of equations with
%                      elements as distal revolute axis intersection points
%
%         it_num - number of iterations required
%


% Get size of system
[row,col] = size(X_guess);

% Initialize the matrices and vectors
partials = zeros(row,row);
delta_x = ones(row,col);
x = zeros(row,col);
f = zeros(row,col);

it_num = 0;
it_stuck = 0; % Iteration divergence flag

x_next = X_guess;
checker = 0;

while norm(delta_x) > 0.00001

   x = x_next;
   it_num = it_num + 1;

   % Iteration stop if diverging
   if it_num >= 25
      it_stuck = 1;
```

```
        break;
    end;

    % Compute equations 1 thru 9
    f(1) = (x(1)-m(1,1))^2 + (x(2)-m(2,1))^2 + (x(3)-m(3,1))^2 -
        (l(4)^2+g_d(2)^2);
    f(2) = (x(1)-m(1,2))^2 + (x(2)-m(2,2))^2 + (x(3)-m(3,2))^2 -
        (l(5)^2+g_d(3)^2);
    f(3) = (x(4)-x(1))^2 + (x(5)-x(2))^2 + (x(6)-x(3))^2 - (2*g_d(3))^2;
    f(4) = (x(4)-m(1,3))^2 + (x(5)-m(2,3))^2 + (x(6)-m(3,3))^2 -
        (l(6)^2+g_d(1)^2);
    f(5) = (x(4)-m(1,2))^2 + (x(5)-m(2,2))^2 + (x(6)-m(3,2))^2 -
        (l(5)^2+g_d(3)^2);
    f(6) = (x(7)-x(4))^2 + (x(8)-x(5))^2 + (x(9)-x(6))^2 - (2*g_d(1))^2;
    f(7) = (x(7)-m(1,1))^2 + (x(8)-m(2,1))^2 + (x(9)-m(3,1))^2 -
        (l(4)^2+g_d(2)^2);
    f(8) = (x(7)-m(1,3))^2 + (x(8)-m(2,3))^2 + (x(9)-m(3,3))^2 -
        (l(6)^2+g_d(1)^2);
    f(9) = (x(1)-x(7))^2 + (x(2)-x(8))^2 + (x(3)-x(9))^2 - (2*g_d(2))^2;

    if norm(f) <= 0.00001
        x_done = x;
        break;
    else

        % Form matrix of partial derivatives
        partials(1,1:3) = [ 2*(x(1)-m(1,1)) 2*(x(2)-m(2,1)) 2*(x(3)-m(3,1)) ];
        partials(2,1:3) = [ 2*(x(1)-m(1,2)) 2*(x(2)-m(2,2)) 2*(x(3)-m(3,2)) ];
        partials(3,4:6) = [ 2*(x(4)-x(1)) 2*(x(5)-x(2)) 2*(x(6)-x(3)) ];
        partials(3,1:3) = -partials(3,4:6);
        partials(4,4:6) = [ 2*(x(4)-m(1,3)) 2*(x(5)-m(2,3)) 2*(x(6)-m(3,3)) ];
        partials(5,4:6) = [ 2*(x(4)-m(1,2)) 2*(x(5)-m(2,2)) 2*(x(6)-m(3,2)) ];
        partials(6,7:9) = [ 2*(x(7)-x(4)) 2*(x(8)-x(5)) 2*(x(9)-x(6)) ];
        partials(6,4:6) = -partials(6,7:9);
        partials(7,7:9) = [ 2*(x(7)-m(1,1)) 2*(x(8)-m(2,1)) 2*(x(9)-m(3,1)) ];
        partials(8,7:9) = [ 2*(x(7)-m(1,3)) 2*(x(8)-m(2,3)) 2*(x(9)-m(3,3)) ];
        partials(9,1:3) = [ 2*(x(1)-x(7)) 2*(x(2)-x(8)) 2*(x(3)-x(9)) ];
        partials(9,7:9) = -partials(9,1:3);

        % Solve System of Linear Equations
        delta_x = inv(partials)*(-f);

        % Compute next x
        x_next = x + delta_x;

    end;

end;

x_done = x_next;
```

A1.3: Rotation Matrix Function

```
function [R] = rotation(k,theta)

% Rotation matrix generator function
%
% By:     Greg Saccoccio
% Date:   8/30/96
%
% Input:  3x1 Vector k, the axis of rotation
%         Angle theta (in radians)
%
% Output: Matrix R
%

% Check dimensions of vector k
[m,n] = size(k);
if m~= 3
   error('Vector has too many rows.');
elseif n~=1
   error('Vector has too many columns.');
end;

% Using equation 2.80 from Craig (1989)

v = 1 - cos(theta);
c = cos(theta);
s = sin(theta);

R = [ k(1)^2*v+c          k(1)*k(2)*v-k(3)*s  k(1)*k(3)*v+k(2)*s
      k(1)*k(2)*v+k(3)*s      k(2)^2*v+c      k(2)*k(3)*v-k(1)*s
      k(1)*k(3)*v-k(2)*s  k(2)*k(3)*v+k(1)*s      k(3)^2*v+c ];
```

Vita

Gregory Nicholas Saccoccio was born on August 9, 1972, in the smallest of states, Rhode Island. He enrolled at Boston University in the fall of 1990, and received his Bachelor of Science degree in mechanical engineering in May, 1994. While completing his undergraduate studies, he worked as an engineering co-op for eight months at the Charles Stark Draper Laboratory in Cambridge, Massachusetts. In the fall of 1994, he began graduate studies at Virginia Polytechnic Institute and State University, with a focus in mechanical design. As a graduate student, he participated in the Siemens Automotive design project group and was employed as a summer intern with Merck & Co., Inc. Upon completion of his Master of Science degree, he will begin employment as a design engineer with the Lockheed Martin Skunk Works in Palmdale, California.

