Kinematic Control Experiments with Trussarm, a Variable-Geometry-Truss Manipulator

by

Regina Sun-Kyung Lee

A thesis submitted in conformity with the requirements for the Degree of Doctor of Philosophy
University of Toronto
Institute for Aerospace Studies

©Copyright by Regina Sun-Kyung Lee, 2000
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-49815-8
Abstract

Trussarm is a Variable-Geometry-Truss (VGT) manipulator, developed and constructed at the University of Toronto Institute for Aerospace Studies. The objective of this dissertation is to examine this new technology to determine its kinematics and control characteristics. In order to examine the kinematics properties of Trussarm, a kinematics model is developed. An algorithm to represent the macroscopic geometry of Trussarm as an equivalent discrete model (a serial open chain of simple 3-DOF gimbal modules) is presented. Redundancy resolution based on the equivalent discrete model is developed, and the kinematics and differential kinematics of a Trussarm module are derived. Through experiments and simulations, the control characteristics of Trussarm including workspace, dexterity, speed, and tracking accuracy, are determined. Furthermore, a new control algorithm is developed to improve the control performance of Trussarm. The experimental results suggest that Trussarm possesses many advantages over conventional manipulators although its accuracy suffers from its physical size and complexity. However, the accuracy may be improved with closed-loop position control.
Acknowledgments

I would like to express my sincere gratitude to many people who made this work possible.

First of all, I would like to thank my supervisor, Prof. Peter C. Hughes, for his academic and technical advice throughout my graduate studies. I also would like to thank my doctoral committee members, Prof. Gabriele M.T. D’Eleuterio and Dr. Frank Naccarato, for their technical advice in all aspects of this work. Much of the work on the kinematics development in Chapters 3 and 4 is the result of conversations with Dr. Kourosh Zanganeh. He patiently taught me everything about hyper-redundant manipulators.

The conversations with my colleagues in the Spacecraft Dynamics and Robotics group have been invaluable to me. I am especially indebted to Thierry Cherpillod for his technical support on the design and construction of Trussarm. Much of the work on the design of Trussarm was also provided by many previous students, including Stephen Oikawa and Roger Hertz. Also, many thanks to Andrew Allen for proofreading parts of this document.

I also would like to acknowledge the Natural Science and Engineering Research Council (NSERC), ISTS (now CresTech), University of Toronto, and Dynacon Inc. for financial support during my doctoral program.
I also wish to thank my parents, Sang-Wook and Jung-Ja Lee, and my sister, Sophia, for their love and support. Their constant encouragement and patience have been and will always be appreciated. Lastly, I would like to take this opportunity to thank my husband, Thierry. Without his love and support and most of all, his faith in me, this work would not have been possible.

July, 1999
Maple, Ontario
Contents

Abstract ............................................. i
Acknowledgments ................................... ii

List of Figures ...................................... viii

Chapter 1 Introduction ............................... 1
  1.1 Background ..................................... 1
    1.1.1 Hyper Redundant Manipulators ............. 1
    1.1.2 Variable Geometry Truss Manipulators ... 2
  1.2 Trussarm ...................................... 4
  1.3 Motivation and Purpose ....................... 4
  1.4 Literature Review ............................ 6
    1.4.1 Kinematics Analysis of Hyper-Redundant VGT Manipulators .......... 6
    1.4.2 Design of and Experiments with VGT Manipulators .......... 8
    1.4.3 Kinematics Analysis of Trussarm ........... 14
  1.5 Overview of this Document ................... 15

Chapter 2 System Description ..................... 17
  2.1 Actuators and Servomechanisms ............... 17
    2.1.1 Trussarm Actuator ....................... 17
2.1.2 Servomechanism ........................................ 19

2.2 Supporting Structure ........................................ 20
2.2.1 Modules .................................................. 20
2.2.2 Passive Members ........................................ 20
2.2.3 Trussarm Coordinate System ............................. 21

2.3 Trussarm Control Environment (TCE) ....................... 22
2.3.1 Trajectory Planning ....................................... 23
2.3.2 Hyper-Redundancy Resolution ........................... 23
2.3.3 Kinematics ............................................... 23
2.3.4 GALIL commands ......................................... 23
2.3.5 Motion Execution ......................................... 24
2.3.6 Cartesian Control ......................................... 24

Chapter 3 Kinematics ............................................ 25
3.1 Kinematics of an Equivalent Discrete Model ............... 26
3.1.1 3-DOF Extensible Gimbal Module ....................... 26
3.1.2 Chain of Gimbal Modules ................................ 28
3.2 Kinematics of a Trussarm Module ........................... 31
3.2.1 Inverse Kinematics ....................................... 31
3.2.2 Forward Kinematics ..................................... 35
3.3 Joint-Limit Avoidance Problems ............................ 37

Chapter 4 Differential Kinematics .............................. 42
4.1 Resolved Motion Rate Control ............................... 43
4.2 Jacobian Matrix and Hessian ................................ 44
4.3 Differential Kinematics of a Trussarm Module .............. 50
4.3.1 Differential Inverse Kinematics ........................ 51
Chapter 5 Performance Measurements

5.1 Workspace Analysis
   5.1.1 Definition
   5.1.2 Workspace Measurement Methods
   5.1.3 Trussarm Workspace Experiments
   5.1.4 Controllable Workspace
   5.1.5 Reach Hierarchy Algorithm

5.2 Dexterity Analysis
   5.2.1 Definition
   5.2.2 Dexterity Measurement

5.3 Repeatability Analysis
   5.3.1 Definition
   5.3.2 Repeatability Measurement Method
   5.3.3 Trussarm Repeatability

5.4 Speed Analysis
   5.4.1 Actuator Speed
   5.4.2 Tip Speed

5.5 Accuracy Analysis
   5.5.1 Definition
   5.5.2 Position Control
   5.5.3 Kinematic (Tracking) Control
   5.5.4 On-Line Cartesian Control
   5.5.5 Velocity Tracking Control

Chapter 6 Control System Design
6.1 Control Design ........................................ 103
6.2 Simulation Test ......................................... 104
6.3 Implementation ........................................ 108

Chapter 7 Conclusion .................................. 110
  7.1 Final Remarks ........................................ 110
  7.2 Contributions ........................................ 112
  7.3 Future Work .......................................... 113
References ............................................... 115

Appendix A An Example of a Constrained Optimization Problem 121
List of Figures

1.1 A Discrete Morphology Hyper-redundant Manipulator, Caltech ............................................ 2
1.2 Trussarm ........................................................................................................................................ 3
1.3 3-DOF VGT at NASA .................................................................................................................. 9
1.4 Trussarm Mark I at UTIAS .......................................................................................................... 9
1.5 Tapered Two-bay Octahedral-octahedral VGT at NASA LRC .................................................. 10
1.6 Serpentine Truss Manipulator, NASA Kennedy Space Center ............................................... 10
1.7 Planar VGT, Caltech .................................................................................................................. 11
1.8 Tetrobot ......................................................................................................................................... 11
1.9 Payload Inspection and Processing System .................................................................................. 13

2.1 Trussarm Module ........................................................................................................................ 20
2.2 Actuator Geometry ....................................................................................................................... 21
2.3 Hinge Assembly ........................................................................................................................... 21
2.4 Trussarm Control System ............................................................................................................ 22

3.1 3-DOF extensible gimbal module ............................................................................................... 26
3.2 Kinematics of a discrete model .................................................................................................. 27
3.3 Vectors in the kinematics of a Trussarm module ....................................................................... 41

5.1 Trussarm Workspace by Extension Method .............................................................................. 61
5.2 Trussarm Workspace by Extension Method, \(zz\)-plane
5.3 Workspace Boundary Approximation
5.4 Workspace of 1 Module
5.5 Workspace of 2 Modules
5.6 Workspace of 3 Modules
5.7 Workspace of 4 Modules
5.8 Trussarm Workspace by Sweeping Method, \(zz\)-plane
5.9 Trussarm Workspace by Sweeping Method, \(zz\)-plane
5.10 Home Configuration
5.11 J-Shape Configuration
5.12 Contracted Configuration
5.13 Extended Configuration
5.14 Positive \(x\) Reach
5.15 Negative \(x\) Reach
5.16 Workspace of PUMA761 manipulator
5.17 Trussarm Controllable Workspace, \(zz\)-plane
5.18 Trussarm Controllable Workspace, \(xy\)-plane
5.19 Controllable Workspace with Reach Hierarchy Algorithm
5.20 Dexterity Measures along a curve (\(zz\)-plane)
5.21 Dexterity Measures along a straight line (\(z\)-axis)
5.22 Posture 1
5.23 Posture 2
5.24 Posture 3
5.25 Posture 4
5.26 Dexterity Measures at Posture 1, 2, 3 and 4
5.27 Setup for Repeatability Analysis
5.28 Repeatability Test Result 1
5.29 Repeatability Test Result 2 ........................................... 84
5.30 Repeatability Test Result 3 ........................................... 84
5.31 Vibration During Acceleration and Deceleration ................. 86
5.32 The 12-inch Pick-and-Place Maneuver .............................. 87
5.33 Square Trajectory in Accuracy Analysis ............................ 90
5.34 Tracking Accuracy Analysis: Square Trajectory, n=2, xz-plane 91
5.35 Configuration of Trussarm: Square Trajectory, n=2, xz-plane 91
5.36 Configuration of Trussarm: Square Trajectory, n=4, xy-plane 92
5.37 Tracking Accuracy Analysis: Square Trajectory, n=4, xy-plane 93
5.38 Tracking Accuracy Analysis: Square Trajectory, n=4, xz-plane 93
5.39 Tracking Accuracy Analysis: Circular Trajectory, n=4, xz-plane 94
5.40 Velocity Tracking Control Simulation Results, n_t = 20 ........ 98
5.41 Velocity Tracking Control Simulation Results, n_t = 1000 ...... 99
5.42 Second norm of the error vs. number of trajectory points ..... 99
5.43 Velocity Tracking Control, n_t = 50, xz-plane .................... 102

6.1 Augmented Kinematic Control Algorithm ........................... 105
6.2 Error obtained from the Simulation of Closed-Loop Control, Square Trajectory, n=2, xz-plane .............................. 106
6.3 Simulation Results from Closed-Loop Control, Square Trajec-
tory, n=2, xz-plane .................................................. 107
6.4 Simulation Results from Closed-Loop Control, Circular Trajec-
tory, n=2, xz-plane .................................................. 107
6.5 Tracking Error vs. Number of Iteration around Fine Positioning
Control ................................................................. 108
6.6 Results from Closed-Loop Control, Square Trajectory, n=2, xz-
plane ................................................................. 109
6.7 Results from Closed-Loop Control, Circular Trajectory, \( n=2 \),
\[ zz \text{-plane} \]
Chapter 1

Introduction

1.1 Background

1.1.1 Hyper Redundant Manipulators

Hyper-redundant manipulators are those that have very large degrees of kinematic redundancy. These manipulators are often used to perform tasks that conventional robots are incapable of accomplishing. Examples include manipulators in a highly constrained environment, snake-like maneuvers, and the grasping of objects. As Immega and Antonelli (1995) pointed out, "the possibilities [of the applications of hyper-redundant manipulators] are limited only by the imagination."

On the other hand, an efficient hyper-redundant manipulation highly depends on the available methods for resolving the hyper-redundant kinematics. Most known methods of redundancy resolution lead to serial computations, hence their computational complexity grows with the manipulator's degrees of freedom. Due to the computational difficulties associated with hyper-redundancy, hyper-redundant manipulators have been considered inapplicable
Hyper-redundant manipulators are generally grouped into three categories according to their morphologies: continuous morphology, discrete morphology, and Variable-Geometry-Truss (VGT). A tentacle is an example of a continuous morphology hyper-redundant manipulator. A discrete morphology hyper-redundant manipulator is a long chain of serial linkages resulting in a manipulator with a very large number of degrees of freedom. A spatial hyper-redundant manipulator of discrete morphology (Figure 1.1) is under development at the California Institute of Technology. Trussarm (Figure 1.2) is an example of a hyper-redundant Variable-Geometry-Truss manipulator.

1.1.2 Variable Geometry Truss Manipulators

The origin of Variable-Geometry-Truss (VGT) structures dates back to the early 80's when deployable and stackable structures were first considered for space applications. The VGT concept was derived from the concept of 'adaptive', or 'smart' [Robertshaw and Reinholtz, 1988] structures. Since then,
numerous applications for VGT structures have been examined, including the manipulation of payloads.

VGT manipulators consist of a series of self-contained modules. Each module, in turn, is a parallel manipulator with variable-length members. The advantages of a VGT manipulator, compared to a conventional manipulator of the same volume and mass, were realized early in the development: stiffness, modularity, flexibility, and dexterity. Because of these advantages, VGT manipulators have been considered for applications in space and undersea explorations and inspection. Examples of proposed applications for VGT manipulators include waste storage tank remediation [Salerno and Reinholtz, 1994], on-site equipment removal [Williams and Mayhew, 1997a], and payload inspection [Williams and Mayhew, 1997b].

More recently, VGT technology has been also considered for manufac-
turing and automation applications. Due to the modularity in VGT manipulators, they have advantages for tasks that require changes in configuration, or tasks in a constrained environment.

1.2 Trussarm

The term 'Trussarm' refers to a modular VGT manipulator design developed at the University of Toronto Institute for Aerospace Studies (UTIAS). Laboratory prototypes have been built to support the study of structures, manipulator design, and the kinematics, dynamics, and control associated with VGT manipulators. The first Trussarm, Trussarm Mark I, was built in the early 90's as a proof-of-concept VGT manipulator. As shown in Figure 1.4, Trussarm Mark I is a 3 degree-of-freedom, double-octahedral manipulator.

More recently, a 12-DOF prototype namely Trussarm Mark II (which will be simply referred to as 'Trussarm' hereinafter), was built. Trussarm consists of four identical 3-degree-of-freedom (3-DOF) modules that are connected in series. Each module is a double-octahedral parallel manipulator consisting of two identical octahedral modules sharing a common triangular facet. Moreover, each double-octahedral parallel manipulator has three extensible links that can be adjusted in lengths continuously, using three prismatic actuators.

1.3 Motivation and Purpose

The main objective of this dissertation is to develop a kinematics control scheme for a hyper-redundant Variable-Geometry-Truss (VGT) manipulator, such as Trussarm, and to investigate its control characteristics through experiments and simulations. The results obtained from this dissertation can be
used to explore industrial applications.

In order to develop an efficient control scheme for Trussarm, the following research objectives have been proposed:

- to efficiently model the kinematics of Trussarm
- to analyze the performance characteristics of Trussarm
- to design and implement control schemes for Trussarm

The kinematics of Trussarm has been previously presented in (Sallmen, 1993). However, the formulation of the kinematics used here takes advantage of the structural symmetry of Trussarm, thus providing a physically meaningful model with a very simple mathematical presentation. One of the contributions of this dissertation in the field of VGT kinematics is the development of a kinematics model at the velocity level. The model presented here is computationally simple and easy to implement.

Many hyper-redundant VGT manipulators remained largely at the experimental level, primarily due to the difficulties in the implementation of control schemes. As a result, many characteristics of VGT manipulators have been examined only through simulations and experimental verifications have not been performed. Throughout this dissertation, a number of experiments designed to examine the control characteristics of Trussarm are presented. The experimental results are compared with simulation results and suggestions for progressive improvements are made. The experiments presented here may be adopted for other VGT manipulators to examine their applicability to practical applications.

A control algorithm based on the kinematics control experiments, is proposed to improve tracking accuracy. The control design employs a two-
layer approach to the kinematics control. The method is implemented and verified through a series of simulations and experiments.

1.4 Literature Review

1.4.1 Kinematics Analysis of Hyper-Redundant VGT Manipulators

A common approach to the kinematics problem for a hyper-redundant VGT manipulator is configuration control. In this approach, the shape of the backbone curve — a curve that captures the macroscopic geometric features of the manipulator — is controlled throughout the motion in order to achieve a set of predefined objectives. These objectives can depend on the kinematics, dynamics, or other relevant features of the system. Many papers have been written on configuration control.

Salerno, Reinholtz and Dhande (1990) used cubic parametric curves to capture the geometry of a long chain of VGT manipulators. They explained the advantages of the cubic curves: “the parametric description allows for curve slopes to pass through the vertical, without numerical difficulties” and also “allows for more flexibility in controlling the curve shape.”

Naccarato and Hughes (1991) introduced the concept of reference shape curves, which is equivalent to the backbone curves. To model the kinematics of the reference shape curve, several space curves were suggested including Bezier curves, B-splines, and Beta-splines. Naccarato (1994) used Bezier curves to model the kinematics of Trussarm.

The above two examples are both extrinsic schemes to parameterize a backbone curve. The backbone curve is modeled by mathematic space curves
such as splines. An alternative is to adopt an _intrinsic scheme_ where a backbone curve is defined by its intrinsic representation in terms of a natural parameter. Chirikjian and Burdick (1990 - 1994) used this scheme to define the backbone curve by using four independent shape functions. Their method was referred to as the continuum approach and has been applied in numerous applications of hyper-redundant manipulators.

Zanganeh and Angeles (1995) argued that “the continuum approach requires some intuition in the choice of the modes and is not a straightforward task.” An _extrinsic scheme_ was preferred and it was suggested to use splines for parameterization of the backbone curves. Regarding the reference shape curve method suggested by Naccarato and Hughes (1991), Zanganeh and Angeles suggested that Bezier curves and B-splines do not provide a direct means for incorporating the prescribed orientation data of the end-effector in the curve. Instead, they proposed a parameterization of the backbone curve using cubic and quintic splines. One of the advantages of this method is that the user can directly incorporate the end-effector pose constraints into parametric equations of the backbone curve.

Zanganeh, Lee and Hughes (1997) used a series of 3-DOF _extensible gimbal modules_ (also referred to as “equivalent serial mechanism”) to represent a backbone curve. The method was referred to as a discrete model approach and applied to Trussarm. Williams and Mayhew (1997a) applied a similar approach to another VGT manipulator and the method was referred to as the virtual serial manipulator approach.

In general, configuration control is used because of computation efficiency. The procedure is also often well-suited for parallel processing. However, configuration control does not permit optimization of other performance criteria such as joint limit avoidance and singularity avoidance [Williams and

1.4.2 Design of and Experiments with VGT Manipulators

While many papers have been written on the design and kinematical analysis of VGT manipulators, only a handful of VGT manipulators have been actually constructed and experimentally analyzed. To date, VGT manipulators that have been constructed are:

- 3-DOF Double-octahedral VGT at NASA Langley Research Center (LRC) [Williams, 1994]  

- Two-bay, Octahedral-octahedral VGT at NASA LRC [Robertshaw and Reinholtz, 1988]

- Tapered Two-bay, Octahedral-octahedral VGT at NASA LRC, 2 [Williams and Mayhew, 1997a]

- Trussarm Mark I [Hughes, et al., 1991]

- Serpentine Truss Manipulator (STM), a 16-DOF tetrahedral VGT, at NASA Kennedy Space Center [Williams and Mayhew, 1997b]

1Some figures shown in this chapter have been collected from the world wide web. Copyrights of the figures are reserved by the authors of the web pages.

2In collaboration with Department of Energy, Battelle, Pacific Northwest Laboratory, and Oak Ridge National Laboratory.
The 3-DOF VGT at NASA (Figure 1.3) and the Trussarm Mark I (Figure 1.4) are both 3-DOF VGT manipulators. In fact, they may be utilized as building blocks for a larger and possibly redundant VGT manipulator. The 3-DOF NASA VGT bears a strong resemblance to the Trussarm Mark I. In his study, Williams (1994) modeled a 3-DOF VGT as an extensible gimbal, which later extended to the development of the Virtual Serial Manipulator Approach to the two-bay VGT kinematics. Here, the term 'bay' refers to a module. This is similar to the discrete model approach to Trussarm kinematics discussed in
Figure 1.5: Tapered Two-bay Octahedral-octahedral VGT at NASA LRC

Figure 1.6: Serpentine Truss Manipulator, NASA Kennedy Space Center
Figure 1.7: Planar VGT, Caltech

Figure 1.8: Tetrobot
Chapter 3.

The 3-DOF NASA VGT, two-bay VGT (figure not available) and the tapered two-bay VGT (Figure 1.5) seem to belong to the same family of manipulators developed by LRC, which has been involved with numerous experiments related to the analysis of VGT manipulators.

Robertshaw and Reinholtz’s two-bay VGT is one of the first VGT manipulators developed by LRC to demonstrate deployment concepts. This spatial VGT is a two-bay, statically determinate, octahedral-octahedral structure with three actuators, providing 3 degrees of freedom. In total, the structure is nearly 4.3 meters in length. Robertshaw and Reinholtz (1988) demonstrated the accuracy of this long manipulator by drawing on a surface with a pen at the end of the manipulator.

The tapered two-bay VGT is a part of the Selective Equipment Removal System (SERS). SERS was designed for removing radioactive equipment and nuclear materials from nuclear waste sites. One of the subsystems of SERS is an 8-DOF deployment manipulator, which is a tapered two-bay VGT mounted onto a 2-DOF serial joint. The dimensions of the first VGT module are identical to the 3-DOF VGT in Figure 1.3. The second module is smaller in size.

A 30-DOF planar VGT (Figure 1.7) was designed and constructed at the California Institute of Technology (Caltech) to validate their analytical work in kinematics and dynamics of hyper-redundant manipulators. Chirikjian and Burdick (1993) wrote that the VGT was chosen among several morphologies of hyper-redundant manipulators for “practical implementation” and the planar manipulator instead of spatial manipulator “for simplicity’s sake”. This planar VGT manipulator consists of 10 identical modules. Each module is, in turn, a 3-DOF planar parallel manipulator. The manipulator can contract to a minimum length of 3.66 meters and extend to a maximum length of 5.49
The manipulator weighs less than 55 kilograms. The 30-DOF planar VGT manipulator has the advantage of being modular and extremely easy to assemble and disassemble; however, it shares with Trussarm an accuracy problem of end-effector positioning. Furthermore, since it is a planar manipulator, it would have little practicality for most industrial applications.

'Serpentine' is simply another term used to describe a hyper-redundant manipulator. 'Snake,' 'tentacle,' 'highly articulated' and 'highly redundant' also denote the same family of manipulators. The serpentine manipulator (Figure 1.6) described by Williams and Mayhew (1997a) is a 16-DOF tetrahedral VGT manipulator developed at the NASA Kennedy Space Center. This manipulator is a subsystem of the Payload Inspection and Processing System (Figure 1.9).

The kinematics modeling and configuration control algorithm, referred to as the follow-the-leader algorithm, has been developed and applied to this manipulator. Based on the experience learned from this exercise, several recommendations were made by Williams and Mayhew including reducing the actuator noise, and ensuring that the control system allows all motors to reach
each commanded set in the same time interval.

The TETROBOT (Figure 1.8) developed at Rensselaer is a modular and hyper-redundant VGT mechanism. The TETROBOT family of robots use a joint mechanism called Concentric Multilink Spherical Joint, that supports modularity in not only the hardware configuration but also the software architecture. The system is reconfigurable with different applications. Three examples given by Hamlin and Sanderson (1997) are the double-octahedral and tetrahedral VGT manipulators and the six-legged walker. The double-octahedral TETROBOT has the same configuration as the Trussarm Mark I and the 3-DOF NASA VGT. The main difference is that 12 of the 21 links in the double-octahedral TETROBOT are actuators. The manipulator is 1.5 meters in length, and can move vertically almost 1 meter. The payload is 80 kilograms and the peak end-effector speed is 6 cm/s. The tetrahedral manipulator TETROBOT is composed of 4 modules with all members actuated. The tip velocity is approximately 8 cm/s and the theoretical payload is 40 kg.

1.4.3 Kinematics Analysis of Trussarm

Two approaches have been taken to the kinematics problem of Trussarm: the link-length method and the petal angle method.

The petal angle method was used by Hughes and Naccarato (1991), Salmen (1993), and Naccarato (1994). In all three cases, a closed-form solution was achieved for the inverse kinematics, but not for the forward kinematics.

Oikawa (1995) briefly considered the differential inverse kinematics of a Trussarm module using the petal angle method. He demonstrated how complicated the solution of the differential kinematics problem is.

The link-length method was examined by Hughes, Sincarsin and Carroll
(1991), and Naccarato (1994). Instead of the petal angles, the link-length method uses the coordinates of the node points as variables.

The differential kinematics based on the link-length method was also studied by Naccarato (1994) and Oikawa (1995). The input to the differential inverse kinematics problem consists of the coordinates of six moving nodes of a Trussarm module, resulting in 18 independent coordinates. Accordingly, the size of the system Jacobian is $18 \times 18$. This requires over 600 times more computation than the $3 \times 3$ matrix in the petal-angle method. Thus, the petal-angle method is computationally more efficient than link-length methods.

1.5 Overview of this Document

The system description of Trussarm is presented in Chapter 2. Components of Trussarm hardware are divided according to their functions into three groups: servomechanism, supporting structure, and control system. The physical layout of each component is described and their functions summarized. Several technologies related to Trussarm geometry and kinematics are introduced. For instance, the definition of Trussarm joint coordinates is stated. Furthermore, the difference between joint coordinates and motor coordinates are given. The frame of reference used in the following chapters is also defined in this chapter in terms of the physical components of a Trussarm module.

In Chapter 3, the kinematics model of Trussarm is derived; a modified discrete model approach is used. A configuration optimization technique is applied to resolve the redundancy in kinematics. The forward and inverse kinematics of each module are solved in terms of joint variables ($\theta$, $\phi$ and $l$) related to the discrete model. The joint limit avoidance problem is also briefly considered.
Chapter 4 outlines the formulation of the differential kinematics of Trussarm. A Jacobian matrix and Hessian are derived using a discrete model. For \( n \) modules (a \( 3n \)-DOF system) the system Jacobian matrix is \( 3n \times 3n \). Elements of this matrix are expressed in terms of \( (\theta, \phi, l) \). The Jacobian matrix of the Trussarm module is also solved.

In Chapter 5, engineering performance data are discussed. Experiments designed to measure the workspace, dexterity, repeatability, speed, and accuracy are described. Through the series of simulation tests, these measurements are predicted, then the results are verified with experiments.

Chapter 6 summarizes a kinematic control system design. The proposed control scheme is to apply coarse control to bring the end-effector close to the target posture, then to apply fine position control based on the sensor information. Simulation results and experiments prove that sufficiently high accuracy in positioning can be achieved by using this proposed control method.

The conclusions are drawn and suggestions for future Trussarm research are made in Chapter 7.
Chapter 2

System Description

In general, a manipulator possesses three attributes: actuators, supporting structures, and some intelligence to control the actuators. Some applications require workspace feedback sensors to achieve the higher performance. In the current configuration, Trussarm possesses only the joint level feedback. This chapter describes the three attributes of Trussarm.

2.1 Actuators and Servomechanisms

2.1.1 Trussarm Actuator

The Trussarm actuators are the most basic substructures of the Trussarm control system. They are independent linear actuators, each driven by a DC motor and a gear reduction assembly (pulleys and a drive belt). The motion of each motor is monitored and controlled by an optical encoder and servomechanism, respectively. The state of each actuator, i.e., the distance that the lead screw
has traveled from the zero position, is related to the encoder counts \((M)\) as

\[
M = 4M_r G L
\]  

(2.1)

where \(M_r\) is the number of encoder counts per motor revolution, \(G\) is the gear ratio, and \(L\) the lead screw travel ratio. The factor 4 is included because the Trussarm encoders operate in the quadrature mode. In the current configuration, the ratios are 2,771,638, 1,422,878, 1,767,652, and 2,660,782 for module 1, 2, 3 and 4, respectively. The specification of the actuators is given in Table 2.1.

The transmission ratio, calculated in Table 5.1, is utilized to transfer joint coordinates (actuator lengths) to motor coordinates (encoder counts). For example, the maximum extension of a Trussarm actuator is 0.0762 meters. For actuator 1, the maximum extension in encoder counts is \(0.0762 \times 2,771,648 = 211,199\) counts. For actuator 12, the same length equals \(0.0762 \times 1,351,734 = 103,002\) counts.

<table>
<thead>
<tr>
<th>Motors</th>
<th>Transmission Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ASTRO</td>
<td>360</td>
</tr>
<tr>
<td>2 Clifton</td>
<td>192</td>
</tr>
<tr>
<td>3 Clifton</td>
<td>192</td>
</tr>
<tr>
<td>4 Clifton</td>
<td>192</td>
</tr>
</tbody>
</table>

Table 2.1: Trussarm actuator specifications
Table 2.2: Trussarm servomechanism PID gains

<table>
<thead>
<tr>
<th></th>
<th>Actuator 1</th>
<th>Actuator 2</th>
<th>Actuator 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAY1</td>
<td>$K_D$</td>
<td>246.15</td>
<td>267.54</td>
</tr>
<tr>
<td></td>
<td>$K_P$</td>
<td>65.02</td>
<td>50.01</td>
</tr>
<tr>
<td></td>
<td>$K_I$</td>
<td>8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>BAY2</td>
<td>$K_D$</td>
<td>140.04</td>
<td>140.03</td>
</tr>
<tr>
<td></td>
<td>$K_P$</td>
<td>27.76</td>
<td>27.76</td>
</tr>
<tr>
<td></td>
<td>$K_I$</td>
<td>8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>BAY3</td>
<td>$K_D$</td>
<td>299.97</td>
<td>304.21</td>
</tr>
<tr>
<td></td>
<td>$K_P$</td>
<td>40.02</td>
<td>50.37</td>
</tr>
<tr>
<td></td>
<td>$K_I$</td>
<td>8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>BAY4</td>
<td>$K_D$</td>
<td>277.66</td>
<td>277.66</td>
</tr>
<tr>
<td></td>
<td>$K_P$</td>
<td>32.12</td>
<td>40.26</td>
</tr>
<tr>
<td></td>
<td>$K_I$</td>
<td>8.0</td>
<td>8.0</td>
</tr>
</tbody>
</table>

2.1.2 Servomechanism

The servomechanism of each actuator is controlled by GALIL motion controllers. Two DMC1080 controllers have been installed on the control computer to provide PID control. DMC1080’s operate in the bus structure.

Each DMC1080 is capable of providing control up to eight axes. For Trussarm, only six axes are servoed by each DMC1080. With two DMC1080’s, the control system provides servomechanisms to all twelve axes.

The performance of the servomechanism has been adjusted using the software, GALIL Motion Control - SDK100, provided by GALIL communication Inc. The gains selected for Trussarm actuators are shown in Table 2.2 and are based on the step response of each motor.
2.2 Supporting Structure

2.2.1 Modules

Each Trussarm module is a double-octahedral parallel manipulator and possesses three actuators of the linear electric lead-screw type. The lead screw assembly acts as a bearing that converts rotary motion to linear motion. Each actuator is fastened at one end directly to the hinge assembly that connects the actuator to the other end of the next actuator. As a result, three actuators form a triangular actuator plane (see Figure 2.2). When fully extended each actuator measures 46.4 cm from the to center of the hinge mechanism. The length of a fully retracted actuator is 31.2 cm. Thus, Trussarm actuators have a 149 % extension ratio.

2.2.2 Passive Members

Each corner of the actuator plane is connected to a pin-joined hinge, which in turn is connected to a 120-degree aluminum piece. From the 120-degree connector, two aluminum passive members are connected 60 degrees apart.
See Figure 2.3. This assembly provides the rotational degrees of freedom needed at the juncture between the passive structure and the actuator plane. The aluminum passive members make up six lateral triangles. In the current configuration, each member is made of 0.95 cm aluminum tubing.

At the base of the lateral triangles, three passive members, made of 0.64 cm drill rod, form a base or top triangular plane. A top plane of one module and a base plane of the next module are connected with a passive connector to connect contiguous modules.

### 2.2.3 Trussarm Coordinate System

Consider the base plane of module 1, shown in Figure 2.1. The three corners of the plane are denoted \( b_{01} \), \( b_{02} \), \( b_{03} \), in clockwise direction. The geometric center of the base triangle, \( O_0 \), is denoted the center of the base frame, \( F_0 \). Moreover, we define the \( x \)-axis of \( F_0 \) as a unit vector through \( b_{03} \), and the \( z \)-axis as the vector out of the base plane. The \( y \)-axis is defined as the vector orthonormal to the \( x \) and \( z \)-axes. The actuator above \( b_{01} \) is denoted actuator 1 and actuators 2 and 3 are numbered in clockwise direction.

The geometric center of the top plane, \( O_1 \), is the center of \( F_1 \). The
position of $O_1$ expressed in $F_0$ and the orientation of $F_1$ with respect to $F_0$ describe the configuration of module 1. $F_1$ also coincides with the base frame for module 2. Therefore, for $n$ modules, there are $n + 1$ frames in total.

In the actual manipulator, the base plane of module $i + 1$ and the top plane of module $i$ are separated by a small offset in the passive connector. To simplify the kinematics, the $F_i$ are defined in the mid-plane between these two physical planes.

### 2.3 Trussarm Control Environment (TCE)

The purpose of the control unit is to servo all the actuators simultaneously in such a way that the manipulator performs the desired task. One of the simplest tasks proposed for Trussarm is trajectory tracking, such as tracing a prescribed end-effector trajectory.

The control architecture is shown in Figure 2.4. The main unit for the Trussarm control system is the Trussarm Control Environment (TCE). TCE was designed in Microsoft Visual C++, based on the DMC communication.
software provided by GALIL communication Inc.

In addition to the basic functionality required for simple joint control, TCE is capable of number of functions.

2.3.1 Trajectory Planning

In order to demonstrate the kinematic control, a simple path generation function has been added to TCE. A linear, circular, or rectangular path may be generated from the path generation menu. Based on geometric information, such as the circular path, the corresponding position and orientation of the end-effector is computed in three dimensional space.

2.3.2 Hyper-Redundancy Resolution

Given the Cartesian coordinates (position and/or orientation) of the end-effector, the configuration of each module can be computed using an equivalent discrete model approach to hyper-redundancy resolution.

2.3.3 Kinematics

TCE is capable of solving both inverse and forward kinematics of Trussarm modules. Once the configurations of all modules are obtained from the hyper-redundancy resolution process, the information is then used to solve the inverse kinematics, which yields the desired actuator lengths.

2.3.4 GALIL commands

For a given end-effector path, two DMC files are prepared by TCE. Each file contains a trajectory for six actuators (in motor counts). The desired
actuator lengths, or joint coordinates, are converted into motor counts. At the beginning of each DMC file, the servo speed is adjusted.

2.3.5 Motion Execution

The TCE is designed to communicate with 1 or 2 DMC 1080 boards. For motion control with 6 or fewer actuators, only one controller may be used. For motion control with 7 or more actuators, both boards need to be selected. For the kinematic control, two DMC files must be externally downloaded to DMC1080 boards. Then, from the motion menu, the programs on two boards are executed simultaneously.

2.3.6 Cartesian Control

Using the terminal mode, the current Cartesian position of the end-effector may be monitored and commanded.
Chapter 3

Kinematics

The motion of a manipulator is often specified by the position and/or orientation of its end-effector. If the manipulator is redundant, some functional specification is additionally required to resolve the redundancy. Hamlin and Sanderson (1997) defined this problem as the kinematic control problem. In their paper, the term “kinematic control” was used to describe "the static positioning of a manipulator, moving slowly through a set of predefined trajectory points." In kinematic control of a hyper-redundant manipulator, three tasks are involved: redundancy resolution, manipulator kinematics, and control. In this chapter, first two topics in kinematic control are considered. The control methods are considered in Chapter 6.

A configuration optimization is an approach to the redundancy resolution problem. A proposed scheme for configuration optimization by discrete model approach, first developed by Zanganeh, the author and Hughes (1997), is presented in Section 3.1. In Section 3.2, the forward and inverse kinematics of a Trussarm module are examined. In Section 3.3, another issue in kinematic control, joint limit avoidance, is considered.
3.1 Kinematics of an Equivalent Discrete Model

3.1.1 3-DOF Extensible Gimbal Module

A discrete model that approximates the backbone curve in a piecewise-linear fashion was introduced by Zanganeh, Lee and Hughes (1997). A discrete model of a hyper-redundant manipulator is obtained by a concatenation of 3-DOF extensible gimbal modules. Each module has, in turn, three springs that provide compliance for the three motion freedoms.

The geometry and kinematics of a gimbal module is presented by three independent variables, $\theta$, $\phi$, and $l$. As shown in Figure 3.1(i), a gimbal module consists of a universal joint with two rotational DOF and one prismatic joint. This model is modified to take advantage of the geometric symmetry (about the actuator plane) of the module. Instead of one prismatic joint, two coupled prismatic joints are used (Figure 3.1(ii)). The local frame $\mathcal{F}_i$ is assigned to the $i$th gimbal module such that its center is at the base of the first prismatic joint and the $x$- and $y$-axes are parallel to the rotation axes of the universal joint. The orientation of the $i$th frame, $\mathcal{F}_i$, with respect to $\mathcal{F}_{i-1}$, is described
thus:

\[ Q_i = \begin{bmatrix} \cos \phi_i & 0 & \sin \phi_i \\ \sin \theta_i \sin \phi_i & \cos \theta_i & -\sin \theta_i \cos \phi_i \\ -\cos \theta_i \sin \phi_i & \sin \theta_i & \cos \theta_i \cos \phi_i \end{bmatrix} \]  

(3.1)

Also the position of the origin of \( F_i \), expressed in \( F_{i-1} \), is

\[ p_i = (I + Q_i) \begin{bmatrix} 0 \\ 0 \\ l_i \end{bmatrix} \]  

(3.2)

where \( I \) is the \( 3 \times 3 \) identity matrix. The kinematics of an extensible gimbal modules was also examined by Williams (1994).
3.1.2 Chain of Gimbal Modules

Let us now consider a chain of \( n \)-gimbal modules, referred to as the equivalent discrete model. Expressed in the base frame, \( \mathcal{F}_0 \), the position and orientation of the end effector (or the tip of the manipulator) are

\[
g = \sum_{i=1}^{n} p_i = \bar{l}_1 + Q_1 \bar{l}_2 + Q_1 Q_2 \bar{l}_3 + \cdots + Q_1 Q_2 \cdots Q_n \bar{l}_{n+1} + Q_1 Q_2 \cdots Q_n t
\]

\[
h = E \hat{n}_n
\]

where \( \bar{t} \) and \( Q_t \) represent position and orientation of the end effector with respect to \( \mathcal{F}_n \), and also,

\[
\bar{l}_i = \begin{bmatrix} 0 \\ 0 \\ l_{i-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ l_i \end{bmatrix} \quad i = 1, \ldots, n + 1
\]

where \( l_0 = l_{n+1} = 0 \). Also, \( E \) is a selection matrix and \( \hat{k} \) is a unit vector,

\[
E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \hat{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\]

The inverse kinematics of the equivalent discrete model is much more complicated than the forward kinematics. This problem is often referred to as the redundancy resolution problem, since the number of unknown variables is much greater than the number of constraints. The redundancy resolution methods are often at the velocity level. They are based on the manipulator Jacobian pseudo-inverse. The main goal is to minimize a cost function, \( Z(\dot{q}) \), and to solve for the joint variables, \( q \).
If the manipulator possesses a large number of degrees of redundancy, i.e., if the manipulator is a hyper-redundant manipulator, its kinematics can not be easily formulated using an inverse Jacobian since it is difficult to build a mathematical model. An alternative technique to resolve hyper-redundancy is configuration control.

The method proposed by Zanganeh, Lee and Hughes (1997) is to minimize the cost function, $Z$, of the system, given as

$$Z = \sum_{i=1}^{n} K_{\theta_i} \theta_i^2 + \sum_{i=1}^{n} K_{\phi_i} \phi_i^2 + \sum_{i=1}^{n} K_{l_i} (l_i - \bar{l}_i)^2$$  \hspace{1cm} (3.7)$$

while satisfying the kinematic constraints:

$$g(\theta, \phi, l) = x_{ee}$$  \hspace{1cm} (3.8)$$
$$h(\theta, \phi, l) = E \hat{n}_{ee}$$  \hspace{1cm} (3.9)$$

where $x_{ee}$ and $\hat{n}_{ee}$ are the desired position and orientation of the end-effector.

The Lagrangian approach is adopted to find the solution set, $q$ where

$$q \equiv \text{col}\{q_1, \cdots, q_n\}$$  \hspace{1cm} (3.10)$$

and

$$q_i \equiv \text{col}\{\theta_i, \phi_i, l_i\} \hspace{1cm} i = 1, \cdots, n$$  \hspace{1cm} (3.11)$$

If we define $w(q)$ and $v$ as

$$w(q) = \begin{bmatrix} g(q) \\ h(q) \end{bmatrix} \hspace{1cm} v = \begin{bmatrix} x_{ee} \\ E \hat{n}_{ee} \end{bmatrix}$$  \hspace{1cm} (3.12)$$

then the Lagrangian is

$$L(q) = \lambda^T (w - v) + Z$$  \hspace{1cm} (3.13)$$
and thus, the normality conditions are

\[
\begin{align*}
\frac{\partial L}{\partial \lambda} & \equiv (w - v)^T = 0_{1 \times m} & (3.14) \\
\frac{\partial L}{\partial q} & \equiv \lambda^T \partial w + \frac{\partial Z}{\partial q} = 0_{1 \times p} & (3.15)
\end{align*}
\]

where \( \lambda \) is the vector of Lagrangian multipliers. For \( n \) gimbal modules (\( p = 3n \)-DOF system) with \( m \) kinematics constraints (\( m = \dim(g) + \dim(h) \)), eqs. (3.14) and (3.15) lead to \( p + m \) nonlinear equations in terms of \( q \) and \( \lambda \).

The \( w \) term in eq. (3.14) has already been derived in eqs. (3.3) and (3.4). Now let us consider the \( \frac{\partial w}{\partial q} \) term in eq. (3.15). From eq. (3.7), we have

\[
\begin{align*}
\frac{\partial Z}{\partial \theta_i} & = K_{\theta} \theta_i & (3.16) \\
\frac{\partial Z}{\partial \phi_i} & = K_{\phi} \phi_i & (3.17) \\
\frac{\partial Z}{\partial l_i} & = K_{l}(l_i - \bar{l}_i) & (3.18)
\end{align*}
\]

Next, consider \( \frac{\partial w}{\partial q} \) term. From eqs. (3.3) and (3.4), the following equations may be derived:

\[
w = \begin{bmatrix} \bar{p}_1 \\ \bar{q}_1 \end{bmatrix} \quad (3.19)
\]

where

\[
\begin{align*}
\bar{p}_i & = \bar{l}_i + Q_i \bar{p}_{i+1} & (3.20) \\
\bar{q}_i & = Q_i \bar{q}_{i+1} & (3.21)
\end{align*}
\]

for \( i = 1, \ldots, n \) with \( \bar{p}_{n+1} = \bar{l}_{n+1} + t \) and \( \bar{q}_{n+1} = Q_i \bar{k} \). By definition, \( \frac{\partial w}{\partial q} \) is a \( m \times p \) matrix:

\[
\frac{\partial w}{\partial q} = \begin{bmatrix}
\frac{\partial q_1}{\partial \theta_1} & \frac{\partial q_1}{\partial \phi_1} & \ldots & \frac{\partial q_1}{\partial \theta_m} \\
\frac{\partial q_2}{\partial \theta_1} & \frac{\partial q_2}{\partial \phi_1} & \ldots & \frac{\partial q_2}{\partial \theta_m} \\
& \ddots & \ddots & \ddots \\
& & \frac{\partial q_p}{\partial \theta_1} & \frac{\partial q_p}{\partial \phi_1} & \ldots & \frac{\partial q_p}{\partial \theta_m}
\end{bmatrix} \quad (3.22)
\]
for \( i = 1, \ldots, n \). For each \( i \), the functions can be differentiated with respect to \( q_i \) as follows:

\[
\frac{\partial g}{\partial \theta_i} = \tilde{Q}_{i-1} \frac{\partial Q_i}{\partial \theta_i} \tilde{p}_{i+1} \\
\frac{\partial g}{\partial \phi_i} = \tilde{Q}_{i-1} \frac{\partial Q_i}{\partial \phi_i} \tilde{p}_{i+1} \\
\frac{\partial g}{\partial l_i} = \tilde{Q}_{i-1}(I + Q_i)k
\]

(3.23) \hspace{1cm} (3.24) \hspace{1cm} (3.25)

where \( \tilde{Q}_i = Q_1 \ldots Q_i \) and \( \tilde{Q}_0 = I \).

\[
\frac{\partial h}{\partial \theta_i} = E\tilde{Q}_{i-1} \frac{\partial Q_i}{\partial \theta_i} \tilde{q}_{i+1} \\
\frac{\partial h}{\partial \phi_i} = E\tilde{Q}_{i-1} \frac{\partial Q_i}{\partial \phi_i} \tilde{q}_{i+1} \\
\frac{\partial h}{\partial l_i} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T
\]

(3.26) \hspace{1cm} (3.27) \hspace{1cm} (3.28)

Equations (3.14) and (3.15) may be solved using any numerical method for nonlinear systems of equations. Examples include the multidimensional secant methods such as Newton's method, quasi-Newton method, and the Broyden's method. In the controller design, Broyden's method was employed. Chapter 9 of Press, et al. (1992) may be referred to for more details.

### 3.2 Kinematics of a Trussarm Module

#### 3.2.1 Inverse Kinematics

In this section, the inverse kinematics problem for a Trussarm module is considered. The objective is to solve for the actuator lengths of the \( j \)th module, \( k_j \), when the corresponding gimbal module's joint variable \( q_j \) are specified. Since only one Trussarm module is considered at a time, the subscript \( j \) will be omitted in this and the following section.
Consider the geometry of a module shown in Figure 3.3. It can be observed that the vector \( \mathbf{d} \) is the sum of \( l_1 \) and \( l_2 \), i.e.,

\[
\mathbf{d} = l_1 + l_2 = l(\mathbf{n}_0 + \mathbf{n}_2) = d\mathbf{n}_1
\]  
(3.29)

where \( \mathbf{n}_1 \) is the unit normal of the actuator plane. The unit direction vector of the top triangle, \( \mathbf{n}_2 \), may be computed using \( \theta \) and \( \phi \):

\[
\mathbf{n}_2 = \mathbf{R}_{02} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\]  
(3.30)

where

\[
\mathbf{R}_{02} = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ \sin \theta \sin \phi & \cos \theta & -\sin \theta \cos \phi \\ -\cos \theta \sin \phi & \sin \theta & \cos \theta \cos \phi \end{bmatrix}
\]  
(3.31)

Thus,

\[
\mathbf{n}_2 = \begin{bmatrix} \sin \phi \\ -\sin \theta \cos \phi \\ \cos \theta \cos \phi \end{bmatrix}
\]  
(3.32)

\( \mathbf{n}_1 \) is then calculated from \( \mathbf{n}_2 \):

\[
\mathbf{n}_1 = \frac{\mathbf{n}_0 + \mathbf{n}_2}{||\mathbf{n}_0 + \mathbf{n}_2||}
\]  
(3.33)

By definition, \( \mathbf{n}_0 \equiv \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \) since all the vectors are defined in \( \mathcal{F}_{j-1} \), whose \( z \)-axis is parallel to \( \mathbf{n}_0 \). Hence,

\[
||\mathbf{n}_0 + \mathbf{n}_2|| = \sqrt{\sin^2 \phi + \sin^2 \theta \cos^2 \phi + (\cos \theta \cos \phi + 1)^2}
\]  
(3.34)

\[
= \sqrt{\sin^2 \phi + (\sin^2 \theta + \cos^2 \theta) \cos^2 \phi + 2 \cos \theta \cos \phi + 1}
\]  
(3.35)

\[
= \sqrt{1 + 2 \cos \theta \cos \phi + 1}
\]  
(3.36)

\[
= \sqrt{2(1 + \cos \theta \cos \phi)}
\]  
(3.37)
Let us define this scalar as $\alpha$. Then, $\hat{\mathbf{n}}_1$ is

$$
\hat{\mathbf{n}}_1 = \frac{1}{\alpha} \begin{bmatrix}
\sin \phi \\
-\sin \theta \cos \phi \\
\cos \theta \cos \phi + 1
\end{bmatrix}
$$

(3.38)

Since $\mathbf{d}$ is perpendicular to the actuator plane, $\mathbf{d}$ is perpendicular to any vector lying on this plane. Using this relation, we can calculate the petal angles, $\gamma_1$, $\gamma_2$, and $\gamma_3$. From Figure 3.3, it is shown that $\delta_2$ (the offset of the actuator plane from the plane formed by petal nodes, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$) lies in the direction of $\hat{\mathbf{n}}_1$. Hence, for $i = 1, 2, 3$, we have

$$
[\mathbf{r}_i - (l\hat{\mathbf{n}}_0 - \delta_2\hat{\mathbf{n}}_1)]^T \mathbf{d} = 0
$$

(3.39)

$$
\mathbf{r}_i = (c_1 + c_2 \cos \gamma_i) \mathbf{e}_i + (\delta_1 + c_2 \sin \gamma_i) \hat{\mathbf{n}}_0
$$

(3.40)

where $\mathbf{e}_i$ is the $i$th column of the matrix $\mathbf{e}$ that represent the the base plane, given as

$$
\mathbf{e} = \begin{bmatrix}
\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -1 \\
\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\
0 & 0 & 0
\end{bmatrix}
$$

(3.41)

Also,

$$
c_2^2 = l_{02}^2 - l_{01}^2/4
$$

(3.42)

$$
c_1 = \frac{l_{01}}{2\sqrt{3}}
$$

(3.43)

$l_{02}$ is the side length of an isometric petal triangle and $l_{01}$ is the side length of...
the equilateral base or top triangles. For Trussarm II:

\[ l_{01} = 29.14 \text{ cm} \] (3.44)

\[ l_{02} = 27.33 \text{ cm} \] (3.45)

\[ \delta_1 = 0.95 \text{ cm} \] (3.46)

\[ \delta_2 = 4.83 \text{ cm} \] (3.47)

Refer to Oikawa (1995) and Lee (1995) for more details of Trussarm geometric constants.

Substituting eq.(3.40) into eq.(3.39), we obtain the following equation for the petal angles:

\[
[(c_1 + c_2 \cos \gamma_i)e_i + (\delta_1 + c_2 \sin \gamma_i)n_0 - (l_0 - \delta_2 n_0)]^T d = 0
\] (3.48)

Eq.(3.48) may be simplified:

\[ a_i \cos \gamma_i + b \sin \gamma_i = f_i \] (3.49)

where

\[ a_i = e_i^T (\hat{n}_0 + \hat{n}_2)l \] (3.50)

\[ b = n_0^T (\hat{n}_0 + \hat{n}_2)l \] (3.51)

\[ f_i = \frac{1}{c_2} [-c_2a_i + (l - \delta_1)b - \delta_2 n_0 (\hat{n}_0 + \hat{n}_2)] \] (3.52)

Thus the petal angles are

\[ \gamma_i = \tan^{-1} \left( \frac{b}{a_i} \right) - \tan^{-1} \left( \frac{\sqrt{a_i^2 + b^2} - f_i}{f_i} \right) \] (3.53)

Once the petal angles are calculated, nodal coordinates are found from eq.(3.40).

Lastly, the actuator lengths are computed from the nodal coordinates:

\[ k_1 = ||r_2 - r_3|| \] (3.54)

\[ k_2 = ||r_3 - r_1|| \] (3.55)

\[ k_3 = ||r_1 - r_2|| \] (3.56)
The remaining nodes on the base triangle and the top triangles are determined by simple trigonometric relations. The base nodes are

\[ b_{01} = l_{01} \begin{bmatrix} -\frac{1}{2\sqrt{3}} \\ -\frac{1}{2} \\ 0 \end{bmatrix}, \quad b_{02} = l_{01} \begin{bmatrix} -\frac{1}{2\sqrt{3}} \\ \frac{1}{2} \\ 0 \end{bmatrix}, \quad b_{03} = l_{01} \begin{bmatrix} \frac{1}{\sqrt{3}} \\ 0 \\ 0 \end{bmatrix} \]

and, for \( i = 1, 2, 3, \)

\[ b_i = b_{0i} + \delta_1 \hat{n}_0 \]

(3.58)

The top nodes for \( i = 1, 2, 3, \) are computed using symmetric properties, i.e.,

\[ t_{0i} = R_{02} b_{0i} + d \]

(3.59)

\[ t_i = t_{0i} - \delta_1 \hat{n}_2 \]

(3.60)

and

\[ r_{3+i} = r_i + \delta_2 \hat{n}_1 \]

(3.61)

\[ r_{6+i} = r_i + \delta_2 \hat{n}_1 \]

(3.62)

### 3.2.2 Forward Kinematics

The forward kinematics problem deals with solving for end-effector position when the actuator lengths are given. In this section the solution for \( q = (\theta, \phi, l) \) are considered when \( k = (k_1, k_2, k_3) \) is given.
Upon substituting eq.(3.40) into eqs. (3.54), (3.55), and (3.56), we get

\[
\begin{align*}
\kappa_1^2 &= \frac{1}{4}(3c_1 + c_2 \cos \gamma_2 + 2c_2 \cos \gamma_3)^2 \\
&\quad \quad + \frac{3}{4}(c_1 + c_2 \cos \gamma_2)^2 + c_2^2(\sin \gamma_2 - \sin \gamma_3)^2 \\
\kappa_2^2 &= \frac{1}{4}(3c_1 + c_2 \cos \gamma_1 + 2c_2 \cos \gamma_3)^2 \\
&\quad \quad + \frac{3}{4}(c_1 + c_2 \cos \gamma_1)^2 + c_2^2(\sin \gamma_1 - \sin \gamma_3)^2 \\
\kappa_3^2 &= \frac{1}{4}c_2^2(\cos \gamma_2 - \cos \gamma_1)^2 \\
&\quad \quad + \frac{3}{4}(2c_1 + c_2 \cos \gamma_2 + c_2 \cos \gamma_1)^2 + c_2^2(\sin \gamma_2 - \sin \gamma_1)^2
\end{align*}
\] (3.63) (3.64) (3.65)

These equations can be simplified to

\[
\begin{align*}
\kappa_1^2 &= 2c_2^2(1 - \cos (\gamma_3 - \gamma_2)) + 3(c_1 + c_2 \cos \gamma_3)(c_1 + c_2 \cos \gamma_2) \\
\kappa_2^2 &= 2c_2^2(1 - \cos (\gamma_1 - \gamma_3)) + 3(c_1 + c_2 \cos \gamma_1)(c_1 + c_2 \cos \gamma_3) \\
\kappa_3^2 &= 2c_2^2(1 - \cos (\gamma_2 - \gamma_1)) + 3(c_1 + c_2 \cos \gamma_2)(c_1 + c_2 \cos \gamma_1)
\end{align*}
\] (3.66) (3.67) (3.68)

These nonlinear equations are solved for \(\gamma_1\), \(\gamma_2\), and \(\gamma_3\) numerically. Then,

\[
\hat{n}_1 = \frac{(r_2 - r_3) \times (r_1 - r_2)}{||(r_2 - r_3) \times (r_1 - r_2)||}
\] (3.69)

Moreover, using eq.(3.38), we can write

\[
\hat{n}_{1z} = \frac{1 + \cos \theta \cos \phi}{\alpha} = \frac{\alpha^2/2}{\alpha} = \frac{\alpha}{2}
\] (3.70)

or, \(\alpha = 2\hat{n}_{1z}\). Also, from eq.(3.38), we can derive the following relations for \(\hat{n}_2\):

\[
\hat{n}_2 \equiv \alpha\hat{n}_1 - \hat{n}_0 = \begin{bmatrix} \sin \phi \\ -\sin \theta \cos \phi \\ \cos \theta \cos \phi \end{bmatrix}
\] (3.71)
Therefore, for $0 \leq \theta, \phi \leq \frac{\pi}{2}$

$$\theta = -\tan^{-1}\left(\frac{\dot{n}_{2z}}{\dot{n}_{2s}}\right) \quad (3.72)$$

$$\phi = \tan^{-1}\left(\frac{\dot{n}_{2s}}{\sqrt{\dot{n}_{2s}^2 + \dot{n}_{2z}^2}}\right) \quad (3.73)$$

for $0 \geq \theta, \phi \geq \frac{\pi}{2}$. Given the workspace of Trussarm module, this restriction is reasonable.

Equation (3.48) can be written as follows:

$$[r_2 + \delta_2 \dot{n}_1 - l \dot{n}_0]T[(\dot{n}_0 + \dot{n}_2)] = 0 \quad (3.74)$$

Thus, we can solve the above equation for $l$, i.e.,

$$l = \frac{(r_2 + \delta_2 \dot{n}_1)T(\dot{n}_0 + \dot{n}_2)}{\dot{n}_0^T(\dot{n}_0 + \dot{n}_2)} \quad (3.75)$$

The inverse kinematics solution presented here is based on the petal angle method. By introducing a new set of input variables, $\theta$, $\phi$ and $l$, the requirement to solve the forward kinematics was removed. The algorithm is now in closed form, and also could be computed in parallel fashion. The proposed forward kinematics could also be solved by parallel computation. Unfortunately, the explicit closed-form solution for the forward kinematics problem has not yet been obtained. The solution for the differential forward kinematics is presented for the first time in Chapter 4.

### 3.3 Joint-Limit Avoidance Problems

The redundancy resolution scheme presented in Section 3.1 does not guarantee that the resulting optimal configuration remains within the actuator limits. Hence, the next logical step is to develop a hyper-redundancy resolution scheme that includes the actuator limits in the configuration control.
The existing Trussarm actuators are capable of linear motion between ±0.0762 m from their *baselengths*.\(^1\) Thus, for \(i = 1, \ldots, 3n\),

\[
-0.0762 \leq k_i \leq 0.0762
\]  
\[(3.76)\]

Let us also define \(k_j\), a three dimensional vector whose components are three actuator lengths of the \(j\)th bay. Hence, \(k_j\) is a function of \(\theta_j\), \(\phi_j\) and \(l_j\). Also, we define a new vector \(k\) as \(\text{col}\{k_1, k_2, \ldots, k_{3n}\}\).

The objective of the problem is now to minimize the cost function, \(Z\), while the following kinematic constraints and the following inequality constraints are met:

\[
\mathbf{w} - \mathbf{u} = 0
\]  
\[(3.77)\]

\[
|k_i(q)| - k_{\text{max}} \leq 0 \quad i = 1, \ldots, 3n
\]  
\[(3.78)\]

A nonlinear optimization problem with both equality and inequality constraints is often solved numerically. As shown in an example in Appendix A, it is very difficult to find a computationally efficient algorithm for the problem. For more details on the optimization techniques, the readers can refer to Strang (1996) or Vanderplaats (1984).

Vanderplaats (1984) examined a general approach to the constrained optimization problem called *sequential unconstrained minimization techniques* (SUMT). The classical approach to using SUMT is to create a pseudo-objective function of the form

\[
\Phi(q) = Z(q) + r_\eta P(q)
\]  
\[(3.79)\]

where \(Z(q)\) is the original objective function, \(P(q)\) is an imposed penalty function, and the scalar \(r_\eta\) is a multiplier which determines the magnitude of the

\(^{1}\text{Baselength refers to 0.38 m from one apex of the actuator plane (triangle) to another.}\)
penalty. The subscript \( n \) is the unconstrained minimization number. For more details, see Vanderplaats (1984) and Fiacco and McCormick (1968). Three traditional approaches to SUMT are "the exterior penalty function method," "the interior penalty function method," and "the extended interior penalty function method." The first method is to penalize the objective function only when constraints are violated. It is the easiest method to incorporate into the design algorithm. The second method penalizes the objective function as the design approaches a constraint, and the last method is the combination of the first two.

In Trussarm's kinematics analysis, the exterior penalty function method was applied. The pseudo-objective function for the problem can be defined as

\[
\Phi(q) = Z(q) + r_n \left( \sum_{j=1}^{m} \{w_j - v_j\}^2 + \sum_{j=1}^{3n} \{\max[0, \xi_j(q)]\}^2 \right)
\]

(3.80)

where \( \xi_i = |k_j| - k_{\text{max}} \). The problem is solved in three steps:

1. Minimize \( \Phi(q) \) with initial guess for \( q \), a small value of \( r_n \) and a constant \( \gamma \) (usually 3).

2. If the solution converges — that is, if the solution \( q \) satisfies the kinematic constraint, \( w(q) = v \) — end the procedure.

3. If not, the penalty parameter \( r_n \) is increased to \( \gamma r_n \) and the procedure is repeated.

In order to minimize \( \Phi(q) \), a set of \( 3n \) nonlinear equations are obtained and solved using Broyden's method:

\[
\frac{\partial \Phi}{\partial q} \equiv \frac{\partial Z}{\partial q} + 2r_n \left[ (w - v)^{T} \frac{\partial w}{\partial q} + y^{T} \frac{\partial k}{\partial q} \right] = 0
\]

(3.81)
where

\[ y = \left[ \max[0, \xi_1] \quad \cdots \quad \max[0, \xi_n] \right]^T \]  \hspace{1cm} (3.82)

The solution for \( \frac{\partial y}{\partial q} \) was presented in Section 3.1. To evaluate \( \frac{\partial k}{\partial q} \), let us define \( q_j \) as the joint variables of the \( j \)th gimbal module, \( \theta_j, \phi_j \) and \( l_j \). Then, \( \frac{\partial k}{\partial q} \) could be written as

\[
\frac{\partial k}{\partial q} = \begin{bmatrix}
\frac{\partial k_1}{\partial q_1} & 0 & \cdots & 0 \\
0 & \frac{\partial k_2}{\partial q_2} & \cdots & 0 \\
0 & 0 & \cdots & \frac{\partial k_n}{\partial q_n}
\end{bmatrix} \hspace{1cm} (3.83)
\]

\( \frac{\partial k_j}{\partial q_j} \) is the inverse Jacobian of the \( j \)th Trussarm module and its solution will be discussed in Section 4.3.

The algorithm is relatively easy to program and also reasonably efficient.
Figure 3.3: Vectors in the kinematics of a Trussarm module
Chapter 4

Differential Kinematics

In the previous chapter, it was shown how difficult it is to find an explicit closed-form solution to the kinematics problem of Trussarm. In general, solutions to the kinematic problem require numerical approach. An alternative approach to the kinematic control problem is to examine the differential relationship between the end-effector configuration, \( \mathbf{v} (x_{\infty} \text{ and } \hat{n}_{\infty}) \), and the joint variables (or design variables), \( q \).

Previously, the function \( w(q) \) was defined as

\[
w(q) = v
\]

If the above function is differentiated with respect to time, we get

\[
\left( \frac{\partial w}{\partial q} \right) \dot{q} = \dot{v}
\]

If the Jacobian matrix, \( \frac{\partial w}{\partial q} \), is invertible, i.e., if the manipulator is not redundant and has no kinematic singularities, the joint velocity vector \( \dot{q} \) can be easily obtained by

\[
\dot{q} = \left( \frac{\partial w}{\partial q} \right)^{-1} \dot{v}
\]
However, if the manipulator is redundant, some redundancy resolution method needs to be employed. The objective of this chapter is to find a solution to $\dot{q}$, when $\dot{v}$ is specified.

The control technique described in eq.(4.3) is commonly referred to as resolved motion rate control (RMRC). Schilling (1990) defined RMRC as a "problem to express the actuator velocities as functions of the velocities of the task coordinates."

In Section 4.1, the RMRC of Trussarm is discussed. The Jacobian matrix and Hessian are derived in Section 4.2. In Section 4.3, the differential kinematics of a Trussarm module is presented. The actuator velocity vector, $\dot{k}$, is related to the equivalent discrete model's joint velocity vector, $\dot{q}$, and its inverse function is presented.

### 4.1 Resolved Motion Rate Control

Equation eq.(3.15) may be written as

$$W\lambda + z = 0_{p \times 1}$$

where $W = (\frac{\partial \omega}{\partial q})^T$ and $z = (\frac{\partial \omega}{\partial q})^T$. If we partition $W$ and $z$ thus,

$$W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

where, dim($W_1$)=$m \times m$, dim($W_2$)=$(p - m) \times m$, dim($z_1$)=$m \times 1$, and dim($z_2$)=$(p - m) \times 1$, then we can write

$$W_1\lambda = -z_1 \quad W_2\lambda = -z_2$$

Thus,

$$-W_2(W_1)^{-1}z_1 + z_2 = 0_{(p-m) \times 1}$$

43
Let us define the left-hand-side of the above equation as \( N \). The problem in hand can be written as follows:

\[
\begin{align*}
    \mathbf{w}(q) - \mathbf{v} &= 0_{m \times 1} \\
    \mathbf{N}(q) &= 0_{(p-m) \times 1}
\end{align*}
\]  

(4.8)

Compared to eqs. (3.14) and (3.15), the total number of equations has been decreased from \( m + p \) to \( p \). Upon differentiating eq.(4.8), with respect to time, we obtain

\[
\begin{align*}
    \frac{\partial \mathbf{w}}{\partial \mathbf{q}} \frac{d \mathbf{q}}{dt} &= \frac{d \mathbf{v}}{dt} \\
    \frac{\partial \mathbf{N}}{\partial \mathbf{q}} \frac{d \mathbf{q}}{dt} &= 0_{(p-m) \times 1}
\end{align*}
\]  

(4.9)

or,

\[
\mathbf{W}_e \dot{\mathbf{q}} = \dot{\mathbf{v}}_e
\]  

(4.10)

where equivalent matrix and vector \( \mathbf{W}_e \) and \( \mathbf{v}_e \) are defined as:

\[
\mathbf{W}_e = \begin{bmatrix} \frac{\partial \mathbf{w}}{\partial \mathbf{q}} \\ \frac{\partial \mathbf{N}}{\partial \mathbf{q}} \end{bmatrix} \quad \dot{\mathbf{v}}_e = \begin{bmatrix} \dot{\mathbf{v}} \\ 0_{(p-m) \times 1} \end{bmatrix}
\]  

(4.11)

Thus, provided \( \mathbf{W}_e \) is not singular, we have

\[
\dot{\mathbf{q}} = \mathbf{W}_e^{-1} \dot{\mathbf{v}}_e
\]  

(4.12)

### 4.2 Jacobian Matrix and Hessian

In this section, we concentrate on the solution for \( \mathbf{W}_e \) in eq.(4.11). Note that the analytical solution for \( \frac{\partial \mathbf{w}}{\partial \mathbf{q}} \) is given in Section 3.1.2. The solution for \( \frac{\partial \mathbf{N}}{\partial \mathbf{q}} \) is derived here. By definition,

\[
\frac{\partial \mathbf{N}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial \mathbf{N}}{\partial q_1} & \frac{\partial \mathbf{N}}{\partial q_2} & \cdots & \frac{\partial \mathbf{N}}{\partial q_n} \end{bmatrix}
\]  

(4.13)
From eq.(4.7), we can derive the relation below:

$$\frac{\partial N}{\partial q_j} = -\frac{\partial W_2}{\partial q_j} W_1^{-1} z_1 - W_2 \frac{\partial W_1^{-1}}{\partial q_j} z_1 - W_2 (W_1)^{-1} \frac{\partial z_1}{\partial q_j} + \frac{\partial z_2}{\partial q_j} \tag{4.14}$$

where $\frac{\partial z_1}{\partial q_j}$ and $\frac{\partial W_1}{\partial q_j}$ are the first $m$ rows of $\frac{\partial q_1}{\partial q_j} (\frac{\partial Z}{\partial q})^T$ and $\frac{\partial q_1}{\partial q_j} (\frac{\partial W_2}{\partial q})^T$, respectively, while $\frac{\partial z_2}{\partial q_j}$ and $\frac{\partial W_2}{\partial q_j}$ are the remaining $(p - m)$ rows. From eqs. (3.16), (3.17) and (3.18), the following derivatives may be obtained:

$$\frac{\partial}{\partial q_j} \left( \frac{\partial Z}{\partial q_i} \right) = 0 \tag{4.15}$$

for $i, j \in \{1, \ldots, n, i \neq j\}$, where $q_j$ is one of the joint variables associated with the $j'$th module, $(\theta_j, \phi_j, l_j)$. Moreover,

$$\frac{\partial^2 Z}{\partial \theta_i^2} = K_\theta, \quad \frac{\partial^2 Z}{\partial \phi_i^2} = K_\phi, \quad \frac{\partial^2 Z}{\partial l_i^2} = K_l \tag{4.16}$$

In matrix form,

$$\frac{\partial^2 Z}{\partial q_i^2} = \begin{bmatrix} K_\theta & 0 & 0 \\ 0 & K_\phi & 0 \\ 0 & 0 & K_l \end{bmatrix} \tag{4.17}$$

Also, from eq.(3.22), we can write $\frac{\partial W}{\partial q_j}$ in the form

$$\frac{\partial W}{\partial q_j} = \begin{bmatrix} \frac{\partial}{\partial q_j} (\frac{\partial g}{\partial \theta}) & \frac{\partial}{\partial q_j} (\frac{\partial g}{\partial \phi}) & \cdots & \frac{\partial}{\partial q_j} (\frac{\partial g}{\partial n}) \\ \frac{\partial}{\partial q_j} (\frac{\partial h}{\partial \theta}) & \frac{\partial}{\partial q_j} (\frac{\partial h}{\partial \phi}) & \cdots & \frac{\partial}{\partial q_j} (\frac{\partial h}{\partial m}) \end{bmatrix}^T$$

Equations 3.23 through (3.28) can be used to calculate $\frac{\partial}{\partial q_j} (\frac{\partial g}{\partial \theta})$ and $\frac{\partial}{\partial q_j} (\frac{\partial h}{\partial \phi})$.

Since $Q_i$ is a function of $\theta_i, \phi_i$ and $l_i$ only, $\frac{\partial Q_i}{\partial q_j} = 0$. Furthermore, we define

$$Q_{j+1,i-1} = Q_{j+1} Q_{j+2} \cdots Q_{i-1} \tag{4.19}$$

45
Then, for \( j < i \), differentiate eq.(3.23) with respect to \( \theta_j \) to obtain
\[
\frac{\partial}{\partial \theta_j} \left( \frac{\partial g}{\partial \theta_i} \right) = \tilde{Q}_{j-1} \frac{\partial Q_{ji}}{\partial \theta_j} \tilde{Q}_{j+1,i-1} \frac{\partial Q_{ii}}{\partial \theta_i} \tilde{p}_{i+1} \tag{4.20}
\]

Similarly,
\[
\frac{\partial}{\partial \theta_j} \left( \frac{\partial g}{\partial \phi_i} \right) = \tilde{Q}_{j-1} \frac{\partial Q_{ji}}{\partial \theta_j} \tilde{Q}_{j+1,i-1} \frac{\partial Q_{ii}}{\partial \phi_i} \tilde{q}_{i+1} \tag{4.21}
\]
\[
\frac{\partial}{\partial \theta_j} \left( \frac{\partial g}{\partial l_i} \right) = 0_{3 \times 1} \tag{4.22}
\]

From eq.(3.26), eq.(3.27) and eq.(3.28) we get
\[
\frac{\partial}{\partial \theta_j} \left( \frac{\partial h}{\partial \theta_i} \right) = \tilde{E} \tilde{Q}_{j-1} \frac{\partial Q_{ji}}{\partial \theta_j} \tilde{Q}_{j+1,i-1} \frac{\partial Q_{ii}}{\partial \theta_i} \tilde{q}_{i+1} \tag{4.23}
\]
\[
\frac{\partial}{\partial \theta_j} \left( \frac{\partial h}{\partial \phi_i} \right) = \tilde{E} \tilde{Q}_{j-1} \frac{\partial Q_{ji}}{\partial \theta_j} \tilde{Q}_{j+1,i-1} \frac{\partial Q_{ii}}{\partial \phi_i} \tilde{q}_{i+1} \tag{4.24}
\]
\[
\frac{\partial}{\partial \theta_j} \left( \frac{\partial h}{\partial l_i} \right) = 0_{2 \times 1} \tag{4.25}
\]

The same rule applies to the derivatives with respect to \( \phi_j \) and \( l_j \).
\[
\frac{\partial}{\partial \phi_j} \left( \frac{\partial g}{\partial \theta_i} \right) = \tilde{Q}_{j-1} \frac{\partial Q_{ji}}{\partial \phi_j} \tilde{Q}_{j+1,i-1} \frac{\partial Q_{ii}}{\partial \theta_i} \tilde{p}_{i+1} \tag{4.26}
\]
\[
\frac{\partial}{\partial \phi_j} \left( \frac{\partial g}{\partial \phi_i} \right) = \tilde{Q}_{j-1} \frac{\partial Q_{ji}}{\partial \phi_j} \tilde{Q}_{j+1,i-1} \frac{\partial Q_{ii}}{\partial \phi_i} \tilde{p}_{i+1} \tag{4.27}
\]
\[
\frac{\partial}{\partial \phi_j} \left( \frac{\partial g}{\partial l_i} \right) = 0_{3 \times 1} \tag{4.28}
\]
\[
\frac{\partial}{\partial \phi_j} \left( \frac{\partial h}{\partial \theta_i} \right) = \tilde{E} \tilde{Q}_{j-1} \frac{\partial Q_{ji}}{\partial \phi_j} \tilde{Q}_{j+1,i-1} \frac{\partial Q_{ii}}{\partial \theta_i} \tilde{q}_{i+1} \tag{4.29}
\]
\[
\frac{\partial}{\partial \phi_j} \left( \frac{\partial h}{\partial \phi_i} \right) = \tilde{E} \tilde{Q}_{j-1} \frac{\partial Q_{ji}}{\partial \phi_j} \tilde{Q}_{j+1,i-1} \frac{\partial Q_{ii}}{\partial \phi_i} \tilde{q}_{i+1} \tag{4.30}
\]
\[
\frac{\partial}{\partial \phi_j} \left( \frac{\partial h}{\partial l_i} \right) = 0_{2 \times 1} \tag{4.31}
\]
\[
\frac{\partial}{\partial l_j} \left( \frac{\partial g}{\partial \theta_i} \right) = \frac{\partial}{\partial l_j} \left( \frac{\partial g}{\partial \phi_i} \right) = \frac{\partial}{\partial l_j} \left( \frac{\partial g}{\partial l_i} \right) = 0_{3 \times 1} \tag{4.32}
\]
\[
\frac{\partial}{\partial l_j} \left( \frac{\partial h}{\partial \theta_i} \right) = \frac{\partial}{\partial l_j} \left( \frac{\partial h}{\partial \phi_i} \right) = \frac{\partial}{\partial l_j} \left( \frac{\partial h}{\partial l_i} \right) = 0_{2 \times 1} \tag{4.33}
\]
Next consider the second derivative of $w$, $\frac{\partial^2 w}{\partial \theta_i \partial \theta_j}$, where $i = j$. First differentiate eq.(3.23) with respect to $\theta_j$ to get

$$\frac{\partial^2 g}{\partial \theta_j^2} = \tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \theta_j^2} \tilde{p}_{i+1} \quad (4.34)$$

Similarly, the derivatives with respect to $\theta_j$ are,

$$\frac{\partial}{\partial \theta_j} \left( \frac{\partial g}{\partial \phi_j} \right) = \tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \theta_j \phi_j} \tilde{p}_{i+1} \quad (4.35)$$

$$\frac{\partial}{\partial \phi_j} \left( \frac{\partial g}{\partial \phi_j} \right) = \tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \phi_j^2} \tilde{p}_{i+1} \quad (4.36)$$

$$\frac{\partial^2 h}{\partial \theta_j^2} = E\tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \theta_j^2} \tilde{q}_{i+1} \quad (4.37)$$

$$\frac{\partial}{\partial \phi_j} \left( \frac{\partial h}{\partial \phi_j} \right) = E\tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \phi_j^2} \tilde{q}_{i+1} \quad (4.38)$$

$$\frac{\partial}{\partial \theta_j} \left( \frac{\partial h}{\partial \phi_j} \right) = 0_{2 \times 1} \quad (4.39)$$

Next, consider the derivatives with respect to $\phi_j$

$$\frac{\partial}{\partial \phi_j} \left( \frac{\partial g}{\partial \theta_j} \right) = \tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \theta_j \phi_j} \tilde{p}_{i+1} \quad (4.40)$$

$$\frac{\partial g}{\partial \phi_j} = \tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \phi_j^2} \tilde{p}_{i+1} \quad (4.41)$$

$$\frac{\partial}{\partial \phi_j} \left( \frac{\partial g}{\partial \phi_j} \right) = \tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \phi_j^2} \tilde{p}_{i+1} \quad (4.42)$$

$$\frac{\partial}{\partial \phi_j} \left( \frac{\partial h}{\partial \phi_j} \right) = E\tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \phi_j^2} \tilde{q}_{i+1} \quad (4.43)$$

$$\frac{\partial h}{\partial \phi_j} = E\tilde{Q}_{j-1} \frac{\partial^2 Q_j}{\partial \phi_j^2} \tilde{q}_{i+1} \quad (4.44)$$

$$\frac{\partial}{\partial \phi_j} \left( \frac{\partial h}{\partial \phi_j} \right) = 0_{2 \times 1} \quad (4.45)$$
Finally, the derivatives with respect to \( l_j \) are
\[
\frac{\partial}{\partial l_j} \left( \frac{\partial g}{\partial \phi_j} \right) = \frac{\partial}{\partial l_j} \left( \frac{\partial g}{\partial \phi_j} \right) = \frac{\partial^2 g}{\partial l_j^2} = 0_{3 \times 1} \tag{4.46}
\]
\[
\frac{\partial}{\partial l_j} \left( \frac{\partial h}{\partial \phi_j} \right) = \frac{\partial}{\partial l_j} \left( \frac{\partial h}{\partial \phi_j} \right) = \frac{\partial^2 h}{\partial l_j^2} = 0_{2 \times 1} \tag{4.47}
\]

So far, the elements in \( w_{\theta_j} \) have been expressed in terms of the first and second derivatives of the rotation matrix, \( Q_j \), \( \frac{\partial Q_j}{\partial \theta_j} \) and \( \frac{\partial^2 Q_j}{\partial \theta_j^2} \). We now derive the expression for these derivatives. First consider \( \frac{\partial Q_j}{\partial \theta_j} \).

\[
\frac{\partial Q_j}{\partial \theta_j} = \begin{bmatrix}
0 & 0 & 0 \\
\cos \theta_j \sin \phi_j & -\sin \theta_j & -\cos \theta_j \cos \phi_j \\
\sin \theta_j \sin \phi_j & \cos \theta_j & -\sin \theta_j \cos \phi_j
\end{bmatrix} \tag{4.48}
\]
\[
\frac{\partial Q_j}{\partial \phi_j} = \begin{bmatrix}
-\sin \phi_j & 0 & \cos \phi_j \\
\sin \theta_j \cos \phi_j & 0 & \sin \theta_j \sin \phi_j \\
-\cos \theta_j \cos \phi_j & 0 & -\cos \theta_j \sin \phi_j
\end{bmatrix} \tag{4.49}
\]
\[
\frac{\partial^2 Q_j}{\partial l_j} = 0_{3 \times 3} \tag{4.50}
\]

The second derivatives of the rotation matrices \( Q_j \) may be derived from the above equations.

\[
\frac{\partial^2 Q_j}{\partial \theta_j^2} = \begin{bmatrix}
0 & 0 & 0 \\
-\sin \theta_j \sin \phi_j & -\cos \theta_j & \sin \theta_j \cos \phi_j \\
\cos \theta_j \sin \phi_j & -\sin \theta_j & -\cos \theta_j \cos \phi_j
\end{bmatrix} \tag{4.51}
\]
\[
\frac{\partial^2 Q_j}{\partial \theta_j \phi_j} = \begin{bmatrix}
0 & 0 & 0 \\
\cos \theta_j \cos \phi_j & 0 & \cos \theta_j \sin \phi_j \\
\sin \theta_j \cos \phi_j & 0 & \sin \theta_j \sin \phi_j
\end{bmatrix} \tag{4.52}
\]
\[
\frac{\partial^2 Q_j}{\partial \phi_j^2} = \begin{bmatrix}
-\cos \phi_j & 0 & -\sin \phi_j \\
-\sin \theta_j \sin \phi_j & 0 & \sin \theta_j \cos \phi_j \\
\cos \theta_j \sin \phi_j & 0 & -\cos \theta_j \cos \phi_j
\end{bmatrix} \tag{4.53}
\]
Thus far we have derived the solutions for $\frac{\partial x}{\partial q_j}$ and $\frac{\partial W}{\partial q_j}$ in eq.(4.14), which is part of RMRC equation, (4.12). We are still missing the solution for $W^{-1}$ and $\frac{\partial W^{-1}}{\partial q_j}$ in order to complete eq.(4.14). Since the dimensions of $W_1$ depends on the number of kinematic constraints, $m$, we can not find the general solution for the problem. However, an example case with $m = 3$ is considered here.

For $m = 3$, $W_1$ may be written as

$$W_1 = \begin{bmatrix} \frac{\partial q_1}{\partial \theta_1} & \frac{\partial q_2}{\partial \theta_1} & \frac{\partial q_3}{\partial \theta_1} \\ \frac{\partial q_1}{\partial \theta_2} & \frac{\partial q_2}{\partial \theta_2} & \frac{\partial q_3}{\partial \theta_2} \\ \frac{\partial q_1}{\partial \theta_3} & \frac{\partial q_2}{\partial \theta_3} & \frac{\partial q_3}{\partial \theta_3} \end{bmatrix} \tag{4.54}$$

The adjoint matrix of $W_1$ is denoted by $\tilde{W}_1$, where

$$\tilde{W}_1 = \begin{bmatrix} \frac{\partial q_1}{\partial \theta_1} \frac{\partial q_2}{\partial \theta_1} \frac{\partial q_3}{\partial \theta_1} & \frac{\partial q_1}{\partial \theta_1} \frac{\partial q_2}{\partial \theta_1} \frac{\partial q_3}{\partial \theta_1} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \frac{\partial q_1}{\partial \theta_3} \frac{\partial q_2}{\partial \theta_3} \frac{\partial q_3}{\partial \theta_3} \end{bmatrix} \tag{4.55}$$

And, the determinant of $W_1$ is obtained as

$$w_d = |W_1| = \frac{\partial q_1}{\partial \theta_1}(\tilde{W}_1(1,1)) + \frac{\partial q_2}{\partial \theta_1}(\tilde{W}_1(1,2)) + \frac{\partial q_3}{\partial \theta_1}(\tilde{W}_1(1,3)) \tag{4.56}$$

The corresponding derivatives of $W_1$ with respect to $q_j$ take on the forms

$$\frac{\partial W_1}{\partial q_j} = \frac{1}{w_d} \frac{\partial \tilde{W}_1}{\partial q_j} - \frac{1}{w_d^2} \frac{\partial w_d}{\partial q_j} \tilde{W}_1 \tag{4.57}$$
The elements of $\frac{\partial \mathbf{W}_1}{\partial q_j}$ can be expressed as follows.

\[
\frac{\partial \mathbf{W}_1}{\partial q_j}(1,1) = \frac{\partial}{\partial q_j}\left(\frac{\partial g_2}{\partial \phi_1}\right) \frac{\partial g_3}{\partial l_1} + \frac{\partial g_2}{\partial \phi_1} \frac{\partial}{\partial q_j}\left(\frac{\partial g_3}{\partial l_1}\right) - \frac{\partial}{\partial q_j}\left(\frac{\partial g_3}{\partial \theta_1}\right) \frac{\partial g_2}{\partial l_1} - \frac{\partial g_3}{\partial \theta_1} \frac{\partial}{\partial q_j}\left(\frac{\partial g_2}{\partial l_1}\right)
\]

(4.58)

\[
\frac{\partial \mathbf{W}_1}{\partial q_j}(2,2) = \frac{\partial}{\partial q_j}\left(\frac{\partial g_3 g_1}{\partial \phi_1 \partial l_1}\right) + \frac{\partial g_3}{\partial \phi_1} \frac{\partial}{\partial q_j}\left(\frac{\partial g_1}{\partial l_1}\right) - \frac{\partial}{\partial q_j}\left(\frac{\partial g_1}{\partial \theta_1}\right) \frac{\partial g_3}{\partial l_1} - \frac{\partial g_1}{\partial \theta_1} \frac{\partial}{\partial q_j}\left(\frac{\partial g_3}{\partial l_1}\right)
\]

(4.59)

\[
\vdots
\]

\[
\frac{\partial \mathbf{W}_1}{\partial q_j}(3,3) = \frac{\partial}{\partial q_j}\left(\frac{\partial g_1}{\partial \theta_1}\right) \frac{\partial g_2}{\partial \phi_1} + \frac{\partial g_1}{\partial \theta_1} \frac{\partial}{\partial q_j}\left(\frac{\partial g_2}{\partial \phi_1}\right) - \frac{\partial}{\partial q_j}\left(\frac{\partial g_2}{\partial \theta_1}\right) \frac{\partial g_1}{\partial l_1} - \frac{\partial g_2}{\partial \theta_1} \frac{\partial}{\partial q_j}\left(\frac{\partial g_1}{\partial l_1}\right)
\]

(4.60)

Lastly, the derivative of the determinant of $\mathbf{W}_1$, $w_d$, may be derived from eq. (4.56) as

\[
\frac{\partial w_d}{\partial q_j} = \frac{\partial}{\partial q_j}\left(\frac{\partial g_1}{\partial \theta_1}\right) \left(\mathbf{W}_1(1,1)\right) + \frac{\partial g_1}{\partial \theta_1} \frac{\partial}{\partial q_j}\left(\mathbf{W}_1(1,1)\right)
\]

\[
+ \frac{\partial}{\partial q_j}\left(\frac{\partial g_2}{\partial \theta_1}\right) \left(\mathbf{W}_1(1,2)\right) + \frac{\partial g_2}{\partial \theta_1} \frac{\partial}{\partial q_j}\left(\mathbf{W}_1(1,2)\right)
\]

\[
+ \frac{\partial}{\partial q_j}\left(\frac{\partial g_3}{\partial \theta_1}\right) \left(\mathbf{W}_1(1,3)\right) \frac{\partial g_3}{\partial \theta_1} \frac{\partial}{\partial q_j}\left(\mathbf{W}_1(1,3)\right)
\]

(4.61)

4.3 Differential Kinematics of a Trussarm Module

Based on the solution of kinematic problem presented in Sections 3.2.1 and 3.2.2, differential kinematic relations for a module are derived here. The subscript $j$ of $q$ and $k$, i.e., the module number, is omitted in this and the following sections.

50
4.3.1 Differential Inverse Kinematics

The objective of this section is to derive the algebraic solution to the inverse Jacobian matrix, $G$:

$$G \equiv J^{-1} = \frac{\partial k}{\partial q}$$  \hspace{1cm} (4.62)

where, $k = (k_1, k_2, k_3)$, and $q = (\theta, \phi, l)$. Hence,

$$G \equiv J^{-1} = \begin{bmatrix}
\frac{\partial k_1}{\partial \theta} & \frac{\partial k_1}{\partial \phi} & \frac{\partial k_1}{\partial l} \\
\frac{\partial k_2}{\partial \theta} & \frac{\partial k_2}{\partial \phi} & \frac{\partial k_2}{\partial l} \\
\frac{\partial k_3}{\partial \theta} & \frac{\partial k_3}{\partial \phi} & \frac{\partial k_3}{\partial l}
\end{bmatrix}$$  \hspace{1cm} (4.63)

Equations (3.54), (3.55) and (3.56) may be written as follows:

$$k_1^2 = (r_2 - r_3)^T(r_2 - r_3)$$  \hspace{1cm} (4.64)

$$k_2^2 = (r_3 - r_1)^T(r_3 - r_1)$$  \hspace{1cm} (4.65)

$$k_3^2 = (r_1 - r_2)^T(r_1 - r_2)$$  \hspace{1cm} (4.66)

Differentiating the above equations with respect to $q_i$, for $i = 1, 2, \text{ and } 3$, yields:

$$k_1 \frac{\partial k_1}{\partial q_i} = (r_2 - r_3)^T \frac{\partial}{\partial q_i} (r_2 - r_3)$$  \hspace{1cm} (4.67)

$$k_2 \frac{\partial k_2}{\partial q_i} = (r_3 - r_1)^T \frac{\partial}{\partial q_i} (r_3 - r_1)$$  \hspace{1cm} (4.68)

$$k_3 \frac{\partial k_3}{\partial q_i} = (r_1 - r_2)^T \frac{\partial}{\partial q_i} (r_1 - r_2)$$  \hspace{1cm} (4.69)

In order to consider derivatives with respect to all three independent variables, $\theta$, $\phi$ and $l$, let us define an operator, $\frac{\partial}{\partial q}$, as follows:

$$\frac{\partial}{\partial q} = \begin{bmatrix}
\frac{\partial}{\partial \theta} & \frac{\partial}{\partial \phi} & \frac{\partial}{\partial l}
\end{bmatrix}$$  \hspace{1cm} (4.70)
Then, for for $j = 1, 2, 3$, we have

$$\frac{\partial r_j}{\partial q_i} = \begin{bmatrix} \frac{\partial r_i}{\partial q} & \frac{\partial r_j}{\partial q} & \frac{\partial r_l}{\partial q} \end{bmatrix}$$  

(4.71)

Upon solving eqs. (4.67), (4.68) and (4.69) for $\frac{\partial r_i}{\partial q_i}$, we obtain

$$G = \begin{bmatrix} \frac{1}{k_1}(r_2 - r_3)^T \frac{\partial}{\partial q_1}(r_2 - r_3) \\ \frac{1}{k_2}(r_3 - r_1)^T \frac{\partial}{\partial q_2}(r_3 - r_1) \\ \frac{1}{k_3}(r_1 - r_2)^T \frac{\partial}{\partial q_3}(r_1 - r_2) \end{bmatrix}$$  

(4.72)

Also, eq.(3.40) can be differentiated to find an expression for $\frac{\partial r_i}{\partial q_i}$, namely,

$$\frac{\partial r_i}{\partial q_i} = (-c_2 \frac{\partial r_i}{\partial q_i} \sin \gamma_j) e_j + (c_2 \frac{\partial r_i}{\partial q_i} \cos \gamma_j) \hat{n}_0$$

$$= -\frac{\partial r_i}{\partial q_i} (c_2 \sin \gamma_j e_j - c_2 \cos \gamma_j \hat{n}_0)$$  

(4.73)

Here, the only unknown variables are $\frac{\partial r_i}{\partial q_i}$. From eq.(3.49), we have

$$\frac{\partial a_j}{\partial q_i} \cos \gamma_j - a_j \sin \gamma_j \frac{\partial r_i}{\partial q_i} + \frac{\partial b_i}{\partial q_i} \sin \gamma_j + b \frac{\partial r_i}{\partial q_i} \cos \gamma_j = \frac{\partial f_i}{\partial q_i}$$  

(4.74)

Rearrange this equation to solve for $\frac{\partial r_i}{\partial q_i}$.

$$\frac{\partial \gamma_j}{\partial q_i} = \frac{\frac{\partial f_i}{\partial q_i} - \frac{\partial a_i}{\partial q_i} \cos \gamma_j - \frac{\partial b_i}{\partial q_i} \sin \gamma_j}{b \cos \gamma_j - a_j \sin \gamma_j}$$  

(4.75)

In order to solve this equation, we now need the solutions for $\frac{\partial a_i}{\partial q_i}$, $\frac{\partial f_i}{\partial q_i}$, $\frac{\partial b_i}{\partial q_i}$ and $\frac{\partial f_i}{\partial q_i}$. First, let us consider $\frac{\partial a_i}{\partial q_i}$. From eq.(3.50):

$$\frac{\partial a_i}{\partial q_i} = c_j \left[ \frac{1}{l} \frac{\partial \hat{n}_2}{\partial q_i} + (\hat{n}_0 + \hat{n}_2) \frac{\partial l}{\partial q_i} \right]$$  

(4.76)

If we define new vectors $a = \text{col}\{a_1, a_2, a_3\}$ and $f = \text{col}\{f_1, f_2, f_3\}$, their derivatives may be expressed in the matrix form as

$$\frac{\partial a}{\partial q} = \begin{bmatrix} \frac{\partial a_1}{\partial q} & \frac{\partial a_2}{\partial q} & \frac{\partial a_3}{\partial q} \end{bmatrix}$$  

(4.77)
and similar for $\frac{\partial f}{\partial q}$. Equation (4.76) then can be written as

$$\frac{\partial a}{\partial q} = e^T \left[ \frac{\partial \hat{n}_2}{\partial q} l + (\hat{n}_0 + \hat{n}_2) \frac{\partial l}{\partial q} \right]$$

$$= e^T \begin{bmatrix} 0 & l \cos \phi & 0 \\ l \cos \theta \cos \phi & l \sin \theta \sin \phi & 0 \\ -l \sin \theta \cos \phi & -l \cos \theta \sin \phi & 1 + \cos \theta \cos \phi \end{bmatrix}$$

(4.78)

Recall that $e$ is defined in eq.(3.41).

Next, for $\frac{\partial f}{\partial q}$ differentiate eq.(3.52).

$$\frac{\partial f}{\partial q} = \frac{1}{c_2} - c_1 \frac{\partial a}{\partial q} + I_{3 \times 1} \left[ \frac{\partial b}{\partial q} + (l - \delta_1) \frac{\partial b}{\partial q} - \delta_2 \frac{\partial q}{\partial q} \right]$$

(4.79)

where

$$\frac{\partial q}{\partial q} = (\hat{n}_0 + \hat{n}_2)^T \left( \frac{\partial \hat{n}_1}{\partial q} l + \hat{n}_1^T \left( \frac{\partial \hat{n}_2}{\partial q} l + (\partial l/\partial q) \right) \right)$$

(4.80)

Third, differentiate eq.(3.51) to find $\frac{\partial b}{\partial q}$.

$$\frac{\partial b}{\partial q} = \hat{n}_0^T \left[ \frac{\partial \hat{n}_2}{\partial q} l + (\hat{n}_0 + \hat{n}_2) \frac{\partial l}{\partial q} \right]$$

$$= \begin{bmatrix} -l \sin \theta \cos \phi & -l \cos \theta \sin \phi & 1 + \cos \theta \cos \phi \end{bmatrix}$$

(4.81)

Lastly, $\frac{\partial l}{\partial q}$ is simply,

$$\frac{\partial l}{\partial q} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

(4.82)

Up to this point, derivatives have been expressed in terms of derivatives of unit normals, $\hat{n}_1$ and $\hat{n}_2$. From (3.32),

$$\frac{\partial \hat{n}_2}{\partial q} = \begin{bmatrix} 0 & \cos \phi & 0 \\ -\cos \theta \cos \phi & \sin \theta \sin \phi & 0 \\ -l \sin \theta \cos \phi & -l \cos \theta \sin \phi & 0 \end{bmatrix}$$

(4.83)
From eq. (3.38), we have

\[
\frac{\partial \hat{n}_1}{\partial q} = \frac{1}{\alpha^2} \left[ \alpha \frac{\partial \hat{n}_2}{\partial q} - \frac{\partial \alpha}{\partial q} (\hat{n}_0 + \hat{n}_2) \right]
\]

(4.84)

where

\[
\frac{\partial \alpha}{\partial q} = \frac{1}{\alpha} \begin{bmatrix} -\sin \theta \cos \phi & -\cos \theta \sin \phi & 0 \end{bmatrix}
\]

(4.85)

### 4.3.2 Differential Forward Kinematics

The forward Jacobian, \( J \), is developed in this section. At the outset, we note that

\[
J = \frac{\partial q}{\partial k}
\]

(4.86)

Differentiating eqs. (3.66), (3.67), and (3.68) with respect to \( k \), yields:

\[
\begin{bmatrix} 2k_1 & 0 & 0 \end{bmatrix} = 2c_2^2 \sin(\gamma_3 - \gamma_2) \left( \frac{\partial \gamma_3}{\partial k} - \frac{\partial \gamma_2}{\partial k} \right)
\]

\[-3c_2 \left[ \frac{\partial \gamma_2}{\partial k} \sin \gamma_2 (c_1 + c_2 \cos \gamma_3) + \frac{\partial \gamma_3}{\partial k} \sin \gamma_3 (c_1 + c_2 \cos \gamma_2) \right]
\]

\[= \alpha_1 \frac{\partial \gamma_2}{\partial k} - \beta_1 \frac{\partial \gamma_3}{\partial k}
\]

(4.87)

\[
\begin{bmatrix} 0 & 2k_2 & 0 \end{bmatrix} = 2c_2^2 \sin(\gamma_1 - \gamma_3) \left( \frac{\partial \gamma_1}{\partial k} - \frac{\partial \gamma_3}{\partial k} \right)
\]

\[-3c_2 \left[ \frac{\partial \gamma_3}{\partial k} \sin \gamma_3 (c_1 + c_2 \cos \gamma_1) + \frac{\partial \gamma_1}{\partial k} \sin \gamma_1 (c_1 + c_2 \cos \gamma_3) \right]
\]

\[= \alpha_2 \frac{\partial \gamma_3}{\partial k} - \beta_2 \frac{\partial \gamma_1}{\partial k}
\]

(4.88)

\[
\begin{bmatrix} 0 & 0 & 2k_3 \end{bmatrix} = 2c_2^2 \sin(\gamma_2 - \gamma_1) \left( \frac{\partial \gamma_2}{\partial k} - \frac{\partial \gamma_1}{\partial k} \right)
\]

\[-3c_2 \left[ \frac{\partial \gamma_1}{\partial k} \sin \gamma_1 (c_1 + c_2 \cos \gamma_2) + \frac{\partial \gamma_2}{\partial k} \sin \gamma_2 (c_1 + c_2 \cos \gamma_1) \right]
\]

\[= \alpha_3 \frac{\partial \gamma_1}{\partial k} - \beta_3 \frac{\partial \gamma_2}{\partial k}
\]

(4.89)
where $\alpha_i$ and $\beta_i$ for $i = 1, 2, 3$ are known constants. For example,

$$\alpha_1 = -2c_2^2 \sin(\gamma_3 - \gamma_2) - 3c_2(c_1 + c_2 \cos \gamma_3) \sin \gamma_2$$

$$\beta_1 = 2c_2^2 \sin(\gamma_3 - \gamma_2) - 3c_2(c_1 + c_2 \cos \gamma_3) \sin \gamma_3$$

In matrix form,

$$\begin{bmatrix} \alpha_1 & -\beta_1 \\ -\beta_2 & 0 & \alpha_2 \\ \alpha_3 & -\beta_3 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \gamma_i}{\partial k_i}^T \\ \frac{\partial \gamma_j}{\partial k_i}^T \\ \frac{\partial \gamma_j}{\partial k_i}^T \end{bmatrix} = 2\tilde{k}$$

where $0$ is the $3 \times 3$ zero matrix, $\alpha_1 \equiv \text{diag}(\alpha_1, \alpha_1, \alpha_1)$, $\beta_1 \equiv \text{diag}(\beta_1, \beta_1, \beta_1)$, and so on, and $\tilde{k} = \text{col}\{k_1, 0, 0, 0, k_2, 0, 0, 0, k_3\}$. The above equation is solved for $\frac{\partial \gamma_i}{\partial k_i}$ for $i$ and $j = 1, 2, 3$. Moreover, by differentiating eq.(3.40) with respect to $k$, we can also find $\frac{\partial \gamma_i}{\partial k_i}$ as

$$\frac{\partial r_j}{\partial k_i} = (-c_2^2 \frac{\partial \gamma_i}{\partial k_i} \sin \gamma_j) e_j + (c_2 \frac{\partial \gamma_i}{\partial k_i} \cos \gamma_j) \hat{n}_0$$

$$= -\frac{\partial \gamma_i}{\partial k_i} (c_2 \sin \gamma_j e_j - c_2 \cos \gamma_j \hat{n}_0)$$

From the definition of the unit normal of actuator plane, $\hat{n}_1$, the following relation is derived.

$$\frac{\partial \hat{n}_1}{\partial k_i} = \frac{\partial}{\partial k_i} \left( \frac{\Delta r_1 \times \Delta r_2}{u_1} \right)$$

where

$$\Delta r_1 = r_2 - r_3$$

$$\Delta r_2 = r_1 - r_2$$

$$u_1 = ||\Delta r_1 \times \Delta r_2||$$
therefore,
\[
\frac{\partial \mathbf{n}_1}{\partial k_i} = \frac{1}{u_i^2} \left[ u_1 \frac{\partial (\Delta r_1 \times \Delta r_2)}{\partial k_i} - (\Delta r_1 \times \Delta r_2) \frac{\partial u_1}{\partial k_i} \right] \\
= \frac{1}{u_i^2} \left[ u_1 \left( \frac{\partial \Delta r_1}{\partial k_i} \times \Delta r_2 + \Delta r_1 \times \frac{\partial \Delta r_2}{\partial k_i} \right) - \Delta r_1 \times \Delta r_2 \frac{\partial u_1}{\partial k_i} \right]
\]
(4.98)

Note that
\[
\frac{\partial \Delta r_1}{\partial k_i} = \frac{\partial r_1}{\partial k_i} - \frac{\partial r_2}{\partial k_i}
\]
(4.99)

eq.(4.97) may be expanded using the properties of the cross product, i.e.,
\[
\frac{\partial u_1}{\partial k_i} = \frac{1}{2u_1} \frac{\partial}{\partial k_i} \left[ \left( \Delta r_1 \times \Delta r_2 \right)^T \left( \Delta r_1 \times \Delta r_2 \right) \right] \\
= \frac{1}{2u_1} \frac{\partial}{\partial k_i} \left( \Delta r_1^T \Delta r_1 \right) \left( \Delta r_2^T \Delta r_2 \right) - \left( \Delta r_1^T \Delta r_2 \right)^2 \\
= \frac{1}{u_1} \left[ \left( \frac{\partial \Delta r_1^T}{\partial k_i} \Delta r_1 \right) \left( \Delta r_2^T \Delta r_2 \right) + \left( \Delta r_1^T \Delta r_1 \right) \left( \frac{\partial \Delta r_2^T}{\partial k_i} \Delta r_2 \right) \right] \\
- \frac{1}{u_1} \left[ \left( \Delta r_1^T \Delta r_2 \right) \left( \frac{\partial \Delta r_1}{\partial k_i} \Delta r_2 + \Delta r_1^T \frac{\partial \Delta r_2}{\partial k_i} \right) \right]
\]
(4.100)

Recall that, \( \hat{n}_2 \equiv \alpha \hat{n}_1 - \hat{n}_0 \), and \( \alpha \equiv \sqrt{2(1 + \cos \theta \cos \phi)} \). Hence,
\[
\frac{\partial \hat{n}_1}{\partial k_i} = \frac{1}{\alpha^2} \left( \frac{\partial \hat{n}_2}{\partial k_i} \alpha - \frac{\partial \alpha}{\partial k_i} \left( \hat{n}_0 + \hat{n}_2 \right) \right)
\]
(4.101)

Rearrange the above equation for \( \hat{n}_2 \):
\[
\frac{\partial \hat{n}_2}{\partial k_i} = \alpha \frac{\partial \hat{n}_1}{\partial k_i} + \frac{1}{\alpha} \frac{\partial \alpha}{\partial k_i} \left( \hat{n}_0 + \hat{n}_2 \right)
\]
(4.102)

But, \( \alpha \equiv 2 \hat{n}_1 \). Therefore,
\[
\frac{\partial \alpha}{\partial k_i} = 2 \frac{\partial \hat{n}_1}{\partial k_i}
\]
(4.103)

\(^1\)Recall the property of cross products
\[
\|a \times b\|^2 = \|a\|^2\|b\|^2 - (a^T b)^2
\]
Similarly, \( \hat{n}_2 \) is a function of \( \theta \) and \( \phi \). Therefore, from eq. (3.71), we have

\[
\frac{\partial \phi}{\partial k_i} = \sec \phi \frac{\partial \hat{n}_{2x}}{\partial k_i} \tag{4.104}
\]

\[
\frac{\partial \theta}{\partial k_i} = \sec \theta \sec \phi \left( -\frac{\partial \hat{n}_{2y}}{\partial k_i} + \sin \theta \sin \phi \frac{\partial \phi}{\partial k_i} \right) \tag{4.105}
\]

Finally, from eq. (3.75):

\[
\frac{\partial l}{\partial k_i} = \frac{1}{\beta^2} \left( \frac{\partial u_2}{\partial k_i} - \frac{\partial \hat{n}_2}{\partial k_i} \right) \tag{4.106}
\]

where

\[
\beta \equiv 1 + \cos \theta \cos \phi = \frac{\alpha^2}{2} \tag{4.107}
\]

\[
u_2 \equiv (\hat{r}_2 + \delta_2 \hat{n}_1)^T (\hat{n}_0 + \hat{n}_2) \tag{4.108}
\]

\[
\frac{\partial u_2}{\partial k_i} \equiv \left( \frac{\partial \hat{r}_2}{\partial k_i} + \delta_2 \frac{\partial \hat{n}_1}{\partial k_i} \right)^T (\hat{n}_0 + \hat{n}_2) + (\hat{r}_2 + \delta_2 \hat{n}_1)^T \frac{\partial \hat{n}_2}{\partial k_i} \tag{4.109}
\]

The Jacobian is, in fact, the collection of eqs. (4.105), (4.104) and (4.106), i.e.,

\[
J = \begin{bmatrix}
\frac{\partial \phi}{\partial k_1} & \frac{\partial \phi}{\partial k_2} & \frac{\partial \phi}{\partial k_3} \\
\frac{\partial \theta}{\partial k_1} & \frac{\partial \theta}{\partial k_2} & \frac{\partial \theta}{\partial k_3} \\
\frac{\partial \hat{n}_{2x}}{\partial k_1} & \frac{\partial \hat{n}_{2x}}{\partial k_2} & \frac{\partial \hat{n}_{2x}}{\partial k_3}
\end{bmatrix} \tag{4.110}
\]
Chapter 5

Performance Measurements

The robotic applications have been proposed in numerous areas where the use of humans is impractical and undesirable. The following areas are examples of the application of industrial manipulators:

- material transfer tasks, such as molding and stamping
- assembly tasks such as welding and grinding
- tasks in hazardous environments such as undersea and planetary exploration, and satellite retrieval and repair.

The performance of an industrial manipulator is often determined by several criteria including:

- workspace
- dexterity
- repeatability
- speed, and
Depending on the application of the manipulator, the significance of each property may vary. For example, for a positioning application such as a pick-and-place maneuver, speed and accuracy are the most important characteristics while dexterity is not as significant.

In order to examine Trussarm's practicality as an industrial manipulator, these properties must be determined. This chapter is about such measurements. Throughout this chapter, the control experiments designed to measure some of the properties are outlined and performance characteristics are discussed. The measurement methods introduced here may be expanded to other VGT manipulators.

Sections 5.1 through 5.4 focus on the general characteristics of Trussarm. A more in-depth discussion is provided for accuracy of the kinematic control in Section 5.5. The kinematics model, developed in the previous chapters, are applied to implement kinematic control algorithm. In Sections 5.5.2 and 5.5.3, the forward and inverse kinematics algorithm presented in Chapter 3 are applied for position control and tracking control, respectively. In Section 5.5.5, resolved-motion rate control developed in Chapter 4 is applied to examine the velocity control.

5.1 Workspace Analysis

5.1.1 Definition

The workspace of a manipulator is the total volume swept out by the end-effector as the manipulator executes all possible motions. Two types of workspace are often considered: a reachable workspace and a dexterous workspace. Spong
and Vidyasager (1989) defined the reachable workspace as "the entire set of points in space reachable by the manipulator," whereas the dexterous workspace is the set of points that "the manipulator can reach with an arbitrary orientation of the end effector." In this section, only the reachable workspace of Trussarm is considered. The dexterous workspace of Trussarm will be considered in the following section.

5.1.2 Workspace Measurement Methods

The methods to measure workspaces have been examined by several authors including Korein (1985) and Kwon et. al. (1994). One commonly used method is the ‘extension method.’ A set of points in the workspace is generated by stepping each actuator through its range. For Trussarm with \( n \) modules (that is, \( 3n \) actuators), \( r^{3n} \) combinations of joint coordinates have to be considered, where \( r \) is the number of discrete joint settings. For example, when three settings (maximum, center and minimum actuator lengths) are used for 12-DOF Trussarm, \( 3^{12} = 531,441 \) points have to be considered. It is extremely difficult to examine that many points to derive a mathematical description of the workspace.

The extension method was first applied to Trussarm by Sallmen (1993) for one module. Sallmen described the workspace of a Trussarm module as an umbrella with three corners. The workspace of 12-DOF Trussarm was studied by Hughes et al. (1999). In both studies, Trussarm workspace was computed by solving forward kinematics at each actuator lengths at discrete actuator displacements. The result found in Hughes et al. (1999) is shown in Figure 5.1. Similar simulation was performed with a set of actuator displacements that yield end-effector positions on the \( xz \)-plane. Such joint states can be
Figure 5.1: Trussarm Workspace by Extension Method
Figure 5.2: Trussarm Workspace by Extension Method, $xz$-plane
uniquely defined as $k_{3i+1} = k_{3i+2}$ for $i = 0, 1, \ldots, n - 1$. To keep the number of discrete configurations small, three joint states are chosen: $k_{\text{max}}$, 0, and $k_{\text{min}}$. The resulting joint sets are combination of the following settings for each module: $(k_{\text{max}}, k_{\text{max}}, k_{\text{min}})$, $(k_{\text{max}}, k_{\text{max}}, 0)$, $(k_{\text{max}}, k_{\text{max}}, k_{\text{max}})$, $(0, 0, k_{\text{max}})$, $(0, 0, 0)$, $(0, 0, k_{\text{min}})$, $(k_{\text{min}}, k_{\text{min}}, k_{\text{max}})$, $(k_{\text{min}}, k_{\text{min}}, 0)$, and $(k_{\text{min}}, k_{\text{min}}, k_{\text{min}})$. For 4 modules, the total number of discrete configuration is $9^4 = 6,561$. Result of the simulation test is shown in Figure 5.2.

Another method is the 'sweeping method.' This method was proposed for an open-link kinematic chain by Korein (1985). First, the last joint in the chain is swept through its range to generate $W_1$. Then, the second joint is swept to generate $W_2$ and so on until the last joint is swept.

Since Trussarm is not an open-link serial manipulator, the sweeping method can not be directly applied. The method proposed here is a combination of the extension method and the sweeping method. First, the extension method is applied to the $n$th module to determine the workspace of the last module. Then, joints in the module $n-1$ are varied to determine the workspace of chain of module $n$ and $n-1$, and so forth. As in the previous section, the workspace on $xz$-plane is considered here. The boundary of the workspace is approximated as follows: (refer to Figure 5.3.)

- Choose a point in space, $c_o$.
- Compute the distance, $r_i$ and angle, $\theta_i$ from $c_o$ to an end-effector position within the workspace, $p_i$:

$$ r_i = |p_i - c_o| $$

$$ \theta_i = \tan^{-1}\left(\frac{C_{o,x} - p_{i,x}}{C_{o,x} - p_{i,z}}\right) $$

63
Determine points $p_j$ where its radius is the maximum and minimum among all $p_i$'s such that $\theta_{j-1} \leq \theta_i \leq \theta_j$ ($\theta_j = \frac{2\pi}{N}$), $j = 1, \ldots, N$. $N$ can be arbitrarily chosen for the accuracy of the workspace representation.

The workspace boundary can be approximated as the curve on which $p_j$ lie.

This method of approximation can be easily extended to the three dimensional representation of the workspace.

Figure 5.4 shows the workspace of one module generated by the sweeping method. The nine actuator displacements that were used in the extension methods are also marked on the same graph to permit comparison between the two methods. Figure 5.5, 5.6 and 5.7 show the workspace of 2, 3 and 4 modules respectively. The boundary of workspace is marked with a dotted line. The shaded region on each plot is the workspace of chain 1, 2 and 3, respectively. Figure 5.8 is the collection of the results from Figure 5.5, 5.5, 5.6 and 5.7. It can be easily seen how quickly the workspace of Trussarm grows with the number of modules.

Lastly, Figure 5.8 is the combination of results from the extension method and the sweeping method. From this comparison, it is noted that the extension and sweeping methods provide very similar results. However, because sweeping is only the first-order approximation, it does not yield all possible points. Instead, only a good approximation can be obtained. On the other hand, this method is more computationally efficient and less time consuming.
Figure 5.3: Workspace Boundary Approximation

(i) Choose \( C \) in space

(ii) Compute \( r \) and \( \theta \) for \( p \).

(iii) Determine \( r_{max} \) and \( r_{min} \).

(iv) Approximate the workspace boundary.
Figure 5.4: Workspace of 1 Module  Figure 5.5: Workspace of 2 Modules

Figure 5.6: Workspace of 3 Modules  Figure 5.7: Workspace of 4 Modules
Figure 5.8: Trussarm Workspace by Sweeping Method, $xz$-plane

Figure 5.9: Trussarm Workspace by Sweeping Method, $xz$-plane
5.1.3 **Trussarm Workspace Experiments**

Next, the focus is placed on the experiments, designed to demonstrate the capability to reach the predicted workspace limits. First, a computer simulation with animation was developed to simulate the motion during the workspace experiments. Figures 5.10 through 5.15 show some examples from the computer simulation. Table 3.1 lists the actuator lengths in cm used in the simulation tests and also the resulting end-effector coordinates. Even though each of the twelve actuators can extend and contract up to ± 7.62 cm, a ± 4.0 to ± 6.0 cm extension was chosen for this experiment, to allow for a safety margin.

![Figure 5.10: Home Configuration](image)

![Figure 5.11: J-Shape Configuration](image)

<table>
<thead>
<tr>
<th>Configuration</th>
<th>(l_{1,4,7,10})</th>
<th>(l_{2,5,8,11})</th>
<th>(l_{3,6,9,12})</th>
<th>End-Effector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home Position</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>(0.0, 0.0, 193.5)</td>
</tr>
<tr>
<td>Contracted</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>(0.0, 0.0, 167.5)</td>
</tr>
<tr>
<td>Extended</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>(0.0, 0.0, 210.9)</td>
</tr>
<tr>
<td>Positive X Reach</td>
<td>4.0</td>
<td>4.0</td>
<td>-4.0</td>
<td>(132.4, 0.0, 111.4)</td>
</tr>
<tr>
<td>Negative X Reach</td>
<td>-4.0</td>
<td>-4.0</td>
<td>4.0</td>
<td>(-122.6, 0.0, 112.5)</td>
</tr>
<tr>
<td>J-Shape</td>
<td>-4.0 / 4.0</td>
<td>-4.0 / 4.0</td>
<td>4.0 / -4.0</td>
<td>(80.3, 0.0, 168.1)</td>
</tr>
</tbody>
</table>

Table 5.1: Trussarm Actuator Lengths in Workspace Demonstration
Figure 5.12: Contracted Configuration  Figure 5.13: Extended Configuration

Figure 5.14: Positive $z$ Reach  Figure 5.15: Negative $z$ Reach
The series of motions illustrated in Figures 5.10 through 5.15 were performed with Trussarm at a relatively low speed. The motor coordinates were computed from the joint coordinates in Table 5.1. A set of DMC programs were used to perform the experiments. Using the high-level control software (referred to as the Trussarm Control Environment in Chapter 2) as an interface tool, the motor commands were executed for all twelve actuators simultaneously.

The following is an excerpt of the DMC programs for Trussarm workspace tests:

```
SH XYZEFG
PA 0, 0, 0, 0, 0, 0
BG
AN
CM XYZEFG
DT 8
CD -9288, -9288, -9288, -3378, -3378, -3378
WC
:
DTC
CD 0,0,0,0,0
PA 0,0,0,0,0
BG
EN
```

Note that throughout the program, the DMC commands DT and CD are repeatedly used. These commands are used for contour mode motions. The contour mode results in smoother motion of the manipulator. The contour is characterized by the position increments (CD) of each actuator and the time increments (DT) between each point on the contour.

During the experiment it was observed that as Trussarm reached the edge of the workspace, the motion of the manipulator was significantly restrained due to the structural limitation. It seems that the excessive loads...
on some members caused this problem. For example, in the Positive $x$-Reach configuration, actuator 3 was under a very large load. It was noted that at this point the actuator was, in fact, moving slower and in a jerky manner. When the commanded joint length exceeded 6 cm, the motion of the actuator came nearly to a stop. Since the gravitational effect is one of the most significant factors on the structural limitation, the size of the functional workspace highly depends on the design and size of the actuators.

While the nonlinear characteristics or backlash of the individual actuators cannot be modeled quantitatively at this time, it was clear that this uncertainty had a significant effect on the performance of Trussarm. This result suggests that Trussarm, or any other Trussarm-like VGT manipulators, may have 'functional' workspace within its theoretical workspace. For Trussarm, this functional workspace extends approximately 1.0 meters in radius and 0.5 meters in height. Recall that in home configuration, Trussarm is 1.9 meters in length.

The workspace of Trussarm is comparable to that of conventional industrial manipulators. For example, PUMA761 is an example of a typical industrial manipulator. As shown in Figure 5.16, PUMA761 is approximately 1.5 meters in length and its workspace extends 1.2 meters in radius.

### 5.1.4 Controllable Workspace

So far, the workspace has been considered as a set of all possible points in space that may be reached by solving the forward kinematics, given the joint coordinates. For a complex manipulator such as Trussarm, it is not guaranteed that every point in such space may be actually reached when the inverse kinematics is solved. In fact, only a subset of points in the reachable workspace
may be reached using the inverse kinematics algorithm (called the 'discrete model approach') derived in Chapter 3. This subset of workspace based on the solution of the inverse kinematics problem is referred to as the controllable workspace here, since the position control of the end-effector is possible only within this space.

The size of the controllable workspace varies with many factors such as redundancy resolution algorithm, joint limit and singularity avoidance methods, and accuracy in the kinematics model. The ratio between controllable workspace and reachable workspace may be used as a measure of control algorithm.

Figure 5.17 shows the cut section of Trussarm's controllable workspace on the $xz$-plane ($y = 0$). Note that many points in the reachable workspace are in controllable workspace. However, there are points near the edge of the workspace that are not controllable.
reachable workspace that cannot be reached by the discrete model approach. In Figure 5.17, 5.18, and 5.19, 'o' was marked when the inverse kinematics was successfully solved with joint coordinates within the joint limits, and 'x' was marked when it failed. Approximately 69 percent of the reachable workspace on the xz-plane is in the controllable workspace.

The controllable workspace is smaller when the cut section on xy-plane (z = 1.8 m) is considered as shown in Figure 5.18. When z = 1.8 m, the workspace spans approximately 1.6 meters in the x- and y-directions.

5.1.5 Reach Hierarchy Algorithm

In order to utilize the entire workspace, a control algorithm that makes use of the workspace information needs to be implemented. One approach is the reach hierarchy algorithm, adopted from [Korein, 1985]. An example of a modified reached hierarchy algorithm is now given for n modules. The concept of the reach hierarchy algorithm is,

1. First, define the workspace of chain i as $W_i$, where chain i consists of
modules $i$ through $n$.

2. If the desired end-effector position is not in $W_1$, then halt immediately.

3. Otherwise, adjust module 1 enough to bring the goal point within $W_2$.

4. Repeat the last step for each remaining chain, adjusting module $i$ to bring the goal point within $W_{i+1}$ for $i = 2, 3, \ldots, n$.

Figure 5.19 shows the controllable workspace slice at $y = 0$, when the reach hierarchy algorithm is applied. Approximately 92 percent of the reachable workspace can be reached using this algorithm. Compared to 69 percent in the discrete model approach, 23 percent increase is obtained. The 8 percent of the reachable workspace that can not be reached with reach hierarchy algorithm may be reached if an exact model of workspace is available. As shown in Figure 5.9, the workspace model is only an approximation, thus allowing 8 percent error.
5.2 Dexterity Analysis

5.2.1 Definition

One of the advantages of a redundant manipulator is its dexterity. The concept of dexterity can be interpreted in several ways. For example, a commonly accepted physical concept is one's ability to use one's hands, one's body or even one's mind. In robotics, dexterity has been interpreted as a specification of the kinematic or dynamic response of a manipulator [Yoshikawa, 1985]. A similar concept has been referred to as 'manipulability.' In this dissertation, we adopt the definition of dexterity by Park and Brockett (1994). Dexterity is "the ability to move and apply forces in arbitrary directions as easily as possible." As implied in the definition, dexterity can be considered in both its kinematic aspects and in its dynamic aspects. The focus here remains on the kinematic aspects of dexterity.
In kinematic control, dexterity measurement may be used to determine optimal 'working position' for the manipulator, or to find the 'optimal posture' for a given end-effector position and/or orientation. In this section, examples for both applications are discussed.

5.2.2 Dexterity Measurement

A survey of methods to quantitatively represent local dexterity was conducted by Klein and Blaho (1987). First, Yoshikawa (1985a) measured local dexterity by the determinant of $JJ^T$:

$$\eta = \sqrt{\text{det}(JJ^T)}$$

(5.3)

When $\eta$ approaches zero, the configuration of the manipulator is considered to be close to the singularity.

Second, the condition number of the transpose of the Jacobian, $c(J^T)$ was used as a measure of workspace quality [Salisbury and Craig, 1982], [Gosselin, 1990]:

$$c = \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}$$

(5.4)

where $\sigma_{\text{max}}$ and $\sigma_{\text{min}}$ are the largest and smallest singular values of the system Jacobian matrix. This is a measure of the accuracy with which forces can be exerted. [Yoshikawa, 1985a] The optimal value for this measurement is 1.

Third, the minimum singular value, $\sigma_{\text{min}}$ was also used as the measure of nearness to the singularity [Klein and Blaho, 1987]. The singularity is marked by the minimum singular value approaching zero.

Finally, joint range availability was used by Liegeois (1977):

$$E = \sum_{i=1}^{n} (q_i - q_i^c)^2$$

(5.5)
where \( q_i^c \) is the center of the range of each actuator. This method provides the 'naturalness' or 'evenness' of the manipulator's posture [Klein and Blaho, 1987].

In order to compare these measures of local dexterity, two simple joint trajectories were chosen for Trussarm. The resulting end-effector trajectories are a curve on the \( xz \)-plane and a straight line along the \( z \)-axis. A simulation test was performed along these trajectories and four values \( (\eta, c, \sigma_{\min} \text{ and } E) \) were calculated along the trajectory.

As shown in Figures 5.20 and 5.21, the variation in these measures is significant. Klein and Blaho conducted similar experiments and concluded that “it is not possible to conclude that one is superior to the others since the measures have slightly different purposes.”

In general, the results suggest that the dexterous workspace of Trussarm at the center of its workspace. Here, Trussarm is in the straight posture with very small changes in actuator lengths. Therefore, Trussarm is capable of performing more dextrous tasks in this area.

In addition to the determination of the dextrous areas in the workspace, dexterity may be also used as a design criterion or to select the optimal (or the most natural) posture for a given end-effector position and/or orientation. Figure 5.22 through 5.25 show four distinct postures of Trussarm, from which the end-effector can be located at \((0,0,1.85) \text{ m.}

Postures 1 and 2 are with constant positive and negative curvature. Posture 3 is when the changes in actuator lengths are equal throughout all four modules. Posture 4 is when the length change of actuators in module 1 are maximized and the other modules remain unmodified. Dexterity Measurements at four postures are shown in Figure 5.26. These measurements suggest that the second posture is the most natural posture with highest dexterity.
Figure 5.20: Dexterity Measures along a curve (xz-plane)
Figure 5.21: Dexterity Measures along a straight line (z-axis)
Figure 5.22: Posture 1  
Figure 5.23: Posture 2  
Figure 5.24: Posture 3  
Figure 5.25: Posture 4
5.3 Repeatability Analysis

5.3.1 Definition

Next, the repeatability of Trussarm is examined. The repeatability of a manipulator is a measure of how precisely the end-effector can be located, repeatedly.
5.3.2 Repeatability Measurement Method

In order to measure the repeatability, a very fine pen was used as a tool. A white board was set up such that pen tip would touch the surface only when it comes to a stop after the motion. Unfortunately, due to the structural vibration, short traces of the tool path were marked on some of the test results. While computing repeatability, these traces of the end-effector were ignored. See Figure 5.27.

Trussarm was moved from its home configuration to a test position and back to the home configuration. The motion was repeated 10 times for three test points. The result was scanned and the radius R of the circle surrounding the scattered points measured. The test points used are shown in Table 5.2.

<table>
<thead>
<tr>
<th>Test</th>
<th>( P_{ee} )</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>20,10,-10</td>
<td>0.14 cm</td>
</tr>
<tr>
<td>Test 2</td>
<td>10,15,0</td>
<td>0.11 cm</td>
</tr>
<tr>
<td>Test 3</td>
<td>10,10,10</td>
<td>0.10 cm</td>
</tr>
</tbody>
</table>

Table 5.2: Test Points Used in Repeatability Analysis

In order to measure the repeatability of the manipulator more accurately, a better method to locate the end-effector tip position is desired. For future reference, three-dimensional position sensor or a vision system are recommended.

5.3.3 Trussarm Repeatability

Based on the method used, it may be concluded that Trussarm’s repeatability is 1.4 mm. In other words, Trussarm is capable of moving its end-effector to a given location with 1.4 mm precision. Figures 5.28, 5.29 and 5.30 show the
Figure 5.27: Setup for Repeatability Analysis
sample results from these tests.

On comparison, the repeatability of the PUMA 700 series manipulators is ±0.2 mm. The repeatability of A645 manipulator (designed by CRS Robotics Copr., Burlington, Ontario) is ±0.05 mm.
5.4 Speed Analysis

5.4.1 Actuator Speed

As explained in Chapter 2, Trussarm’s actuators are independent linear (translational) actuators. Each is driven by a DC motor and a gear reduction assembly. The motors are rated at 6,500 RPM (maximum speed with no load). However, amplifiers that supply the power to the motors limit the actuator speed to 2,925 RPM. Since quadrature encoders output 4 counts per motor rotation, 2,925 PRM corresponds to $4 \times 2,925 = 37,440$ counts/sec. This speed has been used as a guideline for the maximum actuator speed. 37,400 counts/sec corresponds to approximately 1.4, 2.6, 2.1 and 1.4 cm/sec for actuators in Modules 1, 2, 3 and 4, respectively.

5.4.2 Tip Speed

Once the maximum actuator speeds are determined, the maximum tip speed can be either calculated using the forward kinematics model or measured experimentally. End-effector (a pen at $t=(-0.15, 0.28, 0)m$) position is ($-0.15, 0.28, 1.675$) and ($-0.15, 0.28, 2.109$) m when all actuators are extended and retracted to 80 percent of their maximum/minimum extension (6 cm and -6 cm, respectively), This vertical distance of 43.4 cm was used as a test path. At the uniform joint speed of 0.5 cm/s, a 43.4 cm path can be traced at the rate of $43.4 \text{ cm} / (12 \text{ cm} / 0.5 \text{ cm/s}) = 1.8 \text{ cm/s}$. Or, at 1.4 cm/s joint speed, 5.0 cm/s tip speed may be achieved.

For a horizontal path, a straight line from $(0,0,0)$ to $(20,0,0)$ cm was used, as shown in Figure 5.31. With a tool vector, $t = (-15, 28, 0) \text{ cm}$, an end-effector position of $(20, 0, 0) \text{ cm}$ can be achieved with the motor coordinates
shown in Table 5.3. The maximum joint speed is limited by actuator 3 to

<table>
<thead>
<tr>
<th>$p_{cc}$</th>
<th>0,0,0</th>
<th>20,0,0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_{1,2,3}$</td>
<td>0,0,0</td>
<td>-1038,-1088,-36432</td>
</tr>
<tr>
<td>$l_{4,5,6}$</td>
<td>0,0,0</td>
<td>-233,-260,-13345</td>
</tr>
<tr>
<td>$l_{7,8,9}$</td>
<td>0,0,0</td>
<td>-825,-772,-10591</td>
</tr>
<tr>
<td>$l_{10,11,12}$</td>
<td>0,0,0</td>
<td>-3449,-3097,-8058</td>
</tr>
</tbody>
</table>

Table 5.3: Test Points Used in Speed Analysis

36,432/37,440 $\approx$ 1 sec. Thus, the maximum tip speed for horizontal motion is 20 cm per 1 sec = 20 cm/s.

It should be noted that at this speed, structural vibration becomes evident during the acceleration and deceleration phases. This can be seen in Figure 5.31 which is a scanned image of the two end points of the horizontal test trajectory. It may be observed that the end-effector moves almost $\pm 1.0$ cm when accelerating and decelerating.

A more standard test in robotics for speed evaluation is the '12-inch pick-and-place' maneuver. Trussarm is capable of this pick-and-place maneuver in many directions. For test purposes, the three easiest paths were chosen: a 12-inch path along the x-, y- and z-axes. The maneuver is as follows: starting from the home position (0,0),

1. Move to the left by 6 inches, $(-6,0)$
2. Move down by 1 inch, \((-6,-1)\)

3. Move up by 1 inch, \((-6,0)\)

4. Move to the right by 12 inches, \((6,0)\)

5. Move down by 1 inch, \((6,-1)\)

6. Move up by 1 inch, \((6,0)\)

7. Move back to the home position, \((0,0)\)

In total, the end-effector path is 28 inches (71.1 cm), as shown in Figure 5.32.

Throughout the series of pick-and-place maneuvers, the maneuver times were determined. The sample results are shown in Table 5.4. The vertical and horizontal speeds are speeds of the end-effector along paths 1, 4, 7 and 2, 3, 5, 6. The total time, \(t_{\text{total}}\) in Table 5.4 corresponds to:

\[
t_{\text{total}} = \frac{24 \times 2.54}{V_h} + \frac{4 \times 2.54}{V_v}
\]  

\[\text{(5.6)}\]

Table 5.4: 12-inch Pick-And-Place Test Results
In conclusion, the maximum tip speed of Trussarm is 20 cm/s in horizontal motion, and 5 cm/s in vertical motion. Trussarm is capable of performing a standard 12-inch pick-and-place maneuver in less than 7 seconds horizontally, or 20 seconds vertically.

It appears that Trussarm is at disadvantage in speed when compared to conventional manipulators. The maximum linear tip speeds of both the PUMA 700 series robots and A465 are 1.0 m/s.

5.5 Accuracy Analysis

5.5.1 Definition

The accuracy of a manipulator is a measure of how accurately the manipulator can be directed to a given Cartesian point or trajectory within its workspace. First, the accuracy of Trussarm under kinematic control derived in Chapter 3 is considered. The position accuracy and tracking accuracy are examined separately. Next, the resolved-motion-rate control, derived in Chapter 4, is applied and the resulting accuracy is calculated.

It is noted that the accuracy of a manipulator highly depend on the configuration of the manipulator, and therefore, the sample results presented here only provides a foundation for further research.

5.5.2 Position Control

A set of desired joint trajectories was generated and applied to Trussarm, and the position of the end-effector was recorded, using a pen as a tool. The setup is shown in Figure 5.27. The forward kinematics was solved to compute the Cartesian coordinates of the end-effector that corresponded to the desired
joint coordinates. Then, the results from forward kinematics and the actual location of the pen were compared to compute the error in positioning.

Samples of the experimental data in [cm] are shown in Table 5.5. where

<table>
<thead>
<tr>
<th></th>
<th>$l_{1,2,3}$</th>
<th>$l_{4,5,6}$</th>
<th>$l_{7,8,9}$</th>
<th>$l_{10,11,12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample 1</td>
<td>0.4, 4</td>
<td>0.0, 0</td>
<td>0.0, 0</td>
<td>0.0, 0</td>
</tr>
<tr>
<td>Sample 2</td>
<td>0, -4, -4</td>
<td>0.0, 0</td>
<td>0.0, 0</td>
<td>0.0, 0</td>
</tr>
<tr>
<td>Sample 3</td>
<td>3.1, 3.1, 3.1</td>
<td>3.1, 3.1, 3.1</td>
<td>3.1, 3.1, 3.1</td>
<td>3.1, 3.1, 3.1</td>
</tr>
<tr>
<td>Sample 4</td>
<td>-2.9, -3.0, 2.2</td>
<td>-2.9, -2.9, -2.1</td>
<td>-2.9, -2.8, -2.5</td>
<td>-2.9, -2.7, -2.6</td>
</tr>
<tr>
<td>Sample 5</td>
<td>-0.4, 1.6, 0.1</td>
<td>-0.7, -0.2, -0.1</td>
<td>-1.0, -1.4, -1.2</td>
<td>-1.2, -1.2, -1.2</td>
</tr>
<tr>
<td>Sample 6</td>
<td>0, 0, 1.3</td>
<td>0, 0, 0.9</td>
<td>0, 0, 0.6</td>
<td>0.1, 0.1, 0.3</td>
</tr>
</tbody>
</table>

| Sample 1 | $x_a$ | $y_a$ | $z_a$ | $x_d$ | $y_d$ | $z_d$ | $|\delta P_{ae}|$ (\%) |
|----------|-------|-------|-------|-------|-------|-------|-----------------|
| Sample 1 | -35.2 | 0     | -9.4  | -37.1 | 0     | 0     | 5.2             |
| Sample 2 | 31.9  | 0     | 2.8   | 33.5  | 0     | 1.9   | 5.5             |
| Sample 3 | 0     | 0     | 10.1  | 0     | 0     | 10    | 1.0             |
| Sample 4 | 9.4   | 0     | 10.1  | 10.0  | 0     | 10.0  | 4.2             |
| Sample 5 | 0     | 14.1  | 0.2   | 0     | 15.0  | 0     | 6.1             |
| Sample 6 | 19.1  | 0     | 0.1   | 20.0  | 0     | 0     | 4.5             |

Table 5.5: Sample Experimental Data for Position Accuracy Analysis

$P_a = (x_a, y_a, z_a)$ is the actual pen position and $P_d = (x_d, y_d, z_d)$ is the desired Cartesian coordinates. The position error, $|\delta P_{ae}|$ is defined as:

$$
|\delta P_{ae}| = \sqrt{\frac{(x_a - x_d)^2 + (y_a - y_d)^2 + (z_a - z_d)^2}{x_d^2 + y_d^2 + z_d^2}}
$$

(5.7)

It was found that the position error was always $\leq 6\%$. Note that the position error also includes the measurement error.

5.5.3 Kinematic (Tracking) Control

Two types of Cartesian trajectories were chosen to determine the tracking accuracy.
• A square, 10 cm to the side

• A circle, 10 cm in radius

Starting at the home position, the pen was moved to the upper left corner of the $xz$-plane, $(-5.0, 0, -5.0)$ cm, then the square was traced clockwise. See Figure 5.33. The speed of the pen was chosen to be 1.25 cm/s. The result from the test conducted with Modules 3 and 4 only is shown in Figure 5.34. Figure 5.35 shows the configuration of the backbone curve while tracking a square with 2 modules. The performance appears to be very good in the $z$-direction, while there is a 2 mm discrepancy in $x$.

The next test followed the same trajectory as the first test, except that all four modules participated in this test. Figure 5.38 shows the resulting trajectory of the pen. The tracking error increased from 2% to 2.7%. Similar results were observed in the circular trajectory, shown in Figure 5.39.

The tracking error increases when the square is drawn on the $xy$-plane. The results are shown in Figure 5.37. It seems that the position of the tool
Figure 5.34: Tracking Accuracy Analysis: Square Trajectory, $n=2$, $xz$-plane

Figure 5.35: Configuration of Trussarm: Square Trajectory, $n=2$, $xz$-plane
Figure 5.36: Configuration of Trussarm: Square Trajectory, $n=4$, $xy$-plane

Figure 5.37: Tracking Accuracy Analysis: Square Trajectory, $n=4$, $xy$-plane

(pen, $t = (-0.15, 0.28, 0)$ cm) has much more effect on tracking accuracy in the $x$- and $y$- directions than in the $z$-direction. Tracing a square on the $xy$-plane is shown in Figure 5.36.

On average, tracking error in the $z$-direction is smaller than in either $x$ or $y$ directions. The tracking errors are approximately 2% vertically and 7% horizontally. The worst case is when tracking a path on a horizontal surface:
Figure 5.38: Tracking Accuracy Analysis: Square Trajectory, \( n=4 \), \( xx \)-plane

Figure 5.39: Tracking Accuracy Analysis: Circular Trajectory, \( n=4 \), \( zz \)-plane
a tracking error as large as 11% has been observed.

An excerpt of the DMC programs for tracking a square on the \(zz\)-plane is shown below:

```
SH XYZEFG
SP  5427,  6858,  4420,, 2732,  2977,  2556
PA -43424, -34851, -35367,, -21869, -23821, -20456
BG
AM
VT 1000
SP  4635,  4594,  4655,, 2381,  2375,  2385
PA -6338, -18083,  1955,, -2807, -4817, -1369
BG
AM
VT 1000
SP  5080,  5033,  5114,, 2609,  2602,  2614
PA -34309,  22175,  42873,, 18072,  16005,  19550
BG
AM
VT 1000
EN
```

PA is the absolute position command, and SP is the command to set the joint speed. As shown in the above DMC program, both speed and position at each control point along the trajectory was specified to ensure the smooth motion between points. A very simple approximation was utilized to compute the speed of each joint. After solving the inverse kinematics to calculate the desired joint coordinates, the differences between desired coordinates and current coordinates were computed. The speed of each actuator was approximated as follows:

\[
i_t = \frac{(l_{\text{desired}} - l_{\text{current}})}{\Delta T}
\]  

(5.8)
for $i = 1, 2, \ldots, 12$ and $\Delta T$ is the constant time taken between two points on a path. To maintain 1.25 cm/s tip speed, $\Delta T$ was set to be 4 seconds, when 8 control points were used.

Note that the control scheme introduced here assumes infinite acceleration and negative infinite deceleration in the joint motion. In practice, only a constant acceleration and deceleration may be applied. In other words, the trapezoidal speed profile was implemented through the servo (inner) control. In addition to the trapezoidal profile, Salmen (1994) also considered triangular and sinusoidal velocity profiles for velocity control of Trussarm Mark I. However, since the acceleration of Trussarm actuators is sufficiently high (maximum acceleration has been rated at 2.7 cm/s$^2$), the effect of joint acceleration on performance was considered negligible.

5.5.4 On-Line Cartesian Control

It should be noted that, up to this point in this dissertation, all the calculations were performed off-line in the MATLAB environment [Natic, 1992]. The next set of experiments involved Cartesian control with on-line computation. In other words, the same calculations for forward and inverse kinematics were performed in real-time using the Trussarm Control Environment software.

In order to implement real-time control, a few commands were added to the existing DMC commands, listed in Table 5.6. When these commands are issued by the operator, the Trussarm Control Environment software takes the necessary input from both Trussarm and the user input before undertaking a series of actions. For example, when the command wa is issued, the controller first sends TP command (a DMC command to retrieve the current motor counts) and, based on the the response it receives from the DMC boards, it
TW tell the current world coordinates (assuming no end-effector displacement)
WA (desired) world absolute coordinates: x, y, z, n_x, n_y
WR (desired) world relative coordinates: δx, δy, δz
TO tell the current orientation: ϕ(yaw), ψ(pitch), θ(roll)
OA (desired) absolute orientation: ϕ(yaw), ψ(pitch), θ(roll)
OR (desired) relative orientation: δϕ, δψ, δθ
CA (desired) world absolute coordinates without orientation constraints
CR (desired) world relative coordinates without orientation constraints

Table 5.6: DMC Commands for Cartesian Kinematic Control

solves for the forward kinematics to determine the current world coordinates.
Then, it proceeds to add the commanded relative coordinates in order to com-
pute the desired coordinates. Third, the inverse kinematics is solved to find
joint coordinates. The joint limits are also checked at this stage. Finally, the
appropriate transmission ratios are applied to the joint coordinates to compute
the motor coordinates and this new set of motor counts is sent off to DMC
boards using a DMC command, PA.

The experiments discussed in Section 5.5.2 were repeated using Car-tes-
sian control commands and the same results as in Table 5.5 were obtained.

5.5.5 Velocity Tracking Control

The problem addressed in this section is to track the end-effector velocity of
Trussarm. The resolved motion rate control, derived in Chapter 4, is applied.
In some applications of velocity tracking control, the ultimate goal is to achieve
an end-effector position, rather than velocity. In such cases, a position regu-
lator or dead-reckoning needs to be employed in addition to the resolved motion
rate control. The implementation of the dead-reckoning is beyond the scope
of this dissertation, and thus is not considered here.
The same circular trajectory used in the tracking control experiments is used here again. Note that a point on a circle, $x_{ee}^i$, for $0 \leq i \leq n_t$, may be expressed as follows:  
\[
 x_{ee}^i = p_c + rR_{oc} \begin{bmatrix} \cos \eta_i & \sin \eta_i & 0 \end{bmatrix}^T 
\] (5.9)
where $n_t$ is the number of trajectory points on the circle, $0 \leq \eta_i (= \frac{2\pi i}{n_t}) \leq 2\pi$, $p_c$ is the position vector of the center of the circle, $r$ is the radius of the circle, and $R_{oc}$ denotes the orientation of the plane on which the circle lies. For example, the circle used in tracking control on $xy$-plane may be created with $p_c = (0, 0, 1.9345)$, $r = 0.1$ meters, and $R_{oc} = I$. The direction normal $\hat{n}_{ee}$ is $R_{oc} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

The corresponding velocity profile is given as
\[
 \dot{x}_{ee}^i = \frac{d\eta}{dt} r R_{oc} \begin{bmatrix} -\sin \eta_i \\ \cos \eta_i \\ 0 \end{bmatrix} 
\] (5.10)
and $\hat{n}_{ee} = (0, 0, 1)$ where $\frac{d\eta}{dt}$ is the angular speed of the end-effector around the circular trajectory. For example, if it takes 40 seconds to complete the trajectory, $\frac{d\eta}{dt} = \frac{2\pi}{40}$ [rad/sec].

Several simulation tests were performed with various $n_t$ and some examples with $r = 0.4$m, and $p_c = (0, 0, 1.75)$, are shown in Figures 5.40 and 5.41. The solid line in the figures represent the desired trajectory, $\hat{x}_{ee}$, obtained by numerically integrating $\dot{x}_{ee}^i$:
\[
 \hat{x}_{ee}^{i+1} = \hat{x}_{ee}^i + \hat{x}_{ee}^i \delta t, \quad 1 \leq i \leq n_t 
\] (5.11)
for $\hat{x}_{ee}^0 = x^0$. It was assumed that the speed of the end-effector, $\dot{\eta}$, is constant and equal to 1 rad/sec in all simulations reported here.

\[1\] Superscript 'i' denotes the discrete time step, not exponentials.
Figure 5.40: Velocity Tracking Control Simulation Results, $n_t = 20$

Figure 5.41: Velocity Tracking Control Simulation Results, $n_t = 1000$

Figure 5.42: Second norm of the error vs. number of trajectory points
The dotted lines are the result of the simulated resolved motion rate control. The initial configuration, \( q^0 \), was calculated by configuration optimization as discussed in Chapter 2. The joint velocities \( \dot{q}^i \) was calculated as described in Section 4.1, i.e.,

\[
\dot{q}^i = W^{-1}_e(\hat{q}^i)\dot{v}_e^i \quad 0 \leq i \leq n_t
\]

where \( \dot{v}_e \) is defined in eq.(4.11) as

\[
\dot{v}_e = \begin{bmatrix}
\dot{x}_{ee} \\
\dot{E}_{ee} \\
0_{(p-m)\times1}
\end{bmatrix}
\text{ if } m=5
\]

\[
\dot{v}_e = \begin{bmatrix}
\dot{x}_{ee} \\
0_{(p-m)\times1}
\end{bmatrix}
\text{ if } m=3
\]

Then, the approximate end-effector trajectory, \( \hat{x} \), was calculated as

\[
\hat{q}^{i+1} = \hat{q}^i + \dot{q}^i \delta t, \quad \hat{x}_{ee}^{i+1} = w(\hat{q}^{i+1})
\]

Comparing Figure 5.40 and 5.41, it may be noted that, as \( n_t \) increases, the error between \( \hat{x}_{ee} \) and \( \hat{x}_{ee} \) decreases. The simulation tests were performed with \( n_t = 20, 50, 100, 150, 200, \) and \( 1000 \), and the norm of the errors are plotted against \( n_t \) in Figure 5.42. From this figure, it can be concluded that in velocity tracking control, discrepancy in end-effector position is significant and relatively small steps in trajectory would lead to higher accuracy.

The velocity tracking control was implemented on Trussarm using DMC programs. An excerpt from an example is shown below:
In this test, actuator velocity was calculated based on desired end-effector velocity, given by eq.(5.10). The actuator position was approximated as

$$\hat{k} = J_a^{-1} W_e^{-1} \hat{v}_e$$

$$\hat{k}^{i+1} = \hat{k}^i + \hat{k} \delta t$$  \hspace{1cm} (5.16)

where $\hat{k}^0 = k(0)$, solved by inverse kinematics, and

$$J_a = \begin{bmatrix} J_1 & I_3 & I_3 & I_3 \\ I_3 & J_2 & I_3 & I_3 \\ I_3 & I_3 & J_3 & I_3 \\ I_3 & I_3 & I_3 & J_4 \end{bmatrix}$$  \hspace{1cm} (5.17)

$J_j$ is the Jacobian matrix of the $j$th module and $I_3$ is the $3 \times 3$ identity matrix. See eq.(4.11) for definitions of $W_e^{-1}$ and $\hat{v}_e$.

As predicted from the simulation, the test resulted in significant error in end-effector position. An example shown in Figure 5.43 was performed with 4 modules, on the $xz$-plane, $n_t = 50$ and $\dot{\eta} = \frac{\pi}{20}$ rad/sec, or, end-effector velocity of $1.57$ cm/sec. The test was repeated four times and it is shown that position error accumulates under velocity control.
Figure 5.43: Velocity Tracking Control, $n_t = 50$, $xz$-plane
Chapter 6

Control System Design

As demonstrated in Chapter 5, tracking accuracy of Trussarm is poor in some trajectories. In order to improve the performance of kinematic control, a simple, yet effective feedback control strategy is proposed for tracking control.

6.1 Control Design

The proposed control system consists of two layers of control: coarse control and fine positioning control.

The coarse control algorithm is simply to apply the inverse kinematics algorithm, derived in Chapter 3, as it was done in Section 5.5. Because feedback is of joint coordinates and not of end-effector's Cartesian coordinates, this control scheme alone can achieve only coarse positioning.

The fine positioning is based on the global-feedback from Cartesian position sensors. The feedback law is derived by considering the system Jacobian at intermediate configurations along the desired trajectory. After the end-effector is positioned close to its desired position using coarse positioning,
sensors collect feedback information regarding its actual position. The difference between the actual Cartesian coordinates, \( \bar{v} \), and desired coordinates, \( v \), is then calculated:

\[
\delta v = v - \bar{v}
\]  

(6.1)

Recall that the size of \( v \) is equal to the number of kinematic constraints, \( m \). Based on eq.(4.9), \( \delta v_e \) is then defined.

\[
\delta v_e = \begin{bmatrix}
\delta v \\
0_{(p-m) \times 1}
\end{bmatrix}
\]  

(6.2)

We pre-multiply \( \delta v_e \) by the system Jacobian matrix to yield the change in joint coordinates, \( \delta k \):

\[
\delta k = J_a^{-1}W_e^{-1}\delta v_e
\]  

(6.3)

Note that the system Jacobian matrix \( J_a^{-1} \) and \( W_e^{-1} \) have been developed in Chapter 4. Finally, the joint coordinates are adjusted accordingly:

\[
k_{\text{mod}} = k + \delta k
\]  

(6.4)

The performance may be improved even further by applying the feedback control iteratively. A block diagram for this augmented control scheme is shown in Figure 6.1(i).

### 6.2 Simulation Test

The simulation software was modified to implement the proposed control method. First, a linear error model was added to the forward kinematics to generate the Cartesian coordinates that are similar to the actual coordinates. This algorithm is referred to as a Trussarm model hereinafter. The
Figure 6.1: Augmented Kinematic Control Algorithm
Trussarm model was based on the series of data collected from the previous experiments.

The same trajectories (circular and square paths) used in accuracy analysis were used for closed-loop control again. A solid line in Figure 6.3 and 6.4 represents the end-effector trajectory, \( \bar{v} \) when only the coarse positioning is applied. A dotted line is the desired trajectory, \( v \), and the dashed line is the trajectory when closed-loop control is applied. The improvement in the tracking accuracy is significant. For example, the maximum tracking error in square trajectory is 1.3 mm for coarse control only. When fine control is applied additionally, tracking error is reduced to 0.2 mm.

In order to represent the tracking performance more quantitatively, the discrepancy in end-effector position, \( \delta v \), was calculated throughout the trajectory. Then, the second norm of this value for the entire trajectory was computed. The result is shown for square trajectory in Figure 6.2. The second norm of tracking error is plotted against the number of times fine positioning is applied in Figure 6.5.
Figure 6.3: Simulation Results from Closed-Loop Control, Square Trajectory, \(n=2\), \(xz\)-plane

Figure 6.4: Simulation Results from Closed-Loop Control, Circular Trajectory, \(n=2\), \(xz\)-plane
Figure 6.5: Tracking Error vs. Number of Iteration around Fine Positioning Control

6.3 Implementation

At the time of writing, a global sensor was not available. Thus, the proposed method was modified to calculate the actual Trussarm end-effector position, \( \vec{v} \), using the Trussarm model, instead of measuring with sensors. Figure 6.1(ii) shows the architecture for the control method. Figure 6.1(iii) shows how the fine positioning control can be applied repeatedly in order to improve the performance.

Due to the structural failure on module 2, only modules 3 and 4 were used for the closed-loop control tests. Since a 10 cm circular trajectory cannot be achieved with two modules, the circular trajectory was reduced in its size to the radius of \( r=5 \) cm. Figures 6.6 and 6.7 show the sample results from the experiments. Both trajectories were on \( \text{zz-plane} \), drawn with modules 3 and 4 at constant tip speed of 1.25 cm/s.
Figure 6.6: Results from Closed-Loop Control, Square Trajectory, \( n=2 \), \( xz \)-plane

Figure 6.7: Results from Closed-Loop Control, Circular Trajectory, \( n=2 \), \( xz \)-plane
Chapter 7

Conclusion

7.1 Final Remarks

In this work, a number of issues regarding the control characteristics of Trussarm — a 12-DOF VGT manipulator — were examined.

First, the kinematics was derived by adopting a discrete model approach. The geometry of each module was represented by a 3-DOF gimbal. A gimbal consists of one universal revolute joint, which provide two rotational degrees of freedom, $\theta$ and $\phi$, and a coupled prismatic joint, $l$. The macroscopic geometry is then represented by a series of 3-DOF gimbal modules. Redundancy was resolved using a configuration optimization technique and a procedure for solving kinematics was described. The forward and inverse kinematics of a module were also revisited. A similar approach was taken to derive a kinematics model at the velocity level.

A benefit of the proposed method is that it models a complex VGT manipulator as kinematically equivalent to, yet much simpler than, a serial manipulator with a simple mathematical presentation. The kinematics of the
equivalent gimbal mechanism is physically meaningful and can be easily interpreted. For example, a $\theta$ of zero degrees (rotation about $x$-axis) implies that the end-effector position is on the $xz$-plane. The positive and negative $\phi$ (rotation about the $y$-axis) causes an end-effector motion in either the negative or positive $x$-direction. On the other hand, the relation between the motion of the end-effector and the lengths of the actuators is nonlinear and can not be described simply.

This method can be applied to any VGT manipulator whose module can be modeled as a 3 DOF gimbal. If a VGT module has more than 3-DOF or is not symmetric about the mid-plane, its geometry can not be represented by a gimbal module with a coupled prismatic joint. However, a similar technique may be applied using a gimbal module with more degrees of freedom, or even a different configuration.

When compared to other methods, the discrete model approach shares many advantages of configuration control techniques. The method is well suited for parallel processing techniques. The model can be easily extended in order to handle joint limit avoidance; and, the model can be also used for other types of VGT manipulators.

In Chapter 5, the characteristics of Trussarm were examined. The workspace of Trussarm was represented using extension method and sweeping method. A new concept of controllable workspace, a subset of points in the reachable workspace that can be reached by the inverse kinematics algorithm, was introduced. Using the proposed kinematics model, the controllable workspace encompasses almost 70% of the reachable workspace of Trussarm. Based on the information on the Trussarm workspace, a Reach Hierarchy Algorithm was applied to maximize the controllable workspace. With this algorithm, over 90% of the reachable workspace was controllable.
The dexterity of Trussarm was examined using four methods. Tests of repeatability indicated a 1 mm variability. The speed analysis showed that Trussarm's maximum tip speed is 5 and 20 cm/s vertically and horizontally. Furthermore, Trussarm is capable of performing a standard 12-inch pick-and-place maneuver in less than 7 seconds horizontally, of less than 20 seconds for a vertical maneuver.

Positioning errors of about 6% and tracking errors of about 2% to 7% were observed when kinematic control was applied. In order to improve the tracking errors, a two-stage control method was applied.

The present documentation provides a framework for future research on the design and control of VGT manipulators for practical applications. The experiments described in this dissertation provide means to measure qualities of VGT manipulators in order to compare them with the conventional industrial manipulators. Even though the current Trussarm is a laboratory model, designed for research purposes only, the characteristics of Trussarm found in this documentation show the promising future in the areas of manufacturing and automation. The results obtained from this work may be used to explore the industrial applications of Trussarm and other VGT manipulators.

7.2 Contributions

1. The kinematics of Trussarm were derived by adopting a discrete model approach.

2. Analytical solutions to Trussarm differential kinematics was derived.

3. A joint-limit avoidance algorithm was implemented.

4. The workspace of Trussarm was modeled.
5. The controllable workspace of Trussarm was defined.

6. An algorithm to maximize the workspace was applied.

7. The dexterity of Trussarm was computed.

8. The tip speed of Trussarm was measured.

9. The positioning and tracking accuracy of Trussarm under the kinematic control were determined.

10. A closed-loop control algorithm to improve the tracking accuracy was designed and the performance was verified experimentally.

7.3 Future Work

While this dissertation provides grounds for industrial implementation for Trussarm technology, there are still many areas of research that may be pursued for future improvement in Trussarm performance. A few suggestions are made here.

Firstly, as demonstrated in this dissertation, the prototype Trussarm requires much improvement in mechanical design to minimize the 'slop' in its joints.

Secondly, in Chapter 5 the dexterity of Trussarm was considered in a local sense. Park and Blockett (1994) also examined the mathematical representation of the dexterity of a manipulator in global sense. The global dexterity, or overall dexterity, can be used as a design criterion. However, to represent the overall dexterity mathematically is a challenging problem since the measurement should not depend on the choice of the coordinates, or the
size of the manipulator. In addition, a mathematical description of the manipulator's workspace should be available. In future, the global dexterity of Trussarm may be examined in order to improve the design of the manipulator.

Thirdly, the results of the experiments suggest that feedback control would improve the accuracy of Trussarm. Further investigation in the selection of position sensors is recommended.

Lastly, in order to achieve a higher level of performance in robotic manipulation, it is essential to consider manipulator dynamics. This is especially true in applications that involve precision force interaction with the environment, or that involve high-speed motion. Trussarm dynamics have been considered by the author [Lee and Zanganeh, 1997]. The next step would validate and improve the proposed Trussarm dynamics model using the actual joint and force sensor data from Trussarm.
References


Park, F.C. and Brockett, R.W. "Kinematic Dexterity of Robotic Mechanisms," 


Appendix A

An Example of a Constrained Optimization Problem

Before we discuss an example of the constrained optimization problem, let us consider the nature of the inequality constraints. Those which alter the minimum are *active* inequality constraints, while the others are *inactive*.

For a system of order \( r \), with \( m \) inequality constraints, \( A_i(x) \leq b_i \), the solution of the minimization problem satisfies the Kuhn-Tucker optimality condition:

\[
\frac{\partial Z}{\partial x_j} + y_1 \frac{\partial A_1}{\partial x_j} + \cdots + y_m \frac{\partial A_m}{\partial x_j} = 0 \tag{A.1}
\]

for \( j = 1, \ldots, r \), with \( y \) and \( z \) also subject to

\[
y_i \geq 0, \quad A_i(x) \leq b_i \quad y_i(A_i(x) - b_i) = 0 \tag{A.2}
\]

for \( i = 1, \ldots, m \). Equations (A.1) and (A.2) consist of \( r + m \) equations. In eq. (A.2), however, we cannot predict which equations are right. If the constraint is active, \( A_i(x) \leq b_i \), otherwise, \( y_i \geq 0 \). To demonstrate the complexity of the
problem, let us examine a system with two variables, \( x_1 \) and \( x_2 \), and with two constraints, \( x_1 \leq b_1 \) and \( x_2 \leq b_2 \). The objective function is

\[
Z = x_1^2 + x_2^2
\]  

(A.3)

Let us also define Lagrangian multipliers \( y_1 \) and \( y_2 \). Then, the problem is to find \( x_1 \) and \( x_2 \) such that

\[
\frac{\partial L}{\partial x} = 0 \quad \text{(A.4)}
\]

\[
\frac{\partial L}{\partial y} = 0 \quad \text{(A.5)}
\]

where

\[
L = Z + y^T(x - b)
\]  

(A.6)

There are four distinct cases: (1) \( b_1 > 0, b_2 > 0 \), (2) \( b_1 > 0, b_2 < 0 \), (3) \( b_1 < 0, b_2 > 0 \), and (4) \( b_1 < 0, b_2 < 0 \).

Case 1

Both inequality constraints are inactive. Therefore, \( y_1 = 0 \) and \( y_2 = 0 \).

\[
\frac{\partial L}{\partial x_1} = 2x_1 = 0 \quad \text{(A.7)}
\]

\[
\frac{\partial L}{\partial x_2} = 2x_2 = 0 \quad \text{(A.8)}
\]

Therefore, \( x_1 = 0, x_2 = 0 \), and the optimal value for \( Z \) is \( Z^* = 0 \).

Case 2

The first inequality constraint is inactive; the second is active. Therefore, \( y_1 = 0 \) and \( y_2 > 0 \). Moreover,

\[
\frac{\partial L}{\partial x_1} = 2x_1 = 0 \quad \text{(A.9)}
\]

\[
\frac{\partial L}{\partial x_2} = 2x_2 + y_2 = 0 \quad \text{(A.10)}
\]

\[
\frac{\partial L}{\partial y_2} = x_2 - b_2 \quad \text{(A.11)}
\]
Thus, \( x_1 = 0, \ x_2 = b_2, \ y_2 = -2b_2 > 0, \) and \( Z^* = b_2^2. \)

**Case 3**

The first inequality constraint is active; the second is inactive. Therefore, \( y_1 > 0 \) and \( y_2 = 0. \) In this case, we have

\[
\begin{align*}
\frac{\partial L}{\partial x_1} & = 2x_1 + y_1 = 0 \\
\frac{\partial L}{\partial x_2} & = 2x_2 = 0 \\
\frac{\partial L}{\partial y_1} & = x_1 - b_1 \\
\end{align*}
\]

Hence, \( x_1 = b_1, \ x_2 = 0, \ y_1 = -2b_1 > 0, \) and \( Z^* = b_1^2. \)

**Case 4**

Both constraints are active. Therefore, \( y_1 > 0, y_2 > 0 \) and

\[
\begin{align*}
\frac{\partial L}{\partial x_1} & = 2x_1 + y_1 = 0 \\
\frac{\partial L}{\partial x_2} & = 2x_2 + y_2 = 0 \\
\frac{\partial L}{\partial y_1} & = x_1 - b_1 \\
\frac{\partial L}{\partial y_1} & = x_2 - b_2
\end{align*}
\]

Thus, \( x_1 = b_1, \ x_2 = b_2, \ y_1 = -2b_1 > 0, \ y_2 = -2b_2 > 0 \) and \( Z^* = b_1^2 + b_2^2. \)

In general, for a system with \( m \) inequality constraints, \( 2^m \) combinations are required to be examined. A good algorithm finds the right combination without trying them all. For large \( m, \) the problem is still difficult to compute in real-time no matter how "good" the algorithm is. The joint-limit avoidance problem presented in Section 3.3 incorporates 12 inequality constraints, yielding \( 2^{12} = 4096 \) combinations to consider!